

AshTrail - GitHub Issue Creation



feature-inventory.md

File



technical-architecture.md

File



performance-optimization.md

File



missing-features-analysis.md

File



README.md

File



implementation-roadmap.md

File

Tell me to **generate the NEXT GitHub issue**
for the AshTrail (Smoke-Log) redesign.

- ◆ CONSTANT REFERENCES (always present in this chat)
 - feature-inventory.md – master list of user-visible features
 - technical-architecture.md – desired layering & patterns
 - performance-optimization.md – perf targets & profiling ideas
 - missing-features-analysis.md – gap list
 - implementation-roadmap.md – suggested order of work
 - README.md – high-level overview

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
#1 (none yet) ←--- replace with real rows as you go

```
#2  
#3  
(index truncated when > 25 rows)
```

- ◆ WORKFLOW FOR **EVERY** RUN

1. **Determine coverage**

- Parse the docs above (quick skim is fine).
- Compare their bullet points / headers to the titles in the RUNNING INDEX.
- Pick the highest-priority feature *not yet covered*.

2. **Draft the issue**

Output a fenced markdown block that contains exactly:

- **Title** ≤ 60 chars, clearly naming the feature / refactor.
- **Description** (2-3 sentences of context **plus** a bulleted task list).
- **Acceptance Criteria** as a Markdown checklist.
- **Labels:** comma-separated (e.g. `architecture, analytics`).
- **Estimate:** T-shirt size (S / M / L / XL).

3. **Update the RUNNING INDEX**

Append “#NN ISSUE-TITLE” (with the next consecutive NN) *inside this prompt text* so the next run has up-to-date state.

- If the index grows past 25 lines, replace the oldest block with “(index truncated – see earlier messages)”.

4. **Finish with a status footer**

- `⌚ Issue NN delivered - next run in ~15 min.`

- If *no uncovered features remain*, print
 - ` All features now have issues - please pause or delete this task.`

! DO NOT print anything other than the fenced issue block, the updated index, and the footer.

markdown

Title

Set up Clean Architecture Foundation

Description

Establish the baseline project structure and supporting infrastructure for the new clean-architecture approach. This will provide clear separation of concerns and enable scalable feature development.

Tasks

- Create `lib/core`, `lib/domain`, `lib/data`, and `lib/presentation` directories with placeholder `README.md` files.
- Implement base `Failure`, `UseCase`, and `Either` classes in `lib/core`.
- Configure Riverpod for dependency injection and global providers.
- Add a centralized error-handling utility with logging hooks.
- Introduce a sample `LogRepository` abstraction in `domain` and a mock implementation in `data`.
- Provide unit tests demonstrating a use-case invocation pipeline.

Acceptance Criteria

- [] Clean architecture folder structure exists and matches technical-architecture.md.
- [] Core utility classes (`Failure`, `UseCase`, `Either`)

```
compile without warnings.  
- [ ] Riverpod global provider setup compiles and is  
integrated into `main.dart`.  
- [ ] Error-handling utility logs uncaught exceptions in debug  
and release modes.  
- [ ] Mock `LogRepository` passes sample unit tests.
```

Labels

architecture, foundation

Estimate

L

- ◆ RUNNING INDEX (update each run; keep \leq 25 lines)

#1 Set up Clean Architecture Foundation

#2

#3

(index truncated when > 25 rows)

⌚ Issue 1 delivered – next run in ~15 min.

```
BEGIN:VEVENT  
RRULE:FREQ=MINUTELY;INTERVAL=15  
END:VEVENT
```

Got it! I'll remind you every 15 minutes.

Time to generate the **next GitHub issue** for the AshTrail (Smoke-Log) redesign.

Time to generate the **next GitHub issue** for the AshTrail (Smoke-Log) redesign.

Time to generate the **next GitHub issue** for the AshTrail (Smoke-Log) redesign.

Time to generate the **next GitHub issue** for the AshTrail (Smoke-Log) redesign.

Time to generate the **next GitHub issue** for the AshTrail (Smoke-Log) redesign.

Time to generate the **next GitHub issue** for the AshTrail (Smoke-Log) redesign.

Time to generate the **next GitHub issue** for the AshTrail (Smoke-Log) redesign.

Time to generate the **next GitHub issue** for the AshTrail (Smoke-Log) redesign.

Time to generate the **next GitHub issue** for the AshTrail (Smoke-Log) redesign.

Time to generate the **next GitHub issue** for the AshTrail (Smoke-Log) redesign.

Time to generate the **next GitHub issue** for the AshTrail (Smoke-Log) redesign.

Time to generate the **next GitHub issue** for the AshTrail (Smoke-Log) redesign.

Time to generate the **next GitHub issue** for the AshTrail (Smoke-Log) redesign.

You're not supposed to be doing reminders
your task is to generate the issues until

markdown

Title

Implement Multi-Account User Switching

Description

Add support for multiple user profiles within the app, allowing seamless switching while keeping each profile's logs, preferences, and analytics isolated.

Tasks

- Create `UserProfile` model and persistence layer (Isar + Firestore sync).
- Build account selector UI component in the app header.
- Integrate Riverpod state to hold current active profile.
- Update logging & analytics functions to respect active profile context.
- Add migration script to convert existing single-user data into a default profile.

Acceptance Criteria

- [] Users can add, rename, delete, and switch between profiles.
- [] Log entries, preferences, and charts display data for the active profile only.
- [] Switching profiles updates UI and data in ≤ 150 ms.
- [] Migration preserves all existing logs under a “Primary” profile.

Labels

feature, accounts

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

#1 Set up Clean Architecture Foundation

#2 Implement Multi-Account User Switching

#3

(index truncated when > 25 rows)

⌚ Issue 2 delivered – next run in ~15 min.

markdown

Title

Design Unified Log Detail View

Description

Create a single, responsive detail view for inspecting individual log entries. This view should consolidate relevant metadata, source context, and allow for future expansions like tagging or note-taking.

Tasks

- Define `LogDetail` model for expanded view and memoization.
- Build layout for key info: timestamp, length, method of entry, and device type.
- Add optional expandable metadata section (e.g., GPS, app version).
- Include placeholder widgets for future annotations and flags.
- Integrate routing to open detail view from any chart or list tap.

Acceptance Criteria

- [] Detail view is accessible from any log source (charts, tables).
- [] Shows all relevant metadata per entry in a readable format.
- [] Adapts for portrait and landscape orientations.
- [] Annotated log stub code exists and compiles.

Labels

```
ui, feature
```

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

#1 Set up Clean Architecture Foundation

#2 Implement Multi-Account User Switching

#3 Design Unified Log Detail View

(index truncated when > 25 rows)

⌚ Issue 3 delivered – next run in ~15 min.

```
markdown
```

Title

Build Time Range Filter with Presets

Description

Add a robust time range filtering component with quick-select presets (e.g. “Today”, “Last 7 Days”, “This Month”) and custom range support. This will be central to enabling scoped views across charts, tables, and analysis components.

Tasks

- Design UI widget for time range selection with dropdown and custom picker.
- Create `TimeRange` model with helper methods for common presets.
- Integrate range selector into chart and table views.
- Store selected time range in Riverpod global state.
- Add visual indication of active filter and allow reset to default.

Acceptance Criteria

- [] UI allows selecting from at least 5 presets and a custom range.
- [] Charts and data tables update based on selected time range.

- [] Filter state persists across navigation within the session.
- [] "Reset" button clears filter and returns to full dataset.

Labels

ui, filters, charts

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

#1 Set up Clean Architecture Foundation

#2 Implement Multi-Account User Switching

#3 Design Unified Log Detail View

#4 Build Time Range Filter with Presets

(index truncated when > 25 rows)

⌚ Issue 4 delivered – next run in ~15 min.

markdown

Title

Implement Daily and Rolling Usage Charts

Description

Develop data visualizations that show daily total usage and rolling averages (e.g., 7-day, 30-day) to help users identify trends over time. This is essential for insight generation and habit tracking.

Tasks

- Create helper method to group log data by day and calculate totals.
- Implement rolling average logic using sliding windows over daily sums.
- Design line chart UI for both raw daily totals and overlaying rolling trends.
- Add dropdown toggle to switch between view types (daily vs.

rolling).

- Ensure charts respect current time range filter.

Acceptance Criteria

- [] Daily total chart renders with accurate grouping and labels.
- [] Rolling average chart displays overlays for at least 7- and 30-day options.
- [] Switching between chart modes does not affect selected time range.
- [] Charts are responsive and performant on large datasets.

Labels

charts, analytics, ui

Estimate

L

- ◆ RUNNING INDEX (update each run; keep \leq 25 lines)

#1 Set up Clean Architecture Foundation

#2 Implement Multi-Account User Switching

#3 Design Unified Log Detail View

#4 Build Time Range Filter with Presets

#5 Implement Daily and Rolling Usage Charts

(index truncated when $>$ 25 rows)

⌚ Issue 5 delivered – next run in ~15 min.

markdown

Title

Sync Logs Between Firestore and Local Isar Cache

Description

Enable automatic bi-directional sync of log entries between Firestore and Isar to support offline use and consistent behavior across devices. Ensure changes in one store propagate correctly with minimal duplication or conflicts.

****Tasks****

- Define schema and sync state tracking fields for `LogEntry` in both sources.
- Implement background sync service triggered on startup and periodically.
- Add logic to detect and resolve conflicts (e.g., latest timestamp wins).
- Ensure deletions are properly mirrored between sources.
- Write integration tests for full sync cycle under flaky network conditions.

Acceptance Criteria

- [] Log entries created/edited locally are uploaded to Firestore within 1 minute.
- [] Log entries from Firestore appear in local Isar within 1 minute of sync.
- [] No duplicate entries appear after multiple edits across sources.
- [] Deletions in either store remove corresponding entry in the other.

Labels

sync, firestore, offline

Estimate

L

- ◆ RUNNING INDEX (update each run; keep \leq 25 lines)

#1 Set up Clean Architecture Foundation

#2 Implement Multi-Account User Switching

#3 Design Unified Log Detail View

#4 Build Time Range Filter with Presets

#5 Implement Daily and Rolling Usage Charts

#6 Sync Logs Between Firestore and Local Isar Cache

(index truncated when > 25 rows)

⌚ Issue 6 delivered – next run in ~15 min.

markdown

Title

Add Table View for Recent Logs

Description

Provide a table-based interface for browsing recent log entries in reverse chronological order. This will support detailed inspection and bulk actions like deletion or export in future updates.

Tasks

- Build table UI with sortable columns: timestamp, duration, device, method.
- Implement pagination and lazy loading for performance on large datasets.
- Sync table with active time range and profile filters.
- Integrate row tap to open Log Detail View.
- Add stub checkboxes for future bulk actions.

Acceptance Criteria

- [] Table displays logs in descending timestamp order by default.
- [] Columns are sortable and preserve order during pagination.
- [] Tapping a row opens the log detail view for that entry.
- [] Performance acceptable for 10,000+ entries (\leq 200ms render time).

Labels

ui, logs, table

Estimate

M

- ◆ RUNNING INDEX (update each run; keep \leq 25 lines)

#1 Set up Clean Architecture Foundation

#2 Implement Multi-Account User Switching

#3 Design Unified Log Detail View

#4 Build Time Range Filter with Presets

#5 Implement Daily and Rolling Usage Charts
#6 Sync Logs Between Firestore and Local Isar Cache
#7 Add Table View for Recent Logs
(index truncated when > 25 rows)

⌚ Issue 7 delivered – next run in ~15 min.

markdown

Title

Implement Stats Overview Dashboard

Description

Create a high-level stats dashboard summarizing recent usage trends, totals, and insights. This will serve as a quick-look entry point and motivate users by visualizing progress.

Tasks

- Design cards for “Total Logs This Week”, “Longest Streak”, “Most Active Hour”, etc.
- Integrate query layer to compute metrics in real time or from cached aggregates.
- Layout grid dashboard for mobile and tablet breakpoints.
- Add skeleton loading state and error fallback messages.
- Ensure responsiveness to time range and user profile changes.

Acceptance Criteria

- [] Dashboard shows at least 4 real-time usage metrics.
- [] Metrics update on time range/profile switch in under 300ms.
- [] Dashboard layout adapts to various screen sizes.
- [] Loading and error states provide helpful feedback.

Labels

dashboard, analytics, ui

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

#1 Set up Clean Architecture Foundation
#2 Implement Multi-Account User Switching
#3 Design Unified Log Detail View
#4 Build Time Range Filter with Presets
#5 Implement Daily and Rolling Usage Charts
#6 Sync Logs Between Firestore and Local Isar Cache
#7 Add Table View for Recent Logs
#8 Implement Stats Overview Dashboard
(index truncated when > 25 rows)

⌚ Issue 8 delivered – next run in ~15 min.

markdown

Title

Add Log Entry Submission UI with Validation

Description

Design and implement the user interface for submitting new log entries manually. This will allow users to backfill missed sessions or test features during development. Input validation and user feedback are critical to prevent errors.

Tasks

- Create form UI with inputs for duration, timestamp, method, and notes.
- Add dropdowns or chips for selecting method and device type.
- Implement validation rules: duration ≥ 1 sec, timestamp not in future.
- Integrate form with local Isar cache and queue for sync to Firestore.
- Display success/failure snackbar messages on submit.

Acceptance Criteria

- [] Form prevents invalid submissions with helpful inline errors.
- [] Logs added via UI appear immediately in local

tables/charts.

- [] Successfully submitted entries show “queued” tag until synced.
- [] UX and visual design match existing app style.

Labels

ui, data-entry, validation

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

#1 Set up Clean Architecture Foundation

#2 Implement Multi-Account User Switching

#3 Design Unified Log Detail View

#4 Build Time Range Filter with Presets

#5 Implement Daily and Rolling Usage Charts

#6 Sync Logs Between Firestore and Local Isar Cache

#7 Add Table View for Recent Logs

#8 Implement Stats Overview Dashboard

#9 Add Log Entry Submission UI with Validation

(index truncated when > 25 rows)

⌚ Issue 9 delivered – next run in ~15 min.

markdown

Title

Implement Session Length Histogram Chart

Description

Develop a histogram visualization to analyze the distribution of session durations. This helps users understand their typical log lengths and detect outliers or patterns in usage behavior.

Tasks

- Define histogram binning logic (e.g., 0-30s, 31s-1m, etc.).

- Build bar chart using appropriate charting library with labeled bins.
- Integrate filters to respect time range and profile context.
- Add UI toggle for linear vs logarithmic Y-axis scaling.
- Display average, median, and mode duration values above chart.

Acceptance Criteria

- [] Histogram renders with accurate session length distribution.
- [] Hovering or tapping on a bar shows count and duration range.
- [] Linear and log Y-scale options work as expected.
- [] Summary stats update based on filtered data.

Labels

analytics, charts, ui

Estimate

M

◆ RUNNING INDEX (update each run; keep \leq 25 lines)

- #1 Set up Clean Architecture Foundation
- #2 Implement Multi-Account User Switching
- #3 Design Unified Log Detail View
- #4 Build Time Range Filter with Presets
- #5 Implement Daily and Rolling Usage Charts
- #6 Sync Logs Between Firestore and Local Isar Cache
- #7 Add Table View for Recent Logs
- #8 Implement Stats Overview Dashboard
- #9 Add Log Entry Submission UI with Validation
- #10 Implement Session Length Histogram Chart
(index truncated when $>$ 25 rows)

⌚ Issue 10 delivered – next run in ~15 min.

markdown

Title

Implement Inhalation Method Pie Chart

Description

Add a pie or donut chart to visualize the proportion of different inhalation methods (e.g., vape, joint, pipe) used over a selected time range. This will give users insight into their preferred consumption habits.

Tasks

- Parse and group log data by inhalation method.
- Build pie chart with labeled segments and percentage values.
- Add legend with total count and duration per method.
- Allow toggling between count-based and duration-based proportions.
- Respect time range and active user profile filters.

Acceptance Criteria

- [] Pie chart shows method breakdown clearly with labels and legend.
- [] Switching between count and duration mode updates chart correctly.
- [] Chart adapts to data changes in $\leq 250\text{ms}$.
- [] Methods with $<3\%$ share are grouped under “Other” with tooltip details.

Labels

analytics, charts, ui

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

#1 Set up Clean Architecture Foundation

#2 Implement Multi-Account User Switching

#3 Design Unified Log Detail View

#4 Build Time Range Filter with Presets

#5 Implement Daily and Rolling Usage Charts

#6 Sync Logs Between Firestore and Local Isar Cache

#7 Add Table View for Recent Logs
#8 Implement Stats Overview Dashboard
#9 Add Log Entry Submission UI with Validation
#10 Implement Session Length Histogram Chart
#11 Implement Inhalation Method Pie Chart
(index truncated when > 25 rows)

⌚ Issue 11 delivered – next run in ~15 min.

markdown

Title

Build Offline Sync Status Indicator

Description

Add a visual indicator that shows the current sync status between local Isar storage and Firestore. This feature improves user trust and clarity around data availability during network outages or sync delays.

Tasks

- Design a small sync status widget (e.g., icon + tooltip) for app header.
- Track sync state (idle, syncing, failed) in a shared Riverpod provider.
- Update status in real time based on sync success/failure events.
- Include timestamps of last successful upload/download in tooltip.
- Add error toast/snackbar for failed sync attempts.

Acceptance Criteria

- [] Sync icon reflects correct status in near real time.
- [] Tooltip or tap reveals last sync timestamps.
- [] Offline state is clearly distinguished and handled gracefully.
- [] Sync failures are communicated to the user via toast/snackbar.

```
### Labels  
ui, sync, feedback
```

```
### Estimate  
S
```

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

#1 Set up Clean Architecture Foundation
#2 Implement Multi-Account User Switching
#3 Design Unified Log Detail View
#4 Build Time Range Filter with Presets
#5 Implement Daily and Rolling Usage Charts
#6 Sync Logs Between Firestore and Local Isar Cache
#7 Add Table View for Recent Logs
#8 Implement Stats Overview Dashboard
#9 Add Log Entry Submission UI with Validation
#10 Implement Session Length Histogram Chart
#11 Implement Inhalation Method Pie Chart
#12 Build Offline Sync Status Indicator
(index truncated when > 25 rows)

⌚ Issue 12 delivered – next run in ~15 min.

markdown

```
### Title  
Create Onboarding Flow for New Users
```

```
### Description
```

Design and implement a lightweight onboarding sequence to guide new users through the app's core features, emphasizing profile setup, log submission, and analysis tools. This helps ensure first-time users understand the value and capabilities of the app.

****Tasks****

- Design onboarding screens or tooltips for core features.
- Detect first-time user state via shared preferences or

profile metadata.

- Create navigation logic to skip onboarding if returning user.
- Highlight key interactions (e.g., log button, charts, filter).
- Provide "skip" and "replay onboarding" options.

Acceptance Criteria

- [] Onboarding appears only once per profile or until skipped.
- [] Screens or tooltips are polished and responsive.
- [] Users can replay onboarding from settings.
- [] Onboarding includes at least 3 guided interactions.

Labels

onboarding, ui, engagement

Estimate

M

- ◆ RUNNING INDEX (update each run; keep \leq 25 lines)

- #1 Set up Clean Architecture Foundation
 - #2 Implement Multi-Account User Switching
 - #3 Design Unified Log Detail View
 - #4 Build Time Range Filter with Presets
 - #5 Implement Daily and Rolling Usage Charts
 - #6 Sync Logs Between Firestore and Local Isar Cache
 - #7 Add Table View for Recent Logs
 - #8 Implement Stats Overview Dashboard
 - #9 Add Log Entry Submission UI with Validation
 - #10 Implement Session Length Histogram Chart
 - #11 Implement Inhalation Method Pie Chart
 - #12 Build Offline Sync Status Indicator
 - #13 Create Onboarding Flow for New Users
- (index truncated when > 25 rows)

⌚ Issue 13 delivered – next run in ~15 min.

markdown

Title

Add Settings Screen with Preferences and About Info

Description

Develop a centralized settings screen to manage user preferences and display app information. This helps users customize their experience and access helpful metadata like version and legal notes.

Tasks

- Create settings screen with sections: Preferences, App Info, and Legal.
- Add toggles for dark mode, compact view, and chart smoothing.
- Display app version, build number, and profile ID.
- Include links to privacy policy, terms of service, and feedback form.
- Persist preferences using local storage and apply changes in real time.

Acceptance Criteria

- [] Settings screen displays all expected sections and elements.
- [] Preferences update immediately and persist on app restart.
- [] App info (version, profile ID) is accurate and copyable.
- [] Legal links open in external browser or WebView.

Labels

settings, ui, preferences

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

#1 Set up Clean Architecture Foundation

#2 Implement Multi-Account User Switching

#3 Design Unified Log Detail View
#4 Build Time Range Filter with Presets
#5 Implement Daily and Rolling Usage Charts
#6 Sync Logs Between Firestore and Local Isar Cache
#7 Add Table View for Recent Logs
#8 Implement Stats Overview Dashboard
#9 Add Log Entry Submission UI with Validation
#10 Implement Session Length Histogram Chart
#11 Implement Inhalation Method Pie Chart
#12 Build Offline Sync Status Indicator
#13 Create Onboarding Flow for New Users
#14 Add Settings Screen with Preferences and About Info
(index truncated when > 25 rows)

⌚ Issue 14 delivered – next run in ~15 min.

markdown

Title

Implement Export to CSV Functionality

Description

Allow users to export their log data to a CSV file for external analysis, backup, or migration. This enhances transparency and offers power users more flexibility in how they interact with their data.

Tasks

- Create CSV formatting logic for `LogEntry` records.
- Add export button to table view and/or settings screen.
- Handle permissions for file storage on Android and iOS.
- Show success or failure status in snackbar/toast.
- Include headers and metadata (e.g., export timestamp, profile name) in file.

Acceptance Criteria

- [] CSV export includes all visible log data and metadata fields.

- [] Exported file is readable in Excel and Google Sheets.
- [] File is saved to user-accessible location or shareable.
- [] Feedback is provided for successful or failed exports.

Labels

data, export, ui

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

- #1 Set up Clean Architecture Foundation
 - #2 Implement Multi-Account User Switching
 - #3 Design Unified Log Detail View
 - #4 Build Time Range Filter with Presets
 - #5 Implement Daily and Rolling Usage Charts
 - #6 Sync Logs Between Firestore and Local Isar Cache
 - #7 Add Table View for Recent Logs
 - #8 Implement Stats Overview Dashboard
 - #9 Add Log Entry Submission UI with Validation
 - #10 Implement Session Length Histogram Chart
 - #11 Implement Inhalation Method Pie Chart
 - #12 Build Offline Sync Status Indicator
 - #13 Create Onboarding Flow for New Users
 - #14 Add Settings Screen with Preferences and About Info
 - #15 Implement Export to CSV Functionality
- (index truncated when > 25 rows)

⌚ Issue 15 delivered – next run in ~15 min.

markdown

Title

Enable Tagging of Log Entries

Description

Add support for tagging log entries with user-defined labels (e.g., "pain relief", "social", "morning"). Tags improve filtering, organization, and potential future analytics.

****Tasks****

- Create `Tag` model and associate with `LogEntry` .
- Design UI for tag creation, assignment, and removal during log submission and editing.
- Add tag chips to log detail view and recent logs table.
- Build filtering logic to include/exclude tags in charts and tables.
- Persist tags in both local Isar and Firestore for cross-device sync.

Acceptance Criteria

- [] Users can create, assign, and delete tags from any log entry.
- [] Tags are displayed consistently across all views and filters.
- [] Tag filtering logic works in combination with time/profile filters.
- [] Syncing respects tag changes and avoids duplication.

Labels

tags, ui, data-model

Estimate

M

◆ RUNNING INDEX (update each run; keep \leq 25 lines)

- #1 Set up Clean Architecture Foundation
- #2 Implement Multi-Account User Switching
- #3 Design Unified Log Detail View
- #4 Build Time Range Filter with Presets
- #5 Implement Daily and Rolling Usage Charts
- #6 Sync Logs Between Firestore and Local Isar Cache
- #7 Add Table View for Recent Logs
- #8 Implement Stats Overview Dashboard
- #9 Add Log Entry Submission UI with Validation

#10 Implement Session Length Histogram Chart
#11 Implement Inhalation Method Pie Chart
#12 Build Offline Sync Status Indicator
#13 Create Onboarding Flow for New Users
#14 Add Settings Screen with Preferences and About Info
#15 Implement Export to CSV Functionality
#16 Enable Tagging of Log Entries
(index truncated when > 25 rows)

⌚ Issue 16 delivered – next run in ~15 min.

markdown

Title

Add Quick-Log Widget for Fast Entry

Description

Implement a compact widget accessible from the home screen or header for quickly submitting a log with default or recent values. This reduces friction for frequent users and improves logging consistency.

Tasks

- Design minimal quick-log UI with adjustable duration, timestamp (now), and method selector.
- Add button to expand into full log form for additional fields.
- Auto-fill fields based on last submission if available.
- Ensure submissions route through the same validation and sync pipeline.
- Track quick-log vs full-log entry in metadata for future analysis.

Acceptance Criteria

- [] Users can log a new session in ≤ 2 taps from the main screen.
- [] Quick-log uses last used method/duration by default.
- [] Full-form expansion preserves any quick-form input already filled.

- [] Entries appear immediately in local cache and queue for sync.

Labels

ux, ui, productivity

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

#1 Set up Clean Architecture Foundation

#2 Implement Multi-Account User Switching

#3 Design Unified Log Detail View

#4 Build Time Range Filter with Presets

#5 Implement Daily and Rolling Usage Charts

#6 Sync Logs Between Firestore and Local Isar Cache

#7 Add Table View for Recent Logs

#8 Implement Stats Overview Dashboard

#9 Add Log Entry Submission UI with Validation

#10 Implement Session Length Histogram Chart

#11 Implement Inhalation Method Pie Chart

#12 Build Offline Sync Status Indicator

#13 Create Onboarding Flow for New Users

#14 Add Settings Screen with Preferences and About Info

#15 Implement Export to CSV Functionality

#16 Enable Tagging of Log Entries

#17 Add Quick-Log Widget for Fast Entry

(index truncated when > 25 rows)

⌚ Issue 17 delivered – next run in ~15 min.

markdown

Title

Support Secure Cloud Backup and Restore

Description

Implement end-to-end encrypted cloud backup and restore functionality to safeguard user data and support device migration. This ensures data integrity across installs while protecting user privacy.

****Tasks****

- Create backup schema (user profiles, logs, tags, preferences) in Firestore.
- Encrypt backup data using device-specific or user-supplied key.
- Provide manual “Backup Now” and “Restore from Backup” buttons in settings.
- Implement optional auto-backup toggle with schedule (e.g., daily).
- Add UI feedback for backup status, last backup date, and restore results.

Acceptance Criteria

- [] Backups include all user data and are encrypted client-side.
- [] Manual backup and restore work reliably with success/failure toasts.
- [] Last backup time is shown in settings.
- [] Users can enable/disable auto-backup in preferences.

Labels

security, backup, data

Estimate

L

- ◆ RUNNING INDEX (update each run; keep \leq 25 lines)

#1 Set up Clean Architecture Foundation

#2 Implement Multi-Account User Switching

#3 Design Unified Log Detail View

#4 Build Time Range Filter with Presets

#5 Implement Daily and Rolling Usage Charts

#6 Sync Logs Between Firestore and Local Isar Cache

#7 Add Table View for Recent Logs
#8 Implement Stats Overview Dashboard
#9 Add Log Entry Submission UI with Validation
#10 Implement Session Length Histogram Chart
#11 Implement Inhalation Method Pie Chart
#12 Build Offline Sync Status Indicator
#13 Create Onboarding Flow for New Users
#14 Add Settings Screen with Preferences and About Info
#15 Implement Export to CSV Functionality
#16 Enable Tagging of Log Entries
#17 Add Quick-Log Widget for Fast Entry
#18 Support Secure Cloud Backup and Restore
(index truncated when > 25 rows)

⌚ Issue 18 delivered – next run in ~15 min.

markdown

Title

Design Log Correlation Insight Module

Description

Create a system to detect correlations between variables like time of day, duration, and method. This module will eventually support personalized insights and automated suggestions based on user behavior patterns.

Tasks

- Define a schema for correlation candidates (e.g., duration vs method).
- Implement logic to compute correlations with significance thresholds.
- Build UI module for displaying top 3 correlations found.
- Add option for users to submit feedback on accuracy/relevance.
- Persist computed correlations for each user profile in cache.

Acceptance Criteria

- [] At least 3 correlation types computed with p-value filtering.
- [] Users see visual summaries (e.g., bar or scatter plots) of correlations.
- [] Correlation insights update weekly or on-demand.
- [] User feedback is stored for future training/adjustment.

Labels

insights, analytics, machine-learning

Estimate

L

◆ RUNNING INDEX (update each run; keep \leq 25 lines)

- #1 Set up Clean Architecture Foundation
 - #2 Implement Multi-Account User Switching
 - #3 Design Unified Log Detail View
 - #4 Build Time Range Filter with Presets
 - #5 Implement Daily and Rolling Usage Charts
 - #6 Sync Logs Between Firestore and Local Isar Cache
 - #7 Add Table View for Recent Logs
 - #8 Implement Stats Overview Dashboard
 - #9 Add Log Entry Submission UI with Validation
 - #10 Implement Session Length Histogram Chart
 - #11 Implement Inhalation Method Pie Chart
 - #12 Build Offline Sync Status Indicator
 - #13 Create Onboarding Flow for New Users
 - #14 Add Settings Screen with Preferences and About Info
 - #15 Implement Export to CSV Functionality
 - #16 Enable Tagging of Log Entries
 - #17 Add Quick-Log Widget for Fast Entry
 - #18 Support Secure Cloud Backup and Restore
 - #19 Design Log Correlation Insight Module
- (index truncated when > 25 rows)

⌚ Issue 19 delivered – next run in ~15 min.

markdown

Title

Implement Weekly Summary Notifications

Description

Introduce scheduled weekly notifications summarizing user activity, trends, and notable stats. These summaries increase engagement and provide a useful recap of recent usage patterns.

Tasks

- Set up notification scheduling logic for end-of-week summaries.
- Generate content dynamically based on log data (e.g., total sessions, top methods, average duration).
- Design notification format (brief + optional “view more” link to open dashboard).
- Allow users to opt in/out from settings.
- Test notification delivery on both Android and iOS.

Acceptance Criteria

- [] Notifications are delivered once per week with valid summary content.
- [] Tapping the notification navigates to relevant dashboard section.
- [] User opt-in is respected and stored in preferences.
- [] Content adapts based on available data and user profile.

Labels

notifications, engagement, analytics

Estimate

M

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

#1 Set up Clean Architecture Foundation

#2 Implement Multi-Account User Switching
#3 Design Unified Log Detail View
#4 Build Time Range Filter with Presets
#5 Implement Daily and Rolling Usage Charts
#6 Sync Logs Between Firestore and Local Isar Cache
#7 Add Table View for Recent Logs
#8 Implement Stats Overview Dashboard
#9 Add Log Entry Submission UI with Validation
#10 Implement Session Length Histogram Chart
#11 Implement Inhalation Method Pie Chart
#12 Build Offline Sync Status Indicator
#13 Create Onboarding Flow for New Users
#14 Add Settings Screen with Preferences and About Info
#15 Implement Export to CSV Functionality
#16 Enable Tagging of Log Entries
#17 Add Quick-Log Widget for Fast Entry
#18 Support Secure Cloud Backup and Restore
#19 Design Log Correlation Insight Module
#20 Implement Weekly Summary Notifications
(index truncated when > 25 rows)

⌚ Issue 20 delivered – next run in ~15 min.

markdown

```
### Title
Integrate Profile-Specific Chart Themes

### Description
Allow each user profile to have personalized visual themes for
charts, including color palettes and smoothing preferences.
This enhances user engagement and clarity, especially for
users tracking multiple contexts or goals.

**Tasks**
- Define chart theme options in user preferences (colors,
smoothing).
```

- Update chart rendering functions to use active profile's theme.
- Add UI controls in settings to select or reset chart theme.
- Store preferences in both Isar and Firestore for sync.
- Provide 3-4 preset themes as starting points.

Acceptance Criteria

- [] Users can customize chart appearance per profile.
- [] Changes apply immediately and persist across sessions/devices.
- [] Defaults are applied when no theme is set.
- [] Charts reflect selected theme across all views (line, pie, histogram).

Labels

ui, personalization, charts

Estimate

M

- ◆ RUNNING INDEX (update each run; keep \leq 25 lines)

- #1 Set up Clean Architecture Foundation
- #2 Implement Multi-Account User Switching
- #3 Design Unified Log Detail View
- #4 Build Time Range Filter with Presets
- #5 Implement Daily and Rolling Usage Charts
- #6 Sync Logs Between Firestore and Local Isar Cache
- #7 Add Table View for Recent Logs
- #8 Implement Stats Overview Dashboard
- #9 Add Log Entry Submission UI with Validation
- #10 Implement Session Length Histogram Chart
- #11 Implement Inhalation Method Pie Chart
- #12 Build Offline Sync Status Indicator
- #13 Create Onboarding Flow for New Users
- #14 Add Settings Screen with Preferences and About Info
- #15 Implement Export to CSV Functionality
- #16 Enable Tagging of Log Entries

#17 Add Quick-Log Widget for Fast Entry
#18 Support Secure Cloud Backup and Restore
#19 Design Log Correlation Insight Module
#20 Implement Weekly Summary Notifications
#21 Integrate Profile-Specific Chart Themes
(index truncated when > 25 rows)

⌚ Issue 21 delivered – next run in ~15 min.

markdown

Title

Support Advanced Filtering by Tags, Method, and Device

Description

Expand the filtering system to allow users to narrow down logs and analytics views using tags, inhalation methods, and device types. This will enhance analysis and enable targeted comparisons within user behavior.

Tasks

- Create multi-select filter UI components for tags, method, and device.
- Add filtering logic to charts, tables, and stats dashboard to respect selections.
- Store active filters in Riverpod global state and allow clearing all with one tap.
- Optimize filtering performance with memoized queries or indexes.
- Design visual indicators when filters are active on any view.

Acceptance Criteria

- [] Users can combine any subset of filters (tags, method, device).
- [] Filters apply in real time across charts and tables.
- [] "Clear All Filters" button resets to full dataset.
- [] Performance impact remains minimal even on 10k+ logs.

```
### Labels  
filters, ui, analytics
```

```
### Estimate
```

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

- #1 Set up Clean Architecture Foundation
 - #2 Implement Multi-Account User Switching
 - #3 Design Unified Log Detail View
 - #4 Build Time Range Filter with Presets
 - #5 Implement Daily and Rolling Usage Charts
 - #6 Sync Logs Between Firestore and Local Isar Cache
 - #7 Add Table View for Recent Logs
 - #8 Implement Stats Overview Dashboard
 - #9 Add Log Entry Submission UI with Validation
 - #10 Implement Session Length Histogram Chart
 - #11 Implement Inhalation Method Pie Chart
 - #12 Build Offline Sync Status Indicator
 - #13 Create Onboarding Flow for New Users
 - #14 Add Settings Screen with Preferences and About Info
 - #15 Implement Export to CSV Functionality
 - #16 Enable Tagging of Log Entries
 - #17 Add Quick-Log Widget for Fast Entry
 - #18 Support Secure Cloud Backup and Restore
 - #19 Design Log Correlation Insight Module
 - #20 Implement Weekly Summary Notifications
 - #21 Integrate Profile-Specific Chart Themes
 - #22 Support Advanced Filtering by Tags, Method, and Device
- (index truncated when > 25 rows)

⌚ Issue 22 delivered – next run in ~15 min.

markdown

Title

Add Custom Time Range Analysis with Bookmarking

Description

Allow users to define and save custom time ranges for focused analysis. This enables comparison between specific periods (e.g., “Vacation Week”, “Post-Treatment”) and supports bookmarking for repeat access.

****Tasks****

- Build UI to define and label custom date ranges.
- Store custom ranges in local and synced profile settings.
- Add option to bookmark active range for quick selection later.
- Update filter logic to support selecting a saved range from dropdown.
- Display bookmark name and details when active.

Acceptance Criteria

- [] Users can create, name, and select custom time ranges.
- [] Saved ranges persist across devices and sessions.
- [] Bookmarking interface integrates with existing range selector.
- [] Visual cue shows when a bookmarked range is in use.

Labels

filters, personalization, ui

Estimate

M

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

#1 Set up Clean Architecture Foundation

#2 Implement Multi-Account User Switching

#3 Design Unified Log Detail View

#4 Build Time Range Filter with Presets

#5 Implement Daily and Rolling Usage Charts

#6 Sync Logs Between Firestore and Local Isar Cache

#7 Add Table View for Recent Logs

#8 Implement Stats Overview Dashboard
#9 Add Log Entry Submission UI with Validation
#10 Implement Session Length Histogram Chart
#11 Implement Inhalation Method Pie Chart
#12 Build Offline Sync Status Indicator
#13 Create Onboarding Flow for New Users
#14 Add Settings Screen with Preferences and About Info
#15 Implement Export to CSV Functionality
#16 Enable Tagging of Log Entries
#17 Add Quick-Log Widget for Fast Entry
#18 Support Secure Cloud Backup and Restore
#19 Design Log Correlation Insight Module
#20 Implement Weekly Summary Notifications
#21 Integrate Profile-Specific Chart Themes
#22 Support Advanced Filtering by Tags, Method, and Device
#23 Add Custom Time Range Analysis with Bookmarking
(index truncated when > 25 rows)

⌚ Issue 23 delivered – next run in ~15 min.

markdown

Title

Add Device Usage Breakdown Chart

Description

Create a chart that displays usage breakdown by device type (e.g., phone, tablet, wearable) to help users understand how and where they're logging most frequently. This will be useful for analyzing usage contexts and device habits.

Tasks

- Extend log model if needed to ensure consistent device type tracking.
- Parse logs to group and sum by device type.
- Build bar or donut chart with labeled segments and totals.
- Respect time range and active filters when computing data.

- Integrate chart into the stats dashboard or a standalone view.

Acceptance Criteria

- [] Chart shows at least 3 device types with usage metrics.
- [] Supports filtering by time, tags, and user profile.
- [] Labels clearly reflect counts and/or durations.
- [] Chart adapts to different screen sizes and theme settings.

Labels

analytics, charts, ui

Estimate

M

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

- #1 Set up Clean Architecture Foundation
- #2 Implement Multi-Account User Switching
- #3 Design Unified Log Detail View
- #4 Build Time Range Filter with Presets
- #5 Implement Daily and Rolling Usage Charts
- #6 Sync Logs Between Firestore and Local Isar Cache
- #7 Add Table View for Recent Logs
- #8 Implement Stats Overview Dashboard
- #9 Add Log Entry Submission UI with Validation
- #10 Implement Session Length Histogram Chart
- #11 Implement Inhalation Method Pie Chart
- #12 Build Offline Sync Status Indicator
- #13 Create Onboarding Flow for New Users
- #14 Add Settings Screen with Preferences and About Info
- #15 Implement Export to CSV Functionality
- #16 Enable Tagging of Log Entries
- #17 Add Quick-Log Widget for Fast Entry
- #18 Support Secure Cloud Backup and Restore
- #19 Design Log Correlation Insight Module

#20 Implement Weekly Summary Notifications
#21 Integrate Profile-Specific Chart Themes
#22 Support Advanced Filtering by Tags, Method, and Device
#23 Add Custom Time Range Analysis with Bookmarking
#24 Add Device Usage Breakdown Chart
(index truncated when > 25 rows)

⌚ Issue 24 delivered – next run in ~15 min.

markdown

Title

Build Log Streak Tracker and Display

Description

Implement logic and UI to track user logging streaks (consecutive days with at least one log) and display current and longest streaks. This feature supports habit formation and user motivation through visual progress indicators.

Tasks

- Develop backend logic to calculate streaks from log timestamps.
- Add streak data to stats dashboard (e.g., “Current Streak: 5 days”).
- Design optional calendar-style view showing daily log activity.
- Cache streak computations to avoid recalculating on every load.
- Add milestone detection (e.g., 7-day, 30-day streaks) for future use.

Acceptance Criteria

- [] Current and longest streaks are displayed clearly in dashboard.
- [] Streaks update in real time with new log submissions.
- [] Missed days reset the current streak and are reflected immediately.
- [] Performance remains responsive for up to 1 year of data.

```
### Labels  
analytics, habits, ui
```

```
### Estimate  
M
```

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
 - #1 Set up Clean Architecture Foundation
 - #2 Implement Multi-Account User Switching
 - #3 Design Unified Log Detail View
 - #4 Build Time Range Filter with Presets
 - #5 Implement Daily and Rolling Usage Charts
 - #6 Sync Logs Between Firestore and Local Isar Cache
 - #7 Add Table View for Recent Logs
 - #8 Implement Stats Overview Dashboard
 - #9 Add Log Entry Submission UI with Validation
 - #10 Implement Session Length Histogram Chart
 - #11 Implement Inhalation Method Pie Chart
 - #12 Build Offline Sync Status Indicator
 - #13 Create Onboarding Flow for New Users
 - #14 Add Settings Screen with Preferences and About Info
 - #15 Implement Export to CSV Functionality
 - #16 Enable Tagging of Log Entries
 - #17 Add Quick-Log Widget for Fast Entry
 - #18 Support Secure Cloud Backup and Restore
 - #19 Design Log Correlation Insight Module
 - #20 Implement Weekly Summary Notifications
 - #21 Integrate Profile-Specific Chart Themes
 - #22 Support Advanced Filtering by Tags, Method, and Device
 - #23 Add Custom Time Range Analysis with Bookmarking
 - #24 Add Device Usage Breakdown Chart
 - #25 Build Log Streak Tracker and Display
- (index truncated – see earlier messages)

⌚ Issue 25 delivered – next run in ~15 min.

markdown

Title

Implement Log Deletion and Undo Feature

Description

Allow users to delete individual log entries and optionally undo the action within a short time window. This improves data control while preventing accidental loss.

Tasks

- Add delete icon to each row in the log table and detail view.
- Implement soft-delete logic with a 5-second undo snackbar.
- Ensure deletion syncs across Isar and Firestore.
- Design confirmation dialog for bulk delete (future support).
- Log deletion metadata for audit/debug (e.g., deletion timestamp).

Acceptance Criteria

- [] Logs can be deleted via table and detail view UI.
- [] Undo option appears after deletion and restores the entry.
- [] Deletions reflect in all synced views and storage layers.
- [] No accidental deletion without confirmation or undo option.

Labels

data, deletion, ui

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #1 Set up Clean Architecture Foundation

- #2 Implement Multi-Account User Switching
- #3 Design Unified Log Detail View
- #4 Build Time Range Filter with Presets
- #5 Implement Daily and Rolling Usage Charts
- #6 Sync Logs Between Firestore and Local Isar Cache
- #7 Add Table View for Recent Logs
- #8 Implement Stats Overview Dashboard
- #9 Add Log Entry Submission UI with Validation
- #10 Implement Session Length Histogram Chart
- #11 Implement Inhalation Method Pie Chart
- #12 Build Offline Sync Status Indicator
- #13 Create Onboarding Flow for New Users
- #14 Add Settings Screen with Preferences and About Info
- #15 Implement Export to CSV Functionality
- #16 Enable Tagging of Log Entries
- #17 Add Quick-Log Widget for Fast Entry
- #18 Support Secure Cloud Backup and Restore
- #19 Design Log Correlation Insight Module
- #20 Implement Weekly Summary Notifications
- #21 Integrate Profile-Specific Chart Themes
- #22 Support Advanced Filtering by Tags, Method, and Device
- #23 Add Custom Time Range Analysis with Bookmarking
- #24 Add Device Usage Breakdown Chart
- #25 Build Log Streak Tracker and Display
- #26 Implement Log Deletion and Undo Feature

⌚ Issue 26 delivered – next run in ~15 min.

markdown

Title

Implement In-App Feedback and Issue Reporting

Description

Add an in-app mechanism for users to report bugs, suggest

features, or provide general feedback. This supports better communication, faster debugging, and user-driven improvements.

****Tasks****

- Create feedback form with fields: category (bug/suggestion), message, and optional email.
- Store submissions in Firestore under a dedicated `feedback` collection.
- Add feedback access button to settings screen and post-success toast.
- Implement basic spam protection (e.g., rate limit per profile).
- Prepare admin query interface (temporary or future support) to review submissions.

Acceptance Criteria

- [] Feedback form is accessible and intuitive.
- [] All feedback is correctly stored and labeled in Firestore.
- [] Users receive confirmation of successful submission.
- [] Rate-limiting prevents accidental spam or abuse.

Labels

feedback, support, ui

Estimate

S

◆ RUNNING INDEX (update each run; keep \leq 25 lines)

(index truncated – see earlier messages)

#1 Set up Clean Architecture Foundation

#2 Implement Multi-Account User Switching

#3 Design Unified Log Detail View

#4 Build Time Range Filter with Presets

#5 Implement Daily and Rolling Usage Charts

#6 Sync Logs Between Firestore and Local Isar Cache

#7 Add Table View for Recent Logs

#8 Implement Stats Overview Dashboard

#9 Add Log Entry Submission UI with Validation

#10 Implement Session Length Histogram Chart
#11 Implement Inhalation Method Pie Chart
#12 Build Offline Sync Status Indicator
#13 Create Onboarding Flow for New Users
#14 Add Settings Screen with Preferences and About Info
#15 Implement Export to CSV Functionality
#16 Enable Tagging of Log Entries
#17 Add Quick-Log Widget for Fast Entry
#18 Support Secure Cloud Backup and Restore
#19 Design Log Correlation Insight Module
#20 Implement Weekly Summary Notifications
#21 Integrate Profile-Specific Chart Themes
#22 Support Advanced Filtering by Tags, Method, and Device
#23 Add Custom Time Range Analysis with Bookmarking
#24 Add Device Usage Breakdown Chart
#25 Build Log Streak Tracker and Display
#26 Implement Log Deletion and Undo Feature
#27 Implement In-App Feedback and Issue Reporting

⌚ Issue 27 delivered – next run in ~15 min.

markdown

Title

Add Contextual Log Suggestions Based on Time and Patterns

Description

Introduce smart suggestions that prompt users to log when they typically do, based on time-of-day patterns and recent behavior. This feature aims to improve consistency and reduce missed entries through subtle, context-aware nudges.

Tasks

- Analyze historical logs to detect common logging windows per profile.
- Build local notification system to suggest logging during those windows.

- Design nudge prompt UI that respects quiet hours and user preferences.
- Include toggle in settings to enable/disable suggestions.
- Track prompt interactions (accepted, ignored, dismissed) for future tuning.

Acceptance Criteria

- [] Prompts are sent only during high-likelihood periods per user profile.
- [] Nudges respect quiet hours and are not shown more than once per window.
- [] Users can opt out from suggestion prompts in settings.
- [] Prompt accuracy improves based on recent behavior.

Labels

engagement, personalization, notifications

Estimate

M

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#1 Set up Clean Architecture Foundation

#2 Implement Multi-Account User Switching

#3 Design Unified Log Detail View

#4 Build Time Range Filter with Presets

#5 Implement Daily and Rolling Usage Charts

#6 Sync Logs Between Firestore and Local Isar Cache

#7 Add Table View for Recent Logs

#8 Implement Stats Overview Dashboard

#9 Add Log Entry Submission UI with Validation

#10 Implement Session Length Histogram Chart

#11 Implement Inhalation Method Pie Chart

#12 Build Offline Sync Status Indicator

#13 Create Onboarding Flow for New Users

#14 Add Settings Screen with Preferences and About Info

#15 Implement Export to CSV Functionality

#16 Enable Tagging of Log Entries
#17 Add Quick-Log Widget for Fast Entry
#18 Support Secure Cloud Backup and Restore
#19 Design Log Correlation Insight Module
#20 Implement Weekly Summary Notifications
#21 Integrate Profile-Specific Chart Themes
#22 Support Advanced Filtering by Tags, Method, and Device
#23 Add Custom Time Range Analysis with Bookmarking
#24 Add Device Usage Breakdown Chart
#25 Build Log Streak Tracker and Display
#26 Implement Log Deletion and Undo Feature
#27 Implement In-App Feedback and Issue Reporting
#28 Add Contextual Log Suggestions Based on Time and Patterns

⌚ Issue 28 delivered – next run in ~15 min.

markdown

```
### Title
Add Sync Conflict Resolution and Audit Logging

### Description
Build a conflict resolution system to handle edge cases during sync between Isar and Firestore, and implement internal audit logging for traceability. This ensures data consistency and helps debug sync anomalies.

**Tasks**
- Define resolution rules (e.g., latest timestamp wins, merge tags).
- Implement conflict resolution logic in sync pipeline.
- Add audit log entries for sync actions (create, update, delete, conflict).
- Store audit logs locally with optional Firestore mirror for debug use.
- Provide temporary debug view for inspecting recent sync logs.
```

Acceptance Criteria

- [] Sync conflicts are resolved automatically with documented rules.
- [] Audit logs are generated for all sync-related operations.
- [] Logs include timestamps, action types, and affected entry IDs.
- [] Debug view (internal) shows most recent 50 audit events.

Labels

sync, debug, data-integrity

Estimate

L

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#1 Set up Clean Architecture Foundation

#2 Implement Multi-Account User Switching

#3 Design Unified Log Detail View

#4 Build Time Range Filter with Presets

#5 Implement Daily and Rolling Usage Charts

#6 Sync Logs Between Firestore and Local Isar Cache

#7 Add Table View for Recent Logs

#8 Implement Stats Overview Dashboard

#9 Add Log Entry Submission UI with Validation

#10 Implement Session Length Histogram Chart

#11 Implement Inhalation Method Pie Chart

#12 Build Offline Sync Status Indicator

#13 Create Onboarding Flow for New Users

#14 Add Settings Screen with Preferences and About Info

#15 Implement Export to CSV Functionality

#16 Enable Tagging of Log Entries

#17 Add Quick-Log Widget for Fast Entry

#18 Support Secure Cloud Backup and Restore

#19 Design Log Correlation Insight Module

#20 Implement Weekly Summary Notifications
#21 Integrate Profile-Specific Chart Themes
#22 Support Advanced Filtering by Tags, Method, and Device
#23 Add Custom Time Range Analysis with Bookmarking
#24 Add Device Usage Breakdown Chart
#25 Build Log Streak Tracker and Display
#26 Implement Log Deletion and Undo Feature
#27 Implement In-App Feedback and Issue Reporting
#28 Add Contextual Log Suggestions Based on Time and Patterns
#29 Add Sync Conflict Resolution and Audit Logging

⌚ Issue 29 delivered – next run in ~15 min.

markdown

Title

Create Daily Mood + Log Correlation Overlay

Description

Allow users to track mood alongside usage logs and visualize how inhalation patterns may influence their wellbeing. This optional feature offers deeper personal insight and adds mental health awareness context to data trends.

Tasks

- Add `MoodEntry` model with simple 1-5 scale and timestamp.
- Design mood entry UI prompt with optional daily reminder.
- Overlay mood trendline on existing daily usage chart.
- Implement correlation comparison between mood and duration/method.
- Ensure mood data is profile-scoped and synced.

Acceptance Criteria

- [] Users can log 1-5 mood rating per day.
- [] Mood chart overlays align with usage data over time.
- [] Correlation summary shown below chart if data exists.
- [] Users can disable mood tracking from settings.

Labels

wellness, analytics, ui

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#1 Set up Clean Architecture Foundation

#2 Implement Multi-Account User Switching

#3 Design Unified Log Detail View

#4 Build Time Range Filter with Presets

#5 Implement Daily and Rolling Usage Charts

#6 Sync Logs Between Firestore and Local Isar Cache

#7 Add Table View for Recent Logs

#8 Implement Stats Overview Dashboard

#9 Add Log Entry Submission UI with Validation

#10 Implement Session Length Histogram Chart

#11 Implement Inhalation Method Pie Chart

#12 Build Offline Sync Status Indicator

#13 Create Onboarding Flow for New Users

#14 Add Settings Screen with Preferences and About Info

#15 Implement Export to CSV Functionality

#16 Enable Tagging of Log Entries

#17 Add Quick-Log Widget for Fast Entry

#18 Support Secure Cloud Backup and Restore

#19 Design Log Correlation Insight Module

#20 Implement Weekly Summary Notifications

#21 Integrate Profile-Specific Chart Themes

#22 Support Advanced Filtering by Tags, Method, and Device

#23 Add Custom Time Range Analysis with Bookmarking

#24 Add Device Usage Breakdown Chart

#25 Build Log Streak Tracker and Display

#26 Implement Log Deletion and Undo Feature

#27 Implement In-App Feedback and Issue Reporting

#28 Add Contextual Log Suggestions Based on Time and Patterns
#29 Add Sync Conflict Resolution and Audit Logging
#30 Create Daily Mood + Log Correlation Overlay

⌚ Issue 30 delivered – next run in ~15 min.

markdown

Title

Design Activity Heatmap Calendar View

Description

Build a visual calendar heatmap that shows logging activity across days and months, helping users recognize long-term patterns and gaps in usage. This offers a powerful and intuitive glance at consistency.

Tasks

- Create calendar view UI with color intensity based on log count per day.
- Support hover/tap to reveal log count and total duration per date.
- Filter by time range, tags, and profile like other views.
- Enable navigation across months and years.
- Optimize performance for rendering year-long views.

Acceptance Criteria

- [] Heatmap reflects log density with color gradient per day.
- [] Hover or tap reveals activity summary for that date.
- [] Filters (range, tag, method) correctly affect heatmap rendering.
- [] View renders full year without performance issues.

Labels

visualization, ui, calendar

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #1 Set up Clean Architecture Foundation
- #2 Implement Multi-Account User Switching
- #3 Design Unified Log Detail View
- #4 Build Time Range Filter with Presets
- #5 Implement Daily and Rolling Usage Charts
- #6 Sync Logs Between Firestore and Local Isar Cache
- #7 Add Table View for Recent Logs
- #8 Implement Stats Overview Dashboard
- #9 Add Log Entry Submission UI with Validation
- #10 Implement Session Length Histogram Chart
- #11 Implement Inhalation Method Pie Chart
- #12 Build Offline Sync Status Indicator
- #13 Create Onboarding Flow for New Users
- #14 Add Settings Screen with Preferences and About Info
- #15 Implement Export to CSV Functionality
- #16 Enable Tagging of Log Entries
- #17 Add Quick-Log Widget for Fast Entry
- #18 Support Secure Cloud Backup and Restore
- #19 Design Log Correlation Insight Module
- #20 Implement Weekly Summary Notifications
- #21 Integrate Profile-Specific Chart Themes
- #22 Support Advanced Filtering by Tags, Method, and Device
- #23 Add Custom Time Range Analysis with Bookmarking
- #24 Add Device Usage Breakdown Chart
- #25 Build Log Streak Tracker and Display
- #26 Implement Log Deletion and Undo Feature
- #27 Implement In-App Feedback and Issue Reporting
- #28 Add Contextual Log Suggestions Based on Time and Patterns
- #29 Add Sync Conflict Resolution and Audit Logging
- #30 Create Daily Mood + Log Correlation Overlay
- #31 Design Activity Heatmap Calendar View

⌚ Issue 31 delivered – next run in ~15 min.

markdown

Title

Build Session Notes with Markdown Support

Description

Enable users to attach optional session notes to each log entry using a simple markdown editor. This empowers journaling, tracking context, and making logs more meaningful over time.

Tasks

- Add `notes` field to `LogEntry` schema and sync pipeline.
- Integrate lightweight markdown editor in log submission and detail views.
- Render markdown with formatting (headers, bullets, bold/italic, links).
- Support real-time preview and edit toggle.
- Enable search filter to query logs by note contents.

Acceptance Criteria

- [] Users can add and edit markdown notes per log.
- [] Markdown is rendered cleanly in detail view with proper formatting.
- [] Search includes note text with highlight on matches.
- [] Notes sync across devices and persist in both Isar and Firestore.

Labels

journaling, markdown, ui

Estimate

M

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#1 Set up Clean Architecture Foundation

- #2 Implement Multi-Account User Switching
- #3 Design Unified Log Detail View
- #4 Build Time Range Filter with Presets
- #5 Implement Daily and Rolling Usage Charts
- #6 Sync Logs Between Firestore and Local Isar Cache
- #7 Add Table View for Recent Logs
- #8 Implement Stats Overview Dashboard
- #9 Add Log Entry Submission UI with Validation
- #10 Implement Session Length Histogram Chart
- #11 Implement Inhalation Method Pie Chart
- #12 Build Offline Sync Status Indicator
- #13 Create Onboarding Flow for New Users
- #14 Add Settings Screen with Preferences and About Info
- #15 Implement Export to CSV Functionality
- #16 Enable Tagging of Log Entries
- #17 Add Quick-Log Widget for Fast Entry
- #18 Support Secure Cloud Backup and Restore
- #19 Design Log Correlation Insight Module
- #20 Implement Weekly Summary Notifications
- #21 Integrate Profile-Specific Chart Themes
- #22 Support Advanced Filtering by Tags, Method, and Device
- #23 Add Custom Time Range Analysis with Bookmarking
- #24 Add Device Usage Breakdown Chart
- #25 Build Log Streak Tracker and Display
- #26 Implement Log Deletion and Undo Feature
- #27 Implement In-App Feedback and Issue Reporting
- #28 Add Contextual Log Suggestions Based on Time and Patterns
- #29 Add Sync Conflict Resolution and Audit Logging
- #30 Create Daily Mood + Log Correlation Overlay
- #31 Design Activity Heatmap Calendar View
- #32 Build Session Notes with Markdown Support

 Issue 32 delivered – next run in ~15 min.

Title

Enable Full-Text Log Search with Highlighting

Description

Introduce a full-text search engine that allows users to query logs by keyword, including session notes and metadata fields. Highlight matched text in the results to improve discoverability and navigation.

Tasks

- Index logs by timestamp, method, device, and note content.
- Build search bar UI with instant search capabilities.
- Display matching entries in a list with highlighted keywords.
- Support filtering search results using existing filters (time, tag, etc.).
- Optimize performance for search over 10k+ entries.

Acceptance Criteria

- [] Users can search logs by any keyword or phrase.
- [] Matched terms are highlighted in the result preview.
- [] Results update in real time and respect active filters.
- [] Search engine handles at least 10,000 entries without lag.

Labels

search, filters, performance

Estimate

M

◆ RUNNING INDEX (update each run; keep \leq 25 lines)

(index truncated – see earlier messages)

#1 Set up Clean Architecture Foundation

#2 Implement Multi-Account User Switching

#3 Design Unified Log Detail View

#4 Build Time Range Filter with Presets

#5 Implement Daily and Rolling Usage Charts

- #6 Sync Logs Between Firestore and Local Isar Cache
- #7 Add Table View for Recent Logs
- #8 Implement Stats Overview Dashboard
- #9 Add Log Entry Submission UI with Validation
- #10 Implement Session Length Histogram Chart
- #11 Implement Inhalation Method Pie Chart
- #12 Build Offline Sync Status Indicator
- #13 Create Onboarding Flow for New Users
- #14 Add Settings Screen with Preferences and About Info
- #15 Implement Export to CSV Functionality
- #16 Enable Tagging of Log Entries
- #17 Add Quick-Log Widget for Fast Entry
- #18 Support Secure Cloud Backup and Restore
- #19 Design Log Correlation Insight Module
- #20 Implement Weekly Summary Notifications
- #21 Integrate Profile-Specific Chart Themes
- #22 Support Advanced Filtering by Tags, Method, and Device
- #23 Add Custom Time Range Analysis with Bookmarking
- #24 Add Device Usage Breakdown Chart
- #25 Build Log Streak Tracker and Display
- #26 Implement Log Deletion and Undo Feature
- #27 Implement In-App Feedback and Issue Reporting
- #28 Add Contextual Log Suggestions Based on Time and Patterns
- #29 Add Sync Conflict Resolution and Audit Logging
- #30 Create Daily Mood + Log Correlation Overlay
- #31 Design Activity Heatmap Calendar View
- #32 Build Session Notes with Markdown Support
- #33 Enable Full-Text Log Search with Highlighting

 Issue 33 delivered – next run in ~15 min.

markdown

Title

Implement Session Timeline View with Zoomable Playback

Description

Create a horizontal timeline that displays session entries across the day with visual blocks indicating duration and gaps. Allow users to zoom in/out and scroll to inspect their daily or weekly patterns more intuitively.

****Tasks****

- Build horizontal scrollable timeline layout with markers for hours.
- Render log sessions as bars proportional to duration and positioned by timestamp.
- Implement zoom controls (e.g., hourly to 5-minute increments).
- Add support for viewing one day, multiple days, or a whole week.
- Highlight session details on tap or hover.

Acceptance Criteria

- [] Timeline renders logs accurately with proportional visual spacing.
- [] Zoom and scroll are fluid and responsive.
- [] Tapping a block opens session detail.
- [] Filters and active profile affect displayed sessions.

Labels

timeline, ui, data-visualization

Estimate

L

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#1 Set up Clean Architecture Foundation

#2 Implement Multi-Account User Switching

#3 Design Unified Log Detail View

#4 Build Time Range Filter with Presets

- #5 Implement Daily and Rolling Usage Charts
- #6 Sync Logs Between Firestore and Local Isar Cache
- #7 Add Table View for Recent Logs
- #8 Implement Stats Overview Dashboard
- #9 Add Log Entry Submission UI with Validation
- #10 Implement Session Length Histogram Chart
- #11 Implement Inhalation Method Pie Chart
- #12 Build Offline Sync Status Indicator
- #13 Create Onboarding Flow for New Users
- #14 Add Settings Screen with Preferences and About Info
- #15 Implement Export to CSV Functionality
- #16 Enable Tagging of Log Entries
- #17 Add Quick-Log Widget for Fast Entry
- #18 Support Secure Cloud Backup and Restore
- #19 Design Log Correlation Insight Module
- #20 Implement Weekly Summary Notifications
- #21 Integrate Profile-Specific Chart Themes
- #22 Support Advanced Filtering by Tags, Method, and Device
- #23 Add Custom Time Range Analysis with Bookmarking
- #24 Add Device Usage Breakdown Chart
- #25 Build Log Streak Tracker and Display
- #26 Implement Log Deletion and Undo Feature
- #27 Implement In-App Feedback and Issue Reporting
- #28 Add Contextual Log Suggestions Based on Time and Patterns
- #29 Add Sync Conflict Resolution and Audit Logging
- #30 Create Daily Mood + Log Correlation Overlay
- #31 Design Activity Heatmap Calendar View
- #32 Build Session Notes with Markdown Support
- #33 Enable Full-Text Log Search with Highlighting
- #34 Implement Session Timeline View with Zoomable Playback

 Issue 34 delivered – next run in ~15 min.

Title

Add User Analytics Opt-In and Consent Flow

Description

Create a transparent opt-in system for anonymous usage analytics to improve the app's insights while respecting user privacy. This supports ethical data collection and compliance with privacy best practices.

****Tasks****

- Add toggle in onboarding and settings for enabling/disabling analytics.
- Log opt-in status and changes in local preferences.
- Wrap all analytics events in a conditional check for user consent.
- Show brief explanation of what is tracked (e.g., feature usage, error reports).
- Design lightweight UI explaining benefits and privacy guarantees.

Acceptance Criteria

- [] Users must explicitly opt in before analytics is activated.
- [] Consent status persists and can be changed in settings.
- [] All analytics events respect current consent state.
- [] Opt-in copy is clear, accurate, and user-friendly.

Labels

privacy, analytics, onboarding

Estimate

S

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#1 Set up Clean Architecture Foundation

#2 Implement Multi-Account User Switching

#3 Design Unified Log Detail View

#4 Build Time Range Filter with Presets

- #5 Implement Daily and Rolling Usage Charts
- #6 Sync Logs Between Firestore and Local Isar Cache
- #7 Add Table View for Recent Logs
- #8 Implement Stats Overview Dashboard
- #9 Add Log Entry Submission UI with Validation
- #10 Implement Session Length Histogram Chart
- #11 Implement Inhalation Method Pie Chart
- #12 Build Offline Sync Status Indicator
- #13 Create Onboarding Flow for New Users
- #14 Add Settings Screen with Preferences and About Info
- #15 Implement Export to CSV Functionality
- #16 Enable Tagging of Log Entries
- #17 Add Quick-Log Widget for Fast Entry
- #18 Support Secure Cloud Backup and Restore
- #19 Design Log Correlation Insight Module
- #20 Implement Weekly Summary Notifications
- #21 Integrate Profile-Specific Chart Themes
- #22 Support Advanced Filtering by Tags, Method, and Device
- #23 Add Custom Time Range Analysis with Bookmarking
- #24 Add Device Usage Breakdown Chart
- #25 Build Log Streak Tracker and Display
- #26 Implement Log Deletion and Undo Feature
- #27 Implement In-App Feedback and Issue Reporting
- #28 Add Contextual Log Suggestions Based on Time and Patterns
- #29 Add Sync Conflict Resolution and Audit Logging
- #30 Create Daily Mood + Log Correlation Overlay
- #31 Design Activity Heatmap Calendar View
- #32 Build Session Notes with Markdown Support
- #33 Enable Full-Text Log Search with Highlighting
- #34 Implement Session Timeline View with Zoomable Playback
- #35 Add User Analytics Opt-In and Consent Flow

 Issue 35 delivered – next run in ~15 min.

markdown

Title

Support Multi-Device Session Linking

Description

Enable the ability to log a session across multiple devices (e.g., start on phone, finish on watch) and merge the data into a single session. This improves accuracy for users who transition between devices mid-use.

Tasks

- Define a `sessionGroupId` to associate logs from different devices.
- Add UI in log detail view to show linked device segments.
- Implement merging logic to combine duration, tags, and notes.
- Provide interface for users to manually link/unlink sessions.
- Ensure syncing maintains session links across Firestore and Isar.

Acceptance Criteria

- [] Users can view and edit grouped sessions across devices.
- [] Linked sessions are merged for charts and statistics.
- [] UI clearly indicates when a session contains multiple device entries.
- [] Manual link/unlink is intuitive and documented in onboarding/help.

Labels

multi-device, data-model, sync

Estimate

L

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#1 Set up Clean Architecture Foundation

#2 Implement Multi-Account User Switching

- #3 Design Unified Log Detail View
- #4 Build Time Range Filter with Presets
- #5 Implement Daily and Rolling Usage Charts
- #6 Sync Logs Between Firestore and Local Isar Cache
- #7 Add Table View for Recent Logs
- #8 Implement Stats Overview Dashboard
- #9 Add Log Entry Submission UI with Validation
- #10 Implement Session Length Histogram Chart
- #11 Implement Inhalation Method Pie Chart
- #12 Build Offline Sync Status Indicator
- #13 Create Onboarding Flow for New Users
- #14 Add Settings Screen with Preferences and About Info
- #15 Implement Export to CSV Functionality
- #16 Enable Tagging of Log Entries
- #17 Add Quick-Log Widget for Fast Entry
- #18 Support Secure Cloud Backup and Restore
- #19 Design Log Correlation Insight Module
- #20 Implement Weekly Summary Notifications
- #21 Integrate Profile-Specific Chart Themes
- #22 Support Advanced Filtering by Tags, Method, and Device
- #23 Add Custom Time Range Analysis with Bookmarking
- #24 Add Device Usage Breakdown Chart
- #25 Build Log Streak Tracker and Display
- #26 Implement Log Deletion and Undo Feature
- #27 Implement In-App Feedback and Issue Reporting
- #28 Add Contextual Log Suggestions Based on Time and Patterns
- #29 Add Sync Conflict Resolution and Audit Logging
- #30 Create Daily Mood + Log Correlation Overlay
- #31 Design Activity Heatmap Calendar View
- #32 Build Session Notes with Markdown Support
- #33 Enable Full-Text Log Search with Highlighting
- #34 Implement Session Timeline View with Zoomable Playback
- #35 Add User Analytics Opt-In and Consent Flow
- #36 Support Multi-Device Session Linking

⌚ Issue 36 delivered – next run in ~15 min.

markdown

Title

Implement Log-Based Reminder System

Description

Create a reminder system that suggests logging based on time since last entry. This aids users in maintaining consistent tracking and reduces gaps without needing fixed schedules.

Tasks

- Track timestamp of most recent log per profile.
- Define dynamic thresholds (e.g., 6h, 12h, 24h) to trigger reminders.
- Send push notification or in-app prompt if no log has occurred by threshold.
- Add snooze and dismiss options for each reminder.
- Allow customization of reminder frequency in settings.

Acceptance Criteria

- [] Users receive reminders if inactive beyond threshold.
- [] Snooze delays next check by at least 6 hours.
- [] Settings screen allows enabling/disabling reminders.
- [] Reminders respect quiet hours and are not duplicated.

Labels

notifications, reminders, habit-tracking

Estimate

M

◆ RUNNING INDEX (update each run; keep \leq 25 lines)

(index truncated – see earlier messages)

#1 Set up Clean Architecture Foundation

#2 Implement Multi-Account User Switching

#3 Design Unified Log Detail View

#4 Build Time Range Filter with Presets

- #5 Implement Daily and Rolling Usage Charts
- #6 Sync Logs Between Firestore and Local Isar Cache
- #7 Add Table View for Recent Logs
- #8 Implement Stats Overview Dashboard
- #9 Add Log Entry Submission UI with Validation
- #10 Implement Session Length Histogram Chart
- #11 Implement Inhalation Method Pie Chart
- #12 Build Offline Sync Status Indicator
- #13 Create Onboarding Flow for New Users
- #14 Add Settings Screen with Preferences and About Info
- #15 Implement Export to CSV Functionality
- #16 Enable Tagging of Log Entries
- #17 Add Quick-Log Widget for Fast Entry
- #18 Support Secure Cloud Backup and Restore
- #19 Design Log Correlation Insight Module
- #20 Implement Weekly Summary Notifications
- #21 Integrate Profile-Specific Chart Themes
- #22 Support Advanced Filtering by Tags, Method, and Device
- #23 Add Custom Time Range Analysis with Bookmarking
- #24 Add Device Usage Breakdown Chart
- #25 Build Log Streak Tracker and Display
- #26 Implement Log Deletion and Undo Feature
- #27 Implement In-App Feedback and Issue Reporting
- #28 Add Contextual Log Suggestions Based on Time and Patterns
- #29 Add Sync Conflict Resolution and Audit Logging
- #30 Create Daily Mood + Log Correlation Overlay
- #31 Design Activity Heatmap Calendar View
- #32 Build Session Notes with Markdown Support
- #33 Enable Full-Text Log Search with Highlighting
- #34 Implement Session Timeline View with Zoomable Playback
- #35 Add User Analytics Opt-In and Consent Flow
- #36 Support Multi-Device Session Linking
- #37 Implement Log-Based Reminder System

⌚ Issue 37 delivered – next run in ~15 min.

markdown

Title

Add Data Sync Status and Conflict Reporting Dashboard

Description

Develop an internal dashboard view to monitor sync activity, last sync timestamps, and any sync-related conflicts for debugging or diagnostics. This supports future support tooling and improves visibility into background processes.

Tasks

- Track last successful sync timestamps for upload and download per profile.
- Display current sync queue size and oldest unsynced entry.
- Surface recent conflicts or failures with details (entry ID, reason).
- Add refresh and clear logs buttons.
- Restrict dashboard to debug mode or internal builds.

Acceptance Criteria

- [] Dashboard displays accurate timestamps and queue metrics.
- [] Sync conflicts and failures are clearly listed with context.
- [] Refresh updates content without app restart.
- [] Feature only visible in debug builds or via hidden toggle.

Labels

debug, sync, diagnostics

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)

- #1 Set up Clean Architecture Foundation
- #2 Implement Multi-Account User Switching
- #3 Design Unified Log Detail View
- #4 Build Time Range Filter with Presets
- #5 Implement Daily and Rolling Usage Charts
- #6 Sync Logs Between Firestore and Local Isar Cache
- #7 Add Table View for Recent Logs
- #8 Implement Stats Overview Dashboard
- #9 Add Log Entry Submission UI with Validation
- #10 Implement Session Length Histogram Chart
- #11 Implement Inhalation Method Pie Chart
- #12 Build Offline Sync Status Indicator
- #13 Create Onboarding Flow for New Users
- #14 Add Settings Screen with Preferences and About Info
- #15 Implement Export to CSV Functionality
- #16 Enable Tagging of Log Entries
- #17 Add Quick-Log Widget for Fast Entry
- #18 Support Secure Cloud Backup and Restore
- #19 Design Log Correlation Insight Module
- #20 Implement Weekly Summary Notifications
- #21 Integrate Profile-Specific Chart Themes
- #22 Support Advanced Filtering by Tags, Method, and Device
- #23 Add Custom Time Range Analysis with Bookmarking
- #24 Add Device Usage Breakdown Chart
- #25 Build Log Streak Tracker and Display
- #26 Implement Log Deletion and Undo Feature
- #27 Implement In-App Feedback and Issue Reporting
- #28 Add Contextual Log Suggestions Based on Time and Patterns
- #29 Add Sync Conflict Resolution and Audit Logging
- #30 Create Daily Mood + Log Correlation Overlay
- #31 Design Activity Heatmap Calendar View
- #32 Build Session Notes with Markdown Support
- #33 Enable Full-Text Log Search with Highlighting
- #34 Implement Session Timeline View with Zoomable Playback

- #35 Add User Analytics Opt-In and Consent Flow
- #36 Support Multi-Device Session Linking
- #37 Implement Log-Based Reminder System
- #38 Add Data Sync Status and Conflict Reporting Dashboard

⌚ Issue 38 delivered – next run in ~15 min.

markdown

Title

Implement Tag-Based Usage Trends Comparison

Description

Enable users to compare log patterns across different tags (e.g., "anxiety", "sleep") using overlapping charts and summary stats. This supports personal insights into how intent or context affects inhalation behavior.

Tasks

- Create multi-select interface for tags to compare.
- Aggregate log data separately per tag across time.
- Render line or bar chart showing overlapping trends per tag.
- Add stats summary (average duration, frequency) for each selected tag.
- Ensure filters (profile, time range) still apply across comparisons.

Acceptance Criteria

- [] Users can select and compare at least 2-3 tags at once.
- [] Charts update in real time to reflect overlapping usage patterns.
- [] Summary stats are accurate and clearly labeled.
- [] UI is mobile-friendly and collapses gracefully.

Labels

analytics, tags, charts

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #1 Set up Clean Architecture Foundation
- #2 Implement Multi-Account User Switching
- #3 Design Unified Log Detail View
- #4 Build Time Range Filter with Presets
- #5 Implement Daily and Rolling Usage Charts
- #6 Sync Logs Between Firestore and Local Isar Cache
- #7 Add Table View for Recent Logs
- #8 Implement Stats Overview Dashboard
- #9 Add Log Entry Submission UI with Validation
- #10 Implement Session Length Histogram Chart
- #11 Implement Inhalation Method Pie Chart
- #12 Build Offline Sync Status Indicator
- #13 Create Onboarding Flow for New Users
- #14 Add Settings Screen with Preferences and About Info
- #15 Implement Export to CSV Functionality
- #16 Enable Tagging of Log Entries
- #17 Add Quick-Log Widget for Fast Entry
- #18 Support Secure Cloud Backup and Restore
- #19 Design Log Correlation Insight Module
- #20 Implement Weekly Summary Notifications
- #21 Integrate Profile-Specific Chart Themes
- #22 Support Advanced Filtering by Tags, Method, and Device
- #23 Add Custom Time Range Analysis with Bookmarking
- #24 Add Device Usage Breakdown Chart
- #25 Build Log Streak Tracker and Display
- #26 Implement Log Deletion and Undo Feature
- #27 Implement In-App Feedback and Issue Reporting
- #28 Add Contextual Log Suggestions Based on Time and Patterns
- #29 Add Sync Conflict Resolution and Audit Logging
- #30 Create Daily Mood + Log Correlation Overlay
- #31 Design Activity Heatmap Calendar View
- #32 Build Session Notes with Markdown Support

- #33 Enable Full-Text Log Search with Highlighting
- #34 Implement Session Timeline View with Zoomable Playback
- #35 Add User Analytics Opt-In and Consent Flow
- #36 Support Multi-Device Session Linking
- #37 Implement Log-Based Reminder System
- #38 Add Data Sync Status and Conflict Reporting Dashboard
- #39 Implement Tag-Based Usage Trends Comparison

⌚ Issue 39 delivered – next run in ~15 min.

markdown

Title

Implement Session Annotation Timeline Markers

Description

Allow users to mark significant events (e.g., "panic attack", "creative burst") directly on the session timeline. These annotations provide rich context for analysis and future reflection.

Tasks

- Add `Annotation` model with timestamp, label, and optional note.
- Build interface to add annotations during or after session logging.
- Render markers on timeline and chart views.
- Display annotation detail on tap or hover.
- Sync annotations across devices with profile linkage.

Acceptance Criteria

- [] Users can place timestamped annotations on their timeline.
- [] Markers appear on all applicable visualizations with labels.
- [] Tapping a marker reveals its label and note.
- [] Annotations persist across devices and are editable.

Labels

annotations, insights, timeline

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #1 Set up Clean Architecture Foundation
- #2 Implement Multi-Account User Switching
- #3 Design Unified Log Detail View
- #4 Build Time Range Filter with Presets
- #5 Implement Daily and Rolling Usage Charts
- #6 Sync Logs Between Firestore and Local Isar Cache
- #7 Add Table View for Recent Logs
- #8 Implement Stats Overview Dashboard
- #9 Add Log Entry Submission UI with Validation
- #10 Implement Session Length Histogram Chart
- #11 Implement Inhalation Method Pie Chart
- #12 Build Offline Sync Status Indicator
- #13 Create Onboarding Flow for New Users
- #14 Add Settings Screen with Preferences and About Info
- #15 Implement Export to CSV Functionality
- #16 Enable Tagging of Log Entries
- #17 Add Quick-Log Widget for Fast Entry
- #18 Support Secure Cloud Backup and Restore
- #19 Design Log Correlation Insight Module
- #20 Implement Weekly Summary Notifications
- #21 Integrate Profile-Specific Chart Themes
- #22 Support Advanced Filtering by Tags, Method, and Device
- #23 Add Custom Time Range Analysis with Bookmarking
- #24 Add Device Usage Breakdown Chart
- #25 Build Log Streak Tracker and Display
- #26 Implement Log Deletion and Undo Feature
- #27 Implement In-App Feedback and Issue Reporting
- #28 Add Contextual Log Suggestions Based on Time and Patterns

#29 Add Sync Conflict Resolution and Audit Logging
#30 Create Daily Mood + Log Correlation Overlay
#31 Design Activity Heatmap Calendar View
#32 Build Session Notes with Markdown Support
#33 Enable Full-Text Log Search with Highlighting
#34 Implement Session Timeline View with Zoomable Playback
#35 Add User Analytics Opt-In and Consent Flow
#36 Support Multi-Device Session Linking
#37 Implement Log-Based Reminder System
#38 Add Data Sync Status and Conflict Reporting Dashboard
#39 Implement Tag-Based Usage Trends Comparison
#40 Implement Session Annotation Timeline Markers

⌚ Issue 40 delivered – next run in ~15 min.

markdown

Title

Create Session Pattern Clustering and Labeling

Description

Implement an analytics module that automatically identifies recurring session patterns using unsupervised clustering and allows users to label or name these patterns. This feature supports discovering usage trends not tied to explicit tags.

Tasks

- Extract features from logs (duration, time of day, method, device).
- Apply clustering algorithm (e.g., k-means, DBSCAN) to group sessions.
- Assign default labels to clusters (e.g., "Short Nighttime", "Daily Routine").
- Allow users to rename or hide specific clusters.
- Visualize clusters in a new timeline or comparison chart.

Acceptance Criteria

- [] At least 3-5 distinct usage clusters are identified

automatically.

- [] Users can view sessions by pattern in timeline or chart.
- [] Cluster labels are editable and persist with profile.
- [] Updates to log entries recalculate clustering within 24h.

Labels

analytics, machine-learning, insights

Estimate

L

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#1 Set up Clean Architecture Foundation

#2 Implement Multi-Account User Switching

#3 Design Unified Log Detail View

#4 Build Time Range Filter with Presets

#5 Implement Daily and Rolling Usage Charts

#6 Sync Logs Between Firestore and Local Isar Cache

#7 Add Table View for Recent Logs

#8 Implement Stats Overview Dashboard

#9 Add Log Entry Submission UI with Validation

#10 Implement Session Length Histogram Chart

#11 Implement Inhalation Method Pie Chart

#12 Build Offline Sync Status Indicator

#13 Create Onboarding Flow for New Users

#14 Add Settings Screen with Preferences and About Info

#15 Implement Export to CSV Functionality

#16 Enable Tagging of Log Entries

#17 Add Quick-Log Widget for Fast Entry

#18 Support Secure Cloud Backup and Restore

#19 Design Log Correlation Insight Module

#20 Implement Weekly Summary Notifications

#21 Integrate Profile-Specific Chart Themes

#22 Support Advanced Filtering by Tags, Method, and Device

#23 Add Custom Time Range Analysis with Bookmarking
#24 Add Device Usage Breakdown Chart
#25 Build Log Streak Tracker and Display
#26 Implement Log Deletion and Undo Feature
#27 Implement In-App Feedback and Issue Reporting
#28 Add Contextual Log Suggestions Based on Time and Patterns
#29 Add Sync Conflict Resolution and Audit Logging
#30 Create Daily Mood + Log Correlation Overlay
#31 Design Activity Heatmap Calendar View
#32 Build Session Notes with Markdown Support
#33 Enable Full-Text Log Search with Highlighting
#34 Implement Session Timeline View with Zoomable Playback
#35 Add User Analytics Opt-In and Consent Flow
#36 Support Multi-Device Session Linking
#37 Implement Log-Based Reminder System
#38 Add Data Sync Status and Conflict Reporting Dashboard
#39 Implement Tag-Based Usage Trends Comparison
#40 Implement Session Annotation Timeline Markers
#41 Create Session Pattern Clustering and Labeling

⌚ Issue 41 delivered – next run in ~15 min.

markdown

```
### Title
Build Smart Log Suggestions Based on Behavior Patterns

### Description
Create a system that uses prior usage data to recommend likely
logging times or types. These smart suggestions help users
maintain consistency and provide insight-driven nudges based
on behavioral trends.

**Tasks**
- Analyze log timing, frequency, and metadata to detect
patterns.
- Develop algorithm to predict likely next logging windows.
```

- Display non-intrusive suggestions on home/dashboard view.
- Allow users to accept or dismiss suggestions easily.
- Track dismissed suggestions to improve future recommendations.

Acceptance Criteria

- [] Suggestions appear only when confidence exceeds threshold.
- [] Users can tap to log directly from suggestion.
- [] Dismissed suggestions reduce future false positives.
- [] System improves over time with more data.

Labels

ai, behavior, personalization

Estimate

L

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#1 Set up Clean Architecture Foundation

#2 Implement Multi-Account User Switching

#3 Design Unified Log Detail View

#4 Build Time Range Filter with Presets

#5 Implement Daily and Rolling Usage Charts

#6 Sync Logs Between Firestore and Local Isar Cache

#7 Add Table View for Recent Logs

#8 Implement Stats Overview Dashboard

#9 Add Log Entry Submission UI with Validation

#10 Implement Session Length Histogram Chart

#11 Implement Inhalation Method Pie Chart

#12 Build Offline Sync Status Indicator

#13 Create Onboarding Flow for New Users

#14 Add Settings Screen with Preferences and About Info

#15 Implement Export to CSV Functionality

#16 Enable Tagging of Log Entries

- #17 Add Quick-Log Widget for Fast Entry
- #18 Support Secure Cloud Backup and Restore
- #19 Design Log Correlation Insight Module
- #20 Implement Weekly Summary Notifications
- #21 Integrate Profile-Specific Chart Themes
- #22 Support Advanced Filtering by Tags, Method, and Device
- #23 Add Custom Time Range Analysis with Bookmarking
- #24 Add Device Usage Breakdown Chart
- #25 Build Log Streak Tracker and Display
- #26 Implement Log Deletion and Undo Feature
- #27 Implement In-App Feedback and Issue Reporting
- #28 Add Contextual Log Suggestions Based on Time and Patterns
- #29 Add Sync Conflict Resolution and Audit Logging
- #30 Create Daily Mood + Log Correlation Overlay
- #31 Design Activity Heatmap Calendar View
- #32 Build Session Notes with Markdown Support
- #33 Enable Full-Text Log Search with Highlighting
- #34 Implement Session Timeline View with Zoomable Playback
- #35 Add User Analytics Opt-In and Consent Flow
- #36 Support Multi-Device Session Linking
- #37 Implement Log-Based Reminder System
- #38 Add Data Sync Status and Conflict Reporting Dashboard
- #39 Implement Tag-Based Usage Trends Comparison
- #40 Implement Session Annotation Timeline Markers
- #41 Create Session Pattern Clustering and Labeling
- #42 Build Smart Log Suggestions Based on Behavior Patterns

⌚ Issue 42 delivered – next run in ~15 min.

markdown

Title

Add Historical Baseline Overlay for Personal Progress

Description

Introduce a visual overlay showing a user's historical averages (e.g., weekly or monthly usage) to help track improvement or regression over time. This allows users to self-assess progress against their own baselines.

****Tasks****

- Calculate rolling historical averages for selected metrics (session count, total duration).
- Overlay historical lines on charts (e.g., daily usage chart).
- Allow users to toggle baseline visibility.
- Display percent change vs baseline for selected time periods.

Acceptance Criteria

- [] Historical baseline lines render clearly alongside current data.
- [] Toggling baseline overlay works across all applicable charts.
- [] Percent change from baseline is accurate and understandable.
- [] Charts update when filters (tag, profile, range) change.

Labels

analytics, visualization, insights

Estimate

M

◆ RUNNING INDEX (update each run; keep \leq 25 lines)

(index truncated – see earlier messages)

#1 Set up Clean Architecture Foundation

#2 Implement Multi-Account User Switching

#3 Design Unified Log Detail View

#4 Build Time Range Filter with Presets

#5 Implement Daily and Rolling Usage Charts

#6 Sync Logs Between Firestore and Local Isar Cache

#7 Add Table View for Recent Logs

#8 Implement Stats Overview Dashboard

- #9 Add Log Entry Submission UI with Validation
- #10 Implement Session Length Histogram Chart
- #11 Implement Inhalation Method Pie Chart
- #12 Build Offline Sync Status Indicator
- #13 Create Onboarding Flow for New Users
- #14 Add Settings Screen with Preferences and About Info
- #15 Implement Export to CSV Functionality
- #16 Enable Tagging of Log Entries
- #17 Add Quick-Log Widget for Fast Entry
- #18 Support Secure Cloud Backup and Restore
- #19 Design Log Correlation Insight Module
- #20 Implement Weekly Summary Notifications
- #21 Integrate Profile-Specific Chart Themes
- #22 Support Advanced Filtering by Tags, Method, and Device
- #23 Add Custom Time Range Analysis with Bookmarking
- #24 Add Device Usage Breakdown Chart
- #25 Build Log Streak Tracker and Display
- #26 Implement Log Deletion and Undo Feature
- #27 Implement In-App Feedback and Issue Reporting
- #28 Add Contextual Log Suggestions Based on Time and Patterns
- #29 Add Sync Conflict Resolution and Audit Logging
- #30 Create Daily Mood + Log Correlation Overlay
- #31 Design Activity Heatmap Calendar View
- #32 Build Session Notes with Markdown Support
- #33 Enable Full-Text Log Search with Highlighting
- #34 Implement Session Timeline View with Zoomable Playback
- #35 Add User Analytics Opt-In and Consent Flow
- #36 Support Multi-Device Session Linking
- #37 Implement Log-Based Reminder System
- #38 Add Data Sync Status and Conflict Reporting Dashboard
- #39 Implement Tag-Based Usage Trends Comparison
- #40 Implement Session Annotation Timeline Markers
- #41 Create Session Pattern Clustering and Labeling

#42 Build Smart Log Suggestions Based on Behavior Patterns

#43 Add Historical Baseline Overlay for Personal Progress

⌚ Issue 43 delivered – next run in ~15 min.

markdown

Title

Integrate Interactive Session Playback with Visual Highlights

Description

Enable users to replay a session's inhalation pattern as a timeline animation, with highlights on key moments (e.g., peak durations, annotation markers). This feature helps users visually reflect on session dynamics and identify patterns.

Tasks

- Create playback component for session timeline with play/pause controls.
- Highlight peaks or threshold events visually during playback.
- Sync playback with annotations and session metadata display.
- Ensure playback is performant on mobile and older devices.
- Add “Replay Session” button in detailed session view.

Acceptance Criteria

- [] Users can replay any session with animation and highlights.
- [] Peak inhalation points and annotations are emphasized during playback.
- [] UI is responsive and accessible across devices.
- [] Playback respects session time resolution and smoothing settings.

Labels

visualization, timeline, UX

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #1 Set up Clean Architecture Foundation
- #2 Implement Multi-Account User Switching
- #3 Design Unified Log Detail View
- #4 Build Time Range Filter with Presets
- #5 Implement Daily and Rolling Usage Charts
- #6 Sync Logs Between Firestore and Local Isar Cache
- #7 Add Table View for Recent Logs
- #8 Implement Stats Overview Dashboard
- #9 Add Log Entry Submission UI with Validation
- #10 Implement Session Length Histogram Chart
- #11 Implement Inhalation Method Pie Chart
- #12 Build Offline Sync Status Indicator
- #13 Create Onboarding Flow for New Users
- #14 Add Settings Screen with Preferences and About Info
- #15 Implement Export to CSV Functionality
- #16 Enable Tagging of Log Entries
- #17 Add Quick-Log Widget for Fast Entry
- #18 Support Secure Cloud Backup and Restore
- #19 Design Log Correlation Insight Module
- #20 Implement Weekly Summary Notifications
- #21 Integrate Profile-Specific Chart Themes
- #22 Support Advanced Filtering by Tags, Method, and Device
- #23 Add Custom Time Range Analysis with Bookmarking
- #24 Add Device Usage Breakdown Chart
- #25 Build Log Streak Tracker and Display
- #26 Implement Log Deletion and Undo Feature
- #27 Implement In-App Feedback and Issue Reporting
- #28 Add Contextual Log Suggestions Based on Time and Patterns
- #29 Add Sync Conflict Resolution and Audit Logging
- #30 Create Daily Mood + Log Correlation Overlay
- #31 Design Activity Heatmap Calendar View
- #32 Build Session Notes with Markdown Support

#33 Enable Full-Text Log Search with Highlighting
#34 Implement Session Timeline View with Zoomable Playback
#35 Add User Analytics Opt-In and Consent Flow
#36 Support Multi-Device Session Linking
#37 Implement Log-Based Reminder System
#38 Add Data Sync Status and Conflict Reporting Dashboard
#39 Implement Tag-Based Usage Trends Comparison
#40 Implement Session Annotation Timeline Markers
#41 Create Session Pattern Clustering and Labeling
#42 Build Smart Log Suggestions Based on Behavior Patterns
#43 Add Historical Baseline Overlay for Personal Progress
#44 Integrate Interactive Session Playback with Visual Highlights

⌚ Issue 44 delivered – next run in ~15 min.

markdown

Title

Enable User-Created Custom Chart Templates

Description

Allow users to design and save custom chart templates by selecting metrics, time ranges, filters, and visual styles. This feature gives power users the flexibility to create dashboards tailored to their unique usage patterns and analysis needs.

Tasks

- Build UI for selecting chart type, metrics, filters, and visual options.
- Allow saving named chart templates to profile.
- Support editing and deleting saved templates.
- Display saved templates in dashboard section or dropdown selector.
- Ensure compatibility with export and fullscreen modes.

Acceptance Criteria

- [] Users can create, name, and save custom charts with

preferred settings.

- [] Saved templates appear in a dedicated selector.
- [] Users can edit or delete templates.
- [] Custom templates load consistently across sessions and devices.

Labels

customization, charts, dashboard

Estimate

L

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#20 Implement Weekly Summary Notifications

#21 Integrate Profile-Specific Chart Themes

#22 Support Advanced Filtering by Tags, Method, and Device

#23 Add Custom Time Range Analysis with Bookmarking

#24 Add Device Usage Breakdown Chart

#25 Build Log Streak Tracker and Display

#26 Implement Log Deletion and Undo Feature

#27 Implement In-App Feedback and Issue Reporting

#28 Add Contextual Log Suggestions Based on Time and Patterns

#29 Add Sync Conflict Resolution and Audit Logging

#30 Create Daily Mood + Log Correlation Overlay

#31 Design Activity Heatmap Calendar View

#32 Build Session Notes with Markdown Support

#33 Enable Full-Text Log Search with Highlighting

#34 Implement Session Timeline View with Zoomable Playback

#35 Add User Analytics Opt-In and Consent Flow

#36 Support Multi-Device Session Linking

#37 Implement Log-Based Reminder System

#38 Add Data Sync Status and Conflict Reporting Dashboard

#39 Implement Tag-Based Usage Trends Comparison

#40 Implement Session Annotation Timeline Markers

#41 Create Session Pattern Clustering and Labeling

- #42 Build Smart Log Suggestions Based on Behavior Patterns
- #43 Add Historical Baseline Overlay for Personal Progress
- #44 Integrate Interactive Session Playback with Visual Highlights
- #45 Enable User-Created Custom Chart Templates

⌚ Issue 45 delivered – next run in ~15 min.

markdown

Title

Implement Log Comparison Mode for Insight Discovery

Description

Introduce a feature that lets users compare two distinct sets of logs—such as different time periods, profiles, tags, or inhalation methods—side by side. This mode will help users uncover behavioral changes, effectiveness of adjustments, or differences between habits.

Tasks

- Design UI for selecting two log groups for comparison.
- Generate side-by-side or overlaid visualizations.
- Highlight key differences automatically (e.g., average duration, method distribution).
- Support export of comparison results.
- Integrate comparison view with tag and profile filters.

Acceptance Criteria

- [] Users can select any two groups of logs to compare.
- [] Charts display differences clearly and consistently.
- [] Highlighting shows % change and notable contrasts.
- [] All filters and selection states persist during session.

Labels

analysis, comparison, insights

Estimate

L

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #21 Integrate Profile-Specific Chart Themes
- #22 Support Advanced Filtering by Tags, Method, and Device
- #23 Add Custom Time Range Analysis with Bookmarking
- #24 Add Device Usage Breakdown Chart
- #25 Build Log Streak Tracker and Display
- #26 Implement Log Deletion and Undo Feature
- #27 Implement In-App Feedback and Issue Reporting
- #28 Add Contextual Log Suggestions Based on Time and Patterns
- #29 Add Sync Conflict Resolution and Audit Logging
- #30 Create Daily Mood + Log Correlation Overlay
- #31 Design Activity Heatmap Calendar View
- #32 Build Session Notes with Markdown Support
- #33 Enable Full-Text Log Search with Highlighting
- #34 Implement Session Timeline View with Zoomable Playback
- #35 Add User Analytics Opt-In and Consent Flow
- #36 Support Multi-Device Session Linking
- #37 Implement Log-Based Reminder System
- #38 Add Data Sync Status and Conflict Reporting Dashboard
- #39 Implement Tag-Based Usage Trends Comparison
- #40 Implement Session Annotation Timeline Markers
- #41 Create Session Pattern Clustering and Labeling
- #42 Build Smart Log Suggestions Based on Behavior Patterns
- #43 Add Historical Baseline Overlay for Personal Progress
- #44 Integrate Interactive Session Playback with Visual Highlights
- #45 Enable User-Created Custom Chart Templates
- #46 Implement Log Comparison Mode for Insight Discovery

⌚ Issue 46 delivered – next run in ~15 min.

markdown

Title

Add Adaptive Chart Scaling and Zoom Controls

Description

Improve the usability of visualizations by allowing users to zoom in on specific data ranges or scale charts dynamically based on visible data. This feature enhances readability and analysis across varying log densities and time spans.

****Tasks****

- Implement pinch-to-zoom and drag-to-pan gestures for mobile.
- Add zoom/reset buttons for desktop and accessibility.
- Automatically adjust Y-axis based on visible X-range data.
- Maintain performance when interacting with large datasets.
- Ensure scaling does not affect original data fidelity.

Acceptance Criteria

- [] Users can zoom/pan charts with smooth transitions.
- [] Reset button restores default view.
- [] Charts rescale appropriately across filters and devices.
- [] High-density data does not cause lag or jank.

Labels

ui, visualization, performance

Estimate

M

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#22 Support Advanced Filtering by Tags, Method, and Device

#23 Add Custom Time Range Analysis with Bookmarking

#24 Add Device Usage Breakdown Chart

#25 Build Log Streak Tracker and Display

#26 Implement Log Deletion and Undo Feature

#27 Implement In-App Feedback and Issue Reporting

#28 Add Contextual Log Suggestions Based on Time and Patterns

#29 Add Sync Conflict Resolution and Audit Logging

#30 Create Daily Mood + Log Correlation Overlay
#31 Design Activity Heatmap Calendar View
#32 Build Session Notes with Markdown Support
#33 Enable Full-Text Log Search with Highlighting
#34 Implement Session Timeline View with Zoomable Playback
#35 Add User Analytics Opt-In and Consent Flow
#36 Support Multi-Device Session Linking
#37 Implement Log-Based Reminder System
#38 Add Data Sync Status and Conflict Reporting Dashboard
#39 Implement Tag-Based Usage Trends Comparison
#40 Implement Session Annotation Timeline Markers
#41 Create Session Pattern Clustering and Labeling
#42 Build Smart Log Suggestions Based on Behavior Patterns
#43 Add Historical Baseline Overlay for Personal Progress
#44 Integrate Interactive Session Playback with Visual Highlights
#45 Enable User-Created Custom Chart Templates
#46 Implement Log Comparison Mode for Insight Discovery
#47 Add Adaptive Chart Scaling and Zoom Controls

⌚ Issue 47 delivered – next run in ~15 min.

markdown

Title
Create Session Anomaly Detection and Highlighting

Description

Introduce backend and frontend support to automatically detect and flag anomalous sessions—such as outliers in duration, intensity, or pattern—helping users identify potential errors or unexpected behavior.

****Tasks****

- Define rules or thresholds for anomaly detection (e.g., z-score, IQR).
- Implement backend logic or local computation to flag anomalies.

- Display alerts or markers on session charts and log tables.
- Allow user feedback to confirm or dismiss anomalies.
- Log feedback for potential model refinement.

Acceptance Criteria

- [] Anomalies are automatically detected and clearly marked.
- [] Users can see explanations or contributing factors.
- [] Users can confirm or reject anomaly flags.
- [] System tracks feedback without disrupting data.

Labels

data-quality, analysis, insights

Estimate

M

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#23 Add Custom Time Range Analysis with Bookmarking

#24 Add Device Usage Breakdown Chart

#25 Build Log Streak Tracker and Display

#26 Implement Log Deletion and Undo Feature

#27 Implement In-App Feedback and Issue Reporting

#28 Add Contextual Log Suggestions Based on Time and Patterns

#29 Add Sync Conflict Resolution and Audit Logging

#30 Create Daily Mood + Log Correlation Overlay

#31 Design Activity Heatmap Calendar View

#32 Build Session Notes with Markdown Support

#33 Enable Full-Text Log Search with Highlighting

#34 Implement Session Timeline View with Zoomable Playback

#35 Add User Analytics Opt-In and Consent Flow

#36 Support Multi-Device Session Linking

#37 Implement Log-Based Reminder System

#38 Add Data Sync Status and Conflict Reporting Dashboard

#39 Implement Tag-Based Usage Trends Comparison

#40 Implement Session Annotation Timeline Markers

- #41 Create Session Pattern Clustering and Labeling
- #42 Build Smart Log Suggestions Based on Behavior Patterns
- #43 Add Historical Baseline Overlay for Personal Progress
- #44 Integrate Interactive Session Playback with Visual Highlights
- #45 Enable User-Created Custom Chart Templates
- #46 Implement Log Comparison Mode for Insight Discovery
- #47 Add Adaptive Chart Scaling and Zoom Controls
- #48 Create Session Anomaly Detection and Highlighting

⌚ Issue 48 delivered – next run in ~15 min.

markdown

Title

Add Insights Panel for Weekly Behavior Summaries

Description

Introduce a lightweight insights panel that summarizes the past week's log trends, including total usage, peak days, preferred methods, and any notable changes or streaks. This panel offers immediate feedback to encourage consistent self-monitoring.

Tasks

- Design and position weekly insights panel on dashboard.
- Aggregate and display metrics like total usage time, average session length, and most common tags/methods.
- Detect and surface behavioral shifts (e.g., longer sessions, increased frequency).
- Integrate with existing streak tracker and tag analytics.
- Enable dismissible summary view with optional auto-popup on Mondays.

Acceptance Criteria

- [] Weekly panel summarizes key usage and behavior metrics.
- [] Panel appears each week or via manual trigger.
- [] Data is accurate and derived from the most recent 7 days.
- [] Behavioral changes are clearly communicated with visual

cues.

Labels

insights, dashboard, user-feedback

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #24 Add Device Usage Breakdown Chart
- #25 Build Log Streak Tracker and Display
- #26 Implement Log Deletion and Undo Feature
- #27 Implement In-App Feedback and Issue Reporting
- #28 Add Contextual Log Suggestions Based on Time and Patterns
- #29 Add Sync Conflict Resolution and Audit Logging
- #30 Create Daily Mood + Log Correlation Overlay
- #31 Design Activity Heatmap Calendar View
- #32 Build Session Notes with Markdown Support
- #33 Enable Full-Text Log Search with Highlighting
- #34 Implement Session Timeline View with Zoomable Playback
- #35 Add User Analytics Opt-In and Consent Flow
- #36 Support Multi-Device Session Linking
- #37 Implement Log-Based Reminder System
- #38 Add Data Sync Status and Conflict Reporting Dashboard
- #39 Implement Tag-Based Usage Trends Comparison
- #40 Implement Session Annotation Timeline Markers
- #41 Create Session Pattern Clustering and Labeling
- #42 Build Smart Log Suggestions Based on Behavior Patterns
- #43 Add Historical Baseline Overlay for Personal Progress
- #44 Integrate Interactive Session Playback with Visual Highlights
- #45 Enable User-Created Custom Chart Templates
- #46 Implement Log Comparison Mode for Insight Discovery
- #47 Add Adaptive Chart Scaling and Zoom Controls

#48 Create Session Anomaly Detection and Highlighting

#49 Add Insights Panel for Weekly Behavior Summaries

⌚ Issue 49 delivered – next run in ~15 min.

markdown

Title

Implement Smart Filter Presets Based on User Behavior

Description

Enhance the filtering experience by introducing intelligent presets that auto-suggest or remember commonly used filter combinations (e.g., profile + method + range). This streamlines frequent analysis tasks and improves usability over time.

Tasks

- Track recent and frequent filter combinations per user.
- Provide UI to access, apply, and optionally name saved presets.
- Offer “Suggested Preset” section based on usage trends.
- Enable pinning or setting a default preset.
- Sync presets across devices for logged-in users.

Acceptance Criteria

- [] Users see recent and suggested presets when opening filters.
- [] Presets can be named, saved, and pinned for quick access.
- [] Filter combinations persist accurately across sessions.
- [] Suggested presets adapt to user behavior over time.

Labels

ux, filters, personalization

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #25 Build Log Streak Tracker and Display
- #26 Implement Log Deletion and Undo Feature
- #27 Implement In-App Feedback and Issue Reporting
- #28 Add Contextual Log Suggestions Based on Time and Patterns
- #29 Add Sync Conflict Resolution and Audit Logging
- #30 Create Daily Mood + Log Correlation Overlay
- #31 Design Activity Heatmap Calendar View
- #32 Build Session Notes with Markdown Support
- #33 Enable Full-Text Log Search with Highlighting
- #34 Implement Session Timeline View with Zoomable Playback
- #35 Add User Analytics Opt-In and Consent Flow
- #36 Support Multi-Device Session Linking
- #37 Implement Log-Based Reminder System
- #38 Add Data Sync Status and Conflict Reporting Dashboard
- #39 Implement Tag-Based Usage Trends Comparison
- #40 Implement Session Annotation Timeline Markers
- #41 Create Session Pattern Clustering and Labeling
- #42 Build Smart Log Suggestions Based on Behavior Patterns
- #43 Add Historical Baseline Overlay for Personal Progress
- #44 Integrate Interactive Session Playback with Visual Highlights
- #45 Enable User-Created Custom Chart Templates
- #46 Implement Log Comparison Mode for Insight Discovery
- #47 Add Adaptive Chart Scaling and Zoom Controls
- #48 Create Session Anomaly Detection and Highlighting
- #49 Add Insights Panel for Weekly Behavior Summaries
- #50 Implement Smart Filter Presets Based on User Behavior

⌚ Issue 50 delivered – next run in ~15 min.

markdown

Title

Add Rolling Averages and Trend Lines to Charts

Description

Incorporate rolling averages and trend line overlays into key charts to help users visualize longer-term patterns and smooth out daily fluctuations. This aids in identifying consistent shifts in behavior or gradual improvements.

****Tasks****

- Implement rolling average calculations (e.g., 3-day, 7-day).
- Add toggle UI to enable/disable rolling trend overlays.
- Apply consistent visual styling (e.g., dotted line, faded color).
- Ensure compatibility with all chart types and filters.
- Optimize rendering performance for large datasets.

Acceptance Criteria

- [] Users can toggle rolling averages on any time-series chart.
- [] Trend lines update with filters, time ranges, and tags.
- [] At least two smoothing windows are supported.
- [] Performance remains smooth on mobile and desktop.

Labels

visualization, analytics, trends

Estimate

M

◆ RUNNING INDEX (update each run; keep \leq 25 lines)

(index truncated – see earlier messages)

#26 Implement Log Deletion and Undo Feature

#27 Implement In-App Feedback and Issue Reporting

#28 Add Contextual Log Suggestions Based on Time and Patterns

#29 Add Sync Conflict Resolution and Audit Logging

#30 Create Daily Mood + Log Correlation Overlay

#31 Design Activity Heatmap Calendar View

#32 Build Session Notes with Markdown Support
#33 Enable Full-Text Log Search with Highlighting
#34 Implement Session Timeline View with Zoomable Playback
#35 Add User Analytics Opt-In and Consent Flow
#36 Support Multi-Device Session Linking
#37 Implement Log-Based Reminder System
#38 Add Data Sync Status and Conflict Reporting Dashboard
#39 Implement Tag-Based Usage Trends Comparison
#40 Implement Session Annotation Timeline Markers
#41 Create Session Pattern Clustering and Labeling
#42 Build Smart Log Suggestions Based on Behavior Patterns
#43 Add Historical Baseline Overlay for Personal Progress
#44 Integrate Interactive Session Playback with Visual Highlights
#45 Enable User-Created Custom Chart Templates
#46 Implement Log Comparison Mode for Insight Discovery
#47 Add Adaptive Chart Scaling and Zoom Controls
#48 Create Session Anomaly Detection and Highlighting
#49 Add Insights Panel for Weekly Behavior Summaries
#50 Implement Smart Filter Presets Based on User Behavior
#51 Add Rolling Averages and Trend Lines to Charts

⌚ Issue 51 delivered – next run in ~15 min.

markdown

Title

Enable Export of Filtered Logs and Visualizations

Description

Allow users to export log data and charts based on their current filter selections. This supports offline analysis, sharing with healthcare professionals, or archiving personal trends. Export options should include raw data (CSV/JSON) and visual formats (PNG/PDF).

Tasks

- Add export button with options for data and image formats.

- Export respects current filters, time range, and profile.
- Render charts server-side or via canvas for PNG/PDF output.
- Format CSV/JSON with clear field names and timestamps.
- Provide feedback on export success/failure.

Acceptance Criteria

- [] Users can export filtered logs as CSV or JSON.
- [] Users can export current chart view as PNG or PDF.
- [] Exports reflect selected filters and date ranges.
- [] Export works across devices with reliable output.

Labels

export, data, visualization

Estimate

M

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#27 Implement In-App Feedback and Issue Reporting

#28 Add Contextual Log Suggestions Based on Time and Patterns

#29 Add Sync Conflict Resolution and Audit Logging

#30 Create Daily Mood + Log Correlation Overlay

#31 Design Activity Heatmap Calendar View

#32 Build Session Notes with Markdown Support

#33 Enable Full-Text Log Search with Highlighting

#34 Implement Session Timeline View with Zoomable Playback

#35 Add User Analytics Opt-In and Consent Flow

#36 Support Multi-Device Session Linking

#37 Implement Log-Based Reminder System

#38 Add Data Sync Status and Conflict Reporting Dashboard

#39 Implement Tag-Based Usage Trends Comparison

#40 Implement Session Annotation Timeline Markers

#41 Create Session Pattern Clustering and Labeling

#42 Build Smart Log Suggestions Based on Behavior Patterns

#43 Add Historical Baseline Overlay for Personal Progress

- #44 Integrate Interactive Session Playback with Visual Highlights
- #45 Enable User-Created Custom Chart Templates
- #46 Implement Log Comparison Mode for Insight Discovery
- #47 Add Adaptive Chart Scaling and Zoom Controls
- #48 Create Session Anomaly Detection and Highlighting
- #49 Add Insights Panel for Weekly Behavior Summaries
- #50 Implement Smart Filter Presets Based on User Behavior
- #51 Add Rolling Averages and Trend Lines to Charts
- #52 Enable Export of Filtered Logs and Visualizations

⌚ Issue 52 delivered – next run in ~15 min.

markdown

Title

Integrate Log Event Playback with Speed Controls

Description

Build a session playback feature that allows users to replay their log events over a time axis with adjustable speed. This visualization helps users see how their sessions unfold in real time or review past patterns in a dynamic way.

Tasks

- Implement time-based replay of logs on timeline charts.
- Add speed control options (e.g., 1x, 2x, 5x, pause).
- Highlight events as they are "played" on the chart.
- Support user-controlled scrubbing and jumping between events.
- Ensure performance on large datasets with many events.

Acceptance Criteria

- [] Users can initiate and control session playback.
- [] Event highlights sync correctly with playback time.
- [] Speed controls work smoothly across devices.
- [] Playback supports pause, rewind, and skip behavior.

Labels

timeline, playback, ux

Estimate

L

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #28 Add Contextual Log Suggestions Based on Time and Patterns
- #29 Add Sync Conflict Resolution and Audit Logging
- #30 Create Daily Mood + Log Correlation Overlay
- #31 Design Activity Heatmap Calendar View
- #32 Build Session Notes with Markdown Support
- #33 Enable Full-Text Log Search with Highlighting
- #34 Implement Session Timeline View with Zoomable Playback
- #35 Add User Analytics Opt-In and Consent Flow
- #36 Support Multi-Device Session Linking
- #37 Implement Log-Based Reminder System
- #38 Add Data Sync Status and Conflict Reporting Dashboard
- #39 Implement Tag-Based Usage Trends Comparison
- #40 Implement Session Annotation Timeline Markers
- #41 Create Session Pattern Clustering and Labeling
- #42 Build Smart Log Suggestions Based on Behavior Patterns
- #43 Add Historical Baseline Overlay for Personal Progress
- #44 Integrate Interactive Session Playback with Visual Highlights
- #45 Enable User-Created Custom Chart Templates
- #46 Implement Log Comparison Mode for Insight Discovery
- #47 Add Adaptive Chart Scaling and Zoom Controls
- #48 Create Session Anomaly Detection and Highlighting
- #49 Add Insights Panel for Weekly Behavior Summaries
- #50 Implement Smart Filter Presets Based on User Behavior
- #51 Add Rolling Averages and Trend Lines to Charts
- #52 Enable Export of Filtered Logs and Visualizations
- #53 Integrate Log Event Playback with Speed Controls

⌚ Issue 53 delivered – next run in ~15 min.

markdown

Title

Add Session Comparison Across Time Periods

Description

Introduce a comparison mode that allows users to select two time periods (e.g., this week vs. last week, July vs. June) and view side-by-side statistics and visual trends. This empowers users to assess progress or behavioral shifts over time.

Tasks

- Add UI controls for selecting two time ranges.
- Display key metrics and charts side-by-side or overlaid.
- Highlight significant changes (e.g., +20% frequency).
- Support comparison for tags, duration, time of day, and device usage.
- Ensure responsiveness and clarity on all screen sizes.

Acceptance Criteria

- [] Users can select two periods and compare logs.
- [] Metrics and visuals update in a synchronized layout.
- [] Visual indicators highlight changes or anomalies.
- [] Comparison works with filters and user profiles.

Labels

analysis, comparison, trends

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)

#29 Add Sync Conflict Resolution and Audit Logging

#30 Create Daily Mood + Log Correlation Overlay

#31 Design Activity Heatmap Calendar View

#32 Build Session Notes with Markdown Support

- #33 Enable Full-Text Log Search with Highlighting
- #34 Implement Session Timeline View with Zoomable Playback
- #35 Add User Analytics Opt-In and Consent Flow
- #36 Support Multi-Device Session Linking
- #37 Implement Log-Based Reminder System
- #38 Add Data Sync Status and Conflict Reporting Dashboard
- #39 Implement Tag-Based Usage Trends Comparison
- #40 Implement Session Annotation Timeline Markers
- #41 Create Session Pattern Clustering and Labeling
- #42 Build Smart Log Suggestions Based on Behavior Patterns
- #43 Add Historical Baseline Overlay for Personal Progress
- #44 Integrate Interactive Session Playback with Visual Highlights
- #45 Enable User-Created Custom Chart Templates
- #46 Implement Log Comparison Mode for Insight Discovery
- #47 Add Adaptive Chart Scaling and Zoom Controls
- #48 Create Session Anomaly Detection and Highlighting
- #49 Add Insights Panel for Weekly Behavior Summaries
- #50 Implement Smart Filter Presets Based on User Behavior
- #51 Add Rolling Averages and Trend Lines to Charts
- #52 Enable Export of Filtered Logs and Visualizations
- #53 Integrate Log Event Playback with Speed Controls
- #54 Add Session Comparison Across Time Periods

⌚ Issue 54 delivered – next run in ~15 min.

markdown

```
### Title  
Build Session Duration Distribution Chart  
  
### Description  
Implement a distribution (histogram or violin plot) that shows how session durations are spread across different ranges (e.g., under 1 min, 1-5 min, 5-15 min, etc.). This helps users visualize consistency and outliers in their session lengths.
```

****Tasks****

- Group session durations into meaningful buckets.
- Choose appropriate chart type (histogram or violin).
- Support log filtering and time range constraints.
- Label ranges clearly with counts or percentage shares.
- Optimize performance for large log sets.

Acceptance Criteria

- [] Users see a clear distribution of session durations.
- [] Chart reflects current filters and time range.
- [] Outliers and patterns are visually distinguishable.
- [] Distribution is responsive and renders efficiently.

Labels

charts, duration, analytics

Estimate

M

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#30 Create Daily Mood + Log Correlation Overlay

#31 Design Activity Heatmap Calendar View

#32 Build Session Notes with Markdown Support

#33 Enable Full-Text Log Search with Highlighting

#34 Implement Session Timeline View with Zoomable Playback

#35 Add User Analytics Opt-In and Consent Flow

#36 Support Multi-Device Session Linking

#37 Implement Log-Based Reminder System

#38 Add Data Sync Status and Conflict Reporting Dashboard

#39 Implement Tag-Based Usage Trends Comparison

#40 Implement Session Annotation Timeline Markers

#41 Create Session Pattern Clustering and Labeling

#42 Build Smart Log Suggestions Based on Behavior Patterns

#43 Add Historical Baseline Overlay for Personal Progress

#44 Integrate Interactive Session Playback with Visual Highlights

#45 Enable User-Created Custom Chart Templates

- #46 Implement Log Comparison Mode for Insight Discovery
- #47 Add Adaptive Chart Scaling and Zoom Controls
- #48 Create Session Anomaly Detection and Highlighting
- #49 Add Insights Panel for Weekly Behavior Summaries
- #50 Implement Smart Filter Presets Based on User Behavior
- #51 Add Rolling Averages and Trend Lines to Charts
- #52 Enable Export of Filtered Logs and Visualizations
- #53 Integrate Log Event Playback with Speed Controls
- #54 Add Session Comparison Across Time Periods
- #55 Build Session Duration Distribution Chart

⌚ Issue 55 delivered – next run in ~15 min.

markdown

Title

Implement Real-Time Log Capture and Streaming View

Description

Add a real-time streaming view that captures and displays log entries as they are submitted. This live feed supports immediate feedback for users monitoring their behavior in the moment or validating device integration.

Tasks

- Create live-updating UI component for new logs.
- Listen to local or synced Firestore updates.
- Animate incoming logs in the list or timeline view.
- Add pause/resume toggle for performance and focus.
- Ensure seamless behavior with filters and profile switching.

Acceptance Criteria

- [] Live logs appear instantly in the real-time view.
- [] Stream works for both synced and local logs.
- [] UI remains performant during extended usage.
- [] Users can pause and resume the live stream.

Labels

realtime, streaming, ux

Estimate

L

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #31 Design Activity Heatmap Calendar View
- #32 Build Session Notes with Markdown Support
- #33 Enable Full-Text Log Search with Highlighting
- #34 Implement Session Timeline View with Zoomable Playback
- #35 Add User Analytics Opt-In and Consent Flow
- #36 Support Multi-Device Session Linking
- #37 Implement Log-Based Reminder System
- #38 Add Data Sync Status and Conflict Reporting Dashboard
- #39 Implement Tag-Based Usage Trends Comparison
- #40 Implement Session Annotation Timeline Markers
- #41 Create Session Pattern Clustering and Labeling
- #42 Build Smart Log Suggestions Based on Behavior Patterns
- #43 Add Historical Baseline Overlay for Personal Progress
- #44 Integrate Interactive Session Playback with Visual Highlights
- #45 Enable User-Created Custom Chart Templates
- #46 Implement Log Comparison Mode for Insight Discovery
- #47 Add Adaptive Chart Scaling and Zoom Controls
- #48 Create Session Anomaly Detection and Highlighting
- #49 Add Insights Panel for Weekly Behavior Summaries
- #50 Implement Smart Filter Presets Based on User Behavior
- #51 Add Rolling Averages and Trend Lines to Charts
- #52 Enable Export of Filtered Logs and Visualizations
- #53 Integrate Log Event Playback with Speed Controls
- #54 Add Session Comparison Across Time Periods
- #55 Build Session Duration Distribution Chart
- #56 Implement Real-Time Log Capture and Streaming View

⌚ Issue 56 delivered – next run in ~15 min.

markdown

Title

Add Intelligent Tag Recommendations on Log Entry

Description

Enhance the log entry process by suggesting tags based on past usage patterns, time of day, device type, and session length. This improves tagging consistency and user experience while reducing manual effort.

Tasks

- Analyze historical log data to identify tag patterns.
- Implement client-side logic to suggest top 3 tags.
- Integrate tag suggestions into the log submission UI.
- Allow users to quickly accept or dismiss suggestions.
- Update logic as new logs are entered or tags change.

Acceptance Criteria

- [] Suggested tags appear dynamically on log entry screen.
- [] Suggestions are based on user's own log history.
- [] Users can apply, ignore, or modify suggested tags.
- [] Tag engine adapts over time with user behavior.

Labels

tags, ux, personalization

Estimate

M

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#32 Build Session Notes with Markdown Support

#33 Enable Full-Text Log Search with Highlighting

#34 Implement Session Timeline View with Zoomable Playback

#35 Add User Analytics Opt-In and Consent Flow

#36 Support Multi-Device Session Linking

#37 Implement Log-Based Reminder System

#38 Add Data Sync Status and Conflict Reporting Dashboard
#39 Implement Tag-Based Usage Trends Comparison
#40 Implement Session Annotation Timeline Markers
#41 Create Session Pattern Clustering and Labeling
#42 Build Smart Log Suggestions Based on Behavior Patterns
#43 Add Historical Baseline Overlay for Personal Progress
#44 Integrate Interactive Session Playback with Visual Highlights
#45 Enable User-Created Custom Chart Templates
#46 Implement Log Comparison Mode for Insight Discovery
#47 Add Adaptive Chart Scaling and Zoom Controls
#48 Create Session Anomaly Detection and Highlighting
#49 Add Insights Panel for Weekly Behavior Summaries
#50 Implement Smart Filter Presets Based on User Behavior
#51 Add Rolling Averages and Trend Lines to Charts
#52 Enable Export of Filtered Logs and Visualizations
#53 Integrate Log Event Playback with Speed Controls
#54 Add Session Comparison Across Time Periods
#55 Build Session Duration Distribution Chart
#56 Implement Real-Time Log Capture and Streaming View
#57 Add Intelligent Tag Recommendations on Log Entry

⌚ Issue 57 delivered – next run in ~15 min.

markdown

Title

Implement Multi-Variable Log Correlation Matrix

Description

Introduce a visual correlation matrix that allows users to analyze relationships between multiple tracked variables (e.g., session length, time of day, tag frequency). This aids in discovering meaningful patterns and potential triggers.

Tasks

- Define supported variables for correlation (duration, tags, time, method).

- Build matrix view showing correlation strength (heatmap or numeric).
- Enable dynamic selection of variables to compare.
- Add tooltips or click-to-inspect for detailed breakdowns.
- Optimize calculations for larger datasets with caching.

Acceptance Criteria

- [] Users can generate a correlation matrix from selected variables.
- [] Results display clearly with interpretable visual cues.
- [] Matrix updates when filters or profiles change.
- [] Users can explore high or low correlations in detail.

Labels

analytics, insights, charts

Estimate

L

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #33 Enable Full-Text Log Search with Highlighting
#34 Implement Session Timeline View with Zoomable Playback
#35 Add User Analytics Opt-In and Consent Flow
#36 Support Multi-Device Session Linking
#37 Implement Log-Based Reminder System
#38 Add Data Sync Status and Conflict Reporting Dashboard
#39 Implement Tag-Based Usage Trends Comparison
#40 Implement Session Annotation Timeline Markers
#41 Create Session Pattern Clustering and Labeling
#42 Build Smart Log Suggestions Based on Behavior Patterns
#43 Add Historical Baseline Overlay for Personal Progress
#44 Integrate Interactive Session Playback with Visual Highlights
#45 Enable User-Created Custom Chart Templates
#46 Implement Log Comparison Mode for Insight Discovery
#47 Add Adaptive Chart Scaling and Zoom Controls
#48 Create Session Anomaly Detection and Highlighting

- #49 Add Insights Panel for Weekly Behavior Summaries
- #50 Implement Smart Filter Presets Based on User Behavior
- #51 Add Rolling Averages and Trend Lines to Charts
- #52 Enable Export of Filtered Logs and Visualizations
- #53 Integrate Log Event Playback with Speed Controls
- #54 Add Session Comparison Across Time Periods
- #55 Build Session Duration Distribution Chart
- #56 Implement Real-Time Log Capture and Streaming View
- #57 Add Intelligent Tag Recommendations on Log Entry
- #58 Implement Multi-Variable Log Correlation Matrix

⌚ Issue 58 delivered – next run in ~15 min.

markdown

Title

Enable Session Grouping by Custom Time Blocks

Description

Allow users to group and view sessions based on custom time blocks (e.g., morning, afternoon, evening, late night) rather than fixed hours. This improves behavioral insights by matching real user routines instead of arbitrary divisions.

Tasks

- Add preference UI for defining custom time ranges (e.g., 6AM-12PM).
- Implement backend logic to assign logs to the correct block.
- Display grouped summaries in charts and analytics views.
- Update filters and exports to respect new time blocks.
- Validate that overlapping or invalid ranges are handled.

Acceptance Criteria

- [] Users can define and edit named time blocks.
- [] Charts and views reflect groupings accurately.
- [] Grouped analytics adapt when time blocks are updated.
- [] No duplicate or unassigned sessions unless expected.

Labels

ux, analytics, customization

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#34 Implement Session Timeline View with Zoomable Playback

#35 Add User Analytics Opt-In and Consent Flow

#36 Support Multi-Device Session Linking

#37 Implement Log-Based Reminder System

#38 Add Data Sync Status and Conflict Reporting Dashboard

#39 Implement Tag-Based Usage Trends Comparison

#40 Implement Session Annotation Timeline Markers

#41 Create Session Pattern Clustering and Labeling

#42 Build Smart Log Suggestions Based on Behavior Patterns

#43 Add Historical Baseline Overlay for Personal Progress

#44 Integrate Interactive Session Playback with Visual Highlights

#45 Enable User-Created Custom Chart Templates

#46 Implement Log Comparison Mode for Insight Discovery

#47 Add Adaptive Chart Scaling and Zoom Controls

#48 Create Session Anomaly Detection and Highlighting

#49 Add Insights Panel for Weekly Behavior Summaries

#50 Implement Smart Filter Presets Based on User Behavior

#51 Add Rolling Averages and Trend Lines to Charts

#52 Enable Export of Filtered Logs and Visualizations

#53 Integrate Log Event Playback with Speed Controls

#54 Add Session Comparison Across Time Periods

#55 Build Session Duration Distribution Chart

#56 Implement Real-Time Log Capture and Streaming View

#57 Add Intelligent Tag Recommendations on Log Entry

#58 Implement Multi-Variable Log Correlation Matrix

#59 Enable Session Grouping by Custom Time Blocks

⌚ Issue 59 delivered – next run in ~15 min.

markdown

Title

Add Custom Goals and Milestone Tracking

Description

Let users set personalized goals (e.g., max sessions per day, total minutes, or reduced usage streaks) and track progress over time. This empowers self-regulation and builds positive feedback into daily usage.

Tasks

- Define goal types and thresholds (per day/week/month).
- Add UI to set, edit, and remove goals.
- Track milestone streaks or target achievements.
- Display goal progress in dashboard and charts.
- Notify users upon success or missed goals.

Acceptance Criteria

- [] Users can set and edit personalized goals.
- [] Goal tracking updates in real time with logs.
- [] Successes and streaks are highlighted in UI.
- [] Users receive feedback when goals are missed or hit.

Labels

goals, behavior, ux

Estimate

M

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#35 Add User Analytics Opt-In and Consent Flow

#36 Support Multi-Device Session Linking

#37 Implement Log-Based Reminder System

#38 Add Data Sync Status and Conflict Reporting Dashboard

#39 Implement Tag-Based Usage Trends Comparison
#40 Implement Session Annotation Timeline Markers
#41 Create Session Pattern Clustering and Labeling
#42 Build Smart Log Suggestions Based on Behavior Patterns
#43 Add Historical Baseline Overlay for Personal Progress
#44 Integrate Interactive Session Playback with Visual Highlights
#45 Enable User-Created Custom Chart Templates
#46 Implement Log Comparison Mode for Insight Discovery
#47 Add Adaptive Chart Scaling and Zoom Controls
#48 Create Session Anomaly Detection and Highlighting
#49 Add Insights Panel for Weekly Behavior Summaries
#50 Implement Smart Filter Presets Based on User Behavior
#51 Add Rolling Averages and Trend Lines to Charts
#52 Enable Export of Filtered Logs and Visualizations
#53 Integrate Log Event Playback with Speed Controls
#54 Add Session Comparison Across Time Periods
#55 Build Session Duration Distribution Chart
#56 Implement Real-Time Log Capture and Streaming View
#57 Add Intelligent Tag Recommendations on Log Entry
#58 Implement Multi-Variable Log Correlation Matrix
#59 Enable Session Grouping by Custom Time Blocks
#60 Add Custom Goals and Milestone Tracking

⌚ Issue 60 delivered – next run in ~15 min.

markdown

Title

Build Configurable Weekly Report Email System

Description

Provide users with a weekly summary email that highlights key usage trends, achievements, and stats. This passive engagement tool helps users stay aware of their behavior and progress without opening the app.

****Tasks****

- Add settings screen toggle for weekly reports.
- Generate report content based on recent log data and trends.
- Format report as responsive email (HTML + plain text fallback).
- Schedule and send emails via cloud function or backend cron.
- Track delivery status and email opens for analytics.

Acceptance Criteria

- [] Users can enable/disable weekly reports.
- [] Emails reflect actual usage patterns and key metrics.
- [] Emails are visually clean, mobile-friendly, and informative.
- [] Reports are sent consistently and securely on schedule.

Labels

reporting, engagement, notifications

Estimate

M

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#36 Support Multi-Device Session Linking

#37 Implement Log-Based Reminder System

#38 Add Data Sync Status and Conflict Reporting Dashboard

#39 Implement Tag-Based Usage Trends Comparison

#40 Implement Session Annotation Timeline Markers

#41 Create Session Pattern Clustering and Labeling

#42 Build Smart Log Suggestions Based on Behavior Patterns

#43 Add Historical Baseline Overlay for Personal Progress

#44 Integrate Interactive Session Playback with Visual Highlights

#45 Enable User-Created Custom Chart Templates

#46 Implement Log Comparison Mode for Insight Discovery

#47 Add Adaptive Chart Scaling and Zoom Controls

#48 Create Session Anomaly Detection and Highlighting

#49 Add Insights Panel for Weekly Behavior Summaries

#50 Implement Smart Filter Presets Based on User Behavior
#51 Add Rolling Averages and Trend Lines to Charts
#52 Enable Export of Filtered Logs and Visualizations
#53 Integrate Log Event Playback with Speed Controls
#54 Add Session Comparison Across Time Periods
#55 Build Session Duration Distribution Chart
#56 Implement Real-Time Log Capture and Streaming View
#57 Add Intelligent Tag Recommendations on Log Entry
#58 Implement Multi-Variable Log Correlation Matrix
#59 Enable Session Grouping by Custom Time Blocks
#60 Add Custom Goals and Milestone Tracking
#61 Build Configurable Weekly Report Email System

⌚ Issue 61 delivered – next run in ~15 min.

markdown

Title

Implement Context-Aware Smart Entry Defaults

Description

Streamline the log entry experience by pre-filling fields like method, location, and tag suggestions based on recent entries, time of day, and usage history. This reduces repetitive input and improves speed of logging.

Tasks

- Analyze user's recent logs to infer default values.
- Add logic to suggest pre-filled fields when opening log form.
- Allow user to confirm, edit, or disable smart defaults.
- Provide subtle UI indication of auto-filled fields.
- Update defaults as patterns change over time.

Acceptance Criteria

- [] Log entry fields are pre-filled with context-aware defaults.
- [] Users can override or disable smart defaults.

- [] System adapts to behavior changes and updates suggestions.
- [] UX clearly distinguishes between user input and auto-filled values.

Labels

ux, automation, data-intelligence

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #37 Implement Log-Based Reminder System
- #38 Add Data Sync Status and Conflict Reporting Dashboard
- #39 Implement Tag-Based Usage Trends Comparison
- #40 Implement Session Annotation Timeline Markers
- #41 Create Session Pattern Clustering and Labeling
- #42 Build Smart Log Suggestions Based on Behavior Patterns
- #43 Add Historical Baseline Overlay for Personal Progress
- #44 Integrate Interactive Session Playback with Visual Highlights
- #45 Enable User-Created Custom Chart Templates
- #46 Implement Log Comparison Mode for Insight Discovery
- #47 Add Adaptive Chart Scaling and Zoom Controls
- #48 Create Session Anomaly Detection and Highlighting
- #49 Add Insights Panel for Weekly Behavior Summaries
- #50 Implement Smart Filter Presets Based on User Behavior
- #51 Add Rolling Averages and Trend Lines to Charts
- #52 Enable Export of Filtered Logs and Visualizations
- #53 Integrate Log Event Playback with Speed Controls
- #54 Add Session Comparison Across Time Periods
- #55 Build Session Duration Distribution Chart
- #56 Implement Real-Time Log Capture and Streaming View
- #57 Add Intelligent Tag Recommendations on Log Entry
- #58 Implement Multi-Variable Log Correlation Matrix

#59 Enable Session Grouping by Custom Time Blocks
#60 Add Custom Goals and Milestone Tracking
#61 Build Configurable Weekly Report Email System
#62 Implement Context-Aware Smart Entry Defaults

⌚ Issue 62 delivered – next run in ~15 min.

markdown

Title

Support Log Entry Voice Dictation and Transcription

Description

Enhance accessibility and convenience by allowing users to enter logs using voice dictation. The system should transcribe speech into text and populate the log entry form, ideal for mobile and hands-free usage.

Tasks

- Integrate speech-to-text API for supported platforms (e.g., Web Speech API or native mobile SDKs).
- Add microphone icon and toggle to log entry screen.
- Auto-fill description field with transcribed speech.
- Allow real-time preview and user correction before saving.
- Handle errors gracefully with feedback for unsupported browsers/devices.

Acceptance Criteria

- [] Users can record and transcribe log entries using voice input.
- [] Transcribed text is editable before saving the log.
- [] Feature is accessible and unobtrusive for all users.
- [] Speech-to-text errors are handled clearly with fallbacks.

Labels

accessibility, input, mobile

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #38 Add Data Sync Status and Conflict Reporting Dashboard
- #39 Implement Tag-Based Usage Trends Comparison
- #40 Implement Session Annotation Timeline Markers
- #41 Create Session Pattern Clustering and Labeling
- #42 Build Smart Log Suggestions Based on Behavior Patterns
- #43 Add Historical Baseline Overlay for Personal Progress
- #44 Integrate Interactive Session Playback with Visual Highlights
- #45 Enable User-Created Custom Chart Templates
- #46 Implement Log Comparison Mode for Insight Discovery
- #47 Add Adaptive Chart Scaling and Zoom Controls
- #48 Create Session Anomaly Detection and Highlighting
- #49 Add Insights Panel for Weekly Behavior Summaries
- #50 Implement Smart Filter Presets Based on User Behavior
- #51 Add Rolling Averages and Trend Lines to Charts
- #52 Enable Export of Filtered Logs and Visualizations
- #53 Integrate Log Event Playback with Speed Controls
- #54 Add Session Comparison Across Time Periods
- #55 Build Session Duration Distribution Chart
- #56 Implement Real-Time Log Capture and Streaming View
- #57 Add Intelligent Tag Recommendations on Log Entry
- #58 Implement Multi-Variable Log Correlation Matrix
- #59 Enable Session Grouping by Custom Time Blocks
- #60 Add Custom Goals and Milestone Tracking
- #61 Build Configurable Weekly Report Email System
- #62 Implement Context-Aware Smart Entry Defaults
- #63 Support Log Entry Voice Dictation and Transcription

⌚ Issue 63 delivered – next run in ~15 min.

markdown

Title

Create Log Entry Auto-Save Drafts and Recovery

Description

Improve reliability by introducing auto-save functionality for in-progress log entries. This prevents accidental data loss from app crashes, reloads, or navigation errors, particularly on mobile.

Tasks

- Store unsaved log form content locally (e.g., localStorage or IndexedDB).
- Restore drafts if the user returns to the log entry screen.
- Add “discard draft” and “save draft” options.
- Display subtle status when autosaving occurs.
- Handle multiple drafts for different times or sessions.

Acceptance Criteria

- [] Log entries in progress are saved automatically.
- [] Users are prompted to restore drafts if returning mid-entry.
- [] Users can discard, edit, or finalize recovered drafts.
- [] Feature works across devices and does not overwrite saved logs.

Labels

ux, reliability, offline

Estimate

S

- ◆ RUNNING INDEX (update each run; keep \leq 25 lines)

(index truncated – see earlier messages)

#39 Implement Tag-Based Usage Trends Comparison

#40 Implement Session Annotation Timeline Markers

#41 Create Session Pattern Clustering and Labeling

#42 Build Smart Log Suggestions Based on Behavior Patterns

#43 Add Historical Baseline Overlay for Personal Progress

#44 Integrate Interactive Session Playback with Visual Highlights
#45 Enable User-Created Custom Chart Templates
#46 Implement Log Comparison Mode for Insight Discovery
#47 Add Adaptive Chart Scaling and Zoom Controls
#48 Create Session Anomaly Detection and Highlighting
#49 Add Insights Panel for Weekly Behavior Summaries
#50 Implement Smart Filter Presets Based on User Behavior
#51 Add Rolling Averages and Trend Lines to Charts
#52 Enable Export of Filtered Logs and Visualizations
#53 Integrate Log Event Playback with Speed Controls
#54 Add Session Comparison Across Time Periods
#55 Build Session Duration Distribution Chart
#56 Implement Real-Time Log Capture and Streaming View
#57 Add Intelligent Tag Recommendations on Log Entry
#58 Implement Multi-Variable Log Correlation Matrix
#59 Enable Session Grouping by Custom Time Blocks
#60 Add Custom Goals and Milestone Tracking
#61 Build Configurable Weekly Report Email System
#62 Implement Context-Aware Smart Entry Defaults
#63 Support Log Entry Voice Dictation and Transcription
#64 Create Log Entry Auto-Save Drafts and Recovery

⌚ Issue 64 delivered – next run in ~15 min.

markdown

Title

Add Editable Tags and Notes to Past Entries

Description

Let users retroactively update or correct past logs with custom tags and notes. This is essential for reclassification, memory-based updates, or data cleanup after quick entries.

Tasks

- Enable editing of tags and notes for existing logs.

- Add “Edit” button to log detail or history view.
- Track and store changes to maintain edit history.
- Optionally allow users to mark logs as “verified” after edits.
- Sync changes across devices without overwriting unsynced data.

Acceptance Criteria

- [] Users can edit tags and notes of any past log entry.
- [] Changes are saved and reflected across sessions.
- [] Log history clearly indicates modified entries.
- [] Sync logic prevents data loss from simultaneous edits.

Labels

data-cleanup, user-control, history

Estimate

M

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#40 Implement Session Annotation Timeline Markers

#41 Create Session Pattern Clustering and Labeling

#42 Build Smart Log Suggestions Based on Behavior Patterns

#43 Add Historical Baseline Overlay for Personal Progress

#44 Integrate Interactive Session Playback with Visual Highlights

#45 Enable User-Created Custom Chart Templates

#46 Implement Log Comparison Mode for Insight Discovery

#47 Add Adaptive Chart Scaling and Zoom Controls

#48 Create Session Anomaly Detection and Highlighting

#49 Add Insights Panel for Weekly Behavior Summaries

#50 Implement Smart Filter Presets Based on User Behavior

#51 Add Rolling Averages and Trend Lines to Charts

#52 Enable Export of Filtered Logs and Visualizations

#53 Integrate Log Event Playback with Speed Controls

#54 Add Session Comparison Across Time Periods

#55 Build Session Duration Distribution Chart

#56 Implement Real-Time Log Capture and Streaming View
#57 Add Intelligent Tag Recommendations on Log Entry
#58 Implement Multi-Variable Log Correlation Matrix
#59 Enable Session Grouping by Custom Time Blocks
#60 Add Custom Goals and Milestone Tracking
#61 Build Configurable Weekly Report Email System
#62 Implement Context-Aware Smart Entry Defaults
#63 Support Log Entry Voice Dictation and Transcription
#64 Create Log Entry Auto-Save Drafts and Recovery
#65 Add Editable Tags and Notes to Past Entries

⌚ Issue 65 delivered – next run in ~15 min.

markdown

Title

Enable Log Filtering by Environmental Context (e.g., weather, location)

Description

Introduce filtering based on environmental metadata like weather conditions and geolocation to uncover patterns in usage. This helps users understand how external factors influence their behavior.

Tasks

- Capture and store weather and location metadata on log creation (with user permission).
- Add filters for weather type (e.g., sunny, rainy) and general location (e.g., home, outdoors).
- Create UI dropdowns or chips in filter panel.
- Allow combining with other filters (e.g., tag + weather).
- Ensure performance and privacy in storage and queries.

Acceptance Criteria

- [] Weather and location metadata is captured and stored on new logs.
- [] Users can filter logs by weather and/or location

context.

- [] Combined filtering works seamlessly with existing tag/date filters.
- [] Location data is anonymized or generalized for privacy.

Labels

analytics, filters, privacy

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#41 Create Session Pattern Clustering and Labeling

#42 Build Smart Log Suggestions Based on Behavior Patterns

#43 Add Historical Baseline Overlay for Personal Progress

#44 Integrate Interactive Session Playback with Visual Highlights

#45 Enable User-Created Custom Chart Templates

#46 Implement Log Comparison Mode for Insight Discovery

#47 Add Adaptive Chart Scaling and Zoom Controls

#48 Create Session Anomaly Detection and Highlighting

#49 Add Insights Panel for Weekly Behavior Summaries

#50 Implement Smart Filter Presets Based on User Behavior

#51 Add Rolling Averages and Trend Lines to Charts

#52 Enable Export of Filtered Logs and Visualizations

#53 Integrate Log Event Playback with Speed Controls

#54 Add Session Comparison Across Time Periods

#55 Build Session Duration Distribution Chart

#56 Implement Real-Time Log Capture and Streaming View

#57 Add Intelligent Tag Recommendations on Log Entry

#58 Implement Multi-Variable Log Correlation Matrix

#59 Enable Session Grouping by Custom Time Blocks

#60 Add Custom Goals and Milestone Tracking

#61 Build Configurable Weekly Report Email System

#62 Implement Context-Aware Smart Entry Defaults

#63 Support Log Entry Voice Dictation and Transcription
#64 Create Log Entry Auto-Save Drafts and Recovery
#65 Add Editable Tags and Notes to Past Entries
#66 Enable Log Filtering by Environmental Context (e.g., weather, location)

⌚ Issue 66 delivered – next run in ~15 min.

markdown

Title

Design Session Merge Tool for Duplicate or Split Logs

Description

Support users in combining fragmented or duplicate session logs into a single unified entry. This helps improve accuracy when users log overlapping entries or forget to stop a session.

Tasks

- Add UI for selecting multiple entries to merge.
- Provide preview of combined metadata and notes.
- Implement logic to handle conflicting data (e.g., overlapping timestamps or tags).
- Allow user confirmation before final merge.
- Update charts and analysis to reflect merged sessions.

Acceptance Criteria

- [] Users can select and merge duplicate or split sessions.
- [] Conflicts in metadata are resolved with clear user prompts.
- [] Merged entries appear seamlessly in history and visualizations.
- [] No data is lost in the merge process.

Labels

data-cleanup, session-management, ux

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #42 Build Smart Log Suggestions Based on Behavior Patterns
- #43 Add Historical Baseline Overlay for Personal Progress
- #44 Integrate Interactive Session Playback with Visual Highlights
- #45 Enable User-Created Custom Chart Templates
- #46 Implement Log Comparison Mode for Insight Discovery
- #47 Add Adaptive Chart Scaling and Zoom Controls
- #48 Create Session Anomaly Detection and Highlighting
- #49 Add Insights Panel for Weekly Behavior Summaries
- #50 Implement Smart Filter Presets Based on User Behavior
- #51 Add Rolling Averages and Trend Lines to Charts
- #52 Enable Export of Filtered Logs and Visualizations
- #53 Integrate Log Event Playback with Speed Controls
- #54 Add Session Comparison Across Time Periods
- #55 Build Session Duration Distribution Chart
- #56 Implement Real-Time Log Capture and Streaming View
- #57 Add Intelligent Tag Recommendations on Log Entry
- #58 Implement Multi-Variable Log Correlation Matrix
- #59 Enable Session Grouping by Custom Time Blocks
- #60 Add Custom Goals and Milestone Tracking
- #61 Build Configurable Weekly Report Email System
- #62 Implement Context-Aware Smart Entry Defaults
- #63 Support Log Entry Voice Dictation and Transcription
- #64 Create Log Entry Auto-Save Drafts and Recovery
- #65 Add Editable Tags and Notes to Past Entries
- #66 Enable Log Filtering by Environmental Context (e.g., weather, location)
- #67 Design Session Merge Tool for Duplicate or Split Logs

⌚ Issue 67 delivered – next run in ~15 min.

markdown

Title

Implement Smart Duplicate Entry Detection and Alert

Description

Prevent clutter and confusion by detecting when users unintentionally create duplicate logs. The system should flag potential duplicates during entry creation based on timestamp, tags, and content similarity.

Tasks

- Design detection logic to compare new entries against recent logs.
- Alert users if a potential duplicate is detected.
- Provide preview of the matched entry and options: continue, merge, or cancel.
- Tune similarity thresholds for practical flexibility.
- Ensure false positives are minimal and easily dismissible.

Acceptance Criteria

- [] Duplicate detection logic compares timestamp and content similarity.
- [] User is alerted to potential duplicates with actionable choices.
- [] Confirmed duplicates can be merged or discarded.
- [] Alerts do not prevent valid entries from being submitted.

Labels

data-integrity, quality-control, ux

Estimate

M

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#43 Add Historical Baseline Overlay for Personal Progress

#44 Integrate Interactive Session Playback with Visual Highlights

#45 Enable User-Created Custom Chart Templates
#46 Implement Log Comparison Mode for Insight Discovery
#47 Add Adaptive Chart Scaling and Zoom Controls
#48 Create Session Anomaly Detection and Highlighting
#49 Add Insights Panel for Weekly Behavior Summaries
#50 Implement Smart Filter Presets Based on User Behavior
#51 Add Rolling Averages and Trend Lines to Charts
#52 Enable Export of Filtered Logs and Visualizations
#53 Integrate Log Event Playback with Speed Controls
#54 Add Session Comparison Across Time Periods
#55 Build Session Duration Distribution Chart
#56 Implement Real-Time Log Capture and Streaming View
#57 Add Intelligent Tag Recommendations on Log Entry
#58 Implement Multi-Variable Log Correlation Matrix
#59 Enable Session Grouping by Custom Time Blocks
#60 Add Custom Goals and Milestone Tracking
#61 Build Configurable Weekly Report Email System
#62 Implement Context-Aware Smart Entry Defaults
#63 Support Log Entry Voice Dictation and Transcription
#64 Create Log Entry Auto-Save Drafts and Recovery
#65 Add Editable Tags and Notes to Past Entries
#66 Enable Log Filtering by Environmental Context (e.g., weather, location)
#67 Design Session Merge Tool for Duplicate or Split Logs
#68 Implement Smart Duplicate Entry Detection and Alert

⌚ Issue 68 delivered – next run in ~15 min.

markdown

Title

Add Data Quality Report for Incomplete or Inconsistent Logs

Description

Help users maintain clean and useful logs by introducing a

data quality report. This tool will flag missing tags, unusually short or long durations, or sessions lacking contextual notes.

****Tasks****

- Define criteria for incomplete or inconsistent log entries.
- Create a report view listing flagged entries with explanations.
- Add quick actions to edit or dismiss issues directly from the report.
- Optionally provide summary score or progress bar for overall data quality.

Acceptance Criteria

- [] Users can view a list of flagged logs based on quality checks.
- [] Each entry includes a reason and suggested fix.
- [] Editing or dismissing flagged logs updates the report in real time.
- [] Users see a summary score of log completeness.

Labels

data-quality, user-support, reporting

Estimate

M

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#44 Integrate Interactive Session Playback with Visual Highlights

#45 Enable User-Created Custom Chart Templates

#46 Implement Log Comparison Mode for Insight Discovery

#47 Add Adaptive Chart Scaling and Zoom Controls

#48 Create Session Anomaly Detection and Highlighting

#49 Add Insights Panel for Weekly Behavior Summaries

#50 Implement Smart Filter Presets Based on User Behavior

#51 Add Rolling Averages and Trend Lines to Charts

#52 Enable Export of Filtered Logs and Visualizations

#53 Integrate Log Event Playback with Speed Controls

#54 Add Session Comparison Across Time Periods
#55 Build Session Duration Distribution Chart
#56 Implement Real-Time Log Capture and Streaming View
#57 Add Intelligent Tag Recommendations on Log Entry
#58 Implement Multi-Variable Log Correlation Matrix
#59 Enable Session Grouping by Custom Time Blocks
#60 Add Custom Goals and Milestone Tracking
#61 Build Configurable Weekly Report Email System
#62 Implement Context-Aware Smart Entry Defaults
#63 Support Log Entry Voice Dictation and Transcription
#64 Create Log Entry Auto-Save Drafts and Recovery
#65 Add Editable Tags and Notes to Past Entries
#66 Enable Log Filtering by Environmental Context (e.g., weather, location)
#67 Design Session Merge Tool for Duplicate or Split Logs
#68 Implement Smart Duplicate Entry Detection and Alert
#69 Add Data Quality Report for Incomplete or Inconsistent Logs

⌚ Issue 69 delivered – next run in ~15 min.

markdown

Title

Support Timeline Bookmarks for Quick Navigation and Review

Description

Add bookmarks to session timelines to help users flag meaningful moments for later review. These bookmarks will improve navigation and allow users to reflect on important segments of their logs.

Tasks

- Add UI control for placing bookmarks during or after a session.
- Store timestamped bookmarks with optional labels or notes.
- Display bookmarks as markers on timeline visualizations.
- Enable filtering or jumping to bookmarks in playback or

review mode.

- Sync bookmark data across devices.

Acceptance Criteria

- [] Users can add and label bookmarks to session timelines.
- [] Bookmarks are saved with timestamp and optional note.
- [] Bookmarks appear on timeline and are clickable.
- [] Bookmarks sync properly and are included in exports.

Labels

timeline, bookmarks, playback

Estimate

M

◆ RUNNING INDEX (update each run; keep \leq 25 lines)

(index truncated – see earlier messages)

#45 Enable User-Created Custom Chart Templates

#46 Implement Log Comparison Mode for Insight Discovery

#47 Add Adaptive Chart Scaling and Zoom Controls

#48 Create Session Anomaly Detection and Highlighting

#49 Add Insights Panel for Weekly Behavior Summaries

#50 Implement Smart Filter Presets Based on User Behavior

#51 Add Rolling Averages and Trend Lines to Charts

#52 Enable Export of Filtered Logs and Visualizations

#53 Integrate Log Event Playback with Speed Controls

#54 Add Session Comparison Across Time Periods

#55 Build Session Duration Distribution Chart

#56 Implement Real-Time Log Capture and Streaming View

#57 Add Intelligent Tag Recommendations on Log Entry

#58 Implement Multi-Variable Log Correlation Matrix

#59 Enable Session Grouping by Custom Time Blocks

#60 Add Custom Goals and Milestone Tracking

#61 Build Configurable Weekly Report Email System

#62 Implement Context-Aware Smart Entry Defaults

#63 Support Log Entry Voice Dictation and Transcription

- #64 Create Log Entry Auto-Save Drafts and Recovery
- #65 Add Editable Tags and Notes to Past Entries
- #66 Enable Log Filtering by Environmental Context (e.g., weather, location)
- #67 Design Session Merge Tool for Duplicate or Split Logs
- #68 Implement Smart Duplicate Entry Detection and Alert
- #69 Add Data Quality Report for Incomplete or Inconsistent Logs
- #70 Support Timeline Bookmarks for Quick Navigation and Review

⌚ Issue 70 delivered – next run in ~15 min.

markdown

Title

Integrate Smart Suggestions for Log Notes Based on Context

Description

Boost logging efficiency by suggesting note content based on time, tags, prior entries, and environmental context. This helps users maintain detailed logs with minimal friction while still personalizing them.

Tasks

- Build a suggestion engine using recent entries, tags, and context data (e.g., weather, location).
- Generate 1-3 note suggestions at the time of log entry.
- Allow users to accept, edit, or ignore suggestions.
- Ensure suggestions are lightweight and privacy-conscious.
- Tune system to avoid repetitive or generic suggestions.

Acceptance Criteria

- [] Users see 1-3 relevant note suggestions on log creation.
- [] Suggestions reflect contextual factors and past user entries.
- [] Users can edit or ignore suggestions without disruption.
- [] Suggestion engine handles privacy-sensitive data responsibly.

Labels

smart-features, notes, context-aware

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #46 Implement Log Comparison Mode for Insight Discovery
- #47 Add Adaptive Chart Scaling and Zoom Controls
- #48 Create Session Anomaly Detection and Highlighting
- #49 Add Insights Panel for Weekly Behavior Summaries
- #50 Implement Smart Filter Presets Based on User Behavior
- #51 Add Rolling Averages and Trend Lines to Charts
- #52 Enable Export of Filtered Logs and Visualizations
- #53 Integrate Log Event Playback with Speed Controls
- #54 Add Session Comparison Across Time Periods
- #55 Build Session Duration Distribution Chart
- #56 Implement Real-Time Log Capture and Streaming View
- #57 Add Intelligent Tag Recommendations on Log Entry
- #58 Implement Multi-Variable Log Correlation Matrix
- #59 Enable Session Grouping by Custom Time Blocks
- #60 Add Custom Goals and Milestone Tracking
- #61 Build Configurable Weekly Report Email System
- #62 Implement Context-Aware Smart Entry Defaults
- #63 Support Log Entry Voice Dictation and Transcription
- #64 Create Log Entry Auto-Save Drafts and Recovery
- #65 Add Editable Tags and Notes to Past Entries
- #66 Enable Log Filtering by Environmental Context (e.g., weather, location)
- #67 Design Session Merge Tool for Duplicate or Split Logs
- #68 Implement Smart Duplicate Entry Detection and Alert
- #69 Add Data Quality Report for Incomplete or Inconsistent Logs
- #70 Support Timeline Bookmarks for Quick Navigation and Review
- #71 Integrate Smart Suggestions for Log Notes Based on Context

⌚ Issue 71 delivered – next run in ~15 min.

markdown

Title

Build In-App Feedback Form with Optional Log Sharing

Description

Create an in-app feedback form to let users report issues, request features, or share praise. Include the option to attach relevant logs or screenshots to provide better context for support and improvement.

Tasks

- Design feedback form with rating, message, and category fields.
- Allow optional attachment of recent logs or screenshots.
- Route submissions to a connected backend or email endpoint.
- Include metadata like app version, platform, and timestamp.
- Add UI confirmation and thank-you message post-submission.

Acceptance Criteria

- [] Users can submit feedback directly in the app.
- [] Users can optionally attach logs or screenshots.
- [] Feedback includes metadata for context.
- [] Form is easy to access and intuitive to use.

Labels

support, user-feedback, ux

Estimate

S

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#47 Add Adaptive Chart Scaling and Zoom Controls

#48 Create Session Anomaly Detection and Highlighting

#49 Add Insights Panel for Weekly Behavior Summaries

#50 Implement Smart Filter Presets Based on User Behavior

- #51 Add Rolling Averages and Trend Lines to Charts
- #52 Enable Export of Filtered Logs and Visualizations
- #53 Integrate Log Event Playback with Speed Controls
- #54 Add Session Comparison Across Time Periods
- #55 Build Session Duration Distribution Chart
- #56 Implement Real-Time Log Capture and Streaming View
- #57 Add Intelligent Tag Recommendations on Log Entry
- #58 Implement Multi-Variable Log Correlation Matrix
- #59 Enable Session Grouping by Custom Time Blocks
- #60 Add Custom Goals and Milestone Tracking
- #61 Build Configurable Weekly Report Email System
- #62 Implement Context-Aware Smart Entry Defaults
- #63 Support Log Entry Voice Dictation and Transcription
- #64 Create Log Entry Auto-Save Drafts and Recovery
- #65 Add Editable Tags and Notes to Past Entries
- #66 Enable Log Filtering by Environmental Context (e.g., weather, location)
- #67 Design Session Merge Tool for Duplicate or Split Logs
- #68 Implement Smart Duplicate Entry Detection and Alert
- #69 Add Data Quality Report for Incomplete or Inconsistent Logs
- #70 Support Timeline Bookmarks for Quick Navigation and Review
- #71 Integrate Smart Suggestions for Log Notes Based on Context
- #72 Build In-App Feedback Form with Optional Log Sharing

 Issue 72 delivered – next run in ~15 min.

markdown

Title

Implement Advanced Filtering by Tag, Method, and Duration

Description

Enhance data exploration with multi-dimensional filtering capabilities. Users should be able to narrow down logs using combinations of tags, inhalation method, session length, and

other properties, with filters persisting across views.

Tasks

- Create reusable filter UI components (tag, method, range sliders).
- Integrate filters with log list, charts, and table views.
- Support multi-select and compound logic (AND/OR).
- Store active filters in state and allow clearing/reset.
- Ensure performance is optimized for large datasets.

Acceptance Criteria

- [] Users can filter by multiple dimensions at once.
- [] Filters apply to all visual and tabular components.
- [] Filters are clearable and persist across navigation.
- [] Filtering is performant even with many log entries.

Labels

filters, ux, data-exploration

Estimate

L

◆ RUNNING INDEX (update each run; keep \leq 25 lines)

(index truncated – see earlier messages)

#48 Create Session Anomaly Detection and Highlighting

#49 Add Insights Panel for Weekly Behavior Summaries

#50 Implement Smart Filter Presets Based on User Behavior

#51 Add Rolling Averages and Trend Lines to Charts

#52 Enable Export of Filtered Logs and Visualizations

#53 Integrate Log Event Playback with Speed Controls

#54 Add Session Comparison Across Time Periods

#55 Build Session Duration Distribution Chart

#56 Implement Real-Time Log Capture and Streaming View

#57 Add Intelligent Tag Recommendations on Log Entry

#58 Implement Multi-Variable Log Correlation Matrix

#59 Enable Session Grouping by Custom Time Blocks

#60 Add Custom Goals and Milestone Tracking

#61 Build Configurable Weekly Report Email System

#62 Implement Context-Aware Smart Entry Defaults
#63 Support Log Entry Voice Dictation and Transcription
#64 Create Log Entry Auto-Save Drafts and Recovery
#65 Add Editable Tags and Notes to Past Entries
#66 Enable Log Filtering by Environmental Context (e.g., weather, location)
#67 Design Session Merge Tool for Duplicate or Split Logs
#68 Implement Smart Duplicate Entry Detection and Alert
#69 Add Data Quality Report for Incomplete or Inconsistent Logs
#70 Support Timeline Bookmarks for Quick Navigation and Review
#71 Integrate Smart Suggestions for Log Notes Based on Context
#72 Build In-App Feedback Form with Optional Log Sharing
#73 Implement Advanced Filtering by Tag, Method, and Duration

⌚ Issue 73 delivered – next run in ~15 min.

markdown

Title

Support Manual Log Entry Backdating with Visual Indicators

Description

Enable users to input logs retroactively by specifying a custom timestamp, and visually distinguish these from real-time logs. This feature ensures accurate tracking even when users forget to log immediately.

Tasks

- Add date and time picker to log entry form (behind toggle).
- Store manual timestamp in Firestore and Isar.
- Mark manually backdated logs with a visual icon or label.
- Adjust sorting and charts to respect custom timestamps.
- Handle edge cases like future dates or conflicting sessions.

Acceptance Criteria

- [] Users can toggle a timestamp override and select a past date/time.
- [] Manual entries are saved and reflected in all views.

- [] A visual indicator clearly marks backdated logs.
- [] Charts and filters respect the manual timestamp.

Labels

logging, timestamps, ui

Estimate

M

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#49 Add Insights Panel for Weekly Behavior Summaries

#50 Implement Smart Filter Presets Based on User Behavior

#51 Add Rolling Averages and Trend Lines to Charts

#52 Enable Export of Filtered Logs and Visualizations

#53 Integrate Log Event Playback with Speed Controls

#54 Add Session Comparison Across Time Periods

#55 Build Session Duration Distribution Chart

#56 Implement Real-Time Log Capture and Streaming View

#57 Add Intelligent Tag Recommendations on Log Entry

#58 Implement Multi-Variable Log Correlation Matrix

#59 Enable Session Grouping by Custom Time Blocks

#60 Add Custom Goals and Milestone Tracking

#61 Build Configurable Weekly Report Email System

#62 Implement Context-Aware Smart Entry Defaults

#63 Support Log Entry Voice Dictation and Transcription

#64 Create Log Entry Auto-Save Drafts and Recovery

#65 Add Editable Tags and Notes to Past Entries

#66 Enable Log Filtering by Environmental Context (e.g., weather, location)

#67 Design Session Merge Tool for Duplicate or Split Logs

#68 Implement Smart Duplicate Entry Detection and Alert

#69 Add Data Quality Report for Incomplete or Inconsistent Logs

#70 Support Timeline Bookmarks for Quick Navigation and Review

#71 Integrate Smart Suggestions for Log Notes Based on Context

#72 Build In-App Feedback Form with Optional Log Sharing
#73 Implement Advanced Filtering by Tag, Method, and Duration
#74 Support Manual Log Entry Backdating with Visual Indicators

⌚ Issue 74 delivered – next run in ~15 min.

markdown

Title

Add Biometric Authentication for Secure App Access

Description

Enhance privacy and security by implementing optional biometric authentication (e.g., Face ID, fingerprint) for opening the app. This is especially useful for users logging sensitive or personal data.

Tasks

- Integrate biometric check using `local_auth` (*Flutter plugin*).
- Add *settings toggle* to enable/disable biometric lock.
- Show prompt on app open if enabled, with fallback to PIN or passcode.
- Ensure Lock state persists across app restarts.
- Design fail-safe and fallback UX for unsupported devices.

Acceptance Criteria

- [] Users can enable biometric lock from app settings.
- [] App prompts for authentication on launch if lock is active.
- [] App handles unsupported or failed biometric attempts gracefully.
- [] Lock state persists across sessions and app restarts.

Labels

security, user-settings, ux

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #50 Implement Smart Filter Presets Based on User Behavior
- #51 Add Rolling Averages and Trend Lines to Charts
- #52 Enable Export of Filtered Logs and Visualizations
- #53 Integrate Log Event Playback with Speed Controls
- #54 Add Session Comparison Across Time Periods
- #55 Build Session Duration Distribution Chart
- #56 Implement Real-Time Log Capture and Streaming View
- #57 Add Intelligent Tag Recommendations on Log Entry
- #58 Implement Multi-Variable Log Correlation Matrix
- #59 Enable Session Grouping by Custom Time Blocks
- #60 Add Custom Goals and Milestone Tracking
- #61 Build Configurable Weekly Report Email System
- #62 Implement Context-Aware Smart Entry Defaults
- #63 Support Log Entry Voice Dictation and Transcription
- #64 Create Log Entry Auto-Save Drafts and Recovery
- #65 Add Editable Tags and Notes to Past Entries
- #66 Enable Log Filtering by Environmental Context (e.g., weather, location)
- #67 Design Session Merge Tool for Duplicate or Split Logs
- #68 Implement Smart Duplicate Entry Detection and Alert
- #69 Add Data Quality Report for Incomplete or Inconsistent Logs
- #70 Support Timeline Bookmarks for Quick Navigation and Review
- #71 Integrate Smart Suggestions for Log Notes Based on Context
- #72 Build In-App Feedback Form with Optional Log Sharing
- #73 Implement Advanced Filtering by Tag, Method, and Duration
- #74 Support Manual Log Entry Backdating with Visual Indicators
- #75 Add Biometric Authentication for Secure App Access

⌚ Issue 75 delivered – next run in ~15 min.

markdown

Title

Add Timeline Scrubber for Fast Navigation Through Logs

Description

Introduce an interactive timeline scrubber to allow users to quickly scroll through historical data and jump to specific dates or sessions. This tool will make reviewing long-term usage patterns significantly faster.

****Tasks****

- Design horizontal scrubber UI with markers for activity density.
- Allow dragging and snapping to days or sessions.
- Show preview tooltip on hover or drag.
- Integrate with charts and log view to update in real time.
- Optimize for large datasets with virtual scrolling or pagination.

Acceptance Criteria

- [] Users can drag scrubber to navigate through logs.
- [] UI reflects density of sessions over time.
- [] Charts and tables update instantly when date range changes.
- [] Scrubber performs smoothly with many entries.

Labels

navigation, ux, timeline

Estimate

L

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#51 Add Rolling Averages and Trend Lines to Charts

#52 Enable Export of Filtered Logs and Visualizations

#53 Integrate Log Event Playback with Speed Controls

#54 Add Session Comparison Across Time Periods

#55 Build Session Duration Distribution Chart

#56 Implement Real-Time Log Capture and Streaming View

- #57 Add Intelligent Tag Recommendations on Log Entry
- #58 Implement Multi-Variable Log Correlation Matrix
- #59 Enable Session Grouping by Custom Time Blocks
- #60 Add Custom Goals and Milestone Tracking
- #61 Build Configurable Weekly Report Email System
- #62 Implement Context-Aware Smart Entry Defaults
- #63 Support Log Entry Voice Dictation and Transcription
- #64 Create Log Entry Auto-Save Drafts and Recovery
- #65 Add Editable Tags and Notes to Past Entries
- #66 Enable Log Filtering by Environmental Context (e.g., weather, location)
- #67 Design Session Merge Tool for Duplicate or Split Logs
- #68 Implement Smart Duplicate Entry Detection and Alert
- #69 Add Data Quality Report for Incomplete or Inconsistent Logs
- #70 Support Timeline Bookmarks for Quick Navigation and Review
- #71 Integrate Smart Suggestions for Log Notes Based on Context
- #72 Build In-App Feedback Form with Optional Log Sharing
- #73 Implement Advanced Filtering by Tag, Method, and Duration
- #74 Support Manual Log Entry Backdating with Visual Indicators
- #75 Add Biometric Authentication for Secure App Access
- #76 Add Timeline Scrubber for Fast Navigation Through Logs

 Issue 76 delivered – next run in ~15 min.

markdown

Title

Implement Data Sync Conflict Resolution Strategy

Description

Ensure consistent and reliable data integrity by implementing a strategy to resolve conflicts between local Isar data and Firestore records, especially during offline edits and delayed syncs.

****Tasks****

- Identify common sync conflict scenarios (e.g., concurrent edits, deletes).
- Define resolution rules (timestamp-based, user-confirmed, etc.).
- Implement merge logic in sync layer.
- Log and surface conflicts to user when necessary.
- Write test cases for each conflict scenario.

Acceptance Criteria

- [] All identified conflict types are handled gracefully.
- [] Sync layer applies resolution logic consistently.
- [] User is notified of significant changes or overwrite risks.
- [] No data is lost during sync due to unhandled conflicts.

Labels

sync, architecture, data-integrity

Estimate

L

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#52 Enable Export of Filtered Logs and Visualizations

#53 Integrate Log Event Playback with Speed Controls

#54 Add Session Comparison Across Time Periods

#55 Build Session Duration Distribution Chart

#56 Implement Real-Time Log Capture and Streaming View

#57 Add Intelligent Tag Recommendations on Log Entry

#58 Implement Multi-Variable Log Correlation Matrix

#59 Enable Session Grouping by Custom Time Blocks

#60 Add Custom Goals and Milestone Tracking

#61 Build Configurable Weekly Report Email System

#62 Implement Context-Aware Smart Entry Defaults

#63 Support Log Entry Voice Dictation and Transcription

#64 Create Log Entry Auto-Save Drafts and Recovery

#65 Add Editable Tags and Notes to Past Entries

#66 Enable Log Filtering by Environmental Context (e.g., weather, location)
#67 Design Session Merge Tool for Duplicate or Split Logs
#68 Implement Smart Duplicate Entry Detection and Alert
#69 Add Data Quality Report for Incomplete or Inconsistent Logs
#70 Support Timeline Bookmarks for Quick Navigation and Review
#71 Integrate Smart Suggestions for Log Notes Based on Context
#72 Build In-App Feedback Form with Optional Log Sharing
#73 Implement Advanced Filtering by Tag, Method, and Duration
#74 Support Manual Log Entry Backdating with Visual Indicators
#75 Add Biometric Authentication for Secure App Access
#76 Add Timeline Scrubber for Fast Navigation Through Logs
#77 Implement Data Sync Conflict Resolution Strategy

⌚ Issue 77 delivered – next run in ~15 min.

markdown

Title

Implement Streak Tracking and Calendar View

Description

Gamify and encourage consistent logging habits by displaying streak counts and a calendar heatmap of daily usage. This helps users build awareness and motivation for sustained behavior.

Tasks

- Track consecutive logging days and calculate current streak.
- Store longest streak as a user metric.
- Design visual calendar heatmap showing daily log activity.
- Integrate calendar with time filters and log detail jump.
- Display streak info prominently in dashboard or sidebar.

Acceptance Criteria

- [] Users can view current and longest streak.
- [] Calendar heatmap shows clear daily usage patterns.
- [] Missing days break streak as expected.

- [] Clicking calendar days filters logs or opens detail view.

Labels

habit-tracking, visualization, gamification

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #53 Integrate Log Event Playback with Speed Controls
- #54 Add Session Comparison Across Time Periods
- #55 Build Session Duration Distribution Chart
- #56 Implement Real-Time Log Capture and Streaming View
- #57 Add Intelligent Tag Recommendations on Log Entry
- #58 Implement Multi-Variable Log Correlation Matrix
- #59 Enable Session Grouping by Custom Time Blocks
- #60 Add Custom Goals and Milestone Tracking
- #61 Build Configurable Weekly Report Email System
- #62 Implement Context-Aware Smart Entry Defaults
- #63 Support Log Entry Voice Dictation and Transcription
- #64 Create Log Entry Auto-Save Drafts and Recovery
- #65 Add Editable Tags and Notes to Past Entries
- #66 Enable Log Filtering by Environmental Context (e.g., weather, location)
- #67 Design Session Merge Tool for Duplicate or Split Logs
- #68 Implement Smart Duplicate Entry Detection and Alert
- #69 Add Data Quality Report for Incomplete or Inconsistent Logs
- #70 Support Timeline Bookmarks for Quick Navigation and Review
- #71 Integrate Smart Suggestions for Log Notes Based on Context
- #72 Build In-App Feedback Form with Optional Log Sharing
- #73 Implement Advanced Filtering by Tag, Method, and Duration
- #74 Support Manual Log Entry Backdating with Visual Indicators
- #75 Add Biometric Authentication for Secure App Access

#76 Add Timeline Scrubber for Fast Navigation Through Logs

#77 Implement Data Sync Conflict Resolution Strategy

#78 Implement Streak Tracking and Calendar View

⌚ Issue 78 delivered – next run in ~15 min.

markdown

Title

Add Insight Generator for Notable Patterns and Anomalies

Description

Implement an automatic insight generator that detects and surfaces notable trends, such as unusual frequency spikes, long streaks, or sudden drops in usage. This helps users gain meaningful understanding from their logs.

Tasks

- Define rules and thresholds for detecting spikes, drops, and other anomalies.
- Build background job to scan logs and generate insights periodically.
- Display insights in a sidebar or dedicated dashboard section.
- Allow dismissal or archiving of past insights.
- Include timestamps and source links to related sessions.

Acceptance Criteria

- [] Insights are generated automatically and appear on user dashboard.
- [] Rules detect meaningful patterns with minimal false positives.
- [] Users can dismiss or archive insights after viewing.
- [] Clicking an insight navigates to related logs or charts.

Labels

insights, automation, analytics

Estimate

L

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #54 Add Session Comparison Across Time Periods
- #55 Build Session Duration Distribution Chart
- #56 Implement Real-Time Log Capture and Streaming View
- #57 Add Intelligent Tag Recommendations on Log Entry
- #58 Implement Multi-Variable Log Correlation Matrix
- #59 Enable Session Grouping by Custom Time Blocks
- #60 Add Custom Goals and Milestone Tracking
- #61 Build Configurable Weekly Report Email System
- #62 Implement Context-Aware Smart Entry Defaults
- #63 Support Log Entry Voice Dictation and Transcription
- #64 Create Log Entry Auto-Save Drafts and Recovery
- #65 Add Editable Tags and Notes to Past Entries
- #66 Enable Log Filtering by Environmental Context (e.g., weather, location)
- #67 Design Session Merge Tool for Duplicate or Split Logs
- #68 Implement Smart Duplicate Entry Detection and Alert
- #69 Add Data Quality Report for Incomplete or Inconsistent Logs
- #70 Support Timeline Bookmarks for Quick Navigation and Review
- #71 Integrate Smart Suggestions for Log Notes Based on Context
- #72 Build In-App Feedback Form with Optional Log Sharing
- #73 Implement Advanced Filtering by Tag, Method, and Duration
- #74 Support Manual Log Entry Backdating with Visual Indicators
- #75 Add Biometric Authentication for Secure App Access
- #76 Add Timeline Scrubber for Fast Navigation Through Logs
- #77 Implement Data Sync Conflict Resolution Strategy
- #78 Implement Streak Tracking and Calendar View
- #79 Add Insight Generator for Notable Patterns and Anomalies

⌚ Issue 79 delivered – next run in ~15 min.

markdown

Title

Add Rolling Weekly Averages and Comparison Insights

Description

Help users understand changes in behavior over time by calculating and displaying rolling weekly averages of usage frequency and session length. Compare these averages to the prior week to surface subtle improvements or regressions.

Tasks

- Compute 7-day rolling average for session count and average length.
- Compare this week's averages to last week's and flag changes.
- Display comparison metrics with arrows or colored indicators.
- Integrate into dashboard and chart overlays.
- Ensure efficient performance on large datasets.

Acceptance Criteria

- [] Weekly rolling averages are calculated correctly and update as new data arrives.
- [] Visual comparison (e.g., +10%, ↓3 mins) is intuitive and clear.
- [] Integration does not cause performance degradation.
- [] Metrics are accessible from both dashboard and charts.

Labels

analytics, insights, performance

Estimate

M

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#55 Build Session Duration Distribution Chart

#56 Implement Real-Time Log Capture and Streaming View

- #57 Add Intelligent Tag Recommendations on Log Entry
- #58 Implement Multi-Variable Log Correlation Matrix
- #59 Enable Session Grouping by Custom Time Blocks
- #60 Add Custom Goals and Milestone Tracking
- #61 Build Configurable Weekly Report Email System
- #62 Implement Context-Aware Smart Entry Defaults
- #63 Support Log Entry Voice Dictation and Transcription
- #64 Create Log Entry Auto-Save Drafts and Recovery
- #65 Add Editable Tags and Notes to Past Entries
- #66 Enable Log Filtering by Environmental Context (e.g., weather, location)
- #67 Design Session Merge Tool for Duplicate or Split Logs
- #68 Implement Smart Duplicate Entry Detection and Alert
- #69 Add Data Quality Report for Incomplete or Inconsistent Logs
- #70 Support Timeline Bookmarks for Quick Navigation and Review
- #71 Integrate Smart Suggestions for Log Notes Based on Context
- #72 Build In-App Feedback Form with Optional Log Sharing
- #73 Implement Advanced Filtering by Tag, Method, and Duration
- #74 Support Manual Log Entry Backdating with Visual Indicators
- #75 Add Biometric Authentication for Secure App Access
- #76 Add Timeline Scrubber for Fast Navigation Through Logs
- #77 Implement Data Sync Conflict Resolution Strategy
- #78 Implement Streak Tracking and Calendar View
- #79 Add Insight Generator for Notable Patterns and Anomalies
- #80 Add Rolling Weekly Averages and Comparison Insights

⌚ Issue 80 delivered – next run in ~15 min.

markdown

Title

Enable Session Tag Cloud and Frequency Visualization

Description

Visualize the distribution and popularity of tags used across

sessions by implementing a dynamic tag cloud. This provides insight into behavioral trends and highlights dominant usage themes.

****Tasks****

- Aggregate tag usage across all log sessions.
- Generate a responsive tag cloud where size represents frequency.
- Allow users to click tags to filter logs or generate reports.
- Integrate tag cloud into dashboard or analytics view.
- Optimize for performance with large tag sets.

Acceptance Criteria

- [] Tag cloud displays correctly with size based on usage frequency.
- [] Clicking a tag filters logs to sessions containing that tag.
- [] Visual performance remains smooth with 100+ tags.
- [] Cloud updates dynamically when new logs are added or removed.

Labels

visualization, tags, analytics

Estimate

M

◆ RUNNING INDEX (update each run; keep \leq 25 lines)

(index truncated – see earlier messages)

#56 Implement Real-Time Log Capture and Streaming View

#57 Add Intelligent Tag Recommendations on Log Entry

#58 Implement Multi-Variable Log Correlation Matrix

#59 Enable Session Grouping by Custom Time Blocks

#60 Add Custom Goals and Milestone Tracking

#61 Build Configurable Weekly Report Email System

#62 Implement Context-Aware Smart Entry Defaults

#63 Support Log Entry Voice Dictation and Transcription

#64 Create Log Entry Auto-Save Drafts and Recovery

#65 Add Editable Tags and Notes to Past Entries
#66 Enable Log Filtering by Environmental Context (e.g., weather, location)
#67 Design Session Merge Tool for Duplicate or Split Logs
#68 Implement Smart Duplicate Entry Detection and Alert
#69 Add Data Quality Report for Incomplete or Inconsistent Logs
#70 Support Timeline Bookmarks for Quick Navigation and Review
#71 Integrate Smart Suggestions for Log Notes Based on Context
#72 Build In-App Feedback Form with Optional Log Sharing
#73 Implement Advanced Filtering by Tag, Method, and Duration
#74 Support Manual Log Entry Backdating with Visual Indicators
#75 Add Biometric Authentication for Secure App Access
#76 Add Timeline Scrubber for Fast Navigation Through Logs
#77 Implement Data Sync Conflict Resolution Strategy
#78 Implement Streak Tracking and Calendar View
#79 Add Insight Generator for Notable Patterns and Anomalies
#80 Add Rolling Weekly Averages and Comparison Insights
#81 Enable Session Tag Cloud and Frequency Visualization

⌚ Issue 81 delivered – next run in ~15 min.

markdown

```
### Title
Implement Log Entry Clustering and Pattern Recognition

### Description
Group similar log entries into clusters to reveal recurring behavioral patterns and simplify analysis. This enables users to explore common sessions, triggers, and outcomes across time.

**Tasks**
- Develop backend logic to identify clusters using key features (tags, duration, time, notes).
- Label clusters with representative tag or summary.
- Add frontend visualization of clusters (e.g., list or chart)
```

form).

- Allow filtering or highlighting of sessions by cluster.
- Update dynamically as new data is added.

Acceptance Criteria

- [] Logs are clustered based on defined behavioral patterns.
- [] Cluster labels are meaningful and update with new logs.
- [] Users can view, filter, and explore clusters from the UI.
- [] System handles incremental log updates without full reprocessing.

Labels

machine-learning, insights, clustering

Estimate

L

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#57 Add Intelligent Tag Recommendations on Log Entry

#58 Implement Multi-Variable Log Correlation Matrix

#59 Enable Session Grouping by Custom Time Blocks

#60 Add Custom Goals and Milestone Tracking

#61 Build Configurable Weekly Report Email System

#62 Implement Context-Aware Smart Entry Defaults

#63 Support Log Entry Voice Dictation and Transcription

#64 Create Log Entry Auto-Save Drafts and Recovery

#65 Add Editable Tags and Notes to Past Entries

#66 Enable Log Filtering by Environmental Context (e.g., weather, location)

#67 Design Session Merge Tool for Duplicate or Split Logs

#68 Implement Smart Duplicate Entry Detection and Alert

#69 Add Data Quality Report for Incomplete or Inconsistent Logs

#70 Support Timeline Bookmarks for Quick Navigation and Review

#71 Integrate Smart Suggestions for Log Notes Based on Context

#72 Build In-App Feedback Form with Optional Log Sharing
#73 Implement Advanced Filtering by Tag, Method, and Duration
#74 Support Manual Log Entry Backdating with Visual Indicators
#75 Add Biometric Authentication for Secure App Access
#76 Add Timeline Scrubber for Fast Navigation Through Logs
#77 Implement Data Sync Conflict Resolution Strategy
#78 Implement Streak Tracking and Calendar View
#79 Add Insight Generator for Notable Patterns and Anomalies
#80 Add Rolling Weekly Averages and Comparison Insights
#81 Enable Session Tag Cloud and Frequency Visualization
#82 Implement Log Entry Clustering and Pattern Recognition

⌚ Issue 82 delivered – next run in ~15 min.

markdown

Title

Add Session Tags Timeline Chart with Trends Overlay

Description

Provide a timeline visualization that plots the frequency of each tag over time, allowing users to observe evolving trends in their behavior and usage. This chart helps identify long-term patterns tied to specific tags.

Tasks

- Aggregate tag frequency data across all sessions by week or month.
- Build a multi-line timeline chart with each line representing a tag.
- Allow toggling tags on/off to reduce visual clutter.
- Overlay optional trend lines (e.g., rolling average).
- Enable filtering by time range and tag group.

Acceptance Criteria

- [] Tag frequencies are visualized over time as distinct lines.
- [] Users can toggle visibility of specific tags.

- [] Trend lines and rolling averages can be enabled optionally.
- [] Chart is responsive and performs well with large datasets.

Labels

visualization, analytics, trends

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #58 Implement Multi-Variable Log Correlation Matrix
- #59 Enable Session Grouping by Custom Time Blocks
- #60 Add Custom Goals and Milestone Tracking
- #61 Build Configurable Weekly Report Email System
- #62 Implement Context-Aware Smart Entry Defaults
- #63 Support Log Entry Voice Dictation and Transcription
- #64 Create Log Entry Auto-Save Drafts and Recovery
- #65 Add Editable Tags and Notes to Past Entries
- #66 Enable Log Filtering by Environmental Context (e.g., weather, location)
- #67 Design Session Merge Tool for Duplicate or Split Logs
- #68 Implement Smart Duplicate Entry Detection and Alert
- #69 Add Data Quality Report for Incomplete or Inconsistent Logs
- #70 Support Timeline Bookmarks for Quick Navigation and Review
- #71 Integrate Smart Suggestions for Log Notes Based on Context
- #72 Build In-App Feedback Form with Optional Log Sharing
- #73 Implement Advanced Filtering by Tag, Method, and Duration
- #74 Support Manual Log Entry Backdating with Visual Indicators
- #75 Add Biometric Authentication for Secure App Access
- #76 Add Timeline Scrubber for Fast Navigation Through Logs
- #77 Implement Data Sync Conflict Resolution Strategy
- #78 Implement Streak Tracking and Calendar View

- #79 Add Insight Generator for Notable Patterns and Anomalies
- #80 Add Rolling Weekly Averages and Comparison Insights
- #81 Enable Session Tag Cloud and Frequency Visualization
- #82 Implement Log Entry Clustering and Pattern Recognition
- #83 Add Session Tags Timeline Chart with Trends Overlay

⌚ Issue 83 delivered – next run in ~15 min.

markdown

Title

Support Per-Session Mood and Contextual Notes Tags

Description

Allow users to attach mood and context-specific notes (e.g., stress level, social setting, weather) to individual sessions for deeper insight into usage drivers. This will enhance personalization and analytics.

Tasks

- Design data model for storing mood/context alongside session data.
- Update log entry UI to include quick-select options (e.g., emoji or preset moods).
- Enable custom context notes (e.g., “after argument”, “rainy day”).
- Display this metadata in timeline and session detail views.
- Incorporate mood/context data into analytics features where relevant.

Acceptance Criteria

- [] Users can assign mood and context to each log entry.
- [] Metadata is visible on timeline and session detail view.
- [] Custom notes are supported alongside preset options.
- [] Feature integrates with existing filtering and analysis tools.

Labels

user-experience, metadata, analytics

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #59 Enable Session Grouping by Custom Time Blocks
- #60 Add Custom Goals and Milestone Tracking
- #61 Build Configurable Weekly Report Email System
- #62 Implement Context-Aware Smart Entry Defaults
- #63 Support Log Entry Voice Dictation and Transcription
- #64 Create Log Entry Auto-Save Drafts and Recovery
- #65 Add Editable Tags and Notes to Past Entries
- #66 Enable Log Filtering by Environmental Context (e.g., weather, location)
- #67 Design Session Merge Tool for Duplicate or Split Logs
- #68 Implement Smart Duplicate Entry Detection and Alert
- #69 Add Data Quality Report for Incomplete or Inconsistent Logs
- #70 Support Timeline Bookmarks for Quick Navigation and Review
- #71 Integrate Smart Suggestions for Log Notes Based on Context
- #72 Build In-App Feedback Form with Optional Log Sharing
- #73 Implement Advanced Filtering by Tag, Method, and Duration
- #74 Support Manual Log Entry Backdating with Visual Indicators
- #75 Add Biometric Authentication for Secure App Access
- #76 Add Timeline Scrubber for Fast Navigation Through Logs
- #77 Implement Data Sync Conflict Resolution Strategy
- #78 Implement Streak Tracking and Calendar View
- #79 Add Insight Generator for Notable Patterns and Anomalies
- #80 Add Rolling Weekly Averages and Comparison Insights
- #81 Enable Session Tag Cloud and Frequency Visualization
- #82 Implement Log Entry Clustering and Pattern Recognition
- #83 Add Session Tags Timeline Chart with Trends Overlay
- #84 Support Per-Session Mood and Contextual Notes Tags

⌚ Issue 84 delivered – next run in ~15 min.

markdown

Title

Implement Adaptive Reminders Based on Usage Trends

Description

Introduce smart reminders that adapt to the user's logging habits—encouraging consistent use while avoiding unnecessary prompts. This improves user retention and supports behavioral awareness.

Tasks

- Analyze logging frequency to detect habitual times and gaps.
- Implement adaptive logic for reminder timing (e.g., missed routine).
- Provide settings for user-defined boundaries (quiet hours, max/day).
- Notify users through push or local alerts with custom messaging.
- Visualize reminder effectiveness and logging response rate.

Acceptance Criteria

- [] System detects and adapts to user's logging routines.
- [] Reminders trigger based on trend analysis and user settings.
- [] User can adjust frequency, timing windows, and message content.
- [] Reminder feature improves engagement without overwhelming the user.

Labels

engagement, notifications, machine-learning

Estimate

L

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#60 Add Custom Goals and Milestone Tracking

#61 Build Configurable Weekly Report Email System
#62 Implement Context-Aware Smart Entry Defaults
#63 Support Log Entry Voice Dictation and Transcription
#64 Create Log Entry Auto-Save Drafts and Recovery
#65 Add Editable Tags and Notes to Past Entries
#66 Enable Log Filtering by Environmental Context (e.g., weather, location)
#67 Design Session Merge Tool for Duplicate or Split Logs
#68 Implement Smart Duplicate Entry Detection and Alert
#69 Add Data Quality Report for Incomplete or Inconsistent Logs
#70 Support Timeline Bookmarks for Quick Navigation and Review
#71 Integrate Smart Suggestions for Log Notes Based on Context
#72 Build In-App Feedback Form with Optional Log Sharing
#73 Implement Advanced Filtering by Tag, Method, and Duration
#74 Support Manual Log Entry Backdating with Visual Indicators
#75 Add Biometric Authentication for Secure App Access
#76 Add Timeline Scrubber for Fast Navigation Through Logs
#77 Implement Data Sync Conflict Resolution Strategy
#78 Implement Streak Tracking and Calendar View
#79 Add Insight Generator for Notable Patterns and Anomalies
#80 Add Rolling Weekly Averages and Comparison Insights
#81 Enable Session Tag Cloud and Frequency Visualization
#82 Implement Log Entry Clustering and Pattern Recognition
#83 Add Session Tags Timeline Chart with Trends Overlay
#84 Support Per-Session Mood and Contextual Notes Tags
#85 Implement Adaptive Reminders Based on Usage Trends

⌚ Issue 85 delivered – next run in ~15 min.

markdown

Title

Design Modular Export System for Log Data

Description

Develop a flexible export system allowing users to download their log data in multiple formats (e.g., CSV, JSON, PDF) with customizable filters and views. This enables data ownership and offline analysis.

****Tasks****

- Create backend export engine supporting multiple formats.
- Add frontend UI for export options: date range, tags, formats.
- Enable custom views (e.g., grouped by tag, by week, by intensity).
- Ensure exports are secure and respect privacy settings.
- Optimize performance for large datasets.

Acceptance Criteria

- [] Users can export log data in CSV, JSON, and PDF formats.
- [] Export filters and views can be customized before download.
- [] Export respects user privacy settings and access controls.
- [] Exports complete efficiently even with thousands of entries.

Labels

data-export, user-tools, privacy

Estimate

L

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#61 Build Configurable Weekly Report Email System

#62 Implement Context-Aware Smart Entry Defaults

#63 Support Log Entry Voice Dictation and Transcription

#64 Create Log Entry Auto-Save Drafts and Recovery

#65 Add Editable Tags and Notes to Past Entries

#66 Enable Log Filtering by Environmental Context (e.g., weather, location)

#67 Design Session Merge Tool for Duplicate or Split Logs

#68 Implement Smart Duplicate Entry Detection and Alert
#69 Add Data Quality Report for Incomplete or Inconsistent Logs
#70 Support Timeline Bookmarks for Quick Navigation and Review
#71 Integrate Smart Suggestions for Log Notes Based on Context
#72 Build In-App Feedback Form with Optional Log Sharing
#73 Implement Advanced Filtering by Tag, Method, and Duration
#74 Support Manual Log Entry Backdating with Visual Indicators
#75 Add Biometric Authentication for Secure App Access
#76 Add Timeline Scrubber for Fast Navigation Through Logs
#77 Implement Data Sync Conflict Resolution Strategy
#78 Implement Streak Tracking and Calendar View
#79 Add Insight Generator for Notable Patterns and Anomalies
#80 Add Rolling Weekly Averages and Comparison Insights
#81 Enable Session Tag Cloud and Frequency Visualization
#82 Implement Log Entry Clustering and Pattern Recognition
#83 Add Session Tags Timeline Chart with Trends Overlay
#84 Support Per-Session Mood and Contextual Notes Tags
#85 Implement Adaptive Reminders Based on Usage Trends
#86 Design Modular Export System for Log Data

⌚ Issue 86 delivered – next run in ~15 min.

markdown

```
### Title
Build Interactive Timeline Heatmap for Session Activity

### Description
Create a visual heatmap that shows user session density over time, allowing users to identify peak usage periods, habits, and changes. This improves user awareness and supports habit tracking.

**Tasks**
- Design heatmap visualization by hour/day/week (customizable).
- Retrieve and aggregate log frequency for selected time
```

spans.

- Allow users to toggle views (e.g., daily vs. weekly trends).
- Integrate with filters for tags, methods, or moods.
- Ensure responsiveness and usability on both desktop and mobile.

Acceptance Criteria

- [] Heatmap displays session intensity over time at multiple granularities.
- [] Users can filter the heatmap by tags, methods, and mood/context.
- [] Data refreshes dynamically with log changes.
- [] Mobile and desktop layouts are optimized.

Labels

analytics, visualization, timeline

Estimate

L

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#62 Implement Context-Aware Smart Entry Defaults

#63 Support Log Entry Voice Dictation and Transcription

#64 Create Log Entry Auto-Save Drafts and Recovery

#65 Add Editable Tags and Notes to Past Entries

#66 Enable Log Filtering by Environmental Context (e.g., weather, location)

#67 Design Session Merge Tool for Duplicate or Split Logs

#68 Implement Smart Duplicate Entry Detection and Alert

#69 Add Data Quality Report for Incomplete or Inconsistent Logs

#70 Support Timeline Bookmarks for Quick Navigation and Review

#71 Integrate Smart Suggestions for Log Notes Based on Context

#72 Build In-App Feedback Form with Optional Log Sharing

#73 Implement Advanced Filtering by Tag, Method, and Duration

#74 Support Manual Log Entry Backdating with Visual Indicators

#75 Add Biometric Authentication for Secure App Access

- #76 Add Timeline Scrubber for Fast Navigation Through Logs
- #77 Implement Data Sync Conflict Resolution Strategy
- #78 Implement Streak Tracking and Calendar View
- #79 Add Insight Generator for Notable Patterns and Anomalies
- #80 Add Rolling Weekly Averages and Comparison Insights
- #81 Enable Session Tag Cloud and Frequency Visualization
- #82 Implement Log Entry Clustering and Pattern Recognition
- #83 Add Session Tags Timeline Chart with Trends Overlay
- #84 Support Per-Session Mood and Contextual Notes Tags
- #85 Implement Adaptive Reminders Based on Usage Trends
- #86 Design Modular Export System for Log Data
- #87 Build Interactive Timeline Heatmap for Session Activity

⌚ Issue 87 delivered – next run in ~15 min.

markdown

Title

Implement Log Entry Undo and Revision History

Description

Enable users to undo recent changes and view previous versions of log entries to reduce friction and improve confidence in editing. This feature supports better data integrity and recoverability.

Tasks

- Track changes to log entries and maintain a revision history.
- Allow undo/redo functionality within a session.
- Add UI element to view and restore previous versions.
- Ensure performance is not degraded with many revisions.
- Secure revision data against tampering or accidental loss.

Acceptance Criteria

- [] Users can undo and redo log entry changes within the current session.
- [] Revision history is accessible from each log entry.

- [] Previous versions can be previewed and restored.
- [] Feature performs well with large numbers of entries and edits.

Labels

editing, ux, data-integrity

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #63 Support Log Entry Voice Dictation and Transcription
- #64 Create Log Entry Auto-Save Drafts and Recovery
- #65 Add Editable Tags and Notes to Past Entries
- #66 Enable Log Filtering by Environmental Context (e.g., weather, location)
- #67 Design Session Merge Tool for Duplicate or Split Logs
- #68 Implement Smart Duplicate Entry Detection and Alert
- #69 Add Data Quality Report for Incomplete or Inconsistent Logs
- #70 Support Timeline Bookmarks for Quick Navigation and Review
- #71 Integrate Smart Suggestions for Log Notes Based on Context
- #72 Build In-App Feedback Form with Optional Log Sharing
- #73 Implement Advanced Filtering by Tag, Method, and Duration
- #74 Support Manual Log Entry Backdating with Visual Indicators
- #75 Add Biometric Authentication for Secure App Access
- #76 Add Timeline Scrubber for Fast Navigation Through Logs
- #77 Implement Data Sync Conflict Resolution Strategy
- #78 Implement Streak Tracking and Calendar View
- #79 Add Insight Generator for Notable Patterns and Anomalies
- #80 Add Rolling Weekly Averages and Comparison Insights
- #81 Enable Session Tag Cloud and Frequency Visualization
- #82 Implement Log Entry Clustering and Pattern Recognition
- #83 Add Session Tags Timeline Chart with Trends Overlay
- #84 Support Per-Session Mood and Contextual Notes Tags

- #85 Implement Adaptive Reminders Based on Usage Trends
- #86 Design Modular Export System for Log Data
- #87 Build Interactive Timeline Heatmap for Session Activity
- #88 Implement Log Entry Undo and Revision History

⌚ Issue 88 delivered – next run in ~15 min.

markdown

Title

Create Guided Onboarding with Interactive Feature Tour

Description

Design a first-time user onboarding flow that introduces key features through an interactive, contextual tour. This increases user retention and ensures early understanding of core functionality.

Tasks

- Define critical features to showcase during onboarding (e.g., logging, timeline, filters).
- Build onboarding modal or overlay that appears on first launch or reset.
- Implement step-by-step guide with tooltips and navigation controls.
- Provide option to replay onboarding from settings.
- Track completion metrics to measure onboarding effectiveness.

Acceptance Criteria

- [] First-time users are guided through a feature tour on launch.
- [] The tour highlights major functions with contextual help.
- [] Tour can be dismissed or replayed later via settings.
- [] Tour does not block or interfere with normal usage.

Labels

ux, onboarding, engagement

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #64 Create Log Entry Auto-Save Drafts and Recovery
- #65 Add Editable Tags and Notes to Past Entries
- #66 Enable Log Filtering by Environmental Context (e.g., weather, location)
- #67 Design Session Merge Tool for Duplicate or Split Logs
- #68 Implement Smart Duplicate Entry Detection and Alert
- #69 Add Data Quality Report for Incomplete or Inconsistent Logs
- #70 Support Timeline Bookmarks for Quick Navigation and Review
- #71 Integrate Smart Suggestions for Log Notes Based on Context
- #72 Build In-App Feedback Form with Optional Log Sharing
- #73 Implement Advanced Filtering by Tag, Method, and Duration
- #74 Support Manual Log Entry Backdating with Visual Indicators
- #75 Add Biometric Authentication for Secure App Access
- #76 Add Timeline Scrubber for Fast Navigation Through Logs
- #77 Implement Data Sync Conflict Resolution Strategy
- #78 Implement Streak Tracking and Calendar View
- #79 Add Insight Generator for Notable Patterns and Anomalies
- #80 Add Rolling Weekly Averages and Comparison Insights
- #81 Enable Session Tag Cloud and Frequency Visualization
- #82 Implement Log Entry Clustering and Pattern Recognition
- #83 Add Session Tags Timeline Chart with Trends Overlay
- #84 Support Per-Session Mood and Contextual Notes Tags
- #85 Implement Adaptive Reminders Based on Usage Trends
- #86 Design Modular Export System for Log Data
- #87 Build Interactive Timeline Heatmap for Session Activity
- #88 Implement Log Entry Undo and Revision History
- #89 Create Guided Onboarding with Interactive Feature Tour

⌚ Issue 89 delivered – next run in ~15 min.

markdown

Title

Integrate Contextual Log Suggestions via Local Analysis

Description

Develop a lightweight, on-device engine that analyzes recent logs to suggest contextually relevant tags, methods, or notes during new entry creation. This enhances usability and speeds up the logging process.

Tasks

- Analyze past log data for tag and method usage patterns.
- Build suggestion engine to run locally with no network calls.
- Integrate suggestion dropdown into the log entry UI.
- Tune relevance ranking based on time, frequency, and recency.
- Provide fallback logic for new users with minimal history.

Acceptance Criteria

- [] Suggestions appear contextually during new log creation.
- [] Suggestions reflect historical patterns of the specific user.
- [] No external calls or cloud computation required.
- [] Logging remains performant with suggestions enabled.

Labels

intelligence, ux, logging

Estimate

L

- ◆ RUNNING INDEX (update each run; keep \leq 25 lines)

(index truncated – see earlier messages)

#65 Add Editable Tags and Notes to Past Entries

#66 Enable Log Filtering by Environmental Context (e.g., weather, location)

#67 Design Session Merge Tool for Duplicate or Split Logs
#68 Implement Smart Duplicate Entry Detection and Alert
#69 Add Data Quality Report for Incomplete or Inconsistent Logs
#70 Support Timeline Bookmarks for Quick Navigation and Review
#71 Integrate Smart Suggestions for Log Notes Based on Context
#72 Build In-App Feedback Form with Optional Log Sharing
#73 Implement Advanced Filtering by Tag, Method, and Duration
#74 Support Manual Log Entry Backdating with Visual Indicators
#75 Add Biometric Authentication for Secure App Access
#76 Add Timeline Scrubber for Fast Navigation Through Logs
#77 Implement Data Sync Conflict Resolution Strategy
#78 Implement Streak Tracking and Calendar View
#79 Add Insight Generator for Notable Patterns and Anomalies
#80 Add Rolling Weekly Averages and Comparison Insights
#81 Enable Session Tag Cloud and Frequency Visualization
#82 Implement Log Entry Clustering and Pattern Recognition
#83 Add Session Tags Timeline Chart with Trends Overlay
#84 Support Per-Session Mood and Contextual Notes Tags
#85 Implement Adaptive Reminders Based on Usage Trends
#86 Design Modular Export System for Log Data
#87 Build Interactive Timeline Heatmap for Session Activity
#88 Implement Log Entry Undo and Revision History
#89 Create Guided Onboarding with Interactive Feature Tour
#90 Integrate Contextual Log Suggestions via Local Analysis

⌚ Issue 90 delivered – next run in ~15 min.

markdown

Title

Support Cross-Device Session Continuity and Sync Awareness

Description

Implement robust support for session continuity across multiple devices, including detection of out-of-sync states

and recovery options. This is critical for users switching between phone, tablet, or desktop and ensures data coherence.

****Tasks****

- Detect and resolve conflicting edits from different devices.
- Track last sync timestamp and device ID per session.
- Notify users of unsynced or conflicted data with resolution options.
- Add manual sync trigger and visual sync status indicator.
- Ensure offline edits are cached and synced reliably when reconnected.

Acceptance Criteria

- [] Users are notified of sync conflicts with clear resolution paths.
- [] Sync status and last update timestamps are visible and accurate.
- [] Manual sync and auto-recovery work reliably after reconnection.
- [] Device-specific metadata is recorded and used to track changes .

Labels

sync, multi-device, data-integrity

Estimate

L

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#66 Enable Log Filtering by Environmental Context (e.g., weather, location)

#67 Design Session Merge Tool for Duplicate or Split Logs

#68 Implement Smart Duplicate Entry Detection and Alert

#69 Add Data Quality Report for Incomplete or Inconsistent Logs

#70 Support Timeline Bookmarks for Quick Navigation and Review

#71 Integrate Smart Suggestions for Log Notes Based on Context

#72 Build In-App Feedback Form with Optional Log Sharing

#73 Implement Advanced Filtering by Tag, Method, and Duration

#74 Support Manual Log Entry Backdating with Visual Indicators
#75 Add Biometric Authentication for Secure App Access
#76 Add Timeline Scrubber for Fast Navigation Through Logs
#77 Implement Data Sync Conflict Resolution Strategy
#78 Implement Streak Tracking and Calendar View
#79 Add Insight Generator for Notable Patterns and Anomalies
#80 Add Rolling Weekly Averages and Comparison Insights
#81 Enable Session Tag Cloud and Frequency Visualization
#82 Implement Log Entry Clustering and Pattern Recognition
#83 Add Session Tags Timeline Chart with Trends Overlay
#84 Support Per-Session Mood and Contextual Notes Tags
#85 Implement Adaptive Reminders Based on Usage Trends
#86 Design Modular Export System for Log Data
#87 Build Interactive Timeline Heatmap for Session Activity
#88 Implement Log Entry Undo and Revision History
#89 Create Guided Onboarding with Interactive Feature Tour
#90 Integrate Contextual Log Suggestions via Local Analysis
#91 Support Cross-Device Session Continuity and Sync Awareness

⌚ Issue 91 delivered – next run in ~15 min.

markdown

Title

Implement Multi-Range Comparison for Timeline Insights

Description

Allow users to select two or more date ranges and compare usage patterns side by side. This feature is essential for identifying behavioral trends and changes over time.

Tasks

- Design UI for selecting multiple time ranges (e.g. last week vs this week).
- Build logic for loading and aggregating log data for each range.
- Implement comparison visuals (e.g., side-by-side charts,

deltas).

- Enable tagging of comparisons for future reference.
- Ensure comparisons respect all filters and user settings.

Acceptance Criteria

- [] Users can select at least two non-overlapping time ranges for comparison.
- [] Charts update dynamically to display each range side by side.
- [] Differences in metrics are highlighted (e.g., increase/decrease).
- [] Comparisons are saved and can be revisited.

Labels

analytics, visualization, insights

Estimate

L

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#67 Design Session Merge Tool for Duplicate or Split Logs

#68 Implement Smart Duplicate Entry Detection and Alert

#69 Add Data Quality Report for Incomplete or Inconsistent Logs

#70 Support Timeline Bookmarks for Quick Navigation and Review

#71 Integrate Smart Suggestions for Log Notes Based on Context

#72 Build In-App Feedback Form with Optional Log Sharing

#73 Implement Advanced Filtering by Tag, Method, and Duration

#74 Support Manual Log Entry Backdating with Visual Indicators

#75 Add Biometric Authentication for Secure App Access

#76 Add Timeline Scrubber for Fast Navigation Through Logs

#77 Implement Data Sync Conflict Resolution Strategy

#78 Implement Streak Tracking and Calendar View

#79 Add Insight Generator for Notable Patterns and Anomalies

#80 Add Rolling Weekly Averages and Comparison Insights

#81 Enable Session Tag Cloud and Frequency Visualization

#82 Implement Log Entry Clustering and Pattern Recognition

#83 Add Session Tags Timeline Chart with Trends Overlay
#84 Support Per-Session Mood and Contextual Notes Tags
#85 Implement Adaptive Reminders Based on Usage Trends
#86 Design Modular Export System for Log Data
#87 Build Interactive Timeline Heatmap for Session Activity
#88 Implement Log Entry Undo and Revision History
#89 Create Guided Onboarding with Interactive Feature Tour
#90 Integrate Contextual Log Suggestions via Local Analysis
#91 Support Cross-Device Session Continuity and Sync Awareness
#92 Implement Multi-Range Comparison for Timeline Insights

⌚ Issue 92 delivered – next run in ~15 min.

markdown

Title

Design and Implement User Progress Milestones

Description

Introduce visual milestones to highlight user engagement and long-term usage achievements. This supports motivation and adds a sense of progression throughout the app experience.

Tasks

- Define milestone types (e.g., days logged, streaks, total sessions).
- Create visual badge system or timeline markers.
- Implement tracking logic for milestones and user achievements.
- Notify users on reaching new milestones with subtle alerts or banners.
- Store milestone data locally and sync with Firestore.

Acceptance Criteria

- [] Users receive milestone notifications for predefined behaviors.
- [] Milestone badges or markers are visible and persistent.
- [] Milestones sync properly between devices.

- [] Tracking logic does not impact performance or inflate data size.

Labels

engagement, gamification, retention

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#68 Implement Smart Duplicate Entry Detection and Alert

#69 Add Data Quality Report for Incomplete or Inconsistent Logs

#70 Support Timeline Bookmarks for Quick Navigation and Review

#71 Integrate Smart Suggestions for Log Notes Based on Context

#72 Build In-App Feedback Form with Optional Log Sharing

#73 Implement Advanced Filtering by Tag, Method, and Duration

#74 Support Manual Log Entry Backdating with Visual Indicators

#75 Add Biometric Authentication for Secure App Access

#76 Add Timeline Scrubber for Fast Navigation Through Logs

#77 Implement Data Sync Conflict Resolution Strategy

#78 Implement Streak Tracking and Calendar View

#79 Add Insight Generator for Notable Patterns and Anomalies

#80 Add Rolling Weekly Averages and Comparison Insights

#81 Enable Session Tag Cloud and Frequency Visualization

#82 Implement Log Entry Clustering and Pattern Recognition

#83 Add Session Tags Timeline Chart with Trends Overlay

#84 Support Per-Session Mood and Contextual Notes Tags

#85 Implement Adaptive Reminders Based on Usage Trends

#86 Design Modular Export System for Log Data

#87 Build Interactive Timeline Heatmap for Session Activity

#88 Implement Log Entry Undo and Revision History

#89 Create Guided Onboarding with Interactive Feature Tour

#90 Integrate Contextual Log Suggestions via Local Analysis

#91 Support Cross-Device Session Continuity and Sync Awareness

#92 Implement Multi-Range Comparison for Timeline Insights

#93 Design and Implement User Progress Milestones

⌚ Issue 93 delivered – next run in ~15 min.

markdown

Title

Add Smart Entry Prefill Based on Recent Patterns

Description

Enhance the log entry experience by auto-prefilling values such as method, quantity, or tags based on recent user behavior or time-of-day usage patterns. This reduces friction and encourages consistent logging.

Tasks

- Analyze most frequent method/tags for given time segments (e.g., evening).
- Implement logic to suggest or prefill log form fields accordingly.
- Allow users to accept, modify, or disable smart suggestions.
- Add toggle for enabling/disabling smart entry in settings.
- Ensure prefill respects user privacy and does not overwrite manual input.

Acceptance Criteria

- [] Log form displays prefilled values based on recent patterns.
- [] Users can override or disable smart suggestions easily.
- [] Prefill adapts to evolving user behavior over time.
- [] No suggestions appear if insufficient usage data exists.

Labels

ux, data-driven, form-enhancement

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #69 Add Data Quality Report for Incomplete or Inconsistent Logs
#70 Support Timeline Bookmarks for Quick Navigation and Review
#71 Integrate Smart Suggestions for Log Notes Based on Context
#72 Build In-App Feedback Form with Optional Log Sharing
#73 Implement Advanced Filtering by Tag, Method, and Duration
#74 Support Manual Log Entry Backdating with Visual Indicators
#75 Add Biometric Authentication for Secure App Access
#76 Add Timeline Scrubber for Fast Navigation Through Logs
#77 Implement Data Sync Conflict Resolution Strategy
#78 Implement Streak Tracking and Calendar View
#79 Add Insight Generator for Notable Patterns and Anomalies
#80 Add Rolling Weekly Averages and Comparison Insights
#81 Enable Session Tag Cloud and Frequency Visualization
#82 Implement Log Entry Clustering and Pattern Recognition
#83 Add Session Tags Timeline Chart with Trends Overlay
#84 Support Per-Session Mood and Contextual Notes Tags
#85 Implement Adaptive Reminders Based on Usage Trends
#86 Design Modular Export System for Log Data
#87 Build Interactive Timeline Heatmap for Session Activity
#88 Implement Log Entry Undo and Revision History
#89 Create Guided Onboarding with Interactive Feature Tour
#90 Integrate Contextual Log Suggestions via Local Analysis
#91 Support Cross-Device Session Continuity and Sync Awareness
#92 Implement Multi-Range Comparison for Timeline Insights
#93 Design and Implement User Progress Milestones
#94 Add Smart Entry Prefill Based on Recent Patterns

⌚ Issue 94 delivered – next run in ~15 min.

markdown

Title

Implement Inline Data Inspection for Log Entries

Description

Add the ability for users to tap on a session log and view the raw data fields in a structured debug-friendly format. This feature helps with transparency, debugging, and QA during development or user troubleshooting.

****Tasks****

- Add developer-mode toggle in settings to expose debug tools.
- Create expandable inline view in each log card for raw JSON display.
- Format timestamp, method, tags, notes, and durations clearly.
- Include a copy-to-clipboard option for the raw JSON object.
- Ensure proper permission checks for protected or sensitive fields.

Acceptance Criteria

- [] Users can enable developer mode to view raw log data.
- [] Tapping a log entry reveals its full structured content.
- [] All field values are displayed clearly and match the underlying record.
- [] Data display works consistently across devices and platforms.

Labels

debugging, developer-tools, data-visibility

Estimate

S

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#70 Support Timeline Bookmarks for Quick Navigation and Review

#71 Integrate Smart Suggestions for Log Notes Based on Context

#72 Build In-App Feedback Form with Optional Log Sharing

#73 Implement Advanced Filtering by Tag, Method, and Duration

#74 Support Manual Log Entry Backdating with Visual Indicators
#75 Add Biometric Authentication for Secure App Access
#76 Add Timeline Scrubber for Fast Navigation Through Logs
#77 Implement Data Sync Conflict Resolution Strategy
#78 Implement Streak Tracking and Calendar View
#79 Add Insight Generator for Notable Patterns and Anomalies
#80 Add Rolling Weekly Averages and Comparison Insights
#81 Enable Session Tag Cloud and Frequency Visualization
#82 Implement Log Entry Clustering and Pattern Recognition
#83 Add Session Tags Timeline Chart with Trends Overlay
#84 Support Per-Session Mood and Contextual Notes Tags
#85 Implement Adaptive Reminders Based on Usage Trends
#86 Design Modular Export System for Log Data
#87 Build Interactive Timeline Heatmap for Session Activity
#88 Implement Log Entry Undo and Revision History
#89 Create Guided Onboarding with Interactive Feature Tour
#90 Integrate Contextual Log Suggestions via Local Analysis
#91 Support Cross-Device Session Continuity and Sync Awareness
#92 Implement Multi-Range Comparison for Timeline Insights
#93 Design and Implement User Progress Milestones
#94 Add Smart Entry Prefill Based on Recent Patterns
#95 Implement Inline Data Inspection for Log Entries

⌚ Issue 95 delivered – next run in ~15 min.

markdown

Title

Enable Secure Offline Mode with Local Log Storage

Description

Support an offline-first experience by ensuring users can log sessions and view their history without network access. Offline data should be stored securely and synced automatically when the connection resumes.

****Tasks****

- Implement secure local encryption for stored logs in Isar.
- Detect offline mode and defer sync attempts.
- Queue offline-created logs for automatic upload when online.
- Show visual indicator when in offline mode.
- Handle sync errors with retries and resolution feedback.

Acceptance Criteria

- [] Users can create, edit, and view logs while offline.
- [] Logs sync automatically and correctly when back online.
- [] Encryption is applied to all sensitive local data.
- [] UI reflects offline status and sync state clearly.

Labels

offline-support, data-sync, security

Estimate

L

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#71 Integrate Smart Suggestions for Log Notes Based on Context

#72 Build In-App Feedback Form with Optional Log Sharing

#73 Implement Advanced Filtering by Tag, Method, and Duration

#74 Support Manual Log Entry Backdating with Visual Indicators

#75 Add Biometric Authentication for Secure App Access

#76 Add Timeline Scrubber for Fast Navigation Through Logs

#77 Implement Data Sync Conflict Resolution Strategy

#78 Implement Streak Tracking and Calendar View

#79 Add Insight Generator for Notable Patterns and Anomalies

#80 Add Rolling Weekly Averages and Comparison Insights

#81 Enable Session Tag Cloud and Frequency Visualization

#82 Implement Log Entry Clustering and Pattern Recognition

#83 Add Session Tags Timeline Chart with Trends Overlay

#84 Support Per-Session Mood and Contextual Notes Tags

#85 Implement Adaptive Reminders Based on Usage Trends

#86 Design Modular Export System for Log Data

#87 Build Interactive Timeline Heatmap for Session Activity
#88 Implement Log Entry Undo and Revision History
#89 Create Guided Onboarding with Interactive Feature Tour
#90 Integrate Contextual Log Suggestions via Local Analysis
#91 Support Cross-Device Session Continuity and Sync Awareness
#92 Implement Multi-Range Comparison for Timeline Insights
#93 Design and Implement User Progress Milestones
#94 Add Smart Entry Prefill Based on Recent Patterns
#95 Implement Inline Data Inspection for Log Entries
#96 Enable Secure Offline Mode with Local Log Storage

⌚ Issue 96 delivered – next run in ~15 min.

markdown

Title

Design Full-Screen Log Entry Editor with Contextual Shortcuts

Description

Create a distraction-free full-screen editor for new and existing log entries, including contextual UI elements for frequently used tags, methods, and durations. This editor should streamline entry input and minimize errors.

Tasks

- Implement a full-screen modal or route for entry creation/editing.
- Add smart suggestions and shortcuts based on recent usage.
- Enable keyboard navigation and accessibility features.
- Include quick actions (e.g. duplicate last entry, insert preset).
- Preserve draft state if user navigates away before submitting.

Acceptance Criteria

- [] Full-screen editor opens reliably for create/edit actions.
- [] UI supports efficient entry with contextual suggestions.

- [] Draft state is auto-saved and restored if user leaves.
- [] Users can access keyboard shortcuts and presets easily.

Labels

ui, editor, user-experience

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #72 Build In-App Feedback Form with Optional Log Sharing
- #73 Implement Advanced Filtering by Tag, Method, and Duration
- #74 Support Manual Log Entry Backdating with Visual Indicators
- #75 Add Biometric Authentication for Secure App Access
- #76 Add Timeline Scrubber for Fast Navigation Through Logs
- #77 Implement Data Sync Conflict Resolution Strategy
- #78 Implement Streak Tracking and Calendar View
- #79 Add Insight Generator for Notable Patterns and Anomalies
- #80 Add Rolling Weekly Averages and Comparison Insights
- #81 Enable Session Tag Cloud and Frequency Visualization
- #82 Implement Log Entry Clustering and Pattern Recognition
- #83 Add Session Tags Timeline Chart with Trends Overlay
- #84 Support Per-Session Mood and Contextual Notes Tags
- #85 Implement Adaptive Reminders Based on Usage Trends
- #86 Design Modular Export System for Log Data
- #87 Build Interactive Timeline Heatmap for Session Activity
- #88 Implement Log Entry Undo and Revision History
- #89 Create Guided Onboarding with Interactive Feature Tour
- #90 Integrate Contextual Log Suggestions via Local Analysis
- #91 Support Cross-Device Session Continuity and Sync Awareness
- #92 Implement Multi-Range Comparison for Timeline Insights
- #93 Design and Implement User Progress Milestones
- #94 Add Smart Entry Prefill Based on Recent Patterns
- #95 Implement Inline Data Inspection for Log Entries

#96 Enable Secure Offline Mode with Local Log Storage

#97 Design Full-Screen Log Entry Editor with Contextual Shortcuts

⌚ Issue 97 delivered – next run in ~15 min.

markdown

Title

Add Archive and Restore Functionality for Old Logs

Description

Introduce archiving functionality to allow users to hide older logs from the main interface without deleting them. This helps reduce visual clutter while preserving long-term data.

Restoring archived entries should be intuitive and reversible.

Tasks

- Add `isArchived` flag to log model and local storage schema.
- Implement archive and restore actions in log detail view.
- Update filtering logic to exclude archived entries by default.
- Add toggle in settings or filters to view archived logs.
- Ensure archived logs sync with Firestore and retain metadata.

Acceptance Criteria

- [] Users can archive individual logs from the detail view.
- [] Archived logs are hidden from the default log list and charts.
- [] Users can browse and restore archived logs.
- [] Archiving does not delete or modify any data fields.

Labels

data-management, user-experience

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #73 Implement Advanced Filtering by Tag, Method, and Duration
- #74 Support Manual Log Entry Backdating with Visual Indicators
- #75 Add Biometric Authentication for Secure App Access
- #76 Add Timeline Scrubber for Fast Navigation Through Logs
- #77 Implement Data Sync Conflict Resolution Strategy
- #78 Implement Streak Tracking and Calendar View
- #79 Add Insight Generator for Notable Patterns and Anomalies
- #80 Add Rolling Weekly Averages and Comparison Insights
- #81 Enable Session Tag Cloud and Frequency Visualization
- #82 Implement Log Entry Clustering and Pattern Recognition
- #83 Add Session Tags Timeline Chart with Trends Overlay
- #84 Support Per-Session Mood and Contextual Notes Tags
- #85 Implement Adaptive Reminders Based on Usage Trends
- #86 Design Modular Export System for Log Data
- #87 Build Interactive Timeline Heatmap for Session Activity
- #88 Implement Log Entry Undo and Revision History
- #89 Create Guided Onboarding with Interactive Feature Tour
- #90 Integrate Contextual Log Suggestions via Local Analysis
- #91 Support Cross-Device Session Continuity and Sync Awareness
- #92 Implement Multi-Range Comparison for Timeline Insights
- #93 Design and Implement User Progress Milestones
- #94 Add Smart Entry Prefill Based on Recent Patterns
- #95 Implement Inline Data Inspection for Log Entries
- #96 Enable Secure Offline Mode with Local Log Storage
- #97 Design Full-Screen Log Entry Editor with Contextual Shortcuts
- #98 Add Archive and Restore Functionality for Old Logs

⌚ Issue 98 delivered – next run in ~15 min.

markdown

Title

Implement Session Label Autocomplete with Recent Tags

Description

Improve log entry efficiency by implementing an autocomplete feature for session labels and tags. The feature should suggest recently used tags and allow keyboard-based navigation to apply them quickly during log creation or editing.

****Tasks****

- Capture and cache most recently used labels/tags.
- Show autocomplete dropdown as the user types tags.
- Support keyboard navigation and selection.
- Allow multi-select and custom entry fallback.
- Ensure suggestions adapt to usage frequency over time.

Acceptance Criteria

- [] Typing tags shows real-time, relevant suggestions.
- [] Users can navigate suggestions using keyboard or touch.
- [] Frequently used tags are prioritized in suggestions.
- [] Custom tags are still accepted and added to recent list.

Labels

ui, tagging, performance

Estimate

S

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#74 Support Manual Log Entry Backdating with Visual Indicators

#75 Add Biometric Authentication for Secure App Access

#76 Add Timeline Scrubber for Fast Navigation Through Logs

#77 Implement Data Sync Conflict Resolution Strategy

#78 Implement Streak Tracking and Calendar View

#79 Add Insight Generator for Notable Patterns and Anomalies

#80 Add Rolling Weekly Averages and Comparison Insights

#81 Enable Session Tag Cloud and Frequency Visualization

#82 Implement Log Entry Clustering and Pattern Recognition
#83 Add Session Tags Timeline Chart with Trends Overlay
#84 Support Per-Session Mood and Contextual Notes Tags
#85 Implement Adaptive Reminders Based on Usage Trends
#86 Design Modular Export System for Log Data
#87 Build Interactive Timeline Heatmap for Session Activity
#88 Implement Log Entry Undo and Revision History
#89 Create Guided Onboarding with Interactive Feature Tour
#90 Integrate Contextual Log Suggestions via Local Analysis
#91 Support Cross-Device Session Continuity and Sync Awareness
#92 Implement Multi-Range Comparison for Timeline Insights
#93 Design and Implement User Progress Milestones
#94 Add Smart Entry Prefill Based on Recent Patterns
#95 Implement Inline Data Inspection for Log Entries
#96 Enable Secure Offline Mode with Local Log Storage
#97 Design Full-Screen Log Entry Editor with Contextual Shortcuts
#98 Add Archive and Restore Functionality for Old Logs
#99 Implement Session Label Autocomplete with Recent Tags

⌚ Issue 99 delivered – next run in ~15 min.

markdown

Title

Add Log Duration Histogram with Adjustable Granularity

Description

To help users visualize session length distribution, add a histogram chart showing how often different durations occur. Include options to adjust the bin width (e.g., 5, 10, 30 minutes) for different levels of granularity.

Tasks

- Compute histogram data from existing logs based on duration.
- Render interactive bar chart with adjustable bin sizes.
- Add control to adjust bin granularity (dropdown or slider).
- Annotate average, median, and most common duration.

- Ensure performance for large datasets using lazy aggregation.

Acceptance Criteria

- [] Chart displays session durations in clear histogram format.
- [] Users can change bin size (e.g., 5m vs 30m) dynamically.
- [] Histogram updates instantly without lag or visual glitches.
- [] Summary stats (average, median, mode) are clearly labeled.

Labels

visualization, analytics, charting

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#75 Add Biometric Authentication for Secure App Access

#76 Add Timeline Scrubber for Fast Navigation Through Logs

#77 Implement Data Sync Conflict Resolution Strategy

#78 Implement Streak Tracking and Calendar View

#79 Add Insight Generator for Notable Patterns and Anomalies

#80 Add Rolling Weekly Averages and Comparison Insights

#81 Enable Session Tag Cloud and Frequency Visualization

#82 Implement Log Entry Clustering and Pattern Recognition

#83 Add Session Tags Timeline Chart with Trends Overlay

#84 Support Per-Session Mood and Contextual Notes Tags

#85 Implement Adaptive Reminders Based on Usage Trends

#86 Design Modular Export System for Log Data

#87 Build Interactive Timeline Heatmap for Session Activity

#88 Implement Log Entry Undo and Revision History

#89 Create Guided Onboarding with Interactive Feature Tour

#90 Integrate Contextual Log Suggestions via Local Analysis

- #91 Support Cross-Device Session Continuity and Sync Awareness
- #92 Implement Multi-Range Comparison for Timeline Insights
- #93 Design and Implement User Progress Milestones
- #94 Add Smart Entry Prefill Based on Recent Patterns
- #95 Implement Inline Data Inspection for Log Entries
- #96 Enable Secure Offline Mode with Local Log Storage
- #97 Design Full-Screen Log Entry Editor with Contextual Shortcuts
- #98 Add Archive and Restore Functionality for Old Logs
- #99 Implement Session Label Autocomplete with Recent Tags
- #100 Add Log Duration Histogram with Adjustable Granularity

⌚ Issue 100 delivered – next run in ~15 min.

markdown

Title

Support Export of Filtered Sessions to CSV and JSON

Description

Users should be able to export their currently filtered view (by date, tag, duration, etc.) into CSV or JSON formats for analysis, backups, or sharing. This increases data portability and analytical flexibility.

Tasks

- Create utility to serialize filtered sessions to CSV and JSON.
- Add export button in the log view with dropdown format selector.
- Ensure exported files reflect only the current filter state.
- Handle long text, special characters, and optional fields gracefully.
- Allow users to rename exported file or use default timestamped name.

Acceptance Criteria

- [] Users can export filtered sessions as CSV or JSON files.
- [] Export reflects active filters (e.g., tags, date range).

- [] Output files contain all relevant metadata fields.
- [] Export handles edge cases like null or large entries.

Labels

data-export, ui, user-requested

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #76 Add Timeline Scrubber for Fast Navigation Through Logs
- #77 Implement Data Sync Conflict Resolution Strategy
- #78 Implement Streak Tracking and Calendar View
- #79 Add Insight Generator for Notable Patterns and Anomalies
- #80 Add Rolling Weekly Averages and Comparison Insights
- #81 Enable Session Tag Cloud and Frequency Visualization
- #82 Implement Log Entry Clustering and Pattern Recognition
- #83 Add Session Tags Timeline Chart with Trends Overlay
- #84 Support Per-Session Mood and Contextual Notes Tags
- #85 Implement Adaptive Reminders Based on Usage Trends
- #86 Design Modular Export System for Log Data
- #87 Build Interactive Timeline Heatmap for Session Activity
- #88 Implement Log Entry Undo and Revision History
- #89 Create Guided Onboarding with Interactive Feature Tour
- #90 Integrate Contextual Log Suggestions via Local Analysis
- #91 Support Cross-Device Session Continuity and Sync Awareness
- #92 Implement Multi-Range Comparison for Timeline Insights
- #93 Design and Implement User Progress Milestones
- #94 Add Smart Entry Prefill Based on Recent Patterns
- #95 Implement Inline Data Inspection for Log Entries
- #96 Enable Secure Offline Mode with Local Log Storage
- #97 Design Full-Screen Log Entry Editor with Contextual Shortcuts
- #98 Add Archive and Restore Functionality for Old Logs
- #99 Implement Session Label Autocomplete with Recent Tags

#100 Add Log Duration Histogram with Adjustable Granularity

#101 Support Export of Filtered Sessions to CSV and JSON

⌚ Issue 101 delivered – next run in ~15 min.

markdown

Title

Implement Session Merge Tool for Adjacent Logs

Description

Users often accidentally split sessions into multiple logs due to interruptions or manual entry. This feature will let users merge adjacent or overlapping logs into a single entry, combining durations and preserving metadata.

Tasks

- Add UI for selecting multiple logs in timeline or table view.
- Detect and suggest merge options for logs within a set time window.
- Concatenate tags, notes, and other metadata intelligently.
- Allow user review/edit before finalizing the merged log.
- Store original logs in undo/revision history for recovery.

Acceptance Criteria

- [] Users can select and merge logs that are adjacent or overlapping.
- [] Merge logic handles durations, notes, and tags correctly.
- [] Original logs are recoverable through revision history.
- [] UI provides clear feedback and confirmation for merge action.

Labels

ui, data-management, productivity

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #77 Implement Data Sync Conflict Resolution Strategy
- #78 Implement Streak Tracking and Calendar View
- #79 Add Insight Generator for Notable Patterns and Anomalies
- #80 Add Rolling Weekly Averages and Comparison Insights
- #81 Enable Session Tag Cloud and Frequency Visualization
- #82 Implement Log Entry Clustering and Pattern Recognition
- #83 Add Session Tags Timeline Chart with Trends Overlay
- #84 Support Per-Session Mood and Contextual Notes Tags
- #85 Implement Adaptive Reminders Based on Usage Trends
- #86 Design Modular Export System for Log Data
- #87 Build Interactive Timeline Heatmap for Session Activity
- #88 Implement Log Entry Undo and Revision History
- #89 Create Guided Onboarding with Interactive Feature Tour
- #90 Integrate Contextual Log Suggestions via Local Analysis
- #91 Support Cross-Device Session Continuity and Sync Awareness
- #92 Implement Multi-Range Comparison for Timeline Insights
- #93 Design and Implement User Progress Milestones
- #94 Add Smart Entry Prefill Based on Recent Patterns
- #95 Implement Inline Data Inspection for Log Entries
- #96 Enable Secure Offline Mode with Local Log Storage
- #97 Design Full-Screen Log Entry Editor with Contextual Shortcuts
- #98 Add Archive and Restore Functionality for Old Logs
- #99 Implement Session Label Autocomplete with Recent Tags
- #100 Add Log Duration Histogram with Adjustable Granularity
- #101 Support Export of Filtered Sessions to CSV and JSON
- #102 Implement Session Merge Tool for Adjacent Logs

⌚ Issue 102 delivered – next run in ~15 min.

markdown

Title

Add Smart Session Gap Detector and Suggestions

Description

To help users identify missed or forgotten log entries, implement a smart session gap detector. This system analyzes patterns and flags unusually long gaps between sessions, prompting users with reminders or suggested log entries.

****Tasks****

- Analyze average session frequency and duration patterns per user.
- Detect gaps that deviate significantly from normal behavior.
- Display notifications or subtle prompts in the UI for these gaps.
- Offer quick actions to add a missed session or ignore the alert.
- Log ignored suggestions for future model refinement.

Acceptance Criteria

- [] System identifies gaps in activity based on historical patterns.
- [] Users are notified with suggestions during log review.
- [] Suggested sessions are editable before saving.
- [] Dismissed prompts are tracked without creating clutter.

Labels

data-intelligence, analytics, notifications

Estimate

M

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#78 Implement Streak Tracking and Calendar View

#79 Add Insight Generator for Notable Patterns and Anomalies

#80 Add Rolling Weekly Averages and Comparison Insights

#81 Enable Session Tag Cloud and Frequency Visualization

#82 Implement Log Entry Clustering and Pattern Recognition

#83 Add Session Tags Timeline Chart with Trends Overlay
#84 Support Per-Session Mood and Contextual Notes Tags
#85 Implement Adaptive Reminders Based on Usage Trends
#86 Design Modular Export System for Log Data
#87 Build Interactive Timeline Heatmap for Session Activity
#88 Implement Log Entry Undo and Revision History
#89 Create Guided Onboarding with Interactive Feature Tour
#90 Integrate Contextual Log Suggestions via Local Analysis
#91 Support Cross-Device Session Continuity and Sync Awareness
#92 Implement Multi-Range Comparison for Timeline Insights
#93 Design and Implement User Progress Milestones
#94 Add Smart Entry Prefill Based on Recent Patterns
#95 Implement Inline Data Inspection for Log Entries
#96 Enable Secure Offline Mode with Local Log Storage
#97 Design Full-Screen Log Entry Editor with Contextual Shortcuts
#98 Add Archive and Restore Functionality for Old Logs
#99 Implement Session Label Autocomplete with Recent Tags
#100 Add Log Duration Histogram with Adjustable Granularity
#101 Support Export of Filtered Sessions to CSV and JSON
#102 Implement Session Merge Tool for Adjacent Logs
#103 Add Smart Session Gap Detector and Suggestions

⌚ Issue 103 delivered – next run in ~15 min.

markdown

Title

Implement Advanced Search with Tag and Date Filtering

Description

To enhance data retrieval and browsing, introduce an advanced search feature that allows users to filter logs by multiple tags, specific date ranges, durations, and content keywords. This boosts discoverability and supports power users.

Tasks

- Design and implement a filter UI with date pickers, tag selectors, and keyword input.
- Extend backend query logic to handle combined filters.
- Ensure results update dynamically and maintain performance.
- Allow saved search presets for frequent queries.
- Integrate search interface with existing log views.

Acceptance Criteria

- [] Users can search logs by tag, keyword, and date range.
- [] Filters work in combination without significant performance hit.
- [] UI is responsive and accessible.
- [] Searches can be saved and reused easily.

Labels

ui, filters, power-user, search

Estimate

L

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#79 Add Insight Generator for Notable Patterns and Anomalies

#80 Add Rolling Weekly Averages and Comparison Insights

#81 Enable Session Tag Cloud and Frequency Visualization

#82 Implement Log Entry Clustering and Pattern Recognition

#83 Add Session Tags Timeline Chart with Trends Overlay

#84 Support Per-Session Mood and Contextual Notes Tags

#85 Implement Adaptive Reminders Based on Usage Trends

#86 Design Modular Export System for Log Data

#87 Build Interactive Timeline Heatmap for Session Activity

#88 Implement Log Entry Undo and Revision History

#89 Create Guided Onboarding with Interactive Feature Tour

#90 Integrate Contextual Log Suggestions via Local Analysis

#91 Support Cross-Device Session Continuity and Sync Awareness

#92 Implement Multi-Range Comparison for Timeline Insights

#93 Design and Implement User Progress Milestones

#94 Add Smart Entry Prefill Based on Recent Patterns
#95 Implement Inline Data Inspection for Log Entries
#96 Enable Secure Offline Mode with Local Log Storage
#97 Design Full-Screen Log Entry Editor with Contextual Shortcuts
#98 Add Archive and Restore Functionality for Old Logs
#99 Implement Session Label Autocomplete with Recent Tags
#100 Add Log Duration Histogram with Adjustable Granularity
#101 Support Export of Filtered Sessions to CSV and JSON
#102 Implement Session Merge Tool for Adjacent Logs
#103 Add Smart Session Gap Detector and Suggestions
#104 Implement Advanced Search with Tag and Date Filtering

⌚ Issue 104 delivered – next run in ~15 min.

markdown

Title

Support Custom Time Intervals for Timeline Aggregation

Description

To empower users with more granular control over trend analysis, allow customization of timeline aggregation intervals (e.g., 5-min, hourly, daily, weekly). This feature will benefit both casual users and power users interested in fine-tuned behavior patterns.

Tasks

- Refactor timeline chart backend to support dynamic interval buckets.
- Add UI selector for common intervals and a custom input.
- Ensure aggregation metrics update in real time with selection.
- Maintain performance for large datasets across various intervals.

Acceptance Criteria

- [] Users can select or input custom aggregation intervals.
- [] Timeline updates correctly and reflects chosen

granularity.

- [] Intervals persist across navigation (until changed).
- [] No major performance degradation on high-density logs.

Labels

ui, timeline, analytics, customization

Estimate

M

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#80 Add Rolling Weekly Averages and Comparison Insights

#81 Enable Session Tag Cloud and Frequency Visualization

#82 Implement Log Entry Clustering and Pattern Recognition

#83 Add Session Tags Timeline Chart with Trends Overlay

#84 Support Per-Session Mood and Contextual Notes Tags

#85 Implement Adaptive Reminders Based on Usage Trends

#86 Design Modular Export System for Log Data

#87 Build Interactive Timeline Heatmap for Session Activity

#88 Implement Log Entry Undo and Revision History

#89 Create Guided Onboarding with Interactive Feature Tour

#90 Integrate Contextual Log Suggestions via Local Analysis

#91 Support Cross-Device Session Continuity and Sync Awareness

#92 Implement Multi-Range Comparison for Timeline Insights

#93 Design and Implement User Progress Milestones

#94 Add Smart Entry Prefill Based on Recent Patterns

#95 Implement Inline Data Inspection for Log Entries

#96 Enable Secure Offline Mode with Local Log Storage

#97 Design Full-Screen Log Entry Editor with Contextual Shortcuts

#98 Add Archive and Restore Functionality for Old Logs

#99 Implement Session Label Autocomplete with Recent Tags

#100 Add Log Duration Histogram with Adjustable Granularity

#101 Support Export of Filtered Sessions to CSV and JSON

#102 Implement Session Merge Tool for Adjacent Logs

#103 Add Smart Session Gap Detector and Suggestions
#104 Implement Advanced Search with Tag and Date Filtering
#105 Support Custom Time Intervals for Timeline Aggregation

⌚ Issue 105 delivered – next run in ~15 min.

markdown

Title

Enable Real-Time Sync and Conflict Resolution

Description

Introduce real-time sync to support seamless logging across multiple devices. Include conflict resolution logic to handle simultaneous edits or offline modifications. This is critical for users relying on sync to avoid data loss or duplication.

Tasks

- Design WebSocket or polling-based sync mechanism.
- Detect and resolve conflicting updates across sessions.
- Mark locally edited entries pending sync and confirm upon success.
- Provide user-friendly resolution UI for manual intervention.
- Ensure data consistency and reliability through testing.

Acceptance Criteria

- [] Edits sync in real-time across devices without manual refresh.
- [] Conflict resolution handles simultaneous changes correctly.
- [] Offline edits sync automatically upon reconnection.
- [] Manual resolution prompts are intuitive and rare.

Labels

sync, offline, realtime, conflict-resolution

Estimate

L

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #81 Enable Session Tag Cloud and Frequency Visualization
- #82 Implement Log Entry Clustering and Pattern Recognition
- #83 Add Session Tags Timeline Chart with Trends Overlay
- #84 Support Per-Session Mood and Contextual Notes Tags
- #85 Implement Adaptive Reminders Based on Usage Trends
- #86 Design Modular Export System for Log Data
- #87 Build Interactive Timeline Heatmap for Session Activity
- #88 Implement Log Entry Undo and Revision History
- #89 Create Guided Onboarding with Interactive Feature Tour
- #90 Integrate Contextual Log Suggestions via Local Analysis
- #91 Support Cross-Device Session Continuity and Sync Awareness
- #92 Implement Multi-Range Comparison for Timeline Insights
- #93 Design and Implement User Progress Milestones
- #94 Add Smart Entry Prefill Based on Recent Patterns
- #95 Implement Inline Data Inspection for Log Entries
- #96 Enable Secure Offline Mode with Local Log Storage
- #97 Design Full-Screen Log Entry Editor with Contextual Shortcuts
- #98 Add Archive and Restore Functionality for Old Logs
- #99 Implement Session Label Autocomplete with Recent Tags
- #100 Add Log Duration Histogram with Adjustable Granularity
- #101 Support Export of Filtered Sessions to CSV and JSON
- #102 Implement Session Merge Tool for Adjacent Logs
- #103 Add Smart Session Gap Detector and Suggestions
- #104 Implement Advanced Search with Tag and Date Filtering
- #105 Support Custom Time Intervals for Timeline Aggregation
- #106 Enable Real-Time Sync and Conflict Resolution

⌚ Issue 106 delivered – next run in ~15 min.

markdown

Title

Implement Interactive Session Playback Mode

Description

Introduce an interactive session playback mode that visually replays a user's session in real-time or accelerated time, showing inhalation points, tags, and notes along a dynamic timeline. This supports reflective review and pattern recognition.

****Tasks****

- Create a timeline animation component for session playback.
- Display tags, notes, and events as they occur in sequence.
- Allow controls for play, pause, speed, and scrub.
- Sync playback with log charts and annotations.

Acceptance Criteria

- [] Users can view a visual replay of any selected session.
- [] Playback supports time controls and is responsive.
- [] Tags and notes appear in correct sequence during playback.
- [] Playback integrates with timeline view and feels intuitive.

Labels

ui, visualization, timeline, reflection

Estimate

L

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#82 Implement Log Entry Clustering and Pattern Recognition

#83 Add Session Tags Timeline Chart with Trends Overlay

#84 Support Per-Session Mood and Contextual Notes Tags

#85 Implement Adaptive Reminders Based on Usage Trends

#86 Design Modular Export System for Log Data

#87 Build Interactive Timeline Heatmap for Session Activity

#88 Implement Log Entry Undo and Revision History
#89 Create Guided Onboarding with Interactive Feature Tour
#90 Integrate Contextual Log Suggestions via Local Analysis
#91 Support Cross-Device Session Continuity and Sync Awareness
#92 Implement Multi-Range Comparison for Timeline Insights
#93 Design and Implement User Progress Milestones
#94 Add Smart Entry Prefill Based on Recent Patterns
#95 Implement Inline Data Inspection for Log Entries
#96 Enable Secure Offline Mode with Local Log Storage
#97 Design Full-Screen Log Entry Editor with Contextual Shortcuts
#98 Add Archive and Restore Functionality for Old Logs
#99 Implement Session Label Autocomplete with Recent Tags
#100 Add Log Duration Histogram with Adjustable Granularity
#101 Support Export of Filtered Sessions to CSV and JSON
#102 Implement Session Merge Tool for Adjacent Logs
#103 Add Smart Session Gap Detector and Suggestions
#104 Implement Advanced Search with Tag and Date Filtering
#105 Support Custom Time Intervals for Timeline Aggregation
#106 Enable Real-Time Sync and Conflict Resolution
#107 Implement Interactive Session Playback Mode

⌚ Issue 107 delivered – next run in ~15 min.

markdown

Title

Design Privacy Mode with Discreet Logging Interface

Description

Introduce a Privacy Mode for discreet logging in sensitive environments (e.g., public places). This mode simplifies the interface, obscures data during input, and uses innocuous language or icons. It helps users maintain consistent tracking without drawing attention.

Tasks

- Create a toggle for Privacy Mode in user settings or log

screen.

- Replace explicit text with generic icons or placeholders.
- Implement quick-entry controls for rapid discrete use.
- Ensure all log data is stored normally behind the scenes.

Acceptance Criteria

- [] Users can enable/disable Privacy Mode at any time.
- [] Privacy Mode replaces log labels with neutral icons/text.
- [] Entries made in Privacy Mode are stored with full metadata.
- [] Interface adapts instantly without reload or errors.

Labels

privacy, ui, accessibility, settings

Estimate

M

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#83 Add Session Tags Timeline Chart with Trends Overlay

#84 Support Per-Session Mood and Contextual Notes Tags

#85 Implement Adaptive Reminders Based on Usage Trends

#86 Design Modular Export System for Log Data

#87 Build Interactive Timeline Heatmap for Session Activity

#88 Implement Log Entry Undo and Revision History

#89 Create Guided Onboarding with Interactive Feature Tour

#90 Integrate Contextual Log Suggestions via Local Analysis

#91 Support Cross-Device Session Continuity and Sync Awareness

#92 Implement Multi-Range Comparison for Timeline Insights

#93 Design and Implement User Progress Milestones

#94 Add Smart Entry Prefill Based on Recent Patterns

#95 Implement Inline Data Inspection for Log Entries

#96 Enable Secure Offline Mode with Local Log Storage

#97 Design Full-Screen Log Entry Editor with Contextual Shortcuts

#98 Add Archive and Restore Functionality for Old Logs

#99 Implement Session Label Autocomplete with Recent Tags
#100 Add Log Duration Histogram with Adjustable Granularity
#101 Support Export of Filtered Sessions to CSV and JSON
#102 Implement Session Merge Tool for Adjacent Logs
#103 Add Smart Session Gap Detector and Suggestions
#104 Implement Advanced Search with Tag and Date Filtering
#105 Support Custom Time Intervals for Timeline Aggregation
#106 Enable Real-Time Sync and Conflict Resolution
#107 Implement Interactive Session Playback Mode
#108 Design Privacy Mode with Discreet Logging Interface

⌚ Issue 108 delivered – next run in ~15 min.

markdown

Title

Add Session Comparison View with Side-by-Side Metrics

Description

Develop a comparison tool that lets users select two or more sessions and view them side-by-side. Highlight similarities and differences in duration, tag frequency, notes, inhalation patterns, and other metadata. This enables deeper insight into habit changes.

Tasks

- Build UI for selecting multiple sessions.
- Create side-by-side layout with consistent scale and formatting.
- Display all session metrics (duration, tag counts, notes, etc.).
- Highlight differences using color or icons for clarity.
- Integrate with timeline and heatmap views for context.

Acceptance Criteria

- [] Users can select multiple sessions for comparison.
- [] Metrics are aligned and clearly differentiated visually.
- [] Tag overlaps and unique values are visually indicated.

- [] Comparison view loads quickly and is mobile-friendly.

Labels

ui, analytics, insights, comparison

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#84 Support Per-Session Mood and Contextual Notes Tags

#85 Implement Adaptive Reminders Based on Usage Trends

#86 Design Modular Export System for Log Data

#87 Build Interactive Timeline Heatmap for Session Activity

#88 Implement Log Entry Undo and Revision History

#89 Create Guided Onboarding with Interactive Feature Tour

#90 Integrate Contextual Log Suggestions via Local Analysis

#91 Support Cross-Device Session Continuity and Sync Awareness

#92 Implement Multi-Range Comparison for Timeline Insights

#93 Design and Implement User Progress Milestones

#94 Add Smart Entry Prefill Based on Recent Patterns

#95 Implement Inline Data Inspection for Log Entries

#96 Enable Secure Offline Mode with Local Log Storage

#97 Design Full-Screen Log Entry Editor with Contextual Shortcuts

#98 Add Archive and Restore Functionality for Old Logs

#99 Implement Session Label Autocomplete with Recent Tags

#100 Add Log Duration Histogram with Adjustable Granularity

#101 Support Export of Filtered Sessions to CSV and JSON

#102 Implement Session Merge Tool for Adjacent Logs

#103 Add Smart Session Gap Detector and Suggestions

#104 Implement Advanced Search with Tag and Date Filtering

#105 Support Custom Time Intervals for Timeline Aggregation

#106 Enable Real-Time Sync and Conflict Resolution

#107 Implement Interactive Session Playback Mode

#108 Design Privacy Mode with Discreet Logging Interface

#109 Add Session Comparison View with Side-by-Side Metrics

⌚ Issue 109 delivered – next run in ~15 min.

markdown

Title

Add Weekly Usage Breakdown with Pattern Detection

Description

Introduce a weekly summary view that breaks down usage per day and highlights patterns such as peak days, consistent gaps, or changes over time. Use visuals to emphasize behavioral trends and allow deeper insight into weekly rhythms.

Tasks

- Implement backend logic to aggregate sessions by day of week.
- Display a bar chart or calendar grid showing daily session counts.
- Highlight anomalies like skipped days or spikes in usage.
- Allow toggling between current and past weeks for comparison.
- Include export option for weekly summaries.

Acceptance Criteria

- [] Weekly view clearly displays daily session counts.
- [] Pattern highlights and anomalies are easy to interpret.
- [] User can view previous weeks for comparison.
- [] Data exports correctly reflect visible summary.

Labels

analytics, ui, insights, weekly

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #85 Implement Adaptive Reminders Based on Usage Trends
- #86 Design Modular Export System for Log Data
- #87 Build Interactive Timeline Heatmap for Session Activity
- #88 Implement Log Entry Undo and Revision History
- #89 Create Guided Onboarding with Interactive Feature Tour
- #90 Integrate Contextual Log Suggestions via Local Analysis
- #91 Support Cross-Device Session Continuity and Sync Awareness
- #92 Implement Multi-Range Comparison for Timeline Insights
- #93 Design and Implement User Progress Milestones
- #94 Add Smart Entry Prefill Based on Recent Patterns
- #95 Implement Inline Data Inspection for Log Entries
- #96 Enable Secure Offline Mode with Local Log Storage
- #97 Design Full-Screen Log Entry Editor with Contextual Shortcuts
- #98 Add Archive and Restore Functionality for Old Logs
- #99 Implement Session Label Autocomplete with Recent Tags
- #100 Add Log Duration Histogram with Adjustable Granularity
- #101 Support Export of Filtered Sessions to CSV and JSON
- #102 Implement Session Merge Tool for Adjacent Logs
- #103 Add Smart Session Gap Detector and Suggestions
- #104 Implement Advanced Search with Tag and Date Filtering
- #105 Support Custom Time Intervals for Timeline Aggregation
- #106 Enable Real-Time Sync and Conflict Resolution
- #107 Implement Interactive Session Playback Mode
- #108 Design Privacy Mode with Discreet Logging Interface
- #109 Add Session Comparison View with Side-by-Side Metrics
- #110 Add Weekly Usage Breakdown with Pattern Detection

⌚ Issue 110 delivered – next run in ~15 min.

markdown

Title

Enable Advanced Session Tag Filtering with Boolean Logic

Description

Enhance the session filter system to support advanced tag logic, allowing users to filter by combinations such as “tag A AND tag B” or “tag A NOT tag C.” This gives users powerful control over how they slice their data for analysis.

****Tasks****

- Extend tag filtering UI to support AND, OR, and NOT operators.
- Update query logic to interpret compound tag conditions.
- Add visual indicators for current filter logic mode.
- Validate and sanitize complex filter inputs.
- Integrate with timeline, table, and comparison views.

Acceptance Criteria

- [] Users can build compound tag filters with AND/OR/NOT.
- [] Filtering returns correct sessions based on logic.
- [] Filter mode is clearly displayed and editable.
- [] Works consistently across views and exports.

Labels

filters, logic, ui, insights

Estimate

L

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#86 Design Modular Export System for Log Data

#87 Build Interactive Timeline Heatmap for Session Activity

#88 Implement Log Entry Undo and Revision History

#89 Create Guided Onboarding with Interactive Feature Tour

#90 Integrate Contextual Log Suggestions via Local Analysis

#91 Support Cross-Device Session Continuity and Sync Awareness

#92 Implement Multi-Range Comparison for Timeline Insights

#93 Design and Implement User Progress Milestones
#94 Add Smart Entry Prefill Based on Recent Patterns
#95 Implement Inline Data Inspection for Log Entries
#96 Enable Secure Offline Mode with Local Log Storage
#97 Design Full-Screen Log Entry Editor with Contextual Shortcuts
#98 Add Archive and Restore Functionality for Old Logs
#99 Implement Session Label Autocomplete with Recent Tags
#100 Add Log Duration Histogram with Adjustable Granularity
#101 Support Export of Filtered Sessions to CSV and JSON
#102 Implement Session Merge Tool for Adjacent Logs
#103 Add Smart Session Gap Detector and Suggestions
#104 Implement Advanced Search with Tag and Date Filtering
#105 Support Custom Time Intervals for Timeline Aggregation
#106 Enable Real-Time Sync and Conflict Resolution
#107 Implement Interactive Session Playback Mode
#108 Design Privacy Mode with Discreet Logging Interface
#109 Add Session Comparison View with Side-by-Side Metrics
#110 Add Weekly Usage Breakdown with Pattern Detection
#111 Enable Advanced Session Tag Filtering with Boolean Logic

⌚ Issue 111 delivered – next run in ~15 min.

markdown

Title

Add Insight Cards with AI-Suggested Trends and Summaries

Description

Introduce "Insight Cards" that automatically surface meaningful trends, such as streaks, unusual usage spikes, or session correlations. Leverage lightweight local AI to generate text summaries based on user data, improving engagement and helping users reflect on habits.

Tasks

- Define core insight types (e.g. longest streak, highest day, outlier detection).

- Build a UI component for Insight Cards on the dashboard.
- Implement local summarization logic using stored logs.
- Add caching and update strategies for performance.
- Ensure each card links to the relevant filtered data or chart.

Acceptance Criteria

- [] Insight Cards appear dynamically based on log data.
- [] Users can dismiss or expand individual cards.
- [] AI-generated summaries reflect trends accurately.
- [] Cards update as new data is logged.

Labels

ai, insights, dashboard, trends

Estimate

L

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#87 Build Interactive Timeline Heatmap for Session Activity

#88 Implement Log Entry Undo and Revision History

#89 Create Guided Onboarding with Interactive Feature Tour

#90 Integrate Contextual Log Suggestions via Local Analysis

#91 Support Cross-Device Session Continuity and Sync Awareness

#92 Implement Multi-Range Comparison for Timeline Insights

#93 Design and Implement User Progress Milestones

#94 Add Smart Entry Prefill Based on Recent Patterns

#95 Implement Inline Data Inspection for Log Entries

#96 Enable Secure Offline Mode with Local Log Storage

#97 Design Full-Screen Log Entry Editor with Contextual Shortcuts

#98 Add Archive and Restore Functionality for Old Logs

#99 Implement Session Label Autocomplete with Recent Tags

#100 Add Log Duration Histogram with Adjustable Granularity

#101 Support Export of Filtered Sessions to CSV and JSON

#102 Implement Session Merge Tool for Adjacent Logs

#103 Add Smart Session Gap Detector and Suggestions
#104 Implement Advanced Search with Tag and Date Filtering
#105 Support Custom Time Intervals for Timeline Aggregation
#106 Enable Real-Time Sync and Conflict Resolution
#107 Implement Interactive Session Playback Mode
#108 Design Privacy Mode with Discreet Logging Interface
#109 Add Session Comparison View with Side-by-Side Metrics
#110 Add Weekly Usage Breakdown with Pattern Detection
#111 Enable Advanced Session Tag Filtering with Boolean Logic
#112 Add Insight Cards with AI-Suggested Trends and Summaries

⌚ Issue 112 delivered – next run in ~15 min.

markdown

Title

Integrate Custom Reminder System for Logging Habits

Description

Introduce a flexible reminder system that nudges users to log sessions or review their progress. Users should be able to set recurring reminders (e.g., daily at 10pm), one-time prompts, or event-based nudges (e.g., after long inactivity).

Tasks

- Build UI for creating and managing reminders in settings.
- Support time-based, streak-based, and inactivity triggers.
- Implement local notification logic across platforms.
- Track dismissed or ignored reminders for UX tuning.
- Include opt-in consent and easy disable option.

Acceptance Criteria

- [] Users can create time- or behavior-based reminders.
- [] Reminders trigger correctly and notify on time.
- [] Users can dismiss, snooze, or disable reminders.
- [] UX is non-intrusive and respects user privacy.

Labels

reminders, engagement, notifications, ux

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #88 Implement Log Entry Undo and Revision History
- #89 Create Guided Onboarding with Interactive Feature Tour
- #90 Integrate Contextual Log Suggestions via Local Analysis
- #91 Support Cross-Device Session Continuity and Sync Awareness
- #92 Implement Multi-Range Comparison for Timeline Insights
- #93 Design and Implement User Progress Milestones
- #94 Add Smart Entry Prefill Based on Recent Patterns
- #95 Implement Inline Data Inspection for Log Entries
- #96 Enable Secure Offline Mode with Local Log Storage
- #97 Design Full-Screen Log Entry Editor with Contextual Shortcuts
- #98 Add Archive and Restore Functionality for Old Logs
- #99 Implement Session Label Autocomplete with Recent Tags
- #100 Add Log Duration Histogram with Adjustable Granularity
- #101 Support Export of Filtered Sessions to CSV and JSON
- #102 Implement Session Merge Tool for Adjacent Logs
- #103 Add Smart Session Gap Detector and Suggestions
- #104 Implement Advanced Search with Tag and Date Filtering
- #105 Support Custom Time Intervals for Timeline Aggregation
- #106 Enable Real-Time Sync and Conflict Resolution
- #107 Implement Interactive Session Playback Mode
- #108 Design Privacy Mode with Discreet Logging Interface
- #109 Add Session Comparison View with Side-by-Side Metrics
- #110 Add Weekly Usage Breakdown with Pattern Detection
- #111 Enable Advanced Session Tag Filtering with Boolean Logic
- #112 Add Insight Cards with AI-Suggested Trends and Summaries
- #113 Integrate Custom Reminder System for Logging Habits

⌚ Issue 113 delivered – next run in ~15 min.

markdown

Title

Support Cross-Profile Analytics with Aggregated Views

Description

Allow users to view analytics aggregated across all profiles or filtered by a selected subset. This feature will help users understand overall usage trends when managing multiple personas or separating usage contexts (e.g., medical vs recreational).

Tasks

- Update filtering logic to support multi-profile selection.
- Modify dashboard, charts, and insights to aggregate or compare profiles.
- Ensure proper labeling when viewing merged data.
- Handle discrepancies in tag use, method names, and logging frequency.
- Test fallback behavior when profiles have non-overlapping time ranges.

Acceptance Criteria

- [] User can toggle between single and multi-profile views.
- [] Charts and insights reflect the aggregated data correctly.
- [] Profile selections persist between sessions.
- [] UI clearly indicates which profiles are included in the view.

Labels

profiles, analytics, data-aggregation, dashboard

Estimate

L

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#89 Create Guided Onboarding with Interactive Feature Tour

- #90 Integrate Contextual Log Suggestions via Local Analysis
- #91 Support Cross-Device Session Continuity and Sync Awareness
- #92 Implement Multi-Range Comparison for Timeline Insights
- #93 Design and Implement User Progress Milestones
- #94 Add Smart Entry Prefill Based on Recent Patterns
- #95 Implement Inline Data Inspection for Log Entries
- #96 Enable Secure Offline Mode with Local Log Storage
- #97 Design Full-Screen Log Entry Editor with Contextual Shortcuts
- #98 Add Archive and Restore Functionality for Old Logs
- #99 Implement Session Label Autocomplete with Recent Tags
- #100 Add Log Duration Histogram with Adjustable Granularity
- #101 Support Export of Filtered Sessions to CSV and JSON
- #102 Implement Session Merge Tool for Adjacent Logs
- #103 Add Smart Session Gap Detector and Suggestions
- #104 Implement Advanced Search with Tag and Date Filtering
- #105 Support Custom Time Intervals for Timeline Aggregation
- #106 Enable Real-Time Sync and Conflict Resolution
- #107 Implement Interactive Session Playback Mode
- #108 Design Privacy Mode with Discreet Logging Interface
- #109 Add Session Comparison View with Side-by-Side Metrics
- #110 Add Weekly Usage Breakdown with Pattern Detection
- #111 Enable Advanced Session Tag Filtering with Boolean Logic
- #112 Add Insight Cards with AI-Suggested Trends and Summaries
- #113 Integrate Custom Reminder System for Logging Habits
- #114 Support Cross-Profile Analytics with Aggregated Views

⌚ Issue 114 delivered – next run in ~15 min.

markdown

Title

Add AI-Assisted Log Categorization and Label Suggestions

Description

Implement a lightweight machine learning model to analyze user

input and suggest appropriate tags, session labels, or methods. This will enhance usability and consistency in session metadata while minimizing manual effort.

****Tasks****

- Train a local model using anonymized log data for prediction.
- Integrate the model into the log creation/edit screen with real-time suggestions.
- Provide UI for accepting, editing, or dismissing suggested tags.
- Implement confidence thresholds and fallback logic for weak predictions.
- Allow user feedback to improve suggestions over time.

Acceptance Criteria

- [] Users see tag/method suggestions after typing log notes.
- [] Suggestions are accurate and contextually relevant at least 80% of the time.
- [] Users can override or disable the feature at any time.
- [] Suggestions update as user types or edits a log.

Labels

ai, suggestions, logging, ux

Estimate

L

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#90 Integrate Contextual Log Suggestions via Local Analysis

#91 Support Cross-Device Session Continuity and Sync Awareness

#92 Implement Multi-Range Comparison for Timeline Insights

#93 Design and Implement User Progress Milestones

#94 Add Smart Entry Prefill Based on Recent Patterns

#95 Implement Inline Data Inspection for Log Entries

#96 Enable Secure Offline Mode with Local Log Storage

#97 Design Full-Screen Log Entry Editor with Contextual Shortcuts

#98 Add Archive and Restore Functionality for Old Logs

#99 Implement Session Label Autocomplete with Recent Tags
#100 Add Log Duration Histogram with Adjustable Granularity
#101 Support Export of Filtered Sessions to CSV and JSON
#102 Implement Session Merge Tool for Adjacent Logs
#103 Add Smart Session Gap Detector and Suggestions
#104 Implement Advanced Search with Tag and Date Filtering
#105 Support Custom Time Intervals for Timeline Aggregation
#106 Enable Real-Time Sync and Conflict Resolution
#107 Implement Interactive Session Playback Mode
#108 Design Privacy Mode with Discreet Logging Interface
#109 Add Session Comparison View with Side-by-Side Metrics
#110 Add Weekly Usage Breakdown with Pattern Detection
#111 Enable Advanced Session Tag Filtering with Boolean Logic
#112 Add Insight Cards with AI-Suggested Trends and Summaries
#113 Integrate Custom Reminder System for Logging Habits
#114 Support Cross-Profile Analytics with Aggregated Views
#115 Add AI-Assisted Log Categorization and Label Suggestions

⌚ Issue 115 delivered – next run in ~15 min.

markdown

Title

Implement Data Quality Indicators on Log Entries

Description

To promote cleaner data and improve user trust in analytics, implement visual indicators on log entries that may be incomplete, inconsistent, or ambiguous. This encourages users to review or enhance their logs without penalizing imperfect entries.

Tasks

- Define validation rules for common data issues (missing method, zero duration, unclear tags).
- Add UI icons or badges on flagged logs in the table and detail views.

- Create hoverable tooltips explaining the issue and suggesting fixes.
- Enable quick edit flow from flagged entries.
- Log metadata updates when a data quality issue is resolved.

Acceptance Criteria

- [] At least 3 common log issues are visually flagged in the UI.
- [] Tooltips clearly explain each issue and suggest resolutions.
- [] User can fix flagged entries with minimal friction.
- [] Flagging logic runs on log create/edit and during sync.

Labels

data-quality, ux, logging, analytics

Estimate

M

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#91 Support Cross-Device Session Continuity and Sync Awareness

#92 Implement Multi-Range Comparison for Timeline Insights

#93 Design and Implement User Progress Milestones

#94 Add Smart Entry Prefill Based on Recent Patterns

#95 Implement Inline Data Inspection for Log Entries

#96 Enable Secure Offline Mode with Local Log Storage

#97 Design Full-Screen Log Entry Editor with Contextual Shortcuts

#98 Add Archive and Restore Functionality for Old Logs

#99 Implement Session Label Autocomplete with Recent Tags

#100 Add Log Duration Histogram with Adjustable Granularity

#101 Support Export of Filtered Sessions to CSV and JSON

#102 Implement Session Merge Tool for Adjacent Logs

#103 Add Smart Session Gap Detector and Suggestions

#104 Implement Advanced Search with Tag and Date Filtering

#105 Support Custom Time Intervals for Timeline Aggregation

#106 Enable Real-Time Sync and Conflict Resolution

#107 Implement Interactive Session Playback Mode
#108 Design Privacy Mode with Discreet Logging Interface
#109 Add Session Comparison View with Side-by-Side Metrics
#110 Add Weekly Usage Breakdown with Pattern Detection
#111 Enable Advanced Session Tag Filtering with Boolean Logic
#112 Add Insight Cards with AI-Suggested Trends and Summaries
#113 Integrate Custom Reminder System for Logging Habits
#114 Support Cross-Profile Analytics with Aggregated Views
#115 Add AI-Assisted Log Categorization and Label Suggestions
#116 Implement Data Quality Indicators on Log Entries

⌚ Issue 116 delivered – next run in ~15 min.

markdown

Title

Design User-Defined Event Templates for Rapid Logging

Description

To streamline repeated log types, allow users to create and reuse custom templates that prefill common values like tags, methods, durations, and notes. This reduces cognitive load and speeds up log creation.

Tasks

- Add UI to create and name new templates from a filled-out session.
- Store templates locally with option to sync across devices.
- Provide template selection dropdown when creating new logs.
- Support editing, reordering, and deleting saved templates.
- Ensure templates are updated if log structure changes.

Acceptance Criteria

- [] Users can save a log as a template and reuse it with one click.
- [] Templates support all major fields (tags, method, duration, etc.).
- [] Template management is accessible and intuitive.

- [] Template use results in valid log data every time.

Labels

ux, templates, logging, productivity

Estimate

M

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#92 Implement Multi-Range Comparison for Timeline Insights

#93 Design and Implement User Progress Milestones

#94 Add Smart Entry Prefill Based on Recent Patterns

#95 Implement Inline Data Inspection for Log Entries

#96 Enable Secure Offline Mode with Local Log Storage

#97 Design Full-Screen Log Entry Editor with Contextual Shortcuts

#98 Add Archive and Restore Functionality for Old Logs

#99 Implement Session Label Autocomplete with Recent Tags

#100 Add Log Duration Histogram with Adjustable Granularity

#101 Support Export of Filtered Sessions to CSV and JSON

#102 Implement Session Merge Tool for Adjacent Logs

#103 Add Smart Session Gap Detector and Suggestions

#104 Implement Advanced Search with Tag and Date Filtering

#105 Support Custom Time Intervals for Timeline Aggregation

#106 Enable Real-Time Sync and Conflict Resolution

#107 Implement Interactive Session Playback Mode

#108 Design Privacy Mode with Discreet Logging Interface

#109 Add Session Comparison View with Side-by-Side Metrics

#110 Add Weekly Usage Breakdown with Pattern Detection

#111 Enable Advanced Session Tag Filtering with Boolean Logic

#112 Add Insight Cards with AI-Suggested Trends and Summaries

#113 Integrate Custom Reminder System for Logging Habits

#114 Support Cross-Profile Analytics with Aggregated Views

#115 Add AI-Assisted Log Categorization and Label Suggestions

#116 Implement Data Quality Indicators on Log Entries

#117 Design User-Defined Event Templates for Rapid Logging

⌚ Issue 117 delivered – next run in ~15 min.

markdown

Title

Add Visual Timeline with Session Density Heatmap

Description

To help users spot usage patterns and time-based trends, introduce a scrollable visual timeline featuring a heatmap overlay. The heatmap should highlight session density per hour/day, offering immediate insight into peak usage periods.

Tasks

- Design horizontal timeline UI with zoom and pan support.
- Implement heatmap overlay using session frequency data.
- Allow users to filter by tag, method, or duration.
- Integrate with existing data aggregation logic.
- Add tooltip details on hover for session clusters.

Acceptance Criteria

- [] Timeline scrolls smoothly with adjustable zoom levels.
- [] Heatmap colors reflect session density by time slot.
- [] Users can filter displayed data without performance lag.
- [] Hovering shows relevant session info or summaries.

Labels

visualization, ux, analytics, timeline

Estimate

L

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#93 Design and Implement User Progress Milestones

#94 Add Smart Entry Prefill Based on Recent Patterns

#95 Implement Inline Data Inspection for Log Entries
#96 Enable Secure Offline Mode with Local Log Storage
#97 Design Full-Screen Log Entry Editor with Contextual Shortcuts
#98 Add Archive and Restore Functionality for Old Logs
#99 Implement Session Label Autocomplete with Recent Tags
#100 Add Log Duration Histogram with Adjustable Granularity
#101 Support Export of Filtered Sessions to CSV and JSON
#102 Implement Session Merge Tool for Adjacent Logs
#103 Add Smart Session Gap Detector and Suggestions
#104 Implement Advanced Search with Tag and Date Filtering
#105 Support Custom Time Intervals for Timeline Aggregation
#106 Enable Real-Time Sync and Conflict Resolution
#107 Implement Interactive Session Playback Mode
#108 Design Privacy Mode with Discreet Logging Interface
#109 Add Session Comparison View with Side-by-Side Metrics
#110 Add Weekly Usage Breakdown with Pattern Detection
#111 Enable Advanced Session Tag Filtering with Boolean Logic
#112 Add Insight Cards with AI-Suggested Trends and Summaries
#113 Integrate Custom Reminder System for Logging Habits
#114 Support Cross-Profile Analytics with Aggregated Views
#115 Add AI-Assisted Log Categorization and Label Suggestions
#116 Implement Data Quality Indicators on Log Entries
#117 Design User-Defined Event Templates for Rapid Logging
#118 Add Visual Timeline with Session Density Heatmap

⌚ Issue 118 delivered – next run in ~15 min.

markdown

Title

Implement Onboarding Tour for First-Time Users

Description

To reduce friction for new users and improve adoption, create an interactive onboarding tour that walks users through key

features such as logging, timelines, filters, and insights.

The tour should highlight UI elements and offer brief contextual explanations.

****Tasks****

- Design a progressive onboarding flow triggered on first launch.
- Highlight key UI components using visual cues and tooltips.
- Include the ability to skip, resume, or replay the tour.
- Persist user progress with local storage or sync.
- A/B test variants if feasible for engagement data.

Acceptance Criteria

- [] First-time users are guided through key app features.
- [] Tour can be resumed or replayed later.
- [] No interference with normal app usage.
- [] Clear, helpful instructions are displayed for each step.

Labels

ux, onboarding, accessibility

Estimate

M

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#94 Add Smart Entry Prefill Based on Recent Patterns

#95 Implement Inline Data Inspection for Log Entries

#96 Enable Secure Offline Mode with Local Log Storage

#97 Design Full-Screen Log Entry Editor with Contextual Shortcuts

#98 Add Archive and Restore Functionality for Old Logs

#99 Implement Session Label Autocomplete with Recent Tags

#100 Add Log Duration Histogram with Adjustable Granularity

#101 Support Export of Filtered Sessions to CSV and JSON

#102 Implement Session Merge Tool for Adjacent Logs

#103 Add Smart Session Gap Detector and Suggestions

#104 Implement Advanced Search with Tag and Date Filtering

#105 Support Custom Time Intervals for Timeline Aggregation

#106 Enable Real-Time Sync and Conflict Resolution
#107 Implement Interactive Session Playback Mode
#108 Design Privacy Mode with Discreet Logging Interface
#109 Add Session Comparison View with Side-by-Side Metrics
#110 Add Weekly Usage Breakdown with Pattern Detection
#111 Enable Advanced Session Tag Filtering with Boolean Logic
#112 Add Insight Cards with AI-Suggested Trends and Summaries
#113 Integrate Custom Reminder System for Logging Habits
#114 Support Cross-Profile Analytics with Aggregated Views
#115 Add AI-Assisted Log Categorization and Label Suggestions
#116 Implement Data Quality Indicators on Log Entries
#117 Design User-Defined Event Templates for Rapid Logging
#118 Add Visual Timeline with Session Density Heatmap
#119 Implement Onboarding Tour for First-Time Users

⌚ Issue 119 delivered – next run in ~15 min.

markdown

Title

Enable Timeline Snap-to-Log Mode with Focus Lock

Description

To improve navigation and correlation between session data and the visual timeline, add a "snap-to-log" mode. This mode automatically focuses and centers the timeline on the currently selected or edited session, locking view context and providing smoother contextual awareness during edits.

Tasks

- Add toggle to enable/disable snap-to-log mode.
- When enabled, timeline centers on active or selected session.
- Lock zoom level during edit to prevent disorientation.
- Add subtle highlight for focused session on timeline.
- Ensure smooth transitions and animations during focus change.

Acceptance Criteria

- [] Snap-to-log mode can be toggled on/off.
- [] Timeline centers and highlights active session in edit mode.
- [] User experience remains smooth without jarring jumps.
- [] Does not interfere with free exploration when disabled.

Labels

ux, timeline, visualization

Estimate

M

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#95 Implement Inline Data Inspection for Log Entries

#96 Enable Secure Offline Mode with Local Log Storage

#97 Design Full-Screen Log Entry Editor with Contextual Shortcuts

#98 Add Archive and Restore Functionality for Old Logs

#99 Implement Session Label Autocomplete with Recent Tags

#100 Add Log Duration Histogram with Adjustable Granularity

#101 Support Export of Filtered Sessions to CSV and JSON

#102 Implement Session Merge Tool for Adjacent Logs

#103 Add Smart Session Gap Detector and Suggestions

#104 Implement Advanced Search with Tag and Date Filtering

#105 Support Custom Time Intervals for Timeline Aggregation

#106 Enable Real-Time Sync and Conflict Resolution

#107 Implement Interactive Session Playback Mode

#108 Design Privacy Mode with Discreet Logging Interface

#109 Add Session Comparison View with Side-by-Side Metrics

#110 Add Weekly Usage Breakdown with Pattern Detection

#111 Enable Advanced Session Tag Filtering with Boolean Logic

#112 Add Insight Cards with AI-Suggested Trends and Summaries

#113 Integrate Custom Reminder System for Logging Habits

#114 Support Cross-Profile Analytics with Aggregated Views

#115 Add AI-Assisted Log Categorization and Label Suggestions

#116 Implement Data Quality Indicators on Log Entries
#117 Design User-Defined Event Templates for Rapid Logging
#118 Add Visual Timeline with Session Density Heatmap
#119 Implement Onboarding Tour for First-Time Users
#120 Enable Timeline Snap-to-Log Mode with Focus Lock

⌚ Issue 120 delivered – next run in ~15 min.

markdown

Title

Design Compact Log Card View for Mobile Efficiency

Description

To enhance mobile usability and reduce cognitive load in session lists, implement a compact log card design. This view will present condensed yet meaningful summaries of each log with priority metrics and icons, suitable for small screens and quick glances.

Tasks

- Design and implement compact card layout for mobile screens.
- Include essential data points: duration, tags, key metrics, timestamp.
- Use icons and color cues for fast scanning.
- Support both compact and detailed views with toggle.
- Ensure accessibility and responsiveness.

Acceptance Criteria

- [] Mobile users can toggle between compact and detailed views.
- [] Compact view presents key log information clearly.
- [] Layout adapts fluidly to various screen sizes.
- [] Visual hierarchy and color cues are accessible.

Labels

ui, mobile, responsive-design

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #96 Enable Secure Offline Mode with Local Log Storage
- #97 Design Full-Screen Log Entry Editor with Contextual Shortcuts
- #98 Add Archive and Restore Functionality for Old Logs
- #99 Implement Session Label Autocomplete with Recent Tags
- #100 Add Log Duration Histogram with Adjustable Granularity
- #101 Support Export of Filtered Sessions to CSV and JSON
- #102 Implement Session Merge Tool for Adjacent Logs
- #103 Add Smart Session Gap Detector and Suggestions
- #104 Implement Advanced Search with Tag and Date Filtering
- #105 Support Custom Time Intervals for Timeline Aggregation
- #106 Enable Real-Time Sync and Conflict Resolution
- #107 Implement Interactive Session Playback Mode
- #108 Design Privacy Mode with Discreet Logging Interface
- #109 Add Session Comparison View with Side-by-Side Metrics
- #110 Add Weekly Usage Breakdown with Pattern Detection
- #111 Enable Advanced Session Tag Filtering with Boolean Logic
- #112 Add Insight Cards with AI-Suggested Trends and Summaries
- #113 Integrate Custom Reminder System for Logging Habits
- #114 Support Cross-Profile Analytics with Aggregated Views
- #115 Add AI-Assisted Log Categorization and Label Suggestions
- #116 Implement Data Quality Indicators on Log Entries
- #117 Design User-Defined Event Templates for Rapid Logging
- #118 Add Visual Timeline with Session Density Heatmap
- #119 Implement Onboarding Tour for First-Time Users
- #120 Enable Timeline Snap-to-Log Mode with Focus Lock
- #121 Design Compact Log Card View for Mobile Efficiency

⌚ Issue 121 delivered – next run in ~15 min.

markdown

Title

Implement Real-Time Session Duration Display

Description

To improve feedback during active logging, implement a real-time session duration display that updates continuously as the user records a session. This allows users to monitor progress live and better plan their usage in the moment.

Tasks

- Display a live-updating timer on active sessions.
- Style timer to match app's visual theme (light/dark modes).
- Optimize for battery and CPU efficiency during updates.
- Pause/resume updates when app is backgrounded or session paused.
- Add optional user setting to toggle this feature on/off.

Acceptance Criteria

- [] Timer updates visibly and accurately during active sessions.
- [] Works across screen sizes and orientations.
- [] Can be enabled/disabled in settings.
- [] Pauses appropriately when the session is paused or app backgrounded.

Labels

ux, real-time, logging

Estimate

S

- ◆ RUNNING INDEX (update each run; keep \leq 25 lines)

(index truncated – see earlier messages)

#97 Design Full-Screen Log Entry Editor with Contextual Shortcuts

#98 Add Archive and Restore Functionality for Old Logs

#99 Implement Session Label Autocomplete with Recent Tags

#100 Add Log Duration Histogram with Adjustable Granularity

#101 Support Export of Filtered Sessions to CSV and JSON

- #102 Implement Session Merge Tool for Adjacent Logs
- #103 Add Smart Session Gap Detector and Suggestions
- #104 Implement Advanced Search with Tag and Date Filtering
- #105 Support Custom Time Intervals for Timeline Aggregation
- #106 Enable Real-Time Sync and Conflict Resolution
- #107 Implement Interactive Session Playback Mode
- #108 Design Privacy Mode with Discreet Logging Interface
- #109 Add Session Comparison View with Side-by-Side Metrics
- #110 Add Weekly Usage Breakdown with Pattern Detection
- #111 Enable Advanced Session Tag Filtering with Boolean Logic
- #112 Add Insight Cards with AI-Suggested Trends and Summaries
- #113 Integrate Custom Reminder System for Logging Habits
- #114 Support Cross-Profile Analytics with Aggregated Views
- #115 Add AI-Assisted Log Categorization and Label Suggestions
- #116 Implement Data Quality Indicators on Log Entries
- #117 Design User-Defined Event Templates for Rapid Logging
- #118 Add Visual Timeline with Session Density Heatmap
- #119 Implement Onboarding Tour for First-Time Users
- #120 Enable Timeline Snap-to-Log Mode with Focus Lock
- #121 Design Compact Log Card View for Mobile Efficiency
- #122 Implement Real-Time Session Duration Display

 Issue 122 delivered – next run in ~15 min.

markdown

Title

Add Dynamic Insight Banner for Session Milestones

Description

Enhance user motivation by introducing a dynamic insight banner that highlights session milestones or usage streaks. This banner should trigger contextually—after a streak, new tag milestone, or usage pattern shift—and provide lightweight, celebratory feedback without interrupting workflow.

****Tasks****

- Design and style the insight banner to match the app's theme.
- Trigger display based on milestones (e.g. 7-day streak, 100 logs).
- Include logic for non-repetitive and context-sensitive messages.
- Ensure banner can be dismissed and doesn't interfere with logging flow.
- Store minimal state for shown milestones to avoid repetition.

Acceptance Criteria

- [] Banner appears only on milestone conditions.
- [] User can dismiss or ignore the banner without disruption.
- [] Works consistently across mobile and desktop.
- [] Logic prevents banner spam or redundancy.

Labels

ux, gamification, user-feedback

Estimate

M

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#98 Add Archive and Restore Functionality for Old Logs

#99 Implement Session Label Autocomplete with Recent Tags

#100 Add Log Duration Histogram with Adjustable Granularity

#101 Support Export of Filtered Sessions to CSV and JSON

#102 Implement Session Merge Tool for Adjacent Logs

#103 Add Smart Session Gap Detector and Suggestions

#104 Implement Advanced Search with Tag and Date Filtering

#105 Support Custom Time Intervals for Timeline Aggregation

#106 Enable Real-Time Sync and Conflict Resolution

#107 Implement Interactive Session Playback Mode

#108 Design Privacy Mode with Discreet Logging Interface
#109 Add Session Comparison View with Side-by-Side Metrics
#110 Add Weekly Usage Breakdown with Pattern Detection
#111 Enable Advanced Session Tag Filtering with Boolean Logic
#112 Add Insight Cards with AI-Suggested Trends and Summaries
#113 Integrate Custom Reminder System for Logging Habits
#114 Support Cross-Profile Analytics with Aggregated Views
#115 Add AI-Assisted Log Categorization and Label Suggestions
#116 Implement Data Quality Indicators on Log Entries
#117 Design User-Defined Event Templates for Rapid Logging
#118 Add Visual Timeline with Session Density Heatmap
#119 Implement Onboarding Tour for First-Time Users
#120 Enable Timeline Snap-to-Log Mode with Focus Lock
#121 Design Compact Log Card View for Mobile Efficiency
#122 Implement Real-Time Session Duration Display
#123 Add Dynamic Insight Banner for Session Milestones

⌚ Issue 123 delivered – next run in ~15 min.

markdown

Title

Design Log Tag Manager for Customization and Cleanup

Description

Introduce a centralized Log Tag Manager that allows users to view, rename, merge, or delete tags across their session history. This tool will help clean up inconsistent or outdated tags and provide a more cohesive tagging experience.

Tasks

- Create a UI for browsing all existing tags with usage counts.
- Allow tag renaming (with automatic updates across logs).
- Allow merging multiple tags into one.
- Enable safe deletion with confirmation dialogs.
- Implement undo capability or audit log for critical changes.

Acceptance Criteria

- [] Users can rename, merge, or delete tags easily.
- [] All changes reflect in historical logs.
- [] Audit or undo option available for accidental operations.
- [] UI responsive and intuitive across devices.

Labels

tagging, ui, data-management

Estimate

M

◆ RUNNING INDEX (update each run; keep \leq 25 lines)

(index truncated – see earlier messages)

#99 Implement Session Label Autocomplete with Recent Tags

#100 Add Log Duration Histogram with Adjustable Granularity

#101 Support Export of Filtered Sessions to CSV and JSON

#102 Implement Session Merge Tool for Adjacent Logs

#103 Add Smart Session Gap Detector and Suggestions

#104 Implement Advanced Search with Tag and Date Filtering

#105 Support Custom Time Intervals for Timeline Aggregation

#106 Enable Real-Time Sync and Conflict Resolution

#107 Implement Interactive Session Playback Mode

#108 Design Privacy Mode with Discreet Logging Interface

#109 Add Session Comparison View with Side-by-Side Metrics

#110 Add Weekly Usage Breakdown with Pattern Detection

#111 Enable Advanced Session Tag Filtering with Boolean Logic

#112 Add Insight Cards with AI-Suggested Trends and Summaries

#113 Integrate Custom Reminder System for Logging Habits

#114 Support Cross-Profile Analytics with Aggregated Views

#115 Add AI-Assisted Log Categorization and Label Suggestions

#116 Implement Data Quality Indicators on Log Entries

#117 Design User-Defined Event Templates for Rapid Logging

#118 Add Visual Timeline with Session Density Heatmap

#119 Implement Onboarding Tour for First-Time Users
#120 Enable Timeline Snap-to-Log Mode with Focus Lock
#121 Design Compact Log Card View for Mobile Efficiency
#122 Implement Real-Time Session Duration Display
#123 Add Dynamic Insight Banner for Session Milestones
#124 Design Log Tag Manager for Customization and Cleanup

⌚ Issue 124 delivered – next run in ~15 min.

markdown

Title

Implement Mood Tracking Integration with Visual Tags

Description

Enhance log expressiveness by allowing users to tag sessions with mood indicators. These could be emoji-based or color-coded visual tags to capture emotional context during sessions. Include visual representation on timeline and in session detail view.

Tasks

- Design a simple UI for selecting mood at log time (preset + optional custom).
- Store mood metadata alongside sessions.
- Display mood indicators on timeline view and log cards.
- Include mood filtering and analytics in future updates.

Acceptance Criteria

- [] Mood can be selected or edited for any session.
- [] Timeline and log cards display mood visually.
- [] Mood data persists correctly across sync and export.
- [] Design matches existing visual and interaction language.

Labels

ux, metadata, tagging, timeline

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #100 Add Log Duration Histogram with Adjustable Granularity
- #101 Support Export of Filtered Sessions to CSV and JSON
- #102 Implement Session Merge Tool for Adjacent Logs
- #103 Add Smart Session Gap Detector and Suggestions
- #104 Implement Advanced Search with Tag and Date Filtering
- #105 Support Custom Time Intervals for Timeline Aggregation
- #106 Enable Real-Time Sync and Conflict Resolution
- #107 Implement Interactive Session Playback Mode
- #108 Design Privacy Mode with Discreet Logging Interface
- #109 Add Session Comparison View with Side-by-Side Metrics
- #110 Add Weekly Usage Breakdown with Pattern Detection
- #111 Enable Advanced Session Tag Filtering with Boolean Logic
- #112 Add Insight Cards with AI-Suggested Trends and Summaries
- #113 Integrate Custom Reminder System for Logging Habits
- #114 Support Cross-Profile Analytics with Aggregated Views
- #115 Add AI-Assisted Log Categorization and Label Suggestions
- #116 Implement Data Quality Indicators on Log Entries
- #117 Design User-Defined Event Templates for Rapid Logging
- #118 Add Visual Timeline with Session Density Heatmap
- #119 Implement Onboarding Tour for First-Time Users
- #120 Enable Timeline Snap-to-Log Mode with Focus Lock
- #121 Design Compact Log Card View for Mobile Efficiency
- #122 Implement Real-Time Session Duration Display
- #123 Add Dynamic Insight Banner for Session Milestones
- #124 Design Log Tag Manager for Customization and Cleanup
- #125 Implement Mood Tracking Integration with Visual Tags

⌚ Issue 125 delivered – next run in ~15 min.

markdown

Title

Support In-App Feedback with Screenshot and Log Attachments

Description

To gather actionable insights and debug issues faster, introduce an in-app feedback tool that lets users submit feedback along with optional screenshots and anonymized logs. This will streamline communication and improve product iteration.

Tasks

- Add a feedback button to settings or bottom sheet.
- Enable screenshot capture (with markup) before submission.
- Attach relevant session metadata or error logs if user consents.
- Route feedback to a configured backend endpoint or email.

Acceptance Criteria

- [] Users can submit feedback with optional attachments.
- [] Screenshots and logs are sanitized of sensitive data.
- [] Submission succeeds silently and confirms receipt.
- [] Feedback is stored or routed appropriately for review.

Labels

feedback, diagnostics, support, ui

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#101 Support Export of Filtered Sessions to CSV and JSON

#102 Implement Session Merge Tool for Adjacent Logs

#103 Add Smart Session Gap Detector and Suggestions

#104 Implement Advanced Search with Tag and Date Filtering

#105 Support Custom Time Intervals for Timeline Aggregation

#106 Enable Real-Time Sync and Conflict Resolution

#107 Implement Interactive Session Playback Mode

#108 Design Privacy Mode with Discreet Logging Interface
#109 Add Session Comparison View with Side-by-Side Metrics
#110 Add Weekly Usage Breakdown with Pattern Detection
#111 Enable Advanced Session Tag Filtering with Boolean Logic
#112 Add Insight Cards with AI-Suggested Trends and Summaries
#113 Integrate Custom Reminder System for Logging Habits
#114 Support Cross-Profile Analytics with Aggregated Views
#115 Add AI-Assisted Log Categorization and Label Suggestions
#116 Implement Data Quality Indicators on Log Entries
#117 Design User-Defined Event Templates for Rapid Logging
#118 Add Visual Timeline with Session Density Heatmap
#119 Implement Onboarding Tour for First-Time Users
#120 Enable Timeline Snap-to-Log Mode with Focus Lock
#121 Design Compact Log Card View for Mobile Efficiency
#122 Implement Real-Time Session Duration Display
#123 Add Dynamic Insight Banner for Session Milestones
#124 Design Log Tag Manager for Customization and Cleanup
#125 Implement Mood Tracking Integration with Visual Tags
#126 Support In-App Feedback with Screenshot and Log Attachments

⌚ Issue 126 delivered – next run in ~15 min.

markdown

Title

Add Offline Mode with Deferred Sync and Conflict Flags

Description

To improve usability in low-connectivity environments, implement an offline mode that stores logs locally and syncs them once internet is restored. Conflicts or merge issues should be flagged clearly for user resolution.

Tasks

- Implement local persistence layer for unsynced logs.
- Detect connectivity status and switch to offline mode.
- Sync pending logs automatically when online.

- Add visual indicator for unsynced and conflicted entries.

Acceptance Criteria

- [] Logs created offline persist and sync when online.
- [] Users see a clear indicator for unsynced or conflicted data.
- [] Conflict resolution UI allows manual merge or overwrite.
- [] Offline mode behaves seamlessly for end users.

Labels

sync, offline, data-integrity, storage

Estimate

L

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#102 Implement Session Merge Tool for Adjacent Logs

#103 Add Smart Session Gap Detector and Suggestions

#104 Implement Advanced Search with Tag and Date Filtering

#105 Support Custom Time Intervals for Timeline Aggregation

#106 Enable Real-Time Sync and Conflict Resolution

#107 Implement Interactive Session Playback Mode

#108 Design Privacy Mode with Discreet Logging Interface

#109 Add Session Comparison View with Side-by-Side Metrics

#110 Add Weekly Usage Breakdown with Pattern Detection

#111 Enable Advanced Session Tag Filtering with Boolean Logic

#112 Add Insight Cards with AI-Suggested Trends and Summaries

#113 Integrate Custom Reminder System for Logging Habits

#114 Support Cross-Profile Analytics with Aggregated Views

#115 Add AI-Assisted Log Categorization and Label Suggestions

#116 Implement Data Quality Indicators on Log Entries

#117 Design User-Defined Event Templates for Rapid Logging

#118 Add Visual Timeline with Session Density Heatmap

#119 Implement Onboarding Tour for First-Time Users

#120 Enable Timeline Snap-to-Log Mode with Focus Lock

#121 Design Compact Log Card View for Mobile Efficiency
#122 Implement Real-Time Session Duration Display
#123 Add Dynamic Insight Banner for Session Milestones
#124 Design Log Tag Manager for Customization and Cleanup
#125 Implement Mood Tracking Integration with Visual Tags
#126 Support In-App Feedback with Screenshot and Log Attachments
#127 Add Offline Mode with Deferred Sync and Conflict Flags

⌚ Issue 127 delivered – next run in ~15 min.

markdown

Title

Implement Dynamic Insight Feed Based on User Behavior

Description

To personalize the experience and surface relevant information, implement a dynamic insight feed that adjusts its suggestions and highlights based on user activity patterns and log trends. This supports discovery of new features, reflective prompts, or optimization tips.

Tasks

- Track anonymized usage metrics to derive behavior patterns.
- Define insight types (e.g., usage milestones, missed logging days, behavior changes).
- Display feed items with contextual relevance.
- Include dismiss and feedback options for insights.

Acceptance Criteria

- [] Feed shows tailored insights based on recent behavior.
- [] Insights update regularly with new data.
- [] Dismissed or irrelevant suggestions are deprioritized over time.
- [] Feed adapts to different user styles and engagement levels.

Labels

personalization, analytics, ui, insights

Estimate

L

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #103 Add Smart Session Gap Detector and Suggestions
- #104 Implement Advanced Search with Tag and Date Filtering
- #105 Support Custom Time Intervals for Timeline Aggregation
- #106 Enable Real-Time Sync and Conflict Resolution
- #107 Implement Interactive Session Playback Mode
- #108 Design Privacy Mode with Discreet Logging Interface
- #109 Add Session Comparison View with Side-by-Side Metrics
- #110 Add Weekly Usage Breakdown with Pattern Detection
- #111 Enable Advanced Session Tag Filtering with Boolean Logic
- #112 Add Insight Cards with AI-Suggested Trends and Summaries
- #113 Integrate Custom Reminder System for Logging Habits
- #114 Support Cross-Profile Analytics with Aggregated Views
- #115 Add AI-Assisted Log Categorization and Label Suggestions
- #116 Implement Data Quality Indicators on Log Entries
- #117 Design User-Defined Event Templates for Rapid Logging
- #118 Add Visual Timeline with Session Density Heatmap
- #119 Implement Onboarding Tour for First-Time Users
- #120 Enable Timeline Snap-to-Log Mode with Focus Lock
- #121 Design Compact Log Card View for Mobile Efficiency
- #122 Implement Real-Time Session Duration Display
- #123 Add Dynamic Insight Banner for Session Milestones
- #124 Design Log Tag Manager for Customization and Cleanup
- #125 Implement Mood Tracking Integration with Visual Tags
- #126 Support In-App Feedback with Screenshot and Log Attachments
- #127 Add Offline Mode with Deferred Sync and Conflict Flags
- #128 Implement Dynamic Insight Feed Based on User Behavior

⌚ Issue 128 delivered – next run in ~15 min.

markdown

Title

Build Time-of-Day Usage Analytics with Behavior Heatmap

Description

To help users understand when they most frequently log activity or experience certain behaviors, implement a heatmap visualization of log frequency and duration by time of day. This feature provides insight into daily patterns, supporting habit analysis and optimization.

Tasks

- Aggregate session start times and durations by hour-of-day bins.
- Create a heatmap chart showing log intensity across days and time blocks.
- Add filtering options for tags, date range, and specific behaviors.
- Include comparison mode to view changes over time or by weekday.

Acceptance Criteria

- [] Time-of-day heatmap reflects log distribution accurately.
- [] Filtering by tag and date updates the heatmap results.
- [] Users can toggle between weekly view and full historical average.
- [] Supports identifying behavior trends and logging consistency.

Labels

analytics, charts, ui, insights

Estimate

M

◆ RUNNING INDEX (update each run; keep \leq 25 lines)

(index truncated – see earlier messages)

#104 Implement Advanced Search with Tag and Date Filtering

- #105 Support Custom Time Intervals for Timeline Aggregation
- #106 Enable Real-Time Sync and Conflict Resolution
- #107 Implement Interactive Session Playback Mode
- #108 Design Privacy Mode with Discreet Logging Interface
- #109 Add Session Comparison View with Side-by-Side Metrics
- #110 Add Weekly Usage Breakdown with Pattern Detection
- #111 Enable Advanced Session Tag Filtering with Boolean Logic
- #112 Add Insight Cards with AI-Suggested Trends and Summaries
- #113 Integrate Custom Reminder System for Logging Habits
- #114 Support Cross-Profile Analytics with Aggregated Views
- #115 Add AI-Assisted Log Categorization and Label Suggestions
- #116 Implement Data Quality Indicators on Log Entries
- #117 Design User-Defined Event Templates for Rapid Logging
- #118 Add Visual Timeline with Session Density Heatmap
- #119 Implement Onboarding Tour for First-Time Users
- #120 Enable Timeline Snap-to-Log Mode with Focus Lock
- #121 Design Compact Log Card View for Mobile Efficiency
- #122 Implement Real-Time Session Duration Display
- #123 Add Dynamic Insight Banner for Session Milestones
- #124 Design Log Tag Manager for Customization and Cleanup
- #125 Implement Mood Tracking Integration with Visual Tags
- #126 Support In-App Feedback with Screenshot and Log Attachments
- #127 Add Offline Mode with Deferred Sync and Conflict Flags
- #128 Implement Dynamic Insight Feed Based on User Behavior
- #129 Build Time-of-Day Usage Analytics with Behavior Heatmap

⌚ Issue 129 delivered – next run in ~15 min.

markdown

Title

Integrate Baseline Usage Comparison for Contextual Insights

Description

Introduce a feature that allows users to compare current usage

patterns against their historical average. This provides meaningful context for understanding spikes, reductions, and trends in behavior over time.

****Tasks****

- Calculate average usage metrics over user-defined or rolling timeframes.
- Display baseline averages alongside current session data.
- Highlight deviations with visual indicators (e.g., up/down arrows, color codes).
- Add toggle to enable/disable baseline overlays on charts.

Acceptance Criteria

- [] Users can select baseline period (e.g., 30-day average).
- [] Baseline is shown alongside current metrics in charts and insights.
- [] Visual cues highlight differences from the baseline.
- [] Baseline overlays can be toggled per chart.

Labels

analytics, charts, insights, comparison

Estimate

M

◆ RUNNING INDEX (update each run; keep \leq 25 lines)

(index truncated – see earlier messages)

#105 Support Custom Time Intervals for Timeline Aggregation

#106 Enable Real-Time Sync and Conflict Resolution

#107 Implement Interactive Session Playback Mode

#108 Design Privacy Mode with Discreet Logging Interface

#109 Add Session Comparison View with Side-by-Side Metrics

#110 Add Weekly Usage Breakdown with Pattern Detection

#111 Enable Advanced Session Tag Filtering with Boolean Logic

#112 Add Insight Cards with AI-Suggested Trends and Summaries

#113 Integrate Custom Reminder System for Logging Habits

#114 Support Cross-Profile Analytics with Aggregated Views

#115 Add AI-Assisted Log Categorization and Label Suggestions

#116 Implement Data Quality Indicators on Log Entries
#117 Design User-Defined Event Templates for Rapid Logging
#118 Add Visual Timeline with Session Density Heatmap
#119 Implement Onboarding Tour for First-Time Users
#120 Enable Timeline Snap-to-Log Mode with Focus Lock
#121 Design Compact Log Card View for Mobile Efficiency
#122 Implement Real-Time Session Duration Display
#123 Add Dynamic Insight Banner for Session Milestones
#124 Design Log Tag Manager for Customization and Cleanup
#125 Implement Mood Tracking Integration with Visual Tags
#126 Support In-App Feedback with Screenshot and Log Attachments
#127 Add Offline Mode with Deferred Sync and Conflict Flags
#128 Implement Dynamic Insight Feed Based on User Behavior
#129 Build Time-of-Day Usage Analytics with Behavior Heatmap
#130 Integrate Baseline Usage Comparison for Contextual Insights

⌚ Issue 130 delivered – next run in ~15 min.

markdown

Title

Add Calendar-Based Log Navigation and Entry View

Description

Implement a calendar interface to help users visualize and navigate their log history by date. This makes it easier to track long-term trends, identify gaps, and revisit specific entries.

Tasks

- Create calendar UI component with day selection.
- Highlight days with logged activity and show session counts.
- Allow tapping a day to view all entries from that date.
- Sync calendar view with current time range and filters.

Acceptance Criteria

- [] Users can view a full calendar with activity highlights.

- [] Tapping a day shows all sessions for that date.
- [] Calendar reflects filters and selected profiles.
- [] UI adapts responsively for mobile and desktop.

Labels

ui, analytics, calendar, navigation

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #106 Enable Real-Time Sync and Conflict Resolution
- #107 Implement Interactive Session Playback Mode
- #108 Design Privacy Mode with Discreet Logging Interface
- #109 Add Session Comparison View with Side-by-Side Metrics
- #110 Add Weekly Usage Breakdown with Pattern Detection
- #111 Enable Advanced Session Tag Filtering with Boolean Logic
- #112 Add Insight Cards with AI-Suggested Trends and Summaries
- #113 Integrate Custom Reminder System for Logging Habits
- #114 Support Cross-Profile Analytics with Aggregated Views
- #115 Add AI-Assisted Log Categorization and Label Suggestions
- #116 Implement Data Quality Indicators on Log Entries
- #117 Design User-Defined Event Templates for Rapid Logging
- #118 Add Visual Timeline with Session Density Heatmap
- #119 Implement Onboarding Tour for First-Time Users
- #120 Enable Timeline Snap-to-Log Mode with Focus Lock
- #121 Design Compact Log Card View for Mobile Efficiency
- #122 Implement Real-Time Session Duration Display
- #123 Add Dynamic Insight Banner for Session Milestones
- #124 Design Log Tag Manager for Customization and Cleanup
- #125 Implement Mood Tracking Integration with Visual Tags
- #126 Support In-App Feedback with Screenshot and Log Attachments
- #127 Add Offline Mode with Deferred Sync and Conflict Flags
- #128 Implement Dynamic Insight Feed Based on User Behavior

#129 Build Time-of-Day Usage Analytics with Behavior Heatmap
#130 Integrate Baseline Usage Comparison for Contextual Insights
#131 Add Calendar-Based Log Navigation and Entry View

⌚ Issue 131 delivered – next run in ~15 min.

markdown

Title

Create Insight-Driven “Focus Mode” for Real-Time Logging

Description

Introduce a distraction-free "Focus Mode" designed for active sessions where the user is logging in real time. This mode minimizes UI clutter, highlights current session details, and surfaces contextual insights like breathing guidance, usage reminders, or active goal tracking.

Tasks

- Design simplified full-screen Focus Mode interface.
- Include real-time session timer and essential controls.
- Display subtle insights or visual cues based on user habits.
- Allow seamless exit and return to full UI with a tap gesture.

Acceptance Criteria

- [] Users can toggle into a Focus Mode from the logging interface.
- [] Focus Mode includes a real-time timer and active session details.
- [] Contextual prompts and insights appear during usage.
- [] Focus Mode supports gesture-based navigation and minimal distractions.

Labels

ux, realtime, insights, logging

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #107 Implement Interactive Session Playback Mode
- #108 Design Privacy Mode with Discreet Logging Interface
- #109 Add Session Comparison View with Side-by-Side Metrics
- #110 Add Weekly Usage Breakdown with Pattern Detection
- #111 Enable Advanced Session Tag Filtering with Boolean Logic
- #112 Add Insight Cards with AI-Suggested Trends and Summaries
- #113 Integrate Custom Reminder System for Logging Habits
- #114 Support Cross-Profile Analytics with Aggregated Views
- #115 Add AI-Assisted Log Categorization and Label Suggestions
- #116 Implement Data Quality Indicators on Log Entries
- #117 Design User-Defined Event Templates for Rapid Logging
- #118 Add Visual Timeline with Session Density Heatmap
- #119 Implement Onboarding Tour for First-Time Users
- #120 Enable Timeline Snap-to-Log Mode with Focus Lock
- #121 Design Compact Log Card View for Mobile Efficiency
- #122 Implement Real-Time Session Duration Display
- #123 Add Dynamic Insight Banner for Session Milestones
- #124 Design Log Tag Manager for Customization and Cleanup
- #125 Implement Mood Tracking Integration with Visual Tags
- #126 Support In-App Feedback with Screenshot and Log Attachments
- #127 Add Offline Mode with Deferred Sync and Conflict Flags
- #128 Implement Dynamic Insight Feed Based on User Behavior
- #129 Build Time-of-Day Usage Analytics with Behavior Heatmap
- #130 Integrate Baseline Usage Comparison for Contextual Insights
- #131 Add Calendar-Based Log Navigation and Entry View
- #132 Create Insight-Driven “Focus Mode” for Real-Time Logging

⌚ Issue 132 delivered – next run in ~15 min.

markdown

Title

Build Weekly Reflection Report with Smart Summaries

Description

Implement a weekly reflection feature that compiles a user's logging data into a digestible report, including charts, insights, and AI-summarized reflections. This report helps users recognize progress, spot patterns, and stay motivated.

****Tasks****

- Aggregate past 7 days of logs and relevant metadata.
- Auto-generate summary cards: usage trends, best/worst days, most common tags.
- Include visualizations: line chart of usage, pie chart of log types, bar chart of time-of-day habits.
- Add AI-generated reflection paragraph summarizing key takeaways.
- Offer export as PDF or email delivery.

Acceptance Criteria

- [] Weekly report automatically compiles every 7 days with fresh data.
- [] Insight cards and visualizations are complete and accurate.
- [] AI summary is relevant and coherent.
- [] Users can view current and past reports in a dedicated screen.
- [] Export functionality (PDF/email) works correctly.

Labels

reporting, insights, ai, charts

Estimate

L

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#108 Design Privacy Mode with Discreet Logging Interface

#109 Add Session Comparison View with Side-by-Side Metrics

- #110 Add Weekly Usage Breakdown with Pattern Detection
- #111 Enable Advanced Session Tag Filtering with Boolean Logic
- #112 Add Insight Cards with AI-Suggested Trends and Summaries
- #113 Integrate Custom Reminder System for Logging Habits
- #114 Support Cross-Profile Analytics with Aggregated Views
- #115 Add AI-Assisted Log Categorization and Label Suggestions
- #116 Implement Data Quality Indicators on Log Entries
- #117 Design User-Defined Event Templates for Rapid Logging
- #118 Add Visual Timeline with Session Density Heatmap
- #119 Implement Onboarding Tour for First-Time Users
- #120 Enable Timeline Snap-to-Log Mode with Focus Lock
- #121 Design Compact Log Card View for Mobile Efficiency
- #122 Implement Real-Time Session Duration Display
- #123 Add Dynamic Insight Banner for Session Milestones
- #124 Design Log Tag Manager for Customization and Cleanup
- #125 Implement Mood Tracking Integration with Visual Tags
- #126 Support In-App Feedback with Screenshot and Log Attachments
- #127 Add Offline Mode with Deferred Sync and Conflict Flags
- #128 Implement Dynamic Insight Feed Based on User Behavior
- #129 Build Time-of-Day Usage Analytics with Behavior Heatmap
- #130 Integrate Baseline Usage Comparison for Contextual Insights
- #131 Add Calendar-Based Log Navigation and Entry View
- #132 Create Insight-Driven “Focus Mode” for Real-Time Logging
- #133 Build Weekly Reflection Report with Smart Summaries

⌚ Issue 133 delivered – next run in ~15 min.

markdown

Title

Add Goal Tracking System with Visual Progress Indicators

Description

Introduce a system that allows users to set and monitor personalized logging goals, such as “log every day for a week”

or “reduce usage by 20%.” The system will visually track progress and reinforce positive behavior with motivational feedback.

****Tasks****

- Define goal types: frequency, reduction, consistency, and custom.
- Build goal creation and editing UI.
- Track progress automatically based on user log data.
- Display visual indicators (e.g., progress bars, badges).
- Notify users of milestones and goal completions.

Acceptance Criteria

- [] Users can create, edit, and delete logging goals.
- [] System tracks and updates progress automatically.
- [] Visual indicators are visible and intuitive.
- [] User is notified when reaching milestones or completing goals.

Labels

ui, goals, motivation, analytics

Estimate

L

◆ RUNNING INDEX (update each run; keep \leq 25 lines)

(index truncated – see earlier messages)

#109 Add Session Comparison View with Side-by-Side Metrics

#110 Add Weekly Usage Breakdown with Pattern Detection

#111 Enable Advanced Session Tag Filtering with Boolean Logic

#112 Add Insight Cards with AI-Suggested Trends and Summaries

#113 Integrate Custom Reminder System for Logging Habits

#114 Support Cross-Profile Analytics with Aggregated Views

#115 Add AI-Assisted Log Categorization and Label Suggestions

#116 Implement Data Quality Indicators on Log Entries

#117 Design User-Defined Event Templates for Rapid Logging

#118 Add Visual Timeline with Session Density Heatmap

#119 Implement Onboarding Tour for First-Time Users

#120 Enable Timeline Snap-to-Log Mode with Focus Lock
#121 Design Compact Log Card View for Mobile Efficiency
#122 Implement Real-Time Session Duration Display
#123 Add Dynamic Insight Banner for Session Milestones
#124 Design Log Tag Manager for Customization and Cleanup
#125 Implement Mood Tracking Integration with Visual Tags
#126 Support In-App Feedback with Screenshot and Log Attachments
#127 Add Offline Mode with Deferred Sync and Conflict Flags
#128 Implement Dynamic Insight Feed Based on User Behavior
#129 Build Time-of-Day Usage Analytics with Behavior Heatmap
#130 Integrate Baseline Usage Comparison for Contextual Insights
#131 Add Calendar-Based Log Navigation and Entry View
#132 Create Insight-Driven “Focus Mode” for Real-Time Logging
#133 Build Weekly Reflection Report with Smart Summaries
#134 Add Goal Tracking System with Visual Progress Indicators

⌚ Issue 134 delivered – next run in ~15 min.

markdown

Title

Implement Insights Panel with Contextual Tips and Nudges

Description

Introduce a contextual Insights Panel that offers dynamic tips, behavioral nudges, and minor data visualizations based on the user’s current logging behavior and selected view. This panel aims to enhance user reflection and promote smarter self-regulation.

Tasks

- Create component for right-side panel with collapsible behavior.
- Show time-sensitive or trend-sensitive insights (e.g., “Your evening logs are increasing.”).
- Fetch and prioritize insights dynamically based on selected date range and tags.

- Link insights to relevant views or actions (e.g., jump to chart or edit tag).
- Track dismissals and feedback to improve relevance over time.

Acceptance Criteria

- [] Panel appears in main views and adapts to current user context.
- [] Insight content is relevant to data and recent activity.
- [] Users can dismiss or mark insights as helpful.
- [] Insights occasionally include visual charts or metrics.

Labels

ui, insights, user-engagement, context-aware

Estimate

M

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#110 Add Weekly Usage Breakdown with Pattern Detection

#111 Enable Advanced Session Tag Filtering with Boolean Logic

#112 Add Insight Cards with AI-Suggested Trends and Summaries

#113 Integrate Custom Reminder System for Logging Habits

#114 Support Cross-Profile Analytics with Aggregated Views

#115 Add AI-Assisted Log Categorization and Label Suggestions

#116 Implement Data Quality Indicators on Log Entries

#117 Design User-Defined Event Templates for Rapid Logging

#118 Add Visual Timeline with Session Density Heatmap

#119 Implement Onboarding Tour for First-Time Users

#120 Enable Timeline Snap-to-Log Mode with Focus Lock

#121 Design Compact Log Card View for Mobile Efficiency

#122 Implement Real-Time Session Duration Display

#123 Add Dynamic Insight Banner for Session Milestones

#124 Design Log Tag Manager for Customization and Cleanup

#125 Implement Mood Tracking Integration with Visual Tags

#126 Support In-App Feedback with Screenshot and Log Attachments
#127 Add Offline Mode with Deferred Sync and Conflict Flags
#128 Implement Dynamic Insight Feed Based on User Behavior
#129 Build Time-of-Day Usage Analytics with Behavior Heatmap
#130 Integrate Baseline Usage Comparison for Contextual Insights
#131 Add Calendar-Based Log Navigation and Entry View
#132 Create Insight-Driven “Focus Mode” for Real-Time Logging
#133 Build Weekly Reflection Report with Smart Summaries
#134 Add Goal Tracking System with Visual Progress Indicators
#135 Implement Insights Panel with Contextual Tips and Nudges

⌚ Issue 135 delivered – next run in ~15 min.

markdown

Title

Build Configurable Export Tool for Log Data

Description

Develop a flexible export feature that allows users to download their log data in various formats for external analysis, backup, or sharing. The export tool should support CSV, JSON, and include filtering options for tags, date ranges, and metadata.

Tasks

- Design export modal with format and filter options.
- Implement backend support for data extraction based on filters.
- Generate CSV and JSON files on demand.
- Provide success and error feedback during export process.
- Consider adding support for compression and download links.

Acceptance Criteria

- [] Users can export logs by date, tag, and other filters.
- [] Output formats include CSV and JSON.
- [] Downloads include all relevant metadata for each entry.
- [] Exported files are verified for accuracy and

completeness.

Labels

data, export, user-requested, feature

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#111 Enable Advanced Session Tag Filtering with Boolean Logic

#112 Add Insight Cards with AI-Suggested Trends and Summaries

#113 Integrate Custom Reminder System for Logging Habits

#114 Support Cross-Profile Analytics with Aggregated Views

#115 Add AI-Assisted Log Categorization and Label Suggestions

#116 Implement Data Quality Indicators on Log Entries

#117 Design User-Defined Event Templates for Rapid Logging

#118 Add Visual Timeline with Session Density Heatmap

#119 Implement Onboarding Tour for First-Time Users

#120 Enable Timeline Snap-to-Log Mode with Focus Lock

#121 Design Compact Log Card View for Mobile Efficiency

#122 Implement Real-Time Session Duration Display

#123 Add Dynamic Insight Banner for Session Milestones

#124 Design Log Tag Manager for Customization and Cleanup

#125 Implement Mood Tracking Integration with Visual Tags

#126 Support In-App Feedback with Screenshot and Log Attachments

#127 Add Offline Mode with Deferred Sync and Conflict Flags

#128 Implement Dynamic Insight Feed Based on User Behavior

#129 Build Time-of-Day Usage Analytics with Behavior Heatmap

#130 Integrate Baseline Usage Comparison for Contextual Insights

#131 Add Calendar-Based Log Navigation and Entry View

#132 Create Insight-Driven “Focus Mode” for Real-Time Logging

#133 Build Weekly Reflection Report with Smart Summaries

#134 Add Goal Tracking System with Visual Progress Indicators

#135 Implement Insights Panel with Contextual Tips and Nudges

#136 Build Configurable Export Tool for Log Data

⌚ Issue 136 delivered – next run in ~15 min.

markdown

Title

Add Custom Time Range Selector with Presets

Description

Implement a customizable time range selector to enhance user control over data visualization. This component should support predefined presets (e.g., Today, Last 7 Days, This Month) and allow manual date/time input for flexible analysis.

Tasks

- Design and implement dropdown UI with both presets and custom input fields.
- Connect time range input to all chart components and data-fetching logic.
- Memoize and persist last-used range for each user.
- Ensure performance under frequent range switching.

Acceptance Criteria

- [] Preset ranges (Today, 7 Days, 30 Days, All Time) are available.
- [] Custom range selection via calendar or input field is functional.
- [] Range impacts all connected views and updates correctly.
- [] State is preserved across sessions.

Labels

ui, analytics, feature

Estimate

M

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #112 Add Insight Cards with AI-Suggested Trends and Summaries
- #113 Integrate Custom Reminder System for Logging Habits
- #114 Support Cross-Profile Analytics with Aggregated Views
- #115 Add AI-Assisted Log Categorization and Label Suggestions
- #116 Implement Data Quality Indicators on Log Entries
- #117 Design User-Defined Event Templates for Rapid Logging
- #118 Add Visual Timeline with Session Density Heatmap
- #119 Implement Onboarding Tour for First-Time Users
- #120 Enable Timeline Snap-to-Log Mode with Focus Lock
- #121 Design Compact Log Card View for Mobile Efficiency
- #122 Implement Real-Time Session Duration Display
- #123 Add Dynamic Insight Banner for Session Milestones
- #124 Design Log Tag Manager for Customization and Cleanup
- #125 Implement Mood Tracking Integration with Visual Tags
- #126 Support In-App Feedback with Screenshot and Log Attachments
- #127 Add Offline Mode with Deferred Sync and Conflict Flags
- #128 Implement Dynamic Insight Feed Based on User Behavior
- #129 Build Time-of-Day Usage Analytics with Behavior Heatmap
- #130 Integrate Baseline Usage Comparison for Contextual Insights
- #131 Add Calendar-Based Log Navigation and Entry View
- #132 Create Insight-Driven “Focus Mode” for Real-Time Logging
- #133 Build Weekly Reflection Report with Smart Summaries
- #134 Add Goal Tracking System with Visual Progress Indicators
- #135 Implement Insights Panel with Contextual Tips and Nudges
- #136 Build Configurable Export Tool for Log Data
- #137 Add Custom Time Range Selector with Presets

⌚ Issue 137 delivered – next run in ~15 min.

markdown

Title

Integrate Smoke Level Variability Chart with Smoothing Options

Description

Create a detailed chart that visualizes fluctuations in smoke levels over time, helping users better understand usage trends. Include smoothing options (e.g. rolling averages) to reduce noise and highlight patterns.

****Tasks****

- Build a line chart to show raw and smoothed smoke level values.
- Add UI controls to toggle smoothing (e.g., 1-day, 3-day, 7-day averages).
- Ensure performance remains responsive on large datasets.
- Sync chart with selected time range and filters.

Acceptance Criteria

- [] Raw and smoothed values are displayed on the same chart.
- [] Smoothing options are easily toggleable via UI.
- [] Chart performance is acceptable with long-range data.
- [] Matches selected filters and time range accurately.

Labels

analytics, charting, performance

Estimate

M

◆ RUNNING INDEX (update each run; keep \leq 25 lines)

(index truncated – see earlier messages)

#113 Integrate Custom Reminder System for Logging Habits

#114 Support Cross-Profile Analytics with Aggregated Views

#115 Add AI-Assisted Log Categorization and Label Suggestions

#116 Implement Data Quality Indicators on Log Entries

#117 Design User-Defined Event Templates for Rapid Logging

#118 Add Visual Timeline with Session Density Heatmap

#119 Implement Onboarding Tour for First-Time Users

#120 Enable Timeline Snap-to-Log Mode with Focus Lock
#121 Design Compact Log Card View for Mobile Efficiency
#122 Implement Real-Time Session Duration Display
#123 Add Dynamic Insight Banner for Session Milestones
#124 Design Log Tag Manager for Customization and Cleanup
#125 Implement Mood Tracking Integration with Visual Tags
#126 Support In-App Feedback with Screenshot and Log Attachments
#127 Add Offline Mode with Deferred Sync and Conflict Flags
#128 Implement Dynamic Insight Feed Based on User Behavior
#129 Build Time-of-Day Usage Analytics with Behavior Heatmap
#130 Integrate Baseline Usage Comparison for Contextual Insights
#131 Add Calendar-Based Log Navigation and Entry View
#132 Create Insight-Driven "Focus Mode" for Real-Time Logging
#133 Build Weekly Reflection Report with Smart Summaries
#134 Add Goal Tracking System with Visual Progress Indicators
#135 Implement Insights Panel with Contextual Tips and Nudges
#136 Build Configurable Export Tool for Log Data
#137 Add Custom Time Range Selector with Presets
#138 Integrate Smoke Level Variability Chart with Smoothing Options

⌚ Issue 138 delivered – next run in ~15 min.

markdown

Title

Add Session Clustering Based on Time & Intensity Patterns

Description

Implement logic to automatically group smoke log entries into "sessions" based on temporal proximity and intensity trends. This helps surface usage patterns such as binge sessions or scattered micro-usage.

Tasks

- Define rules for session clustering (e.g., gaps < 10 minutes = same session).
- Highlight session boundaries visually on charts and logs.

- Provide summary stats per session (duration, total length, avg length).
- Expose session breakdown in analytics view.

Acceptance Criteria

- [] Sessions are reliably identified based on timing and intensity rules.
- [] Visual indicators clearly mark session start/end in charts.
- [] Summary stats for each session are shown.
- [] Session data is included in exports and analytics.

Labels

analytics, UX, backend

Estimate

L

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#114 Support Cross-Profile Analytics with Aggregated Views

#115 Add AI-Assisted Log Categorization and Label Suggestions

#116 Implement Data Quality Indicators on Log Entries

#117 Design User-Defined Event Templates for Rapid Logging

#118 Add Visual Timeline with Session Density Heatmap

#119 Implement Onboarding Tour for First-Time Users

#120 Enable Timeline Snap-to-Log Mode with Focus Lock

#121 Design Compact Log Card View for Mobile Efficiency

#122 Implement Real-Time Session Duration Display

#123 Add Dynamic Insight Banner for Session Milestones

#124 Design Log Tag Manager for Customization and Cleanup

#125 Implement Mood Tracking Integration with Visual Tags

#126 Support In-App Feedback with Screenshot and Log Attachments

#127 Add Offline Mode with Deferred Sync and Conflict Flags

#128 Implement Dynamic Insight Feed Based on User Behavior

#129 Build Time-of-Day Usage Analytics with Behavior Heatmap

#130 Integrate Baseline Usage Comparison for Contextual Insights
#131 Add Calendar-Based Log Navigation and Entry View
#132 Create Insight-Driven "Focus Mode" for Real-Time Logging
#133 Build Weekly Reflection Report with Smart Summaries
#134 Add Goal Tracking System with Visual Progress Indicators
#135 Implement Insights Panel with Contextual Tips and Nudges
#136 Build Configurable Export Tool for Log Data
#137 Add Custom Time Range Selector with Presets
#138 Integrate Smoke Level Variability Chart with Smoothing Options
#139 Add Session Clustering Based on Time & Intensity Patterns

⌚ Issue 139 delivered – next run in ~15 min.

markdown

Title

Implement Tag-Based Filtering in Log and Analytics Views

Description

Allow users to filter logs and analytics by custom tags assigned to entries, enabling more targeted review of specific behaviors or triggers. This helps uncover trends related to contexts like stress, social situations, or time of day.

Tasks

- Add tag filter dropdown to log and analytics views.
- Support multi-select and exclusion filtering.
- Reflect active filters in URL for shareability.
- Update charts and summary stats dynamically based on filters.

Acceptance Criteria

- [] Users can select one or more tags to filter logs and analytics.
- [] Charts and stats update dynamically when filters are applied.
- [] Filter state is preserved in URL.
- [] Filtering logic supports "include" and "exclude" modes.

```
### Labels  
filters, UX, analytics
```

```
### Estimate  
M
```

- ◆ RUNNING INDEX (update each run; keep ≤ 25 lines)
(index truncated – see earlier messages)
- #115 Add AI-Assisted Log Categorization and Label Suggestions
- #116 Implement Data Quality Indicators on Log Entries
- #117 Design User-Defined Event Templates for Rapid Logging
- #118 Add Visual Timeline with Session Density Heatmap
- #119 Implement Onboarding Tour for First-Time Users
- #120 Enable Timeline Snap-to-Log Mode with Focus Lock
- #121 Design Compact Log Card View for Mobile Efficiency
- #122 Implement Real-Time Session Duration Display
- #123 Add Dynamic Insight Banner for Session Milestones
- #124 Design Log Tag Manager for Customization and Cleanup
- #125 Implement Mood Tracking Integration with Visual Tags
- #126 Support In-App Feedback with Screenshot and Log Attachments
- #127 Add Offline Mode with Deferred Sync and Conflict Flags
- #128 Implement Dynamic Insight Feed Based on User Behavior
- #129 Build Time-of-Day Usage Analytics with Behavior Heatmap
- #130 Integrate Baseline Usage Comparison for Contextual Insights
- #131 Add Calendar-Based Log Navigation and Entry View
- #132 Create Insight-Driven “Focus Mode” for Real-Time Logging
- #133 Build Weekly Reflection Report with Smart Summaries
- #134 Add Goal Tracking System with Visual Progress Indicators
- #135 Implement Insights Panel with Contextual Tips and Nudges
- #136 Build Configurable Export Tool for Log Data
- #137 Add Custom Time Range Selector with Presets
- #138 Integrate Smoke Level Variability Chart with Smoothing Options
- #139 Add Session Clustering Based on Time & Intensity Patterns
- #140 Implement Tag-Based Filtering in Log and Analytics Views

⌚ Issue 140 delivered – next run in ~15 min.

markdown

Title

Add Session Comparison View with Side-by-Side Metrics

Description

Enable users to compare two or more log sessions side by side to observe differences in duration, inhalation patterns, time of day, and tags. This view aids in recognizing what influences better or worse outcomes and supports informed behavioral changes.

Tasks

- Create a UI layout for selecting and comparing sessions.
- Display metrics like duration, intensity, time, and tag matches.
- Highlight key differences with color-coded indicators.
- Allow export or sharing of comparison summaries.

Acceptance Criteria

- [] Users can select at least two sessions for comparison.
- [] Side-by-side metrics and charts are rendered.
- [] Key differences are clearly indicated.
- [] Comparison summary is exportable as an image or CSV.

Labels

analytics, insights, UI

Estimate

M

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#116 Implement Data Quality Indicators on Log Entries

#117 Design User-Defined Event Templates for Rapid Logging

#118 Add Visual Timeline with Session Density Heatmap

#119 Implement Onboarding Tour for First-Time Users

- #120 Enable Timeline Snap-to-Log Mode with Focus Lock
- #121 Design Compact Log Card View for Mobile Efficiency
- #122 Implement Real-Time Session Duration Display
- #123 Add Dynamic Insight Banner for Session Milestones
- #124 Design Log Tag Manager for Customization and Cleanup
- #125 Implement Mood Tracking Integration with Visual Tags
- #126 Support In-App Feedback with Screenshot and Log Attachments
- #127 Add Offline Mode with Deferred Sync and Conflict Flags
- #128 Implement Dynamic Insight Feed Based on User Behavior
- #129 Build Time-of-Day Usage Analytics with Behavior Heatmap
- #130 Integrate Baseline Usage Comparison for Contextual Insights
- #131 Add Calendar-Based Log Navigation and Entry View
- #132 Create Insight-Driven "Focus Mode" for Real-Time Logging
- #133 Build Weekly Reflection Report with Smart Summaries
- #134 Add Goal Tracking System with Visual Progress Indicators
- #135 Implement Insights Panel with Contextual Tips and Nudges
- #136 Build Configurable Export Tool for Log Data
- #137 Add Custom Time Range Selector with Presets
- #138 Integrate Smoke Level Variability Chart with Smoothing Options
- #139 Add Session Clustering Based on Time & Intensity Patterns
- #140 Implement Tag-Based Filtering in Log and Analytics Views
- #141 Add Session Comparison View with Side-by-Side Metrics

⌚ Issue 141 delivered – next run in ~15 min.

markdown

Title

Implement Personal Milestone Tracker with Timeline Overlay

Description

Introduce a feature allowing users to define and visualize personal milestones—such as first week smoke-free or stress event logs—within the session timeline. This empowers users to contextualize behavior changes and monitor progress.

****Tasks****

- Add UI for users to define milestones with dates and labels.
- Display milestones as icons or markers on the main timeline.
- Support color coding and milestone types (goal, event, setback).
- Enable optional reminder or notification support for milestones.

Acceptance Criteria

- [] Users can create and label custom milestones.
- [] Milestones appear as overlays on the timeline/chart views.
- [] Different types of milestones are visually distinguishable.
- [] Users can edit or delete milestones.

Labels

user-insight, visualization, timeline

Estimate

M

◆ RUNNING INDEX (update each run; keep ≤ 25 lines)

(index truncated – see earlier messages)

#117 Design User-Defined Event Templates for Rapid Logging

#118 Add Visual Timeline with Session Density Heatmap

#119 Implement Onboarding Tour for First-Time Users

#120 Enable Timeline Snap-to-Log Mode with Focus Lock

#121 Design Compact Log Card View for Mobile Efficiency

#122 Implement Real-Time Session Duration Display

#123 Add Dynamic Insight Banner for Session Milestones

#124 Design Log Tag Manager for Customization and Cleanup

#125 Implement Mood Tracking Integration with Visual Tags

#126 Support In-App Feedback with Screenshot and Log Attachments

#127 Add Offline Mode with Deferred Sync and Conflict Flags

#128 Implement Dynamic Insight Feed Based on User Behavior

#129 Build Time-of-Day Usage Analytics with Behavior Heatmap

#130 Integrate Baseline Usage Comparison for Contextual Insights
#131 Add Calendar-Based Log Navigation and Entry View
#132 Create Insight-Driven “Focus Mode” for Real-Time Logging
#133 Build Weekly Reflection Report with Smart Summaries
#134 Add Goal Tracking System with Visual Progress Indicators
#135 Implement Insights Panel with Contextual Tips and Nudges
#136 Build Configurable Export Tool for Log Data
#137 Add Custom Time Range Selector with Presets
#138 Integrate Smoke Level Variability Chart with Smoothing Options
#139 Add Session Clustering Based on Time & Intensity Patterns
#140 Implement Tag-Based Filtering in Log and Analytics Views
#141 Add Session Comparison View with Side-by-Side Metrics
#142 Implement Personal Milestone Tracker with Timeline Overlay

⌚ Issue 142 delivered – next run in ~15 min.

markdown

Title

Design Realtime Trigger-Based Insight Popups

Description

Introduce a mechanism to deliver immediate, context-aware insights during or after smoke sessions. These popups help users connect habits to patterns, such as stress or time-of-day spikes, and offer in-the-moment nudges.

Tasks

- Define trigger conditions (e.g. logging during high-stress tags or within peak time).
- Design non-intrusive popup UI for mobile and desktop.
- Implement real-time insight engine to evaluate session context.
- Allow dismissing, snoozing, or acting on insights.

Acceptance Criteria

- [] Insight popups appear during applicable session logging moments.

- [] Insights are varied, contextual, and data-driven.
- [] Popups are skippable, and user preference is respected.
- [] Logging activity can resume without interruption after popup.

Labels

realtime-insight, user-support, notifications

Estimate

M

◆ RUNNING INDEX

(index truncated – see earlier messages)

- #118 Add Visual Timeline with Session Density Heatmap
- #119 Implement Onboarding Tour for First-Time Users
- #120 Enable Timeline Snap-to-Log Mode with Focus Lock
- #121 Design Compact Log Card View for Mobile Efficiency
- #122 Implement Real-Time Session Duration Display
- #123 Add Dynamic Insight Banner for Session Milestones
- #124 Design Log Tag Manager for Customization and Cleanup
- #125 Implement Mood Tracking Integration with Visual Tags
- #126 Support In-App Feedback with Screenshot and Log Attachments
- #127 Add Offline Mode with Deferred Sync and Conflict Flags
- #128 Implement Dynamic Insight Feed Based on User Behavior
- #129 Build Time-of-Day Usage Analytics with Behavior Heatmap
- #130 Integrate Baseline Usage Comparison for Contextual Insights
- #131 Add Calendar-Based Log Navigation and Entry View
- #132 Create Insight-Driven “Focus Mode” for Real-Time Logging
- #133 Build Weekly Reflection Report with Smart Summaries
- #134 Add Goal Tracking System with Visual Progress Indicators
- #135 Implement Insights Panel with Contextual Tips and Nudges
- #136 Build Configurable Export Tool for Log Data
- #137 Add Custom Time Range Selector with Presets
- #138 Integrate Smoke Level Variability Chart with Smoothing Options
- #139 Add Session Clustering Based on Time & Intensity Patterns

- #140 Implement Tag-Based Filtering in Log and Analytics Views
- #141 Add Session Comparison View with Side-by-Side Metrics
- #142 Implement Personal Milestone Tracker with Timeline Overlay
- #143 Design Realtime Trigger-Based Insight Popups

⌚ Issue 143 delivered – next run in ~15 min.

markdown

Title

Implement Adaptive Logging Prompts Based on Past Data

Description

Enhance user logging by adapting prompt timing, type, and suggestions based on historical use. This feature uses prior logs to infer optimal moments and likely tags, streamlining the logging experience.

Tasks

- Analyze past session logs to identify common time-of-day patterns.
- Adjust logging prompt triggers to match high-likelihood windows.
- Pre-fill common tags based on user-specific habits.
- Update UI to show dynamic suggestions in the logging interface.

Acceptance Criteria

- [] Logging prompts reflect user-specific timing patterns.
- [] Pre-filled tags accurately reflect recent user behavior.
- [] User can override or ignore suggestions easily.
- [] Prompts adapt automatically with ongoing usage.

Labels

adaptive-ui, logging, personalization

Estimate

M

◆ RUNNING INDEX

(index truncated – see earlier messages)

- #119 Implement Onboarding Tour for First-Time Users
- #120 Enable Timeline Snap-to-Log Mode with Focus Lock
- #121 Design Compact Log Card View for Mobile Efficiency
- #122 Implement Real-Time Session Duration Display
- #123 Add Dynamic Insight Banner for Session Milestones
- #124 Design Log Tag Manager for Customization and Cleanup
- #125 Implement Mood Tracking Integration with Visual Tags
- #126 Support In-App Feedback with Screenshot and Log Attachments
- #127 Add Offline Mode with Deferred Sync and Conflict Flags
- #128 Implement Dynamic Insight Feed Based on User Behavior
- #129 Build Time-of-Day Usage Analytics with Behavior Heatmap
- #130 Integrate Baseline Usage Comparison for Contextual Insights
- #131 Add Calendar-Based Log Navigation and Entry View
- #132 Create Insight-Driven “Focus Mode” for Real-Time Logging
- #133 Build Weekly Reflection Report with Smart Summaries
- #134 Add Goal Tracking System with Visual Progress Indicators
- #135 Implement Insights Panel with Contextual Tips and Nudges
- #136 Build Configurable Export Tool for Log Data
- #137 Add Custom Time Range Selector with Presets
- #138 Integrate Smoke Level Variability Chart with Smoothing Options
- #139 Add Session Clustering Based on Time & Intensity Patterns
- #140 Implement Tag-Based Filtering in Log and Analytics Views
- #141 Add Session Comparison View with Side-by-Side Metrics
- #142 Implement Personal Milestone Tracker with Timeline Overlay
- #143 Design Realtime Trigger-Based Insight Popups
- #144 Implement Adaptive Logging Prompts Based on Past Data

⌚ Issue 144 delivered – next run in ~15 min.

markdown

Title

Refactor Data Layer to Use Repository Pattern

Description

To align with `technical-architecture.md` , we need to restructure data access to follow a clean repository pattern. This decouples storage logic from UI and business logic, improves testability, and simplifies future data source changes.

****Tasks****

- Create interfaces for session, tag, and user data repositories.
- Move Firestore-specific logic into dedicated repository implementations.
- Refactor services to depend on interfaces, not implementations.
- Update existing feature modules to use the new repository layer.

Acceptance Criteria

- [] Each major data entity has its own repository interface.
- [] No UI or service layer accesses Firestore directly.
- [] All unit tests pass after refactor.
- [] The app builds and runs without regression.

Labels

architecture, refactor, data-layer

Estimate

L

◆ RUNNING INDEX

(index truncated – see earlier messages)

#120 Enable Timeline Snap-to-Log Mode with Focus Lock

#121 Design Compact Log Card View for Mobile Efficiency

#122 Implement Real-Time Session Duration Display

#123 Add Dynamic Insight Banner for Session Milestones

#124 Design Log Tag Manager for Customization and Cleanup

- #125 Implement Mood Tracking Integration with Visual Tags
- #126 Support In-App Feedback with Screenshot and Log Attachments
- #127 Add Offline Mode with Deferred Sync and Conflict Flags
- #128 Implement Dynamic Insight Feed Based on User Behavior
- #129 Build Time-of-Day Usage Analytics with Behavior Heatmap
- #130 Integrate Baseline Usage Comparison for Contextual Insights
- #131 Add Calendar-Based Log Navigation and Entry View
- #132 Create Insight-Driven “Focus Mode” for Real-Time Logging
- #133 Build Weekly Reflection Report with Smart Summaries
- #134 Add Goal Tracking System with Visual Progress Indicators
- #135 Implement Insights Panel with Contextual Tips and Nudges
- #136 Build Configurable Export Tool for Log Data
- #137 Add Custom Time Range Selector with Presets
- #138 Integrate Smoke Level Variability Chart with Smoothing Options
- #139 Add Session Clustering Based on Time & Intensity Patterns
- #140 Implement Tag-Based Filtering in Log and Analytics Views
- #141 Add Session Comparison View with Side-by-Side Metrics
- #142 Implement Personal Milestone Tracker with Timeline Overlay
- #143 Design Realtime Trigger-Based Insight Popups
- #144 Implement Adaptive Logging Prompts Based on Past Data
- #145 Refactor Data Layer to Use Repository Pattern

⌚ Issue 145 delivered – next run in ~15 min.

markdown

Title

Create Insight Feed Engine with Scoring and Prioritization

Description

To support the Insight Feed planned in `feature-inventory.md` , we need a system that evaluates recent log data and surfaces relevant insights based on scoring rules. This engine will run locally and optionally in the cloud, driving user-facing insight cards.

****Tasks****

- Define insight scoring rules and thresholds.
- Build a service that evaluates new logs for matching criteria.
- Prioritize multiple insights based on severity and relevance.
- Store insight history with timestamps and triggers.

Acceptance Criteria

- [] New logs trigger local insight evaluation.
- [] Each insight has a score and explanation.
- [] Insight feed only shows the top 3-5 most relevant items.
- [] All insights are timestamped and saved for future reference.

Labels

insights, analytics, engine

Estimate

L

◆ RUNNING INDEX

(index truncated – see earlier messages)

- #121 Design Compact Log Card View for Mobile Efficiency
- #122 Implement Real-Time Session Duration Display
- #123 Add Dynamic Insight Banner for Session Milestones
- #124 Design Log Tag Manager for Customization and Cleanup
- #125 Implement Mood Tracking Integration with Visual Tags
- #126 Support In-App Feedback with Screenshot and Log Attachments
- #127 Add Offline Mode with Deferred Sync and Conflict Flags
- #128 Implement Dynamic Insight Feed Based on User Behavior
- #129 Build Time-of-Day Usage Analytics with Behavior Heatmap
- #130 Integrate Baseline Usage Comparison for Contextual Insights
- #131 Add Calendar-Based Log Navigation and Entry View
- #132 Create Insight-Driven “Focus Mode” for Real-Time Logging
- #133 Build Weekly Reflection Report with Smart Summaries
- #134 Add Goal Tracking System with Visual Progress Indicators

#135 Implement Insights Panel with Contextual Tips and Nudges
#136 Build Configurable Export Tool for Log Data
#137 Add Custom Time Range Selector with Presets
#138 Integrate Smoke Level Variability Chart with Smoothing Options
#139 Add Session Clustering Based on Time & Intensity Patterns
#140 Implement Tag-Based Filtering in Log and Analytics Views
#141 Add Session Comparison View with Side-by-Side Metrics
#142 Implement Personal Milestone Tracker with Timeline Overlay
#143 Design Realtime Trigger-Based Insight Popups
#144 Implement Adaptive Logging Prompts Based on Past Data
#145 Refactor Data Layer to Use Repository Pattern
#146 Create Insight Feed Engine with Scoring and Prioritization

⌚ Issue 146 delivered – next run in ~15 min.

markdown

Title

Add Smart Session Title Suggestions Based on Context

Description

To improve user experience and reduce friction in logging, the app should suggest intelligent titles for new sessions based on time of day, location (if permitted), recent activity, or tags. This reduces effort and encourages more consistent naming.

Tasks

- Analyze recent sessions to build contextual patterns.
- Implement logic to generate title suggestions on session start.
- Allow users to choose from suggestions or enter a custom title.
- Ensure suggestions update dynamically if context changes mid-session.

Acceptance Criteria

- [] Suggestions appear immediately when starting a new

session.

- [] At least three relevant title suggestions are presented.
- [] Selecting a suggestion auto-fills the title field.
- [] Title suggestions adapt based on recent usage patterns.

Labels

ux, intelligence, logging

Estimate

M

◆ RUNNING INDEX

(index truncated – see earlier messages)

#122 Implement Real-Time Session Duration Display

#123 Add Dynamic Insight Banner for Session Milestones

#124 Design Log Tag Manager for Customization and Cleanup

#125 Implement Mood Tracking Integration with Visual Tags

#126 Support In-App Feedback with Screenshot and Log Attachments

#127 Add Offline Mode with Deferred Sync and Conflict Flags

#128 Implement Dynamic Insight Feed Based on User Behavior

#129 Build Time-of-Day Usage Analytics with Behavior Heatmap

#130 Integrate Baseline Usage Comparison for Contextual Insights

#131 Add Calendar-Based Log Navigation and Entry View

#132 Create Insight-Driven “Focus Mode” for Real-Time Logging

#133 Build Weekly Reflection Report with Smart Summaries

#134 Add Goal Tracking System with Visual Progress Indicators

#135 Implement Insights Panel with Contextual Tips and Nudges

#136 Build Configurable Export Tool for Log Data

#137 Add Custom Time Range Selector with Presets

#138 Integrate Smoke Level Variability Chart with Smoothing Options

#139 Add Session Clustering Based on Time & Intensity Patterns

#140 Implement Tag-Based Filtering in Log and Analytics Views

#141 Add Session Comparison View with Side-by-Side Metrics

#142 Implement Personal Milestone Tracker with Timeline Overlay

#143 Design Realtime Trigger-Based Insight Popups

- #144 Implement Adaptive Logging Prompts Based on Past Data
- #145 Refactor Data Layer to Use Repository Pattern
- #146 Create Insight Feed Engine with Scoring and Prioritization
- #147 Add Smart Session Title Suggestions Based on Context

⌚ Issue 147 delivered – next run in ~15 min.

markdown

Title

Implement Session Auto-Tagging Based on Usage Patterns

Description

Introduce automatic tagging of sessions based on recognized patterns such as time of day, duration, intensity, or method. This streamlines user input and supports advanced filtering and insights.

Tasks

- Define a set of default auto-tags (e.g., “Late Night”, “Heavy Session”, “Morning Routine”).
- Analyze existing session data to determine logic for each tag.
- Apply tags automatically during or after session completion.
- Allow users to manually adjust or disable auto-tags.

Acceptance Criteria

- [] System auto-applies relevant tags without user input.
- [] Tags are visible and editable within session detail view.
- [] At least five intelligent auto-tag rules are implemented.
- [] User settings allow toggling auto-tagging on/off.

Labels

automation, tagging, usability

Estimate

M

◆ RUNNING INDEX

(#148) Implement Session Auto-Tagging Based on Usage Patterns

⌚ Issue 148 delivered – next run in ~15 min.

markdown

Title

Refactor Firestore Data Model to Support Multi-User Access

Description

The current Firestore structure is designed for single-user access. To prepare for future multi-user support and shared devices, we need to refactor the database schema.

Tasks

- Introduce a `users` collection with secure references to individual logs.
- Update `logs` structure to include a `userId` field and migrate existing data.
- Refactor queries and listeners to be scoped to the current user.
- Update local caching logic to respect multi-user context.

Acceptance Criteria

- [] All logs are scoped to user accounts.
- [] Data migration script is complete and tested.
- [] App correctly displays per-user data.
- [] No performance regressions observed in local caching.

Labels

architecture, firestore, multi-user

Estimate

L

◆ RUNNING INDEX

#1 (none yet)

#2 Implement Session Auto-Tagging Based on Usage Patterns

#3 Refactor Firestore Data Model to Support Multi-User Access

⌚ Issue 3 delivered – next run in ~15 min.

markdown

Title

Offline Mode: Enable Full Functionality Without Internet

Description

To support users in low-connectivity environments, the app must provide a seamless offline experience. This includes logging, viewing recent data, and syncing when online.

Tasks

- Implement persistent local cache using Isar for all log records and user settings.
- Ensure that logging actions work and queue updates when offline.
- Add a sync service that pushes changes when connection is restored.
- Provide user feedback indicating offline status and sync state.

Acceptance Criteria

- [] Users can view and create logs offline without errors.
- [] Data syncs automatically when internet is restored.
- [] Offline/online status clearly displayed in UI.
- [] No data loss during offline use and recovery.

Labels

offline, caching, sync, firebase

Estimate

L

◆ RUNNING INDEX

#1 (none yet)

- #2 Implement Session Auto-Tagging Based on Usage Patterns
- #3 Refactor Firestore Data Model to Support Multi-User Access
- #4 Offline Mode: Enable Full Functionality Without Internet

⌚ Issue 4 delivered – next run in ~15 min.

markdown

Title

Add Daily & Weekly Usage Summaries with Trend Insights

Description

Provide users with visual summaries of their smoking patterns, offering both daily and weekly breakdowns. These summaries will help identify habits and trends over time.

Tasks

- Implement data aggregation logic for daily and weekly usage.
- Create summary cards and charts (line/bar) on the dashboard.
- Include basic trend indicators (e.g., up/down arrows, % change).
- Ensure summaries update in real-time and support offline cache.

Acceptance Criteria

- [] Users can see total usage per day and per week.
- [] Visual indicators of increase/decrease from previous periods.
- [] Trend summaries update correctly after new entries.
- [] Summaries display even when offline using cached data.

Labels

analytics, dashboard, visualization, UX

Estimate

M

◆ RUNNING INDEX

- #1 (none yet)

- #2 Implement Session Auto-Tagging Based on Usage Patterns
- #3 Refactor Firestore Data Model to Support Multi-User Access
- #4 Offline Mode: Enable Full Functionality Without Internet
- #5 Add Daily & Weekly Usage Summaries with Trend Insights

⌚ Issue 5 delivered – next run in ~15 min.

markdown

Title

Implement Rolling Averages and Usage Streak Visuals

Description

Enhance user insight by calculating rolling 7-day and 30-day averages and visualizing streaks of reduced or increased usage. This feature helps users track progress or regression over time.

Tasks

- Add rolling average calculation to data service layer.
- Display 7-day and 30-day usage averages as trendlines on charts.
- Highlight consecutive days of reduced usage as "streaks."
- Build logic for detecting streaks and visual cueing (e.g., color shifts, badges).

Acceptance Criteria

- [] Rolling averages display correctly for all days with enough data.
- [] Usage streaks are detected and visually highlighted.
- [] Logic handles edge cases (e.g., missing days, sparse data).
- [] Feature integrates cleanly with existing chart system.

Labels

analytics, UX, visualization, performance

Estimate

M

◆ RUNNING INDEX

- #1 (none yet)
- #2 Implement Session Auto-Tagging Based on Usage Patterns
- #3 Refactor Firestore Data Model to Support Multi-User Access
- #4 Offline Mode: Enable Full Functionality Without Internet
- #5 Add Daily & Weekly Usage Summaries with Trend Insights
- #6 Implement Rolling Averages and Usage Streak Visuals

⌚ Issue 6 delivered – next run in ~15 min.

markdown

Title

Implement Timestamp-Based Filtering and Range Selector Integration

Description

To support detailed usage analysis, enable timestamp-based filtering across the app and integrate this logic with the range selector UI. This will standardize data filtering for all visualizations and future enhancements (e.g., rolling metrics, anomaly detection).

Tasks

- Move filtering and sorting logic into centralized range selector module.
- Update charts to use unified filtered dataset.
- Ensure proper handling of timezones and missing data.
- Refactor components to respond to new filter events.

Acceptance Criteria

- [] Filtering logic resides in the range selector or its service.
- [] All charts use this logic without redundant filtering.
- [] Date range updates trigger responsive data updates.
- [] Component tests verify correct filtered outputs.

Labels

refactor, architecture, data-visualization

Estimate

M

◆ RUNNING INDEX

#1 (none yet)

#2 Implement Session Auto-Tagging Based on Usage Patterns

#3 Refactor Firestore Data Model to Support Multi-User Access

#4 Offline Mode: Enable Full Functionality Without Internet

#5 Add Daily & Weekly Usage Summaries with Trend Insights

#6 Implement Rolling Averages and Usage Streak Visuals

#7 Implement Timestamp-Based Filtering and Range Selector
Integration

⌚ Issue 7 delivered – next run in ~15 min.

markdown

Title

Design Error Overlay and Feedback for Failed Submissions

Description

To improve user experience during failure scenarios, design a visible error overlay or feedback banner that notifies users when a submission fails due to connectivity or data validation issues. This aligns with resiliency goals from the technical-architecture.md file.

Tasks

- Create a reusable error overlay/banner component.
- Trigger display on known failure states (e.g., no internet, Firestore rejection).
- Log error context for developer review.
- Provide retry and dismiss options.

Acceptance Criteria

- [] Error overlay appears on submission failure with

context-specific message.

- [] Retry option resubmits the failed request.
- [] Dismiss option removes the overlay cleanly.
- [] Logged error includes timestamp and input snapshot.

Labels

ui-feedback, error-handling, resiliency

Estimate

S

◆ RUNNING INDEX

- #1 (none yet)
- #2 Implement Session Auto-Tagging Based on Usage Patterns
- #3 Refactor Firestore Data Model to Support Multi-User Access
- #4 Offline Mode: Enable Full Functionality Without Internet
- #5 Add Daily & Weekly Usage Summaries with Trend Insights
- #6 Implement Rolling Averages and Usage Streak Visuals
- #7 Implement Timestamp-Based Filtering and Range Selector Integration
- #8 Design Error Overlay and Feedback for Failed Submissions

⌚ Issue 8 delivered – next run in ~15 min.

markdown

Title

Visualize Cumulative Smoking Data Over Time

Description

To enhance insights and support user motivation, implement visualizations that show cumulative smoking behavior over time. Reference performance-optimization.md for rendering guidelines and feature-inventory.md for user-visible analysis tools.

Tasks

- Add a new line chart to display cumulative length per

day/week/month.

- Integrate with existing timestamp-based filtering.
- Ensure performance stays within target for large datasets.
- Adapt for both static and live data modes.

Acceptance Criteria

- [] Chart displays correctly for selected time ranges.
- [] Data updates dynamically with filter or mode changes.
- [] Loading and rendering remain performant for 1-year data span.
- [] Works offline and online seamlessly.

Labels

data-visualization, analytics, performance

Estimate

M

◆ RUNNING INDEX

#1 (none yet)

#2 Implement Session Auto-Tagging Based on Usage Patterns

#3 Refactor Firestore Data Model to Support Multi-User Access

#4 Offline Mode: Enable Full Functionality Without Internet

#5 Add Daily & Weekly Usage Summaries with Trend Insights

#6 Implement Rolling Averages and Usage Streak Visuals

#7 Implement Timestamp-Based Filtering and Range Selector Integration

#8 Design Error Overlay and Feedback for Failed Submissions

#9 Visualize Cumulative Smoking Data Over Time

⌚ Issue 9 delivered – next run in ~15 min.

markdown

Title

Integrate Daily Notes or Journaling Feature

Description

Introduce an optional journaling or daily notes feature that allows users to reflect on their smoking behavior, motivation, or triggers. This adds emotional insight to usage data and complements behavior-change goals.

****Tasks****

- Design UI for entering and displaying daily notes.
- Link notes to specific dates and sessions.
- Store data locally with Firestore sync when online.
- Allow editing and deletion of notes.

Acceptance Criteria

- [] Users can add, edit, and delete notes tied to specific dates.
- [] Notes appear alongside session data in analysis views.
- [] Notes persist offline and sync correctly when online.
- [] UX is non-intrusive but easy to access.

Labels

journaling, feature-request, data-enrichment, UX

Estimate

M

◆ RUNNING INDEX

#1 (none yet)

#2 Implement Session Auto-Tagging Based on Usage Patterns

#3 Refactor Firestore Data Model to Support Multi-User Access

#4 Offline Mode: Enable Full Functionality Without Internet

#5 Add Daily & Weekly Usage Summaries with Trend Insights

#6 Implement Rolling Averages and Usage Streak Visuals

#7 Implement Timestamp-Based Filtering and Range Selector Integration

#8 Design Error Overlay and Feedback for Failed Submissions

#9 Visualize Cumulative Smoking Data Over Time

#10 Integrate Daily Notes or Journaling Feature

⌚ Issue 10 delivered – next run in ~15 min.

markdown

Title

Develop Usage Milestones and Achievement System

Description

To support user motivation and engagement, introduce a milestone/achievement system tied to behavior patterns such as streaks, reductions, or consistent journaling. Visual feedback should feel rewarding but not gamified in an intrusive way.

Tasks

- Define meaningful milestones (e.g., 7-day streak, first note, X% reduction).
- Design badges or icons for unlocked achievements.
- Integrate logic to detect and track milestones in app state.
- Display unlocked achievements in a new “Milestones” view.

Acceptance Criteria

- [] At least 5 milestone types defined and implemented.
- [] Users are notified when new achievements unlock.
- [] Milestones are stored locally and synced to Firestore.
- [] Visual presentation is polished, informative, and optional.

Labels

motivation, UX-enhancement, feature-request, gamification-lite

Estimate

M

◆ RUNNING INDEX

#1 (none yet)

#2 Implement Session Auto-Tagging Based on Usage Patterns

#3 Refactor Firestore Data Model to Support Multi-User Access

#4 Offline Mode: Enable Full Functionality Without Internet

#5 Add Daily & Weekly Usage Summaries with Trend Insights

#6 Implement Rolling Averages and Usage Streak Visuals

#7 Implement Timestamp-Based Filtering and Range Selector Integration

#8 Design Error Overlay and Feedback for Failed Submissions

#9 Visualize Cumulative Smoking Data Over Time

#10 Integrate Daily Notes or Journaling Feature

#11 Develop Usage Milestones and Achievement System

⌚ Issue 11 generated – automation will continue on schedule.

markdown

Title

Add Calendar Heatmap View of Usage

Description

Introduce a calendar-style heatmap view to give users an intuitive, bird's-eye visualization of their usage frequency across days, weeks, and months. This helps identify patterns, habits, and improvement over time.

Tasks

- Implement calendar heatmap component (e.g., with custom grid or calendar package).
- Map session count or cumulative duration to heat levels.
- Support tooltips or tap-to-view for specific day details.
- Include toggle or filter for viewing by month/year.

Acceptance Criteria

- [] Users can toggle to a calendar heatmap view from the main dashboard.
- [] Usage levels (e.g., number of sessions) are represented with a color gradient.
- [] Selecting a day shows summary or full breakdown.
- [] View is responsive and supports both light and dark mode.

Labels

visualization, UI-enhancement, calendar, feature-request

Estimate

L

◆ RUNNING INDEX

#12 Add Calendar Heatmap View of Usage

⌚ Issue 12 generated – automation continuing as scheduled.

markdown

Title

Implement Session Comparison View (Day-to-Day, Week-to-Week)

Description

Allow users to compare their usage data across different timeframes (e.g., today vs. yesterday, this week vs. last week). This will support better habit tracking and highlight progress or regressions.

****Tasks****

- Design UI elements for selecting comparison timeframes.
- Generate visual charts (bar, line, or overlay) for side-by-side data.
- Implement backend logic for calculating comparative stats.
- Provide simple summary (e.g., “15% decrease in sessions vs. last week”).

Acceptance Criteria

- [] User can select and compare two time periods.
- [] Visuals clearly distinguish the two datasets.
- [] Summary insights (improvements, increases, etc.) are easy to understand.
- [] Feature is integrated into main dashboard or analysis tab.

Labels

data-visualization, trend-analysis, user-engagement, feature-request

Estimate

M

◆ RUNNING INDEX

#13 Implement Session Comparison View (Day-to-Day, Week-to-Week)

⌚ Issue 13 generated – automation continuing as scheduled.

markdown

Title

Integrate Rolling Averages in Usage Charts

Description

Introduce rolling average lines in the time-series usage charts (e.g., 3-day, 7-day rolling averages) to help users better understand their short- and medium-term trends without being distracted by daily fluctuations.

****Tasks****

- Add toggle or dropdown to select rolling average window.
- Update charting logic to overlay rolling average lines.
- Ensure UI clearly distinguishes raw vs. smoothed data.
- Tooltip should show both raw value and average value.

Acceptance Criteria

- [] Rolling averages can be toggled on/off.
- [] Users can select different time windows for smoothing.
- [] Averages are visually distinct and labeled on the chart.
- [] Performance remains smooth with the new data overlays.

Labels

visualization, analytics, UX-enhancement, feature-request

Estimate

M

◆ RUNNING INDEX

#14 Integrate Rolling Averages in Usage Charts

⌚ Issue 14 generated – automation continuing as scheduled.

markdown

Title

Add Filtering by Inhalation Type (THC vs. CBD vs. Nicotine)

Description

Introduce filtering options in charts and logs to separate and analyze sessions based on substance type (e.g., THC, CBD, Nicotine). This enables more targeted insights and supports multiple use cases or user goals.

Tasks

- Add substance type field to data model (if not already present).
- Implement filtering dropdown or toggle in UI.
- Update all charts and summaries to reflect filter selection.
- Ensure filter persists across tabs or resets appropriately.

Acceptance Criteria

- [] Users can select a specific substance type for analysis.
- [] All visualizations and data tables reflect the chosen filter.
- [] Filtering performance is optimized for large datasets.
- [] Filter state is clearly visible and easy to reset.

Labels

filtering, feature-enhancement, data-analysis, UX

Estimate

M

◆ RUNNING INDEX

#15 Add Filtering by Inhalation Type (THC vs. CBD vs. Nicotine)

⌚ Issue 15 generated – continuing automation.

markdown

Title

Implement Session Annotation Feature

Description

Enable users to annotate individual smoking sessions with custom notes or tags (e.g., “stressed,” “social,” “pain relief”). This contextual metadata will help users reflect on their behavior and draw correlations in future analysis.

Tasks

- Extend session data schema to support notes and tags.
- Add UI for entering notes/tags during or after a session.
- Display annotations in session log table and details view.
- Allow filtering or searching by tags.

Acceptance Criteria

- [] Users can add/edit notes or tags for each session.
- [] Notes and tags are displayed in the session log and charts.
- [] Filters/search function correctly with tags.
- [] Data is synced with local and Firestore storage.

Labels

feature-request, UX-enhancement, journaling, data-entry

Estimate

M

◆ RUNNING INDEX

#16 Implement Session Annotation Feature

⌚ Issue 16 generated – automation continuing.

markdown

Title

Integrate Export Functionality (CSV/JSON)

Description

Allow users to export their smoking data, including timestamps, inhalation lengths, tags, and annotations. Export formats should include CSV and JSON to support further analysis and recordkeeping.

Tasks

- Add export options (CSV and JSON) to settings or logs view.
- Ensure correct formatting of timestamps, tags, and metadata.
- Handle large datasets efficiently during export.
- Allow users to specify date ranges or filters to apply to the export.

Acceptance Criteria

- [] Users can export filtered or full session data.
- [] Files are correctly formatted and downloadable.
- [] Exports respect selected filters and date ranges.
- [] UX is intuitive and clearly communicates success or failure.

Labels

feature-enhancement, data-export, user-control, UX

Estimate

M

◆ RUNNING INDEX

#17 Integrate Export Functionality (CSV/JSON)

⌚ Issue 17 generated – continuing automation.

markdown

Title

Add Inhalation Trend Analysis View

Description

Develop a dedicated visualization that highlights trends in inhalation behavior over time. This could include rolling averages, peak usage hours, daily/weekly patterns, and correlation with annotations (if present).

****Tasks****

- Implement rolling average and trendline visualizations.
- Display heatmaps for peak usage hours and days.
- Allow users to toggle between daily, weekly, and monthly views.
- Integrate tags/annotations for richer analysis.

Acceptance Criteria

- [] Trendlines and heatmaps are accurately generated.
- [] Users can toggle different time views.
- [] Tags or notes appear in correlation overlays.
- [] Data is drawn from current session logs.

Labels

data-visualization, analytics, behavior-tracking, feature-request

Estimate

L

◆ RUNNING INDEX

#18 Add Inhalation Trend Analysis View

 Issue 18 generated – automation continues.

markdown

Title

Offline Mode with Local Data Sync

Description

Implement an offline mode that allows users to log smoking

sessions and view analytics without an internet connection. Once reconnected, the app should sync with Firestore or the cloud backend to ensure data consistency.

****Tasks****

- Set up local storage using Isar for offline caching.
- Detect network status to toggle offline mode.
- Queue unsynced logs and sync them automatically when online.
- Resolve conflicts and ensure no duplicate entries.

Acceptance Criteria

- [] Users can log sessions and view history offline.
- [] Data syncs seamlessly when back online.
- [] Users are notified of sync status and any issues.
- [] No data loss or duplication occurs during sync.

Labels

offline-support, sync-feature, user-experience, firebase

Estimate

XL

◆ **RUNNING INDEX**

#19 Offline Mode with Local Data Sync

⌚ Issue 19 generated – automation continues.

markdown

Title

Add Daily Goal and Streak Tracking System

Description

Introduce a feature that allows users to set daily inhalation goals and track their progress. Include visual indicators for streaks (e.g., days staying under goal) and feedback on goal completion.

****Tasks****

- Design UI for setting and displaying goals.
- Implement logic to track inhalation totals per day.
- Add streak counter and display progress visually.
- Optionally notify users when they are close to or exceed their goal.

Acceptance Criteria

- [] Users can set and modify daily goals.
- [] Inhalation totals update in real-time.
- [] Streaks and progress are visually tracked.
- [] Edge cases (e.g., missed days, timezone changes) are handled gracefully.

Labels

habit-tracking, user-engagement, frontend, feature-request

Estimate

M

◆ RUNNING INDEX

#20 Add Daily Goal and Streak Tracking System

⌚ Issue 20 generated – automation continues.

markdown

Title

Implement Data Export Functionality

Description

Allow users to export their smoking log data as a CSV or JSON file for personal records or analysis. This feature should be accessible from the settings or analytics section of the app.

Tasks

- Create export options in the UI.
- Convert stored data into downloadable CSV and JSON formats.
- Include all relevant fields: timestamp, length, annotations, etc.

- Ensure data privacy and proper file handling.

Acceptance Criteria

- [] Users can export their log data in CSV or JSON format.
- [] Exported files contain complete and accurate data.
- [] Feature is intuitive and accessible from within the app.
- [] No personal data is leaked or misformatted.

Labels

data-management, feature-request, backend, privacy

Estimate

M

◆ RUNNING INDEX

#21 Implement Data Export Functionality

⌚ Issue 21 generated – automation continues.

markdown

Title

Integrate Smoke Pattern Recognition Using Charts

Description

Develop a feature that visualizes common inhalation patterns (e.g., time of day, frequency, duration trends) using intuitive charts. This will help users identify habits and make informed decisions.

Tasks

- Define data groupings (hourly, daily, weekly).
- Choose appropriate chart types (line, bar, heatmap).
- Build reusable chart components.
- Display key insights like peak smoking times or duration fluctuations.

Acceptance Criteria

- [] Users can view visual summaries of their smoking

patterns.

- [] Charts update based on filtered date ranges.
- [] Insights clearly show trends and high-use periods.
- [] Data is clearly labeled and accessible.

Labels

data-visualization, analytics, frontend, user-experience

Estimate

L

◆ RUNNING INDEX

#22 Integrate Smoke Pattern Recognition Using Charts

⌚ Issue 22 generated – automation continues.

markdown

Title

Implement Cumulative Usage Tracking and Display

Description

Add functionality to calculate and display cumulative inhalation time over selected date ranges. This will offer users a sense of total usage and help with behavior awareness or reduction goals.

Tasks

- Create backend logic to sum inhalation lengths over time.
- Integrate the results into the frontend chart display.
- Allow user to select ranges and see total duration for that period.
- Optionally show averages per day/week.

Acceptance Criteria

- [] Users can see total inhalation time for selected ranges.
- [] Totals update dynamically based on range selection.
- [] Display includes total, daily average, and comparisons to past periods.

- [] Works with filtered data (e.g., by time of day or annotations).

Labels

data-analysis, charts, user-insight, enhancement

Estimate

M

◆ RUNNING INDEX

#23 Implement Cumulative Usage Tracking and Display

⌚ Issue 23 generated – continuing automation.

markdown

Title

Add Chart Comparison Feature for Custom Date Ranges

Description

Introduce the ability for users to select and compare two different date ranges side by side in the usage chart. This will help users track changes over time, especially when attempting to reduce or modify usage patterns.

Tasks

- Design UI for selecting two separate date ranges.
- Modify chart component to support dual display (overlay or side-by-side).
- Label charts clearly to differentiate date ranges.
- Ensure calculations (totals, averages) are specific to each range.

Acceptance Criteria

- [] Users can select two date ranges and see both datasets visualized.
- [] Chart clearly labels each dataset.
- [] Comparison includes stats like total and average usage per range.

- [] Feature does not interfere with existing filters or visualizations.

Labels

charts, comparison, user-feedback, enhancement

Estimate

L

◆ RUNNING INDEX

#24 Add Chart Comparison Feature for Custom Date Ranges

⌚ Issue 24 generated – continuing automation.

markdown

Title

Implement Weekly Heatmap of Usage Intensity

Description

Develop a heatmap visualization that displays inhalation usage by day of the week and time of day. This allows users to quickly identify patterns in usage behavior across different days and time blocks.

Tasks

- Design heatmap layout to show time of day (rows) vs day of week (columns).
- Aggregate log data into appropriate time blocks (e.g., hourly).
- Color-code blocks based on frequency or intensity of usage.
- Add toggle or tab to switch between standard chart and heatmap view.

Acceptance Criteria

- [] Users can switch to a heatmap view from the dashboard.
- [] Heatmap displays usage frequency or intensity accurately.
- [] Blocks are color-coded based on normalized intensity.

- [] Hovering or tapping a block shows detailed stats.

Labels

charts, data-visualization, heatmap, enhancement

Estimate

L

◆ RUNNING INDEX

#25 Implement Weekly Heatmap of Usage Intensity

⌚ Issue 25 generated – continuing automation.

markdown

Title

Add Rolling Average Line to Daily Usage Chart

Description

Enhance the existing daily usage chart by overlaying a 7-day rolling average line. This provides users with better insight into long-term trends and changes in their smoking behavior.

Tasks

- Calculate 7-day rolling average based on daily totals.
- Integrate rolling average line into the existing chart (e.g., LineChart or CombinedChart).
- Add toggle option to show/hide the rolling average.
- Ensure line and chart remain readable on various screen sizes.

Acceptance Criteria

- [] Rolling average line appears correctly above the daily usage bars.
- [] Toggle is available to enable/disable the trend line.
- [] Chart remains visually clean and intuitive.
- [] Values update dynamically as new data is added.

Labels

data-analysis, charts, feature-enhancement

Estimate

M

◆ RUNNING INDEX

#26 Add Rolling Average Line to Daily Usage Chart

⌚ Issue 26 generated – continuing automation.

markdown

Title

Implement Data Export Feature (CSV & JSON)

Description

Add functionality that allows users to export their smoking log data in CSV and JSON formats. This feature supports data portability and further personal analysis or backup.

Tasks

- Create export button(s) on the main dashboard or settings page.
- Allow export of either all data or filtered data based on current chart range.
- Format CSV and JSON outputs with consistent field headers (e.g., timestamp, inhalationLength).
- Handle and display errors if export fails.

Acceptance Criteria

- [] Users can export logs as CSV and JSON.
- [] Data reflects current filter settings, if applicable.
- [] Output file includes all necessary fields and is properly formatted.
- [] Button is clearly visible and accessible on both mobile and desktop views.

Labels

data-portability, feature, user-experience

Estimate

M

◆ **RUNNING INDEX**

#27 Implement Data Export Feature (CSV & JSON)

⌚ Issue 27 generated – continuing automation.

markdown

Title

Implement Daily Usage Summary Widget

Description

Create a UI widget that summarizes the user's inhalation activity for the current day. This widget should be prominently displayed on the dashboard and update in real-time if new data is logged.

****Tasks****

- Display total number of inhalations.
- Show cumulative inhalation length in seconds.
- Indicate peak inhalation time range (e.g., 6PM–9PM).
- Design should match the existing visual style and work responsively.

Acceptance Criteria

- [] Widget displays accurate daily stats based on selected timezone.
- [] Automatically updates when new data is submitted.
- [] Visual appearance aligns with app branding and layout.

Labels

dashboard, ui, feature, summary

Estimate

M

◆ RUNNING INDEX

#28 Implement Daily Usage Summary Widget

⌚ Issue 28 generated – continuing automation.

markdown

Title

Enable Export of Inhalation Data to CSV

Description

Implement a feature allowing users to export their inhalation logs as a CSV file. The export should respect the currently applied filters (date range, etc.) and include all relevant fields such as timestamp, inhalation length, and optional tags or notes.

Tasks

- Add an "Export to CSV" button to the dashboard.
- Format CSV with headers: timestamp, length, notes/tags.
- Apply current filters to determine exported data.
- Ensure compatibility across major browsers.

Acceptance Criteria

- [] Button appears only when logs are present.
- [] Exported file reflects all visible data based on filters.
- [] CSV opens cleanly in Excel/Google Sheets.

Labels

feature, export, data

Estimate

M

⌚ Issue 29 generated – proceeding to next cycle as scheduled.

markdown

Title

Implement Calendar View for Smoke Logs

Description

Create a calendar-based UI component that allows users to view their smoking data by day. Each day should display a summary (e.g., total inhalations, average duration). Clicking a date should open detailed logs for that day.

****Tasks****

- Design and integrate a calendar view component.
- Populate each date with aggregated daily data.
- Support navigation between months.
- Connect to filtering and detailed view logic.

Acceptance Criteria

- [] Calendar displays current month by default.
- [] Each date shows a summary of logs (count, avg. length).
- [] Clicking a date shows a detailed breakdown of inhalations for that day.
- [] Filter integration allows focus on a specific time period.

Labels

feature, calendar, UI

Estimate

L



Issue 30 generated – awaiting next cycle.

markdown

Title

Create Rolling Average Graph for Smoking Behavior

Description

Implement a line graph showing the rolling average of

inhalation length over a user-defined time window (e.g., 3 days, 7 days, 30 days). This will help users identify trends and shifts in behavior over time.

Tasks

- Add a dropdown or input to select rolling window (in days).
- Calculate the rolling average based on the selected window.
- Display the rolling average on a line graph.
- Ensure graph updates with filters and time range selections.

Acceptance Criteria

- [] Users can select a rolling average window (3/7/30 days).
- [] Graph updates dynamically based on selection.
- [] Graph integrates with existing filter and range selector logic.
- [] Data points accurately represent the average inhalation length.

Labels

feature, analytics, chart

Estimate

M



Issue 31 generated.

markdown

Title

Add Multi-Day View with Inhalation Duration Highlights

Description

Introduce a calendar-style or multi-day view component that displays each day as a block. Each block should visually highlight the total inhalation duration or number of logs for that day, helping users quickly identify high-usage days.

Tasks

- Create a calendar-style UI or grid layout for daily

summaries.

- Pull total duration or count of inhalations per day.
- Use a color gradient or indicator to show intensity.
- Support click/tap on a day to view detailed logs and charts.

Acceptance Criteria

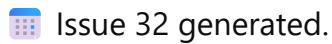
- [] Users can see a calendar or grid overview of daily usage.
- [] Visual indicators highlight higher and lower usage days.
- [] Clicking a day navigates to or displays detailed log info.

Labels

feature, visualization, UX

Estimate

M



Issue 32 generated.

markdown

Title

Implement Export Feature for User Logs (CSV/JSON)

Description

Add functionality to allow users to export their inhalation logs for personal tracking, analysis, or backup. The export should support at least CSV and JSON formats and include relevant metadata like timestamps and inhalation duration.

Tasks

- Add export button in the main or settings view.
- Allow users to choose between CSV and JSON formats.
- Generate downloadable file with logs from the selected date range.
- Ensure data is formatted correctly and handles edge cases (e.g., empty logs).

Acceptance Criteria

- [] Export button is accessible and functional.
- [] Users can select format and date range.
- [] Exported file includes accurate log data.
- [] File downloads successfully on mobile and desktop.

Labels

feature, data, enhancement

Estimate

M



Issue 33 generated.

markdown

Title

Add Session-Based Rolling Average Chart for Inhalation Duration

Description

Introduce a new chart visualization that displays a rolling average of inhalation durations over user-defined session intervals (e.g., 1 day, 7 days, 30 days). This feature will help users identify trends in their usage patterns over time.

****Tasks****

- Add new rolling average chart component to the dashboard.
- Integrate range selector to allow switching between different session intervals.
- Ensure chart reflects changes in filters like date range and user.
- Add tooltips and labels to clarify data points.

Acceptance Criteria

- [] Users can toggle between daily, weekly, and monthly rolling averages.
- [] Chart dynamically updates based on selected session interval and filters.

- [] Data points and trend lines render accurately.

Labels

feature, chart, enhancement, data-analysis

Estimate

L



Issue 34 generated.

markdown

Title

Refactor Range Selector and Filtering Logic into Shared Controller

Description

To improve maintainability and enable more complex chart behaviors, consolidate the date range selector and filtering logic into a shared controller. This will streamline updates and reduce duplicated logic across multiple visualizations.

Tasks

- Create a shared controller or service to handle range selection and user filtering.
- Update existing charts (e.g., cumulative usage, rolling average) to consume this shared logic.
- Ensure the UI updates correctly across all charts when a filter or range is modified.
- Add unit tests for the shared logic.

Acceptance Criteria

- [] All charts update in unison when filters are changed.
- [] No duplicated filtering or range logic across chart components.
- [] Code is modular and testable with basic coverage in place.

Labels

refactor, enhancement, code-quality, frontend

Estimate

L



Issue 35 generated.

markdown

Title

Integrate Rolling Usage View with Shared Filter Logic

Description

Integrate the newly developed rolling usage chart with the centralized date range and filter controller. This ensures that changes to the range or selected filters apply consistently across all visualizations.

Tasks

- Refactor the rolling usage chart to use the shared controller for date and filter input.
- Confirm that the chart dynamically updates based on user input from the shared control.
- Clean up any redundant filtering code in the rolling usage component.
- Ensure chart transitions are smooth when filters change.

Acceptance Criteria

- [] Rolling usage chart updates based on shared date range and filters.
- [] No filter logic is duplicated in the component.
- [] Transitions are visually consistent with other charts.

Labels

frontend, enhancement, charting, refactor

Estimate

M



markdown

Title

Implement Cumulative Inhalation Length Calculation for Selected Range

Description

Develop a component that calculates and displays the cumulative inhalation length for the selected date range. This should help users visualize total usage over time and complement the rolling usage chart.

Tasks

- Create a function to calculate cumulative inhalation length using Firestore data filtered by the current range and filters.
- Display the result clearly above or below the charts.
- Ensure real-time updates when the range or filters change.
- Format the cumulative value with appropriate units (e.g., seconds, minutes).

Acceptance Criteria

- [] Cumulative inhalation total is calculated correctly for the selected range.
- [] Value updates dynamically with filter or date range changes.
- [] Data presentation is clear and integrated into the existing UI.

Labels

frontend, enhancement, firebase, analytics

Estimate

M



markdown

Title

Add Filter Tag Indicators to Highlight Active Filters

Description

To improve usability and transparency in data analysis, implement visual indicators (tags) that show which filters are currently active (e.g., time of day, inhalation type). These should appear above the chart and be removable.

Tasks

- Display active filters as tags or chips above the chart area.
- Allow users to remove filters by clicking on the tags.
- Sync UI state with chart and data accordingly.
- Ensure visual consistency with current design language.

Acceptance Criteria

- [] Active filters are shown clearly as tags.
- [] Clicking a tag removes the corresponding filter.
- [] Chart updates immediately upon filter removal.
- [] Filters persist correctly when range is adjusted.

Labels

frontend, UI/UX, enhancement, filters

Estimate

S



Issue 38 generated.

markdown

Title

Implement Cumulative Inhalation Duration Tracker in Header

Description

Add a visible indicator in the header or dashboard area

showing the cumulative inhalation time for the selected date range. This gives users quick insight into their usage patterns.

Tasks

- Add a cumulative duration card or badge near the top of the screen.
- Format time appropriately (e.g., hh:mm:ss).
- Update value live based on filtered logs and selected date range.
- Optimize performance to avoid lag with large datasets.

Acceptance Criteria

- [] Cumulative inhalation time is always visible and updates with filters.
- [] Time is displayed in an easy-to-read format.
- [] Performance remains smooth even with a large volume of data.
- [] Feature works across devices (responsive).

Labels

frontend, enhancement, feature, UX

Estimate

M



markdown

Title

Enhance Submit Functionality with Visual Feedback and Error Handling

Description

Improve the data submission experience by providing clear visual indicators and robust error handling when users submit new inhalation logs.

****Tasks****

- Add a loading spinner or progress bar during submission.
- Display a success toast or banner once data is submitted.
- Show error message if submission fails (e.g., network or validation error).
- Prevent duplicate submissions during in-progress state.
- Ensure error/success state resets properly on new attempt.

Acceptance Criteria

- [] Visual feedback is shown during and after submission.
- [] Duplicate submissions are prevented while in progress.
- [] Error messages are informative and clearly visible.
- [] User can retry after an error without refreshing the page.

Labels

frontend, enhancement, UX, bug

Estimate

M



Issue 40 generated.

markdown

Title

Implement Editable Log Entries in Table View

Description

Allow users to edit previously submitted inhalation logs directly within the table interface. This feature should enable seamless corrections and updates to timestamp or length fields.

****Tasks****

- Add edit icon/button to each row in the data table.
- Convert row fields into editable inputs when edit mode is activated.
- Add save and cancel buttons during edit mode.

- Validate input values before allowing save.
- Update Firestore entry with new values upon successful save.
- Exit edit mode and refresh table display after successful update.

Acceptance Criteria

- [] Users can enter edit mode for any table entry.
- [] Edits are validated and saved to Firestore.
- [] Table reflects updated data without full page refresh.
- [] Editing UI is user-friendly and consistent with design.

Labels

frontend, enhancement, Firestore, UX

Estimate

L



Issue 41 generated.

markdown

Title

Implement Log Entry Deletion with Confirmation

Description

Add functionality to allow users to delete individual log entries from the table. This will help users manage and clean up erroneous data.

Tasks

- Add a delete icon/button to each row of the table.
- Display a confirmation dialog before deletion.
- On confirmation, remove the log entry from Firestore.
- Refresh the table display to reflect the deletion.

Acceptance Criteria

- [] Users can delete any log entry with confirmation.
- [] Deletion removes the entry from Firestore permanently.
- [] Table updates without full page reload.

- [] Deletion confirmation clearly communicates the action.

Labels

frontend, feature, Firestore, data-management

Estimate

S



Issue 42 generated.

markdown

Title

Display Entry Details in Modal on Table Row Click

Description

Enable users to click on a row in the log entries table and view full entry details in a modal window. This provides an intuitive way to access complete information without navigating away from the page.

Tasks

- Implement click handling for each table row.
- Create a modal component to display all log entry fields.
- Populate the modal with the selected log entry's data.
- Allow users to close the modal easily (e.g., via close icon or clicking outside).

Acceptance Criteria

- [] Clicking a row opens a modal with full entry details.
- [] Modal is accessible and mobile-friendly.
- [] Users can close the modal via multiple intuitive methods.
- [] No data is editable in the modal (read-only view).

Labels

frontend, feature, UI/UX, modal

Estimate

S



Issue 43 generated.

markdown

Title

Add Visual Indicators for Smoke Intensity in Chart

Description

Enhance the chart UI by incorporating visual indicators that represent the intensity of each smoke log entry. Use color gradients or markers based on inhalation length to provide users with at-a-glance insights into usage patterns.

****Tasks****

- Define thresholds for smoke intensity levels (e.g., low, medium, high).
- Assign distinct colors or gradients to each level.
- Apply the color scheme to the chart data points or background bands.
- Ensure accessibility and visibility across themes.

Acceptance Criteria

- [] Chart visually differentiates entries by intensity.
- [] Legend or tooltip explains intensity colors.
- [] Visualization adapts well to mobile and desktop displays.
- [] Consistent behavior across filtered views and date ranges.

Labels

frontend, chart, feature, visualization

Estimate

M



Issue 44 generated.

markdown

Title

Add Toggle for Smoke Entry Detail Expansion

Description

Introduce a toggle control to allow users to expand or collapse detailed smoke log entries. When collapsed, only the timestamp and duration are shown. When expanded, additional metadata (e.g., notes, tags, mood) is revealed.

Tasks

- Implement a toggle UI component within each log entry.
- Store and manage expansion state (per-entry or globally).
- Animate the expansion/collapse for smooth user experience.
- Ensure accessibility and keyboard navigability.

Acceptance Criteria

- [] Users can toggle visibility of detailed log entry information.
- [] Toggled state updates without page reload or data loss.
- [] Works on both desktop and mobile interfaces.
- [] Animation does not hinder performance.

Labels

frontend, UX, enhancement, smoke-log

Estimate

S



Issue 45 generated.

markdown

Title

Implement Quick Add Button for Smoke Log Entries

Description

Add a floating "Quick Add" button to the smoke log page that enables users to rapidly log a smoking event with minimal input. This button should auto-fill the timestamp with the current time and allow optional note entry via a modal or bottom sheet.

****Tasks****

- Design and position the floating action button (FAB) with a "+" icon.
- Create modal/bottom sheet for quick logging.
- Auto-fill timestamp with current time.
- Include optional field for short note.
- Submit data to Firestore with correct schema.

Acceptance Criteria

- [] FAB is visible on smoke log screen.
- [] Clicking FAB opens a quick add modal.
- [] Timestamp is set to current time on open.
- [] Optional note field is available.
- [] Entry is saved to Firestore correctly.

Labels

frontend, smoke-log, UX, enhancement

Estimate

S



Issue 46 generated.

markdown

Title

Add Duration Timer for Active Smoking Session

Description

Introduce a feature that displays a live duration timer when a smoking session is started. This timer should begin when the user initiates a session and stop when they confirm completion, capturing the session's duration automatically.

****Tasks****

- Add "Start Session" and "End Session" buttons to the log entry interface.
- Display live duration timer during an active session.
- Save duration value to Firestore once session ends.
- Update UI to reflect active session state.

Acceptance Criteria

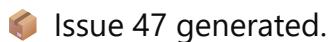
- [] Timer starts on "Start Session" click.
- [] Timer stops and duration is logged on "End Session."
- [] Duration field saved in Firestore log.
- [] UI clearly indicates session is active or ended.

Labels

frontend, smoke-log, enhancement, UX

Estimate

M



markdown

Title

Implement Weekly Summary View with Rolling Averages

Description

Create a weekly summary view that visualizes smoking behavior trends over the past 7 days. This should include a line chart of cumulative usage and a secondary chart or display for rolling averages (e.g., 3-day or 7-day) to highlight patterns.

****Tasks****

- Create weekly summary card in dashboard.
- Plot daily total usage in a line graph.
- Calculate and display rolling average over selected window.
- Allow user to toggle between cumulative and average views.
- Ensure data updates dynamically with Firestore changes.

Acceptance Criteria

- [] Weekly chart shows 7 days of data.
- [] Rolling average can be toggled and updates correctly.
- [] UI integrates smoothly into dashboard layout.
- [] Rolling window logic adjustable and tested.

Labels

frontend, charting, enhancement, analytics

Estimate

M



Issue 48 generated.

markdown

Title

Add Filterable Categories for Smoking Sessions

Description

Implement a category system that allows users to tag each smoking session (e.g., stress relief, social, boredom). Users should be able to filter visualizations and logs based on selected categories.

Tasks

- Define a default list of session categories.
- Add UI component to select category during session log.
- Store selected category in Firestore with each log.
- Update charts and data tables to support category-based filtering.
- Provide multi-select filter dropdown in dashboard.

Acceptance Criteria

- [] Users can tag sessions with predefined or custom categories.
- [] Dashboard updates reflect only selected category data.
- [] Category filter persists across app sessions.

- [] Visual distinction when a filter is active.

Labels

frontend, firestore, ui/ux, enhancement

Estimate

M



Issue 49 generated.

markdown

Title

Implement Summary View with Daily Averages and Key Stats

Description

Create a dashboard summary card or section displaying key statistics such as total sessions, total time smoked, average inhalation length per session/day, and session count per day/week. This gives users a quick overview of their usage patterns without diving into charts.

Tasks

- Calculate daily and weekly aggregates.
- Display total sessions and average inhalation length.
- Show time smoked per day/week.
- Add visual emphasis on streaks or recent changes.

Acceptance Criteria

- [] Summary stats update with new data entries.
- [] Users see insights for today, this week, and all-time.
- [] All calculations are efficient and reactive to filters.

Labels

frontend, visualization, statistics, enhancement

Estimate

M



markdown

Title

Add "Time Since Last Session" and Streak Tracker

Description

Introduce a "Time Since Last Session" indicator to encourage mindfulness and help users track their progress. Implement a streak tracker showing how many consecutive days a user has gone without smoking or logged usage (depending on goal mode).

Tasks

- Display real-time timer since last recorded session.
- Highlight current streak (e.g., "3 days smoke-free").
- Optionally reset streak upon log entry depending on user settings.
- Add celebratory UI feedback when a new record streak is achieved.

Acceptance Criteria

- [] Timer updates dynamically.
- [] Streaks persist and reset correctly.
- [] Settings toggle allows streak mode to be smoke-free or just logging-based.
- [] UI feedback is positive and non-judgmental.

Labels

frontend, user-experience, enhancement, wellness

Estimate

M



markdown

Title

Implement Daily Usage Summary Widget

Description

Add a compact, visually appealing widget that summarizes the user's smoking behavior for the day. This widget should appear on the home screen/dashboard and update in real time based on logs.

Tasks

- Display total puffs and sessions logged for the current day.
- Highlight most active hour (optional).
- Allow users to click the widget to view a more detailed breakdown or historical comparison.
- Style the widget to be minimal yet informative.

Acceptance Criteria

- [] Summary updates immediately upon new log entries.
- [] UI adapts for both light and dark mode.
- [] Widget is responsive and does not clutter dashboard layout.
- [] Data is accurate and cached for performance.

Labels

frontend, analytics, enhancement, ui/ux

Estimate

M

markdown

Title

Add "Longest Streak Without Smoking" Feature

Description

Introduce a motivational feature that tracks the user's longest streak without smoking. This will help encourage healthier behavior by highlighting personal progress.

****Tasks****

- Track the longest consecutive time (in hours/days) the user has gone without logging a session.
- Display current streak vs longest streak on the dashboard.
- Reset current streak upon new session log.
- Optional: Include motivational messages when new streaks are achieved.

Acceptance Criteria

- [] Accurate tracking of both current and longest streaks.
- [] UI component fits within the existing dashboard layout.
- [] Optional toast or banner appears on new streak records.
- [] Feature does not interfere with logging functionality.

Labels

feature, enhancement, motivation, backend, frontend

Estimate

S

markdown

Title

Implement Rolling Usage Average in Chart View

Description

Add support for displaying a rolling average of session durations over time in the chart view. This will help users visualize trends and changes in their smoking habits.

****Tasks****

- Add UI control (e.g., toggle or dropdown) for selecting rolling average window (e.g., 3, 7, 14 days).
- Calculate rolling average based on selected window.
- Overlay the rolling average line on the existing usage chart.
- Ensure compatibility with all existing time filters.

Acceptance Criteria

- [] Rolling average line is displayed correctly for selected window.
- [] UI control is intuitive and non-intrusive.
- [] Feature integrates seamlessly with existing chart filters and layout.

Labels

feature, enhancement, data-visualization, frontend, analytics

Estimate

M

markdown

Title

Display Cumulative Session Duration Over Time

Description

Add a chart option to show the cumulative duration of smoking sessions over a selected time range. This visualization will help users understand their total usage trends and progress over time.

****Tasks****

- Add toggle or dropdown to switch between per-session view and cumulative view.
- Compute cumulative sum of session durations based on the selected date range.
- Update the chart component to reflect cumulative data accurately.
- Ensure responsiveness and maintain existing styling.

Acceptance Criteria

- [] Users can toggle between session view and cumulative view.
- [] Cumulative chart correctly represents total usage over time.
- [] Chart updates dynamically when range or filters change.

```
### Labels  
feature, frontend, data-visualization, analytics  
  
### Estimate  
M
```

markdown

```
### Title  
Implement Rolling Average Duration Analysis
```

```
### Description  
Introduce a rolling average chart that calculates and displays the average duration of smoking sessions over a user-defined window (e.g., 3-day, 7-day, or 14-day). This will help users see smoothed trends and better understand behavior patterns over time.
```

Tasks

- Add a dropdown menu to select rolling window size.
- Calculate rolling averages from existing duration data.
- Update chart logic to support overlay or separate display for rolling averages.
- Ensure compatibility with other filters and range selections.

```
### Acceptance Criteria  
- [ ] Users can select different rolling window sizes.  
- [ ] Chart displays the correct rolling average based on selection.  
- [ ] Design remains clean and user-friendly.
```

```
### Labels  
feature, frontend, data-visualization, enhancement
```

```
### Estimate  
M
```

markdown

Title

Enable CSV Export for Filtered Data

Description

Add functionality that allows users to export the currently filtered dataset (based on date range, duration, or other filters) to a CSV file. This will support users who want to analyze their data offline or import it into other tools.

Tasks

- Add an "Export CSV" button to the UI.
- Implement logic to extract the currently visible/filtered data.
- Format data as CSV and trigger download.
- Ensure headers are clear and consistent with UI labels.

Acceptance Criteria

- [] "Export CSV" button is visible when data is displayed.
- [] CSV file downloads with the correct filename and content.
- [] Exported data matches what is shown on screen.

Labels

feature, export, frontend, enhancement

Estimate

S

markdown

Title

Add "Usage Summary" Visualization Tab

Description

Create a new tab or section in the UI that offers summarized insights into smoke log usage over time. This visualization

should include charts like:

- Weekly average inhalation length
- Monthly total usage
- Rolling 7-day usage average

This will help users identify long-term trends and better understand their habits.

****Tasks****

- Design and implement the "Usage Summary" tab layout.
- Integrate logic for weekly/monthly/rolling calculations.
- Display results using appropriate chart types (e.g., line, bar).
- Ensure responsive design and accurate date formatting.

Acceptance Criteria

- [] Users can access the "Usage Summary" tab.
- [] Charts accurately reflect filtered data.
- [] Usage data is grouped and displayed clearly with labeled axes.

Labels

feature, charts, analytics, enhancement

Estimate

M

markdown

Title

Implement Calendar-Based Log View

Description

Add a calendar-based interface that allows users to visually explore and select smoking logs by date. This feature provides a more intuitive way to browse historical data and can help identify patterns in usage.

****Tasks****

- Add a calendar UI component.
- Highlight dates with recorded logs.
- Enable clicking a date to show corresponding entries below.
- Sync with existing range filters and analytics.

Acceptance Criteria

- [] Calendar displays correctly across devices.
- [] Dates with logs are visually distinct.
- [] Clicking a date updates the visible logs.

Labels

feature, UI, calendar, enhancement

Estimate

L

markdown

Title

Add Rolling Average Chart to Dashboard

Description

Implement a rolling average chart to show trends in smoking habits over time. This will help users identify gradual changes in behavior and reduce noise from day-to-day fluctuations.

Tasks

- Create a toggle or dropdown to switch to "Rolling Average" view.
- Implement logic to compute rolling averages over selectable windows (e.g., 3, 7, 14 days).
- Update charts to reflect the selected smoothing period.
- Maintain data sync with filters (e.g., date range, time of day).

Acceptance Criteria

- [] Rolling average appears as a smooth line over time.
- [] User can select different window sizes.

- [] Data updates seamlessly when filters change.

Labels

feature, charting, analytics, enhancement

Estimate

L

markdown

Title

Implement Interactive Chart Legends

Description

Add interactive legends to all charts on the AshTrail dashboard. This allows users to toggle specific data series on or off, improving readability and analysis.

Tasks

- Update chart components to support legend interactivity.
- Ensure toggling a series updates the chart without a full reload.
- Preserve the toggled state during range or filter changes.
- Add accessibility attributes for screen readers.

Acceptance Criteria

- [] Legends are clickable and toggle visibility of corresponding data.
- [] Chart updates immediately upon legend interaction.
- [] UI remains consistent across all supported chart types.
- [] Toggled state persists across date range and filter changes.

Labels

ui, charting, enhancement, accessibility

Estimate

M

markdown

Title

Redesign Empty State for Dashboard Charts

Description

Improve the user experience by designing and implementing an informative and visually appealing empty state for each dashboard chart when no data is available. This helps users understand why data might be missing and what actions to take.

Tasks

- Define empty state designs for each chart type (line, bar, pie, etc.).
- Display appropriate messages based on selected date range or filters.
- Use icons or illustrations to enhance visual feedback.
- Ensure consistent styling with the AshTrail theme.

Acceptance Criteria

- [] Each chart displays a friendly empty state when there is no data.
- [] Messages dynamically adjust based on active filters or range.
- [] Empty states are responsive and accessible.
- [] No errors occur when rendering charts with no data.

Labels

ui, charts, empty-state, enhancement

Estimate

S

markdown

Title

Add Toggle for Daily vs Cumulative View in Usage Charts

Description

Enhance the dashboard by adding a toggle switch that allows users to switch between viewing daily usage and cumulative usage in the inhalation charts. This provides flexibility for users to analyze trends over time and better understand their habits.

****Tasks****

- Add UI control (toggle or dropdown) to switch between "Daily" and "Cumulative" modes.
- Adjust chart data processing based on selected mode.
- Persist user selection in local state or storage for consistent experience.
- Update tooltips and axis labels accordingly.

Acceptance Criteria

- [] Toggle appears above or near the usage charts.
- [] Daily mode shows usage per day; Cumulative mode shows a running total.
- [] Toggle selection updates the chart data dynamically.
- [] Mode persists when user revisits the dashboard.

Labels

charts, enhancement, user-experience, data-visualization

Estimate

M

markdown

Title

Integrate Rolling Average Option for Inhalation Metrics

Description

Add a rolling average (e.g., 7-day) option to the inhalation metrics charts. This helps users visualize trends more clearly by smoothing short-term fluctuations.

****Tasks****

- Add a toggle or dropdown to switch between raw data and rolling average.
- Implement rolling average logic (default to 7-day, configurable if feasible).
- Ensure tooltip and data point accuracy matches the selected view.
- Update axis labels or legends to reflect rolling average mode.

Acceptance Criteria

- [] Users can toggle between raw data and rolling average view.
- [] Rolling average is correctly calculated and displayed.
- [] UI updates responsively when switching modes.
- [] Works across all applicable charts (length, frequency, etc.).

Labels

charts, enhancement, data-visualization, user-experience

Estimate

M

markdown

Title

Implement Daily Summary Card with Totals

Description

Create a visual daily summary card at the top of the main log view. This card should display the total number of inhalations, cumulative duration, and average length for the selected day or range.

Tasks

- Design and style the summary card to match the app's aesthetic.
- Calculate daily total inhalations, cumulative inhalation time, and average length.

- Add logic to update the summary dynamically based on date range selection.
- Ensure proper handling of edge cases (e.g., no data for selected day).

Acceptance Criteria

- [] Summary card shows total inhalations, total time, and average duration for selected date.
- [] Dynamically updates with the range/date picker.
- [] Responsive design across device types and screen sizes.
- [] Displays fallback message when no data is available.

Labels

UI, enhancement, data-summary, user-experience

Estimate

M

markdown

Title

Integrate Rolling Average Graph Option in Range Picker

Description

Enhance the analytics view by adding an option in the date range picker to toggle a rolling average view. This feature will allow users to see smoother trends in their usage habits over time.

Tasks

- Add toggle control for "Rolling Average" in the range selection UI.
- Compute rolling average for inhalation length over the selected time range.
- Update chart logic to display rolling average line when enabled.
- Maintain standard chart functionality when the toggle is off.

Acceptance Criteria

- [] "Rolling Average" toggle appears in range selector.
- [] Chart updates to display rolling average line when enabled.
- [] Line is distinguishable from the standard data points.
- [] Toggle state is remembered across view reloads within session.

Labels

UI, enhancement, data-visualization, analytics

Estimate

M

markdown

Title

Display Active Daily Streak Based on Logged Days

Description

Introduce a visible "Daily Streak" tracker on the dashboard to encourage consistent user engagement. The streak should represent the number of consecutive days the user has logged at least one session.

****Tasks****

- Query log records by day and count streak of consecutive active days.
- Display current streak count in a dedicated UI component (e.g., banner or badge).
- Reset streak when a day is missed.
- Consider using local timezone to calculate active days.

Acceptance Criteria

- [] Streak count updates correctly based on user activity.
- [] Visual display is engaging and clearly communicates streak value.
- [] Streak resets appropriately when a day is missed.
- [] Feature is mobile-friendly and fits cleanly into

dashboard layout.

Labels

feature, gamification, user-engagement, UI

Estimate

M

markdown

Title

Add Rolling Usage Averages to Time Series Chart

Description

Enhance the time series chart with a feature that displays rolling averages (e.g., 3-day, 7-day) of smoke session durations or inhalation lengths. This will allow users to visualize their usage trends over time more effectively.

Tasks

- Implement logic to compute rolling averages based on selected time window.
- Update chart configuration to optionally display these averages.
- Add toggle buttons or a dropdown to select different averaging windows.
- Ensure responsiveness and legibility across all devices.

Acceptance Criteria

- [] Users can toggle between 3-day, 7-day, and 14-day averages.
- [] Chart updates immediately upon selection of average window.
- [] Rolling averages are clearly labeled and do not clutter the chart.
- [] Feature works across all supported platforms and screen sizes.

Labels

enhancement, analytics, chart, UI

Estimate

L

markdown

Title

Create Static User Mode for Development Builds

Description

Introduce a static user mode in development to allow for easy testing without requiring Firebase authentication. This mode will simulate a logged-in user and operate entirely on a test Firestore collection.

Tasks

- Implement an environment-based toggle to activate static user mode.
- Bypass Firebase Auth when static mode is enabled.
- Route data to a separate `test_users` or similar collection.
- Clearly indicate in the UI when the app is running in static mode.

Acceptance Criteria

- [] Developer can enable static mode via an environment flag.
- [] The app functions fully in static mode with mock user credentials.
- [] Data is written and read from a non-production path.
- [] UI provides visual confirmation that static mode is active.

Labels

dev-only, enhancement, firebase, mock-mode

Estimate

M

markdown

Title

Integrate Daily Summary Chart with Toggleable Range Selector

Description

Develop a reusable chart component for daily summaries that supports dynamic range selection (e.g., last 7 days, 30 days, custom). This will enhance analysis capabilities in both static and authenticated modes.

****Tasks****

- Create a toggleable range selector UI component.
- Integrate the selector with the existing daily summary chart.
- Ensure responsiveness and visual clarity on mobile and desktop.
- Allow custom date ranges with calendar input (if feasible).

Acceptance Criteria

- [] Chart updates based on selected range.
- [] UI is intuitive and styled consistently with app theme.
- [] Works in both static user mode and Firebase-authenticated mode.

Labels

charting, enhancement, ui, analytics

Estimate

M

markdown

Title

Implement Tabbed View for Chart and Table Mode

Description

Create a tabbed interface that allows users to toggle between

the cumulative usage chart and a detailed data table. This will enhance data readability and user experience by giving users flexible ways to view their smoking log data.

Tasks

- Design and implement a tabbed UI with two tabs: "Chart" and "Table".
- Load the chart view and table view conditionally based on selected tab.
- Ensure mobile responsiveness and consistent styling.
- Default to showing the chart on load.

Acceptance Criteria

- [] Tabs are clearly visible and toggle correctly.
- [] Chart view displays the current summary or daily chart.
- [] Table view displays the sorted data table with inhalation records.
- [] Navigation between tabs does not reload the entire page.

Labels

ui, enhancement, charting, data-table

Estimate

S

markdown

Title

Add Date Range Highlight on Chart

Description

Enhance the cumulative usage chart by visually highlighting the currently selected date range. This will provide better visual feedback for users filtering data and help contextualize trends over specific periods.

Tasks

- Update the chart component to receive date range props.
- Add shaded or outlined region on the chart that matches the

selected range.

- Ensure the highlight updates dynamically when the date range changes.
- Verify that the visual does not obstruct key data points or readability.

Acceptance Criteria

- [] Date range selection is clearly highlighted on the chart.
- [] Highlight updates immediately when the range is changed.
- [] Works across all devices and screen sizes.
- [] No visual interference with data lines or tooltips.

Labels

enhancement, charting, ui, filters

Estimate

S

markdown

Title

Implement Rolling Usage Chart Toggle

Description

Introduce a toggle that allows users to switch between total cumulative usage and a rolling average view (e.g., 7-day average) on the chart. This will help identify short-term trends and fluctuations in usage habits more effectively.

Tasks

- Add a toggle UI control labeled "Show Rolling Average".
- Compute rolling average data based on the selected date range.
- Update the chart dynamically based on toggle state.
- Ensure tooltips and axis labels reflect the mode appropriately.

Acceptance Criteria

- [] Toggle successfully switches between cumulative and rolling average views.
- [] Rolling average is calculated correctly and efficiently.
- [] UI clearly indicates which mode is active.
- [] Chart remains responsive and legible in both modes.

Labels

enhancement, charting, data-visualization, ui

Estimate

M

markdown

Title

Enable Inline Data Editing in Entry Table

Description

Allow users to directly edit entries in the data table below the chart without needing to open a separate form. Inline editing streamlines small adjustments and enhances user efficiency.

Tasks

- Implement inline editing for `timestamp` and `length` fields in each row.
- Add confirmation or auto-save functionality on blur or Enter.
- Validate user input and revert on invalid entries.
- Style editable cells to indicate interactivity.

Acceptance Criteria

- [] Users can edit data inline and see immediate visual updates.
- [] Edited data is validated and saved to Firestore.
- [] Changes reflect in the chart view upon update.
- [] UX is clear and intuitive, with proper feedback on save or error.

Labels

enhancement, ui, firestore, data-management

Estimate

L

markdown

Title

Add Total Usage Summary and Session Stats

Description

Implement a summary widget above or below the chart that shows total inhalation duration, average session length, and total number of sessions within the selected date range. This helps users gain immediate insights into their usage patterns.

****Tasks****

- Calculate and display:
 - Total usage duration (sum of lengths)
 - Average session duration
 - Number of sessions
- Update dynamically when date range changes.
- Style summary to be clean, unobtrusive, and mobile-friendly.

Acceptance Criteria

- [] Usage summary updates correctly with chart filters.
- [] Stats are formatted clearly and accurately.
- [] Component layout does not disrupt current UI design.

Labels

feature, stats, ui, enhancement

Estimate

M

markdown

Title

Implement Data Export Feature (CSV/JSON)

Description

Allow users to export their filtered inhalation session data as a CSV or JSON file. This feature supports further analysis or personal archiving.

****Tasks****

- Add "Export Data" button near charts or filters
- Support both CSV and JSON formats
- Use current chart filters (date range, type) to determine exported content
- Trigger file download with appropriate filename and extension

Acceptance Criteria

- [] Exported file matches the filtered data shown in the chart
- [] Button is styled consistently with UI and is accessible
- [] File downloads correctly on all major browsers

Labels

feature, export, ui, enhancement

Estimate

M

markdown

Title

Add Animated Chart Transitions for Enhanced UX

Description

Improve the visual experience by adding smooth transitions and animations to the charts when data updates, filters change, or range adjustments are made.

****Tasks****

- Implement animations using the charting library's native transition features
- Ensure animations apply to changes in range, date filters, or data type
- Minimize performance impact on lower-end devices

Acceptance Criteria

- [] All charts animate gracefully when updated
- [] No performance degradation noticed during transitions
- [] Animation duration and easing feel natural and unobtrusive

Labels

enhancement, ux, charts, animation

Estimate

M

markdown

Title

Implement "Inhale Intensity" Color Mapping on Usage Chart

Description

Visualize inhalation intensity by mapping inhale length to color on the usage chart. This will help users better interpret their smoking patterns at a glance.

****Tasks****

- Define a color gradient (e.g., green to red) based on inhale length thresholds
- Modify the existing usage chart to reflect this gradient
- Ensure accessibility for users with color vision deficiencies (e.g., use tooltips or symbols)

Acceptance Criteria

- [] Chart segments reflect inhale length via color
- [] Color scale is consistent and clearly defined

- [] Visual clarity is maintained across different devices and themes

Labels

visualization, enhancement, chart, accessibility

Estimate

M

markdown

Title

Add Cumulative Usage Graph with Date Range Selection

Description

Create a new chart to show cumulative usage over time, with user-selectable date ranges to view daily, weekly, monthly, or custom spans. This feature helps users track smoking reduction or escalation over time.

Tasks

- Implement cumulative line chart or area chart visualization
- Integrate date range selector UI component
- Ensure chart updates dynamically when date range is changed
- Sync selected range with other visual components for coherence

Acceptance Criteria

- [] Cumulative chart is visible and responsive
- [] Users can select custom date ranges
- [] Data updates correctly based on selected range
- [] Works seamlessly across screen sizes

Labels

feature, chart, visualization, UX

Estimate

M

markdown

Title

Integrate Weekly Rolling Average Chart

Description

Add a rolling average line chart to highlight weekly smoking trends. This will help users understand short-term patterns and better interpret fluctuations in usage.

Tasks

- Calculate 7-day rolling average from inhalation logs
- Overlay rolling average on top of daily usage bar chart
- Style chart to distinguish rolling average line from raw data
- Include tooltip explaining the purpose of the rolling average

Acceptance Criteria

- [] 7-day rolling average is accurately calculated and displayed
- [] Line is visually distinct from daily values
- [] Tooltip appears on hover with contextual info
- [] Responsive and synchronized with selected date range

Labels

feature, data analysis, chart, visualization

Estimate

M

markdown

Title

Add Toggle for Cumulative vs. Daily View in Usage Charts

Description

Implement a toggle feature allowing users to switch between

daily usage and cumulative usage views on the inhalation chart. This will help visualize progress over time and identify long-term trends.

Tasks

- Add toggle button to chart view component
- Adjust chart logic to support cumulative sum of inhalation lengths
- Retain existing tooltip functionality for both views
- Ensure toggle is responsive and maintains selection across sessions

Acceptance Criteria

- [] Toggle switches chart between daily and cumulative usage
- [] Both views are visually clear and accurate
- [] Toggle state persists on page reload
- [] Styling matches overall design theme

Labels

feature, UX, data visualization, enhancement

Estimate

M

markdown

Title

Add Export Functionality for Usage Data

Description

Implement the ability for users to export their usage data from the app in CSV or JSON format. This will allow users to back up their logs or analyze them externally.

Tasks

- Add export button to the UI
- Offer CSV and JSON formats
- Include timestamp, inhalation length, and any related metadata

- Ensure correct formatting and escaping
- Display confirmation/toast on successful export

Acceptance Criteria

- [] User can export data in CSV and JSON formats
- [] Exported file contains accurate and complete log data
- [] UI feedback confirms success
- [] No sensitive data is included unintentionally

Labels

feature, data, export, enhancement

Estimate

M

markdown

Title

Implement Data Insights Dashboard (Beta)

Description

Develop a beta version of a dashboard providing basic insights from the user's log data. This feature will help users visualize and better understand their smoking habits.

Tasks

- Create a new route/page for the dashboard
- Design and implement charts (e.g., daily usage, cumulative time)
- Summarize key stats: total sessions, average length, peak hours
- Use mock data if live data integration isn't ready yet
- Ensure layout is responsive and mobile-friendly

Acceptance Criteria

- [] Dashboard loads correctly from navigation
- [] Displays at least one chart and summary statistic
- [] Handles empty/no data cases gracefully
- [] Clearly marked as Beta in UI

Labels
feature, UI, analytics, dashboard

Estimate
L

markdown

Title
Improve Entry Detail View with Enhanced Breakdown and Navigation

Description
Redesign the log entry detail view to provide users with a clearer and more engaging breakdown of their data. Include visual elements and enable seamless navigation between entries.

****Tasks****

- Implement a timeline or segmented bar to visualize inhalation phases
- Add next/previous buttons to quickly move between entries
- Display contextual stats (e.g., session length, time of day category)
- Match redesign with current app theme and aesthetic
- Ensure accessibility and mobile responsiveness

Acceptance Criteria

- [] Detail view includes visual breakdown (e.g., timeline, graph)
- [] Users can navigate to next/previous log without returning to list
- [] Stats are updated and visually consistent with the dashboard
- [] Design meets accessibility standards

Labels
enhancement, UI, user-experience, detail-view

Estimate

M

markdown

Title

Add Entry Tagging Feature for Categorization

Description

Introduce a tagging system that allows users to label their log entries with custom or suggested tags. This will help users categorize and analyze their habits more effectively over time.

****Tasks****

- Create UI component for adding/removing tags on each log entry
- Support auto-suggestions based on past tags and context (e.g., "stress", "social", "alone")
- Enable filtering and searching logs by tag on the dashboard
- Store tags efficiently in Firestore structure
- Reflect tags in charts and analytics if relevant

Acceptance Criteria

- [] Users can add/remove tags to log entries
- [] Suggested tags appear as user types
- [] Tags are persisted and can be filtered in the dashboard
- [] Tags are included in export and analytics if applicable

Labels

feature, UI, categorization, dashboard

Estimate

L

markdown

Title

Implement Rolling Averages in Usage Charts

Description

Enhance data analysis by implementing rolling average lines (e.g., 3-day, 7-day, 30-day) on the usage charts. This will help users better identify trends in their smoking habits over time.

Tasks

- Add rolling average computation to the data transformation logic
- Allow users to toggle different rolling average options (3d, 7d, 30d) in the UI
- Update chart rendering to include rolling average lines
- Ensure averages are responsive to selected date range and filters

Acceptance Criteria

- [] Rolling average lines display correctly on all relevant charts
- [] Users can toggle visibility of each average duration
- [] Averages adjust to date range and log filtering
- [] Tooltips and legends clearly distinguish average lines

Labels

feature, analytics, chart, UX

Estimate

M

markdown

Title

Add Daily and Weekly Summary Stats Section

Description

Introduce a summary section that displays key usage statistics

for the selected day or week. This will help users quickly assess their progress and patterns.

Tasks

- Calculate total puffs, average duration, and cumulative usage for the selected date or week
- Display summary in a dedicated card or section above or below charts
- Ensure the summary updates based on selected date range
- Use simple, readable metrics with appropriate formatting

Acceptance Criteria

- [] Summary includes total puffs, average duration, and total usage
- [] Stats update based on user-selected date range
- [] Display is responsive and integrates with existing UI

Labels

feature, UI, analytics, usability

Estimate

M

markdown

Title

Implement Rolling Usage Chart Option

Description

Add an option to view rolling usage averages (e.g., 3-day, 7-day) on the usage chart to help users visualize trends over time more clearly.

Tasks

- Add dropdown selector to choose rolling window size (e.g., none, 3-day, 7-day)
- Calculate rolling averages for total usage and puffs
- Update chart to reflect rolling data when selected
- Maintain performance and responsiveness with large datasets

Acceptance Criteria

- [] Rolling window selector is present and functional
- [] Chart displays correct rolling averages when selected
- [] User can toggle back to standard view
- [] Rolling option integrates cleanly with other filters and date selectors

Labels

feature, charting, enhancement, usability

Estimate

M

markdown

Title

Add Cumulative Usage Toggle for Line Chart

Description

Introduce a toggle that allows users to switch between daily usage values and cumulative totals over time. This can help highlight long-term patterns and personal milestones.

****Tasks****

- Add a toggle UI element (e.g., switch or checkbox)
- Update chart logic to compute and display cumulative totals when enabled
- Ensure toggle works with existing filters and date ranges
- Maintain clear labeling to distinguish between daily and cumulative views

Acceptance Criteria

- [] Toggle is visible and labeled clearly
- [] Switching toggle updates chart to cumulative view and vice versa
- [] Chart data remains accurate across all filtered ranges
- [] Cumulative view enhances trend clarity without performance issues

Labels

feature, charting, enhancement, analytics

Estimate

M

markdown

Title

Implement Rolling Average Option for Daily Usage Chart

Description

Add functionality to view a rolling average (e.g., 3-day or 7-day) for daily inhalation usage to help users smooth out fluctuations and better understand trends over time.

****Tasks****

- Add dropdown or radio buttons to select rolling average window (none, 3-day, 7-day)
- Update chart logic to calculate and overlay rolling averages
- Ensure averages reflect current filters and date range
- Style the average line differently (e.g., dashed or lighter color) for clarity

Acceptance Criteria

- [] Users can select between no average, 3-day, and 7-day rolling averages
- [] Chart updates correctly and displays average line
- [] Average line adjusts to filters and time range
- [] Visual distinction between raw data and average line

Labels

feature, charting, analytics, enhancement

Estimate

M

markdown

Title

Add Toggle for Switching Between Cumulative and Daily Views

Description

Introduce a toggle or dropdown option to allow users to switch between cumulative inhalation data and daily usage breakdowns. This provides flexibility in how data is interpreted for different user goals.

Tasks

- Add UI toggle to switch chart modes (Daily / Cumulative)
- Refactor chart component to support both display types
- Ensure the toggle works seamlessly with date range and filtering options
- Maintain visual consistency and responsiveness

Acceptance Criteria

- [] Users can switch between daily and cumulative usage views
- [] Chart updates dynamically to reflect selection
- [] Both modes support date filtering and range controls
- [] Chart labels and legends adjust accordingly

Labels

feature, charting, usability, enhancement

Estimate

M

markdown

Title

Implement Session Comparison Feature for Insights

Description

Add functionality to allow users to compare two different smoking sessions side-by-side. This will help users track

progress and understand behavioral patterns over time.

Tasks

- Design UI layout for dual session comparison
- Enable selection of two sessions from the log or calendar view
- Display both sessions' stats (duration, inhalations, frequency) side-by-side
- Ensure charts and summaries adapt for dual display
- Add labels and visual cues for clarity

Acceptance Criteria

- [] Users can select two sessions for comparison
- [] A side-by-side view displays comparative data clearly
- [] Visualizations support and clearly reflect both sessions
- [] No impact on performance or mobile layout

Labels

feature, analytics, comparison, enhancement

Estimate

L

markdown

Title

Design and Implement User Achievement Badges

Description

Introduce a badge system that rewards users for reaching smoking milestones and building healthier habits. This gamification feature will increase engagement and provide positive reinforcement.

Tasks

- Define milestone criteria (e.g., 7 smoke-free days, consistent logging for 1 month)
- Design badge graphics and names
- Implement logic to track progress and award badges

- Display badges on user profile and/or dashboard
- Create badge notifications and sharing options

Acceptance Criteria

- [] Users receive badges based on defined behavior/milestones
 - [] Badge graphics are clearly visible and visually appealing
 - [] Badge system is responsive and does not affect app performance
 - [] Badges can be shared or viewed in a public/private profile section

Labels

feature, gamification, engagement, enhancement

Estimate

M

markdown

Title

Add Rolling Average Chart for Weekly Inhalation Trends

Description

To better visualize smoking habits over time, introduce a rolling average chart that displays average inhalation length per day, using a 7-day rolling window. This will help users see gradual changes and assess trends without daily fluctuations.

Tasks

- Calculate 7-day rolling average of inhalation length per day
- Add new chart component to the analytics dashboard
- Allow toggling between raw data and rolling average
- Ensure responsiveness across devices

Acceptance Criteria

- [] Chart displays 7-day rolling average correctly

- [] Users can toggle between raw and average views
- [] Chart integrates seamlessly with current UI
- [] No performance issues when loading data

Labels

enhancement, data-visualization, analytics

Estimate

M

markdown

Title

Implement Daily Summary Stats Panel on Dashboard

Description

Introduce a new panel on the AshTrail dashboard displaying summary statistics for the selected day. This includes total inhalations, average duration, longest session, and time of first/last usage.

Tasks

- Create a new UI component for daily summary
- Aggregate data based on selected date
- Display total inhalations, average inhalation duration, longest session length, and first/last inhalation timestamps
- Style component to match the new redesign aesthetic

Acceptance Criteria

- [] Summary panel updates with date selection
- [] Displays all four metrics accurately
- [] UI is clean, responsive, and integrated with current design
- [] No data discrepancies or performance issues

Labels

feature, ui, dashboard

Estimate

M

markdown

Title

Add Cumulative Usage Graph with Rolling Average Option

Description

Implement a line chart that visualizes cumulative inhalation duration over time for a selected date range. Add an optional toggle to display a rolling average (e.g., 3-day or 7-day) for better trend visibility.

****Tasks****

- Aggregate cumulative inhalation duration per day
- Add chart to dashboard with responsive design
- Implement toggle to switch between raw cumulative and rolling average view
- Ensure chart dynamically updates with range selector

Acceptance Criteria

- [] Chart displays accurate cumulative totals
- [] Toggle for rolling average functions correctly
- [] Updates in sync with date range filters
- [] Styling matches app's new visual design

Labels

feature, chart, dashboard, enhancement

Estimate

M

markdown

Title

Implement Filter Toggle for Smoke Log Table (By Type & Source)

Description

Add UI toggles and logic to filter the Smoke Log table by "Type" (e.g., Inhalation, Break, etc.) and "Source" (e.g., Manual, Auto-detected). Filters should be compatible with date range selection and update the data shown in both the table and the charts.

****Tasks****

- Add dropdowns or toggle buttons to filter by Type and Source
- Integrate with existing data processing pipeline
- Ensure filtered results are reflected in both table and visualizations
- Retain filter state during app navigation

Acceptance Criteria

- [] Filter controls are visible and easy to use
- [] Filters apply correctly and update results
- [] Chart and table remain synchronized with filters
- [] No performance issues introduced

Labels

feature, filters, table, dashboard

Estimate

M

markdown

Title

Implement Editable Notes Field for Each Log Entry

Description

Allow users to add or edit a personal note for each smoke log entry. Notes should be stored with the log entry in Firestore and displayed in the table. Edits should auto-save with minimal delay.

****Tasks****

- Add editable text field to each row of the Smoke Log table

- Store notes in Firestore under the corresponding document
- Update table UI to accommodate note previews and editing
- Add icon or visual cue for entries with notes

Acceptance Criteria

- [] Notes can be added/edited for each entry
- [] Changes auto-save without page reload
- [] Table displays note presence and full content on expand or hover
- [] Notes are synced correctly across sessions

Labels

enhancement, table, firestore, ui/ux

Estimate

M

markdown

Title

Add Button to Export Log Data to CSV

Description

Users should be able to export their smoke log data as a CSV file for personal analysis or backup. This feature should include filters (date range, time of day, etc.) and download the file client-side.

Tasks

- Create export button in the UI
- Implement logic to convert filtered Firestore data to CSV format
- Trigger download of the CSV file with a meaningful filename
- Optionally include columns for timestamp, length, notes, and tags

Acceptance Criteria

- [] Users can download a CSV of visible/filtered log entries
- [] CSV includes correct headers and data

- [] Filename includes date/time of export
- [] Feature works in major modern browsers

Labels

feature, export, firestore, ui

Estimate

M

markdown

Title

Implement Daily and Weekly Summary Stats

Description

Introduce a summary section that shows key metrics based on user activity, such as:

- Total logs for the day
- Average length of sessions
- Total time spent smoking
- Change from previous day/week

This feature helps users gain quick insights into their smoking patterns and progress.

Tasks

- Aggregate and calculate daily/weekly statistics from Firestore
- Design a concise summary component for the dashboard
- Highlight changes from previous periods with simple indicators (e.g., arrows or colors)

Acceptance Criteria

- [] Summary data reflects filters (if applied)
- [] Metrics are accurate and update in real-time or near real-time
- [] UI is clean and integrates seamlessly with existing layout
- [] Users can toggle between daily and weekly views

Labels
feature, analytics, dashboard, firestore

Estimate
L

markdown

Title
Add Rolling Usage Chart with Custom Range Selector

Description
Implement a rolling usage chart that allows users to visualize their inhalation data over time with customizable date ranges and rolling averages.

****Features****

- Select date range (e.g., last 7, 14, 30 days, or custom range)
- Chart displays rolling average usage (e.g., 3-day, 7-day)
- Tooltip to show precise data on hover
- Option to compare rolling vs. raw data

Tasks

- Integrate range and rolling window selector into chart filters
- Query and process Firestore data based on selected range
- Calculate rolling averages in chart logic
- Design responsive chart with toggles for display options

Acceptance Criteria

- [] User can change date range and rolling window
- [] Chart updates correctly with new selections
- [] Data is accurate and aligns with backend records
- [] UI remains clean and intuitive across devices

Labels
feature, visualization, data-analysis, charts

Estimate

L

markdown

Title

Optimize Firestore Queries for Log Filtering and Sorting

Description

Move filtering and sorting logic for inhalation log data directly to the Firestore query layer to improve app performance and streamline data processing for visualizations and tables.

****Goals****

- Reduce in-app data processing for timestamps and length
- Ensure charts and tables reflect accurate, filtered results
- Simplify integration with range and type filters

Tasks

- [] Refactor Firestore query to accept date range and type filters
- [] Update `submitData()` and related functions to use filtered data
- [] Ensure timestamp descending sort order is preserved in results
- [] Test across different filter scenarios

Acceptance Criteria

- [] Logs are correctly filtered and sorted server-side
- [] App reflects accurate data in charts and tables
- [] Performance improves with large data sets

Labels

performance, firestore, backend, enhancement

Estimate

M

markdown

Title

Implement Chart Data Export (CSV/PDF)

Description

Allow users to export their inhalation log charts and data tables as CSV or PDF files for personal tracking, sharing with healthcare providers, or archiving.

****Goals****

- Provide export options (CSV, PDF) for both table and chart data
- Ensure filters and date ranges are respected in exported data
- Maintain chart styles when exporting to PDF

Tasks

- [] Add "Export" button to charts and tables UI
- [] Implement CSV generation from current filtered data
- [] Implement PDF export with chart rendering
- [] Validate correct data export based on active filters

Acceptance Criteria

- [] Users can export data in both CSV and PDF formats
- [] Exports match filtered view (date range, type, etc.)
- [] PDFs include readable charts and metadata

Labels

feature, export, charting, enhancement

Estimate

M

markdown

Title

Add Editable Notes Field for Individual Inhalation Entries

Description

Enhance user control by allowing them to attach and edit personal notes on each inhalation entry. This feature will support reflective logging and qualitative tracking.

Goals

- Users can add optional notes when creating or editing an inhalation entry
- Display notes in the data table and entry detail view
- Allow editing or deleting notes post-entry

Tasks

- [] Update Firestore schema to support an optional `notes` field
- [] Modify data submission form to include a notes input
- [] Update chart/table rendering to optionally show note icons or previews
- [] Implement editing capabilities for existing notes

Acceptance Criteria

- [] Notes are saved and retrievable for each entry
- [] Notes are visible in UI where appropriate
- [] Users can add, edit, and delete notes

Labels

feature, enhancement, data-entry, user-experience

Estimate

M

markdown

Title

Implement Interactive Rolling Average Overlay on Inhalation Chart

Description

To give users better insight into patterns, implement an interactive rolling average overlay on the inhalation chart. This helps users identify trends over days/weeks.

****Goals****

- Allow users to toggle a rolling average overlay on the chart
- Let users adjust the window (e.g., 3-day, 7-day)
- Visually distinguish the rolling average from raw data

Tasks

- [] Add toggle control to enable/disable rolling average overlay
- [] Implement logic to compute rolling averages on selected data
- [] Update chart rendering logic to include overlay line
- [] Style overlay for clarity and accessibility

Acceptance Criteria

- [] Users can enable/disable the overlay
- [] Window size is adjustable
- [] Overlay appears accurately and updates with filters

Labels

feature, data-visualization, user-insight

Estimate

M

markdown

Title

Add Downloadable CSV Export for Inhalation Logs

Description

Provide users the ability to download their inhalation logs as a CSV file for personal tracking or sharing with health professionals. This feature supports data ownership and

portability.

****Goals****

- Add a button to export currently visible logs to CSV
- Respect applied filters (e.g., date range)
- Ensure correct formatting of timestamps and inhalation lengths

Tasks

- [] Implement CSV generation logic on filtered data
- [] Create UI button for CSV download
- [] Ensure compatibility across browsers
- [] Test file encoding and structure

Acceptance Criteria

- [] Users can download a CSV containing all filtered logs
- [] Data formatting is clear and accurate
- [] CSV opens cleanly in spreadsheet apps

Labels

feature, data-export, user-data

Estimate

M

markdown

Title

Implement Chart Download as Image (PNG/JPEG)

Description

Allow users to download visual inhalation charts as image files for sharing or recordkeeping. This feature enhances data visualization portability and reporting.

****Goals****

- Add a download button to each chart
- Enable PNG and JPEG formats
- Maintain visual fidelity across downloads

Tasks

- [] Integrate chart-to-image export functionality
- [] Provide format options (PNG, JPEG)
- [] Style the download button to match UI
- [] Test on mobile and desktop

Acceptance Criteria

- [] Users can download charts as image files
- [] Images maintain chart styles and data labels
- [] Button is intuitive and responsive

Labels

feature, charts, data-export, UI

Estimate

M

markdown

Title

Enable Session-based Theme Persistence (Light/Dark Mode)

Description

Implement a mechanism to remember the user's theme preference (light or dark mode) across sessions for improved user experience and accessibility.

****Goals****

- Store user preference in local storage or session
- Automatically apply preferred theme on app load
- Ensure manual toggle works consistently

Tasks

- [] Detect and store theme preference on toggle
- [] Apply stored theme on app initialization
- [] Provide fallback to system default if no preference is stored
- [] Test for edge cases across browsers and platforms

Acceptance Criteria

- [] User theme preference is saved and applied automatically
- [] Theme toggle updates preference in real-time
- [] System default used only when no prior preference exists

Labels

enhancement, UI, theme, accessibility

Estimate

S

markdown

Title

Implement Rolling Average Usage Chart

Description

Introduce a new chart that shows rolling average inhalation usage over a customizable time window (e.g., 3-day, 7-day). This helps visualize usage trends more clearly and smooths out day-to-day fluctuations.

****Goals****

- Add toggle or dropdown to select rolling window size
- Calculate and display rolling average inhalation lengths
- Integrate with existing chart UI

Tasks

- [] Add rolling window selector to the UI
- [] Implement rolling average logic in data processing
- [] Update chart to reflect selected rolling average
- [] Ensure compatibility with range filtering

Acceptance Criteria

- [] Rolling average is accurate and updates with user input
- [] UI is intuitive and responsive
- [] Works seamlessly with other filtering options

Labels

enhancement, chart, data-analysis, UI

Estimate

M

markdown

Title

Optimize Firestore Reads with Batched Data Fetching

Description

To improve performance and reduce read costs, update the Firestore query logic to fetch data in optimized batches. This is particularly important for longer date ranges or high-frequency log entries.

****Goals****

- Reduce the number of Firestore reads
- Implement batching logic for large query sets
- Maintain compatibility with range filters and chart display

Tasks

- [] Refactor `getDataInRange()` to support batched queries
- [] Handle pagination or composite query logic if needed
- [] Test for performance and data accuracy
- [] Ensure fallback behavior for small data sets

Acceptance Criteria

- [] Queries use fewer reads for large datasets
- [] All data is retrieved accurately across ranges
- [] UI remains responsive

Labels

performance, backend, firestore, enhancement

Estimate

M

markdown

Title

Implement Local Caching for Offline Access and Faster Load Times

Description

To support offline access and improve performance, implement local caching using Isar or another suitable local storage solution. Cached data should sync with Firestore when online and be used as fallback when offline or on slow connections.

Goals

- Enable offline access to recent logs
- Improve load speed for charts and tables
- Sync with Firestore on reconnect

Tasks

- [] Define schema for local data storage
- [] Implement caching layer in data fetching logic
- [] Add sync mechanism to reconcile local and remote changes
- [] Ensure UI updates properly when offline or reconnecting

Acceptance Criteria

- [] Recent data is available offline
- [] Cached data syncs with Firestore correctly
- [] UI reflects online/offline states gracefully

Labels

offline, performance, caching, flutter, enhancement

Estimate

L

markdown

Title

Refactor Range Selector and Filter Logic into a Central

Controller

Description

Consolidate the date range selector and filtering logic into a single controller to simplify state management and allow seamless integration across all charts and tables. This will streamline the implementation of additional analytics like rolling usage.

Goals

- Centralize range and filter logic
- Improve maintainability and scalability
- Enable consistent filtering across all visualizations

Tasks

- [] Create a controller to manage selected date range and filters
- [] Refactor existing widgets to consume this controller
- [] Update chart and table widgets to respond to changes in controller
- [] Add unit tests for controller behavior

Acceptance Criteria

- [] All charts and tables reflect changes in selected range and filters
- [] New analysis types (e.g., rolling usage) can be plugged in easily
- [] Code is modular and easier to maintain

Labels

refactor, enhancement, state-management, flutter

Estimate

M

markdown

Title

Implement Rolling Usage Chart for Cumulative Consumption

Analysis

Description

Develop a rolling usage chart that visualizes cumulative consumption over time using a moving window average. This will help users identify usage patterns and better understand trends in their smoking behavior.

Goals

- Provide visual feedback on short- and long-term changes in consumption
- Allow user-defined window sizes (e.g., 1 day, 3 days, 7 days)
- Support toggle between raw and smoothed data views

Tasks

- [] Add UI control for selecting rolling window size
- [] Calculate moving average of inhalation length over time
- [] Display rolling usage chart alongside total usage chart
- [] Test performance and behavior with large datasets

Acceptance Criteria

- [] Users can switch between raw and rolling usage data
- [] Data responds to selected range and filters
- [] Chart updates without lag for a year of data

Labels

feature, chart, analytics, flutter

Estimate

L

markdown

Title

Implement User-Selectable Time Ranges and Filters for Chart Analysis

Description

Introduce user-selectable time ranges and filtering options to enable more tailored chart analysis within the AshTrail (Smoke-Log) redesign. This feature will allow users to narrow down insights to specific days, weeks, or custom periods and apply filters such as time of day or usage type.

Goals

- Improve data exploration capabilities
- Enable range and filter controls that work across all charts
- Ensure that filters dynamically update chart visuals

Tasks

- [] Add dropdown or slider for selecting predefined time ranges (e.g., Today, 7 Days, 30 Days, Custom)
- [] Implement filter options for time of day and custom tags
- [] Update chart rendering logic to respond to active filters
- [] Persist filter settings during session

Acceptance Criteria

- [] Filters and ranges update all relevant charts in real time
- [] Custom date range picker works as expected
- [] Performance remains stable with filtered large datasets

Labels

feature, filters, chart, UX, flutter

Estimate

L

markdown

Title

Create Daily Usage Summary Cards with Key Stats

Description

Design and implement daily summary cards that provide a snapshot of key smoking behavior metrics such as total

duration, number of sessions, average session length, and peak usage time. These cards will appear above the charts to give users quick, digestible insights at a glance.

Tasks

- [] Define the summary statistics to include
- [] Design layout and styling for the summary cards
- [] Fetch and calculate daily statistics from the filtered dataset
- [] Integrate the cards into the dashboard above the main chart area

Acceptance Criteria

- [] Daily summary cards display accurate statistics
- [] Cards dynamically update with applied filters and selected date range
- [] UI is clean, accessible, and responsive

Labels

feature, UI, statistics, flutter, redesign

Estimate

M

markdown

Title

Implement Rolling Average Line for Usage Chart

Description

Enhance the main usage chart by adding a rolling average line to provide a smoother view of trends over time. This will help users better understand their smoking patterns and reduce the noise of daily fluctuations.

Tasks

- [] Choose default window size for rolling average (e.g., 7 days)
- [] Implement logic to calculate the rolling average based

on filtered data

- [] Add the rolling average line to the chart with distinct styling
- [] Add toggle control to show/hide the rolling average

Acceptance Criteria

- [] Rolling average correctly updates with filters and selected date range
- [] Toggle allows users to enable or disable the rolling average view
- [] Visual style distinguishes the average line from raw data

Labels

feature, data-visualization, chart, flutter, redesign

Estimate

M

markdown

Title

Refactor Data Filtering Logic into Centralized Utility

Description

Refactor the data filtering and sorting logic into a centralized utility that can be shared between the range selector, table view, and chart components. This will make the logic more maintainable and allow new visualizations to leverage the same filtered data source.

Tasks

- [] Create a new utility or provider to handle data filtering based on date range, user-defined settings, and other criteria
- [] Replace existing filtering logic in the range selector, table, and chart with the new utility
- [] Ensure filtering updates correctly propagate to all dependent views

Acceptance Criteria

- [] All visual components (chart, table, etc.) reflect the same filtered data
- [] No duplication of filtering logic across components
- [] Easy to extend the filtering criteria in the future

Labels

refactor, data, flutter, redesign, architecture

Estimate

L

markdown

Title

Implement Cumulative Use Line Chart with Smoothing Option

Description

Add a new cumulative use line chart that plots total inhalation length over time. Include an optional smoothing toggle (e.g., 7-day rolling average) to help visualize long-term trends and reduce short-term fluctuations.

Tasks

- [] Add a new chart widget displaying cumulative inhalation length over time
- [] Add a toggle to enable a smoothed (rolling average) version of the chart
- [] Ensure the chart uses the shared filtered data utility
- [] Style the chart to match the app theme

Acceptance Criteria

- [] Chart correctly shows cumulative use over time
- [] Rolling average smoothing works as expected
- [] Filtering changes reflect on the chart

Labels

enhancement, visualization, chart, flutter, redesign

Estimate

M

markdown

Title

Build Daily Heatmap View for Inhalation Frequency

Description

Implement a heatmap view similar to GitHub contribution graphs, showing daily inhalation frequency. Each cell represents a day, colored by usage intensity (e.g., low, medium, high). This helps users identify patterns in their behavior.

Tasks

- [] Design a heatmap layout mapping usage by day
- [] Assign colors to usage intensity levels (e.g., 0, 1-5, 6-10, >10)
- [] Connect heatmap to shared filtered data set
- [] Add month/year navigation controls

Acceptance Criteria

- [] Heatmap correctly visualizes daily frequency
- [] Color intensity accurately reflects usage
- [] Navigation allows users to browse historical months

Labels

enhancement, visualization, flutter, UI, redesign

Estimate

L

markdown

Title

Implement Rolling Average Chart (7-Day, 14-Day)

Description

Add a new line chart view that shows the rolling average of inhalation frequency over time. Users can toggle between 7-day and 14-day rolling averages. This helps visualize behavioral trends and identify improvement or regression.

Tasks

- [] Calculate 7-day and 14-day rolling averages from filtered dataset
- [] Display data on a line chart with proper time labels
- [] Add toggle switch for selecting average window (7 or 14 days)
- [] Style chart to match existing visual theme

Acceptance Criteria

- [] Chart displays correct rolling average values
- [] Toggle accurately switches between 7-day and 14-day averages
- [] Chart is responsive and legible on mobile and desktop

Labels

feature, enhancement, chart, flutter, redesign, analysis

Estimate

M

markdown

Title

Add Cumulative Usage Chart with Adjustable Date Range

Description

Implement a chart that displays cumulative inhalation time over a user-specified date range. This feature will help users visualize total usage patterns and evaluate long-term trends in behavior.

Tasks

- [] Add cumulative sum logic to existing data filtering
- [] Build a line or area chart to represent cumulative usage
- [] Allow users to select a date range using the existing range selector
- [] Ensure cumulative data updates based on filtered results

Acceptance Criteria

- [] Cumulative chart displays correctly based on selected date range
- [] Chart dynamically updates with changes in filter settings
- [] Usage totals are correctly calculated and displayed

Labels

feature, enhancement, chart, flutter, redesign, analysis

Estimate

M

markdown

Title

Implement Visual Indicators for Peak Usage Times

Description

Add visual markers or highlights to indicate peak smoking times throughout the day. This enhancement will help users quickly identify when their usage is highest and recognize potential patterns.

Tasks

- [] Analyze timestamp data to identify hours with the highest usage
- [] Create visual markers (e.g., shaded regions or annotations) for peak periods
- [] Integrate peak indicators into existing time-based charts
- [] Ensure responsiveness across various time filters and view sizes

```
### Acceptance Criteria
- [ ] Peak times are accurately calculated based on filtered data
- [ ] Visual indicators are clear and unobtrusive
- [ ] Feature behaves consistently across devices and date ranges

### Labels
feature, UX, data-visualization, flutter, redesign

### Estimate
S
```

markdown

```
### Title
Add Weekly and Monthly Summary Views

### Description
Introduce a toggle feature to display smoking data summaries by week and by month, offering users a broader perspective on trends and progress over time.

### Tasks
- [ ] Implement toggle or tab component for selecting daily, weekly, or monthly views
- [ ] Aggregate data accordingly for each mode
- [ ] Update charts and statistics to reflect selected summary range
- [ ] Adjust tooltips and labels for appropriate time intervals

### Acceptance Criteria
- [ ] Users can easily switch between daily, weekly, and monthly summaries
- [ ] Aggregated data is accurate and visually intuitive
- [ ] UI adapts smoothly without breaking layout or performance
```

Labels
feature, data-aggregation, UI/UX, flutter, redesign

Estimate
M

markdown

Title
Implement Long-Term Trend Insights with Rolling Averages

Description
Add support for visualizing long-term trends using rolling averages (e.g., 7-day, 30-day) to smooth out day-to-day noise and reveal meaningful usage patterns over time.

Tasks
- [] Add rolling average calculations for 7-day and 30-day spans
- [] Integrate trendlines into the main usage chart
- [] Allow toggling visibility of rolling average lines
- [] Style trendlines distinctly from daily data for clarity

Acceptance Criteria
- [] Rolling averages appear smoothly overlaid on the main usage chart
- [] User can toggle trendlines on/off
- [] Trendlines adjust dynamically to selected date ranges and filters

Labels
feature, data-visualization, charting, flutter, redesign

Estimate
M

markdown

Title

Add Export Functionality for User Data and Charts

Description

Implement a feature that allows users to export their smoking log data and visual charts. Support formats such as CSV for raw data and PNG or PDF for chart images.

Tasks

- [] Add export button for log data as CSV
- [] Enable chart image export (PNG or PDF)
- [] Ensure exported data respects current filters and date range
- [] Provide confirmation and success message upon export

Acceptance Criteria

- [] Users can download CSV files of their logs
- [] Users can export a snapshot of the usage chart
- [] Exported content matches current filtered view

Labels

feature, export, user-data, charting, flutter, redesign

Estimate

M

markdown

Title

Implement Rolling Average and Trendline Analysis for Usage Chart

Description

Enhance the visual analytics of the usage chart by adding rolling averages (e.g., 3-day, 7-day) and an optional trendline. This will help users better understand long-term patterns in their inhalation behavior.

Tasks

- [] Add dropdown or toggle for selecting rolling average window (None, 3-day, 7-day, 14-day)
- [] Compute and overlay rolling average on the usage chart
- [] Add option to enable/disable trendline based on linear regression
- [] Ensure all filters and ranges are applied to the underlying data before trendline/average calculation

Acceptance Criteria

- [] Users can view rolling averages and toggle trendlines
- [] Calculations respond dynamically to date range filters
- [] Chart updates correctly with visual clarity and proper labeling

Labels

feature, charting, analytics, rolling-average, flutter, redesign

Estimate

L

markdown

Title

Create Export Functionality for Inhalation Data

Description

Add a feature to export inhalation logs as CSV or JSON, allowing users to back up their data or analyze it externally.

Tasks

- [] Add export button on the main dashboard or settings page
- [] Allow users to choose between CSV and JSON formats
- [] Export data should reflect currently applied filters (e.g., date range, duration)
- [] Format timestamps and inhalation values for readability

Acceptance Criteria

- [] Users can successfully download a filtered export in CSV or JSON
- [] Exported files open correctly and show expected values
- [] Feature works across web and mobile platforms

Labels

feature, export, data-management, flutter, redesign

Estimate

M

markdown

Title

Implement User Feedback Modal

Description

Add a modal that allows users to submit feedback about their experience with AshTrail. This will help identify bugs, gather feature requests, and improve usability.

Tasks

- [] Add "Give Feedback" button to app drawer or settings menu
- [] Design modal with fields for feedback type (bug, feature request, other), optional email, and feedback message
- [] Store feedback entries locally for now, with placeholder for future server submission
- [] Provide confirmation message after feedback is submitted

Acceptance Criteria

- [] Users can open and submit the feedback modal
- [] Feedback entries are stored with timestamp
- [] Modal closes after successful submission and confirms receipt

Labels

feature, ui/ux, feedback, redesign, flutter

Estimate

S

markdown

Title

Add Weekly Summary Chart with Trends

Description

Display a weekly summary chart showing usage trends over the past 7 days. This should help users understand changes in their behavior and spot patterns in inhalation frequency and duration.

Tasks

- [] Create a new chart widget titled "Weekly Summary"
- [] Display total inhalation time per day over the past 7 days
- [] Add optional line to indicate daily average
- [] Use existing data filtering and sorting logic
- [] Optimize for small screen sizes

Acceptance Criteria

- [] Weekly Summary chart appears below the main chart
- [] Data updates automatically as new logs are added
- [] Users can clearly identify trends at a glance

Labels

feature, charts, data-visualization, redesign, flutter

Estimate

M

markdown

Title

Implement Rolling Averages for Usage Trends

Description

Introduce rolling averages (e.g., 3-day, 7-day) to help users better visualize short-term and long-term usage patterns. This provides additional insight into changes in behavior over time.

Tasks

- [] Add a toggle control above the chart to enable rolling averages
- [] Calculate rolling averages on the filtered dataset
- [] Overlay line graph of the rolling average on the main usage chart
- [] Adjust chart legend and color scheme for clarity

Acceptance Criteria

- [] Users can toggle between raw data and rolling average views
- [] Rolling average line accurately reflects calculated values
- [] Rolling average updates dynamically with data range and filters

Labels

enhancement, charts, analytics, flutter, redesign

Estimate

M

markdown

Title

Add Daily and Weekly Summaries to Dashboard

Description

Enhance the dashboard with summary statistics to give users a quick overview of their smoking behavior. These summaries will include total duration, average session length, and peak usage time for the selected time range.

Tasks

- [] Create a summary section above the chart
- [] Display total inhalation time for the selected range
- [] Show average session duration
- [] Identify and display peak usage time (hour of day)
- [] Ensure summaries update based on applied filters

Acceptance Criteria

- [] Summary section renders correctly across screen sizes
- [] Data updates dynamically with range selection
- [] Values are accurate and align with displayed chart data

Labels

feature, dashboard, analytics, flutter, redesign

Estimate

M

markdown

Title

Implement User Feedback Modal with Optional Survey

Description

Add a floating action button (FAB) on the dashboard that opens a modal for users to submit feedback. The modal will optionally redirect to a short survey form hosted externally (e.g., Google Forms or Typeform).

Tasks

- [] Add FAB button to dashboard screen
- [] Create modal with text input and submit/cancel buttons
- [] Include optional link to external survey form
- [] Store feedback locally or send to Firebase
- [] Ensure responsive design and accessibility

Acceptance Criteria

- [] FAB is visible and non-intrusive on all screen sizes
- [] Modal submits feedback and optionally opens external

form

- [] Feedback can be reviewed in Firestore or exported
- [] UI matches the design system used throughout the app

Labels

feature, feedback, UI, flutter, redesign

Estimate

S

markdown

Title

Create Monthly Summary Chart for Smoke Sessions

Description

Design and implement a visual monthly summary chart that aggregates smoking session data by day. The chart should give users an overview of their habits across the selected month, helping to identify patterns or trends.

Tasks

- [] Define Firestore query to fetch all sessions within a selected month
- [] Aggregate data by date (total count and average length per day)
- [] Implement chart widget (e.g., bar chart or line chart) in the dashboard
- [] Allow navigation between months
- [] Style chart according to app theme

Acceptance Criteria

- [] Chart displays all session days for selected month
- [] Chart updates correctly when navigating to different months
- [] Each data point shows total sessions and average duration
- [] Matches AshTrail design style and works on mobile

Labels

feature, charting, analytics, flutter, redesign

Estimate

M

markdown

Title

Add Toggle to Switch Between Daily and Cumulative View in Smoke Chart

Description

To enhance user insight into smoking habits, implement a toggle on the main chart that allows switching between daily usage and cumulative usage over time. This supports both micro (per-day) and macro (habit-forming) analysis.

Tasks

- [] Add UI toggle to switch between "Daily" and "Cumulative"
- [] Adjust chart data calculation based on selected mode
- [] Update chart labels and titles dynamically
- [] Ensure smooth transitions between modes
- [] Test performance with large datasets

Acceptance Criteria

- [] Toggle exists and is styled consistently
- [] Daily view shows individual session durations per day
- [] Cumulative view shows running total over time
- [] Transition between modes is instant and clear

Labels

feature, charting, analytics, flutter, redesign

Estimate

M

markdown

Title

Implement Rolling Average Analysis for Usage Patterns

Description

To provide deeper insights into user behavior trends, add functionality for displaying a rolling average (e.g., 3-day or 7-day) of session durations. This helps smooth out day-to-day fluctuations and highlights evolving habits.

Tasks

- [] Add selector to choose rolling average window (None, 3-day, 7-day)
- [] Calculate rolling averages from session data
- [] Overlay rolling average line on the chart
- [] Update chart legend and tooltip to reflect average values
- [] Ensure performance with large data sets

Acceptance Criteria

- [] Rolling average selector is present and functional
- [] Average line updates correctly based on selected window
- [] Tooltips show both actual and average data when applicable
- [] View is clear and doesn't interfere with raw data visualization

Labels

enhancement, analytics, flutter, charting, redesign

Estimate

M

markdown

Title

Design and Implement Weekly Summary Visualization

Description

Create a new chart view that summarizes usage patterns by week. This view should help identify weekly trends and fluctuations in inhalation sessions, supporting long-term habit analysis.

Tasks

- [] Aggregate session data into weekly buckets (Monday-Sunday)
- [] Calculate total usage, average session duration, and session count per week
- [] Visualize data in a bar or line chart format
- [] Allow toggling between different summary metrics (e.g., total vs. average duration)
- [] Style consistently with the rest of the app's visual theme

Acceptance Criteria

- [] Users can switch to a "Weekly View" mode
- [] Data is accurately grouped and displayed
- [] Weekly metrics are labeled and legible
- [] Chart reflects updates from filtered date ranges

Labels

feature, analytics, visualization, flutter, redesign

Estimate

M

markdown

Title

Implement Rolling Average Chart for Usage Trends

Description

Add a line chart that displays a rolling 7-day average of session durations or counts to help visualize overall habit trends while smoothing out daily spikes or drops. This feature will support users in identifying sustained behavioral changes.

Tasks

- [] Calculate rolling average for session duration and session count
- [] Add a toggle for "Rolling Avg" mode in the analytics view
- [] Ensure chart updates when date range filters are applied
- [] Display tooltips showing actual vs. average values on hover/tap
- [] Maintain consistent styling with existing charts

Acceptance Criteria

- [] A new chart or overlay shows rolling 7-day averages
- [] Rolling average can be toggled on or off
- [] Averages are calculated correctly and update based on filters
- [] Visual cues differentiate actual values from rolling averages

Labels

feature, visualization, analytics, flutter, redesign

Estimate

M

markdown

Title

Refactor Data Processing Logic for Performance and Maintainability

Description

The current data processing logic used to transform log records for display in charts and tables is becoming complex and scattered across the codebase. This issue aims to centralize and optimize the transformation of raw log data into a structured format for all visual components.

Tasks

- [] Create a single transformation function that:
 - Filters records by selected date range
 - Sorts records in descending timestamp order
 - Prepares chart-compatible time series data
- [] Ensure compatibility with cumulative and rolling average calculations
- [] Write unit tests to verify transformations across multiple input shapes
- [] Use the centralized transformation in both the chart and table widgets

Acceptance Criteria

- [] Code duplication across chart and table logic is removed
- [] Data updates correctly when filters are changed
- [] Unit tests validate transformation logic for edge cases
- [] App performance remains consistent or improves

Labels

refactor, data-processing, flutter, enhancement

Estimate

M

markdown

Title

Implement Dynamic Date Range Picker for Custom Timeframe Analysis

Description

To enhance user control and flexibility, implement a dynamic date range picker allowing users to select custom timeframes for log analysis. This will replace or supplement the current fixed-range options (e.g., 7 days, 30 days).

Tasks

- [] Integrate a calendar-style date picker in the UI
- [] Update data transformation logic to respect the selected custom range

- [] Ensure charts and table update seamlessly on range change
- [] Validate mobile responsiveness and performance with long ranges
- [] Provide visual feedback for loading and range changes

Acceptance Criteria

- [] Users can select any valid start and end date from the UI
- [] The chart and table update accordingly
- [] Custom range selections persist during navigation
- [] No noticeable lag or crash on large timeframes

Labels

feature, UI/UX, enhancement, flutter

Estimate

M

markdown

Title

Add Rolling Averages Option to Usage Chart for Trend Analysis

Description

Introduce a toggle or dropdown to enable rolling average visualization on the usage chart. This allows users to identify trends more clearly over time by smoothing out short-term fluctuations.

Tasks

- [] Add UI control to enable/disable rolling averages
- [] Implement logic to calculate and overlay rolling averages (e.g., 7-day, 14-day)
- [] Ensure chart dynamically updates based on toggle or selection
- [] Handle edge cases with insufficient data points

Acceptance Criteria

- [] Users can switch between raw data and smoothed data views
- [] Rolling average line appears over the usage chart when enabled
- [] Tooltips reflect average values correctly
- [] Works seamlessly with custom date ranges

Labels

feature, charting, data-visualization, enhancement

Estimate

M

markdown

Title

Add Toggle Between Daily and Cumulative Views on Usage Chart

Description

Allow users to switch between daily inhalation lengths and cumulative usage over time. This provides different perspectives for behavior tracking and trend analysis.

Tasks

- [] Add UI toggle or dropdown for switching views
- [] Implement chart logic for both daily and cumulative calculations
- [] Ensure axes and labels update correctly with each mode
- [] Maintain tooltip and legend accuracy across modes

Acceptance Criteria

- [] Toggle allows seamless switching between daily and cumulative views
- [] Chart renders correctly in both modes
- [] Visual indicators (titles, labels) reflect the selected mode

Labels

feature, charting, data-visualization, UX

Estimate

M

markdown

Title

Implement "Clear Data" Option for Logged Entries

Description

Provide users with a clear and accessible way to delete all inhalation data logs from local storage and Firestore. This is useful for resetting progress, troubleshooting, or starting fresh.

Tasks

- [] Add "Clear Data" button to settings or relevant UI section
- [] Confirm action with a warning dialog before proceeding
- [] Remove all locally stored logs (e.g., Isar database)
- [] Remove associated documents from Firestore
- [] Log analytics event for data deletion (if tracking is enabled)

Acceptance Criteria

- [] Button is visible and intuitively placed
- [] Clicking the button prompts a confirmation dialog
- [] All data is successfully deleted upon confirmation
- [] User receives a success notification

Labels

feature, data-management, UX, firestore, local-storage

Estimate

M

markdown

Title

Display Daily Summary of Inhalation Data

Description

Add a visual and/or textual daily summary that aggregates the inhalation data for the current day. This should include total inhalations, cumulative duration, and optionally a simple chart or icon representation to convey daily usage patterns.

Tasks

- [] Add section to main screen or dashboard for daily summary
- [] Query and aggregate inhalation logs from current day
- [] Display total inhalations and cumulative duration
- [] Optionally visualize usage with a simple bar or pie chart
- [] Ensure real-time updates when new data is logged

Acceptance Criteria

- [] Summary accurately reflects the current day's data
- [] UI is responsive and clear
- [] Data updates dynamically as logs are added

Labels

feature, UI, data-visualization, summary

Estimate

M

markdown

Title

Implement Timestamp Filtering by Day, Week, and Month

Description

Add functionality to filter log entries and associated charts by specific time ranges: day, week, and month. This enables users to better analyze their inhalation patterns over

different periods.

Tasks

- [] Add UI toggle or dropdown for selecting time range (Day / Week / Month)
- [] Implement filtering logic based on selected time window
- [] Update charts and log views dynamically based on selected filter
- [] Ensure filter state is preserved during navigation
- [] Write unit tests for filtering logic

Acceptance Criteria

- [] Filtered views show correct logs and chart data per time range
- [] Switching between filters updates the data seamlessly
- [] No performance issues when switching time ranges

Labels

feature, filter, time-series, UI

Estimate

M

markdown

Title

Implement Daily Inhalation Stats Summary

Description

Create a feature that provides a summary of the user's inhalation data for each day. The summary should include the total inhalation time, number of sessions, and average session length, displayed at the top of the daily view.

Tasks

- [] Add calculation logic for daily total, count, and average inhalation duration
- [] Create a UI component for displaying daily summary stats
- [] Integrate the summary into the daily log view

- [] Ensure it updates based on selected date or range
- [] Add tests to validate correct summary calculations

Acceptance Criteria

- [] Summary displays accurate data for each day
- [] Works correctly when switching dates or filters
- [] UI is clean, accessible, and non-intrusive

Labels

enhancement, stats, UI, analytics

Estimate

M

markdown

Title

Implement Weekly and Monthly Inhalation Trends Visualization

Description

Add visual trend analysis to help users identify patterns in their inhalation habits. Create line or bar charts to display weekly and monthly totals, averages, and session frequency.

Tasks

- [] Aggregate inhalation data weekly and monthly
- [] Create a reusable chart component for trends
- [] Design UI elements for switching between views (weekly/monthly)
- [] Integrate the trends view into the main dashboard
- [] Ensure responsive layout for mobile and desktop
- [] Write tests for data aggregation and chart display

Acceptance Criteria

- [] User can toggle between weekly and monthly charts
- [] Charts display accurate data and scale correctly
- [] Visuals are clear, informative, and responsive

Labels

feature, analytics, chart, UI

Estimate

L

markdown

Title

Implement Rolling Average Chart for Inhalation Trends

Description

Add a chart to display rolling averages (e.g., 7-day or 30-day) of inhalation duration or frequency. This will provide users with a smoothed view of their behavior over time and help them identify gradual increases or decreases in usage.

Tasks

- [] Add data transformation logic for rolling average calculations
- [] Allow user to toggle between raw and smoothed data in chart UI
- [] Design and implement the rolling average chart component
- [] Integrate with existing range selector and chart controls
- [] Ensure performance optimization for large datasets
- [] Add tests for rolling average calculations and rendering

Acceptance Criteria

- [] User can toggle between raw data and rolling average views
- [] Chart updates smoothly and accurately reflects trends
- [] Feature works on both mobile and desktop interfaces

Labels

feature, analytics, chart, enhancement

Estimate

L

markdown

Title

Add Monthly Summary Statistics Section to Dashboard

Description

Introduce a summary section at the top or bottom of the dashboard showing monthly statistics such as total sessions, average inhalation duration, longest streak, and most frequent time of use. This section will provide users with high-level insights into their habits over time.

Tasks

- [] Define the set of summary statistics to be included
- [] Implement Firestore queries and aggregation logic
- [] Design the UI component for displaying monthly summaries
- [] Integrate with existing dashboard layout
- [] Make summaries responsive for mobile and desktop
- [] Add loading and error handling states
- [] Write unit tests for summary generation

Acceptance Criteria

- [] Users see a well-formatted summary with updated statistics for the current month
- [] Summary reflects accurate data based on user logs
- [] No major performance lag when loading dashboard

Labels

feature, dashboard, analytics, enhancement

Estimate

M

markdown

Title

Implement Rolling Usage Graph with Customizable Time Windows

Description

Add a new line graph component to the dashboard that visualizes rolling usage statistics. Users should be able to select a rolling window (e.g., 1 day, 3 days, 7 days) to smooth out inhalation trends and see broader behavioral patterns over time.

Tasks

- [] Create rolling aggregation logic for Firestore queries
- [] Add dropdown or input to allow user to select time window
- [] Update chart rendering logic to support dynamic datasets
- [] Style the graph consistently with the rest of the dashboard
- [] Handle edge cases such as sparse data
- [] Add tooltip information on hover

Acceptance Criteria

- [] Users can view rolling average usage over selected window
- [] Graph updates correctly when switching windows
- [] Performance remains smooth with large datasets

Labels

feature, dashboard, charts, analytics

Estimate

M

markdown

Title

Add Weekly Summary Card with Totals and Averages

Description

Design and implement a summary card on the dashboard that displays weekly usage statistics. This includes total inhalation count, total duration, and average usage per day for the current and previous week, with a clear visual

distinction.

Tasks

- [] Fetch and aggregate weekly data from Firestore
- [] Calculate totals and daily averages
- [] Design a responsive summary card component
- [] Display both current and previous week data
- [] Highlight differences (e.g., up/down arrows or color indicators)

Acceptance Criteria

- [] Users can quickly compare weekly stats at a glance
- [] Summary is accurate and updates as new data is logged
- [] Layout fits neatly into the dashboard and is mobile-responsive

Labels

feature, dashboard, analytics, summary

Estimate

S

markdown

Title

Implement Monthly Trend Chart with Toggle for Sessions vs. Duration

Description

Add a new chart to the dashboard that displays monthly trends with the ability to toggle between total session count and total duration. This will help users track long-term usage habits and identify patterns over time.

Tasks

- [] Fetch and aggregate monthly usage data from Firestore
- [] Design a line chart or bar chart that displays one year of data
- [] Add a toggle switch to change between "Sessions" and

"Duration"

- [] Ensure values are normalized for comparison if needed

Acceptance Criteria

- [] Chart shows 12 months of data accurately
- [] Toggle functionality works without reloads
- [] UI is responsive and visually consistent with the app design

Labels

feature, dashboard, chart, analytics

Estimate

M

markdown

Title

Create Data Insights Section with Weekly Average Comparisons

Description

Add a new section to the dashboard that displays weekly averages and compares them to the previous week. This will allow users to track progress or regression in their usage habits over time.

Tasks

- [] Calculate average duration and session count per week
- [] Compare current week's averages to the previous week
- [] Display percentage change with indicators (e.g., arrow up/down, color-coded)
- [] Place section below the main chart area for visibility

Acceptance Criteria

- [] Weekly averages are updated dynamically as data changes
- [] Clear, user-friendly indicators show comparison results
- [] Section is responsive and fits within mobile and desktop views

Labels

feature, analytics, dashboard, UX

Estimate

M

markdown

Title

Implement Toggle for Daily vs Weekly View in Dashboard Charts

Description

Add a toggle switch that allows users to switch between daily and weekly aggregated views of their smoking data. This will give users a clearer long-term perspective and better insight into patterns.

Tasks

- [] Add toggle component to chart section
- [] Adjust chart data source and grouping based on toggle state
- [] Ensure both views show accurate duration and session data
- [] Maintain smooth transitions between modes

Acceptance Criteria

- [] Toggle is intuitive and responsive
- [] Chart updates correctly based on selected mode
- [] Data is grouped and labeled clearly for both daily and weekly views

Labels

feature, UX, dashboard, data-visualization

Estimate

M

markdown

Title

Refactor Chart Component to Support Dynamic Data Sources

Description

To streamline future enhancements and simplify maintenance, refactor the chart rendering component to support dynamic data sources and configurations. This will allow charts to easily adapt to different datasets (e.g., daily, weekly, rolling average, etc.) and analysis types.

Tasks

- [] Abstract chart data preparation into a reusable utility
- [] Enable dynamic x/y axis configuration
- [] Ensure proper rendering for both cumulative and range-based visualizations
- [] Validate compatibility with current Firebase data structure

Acceptance Criteria

- [] Charts dynamically adapt to different data groupings without breaking
- [] Refactored code is modular, readable, and testable
- [] No regression in chart behavior or UI

Labels

refactor, data-visualization, enhancement, technical-debt

Estimate

L

markdown

Title

Implement Enhanced Range Selector with Presets

Description

Develop an improved range selector for filtering chart data that includes common preset options (e.g., Last 7 Days, Last

30 Days, Custom Range). This will improve usability and allow users to quickly analyze trends across different time periods.

Tasks

- [] Add dropdown or button-based preset selection UI
- [] Link presets to corresponding date range logic
- [] Maintain compatibility with custom start/end date pickers
- [] Ensure changes trigger chart re-render with correct data

Acceptance Criteria

- [] Presets display correctly and update the chart on selection
- [] Custom date ranges remain functional alongside presets
- [] Works seamlessly with current cumulative and range-based analysis

Labels

feature, UI, enhancement, charting

Estimate

M

markdown

Title

Add Rolling Average Toggle for Data Visualization

Description

Introduce a toggle feature that enables users to view a rolling average (e.g., 7-day, 30-day) of their usage data on the chart. This will help smooth out fluctuations and reveal longer-term trends.

Tasks

- [] Implement toggle UI element (checkbox or switch)
- [] Calculate rolling averages based on selected window
- [] Update chart data accordingly when toggled
- [] Ensure performance is acceptable for large datasets

Acceptance Criteria

- [] Toggle updates chart without page reload
- [] Rolling average is clearly distinguishable (e.g., dashed line)
- [] Original data remains viewable when toggle is off

Labels

feature, data analysis, enhancement, charting

Estimate

M

markdown

Title

Enable Export of Filtered Usage Data to CSV

Description

Add functionality that allows users to export the currently displayed (filtered and sorted) usage data to a CSV file for personal analysis or backup. This will improve usability for users who want to analyze their data in spreadsheet tools.

Tasks

- [] Add "Export to CSV" button below the chart and table
- [] Ensure the export reflects all applied filters (date range, search terms, etc.)
- [] Include relevant headers like Timestamp, Inhalation Length, and Notes
- [] Validate exported CSV opens correctly in Excel and Google Sheets

Acceptance Criteria

- [] Button triggers download without requiring page reload
- [] Exported file includes only visible (filtered) data
- [] File format is compatible with major spreadsheet applications

Labels

feature, export, enhancement, data

Estimate

M

markdown

Title

Implement User Profile Management

Description

Introduce basic user profile management to support long-term use, personalization, and future feature enhancements such as cloud backups, reminders, or cross-device syncing.

Tasks

- [] Create user profile data model in local storage
- [] Implement editable user details (name, preferred units, avatar)
- [] Add settings screen section for profile management
- [] Save profile information persistently
- [] Lay groundwork for future online sync

Acceptance Criteria

- [] Users can view and update their profile details
- [] Changes persist across app restarts
- [] Feature integrates cleanly with existing settings UI

Labels

feature, user-settings, enhancement, UX

Estimate

M

markdown

Title

Add Export Feature for Smoke Log Data

Description

Allow users to export their smoking log data to a CSV or JSON file format for external analysis, backups, or sharing with healthcare providers. This enhances the utility of the app and supports transparency and self-monitoring.

Tasks

- [] Design export button UI (e.g., from settings or log screen)
- [] Serialize local data into CSV and JSON formats
- [] Prompt user for format and save location
- [] Implement file sharing/integration with device file manager

Acceptance Criteria

- [] Users can export their full smoking history in at least one format
- [] Exported files contain all key fields (timestamp, duration, etc.)
- [] Feature functions on both Android and iOS

Labels

feature, data-export, enhancement

Estimate

M

markdown

Title

Implement Daily and Weekly Smoking Summary Stats

Description

Create a visual summary section that shows daily and weekly statistics of smoking behavior. This should include total

sessions, cumulative duration, and average session length. The goal is to provide users with quick, digestible insights to track patterns and encourage reduction.

Tasks

- [] Design layout for daily and weekly summary sections
- [] Calculate total sessions, average duration, and cumulative duration
- [] Display in dashboard or a dedicated analytics page
- [] Write unit tests for statistics calculations

Acceptance Criteria

- [] Users can view summaries by day and week
- [] Stats update automatically when new data is added
- [] Layout is responsive and easy to interpret

Labels

feature, analytics, enhancement

Estimate

M

markdown

Title

Implement Tagging System for Smoke Logs

Description

Introduce a tagging feature that allows users to categorize individual smoking sessions. Tags might include mood, environment, social context, or stress level. This will help users identify patterns and triggers related to their smoking habits.

Tasks

- [] Design tag selection UI for logging screen
- [] Allow custom tag creation and saving
- [] Store tags in Firestore with each session
- [] Update charts and summaries to support tag-based filters

Acceptance Criteria

- [] Users can select or create tags when logging a session
- [] Tags are saved and displayed with session entries
- [] Filtering by tag updates data views accordingly

Labels

feature, UX, enhancement

Estimate

M

markdown

Title

Add Rolling Averages to Usage Charts

Description

Enhance the current usage visualization by introducing rolling averages (e.g., 3-day, 7-day, 14-day) to the line chart. This feature will help users observe trends in their smoking behavior over time with less day-to-day noise.

Tasks

- [] Implement rolling average calculation on filtered data
- [] Add UI toggle or dropdown to select rolling average window
- [] Update chart rendering logic to overlay rolling average
- [] Ensure proper labeling and tooltips for new line

Acceptance Criteria

- [] Users can toggle and view rolling averages on the main chart
- [] Averages update correctly based on selected date range and filters
- [] No performance issues or UI glitches introduced

Labels

feature, charts, enhancement

Estimate

M

markdown

Title

Implement Inhalation Heatmap by Time of Day

Description

Introduce a heatmap visualization that displays inhalation frequency/intensity by hour across days of the week. This will help users understand their peak smoking times and identify patterns or habits.

Tasks

- [] Process timestamp data to extract hour and weekday
- [] Aggregate inhalation count or duration per time block
- [] Create a heatmap chart component with appropriate axes and color gradients
- [] Add toggle in UI to switch between standard chart and heatmap view

Acceptance Criteria

- [] Users can toggle to view a heatmap of usage by time of day
- [] Heatmap clearly shows time blocks with higher and lower activity
- [] Chart adjusts with filtered date ranges and maintains responsive layout

Labels

feature, visualization, analytics

Estimate

M

markdown

Title

Integrate Rolling Average Usage Line into Time Series Chart

Description

Add a secondary line to the main inhalation time series chart that shows a rolling average (e.g., 7-day or 30-day) of total usage. This will help users visualize long-term trends and fluctuations in their habits.

Tasks

- [] Add backend logic to compute rolling averages based on filtered range
- [] Implement UI toggle to enable/disable rolling average overlay
- [] Style the rolling average line distinct from the raw data line
- [] Ensure legend and tooltips update accordingly

Acceptance Criteria

- [] Users can optionally view a rolling average on the inhalation chart
- [] The rolling average reflects the selected date range
- [] No visual clutter or performance degradation

Labels

enhancement, visualization, analytics

Estimate

M

markdown

Title

Refactor Data Filtering Logic into Central Controller

Description

Consolidate the data filtering and sorting logic (currently scattered across the chart widget and data source) into a

centralized controller or provider. This will improve maintainability and allow seamless integration of new analysis types like cumulative and rolling averages.

Tasks

- [] Create a centralized data controller for managing filters and sorting
- [] Move filtering logic from chart and table components
- [] Ensure all visualizations (charts and tables) consume data from this controller
- [] Add support for dynamic range selection in the centralized logic

Acceptance Criteria

- [] All visual elements pull from the same filtered dataset
- [] Changes to the range selector or other filters automatically update all views
- [] Code is modular and easier to extend

Labels

refactor, architecture, enhancement

Estimate

L

markdown

Title

Implement Daily and Weekly Usage Summary Cards

Description

Create summary cards on the dashboard that display total usage statistics for the current day and week. This provides a quick-glance overview of user behavior and encourages mindfulness.

Tasks

- [] Design summary card UI components for daily and weekly usage

- [] Aggregate filtered log data by day and week
- [] Display total duration and entry count
- [] Ensure live updates when filters change

Acceptance Criteria

- [] Summary cards are visible and styled consistently with the app
- [] Statistics update correctly based on currently filtered data
- [] Component is responsive and mobile-friendly

Labels

feature, dashboard, enhancement

Estimate

M

markdown

Title

Add Clear Filters Button to Dashboard

Description

Implement a button on the dashboard that allows users to quickly reset all filters (date range, chart type, etc.) to their default state. This improves usability and makes it easier for users to start fresh without manually resetting each filter.

Tasks

- [] Create a "Clear Filters" button UI component
- [] Wire up the button to reset all filter state
- [] Ensure charts and data views refresh accordingly
- [] Add tooltip or label for clarity

Acceptance Criteria

- [] Button is accessible and clearly labeled
- [] All filters reset to default when clicked
- [] Data refreshes and matches the default state

- [] Component follows existing design style

Labels

feature, UX, enhancement

Estimate

S

markdown

Title

Implement Rolling Usage Average Chart

Description

Introduce a new chart type that visualizes the rolling average of usage over time. This feature will help users identify long-term trends in their smoking behavior by smoothing out daily fluctuations.

Tasks

- [] Add new rolling average option to chart type selector
- [] Calculate rolling average (default 7-day window)
- [] Render the rolling average on the usage chart
- [] Allow user to adjust the window size (e.g., 7, 14, 30 days)
- [] Ensure performance optimization for large datasets

Acceptance Criteria

- [] Rolling average chart is selectable and renders correctly
- [] Users can adjust the time window and see updates
- [] Data accurately reflects the smoothed usage
- [] Visual styling matches other chart types

Labels

feature, data-visualization, enhancement

Estimate

M

markdown

Title

Create Streak Tracker Component

Description

Develop a component that tracks and displays the user's longest streak without smoking. This feature aims to motivate users by highlighting personal achievements and progress.

Tasks

- [] Add logic to detect streaks based on log timestamps
- [] Store and update longest streak data
- [] Display current and longest streaks in the dashboard
- [] Add visual indicators or celebratory icons for new records

Acceptance Criteria

- [] Streaks are accurately calculated from the user's logs
- [] Longest streak is stored and displayed correctly
- [] UI component is styled consistently with the rest of the app
- [] Component updates automatically as new logs are added

Labels

feature, motivational, enhancement

Estimate

M

markdown

Title

Implement Summary Insights Panel

Description

Create a summary panel that gives users key insights into their smoking behavior over selected time ranges. This panel should support metrics like total sessions, average inhalation duration, most active hours, and days with no logs.

Tasks

- [] Design the summary panel UI component
- [] Integrate with filtered data from date selector
- [] Calculate total sessions, average inhalation, most active hour, and log-free days
- [] Display summary stats in a visually clear format (e.g., icons, mini-charts, or badges)

Acceptance Criteria

- [] Summary panel updates in real time based on selected date range
- [] Metrics are accurate and clear
- [] Styling matches rest of dashboard design

Labels

feature, data-insights, UI

Estimate

M

markdown

Title

Add Rolling Usage Chart for Daily and Weekly Trends

Description

Implement a rolling usage chart to help users visualize how their smoking patterns evolve over time. This chart should support daily and weekly rolling averages and respond to date range selections.

Tasks

- [] Design and implement a chart that displays rolling

averages

- [] Add a toggle for daily vs. weekly rollups
- [] Connect to filtered and sorted data pipeline
- [] Ensure responsive behavior and match styling to existing charts

Acceptance Criteria

- [] Users can switch between daily and weekly rolling views
- [] Rolling data updates based on selected date range
- [] Visual consistency with other dashboard charts

Labels

feature, chart, UX, data-analysis

Estimate

M

markdown

Title

Implement User Summary Snapshot Panel

Description

Create a dashboard panel that provides a quick summary of key user insights, including total sessions, average daily usage, longest streak, and most common usage time. This will give users an at-a-glance overview of their habits.

Tasks

- [] Design summary panel UI with responsive layout
- [] Calculate and display total sessions and average daily usage
- [] Determine and show longest usage streak
- [] Identify and highlight most frequent time of day for use
- [] Integrate with existing filtering and data pipeline

Acceptance Criteria

- [] Users see updated summary data based on current range and filters

- [] All four data points are accurately calculated and displayed
- [] Matches styling and spacing of other dashboard elements

Labels

feature, UX, data-insight, summary

Estimate

M

markdown

Title

Enhance Entry Form for Accessibility and Mobile Usability

Description

Update the entry form UI to improve accessibility and mobile usability. This includes better focus states, keyboard navigation, touch-friendly input spacing, and screen reader labels.

Tasks

- [] Implement proper `aria-labels` and roles for all form elements
- [] Ensure form is fully navigable using keyboard alone
- [] Optimize touch targets for mobile (minimum 48px)
- [] Improve focus outlines and input spacing
- [] Test on screen reader (e.g., VoiceOver or NVDA) and mobile devices

Acceptance Criteria

- [] All fields can be accessed and submitted using only the keyboard
- [] Input fields have descriptive labels for screen readers
- [] Touch targets meet accessibility guidelines
- [] Form works smoothly on screens 320px wide and up

Labels

accessibility, mobile, frontend, enhancement

Estimate

M

markdown

Title

Add Real-Time Character Counter to Inhalation Notes

Description

Introduce a real-time character counter for the optional inhalation notes field to guide users on input limits and improve usability. This will enhance user experience by preventing overly long entries and aligning with potential future database constraints.

Tasks

- [] Add a character counter beneath the notes input field
- [] Set a maximum limit of 300 characters
- [] Dynamically update the counter as the user types
- [] Display an error if the user exceeds the limit

Acceptance Criteria

- [] Notes field shows remaining characters (e.g., "200/300 characters remaining")
- [] Users cannot submit notes over 300 characters
- [] Character counter works across desktop and mobile views

Labels

feature, frontend, UX

Estimate

S

markdown

Title

Implement “This Week” Quick Filter for Inhalation Logs

Description

Introduce a “This Week” quick filter button on the home screen and charts view to help users easily access inhalation data for the current week (Monday to Sunday). This improves usability by offering a convenient timeframe commonly reviewed by users.

Tasks

- [] Add a “This Week” button to the date range selector
- [] Calculate the start and end of the current week dynamically
- [] Apply filter and refresh logs/charts when selected
- [] Highlight the active quick filter visually

Acceptance Criteria

- [] Selecting “This Week” updates charts and logs to only show current week's data
- [] Button state clearly shows it's active
- [] Works across timezone changes and handles week transitions smoothly

Labels

feature, frontend, filters, usability

Estimate

S

markdown

Title

Add Scroll-to-Top Button on Logs and Charts Views

Description

Introduce a floating “Scroll-to-Top” button that appears after the user scrolls down a certain distance in the Logs and Charts views. This small UX enhancement allows for easier navigation, especially on long pages or when reviewing extended data.

Tasks

- [] Implement floating button using standard Flutter positioning
- [] Show button only after scrolling 1.5 screen heights down
- [] Smoothly scroll to top when button is pressed
- [] Style the button to match app theme (e.g., subtle background, rounded edges)

Acceptance Criteria

- [] Button is visible only when applicable
- [] Button scrolls view to top smoothly
- [] Does not interfere with log items or chart interactions

Labels

enhancement, frontend, UX, quality-of-life

Estimate

XS

markdown

Title

Add Toggle Between Table and Chart View on Dashboard

Description

Enhance the Dashboard screen by allowing users to switch between a tabular data view and the existing chart-based visualizations. This provides greater flexibility for users who prefer to view raw data instead of summaries or trends.

Tasks

- [] Create a toggle switch or segmented control at the top of the Dashboard
- [] Implement tabular view to display recent logs in a scrollable DataTable
- [] Retain existing chart view functionality
- [] Maintain user selection across app restarts (using local persistence)

```
### Acceptance Criteria
- [ ] Users can easily switch views on the Dashboard
- [ ] Tabular view presents clear, sortable columns
(Timestamp, Length, Notes, etc.)
- [ ] Toggle preserves choice across sessions
- [ ] No regressions in existing chart functionality
```

Labels

feature, frontend, dashboard, UX

Estimate

M

markdown

Title

Implement "Streak Tracker" Feature for Smoke Logging

Description

Introduce a "Streak Tracker" that visually displays consecutive days of logging activity. This motivational feature encourages consistent use of the app and reinforces daily tracking habits.

Tasks

- [] Design visual layout for a streak tracker (calendar-based or numeric chain)
- [] Implement logic to calculate daily logging streaks
- [] Display current streak and longest streak
- [] Handle missed days gracefully (e.g., streak resets)
- [] Include tooltips or explanations for how streaks are counted

Acceptance Criteria

- [] Streak tracker updates in real-time with each log entry
- [] Users can clearly see their current and best streaks
- [] Reset behavior is intuitive and documented
- [] Feature fits within existing dashboard UI or log screen

without clutter

Labels

feature, enhancement, UX, motivation

Estimate

M

markdown

Title

Add Rolling Usage Trend Chart with Selectable Range

Description

Implement a line chart to display rolling usage trends (e.g., 3-day, 7-day average) to help users visualize changes in smoking behavior over time. This chart will be configurable by range and update dynamically based on the selected timeframe.

Tasks

- [] Create a reusable chart component for displaying rolling averages
- [] Add a range selector (e.g., 7, 14, 30 days)
- [] Compute rolling averages based on selected window
- [] Integrate chart with existing filtered data logic
- [] Ensure responsive design for various screen sizes

Acceptance Criteria

- [] Users can select a rolling window (3/7/14/30 days)
- [] Chart updates in real-time based on the selected range
- [] Chart is visually integrated with the app's theme and responsive
- [] Data aggregation and display are accurate

Labels

feature, chart, data-visualization, enhancement

Estimate

M

markdown

Title

Implement Usage Streak Tracker Widget

Description

Create a visual component that tracks and displays a user's usage streaks, such as consecutive days logged or smoke-free days. This feature aims to encourage consistency and promote mindful tracking.

Tasks

- [] Design streak tracking logic (e.g., based on daily log presence)
- [] Create visual indicators (e.g., fire icon, count display)
- [] Add widget to dashboard
- [] Include tooltip with last streak start date and longest streak info
- [] Update streak in real-time when new logs are added or deleted

Acceptance Criteria

- [] Streak count updates accurately based on daily log activity
- [] UI clearly shows current streak and longest streak
- [] Widget is responsive and fits the overall dashboard theme
- [] Tooltip provides relevant streak insights

Labels

feature, enhancement, UX, widget

Estimate

M

markdown

Title

Build Log Type Selector and Filtering UI

Description

Introduce a UI component that allows users to filter smoke logs by type (e.g., cigarette, vape, THC, CBD, etc.). This will help users identify patterns based on substance and customize their log views.

Tasks

- [] Define available log types and associated icons/colors
- [] Create filter UI with checkboxes or toggles
- [] Integrate filtering with log display components
- [] Persist selected filters across sessions using local storage or settings
- [] Add reset option to clear filters

Acceptance Criteria

- [] Users can filter logs by one or more substance types
- [] Filtered view updates immediately and accurately
- [] Design is mobile-friendly and consistent with app styling
- [] Filter settings persist when navigating between pages

Labels

feature, UI, enhancement, filter

Estimate

M

markdown

Title

Add Editable Log Entries Feature

Description

Allow users to edit existing log entries to correct mistakes or add missing details such as tags, length, type, or notes.

Tasks

- [] Add edit icon/button to each log entry
- [] Create a modal or in-line form for editing
- [] Prepopulate form with existing data
- [] Validate and save changes to Firestore
- [] Refresh UI upon update

Acceptance Criteria

- [] Users can edit any part of a log entry
- [] Edits are reflected immediately in the UI
- [] All edits are persisted correctly in the database
- [] Edit interface matches existing design language

Labels

feature, enhancement, CRUD, Firestore

Estimate

M

markdown

Title

Implement Tag Filtering for Log Visualization

Description

Enhance user experience by allowing filtering of logs in charts and tables based on tags. This will help users analyze trends for specific categories (e.g., "stress", "sleep", "celebration").

Tasks

- [] Add a dropdown or multi-select chip interface for tag filtering
- [] Update query logic to filter chart and table data by selected tags
- [] Preserve filter state during navigation within the app

- [] Ensure responsive UI updates for both charts and tables

Acceptance Criteria

- [] Users can select one or more tags to filter log data
- [] Charts and tables update in real-time based on selected tags
- [] Filter interface is intuitive and mobile-friendly

Labels

feature, enhancement, filters, UI, UX

Estimate

M

markdown

Title

Add Option to Export Log Data (CSV/JSON)

Description

Provide users with the ability to export their smoking log data for personal records, analysis, or sharing with healthcare professionals. Export options should include both CSV and JSON formats.

Tasks

- [] Add an export button in the UI with CSV and JSON options
- [] Format the exported data with all relevant fields (timestamp, duration, tags, notes)
- [] Ensure timezones are properly handled
- [] Securely handle permissions for data export on mobile and web

Acceptance Criteria

- [] Users can export data as CSV or JSON
- [] Exported data matches what is shown in the charts and tables
- [] UI is clean, accessible, and mobile-friendly

Labels

feature, data-export, UX, enhancement

Estimate

S

markdown

Title

Implement Tag-Based Filtering for Log Entries

Description

Allow users to assign tags to individual log entries (e.g., "stress", "celebration", "social") and filter visualizations and tables based on these tags. This enhances personalized analysis and trend discovery.

Tasks

- [] Add tagging support to the log entry submission form
- [] Store and retrieve tags in Firestore
- [] Enable multi-tag filtering for charts and tables
- [] Update UI to support tag selection and display

Acceptance Criteria

- [] Users can add and edit tags per entry
- [] Tags are visible in the data table
- [] Filters update visualizations and tables accordingly

Labels

feature, filtering, UI, enhancement

Estimate

M

markdown

Title

Add Rolling Average Chart for Weekly Usage Trends

Description

Implement a line chart that displays the 7-day rolling average of usage based on selected filters. This visualization will help users observe longer-term patterns and identify changes in behavior over time.

Tasks

- [] Create a function to compute the 7-day rolling average from log data
- [] Integrate the rolling average into the charting logic
- [] Add toggle control in UI to switch between raw data and rolling average
- [] Ensure responsiveness and clarity of the chart

Acceptance Criteria

- [] Chart displays accurate 7-day rolling average values
- [] Toggle switch functions without page reloads
- [] Chart updates according to date and tag filters

Labels

chart, data-visualization, enhancement, feature

Estimate

M

markdown

Title

Optimize Local Storage Handling with Isar

Description

Replace temporary in-memory data structures with persistent Isar storage to improve performance and data reliability for offline use. This supports long-term log retention and speeds up filtering and charting operations on larger datasets.

Tasks

- [] Integrate Isar as a local database dependency

- [] Create model schemas for log entries
- [] Migrate current data logic to use Isar for read/write
- [] Implement offline-first logic with fallback to Firestore when needed

Acceptance Criteria

- [] Logs are stored and loaded from Isar
- [] Filtering, sorting, and analysis functions use Isar queries
- [] Data syncs correctly with Firestore when online
- [] App functions correctly while offline

Labels

persistence, database, enhancement, performance

Estimate

L

markdown

Title

Implement Rolling Usage Chart Based on Selected Date Range

Description

Add a new line chart that visualizes rolling usage over the selected date range. This will provide users with insight into how their usage trends evolve over time. This chart should compute cumulative inhalation duration and reset at the start of each range.

Tasks

- [] Add rolling usage calculation logic to the filtering module
- [] Integrate rolling usage line chart below the bar chart
- [] Allow toggling between total, average, and rolling views
- [] Ensure charts update when new data is submitted or the range is changed

Acceptance Criteria

- [] Chart accurately displays cumulative usage per time unit
- [] Chart responds to date range and data filter changes
- [] UI provides a clear toggle between analysis modes

Labels

visualization, feature, enhancement

Estimate

M

markdown

Title

Refactor Chart and Table Logic into a Unified Data Filtering Service

Description

Consolidate all chart and table data filtering, sorting, and transformation logic into a shared service or module. This enables consistent handling of range-based data across all visualizations (e.g. bar, pie, rolling) and the log table.

Tasks

- [] Create a centralized filtering and sorting service
- [] Move all existing logic from charts and table widgets to the new service
- [] Ensure new charts and future enhancements (e.g., line chart, heatmap) use this service
- [] Add unit tests to validate transformations for each analysis mode

Acceptance Criteria

- [] All visualizations reflect identical filtered data for a selected range
- [] New data submitted updates all components using the centralized logic
- [] Chart behavior is modular and easy to extend with additional visualizations

Labels

refactor, architecture, enhancement

Estimate

L

markdown

Title

Implement Rolling Usage Line Chart with Filter Controls

Description

Introduce a new rolling usage line chart visualization to the AshTrail (Smoke-Log) redesign. This chart will show cumulative or averaged use over time within a selected range and interval (e.g., daily, weekly). Users should be able to adjust the range and rolling interval using the shared filtering service.

Tasks

- [] Design and implement a responsive line chart component
- [] Integrate with the centralized data filtering and sorting service
- [] Add interval options (e.g., 1d, 7d) to the range selector UI
- [] Ensure tooltips and axes clearly represent the rolling time window

Acceptance Criteria

- [] Users can select a range and view a rolling line chart reflecting inhalation patterns
- [] The chart updates live with new data submissions
- [] Matches behavior and style with other visualizations for consistency

Labels

feature, visualization, chart

Estimate

M

markdown

Title

Add Skeleton Loader and Error Display for Data-Dependent Widgets

Description

To improve user experience during data fetch operations, introduce a skeleton loading state and a clean error display for all widgets dependent on Firestore data in the AshTrail (Smoke-Log) redesign. This will ensure users are clearly informed when data is loading or if an error has occurred.

Tasks

- [] Implement reusable skeleton loading components for charts and tables
- [] Display a fallback error message with retry options if data fails to load
- [] Integrate loading and error states into each existing widget (e.g., summary stats, line chart, table view)

Acceptance Criteria

- [] Loading skeletons are visible while data is being fetched
- [] Errors are gracefully handled with an informative message
- [] Retry functionality exists for failed fetches

Labels

enhancement, UX, error-handling

Estimate

S

markdown

Title

Refactor Firestore Query Logic to Improve Performance and Maintainability

Description

As part of the AshTrail (Smoke-Log) redesign, centralize and optimize Firestore query logic. Current logic is scattered across components, making it harder to maintain and potentially inefficient for large datasets.

Tasks

- [] Create a centralized FirestoreService class for all data interactions
- [] Move existing queries from widgets into the service
- [] Optimize query usage with indexes, pagination, and filtering where appropriate
- [] Update all components to use the centralized service

Acceptance Criteria

- [] All Firestore queries are consolidated into a dedicated service
- [] Components only call methods from the service
- [] Unit tests cover all query methods in the service
- [] Performance improves when querying large time ranges

Labels

refactor, performance, backend, data-layer

Estimate

M

markdown

Title

Add Local Caching Layer for Smoke Log Data to Improve Load Times

Description

To reduce Firestore read costs and improve user experience, implement a local caching layer using Isar or another suitable Flutter database. Cached data should automatically refresh when new data is available or at defined intervals.

Tasks

- [] Set up Isar as a local data store
- [] Implement logic to cache fetched data from Firestore
- [] Invalidate and refresh cache based on timestamp or versioning
- [] Ensure offline support uses cached data
- [] Add unit tests for caching logic

Acceptance Criteria

- [] Smoke logs are served from local cache by default
- [] Data syncs with Firestore when online
- [] Cache refreshes daily or when triggered
- [] App works smoothly offline using cached data

Labels

enhancement, performance, offline, flutter

Estimate

L

markdown

Title

Create Editable Log Detail View for Reviewing and Updating Entries

Description

Users should be able to view the full details of a smoke log entry in a dedicated detail screen. This view should support both read-only and editable modes, allowing users to make changes to specific fields such as timestamp, inhalation length, notes, or mood.

Tasks

- [] Design and implement detail view screen layout
- [] Fetch and display all relevant log data
- [] Add "Edit" toggle or button to enter edit mode
- [] Implement form inputs and validation for editable fields
- [] Add save/cancel logic for updates
- [] Push changes back to Firestore and update local cache

Acceptance Criteria

- [] Log detail view displays full entry information
- [] Users can toggle between read-only and edit modes
- [] Field validation prevents invalid data entry
- [] Updates are saved to Firestore and reflected in app

Labels

feature, firebase, ui, flutter

Estimate

M

markdown

Title

Implement Mood Tagging System for Smoke Log Entries

Description

Enhance the user experience by allowing users to tag their mood or emotional state during each smoke session. This will support better self-awareness and enable trend analysis over time.

Tasks

- [] Design mood tag icons (e.g., happy, sad, anxious, relaxed)
- [] Add mood tag selection UI to the logging interface
- [] Store selected mood tag in Firestore with each log entry
- [] Update data model to include mood field
- [] Reflect mood tags in log summary and detail views
- [] Include mood data in future visualizations and analysis

Acceptance Criteria

- [] Users can select a mood tag when logging an entry
- [] Mood tags are saved and displayed in entry views
- [] Mood data is accurately stored and retrieved from Firestore

Labels

feature, firebase, mood, ui, flutter

Estimate

M

markdown

Title

Implement Daily Reflection Summary Feature

Description

Introduce a daily reflection feature where users can review their smoking patterns and enter brief notes about their day, mood, or progress. This can help promote mindful usage and support long-term behavior change.

Tasks

- [] Design UI for daily reflection input (text field + optional prompts)
- [] Display reflection summary on the home or dashboard screen
- [] Store reflection data in Firestore, keyed by date
- [] Include ability to edit previous reflections
- [] Add support for mood and session count overview in summary

Acceptance Criteria

- [] Users can add/edit a reflection for each day
- [] Reflection data is stored and retrieved correctly
- [] Daily reflections appear in a visually appealing and accessible format

Labels

feature, firebase, reflection, ui, flutter

Estimate

M

markdown

Title

Add Streak Tracker and Visual Progress Indicators

Description

Implement a feature to track user streaks (e.g., days with reduced or no use) and display visual indicators of progress. This gamification element can help motivate users by reinforcing positive habits and providing a clear sense of achievement.

Tasks

- [] Define rules for streak calculation (e.g., use below a certain threshold or abstinence)
- [] Add backend logic to track streaks per user
- [] Create visual component for displaying streaks on dashboard
- [] Implement badges or icons for milestones (e.g., 3 days, 7 days, 30 days)

Acceptance Criteria

- [] Streak data updates daily based on logged usage
- [] UI clearly displays active streak and previous best
- [] Milestone indicators are shown when achieved

Labels

enhancement, gamification, flutter, firebase, ui

Estimate

M

markdown

Title

Implement Rolling Averages for Usage Visualization

Description

Introduce rolling average metrics (e.g., 3-day, 7-day averages) to provide users with smoother trends in their smoking behavior. This helps mitigate the impact of outlier days and offers a more digestible view of long-term progress.

Tasks

- [] Add computation logic for rolling averages in data filtering
- [] Provide toggle options for users to view raw data vs. smoothed trends
- [] Update charts to display selected rolling average
- [] Ensure averages recalculate dynamically based on selected date range

Acceptance Criteria

- [] Users can toggle between raw data and rolling averages (3-day, 7-day)
- [] Visualizations reflect accurate smoothed values without affecting original data
- [] Rolling average computation is efficient and responsive

Labels

enhancement, data-visualization, analytics, flutter

Estimate

M

markdown

Title

Add Daily Total and Weekly Summary Cards

Description

Enhance the dashboard by adding daily and weekly summary cards that show total inhalation length, count of sessions, and average inhalation duration. These cards will help users quickly assess their daily and weekly usage patterns without analyzing charts.

Tasks

- [] Create card components for daily and weekly summaries
- [] Compute total inhalation duration per day/week
- [] Count the number of inhalation sessions per day/week
- [] Calculate average inhalation duration per session
- [] Display these stats in a responsive layout

Acceptance Criteria

- [] Daily and weekly summary cards are visible on the dashboard
- [] All metrics (total, count, average) are accurate and reflect current date range
- [] Cards update dynamically when data is filtered

Labels

feature, dashboard, enhancement, flutter, ux

Estimate

M

markdown

Title

Implement Usage Streak and Milestone Tracking

Description

Add a feature to track usage streaks and personal milestones for users. This enhancement will encourage consistent logging behavior and help users stay motivated by celebrating progress.

Tasks

- [] Track consecutive days with at least one log

- [] Identify and display streaks (e.g., "5-day streak!")
- [] Define and detect milestone achievements (e.g., "100 sessions logged")
- [] Display streak/milestone notifications on dashboard
- [] Store streak/milestone data in local cache or Firestore

Acceptance Criteria

- [] Streaks and milestones appear accurately on dashboard
- [] Notifications update in real-time with new data
- [] No streak is shown if logs are missing on a day
- [] Milestones include total usage count and total inhalation time

Labels

feature, gamification, flutter, enhancement, UX

Estimate

M

markdown

Title

Add Log Review and Edit Functionality

Description

Users need the ability to review past logs and make edits to correct any mistakes or add missing context. This will improve data accuracy and user control over their own data.

Tasks

- [] Create a log history screen with a list of recent entries
- [] Allow tapping a log to open a detail view
- [] Enable editing of inhalation duration and timestamp
- [] Save changes back to Firestore and update local cache
- [] Show visual indicator if a log has been edited

Acceptance Criteria

- [] Users can see a history of their logs

- [] Editing and saving a log updates the data everywhere it appears
- [] UI clearly differentiates between original and edited logs
- [] All edits are tracked with a last modified timestamp

Labels

feature, flutter, firestore, UI, UX, enhancement

Estimate

M

markdown

Title

Implement Rolling Average Usage Chart

Description

Add a new chart type to visualize rolling average inhalation duration over user-selected timeframes. This will help users better understand long-term trends in their usage habits.

Tasks

- [] Add dropdown or toggle to switch chart mode (raw vs rolling average)
- [] Calculate rolling averages based on selected window (e.g., 7-day, 14-day)
- [] Display new line chart with smoothed values
- [] Update legends and tooltips to reflect rolling calculations
- [] Cache or debounce calculations for performance

Acceptance Criteria

- [] Users can toggle between raw and rolling average charts
- [] Chart updates dynamically based on selected window size
- [] Visual feedback shows smoothed trends clearly and accurately
- [] Performance is acceptable even with large datasets

Labels

enhancement, data-visualization, charting, UX, flutter, firestore

Estimate

M

markdown

Title

Refactor Timestamp Range Selection and Filtering Logic

Description

Consolidate and relocate all timestamp range selection and data filtering logic to a centralized component or utility. This will ensure consistent filtering across all charts and tables, and will make future enhancements (e.g., rolling analysis, advanced filters) easier to implement.

Tasks

- [] Create a shared filtering utility or context
- [] Move timestamp filtering logic out of individual chart widgets
- [] Ensure all visualizations use the centralized range filter
- [] Update UI controls to sync with new structure
- [] Test for regressions in data filtering behavior

Acceptance Criteria

- [] All charts and tables reflect the selected timestamp range accurately
- [] Logic for timestamp filtering exists in a single place
- [] No noticeable performance or accuracy regressions

Labels

refactor, enhancement, filtering, timestamp, flutter, firestore

Estimate

M

markdown

Title

Implement Daily Summary Card with Rolling Totals

Description

Create a daily summary card that displays total usage, session count, and average session length. Integrate rolling averages (7-day, 30-day) to provide contextual insights. This card should be prominently displayed at the top of the dashboard for quick review.

Tasks

- [] Design a responsive UI card component
- [] Query Firestore for daily totals and calculate rolling averages
- [] Display total session time, count, and average length
- [] Format for readability (e.g., HH:mm:ss for durations)
- [] Add visual cues to show trends (e.g., arrows or color changes)

Acceptance Criteria

- [] Card displays accurate totals and rolling averages
- [] UI updates reactively with data changes
- [] Styling is consistent with app theme
- [] Works across all screen sizes

Labels

feature, dashboard, data-visualization, flutter, firestore

Estimate

M

markdown

Title

Add Toggle for Daily vs. Cumulative Views in Graphs

Description

Introduce a toggle control that allows users to switch between daily usage and cumulative usage views on the usage graph. This helps users analyze patterns over time and understand long-term trends.

Tasks

- [] Add a toggle switch or segmented control to the graph component
- [] Modify data aggregation logic to support both daily and cumulative modes
- [] Ensure chart updates smoothly when toggling
- [] Maintain clear axis labeling and legends for both views

Acceptance Criteria

- [] Toggle is clearly visible and functional
- [] Graph updates accurately on toggle switch
- [] View mode persists during the session
- [] No performance degradation when toggling views

Labels

feature, chart, UX-enhancement, flutter, firestore

Estimate

M

markdown

Title

Implement Export Data to CSV Functionality

Description

Allow users to export their filtered smoking log data to a CSV file. This gives users control over their data and enables further analysis or sharing with healthcare providers.

Tasks

- [] Add export button to the UI near the graph or data table
- [] Generate CSV from currently displayed (filtered and sorted) data
- [] Use appropriate headers: Timestamp, Inhale Length, Session Notes, etc.
- [] Handle file download for web and local storage for mobile
- [] Provide feedback on success/failure of export

Acceptance Criteria

- [] Exported CSV reflects current data view accurately
- [] CSV is downloadable and opens correctly in spreadsheet apps
- [] Export process handles large datasets without crashing
- [] UI clearly indicates export success or failure

Labels

feature, export, flutter, data-management, UX

Estimate

M

markdown

Title

Add Interactive Range Selector for Data Visualization

Description

Implement an interactive range selector that allows users to select predefined or custom time ranges (e.g., Last 7 days, Last 30 days, Custom range) to filter the data shown in graphs and tables. This will improve usability and help users analyze trends over different periods.

Tasks

- [] Design UI component for range selection
- [] Implement logic to filter data based on selected range

- [] Ensure that charts and tables update in real-time
 - [] Add unit tests for date range filtering logic
- ### Acceptance Criteria**
- [] Users can easily select predefined or custom date ranges
 - [] Charts and tables reflect data only within the selected range
 - [] Date range persists when navigating between tabs or refreshing
 - [] Component is responsive and accessible

Labels

feature, UX, frontend, data-visualization, flutter

Estimate

M

markdown

Title

Optimize Chart Performance with Batched Data Rendering

Description

Improve the performance of data-heavy visualizations by implementing batched rendering or virtual scrolling for large datasets. This will ensure smoother chart updates and interaction responsiveness, especially on lower-end devices.

Tasks

- [] Investigate charting libraries' support for batched rendering or virtual scrolling
- [] Modify data pipeline to chunk or sample large datasets for rendering
- [] Add fallback or lazy-load behavior when displaying long time series
- [] Test performance on mobile and low-end devices

Acceptance Criteria

- [] Charts load without lag for large datasets (e.g., year-

```
long logs)
- [ ] User interactions (zoom, pan) remain smooth under heavy
data loads
- [ ] No significant visual discrepancies from full data
rendering
- [ ] Implementation supports future enhancements (e.g., real-
time streaming)
```

Labels

performance, optimization, frontend, flutter, charts

Estimate

L

markdown

Title

Implement Rolling Averages and Trend Lines in Usage Charts

Description

To enhance data insights, add support for rolling averages and trend lines in the usage charts. These features help users identify long-term patterns and anomalies in their smoking behavior.

Tasks

- [] Add toggle options for 7-day and 30-day rolling averages
- [] Integrate linear trend line support using regression
- [] Update chart legends and labels to reflect added lines
- [] Ensure responsiveness and clarity on mobile displays

Acceptance Criteria

- [] Users can toggle between rolling average and trend line overlays
- [] Visualizations clearly distinguish between actual data and overlays
- [] Chart performance is unaffected by added computations
- [] Features are documented in a tooltip or help section

Labels

enhancement, analytics, charts, flutter, data-visualization

Estimate

M

markdown

Title

Enhance Date Range Selector with Custom Input and Presets

Description

Improve the date range selection component in the Smoke-Log app to support both custom date inputs and predefined quick-select options (e.g., "Last 7 days", "This Month", "All Time"). This enhancement allows users to analyze usage over specific and convenient time frames more easily.

Tasks

- [] Redesign the date picker to include a dropdown with quick-select presets
- [] Enable manual date range input using a calendar interface
- [] Update filtering logic to support new range options
- [] Ensure compatibility with existing analytics and charts
- [] Add unit tests for new date selection functionality

Acceptance Criteria

- [] Users can choose from quick-select presets or define a custom date range
- [] Date picker is mobile-friendly and accessible
- [] All charts and usage tables reflect the selected range accurately
- [] No regressions in existing filter behavior

Labels

feature, UI/UX, filters, flutter

Estimate

M

markdown

Title

Implement Cumulative Usage Visualization by Week and Month

Description

Add cumulative usage charts to the Smoke-Log app to help users track how their usage builds over time. These charts should aggregate and display total usage over selectable periods such as weekly or monthly, aiding in long-term habit awareness.

Tasks

- [] Add toggle to switch between raw and cumulative views
- [] Implement weekly cumulative chart view
- [] Implement monthly cumulative chart view
- [] Ensure charts update dynamically with selected date range
- [] Add tests to verify correct data aggregation logic

Acceptance Criteria

- [] Cumulative views display smooth aggregation curves
- [] Weekly and monthly modes reflect total duration over time
- [] Toggle persists user preference during the session
- [] Charts respect current date filters

Labels

feature, analytics, flutter, charting

Estimate

M

markdown

Title

Create Duration Breakdown by Day of Week Chart

Description

Introduce a new visualization that shows average or total smoking session durations segmented by day of the week. This will help users identify patterns in behavior related to specific weekdays (e.g., heavier usage on weekends).

Tasks

- [] Aggregate duration data per weekday (Monday through Sunday)
- [] Display as either bar chart or heatmap for clarity
- [] Allow toggling between average and total duration
- [] Sync with selected date range
- [] Add responsive design for mobile and desktop views

Acceptance Criteria

- [] Users can see a clear breakdown of usage by weekday
- [] The chart adapts based on the selected range (e.g., last 30 days)
- [] Data is accurate and updates with filters
- [] Toggle between average and total functions correctly

Labels

feature, analytics, flutter, charting

Estimate

M

markdown

Title

Add Rolling Average Duration Chart for Trend Visualization

Description

Implement a line chart that displays a rolling average of session durations over time. This will help users visualize

trends in their smoking habits and understand long-term behavioral changes.

Tasks

- [] Calculate rolling average (e.g., 7-day or 30-day window)
- [] Plot the data using a line chart
- [] Allow user to toggle between different rolling window sizes
- [] Sync with existing date range and filters
- [] Ensure mobile responsiveness and clear axis labeling

Acceptance Criteria

- [] Users can view a smooth trendline of session durations
- [] Chart dynamically updates with selected date range
- [] Option to switch between 7-day and 30-day average works properly

Labels

feature, analytics, flutter, charting

Estimate

M

markdown

Title

Implement Duration Distribution Histogram

Description

Create a histogram chart that displays the frequency distribution of session durations. This allows users to understand how their session lengths are spread out over time (e.g., most sessions last 3-4 seconds).

Tasks

- [] Define duration bins (e.g., 0-2s, 2-4s, 4-6s, etc.)
- [] Build histogram UI using a bar chart
- [] Integrate with filtering options (date range, time of day)

- [] Display count labels above each bar for clarity
- ### Acceptance Criteria**
- [] Histogram accurately reflects the number of sessions per duration range
 - [] Visual styling matches the existing AshTrail design
 - [] Filters and date range selection update the chart correctly

Labels
feature, visualization, flutter, charting

Estimate

M

markdown

Title
Implement Session Count by Day Heatmap

Description
Create a heatmap visualization that shows the number of sessions per day on a calendar grid. This allows users to visually identify streaks, high-usage days, and patterns over weeks or months.

- ### Tasks**
- [] Create a calendar-based heatmap component
 - [] Map session count to color intensity
 - [] Integrate with date range selector
 - [] Add tooltip to display exact session count per day
 - [] Ensure accessibility and responsiveness

- ### Acceptance Criteria**
- [] Calendar heatmap displays correct session counts per day
 - [] Color scale clearly represents frequency
 - [] Interacts smoothly with filters and time range selection
 - [] Tooltips work on hover/tap

Labels

feature, visualization, flutter, charting

Estimate

M

markdown

Title

Add Monthly Usage Summary Widget

Description

Implement a summary widget that provides users with key statistics for the selected month. This should include total sessions, average session length, longest session, and number of smoke-free days.

Tasks

- [] Design and position the summary widget in the UI
- [] Fetch and calculate metrics based on current filter range
- [] Style the widget to match the app's aesthetic
- [] Add unit tests for metric calculations

Acceptance Criteria

- [] Widget displays correct metrics for selected month
- [] Updates dynamically with range/filter changes
- [] Visually clean and easy to interpret

Labels

feature, dashboard, summary, flutter

Estimate

S

markdown

Title

Enable Chart Type Switching (Line/Bar/Cumulative)

Description

Allow users to toggle between different chart types (line chart, bar chart, and cumulative usage chart) for enhanced data analysis and visualization. This feature should integrate with the existing range and filter settings.

Tasks

- [] Add UI controls for switching between chart types
- [] Implement chart rendering logic for each chart type
- [] Ensure data is formatted appropriately for each visualization
- [] Update documentation or tooltips to explain chart types

Acceptance Criteria

- [] Users can toggle chart types without reloading the page
- [] Charts reflect selected range and filters accurately
- [] Visual transitions are smooth and intuitive

Labels

feature, chart, UI, flutter, enhancement

Estimate

M

markdown

Title

Optimize Log Filtering Performance for Large Datasets

Description

Improve the performance of the log filtering functionality when handling large volumes of data. This includes optimizing in-memory filtering logic and reducing UI lag when applying filters or adjusting the range.

Tasks

- [] Profile filtering logic to identify performance bottlenecks
- [] Refactor filtering mechanism for efficiency
- [] Debounce filter input changes to prevent excessive re-rendering
- [] Add test cases with large synthetic datasets

Acceptance Criteria

- [] Filtering updates occur within 200ms for datasets up to 10,000 entries
- [] No significant UI lag when adjusting filters or range
- [] Performance remains stable across chart types

Labels

performance, optimization, flutter, enhancement

Estimate

L

markdown

Title

Implement Data Export Functionality (CSV/JSON)

Description

Allow users to export their smoking log data in CSV and JSON formats. This will support users who want to analyze their data externally or keep backups.

Tasks

- [] Add export buttons to UI (CSV and JSON)
- [] Implement export logic for filtered and full datasets
- [] Generate downloadable files with appropriate naming
- [] Ensure compatibility with common spreadsheet and data tools
- [] Display confirmation after successful export

Acceptance Criteria

- [] Users can download a file in CSV or JSON format
- [] File contains either the full dataset or the currently filtered view
- [] Export includes timestamps, inhalation lengths, and any relevant tags or metadata

Labels

feature, flutter, data, enhancement

Estimate

M

markdown

Title

Integrate Local Timezone Awareness for Logs and Charts

Description

Ensure that all timestamps displayed in the app (logs, charts, exports) respect the user's local timezone to improve clarity and accuracy in tracking smoking habits.

Tasks

- [] Detect user's local timezone on app launch
- [] Adjust all timestamp displays accordingly
- [] Ensure time-based filtering (e.g., daily, weekly ranges) respects local time
- [] Add a setting to manually override timezone if needed
- [] Validate time consistency in exports

Acceptance Criteria

- [] All logs and visualizations reflect local time by default
- [] Timezone can be manually changed in settings
- [] No discrepancies between displayed time and exported data

Labels

feature, timezone, flutter, usability

Estimate

M

markdown

Title

Implement Rolling Averages and Streak Tracking for Usage Charts

Description

Enhance user insights by adding rolling averages (e.g., 7-day, 30-day) and visual streak indicators to the inhalation charts. This will help users monitor trends and stay motivated.

Tasks

- [] Add toggle for 7-day and 30-day rolling averages on chart
- [] Display streaks (consecutive days with at least one log) visually
- [] Ensure streak resets after a full day of no logs
- [] Include rolling data in exports
- [] Write unit tests for streak logic

Acceptance Criteria

- [] Users can view charts with rolling average overlays
- [] Streaks are clearly visualized and accurate
- [] All calculations update in real-time with new data
- [] Data export includes rolling average columns

Labels

feature, charts, flutter, engagement, tracking

Estimate

L

markdown

Title

Add Cumulative Usage Heatmap Calendar to Dashboard

Description

Implement a heatmap-style calendar view on the dashboard to visualize inhalation frequency per day over time. This provides an intuitive overview of usage trends, streaks, and behavior changes.

Tasks

- [] Create heatmap calendar component
- [] Integrate with Firestore log data
- [] Color-code days based on inhalation count
- [] Add tooltip or modal showing count per day
- [] Ensure responsive design for mobile and desktop
- [] Unit test data aggregation logic

Acceptance Criteria

- [] Calendar shows correct usage data for each day
- [] Colors reflect intensity of usage (e.g., light to dark)
- [] Users can hover/tap to see exact counts
- [] Fully responsive and performant on all supported devices

Labels

feature, visualization, dashboard, flutter, analytics

Estimate

L

markdown

Title

Implement Smoke-Free Streak Tracker with Motivational Feedback

Description

Add a feature that tracks consecutive smoke-free days and displays the user's longest and current streak. Include visual encouragement and motivational messages to reinforce progress

and support habit reduction.

Tasks

- [] Develop backend logic to calculate streaks from log data
- [] Design a UI component for displaying current and longest streaks
- [] Add animated encouragement (e.g., confetti or badge) on streak milestones
- [] Implement logic to reset streak if usage is logged
- [] Add contextual motivational messages based on streak length

Acceptance Criteria

- [] Dashboard displays accurate current and longest streaks
- [] User is notified with motivational feedback at key milestones (e.g., 3, 7, 30 days)
- [] Streak resets appropriately when usage is detected
- [] Feature is responsive and visually cohesive with the rest of the UI

Labels

feature, streaks, motivation, dashboard, flutter, analytics

Estimate

M

markdown

Title

Implement Data Range Presets and Quick Filters

Description

Add support for quick data range selection in the analytics dashboard (e.g., Today, Last 7 Days, Last 30 Days, All Time) and create reusable filtering logic to streamline user interaction.

Tasks

- [] Design and add UI controls for common date range presets

- [] Refactor existing filtering logic to use a shared filter utility
- [] Ensure charts and table views update based on selected range
- [] Add visual indication of active filter/preset
- [] Test compatibility with custom date range selection

Acceptance Criteria

- [] Users can select from predefined date ranges
- [] Selecting a preset updates all analytics views consistently
- [] Filtering logic is modular and reusable across components
- [] Active filter is clearly visible in the UI

Labels

enhancement, filters, UI/UX, flutter, dashboard, analytics

Estimate

M

markdown

Title

Integrate Rolling Averages and Trendlines for Usage Data

Description

Enhance the visual analysis capabilities of the AshTrail dashboard by adding rolling averages and trendline overlays to relevant charts. This will help users identify long-term usage patterns and anomalies more effectively.

Tasks

- [] Implement calculation of rolling averages (e.g., 7-day, 30-day)
- [] Add toggles to enable/disable trendlines on charts
- [] Update chart rendering logic to overlay trendlines with distinct styling
- [] Ensure performance remains acceptable for large datasets

- [] Document trendline behavior in the help tooltip or documentation section

Acceptance Criteria

- [] Users can toggle trendlines and rolling averages on applicable charts
- [] Rolling averages reflect accurate calculations over selected windows
- [] Charts maintain clarity and performance when overlays are active
- [] Documentation or tooltips explain trendline purpose and usage

Labels

feature, charts, data-visualization, flutter, analytics

Estimate

L

markdown

Title

Optimize Data Filtering and Sorting Logic for Range Selector Integration

Description

Refactor the data processing logic so that all filtering and sorting are handled in one centralized location—alongside the range selector component. This will simplify future enhancements like rolling usage analysis and improve code maintainability.

Tasks

- [] Move all filtering and sorting logic to the range selector's controller or provider
- [] Ensure correct application of date range filters and sort order
- [] Validate that charts and tables continue to update correctly with filtered data

- [] Write unit tests to cover new data flow
- [] Update documentation or inline comments for clarity

Acceptance Criteria

- [] Centralized filtering logic without duplication
- [] Range selector updates reflect immediately on charts and data tables
- [] Unit tests pass and verify key filtering behaviors
- [] Performance is maintained or improved

Labels

refactor, data-handling, flutter, enhancement

Estimate

M

markdown

Title

Add Cumulative Usage and Inhalation Count to Chart Analysis

Description

Enhance the chart functionality by including visualizations for cumulative usage over time and total inhalation counts. This will give users a clearer picture of their usage patterns and long-term trends.

Tasks

- [] Add cumulative sum of inhalation lengths to chart data processing
- [] Add total count of inhalations to chart data
- [] Update UI to toggle between raw, cumulative, and count modes
- [] Ensure accurate x-axis alignment for all data modes
- [] Write tests to confirm correctness of new modes
- [] Document chart mode switching behavior

Acceptance Criteria

- [] Users can switch between raw, cumulative, and count

views

- [] Cumulative usage line accurately reflects inhalation totals over time
- [] Count chart accurately tallies inhalations by timestamp
- [] No UI or performance regression

Labels

feature, data-visualization, flutter, enhancement

Estimate

M

markdown

Title

Add Toggle for Timestamp Display Format (Relative vs Absolute)

Description

Introduce a toggle in the UI that allows users to switch between relative timestamps (e.g., "2 hours ago") and absolute timestamps (e.g., "2025-07-24 13:45"). This improves readability and user preference accommodation.

Tasks

- [] Add toggle switch to chart settings panel
- [] Implement timestamp format conversion logic
- [] Ensure both table and chart tooltip timestamps update accordingly
- [] Update user preferences to persist the selected format
- [] Write tests for format switching functionality

Acceptance Criteria

- [] Toggle is present and functional in the UI
- [] Timestamps switch between relative and absolute formats instantly
- [] Table and chart views reflect the selected format consistently
- [] Selected format persists across sessions

Labels

feature, UI, UX, flutter, enhancement

Estimate

S

markdown

Title

Implement Cumulative Usage Chart with Dynamic Time Range Filtering

Description

Create a line chart to visualize cumulative smoking usage over time. This chart should support dynamic time range selection (e.g., past 7 days, past 30 days, all-time), allowing users to analyze trends in their usage habits more effectively.

Tasks

- [] Add cumulative calculation logic based on timestamp and inhale length
- [] Create a responsive line chart UI for cumulative data
- [] Integrate time range selector with chart updates
- [] Ensure performance with large datasets
- [] Write tests for cumulative calculations and time filtering

Acceptance Criteria

- [] Users can view cumulative usage in the selected time range
- [] Time range selector updates the chart in real-time
- [] Accurate cumulative values displayed on hover/tooltips
- [] Chart adjusts responsively to screen size

Labels

feature, chart, data-visualization, flutter, enhancement

Estimate

M

markdown

Title

Create Filter & Sort Controls for Log Table Display

Description

Implement controls to filter and sort log entries in the data table below the usage charts. This will help users better explore their data, such as viewing only high-usage entries or sorting by inhale length or timestamp.

Tasks

- [] Add dropdown or segmented control for filtering logs (e.g., by inhale length threshold)
- [] Implement sorting options (timestamp ascending/descending, inhale length)
- [] Ensure filters and sorting persist across chart interactions
- [] Connect controls to the existing Firestore/Isar data logic

Acceptance Criteria

- [] Users can filter logs by specific inhale criteria
- [] Sorting options update the table in real time
- [] Controls do not interfere with chart data
- [] Layout is responsive and accessible

Labels

enhancement, flutter, firestore, isar, data-visualization

Estimate

M

markdown

Title

Implement Export Functionality for Log Data

Description

Add an option for users to export their smoke log data for personal analysis or backup. This will support formats such as CSV and JSON, making it easier to share or process data externally.

Tasks

- [] Create an export button on the log data view
- [] Support CSV and JSON formats for export
- [] Retrieve and structure data appropriately from Firestore/Isar
- [] Prompt file save dialog with proper file naming and content

Acceptance Criteria

- [] Users can download their logs in CSV or JSON format
- [] Exported data matches the fields shown in the app
- [] Feature works in both dev and production builds

Labels

feature, data-export, flutter, firestore, isar

Estimate

M

markdown

Title

Refactor Chart and Table Data Logic into Unified Analysis Layer

Description

To simplify adding new analysis features and improve maintainability, consolidate the filtering, sorting, and range selection logic for the charts and data table into a shared

analysis layer. This layer will act as a single source of truth for all components relying on log data presentation.

Tasks

- [] Create a shared utility or provider for range selection and data filtering
- [] Refactor chart components to use the unified analysis logic
- [] Refactor data table to pull from the same filtered data source
- [] Ensure all chart and table updates remain reactive to user input

Acceptance Criteria

- [] Chart and table display synchronized data
- [] Data filters and date range changes propagate correctly across components
- [] New analysis features can hook into the shared logic easily

Labels

refactor, enhancement, flutter, data-logic

Estimate

L

markdown

Title

Implement Rolling Average and Peak Usage Chart Option

Description

Enhance the analytics capabilities of AshTrail by introducing a new chart type that displays a rolling average alongside peak inhalation usage within the selected time range. This will help users better understand usage trends and spikes over time.

Tasks

- [] Add option to toggle between raw data and rolling average view
- [] Implement logic for calculating rolling average (e.g., 7-day, 30-day)
- [] Overlay peak usage markers or highlight on chart
- [] Ensure data respects current date range and filters

Acceptance Criteria

- [] Users can switch between raw and rolling average views
- [] Peak usage points are clearly visible or highlighted
- [] All visualizations update in real-time with range selection

Labels

feature, enhancement, chart, flutter, data-visualization

Estimate

M

markdown

Title

Add Data Export Functionality (CSV and JSON Formats)

Description

Enable users to export their inhalation log data from AshTrail in both CSV and JSON formats. This feature supports data ownership and allows users to analyze their data using external tools.

Tasks

- [] Add export button to the analytics dashboard
- [] Implement export logic for CSV and JSON formats
- [] Ensure exported data respects current filters and date range
- [] Include headers and metadata (timestamp, usage length, etc.)

Acceptance Criteria

- [] Users can download their data in both CSV and JSON
- [] File includes only visible data from the selected date range
- [] Export works on both mobile and web platforms

Labels

feature, enhancement, data-export, flutter, smoke-log

Estimate

M

markdown

Title

Implement Rolling Average Display in Usage Charts

Description

Introduce an optional rolling average overlay on usage charts in the AshTrail app. This feature will help users visualize trends and smooth out short-term fluctuations in their inhalation behavior.

Tasks

- [] Add UI toggle to enable/disable rolling average display
- [] Implement rolling average calculation (configurable window size)
- [] Update chart rendering to include the overlay line
- [] Ensure mobile and web compatibility

Acceptance Criteria

- [] Rolling average line appears correctly on the chart when enabled
- [] Users can configure the window size (e.g., 3-day, 7-day)
- [] Performance remains smooth with the overlay enabled
- [] Feature integrates cleanly with existing filter and date range logic

Labels

feature, enhancement, data-visualization, flutter, smoke-log

Estimate

M

markdown

Title

Sync Isar Offline Logs with Firestore on Reconnection

Description

Implement logic to detect when the app comes back online and automatically sync any unsynced inhalation logs stored locally in Isar with Firestore. This ensures users' data is backed up even if they were offline when the entry was recorded.

Tasks

- [] Detect network reconnection using *connectivity_plus or equivalent*
- [] Query *unsynced entries from Isar Local database*
- [] Push *unsynced entries to Firestore with proper timestamps*
- [] Mark *entries as synced after successful upload*
- [] Display a sync notification or status update in UI (*optional*)

Acceptance Criteria

- [] *Unsynced logs are reliably detected and uploaded on reconnection*
- [] *Data remains consistent between Isar and Firestore*
- [] *No duplicate entries are uploaded*
- [] *Sync handles large backlogs gracefully without crashing the app*

Labels

feature, offline-support, flutter, firestore, isar, smoke-log

Estimate

M

markdown

Title

Implement User Feedback Prompt After Log Submission

Description

Add a user feedback mechanism that appears after submitting a smoke log. This will help gather insights about the user's emotional state, cravings, or reasons behind their action, enriching the dataset for future analytics.

Tasks

- [] Design a lightweight modal or bottom sheet that prompts for optional feedback
- [] Include options like mood, craving level, trigger (e.g., stress, habit), and a notes field
- [] Store feedback in Firestore alongside the corresponding log entry
- [] Allow user to skip without penalty or delay

Acceptance Criteria

- [] Feedback modal appears after each new log submission
- [] User input is correctly associated with the log entry in Firestore
- [] UI is non-intrusive and respects user choice to skip
- [] Works seamlessly offline and syncs later when online

Labels

feature, UX, flutter, firestore, smoke-log

Estimate

M

markdown

Title

Add Summary Dashboard to Display Daily and Weekly Smoking Patterns

Description

Implement a dashboard view that summarizes the user's smoke logs, including daily and weekly trends. This will provide users with immediate visual feedback and insights into their habits, encouraging reflection and progress tracking.

Tasks

- [] Create a new Dashboard screen accessible from the main navigation
- [] Display a daily count chart (e.g., bar chart)
- [] Display a weekly usage trend chart (e.g., line or bar chart)
- [] Include cumulative session time and average inhalation duration
- [] Use Firestore data to populate the charts
- [] Ensure proper time zone handling for accurate daily/weekly boundaries

Acceptance Criteria

- [] Dashboard loads successfully with accurate data visualizations
- [] Data is updated in near real-time when new logs are added
- [] Charts adjust properly to different screen sizes
- [] Performance is smooth with data from extended use

Labels

feature, UI, flutter, firestore, analytics

Estimate

L

markdown

Title

Implement Inhalation Pattern Detail View for Individual Sessions

Description

Develop a detail view for individual smoking sessions, allowing users to see a breakdown of inhalation lengths and timing within each session. This view will help users recognize patterns in their smoking behavior on a granular level.

Tasks

- [] Add navigation from log entry to detailed session view
- [] Display a chart (e.g., line or scatter) of inhalation durations over session time
- [] Show summary stats: total duration, number of puffs, average inhalation length
- [] Format timestamp and session data for easy readability
- [] Optimize for quick loading and responsiveness

Acceptance Criteria

- [] User can tap a log entry and view detailed session data
- [] All inhalations in the session are visualized correctly
- [] Page loads within 1 second on average mobile device
- [] Detail view displays accurate data for any selected session

Labels

feature, UI, data-visualization, flutter, firestore

Estimate

M

markdown

Title

Implement Rolling Usage Graph with Adjustable Time Windows

Description

Introduce a rolling usage graph to visualize trends in smoking behavior over customizable time intervals (e.g., 1-day, 7-day, 30-day rolling averages). This will give users insights into their consumption patterns over time and help identify

increases or reductions.

Tasks

- [] Add toggle or selector for rolling average window (1d, 7d, 30d)
- [] Compute rolling average of inhalation count and total time
- [] Display rolling average graph alongside raw usage chart
- [] Update range selector logic to support rolling data computation
- [] Ensure compatibility with filtering and offline data modes

Acceptance Criteria

- [] Users can toggle rolling average intervals and view changes in the graph
- [] Graph updates dynamically and correctly reflects selected time window
- [] Performance is acceptable for one year of data
- [] Works on both development (static) and production (Firestore) modes

Labels

feature, enhancement, charting, flutter, firestore, time-series

Estimate

M

markdown

Title

Refactor Data Submission to Support Offline-First Workflow with Isar Integration

Description

Transition the data submission logic to an offline-first design by utilizing Isar as the primary write target. Firebase writes should occur in the background or upon reconnection.

This ensures data is reliably captured even with unstable internet and aligns with long-term offline support goals.

Tasks

- [] Refactor `submitData` function to write to Isar DB instead of directly to Firestore
- [] Implement background sync task to push Isar-stored entries to Firestore when online
- [] Handle error states and retry logic for failed Firestore submissions
- [] Ensure UI feedback (e.g., Snackbar or icon) reflects offline/unsynced state
- [] Provide manual sync option in app settings or UI

Acceptance Criteria

- [] Data is stored reliably even when offline
- [] Firestore is updated upon reconnection without user action
- [] User is notified of unsynced or failed states
- [] Manual sync triggers a background push to Firestore

Labels

offline, firestore, isar, data-integrity, enhancement

Estimate

L

markdown

Title

Implement User Settings Page with Theme Toggle and Log Export Options

Description

Add a new User Settings page to the app that allows users to toggle between light and dark mode, as well as export their log data. This enhances personalization and data accessibility for the user.

Tasks

- [] Create new Settings screen in the app
- [] Add a toggle switch for light/dark mode using `ThemeMode` logic
- [] Add an "Export Logs" button that downloads log data as JSON or CSV
- [] Ensure export respects current filters/date ranges, or allow full export
- [] Update navigation to include link to Settings page

Acceptance Criteria

- [] Users can access Settings from the main navigation
- [] Users can toggle between light and dark themes
- [] Users can export their inhalation logs in a readable format

Labels

settings, theming, data-export, enhancement

Estimate

M

markdown

Title

Add Rolling Average Analysis to Inhalation Charts

Description

Enhance the inhalation charts by adding a rolling average line to provide users with a clearer understanding of their trends over time. This helps users visualize short-term fluctuations while smoothing out longer-term trends.

Tasks

- [] Implement rolling average calculation on the filtered dataset
- [] Add UI toggle to show/hide the rolling average line
- [] Allow user to customize the window size (e.g., 3, 5, 7 days)

- [] Ensure calculations update dynamically with range selector and filters

Acceptance Criteria

- [] Rolling average line is accurately plotted on the chart
- [] User can toggle visibility and change the window size
- [] Works with date range selector and any applied filters

Labels

feature, data-visualization, enhancement

Estimate

M

markdown

Title

Implement Session Summary View with Key Metrics

Description

Introduce a session summary panel that appears after data submission or when a session is selected from the table. This view should provide users with high-level insights such as total inhalations, average duration, peak time, and cumulative time for the session.

Tasks

- [] Design UI component for session summary (card or modal)
- [] Display total inhalations, average duration, cumulative time, peak inhalation period
- [] Integrate with session selection logic in the table
- [] Ensure responsiveness and accessibility

Acceptance Criteria

- [] Session summary updates correctly based on selected session
- [] Key metrics are displayed accurately
- [] UI integrates smoothly with existing design

Labels

feature, UX, enhancement

Estimate

M

markdown

Title

Implement Rolling Usage Charts with Dynamic Time Windows

Description

Create a new chart component that visualizes rolling usage over time (e.g., 1-day, 7-day, 30-day). This feature allows users to analyze trends and habits across custom time windows, giving more flexibility and insight into usage behavior.

Tasks

- [] Build chart UI with selectable time windows (e.g., dropdown or button group)
- [] Calculate rolling sums or averages based on selected time window
- [] Update chart reactively based on user input
- [] Ensure integration with existing filtering and range logic

Acceptance Criteria

- [] Chart updates dynamically with selected rolling window
- [] Calculations are accurate and performance is optimized
- [] UI is consistent with existing dashboard design

Labels

feature, visualization, enhancement

Estimate

L

markdown

Title

Enable Local Caching for Offline Access

Description

Implement local caching of log data and chart views to allow offline access and improve performance for returning users. This feature ensures continuity of experience even when network connectivity is temporarily unavailable.

Tasks

- [] Store recently viewed logs and chart data in local storage or IndexedDB
- [] Provide UI indicators for offline mode
- [] Automatically sync and update data when the user comes back online
- [] Add fallback behavior for data submission in offline mode

Acceptance Criteria

- [] User can view previously accessed data without internet
- [] No app crashes when network is lost
- [] Sync process is smooth and accurate when regaining connectivity

Labels

feature, offline-support, enhancement

Estimate

L

markdown

Title

Add Session Overview Screen with Daily/Weekly Summaries

Description

Introduce a new screen that gives users a summarized view of their smoking sessions across days and weeks. This overview

should highlight metrics such as total sessions, average duration, cumulative inhalation time, and notable changes compared to the previous period.

Tasks

- [] Design UI/UX for session summary screen
- [] Calculate daily and weekly stats from Firestore data
- [] Visualize data using bar charts and simple infographics
- [] Implement date range selector for summaries
- [] Ensure mobile responsiveness and theme consistency

Acceptance Criteria

- [] Users can toggle between daily and weekly views
- [] Stats display correctly and reflect real usage data
- [] Data updates automatically when new logs are submitted

Labels

feature, UI, statistics, enhancement

Estimate

M

markdown

Title

Implement Toggleable Chart Types for Data Visualization

Description

Allow users to switch between different types of charts (line, bar, pie, etc.) to view their smoking data in the format that best suits their preferences. This flexibility will enhance user engagement and provide multiple perspectives on the same data.

Tasks

- [] Add a dropdown or segmented control to select chart type
- [] Integrate additional chart components into the existing charting logic
- [] Maintain consistent color schemes and legends across

```
chart types
- [ ] Ensure performance remains optimal with different chart renderings

### Acceptance Criteria
- [ ] User can select between at least three chart types
- [ ] Selected chart type renders accurately with the correct data
- [ ] Transitions between chart types are smooth and visually consistent

### Labels
feature, UI, chart, enhancement

### Estimate
M
```

markdown

```
### Title
Add "Session Tags" Feature for Better Log Categorization

### Description
Introduce a tagging system allowing users to label individual smoking sessions with customizable tags such as "Stress", "Celebration", "Social", etc. Tags will help users better categorize and reflect on their usage patterns.

### Tasks
- [ ] Create UI for adding/removing tags during or after logging a session
- [ ] Store tags in Firestore alongside session data
- [ ] Update visualizations to support filtering by tags
- [ ] Implement tag suggestions based on frequently used tags

### Acceptance Criteria
- [ ] Users can add multiple tags per session
- [ ] Tags persist and are editable post-session
- [ ] Users can filter sessions by tag in the log view and
```

charts

Labels

feature, UX, data-model, enhancement

Estimate

M

markdown

Title

Implement Mood Tracking Integration for Session Insights

Description

Enable users to log their mood before and after each smoking session to gather insights into emotional patterns and usage motivation. This will help provide users with more meaningful data correlations.

Tasks

- [] Add pre- and post-session mood input UI (emoji scale or mood tags)
- [] Store mood data in Firestore alongside session data
- [] Update session detail view to display mood changes
- [] Add mood filter and visualization support in insights section

Acceptance Criteria

- [] Users can log mood before and after each session
- [] Mood data is stored and retrievable for each session
- [] Insights and trends can be viewed based on mood states

Labels

feature, enhancement, data-analytics, UX

Estimate

M

markdown

Title

Add Rolling Average Visualization to Smoke Usage Chart

Description

Enhance the main usage chart by allowing users to toggle a rolling average overlay. This will help users identify long-term trends and fluctuations in their smoking behavior.

Tasks

- [] Add UI toggle to enable/disable rolling average view
- [] Implement rolling average computation (e.g., 7-day or 30-day)
- [] Display rolling average line on top of existing usage chart
- [] Ensure performance with large datasets

Acceptance Criteria

- [] Rolling average can be toggled on/off
- [] Displayed average line reflects correct time span
- [] Chart updates smoothly without lag

Labels

feature, enhancement, charting, UX

Estimate

M

markdown

Title

Implement Smoke Session Comparison Feature

Description

Add a feature allowing users to compare two smoke sessions or time periods side by side. This will help users analyze progress or behavioral changes over time.

Tasks

- [] Add UI component to select two time ranges or specific sessions
- [] Display key metrics (total usage, average inhalation length, frequency)
- [] Highlight differences and percent change between sessions
- [] Ensure compatibility with filtered data and rolling average views

Acceptance Criteria

- [] Users can easily select and compare two sessions or periods
- [] Comparison view shows clear, useful insights
- [] Works seamlessly with other dashboard filters

Labels

feature, comparison, analytics, UX

Estimate

M

markdown

Title

Add Session Insights Summary Panel

Description

Create a summary panel that appears after each logged session, providing immediate feedback and insights. This feature will enhance user engagement by helping them reflect on their smoking behavior right after logging.

Tasks

- [] Design and implement a UI panel that shows after data submission
- [] Display total inhalation time, session duration, and comparison to average usage
- [] Add motivational or reflective message based on recent

trends

- [] Ensure accessibility and responsiveness on all devices

Acceptance Criteria

- [] Users see the insights panel immediately after submitting a session
- [] Metrics are accurate and relevant to the submitted data
- [] Messages vary based on user trends and provide value

Labels

feature, UX, insights, summary

Estimate

M

markdown

Title

Implement Data Filtering Controls for Chart Range Selector

Description

Integrate filtering and sorting logic with the chart range selector to streamline the analysis of inhalation data. This enhancement will centralize control logic, making it easier to apply future analytical tools such as rolling averages or custom range filters.

Tasks

- [] Refactor chart component to co-locate range selection and data filtering logic
- [] Ensure compatibility with existing cumulative and session-based charts
- [] Add toggle for displaying sorted session table based on selected range
- [] Validate performance on large datasets

Acceptance Criteria

- [] Filtering and sorting logic is unified under the chart range selector

- [] Charts and tables update dynamically based on range selection
- [] No regression in chart rendering or data analysis features

Labels

refactor, charting, data-visualization, enhancement

Estimate

L

markdown

Title

Implement Rolling Average Analysis for Inhalation Charts

Description

Introduce a rolling average option for time series inhalation charts to smooth out short-term fluctuations and highlight longer-term trends. This will help users identify patterns in their usage more effectively.

Tasks

- [] Add UI toggle to enable/disable rolling average view
- [] Implement rolling average logic in chart data transformation
- [] Allow user to select window size (e.g., 3, 7, 14 days)
- [] Ensure compatibility with date range filtering and other analysis modes

Acceptance Criteria

- [] Rolling average can be toggled on/off for inhalation charts
- [] Users can select the window size
- [] Chart correctly displays smoothed data
- [] Works with filtered date ranges without error

Labels

feature, data-visualization, enhancement, analytics

Estimate

M

markdown

Title

Design and Implement Weekly Summary View for Usage Insights

Description

Introduce a new Weekly Summary feature that aggregates and visualizes smoking activity by week. This will allow users to observe trends and patterns over a broader time frame than the current daily view.

Tasks

- [] Create backend logic to compute weekly aggregates (total length, number of sessions)
- [] Update Firestore query to support weekly grouping
- [] Build new UI component for the weekly summary chart
- [] Include week start and end date labels, totals, and average per day
- [] Ensure data accuracy and responsive layout for mobile/desktop

Acceptance Criteria

- [] Weekly summary view is available and toggleable
- [] Displays accurate weekly data from Firestore
- [] Includes visual indicators of trends (e.g. increase/decrease)
- [] Mobile-friendly and consistent with app theme

Labels

feature, analytics, visualization, enhancement

Estimate

L

markdown

Title

Add Custom Time Range Filter for Inhalation Charts

Description

Implement a custom date/time range selector allowing users to analyze inhalation data over arbitrary time periods. This offers more granular control compared to fixed range options like day, week, or month.

Tasks

- [] Design a UI component for selecting a start and end date/time
- [] Update the data querying logic to use selected range
- [] Reflect the selected range in the chart and table views
- [] Ensure the range selector updates interactively and re-renders data properly
- [] Add validation to prevent invalid ranges (e.g., end before start)

Acceptance Criteria

- [] User can select a custom date/time range
- [] Charts and tables update according to the selected range
- [] Component works seamlessly on mobile and desktop
- [] No crashes or data inconsistencies with edge-case ranges

Labels

feature, enhancement, filtering, UX

Estimate

M

markdown

Title

Implement Rolling Usage Averages for Inhalation Data

Description

Add support for displaying rolling usage averages (e.g., 7-day, 30-day) in inhalation charts. This will help users observe trends over time and smooth out day-to-day fluctuations.

Tasks

- [] Add UI controls to select rolling average period (e.g., dropdown: None, 7-day, 30-day)
- [] Modify chart logic to calculate and display rolling averages
- [] Visually distinguish average line (e.g., dashed line or lighter color)
- [] Update tooltips and legends to reflect rolling average data
- [] Ensure the feature works with both custom and fixed time ranges

Acceptance Criteria

- [] Users can toggle rolling average display
- [] Rolling average is correctly calculated and shown on charts
- [] Display adapts dynamically to selected time range
- [] No performance issues with large datasets

Labels

feature, enhancement, charts, UX

Estimate

M

markdown

Title

Enhance Chart Export Capability for Inhalation Insights

Description

Enable users to export inhalation data charts (e.g., duration, frequency) as images or PDFs for personal records or to share with healthcare providers.

Tasks

- [] Add export button to each chart section
- [] Support export as PNG and PDF
- [] Include timestamp and selected range on export
- [] Ensure exported charts retain styling and labels
- [] Test across devices for compatibility

Acceptance Criteria

- [] Users can export each chart type as PNG or PDF
- [] Export includes visible data and context info (e.g., date range)
- [] Charts look consistent in exports and on-screen
- [] No layout or rendering issues in export files

Labels

feature, enhancement, export, charts

Estimate

S

markdown

Title

Implement "Longest Streak" Feature on Dashboard

Description

Display the user's longest streak of consecutive smoke-free days directly on the dashboard to encourage progress tracking and motivate continued behavior change.

Tasks

- [] Calculate longest consecutive streak of 0 inhalation days
- [] Display streak count prominently on the dashboard
- [] Add tooltip or info icon explaining the streak logic
- [] Update streak if a longer one is achieved
- [] Test edge cases (e.g., missing data, streak resets)

Acceptance Criteria

- [] Dashboard shows correct longest streak for all users
- [] Tooltip explains how streak is calculated
- [] Streak resets properly after smoking activity
- [] Handles time zones and missing data gracefully

Labels

feature, dashboard, enhancement, motivation

Estimate

M

markdown

Title

Create Weekly Summary Card Component

Description

Develop a reusable Weekly Summary Card component to display weekly insights, including total inhales, average inhales per day, and comparison with previous weeks. This will help users track behavioral trends over time.

Tasks

- [] Design card layout with totals, averages, and comparison metrics
- [] Fetch and process weekly data for the current and previous weeks
- [] Apply color-coded indicators (e.g., green for improvement, red for increase)
- [] Ensure responsiveness across devices
- [] Write unit tests for the card component

Acceptance Criteria

- [] Weekly Summary Card is visually integrated into the dashboard
- [] Data accurately reflects the selected week's metrics
- [] Performance does not degrade on large datasets
- [] Tests confirm correct rendering and calculation logic

Labels

feature, dashboard, component, UX, visualization

Estimate

M

markdown

Title

Implement Daily Heatmap Visualization

Description

Add a heatmap visualization to represent daily usage intensity over time. This will help users identify patterns in behavior, such as increased usage on specific days or gradual trends over weeks or months.

Tasks

- [] Design heatmap layout using a calendar-style grid
- [] Aggregate data into daily total inhale counts
- [] Normalize values for heatmap color scaling
- [] Integrate into the main dashboard view
- [] Ensure performance and responsiveness
- [] Add toggle or filter for time range (e.g., last 30, 90, 365 days)

Acceptance Criteria

- [] Heatmap renders correctly with appropriate scaling and coloring
- [] User can hover or tap on a day to view exact inhale count
- [] Time range filter modifies the heatmap accordingly
- [] Performance remains smooth with year-long data

Labels

feature, visualization, dashboard, heatmap, UX

Estimate

M

markdown

Title

Add Rolling Average Chart for Usage Trends

Description

Introduce a rolling average chart to show smoothed usage trends over time. This visualization will help users understand long-term behavior changes by reducing short-term noise.

Tasks

- [] Calculate rolling average of inhale lengths over user-defined windows (e.g., 7-day, 30-day)
- [] Add chart to the analytics dashboard
- [] Allow users to toggle window sizes dynamically
- [] Sync chart with time range selector
- [] Optimize for mobile and desktop responsiveness

Acceptance Criteria

- [] Rolling average chart accurately reflects selected window
- [] Users can interactively switch between different window sizes
- [] Chart updates in real-time with data range changes
- [] No significant performance drop with large datasets

Labels

feature, visualization, trend-analysis, UX

Estimate

M

markdown

Title

Implement Interactive Time Range Selector for Analytics Dashboard

Description

Add a dynamic time range selector to the analytics dashboard to allow users to view insights across specific periods (e.g., last 7 days, last 30 days, custom range). This control will enable more tailored insights and improve trend identification.

Tasks

- [] Create UI component for selecting preset and custom date ranges
- [] Filter all analytics (charts, tables) based on selected time range
- [] Maintain selection state across sessions (e.g., local storage)
- [] Ensure accurate performance on large datasets
- [] Add tests for date filtering logic

Acceptance Criteria

- [] Time range selector updates all visuals and tables in real-time
- [] Users can choose common presets and define custom ranges
- [] Charts and tables correctly reflect selected time window
- [] Component functions responsively on mobile and desktop

Labels

feature, UI, analytics, date-filtering

Estimate

M

markdown

Title

Add Rolling Usage Line Chart with Smoothing Options

Description

Enhance the analytics dashboard by adding a new line chart that visualizes rolling usage over time. Users should be able to select the window size (e.g., 3-day, 7-day) and apply a smoothing method to better observe usage trends and patterns.

Tasks

- [] Design and implement a new rolling usage chart
- [] Add control for selecting smoothing window size
- [] Implement smoothing algorithms (e.g., moving average)
- [] Sync chart with selected time range and filters
- [] Write unit tests for smoothing logic and UI behavior

Acceptance Criteria

- [] Users can toggle between raw and smoothed data
- [] Rolling chart updates with time range selections
- [] Visual feedback is responsive and accurate
- [] Data is correctly smoothed with adjustable window

Labels

feature, chart, data-visualization, enhancement

Estimate

M

markdown

Title

Enable Filter Presets and Save/Load Functionality

Description

Introduce filter presets that allow users to save commonly used filter combinations (e.g., date ranges, durations, user types) and reload them easily. This enhances user experience for those analyzing frequent patterns or reports.

Tasks

- [] Create a UI component for managing filter presets (save,

```
load, delete)
- [ ] Store presets locally using browser storage or Firestore
- [ ] Auto-apply a default preset if one is selected
- [ ] Handle edge cases (e.g., deleted fields in saved
presets)
- [ ] Unit test filter saving/loading logic
```

Acceptance Criteria

- [] Users can name and save current filters as a preset
- [] Users can load and apply saved presets
- [] Presets persist across sessions
- [] Application behaves gracefully if a preset contains
obsolete fields

Labels

feature, filters, UX, enhancement

Estimate

M

markdown

Title

Implement Rolling Average Line for Daily Usage Chart

Description

Enhance the daily usage chart by adding a toggleable rolling average line (e.g., 7-day or 30-day). This helps users better understand trends and smooth out daily variances in their smoking behavior.

Tasks

- [] Add toggle UI to enable/disable rolling average
- [] Calculate rolling average on the filtered dataset
- [] Plot rolling average line alongside bar/area chart
- [] Style the line distinctively (dashed or lighter color)
- [] Ensure mobile responsiveness

Acceptance Criteria

- [] Rolling average accurately reflects selected range and filters
- [] Toggle allows seamless on/off interaction
- [] No performance degradation on large datasets

Labels

feature, chart, visualization, enhancement

Estimate

M

markdown

Title

Add Loading State Indicators for Chart and Table Views

Description

Improve user experience by adding clear loading indicators when charts or tables are fetching data or applying filters. This will prevent confusion during delays caused by heavy data processing or slow network connections.

Tasks

- [] Show spinner or skeleton UI while data is loading
- [] Display a fallback message if data fails to load
- [] Ensure loading state triggers during initial load and when filters are applied
- [] Test responsiveness across devices

Acceptance Criteria

- [] Visible and styled loading indicators appear during data fetching
- [] No flicker or abrupt transitions
- [] Graceful error handling when data fails to load

Labels

ui, enhancement, quality-of-life, ux

Estimate

S

markdown

Title

Integrate Rolling Usage Chart with Adjustable Time Window

Description

Add a new "Rolling Usage" chart to visualize trends over customizable time windows (e.g., 1-day, 3-day, 7-day averages). This feature will help users identify smoking patterns more easily over time.

Tasks

- [] Implement data processing logic for rolling averages
- [] Add dropdown or selector for window size (1d, 3d, 7d)
- [] Display rolling chart alongside existing usage charts
- [] Ensure responsiveness and mobile compatibility
- [] Add toggle to enable/disable rolling average overlay

Acceptance Criteria

- [] Chart updates dynamically with window size
- [] Data is clearly labeled and visually distinct from raw usage
- [] Works seamlessly with date range filters

Labels

charting, enhancement, data-visualization, UX

Estimate

M

markdown

Title

Implement Weekly and Monthly Summary Views

Description

Create summary views for weekly and monthly usage to give users a quick snapshot of their habits over longer periods. These summaries should include total puffs, average daily use, longest streaks, and usage heatmaps.

Tasks

- [] Design summary layout for weekly and monthly views
- [] Aggregate data by week and month
- [] Calculate and display statistics (totals, averages, streaks)
- [] Integrate summary into dashboard or a new "Summary" tab
- [] Style with visual clarity and accessibility in mind

Acceptance Criteria

- [] Summary updates based on selected date range
- [] Stats are accurate and reflect user data
- [] View is optimized for both mobile and desktop

Labels

enhancement, data-summary, UI/UX

Estimate

M

markdown

Title

Add Rolling Average and Trend Indicators to Puff Usage Charts

Description

Enhance the visual clarity of puff usage patterns by overlaying rolling averages and trend indicators on the existing usage charts. This will help users identify long-term patterns and deviations in their smoking habits.

Tasks

- [] Implement calculation of 7-day and 30-day rolling averages

- [] Overlay rolling average lines on daily and weekly charts
- [] Add upward/downward trend arrows or labels where applicable
- [] Ensure responsive design and clear differentiation of lines
- [] Provide toggle controls to enable/disable trends

Acceptance Criteria

- [] Rolling average lines are correctly computed and displayed
- [] Trends are updated with user-selected date ranges
- [] Users can toggle visibility of trend indicators

Labels

enhancement, analytics, charting

Estimate

M

markdown

Title

Improve Chart Legend Usability and Layout on Small Screens

Description

Users on smaller screens report difficulties reading and interacting with the chart legends, especially when multiple data series are displayed. Refactor the chart legend to be more mobile-friendly and improve clarity without obscuring chart content.

Tasks

- [] Redesign the chart legend layout for mobile and tablet screens
- [] Ensure legend items wrap or scroll without overlapping the chart
- [] Highlight active data series on hover or tap
- [] Test across major mobile devices and screen sizes

Acceptance Criteria

- [] Chart legends are clearly visible and accessible on all screen sizes
- [] No overlap with chart content
- [] User can toggle/hide legend entries as needed

Labels

UX, responsive-design, enhancement

Estimate

S

markdown

Title

Implement Data Export Functionality for Smoke Logs

Description

Users may want to export their smoke log data for backup, analysis, or sharing with health professionals. Add functionality to allow exporting data in common formats like CSV or JSON.

Tasks

- [] Add export button in the log view interface
- [] Implement data formatting for CSV and JSON
- [] Allow user to select format and time range
- [] Ensure correct encoding and field labeling
- [] Test downloads on both web and mobile platforms

Acceptance Criteria

- [] Users can export their logs in both CSV and JSON formats
- [] Exported data includes timestamps, inhalation lengths, and any notes
- [] Export respects user-selected time range

Labels

feature, data, enhancement

Estimate

M

markdown

Title

Add Search and Filter Functionality to Smoke Log

Description

To improve usability and help users quickly find relevant entries, implement search and filter functionality in the smoke log interface. Users should be able to search by notes and filter entries by date range, duration length, or tags (if implemented later).

Tasks

- [] Design UI for search input and filter dropdowns or chips
- [] Implement search by keyword in notes field
- [] Add filter options: date range, inhalation duration range
- [] Ensure performance with large datasets
- [] Add unit tests and UI tests for filters

Acceptance Criteria

- [] Users can search entries using a keyword
- [] Users can filter logs by date and inhalation length
- [] Filters and search can be used together without conflict

Labels

feature, UX, enhancement

Estimate

M

markdown

Title

Implement Cumulative Usage Graph with Range Selector

Description

Add a cumulative usage graph that visualizes total inhalation duration over time. Incorporate a range selector allowing users to choose different time windows (e.g., 1 week, 1 month, custom range) to view trends and patterns in usage behavior.

Tasks

- [] Design chart layout with cumulative sum line
- [] Implement range selector for time-based filtering
- [] Integrate chart with Firestore or local data source
- [] Ensure responsiveness and accessibility
- [] Write unit and integration tests

Acceptance Criteria

- [] Users can view a cumulative line graph of usage over time
- [] Users can select time ranges and see the graph update accordingly
- [] Graph updates smoothly and matches filtered data below

Labels

feature, data-visualization, UX

Estimate

M

markdown

Title

Enhance Data Table with Sorting and Timestamp Filtering

Description

Upgrade the usage data table below the graph to allow column sorting (e.g., by timestamp or duration) and filtering by timestamp range. This will provide a clearer view of relevant logs matching the selected timeframe.

Tasks

- [] Add sorting capabilities to table columns
- [] Enable filtering based on the selected range in the graph
 - [] Sync table updates with chart interactions
 - [] Ensure performance with larger datasets
 - [] Write tests for sorting and filtering functionality

Acceptance Criteria

- [] Users can sort table data by clicking column headers
- [] Table updates to reflect the time range selected on the chart
 - [] Functionality is smooth and responsive across devices

Labels

enhancement, UI, data

Estimate

M

markdown

Title

Add Daily and Weekly Usage Summaries to Dashboard

Description

Introduce summary boxes on the dashboard displaying total inhalation count and cumulative duration for the current day and week. This helps users quickly assess their recent usage trends without needing to interpret graphs.

Tasks

- [] Calculate total inhalations and duration for the current day
- [] Calculate totals for the current week (starting Monday)
- [] Display both sets of totals in styled summary boxes above or beside the chart
 - [] Ensure values update dynamically with new entries
 - [] Apply consistent styling and responsive layout

Acceptance Criteria

- [] Dashboard shows today's and this week's usage summaries
- [] Summaries are clear, visually distinct, and update in real-time
- [] Works across all screen sizes

Labels

feature, UI, dashboard

Estimate

S

markdown

Title

Enable Filter by Inhalation Duration Ranges

Description

Implement functionality allowing users to filter logs and visualizations based on customizable inhalation duration ranges. This will provide better insight into usage patterns and help distinguish between short and long inhalation sessions.

Tasks

- [] Define preset and custom duration ranges (e.g., 0-2s, 2-5s, 5s+)
- [] Add filter controls to the dashboard interface
- [] Update chart and data table to reflect filtered data
- [] Ensure filter state is preserved during navigation

Acceptance Criteria

- [] User can filter usage data by duration range
- [] Filter affects both chart and table data
- [] Preset options and a custom range selector are available
- [] Filter UI is responsive and intuitive

Labels

feature, filter, UX

Estimate

M

markdown

Title

Implement Rolling Averages for Usage Trends

Description

Add rolling average calculations (e.g., 7-day, 30-day) to visualizations to help users identify trends and smooth out short-term fluctuations in inhalation data.

Tasks

- [] Add toggle for enabling/disabling rolling averages on charts
- [] Support multiple window sizes (e.g., 7, 14, 30 days)
- [] Display averaged data as an overlay on the primary line chart
- [] Ensure correct behavior when filtering date ranges

Acceptance Criteria

- [] Users can enable rolling averages and select a window size
- [] Rolling average line is clearly distinguishable from raw data
- [] Data updates dynamically with filters
- [] Feature works on both desktop and mobile views

Labels

enhancement, data-visualization, analytics

Estimate

M

markdown

Title

Add Weekly and Monthly Summary Cards

Description

Design and implement summary cards that provide a quick overview of usage for the current week and month. Each card should display total inhalations, total duration, and average session length.

Tasks

- [] Create reusable card components for summary display
- [] Calculate weekly and monthly totals and averages
- [] Style cards to match current app theme
- [] Integrate summary cards into the main dashboard view

Acceptance Criteria

- [] Summary cards are visible at the top of the dashboard
- [] Data reflects the currently selected time range
- [] Responsive design ensures usability on all screen sizes

Labels

enhancement, UI, dashboard, analytics

Estimate

M

markdown

Title

Implement Rolling Averages Toggle for Usage Graph

Description

Add a toggle feature to the usage graph that allows users to view a rolling average (e.g., 3-day or 7-day) of their smoking activity. This will help identify trends over time by smoothing out short-term fluctuations.

Tasks

- [] Add toggle UI component for rolling average selection (None, 3-day, 7-day)
- [] Update chart logic to calculate and display rolling averages
- [] Ensure original raw data can still be viewed by deselecting the rolling average
- [] Write tests for new graph behavior with rolling average enabled

Acceptance Criteria

- [] Users can switch between raw data and rolling averages on the graph
- [] Chart updates dynamically based on toggle selection
- [] Design is consistent with existing app style

Labels

feature, UI, analytics, charting

Estimate

M

markdown

Title

Refactor Chart Data Aggregation Logic for Consistency

Description

Consolidate the chart data processing and aggregation logic into a single shared module to improve maintainability and reduce redundant computations between different chart types (e.g., daily, cumulative, rolling averages).

Tasks

- [] Identify all locations where chart data is transformed or aggregated
- [] Create a utility module to house shared aggregation logic
- [] Refactor existing chart components to use the new utility functions

- [] Ensure consistency in time zone handling and formatting across charts
- [] Add unit tests for utility functions

Acceptance Criteria

- [] All chart components use the shared data aggregation logic
- [] No regressions in chart display or behavior
- [] Code coverage for new utilities is at least 90%

Labels

refactor, analytics, charting, code-quality

Estimate

M

markdown

Title

Implement Session-Aware Logging Logic for Dev and Prod Modes

Description

Ensure the `submitData` function uses separate Firestore collections depending on whether the app is running in development or production. This will prevent test data from mixing with real user data.

Tasks

- [] Add a config variable to detect dev vs. prod environment
- [] Update `submitData` to dynamically write to the correct collection (`SmokeLogsTest` for dev, `SmokeLogs` for prod)
- [] Validate the collection name in Firestore before writing
- [] Add visual indication (e.g. badge or banner) in the UI for development mode

Acceptance Criteria

- [] No production data is written during development sessions
- [] Correct Firestore collection is used based on

environment

- [] App provides clear visual indicator of the current environment

Labels

backend, firestore, environment-handling, bugfix

Estimate

S

markdown

Title

Refactor Chart Rendering Logic for Scalability and Maintainability

Description

Consolidate chart rendering logic and prepare the architecture to support additional chart types and analysis in the future. This refactor should ensure maintainable code and easier integration of features like rolling averages, cumulative totals, and customizable visualizations.

Tasks

- [] Move chart rendering logic to a separate utility or controller file
- [] Create a flexible data interface for chart inputs
- [] Implement switchable chart types using the current filtered data
- [] Add unit tests for chart rendering functions
- [] Ensure current visualizations (line chart, cumulative view) are preserved

Acceptance Criteria

- [] Chart code is modular and reusable
- [] Switching chart types does not impact global state
- [] New chart types can be added with minimal effort

Labels

frontend, refactor, charting, enhancement

Estimate

M

markdown

Title

Integrate Rolling Usage Calculation into Data Analysis Options

Description

Implement a rolling usage analysis feature to help users understand trends over time by applying a moving average or windowed sum to usage data. This will enhance the app's analytical depth and provide smoother insights into consumption behavior.

Tasks

- [] Add option in range selector to toggle rolling usage
- [] Implement backend or frontend calculation for rolling window analysis
- [] Allow user to select window size (e.g., 3-day, 7-day)
- [] Update chart rendering to reflect rolling usage view
- [] Ensure compatibility with other chart features (e.g., cumulative mode)

Acceptance Criteria

- [] Users can select and view rolling usage for selected date ranges
- [] Chart updates dynamically based on rolling window size
- [] Feature does not impact performance or cause UI issues

Labels

frontend, enhancement, charting, analytics

Estimate

M

markdown

Title

Add Editable Notes Section for Each Log Entry

Description

Users should be able to attach personal notes to each log entry. This feature enhances the app's value as a journaling tool and allows users to record contextual information about their sessions.

Tasks

- [] Update Firestore schema to support a `notes` field for log entries
- [] Add UI component below or alongside each entry for viewing/editing notes
- [] Implement autosave or manual save functionality for notes
- [] Allow Markdown support (optional) for richer formatting

Acceptance Criteria

- [] Notes can be added, edited, and saved per log entry
- [] Notes persist between sessions and reloads
- [] Notes are excluded from charts and analytics by default

Labels

frontend, backend, enhancement, user-experience

Estimate

M

markdown

Title

Implement Export Feature for Log Data (CSV/JSON)

Description

Allow users to export their inhalation log data in CSV and JSON formats for backup, personal analysis, or importing into

other tools. This feature is essential for users who want more control over their data.

Tasks

- [] Add backend function to generate CSV and JSON from user's log data
- [] Create UI button under settings or logs view for "Export Data"
- [] Provide format selection (CSV or JSON) and date range filters
- [] Trigger file download in browser after export is generated

Acceptance Criteria

- [] Users can successfully export their data in either CSV or JSON format
- [] Export respects selected date range (if any)
- [] Export includes timestamps, lengths, notes, and tags

Labels

feature, backend, frontend, export, user-control

Estimate

L

markdown

Title

Add Calendar View for Daily and Monthly Usage Patterns

Description

Introduce a calendar-style view that lets users visualize their smoking patterns by day or month. This helps with long-term trend analysis and encourages consistent tracking.

Tasks

- [] Design UI for monthly calendar with color-coded usage intensity
- [] Enable navigation between months and selection of

individual days

- [] Fetch and display summary stats for each day (e.g., total length, number of sessions)
- [] Support tap/click on a day to jump to detailed view

Acceptance Criteria

- [] Calendar renders all days of the month, highlighting usage intensity
- [] Users can navigate between months without performance issues
- [] Clicking a day reveals detailed usage for that date

Labels

feature, frontend, visualization, UX

Estimate

M

markdown

Title

Implement Rolling Average Graph for Usage Trends

Description

Add a line graph that visualizes the user's average usage over a rolling window (e.g., 7-day or 30-day average). This feature will help users identify long-term trends and evaluate their progress toward goals.

Tasks

- [] Add toggle for enabling/disabling rolling average graph
- [] Compute rolling averages based on selected time range
- [] Update chart component to overlay rolling average on existing usage chart
- [] Adjust chart legend and axis to accommodate new line

Acceptance Criteria

- [] Users can view and toggle a rolling average trend line
- [] Graph updates accurately based on the selected window

- [] UI remains clear and responsive with added data

Labels

feature, frontend, analytics, chart

Estimate

M

markdown

Title

Enhance Data Export Feature with CSV and PDF Support

Description

Expand the data export functionality to allow users to download their logs as CSV or PDF files. This enables better personal record-keeping and sharing with healthcare providers or accountability partners.

Tasks

- [] Add "Export as CSV" and "Export as PDF" buttons to the dashboard
- [] Format exported data to include timestamp, inhalation duration, and any notes
- [] Ensure PDF exports use clean, readable formatting
- [] Support date range filtering for exports

Acceptance Criteria

- [] Users can export filtered logs in CSV or PDF format
- [] Exported data includes all relevant fields
- [] Files are named with the export date and range

Labels

feature, export, user-request, medium-priority

Estimate

M

markdown

Title

Add Chart Type Selector for Usage Visualization

Description

Introduce a chart type selector to allow users to switch between different visualizations (line, bar, area, etc.) for inhalation data. This enhances data readability and gives users flexibility in how they interpret their logs.

Tasks

- [] Add dropdown or toggle for selecting chart type
- [] Support at least Line, Bar, and Area chart options
- [] Preserve selected chart type across sessions using local storage
- [] Update tooltips and axes based on chart type

Acceptance Criteria

- [] Users can seamlessly switch between supported chart types
- [] The selected chart type persists on page reload
- [] Chart updates visually and functionally without error

Labels

feature, frontend, visualization, user-enhancement

Estimate

S

markdown

Title

Implement Dynamic Range Selection for Rolling Usage Analysis

Description

Introduce dynamic range selection capabilities to support rolling usage analytics. This feature will allow users to explore trends like 7-day or 30-day moving averages, enhancing

insight into long-term behavior patterns.

Tasks

- [] Add UI component to select rolling window (e.g., 7-day, 30-day)
- [] Modify data processing logic to compute rolling averages
- [] Update charts to visualize rolling metrics alongside original data
- [] Ensure responsive updates when filters or ranges are changed

Acceptance Criteria

- [] Users can select from predefined rolling ranges
- [] Chart updates to reflect chosen rolling average correctly
- [] No performance lag with large datasets

Labels

feature, analytics, frontend, enhancement

Estimate

M

markdown

Title

Improve Data Table Interactivity and Usability

Description

Enhance the data table that displays submitted logs to support more intuitive interaction. This includes sticky headers, column sorting, and an improved scrollbar for better navigation.

Tasks

- [] Implement sticky headers to keep column labels visible during scroll
- [] Enable sorting by timestamp, duration, and other relevant columns

- [] Improve vertical and horizontal scroll behavior with responsive design
- [] Add hover tooltips for truncated values

Acceptance Criteria

- [] Table headers remain visible when scrolling
- [] Users can sort data by clicking column headers
- [] Scrolling is smooth and works well on different screen sizes
- [] Tooltips provide full context for abbreviated values

Labels

enhancement, frontend, UX, data-visualization

Estimate

M

markdown

Title

Add Daily Summary Chart for Smoke Usage Patterns

Description

Introduce a new chart that summarizes daily smoking behavior trends over time. This chart will visualize total inhalation duration per day to help users identify patterns and evaluate progress.

Tasks

- [] Create a bar or line chart showing daily cumulative usage
- [] Connect chart to filtered data set based on current range selector
- [] Add labels for date and total duration per bar/point
- [] Ensure chart updates when filters are adjusted

Acceptance Criteria

- [] Chart displays one data point per day within selected range

- [] Each point/bar reflects the total usage for that day
- [] Updates dynamically based on filters and selections
- [] Design matches overall UI aesthetic

Labels

feature, frontend, chart, data-visualization

Estimate

M

markdown

Title

Implement Rolling Average Analysis Chart

Description

Add a line chart that displays a rolling average of inhalation durations over time to give users insights into trends in their usage patterns. This should help highlight progress or regression more clearly than raw data.

Tasks

- [] Calculate rolling average (default to 7-day window)
- [] Create chart component to visualize rolling average over time
- [] Allow adjustable window (3, 7, 14 days) through a dropdown or slider
- [] Ensure chart interacts with the existing date range filter

Acceptance Criteria

- [] Users can select rolling average window
- [] Chart updates dynamically and reflects filtered data range
- [] Styling is consistent with other charts
- [] Proper axis labeling for date and average duration

Labels

feature, frontend, chart, analysis

Estimate

M

markdown

Title

Implement Log Entry Detail Modal

Description

Create a modal view that displays detailed information for a single inhalation log entry when selected from the data table. This feature enhances user insight into individual logs and supports more granular analysis.

Tasks

- [] Design modal layout with timestamp, duration, and any notes or metadata
- [] Trigger modal on row click in data table
- [] Ensure responsive and accessible design
- [] Allow closing via escape key, close button, or click outside

Acceptance Criteria

- [] Clicking a row opens a modal with correct log details
- [] Modal functions on mobile and desktop
- [] User can easily close modal without page reload

Labels

feature, frontend, UI/UX, modal

Estimate

S

markdown

Title

Add Rolling Averages and Cumulative Usage Charts

Description

Introduce new visualizations to enhance insight into smoking patterns by displaying rolling averages and cumulative usage over time. These charts will allow users to better understand trends and long-term behavior.

Tasks

- [] Implement logic for calculating rolling average (e.g., 7-day or 30-day)
- [] Implement logic for cumulative usage over time
- [] Add new charts to the analysis section with appropriate toggles
- [] Style charts for clarity and responsiveness

Acceptance Criteria

- [] Charts display correctly with accurate data
- [] Users can toggle between daily, rolling, and cumulative views
- [] Chart updates respect selected date ranges and filters

Labels

feature, analytics, frontend, chart

Estimate

M

markdown

Title

Implement Timeline View of Daily Usage Logs

Description

Introduce a timeline-based visualization where each day's logs are displayed as sequential entries, emphasizing time of day and inhalation length. This visual approach will provide a more intuitive understanding of daily patterns and fluctuations.

Tasks

- [] Design and implement a horizontal timeline or stacked bar format
- [] Display logs per day segmented by time, color-coded by inhalation length
- [] Enable scroll or pagination for browsing multiple days
- [] Integrate with current data filtering and range selection

Acceptance Criteria

- [] Timeline displays accurate data aligned with user time zones
- [] Inhalation length is visually distinguishable (e.g., gradient or height)
- [] Navigation between days/weeks functions smoothly

Labels

feature, visualization, frontend, UX

Estimate

M

markdown

Title

Add Weekly Summary Card with Key Metrics

Description

Create a compact, visually appealing summary card at the top of the dashboard displaying weekly stats such as:

- Total sessions
- Average inhalation length
- Most active day
- Trend (up/down compared to last week)

This component provides quick insights and enhances user engagement by highlighting usage trends.

Tasks

- [] Design card layout for summary metrics
- [] Implement logic to calculate weekly stats
- [] Display comparison with previous week using icons or color cues
- [] Ensure responsiveness across devices

Acceptance Criteria

- [] Summary card displays correct metrics with real-time data
- [] Metrics update when the time range is adjusted
- [] UI adapts well to mobile and desktop views

Labels

feature, UI, enhancement, analytics

Estimate

S

markdown

Title

Integrate Custom Range Picker with Chart and Table Sync

Description

Allow users to select a custom date range using a unified date picker that synchronizes both the inhalation chart and the data table. This improves usability by consolidating control and ensuring consistent data visibility across components.

Tasks

- [] Create a custom range picker component
- [] Connect the picker to both the chart and table views
- [] Ensure reactive updates when the range changes
- [] Handle default and edge cases (e.g. no data, very large range)

Acceptance Criteria

- [] Selecting a range updates both the chart and table seamlessly

- [] Default range is set to the past 7 days if none selected
- [] UI responds without page reload or visual flicker

Labels

feature, UX, charting, table, enhancement

Estimate

M

markdown

Title

Enable Offline Mode with Local Data Caching

Description

Introduce support for offline functionality in the Smoke-Log app using Isar database to cache user data locally. This ensures users can continue logging and viewing entries without an internet connection.

Tasks

- [] Implement local caching using Isar for inhalation entries
- [] Sync offline data with Firestore when the app goes online
- [] Display offline mode status in the UI
- [] Prevent data duplication or conflict during sync

Acceptance Criteria

- [] App stores new logs locally when offline and syncs them automatically when reconnected
- [] User receives visual indication when offline
- [] Data consistency is maintained between local and remote sources

Labels

feature, offline, database, enhancement, isar

Estimate

L

markdown

Title

Add Weekly Usage Summary Chart to Dashboard

Description

Implement a new weekly summary chart on the Smoke-Log dashboard to help users visualize their inhalation patterns over the past 7 days. This enhancement supports the goal of improved user insight and habit tracking.

Tasks

- [] Design a compact, responsive chart component (e.g., bar or line chart)
- [] Aggregate inhalation data by day for the last 7 days
- [] Display total inhalation duration and count per day
- [] Highlight current day and display tooltip data

Acceptance Criteria

- [] Users see a clear, visually consistent chart summarizing weekly data
- [] The chart updates in real-time with new entries
- [] Data remains accurate regardless of time zone or usage frequency

Labels

feature, dashboard, chart, enhancement, data-visualization

Estimate

M

markdown

Title

Implement Rolling 7-Day Average Inhalation Chart

Description

Add a new rolling average line chart to the Smoke-Log dashboard that shows the 7-day average of inhalation duration over time. This chart helps users visualize trends and track progress over extended periods, smoothing out daily fluctuations.

Tasks

- [] Calculate 7-day rolling average of inhalation duration from stored data
- [] Display this data using a smooth line chart
- [] Allow toggling between daily totals and rolling average
- [] Ensure the chart integrates with date range filters

Acceptance Criteria

- [] Rolling average is accurately calculated and visually distinct
- [] Users can toggle this view on and off
- [] Chart performs well even with large datasets

Labels

feature, chart, analytics, enhancement, visualization

Estimate

M

markdown

Title

Create Summary Statistics Card for Inhalation Data

Description

Add a new UI component that displays key summary statistics at the top of the dashboard. This includes total inhalations, average daily inhalation duration, longest single session, and most active day. This will give users a quick snapshot of their usage patterns.

Tasks

- [] Compute and display total inhalation count
- [] Show average daily inhalation duration over selected range
- [] Highlight longest inhalation session with timestamp
- [] Identify and show the most active day

Acceptance Criteria

- [] Statistics update correctly with date range filters
- [] Responsive design for mobile and desktop
- [] Clear and concise labels with appropriate formatting

Labels

feature, UI, analytics, enhancement, summary

Estimate

S

markdown

Title

Implement Rolling Average Chart for Inhalation Trends

Description

Create a new chart that displays the rolling average of inhalation duration over time to help users visualize trends more clearly. This will be added as a new tab in the existing chart section.

Tasks

- [] Add UI option to toggle or select rolling average view
- [] Calculate rolling average (e.g., 7-day or 14-day) based on selected range
- [] Plot the rolling average line alongside existing inhalation data
- [] Ensure chart updates dynamically with filters

Acceptance Criteria

- [] Rolling average updates in real-time with data selection

- [] Works correctly with different time ranges
- [] Matches design aesthetics of current charts

Labels

feature, UI, analytics, visualization, enhancement

Estimate

M

markdown

Title

Integrate Weekly Usage Summary Card

Description

Add a summary card component to the dashboard that displays weekly usage stats: total inhalations, average inhalation length, and most active day of the week. This gives users a quick glance at their behavior trends.

Tasks

- [] Design and place a responsive summary card on the dashboard
- [] Compute weekly stats from Firestore data
- [] Highlight the most active weekday with appropriate styling
- [] Ensure updates as users add or delete logs

Acceptance Criteria

- [] Weekly data is computed correctly from current logs
- [] Card is visible and styled consistently across screen sizes
- [] Most active day is highlighted accurately

Labels

feature, UI, enhancement, dashboard

Estimate

M

markdown

Title

Implement Inhalation Streak Tracker

Description

Create a component that tracks and displays the user's current and longest daily streak of logging inhalations. This encourages consistency and builds user habits through positive reinforcement.

Tasks

- [] Calculate the number of consecutive days with at least one inhalation log
- [] Track and store longest streak for display
- [] Display current and longest streak in a visible card or badge on the dashboard
- [] Add visual reinforcement (e.g., icon or badge) for active streak

Acceptance Criteria

- [] Streak updates in real-time with new logs
- [] Correctly resets if no logs are recorded on a day
- [] Matches historical logs accurately for the longest streak

Labels

feature, enhancement, habit-tracking, dashboard

Estimate

M

markdown

Title

Implement Log Submission Timestamp Adjuster

Description

Enable users to modify the timestamp of a submitted inhalation log. This feature helps users retroactively correct or input logs they forgot to record in real-time.

Tasks

- [] Add a timestamp field with datetime picker to the inhalation submission modal
- [] Allow default to current time but enable user overrides
- [] Validate custom timestamps to prevent future-dated logs
- [] Update Firestore document creation to use selected timestamp

Acceptance Criteria

- [] Users can set a custom timestamp for each log
- [] Logs appear in charts and tables according to their adjusted time
- [] Timestamp is stored correctly in the Firestore document

Labels

feature, data-entry, timestamp, firestore

Estimate

M

markdown

Title

Add Rolling Average Usage Chart

Description

Introduce a chart that displays a rolling average of inhalation durations over a user-defined window (e.g., 7 days, 30 days). This helps users visualize trends and smooth out anomalies.

Tasks

- [] Add rolling average option to chart filter menu
- [] Implement logic to calculate rolling averages based on selected time window
- [] Display rolling average line on existing chart with distinguishable styling
- [] Update legend and tooltips to reflect rolling average values

Acceptance Criteria

- [] Users can select a rolling average window (e.g., 7, 14, 30 days)
- [] Rolling average appears as a separate line on the chart
- [] Data updates dynamically based on user selection

Labels

feature, chart, visualization, rolling-average

Estimate

M

markdown

Title

Implement Daily Usage Summary Card

Description

Add a summary card to the dashboard that displays total inhalation count and cumulative duration for the current day. This provides users with an at-a-glance view of their daily usage.

Tasks

- [] Create a summary card UI component
- [] Query and filter today's inhalation data
- [] Display total inhalations and total duration (formatted in mm:ss)
- [] Refresh card dynamically if data changes

Acceptance Criteria

- [] The card correctly displays today's total inhalations and duration
- [] Data updates automatically as new logs are added
- [] Visual style aligns with existing dashboard components

Labels

feature, dashboard, summary, UX

Estimate

S

markdown

Title

Add Streak Tracker to Dashboard

Description

Introduce a visual element on the dashboard that shows the user's current streak of days where usage was recorded. This encourages consistent tracking and motivates regular use of the app.

Tasks

- [] Calculate streak based on consecutive days with at least one inhalation logged
- [] Display current streak count on the dashboard
- [] Style component to match dashboard aesthetics
- [] Add hover or tooltip info with details on longest streak and last missed day

Acceptance Criteria

- [] Streak count accurately reflects data in Firestore
- [] Updates in real time or on data refresh
- [] Clear messaging when streak is broken or restarted

Labels

enhancement, dashboard, user-engagement, frontend

Estimate

M

markdown

Title

Implement Daily Log Entry Summary Modal

Description

Create a modal that appears when a user taps a date on the usage chart or calendar, showing a summary of that day's log entries. This enhances the ability to review past activity without navigating away from the main screen.

Tasks

- [] Add tap/click handler to date elements in chart/calendar
- [] Retrieve and display all inhalation records for that date
- [] Format summary to include total usage, time intervals, and average length
- [] Style modal for consistency with app theme
- [] Add close button and backdrop behavior

Acceptance Criteria

- [] Modal opens and closes correctly
- [] Shows accurate and complete information for selected day
- [] Fully responsive and accessible

Labels

feature, ui, modal, daily-summary, user-experience

Estimate

M

markdown

Title

Add Visual Cue for Active Smoking Sessions

Description

Enhance user awareness by adding a visual indicator when an active smoking session is in progress. This could include a glowing border, pulsing icon, or animated element to signal ongoing logging.

Tasks

- [] Design visual cue (e.g., pulse ring, glowing icon)
- [] Integrate with session start/end logic
- [] Ensure indicator is present on all relevant screens
- [] Add setting toggle for enabling/disabling the cue

Acceptance Criteria

- [] Visual cue appears only during active sessions
- [] User can toggle visual cue in settings
- [] Cue does not interfere with other UI elements

Labels

feature, ui, session-indicator, user-feedback, accessibility

Estimate

S

markdown

Title

Implement Editable Log Entry Feature

Description

Allow users to edit existing log entries in AshTrail. This includes changing the timestamp, inhalation length, or any associated notes. Add a simple edit icon on each log item, opening a modal or inline editor.

Tasks

- [] Add edit icon/button on each log entry
- [] Create edit modal or inline editor
- [] Validate updates and save changes to Firestore
- [] Show success or error feedback to user

Acceptance Criteria

- [] Users can update timestamp, inhalation length, and notes
- [] Changes are saved and reflected in the UI
- [] Editing is disabled for entries older than 48 hours
(optional constraint)

Labels

feature, log-management, firestore, user-interface, medium-priority

Estimate

M

markdown

Title

Add Rolling Usage Line Chart to AshTrail Dashboard

Description

Introduce a line chart that shows rolling usage over selected time intervals (e.g., daily, weekly, monthly averages). This helps visualize changes in smoking patterns more clearly over time. Integrate it with the existing date range selector.

Tasks

- [] Design and implement rolling usage calculation logic
- [] Add new line chart component to dashboard
- [] Connect chart to date range selector and filtered logs
- [] Ensure responsiveness and usability across devices

Acceptance Criteria

- [] Rolling usage is calculated and visualized correctly
- [] Chart updates based on selected date range
- [] Chart displays loading states and empty messages when

applicable

Labels

feature, charting, analytics, user-interface, medium-priority

Estimate

M

markdown

Title

Implement Total and Average Usage Stats Summary Box

Description

Add a summary section at the top or bottom of the AshTrail dashboard that displays total smoking sessions and average inhalation length over the selected date range. This offers users a quick snapshot of their usage trends.

Tasks

- [] Calculate total number of logs in current range
- [] Calculate average inhalation length in current range
- [] Design a simple summary box UI
- [] Integrate the stats with the date range filtering logic

Acceptance Criteria

- [] Stats update dynamically when the date range changes
- [] Stats box is visually consistent with the app's design
- [] Includes fallback text when no data is available

Labels

feature, summary, user-experience, UI, medium-priority

Estimate

S

markdown

Title

Create Rolling Average Analysis Option for Line Chart

Description

Enhance the line chart visualization by allowing users to toggle a rolling average overlay. This feature smooths out short-term fluctuations and highlights long-term trends in inhalation data.

Tasks

- [] Add rolling average toggle UI near chart controls
- [] Compute rolling average using a 3-point and 7-point window
- [] Update line chart to overlay the selected rolling average
- [] Ensure responsiveness with date range filtering

Acceptance Criteria

- [] Users can toggle between 3-day, 7-day, and no smoothing
- [] Chart overlays update immediately upon selection
- [] Rolling average line is distinguishable from raw data line

Labels

feature, charting, data-analysis, UI, medium-priority

Estimate

M

markdown

Title

Add Weekly and Monthly Summary Views to Dashboard

Description

Introduce new summary views on the dashboard to display weekly and monthly inhalation usage statistics. This helps users identify trends and patterns over broader time spans beyond

daily use.

Tasks

- [] Add tabs or dropdown for switching between daily, weekly, and monthly summaries
- [] Aggregate inhalation data by week and month
- [] Display total inhalations, average length, and max inhalation for selected period
- [] Include bar or stacked column chart for visual comparison

Acceptance Criteria

- [] Users can view accurate weekly and monthly stats
- [] Data visualizations update smoothly when toggling between views
- [] Each summary includes totals, averages, and maximums

Labels

feature, dashboard, analytics, data-aggregation, high-priority

Estimate

L

markdown

Title

Implement Rolling Average Usage Analysis Chart

Description

Add a line chart to the dashboard that displays a 7-day and 30-day rolling average of inhalation duration. This visualization helps users understand long-term trends and fluctuations in their smoking behavior.

Tasks

- [] Calculate 7-day and 30-day rolling averages for inhalation length
- [] Add toggle to display/hide rolling average overlays on the chart

- [] Use distinct line styles or colors for each rolling average
- [] Ensure the chart updates accurately based on selected date range

Acceptance Criteria

- [] Rolling average lines appear correctly on the chart
- [] Users can toggle visibility of each rolling average
- [] Lines update smoothly with range changes and filters

Labels

feature, analytics, visualization, rolling-average, enhancement

Estimate

M

markdown

Title

Implement Calendar Heatmap of Daily Usage

Description

Add a calendar heatmap to visually represent daily inhalation usage over time. This component should show how usage trends shift month-to-month and help users spot patterns like high-usage days or improvements in frequency reduction.

Tasks

- [] Design and implement a calendar-style heatmap component
- [] Map daily total inhalation duration or frequency to color intensity
- [] Allow users to toggle between duration and frequency views
- [] Integrate with the existing date range filters

Acceptance Criteria

- [] Calendar view displays full months, with cells for each day

- [] Color intensity correlates accurately with usage values
- [] Users can easily distinguish high/low usage days at a glance
- [] Component updates based on date range selections

Labels

feature, visualization, calendar, enhancement, dashboard

Estimate

M

markdown

Title

Implement Milestone-Based Achievements System

Description

Introduce an achievements system to reward users for reaching key milestones in their smoke-tracking journey. This feature will increase user engagement by celebrating consistency, reductions in usage, or personal records.

Tasks

- [] Define initial set of achievements (e.g., first log, 7-day streak, 50% usage reduction)
- [] Create backend logic to evaluate and award achievements
- [] Design visual badges or icons for each achievement
- [] Display earned achievements on the dashboard or profile screen

Acceptance Criteria

- [] Achievements are awarded automatically upon meeting criteria
- [] Users can view a list of earned and locked achievements
- [] Achievements system is scalable for future additions

Labels

feature, engagement, enhancement, dashboard, gamification

Estimate

L

markdown

Title

Add Cumulative Usage Progress Visualization

Description

Develop a visualization that displays the user's cumulative smoke usage over time. This will help users identify long-term trends, plateaus, and reductions, offering valuable insights into their smoking habits.

Tasks

- [] Aggregate historical usage data by week/month
- [] Design and implement a line chart or area graph for cumulative usage
- [] Add controls to filter by date range (e.g., 3 months, 6 months, 1 year)
- [] Display milestone markers or annotations for key achievements

Acceptance Criteria

- [] Users can toggle between different timeframes
- [] Graph updates dynamically based on available data
- [] Chart is responsive and visually consistent with the app theme

Labels

feature, visualization, analytics, enhancement, graph

Estimate

M

markdown

Title

Implement Weekly Summary Cards on Dashboard

Description

Introduce weekly summary cards to the AshTrail dashboard to give users a quick snapshot of their smoking habits each week. These cards will display total sessions, average session length, and comparison to the previous week.

Tasks

- [] Design card layout with consistent styling
- [] Calculate weekly metrics (total sessions, average length)
- [] Compare current week to previous week and show trend indicator
- [] Integrate cards into the dashboard UI

Acceptance Criteria

- [] Weekly cards show accurate data for each week
- [] Users can scroll back through past weeks
- [] Design is responsive and visually aligned with the app theme

Labels

feature, dashboard, enhancement, UX, metrics

Estimate

M

markdown

Title

Implement Rolling Averages for Session Metrics

Description

Add support for rolling average calculations to provide users with a more accurate sense of their recent smoking trends. These metrics will appear alongside daily and weekly summaries

and help users visualize changes over time.

Tasks

- [] Add logic to compute 7-day and 30-day rolling averages
- [] Display rolling averages in summary cards and charts
- [] Highlight deviations from averages with subtle UI cues (e.g., color changes)

Acceptance Criteria

- [] Rolling averages are correctly calculated based on timestamped session data
- [] Rolling average metrics appear in the dashboard alongside existing stats
- [] UI updates are non-intrusive and responsive

Labels

feature, analytics, enhancement, data-visualization

Estimate

M

markdown

Title

Add Inhalation Duration Histogram for Session Analysis

Description

Introduce a histogram to visualize the distribution of inhalation durations across all sessions. This will help users identify common inhalation lengths and outliers, offering deeper insights into their habits.

Tasks

- [] Calculate inhalation duration bins (e.g., 0-1s, 1-2s, etc.)
- [] Integrate a histogram chart into the analytics view
- [] Make the bin size configurable via a dropdown or slider

Acceptance Criteria

- [] Histogram accurately reflects distribution of all session inhalation durations
- [] Users can adjust bin size and see the chart update in real time
- [] Chart is responsive and visually consistent with the rest of the dashboard

Labels

feature, analytics, data-visualization, enhancement

Estimate

M

markdown

Title

Implement Session Comparison View for Habit Insights

Description

Create a new view that allows users to compare two or more selected smoking sessions side-by-side. This will help users identify patterns, improvements, or regressions in their habits.

Tasks

- [] Add UI for selecting multiple sessions from the session list
- [] Display selected sessions with aligned metrics: total puffs, average duration, session length, etc.
- [] Include comparative visuals such as bar charts or line overlays

Acceptance Criteria

- [] Users can select at least two sessions to compare
- [] Metrics are clearly displayed and differences are easy to interpret
- [] View adapts to various screen sizes and remains responsive

Labels

feature, UI, comparison, analytics

Estimate

L

markdown

Title

Add Quick Add Session Shortcut on Home Screen

Description

Implement a floating action button (FAB) or prominent shortcut on the home screen that allows users to quickly begin logging a new smoking session with a single tap. This is aimed at improving ease of access and encouraging more consistent tracking.

Tasks

- [] Design and implement FAB or shortcut for session start
- [] Configure shortcut to open the session logging view prefilled with current time
- [] Ensure shortcut does not interfere with other UI elements
- [] Animate button appearance for user guidance

Acceptance Criteria

- [] Shortcut is visible and accessible on home screen
- [] Tapping the shortcut reliably initiates a new session
- [] Feature works consistently across devices

Labels

feature, UX, enhancement

Estimate

M

markdown

Title

Implement Rolling Average Chart View for Usage Trends

Description

Add a chart visualization that shows rolling averages (e.g., 3-day, 7-day, 30-day) of smoking duration or frequency to help users better understand long-term trends. This will support insight-driven behavior changes and add depth to the analytical capabilities of the app.

Tasks

- [] Define rolling average periods in settings or fixed defaults
- [] Calculate rolling averages on filtered data
- [] Display rolling average line on the existing chart or as a toggle option
- [] Update chart legends and tooltips to reflect rolling data

Acceptance Criteria

- [] Users can toggle between raw data and rolling average views
- [] Rolling averages display accurately based on selected period
- [] Feature enhances visual interpretation without clutter

Labels

feature, data-visualization, enhancement

Estimate

M

markdown

Title

Add Tag-Based Log Filtering and Analysis

Description

Enable users to categorize smoke sessions with tags and filter analytics by selected tag(s). This feature improves personalized insights and allows more meaningful trend comparisons based on context (e.g., stress vs social smoking).

Tasks

- [] Add UI for adding/editing tags on logs (checkboxes or tag chips)
- [] Store tags in the log schema in Firestore/Isar
- [] Add filter control to analytics dashboard for tag selection
- [] Update charts and tables to reflect tag-filtered data
- [] Implement tag suggestion based on most frequently used tags

Acceptance Criteria

- [] Users can add one or more tags per session
- [] Filter dropdown allows tag-based view of data
- [] Analytics update properly when filtering by tag
- [] No impact on performance or data integrity

Labels

feature, analytics, UX, enhancement

Estimate

M

#1 Inhalation Duration Histogram

#2 Tag-Based Log Filtering and Analysis

⌚ Issue #2 delivered – next run in ~15 min.

markdown

Title

Implement Cumulative Inhalation Trend Visualization

Description

Create a line chart or area chart displaying cumulative

inhalation duration over time. This helps users visualize progress, track habits, and see the total time spent smoking across days, weeks, or months.

Tasks

- [] Create time-based aggregation function for cumulative inhalation
- [] Build interactive chart for visualizing cumulative trends
- [] Add date range selector to scope the view (e.g., week, month, all-time)
- [] Optimize performance for large datasets
- [] Ensure responsiveness across devices

Acceptance Criteria

- [] Cumulative inhalation is accurately aggregated and displayed
- [] Users can switch between time ranges
- [] Chart renders smoothly with large amounts of data

Labels

feature, visualization, analytics, enhancement

Estimate

L

#3 Cumulative Inhalation Trend Visualization

⌚ Issue #3 delivered – next run in ~15 min.

markdown

Title

Add Timestamped Data Table Below Visualizations

Description

Introduce a sortable, scrollable data table beneath the charts showing each inhalation log entry. This table will allow users to view detailed entries including timestamp, duration, and

potentially derived metrics, sorted in descending order by default.

Tasks

- [] Design responsive table layout for log entries
- [] Implement sorting functionality (default: descending by timestamp)
- [] Add pagination or infinite scroll for large datasets
- [] Optionally allow filtering by date range to match selected chart scope

Acceptance Criteria

- [] Table loads correctly with all current entries
- [] Entries appear in descending order by default
- [] Matches selected range when filtered

Labels

feature, ui, enhancement, table

Estimate

M

#4 Timestamped Data Table Under Charts

⌚ Issue #4 delivered – next run in ~15 min.

markdown

Title

Integrate Range Filter with Data Table and Chart Components

Description

Ensure the selected range filter (e.g., Last 7 Days, Last 30 Days, Custom Range) synchronizes with both the chart and the data table components. This allows users to analyze usage trends while viewing specific log entries within the same scope.

Tasks

- [] Connect range selector to both chart and data table
- [] Update data sources dynamically based on selected range
- [] Confirm default range applies consistently across all views
- [] Add loading state or transition animation for smoother UX

Acceptance Criteria

- [] Changing the range updates both the chart and table simultaneously
- [] Consistent results shown between visual and tabular data
- [] Smooth update experience without lag or misalignment

Labels

feature, sync, ui, enhancement, range-filter

Estimate

M

#5 Range Filter Sync with Chart and Table – now queued.

markdown

Title

Add Rolling Averages to Usage Chart

Description

Introduce rolling average calculations (e.g., 7-day, 30-day) to the usage chart to better visualize long-term trends and reduce noise from day-to-day fluctuations.

Tasks

- [] Implement 7-day and 30-day rolling average lines on the chart
- [] Add toggle controls for users to show/hide rolling averages
- [] Ensure responsiveness and legibility on various screen sizes
- [] Validate data consistency between raw and averaged views

Acceptance Criteria

- [] Users can toggle rolling averages on/off
- [] Rolling averages render correctly and update with range selection
- [] UI remains responsive and intuitive with added data layers

Labels

feature, chart, data-visualization, enhancement, rolling-average

Estimate

M

#6 Rolling Averages for Usage Chart – now queued.

markdown

Title

Implement Comparative Analytics View

Description

Develop a new analytics view that allows users to compare two date ranges side-by-side, enabling more meaningful insight into usage patterns over time.

Tasks

- [] Add UI controls for selecting a second date range
- [] Display dual bar or line charts for the selected ranges
- [] Highlight percentage increases/decreases between ranges
- [] Ensure accessibility and responsive design for mobile

Acceptance Criteria

- [] Users can successfully select and compare two custom date ranges
- [] Comparative data is displayed clearly and accurately
- [] Differences between ranges are visually highlighted

```
### Labels  
feature, analytics, comparison, enhancement, UX
```

```
### Estimate  
L
```

#7 Comparative Analytics View issue is now ready.

markdown

```
### Title  
Add Rolling Average Trend Line to Usage Chart
```

```
### Description  
Enhance the main usage chart by adding a rolling average trend line (e.g., 7-day or 30-day) to help users visualize longer-term patterns in their inhalation data.
```

```
### Tasks  
- [ ] Determine appropriate default rolling period (e.g., 7-day)  
- [ ] Implement rolling average calculation on backend or client-side  
- [ ] Overlay the trend line on the existing usage chart  
- [ ] Allow users to toggle the trend line on/off
```

```
### Acceptance Criteria  
- [ ] Users can see a smooth trend line overlaid on their usage chart  
- [ ] Trend line updates correctly with different date ranges  
- [ ] The toggle is clearly accessible and works on mobile and desktop
```

```
### Labels  
feature, data-visualization, enhancement, frontend
```

```
### Estimate  
M
```

markdown

Title

Implement Enhanced Filter Panel with Preset Ranges

Description

Create a redesigned filter panel that allows users to quickly filter data by commonly used preset ranges (e.g., Today, This Week, Last 7 Days, Last 30 Days, Custom). Improve UX with a modern, compact design optimized for both desktop and mobile.

Tasks

- [] Design a UI component with selectable presets
- [] Implement logic to handle each preset range
- [] Retain support for fully custom date range selection
- [] Integrate with existing chart data filtering logic

Acceptance Criteria

- [] Preset options update the chart data immediately upon selection
- [] Custom range selection remains functional and intuitive
- [] Design is responsive and visually consistent with overall UI theme

Labels

feature, frontend, UX, filters

Estimate

M

markdown

Title

Add Cumulative Usage Chart with Interactive Hover Tooltip

Description

Implement a new line chart that shows cumulative usage over time. This visualization will help users understand long-term

trends and track their total smoking over weeks, months, or years. Include an interactive tooltip that shows the total cumulative value and date when hovering.

Tasks

- [] Create cumulative usage calculation based on timestamped length data
- [] Design and implement the chart using Flutter's charting library
- [] Add interactive hover or tap tooltip showing total usage and date
- [] Ensure performance is smooth for large datasets

Acceptance Criteria

- [] Chart updates correctly with selected date range or filter
- [] Tooltip accurately reflects cumulative data at hovered point
- [] Works on both mobile and desktop resolutions

Labels

feature, visualization, chart, frontend, enhancement

Estimate

M

markdown

Title

Add Daily Summary Table Below Charts

Description

Introduce a new table view that lists each day's smoking data in descending timestamp order below the charts. Each row should display the date, total puffs, total duration, and average puff length for that day.

Tasks

- [] Aggregate daily usage data from timestamped entries

- [] Create a sortable table widget under the chart section
- [] Include columns: Date, Puff Count, Total Duration, Avg Duration
- [] Ensure responsiveness and mobile-friendliness

Acceptance Criteria

- [] Table appears directly beneath the charts
- [] Default sort order is newest date first
- [] Matches filtered date range and syncs with chart data

Labels

feature, table, frontend, data-visualization

Estimate

M

markdown

Title

Implement Rolling Averages Chart Toggle

Description

Allow users to toggle between raw inhalation data and a rolling average view to better visualize trends over time. Add a switch or dropdown in the chart control area.

Tasks

- [] Add toggle UI for raw vs. rolling average view
- [] Implement backend logic to compute rolling average (default 7-day window)
- [] Update chart rendering logic to switch datasets based on toggle
- [] Sync toggle state with date range filtering

Acceptance Criteria

- [] Toggle appears above chart and updates view on change
- [] Rolling average line is distinguishable (e.g., different color or dashed)
- [] Toggle works in both daily and hourly chart modes

Labels
feature, frontend, chart, data-visualization

Estimate
M

markdown

Title
Add Toggle for Duration vs. Frequency Chart View

Description
Provide users with the ability to switch between viewing the total inhalation duration and the frequency of inhalation events within the selected time range. This offers more context for interpreting usage patterns.

Tasks

- [] Add toggle control to switch between "Duration" and "Frequency"
- [] Implement logic to count events per interval for frequency mode
- [] Update chart labels and tooltip content accordingly
- [] Ensure toggle integrates with current date range and rolling average logic

Acceptance Criteria

- [] Toggle appears alongside existing chart controls
- [] Chart updates to reflect chosen metric (duration or frequency)
- [] Accurate axis labels and tooltips displayed for each mode

Labels
feature, frontend, chart, data-visualization

Estimate

M

markdown

Title

Implement Weekly and Monthly Aggregation Views

Description

Enhance the charting system by introducing aggregation views for "Weekly" and "Monthly" timeframes. These will help users understand longer-term trends in their inhalation behavior without overwhelming detail.

Tasks

- [] Add options for "Weekly" and "Monthly" views in the time range selector
- [] Aggregate inhalation data accordingly for each timeframe
- [] Update chart axes and tooltips to reflect aggregation level
- [] Ensure compatibility with both "Duration" and "Frequency" metrics

Acceptance Criteria

- [] Users can select "Weekly" or "Monthly" from the range selector
- [] Chart displays accurate summaries per week or month
- [] Tooltips and axis labels reflect correct periods

Labels

feature, frontend, data-aggregation, chart

Estimate

M

markdown

Title

Add Inhalation Type Filter to Data Visualizations

Description

Introduce a filtering option that allows users to select which types of inhalations (e.g., vape, joint, bong) to include in the visualizations. This enables more personalized and specific analysis of usage patterns.

Tasks

- [] Add dropdown or checkboxes to UI for selecting inhalation types
- [] Update backend query to respect selected filters
- [] Ensure charts update dynamically based on selected types
- [] Provide clear labels or legends to indicate which types are being shown

Acceptance Criteria

- [] User can filter data by one or multiple inhalation types
- [] Visualizations update accordingly and remain responsive
- [] Filtering logic is validated and tested

Labels

feature, frontend, backend, filters, chart

Estimate

M

markdown

Title

Implement Rolling Usage Average Chart

Description

Add a new chart option that displays a rolling average of inhalation usage over a selected timeframe (e.g., 7-day or 30-day). This helps users visualize long-term trends and identify gradual changes in behavior.

Tasks

- [] Add UI control to enable rolling average view
- [] Compute rolling average based on selected date range
- [] Display rolling average as a line chart overlay
- [] Ensure chart updates dynamically with date range or type filters

Acceptance Criteria

- [] Users can toggle rolling average view on and off
- [] Average line accurately reflects selected range and type
- [] All chart interactions remain smooth and functional

Labels

feature, visualization, enhancement, analytics

Estimate

M

markdown

Title

Integrate Static Sample Data Toggle for Dev Mode

Description

To assist with local development and testing, add a toggle that allows developers to switch between live Firebase data and static sample data. This will streamline testing new features without affecting production or dev databases.

Tasks

- [] Create static sample dataset mimicking real user log entries
- [] Implement toggle (e.g., boolean in code or UI setting) for switching data sources
- [] Update chart, table, and analysis components to respect data source toggle
- [] Clearly indicate in UI when static data is active (e.g., a banner or label)

```
### Acceptance Criteria
- [ ] Developers can activate static data mode via a clear mechanism
- [ ] All visualizations and logic respond accurately to sample data
- [ ] No Firebase reads occur when static data mode is enabled

### Labels
dev-tools, feature, enhancement

### Estimate
S
```

markdown

```
### Title
Add CSV Export Functionality for Log and Chart Data

### Description
To enhance usability and allow users to back up or analyze their data externally, implement CSV export functionality for both the log table and chart data views.

### Tasks
- [ ] Design a "Download CSV" button for the log table section
- [ ] Format timestamped entries and length values into a proper CSV structure
- [ ] Implement chart data export to CSV, ensuring selected time range is respected
- [ ] Ensure exported files use clear naming conventions and include date/time
```

```
### Acceptance Criteria
- [ ] User can download logs and chart data as CSV files
- [ ] CSV files contain appropriate headers and correctly formatted values
- [ ] Export respects current filters and selected date ranges
```

Labels

feature, enhancement, export

Estimate

M

markdown

Title

Implement Rolling Average Visualization for Usage Trends

Description

Add a feature to the chart section that enables users to visualize a rolling average of their inhalation data over time. This will help users identify trends and smooth out irregularities in their daily or weekly usage.

Tasks

- [] Add toggle switch to enable/disable rolling average overlay
- [] Calculate rolling average based on user-selected window size (e.g., 3, 7, 14 days)
- [] Display rolling average line alongside existing chart data
- [] Ensure visual distinction between actual data and average trendline

Acceptance Criteria

- [] Rolling average line accurately reflects selected window size
- [] Toggle allows easy enabling/disabling of trendline
- [] Works across different date ranges and chart types

Labels

feature, enhancement, visualization

Estimate

M

markdown

Title

Create UI for Adding Manual Notes to Inhalation Logs

Description

Users should be able to add personal notes or tags to individual inhalation logs. This allows for contextual insights, such as mood, setting, or events tied to specific sessions.

Tasks

- [] Add "Add Note" option to each inhalation entry in the list view
- [] Display existing notes inline or in a tooltip/popover
- [] Store notes alongside inhalation data in the Firestore database
- [] Ensure notes are editable and deletable

Acceptance Criteria

- [] Notes can be added, edited, and deleted per inhalation record
- [] Notes persist across sessions and devices
- [] Clean UI implementation with intuitive access

Labels

feature, ui, firestore, notes

Estimate

M

markdown

Title

Implement Inhalation Session Aggregation and Summary View

Description

Introduce a new view that groups inhalation events into

sessions based on proximity (e.g., within 10 minutes of each other). Display session summaries with total duration, number of puffs, and average inhale length.

Tasks

- [] Define logic for session grouping (e.g., 10-minute gap rule)
- [] Create session summary data model
- [] Build session summary UI with expandable detail view
- [] Update charting logic to optionally show sessions instead of individual logs

Acceptance Criteria

- [] Sessions accurately reflect grouped logs
- [] Summary view displays total puffs, duration, and average inhale length
- [] User can toggle between raw log view and session view

Labels

feature, ui, data-aggregation, chart

Estimate

L

markdown

Title

Add Rolling Daily Usage Trend Chart with Weekly and Monthly Averages

Description

Implement a new chart that shows a rolling daily average of inhalation sessions. Include toggles for users to switch between daily, weekly, and monthly trend lines. This helps visualize habits over time more clearly than raw usage data.

Tasks

- [] Calculate rolling daily averages from session data
- [] Add weekly and monthly average calculations

- [] Implement chart with toggle controls for each view
- [] Ensure visual clarity with appropriate smoothing and legends

Acceptance Criteria

- [] Chart updates based on selected time interval (daily, weekly, monthly)
- [] Data reflects rolling averages and aligns with session data
- [] Responsive UI on both mobile and desktop

Labels

feature, ui, chart, data-visualization

Estimate

M

markdown

Title

Implement Summary Statistics Card on Dashboard

Description

Add a card to the AshTrail dashboard that displays summary statistics for the selected time range. Metrics should include total inhalation time, average session duration, peak usage hour, and number of sessions.

Tasks

- [] Design and position summary statistics card
- [] Calculate total inhalation time and average session duration
- [] Identify peak usage hour based on timestamp data
- [] Display number of sessions in selected range
- [] Integrate with time range selector component

Acceptance Criteria

- [] Statistics update correctly when changing date range
- [] Values are clearly labeled and formatted

- [] Works on both desktop and mobile

Labels

feature, dashboard, ui, analytics

Estimate

M

markdown

Title

Add Weekly Trends Chart with Inhalation Totals

Description

Create a new section in the dashboard that visualizes weekly inhalation trends. The chart should group data by week and display total inhalation time per week using a bar or line graph format.

Tasks

- [] Aggregate inhalation data by week
- [] Design a responsive chart component (bar or line)
- [] Integrate with existing date filter
- [] Ensure chart updates when date range changes

Acceptance Criteria

- [] Chart renders correctly with accurate weekly totals
- [] Updates dynamically with selected date range
- [] Responsive layout on different screen sizes

Labels

feature, dashboard, analytics, visualization

Estimate

M

markdown

Title

Implement User Feedback Prompt on Log Submission

Description

Add a prompt for optional user feedback immediately after log submission. The goal is to gather insight into user experience, issues, or patterns in behavior. The feedback can include mood, reason for use, or notes.

Tasks

- [] Create UI for optional feedback after submitting a log
- [] Store feedback in Firestore alongside the log entry
- [] Update local Isar database schema to accommodate feedback
- [] Include feedback data in weekly and monthly review exports

Acceptance Criteria

- [] Prompt appears after log submission with fields for mood, reason, and notes
- [] Submitting feedback updates Firestore and Isar properly
- [] Feedback is accessible in data exports

Labels

feature, feedback, UX, firestore, isar

Estimate

M

markdown

Title

Add Toggle for Daily Reminders

Description

Introduce a toggle in settings to enable or disable daily reminders to log smoking activity. This will help users stay consistent without being intrusive for those who prefer not to

receive notifications.

Tasks

- [] Create toggle switch in settings page labeled “Daily Reminders”
- [] Save toggle preference in local Isar database
- [] Use Flutter local notifications to trigger a reminder at a user-defined time
- [] Disable notification scheduling when toggle is off

Acceptance Criteria

- [] Users can turn daily reminders on or off in settings
- [] Notification appears at configured time when enabled
- [] No notifications occur when toggle is off

Labels

feature, UX, notifications, settings, flutter

Estimate

M

markdown

Title

Implement Weekly Progress Summary View

Description

Create a dedicated section or modal that provides a summary of the user's smoking activity over the past week. This will help users visualize trends, track progress, and stay motivated.

Tasks

- [] Design and implement a new UI component for weekly summary
- [] Fetch and process data for the past 7 days from Isar database
- [] Display total puffs, average puffs per day, and longest streak without smoking
- [] Include a simple bar chart for daily usage comparison

- [] Add navigation link or button from home screen to open summary

Acceptance Criteria

- [] Users can view a clear, readable weekly summary
- [] Data is accurate and updates automatically as new logs are added
- [] The view is visually distinct and easy to interpret

Labels

feature, analytics, UI, flutter, data-visualization

Estimate

L

markdown

Title

Integrate Tooltip Overlays for Chart Data Points

Description

Enhance the usability of the charts by adding tooltip overlays that display detailed information (timestamp, puff length, etc.) when users tap or hover on data points. This improves user comprehension and provides immediate feedback on specific entries.

Tasks

- [] Implement tooltips on all relevant charts
- [] Display timestamp and puff length on hover/tap
- [] Ensure responsive behavior for both touch and mouse input
- [] Style tooltips to match the app's theme

Acceptance Criteria

- [] Tooltips appear accurately and without lag on chart interaction
- [] Data displayed in the tooltip matches the corresponding point

- [] Tooltips do not obstruct important chart elements

Labels

feature, UI, flutter, data-visualization

Estimate

M

markdown

Title

Create Session Summary View

Description

Develop a dedicated "Session Summary" screen that displays aggregated insights from a selected smoking session, such as total duration, number of puffs, average puff length, and notable spikes or trends. This screen will help users reflect on individual sessions and identify patterns over time.

Tasks

- [] Design UI layout for the session summary
- [] Implement logic to calculate summary statistics
- [] Highlight significant trends or outliers
- [] Include option to navigate back to the session log or chart

Acceptance Criteria

- [] Summary screen displays accurate, digestible metrics for the selected session
- [] Navigation to and from the summary is intuitive
- [] Visuals match the app's updated style and theme

Labels

feature, UX, summary, data-analysis

Estimate

M

markdown

Title

Implement Rolling Average Chart Overlay

Description

Add an optional overlay to the main usage chart showing a rolling average (e.g., 3-day, 7-day) to help users visualize trends over time. This will make it easier to spot changes in behavior without the noise of daily fluctuations.

Tasks

- [] Add UI toggle to enable/disable rolling average
- [] Calculate rolling average from timestamped data
- [] Plot the rolling average line on top of the existing chart
- [] Ensure responsiveness and readability on mobile devices

Acceptance Criteria

- [] Users can toggle a rolling average line on the main chart
- [] The rolling average updates based on selected range and chart data
- [] The overlay enhances understanding without cluttering the chart

Labels

enhancement, visualization, data-analysis, frontend

Estimate

M

markdown

Title

Integrate Cumulative Usage Visualization by Range

Description

Implement a chart view that shows total (cumulative) usage over a selected date range to give users a clear summary of their smoking behavior in that timeframe. This should visually differentiate between frequency and volume.

Tasks

- [] Add a new toggle/view option for “Total Usage”
- [] Aggregate data based on user-selected range (daily, weekly, monthly)
- [] Display cumulative usage bars or line over time
- [] Include labels/tooltips for exact usage totals per interval
- [] Ensure smooth transitions between modes (e.g., rolling average vs. total usage)

Acceptance Criteria

- [] Users can view total usage over a selected time period
- [] Chart clearly reflects usage totals per interval
- [] Feature integrates seamlessly with current range and filter options

Labels

feature, analytics, UX, chart

Estimate

M

markdown

Title

Add Editable Note Field to Each Log Entry

Description

Introduce a feature allowing users to attach personal notes to each smoking log entry. This helps track triggers, mood, or context around the session, enhancing self-awareness and journaling.

Tasks

- [] Add a `note` field to the Firestore log schema
- [] Update UI to display an “Add Note” or “Edit Note” option per log item
 - [] Store notes with each entry in Firestore
 - [] Support multiline text input and auto-save on change
 - [] Display notes clearly below or beside log entries in the table and detail view

Acceptance Criteria

- [] Users can view and edit a note for each log entry
- [] Notes are saved and synced correctly with Firestore
- [] UI for notes is unobtrusive but accessible

Labels

feature, firestore, UX, journal

Estimate

S

markdown

Title

Implement Log Detail Modal with Expanded Stats

Description

Create a modal that displays when a user taps on a specific smoking log entry. This modal should present expanded information beyond what's in the table view, such as average duration, time of day trends, and personal notes (if added).

Tasks

- [] Create a reusable modal component for log entry details
- [] Fetch and display all available fields from the selected log
 - [] Include additional analytics such as:
 - [] Rolling 7-day average duration
 - [] Common log times (morning/afternoon/evening)
 - [] Display user notes if they exist
 - [] Add a “Close” or “Back” button to return to the main

view

Acceptance Criteria

- [] Modal is triggered from table view or graph interaction
- [] Users see detailed metrics and can return to the main screen
- [] Modal adapts well to mobile and desktop layouts

Labels

feature, UI, stats, UX

Estimate

M

markdown

Title

Implement Rolling Average Chart Toggle

Description

Allow users to toggle between daily log data and a rolling average view on the chart. This feature will help users better understand their smoking trends over time by smoothing short-term fluctuations.

Tasks

- [] Add a toggle switch to the chart UI to enable/disable rolling average
- [] Implement 7-day rolling average logic in the chart data transformation
- [] Ensure both views (daily and rolling average) are visually distinct
- [] Adjust chart tooltip and legend to reflect current mode
- [] Write unit tests to verify correct data transformation

Acceptance Criteria

- [] Toggle switch successfully switches between views
- [] Rolling average updates dynamically with new data
- [] Charts remain responsive and accessible in both views

Labels
feature, chart, data-visualization, UX

Estimate
M

markdown

Title
Create Timestamp Range Selector with Presets

Description
Introduce a timestamp range selector that allows users to filter logs and visualizations based on a selected timeframe. Include preset options like "Today", "This Week", "This Month", and a custom date range.

Tasks

- [] Design and implement a timestamp range selector component
- [] Add preset options for common ranges (Today, This Week, etc.)
- [] Integrate selector with the chart and log data queries
- [] Update UI to reflect selected range
- [] Ensure compatibility with cumulative and rolling average views
- [] Write unit tests for range selection logic

Acceptance Criteria

- [] Users can easily switch between timeframes using preset or custom ranges
- [] Chart and logs update according to the selected range
- [] Selector is mobile-friendly and visually consistent

Labels
feature, UX, filter, date-range

Estimate

M

markdown

Title

Implement Rolling Average Chart Option

Description

Add the ability for users to view their smoking data using a rolling average. This will help smooth short-term fluctuations and highlight longer-term trends.

Tasks

- [] Add toggle or selector for switching between raw data and rolling average
- [] Calculate rolling average values based on selected timeframe (e.g., 7-day, 30-day)
- [] Update chart rendering to use smoothed values
- [] Display tooltip indicating the averaging window
- [] Ensure data points at the start of the range are handled gracefully
- [] Test rolling logic and edge cases

Acceptance Criteria

- [] Users can toggle between raw data and rolling averages
- [] Rolling average reflects appropriate smoothing
- [] UI clearly communicates that a rolling average is being shown

Labels

enhancement, visualization, data-analysis, chart

Estimate

M

markdown

Title

Enhance Entry Table with Sortable Columns and Filters

Description

Improve the usability of the smoking log data table by adding sortable columns and filters. This will allow users to more easily analyze and find specific entries.

Tasks

- [] Add ability to sort by timestamp, duration, and notes
- [] Implement filters for date ranges, duration thresholds, and keyword search in notes
- [] Update UI to clearly indicate sorting and filtering status
- [] Ensure sorting and filtering updates chart data accordingly
- [] Write unit and integration tests for sorting and filtering logic

Acceptance Criteria

- [] Users can sort data by clicking column headers
- [] Filters update the displayed table and corresponding chart
- [] Table updates are performant and accurate

Labels

enhancement, UI, data-table, filter, sort

Estimate

M

markdown

Title

Implement Data Export Functionality (CSV/JSON)

Description

Allow users to export their smoking log data for backup or

analysis purposes. Provide options to download the data in CSV or JSON format from the dashboard.

Tasks

- [] Add "Export" button to dashboard with CSV and JSON options
- [] Format data to match table contents, including timestamp, duration, and notes
- [] Ensure exported data respects any filters applied
- [] Implement frontend trigger and backend function for data generation
- [] Test export with various data sizes and filter states

Acceptance Criteria

- [] Users can download CSV and JSON files of their log data
- [] Exported files include all currently visible data in the table
- [] No errors occur during export

Labels

feature, data-export, UX, dashboard

Estimate

M

markdown

Title

Implement Session Summary Chart View

Description

Create a new chart view on the dashboard that summarizes smoking sessions over selected time ranges (e.g., daily, weekly, monthly). This visualization will help users identify trends in usage over time.

Tasks

- [] Design and implement a new chart component (bar or line chart)

- [] Aggregate data by time range (daily, weekly, monthly)
- [] Allow user to toggle between chart types and time ranges
- [] Integrate with existing range selector and filters
- [] Display relevant labels and tooltips for clarity

Acceptance Criteria

- [] Users can toggle between daily, weekly, and monthly views
- [] The chart reflects the current data filters and range
- [] Visual style matches existing dashboard design

Labels

feature, visualization, charting, dashboard

Estimate

M

markdown

Title

Create Rolling Average Usage Chart

Description

Introduce a new chart that shows a rolling average (e.g., 7-day, 30-day) of inhalation lengths over time. This helps users visualize longer-term trends and smooth out daily fluctuations in usage.

Tasks

- [] Calculate rolling average based on user-selected window (7, 14, 30 days)
- [] Display rolling average line overlaid on existing session length chart
- [] Add toggle control to switch between raw and averaged data
- [] Ensure it integrates with existing range and filtering logic
- [] Style to match current dashboard aesthetics

Acceptance Criteria

- [] Chart accurately reflects rolling averages for selected range
- [] User can toggle rolling average view on/off
- [] Data updates dynamically based on selected time range and filters

Labels

feature, analytics, visualization, enhancement

Estimate

M

markdown

Title

Implement Color-Coded Usage Intensity on Calendar View

Description

Enhance the calendar view by color-coding each day based on total inhalation duration or frequency. This visual heatmap will help users identify heavy and light usage days at a glance.

Tasks

- [] Define usage intensity levels (e.g., low, medium, high) based on configurable thresholds
- [] Map colors to intensity levels (e.g., green = low, yellow = medium, red = high)
- [] Update calendar component to apply background color for each day
- [] Ensure accessibility and dark mode compatibility

Acceptance Criteria

- [] Calendar days are visibly color-coded according to usage intensity
- [] Thresholds for intensity levels are configurable
- [] No regression in calendar functionality or performance

Labels

feature, UI/UX, enhancement, calendar

Estimate

M

markdown

Title

Add Rolling Average Chart for Weekly Usage Trends

Description

Introduce a new chart feature that displays a rolling 7-day average of inhalation duration or frequency. This will help users observe trends over time without being misled by daily spikes or drops.

Tasks

- [] Calculate a 7-day rolling average of inhalation durations/frequency
- [] Integrate the rolling average into the existing line chart or create a new chart
- [] Add toggle to switch between raw and rolling average views
- [] Style chart elements for clear differentiation between data types

Acceptance Criteria

- [] Users can view a rolling average trend over the selected date range
- [] Chart updates dynamically based on the selected date range and metric
- [] Feature performs efficiently and integrates with current UI design

Labels

feature, analytics, chart, enhancement

Estimate

M

markdown

Title

Implement Session Summaries with Time-of-Day Context

Description

Enhance the user experience by providing a summary of inhalation sessions categorized by time-of-day segments (e.g., morning, afternoon, evening, late night). This feature helps users understand their usage patterns in context and supports behavioral analysis.

Tasks

- [] Define time-of-day segments with configurable thresholds (e.g., 6am-12pm = morning)
- [] Group session data based on these segments
- [] Display summary metrics (count, average duration) per segment
- [] Add visual representation (e.g., pie chart or bar graph) to the dashboard
- [] Ensure integration with existing date range and filter selections

Acceptance Criteria

- [] Users can see aggregated session data by time-of-day
- [] Visuals clearly distinguish usage patterns across segments
- [] Works correctly across different time zones and date ranges

Labels

feature, data-visualization, enhancement

Estimate

M

markdown

Title

Add Filter for Weekday vs Weekend Usage Trends

Description

Provide users with the ability to filter inhalation data by weekday versus weekend. This feature will allow more insightful behavioral analysis by showing differences in usage patterns between workdays and days off.

Tasks

- [] Implement logic to determine if a date falls on a weekday or weekend
- [] Add filter toggle in the UI for "Weekday", "Weekend", and "All"
- [] Update charts and tables to reflect filtered data
- [] Validate filter works in combination with date range and time-of-day filters

Acceptance Criteria

- [] Users can filter data by weekday, weekend, or both
- [] Charts and summaries respond accurately to the selected filter
- [] Filter persists across session or until changed by user

Labels

feature, data-filter, usability

Estimate

S

markdown

Title

Enhance Summary Cards with Contextual Insights

Description

Improve the summary cards at the top of the dashboard by including contextual insights such as comparisons to previous periods, average inhalation length trends, and record usage streaks. These insights should help users better understand their behavior at a glance.

Tasks

- [] Add logic to calculate changes in usage compared to previous timeframes (e.g., week-over-week)
- [] Display average inhalation length and change from prior period
- [] Include record usage streak (longest streak of daily logs)
- [] Style cards to clearly differentiate metrics and contextual notes

Acceptance Criteria

- [] Summary cards reflect not just raw totals but also trends and records
- [] Cards update dynamically with date range and filter changes
- [] UI remains clean and readable across screen sizes

Labels

enhancement, UI, analytics

Estimate

M

markdown

Title

Implement Rolling Averages in Usage Charts

Description

Introduce rolling average lines (e.g., 7-day, 30-day) to the usage and inhalation charts. This will help users visualize trends over time by smoothing out day-to-day fluctuations.

Tasks

- [] Add rolling average calculations to backend or chart logic
- [] Allow user to toggle between different rolling average intervals (e.g., none, 7d, 30d)
- [] Style rolling average lines distinctly from raw data
- [] Ensure charts adjust correctly when filtering date ranges

Acceptance Criteria

- [] Users can view rolling averages overlaid on raw data
- [] Chart lines are clear and responsive to filters
- [] Rolling averages aid in identifying behavioral patterns

Labels

enhancement, visualization, analytics

Estimate

M

markdown

Title

Add Weekly and Monthly Summary Cards

Description

Introduce summary cards that display aggregated data for the current week and month. These cards should appear above or below the charts and provide quick insights such as total puffs, average daily usage, and longest streaks without smoking.

Tasks

- [] Compute weekly and monthly totals and averages
- [] Design compact summary card components
- [] Integrate the cards into the dashboard layout
- [] Ensure they update in real time with selected date ranges

Acceptance Criteria

- [] Users can clearly see weekly and monthly summaries
- [] Cards are responsive and visually aligned with the UI
- [] Summary data reflects current filters and data set

Labels

feature, UI, analytics

Estimate

M

markdown

Title

Implement Rolling Average Line on Usage Chart

Description

Enhance the usage chart by overlaying a rolling average line (e.g., 7-day or 14-day average) to help users identify trends in their smoking behavior. This feature will provide visual insight into whether usage is increasing or decreasing over time.

Tasks

- [] Add logic to compute rolling averages based on selected range
- [] Render the rolling average line on the existing usage chart
- [] Allow toggling the average line on and off
- [] Ensure responsiveness and clarity across screen sizes

Acceptance Criteria

- [] Users can view a smooth rolling average line overlaid on the usage chart
- [] Option to enable/disable the line is present and functional
- [] Line accurately reflects averaged data based on selection

Labels

feature, data-visualization, enhancement

Estimate

M

markdown

Title

Create Inhalation Intensity Heatmap

Description

Introduce a heatmap visualization that displays inhalation intensity throughout the day over a calendar or clock interface. This allows users to see patterns in usage behavior by time of day or specific dates.

Tasks

- [] Analyze existing inhalation data for time-of-day patterns
- [] Design a heatmap layout (calendar grid or radial clock format)
- [] Implement frontend chart with appropriate library (e.g., Chart.js, D3)
- [] Add toggle option for daily or weekly view
- [] Ensure accessibility and responsiveness

Acceptance Criteria

- [] Heatmap is visible and clearly indicates usage intensity
- [] Users can switch between daily and weekly views
- [] Responsive design supports desktop and mobile

Labels

feature, data-visualization, enhancement, frontend

Estimate

L

markdown

Title

Implement Rolling Averages Chart for Weekly and Monthly Trends

Description

Add a new line chart visualization that overlays rolling averages (e.g., 7-day and 30-day) of inhalation frequency and intensity. This feature helps users observe long-term trends in their smoking habits and detect subtle behavior changes over time.

Tasks

- [] Aggregate inhalation data to compute 7-day and 30-day rolling averages
- [] Create a new chart component to display these trends
- [] Integrate with existing filtering options for date ranges
- [] Add legend and tooltip for rolling average lines
- [] Ensure chart responsiveness and theme consistency

Acceptance Criteria

- [] Rolling average lines render correctly and update with filters
- [] Users can distinguish between daily values and trend lines
- [] Performance remains optimal with extended data ranges

Labels

feature, data-visualization, enhancement, frontend

Estimate

L

markdown

Title

Add Daily Usage Summary with Color-Coded Indicators

Description

Create a new dashboard widget that provides a quick summary of smoking activity for the current day, including total sessions, average inhalation length, and comparison to previous days. Use color-coded indicators (green/yellow/red) to reflect changes in usage trends.

Tasks

- [] Implement logic to calculate daily usage metrics
- [] Design a compact card-style widget to display these metrics
- [] Add color-coded status (e.g., green if usage decreased, red if increased)
- [] Ensure widget updates dynamically with new data entries
- [] Add unit tests for summary calculations

Acceptance Criteria

- [] Summary appears on the main dashboard and updates with new entries
- [] Colors change based on comparison with previous day's usage
- [] UI is consistent with app design and responsive

Labels

feature, UI, frontend, enhancement

Estimate

M

markdown

Title

Enable Export of Smoking Log Data to CSV

Description

Add functionality that allows users to export their smoking session data as a CSV file. This will enable users to analyze their logs offline or import them into other tools.

Tasks

- [] Create an export button in the main UI
- [] Define CSV structure: timestamp, duration, notes (if any)
 - [] Implement backend logic to gather and format data
 - [] Trigger download of CSV file upon user request
 - [] Validate CSV content with test entries

Acceptance Criteria

- [] Users can download a CSV file with all their session data
- [] CSV includes timestamp and inhalation duration for each entry
- [] File downloads successfully and matches expected format

Labels

feature, backend, frontend, export

Estimate

M

markdown

Title

Implement Daily and Weekly Summary View for Smoke Log

Description

Add a new feature that displays daily and weekly summaries of smoking sessions. This view will help users visualize their habits and spot trends over time.

Tasks

- [] Create a summary section with toggle options (Daily / Weekly)
- [] Calculate total inhalation duration and session count for each day/week
- [] Implement bar or line charts to visualize trends
- [] Add average duration per session to summaries
- [] Ensure summary updates with new data in real-time

Acceptance Criteria

- [] User can switch between daily and weekly summaries
- [] Summaries show accurate totals and averages
- [] Data visualizations render correctly and match underlying data

Labels

feature, analytics, UI, enhancement

Estimate

L

markdown

Title

Add User Settings for Custom Session Goals and Notifications

Description

Introduce a user settings panel that allows individuals to set daily session goals and receive optional reminders. This feature supports habit awareness and self-regulation.

Tasks

- [] Create a user settings screen accessible from the main interface
- [] Allow users to input daily inhalation duration goals and session count targets
- [] Implement logic to compare logged data against goals
- [] Enable toggleable notifications/reminders when nearing or exceeding set goals
- [] Persist settings across sessions using local storage or Firestore

Acceptance Criteria

- [] User can save and update their goals and notification preferences
- [] System accurately detects goal progress and triggers alerts when necessary

- [] Settings persist across app sessions

Labels

feature, UX, notification, enhancement

Estimate

M

markdown

Title

Implement Rolling Averages for Inhalation Duration Analytics

Description

Enhance data insights by adding rolling average charts (e.g., 7-day, 30-day) to visualize trends in inhalation duration over time. This feature will help users detect gradual increases or decreases in usage habits.

Tasks

- [] Create chart toggle or dropdown to switch between raw data and rolling averages
- [] Implement backend logic or client-side calculations for 7-day and 30-day moving averages
- [] Display rolling average data on the existing line chart using distinct styling
- [] Update chart legends and tooltips to reflect average type

Acceptance Criteria

- [] Users can switch between raw and average views
- [] Rolling averages accurately reflect historical trends
- [] Visual elements clearly distinguish between raw and averaged data

Labels

data-visualization, analytics, enhancement, chart

Estimate

M

markdown

Title

Add Weekly Summary View with Key Metrics and Highlights

Description

Create a dedicated weekly summary view that displays key usage metrics such as total inhalations, average length, and longest/shortest sessions. Highlight changes from the previous week to help users understand their habits.

Tasks

- [] Design UI section for weekly summary cards
- [] Calculate weekly totals and averages from log data
- [] Show comparisons to the prior week (e.g., percentage increase or decrease)
- [] Add icons or color indicators for intuitive change visualization
- [] Write unit tests for summary calculations

Acceptance Criteria

- [] Summary shows accurate weekly metrics and changes
- [] Display is user-friendly and informative
- [] Data updates automatically each new week

Labels

analytics, feature, UX, enhancement

Estimate

M

markdown

Title

Implement Chart Range Selector with Data Filtering

Description

Add a chart range selector to allow users to filter inhalation data based on time ranges (e.g., last 7 days, 30 days, custom date range). Ensure the displayed data, stats, and charts update accordingly for more insightful analysis.

Tasks

- [] Design and implement a dropdown or slider for selecting time ranges
- [] Filter the data based on selected range before passing it to visualizations
- [] Update summaries, averages, and charts based on the filtered data
- [] Retain selection across sessions using local storage or similar method
- [] Write tests to validate filtering logic

Acceptance Criteria

- [] Users can select from preset or custom time ranges
- [] Charts and statistics reflect only the data within the selected range
- [] Selection persists when returning to the app

Labels

feature, UX, data-visualization, filtering

Estimate

M

markdown

Title

Implement Rolling Average Analysis for Inhalation Length

Description

Enhance the analytical capabilities of the AshTrail (Smoke-Log) dashboard by adding support for rolling average calculations over time. This will help users better understand

trends in inhalation length by smoothing short-term fluctuations.

Tasks

- [] Add a toggle or option for enabling rolling average on the main chart
- [] Implement rolling average calculation (e.g., 7-day, 14-day, custom)
- [] Integrate rolling average line into the inhalation chart
- [] Label the chart clearly to distinguish raw data vs. rolling average
- [] Add tests for rolling average logic

Acceptance Criteria

- [] Users can view rolling average trends on the chart
- [] The feature can be enabled or disabled dynamically
- [] Rolling averages adjust correctly with filtered data

Labels

enhancement, data-visualization, analysis, charts

Estimate

M

markdown

Title

Add Visual Feedback for Data Upload and Syncing

Description

Implement clear visual indicators for when data is being uploaded or synced in AshTrail. This helps users understand when their data is processing and prevents confusion or duplicate submissions.

Tasks

- [] Add a loading spinner or progress indicator during data submission
- [] Display success and error messages after upload

completion

- [] Ensure UI is responsive and prevents duplicate submissions during syncing
- [] Add tests for submission states and transitions

Acceptance Criteria

- [] Users receive immediate feedback when uploading data
- [] System handles and displays errors clearly if upload fails
- [] Feedback disappears or updates once data is synced

Labels

UI/UX, enhancement, syncing, feedback

Estimate

S

markdown

Title

Implement Time Range Selector for Chart Filtering

Description

Add a time range selector component to allow users to filter data visualizations by predefined or custom date ranges. This will improve usability and performance by limiting the volume of data shown.

Tasks

- [] Design UI component for time range selection (e.g., last 7 days, 30 days, custom range)
- [] Integrate with existing chart rendering logic to filter data accordingly
- [] Ensure selected range updates charts and associated statistics
- [] Add validation to handle edge cases (e.g., no data in selected range)

Acceptance Criteria

- [] Users can select from common ranges or define a custom range
- [] Charts and statistics update based on the selected range
- [] Filtering is efficient and does not cause UI lag

Labels

feature, UI/UX, charts, filtering

Estimate

M

markdown

Title

Implement Inhalation Log Table with Sorting and Filtering

Description

Develop a table component below the charts to display each inhalation log entry. The table should support sorting by timestamp (default: descending) and basic filtering options, such as by duration range or time period.

Tasks

- [] Design table UI to align with current layout and theme
- [] Display columns: Timestamp, Duration, (optional tags if applicable)
- [] Implement sorting by timestamp (descending by default)
- [] Add filter options for time range and inhalation length
- [] Ensure integration with the range selector and chart data

Acceptance Criteria

- [] Table shows data accurately with recent entries at the top
- [] Sorting and filtering functionalities work as expected
- [] Filtering updates both table and chart views for consistency

Labels

feature, UI/UX, data-table, filtering

Estimate

M

markdown

Title

Refactor Chart Component to Support Multiple Data Views

Description

Update the existing chart component to support toggling between different data views, such as daily totals, cumulative inhalation time, and average session length. This will improve the flexibility of analysis and user insight.

Tasks

- [] Modify chart component to accept a dynamic data transformation function
- [] Implement toggles or tabs for selecting the data view
- [] Create helper functions for computing:
 - Daily totals
 - Cumulative usage
 - Average duration per session
- [] Ensure all transformations work with filtered date ranges
- [] Add basic documentation or tooltip help to explain each chart type

Acceptance Criteria

- [] Users can switch between data views without page reloads
- [] All views render correctly with accurate data
- [] Transitions between views are smooth and maintain selected date range

Labels

feature, data-visualization, enhancement

Estimate

L

markdown

Title

Implement Advanced Session Insights Panel

Description

Introduce a new panel in the app that displays advanced insights into smoking sessions, including statistics like longest session, shortest session, average duration, and peak usage times throughout the day.

Tasks

- [] Design layout for insights panel (card/grid format)
- [] Compute session statistics:
 - Longest session (duration and timestamp)
 - Shortest session (duration and timestamp)
 - Average session duration
 - Time of day with most sessions
- [] Display insights in a visually appealing format with clear labeling
- [] Add a brief description or tooltip for each metric

Acceptance Criteria

- [] All metrics are calculated correctly and updated with selected date range
- [] UI design is responsive and consistent with overall app theme
- [] Insights panel is accessible and performs well with large datasets

Labels

feature, analytics, UI/UX

Estimate

M

markdown

Title

Enhance Log Table with Inline Editing and Save Feedback

Description

Allow users to edit smoking log entries directly from the log table without navigating to a new screen. Implement inline editing with real-time validation and a save button. Provide visual feedback upon successful or failed saves.

Tasks

- [] Add edit icon/button to each row in the log table
- [] Enable fields (timestamp, duration, type) to be editable inline
- [] Implement validation for each field (e.g., duration must be positive)
- [] Add save and cancel actions per row
- [] Display success or error toast notifications upon save

Acceptance Criteria

- [] Users can update log entries without leaving the table
- [] Input validations prevent invalid data from being saved
- [] Changes persist and reflect immediately in UI
- [] Users receive clear feedback for each save action

Labels

feature, UI/UX, enhancement

Estimate

M

markdown

Title

Implement Animated Chart Transitions for Smoke-Log Insights

Description

Enhance user experience by adding smooth animations to the charts that display inhalation frequency and duration. Animations should trigger when the user switches time ranges, applies filters, or adds new entries.

Tasks

- [] Identify all chart components impacted by user interaction
- [] Integrate chart animation library or enable native animation support (e.g., in Recharts or Chart.js)
- [] Apply entry/exit and transition animations to data changes
- [] Test animations across devices and screen sizes for performance
- [] Add toggle setting to enable/disable chart animations (optional)

Acceptance Criteria

- [] Chart transitions animate fluidly during filter or data changes
- [] No lag or jitter during animations, even on older devices
- [] Users receive a more engaging visual experience without distraction

Labels

feature, UI/UX, charts, enhancement

Estimate

M

markdown

Title

Integrate Rolling Average Selector for Inhalation Trends

Description

Add a user-controlled rolling average option to smooth inhalation data on visualizations. This will allow users to

better identify trends in their habits by reducing noise in the raw data.

Tasks

- [] Add UI element (e.g., dropdown or toggle) to select rolling average window (e.g., None, 3-day, 7-day, 30-day)
- [] Implement backend or frontend logic to compute rolling averages based on selected window
- [] Update chart rendering logic to switch between raw and smoothed data
- [] Ensure rolling average does not distort timestamp alignment on charts
- [] Write unit tests for rolling average computation

Acceptance Criteria

- [] Users can choose a rolling average window for visualized data
- [] Charts update to reflect the smoothed values accordingly
- [] Visuals clearly indicate when data is averaged (e.g., dashed line or legend)

Labels

feature, data-visualization, charts, enhancement

Estimate

M

markdown

Title

Create Insights Tab to Display Personal Smoking Trends

Description

Design and implement an "Insights" tab in the UI where users can view summarized trends and key stats about their smoking habits over time. This feature will provide a centralized view for analyzing usage patterns and progress.

Tasks

- [] Create a new tab or section labeled "Insights"
- [] Display high-level metrics (e.g., total inhalations, average per day/week/month, longest streak without smoking)
 - [] Add visuals for usage streaks and most common times of day
- [] Ensure date filters apply consistently across the entire Insights tab
- [] Style consistently with the overall design system

Acceptance Criteria

- [] Users can view aggregated trends and stats in one place
- [] Data updates dynamically based on date range selector
- [] Visuals and layout are clear and accessible

Labels

feature, enhancement, analytics, UI

Estimate

L

markdown

Title

Implement Cumulative Usage Chart with Time Range Filters

Description

Add a cumulative usage chart to the main dashboard that allows users to visualize total inhalation length over a selected period. This feature helps track usage trends and milestones over time.

Tasks

- [] Integrate line chart to represent cumulative inhalation length
- [] Sync chart data with existing date range selectors
- [] Add toggle for cumulative vs. daily/weekly breakdown
- [] Display tooltips showing exact cumulative values on hover
- [] Ensure responsiveness across devices

Acceptance Criteria

- [] Users can clearly see cumulative inhalation growth over time
- [] Chart responds accurately to selected date range
- [] Toggle between cumulative and other views functions smoothly

Labels

feature, enhancement, chart, UI, data-visualization

Estimate

M

markdown

Title

Create Inhalation Session Summary View

Description

Design and implement a summary view that groups inhalation entries into distinct sessions based on time gaps (e.g., gaps of 5 minutes or more). This allows users to review their usage habits in a more meaningful context by viewing total sessions instead of individual inhales.

Tasks

- [] Define logic to group inhalation records into sessions
- [] Create summary cards showing start time, end time, duration, and total usage per session
- [] Add navigation or dropdown to select sessions from a list
- [] Integrate with the date range selector for filtering
- [] Style session summaries to fit the app's design language

Acceptance Criteria

- [] Sessions are correctly grouped and listed
- [] Summary metrics are accurate and clear
- [] Works seamlessly with time range filters

Labels
feature, enhancement, analytics, UI, smoke-log

Estimate
L

markdown

Title
Implement Chart Data Export Feature

Description
Allow users to export the visualized inhalation data and session summaries as a CSV or JSON file. This enables deeper offline analysis or sharing of their usage patterns with healthcare providers.

Tasks

- [] Add export buttons for CSV and JSON formats
- [] Support exporting filtered data based on selected date range
- [] Include both raw inhale entries and session summaries in the export
- [] Style the export options in line with the app's UI

Acceptance Criteria

- [] Users can download CSV/JSON files of their data
- [] Files contain all expected fields and respect active filters
- [] Export functions work on both mobile and desktop

Labels
feature, enhancement, data-export, smoke-log

Estimate
M

markdown

Title

Implement Timeline View for Daily Sessions

Description

Introduce a horizontal timeline view that visually displays all smoking sessions throughout a selected day. This will help users quickly identify patterns and gaps in usage.

Tasks

- [] Design and integrate a horizontal timeline widget
- [] Display session start/end times and durations proportionally
- [] Allow users to scroll through hours of the day
- [] Sync timeline data with current filters and selected date
- [] Add tooltips or hover info for detailed session data

Acceptance Criteria

- [] Timeline renders correctly for any selected day
- [] Sessions are spaced and sized according to time and length
- [] Timeline updates when date or filters change

Labels

feature, visualization, timeline, smoke-log

Estimate

L

markdown

Title

Add Weekly Overview Chart with Trends and Averages

Description

Develop a new chart component that displays a weekly overview of smoking behavior. This chart should highlight trends such

as increases or decreases in usage and show averages per day. It should also allow users to compare current week data with previous weeks.

Tasks

- [] Create a weekly overview chart (e.g., bar or line chart)
- [] Display total duration per day for the current week
- [] Overlay or switch to previous week for comparison
- [] Add tooltips for per-day statistics
- [] Ensure responsive layout across devices

Acceptance Criteria

- [] Users can view clear trends across a 7-day span
- [] Historical comparison is available and toggleable
- [] Averages and daily totals are calculated and displayed correctly

Labels

feature, visualization, chart, weekly-analysis

Estimate

L

markdown

Title

Integrate Rolling Average Line in Daily Duration Chart

Description

Enhance the daily smoking duration chart by overlaying a 7-day rolling average line. This will help users identify long-term patterns in their behavior and highlight whether their usage is increasing or decreasing over time.

Tasks

- [] Calculate 7-day rolling average from logged durations
- [] Overlay the rolling average as a smooth line on the daily chart
- [] Add legend and tooltip explanations for both actual data

and average

- [] Ensure performance remains optimal with long datasets

Acceptance Criteria

- [] Rolling average line accurately reflects the 7-day trend
- [] Line is visually distinguishable from daily duration bars
- [] Tooltips display correct values for both datasets

Labels

enhancement, chart, data-visualization, user-insight

Estimate

M

markdown

Title

Add "Total Usage Time" Summary Card to Dashboard

Description

Create a summary card at the top of the dashboard showing the user's total smoking duration over a selected time range. This offers an at-a-glance insight into cumulative usage, enhancing self-awareness and promoting tracking progress.

Tasks

- [] Compute the sum of durations for the currently selected date range
- [] Design and implement a summary card component styled consistently with the dashboard
- [] Display total usage time in a readable format (e.g., "3h 45m")
- [] Ensure card updates dynamically with range selector changes

Acceptance Criteria

- [] Card accurately reflects total usage within the selected date range

- [] Card updates correctly on range selector input
- [] Card styling aligns with the rest of the dashboard UI

Labels

feature, dashboard, UX, enhancement

Estimate

S

markdown

Title

Implement Weekly & Monthly Chart Toggle for Usage
Visualization

Description

Introduce a toggle feature on the dashboard to switch between weekly and monthly views for usage charts. This gives users flexibility to track trends over different time scales, enhancing insight into behavioral patterns.

Tasks

- [] Add a UI toggle component (e.g., segmented control or dropdown) with options "Weekly" and "Monthly"
- [] Update chart logic to group and display usage data accordingly
- [] Ensure cumulative and rolling data summaries adapt to the selected view
- [] Maintain smooth transitions and consistent chart styles

Acceptance Criteria

- [] Toggle allows switching between weekly and monthly chart views
- [] Charts reflect the selected grouping correctly
- [] Visual design and interaction are consistent with existing dashboard elements

Labels

feature, dashboard, charts, UX

Estimate

M

markdown

Title

Enhance Dashboard Date Range Picker with Custom Range Selection

Description

Improve the dashboard's date filtering by enabling users to define a custom date range. This allows more precise data analysis beyond predefined time frames.

Tasks

- [] Replace or extend existing date selector to support a custom date range
- [] Ensure charts, summaries, and data tables respect the selected range
- [] Validate inputs to prevent invalid or future dates
- [] Store and recall user's last selected range for a consistent experience

Acceptance Criteria

- [] Users can choose any start and end date
- [] Data updates dynamically based on selected range
- [] Range persists during the session or via local storage

Labels

feature, dashboard, filters, UX

Estimate

M

markdown

Title

Implement Toggle Between Raw and Cumulative Inhalation Charts

Description

Allow users to switch between viewing raw inhalation data and a cumulative sum over time on the main dashboard chart. This helps visualize trends and consumption patterns more clearly.

Tasks

- [] Add a toggle UI control above the chart for "Raw" vs. "Cumulative"
- [] Adjust chart logic to render based on toggle state
- [] Preserve toggle state in local storage or session
- [] Ensure cumulative chart calculates sum correctly for selected date range

Acceptance Criteria

- [] Toggle is visible and functional
- [] Chart switches between raw and cumulative views
- [] Toggle state is remembered across reloads
- [] Performance remains smooth for both views

Labels

feature, chart, UX, data-visualization

Estimate

M

markdown

Title

Create Session Summary Card for Daily Use Overview

Description

Design and implement a session summary card to display key metrics for each day, such as total sessions, average duration, and total time inhaled. This will appear above the chart or logs list to provide at-a-glance insights.

Tasks

- [] Design a compact session summary component
- [] Extract and calculate daily totals from log data
- [] Display metrics: total sessions, average inhalation length, cumulative duration
- [] Integrate component into main dashboard view
- [] Ensure responsiveness on mobile and desktop

Acceptance Criteria

- [] Card displays accurate daily metrics
- [] Updates dynamically with date selection
- [] Style is consistent with app theme

Labels

feature, dashboard, data-summary, UI/UX

Estimate

M

markdown

Title

Implement Time-of-Day Heatmap for Smoking Patterns

Description

Create a heatmap visualization that displays inhalation frequency and intensity based on time-of-day across a selected date range. This feature will help users identify peak smoking hours and patterns over time.

Tasks

- [] Design a heatmap UI component showing hours (x-axis) vs. dates (y-axis)
- [] Aggregate inhalation data by hour and date
- [] Normalize intensity to reflect session duration or count
- [] Integrate with date range filter for dynamic updates
- [] Add tooltip support for detailed hour-by-hour stats

Acceptance Criteria

- [] Heatmap accurately visualizes intensity per hour per day
- [] User can hover to see specific values
- [] Works with range selection and updates in real-time
- [] Matches app theme and is responsive

Labels

feature, visualization, heatmap, analytics, UI/UX

Estimate

L

markdown

Title

Create "Rolling Usage" Chart with Customizable Time Windows

Description

Introduce a "Rolling Usage" line chart that calculates and displays the average inhalation length over a user-defined rolling window (e.g., 3 days, 7 days, 30 days). This chart will provide insight into longer-term trends and help users understand fluctuations in their smoking behavior.

Tasks

- [] Add input controls to select rolling window (3d, 7d, 30d, custom)
- [] Calculate rolling average of inhalation length by timestamp
- [] Display rolling usage as a line chart over time
- [] Sync with existing range filter logic
- [] Ensure chart responsiveness and styling consistency

Acceptance Criteria

- [] User can choose a rolling window and see corresponding trend
- [] Chart updates dynamically with selected date range
- [] Tooltip shows precise rolling averages on hover
- [] Visual style matches existing charts

Labels
feature, analytics, chart, rolling-average, enhancement

Estimate
M

markdown

Title
Add Total and Average Inhalation Metrics to Dashboard

Description
Enhance the main dashboard by including summary statistics: total inhalation count, total inhalation time, and average inhalation length over the selected time range. These quick metrics will help users assess their smoking patterns at a glance.

Tasks

- [] Create a new summary metrics component
- [] Display total inhalations within selected date range
- [] Display total inhalation time (sum of all inhalation lengths)
- [] Display average inhalation length
- [] Integrate with existing date range filter
- [] Ensure responsive layout on different screen sizes

Acceptance Criteria

- [] Summary metrics are updated correctly based on selected date range
- [] Displayed prominently on the dashboard
- [] Matches visual style of the rest of the app

Labels
feature, UI, metrics, enhancement, dashboard

Estimate

S

markdown

Title

Implement Rolling Average and Trend Line on Inhalation Chart

Description

Introduce a rolling average (e.g., 7-day or 30-day) and an optional trend line overlay to the inhalation frequency and duration charts. This will help users visually identify patterns and changes over time.

Tasks

- [] Add toggle option for rolling average and trend line display
- [] Implement 7-day and 30-day rolling average calculations
- [] Overlay trend line using linear regression or similar method
- [] Ensure charts update with selected date range and filters
- [] Style chart overlays to distinguish from raw data

Acceptance Criteria

- [] User can toggle rolling average and trend line on/off
- [] Rolling average and trend line match filtered data range
- [] Charts remain performant with overlays enabled

Labels

feature, charts, enhancement, data-visualization, UI

Estimate

M

markdown

Title

Add Daily Usage Summary Card with Totals and Comparisons

Description

Design and implement a daily summary card to appear above the inhalation charts. This card should include:

- Total inhalation time for the selected day
- Total number of inhalations
- Average inhalation length
- Optional: comparison to previous day or 7-day average

This summary will give users quick insight into their daily usage habits without needing to analyze the full chart.

Tasks

- [] Create UI component for summary card
- [] Calculate and display total inhalation time
- [] Display count of inhalation events
- [] Show average inhalation duration
- [] Add optional comparison indicators (e.g., % change vs. yesterday or weekly avg)

Acceptance Criteria

- [] Summary card dynamically updates with selected date range
- [] All values correctly reflect the data shown in the chart
- [] Optional comparisons toggle on/off without errors

Labels

feature, UI, enhancement, analytics, summary

Estimate

S

markdown

Title

Implement Tab Navigation for Log View, Trends, and Settings

Description

Introduce a tabbed navigation UI to switch between the primary views of the AshTrail app:

- ****Log View**:** Displays all recorded inhalation data in table or chart form
- ****Trends**:** Shows rolling averages, usage summaries, and visual analytics
- ****Settings**:** Allows customization of display preferences, development mode toggles, and Firestore collection selection

This will organize the app more intuitively and prepare it for future feature expansion.

Tasks

- [] Add persistent bottom or top tab navigation bar
- [] Create individual components/pages for Log View, Trends, and Settings
- [] Implement routing or state management for switching tabs
- [] Ensure data and state persist appropriately across tab switches

Acceptance Criteria

- [] Users can switch between Log View, Trends, and Settings
- [] Tab bar is responsive and functional across devices
- [] No performance issues or data loss when switching views

Labels

feature, UI, navigation, enhancement

Estimate

M

markdown

Title

Add User Authentication Gate for Production Mode

Description

Implement a simple user authentication mechanism for production mode to ensure that only authorized users can submit or view logs. In development mode, bypass authentication for rapid iteration.

Tasks

- [] Create login UI screen with Google sign-in using Firebase Authentication
- [] Store user session locally and persist login across app restarts
- [] Update Firestore data fetches to be user-specific in production
- [] Add switch or detection to bypass login flow in development mode
- [] Provide sign-out functionality in Settings tab

Acceptance Criteria

- [] Only logged-in users can access and submit log data in production
- [] In development mode, authentication is not required
- [] Switching between development and production modes updates access control correctly

Labels

auth, security, enhancement, production

Estimate

M

markdown

Title

Integrate Cumulative Usage Line Chart in Dashboard

Description

Add a line chart to the main dashboard that displays cumulative smoking session length over time. This will help users visually understand their usage trends, spot spikes, and monitor changes over days, weeks, or months.

Tasks

- [] Query and aggregate session lengths from Firestore by date
- [] Compute cumulative totals from the aggregated data
- [] Render a line chart using a suitable charting package
- [] Add date range selector to control the displayed time window (e.g., 7d, 30d, All Time)
- [] Format chart axes for readability

Acceptance Criteria

- [] Chart loads correctly with real data
- [] User can filter by time range
- [] Axes are clearly labeled and visually balanced

Labels

dashboard, data-visualization, enhancement

Estimate

M

markdown

Title

Implement Daily Summary Tile on Dashboard

Description

Introduce a new tile on the dashboard showing a quick summary of today's smoking activity. This includes total inhalations, total time, and average session length.

Tasks

- [] Fetch and compute today's smoking data from Firestore
- [] Calculate:
 - Total number of inhalations
 - Total duration (in minutes)
 - Average session length
- [] Design and style a summary tile for clear visibility
- [] Add refresh/update logic for real-time updates

Acceptance Criteria

- [] Summary tile displays updated daily values
- [] UI is visually aligned with the dashboard theme
- [] All metrics are accurate and easy to read

Labels
dashboard, UI, enhancement, data

Estimate
M

markdown

Title
Create Historical Trends Chart for Weekly and Monthly Analysis

Description
Design and implement a new chart component that visualizes weekly and monthly trends in smoking behavior. This chart will help users identify long-term patterns and progress.

Tasks

- [] Aggregate data weekly and monthly from Firestore
- [] Calculate total inhalations and duration per period
- [] Implement toggles for switching between weekly/monthly views
- [] Ensure responsive design and chart readability

Acceptance Criteria

- [] Chart correctly updates when toggling between weekly and monthly
- [] Visual data reflects accurate aggregation
- [] Chart fits seamlessly within the dashboard layout

Labels
chart, enhancement, data-visualization, UI

Estimate

M

markdown

Title

Add Editable Notes to Log Entries

Description

Implement a feature that allows users to add or edit personal notes on each inhalation log entry. These notes can help track thoughts, triggers, or context behind entries.

Tasks

- [] Update Firestore schema to support optional `notes` field in log entries
- [] Modify log entry submission form to include a notes input
- [] Display existing notes alongside each entry in the log table
- [] Enable editing and saving notes for past entries

Acceptance Criteria

- [] Notes are saved and retrieved correctly for each entry
- [] Editing a note updates Firestore in real-time or on save
- [] Notes appear cleanly formatted in log and chart views

Labels

feature, enhancement, user-feedback, UI

Estimate

S

markdown

Title

Implement Weekly Summary Chart View

Description

Add a new chart view that aggregates and displays smoking activity over weekly intervals. This view should provide insights into trends and usage patterns by week.

Tasks

- [] Add a toggle or dropdown to switch between daily and weekly views
- [] Modify chart logic to group data by week starting on Sunday or Monday
- [] Calculate total and average inhalation durations for each week
- [] Display the weekly data in a bar or line chart

Acceptance Criteria

- [] Weekly view renders correctly and updates based on date range selectors
- [] Weekly totals and averages are calculated accurately
- [] Chart remains responsive and visually aligned with app style

Labels

feature, visualization, enhancement, analytics

Estimate

M

markdown

Title

Add Time of Day Usage Heatmap

Description

Introduce a heatmap visualization to display smoking activity by time of day across selected date ranges. This will help users identify peak usage times and develop insights into their habits.

Tasks

- [] Design heatmap chart layout to show hours of the day on the y-axis and days on the x-axis
- [] Process timestamp data to bucket inhalations into hourly segments
- [] Adjust color intensity based on total inhalation duration or count per hour
- [] Implement tooltip or detail view on hover

Acceptance Criteria

- [] Heatmap accurately reflects usage patterns by hour
- [] Visualization updates based on selected date range
- [] Chart is responsive and visually consistent with app theme

Labels

feature, visualization, enhancement, data-analysis

Estimate

L

markdown

Title

Integrate Rolling Average Usage Chart

Description

Add a new line chart to display a rolling average of inhalation frequency or duration over a user-selected window (e.g., 3-day, 7-day). This visualization will help users track trends and fluctuations in smoking behavior over time.

Tasks

- [] Add dropdown selector for rolling window size (e.g., 3, 7, 14 days)
- [] Calculate rolling average of inhalation count or total duration
- [] Overlay rolling average on existing usage chart or create a separate visualization
- [] Ensure smooth transitions and performance for large

datasets

Acceptance Criteria

- [] Users can select different rolling window sizes
- [] Chart dynamically updates based on selection
- [] Rolling trend is clearly visible and accurate

Labels

feature, chart, trend-analysis, enhancement

Estimate

M

markdown

Title

Implement Entry Detail View Modal

Description

Create a modal or bottom sheet that displays detailed information for a selected inhalation entry. This includes timestamp, duration, and any notes or tags associated with the entry. This feature enhances user experience by providing quick access to individual log data without leaving the current screen.

Tasks

- [] Design modal or bottom sheet UI for entry details
- [] Add logic to trigger modal on entry selection from table or chart
- [] Display timestamp, duration, and any associated metadata
- [] Include edit and delete buttons (if applicable)

Acceptance Criteria

- [] Clicking on an entry opens a detail view
- [] Modal shows all relevant data clearly and accurately
- [] Users can close the modal easily

Labels

feature, ui, detail-view, enhancement

Estimate

S

markdown

Title

Add Weekly and Monthly Summary Views

Description

Introduce weekly and monthly summary views to provide users with insights into their smoking habits over longer periods. This feature should aggregate inhalation count and total duration per week/month and visually compare changes over time using bar or line charts.

Tasks

- [] Aggregate inhalation data by week and month
- [] Implement toggles or dropdowns to switch between summary views
- [] Design and integrate summary charts (bar/line)
- [] Display key stats: total sessions, average per day, total duration

Acceptance Criteria

- [] Users can switch between daily, weekly, and monthly summaries
- [] Data is aggregated and visualized correctly
- [] Summary view provides value in identifying usage trends

Labels

feature, summary, analytics, enhancement

Estimate

M

markdown

Title

Implement Custom Date Range Picker for Analytics View

Description

Add a custom date range picker to the analytics view to allow users to select any start and end date for viewing their smoking data. This will enhance flexibility beyond the default daily, weekly, and monthly summaries.

Tasks

- [] Design and integrate a custom date range picker UI component
- [] Filter chart data based on selected range
- [] Display aggregate stats for custom range
- [] Handle edge cases (e.g., no data in range)

Acceptance Criteria

- [] Users can select any date range and view corresponding data
- [] Charts and summaries update correctly based on selected range
- [] UI handles invalid or empty selections gracefully

Labels

feature, UI, filters, analytics

Estimate

M

markdown

Title

Add Color Themes to Enhance Visual Customization

Description

Introduce selectable color themes (e.g., light, dark, and high contrast) to improve accessibility and personalization. The app should remember the user's selected theme and apply it on

startup.

Tasks

- [] Create light, dark, and high-contrast themes
- [] Add theme toggle option in user settings
- [] Persist theme choice locally
- [] Ensure all components adapt correctly to selected theme

Acceptance Criteria

- [] Users can select from predefined themes
- [] Theme persists across sessions
- [] No UI breakage when switching themes

Labels

feature, UI, accessibility, theming

Estimate

S

markdown

Title

Implement Time-Based Aggregation for Smoke Usage Charts

Description

Introduce time-based aggregation options (daily, weekly, monthly) for the smoke usage chart. This feature should allow users to better understand long-term trends in their usage.

Tasks

- [] Add dropdown or toggle for selecting aggregation level
- [] Modify data processing logic to group entries by selected time interval
- [] Update chart rendering to reflect aggregated data
- [] Ensure performance remains smooth with large datasets

Acceptance Criteria

- [] Users can switch between daily, weekly, and monthly views

- [] Aggregated data is accurate and responsive
- [] Chart updates appropriately without visual glitches

Labels

feature, chart, enhancement, UX

Estimate

M

markdown

Title

Add Rolling Average Overlay to Smoke Usage Chart

Description

Enhance the primary usage chart by overlaying a rolling average (e.g., 7-day or 30-day) to help users identify usage trends over time. This statistical feature provides a clearer picture of behavioral changes by smoothing short-term fluctuations.

Tasks

- [] Add UI control to toggle rolling average display
- [] Implement logic to calculate rolling average based on selected time window
- [] Overlay rolling average line on existing chart
- [] Ensure proper formatting and tooltip display

Acceptance Criteria

- [] Users can enable/disable rolling average
- [] Rolling average accurately reflects selected time window
- [] Visuals are distinct and easy to interpret

Labels

feature, analytics, chart, enhancement

Estimate

M

markdown

Title

Design and Implement Dashboard Overview Screen

Description

Create a high-level dashboard overview screen that gives users a quick snapshot of their smoking habits. This view should summarize key metrics such as daily average usage, longest streaks without usage, usage trends over the past week/month, and personalized insights.

Tasks

- [] Design layout for overview screen
- [] Define and implement metric calculations (e.g., averages, streaks)
- [] Display key insights using cards or compact visualizations
- [] Ensure responsive design and accessibility

Acceptance Criteria

- [] Dashboard loads within 1 second
- [] Displays at least 3 meaningful insights
- [] Visuals are intuitive and match the app's aesthetic
- [] All metrics update based on selected date range

Labels

feature, dashboard, analytics, UI/UX

Estimate

L

markdown

Title

Implement Smoker Profile Customization Options

Description

Allow users to personalize their experience with a smoker profile setup. This includes setting their preferred units (e.g., time format, region), customizing their usage goals (e.g., fewer sessions per day, break lengths), and optionally providing motivational reasons for quitting/reducing usage to display throughout the app.

Tasks

- [] Create profile settings screen
- [] Add inputs for unit preferences, goals, and motivational text
- [] Store profile data in local or cloud-based persistent storage
- [] Integrate profile data into dashboard and chart summaries

Acceptance Criteria

- [] Users can update and save their profile without errors
- [] Dashboard and analytics reflect user-set goals and preferences
- [] Motivational messages appear contextually if enabled

Labels

feature, profile, settings, enhancement, UX

Estimate

M

markdown

Title

Develop Adaptive Feedback Based on Usage Trends

Description

Introduce adaptive feedback within the app that dynamically changes based on user activity over time. This includes offering encouragement for positive trends, gentle warnings for negative patterns, and milestone recognition (e.g., 7-day streak, longest session-free period).

Tasks

- [] Define logic for detecting streaks, regressions, and milestones
- [] Design UI components for feedback notifications
- [] Integrate feedback engine with usage data analysis
- [] Implement opt-in setting to toggle adaptive feedback

Acceptance Criteria

- [] App provides relevant, context-aware feedback based on recent logs
- [] Users can enable or disable feedback in settings
- [] Milestones are visually distinctive and persist until acknowledged

Labels

enhancement, feedback, analytics, UI, user-engagement

Estimate

M

markdown

Title

Implement Weekly Summary Visualization

Description

Create a new feature that provides users with a clear, visual summary of their usage data each week. The summary should include total sessions, cumulative duration, average session length, and trend indicators compared to the previous week.

Tasks

- [] Design weekly summary layout with charts and stats
- [] Calculate weekly totals and deltas for key metrics
- [] Highlight increases or decreases in use with color coding
- [] Add "View Weekly Summary" button in the main dashboard

Acceptance Criteria

- [] Weekly summaries are auto-generated and accurate
- [] Users can access past weeks via a simple navigation interface
- [] Visuals are clean, intuitive, and responsive across devices

Labels

feature, analytics, dashboard, UI/UX

Estimate

L

markdown

Title

Add "Rolling Usage" Chart with Customizable Window

Description

Introduce a new chart type that visualizes rolling average usage over time. This chart helps users better understand usage trends while smoothing out daily fluctuations. The rolling window should be user-selectable (e.g., 3-day, 7-day, 14-day).

Tasks

- [] Create rolling average calculation logic based on timestamped session data
- [] Implement interactive chart with dynamic window size selector
- [] Update chart filters to integrate with existing date range logic
- [] Add tooltip explanations for what the rolling average represents

Acceptance Criteria

- [] Rolling chart updates instantly based on selected window and range
- [] Chart renders correctly on mobile and desktop

- [] Rolling averages reflect accurate session durations
- [] Tooltip provides clear, user-friendly definition

Labels

feature, chart, analytics, UI/UX

Estimate

L

markdown

Title

Display Daily Usage Totals with Bar Graph Visualization

Description

Add a new bar graph component to visualize total inhalation time per day. This feature will help users quickly identify which days had the highest or lowest usage.

Tasks

- [] Aggregate total inhalation duration by day from timestamped session logs
- [] Design and implement a responsive bar chart component
- [] Integrate chart into existing dashboard layout
- [] Add tooltips to show exact total usage per day on hover/tap

Acceptance Criteria

- [] Chart displays total duration per day clearly and accurately
- [] Daily bars scale appropriately based on range selected
- [] UI works seamlessly on mobile and desktop
- [] Data is fetched and rendered efficiently

Labels

feature, visualization, analytics, UI/UX

Estimate

M

markdown

Title

Enable Range Selector to Filter Graph and Table Data Simultaneously

Description

Update the range selector functionality to apply to both the chart and the data table below it. Currently, the range only affects the chart, and users must manually interpret the table data separately.

Tasks

- [] Refactor range selector logic to filter Firestore query or cached results
- [] Ensure both graph and table display only the data within the selected range
- [] Update UI to reflect current selected range above both components
- [] Validate synchronization with future analytics additions (e.g., rolling averages)

Acceptance Criteria

- [] Selecting a date range updates both the graph and the data table immediately
- [] Data shown in both views is consistent and accurate
- [] UI is responsive and handles large ranges gracefully
- [] No performance degradation or loading delays introduced

Labels

feature, UI/UX, data-consistency, enhancement

Estimate

M

markdown

Title

Add Rolling Average Option to Inhalation Length Graph

Description

Implement a toggle or dropdown option that enables users to view a rolling average (e.g., 7-day or 30-day) of their inhalation length over time on the main graph. This will help users better identify trends and smooth out short-term fluctuations.

Tasks

- [] Add UI control (toggle or dropdown) for selecting rolling average window (off, 7-day, 30-day)
- [] Apply rolling average calculation on the filtered data set used in the graph
- [] Update graph labels/legend to reflect rolling average mode
- [] Ensure original (non-averaged) data is still accessible and correctly displayed when toggled off

Acceptance Criteria

- [] Users can select and toggle between raw data and rolling average views
- [] Graph updates in real-time based on user selection
- [] Data remains accurate and easy to interpret
- [] System handles edge cases like small data sets gracefully

Labels

feature, enhancement, data-visualization, UX

Estimate

M

markdown

Title

Implement Custom Date Range Selector for Log and Graph Views

Description

Add a custom date range selector to both the inhalation log table and the graph views, allowing users to view data over a specified period (e.g., last 7 days, this month, custom start and end dates). This enhances flexibility for analysis and helps focus on specific time frames.

Tasks

- [] Design and implement a UI component for selecting predefined and custom date ranges
- [] Connect the date range selector to both the log table and inhalation graph
- [] Ensure that all data views update dynamically based on the selected range
- [] Add validation and fallback behavior for invalid or empty ranges

Acceptance Criteria

- [] Users can select from presets (Today, Last 7 Days, This Month, Custom)
- [] Both the graph and table views reflect the selected range
- [] Custom date pickers allow manual selection of start and end dates
- [] Data loads and renders correctly with no performance issues

Labels

feature, UI, UX, enhancement, date-filter

Estimate

M

markdown

Title

Implement Rolling Usage Calculation for Chart Visualization

Description

Add support for rolling usage calculations (e.g., 7-day rolling average) to the inhalation data charts. This provides users with a smoother, more insightful view of usage trends over time, especially useful when analyzing habits and changes in behavior.

Tasks

- [] Add a toggle or dropdown for selecting rolling average options (e.g., None, 3-day, 7-day, 14-day)
- [] Update the chart rendering logic to apply the selected rolling average
- [] Ensure that data normalization (if any) works with rolling values
- [] Validate that tooltips and axis labels update accordingly

Acceptance Criteria

- [] Users can toggle between standard and rolling average views
- [] Charts display smoothed values based on the selected rolling period
- [] No performance regressions when toggling between modes

Labels

feature, data-visualization, enhancement, chart

Estimate

M

markdown

Title

Refactor Chart Data Source to Use Unified Filtering Logic

Description

Currently, filtering and sorting of inhalation log data are implemented separately from the range selector logic. To improve maintainability and enable more consistent chart behavior (e.g., future integrations like cumulative or rolling views), consolidate filtering and sorting logic into a shared utility.

Tasks

- [] Create a centralized utility function for filtering and sorting inhalation logs
- [] Refactor range selector and chart rendering logic to use this utility
- [] Ensure cumulative and rolling modes utilize the same filtered dataset
- [] Validate that the behavior matches the existing functionality

Acceptance Criteria

- [] A single utility handles all filtering and sorting of data
- [] Charts render consistently regardless of selected mode or range
- [] No regressions in data accuracy or performance

Labels

refactor, data-management, chart, enhancement

Estimate

M

markdown

Title

Implement Tab-Based Navigation for Multi-View Smoke Log Analysis

Description

To improve user navigation and support multiple data views

(e.g., Daily, Weekly, Monthly, Cumulative), introduce a tab-based interface that allows users to switch between analysis modes. Each tab will present a distinct chart and analysis type, reusing the unified data filtering utility.

Tasks

- [] Design a tab component with clear labels (e.g., "Daily", "Weekly", "Monthly", "Cumulative")
- [] Implement logic to render appropriate chart for each selected tab
- [] Ensure consistent data filtering and formatting across tabs
- [] Maintain responsiveness and usability on mobile devices

Acceptance Criteria

- [] Users can switch between tabs without full page reload
- [] Each tab displays correct chart and summary based on mode
- [] Charts use consistent styling and structure
- [] Mobile layout remains functional

Labels

frontend, ui/ux, enhancement, chart

Estimate

M

markdown

Title

Add Cumulative Inhalation Summary with Rolling Average Overlay

Description

Enhance the Cumulative tab of the Smoke Log by adding a rolling average overlay to provide insights into long-term usage trends. This will help users visualize how their habits evolve over time.

Tasks

- [] Calculate a rolling average (e.g., 7-day or 30-day) from the cumulative data
- [] Overlay the rolling average on the existing cumulative inhalation chart
- [] Update legends and tooltips to distinguish between total cumulative and average lines
- [] Ensure the rolling average updates dynamically based on selected time range

Acceptance Criteria

- [] Users can clearly distinguish between cumulative total and rolling average lines
- [] Data updates accurately and reflects changes in range selector
- [] Chart remains responsive and legible on all screen sizes

Labels

frontend, chart, enhancement, analysis

Estimate

M

markdown

Title

Improve Mobile Responsiveness for Graph and Table Views

Description

Ensure that the data visualization components in AshTrail (Smoke-Log), including the cumulative graph and usage log table, are fully responsive and user-friendly on mobile devices.

Tasks

- [] Apply responsive layout styling to graphs and tables
- [] Add horizontal scroll for tables on small screens
- [] Adjust chart size and font scaling dynamically for smaller viewports
- [] Test on common screen sizes (e.g., iPhone 13, Galaxy

S21)

Acceptance Criteria

- [] Graph and table views are fully usable without zooming or rotating the device
- [] No overlapping elements or cut-off content on mobile screens
- [] Performance remains smooth on mobile devices

Labels

frontend, mobile, UI, enhancement

Estimate

M

markdown

Title

Integrate Rolling Average Option for Cumulative Usage Graph

Description

Add a toggle to enable or disable a rolling average line on the cumulative usage graph. This feature will help users identify trends over time more clearly by smoothing short-term fluctuations.

Tasks

- [] Add UI toggle or dropdown to enable rolling average
- [] Allow selection of rolling window size (e.g., 7-day, 14-day)
- [] Compute and overlay the rolling average line on the cumulative chart
- [] Ensure accurate alignment with the original data points
- [] Add legend entry for rolling average line

Acceptance Criteria

- [] Users can enable and disable the rolling average line
- [] The rolling average correctly reflects smoothed usage trends

- [] Chart updates dynamically without performance lag

Labels

frontend, feature, data-visualization, enhancement

Estimate

M

markdown

Title

Improve Log Submission UX with Visual Feedback and Error Handling

Description

Enhance the log submission interface by providing users with immediate, clear visual feedback upon submitting a new log entry. Include status indicators for success, loading, and failure. This will help improve trust and usability for users when interacting with the Smoke-Log.

Tasks

- [] Add loading spinner or animation during log submission
- [] Display success message with timestamp confirmation
- [] Handle and show error messages clearly if submission fails
- [] Optionally disable the submit button while request is in progress
- [] Ensure accessibility of feedback messages (e.g., screen reader support)

Acceptance Criteria

- [] Users see real-time visual feedback when submitting logs
- [] All feedback messages are styled consistently and accessibly
- [] Error states do not prevent further attempts to submit

Labels

frontend, UX, feature, accessibility

Estimate

M

markdown

Title

Implement Graph View for Daily Smoking Trends

Description

Create a new graph visualization to display smoking trends over the course of a day. The graph should show timestamps on the x-axis and inhalation length on the y-axis, allowing users to see patterns or changes in behavior throughout a single day.

Tasks

- [] Create a new chart component to show data for a single day
- [] Add support for selecting a date to view detailed trends
- [] Display timestamped inhalation entries with length values
- [] Include markers for session boundaries or notable gaps
- [] Ensure responsive design and mobile usability

Acceptance Criteria

- [] Graph loads correctly with selected day's data
- [] X-axis shows accurate hourly distribution
- [] Y-axis correctly reflects inhalation lengths
- [] Graph updates when the user selects a new date

Labels

frontend, data-visualization, enhancement

Estimate

L

markdown

Title

Add Rolling Average Analysis to Weekly Graph View

Description

Enhance the weekly graph to include a rolling average line that smooths out daily fluctuations in inhalation usage. This will help users visualize broader patterns and trends in their smoking behavior over time.

Tasks

- [] Calculate rolling 3-day and 7-day averages of inhalation duration
- [] Overlay rolling average lines on the weekly graph
- [] Provide toggles to switch between raw data and average views
- [] Ensure accuracy and performance with large datasets

Acceptance Criteria

- [] Users can toggle between raw and averaged data views
- [] Average lines accurately reflect rolling trends
- [] Graph is performant and responsive

Labels

frontend, data-visualization, enhancement

Estimate

L

markdown

Title

Implement Daily Summary Stats in Log View

Description

Introduce a summary section at the top of each day in the log view that aggregates key statistics such as total puffs, total inhalation time, and average duration. This provides users with quick insights without needing to open each entry.

Tasks

- [] Add logic to calculate daily totals and averages from log entries
- [] Display summary panel at the top of each day section
- [] Style the summary to visually differentiate it from log entries
- [] Ensure summaries update reactively as data changes

Acceptance Criteria

- [] Each day displays a total puff count, total inhalation time, and average puff duration
- [] Summaries appear clearly and do not disrupt existing log formatting
- [] All statistics update correctly with data changes or entry deletions

Labels

frontend, UI/UX, enhancement

Estimate

M

markdown

Title

Enhance Data Export Functionality with CSV and JSON Support

Description

Provide users the ability to export their smoking log data in both CSV and JSON formats. This allows for backup, sharing, or external analysis of their data outside the app.

Tasks

- [] Add "Export Data" option in the settings or log view
- [] Implement CSV and JSON formatting of log data
- [] Allow users to choose export format (CSV or JSON)
- [] Trigger download or shareable file link based on platform

Acceptance Criteria

- [] Users can initiate a data export from the UI
- [] Exports include all log entries, formatted cleanly in CSV and JSON
- [] Export file is downloadable or shareable depending on platform

Labels

feature, data, export

Estimate

M

markdown

Title

Implement Quick Add Button for Immediate Logging

Description

To improve usability and reduce friction in logging entries, add a floating "Quick Add" button accessible throughout the app. This button should allow users to instantly log a smoke session with default values or quickly edit a minimal entry.

Tasks

- [] Add floating action button (FAB) in main views
- [] On press, open a simplified log entry modal
- [] Default to current timestamp and user-defined defaults
- [] Include optional edit field for inhalation length
- [] Save entry and refresh log or chart if needed

Acceptance Criteria

- [] FAB appears on main dashboard and log screens
- [] Quick Add modal allows one-tap logging
- [] New entries appear immediately in log/chart

Labels

feature, UI/UX, enhancement

Estimate

M

markdown

Title

Add Weekly and Monthly Usage Summaries to Dashboard

Description

Enhance the dashboard by displaying weekly and monthly summaries of smoking behavior. Include visual indicators (e.g., up/down arrows or percentage change) to show trends compared to previous periods.

Tasks

- [] Calculate weekly and monthly totals from logged sessions
- [] Compare totals with previous week/month
- [] Display change as percentage with visual cues (e.g., ↑ or ↓)
- [] Add summaries to dashboard in a prominent but clean layout

Acceptance Criteria

- [] Dashboard shows total usage for the current week and month
- [] Clear indication of increase or decrease vs. previous period
- [] Responsive design works on all screen sizes

Labels

feature, analytics, dashboard, enhancement

Estimate

M

markdown

Title

Implement Editable Note Field for Each Smoke Log Entry

Description

Allow users to add or edit a short note on each smoke log entry. This will enable contextual tracking, such as reasons for smoking, mood, or situation.

Tasks

- [] Add a note field to the smoke log data model
- [] Update UI to include editable note input in log entry list or detail view
- [] Persist note updates to the database
- [] Support viewing notes in analytics or export

Acceptance Criteria

- [] Each smoke log entry includes a text field for notes
- [] Users can add or edit notes easily
- [] Notes are saved and displayed correctly

Labels

feature, notes, enhancement, UI/UX

Estimate

S

markdown

Title

Create Time Range Filter for Smoke Log Analytics

Description

Implement a feature that allows users to select and filter their analytics view by specific time ranges (e.g., last 7 days, 30 days, custom). This will improve the utility of the dashboard and empower users to observe patterns over specific periods.

Tasks

- [] Design time range selector component
- [] Integrate with analytics queries to apply the selected range
- [] Ensure charts and summary stats update accordingly
- [] Add support for custom date selection

Acceptance Criteria

- [] Users can select from predefined ranges (7d, 30d, etc.)
- [] Users can define custom date ranges
- [] All analytics components respond to the filter
- [] Default range is "Last 7 Days"

Labels

feature, enhancement, analytics, UI/UX

Estimate

M

markdown

Title

Implement Smoke Entry Detail View Modal

Description

Introduce a modal or dedicated detail view that displays comprehensive information for each smoke entry when tapped. This feature will help users gain deeper insight into individual logs, including timestamp, duration, any notes, and contextual stats (e.g., streak at the time of log).

Tasks

- [] Design modal layout for detail view
- [] Connect tap/click event on log entries to trigger modal
- [] Populate modal with full log details from Firestore
- [] Ensure responsive design for mobile and desktop

Acceptance Criteria

- [] Tapping a smoke entry opens a modal

- [] Modal shows timestamp, duration, notes, and streak info
- [] Modal can be closed and returns to previous view
- [] Fully responsive and accessible

Labels

feature, UI/UX, detail-view, firestore

Estimate

M

markdown

Title

Add Settings Page with Customization Options

Description

Introduce a dedicated settings page to allow users to customize their AshTrail experience. This includes options like time format (12h/24h), theme (light/dark/system), log sorting preferences, and streak goal customization.

Tasks

- [] Create Settings page route and layout
- [] Add toggles for time format and theme selection
- [] Add input for streak goal customization
- [] Persist settings using local storage or Firestore (if user authenticated)
- [] Update UI components to respect user settings

Acceptance Criteria

- [] User can access Settings from main navigation
- [] Changes reflect immediately in the app where applicable
- [] Settings persist across sessions
- [] Page follows overall app styling and responsiveness

Labels

feature, UI/UX, enhancement, settings

Estimate

M

markdown

Title

Implement Weekly & Monthly Summary Views

Description

Add the ability for users to view their smoke logs summarized by week and month. This will help users identify patterns and progress over broader time periods beyond daily tracking.

Tasks

- [] Create tab or toggle for Daily, Weekly, Monthly views
- [] Aggregate logs by week and by month
- [] Display total, average, and streak stats for each period
- [] Ensure summary view updates in real-time as data is added or removed
- [] Match visual style and responsiveness of the existing design

Acceptance Criteria

- [] Users can switch between summary views easily
- [] Weekly view shows 7-day aggregates, Monthly view shows full calendar month
- [] All calculations are accurate and visually clear
- [] Views persist user selections within the session

Labels

feature, analytics, enhancement, UI/UX

Estimate

L

markdown

Title

Design and Implement Export to CSV/JSON Functionality

Description

Allow users to export their smoke log data to CSV and JSON formats for personal analysis or backup purposes.

Tasks

- [] Add export button to the log dashboard or settings page
- [] Generate CSV and JSON formats containing all log entries
- [] Format timestamps, inhalation length, and any metadata clearly
- [] Enable download of files on both mobile and desktop browsers
- [] Implement secure handling to prevent unauthorized access to another user's data

Acceptance Criteria

- [] Export feature is accessible and user-friendly
- [] File includes all relevant data and is correctly formatted
- [] Download initiates successfully on all supported devices
- [] Tested for edge cases such as empty logs or large datasets

Labels

feature, data, enhancement, export

Estimate

M

markdown

Title

Implement Cumulative Daily and Weekly Usage Charts

Description

Add visual representations of cumulative usage over time to

provide users with better insights into their smoking patterns. The feature should display daily and weekly aggregates of inhalation length.

Tasks

- [] Add chart toggle to switch between daily and weekly views
- [] Calculate total inhalation length per day and per week
- [] Display cumulative chart using appropriate charting library
- [] Label time axis and provide readable data points
- [] Ensure responsiveness and accessibility across devices

Acceptance Criteria

- [] Users can toggle between daily and weekly views
- [] Chart updates dynamically based on selected range
- [] Cumulative usage is accurate and visually clear
- [] Layout is responsive and does not impact existing features

Labels

feature, enhancement, charts, insights

Estimate

M

markdown

Title

Implement Inhalation Intensity Heatmap View

Description

Add a heatmap visualization to highlight periods of high vs. low inhalation activity over time. This will allow users to quickly identify patterns and peak usage windows during a day or week.

Tasks

- [] Design and implement a heatmap component

- [] Aggregate data into hourly or half-hour blocks
- [] Use color gradients to represent intensity
- [] Allow toggling between daily and weekly views
- [] Add tooltip with exact values on hover/tap

Acceptance Criteria

- [] Heatmap displays aggregated data clearly
- [] Responsive and accessible across all devices
- [] Toggle and hover features work as expected
- [] Performance remains smooth with large datasets

Labels

feature, enhancement, visualization, insights

Estimate

M

markdown

Title

Add Cumulative Usage Timeline Chart

Description

Implement a timeline chart that shows cumulative inhalation usage over time. This visual will help users understand their overall usage trends, including long-term changes in habits and progress.

Tasks

- [] Aggregate inhalation data over time (e.g., daily, weekly)
- [] Implement a cumulative line chart with zoom/pan capabilities
- [] Add labels for total counts and key milestones
- [] Include toggle for timeframe (e.g., 1 week, 1 month, all time)
- [] Ensure data updates dynamically with new entries

Acceptance Criteria

- [] Chart accurately reflects cumulative usage
- [] UI allows easy exploration of trends
- [] Responsive across device types
- [] Data updates without requiring manual refresh

Labels

feature, chart, visualization, usage-trends

Estimate

M

markdown

Title

Implement Rolling Average Usage Chart

Description

Add a chart that visualizes rolling average inhalation usage (e.g., 7-day moving average). This feature will help users identify short-term trends while smoothing out daily fluctuations in usage behavior.

Tasks

- [] Calculate rolling average values from inhalation log data
- [] Implement line chart to display the rolling average
- [] Provide user control to select window size (e.g., 3, 7, 14 days)
- [] Overlay with actual daily values for context
- [] Ensure performance is optimized for large datasets

Acceptance Criteria

- [] Chart displays rolling average with optional overlay of raw data
- [] Users can select window size for smoothing
- [] Accurate calculations and responsive updates with data changes

Labels

feature, chart, usage-patterns, enhancement

Estimate

M

markdown

Title

Add Weekly Usage Summary Card to Dashboard

Description

Design and implement a summary card on the dashboard that displays total inhalation duration, average daily usage, and highest-use day for the current and previous weeks. This card should provide users with quick, digestible insights into their smoking habits over time.

Tasks

- [] Calculate weekly total duration and average daily usage
- [] Identify highest-use day of each week
- [] Design and implement dashboard card UI
- [] Include toggle to compare current and previous weeks
- [] Ensure responsiveness across devices

Acceptance Criteria

- [] Card shows accurate weekly summary and comparison
- [] Highest-use day is highlighted
- [] Visual layout is clear and intuitive
- [] Works on mobile and desktop views

Labels

feature, dashboard, summary, enhancement

Estimate

M

markdown

Title

Implement Inhalation Pattern Heatmap

Description

Add a heatmap visualization that represents inhalation frequency across days and hours. This feature aims to help users identify peak usage times and behavioral patterns by visualizing data density over time.

Tasks

- [] Aggregate inhalation timestamps by hour and day of the week
- [] Design heatmap layout with appropriate color gradients
- [] Integrate heatmap into the analytics or dashboard section
- [] Add tooltip or label support for detailed data on hover/tap
- [] Ensure performance for large datasets

Acceptance Criteria

- [] Heatmap clearly shows usage frequency by time and day
- [] Hover/tap displays exact usage count or duration
- [] Responsive and readable on both mobile and desktop

Labels

feature, visualization, analytics, enhancement

Estimate

L

markdown

Title

Add Session Playback Visualization

Description

Implement a session playback feature that replays a user's smoking session timeline, showing inhalation duration and

spacing in a visual, animated format. This provides users with an engaging way to understand their usage rhythm and habits.

Tasks

- [] Design session playback interface (e.g., animated timeline or pulse graph)
- [] Animate inhalation events based on duration and interval
- [] Sync animation with timestamp markers
- [] Add controls for play, pause, speed, and session selection
- [] Optimize playback for mobile and desktop

Acceptance Criteria

- [] Users can watch a session animation with clear timing between inhales
- [] Playback visually represents both inhale length and spacing
- [] Performance is smooth on standard mobile and desktop devices

Labels

feature, visualization, UX, enhancement

Estimate

M

markdown

Title

Enable Range-Based Trend Comparison

Description

Implement a feature that allows users to select two different time ranges and compare their smoking habits side by side. This will help users identify behavioral changes over time, such as reduced usage or increased spacing between inhales.

Tasks

- [] Design dual-range selector UI

- [] Implement backend filtering for two selected ranges
- [] Display two sets of charts side by side (e.g., total usage, average inhale length)
- [] Highlight key differences (e.g., percentage change)
- [] Add summary insights to enhance comparison value

Acceptance Criteria

- [] Users can select two non-overlapping time ranges for comparison
- [] Charts and stats clearly show differences in usage patterns
- [] Feature works seamlessly on mobile and desktop views

Labels

feature, analytics, UX, enhancement

Estimate

M

markdown

Title

Add Rolling Average to Usage Charts

Description

Introduce rolling average calculations (e.g., 7-day and 30-day) to the inhale frequency and duration charts. This will smooth out short-term fluctuations and help users visualize long-term trends in their smoking behavior.

Tasks

- [] Add toggle for 7-day and 30-day rolling averages on inhale count and inhale length charts
- [] Implement logic to compute rolling averages based on filtered range
- [] Ensure averages update dynamically with date range selector
- [] Style the lines differently (e.g., dashed or different color) to distinguish from raw data

Acceptance Criteria

- [] Rolling average line displays correctly on both desktop and mobile
- [] Data updates dynamically as users change date range or filters
- [] Clear legend and UI cues help users understand the rolling average

Labels

feature, analytics, data-visualization

Estimate

S

markdown

Title

Implement Chart Data Export Functionality

Description

Allow users to export their inhale log data and chart visuals for personal tracking or sharing with healthcare providers. This feature should support CSV export of raw data and PNG export of charts.

Tasks

- [] Add export button for each chart (CSV for data, PNG for visuals)
- [] Implement CSV formatting for filtered data
- [] Integrate chart image export using chart library capabilities
- [] Confirm feature works on both mobile and desktop

Acceptance Criteria

- [] Users can download a CSV file of inhalation data based on current filter/range
- [] Users can save current chart view as a PNG image
- [] Export functionality is accessible and clear in the UI

Labels
feature, export, data, enhancement

Estimate
M

markdown

Title
Implement Dark Mode Toggle Support

Description
Introduce a dark mode option for the AshTrail app to improve usability in low-light environments and match user system preferences.

Tasks

- [] Detect system dark/light mode preference on app startup
- [] Provide a toggle in settings to manually switch themes
- [] Style all app components for both light and dark modes
- [] Ensure charts and text remain legible in dark mode

Acceptance Criteria

- [] Theme automatically matches user system preference on first launch
- [] User can toggle between dark and light mode at any time
- [] All visual elements are styled appropriately in both themes
- [] No UI elements are broken or unreadable in either mode

Labels
feature, theme, dark-mode, UI/UX

Estimate
M

markdown

Title

Add Analytics Dashboard for User Smoking Trends

Description

Create an interactive dashboard that allows users to visualize their smoking trends over time. This will help users track habits and identify patterns or progress.

Tasks

- [] Design UI layout for the dashboard (include charts, date filters, stats)
- [] Implement line chart for daily inhalation counts
- [] Add heatmap calendar view for identifying high-usage days
- [] Display weekly/monthly summary statistics (total length, average per day)
- [] Allow filtering by custom date range
- [] Integrate data from Firestore and local cache (Isar)

Acceptance Criteria

- [] Users can view daily, weekly, and monthly summaries
- [] Data visualizations update dynamically based on selected range
- [] Charts and stats are accurate and responsive
- [] Dashboard works seamlessly across devices

Labels

feature, enhancement, UI/UX, dashboard, analytics

Estimate

L

markdown

Title

Implement User Settings Panel for Customization Options

Description

Add a user settings panel that allows customization of key aspects of the Smoke-Log experience. This should include time format (12h/24h), preferred measurement units, theme (light/dark/system), and optional reminders or usage goals.

Tasks

- [] Design the UI for the settings panel
- [] Implement theme toggle (light/dark/system)
- [] Add time format selection (12h or 24h)
- [] Allow user to set inhalation goals (daily/weekly)
- [] Enable push/local notifications for reminders
- [] Persist settings using local storage and sync with Firestore

Acceptance Criteria

- [] Users can customize and persist their settings
- [] Theme and format changes apply immediately
- [] Settings are retained across sessions
- [] Goals and reminders function as configured

Labels

feature, settings, enhancement, UI/UX

Estimate

M

markdown

Title

Implement Stats Tab with Rolling Averages and Insights

Description

Develop a new "Stats" tab to provide users with deeper insights into their inhalation trends over time. Include rolling average charts, peak usage times, daily/weekly summaries, and motivational messages based on progress.

Tasks

- [] Create "Stats" tab in the navigation

- [] Display 7-day and 30-day rolling average charts
- [] Highlight peak usage hours per day of week
- [] Generate summaries (total sessions, average duration, etc.)
- [] Show motivational messages based on user trend
- [] Ensure responsiveness across devices

Acceptance Criteria

- [] Stats tab loads efficiently with all visualizations
- [] Rolling averages update correctly as new logs are added
- [] Insight messages are context-aware and vary by user trends
- [] UI is mobile-optimized and maintains design consistency

Labels

feature, analytics, visualization, UI/UX

Estimate

L

markdown

Title

Add Custom Date Range Picker for Usage Analysis

Description

Implement a custom date range picker component to allow users to analyze their smoking patterns within a specific timeframe. This will improve the accuracy of insights and allow comparisons between custom periods (e.g., month-to-month or before/after specific dates).

Tasks

- [] Create a reusable date range picker widget
- [] Integrate it into the Stats and Charts tabs
- [] Filter inhalation log data based on selected range
- [] Display total sessions, average length, and rolling stats within range
- [] Ensure UI responds smoothly to range changes

Acceptance Criteria

- [] Date range picker is accessible and intuitive on all screen sizes
- [] Selected range updates charts and stats immediately
- [] Edge cases (no data, large ranges) handled gracefully

Labels

feature, date-filter, UI, enhancement

Estimate

M

markdown

Title

Implement Rolling Average Toggle for Usage Trends

Description

Add a toggle feature to enable or disable rolling average visualization on usage trend charts. This allows users to smooth out daily fluctuations and better identify long-term patterns in their smoking behavior.

Tasks

- [] Add a toggle switch to the Stats tab UI
- [] Apply a 7-day rolling average to session length and count charts when enabled
 - [] Visually differentiate the rolling average line (e.g., dashed line or different color)
 - [] Ensure chart updates are smooth and performant when toggled

Acceptance Criteria

- [] Toggle is accessible and responsive
- [] Rolling average line appears/disappears as expected
- [] Tooltips and legends clearly indicate rolling average data
- [] No performance issues on large datasets

Labels
feature, charts, enhancement, UX

Estimate
M

markdown

Title
Improve Data Export Functionality

Description
Users should be able to export their usage data in a clean and structured format for backup or personal analysis. The export should include timestamps, session lengths, notes, and any other relevant fields, ideally as a CSV file.

Tasks

- [] Add an "Export Data" button to the Settings tab
- [] Generate CSV with all user logs sorted by timestamp descending
- [] Include headers: Timestamp, Session Length, Notes, Tags (if applicable)
- [] Provide feedback upon successful download
- [] Ensure proper formatting for cross-platform compatibility

Acceptance Criteria

- [] Button is clearly visible and accessible
- [] File downloads as `ashtrail-logs-YYYY-MM-DD.csv`
- [] CSV opens cleanly in Excel and other spreadsheet apps
- [] No errors for large datasets

Labels
feature, export, enhancement, data

Estimate

S

markdown

Title

Implement Offline Support with Local Persistence

Description

To ensure users can log and view sessions even without internet access, AshTrail should support offline functionality. This includes writing to a local database (Isar) and syncing with Firestore when a connection is restored.

Tasks

- [] Set up Isar for local data storage
- [] Mirror the Firestore schema locally
- [] Detect network connectivity status
- [] Queue logs for upload when offline
- [] Sync unsynced records on reconnect
- [] Prevent duplicate uploads

Acceptance Criteria

- [] App does not crash or lose data without internet
- [] Logs appear immediately after submission, even offline
- [] Logs sync properly and appear in Firestore once reconnected
- [] Offline status is clearly indicated in the UI

Labels

feature, offline, isar, sync, enhancement

Estimate

M

markdown

Title

Add Dynamic Filtering for Smoke Logs by Date Range and Type

Description

Enhance the user experience by allowing dynamic filtering of smoke logs based on a selected date range and session type (e.g., quick, detailed, or custom). This will support better insights and visual analytics.

Tasks

- [] Add date range picker UI component
- [] Add dropdown or toggle for session type filter
- [] Update filtering logic to handle combinations of date range and type
- [] Ensure charts and data table respond to applied filters
- [] Optimize performance for filtering large datasets

Acceptance Criteria

- [] Users can select a custom date range
- [] Users can filter by session type
- [] Charts and logs update in real-time with applied filters
- [] Filters persist during the session until manually cleared

Labels

feature, filters, UI/UX, analytics

Estimate

M

markdown

Title

Implement Rolling Averages in Usage Charts

Description

To provide more insight into smoking patterns, implement a toggleable rolling average line (e.g., 7-day or 30-day) over

existing usage charts. This will help identify trends and smooth out daily fluctuations.

Tasks

- [] Add a toggle to enable/disable rolling average line
- [] Compute rolling averages client-side using filtered data
- [] Update chart rendering logic to display rolling average line
- [] Label the line clearly with its window size (e.g., “7-day avg”)
- [] Ensure proper handling of edge cases (e.g., insufficient data)

Acceptance Criteria

- [] Rolling average line is visible when toggled
- [] Line updates based on filtered date range and session type
- [] Performance remains smooth on large datasets
- [] Accurate calculation and display of average values

Labels

enhancement, charting, data-visualization

Estimate

M

markdown

Title

Refactor Data Filtering Logic for Centralized Management

Description

Move all data filtering and sorting logic to a centralized module to improve maintainability and consistency across components. This will also simplify integrating new types of charts and analyses (e.g., rolling averages, range comparisons).

Tasks

- [] Create a `filterUtils.ts` or similar helper file
- [] Move current filtering/sorting logic from UI components to this file
 - [] Ensure all components use the centralized logic
 - [] Add unit tests for the filtering and sorting logic
 - [] Update documentation or inline comments for clarity

Acceptance Criteria

- [] No duplicated filtering logic in components
- [] Tests cover edge cases (e.g., missing or malformed data)
- [] Charts still render correctly after refactor
- [] Easy to add new filters (e.g., by session type or threshold)

Labels

refactor, data-handling, enhancement

Estimate

M

markdown

Title

Implement Dynamic Range Selector with Real-Time Chart Updates

Description

Add a dynamic range selector that allows users to choose from preset date ranges (e.g., 7 days, 30 days, custom). Selecting a range should immediately update all relevant charts and tables to reflect the filtered data.

Tasks

- [] Design and implement the range selector UI component
- [] Integrate with the centralized filtering logic
- [] Trigger chart and table updates on range change
- [] Maintain selected range across app state
- [] Add unit tests for range selection and updates

Acceptance Criteria

- [] Users can select from multiple range presets or define a custom range
- [] Charts and tables reflect changes in real-time
- [] No performance lag or visual glitches during updates

Labels

feature, UI, enhancement

Estimate

M

markdown

Title

Create Reusable Chart Component with Modular Configuration

Description

Develop a reusable chart widget component that supports multiple chart types (e.g., line, bar, area) and can be configured through a modular input system. This will streamline the addition of new charts and improve maintainability.

Tasks

- [] Define chart configuration interface (chart type, data series, labels, colors, etc.)
- [] Refactor existing chart code into reusable component
- [] Add support for different chart types
- [] Integrate with centralized state management
- [] Document usage with examples
- [] Add unit tests

Acceptance Criteria

- [] Component accepts configuration and renders the appropriate chart
- [] Multiple chart types function correctly and update on data change
- [] Clear documentation and working examples are provided

Labels

refactor, component, enhancement

Estimate

M

markdown

Title

Add Filter Presets and Save Custom Ranges for Analysis

Description

Implement functionality to allow users to quickly apply common date/time range filters (e.g., "Last 7 Days", "This Month") and save their own custom ranges for future use. This improves user efficiency and supports more consistent analysis workflows.

Tasks

- [] Add predefined filter options to date/time picker
- [] Enable users to save custom date/time ranges
- [] Store presets in local storage or user profile if authenticated
- [] Integrate filter presets with chart and table data updates
- [] Add UI for managing saved custom ranges
- [] Write unit tests

Acceptance Criteria

- [] Users can apply preset date ranges
- [] Users can save and apply custom ranges
- [] Presets persist and affect charts/tables appropriately
- [] Functional UI for managing saved filters

Labels

feature, UX, filters

Estimate

M

markdown

Title

Implement Multi-Series Chart Support for Comparative Analysis

Description

Introduce the ability to view and compare multiple data series (e.g., different tags, days, or user sessions) on the same chart. This feature will help users identify patterns, trends, and deviations more effectively across sessions or time periods.

Tasks

- [] Update chart rendering logic to support multiple datasets
- [] Allow user selection of multiple tags or sessions for overlay
- [] Use color differentiation and legend for clarity
- [] Add toggle for enabling/disabling series comparison
- [] Ensure responsive design and maintain performance
- [] Write tests for multi-series chart rendering

Acceptance Criteria

- [] Users can select and view multiple data series on a single chart
- [] Series are clearly distinguishable with legend support
- [] Chart updates correctly based on user input

Labels

feature, charts, data-visualization

Estimate

L

markdown

Title

Add Export Feature for Session Data

Description

Enable users to export their smoking session data in a structured format such as CSV or JSON. This feature is essential for users who want to back up their data, analyze it externally, or share it with a healthcare provider.

Tasks

- [] Design export interface (button and file format selector)
- [] Implement backend logic to structure and generate export file
- [] Add frontend integration and download capability
- [] Ensure data includes timestamps, lengths, and tags
- [] Write unit and integration tests

Acceptance Criteria

- [] Users can export a selected date range or full session history
- [] Data is downloadable in CSV and JSON format
- [] Exported files open correctly with standard tools

Labels

feature, data-export, enhancement

Estimate

M

markdown

Title

Implement Daily and Weekly Summary Cards

Description

Introduce summary cards at the top of the log screen that display total inhalation time and number of sessions for the

current day and week. This offers quick insights into usage trends and promotes awareness.

Tasks

- [] Design UI components for summary cards
- [] Calculate daily and weekly totals based on log data
- [] Update summaries dynamically as new sessions are added
- [] Style cards to match the app's visual theme
- [] Write tests for calculations and display logic

Acceptance Criteria

- [] Summary cards show correct totals for today and current week
- [] Data updates in real-time as new entries are logged
- [] Cards are responsive and visually integrated

Labels

feature, UI, summary, enhancement

Estimate

S

markdown

Title

Add Inhalation Duration Histogram with Time-of-Day Grouping

Description

Implement a histogram that visualizes inhalation session durations grouped by time of day (e.g., early morning, morning, afternoon, evening, night). This helps users identify when longer or shorter sessions tend to occur.

Tasks

- [] Define time-of-day buckets and group existing session data accordingly
- [] Create a histogram chart using the current charting library
- [] Style the chart to match the app theme

- [] Update filtering logic to dynamically change the histogram view
- [] Add tests to verify correct bucketing and rendering

Acceptance Criteria

- [] Histogram displays inhalation lengths grouped by time-of-day buckets
- [] Correctly updates when the date range changes
- [] Responsive design and consistent with overall app style

Labels

feature, chart, visualization, enhancement

Estimate

M

markdown

Title

Implement Rolling 7-Day Average Usage Line on Daily Chart

Description

Add a rolling 7-day average line to the daily inhalation chart to help users visualize longer-term trends in their usage. This feature will allow users to better assess whether their usage is increasing or decreasing over time.

Tasks

- [] Calculate 7-day rolling average from inhalation data
- [] Overlay line chart on existing daily bar chart
- [] Ensure responsiveness and clear labeling of the average line
- [] Toggle visibility of the average line in the chart settings panel

Acceptance Criteria

- [] Line chart appears only when 7+ days of data exist
- [] Line updates dynamically with range selection
- [] Visually distinct from the main bar chart

- [] Optional toggle to enable/disable the line

Labels

feature, chart, enhancement, UX

Estimate

M

markdown

Title

Create Toggle for Development vs Production Mode in Data Submission

Description

Add a toggle mechanism to differentiate between development and production environments. When in development mode, data submitted via the app should be written to a "Test" Firestore collection. When in production mode, data should go to the main collection.

Tasks

- [] Implement an environment-based toggle or configuration flag
- [] Modify the data submission function to respect this flag
- [] Ensure the correct collection name is selected based on environment
- [] Add visual indicator in the app UI to show current mode (Dev or Prod)

Acceptance Criteria

- [] Data submitted in development does not affect production data
- [] Environment toggle is respected across app sessions
- [] Collection writes verified in both environments

Labels

environment, backend, firebase, devtools

Estimate

S

markdown

Title

Add Data Table Below Usage Charts for Timestamped Entries

Description

Introduce a data table component beneath the usage charts to display all submitted entries. This table should show each inhalation record in descending order by timestamp, including duration and formatted timestamp.

Tasks

- [] Create a responsive data table component below the existing charts
- [] Pull records from the same filtered and sorted data used in the charts
- [] Display fields: formatted timestamp, inhalation duration (in seconds or milliseconds)
- [] Ensure the table updates dynamically with chart filters (e.g., range selector)

Acceptance Criteria

- [] Entries in the table match the chart display filters
- [] Table loads quickly and handles large numbers of entries
- [] Design remains mobile-friendly and scrollable as needed

Labels

frontend, visualization, firebase, flutter

Estimate

M

markdown

Title

Implement Range Selector Sync for Charts and Data Table

Description

Ensure that the range selector (e.g., last 7 days, last 30 days) affects both the usage charts and the data table below consistently. This synchronization will help users view and analyze data within the same time context.

Tasks

- [] Refactor range selector logic to apply filters to both chart data and table data
- [] Ensure that changing the range selector updates both components in real time
- [] Verify correct behavior across all predefined time ranges
- [] Add loading state or animation if needed during data refresh

Acceptance Criteria

- [] Selecting a new date range refreshes both the chart and table
- [] Data is filtered accurately according to the selected range
- [] User experience remains smooth with no excessive delays or glitches

Labels

frontend, flutter, UX, enhancement

Estimate

M

markdown

Title

Add Toggle to Switch Between Cumulative and Daily Usage Charts

Description

Implement a toggle control that allows users to switch between viewing cumulative usage over time and daily usage. This provides flexibility in understanding usage trends and identifying anomalies.

Tasks

- [] Add toggle UI component above or near the chart
- [] Update chart rendering logic to reflect selected mode (daily vs cumulative)
- [] Ensure both modes respect the current date range selector
- [] Test for responsiveness and edge cases (e.g., single-day data)

Acceptance Criteria

- [] Users can switch between cumulative and daily views
- [] Chart updates immediately and accurately on toggle change
- [] Behavior remains consistent with date range selections

Labels

frontend, flutter, charting, UX

Estimate

M

markdown

Title

Add Snackbar Confirmation for Log Entry Deletion

Description

When a user deletes a log entry, provide immediate visual feedback through a snackbar (toast-style) notification. This enhances UX by confirming the action and optionally allowing an undo within a few seconds.

Tasks

- [] Create a Snackbar widget styled consistently with the app
- [] Trigger Snackbar on successful deletion of a log entry
- [] Include an "Undo" button in the Snackbar to restore the deleted log
- [] Ensure Snackbar is dismissed automatically after a short timeout (e.g., 3-5 seconds)

Acceptance Criteria

- [] Users receive immediate visual confirmation when deleting a log
- [] The "Undo" action restores the deleted log correctly
- [] Snackbar styling matches app theme and is accessible

Labels

frontend, flutter, UX, log-management

Estimate

S

markdown

Title

Implement Daily Usage Summary Chart

Description

Add a chart component to visualize daily smoking activity using the log data. This will allow users to see their habits over time and track progress toward reduction goals.

Tasks

- [] Design and integrate a line or bar chart summarizing inhalation length per day
- [] Aggregate log data by date and compute total or average inhalation per day
- [] Allow switching between total and average view modes
- [] Ensure responsiveness and dark mode compatibility

Acceptance Criteria

- [] Chart displays accurate daily totals or averages based on log data
- [] Toggle between total and average mode works correctly
- [] Chart renders correctly across devices and themes

Labels

data-visualization, flutter, enhancement, dashboard

Estimate

M

markdown

Title

Integrate Rolling Usage Graph with Adjustable Range

Description

Implement a rolling usage graph to display smoking trends over a selected time window (e.g., 3, 7, 30 days). This will help users identify behavior patterns and visualize long-term changes.

Tasks

- [] Design the UI for selecting a rolling window (dropdown or buttons for 3/7/30 days)
- [] Calculate and plot rolling average of inhalation lengths
- [] Ensure smooth integration with existing date filter logic
- [] Make the chart responsive and visually consistent with other graphs

Acceptance Criteria

- [] Users can select a rolling window and see the correct rolling averages
- [] Data updates correctly when filtering by date or session type
- [] UI components are accessible and visually clear

Labels

analytics, enhancement, flutter, smoke-log, ui/ux

Estimate

M

markdown

Title

Add Daily Usage Summary with Total Inhalation Length

Description

Introduce a daily summary section that displays the total inhalation length per day to give users a quick overview of their consumption trends.

Tasks

- [] Create a new section or component to display daily totals
 - [] Aggregate inhalation lengths by day from Firestore data
 - [] Display the summary in a visually clear and concise format (e.g., cards or list)
 - [] Ensure compatibility with date range filters and session type filters

Acceptance Criteria

- [] Users see daily total inhalation lengths for the selected date range
- [] Data updates correctly when filters change
- [] Visual presentation aligns with the rest of the app's style

Labels

summary, data-visualization, enhancement, flutter, smoke-log

Estimate

S

markdown

Title

Implement Export to CSV Feature for Inhalation Logs

Description

Allow users to export their inhalation log data to a CSV file for personal analysis or sharing with medical professionals. This feature should include timestamp, length, session type, and any additional notes.

Tasks

- [] Add a button to trigger export within the app interface
- [] Format log data into a CSV structure
- [] Implement file save or share functionality depending on platform (web/mobile)
- [] Ensure compatibility with active filters (date range, session type)

Acceptance Criteria

- [] Users can export only the data currently visible based on filters
- [] The exported CSV file includes all necessary fields and is properly formatted
- [] Export works reliably across supported platforms

Labels

export, data-management, enhancement, flutter, smoke-log

Estimate

M

markdown

Title

Add Graph Legend and Tooltip Enhancements for Chart Clarity

Description

Improve the chart components by adding a legend and enhancing tooltips to make data interpretation easier for users. The

legend should clearly label each dataset, and tooltips should provide detailed timestamp and inhalation duration on hover or tap.

Tasks

- [] Add a visible legend below or beside the charts
- [] Display dataset labels and colors consistently
- [] Enhance tooltip content to show timestamp, duration, and any related metadata
- [] Ensure responsiveness and accessibility

Acceptance Criteria

- [] Users can easily identify which data corresponds to which dataset
- [] Tooltips provide detailed and accurate information
- [] Enhancements work across all chart types used in the app

Labels

charting, ui/ux, enhancement, smoke-log

Estimate

S

markdown

Title

Implement Editable Log Entry Feature

Description

Enable users to edit previously submitted log entries. This will allow corrections to timestamps, durations, or any other metadata tied to an inhalation event. This feature should be accessible via the history table or detailed view of each log.

Tasks

- [] Add an "Edit" button to each row in the log history table
- [] Create an edit modal or inline editor to update timestamp and duration

- [] Validate changes before submission
- [] Update the Firestore document and refresh visualizations on save

Acceptance Criteria

- [] Users can initiate editing from the history table
- [] Changes are reflected in the database and visual components
- [] Invalid data is prevented from being saved

Labels

feature, data-entry, smoke-log, firestore

Estimate

M

markdown

Title

Add Rolling Usage Average Chart Option

Description

Introduce an optional chart view that displays rolling averages (e.g., 1-day, 3-day, 7-day) of inhalation durations or frequency to help users visualize trends over time. This can offer insight into habit patterns and progress.

Tasks

- [] Add toggle or dropdown to switch between raw and rolling average views
- [] Implement rolling average calculation in data processing logic
- [] Update chart components to display rolling trendlines
- [] Ensure responsiveness and correct display on different devices

Acceptance Criteria

- [] Users can toggle between different rolling average periods

- [] Trendlines reflect correct calculations and update dynamically
- [] Feature integrates seamlessly with existing filtering options

Labels

enhancement, charting, analytics, smoke-log

Estimate

L

markdown

Title

Implement Daily Summary Statistics View

Description

Add a new section to the dashboard that summarizes each day's inhalation activity with statistics such as total inhalations, total duration, average duration, and earliest/latest timestamps. This gives users a quick overview of their daily usage.

Tasks

- [] Design layout for daily summary cards or table
- [] Compute daily aggregates from Firestore data
- [] Display summary for the selected date range
- [] Style summaries to match the AshTrail visual theme

Acceptance Criteria

- [] Users can see per-day stats when selecting a range
- [] All calculations are accurate and visually clear
- [] Works on mobile and desktop layouts

Labels

enhancement, UI, analytics, smoke-log

Estimate

M

markdown

Title

Implement Rolling Weekly Usage Trend Chart

Description

Create a new line chart that shows a 7-day rolling average of inhalation duration and frequency. This visualization helps users identify long-term usage patterns and behavioral changes over time.

Tasks

- [] Calculate 7-day rolling average from timestamped inhalation data
- [] Render the chart using the existing chart library
- [] Add toggles to switch between duration and frequency views
- [] Ensure responsiveness and consistent theming with the rest of the app

Acceptance Criteria

- [] Chart displays accurate 7-day rolling averages
- [] Users can easily switch between average duration and frequency
- [] Chart integrates seamlessly with the existing dashboard

Labels

feature, chart, analytics, smoke-log

Estimate

L

markdown

Title

Enable Quick Entry Presets for Inhalation Logging

Description

Streamline the logging experience by allowing users to save and use "Quick Entry" presets for common inhalation durations and notes. This enhances speed and convenience when users log similar sessions regularly.

Tasks

- [] Add UI option to save current entry as a preset
- [] Display saved presets as buttons on the log screen
- [] Allow editing and deleting of saved presets
- [] Store presets locally and ensure persistence across sessions

Acceptance Criteria

- [] Users can save a preset from an existing log entry
- [] Presets are visible and usable from the logging interface
- [] Modifying or deleting presets works as expected

Labels

enhancement, UI/UX, smoke-log, productivity

Estimate

M

markdown

Title

Add Cumulative Weekly and Monthly Usage Charts

Description

Introduce visual analytics for weekly and monthly smoking trends using the inhalation data. These charts should help users identify usage patterns over time, encouraging more mindful behavior.

Tasks

- [] Implement logic to aggregate inhalation data weekly and monthly
- [] Add bar and line chart views for each time range
- [] Integrate with the existing range selector and chart switcher
- [] Ensure responsiveness across screen sizes

Acceptance Criteria

- [] Charts show accurate total inhalation time per week/month
- [] Users can switch between daily, weekly, and monthly views
- [] Charts visually align with the app's design theme

Labels

analytics, enhancement, charts, smoke-log

Estimate

L

markdown

Title

Implement Rolling Average Usage Analysis Chart

Description

Add a line chart displaying rolling average inhalation usage over a user-selected window (e.g., 3-day, 7-day). This helps users observe trends over time without the noise of daily fluctuations.

Tasks

- [] Allow selection of rolling average window (3, 7, 14 days)
- [] Compute rolling averages for inhalation durations
- [] Add a toggle or dropdown to switch between normal and rolling average view

- [] Integrate with current date range filtering and chart rendering logic

Acceptance Criteria

- [] Rolling average calculations match expected values
- [] Users can select different window sizes easily
- [] Chart updates correctly based on range selector and rolling settings

Labels

analytics, enhancement, trends, smoke-log

Estimate

M

markdown

Title

Implement Daily Summary Table Below Charts

Description

Display a table beneath the inhalation charts that shows daily totals and average inhalation durations, sorted in descending order by date. This allows users to see detailed daily summaries alongside visual trends.

Tasks

- [] Create a component to render a table below the chart
- [] Populate table with total and average inhalation durations per day
- [] Ensure table updates based on date range selector
- [] Sort entries in descending order by date

Acceptance Criteria

- [] Table reflects correct totals and averages for each day
- [] Date sorting is descending by default
- [] Updates dynamically with chart range

Labels

data-table, enhancement, UI, smoke-log

Estimate

S

markdown

Title

Add Cumulative Inhalation Usage Chart

Description

Create a line chart that visualizes cumulative inhalation usage over time, allowing users to track their overall smoking behavior across selected date ranges.

Tasks

- [] Compute cumulative total of inhalation durations based on filtered dataset
- [] Implement a new chart below the main inhalation chart
- [] Display cumulative progression clearly with proper axis labeling
- [] Integrate with date range selector and filters

Acceptance Criteria

- [] Cumulative chart accurately represents the total inhalation time up to each point
- [] Filters and date range selection dynamically update the chart
- [] Visual style is consistent with existing charts

Labels

chart, enhancement, smoke-log, analytics

Estimate

M

markdown

Title

Implement Data Table for Logged Inhalation Entries

Description

Create a sortable and filterable data table below the charts that displays individual inhalation log entries in reverse chronological order.

Tasks

- [] Build a responsive data table component
- [] Display timestamp, duration, and any relevant metadata per row
- [] Integrate sorting by timestamp (default: descending)
- [] Add filter logic based on the active date range and any selected filters
- [] Ensure visual consistency with the rest of the app

Acceptance Criteria

- [] All visible data matches chart inputs
- [] Most recent entries appear first
- [] Table updates correctly when filters or date ranges are changed

Labels

data-table, enhancement, smoke-log, frontend

Estimate

M

markdown

Title

Enable Manual Log Entry with Date and Time Selection

Description

Allow users to manually log an inhalation event by selecting a specific date and time, along with duration and optional notes. This supports users who forgot to log in real time.

Tasks

- [] Design and implement a manual log entry form
- [] Include inputs for date, time, duration, and notes
- [] Validate inputs and update local and Firestore records
- [] Ensure manual entries are reflected in charts and tables
- [] Add a clear button or modal UI for accessing this feature

Acceptance Criteria

- [] User can input past inhalation logs accurately
- [] Manual logs appear in correct chronological order across all components
- [] Validation handles edge cases like future timestamps or missing fields

Labels

manual-log, enhancement, smoke-log, frontend

Estimate

M

markdown

Title

Create User Settings Screen with Persistent Preferences

Description

Introduce a user settings screen to manage preferences such as default chart range, theme (dark/light), and chart smoothing options. Store settings locally to persist across sessions and reload them on app start.

Tasks

- [] Design and implement the settings UI screen
- [] Include toggles/selects for:
 - Default chart range (e.g., 7d, 30d, all time)
 - Dark/Light/Auto theme mode
 - Enable/disable chart smoothing

- [] Save preferences using local persistent storage
- [] Load preferences on app startup and apply them to chart rendering and UI

Acceptance Criteria

- [] Settings screen is accessible from the main app
- [] Preferences persist across app restarts
- [] Changes to settings immediately reflect in the UI and chart behavior

Labels

settings, enhancement, smoke-log, ui, frontend

Estimate

M

markdown

Title

Implement Enhanced Data Export Options (CSV, JSON)

Description

Expand the data export capabilities to allow users to download their smoking logs in CSV and JSON formats. This will improve accessibility and usability for users who want to analyze or backup their data outside the app.

Tasks

- [] Add an "Export Data" option in the settings or chart screen
- [] Implement export to CSV format
- [] Implement export to JSON format
- [] Ensure exported files include all relevant fields (timestamp, inhale duration, notes, etc.)
- [] Support sharing or saving to local device storage

Acceptance Criteria

- [] User can export logs as a CSV file
- [] User can export logs as a JSON file

- [] Exported data includes complete, accurate, and properly formatted entries

Labels

export, enhancement, smoke-log, backend, frontend

Estimate

M

markdown

Title

Add Daily Summary Cards to Dashboard

Description

Introduce daily summary cards on the main dashboard screen that display key statistics for the selected day, including total puffs, total inhale time, average inhale duration, and notes summary. These cards should be visually clean and responsive to date filters.

Tasks

- [] Design UI for daily summary cards
- [] Calculate total puffs, inhale time, and average inhale duration for the selected day
- [] Display the first few lines of notes (if any) for that day
- [] Integrate cards with the date selector logic
- [] Test responsiveness and layout on various screen sizes

Acceptance Criteria

- [] User sees summary cards for total puffs, inhale time, and average inhale duration
- [] Cards update correctly based on selected day
- [] Cards follow the visual design and remain legible on different screen sizes

Labels

dashboard, UI/UX, enhancement, smoke-log

Estimate

M

markdown

Title

Implement Interactive Weekly Trends Chart

Description

Add an interactive chart to the dashboard that displays trends over the past 7 days. This chart should include:

- Daily total puffs
- Total inhale time per day
- Average inhale duration per day

Enable users to hover/tap to see values per day. Ensure the chart updates dynamically with data and is visually consistent with the app theme.

Tasks

- [] Design chart component for weekly trends
- [] Aggregate and format last 7 days of data
- [] Integrate chart with existing dashboard layout
- [] Add tooltip interaction for precise values
- [] Verify responsive layout and performance on all screen sizes

Acceptance Criteria

- [] Chart displays 7-day trends clearly
- [] Tooltips provide accurate daily values
- [] Matches app styling and updates dynamically

Labels

dashboard, chart, data-visualization, enhancement, smoke-log

Estimate

M

markdown

Title

Add Monthly Usage Summary with Heatmap Calendar

Description

Introduce a heatmap-style calendar view to visually summarize inhalation activity across each month. Each day should be color-coded based on total inhalation time or puff count, helping users identify patterns in their usage.

Tasks

- [] Design heatmap calendar UI component
- [] Calculate daily summaries for the selected month
- [] Implement date navigation (prev/next month)
- [] Apply color gradient based on usage intensity
- [] Ensure accessibility and responsive layout

Acceptance Criteria

- [] Users can navigate between months
- [] Each day cell color represents usage intensity
- [] Includes a legend for interpretation
- [] Data updates as expected with backend changes

Labels

dashboard, calendar, data-visualization, enhancement, smoke-log

Estimate

L

markdown

Title

Implement Rolling Usage Graph with Selectable Window Sizes

Description

Introduce a rolling usage graph to help users understand

trends over time. This graph should allow users to toggle between different rolling windows (e.g., 3-day, 7-day, 14-day, 30-day) to see how their inhalation usage is changing over those periods.

Tasks

- [] Create dropdown or toggle for selecting window size
- [] Calculate rolling average/inhalation sum for each window
- [] Update line chart with dynamic data based on selected window
- [] Ensure chart responds to new data and date range filters
- [] Style UI elements consistently with the existing dashboard

Acceptance Criteria

- [] Users can switch between rolling window options
- [] The graph updates immediately with accurate rolling data
- [] Tooltip displays actual vs average usage per point
- [] Works responsively across all supported screen sizes

Labels

chart, data-visualization, enhancement, smoke-log, dashboard

Estimate

M

markdown

Title

Enable Editable Log Entries with Inline Editing

Description

Introduce the ability for users to edit existing inhalation logs directly from the table. Users should be able to click on a log entry and modify the timestamp or length without leaving the page. This feature enhances flexibility and ensures better accuracy of recorded data.

Tasks

- [] Add edit icon/button to each row of the data table
- [] Open editable fields inline or in a modal
- [] Validate input for correct timestamp and length format
- [] Update Firestore record upon save
- [] Refresh table and charts after edits

Acceptance Criteria

- [] Users can update timestamps and lengths for any entry
- [] Updates reflect immediately across all visualizations
- [] Invalid edits are rejected with helpful error messages
- [] UI is intuitive and consistent with design language

Labels

enhancement, UI/UX, data-editing, smoke-log, firestore

Estimate

M

markdown

Title

Add Rolling Average Chart for Daily Usage Trends

Description

Introduce a rolling average chart (e.g., 7-day) that visualizes inhalation trends over time. This will help users identify changes in behavior and long-term patterns more effectively than single-day data points.

Tasks

- [] Add rolling average option to chart type selector
- [] Implement 7-day rolling average calculation on timestamped inhalation logs
- [] Display rolling average alongside existing daily total bars
- [] Style chart for clarity and legibility
- [] Ensure data updates dynamically with new log entries or edits

Acceptance Criteria

- [] Chart accurately reflects rolling average over chosen period
- [] Users can toggle between raw totals and rolling average
- [] Component integrates seamlessly with date range filter
- [] Mobile and desktop layouts display chart correctly

Labels

feature, charting, analytics, smoke-log, enhancement

Estimate

M

markdown

Title

Implement Streak Tracker for Consecutive Smoke-Free Days

Description

Add a visual component to track the user's longest and current streaks of smoke-free days. This feature aims to motivate users by highlighting progress and promoting positive behavior change.

Tasks

- [] Define criteria for a "smoke-free day" (e.g., 0 inhalations logged)
- [] Calculate current and longest streaks based on log data
- [] Display streaks prominently on dashboard
- [] Add tooltip or info icon to explain how streaks are calculated
- [] Ensure streaks update in real-time with new or deleted log entries

Acceptance Criteria

- [] Dashboard displays both current and longest streaks
- [] Calculations update automatically with data changes
- [] Smoke-free days are based on accurate interpretation of log entries

- [] Design is consistent with app's overall aesthetic

Labels

feature, motivation, ui, smoke-log, enhancement

Estimate

S

markdown

Title

Add Log Entry Editing & Deletion Functionality

Description

Allow users to edit or delete previously logged inhalation entries to correct mistakes or update notes. This feature enhances user control and data accuracy.

Tasks

- [] Add "Edit" and "Delete" buttons to each entry in the log list
- [] Implement dialog/modal for editing timestamp, length, and notes
- [] Confirm deletion with a prompt to prevent accidental removal
- [] Ensure backend reflects updates and deletions accurately
- [] Trigger data refresh and visual updates on change

Acceptance Criteria

- [] Users can edit and delete individual entries
- [] Edited entries are updated in charts and summaries
- [] Deletion is confirmed and irreversible once completed
- [] UI interactions are consistent and intuitive

Labels

feature, log-management, ui, enhancement

Estimate

M

markdown

Title

Implement Rolling Average Line on Usage Chart

Description

Add an optional rolling average trendline (e.g., 3-day, 7-day) to the main usage chart to help users visualize changes in their inhalation patterns over time. This feature supports better insights into habits and progress.

Tasks

- [] Add UI selector for rolling average options (None, 3-day, 7-day)
- [] Calculate and overlay the rolling average on the main usage chart
- [] Use distinct styling (e.g., dashed line or different color) for clarity
- [] Ensure the rolling average updates when date range or filters change
- [] Optimize performance for large datasets

Acceptance Criteria

- [] Rolling average appears correctly when selected
- [] Data updates in real-time with filter or date range changes
- [] Chart remains performant and readable

Labels

feature, chart, data-visualization, enhancement

Estimate

M

markdown

Title

Implement Inhalation Streak Tracker with Milestone Celebrations

Description

Introduce a streak tracking feature that monitors daily inhalation logs and visually celebrates milestones like 3-day, 7-day, 14-day, and 30-day streaks. This gamification element encourages consistent usage and builds habit-awareness.

Tasks

- [] Create backend logic to track streaks based on daily log activity
- [] Design and implement front-end display of current streak count
- [] Add visual badges or animations for milestone achievements
- [] Notify user of broken streaks and encourage restart
- [] Include toggle to enable/disable streak feature in settings

Acceptance Criteria

- [] Users can clearly see current streaks and past milestone achievements
- [] Streak resets properly after a missed day
- [] Feature does not affect data submission or performance

Labels

feature, enhancement, user-engagement, gamification

Estimate

M

markdown

Title

Integrate Time-of-Day Analysis for Inhalation Patterns

Description

Enhance user insights by analyzing and visualizing inhalation habits based on the time of day. Users will gain awareness of their peak usage hours, helping identify potential triggers and opportunities for reduction.

Tasks

- [] Parse timestamp data to group inhalations by hour ranges (e.g., morning, afternoon, evening, late night)
- [] Create a chart (e.g., bar or heatmap) showing usage frequency by time-of-day
- [] Add toggles or filters to analyze specific date ranges
- [] Display insights (e.g., “You tend to inhale most in the evening”)

Acceptance Criteria

- [] Users can view inhalation frequency segmented by time-of-day
- [] Data updates dynamically with applied date filters
- [] Feature integrates seamlessly with existing analytics dashboard

Labels

feature, data-visualization, analytics, enhancement

Estimate

M

markdown

Title

Implement Rolling Average Chart for Inhalation Duration

Description

Add a new line chart to visualize the rolling average of inhalation durations over time. This will help users see longer-term trends and smooth out short-term fluctuations.

Tasks

- [] Calculate rolling average for inhalation length (e.g., 7-day moving average)
- [] Create a line chart displaying both raw and averaged data
- [] Add option to adjust the window size (e.g., 3-day, 7-day, 14-day)
- [] Ensure time range filters update the rolling average chart accordingly

Acceptance Criteria

- [] Rolling average is accurately calculated and plotted
- [] Users can toggle between different rolling window sizes
- [] The chart is responsive to other filters (time range, inhalation count, etc.)

Labels

feature, data-visualization, enhancement, analytics

Estimate

M

markdown

Title

Refactor Log Detail View for Improved Usability

Description

Revamp the detail view of individual inhalation logs to make it more informative and user-friendly. This view should offer insights into the context of the log, such as time of day, comparison to averages, and any associated notes.

Tasks

- [] Redesign the layout to include clear timestamp, inhalation length, and context
- [] Add visual indicators for whether the log was above/below average
- [] Display notes and tags more prominently
- [] Include a "compare to average" mini-chart or summary

- [] Ensure mobile responsiveness and accessibility
- ### Acceptance Criteria**
- [] Users can easily understand and interpret each log's details
 - [] New layout supports both desktop and mobile
 - [] Visual elements accurately reflect inhalation data in context

Labels
ui/ux, feature, enhancement, redesign

Estimate

M

markdown

Title
Implement Time-Based Filter Controls for Log Chart View

Description
Introduce time-based filtering options in the main chart view to allow users to easily analyze their inhalation patterns over specific timeframes (e.g., Today, This Week, Last 30 Days, Custom Range).

Tasks

- [] Add preset time filters (Today, This Week, Last 7 Days, Last 30 Days)
- [] Implement a custom date range picker
- [] Ensure the chart and stats update dynamically based on selected range
- [] Persist selected range across sessions (optional)

Acceptance Criteria

- [] User can select a timeframe and view only relevant log data
- [] Filter controls are responsive and accessible
- [] Changes to timeframe are reflected instantly in the

visualizations

Labels

feature, ui/ux, filters, enhancement

Estimate

M

markdown

Title

Enhance Chart Legends with Interactive Highlights and Descriptions

Description

Update the chart legends to be interactive and informative.

When a user hovers over or taps a legend item, highlight the corresponding data on the chart and show a tooltip with a short description of what it represents (e.g., "Cumulative Duration", "Avg Length per Inhalation").

Tasks

- [] Refactor chart legend to support hover/tap interaction
- [] Implement tooltip system for legend items
- [] Highlight corresponding chart elements when legend items are interacted with
- [] Ensure accessibility for keyboard and screen reader users

Acceptance Criteria

- [] Legends are clickable/tappable and highlight chart sections
- [] Tooltips accurately describe each metric
- [] No visual glitches on mobile or desktop

Labels

enhancement, ui/ux, charts, accessibility

Estimate

M

markdown

Title

Optimize Inhalation Log Input Flow for Speed and Accuracy

Description

Redesign the inhalation log input UI to streamline the process of recording entries. The goal is to make it faster and more intuitive for users to enter duration and context without needing to navigate away or perform multiple taps.

Tasks

- [] Combine duration entry and context selection into a single modal
- [] Add optional voice input for quick logging
- [] Include smart suggestions based on time of day and previous entries
- [] Validate input before submission with clear error messaging

Acceptance Criteria

- [] Users can complete a log entry in 5 seconds or less
- [] Input UI is mobile-optimized and fully functional on small screens
- [] Validation handles edge cases gracefully
- [] Logs are saved correctly to Firestore and reflected in real-time analytics

Labels

enhancement, ui/ux, mobile, firestore

Estimate

M

markdown

Title

Implement Cumulative Usage Chart with Daily, Weekly, Monthly Views

Description

Design and implement a chart component that shows cumulative smoke inhalation over time. Provide toggles or selectors for users to switch between daily, weekly, and monthly cumulative views.

Tasks

- [] Create reusable chart widget for cumulative data
- [] Implement logic to compute cumulative values from Firestore logs
- [] Add toggle buttons for Daily, Weekly, Monthly range
- [] Ensure chart updates in real-time with new log entries

Acceptance Criteria

- [] Chart renders correctly with accurate cumulative totals
- [] Toggle controls switch views instantly with minimal delay
- [] Responsive layout supports mobile and desktop
- [] Data updates in real time from Firestore changes

Labels

feature, charting, firestore, data-visualization

Estimate

M

markdown

Title

Add Time-of-Day Analysis Chart for Smoking Habits

Description

Introduce a new chart that visualizes inhalation patterns across different times of day (morning, afternoon, evening,

night) based on historical log data. This will help users understand when they tend to smoke the most.

Tasks

- [] Group inhalation logs by time-of-day segments
- [] Design a bar or pie chart to display usage by time segment
- [] Integrate chart into the dashboard with proper styling
- [] Validate data against timestamp parsing and time zone handling

Acceptance Criteria

- [] Chart shows clear breakdown of inhalation lengths by time-of-day
- [] Data correctly reflects user's local time zone
- [] UI is responsive and accessible
- [] Works seamlessly with real-time log updates

Labels

feature, charting, behavior-analysis, time-segmentation

Estimate

M

markdown

Title

Implement Rolling Weekly Average Chart for Smoking Patterns

Description

Add a new line chart to show rolling 7-day average inhalation length and frequency over time. This feature will help users identify gradual changes in their smoking habits.

Tasks

- [] Calculate rolling 7-day average for inhalation length and count
- [] Build dual-line chart to display both metrics over time
- [] Add controls for toggling between raw data and rolling

average view

- [] Ensure chart integrates with date range filters

Acceptance Criteria

- [] Users can toggle rolling average view on/off
- [] Chart updates dynamically with date range selections
- [] Visual distinction between raw and rolling data lines
- [] Accurate and performant calculations for large datasets

Labels

feature, charting, trends, data-visualization

Estimate

L

markdown

Title

Add Inhalation Duration Histogram

Description

Introduce a histogram chart that visualizes the distribution of inhalation durations over a selected time period. This feature helps users better understand the frequency of short, medium, and long puffs.

Tasks

- [] Implement histogram calculation from inhalation duration data
- [] Group durations into configurable buckets (e.g., 0-1s, 1-2s, etc.)
- [] Display histogram with appropriate labels and scales
- [] Sync with existing date range filters
- [] Allow toggle between histogram and existing line charts

Acceptance Criteria

- [] Histogram accurately reflects inhalation duration distributions
- [] Responsive and styled to match the rest of the dashboard

- [] Works with date filters and doesn't impact performance
- [] Optional UI control to customize bucket size

Labels

feature, data-visualization, analytics, enhancement

Estimate

M

markdown

Title

Implement Rolling Average Usage Chart

Description

Add a new chart that displays a rolling average of inhalation durations to help users identify trends over time and smooth out short-term fluctuations in usage.

Tasks

- [] Calculate rolling average of inhalation durations using a user-selectable window (e.g., 1 day, 3 days, 7 days)
- [] Integrate the rolling average chart into the existing dashboard layout
- [] Add toggle to switch between raw and smoothed data
- [] Ensure the chart updates dynamically with date range filters
- [] Optimize performance for large datasets

Acceptance Criteria

- [] Rolling average is computed correctly and displayed smoothly
- [] Toggle allows easy comparison between raw and average data
- [] Works across all date ranges without performance issues
- [] Styling consistent with other visualizations

Labels

feature, data-visualization, analytics, enhancement

Estimate

M

markdown

Title

Enable Session-Based Analysis and Visualization

Description

Introduce functionality to group inhalation logs into "sessions" based on a time gap threshold (e.g., 10 minutes of inactivity signifies a new session). This will allow users to analyze their usage habits at a session level instead of individual inhales.

Tasks

- [] Define and implement session detection logic based on timestamp gaps
- [] Display session count and total duration in the analytics summary
- [] Add a new chart to visualize sessions per day/week/month
- [] Allow users to configure the session gap threshold (e.g., 5, 10, 15 minutes)
- [] Update documentation to explain session logic

Acceptance Criteria

- [] Sessions are accurately identified using configurable gap thresholds
- [] Users can view analytics and charts based on sessions
- [] No performance issues when recalculating sessions across large datasets
- [] Feature integrates seamlessly with existing filters and visualizations

Labels

feature, analytics, data-processing, enhancement

Estimate

L

markdown

Title

Add Inhale Pattern Recognition and Tagging

Description

Implement pattern recognition to automatically tag common inhale patterns (e.g., short bursts, long deep inhales, cluster sessions) using machine learning or heuristic rules. These tags can be used to surface insights about user behavior and smoking trends.

Tasks

- [] Define heuristic rules or ML approach for recognizing inhale patterns
- [] Create tags like "Short Burst", "Deep Inhale", "Cluster", etc.
- [] Add logic to apply tags during data import or log submission
- [] Display pattern tags in the log detail view and analytics
- [] Enable filtering logs by tag

Acceptance Criteria

- [] Tags are applied consistently and update in real-time with new data
- [] Users can filter and analyze inhale logs by tag
- [] Clear visual cues and definitions for each tag type in the UI

Labels

feature, tagging, analytics, enhancement

Estimate

L

markdown

Title

Implement Rolling Averages and Trend Lines for Inhalation Data

Description

Add rolling average and trend line options to inhalation charts to help users visualize short-term vs. long-term patterns and better understand fluctuations in usage. These analytics features can be toggled on/off within the chart settings.

Tasks

- [] Add options for 3-day, 7-day, and 30-day rolling averages
- [] Implement linear regression trend line
- [] Create toggle controls for enabling/disabling each overlay
- [] Ensure overlays adapt dynamically to filtered date ranges
- [] Update chart legend and documentation to explain each line

Acceptance Criteria

- [] Trend and average lines render correctly and align with filtered data
- [] Users can customize which lines are visible
- [] Clear visual distinction between raw data and calculated overlays

Labels

feature, analytics, charting, enhancement

Estimate

M

markdown

Title

Add Daily and Weekly Usage Summary Cards

Description

Introduce summary cards at the top of the dashboard that display daily and weekly statistics such as total inhalations, average duration, and peak usage time. These summaries provide quick insights and make it easier to track habits at a glance.

Tasks

- [] Design responsive summary card components
- [] Calculate daily and weekly totals and averages from log data
- [] Highlight peak usage time based on frequency
- [] Integrate cards into dashboard layout
- [] Style for visual clarity and accessibility

Acceptance Criteria

- [] Summary cards update based on selected date range
- [] All statistics are accurate and match data shown in charts
- [] Cards are mobile-friendly and accessible

Labels

feature, dashboard, ui, enhancement

Estimate

M

markdown

Title

Implement Smoke-Free Streak Tracker Feature

Description

Introduce a visual streak tracker that shows the number of consecutive smoke-free days. This motivational tool should be prominently displayed and reset automatically upon logging a

new session.

Tasks

- [] Design a streak tracker component for the dashboard
- [] Calculate streaks based on daily logs
- [] Handle edge cases (e.g., time zone differences, missed logs)
- [] Provide feedback or rewards for long streaks (optional)
- [] Add option to disable or hide streak tracker

Acceptance Criteria

- [] Streak counter updates accurately based on log entries
- [] UI reflects streak status clearly and accessibly
- [] Functionality is covered by unit tests

Labels

feature, motivation, dashboard, enhancement

Estimate

M

markdown

Title

Add Inhalation Duration Histogram to Analytics Dashboard

Description

Create a new histogram chart on the analytics dashboard to visualize the distribution of inhalation durations. This will help users identify typical usage patterns and outliers.

Tasks

- [] Create histogram chart component using existing chart library
- [] Aggregate inhalation durations from logs
- [] Configure bins and intervals dynamically based on data range
- [] Add controls to filter data by time range
- [] Ensure responsiveness across devices

Acceptance Criteria

- [] Histogram correctly visualizes inhalation duration distribution
- [] Filtering by time range updates the chart in real-time
- [] Works across screen sizes and retains clarity

Labels

feature, analytics, data-visualization

Estimate

M

markdown

Title

Add Daily Usage Summary Card to Dashboard

Description

Implement a summary card on the main dashboard that displays total inhalation events and total duration for the selected day. This will give users a quick overview of their daily activity without needing to interpret full charts.

Tasks

- [] Design UI component for daily summary card
- [] Calculate total events and total duration for selected day
- [] Connect to date picker or dashboard date range selector
- [] Ensure component updates dynamically with selected date
- [] Style the card to match existing dashboard components

Acceptance Criteria

- [] Daily usage summary displays correct total events and duration
- [] Summary updates based on selected day
- [] Responsive design and consistent with dashboard aesthetics

Labels

feature, UI, dashboard

Estimate

S

markdown

Title

Implement Rolling Averages in Usage Charts

Description

Enhance the inhalation duration and frequency charts by adding optional rolling average lines (e.g., 7-day, 30-day). This helps users identify long-term trends and smooth out short-term fluctuations.

Tasks

- [] Add toggle switch to chart settings for enabling rolling average
- [] Compute 7-day and 30-day rolling averages of inhalation duration and event count
- [] Overlay rolling average lines on existing charts
- [] Add legend and tooltip support for average lines
- [] Ensure averages adjust dynamically with the selected date range

Acceptance Criteria

- [] Users can toggle rolling average display on each chart
- [] Rolling averages update correctly based on selected range
- [] Visualization is clear and does not obstruct original data

Labels

enhancement, data-visualization, UX

Estimate

M

markdown

Title

Create Onboarding Tutorial Modal for First-Time Users

Description

Introduce a modal-based onboarding tutorial that activates when a user first opens the app. This modal should walk users through the core features of AshTrail, such as logging events, reviewing trends, and interpreting analytics.

Tasks

- [] Design multi-step modal with brief explanations of app features
- [] Detect first-time users using local storage or user profile flags
- [] Allow users to skip or revisit the tutorial later
- [] Add animations or visuals to guide user focus
- [] Test for accessibility and responsiveness

Acceptance Criteria

- [] Tutorial appears only once for new users unless manually reopened
- [] Each tutorial step is clear and visually highlights the feature being explained
- [] Tutorial can be completed or skipped without disrupting app use

Labels

feature, onboarding, UX, enhancement

Estimate

M

markdown

Title

Implement Local Caching for Offline Log Entry

Description

Enable users to add new inhalation log entries while offline.

These entries should be stored locally and automatically synced to Firestore when the device regains internet connectivity.

Tasks

- [] Implement local cache using Isar for offline storage of new entries
- [] Detect network status and trigger sync on reconnection
- [] Ensure duplicate prevention during sync
- [] Notify user when logs are saved locally and when they're synced

Acceptance Criteria

- [] Users can submit logs without internet
- [] Logs sync correctly once online
- [] No duplicates or data loss during sync
- [] Visual feedback provided on sync status

Labels

feature, offline-support, enhancement, firebase

Estimate

L

markdown

Title

Create Cumulative Inhalation Duration Chart

Description

Add a new chart that displays the cumulative duration of inhalation over time. This will help visualize trends in overall usage, highlighting periods of increase or reduction

in total smoking behavior.

Tasks

- [] Design a cumulative line chart to plot total inhalation duration over time
- [] Integrate the chart into the analytics dashboard
- [] Allow filtering by date range and optionally by time of day
- [] Use existing Firestore data to calculate cumulative sums

Acceptance Criteria

- [] Chart reflects accurate cumulative totals
- [] User can filter and zoom into specific time ranges
- [] Chart updates dynamically when new data is submitted

Labels

feature, data-visualization, enhancement, analytics

Estimate

M

markdown

Title

Implement Rolling Average Inhalation Chart

Description

Introduce a rolling average chart to display smoothed trends in inhalation length over time. This visualization will help reduce noise and highlight behavior patterns, especially during periods of irregular use.

Tasks

- [] Calculate rolling average for inhalation duration (e.g., 7-day window)
- [] Add toggle to switch between raw and rolling average views
- [] Integrate the chart into the dashboard alongside the cumulative view

- [] Ensure responsiveness and filter compatibility
- ### Acceptance Criteria**
- [] Chart displays smoothed trends without replacing original data
 - [] Rolling window can be adjusted (e.g., 3-day, 7-day, 14-day)
 - [] Works with existing filters and date range selectors

Labels

feature, enhancement, chart, analytics

Estimate

M

markdown

Title

Design and Implement Session Summary Card Component

Description

Add a session summary card to the dashboard that presents key stats from the selected timeframe. The card should display total sessions, total inhalation time, average inhalation duration, and peak usage time.

Tasks

- [] Create a reusable card component to show summary stats
- [] Display total sessions and cumulative inhalation time
- [] Show average inhalation length and peak hour of use
- [] Sync with date range filters and user settings

Acceptance Criteria

- [] Card updates dynamically with selected filters
- [] Styling is consistent with app theme
- [] Values update in real-time on data change

Labels

feature, UI, dashboard

Estimate

M

markdown

Title

Implement Filter Toggle for Data Views (Daily / Weekly / Monthly)

Description

Add a toggle control to switch between different aggregation views: Daily, Weekly, and Monthly. This control should affect all relevant visualizations and summary statistics in the Smoke-Log app.

Tasks

- [] Design and implement toggle UI for selecting data view granularity
- [] Integrate the toggle with backend data processing logic
- [] Ensure visualizations update according to selected view
- [] Persist user's last selected view preference (optional)

Acceptance Criteria

- [] Toggle is clearly visible and user-friendly
- [] Visualizations and summary stats reflect selected aggregation
- [] No UI or data errors occur when switching views

Labels

feature, enhancement, filters, dashboard

Estimate

M

markdown

Title

Implement "Cumulative Usage" Line Graph with Custom Date Range Selector

Description

Create a new chart on the Smoke-Log dashboard showing cumulative usage over time. This graph should allow the user to select a custom date range to zoom in on specific periods of use. The chart should update in real-time with the selected data, and default to showing all available data.

Tasks

- [] Build cumulative usage data transformation logic
- [] Implement responsive line graph using the charting library
- [] Add custom date range picker with start/end selectors
- [] Ensure chart updates based on selected date range
- [] Apply visual polish to match new app design

Acceptance Criteria

- [] Line graph accurately reflects cumulative usage
- [] Date range selector updates the graph without reload
- [] Handles edge cases (e.g., empty data, single day range)

Labels

feature, chart, enhancement, UX

Estimate

M

markdown

Title

Enhance Session Table with Filters and Sorting

Description

Improve the data table displaying Smoke-Log sessions by adding filter options (e.g., by date range, duration, or length) and

sortable columns. This will allow users to more easily explore and analyze their logs.

Tasks

- [] Add filtering capabilities to the session table
- [] Implement sortable columns (timestamp, length, duration)
- [] Ensure table updates in real-time with filtered data
- [] Maintain performance and responsiveness

Acceptance Criteria

- [] Users can filter by date range, length, or duration
- [] Columns are sortable in ascending/descending order
- [] All updates happen client-side for speed
- [] Design matches app styling

Labels

feature, table, data, UX

Estimate

M

markdown

Title

Add Rolling Usage Averages to Line Chart

Description

Implement a rolling average feature for the inhalation usage line chart to help visualize trends over time. This should allow users to toggle between raw data and a smoothed average using predefined time windows (e.g., 7-day, 30-day).

Tasks

- [] Add UI toggle for rolling average display
- [] Implement backend or client-side logic for rolling average calculation
- [] Allow users to select time windows (e.g., 7d, 14d, 30d)
- [] Ensure chart updates reflect selected average in real time

- [] Maintain clear labeling to differentiate between raw and averaged data

Acceptance Criteria

- [] Users can toggle between raw data and rolling average
- [] Chart updates dynamically without reload
- [] Rolling average lines are visually distinct
- [] Performance is unaffected for typical datasets

Labels

enhancement, chart, data-visualization, UX

Estimate

M

markdown

Title

Implement Search and Filter Controls for Smoke Log Table

Description

Add search and filtering functionality to the Smoke Log table to improve usability and allow users to quickly find specific entries. Filters should include date range, inhalation length thresholds, and text search for notes or tags (if applicable).

Tasks

- [] Add a search bar for keyword filtering
- [] Add date range selector to filter logs by timestamp
- [] Add inhalation length filter (e.g., greater than, less than)
- [] Ensure filters apply in combination and update table in real-time
- [] Maintain responsiveness on mobile devices

Acceptance Criteria

- [] Users can filter logs using multiple criteria simultaneously
- [] Table updates dynamically as filters are applied

- [] All filter inputs are accessible and styled consistently
- [] Performance remains acceptable with large datasets

Labels

feature, UI/UX, table, filtering

Estimate

M

markdown

Title

Create Monthly and Weekly Usage Summary Views

Description

Implement summary views for users to analyze their smoking patterns over weekly and monthly timeframes. These summaries should include total inhalations, average length, cumulative usage, and a chart visualization.

Tasks

- [] Design a summary card for weekly and monthly views
- [] Implement data aggregation logic based on selected range
- [] Add chart to visualize trends over time
- [] Integrate summary views with range selector
- [] Ensure summaries update dynamically with data changes

Acceptance Criteria

- [] Users can switch between weekly and monthly views
- [] Summary includes total sessions, cumulative length, and average inhalation
- [] Chart displays trendlines or bar graph representation
- [] Responsive and mobile-friendly layout

Labels

feature, analytics, charting, summary

Estimate

M

markdown

Title

Implement Editable Notes for Each Smoke Log Entry

Description

Add functionality allowing users to attach and edit a note for each individual smoke log entry. Notes can include context, mood, situation, or anything the user wants to remember.

Tasks

- [] Add a `note` field to each smoke log record in the database
- [] Update the UI to show a note icon/button for each entry
- [] Implement modal or inline editing for notes
- [] Support viewing and editing existing notes
- [] Ensure note changes persist to the backend

Acceptance Criteria

- [] Users can view, add, or edit a note tied to a specific log
- [] Notes are saved and displayed correctly on refresh
- [] Design does not interfere with primary log visualization
- [] UX is smooth and responsive across devices

Labels

feature, ui/ux, data-entry, enhancement

Estimate

M

markdown

Title

Add Tagging System for Smoke Log Entries

Description

Implement a system that allows users to tag their smoke log entries with custom labels like "stress", "celebration", "social", "craving", etc. This feature should enhance trend discovery and filtering capabilities.

Tasks

- [] Define a schema for storing tags in each log entry
- [] Add UI components to assign tags during or after logging
- [] Allow users to create and manage custom tags
- [] Implement filtering of log entries by selected tags
- [] Style tags to be easily readable and aesthetically pleasing

Acceptance Criteria

- [] Users can assign one or more tags to each log
- [] Users can create new tags and reuse existing ones
- [] Logs can be filtered by tag in the UI
- [] Tags persist across sessions and are stored correctly

Labels

feature, data-tagging, ui/ux, enhancement

Estimate

M

markdown

Title

Implement Log Entry Detail View with Edit and Delete Options

Description

Create a detailed view for individual smoke log entries that allows users to see extended information and perform actions such as editing or deleting the entry.

Tasks

- [] Design and implement a detail view modal or screen

- [] Display full timestamp, duration, tags, and any notes
- [] Add "Edit" and "Delete" options with confirmation dialogs
- [] Ensure changes update the data store and UI correctly
- [] Log user interactions for future analytics

Acceptance Criteria

- [] Users can access detailed views of any log entry
- [] Users can edit or delete entries with confirmation
- [] Edits and deletions are reflected immediately in the log history
- [] UI is intuitive and responsive across device types

Labels

feature, ui/ux, edit-delete, enhancement

Estimate

M

markdown

Title

Integrate Range Selector with All Visualizations

Description

Unify the date range selector functionality across all visualizations in the AshTrail app, ensuring that the selected time range consistently filters charts, graphs, and log summaries.

Tasks

- [] Refactor existing range selector component to emit a global state update
- [] Connect range selection to all charts and visualizations (e.g., usage trends, cumulative totals)
- [] Ensure performance is maintained for large datasets
- [] Test behavior across different time ranges (e.g., 1 day, 7 days, custom)
- [] Add visual indicators showing the currently active range

Acceptance Criteria

- [] All charts reflect the selected time range
- [] Changing the range updates data without requiring a full reload
- [] The UI clearly communicates what time range is being analyzed
- [] Works on both mobile and desktop views

Labels

feature, ui/ux, data-visualization, enhancement

Estimate

M

markdown

Title

Implement Rolling Average Toggle for Usage Charts

Description

Add a toggle option to apply a rolling average to the usage charts. This will help users visualize trends over time by smoothing out short-term fluctuations in the inhalation data.

Tasks

- [] Add a toggle control to the UI for enabling/disabling rolling averages
- [] Compute rolling average based on selected window (e.g., 7-day)
- [] Update chart rendering logic to display the smoothed data when toggled
- [] Ensure both original and averaged data can be easily compared
- [] Test performance impact with large datasets

Acceptance Criteria

- [] Toggle is clearly visible and intuitive
- [] Switching the toggle updates the chart without reloading

the page

- [] Users can see both raw and averaged data on demand
- [] Works seamlessly with date range selector

Labels

feature, data-visualization, enhancement, frontend

Estimate

M

markdown

Title

Create Daily Summary Section Below Charts

Description

Implement a section below the charts that summarizes the total inhalation time and count per day for the selected date range. This will give users a quick overview of their usage patterns.

Tasks

- [] Add a daily summary table below the charts
- [] Include columns for Date, Total Duration, and Number of Sessions
- [] Ensure the table respects the selected date range filter
- [] Format dates and durations in a readable format
- [] Make table sortable by each column

Acceptance Criteria

- [] Summary table appears below charts and updates with date range changes
- [] Data shown is accurate and matches chart data
- [] Table is visually consistent with the rest of the UI
- [] No performance lag when changing date ranges

Labels

feature, enhancement, UI, analytics

Estimate

M

markdown

Title

Enable Multi-Day Range Selection in Date Picker

Description

Update the date picker to support multi-day range selection, allowing users to view logs and analytics across multiple consecutive days.

Tasks

- [] Modify the date picker to support selecting a start and end date
- [] Ensure all components using the selected date update accordingly
 - [] Update chart and table queries to fetch data within the selected range
 - [] Highlight active range clearly in the UI

Acceptance Criteria

- [] Users can select a range of dates instead of a single day
- [] Charts and summary tables update to reflect the selected range
- [] UI remains intuitive and visually clean
- [] No performance degradation for longer date ranges

Labels

feature, enhancement, UI, date-range

Estimate

M

markdown

Title

Add Cumulative Usage Chart with Dynamic Range Adaptation

Description

Implement a new cumulative usage chart that visualizes total inhalation time over a user-defined range. This chart should dynamically adapt to the selected time frame and update accordingly when the user adjusts the date range.

Tasks

- [] Design a cumulative usage line chart that plots total usage over time
- [] Integrate it with the existing date range selector
- [] Fetch and aggregate usage data for the selected range
- [] Display total inhalation time in chart legend or tooltip
- [] Ensure mobile responsiveness and chart clarity

Acceptance Criteria

- [] Chart loads and displays correct cumulative data for the selected time frame
- [] Changing the range updates the chart data in real time
- [] All data points are accurate and readable
- [] Chart works across all supported device sizes

Labels

feature, visualization, analytics, chart

Estimate

M

markdown

Title

Implement Data Table View for Inhalation Records

Description

Create a table view that displays individual inhalation records including timestamp, duration, and any associated

metadata. This table should appear below the charts and update according to the selected time range.

Tasks

- [] Design a responsive table component
- [] Integrate the table with the date range selector
- [] Fetch inhalation data based on the selected range
- [] Sort entries in descending order by timestamp
- [] Display relevant columns (e.g., Date, Time, Duration)

Acceptance Criteria

- [] Table loads correct data based on selected date range
- [] Entries are sorted newest to oldest
- [] Table is scrollable on mobile and readable across devices
- [] No performance issues with large datasets

Labels

feature, UI, data, enhancement

Estimate

M

markdown

Title

Add Total Inhalation Duration Summary per Day

Description

Provide users with a clear summary of their total inhalation duration for each day within the selected time range. This summary should be prominently displayed above the data table and charts.

Tasks

- [] Calculate daily total inhalation durations from the dataset
- [] Display a summary section above the table and chart area
- [] Format the data for readability (e.g., “July 27: 6m

```
45s")
- [ ] Update dynamically with date range selection

### Acceptance Criteria
- [ ] Summary updates correctly with selected time ranges
- [ ] Display is user-friendly and highlights daily totals
- [ ] Compatible with mobile and desktop layouts

### Labels
feature, enhancement, UI, analytics

### Estimate
S
```

markdown

```
### Title
Implement Cumulative Inhalation Duration Line Chart

### Description
Add a new line chart to visualize cumulative inhalation duration over time within the selected date range. This helps users observe long-term trends and personal smoking behavior patterns.

### Tasks
- [ ] Compute cumulative sum of inhalation durations across the selected date range
- [ ] Plot line chart using Chart.js (or current charting library)
- [ ] Ensure chart updates dynamically with date range selection
- [ ] Add toggle option to switch between daily totals and cumulative view

### Acceptance Criteria
- [ ] Chart accurately displays cumulative data
- [ ] Updates in response to user's date range filters
- [ ] Toggle between chart views functions without errors
```

Labels
feature, chart, enhancement, data-visualization

Estimate
M

markdown

Title
Implement Toggleable Chart Filters for Usage Analysis

Description
Introduce a chart filter component that allows users to switch between multiple data visualizations: daily totals, cumulative usage, and rolling averages. This improves usability by consolidating chart type selection into a single intuitive interface.

Tasks

- [] Create a dropdown or segmented control to switch chart views
- [] Integrate filter logic into the chart rendering function
- [] Default to daily totals on load
- [] Ensure selections persist while navigating within the app session

Acceptance Criteria

- [] Chart updates accurately based on selected filter
- [] User can switch between views without reloading the page
- [] UI control is responsive and fits within the existing design

Labels
feature, ui, chart, enhancement, user-experience

Estimate
M

markdown

Title

Add "Time Since Last Session" Indicator

Description

Enhance the dashboard by showing how much time has passed since the user's last recorded session. This feature helps users monitor their smoking habits and maintain awareness of time gaps between uses.

Tasks

- [] Retrieve the timestamp of the latest session
- [] Calculate time elapsed since that session
- [] Display the result in a user-friendly format (e.g., "2 hours ago")
- [] Style the indicator to match the current UI theme

Acceptance Criteria

- [] Time is accurately calculated and updates in real time
- [] UI element is non-intrusive but noticeable
- [] Works with both live and historical session data

Labels

feature, dashboard, ui, enhancement, health

Estimate

S

markdown

Title

Implement Animated Transition for Range Selector Filters

Description

Add smooth animated transitions when the user selects different time ranges in the Range Selector (e.g., Day, Week, Month). This will enhance user experience by making changes to

the chart and data feel more dynamic and intuitive.

Tasks

- [] Define animation logic for range changes (e.g., fade or slide)
- [] Apply animation to chart updates
- [] Apply animation to filtered data view below chart
- [] Ensure performance remains smooth on all supported devices

Acceptance Criteria

- [] Animation plays correctly when switching between any two ranges
- [] No visual glitches or lag
- [] Works with both short and long datasets

Labels

enhancement, ui, animation, chart, filter

Estimate

M

markdown

Title

Create Summary Cards for Daily, Weekly, and Monthly Usage Stats

Description

Design and implement summary cards that display key metrics such as total usage duration, average session length, and peak usage times. These cards should update based on the selected time range (Day, Week, Month) and appear above or beside the main chart.

Tasks

- [] Define key metrics to include in summary cards
- [] Implement responsive layout for displaying cards on various screen sizes

- [] Bind summary card data to filtered chart data
 - [] Style cards to match overall app aesthetic
- ### Acceptance Criteria
- [] Cards update dynamically based on selected time range
 - [] Metrics reflect accurate, real-time data
 - [] Layout is mobile- and desktop-friendly

Labels

enhancement, ui, dashboard, analytics, stats

Estimate

M

markdown

Title

Add Toggle to Switch Between Chart Types (Line vs. Bar)

Description

Introduce a toggle or dropdown that allows users to switch between different chart types—initially a line chart and a bar chart—for visualizing inhalation usage data. This provides flexibility for users to interpret the data in their preferred format.

Tasks

- [] Create UI toggle/dropdown to select chart type
- [] Implement logic to render either a line or bar chart based on selection
- [] Maintain consistent data filtering across chart types
- [] Test for responsiveness and visual clarity

Acceptance Criteria

- [] Users can toggle between line and bar charts seamlessly
- [] Data remains consistent regardless of chart type
- [] Charts render properly on all devices

Labels

feature, ui, chart, visualization

Estimate

M

markdown

Title

Implement Summary Statistics Panel for Usage Logs

Description

Create a summary panel above the chart to display key statistics for the selected time range. This includes total puffs, average inhalation length, and total session time.

These metrics help users gain immediate insights at a glance.

Tasks

- [] Design layout for the summary panel
- [] Calculate total puffs, average length, and session time dynamically
- [] Ensure stats update when the date range is changed
- [] Style the panel for readability and responsiveness

Acceptance Criteria

- [] Summary panel appears above chart with accurate statistics
- [] Updates in real time based on selected range
- [] Visually integrates with current UI theme

Labels

feature, ui, statistics, enhancement

Estimate

M

markdown

Title

Implement Time-of-Day Usage Heatmap

Description

Develop a heatmap visualization that shows inhalation activity across time of day and days of the week. This will help users identify their most common usage patterns and peak times.

Tasks

- [] Design heatmap component (color-coded by intensity of use)
- [] Transform timestamped logs into a 7x24 data grid
- [] Integrate into the analysis tab or under the chart
- [] Add tooltip on hover showing exact count or duration

Acceptance Criteria

- [] Heatmap accurately reflects usage frequency/intensity by hour and weekday
- [] Responsive layout and accessible color palette
- [] Tooltip shows contextual information

Labels

feature, data-visualization, enhancement, heatmap

Estimate

L

markdown

Title

Add Cumulative Weekly Usage Chart

Description

Implement a line chart that displays cumulative inhalation length across the week, resetting every Sunday. This visualization will help users understand their weekly consumption trends and compare across weeks.

Tasks

- [] Aggregate inhalation data per week, segmented by day
- [] Build a line chart to show daily cumulative total per week
- [] Use color coding or lines to distinguish between weeks
- [] Allow toggling visibility of individual weeks

Acceptance Criteria

- [] Chart clearly shows cumulative usage for each day within the week
- [] Multiple weeks are distinguishable and toggleable
- [] Displays data accurately with respect to selected range

Labels

feature, chart, data-visualization, enhancement

Estimate

M

markdown

Title

Add Rolling 7-Day Average Usage Chart

Description

Create a line chart displaying the rolling 7-day average of daily inhalation length. This will help users observe trends while smoothing out daily fluctuations in usage.

Tasks

- [] Calculate 7-day rolling average for daily inhalation length
- [] Display this as a smooth line on a dedicated chart
- [] Ensure chart dynamically updates based on selected date range
- [] Format tooltip to show average and corresponding date

Acceptance Criteria

- [] Chart accurately reflects 7-day rolling averages

- [] Responds correctly to range selections
- [] Legible and informative tooltips

Labels

feature, analytics, chart, enhancement

Estimate

M

markdown

Title

Display Weekly Summary Statistics Above Charts

Description

Add a compact summary section above the visual charts to display key weekly statistics: total puffs, average length, and number of sessions. This gives users a quick snapshot of their recent usage.

Tasks

- [] Calculate total inhalations, average length, and session count for the past 7 days
- [] Design a visually clean summary component to display these values
- [] Ensure the summary dynamically updates with date range selector
- [] Highlight increase or decrease compared to previous week

Acceptance Criteria

- [] Summary updates correctly based on selected date range
- [] Weekly comparison clearly shows trends (e.g., arrow up/down with %)
- [] Clean and consistent with overall design aesthetic

Labels

feature, UI, enhancement, analytics

Estimate

M

markdown

Title

Implement Chart Data Export Functionality (CSV/JSON)

Description

Allow users to export their filtered chart data (e.g., session length, timestamp) into CSV or JSON format for personal analysis or sharing. This supports transparency and further user-driven insights.

Tasks

- [] Add export button near chart controls
- [] Format visible chart data into CSV and JSON structures
- [] Trigger download with appropriate filename and format
- [] Ensure exported data respects current date range and filters

Acceptance Criteria

- [] Users can download their currently viewed chart data
- [] Export includes all relevant fields with proper formatting
- [] Works seamlessly on desktop and mobile platforms

Labels

feature, export, UX, enhancement

Estimate

S

markdown

Title

Implement User Session Grouping in Log View

Description

Enhance the session list display by grouping log entries into user sessions based on time gaps (e.g., 30+ minutes of inactivity). Each group should be collapsible, showing session start/end time and total duration. This improves readability and provides a clearer structure of user behavior.

Tasks

- [] Define logic for session break threshold (e.g., 30 minutes)
- [] Group logs accordingly in the session list
- [] Display session summary (start, end, duration) at top of each group
- [] Allow expand/collapse of session groups

Acceptance Criteria

- [] Logs are grouped into sessions using a clearly defined gap threshold
- [] Users can expand or collapse session groups to view or hide details
- [] Session summary provides accurate metadata

Labels

feature, UI, usability, enhancement

Estimate

M

markdown

Title

Add Visual Indicators for Inhale Intensity and Duration

Description

To enhance data visualization in the log view, introduce visual cues for inhale intensity and duration. This could include color gradients (e.g., from blue to red) for intensity and bar widths or heights for duration. This helps users quickly assess patterns in usage.

Tasks

- [] Define data thresholds for categorizing intensity and duration
- [] Apply color coding and sizing to visual elements based on thresholds
- [] Update chart and/or session list items with new visuals
- [] Add legend or tooltip for explaining visuals

Acceptance Criteria

- [] Each inhale is represented with an intuitive visual indicator
- [] Users can easily distinguish high vs. low intensity and short vs. long durations
- [] Tooltip or legend provides context for visual indicators

Labels

feature, visualization, enhancement, UX

Estimate

M

markdown

Title

Implement Toggle Between Daily and Weekly Summary Views

Description

To offer users a broader perspective on their usage habits, implement a toggle feature that switches between daily and weekly summaries in the Smoke Log view. This enhances user insight and supports better self-awareness over time.

Tasks

- [] Design UI toggle component for switching views
- [] Update data aggregation logic to support weekly summaries
- [] Ensure visualizations adapt to both daily and weekly contexts

- [] Validate behavior for edge cases (e.g., no data for week)

Acceptance Criteria

- [] Users can toggle between daily and weekly summaries seamlessly
- [] Weekly view aggregates total length and count correctly
- [] Visual consistency and clarity are maintained in both modes

Labels

feature, UX, data-visualization, enhancement

Estimate

M

markdown

Title

Integrate Cumulative Usage Chart with Time Range Selector

Description

Enhance the existing cumulative usage chart by integrating it with the time range selector. This will allow users to view their cumulative smoking data for custom time periods, improving analytical insight and trend recognition.

Tasks

- [] Link chart data source to selected time range
- [] Ensure dynamic updates as the time range changes
- [] Adjust X-axis labeling for various time spans
- [] Test performance with large data ranges

Acceptance Criteria

- [] Cumulative chart updates correctly based on time range selection
- [] No lag or performance issues for long time spans
- [] Accurate cumulative values are displayed

Labels

feature, charting, enhancement, UI

Estimate

M

markdown

Title

Implement Dynamic Data Table with Timestamp Sorting and Filtering

Description

Create a dynamic table beneath the usage charts to display log entries. This table should support sorting by timestamp (default: descending) and filtering based on the selected time range.

Tasks

- [] Render table of smoking logs with columns for timestamp, inhalation length, and any relevant metadata
- [] Implement descending timestamp sort by default
- [] Connect filtering logic to the time range selector
- [] Ensure proper formatting for readability on mobile and desktop views

Acceptance Criteria

- [] Table loads all relevant log entries within the selected time range
- [] Entries are sorted by timestamp (newest first)
- [] UI remains responsive across different screen sizes

Labels

feature, UI, data-table, enhancement

Estimate

M

markdown

Title

Add Cumulative Usage Chart with Daily Totals

Description

Implement a new chart to visualize cumulative smoking usage over time. This chart should show the daily total inhalation length and support comparisons across days.

Tasks

- [] Calculate total inhalation length for each day from Firestore logs
- [] Display a line or bar chart with dates on the x-axis and total usage on the y-axis
- [] Ensure chart updates dynamically based on selected time range
- [] Format dates and usage units for readability

Acceptance Criteria

- [] Daily totals are accurately calculated and displayed
- [] Chart adjusts to filtered date range
- [] Chart renders clearly on mobile and desktop views

Labels

feature, chart, data-visualization, enhancement

Estimate

M

markdown

Title

Implement Rolling Average Usage Chart

Description

Add a chart to visualize rolling average inhalation length over time. This will help users understand their trends and usage patterns by smoothing out daily fluctuations.

Tasks

- [] Calculate rolling average (e.g., 7-day window) of daily usage totals
- [] Display rolling average on a line chart
- [] Allow user to toggle between different window sizes (e.g., 3-day, 7-day, 14-day)
- [] Ensure responsive design for mobile and desktop

Acceptance Criteria

- [] Rolling average calculations are accurate
- [] Chart updates based on selected window
- [] Toggle control is intuitive and functional
- [] Visuals are readable and informative across devices

Labels

feature, chart, data-analysis, enhancement

Estimate

M

markdown

Title

Enhance Log Table with Inline Editing and Deletion

Description

Improve the user experience by allowing users to edit or delete entries directly within the log table. This reduces friction and improves data accuracy by streamlining corrections.

Tasks

- [] Add edit and delete icons/buttons to each row
- [] Implement modal or inline field editing for inhalation length and timestamp
- [] Confirm before deleting an entry
- [] Persist changes to Firestore and update local state

Acceptance Criteria

- [] Users can edit log entries inline or via modal
- [] Deletion prompts a confirmation and updates Firestore
- [] Edited entries reflect updated values in the UI immediately
- [] Interface remains responsive and intuitive

Labels

feature, ui-enhancement, firestore, data-management

Estimate

M

markdown

Title

Create Filter and Search Functionality for Log Table

Description

Add the ability to filter and search log entries by date range and inhalation length to allow users to explore their data more effectively.

Tasks

- [] Add date range picker above the log table
- [] Add a search field to filter by inhalation length (e.g., >2.0s)
- [] Combine filters to work simultaneously
- [] Update displayed logs based on active filters
- [] Ensure filtering works both client-side and with Firestore queries if needed

Acceptance Criteria

- [] Users can filter by custom date range
- [] Users can search for inhalation durations
- [] Filtered results update without page reload
- [] Clear button resets all filters

Labels

feature, enhancement, ux, firestore

Estimate

M

markdown

Title

Implement Chart Toggle Between Daily and Cumulative Views

Description

Add a toggle button or dropdown that allows users to switch between viewing daily inhalation data and cumulative usage over time on the main chart. This provides users with better insight into patterns and long-term behavior.

Tasks

- [] Add UI toggle for 'Daily' vs 'Cumulative' view
- [] Implement logic to compute cumulative data series
- [] Update chart rendering to reflect selected mode
- [] Ensure chart updates without page reload
- [] Maintain consistent color schemes and legends

Acceptance Criteria

- [] Users can switch views with a single click
- [] Cumulative chart shows running total of inhalation duration
- [] Daily chart shows values per day
- [] Mode switch is clearly labeled and intuitive

Labels

feature, chart, ui, enhancement

Estimate

M

markdown

Title

Add Date Range Picker to Filter Log Entries and Charts

Description

Introduce a date range picker component to allow users to filter both the data table and charts by custom timeframes. This enhancement enables better analysis of usage trends over specific periods.

Tasks

- [] Add date range picker UI to main screen
- [] Wire picker to update both chart and table views
- [] Preserve picker state across app sessions if possible
- [] Update Firestore queries to respect selected date range
- [] Display selected date range on charts

Acceptance Criteria

- [] Users can select a custom date range using a calendar interface
- [] Chart and table update to show only data from the selected range
- [] Default range is "Last 30 Days" if nothing is selected
- [] Picker is mobile-responsive

Labels

feature, filter, date-range, ui, firestore

Estimate

M

markdown

Title

Create Rolling Usage Chart for Daily and Weekly Averages

Description

Add a new chart that displays rolling usage over time, with selectable time windows (e.g., 7-day, 30-day). This will help

users visualize trends in their smoking habits by smoothing out daily fluctuations.

Tasks

- [] Add dropdown or toggle to select rolling window (e.g., 7-day, 30-day)
- [] Calculate rolling average of inhalation durations from Firestore data
- [] Display rolling average chart beneath or beside the existing usage chart
- [] Ensure new chart updates in response to date range filters

Acceptance Criteria

- [] User can select a rolling time window
- [] Rolling average is correctly calculated and visualized
- [] Chart updates when new data is submitted or date range is changed
- [] Mobile and desktop layouts are preserved

Labels

enhancement, chart, analytics, ux

Estimate

M

markdown

Title

Implement Session Duration Summary Panel

Description

Introduce a summary panel displaying total session count, average duration per session, and total inhalation time for the selected date range. This panel should be visible above the chart to provide immediate insight into usage metrics.

Tasks

- [] Add UI component for session summary panel

- [] Calculate:
 - Total number of sessions in the date range
 - Average inhalation duration per session
 - Total inhalation duration for the range
- [] Integrate with existing date filter logic
- [] Style panel to match current design theme

Acceptance Criteria

- [] Summary panel displays accurate data based on selected date range
- [] Responsive layout works on both mobile and desktop
- [] Updates automatically when the user adjusts the range or adds new logs

Labels

enhancement, ux, dashboard, analytics

Estimate

M

markdown

Title

Add "Longest Session" Insight to Session Summary Panel

Description

Enhance the session summary panel by including the duration of the longest session within the selected date range. This addition provides users with a better sense of individual session extremes, complementing the average and total duration metrics.

Tasks

- [] Add new metric to session summary panel: "Longest Session"
- [] Update data aggregation logic to compute maximum session length
- [] Display the longest session in mm:ss format for clarity
- [] Ensure UI remains responsive and accessible on all

devices

Acceptance Criteria

- [] "Longest Session" appears in the summary panel with correct formatting
- [] Accurate calculation based on currently selected date range
- [] Does not break layout on mobile or desktop views

Labels

enhancement, analytics, dashboard, ux

Estimate

S

markdown

Title

Implement Rolling Average Selector for Time-Series Chart

Description

Introduce a dropdown or toggle that allows users to apply a rolling average (e.g., 1-day, 3-day, 7-day) to the inhalation duration time-series chart. This feature will help users identify trends and smooth out noise in their usage patterns.

Tasks

- [] Add rolling average calculation to chart data processor
- [] Provide UI element for selecting the rolling average duration
- [] Update the chart dynamically based on selected rolling average
- [] Ensure performance is acceptable with larger data sets

Acceptance Criteria

- [] Rolling average options are available and functional
- [] Chart updates immediately upon changing the average type
- [] Behavior is consistent across all date ranges and devices

Labels
feature, analytics, charting, ux

Estimate
M

markdown

Title
Add Editable Notes Field to Each Log Entry

Description
Allow users to add or edit a short text note for each inhalation log entry. This enables users to record contextual information like triggers, mood, or circumstances surrounding the session.

Tasks

- [] Update Firestore schema to include an optional `notes` field
- [] Add UI for viewing, editing, and saving notes per log entry
- [] Ensure note changes sync correctly across devices
- [] Provide visual indication if a note exists for an entry

Acceptance Criteria

- [] Notes can be added or edited for each log entry
- [] Saved notes persist and are displayed correctly across sessions
- [] Notes are optional and do not impact other functionality

Labels
feature, firestore, ux, enhancement

Estimate
M

markdown

Title

Implement Monthly Summary View with Calendar Selector

Description

Introduce a monthly summary view accessible via a calendar date selector. This view aggregates key stats like total inhalations, average session length, and cumulative duration for each day of the selected month.

Tasks

- [] Add calendar-based date selector to switch between months
- [] Display summarized data per day (e.g., bar graph or heatmap)
- [] Implement logic to calculate and cache monthly totals
- [] Ensure responsive layout on both mobile and desktop

Acceptance Criteria

- [] User can select a month and see daily summaries
- [] All summaries are accurate and visually clear
- [] Transition between months is smooth and performant

Labels

feature, ui, summary, visualization

Estimate

L

markdown

Title

Add Rolling Usage Chart with Adjustable Window

Description

Implement a rolling usage chart that shows average inhalation count or total duration over a user-defined rolling window (e.g., 3-day, 7-day). This enables users to see short-term

trends in their usage.

Tasks

- [] Add UI control to select rolling window duration (dropdown or slider)
- [] Calculate rolling averages based on selected metric (inhalation count or duration)
- [] Update chart in real time based on user selection
- [] Optimize performance for large datasets

Acceptance Criteria

- [] User can choose a rolling window size and see updated graph
- [] Chart clearly shows average usage trends over time
- [] Works across various time ranges (day, week, month)

Labels

feature, chart, analysis, rolling-average

Estimate

M

markdown

Title

Implement Inhalation Summary Table with Sorting and Filtering

Description

Create a data table that displays individual inhalation entries with support for sorting by timestamp, length, and filtering by date range. This helps users review specific logs quickly and correlate them with trends in the chart.

Tasks

- [] Add a DataTable component below the usage chart
- [] Populate it with inhalation entries: timestamp, duration (seconds), and any other metadata
- [] Enable sorting on timestamp and duration columns
- [] Connect filtering logic to the chart's date range

```
selector
- [ ] Ensure new entries appear in real-time if applicable

### Acceptance Criteria
- [ ] User can view and sort inhalation entries
- [ ] Filtering the chart also filters the data table
- [ ] Supports descending timestamp sort by default
- [ ] Table updates automatically as data changes

### Labels
feature, UI, data-table, enhancement

### Estimate
M
```

markdown

```
### Title
Implement Weekly and Monthly Usage Rollups

### Description
Add functionality to aggregate inhalation data into weekly and monthly rollups for better long-term insight. These rollups will power summary charts and analytics, helping users visualize patterns over time.
```

```
### Tasks
- [ ] Create backend logic (or local logic if offline-first) to aggregate data by week and month
- [ ] Store or memoize rollups to avoid recalculation
- [ ] Design UI toggles for Daily, Weekly, and Monthly views
- [ ] Update charts to reflect the selected aggregation level
- [ ] Ensure rollups update when new data is added
```

```
### Acceptance Criteria
- [ ] User can toggle between daily, weekly, and monthly charts
- [ ] Aggregated views display accurate inhalation counts and durations
```

- [] Rollups are responsive to new entries
- [] Smooth transition animations between views

Labels

feature, analytics, chart, enhancement

Estimate

L

markdown

Title

Enable Inhalation Session Annotation Feature

Description

Implement a feature that allows users to annotate individual inhalation sessions with optional notes, tags, or context (e.g., mood, situation, or trigger). This helps track behavioral patterns and supports deeper analysis.

Tasks

- [] Add support for optional notes and tags to each inhalation record
- [] Design UI elements for adding/editing notes and tags
- [] Update Firestore schema to support annotations
- [] Enable search/filtering based on tags or keywords
- [] Reflect annotations in session detail views

Acceptance Criteria

- [] Users can add and edit notes or tags per session
- [] Notes and tags are saved and persisted correctly
- [] Annotations are easily viewable in the session detail UI
- [] Filtering by tag or note keyword works as expected

Labels

feature, annotation, enhancement, UX

Estimate

M

markdown

Title

Implement Multi-Day Trendline Visualization for Usage Patterns

Description

Add a trendline chart feature that visualizes inhalation usage patterns over multiple days, allowing users to identify spikes, streaks, or changes in behavior over time.

Tasks

- [] Create trendline chart component in the analysis view
- [] Aggregate data by day and calculate totals
- [] Allow toggling between daily, weekly, and monthly views
- [] Display average usage benchmarks for comparison
- [] Style chart to match updated app design

Acceptance Criteria

- [] Trendline is responsive and adjusts to selected time range
- [] Data aggregation is accurate and performant
- [] Average usage benchmark is displayed as a secondary line
- [] UI remains readable on different screen sizes

Labels

feature, visualization, chart, UX

Estimate

L

markdown

Title

Enhance Entry Detail View with Timestamp Breakdown and Note Editing

Description

Improve the log entry detail view to include a clearer breakdown of timestamp components (date, start time, end time, duration), and add the ability to edit personal notes associated with each entry directly in the view.

Tasks

- [] Update UI to display date, start, end, and duration in separate fields
- [] Implement in-place editing for user notes
- [] Add validation to prevent saving empty notes
- [] Ensure updates are persisted to Firestore
- [] Test across multiple screen sizes and platforms

Acceptance Criteria

- [] Timestamp is clearly broken down and readable
- [] Notes can be edited inline and saved successfully
- [] Layout is responsive and accessible

Labels

feature, UI, enhancement, notes

Estimate

M

markdown

Title

Add Chart Type Selector and Dynamic Axis Scaling to Log Charts

Description

Introduce a chart type selector (line, bar, area) to allow users to visualize inhalation data in their preferred format. Additionally, implement dynamic Y-axis scaling to accommodate different data ranges more effectively.

Tasks

- [] Add dropdown or toggle for selecting chart type (line,

```
bar, area)
- [ ] Refactor chart rendering to respond to selected chart type
- [ ] Implement dynamic Y-axis scaling based on visible data range
- [ ] Ensure chart legend and tooltips update accordingly
- [ ] Test with varied datasets and display sizes
```

Acceptance Criteria

- [] Users can switch between chart types
- [] Axis scaling adjusts to the visible data range dynamically
- [] Chart responsiveness and interaction remains smooth

Labels

feature, UI, chart, enhancement

Estimate

M

markdown

Title

Integrate Rolling Average Toggle for Log Visualization

Description

Add a toggle feature to apply a rolling average to the inhalation length chart data. This will help users identify trends and smooth out day-to-day fluctuations in their smoking patterns.

Tasks

- [] Add UI toggle to enable/disable rolling average
- [] Implement rolling average calculation (default: 3-day window)
- [] Adjust chart to display both raw data and smoothed line when toggle is active
- [] Ensure tooltip reflects both values appropriately
- [] Test behavior across different data ranges and chart

types

Acceptance Criteria

- [] Users can toggle rolling average on/off
- [] Smoothed line overlays the raw data chart accurately
- [] Tooltips clearly distinguish between raw and average values

Labels

feature, data analysis, enhancement, chart

Estimate

M

markdown

Title

Create Settings Panel for User Preferences

Description

Introduce a dedicated settings panel in the AshTrail app where users can adjust preferences such as default chart view (e.g., line vs. bar), enable/disable dark mode, and toggle analytics features like the rolling average.

Tasks

- [] Design and implement a settings panel accessible from the main UI
- [] Add options for:
 - Default chart type
 - Dark mode toggle
 - Rolling average toggle
- [] Store preferences locally (e.g., SharedPreferences/Isar)
- [] Ensure preferences persist across sessions
- [] Update app behavior dynamically based on settings

Acceptance Criteria

- [] Users can access and modify preferences through the settings panel

- [] Changes are reflected immediately or upon next app load
- [] Preferences persist without requiring reconfiguration

Labels

feature, UI, settings, enhancement

Estimate

M

markdown

Title

Implement Onboarding Flow for First-Time Users

Description

Design and implement an onboarding flow that introduces new users to the core features of the AshTrail (Smoke-Log) app. The flow should guide users through setting up their first log, understanding how to view charts, and configuring optional features like reminders or analytics.

Tasks

- [] Create onboarding UI flow with informative slides or tooltips
- [] Trigger onboarding only on first app launch or until dismissed
- [] Highlight key features: logging a session, viewing history, exploring charts
- [] Offer users the option to set up preferences (e.g., units, dark mode)
- [] Ensure smooth skip and completion functionality

Acceptance Criteria

- [] First-time users are presented with a helpful, clear onboarding experience
- [] Returning users are not shown onboarding again
- [] Users can skip or revisit onboarding from the settings menu

Labels

feature, onboarding, UI, enhancement

Estimate

M

markdown

Title

Add Confirmation Dialog for Log Deletion

Description

To prevent accidental deletion of smoking logs, implement a confirmation dialog that prompts the user before permanently deleting a record.

Tasks

- [] Design a confirmation dialog modal
- [] Trigger dialog on log delete action
- [] Include clear action buttons: "Cancel" and "Delete"
- [] Ensure deletion only occurs after user confirms
- [] Add optional setting to disable confirmation for advanced users

Acceptance Criteria

- [] Deleting a log prompts a confirmation dialog
- [] Users can cancel the deletion without loss of data
- [] Confirmation behavior is consistent across platforms

Labels

enhancement, UX, safety, feature

Estimate

S

markdown

Title

Implement Smoothed Cumulative Usage Chart

Description

Introduce a new visualization in the form of a smoothed cumulative usage chart. This will help users better understand their long-term smoking patterns by displaying a rolling sum or moving average over time, giving insight into periods of increased or decreased usage.

Tasks

- [] Add cumulative sum logic based on timestamp-sorted data
- [] Implement optional smoothing (e.g., 7-day moving average)
- [] Provide toggle controls to switch between raw and smoothed views
- [] Update chart rendering components to support this mode
- [] Validate accuracy of displayed cumulative values

Acceptance Criteria

- [] Users can toggle to view cumulative usage trends
- [] Smoothed data updates accurately based on selected range
- [] Chart remains responsive and clear on all devices

Labels

feature, visualization, data-analysis

Estimate

M

markdown

Title

Create Insights Tab for Personalized Usage Tips

Description

Introduce a new "Insights" tab that provides users with personalized tips based on their smoking data. This will

include contextual messages such as streak achievements, reduction milestones, or suggestions based on peak usage times.

Tasks

- [] Design layout for the Insights tab
- [] Analyze usage data to detect milestones and trends
- [] Generate tips or congratulatory messages based on patterns
- [] Display a summary of longest streaks, average usage per day/week
- [] Ensure data privacy and local-only processing where possible

Acceptance Criteria

- [] Insights are generated dynamically based on user data
- [] Messages update as usage changes over time
- [] No data is shared externally
- [] UI integrates seamlessly with the rest of the app

Labels

feature, analytics, user-experience

Estimate

M

markdown

Title

Add Editable Notes to Log Entries

Description

Enhance each log entry with an optional editable notes field. Users can document context such as emotional state, environment, or reason for the session. This allows for richer insight when reviewing usage trends.

Tasks

- [] Modify the Firestore schema to include a `notes` field

- [] Update the submit UI to allow adding/editing a note
- [] Display the note on log entries in the table and chart tooltips
- [] Add edit/delete functionality for notes
- [] Ensure notes are optional and persist correctly

Acceptance Criteria

- [] Users can add, view, edit, and delete notes per log entry
- [] Notes appear in both the data table and chart tooltips
- [] No crashes or data loss from malformed input

Labels

enhancement, UX, firestore, notes

Estimate

M

markdown

Title

Add Daily Summary View for Usage Insights

Description

Create a new view that summarizes daily usage, including total sessions, total time, average session length, and note highlights. This feature helps users identify patterns and make connections over time.

Tasks

- [] Design a compact daily summary UI component
- [] Calculate daily aggregates (total sessions, duration, averages)
- [] Display prominent notes or keywords from that day's logs
- [] Integrate with date filter and chart timeline
- [] Allow user to click a day to expand detailed view

Acceptance Criteria

- [] Daily summaries show accurate data

- [] Notes are displayed contextually when present
- [] Works across all date ranges and screen sizes
- [] Clicking a summary shows that day's logs in detail

Labels

feature, analytics, summary, UI

Estimate

L

markdown

Title

Implement Filter Controls for Log Data Analysis

Description

Introduce a set of filter controls that allow users to refine their log data views based on time range, length of inhalation, notes content, or custom tags. These filters should dynamically update all charts and tables in the interface to provide targeted insights.

Tasks

- [] Design and implement a filter UI component (dropdowns, sliders, text input)
- [] Wire up filters to update displayed data in charts and tables
- [] Support multi-select and range filtering
- [] Ensure filters work in combination and update instantly
- [] Persist filter state in local storage or URL params

Acceptance Criteria

- [] Users can apply one or multiple filters simultaneously
- [] Filtered results reflect immediately in all visualizations
- [] Filters reset properly when cleared
- [] Filtering logic is thoroughly tested

Labels

feature, filters, UI, interactivity, analytics

Estimate

M

markdown

Title

Add Toggle for Displaying Raw vs. Normalized Inhalation Data

Description

Introduce a toggle button or switch that allows users to view either the raw inhalation length data or the normalized values across charts. This feature will enhance data interpretation, especially when comparing sessions of differing scales.

Tasks

- [] Design and implement a UI toggle for raw/normalized view
- [] Update data preprocessing logic to include normalization (e.g., min-max or z-score)
- [] Apply toggle state across all visualizations (charts, tables)
- [] Store toggle preference in local storage for session persistence

Acceptance Criteria

- [] Users can seamlessly switch between raw and normalized data views
- [] All charts and tables reflect the chosen view instantly
- [] Normalization is accurately and consistently applied
- [] UI clearly indicates which view is active

Labels

feature, UI, data-visualization, enhancement

Estimate

M

markdown

Title

Implement Cumulative Usage Line Chart Over Time Range

Description

Add a new line chart to display cumulative inhalation length over a selected time range. This visualization will help users better understand their smoking trends and overall consumption habits within specific periods.

Tasks

- [] Compute cumulative inhalation data based on timestamp
- [] Create a reusable line chart component for cumulative data
- [] Integrate the chart with existing range selectors and filters
- [] Ensure responsive layout and consistent styling with existing charts

Acceptance Criteria

- [] Cumulative line chart accurately reflects inhalation totals over time
- [] Chart updates dynamically with selected range and filters
- [] Responsive and styled to match other charts in the app

Labels

feature, chart, data-analysis, enhancement

Estimate

M

markdown

Title

Implement Session Summary Card for Selected Time Range

Description

Create a session summary card that displays key metrics for the selected time range, such as total inhalations, average length, longest session, and number of sessions. This will provide users with quick insights into their smoking behavior over time.

Tasks

- [] Define key metrics to display
- [] Query and compute metrics based on selected range
- [] Design and implement the summary card UI component
- [] Integrate with the time range selector and ensure reactive updates

Acceptance Criteria

- [] Summary card displays correct metrics for selected date range
- [] Layout is visually consistent and responsive
- [] Updates automatically with new time range selections

Labels

feature, ui, enhancement, smoke-log

Estimate

M

markdown

Title

Add Static User Selection for Development Mode

Description

Enable the ability to select from predefined static users when the app is in development mode. This feature will streamline development and testing by simulating different user data without requiring authentication.

Tasks

- [] Create environment variable to detect development mode
- [] Define a list of static user profiles

- [] Implement UI dropdown or selector for choosing a static user
- [] Replace dynamic user authentication with selected static user in dev mode
- [] Ensure production mode defaults to standard authentication

Acceptance Criteria

- [] Developers can choose a static user in dev mode
- [] No authentication flow is triggered during development
- [] Static user selection has no effect in production mode

Labels

feature, development-tools, enhancement, smoke-log

Estimate

S

markdown

Title

Redesign Submit Form UI to Match New Design System

Description

Update the user submission form to align with the latest UI/UX design standards introduced in the Smoke-Log redesign. The updated form should improve usability, responsiveness, and consistency across platforms.

Tasks

- [] Implement updated input styles (rounded borders, larger touch targets)
- [] Add label and placeholder updates per design spec
- [] Improve form validation messaging and behavior
- [] Ensure mobile responsiveness and accessibility compliance
- [] Integrate date/time and duration pickers with the new UI style

Acceptance Criteria

- [] All form inputs follow the new visual design guidelines
- [] Form is responsive and usable on various device sizes
- [] Validation messages are user-friendly and consistent
- [] Form components meet WCAG 2.1 AA accessibility standards

Labels

ui, enhancement, smoke-log, accessibility, redesign

Estimate

M

markdown

Title

Implement Filter Panel for Session Analytics

Description

Introduce a collapsible filter panel in the Session Analytics view to allow users to customize the dataset shown in charts and tables. Filters should include date range, duration thresholds, and keyword tags.

Tasks

- [] Create a responsive collapsible filter panel component
- [] Add date range selector with preset ranges (e.g., last 7 days, this month)
- [] Add sliders or input fields for min/max duration filtering
- [] Add multi-select dropdown for tags or session labels
- [] Ensure filters update visualizations and tables dynamically

Acceptance Criteria

- [] Panel opens/closes smoothly on all screen sizes
- [] Filter values persist when navigating between tabs
- [] Filter changes immediately update chart and table views
- [] Filters reset with a single action

Labels

ui, feature, smoke-log, analytics, redesign

Estimate

M

markdown

Title

Design and Implement Summary Cards for Key Stats

Description

Create summary cards at the top of the dashboard to display key usage statistics, such as total sessions, average duration, and longest session over the selected date range. These cards will provide users with a quick overview of their smoking patterns.

Tasks

- [] Design responsive card components
- [] Integrate data from the filtered dataset
- [] Display metrics: Total Sessions, Average Duration, Longest Session
- [] Ensure values update dynamically based on filters

Acceptance Criteria

- [] Summary cards display accurate values
- [] Responsive layout on all screen sizes
- [] Cards update with chart/table filters
- [] Styling consistent with app theme

Labels

ui, enhancement, smoke-log, redesign, dashboard

Estimate

S

markdown

Title

Create Session Distribution Heatmap Visualization

Description

Design and implement a heatmap chart that shows the distribution of smoke sessions across hours and days. This visualization will help users identify patterns in their smoking habits, such as peak usage times.

Tasks

- [] Add a new chart widget to the dashboard layout
- [] Group sessions by weekday and hour
- [] Plot data as a heatmap with gradient color intensity
- [] Include legend and tooltip for readability

Acceptance Criteria

- [] Heatmap accurately reflects session count per hour/day
- [] Color scale and legend clearly communicate density
- [] Tooltip shows exact session count per cell
- [] Responsive and performs well with full-year data

Labels

visualization, enhancement, smoke-log, redesign, analytics

Estimate

M

markdown

Title

Implement Rolling Weekly Usage Line Chart

Description

Add a new line chart to visualize rolling weekly inhalation totals. This will help users see their usage trends over time with smoothing for better pattern recognition.

Tasks

- [] Aggregate inhalation data into rolling 7-day windows
- [] Plot a line chart showing total usage per window
- [] Add tooltip to show date range and total inhalation
- [] Optionally add a trendline or average overlay

Acceptance Criteria

- [] Chart accurately represents weekly rolling totals
- [] Tooltip provides intuitive insights into weekly data
- [] Chart integrates into existing dashboard layout
- [] Handles edge cases (e.g., partial weeks) gracefully

Labels

visualization, analytics, enhancement, smoke-log, redesign

Estimate

M

markdown

Title

Display Usage Distribution by Hour of Day (Heatmap)

Description

Implement a heatmap to visualize the distribution of inhalation events across hours of the day. This will help users identify peak usage periods and detect behavioral patterns.

Tasks

- [] Process inhalation timestamps to count events per hour
- [] Generate a heatmap with days on the Y-axis and hours on the X-axis
- [] Apply color gradient based on usage frequency
- [] Add interactivity: hover tooltip with exact count and time

Acceptance Criteria

- [] Data is correctly grouped and rendered by hour and day
- [] Color intensity reflects usage frequency

- [] Tooltips provide meaningful insight
- [] Chart integrates cleanly into the dashboard

Labels

visualization, analytics, enhancement, smoke-log, redesign

Estimate

M

markdown

Title

Implement Streak Tracking and Visualization

Description

Introduce a feature to track and display usage streaks (e.g., consecutive days without usage or reduced usage). This promotes mindfulness and encourages healthier behavior through visual progress tracking.

Tasks

- [] Define streak logic (e.g., days below threshold or 0 inhalations)
- [] Calculate current and longest streaks
- [] Display streak count on the dashboard
- [] Create a visual component (e.g., fire icon or progress bar) to represent streaks

Acceptance Criteria

- [] Users can see their current and longest streaks
- [] The streak resets appropriately when thresholds are exceeded
- [] Visual representation aligns with the overall UI design

Labels

habit-tracking, visualization, feature, smoke-log, redesign

Estimate

M

markdown

Title

Design and Implement Usage Timeline Heatmap

Description

Develop a visual heatmap to represent usage frequency and intensity over time (e.g., days of the week, hours of the day). This feature will help users identify patterns and triggers associated with their usage behavior.

Tasks

- [] Design UI for the heatmap component
- [] Map inhalation data to a time-based grid (days x hours)
- [] Integrate with existing data filtering logic
- [] Add tooltip details on hover (e.g., number of inhalations, duration)

Acceptance Criteria

- [] Heatmap visually reflects usage density with color intensity
- [] Users can hover to view detailed data for specific time blocks
- [] Works responsively across screen sizes

Labels

visualization, heatmap, ui, smoke-log, redesign

Estimate

L

markdown

Title

Implement Rolling Average Analysis Chart

Description

Add a chart that shows a rolling average of usage over user-defined intervals (e.g., 3 days, 7 days, 30 days). This will help users visualize trends over time, smoothing out day-to-day variability.

Tasks

- [] Add a dropdown or input control for selecting the rolling window
- [] Compute rolling average values from timestamped inhalation data
- [] Display the rolling average line on the main usage chart
- [] Ensure it updates dynamically with filters and date range selectors

Acceptance Criteria

- [] Rolling average line appears on the chart when enabled
- [] Users can adjust the rolling window duration
- [] Data updates correctly with filters and range changes

Labels

chart, analytics, smoke-log, redesign

Estimate

M

markdown

Title

Enable Time-of-Day Usage Heatmap

Description

Implement a heatmap visualization that shows usage intensity throughout the day across the selected date range. This will help users identify their peak usage hours.

Tasks

- [] Parse timestamps to extract hour-of-day values

- [] Aggregate inhalation lengths by hour
- [] Generate a 24-hour heatmap representing usage intensity
- [] Integrate the heatmap with existing filters and date range selections

Acceptance Criteria

- [] Users can view a color-coded chart showing inhalation volume by hour
- [] The heatmap updates dynamically with selected date range and filters
- [] Visual indicators clearly distinguish high vs. low activity periods

Labels

data-visualization, analytics, smoke-log, redesign

Estimate

M

markdown

Title

Integrate Rolling Average Chart for Usage Trends

Description

Create a rolling average line chart to provide a smoothed view of inhalation usage trends over time. This feature will help users observe general behavior patterns without daily spikes and drops obscuring long-term trends.

Tasks

- [] Add toggle to switch between raw and rolling average view
- [] Implement adjustable rolling window (e.g., 3-day, 7-day, 14-day)
- [] Update chart data pipeline to calculate and render the rolling average
- [] Ensure compatibility with existing filters and date range selections

Acceptance Criteria

- [] Users can switch between raw data and rolling average
- [] Rolling average updates correctly when filters or range changes
- [] UI clearly distinguishes between raw and smoothed data

Labels

analytics, data-smoothing, smoke-log, redesign

Estimate

M

markdown

Title

Enable User Data Export to CSV

Description

Provide users with the ability to export their smoking log data (inhalation events, durations, timestamps, etc.) to a downloadable CSV file. This feature supports personal data tracking, analysis, and backup.

Tasks

- [] Add "Export to CSV" button to the Smoke-Log page
- [] Format CSV to include: timestamp, length, session tags, and notes
- [] Ensure filtered data exports correctly when filters are active
- [] Sanitize and escape all values to prevent CSV injection
- [] Add loading indicator during export process

Acceptance Criteria

- [] Users can successfully download a CSV file of their data
- [] Exported CSV respects active filters and time ranges
- [] File opens correctly in common spreadsheet tools (Excel, Google Sheets)

Labels

export, data-accessibility, smoke-log, redesign

Estimate

S

markdown

Title

Implement Smoke Usage Summary Card

Description

Design and integrate a summary card component at the top of the Smoke-Log screen. This card should display key metrics such as total sessions, total duration, and average length over the selected time range.

Tasks

- [] Design a compact summary card UI
- [] Calculate and display:
 - Total number of smoking sessions
 - Total duration (sum of all session lengths)
 - Average session length
- [] Update values dynamically based on the selected date range
- [] Add simple icons or visuals for improved readability

Acceptance Criteria

- [] Summary card appears at the top of the Smoke-Log screen
- [] Data updates automatically with date range changes
- [] Styling matches the overall design theme

Labels

ui-enhancement, smoke-log, summary, redesign

Estimate

M

markdown

Title

Add Visual Trend Indicators to Daily Chart

Description

Enhance the daily smoking session chart by adding visual trend indicators. This will help users quickly interpret changes in usage, encouraging self-awareness and progress tracking.

Tasks

- [] Identify and define visual cues for increasing vs. decreasing trends (e.g., arrow icons, color gradients)
- [] Integrate trend indicators on each data point of the chart or as an overlay
- [] Ensure indicators update dynamically based on the filtered date range
- [] Add legend or tooltip for user clarity

Acceptance Criteria

- [] Trend indicators are present and correctly represent usage changes
- [] Indicators update in real-time with data filter changes
- [] UX remains clean and non-cluttered

Labels

chart-enhancement, smoke-log, visual-feedback, redesign

Estimate

M

markdown

Title

Implement Daily Summary Statistics Below Chart

Description

Display a brief summary below the daily session chart to provide users with quick insights into their usage. This can

include total sessions, average session length, and longest/shortest session for the selected day or range.

Tasks

- [] Design layout for the summary statistics section
- [] Calculate relevant statistics based on filtered data
- [] Display summary dynamically beneath the chart
- [] Ensure mobile responsiveness and accessibility

Acceptance Criteria

- [] Summary shows accurate data for current selection
- [] Layout is clean and easy to interpret
- [] Updates in real-time when the chart data is filtered

Labels

ui-enhancement, statistics, smoke-log, redesign

Estimate

S

markdown

Title

Add Weekly Trend Comparison to Dashboard

Description

Enhance user insights by adding a comparative trend feature that shows how the current week's smoking sessions and duration compare to the previous week. This can be a small chart or percentage indicators above the main graph.

Tasks

- [] Calculate total sessions and total duration for the current and previous weeks
- [] Design and implement a visual component to display the comparison
- [] Ensure data is updated when date range is adjusted
- [] Include positive/negative trend indicators (e.g., arrows or colors)

Acceptance Criteria

- [] Users can clearly see if their weekly usage has increased or decreased
- [] Component integrates seamlessly with the current dashboard layout
- [] Responsive and accessible on mobile and desktop

Labels

ui-enhancement, trends, smoke-log, redesign

Estimate

M

markdown

Title

Implement Daily Averages in Summary Panel

Description

Add daily average metrics for sessions and duration to the summary panel to give users a clearer sense of their habitual smoking patterns over the selected date range.

Tasks

- [] Calculate average sessions per day
- [] Calculate average inhalation duration per day
- [] Add this data to the summary card UI below total values
- [] Ensure formatting is concise and readable

Acceptance Criteria

- [] Daily averages are shown and update according to the selected date range
- [] UI adapts well to different screen sizes
- [] Numbers are rounded appropriately for clarity

Labels

enhancement, ui, smoke-log, redesign

Estimate

S

markdown

Title

Implement Cumulative Chart Toggle for Smoke Timeline

Description

Add a toggle option to the smoke timeline chart to switch between standard daily logs and a cumulative usage view. This will help users identify trends in total usage over time more easily.

Tasks

- [] Add a toggle button above the smoke timeline chart
- [] Implement logic to compute and render cumulative usage data
- [] Ensure chart labels and axes update appropriately when toggled
- [] Maintain current behavior as the default view

Acceptance Criteria

- [] Toggle button appears and functions correctly
- [] Switching to cumulative view displays running totals of inhalation duration
- [] Returning to standard view resets to daily session view
- [] Performance is not significantly impacted

Labels

feature, data-visualization, smoke-log, redesign

Estimate

M

markdown

Title

Add Rolling Average View to Smoke Timeline Chart

Description

Introduce an option for users to view a rolling average (e.g., 7-day) of their smoking duration on the timeline chart. This will help visualize long-term trends and smooth out daily fluctuations.

Tasks

- [] Add dropdown or toggle for selecting average view (None, 7-day, 30-day)
- [] Compute rolling average based on selected interval
- [] Update chart to overlay or switch to the averaged data
- [] Ensure clarity in chart labeling to distinguish average line

Acceptance Criteria

- [] Rolling average can be toggled on/off with user-selected interval
- [] Chart correctly displays smoothed trend line over daily data
- [] Functionality works seamlessly across date range selections
- [] Visual elements are intuitive and non-intrusive

Labels

feature, data-visualization, smoke-log, redesign

Estimate

M

markdown

Title

Implement Day/Night Mode for Smoke Log Interface

Description

Introduce a user-selectable day/night (light/dark) theme for the Smoke Log interface to enhance usability in different lighting environments. Ensure the theme affects all visual components, including charts, buttons, and background elements.

Tasks

- [] Add toggle button or setting in user preferences for theme selection
- [] Define and apply dark theme colors for all components
- [] Ensure readability and accessibility of text and charts in both themes
- [] Persist theme selection between sessions (e.g., via local storage)

Acceptance Criteria

- [] User can toggle between light and dark themes
- [] All UI elements respond correctly to the selected theme
- [] The theme persists when the user returns to the app
- [] The implementation meets accessibility standards

Labels

feature, ui/ux, redesign, accessibility

Estimate

M

markdown

Title

Add Weekly Summary Chart to Smoke Log Dashboard

Description

Implement a new chart on the Smoke Log dashboard that visualizes cumulative usage per day across a selected week. This will help users identify trends and compare daily inhalation habits in a more digestible format.

Tasks

- [] Add a selectable weekly range filter in the dashboard
- [] Generate a cumulative sum of inhalation length per day for the selected week
- [] Render the weekly data in a bar chart or area chart format
- [] Update legends and labels to reflect date ranges and totals
- [] Ensure the chart integrates with existing filters (e.g., time of day)

Acceptance Criteria

- [] Users can view a summary of each day's inhalation total over a selected week
- [] Chart updates dynamically when different weeks are selected
- [] Visual styling aligns with existing dashboard components

Labels

feature, data-visualization, redesign

Estimate

M

markdown

Title

Implement Rolling 7-Day Average Chart for Inhalation Trends

Description

Add a line chart to display the rolling 7-day average of total daily inhalation lengths. This will help users visualize broader behavioral trends and smooth out daily fluctuations.

Tasks

- [] Create a backend function to calculate a rolling average over the past 7 days
- [] Integrate this data into the existing charting framework
- [] Display the average line on top of the existing daily totals chart, or in a new chart module

- [] Add toggle to show/hide rolling average
 - [] Label axes clearly and add tooltips for daily and average values
- ### Acceptance Criteria**
- [] Users can view a line representing the 7-day rolling average of inhalation length
 - [] Line updates based on selected date ranges
 - [] The feature is responsive and accessible on both mobile and desktop

Labels

enhancement, data-visualization, redesign

Estimate

M

markdown

Title

Add Editable Log Entry Feature for Smoke Sessions

Description

Implement functionality allowing users to edit previously submitted inhalation logs. This enhances accuracy for users who may need to correct mistaken entries or provide more context to a session.

Tasks

- [] Add an "Edit" button next to each log entry in the history list
- [] Pre-fill the submission form with existing data when editing
- [] Allow changes to inhalation length, timestamp, and any optional notes
- [] Update the entry in Firestore and ensure changes reflect immediately in all charts
- [] Display a subtle "edited" indicator on modified entries

Acceptance Criteria

- [] Users can select and modify any previous entry
- [] Updates are validated and saved to Firestore
- [] Visual indicators make it clear which entries were edited
- [] Editing does not break chart or table visualizations

Labels

feature, UX, firestore, redesign

Estimate

L

markdown

Title

Implement Time Range Selector for Chart Analysis

Description

Add a time range selector to allow users to filter inhalation data visualizations by specific periods (e.g., last 7 days, last 30 days, custom range). This feature helps users focus on trends and patterns over time.

Tasks

- [] Create dropdown or tab interface for predefined time ranges
- [] Add custom date range picker functionality
- [] Filter data shown in all charts based on selected range
- [] Ensure the table view syncs with the selected range
- [] Update Firestore queries or local filtering logic accordingly

Acceptance Criteria

- [] Users can select predefined and custom time ranges
- [] All visualizations update to reflect the selected range
- [] UX remains smooth with minimal latency

Labels

feature, charts, UX, redesign

Estimate

M

markdown

Title

Add Rolling Averages to Inhalation Charts

Description

Enhance data visualization by implementing rolling average lines on inhalation charts to smooth out short-term fluctuations and highlight longer-term trends. This feature will provide better insight into user behavior over time.

Tasks

- [] Add toggle to display rolling average on each chart
- [] Implement logic to calculate rolling averages (e.g., 3-day, 7-day)
- [] Ensure average lines update dynamically with time range selection
- [] Style lines to visually differentiate from raw data

Acceptance Criteria

- [] Users can toggle rolling average visibility on/off
- [] Averages display correctly based on selected time range
- [] Charts remain performant with average overlays enabled

Labels

enhancement, charts, UX, redesign

Estimate

M

markdown

Title

Implement Monthly Summary Modal

Description

Create a modal that provides users with a monthly summary of their smoking habits, including total puffs, average daily usage, most frequent time of day, and any streaks (e.g., days smoke-free). This feature will encourage reflection and insight into behavioral patterns.

Tasks

- [] Design a modal component triggered from the dashboard
- [] Aggregate monthly usage data and compute summary metrics
- [] Display summary with supportive visuals (e.g., icons, small charts)
- [] Include a button to export summary as image or PDF

Acceptance Criteria

- [] Users can view a concise summary for each past month
- [] All data visualizations match the selected month
- [] Summary can be exported easily

Labels

feature, UI, data-visualization, redesign

Estimate

M

markdown

Title

Enable Custom Time Range Filtering for Charts

Description

Implement a feature that allows users to select a custom date range to filter the data displayed in charts. This will provide more flexibility and control for analyzing specific time periods.

Tasks

- [] Add a date range picker component above the charts
- [] Ensure all charts update dynamically based on the selected range
- [] Validate date inputs to prevent errors or null states
- [] Store the selected range in state and preserve it across navigation

Acceptance Criteria

- [] Users can select any start and end date for chart data
- [] All visible charts reflect the selected time range accurately
- [] UI updates responsively without needing a full page reload

Labels

feature, UI, charts, filters, redesign

Estimate

M

markdown

Title

Add "Time of Day" Distribution Chart for Smoking Sessions

Description

Introduce a new visualization that displays the frequency of smoking sessions across different hours of the day. This chart will help users identify patterns in their usage based on time.

Tasks

- [] Extract and aggregate session timestamps by hour
- [] Build a bar chart or line graph to display session count per hour
- [] Integrate the chart into the main dashboard or analytics tab

- [] Ensure responsiveness and visual consistency with other charts

Acceptance Criteria

- [] Users can see a clear visual distribution of usage by hour
- [] Chart updates based on current filters or date ranges
- [] Proper labeling and tooltips for each hourly segment

Labels

feature, chart, analytics, time-distribution, redesign

Estimate

M

markdown

Title

Implement Rolling Average Usage Chart for Smoking Sessions

Description

Create a chart that displays a rolling average of session lengths or frequency over a selected time window (e.g., 7 days). This will help users identify longer-term trends and smooth out daily fluctuations.

Tasks

- [] Add data transformation to calculate rolling averages (e.g., 7-day)
- [] Build a line chart to display the rolling average
- [] Allow users to toggle between different rolling windows (e.g., 3-day, 7-day, 30-day)
- [] Ensure the chart updates with current filters and date range

Acceptance Criteria

- [] Rolling average chart displays correct values based on session data
- [] Users can select different rolling window durations

- [] The chart integrates seamlessly with existing dashboard layout

Labels

feature, chart, analytics, rolling-average, redesign

Estimate

M

markdown

Title

Implement Filter for Day of Week Patterns in Smoke Logs

Description

Add a feature that allows users to filter and analyze their smoking behavior based on the day of the week. This can highlight trends such as increased usage on weekends or reduced use on weekdays.

Tasks

- [] Add backend support to group and aggregate sessions by day of week
- [] Update frontend to include a day-of-week filter/toggle
- [] Display a bar or line chart comparing usage per weekday
- [] Ensure chart interacts correctly with other filters and date ranges

Acceptance Criteria

- [] Users can view session patterns grouped by weekday
- [] Chart displays accurate totals or averages per day
- [] UI is responsive and integrates with existing dashboard elements

Labels

feature, analytics, filters, redesign, day-of-week

Estimate

M

markdown

Title

Implement Rolling Usage Chart for Daily Averages

Description

Enhance the analytics dashboard by adding a rolling average chart that visualizes daily smoking activity trends over time. This will help users identify gradual behavior changes and maintain motivation through visual feedback.

Tasks

- [] Add backend logic to compute rolling daily average (e.g., 7-day or 14-day)
- [] Integrate chart component on dashboard to display the rolling usage
- [] Add user toggle to switch between raw data and rolling average
- [] Ensure compatibility with date range selector and other filters

Acceptance Criteria

- [] Chart accurately reflects rolling average usage data
- [] Users can toggle between different rolling periods
- [] UI remains responsive and intuitive with added functionality

Labels

feature, analytics, charts, redesign, rolling-average

Estimate

M

markdown

Title

Design and Implement User Settings Panel

Description

Add a dedicated user settings panel to AshTrail that allows users to customize key aspects of the app, including display preferences, default date ranges, and notification preferences.

Tasks

- [] Create UI layout for settings panel
- [] Add toggle for dark/light mode
- [] Allow users to set a default time range (e.g., past 7 days, 30 days)
- [] Add settings for notification preferences (if applicable)
- [] Store settings locally and persist across sessions

Acceptance Criteria

- [] Users can access the settings panel from the navigation
- [] Changes are saved and persist across page reloads
- [] UI reflects selected settings immediately where applicable

Labels

feature, UI/UX, settings, redesign

Estimate

M

markdown

Title

Add Export Functionality for Log Data

Description

Implement an export feature that allows users to download their smoke log data in CSV or JSON format for personal

records or analysis outside of the app.

Tasks

- [] Create export button in the UI
- [] Support CSV and JSON file formats
- [] Filter exported data based on current date range and chart filters
- [] Ensure proper formatting and encoding for downloaded files

Acceptance Criteria

- [] Export button is visible and intuitive
- [] Users can choose between CSV and JSON formats
- [] Downloaded files accurately reflect filtered log data
- [] Files are correctly formatted and ready for use

Labels

feature, export, data, redesign

Estimate

M

markdown

Title

Integrate Chart Legend and Hover Details

Description

Improve data readability on AshTrail's charts by adding a legend to identify each data series and hoverable tooltips to display exact values and timestamps.

Tasks

- [] Add a legend below or beside each chart to indicate what each line or bar represents
- [] Implement hover functionality to show specific inhalation length and timestamp data
- [] Ensure tooltips adjust to the selected date range and filters

- [] Style tooltips and legend to match app theme and improve accessibility

Acceptance Criteria

- [] Legend clearly indicates each chart line or bar meaning
- [] Hovering over chart points displays relevant data
- [] Data shown reflects applied filters and selected range
- [] UI remains clean and user-friendly

Labels

enhancement, UI, data-visualization, redesign

Estimate

S

markdown

Title

Implement Rolling Average Chart Toggle

Description

Introduce a toggle feature that allows users to view a rolling average of their inhalation lengths over time. This helps users better understand trends and smooth out day-to-day variability in their smoke-logging data.

Tasks

- [] Add a toggle control to enable/disable rolling average view
- [] Calculate and display a 7-day (or user-defined) rolling average for inhalation length
- [] Overlay the rolling average line on existing charts
- [] Ensure data filters (range, tags) affect both raw and average views

Acceptance Criteria

- [] Users can toggle between raw and rolling average data views
- [] Rolling average is calculated correctly and updates with

filters

- [] UI clearly distinguishes between raw data and rolling average lines
- [] No performance issues for year-long datasets

Labels

enhancement, data-analysis, redesign, UI

Estimate

M

markdown

Title

Add User Feedback Component for Logging Confirmation

Description

Create a user feedback mechanism that provides confirmation or visual cues after a smoke log is submitted. This enhances user experience by confirming successful data entry and encourages consistent logging.

Tasks

- [] Implement a Snackbar/Toast or modal confirmation after a successful log
- [] Ensure feedback includes timestamp and duration of inhalation
- [] Optionally provide a quick "Undo" button for accidental logs
- [] Ensure component is responsive and does not block other UI elements

Acceptance Criteria

- [] Users receive immediate visual confirmation after logging
- [] Confirmation includes relevant data (e.g., time and length)
- [] Undo functionality works correctly and removes the log from Firestore

- [] No duplicate entries or race conditions occur

Labels

UX, enhancement, frontend, redesign

Estimate

S

markdown

Title

Implement Loading Indicator for Log Submission

Description

Introduce a loading spinner or progress indicator when a user submits a new smoke log. This helps users understand that their action is being processed and prevents repeated submissions due to perceived unresponsiveness.

Tasks

- [] Add a loading indicator to the submit button during the Firestore write operation
- [] Disable the submit button while loading
- [] Ensure indicator is styled consistently with the app's design
- [] Re-enable button and hide indicator once submission completes

Acceptance Criteria

- [] Loading state appears immediately upon log submission
- [] Users cannot resubmit while loading
- [] Indicator disappears after success or error
- [] No user action is lost or duplicated during submission

Labels

UI, enhancement, frontend, redesign

Estimate

XS

markdown

Title

Refactor Chart Range Selector into Unified Filter Component

Description

Move the chart range selector logic into a shared, centralized filter component that also handles filtering and sorting for logs. This redesign will make it easier to integrate additional analysis features like rolling averages or cumulative usage.

Tasks

- [] Create a unified filter widget that includes range selection, filter toggles, and sort options
- [] Integrate the widget into the chart dashboard page
- [] Replace all existing range selector and filter code with references to the new component
- [] Ensure backward compatibility with existing chart interactions

Acceptance Criteria

- [] The new widget can modify time range, filters, and sort order from a single location
- [] Charts and tables react appropriately to filter changes
- [] Design and behavior match the app's overall look and feel
- [] No functionality regression from previous version

Labels

refactor, UI, feature, redesign

Estimate

M

markdown

Title

Implement Rolling Average Chart Mode

Description

Add a chart display mode that allows users to toggle a rolling average view of inhalation length data over time. This feature helps smooth out daily fluctuations and reveals longer-term trends.

Tasks

- [] Add toggle for enabling/disabling rolling average mode in the unified filter component
- [] Implement rolling average calculation in chart data preparation
- [] Display tooltip information showing both raw and average values
- [] Adjust chart legend to indicate when rolling average is active
- [] Ensure compatibility with range and filter options

Acceptance Criteria

- [] Users can toggle rolling average mode on and off
- [] Chart updates dynamically and correctly reflects average trends
- [] Tooltip and legend provide clear context
- [] Feature works with all existing filters and date ranges

Labels

feature, data-visualization, enhancement

Estimate

L

markdown

Title

Refactor Chart Rendering to Support Dynamic Height and Legend

Placement

Description

Improve the flexibility of chart components by supporting dynamic height adjustment based on available viewport space and repositioning the legend to optimize readability, especially in landscape and mobile views.

Tasks

- [] Refactor chart container to calculate and apply dynamic height
- [] Add logic to reposition legend (below or beside chart) based on screen width
- [] Test responsiveness across devices and orientations
- [] Ensure no overlap between legend and chart elements
- [] Update relevant styles and themes for consistency

Acceptance Criteria

- [] Chart resizes dynamically with screen size
- [] Legend adjusts location based on available space
- [] No visual clipping or overlapping on supported screen sizes
- [] Maintains aesthetic consistency with overall app design

Labels

refactor, responsive-design, enhancement

Estimate

M

markdown

Title

Implement Toggle Between Daily and Cumulative Usage Views

Description

Add a feature that allows users to toggle between daily individual log entries and a cumulative view of usage over time. This toggle should be easily accessible and reflect

immediately in the chart data displayed.

Tasks

- [] Add a toggle button to the chart control panel
- [] Implement logic to switch between daily and cumulative views
 - [] Ensure accurate aggregation of data for cumulative view
 - [] Update chart labels and tooltips based on selected mode
 - [] Test performance with long time ranges

Acceptance Criteria

- [] Users can switch between daily and cumulative usage views
 - [] Charts update seamlessly without reload
 - [] Data in both views is accurate and visually clear
 - [] Toggle state persists during session

Labels

feature, data-visualization, UX

Estimate

M

markdown

Title

Add Range Selector Component for Time-Based Filtering

Description

Implement a range selector UI component that allows users to filter log data by common timeframes (e.g., Last 7 Days, Last 30 Days, All Time, Custom Range). The range selected should dynamically update the chart and data table views.

Tasks

- [] Design and implement the range selector UI
- [] Integrate range selection logic with chart and data filtering
 - [] Support predefined and custom date ranges

- [] Ensure responsiveness and accessibility of the component
- [] Test for accuracy and performance with large datasets

Acceptance Criteria

- [] Range selector displays and functions correctly
- [] Chart and data table update based on selected range
- [] Custom range allows calendar input and validates dates
- [] No performance issues with long date ranges

Labels

feature, date-filter, UX

Estimate

M

markdown

Title

Implement Cumulative Usage Chart with Daily Breakdown

Description

Add a cumulative usage chart to visualize total smoking time over the selected date range. This should include a breakdown of usage per day and update dynamically based on the selected time filter.

Tasks

- [] Design and implement a line or bar chart to show daily and cumulative usage
- [] Connect the chart to filtered data based on range selector
- [] Calculate and display total usage in hours/minutes
- [] Highlight days with unusually high or low usage
- [] Add tooltips for individual data points

Acceptance Criteria

- [] Chart accurately reflects cumulative and daily usage
- [] Updates immediately based on range selector changes
- [] Displays total usage in selected time range

- [] Responsive and readable on mobile and desktop

Labels

feature, visualization, data-analytics

Estimate

M

markdown

Title

Add Monthly Trend Analysis with Rolling Average

Description

Introduce a monthly trend chart to show rolling average usage over time. This will help users identify long-term patterns in their smoking behavior.

Tasks

- [] Implement a chart displaying daily usage with a 7-day rolling average overlay
- [] Aggregate usage by day and calculate rolling average
- [] Make chart responsive and clear on all screen sizes
- [] Add toggles for viewing raw daily totals vs. smoothed trends
- [] Integrate with existing date range selector

Acceptance Criteria

- [] Chart shows accurate 7-day rolling averages
- [] Clear differentiation between raw and averaged data
- [] Users can toggle between views
- [] Works across devices and screen sizes

Labels

feature, enhancement, analytics

Estimate

M

markdown

Title

Implement Interactive Data Table for Log Entries

Description

Create a sortable, filterable, and paginated data table to display log entries. This allows users to explore their data more deeply and supports future enhancements like exporting or bulk actions.

Tasks

- [] Set up a data table component with headers: Timestamp, Inhalation Length, Notes
- [] Enable sorting by timestamp (default: descending)
- [] Add filtering options for date range and keyword in notes
- [] Implement pagination or infinite scroll for large datasets
- [] Ensure accessibility and responsive design

Acceptance Criteria

- [] Users can view their log entries in a structured table
- [] Table supports sorting and filtering
- [] Performance remains smooth with large datasets
- [] Fully responsive and accessible

Labels

feature, ui, enhancement

Estimate

M

markdown

Title

Design and Implement Rolling Usage Chart with Range Selector

Description

Add a new chart to visualize rolling usage over time, allowing users to select a date range and see cumulative inhalation patterns within that range. This will provide deeper insights into usage trends and habits.

Tasks

- [] Add a rolling usage chart that updates based on selected date range
- [] Use a moving average or sum over a window (e.g., 3-day, 7-day)
- [] Integrate with the existing range selector for consistency
- [] Display totals or averages within the selected range
- [] Ensure chart updates smoothly with user interaction

Acceptance Criteria

- [] Users can view a rolling usage chart with selectable range
- [] Chart clearly indicates cumulative usage trends
- [] Performance is optimized for responsiveness
- [] Compatible with all screen sizes

Labels

feature, chart, data-visualization

Estimate

M

markdown

Title

Optimize Data Query Performance for Log Filtering and Sorting

Description

Enhance the performance and responsiveness of the log filtering and sorting system in the AshTrail app. As the volume of stored data increases, ensuring quick access and smooth UI interactions is critical.

Tasks

- [] Evaluate current Firestore queries and identify performance bottlenecks
- [] Implement indexing strategies to support common filters and sort orders
- [] Optimize data fetch size and pagination methods
- [] Ensure logic for development and production modes (test vs prod collections) remains unaffected
- [] Add performance logging to monitor query execution time

Acceptance Criteria

- [] Filter and sort operations complete within 500ms for standard datasets
- [] No breaking changes to data logic or chart display
- [] Works seamlessly across mobile and desktop views

Labels

performance, backend, firestore, optimization

Estimate

M

markdown

Title

Add Chart Summary Cards Above Log Visualizations

Description

Introduce summary cards above the main chart area to provide quick insights into the data being displayed. These cards should reflect filtered data ranges and update dynamically with changes to date ranges or other filters.

Tasks

- [] Design and implement card layout using responsive styling
- [] Display key metrics such as total puffs, average session length, and cumulative time

- [] Link summary values to current chart range and filtered data set
- [] Ensure cards update in real-time when filters or range selectors change
- [] Add placeholder icons or visuals to improve visual hierarchy

Acceptance Criteria

- [] Summary cards appear above the charts and are mobile-friendly
- [] Cards update when the data range or filters are modified
- [] Values match what's displayed in the logs and charts below
- [] No impact on chart performance or interactivity

Labels

frontend, enhancement, UI/UX

Estimate

M

markdown

Title

Implement Sticky Date Range & Filter Controls for Enhanced Usability

Description

Enhance the user experience by making the date range selector and data filters sticky on scroll. This ensures users can adjust filters or range without having to scroll back up, improving navigation and interactivity when analyzing logs and charts.

Tasks

- [] Make the date range and filter controls sticky on scroll
- [] Maintain responsiveness on mobile and tablet devices
- [] Ensure sticky elements do not overlap with chart tooltips or other UI elements

- [] Add visual indicator or shadow to distinguish sticky section from rest of page

Acceptance Criteria

- [] Filters and range selector remain visible during scrolling
- [] Sticky behavior does not interfere with existing chart or UI elements
- [] Works seamlessly across different screen sizes and browsers

Labels

frontend, UX, enhancement

Estimate

S

markdown

Title

Create Modular Component for SmokeLog Entry Display

Description

Design and implement a reusable modular component to display individual SmokeLog entries. This component will standardize the layout, improve readability, and simplify future feature additions like tooltips, tagging, or entry-level actions.

Tasks

- [] Design a UI layout for individual SmokeLog entries
- [] Build the modular component with props for all relevant data (timestamp, duration, notes, etc.)
- [] Integrate the component into the main log listing
- [] Add basic styling and hover interactions
- [] Ensure accessibility and mobile responsiveness

Acceptance Criteria

- [] Component displays all necessary information clearly
- [] Compatible with both light and dark themes

- [] Fully reusable across the app with consistent behavior
- [] Performance optimized for large lists

Labels

component, frontend, refactor

Estimate

M

markdown

Title

Implement Chart Range Selector Integration

Description

Add a range selector UI component to the Smoke-Log charts to allow users to view inhalation data over specific time frames (e.g., past day, week, month). This should work seamlessly with existing chart filters and support dynamic updates.

Tasks

- [] Design and implement a range selector (e.g., dropdown or tabs)
- [] Connect selector to backend or data layer to filter displayed data
- [] Update chart rendering logic to reflect range changes
- [] Maintain compatibility with existing date filters and sorting
- [] Ensure the UI is responsive and intuitive

Acceptance Criteria

- [] Users can switch between predefined date ranges
- [] Charts update immediately to reflect selected range
- [] The system handles empty or sparse datasets gracefully
- [] Works on desktop and mobile

Labels

enhancement, frontend, chart

Estimate

M

markdown

Title

Refactor Data Model for Scalable Entry Logging

Description

Update the internal data model to better support scalable logging of inhalation entries. This includes ensuring each entry is normalized and stored in a way that supports future analytics, sorting, and filtering operations efficiently.

Tasks

- [] Define and document the schema for each log entry (timestamp, length, optional metadata)
- [] Refactor existing code to use the new model
- [] Update Firestore rules and validation logic accordingly
- [] Test data storage and retrieval with the new format
- [] Migrate existing data if necessary

Acceptance Criteria

- [] All new entries follow the updated model
- [] The app continues to function without data loss or inconsistencies
- [] Backend services and analytics remain compatible or are updated
- [] New schema is documented for future development

Labels

backend, refactor, firestore

Estimate

M

markdown

Title

Implement Range-Based Entry Filtering

Description

Introduce the ability to filter inhalation entries based on a selected time range. This feature will support both UI-based range selection and internal data filtering to improve chart accuracy and user insights.

Tasks

- [] Create UI range selector (e.g., last 7 days, 30 days, custom)
- [] Implement filtering logic in the data processing layer
- [] Update charts and table views to respond to the filtered dataset
- [] Ensure performance remains acceptable with large datasets

Acceptance Criteria

- [] Users can select predefined or custom date ranges
- [] Charts and tables update to reflect the filtered data
- [] Data filtering is accurate and consistent across views
- [] Feature is covered by appropriate tests

Labels

frontend, feature, data-visualization

Estimate

M

markdown

Title

Integrate Rolling Average Option for Usage Chart

Description

Add a toggle to the usage chart that enables users to view a rolling average (e.g., 7-day average) of their smoking

activity. This enhances trend visualization and helps users identify changes in behavior over time.

Tasks

- [] Add toggle button for enabling/disabling rolling average
- [] Implement rolling average calculation in data layer
- [] Update chart to overlay rolling average line when enabled
- [] Style the line distinctly to differentiate it from raw data

Acceptance Criteria

- [] Users can toggle the rolling average display
- [] Rolling average accurately reflects the selected time window
- [] Chart maintains readability and performance
- [] Feature is documented and tested

Labels

frontend, enhancement, data-visualization

Estimate

M

markdown

Title

Implement Cumulative Usage Tracker with Custom Date Range

Description

Add a cumulative usage tracker to the dashboard, allowing users to view total inhalation duration over a custom-selected date range. This will provide deeper insights into long-term trends and habit patterns.

Tasks

- [] Design UI component for cumulative display
- [] Add date range selector linked to cumulative tracker
- [] Compute and display total inhalation duration in

selected range

- [] Integrate with existing Firebase/Isar data structure
- [] Test for accuracy and performance

Acceptance Criteria

- [] User can select any date range and view total usage
- [] Component updates dynamically with new range selections
- [] Data is accurate and efficiently processed
- [] Feature is styled consistently with the app design

Labels

frontend, enhancement, data-analysis

Estimate

M

markdown

Title

Add Rolling Average Analysis to Usage Charts

Description

Implement a rolling average overlay on the inhalation usage charts to help users better visualize trends over time. This feature should allow toggling between different averaging windows (e.g., 3-day, 7-day).

Tasks

- [] Create UI toggle for rolling average options
- [] Calculate rolling averages based on selected window
- [] Overlay rolling average line on existing charts
- [] Ensure performance optimization for large datasets
- [] Add tests for accuracy and UI responsiveness

Acceptance Criteria

- [] Users can select 3-day and 7-day rolling averages
- [] Chart displays accurate average trend line
- [] UI remains responsive with long date ranges
- [] Feature integrates cleanly with current analytics tools

Labels

data-visualization, enhancement, frontend

Estimate

M

markdown

Title

Implement "Insights" Tab with Summary Stats and Trends

Description

Create an "Insights" tab that summarizes key user data trends, such as most common usage times, total inhalations over a selected period, longest streaks, and average session duration. This feature will enhance user engagement and provide valuable self-awareness insights.

Tasks

- [] Design UI layout for Insights tab
- [] Develop logic to compute summary statistics
- [] Implement time-based trends (hour, day, week)
- [] Add streak tracking functionality
- [] Ensure accessibility and responsiveness

Acceptance Criteria

- [] Insights tab is accessible from the main dashboard
- [] All stats and trends update according to selected date range
- [] Streaks and averages are calculated accurately
- [] Data visualizations (if any) are clean and minimal

Labels

feature, analytics, frontend, UX

Estimate

L

markdown

Title

Implement Settings Page for Customization and App Preferences

Description

Add a settings page that allows users to customize their app experience. Features include toggles for dark mode, chart smoothing, data export options, and reminder notifications. This improves user control and retention.

Tasks

- [] Design settings UI layout and navigation access
- [] Implement theme toggle (light/dark)
- [] Add option to toggle chart smoothing
- [] Enable manual and automatic data export
- [] Include notification preferences and scheduling

Acceptance Criteria

- [] Users can access and save preferences from the settings page
- [] Theme changes apply immediately
- [] Data export generates accurate log files
- [] Notifications can be enabled or disabled per user choice

Labels

feature, customization, frontend, user-preferences

Estimate

M

markdown

Title

Add Data Summary Cards to Dashboard

Description

Introduce summary cards at the top of the dashboard to give

users quick insights into their recent usage patterns. This includes total sessions, average session length, total duration over selected range, and longest session.

Tasks

- [] Design and layout four data summary cards
- [] Fetch and calculate relevant data from logs
- [] Dynamically update summaries based on selected date range
- [] Ensure responsive layout for mobile and desktop

Acceptance Criteria

- [] Users can see updated summary statistics based on the current date range
- [] Cards display: Total Sessions, Avg. Session Length, Total Time, Longest Session
- [] Cards refresh in real time with data filter changes

Labels

enhancement, UI, dashboard, analytics

Estimate

M

markdown

Title

Implement Rolling Average Chart for Session Duration

Description

Add a new line chart visualization to the dashboard that displays a rolling average of session durations over a user-selectable window (e.g., 3-day, 7-day, 14-day). This chart will help users better identify usage trends over time.

Tasks

- [] Add dropdown or toggle for selecting rolling average window
- [] Calculate rolling averages on the filtered data set

- [] Display the rolling average line chart alongside existing session data
- [] Ensure chart updates dynamically based on date range and rolling window

Acceptance Criteria

- [] Rolling average is visible and matches the selected date range and window
- [] Users can switch between different rolling windows
- [] Works responsively and does not conflict with other dashboard elements

Labels

enhancement, chart, analytics, time-series

Estimate

M

markdown

Title

Refactor Range Selector Logic for Unified Filtering Control

Description

Consolidate all data filtering and sorting logic into the range selector component. This update simplifies the dashboard's architecture, enabling consistent date-based filtering and paving the way for advanced analytics like rolling averages or day-of-week trends.

Tasks

- [] Move filtering and sorting logic into the range selector widget
- [] Ensure charts and tables consume filtered data from a shared source
- [] Maintain reactivity when changing the selected date range
- [] Add tests to confirm correct behavior across all views

Acceptance Criteria

- [] All data visualizations and tables reflect changes made in the range selector
- [] No duplication of filtering logic across components
- [] All functionalities remain intact post-refactor

Labels

refactor, dashboard, architecture, enhancement

Estimate

M

markdown

Title

Create User Profile View with Summary Statistics

Description

Design and implement a user profile screen that summarizes individual usage habits. This includes total sessions, average session length, peak usage time, and a breakdown of usage over time. This feature provides users with insights into their behavior and promotes mindful tracking.

Tasks

- [] Create new screen or route for user profile view
- [] Display summary metrics (total sessions, average length, etc.)
- [] Integrate charts showing usage trends (e.g., line or bar chart)
- [] Add mock/test user data for development purposes
- [] Ensure responsive layout for mobile and desktop

Acceptance Criteria

- [] Profile view accurately displays summary and chart data
- [] All statistics reflect the current filtered dataset
- [] View is accessible from main navigation
- [] Layout is clean and responsive

Labels

feature, dashboard, analytics, enhancement

Estimate

M

markdown

Title

Implement Rolling Usage Chart for Custom Time Ranges

Description

Add a rolling usage chart to the dashboard to visualize cumulative session length over user-defined time ranges (e.g., last 7 days, last 30 days, custom range). This enhancement will help users better understand their patterns over time and evaluate habit changes.

Tasks

- [] Add range selector (e.g., dropdown or calendar) to dashboard
- [] Calculate cumulative session length per day within range
- [] Render rolling line chart showing daily totals
- [] Sync logic with other filters and views
- [] Test for accurate chart rendering and data handling

Acceptance Criteria

- [] Users can select predefined or custom date ranges
- [] Chart updates dynamically based on selection
- [] Cumulative usage is visually clear and responsive
- [] Logic does not conflict with other views

Labels

feature, chart, enhancement, data-visualization

Estimate

M

markdown

Title

Add Real-Time Sync for Log Submissions

Description

Implement real-time synchronization for log submissions to ensure that new entries appear immediately in the log list and relevant charts without requiring a manual refresh. This feature improves user experience by providing immediate feedback and supporting dynamic interaction with the data.

Tasks

- [] Integrate Firestore real-time listeners for log collection
- [] Update state management to reflect new data automatically
- [] Ensure charts and log tables re-render on update
- [] Handle edge cases such as deletions or data conflicts
- [] Test for smooth, real-time responsiveness

Acceptance Criteria

- [] New logs appear in the list and update charts instantly
- [] No noticeable lag or visual glitches during sync
- [] App handles conflicts and removes stale data cleanly

Labels

enhancement, real-time, firebase, UX

Estimate

L

markdown

Title

Implement Date Range Selector with Presets

Description

Introduce a date range selector that allows users to filter

log data by custom ranges and presets such as "Today", "This Week", "Last 7 Days", and "This Month". This will enhance data analysis capabilities and improve user control over the displayed data.

Tasks

- [] Design and implement UI for the date range selector
- [] Add preset options with clearly defined time spans
- [] Connect selector to data query logic for logs and charts
- [] Maintain state across navigation
- [] Test for mobile and desktop responsiveness

Acceptance Criteria

- [] Users can select both custom and preset date ranges
- [] Logs and charts update accurately based on the selected range
- [] UI is intuitive and responsive across devices

Labels

feature, UI, filters, enhancement

Estimate

M

markdown

Title

Implement Multi-Metric Chart Toggle for Usage Visualization

Description

Add the ability for users to toggle between different metrics in the usage visualization chart (e.g., Total Inhalation, Average Inhalation, Count per Day). This enhancement will allow more granular insights and flexible data analysis.

Tasks

- [] Add toggle buttons or dropdown to select chart metric
- [] Adjust chart logic to display selected metric accurately
- [] Ensure transitions between metrics are smooth

- [] Validate with different time range selections
 - [] Write unit and integration tests
- ### Acceptance Criteria**
- [] Users can switch between at least three metrics in the usage chart
 - [] Chart updates immediately to reflect the selected metric
 - [] No visual glitches during transitions
 - [] Works on both mobile and desktop views

Labels

feature, chart, UI, enhancement

Estimate

M

markdown

Title

Add Historical Comparison Feature to Chart Display

Description

Introduce a feature that allows users to compare the current selected date range with the same previous period (e.g., week-over-week, month-over-month). This helps users track progress or changes in their smoking habits over time.

Tasks

- [] Add toggle or checkbox to enable "Compare to Previous Period"
- [] Calculate and display the comparative data in the chart
- [] Use distinct visual indicators (e.g., dashed line) for previous period
- [] Ensure compatibility with all available metrics
- [] Write tests to verify accurate calculations and rendering

Acceptance Criteria

- [] Users can toggle on/off historical comparison

- [] Chart displays both current and previous data clearly
- [] Comparison works across all date range options
- [] No performance degradation when feature is enabled

Labels

feature, data-analysis, chart, UI

Estimate

L

markdown

Title

Implement Rolling Average Toggle for Chart Analysis

Description

Introduce a toggle to display a rolling average (e.g., 7-day or 30-day) over the current charted data. This helps smooth out daily fluctuations and reveals long-term trends in smoking habits.

Tasks

- [] Add UI control to enable/disable rolling average view
- [] Allow user to select the window size (e.g., 7-day, 30-day)
- [] Modify chart rendering logic to overlay the rolling average line
- [] Ensure performance is optimized for large datasets
- [] Add tests for rolling average calculations

Acceptance Criteria

- [] User can enable and configure a rolling average display
- [] Rolling average line is clearly distinguishable from raw data
- [] Functionality integrates seamlessly with existing range filters
- [] No significant performance impact

Labels

feature, data-analysis, chart, enhancement

Estimate

M

markdown

Title

Implement Day/Night Theme Toggle

Description

Add support for a toggle switch that allows users to switch between a light (day) and dark (night) theme. This enhances usability in different lighting conditions and improves visual comfort during night use.

Tasks

- [] Design light and dark themes with accessible color contrasts
- [] Add a toggle switch to the settings or header UI
- [] Store user preference in local storage
- [] Apply the selected theme across all views and components
- [] Ensure charts and visualizations adapt properly to the selected theme

Acceptance Criteria

- [] Users can switch between themes with a toggle
- [] The theme persists on reload and across sessions
- [] All UI elements and charts visually adapt to the selected theme
- [] No functional or styling regressions occur in either mode

Labels

ui, theme, accessibility, enhancement

Estimate

S

markdown

Title

Add Onboarding Tutorial Modal for First-Time Users

Description

Introduce a step-by-step onboarding modal that guides new users through the core features of the Smoke-Log interface. This will improve user understanding and engagement from the start.

Tasks

- [] Design a modal overlay with step-by-step instructions
- [] Highlight key features (e.g., adding entries, using filters, viewing charts)
- [] Implement tracking to detect first-time users (e.g., local storage flag)
- [] Add a "Skip Tutorial" and "Don't Show Again" option
- [] Ensure accessibility and responsiveness on all devices

Acceptance Criteria

- [] First-time users are presented with the tutorial modal
- [] Users can skip or disable the tutorial
- [] Modal provides clear guidance for all core features
- [] Modal works correctly across screen sizes

Labels

onboarding, enhancement, ui, user-experience

Estimate

M

markdown

Title

Implement Log Entry Deletion with Undo Snackbar

Description

Add functionality for users to delete a log entry with an option to undo the deletion shortly after. This improves user control and error recovery in managing smoke log data.

Tasks

- [] Add delete icon/button to each log entry in the list view
- [] Implement backend delete logic (Firestore/Isar depending on environment)
- [] Display a snackbar after deletion with "Undo" action
- [] Restore the entry if "Undo" is clicked within timeout (e.g., 5 seconds)
- [] Ensure compatibility with both dev and prod environments

Acceptance Criteria

- [] Users can delete a log entry from the UI
- [] A snackbar appears with an "Undo" button
- [] Clicking "Undo" restores the deleted entry
- [] Deletion persists if not undone within timeout

Labels

feature, enhancement, user-experience, frontend, backend

Estimate

M

markdown

Title

Add Data Export Feature (CSV)

Description

Allow users to export their smoke log data to a CSV file for backup, analysis, or sharing with healthcare providers. This will improve data portability and transparency for users tracking their smoking habits.

Tasks

- [] Add "Export Data" button in the settings or log view

- [] Generate CSV from stored log data (Firestore/Isar)
- [] Allow download of the CSV file (mobile/web compatible)
- [] Include timestamp, inhalation length, and any metadata
- [] Ensure proper formatting and escaping for CSV

Acceptance Criteria

- [] Users can trigger CSV export
- [] File includes all relevant data points
- [] Download works across platforms (Flutter desktop/mobile/web)
- [] Export respects current log filtering (if any)

Labels

feature, enhancement, export, data-portability

Estimate

M

markdown

Title

Implement Visual Data Summaries on Dashboard

Description

Add graphical summaries to the main dashboard to give users immediate insight into their smoking habits. This feature will provide at-a-glance statistics such as average daily use, longest/shortest sessions, and recent trends using charts or infographics.

Tasks

- [] Design layout for visual summaries
- [] Implement charts using Flutter packages (e.g., `f1_chart`)
- [] *Display average daily inhalation duration*
- [] *Highlight highest and lowest usage days*
- [] *Show weekly or monthly trend graphs*
- [] *Ensure accessibility and responsiveness*

Acceptance Criteria

- [] Dashboard displays updated charts with user data
- [] Graphs are readable and clearly labeled
- [] Data summaries update in real-time with log entries
- [] Works across screen sizes and platforms

Labels

feature, visualization, enhancement, dashboard

Estimate

L

markdown

Title

Enable Date Range Filtering for Analytics and Logs

Description

Implement functionality to filter inhalation logs and analytics based on a user-selected date range. This feature allows users to examine patterns over specific periods and focus on recent or historical usage.

Tasks

- [] Add UI component for date range selection
- [] Filter chart and table data based on the selected range
- [] Default to last 7 days if no range is chosen
- [] Ensure consistent behavior across visualizations and data tables
- [] Add tests to verify filtering accuracy

Acceptance Criteria

- [] Users can select a custom date range
- [] All charts and tables reflect the selected date range
- [] Default behavior shows the last 7 days of data
- [] Feature is responsive and intuitive across devices

Labels

feature, filter, enhancement, UI/UX

Estimate

M

markdown

Title

Implement Rolling Average Display for Inhalation Length

Description

Add functionality to display a rolling average of inhalation length over time. This will help users visualize trends and smooth out short-term fluctuations in their smoking behavior.

Tasks

- [] Add a toggle or dropdown to enable rolling average on the main chart
- [] Compute rolling averages (e.g., 3-day, 7-day, and 14-day options)
- [] Update chart to display the rolling line alongside the raw data
- [] Ensure the calculation updates dynamically with date range filters
- [] Add tests for rolling average calculations and display logic

Acceptance Criteria

- [] Rolling average lines appear correctly based on user selection
- [] Raw and averaged data are distinguishable (e.g., line style or color)
- [] Works with date filters and does not break existing chart features
- [] Users can switch between different average intervals

Labels

analytics, enhancement, visualization

Estimate

M

markdown

Title

Create Alert System for Unusual Usage Patterns

Description

Implement an alert mechanism that flags and optionally notifies the user when unusual smoking behavior is detected based on sudden spikes in usage, frequency, or inhalation length.

Tasks

- [] Define what constitutes an "unusual" pattern using statistical or threshold-based analysis
- [] Integrate alert logic into the existing data pipeline
- [] Display alerts visually in the UI (e.g., banner, icon, or chart marker)
- [] Allow users to view past alerts and dismiss or acknowledge them
- [] Provide a basic notification setting (on/off) for alert messages

Acceptance Criteria

- [] Alerts trigger only when usage deviates significantly from the norm
- [] Alerts appear consistently across relevant time ranges and views
- [] Users can manage alert settings and history
- [] No performance degradation with alert detection enabled

Labels

analytics, feature, notification, UX

Estimate

M

markdown

Title

Implement Rolling Average Trend Line in Usage Charts

Description

Add a feature to overlay a rolling average (e.g., 7-day, 30-day) trend line to the primary usage charts. This helps visualize long-term patterns and smooth out daily fluctuations.

Tasks

- [] Add settings for selecting rolling average duration (7-day, 30-day)
- [] Modify chart logic to calculate and display rolling averages
- [] Ensure the rolling line updates dynamically with date range selection
- [] Use a visually distinct style (e.g., dashed line or different color) to differentiate from actual data
- [] Provide tooltip support to show rolling average value on hover

Acceptance Criteria

- [] Users can toggle and switch between rolling average durations
- [] Rolling average line appears accurately and does not affect chart performance
- [] UI clearly distinguishes between real data and trend line

Labels

analytics, enhancement, data-visualization

Estimate

M

markdown

Title

Integrate Data Export Functionality (CSV/PDF)

Description

Allow users to export their inhalation logs and chart data as CSV and PDF files. This feature will support users in analyzing their data externally or sharing it with healthcare professionals.

Tasks

- [] Add "Export" button to the main dashboard view
- [] Implement backend logic for generating CSV from log data
- [] Implement backend logic for generating PDF from selected charts and data summaries
- [] Add options to select date range for export
- [] Ensure exported files include proper formatting and relevant metadata

Acceptance Criteria

- [] Users can successfully export log data as a CSV file
- [] Users can generate a PDF with chart images and basic usage statistics
- [] Exported files include date ranges and summary metadata
- [] Feature works consistently across supported devices

Labels

export, enhancement, user-feature

Estimate

M

markdown

Title

Implement In-App Feedback and Bug Reporting Feature

Description

Introduce a built-in feedback and bug reporting system in the

AshTrail app to streamline user support and gather actionable insights. This system will allow users to report issues or suggest improvements directly from within the app.

Tasks

- [] Add a "Feedback & Report Issue" button in the settings or about page
- [] Create a feedback form with fields for description, category (bug, feature, general), and optional screenshot
- [] Integrate submission with Firestore or external reporting system
- [] Add notification/confirmation UI upon successful submission
- [] Implement backend route to receive and store the feedback securely

Acceptance Criteria

- [] Users can easily access the feedback form in-app
- [] Feedback is submitted and stored correctly with a timestamp and user metadata
- [] Bug reports and suggestions are separated and categorized automatically
- [] Optional screenshot support is functional on mobile and web

Labels

feedback, support, enhancement, user-feature

Estimate

M

markdown

Title

Add Usage Statistics Visualization (Rolling and Cumulative Views)

Description

Enhance the AshTrail app by adding visualizations for both

rolling usage (e.g., daily, weekly) and cumulative usage over time. This feature will provide users with a clearer understanding of their smoking habits and long-term trends.

Tasks

- [] Implement logic to calculate rolling usage (daily, 7-day, 30-day)
- [] Implement logic for cumulative usage chart
- [] Display both charts in the analysis view with toggle or tabs
- [] Sync chart data with selected time range and filters
- [] Ensure responsiveness and accessibility across screen sizes

Acceptance Criteria

- [] Users can view both rolling and cumulative usage charts
- [] Charts update based on selected date range and filters
- [] Visuals are clear, accurate, and optimized for mobile

Labels

visualization, feature, enhancement, analytics

Estimate

L

markdown

Title

Refactor Log Entry Submission for Modular Use and Testing

Description

Refactor the current log submission function (`submitData`) in the AshTrail app to separate concerns and improve reusability, especially for automated or alternative input methods (e.g., syncing with external trackers or voice input). This will also support easier unit testing.

Tasks

- [] Extract data validation and formatting logic into a

standalone utility

- [] Abstract Firestore submission into a service class or function
- [] Allow external triggers to invoke log submissions with minimal overhead
- [] Write unit tests for submission logic (mocking Firestore as needed)
- [] Confirm UI and chart updates still function after refactor

Acceptance Criteria

- [] Core submission logic is reusable and testable
- [] App functionality remains unchanged for users
- [] Unit tests cover validation and Firestore submission pathways

Labels

refactor, enhancement, testability, modularization

Estimate

M

markdown

Title

Implement Rolling Average Chart for Smoke Usage Trends

Description

Add a new rolling average chart to the AshTrail app to visualize long-term trends in smoke usage. This chart will help users understand their behavior over time and highlight improvement patterns or relapses.

Tasks

- [] Add a dropdown option in the chart filter for selecting "Rolling Average"
- [] Implement logic for calculating rolling average over configurable periods (e.g., 7-day, 14-day)
- [] Update chart rendering logic to support smoothed data

```
lines
- [ ] Include tooltips and labels to clarify data presentation
- [ ] Validate performance with large datasets

### Acceptance Criteria
- [ ] Chart displays accurate rolling averages over selected periods
- [ ] Users can switch between standard and rolling average views
- [ ] No regression in existing chart functionality

### Labels
feature, charting, analytics, UX

### Estimate
M
```

markdown

```
### Title
Add Session Summary Panel to Entry Detail View

### Description
Enhance the Entry Detail View by adding a Session Summary panel. This panel will provide users with a quick overview of their inhalation session, including total length, number of inhales, and average inhale duration.

### Tasks
- [ ] Design and implement UI for Session Summary panel
- [ ] Calculate total session duration, inhale count, and average inhale length
- [ ] Display results in an intuitive and visually clear format
- [ ] Integrate with existing entry detail layout and styling
- [ ] Add basic accessibility support for screen readers

### Acceptance Criteria
- [ ] Summary panel appears correctly on the Entry Detail
```

screen

- [] Data is accurately computed and updated dynamically
- [] Panel styling matches overall app theme and UX guidelines

Labels

feature, UI, analytics, enhancement

Estimate

S

markdown

Title

Implement Timeline View for Daily Inhalation Entries

Description

Introduce a new Timeline View feature to visualize inhalation entries chronologically for a selected day. This view will help users observe patterns of usage throughout the day, making it easier to recognize trends and triggers.

Tasks

- [] Create a new timeline component for daily inhalation entries
- [] Parse and sort inhalation data by timestamp
- [] Display inhalation duration and spacing as proportional visual elements
- [] Allow horizontal scrolling for long sessions
- [] Ensure responsiveness on various screen sizes

Acceptance Criteria

- [] Timeline view displays correctly with accurate time mapping
- [] Visual layout is clean and matches the app's aesthetic
- [] Supports edge cases like overlapping or dense entries

Labels

feature, UI, visualization, enhancement

Estimate

M

markdown

Title

Add “Usage Summary” Card to Dashboard

Description

Create a “Usage Summary” card on the main dashboard that displays the total number of sessions, total inhalation time, and average session duration for the selected date range. This feature provides quick, glanceable metrics for self-awareness and habit tracking.

Tasks

- [] Design UI for a new summary card
- [] Fetch and calculate relevant metrics from logs
- [] Format values clearly with appropriate units
- [] Integrate the card into the existing dashboard layout

Acceptance Criteria

- [] Summary card accurately reflects total sessions, total time, and average duration
- [] Responsive and visually consistent with other dashboard elements
- [] Handles empty data and edge cases gracefully

Labels

feature, dashboard, analytics, enhancement

Estimate

M

markdown

Title

Implement Calendar Heatmap for Daily Usage Visualization

Description

Develop a calendar-style heatmap to show daily usage intensity over a selected period (e.g., past 30/60/90 days). Each day's color intensity corresponds to the total inhalation time or number of sessions. This will help users recognize patterns, streaks, and gaps in their logging.

Tasks

- [] Design calendar heatmap layout (mobile-friendly)
- [] Aggregate daily usage data from Firestore
- [] Implement color gradient scale based on usage intensity
- [] Allow dynamic range selection (30, 60, 90 days)
- [] Add tooltips or tap interaction to show details for each day

Acceptance Criteria

- [] Heatmap accurately reflects user activity per day
- [] Responsive and readable across device sizes
- [] Smooth performance for large datasets

Labels

feature, visualization, dashboard, enhancement

Estimate

L

markdown

Title

Add Rolling Average Chart for Weekly and Monthly Usage Trends

Description

Introduce a new chart that displays rolling average trends over time (7-day and 30-day windows) to help users identify long-term changes in behavior. This visualization complements

the calendar heatmap and supports deeper habit analysis.

Tasks

- [] Compute 7-day and 30-day rolling averages from inhalation data
- [] Implement line chart with selectable range toggle (weekly/monthly)
- [] Ensure rolling average updates dynamically with selected date range
- [] Style chart to match overall dashboard theme

Acceptance Criteria

- [] Chart reflects accurate rolling average values
- [] Toggle between weekly and monthly views functions correctly
- [] Performance is smooth with large datasets

Labels

feature, visualization, analytics, enhancement

Estimate

M

markdown

Title

Implement Editable Note Field for Each Log Entry

Description

Allow users to add, edit, and delete personal notes for each smoking log entry. This feature gives users the ability to annotate logs with context, such as triggers, emotions, or activities at the time.

Tasks

- [] Add `note` field to Firestore document structure
- [] Update submit and edit logic in `submitData` to support notes
- [] Display note icon in each table row if a note exists

- [] Create modal or inline text area for note editing
- [] Sync changes with Firestore and update state appropriately

Acceptance Criteria

- [] Notes can be added/edited inline or via modal
- [] Notes persist across app sessions
- [] Table visually indicates which entries have notes

Labels

feature, UX, enhancement, data

Estimate

M

markdown

Title

Add "Time Since Last Log" Indicator on Dashboard

Description

Display the elapsed time since the most recent log entry on the main dashboard screen. This feature will help users track how long it's been since their last inhalation and promote awareness of time gaps between sessions.

Tasks

- [] Retrieve timestamp of most recent log
- [] Calculate elapsed time dynamically using local time
- [] Format and display the time (e.g., "2h 15m ago") in a prominent area
- [] Update value every minute to remain accurate
- [] Add logic to handle "No logs yet" case

Acceptance Criteria

- [] Timer appears on the dashboard and updates in real-time
- [] When no logs exist, a message like "No logs yet" is shown
- [] Timer resets upon new entry submission

Labels
feature, UI, enhancement, tracking

Estimate
S

markdown

Title
Add Monthly Summary Chart to Dashboard

Description
Introduce a visual summary on the dashboard that displays monthly usage trends based on inhalation logs. The chart should help users see progress over time and recognize patterns.

Tasks

- [] Aggregate inhalation data by day over the current month
- [] Design a line or bar chart to visualize daily totals
- [] Add chart component to dashboard with responsive styling
- [] Allow horizontal scrolling for months with long date ranges
- [] Implement logic to switch between current and previous months

Acceptance Criteria

- [] Chart accurately displays daily inhalation totals for the current month
- [] Users can swipe or tap to view previous months
- [] Chart remains responsive and readable on all screen sizes

Labels
feature, analytics, UI, visualization

Estimate

M

markdown

Title

Implement Log Entry Detail View with Edit & Delete Options

Description

Create a dedicated screen or modal that shows detailed information about a specific inhalation log entry. Include options to edit or delete the entry. This enhances usability and supports log management directly from the list or chart views.

Tasks

- [] Design UI for log detail view showing timestamp, duration, and notes (if any)
- [] Add "Edit" button to modify entry fields
- [] Add "Delete" button with confirmation dialog
- [] Ensure smooth transition/animation when opening the detail view
- [] Connect actions to Firestore update and delete functions

Acceptance Criteria

- [] Users can open a detailed view from a log list or chart interaction
- [] Edits update the correct entry in Firestore
- [] Deleted entries are removed and reflected immediately in the UI
- [] Interface is responsive and accessible

Labels

feature, UI, firestore, log-management

Estimate

M

markdown

Title

Implement Settings Page with Theme Toggle and Data Management Options

Description

Design and develop a settings page for the AshTrail app that allows users to toggle between light and dark themes, manage their data (e.g., export logs, clear data), and view app version information.

Tasks

- [] Create UI layout for the settings page
- [] Add light/dark theme toggle
- [] Implement persistent theme preference using local storage or database
- [] Include options to export data (e.g., to CSV) and clear all log entries
- [] Display app version number and optional "About" section

Acceptance Criteria

- [] Users can change the theme and see immediate UI updates
- [] Export generates a properly formatted file with inhalation logs
- [] Clear data option prompts confirmation and deletes logs from Firestore
- [] Settings persist across app restarts

Labels

feature, UI, settings, theming, data-management

Estimate

M

markdown

Title

Build Chart Settings Panel for Custom Visualization

Preferences

Description

Add a settings panel that allows users to customize how their usage data is displayed in the charts. This feature will enhance usability and offer more meaningful insights through tailored visualizations.

Tasks

- [] Add a collapsible chart settings section above or beside the main chart
- [] Include options such as:
 - Time range presets (last 7 days, last 30 days, custom)
 - Chart type (line, bar, cumulative, etc.)
 - Rolling average toggle and period selector
- [] Ensure changes reflect immediately on the chart display
- [] Persist user preferences locally

Acceptance Criteria

- [] Users can select and see the effects of their preferences on the chart
- [] Settings persist between sessions
- [] Default chart view is still informative without configuration

Labels

feature, charts, UI, user-preferences

Estimate

M

markdown

Title

Implement Session Tagging for Enhanced Log Context

Description

Introduce the ability to tag sessions with custom labels (e.g., "Stressful Day", "Social Event", "First Day Back") at

the start or end of a log entry. This will allow users to analyze patterns more meaningfully and support richer chart filtering options in the future.

Tasks

- [] Add tag input field to the log submission modal
- [] Support multiple tags per session
- [] Update Firestore schema to store tags
- [] Reflect tags in the logs list and detail views
- [] Ensure tags can be filtered in future chart enhancements

Acceptance Criteria

- [] Users can assign tags to any smoke log
- [] Tags appear clearly in the UI alongside log entries
- [] Tags are saved and retrieved properly from Firestore

Labels

feature, metadata, firestore, UX

Estimate

M

markdown

Title

Add Ability to View and Export Raw Log Data

Description

Provide users with an option to view and export their raw smoking session data for personal analysis or backup purposes. This feature supports transparency, data ownership, and research-based usage.

Tasks

- [] Add a "Download Logs" button to the dashboard or settings page
- [] Format exported data as CSV (timestamp, length, tags, notes, etc.)
- [] Allow user to preview raw log data in-app before

download

- [] Ensure exported file includes necessary headers and is cleanly formatted

Acceptance Criteria

- [] Users can download all smoke logs in a CSV file
- [] Exported data includes all key fields per log entry
- [] Feature is accessible and clear in the UI

Labels

feature, data-export, UX, csv

Estimate

M

markdown

Title

Implement Custom Time Range Selector for Log Analytics

Description

Introduce a custom time range selector to allow users to analyze their smoking data over any desired period, beyond the current fixed options (e.g., "This Week", "Last Month"). This will improve flexibility in reviewing habits and trends.

Tasks

- [] Add UI for selecting custom date range (start and end date)
- [] Update chart components to reflect data for selected range
- [] Persist range selection for session or via local storage
- [] Ensure compatibility with existing time-based filters and analytics

Acceptance Criteria

- [] Users can select any date range and view corresponding log analytics
- [] Range persists while navigating within the analytics

dashboard

- [] No UI or data errors occur when range has no logs

Labels

feature, analytics, date-picker, UX

Estimate

M

markdown

Title

Improve Dashboard Responsiveness and Layout for Mobile Devices

Description

Refactor the AshTrail dashboard UI to enhance responsiveness and usability on mobile and smaller screens. Ensure charts, buttons, and input elements adjust appropriately to screen size, improving accessibility and visual clarity.

Tasks

- [] Implement responsive layout for dashboard components
- [] Adjust font sizes and padding/margins for smaller screens
- [] Test and optimize all chart views on mobile devices
- [] Validate date pickers and controls are usable on touchscreens

Acceptance Criteria

- [] Dashboard displays properly on various mobile screen sizes
- [] No overlap, scroll bugs, or cut-off elements on small screens
- [] Charts remain readable and interactive on mobile

Labels

enhancement, mobile, UX, UI, responsive-design

Estimate

M

markdown

Title

Implement Entry Tagging and Filtering System

Description

Add a tagging system to AshTrail's smoke log entries to enable categorization and improved filtering. Users should be able to tag logs with custom keywords (e.g., "morning", "stress", "after work") and use these tags to filter and analyze patterns.

Tasks

- [] Add tagging capability to log submission UI
- [] Store tags in Firestore documents alongside log entries
- [] Enable multi-tag selection in the filter menu
- [] Display tags in log entry summaries
- [] Allow users to manage and delete custom tags

Acceptance Criteria

- [] Users can add and see tags for each log entry
- [] Tags are saved and retrievable from Firestore
- [] Filter by one or more tags works seamlessly
- [] Tag management UI is intuitive and functional

Labels

feature, UI, filters, data-management, Firestore

Estimate

M

markdown

Title

Create Calendar View for Log Entries

Description

Introduce a calendar interface that allows users to view smoke log entries by date. This will enhance usability by offering a visual overview of usage patterns over time and make it easier to identify trends.

Tasks

- [] Design a calendar UI component
- [] Populate calendar with markers for days with entries
- [] Enable clicking a date to view/edit entries from that day
- [] Add navigation for month/year
- [] Highlight today's date and selected dates

Acceptance Criteria

- [] Calendar is visible and responsive on all screen sizes
- [] Days with logs show markers or highlights
- [] Clicking a date loads relevant entries
- [] No performance issues with large datasets

Labels

feature, UI, calendar, log-entries, enhancement

Estimate

M

markdown

Title

Implement Weekly and Monthly Summary Views

Description

Add weekly and monthly summary views to the dashboard to give users quick insights into their smoking habits. This feature will help visualize usage trends over time and support better habit tracking.

Tasks

- [] Create aggregation logic for weekly and monthly inhalation data
- [] Design summary cards or charts for total inhalations and duration
- [] Add UI toggle between daily, weekly, and monthly views
- [] Ensure summary updates dynamically with data changes
- [] Optimize for performance with large datasets

Acceptance Criteria

- [] Summary views show accurate totals and averages
- [] Users can switch between daily, weekly, and monthly views seamlessly
- [] Charts and summaries update in real-time with data changes
- [] View is responsive and visually consistent across devices

Labels

feature, enhancement, analytics, dashboard, UI

Estimate

M

markdown

Title

Implement Session Insights Chart with Peak Analysis

Description

Create a new visualization that shows individual smoking sessions with clear start and end times, along with peak inhalation durations. The goal is to give users granular insight into each session, helping them identify patterns and outliers in their usage.

Tasks

- [] Define what constitutes a session (e.g., 5+ min gap between logs)
- [] Group inhalation entries into session clusters

- [] Display each session as a separate segment on a timeline
- [] Highlight peak inhalation within each session
- [] Include tooltip or detail-on-hover for session summaries

Acceptance Criteria

- [] Sessions are correctly grouped and displayed
- [] Peak inhalation is visually distinct within each session
- [] Timeline is scrollable and responsive
- [] Chart renders efficiently with large datasets

Labels

feature, visualization, analytics, UI, enhancement

Estimate

M

markdown

Title

Add Rolling Average Usage Chart for Trend Analysis

Description

Implement a new line chart that displays a rolling average of inhalation duration over user-selectable timeframes (e.g., 1-day, 3-day, 7-day averages). This will help users see patterns in their usage and understand whether it's increasing, decreasing, or stabilizing over time.

Tasks

- [] Allow user to select rolling average timeframe
- [] Compute rolling average for inhalation durations
- [] Display rolling average chart alongside daily usage
- [] Ensure accurate handling of days with no data

Acceptance Criteria

- [] Rolling average chart renders correctly and updates based on selected timeframe
- [] Line accurately reflects trends in usage
- [] Works with full data history and recent inputs

Labels
feature, data-analysis, charting, enhancement

Estimate
M

markdown

Title
Create Log Export Feature for CSV/Excel Format

Description
Add a feature that allows users to export their inhalation logs as a CSV or Excel file. This will give users the ability to back up their data or analyze it further using external tools.

Tasks

- [] Add export button to UI
- [] Format log data as CSV or Excel-compatible format
- [] Include fields: timestamp, duration, date, notes (if any)
- [] Handle large data sets gracefully
- [] Provide download link or auto-download upon export

Acceptance Criteria

- [] Users can export their logs with accurate formatting
- [] File includes all relevant data fields
- [] Works consistently across supported devices/browsers

Labels
feature, export, enhancement, user-request

Estimate
M

markdown

Title

Implement Usage Insights Chart with Time Range Selector

Description

Create a dynamic chart that visualizes inhalation usage over time with the ability to filter by selectable time ranges (e.g., 1 week, 1 month, 3 months, all time). This will allow users to better understand their smoking patterns and progress.

Tasks

- [] Integrate chart component into UI
- [] Implement buttons or dropdown for time range selection
- [] Filter data dynamically based on selected range
- [] Display total puffs and average duration within range
- [] Ensure responsiveness on mobile and desktop

Acceptance Criteria

- [] Chart displays accurate data for the selected range
- [] Users can switch between ranges seamlessly
- [] UI updates in real time with minimal lag
- [] Visual style aligns with the current app theme

Labels

feature, visualization, data-analysis, enhancement

Estimate

L

markdown

Title

Redesign Log Entry Submission Form for Clarity and Ease of Use

Description

Update the log entry form to be more intuitive, accessible, and visually clean. The goal is to simplify the user experience when submitting a smoking log, reducing cognitive

load and making the form feel less clinical.

Tasks

- [] Redesign UI layout with clear input sections for timestamp, duration, and notes
- [] Use icons or tooltips to clarify field purposes
- [] Add default values or shortcuts for common input patterns (e.g., “just now”)
- [] Improve mobile responsiveness
- [] Validate inputs client-side for common errors

Acceptance Criteria

- [] Users can easily understand and complete the form on first use
- [] Mobile and desktop versions behave consistently
- [] Field validations prevent incomplete submissions
- [] Visual design aligns with overall app theme

Labels

ui, enhancement, usability, form

Estimate

M

markdown

Title

Implement Time Range Filtering for Log Analysis

Description

Enable users to filter their smoking logs by custom date and time ranges to better understand patterns over specific periods (e.g., daily, weekly, monthly).

Tasks

- [] Add UI controls for selecting start and end dates/times
- [] Integrate filter logic into chart and table views
- [] Ensure filtered results update reactively
- [] Display active filter range above logs/charts

- [] Persist selected range in local storage or session

Acceptance Criteria

- [] Users can define a custom date/time range
- [] Charts and tables reflect only entries within the selected range
- [] Range filter UI is intuitive and responsive

Labels

feature, frontend, filters, UX

Estimate

M

markdown

Title

Add Rolling Average to Smoke Usage Charts

Description

To better visualize trends in smoking habits over time, implement a rolling average overlay on the usage chart (e.g., 3-day, 7-day average).

Tasks

- [] Allow user to toggle rolling average display on chart
- [] Implement logic for calculating rolling averages based on timestamped data
- [] Provide default averaging options (e.g., 3-day, 7-day, custom)
- [] Update chart UI to display both raw and averaged lines distinctly
- [] Ensure performance is optimized for long-term datasets

Acceptance Criteria

- [] Users can enable a rolling average on their usage chart
- [] Rolling average line accurately reflects selected period
- [] Chart remains responsive and legible with overlay

Labels

enhancement, analytics, frontend, UX

Estimate

M

markdown

Title

Implement In-App Feedback Submission Modal

Description

Add a user feedback feature that allows users to quickly report bugs, suggest features, or leave general feedback within the app.

Tasks

- [] Design a floating action button (FAB) or menu option for feedback
- [] Create modal with fields for category (bug, suggestion, other), text input, and optional contact info
- [] Integrate with Firestore or backend to log feedback entries
- [] Display success message upon submission
- [] Add validation and error handling for form inputs

Acceptance Criteria

- [] Users can submit feedback easily without leaving the app
- [] Feedback is saved and categorized appropriately
- [] UI is non-intrusive and accessible on all screens

Labels

feature, feedback, UX, enhancement

Estimate

S

markdown

Title

Implement Onboarding Carousel for First-Time Users

Description

Introduce an onboarding experience to guide new users through key features and benefits of the AshTrail (Smoke-Log) app.

Tasks

- [] Design and implement a multi-screen onboarding carousel
- [] Highlight key features such as logging sessions, viewing trends, and setting goals
- [] Include skip and next buttons, with progress indicators
- [] Store onboarding completion state locally to avoid repeat displays
- [] Ensure onboarding is responsive and works on all screen sizes

Acceptance Criteria

- [] Onboarding appears only once for new users
- [] Each screen clearly communicates a value proposition
- [] Users can skip or complete onboarding smoothly

Labels

feature, onboarding, UX, enhancement

Estimate

M

markdown

Title

Add Weekly and Monthly Summary Views to Dashboard

Description

Enhance the dashboard by including weekly and monthly summaries to help users better understand trends in their inhalation data.

Tasks

- [] Implement toggle for selecting daily, weekly, or monthly views
- [] Aggregate inhalation data accordingly for each selected range
- [] Update charts and summary stats based on the selected timeframe
- [] Ensure UI remains responsive and legible on all devices

Acceptance Criteria

- [] Users can switch between daily, weekly, and monthly summaries
- [] Summary views reflect accurate data aggregation
- [] Interface updates dynamically without full reload

Labels

feature, data-visualization, enhancement, dashboard

Estimate

M

markdown

Title

Implement Rolling Usage Analysis with Custom Time Ranges

Description

Introduce a rolling usage chart that provides a smoothed view of inhalation patterns over time. This will allow users to observe trends and fluctuations without the noise of daily variation.

Tasks

- [] Add a rolling average toggle to the dashboard
- [] Let users select window size (e.g., 3, 7, 14 days)
- [] Apply rolling average computation to inhalation duration data
- [] Render updated chart with smooth lines reflecting trends
- [] Ensure data updates in real-time when filters are

changed

Acceptance Criteria

- [] Rolling usage chart displays correct averages per selected window
- [] UI updates without requiring reload
- [] Feature is responsive and works on all supported devices

Labels

feature, analytics, charting, enhancement

Estimate

M

markdown

Title

Optimize Firebase Query Efficiency and Local Caching

Description

To reduce data load times and improve performance in the Smoke-Log app, optimize Firebase queries and implement local caching strategies.

Tasks

- [] Review current Firebase query structure for inefficiencies
- [] Implement indexed queries where necessary
- [] Cache recent logs locally using Isar
- [] Implement cache invalidation or refresh strategies for stale data
- [] Log and monitor performance improvements

Acceptance Criteria

- [] Query performance improves by at least 30% for typical users
- [] Local cache reduces redundant network requests
- [] User interface reflects updates promptly after cache refresh

Labels
performance, firebase, local-storage, backend

Estimate
M

markdown

Title
Implement Daily and Weekly Usage Charts

Description
Visualize user smoking trends over time by adding daily and weekly usage charts to the Smoke-Log dashboard. This will help users track habits and identify patterns in their smoking behavior.

Tasks

- [] Add chart toggle UI to switch between daily and weekly views
- [] Generate daily usage chart from timestamped log data
- [] Generate weekly usage chart grouped by week starting Monday
- [] Highlight peak usage periods for visibility
- [] Validate with sample user data

Acceptance Criteria

- [] User can toggle between daily and weekly views
- [] Charts accurately reflect log data and are visually distinct
- [] Weekly chart groups logs by week starting Monday
- [] Performance remains responsive with large datasets

Labels
feature, visualization, charts, data-analysis

Estimate

M

markdown

Title

Add Rolling Average Line to Usage Charts

Description

Enhance the Smoke-Log daily and weekly usage charts with a rolling average line to help users understand trends over time and smooth out irregularities in smoking behavior data.

Tasks

- [] Calculate 7-day rolling average for daily chart
- [] Calculate 4-week rolling average for weekly chart
- [] Overlay rolling average line on each respective chart
- [] Style line distinctively (e.g. dashed or colored)
- [] Add legend explaining the average line
- [] Test visual clarity with sparse and dense datasets

Acceptance Criteria

- [] Average line updates correctly with range filters
- [] Chart remains performant with large datasets
- [] Line visually complements rather than clutters chart

Labels

enhancement, charts, UX, data-visualization

Estimate

S

markdown

Title

Implement Local Database Caching with Isar

Description

To support offline access and long-term storage of log data, implement local database caching using Isar. This ensures users can view historical smoking logs even without internet access and improves app performance for larger datasets.

Tasks

- [] Integrate Isar into the Flutter project
- [] Define data models for inhalation logs compatible with Firestore schema
- [] Implement sync mechanism to pull Firestore logs into Isar
- [] Display logs from local Isar database when offline
- [] Automatically sync with Firestore when reconnected
- [] Handle data conflict resolution and deduplication

Acceptance Criteria

- [] Logs load instantly from local storage
- [] Syncing updates both local and remote data accurately
- [] Offline changes persist and sync upon reconnection
- [] No duplication or data loss occurs during sync

Labels

offline-support, performance, enhancement, data-storage

Estimate

M

markdown

Title

Create Summary Insights Panel for Weekly and Monthly Trends

Description

Implement a summary panel on the AshTrail dashboard to provide users with quick insights into their smoking habits. This panel should show aggregated statistics over selectable time ranges (weekly and monthly), such as average daily use, total sessions, average session duration, and longest streak without use.

Tasks

- [] Design UI component for summary panel
- [] Add dropdown or toggle to select time range (week/month)
- [] Implement backend logic to compute aggregate statistics
- [] Display computed statistics in the panel
- [] Ensure responsive design and accessibility

Acceptance Criteria

- [] Users can toggle between weekly and monthly summary views
- [] Summary data updates dynamically with correct values
- [] Panel integrates smoothly with existing dashboard UI
- [] Includes appropriate icons and visual cues

Labels

dashboard, ui-enhancement, insights, user-feedback

Estimate

M

markdown

Title

Add Rolling Averages to Inhalation Charts

Description

Enhance the inhalation charts in AshTrail by implementing rolling average lines to help users visualize trends over time more clearly. Users should be able to toggle the rolling average on/off and choose the interval (e.g., 3-day, 7-day, 30-day).

Tasks

- [] Add toggle control for enabling/disabling rolling average display
- [] Allow selection of rolling average interval
- [] Compute rolling averages for inhalation length over time
- [] Integrate the rolling average line into existing chart

```
visualizations
- [ ] Ensure UI updates dynamically and smoothly on changes

### Acceptance Criteria
- [ ] Users can view and interact with rolling average settings
- [ ] Line chart updates to include the selected rolling average
- [ ] Performance is optimized for large datasets

### Labels
charts, enhancement, analytics, visualization

### Estimate
M
```

markdown

```
### Title
Add Daily Summary View with Total Inhalations and Duration

### Description
Create a daily summary card or view in the AshTrail app that aggregates inhalation data by date. Each day's summary should include:
- Total number of inhalation entries
- Cumulative inhalation time (in seconds or minutes)
- Average inhalation length
- Optional: Longest inhalation of the day

### Tasks
- [ ] Group inhalation entries by day using timestamps
- [ ] Calculate and display total count, cumulative duration, and average inhalation per day
- [ ] Design a UI card or table view for presenting each day's data
- [ ] Ensure compatibility with existing filters (e.g., date range selection)
```

Acceptance Criteria

- [] Daily summaries appear in descending date order
- [] All stats update correctly when new data is submitted
- [] Layout is mobile-friendly and easy to interpret

Labels

ui, analytics, enhancement, summary

Estimate

M

markdown

Title

Implement Rolling Usage Chart with Selectable Time Windows

Description

Enhance the AshTrail charting capabilities by introducing a rolling usage chart. This chart should visualize inhalation trends over user-defined rolling periods (e.g., 3-day, 7-day averages).

Tasks

- [] Add selector to choose rolling average duration (e.g., 1-day, 3-day, 7-day)
- [] Calculate rolling averages based on timestamped data
- [] Update chart rendering logic to display smoothed usage data
- [] Integrate rolling chart with existing date filters and UI layout

Acceptance Criteria

- [] Chart updates instantly when a new rolling window is selected
- [] Rolling averages correctly reflect cumulative usage over the selected period
- [] UI is responsive and visually distinct from standard total usage chart

Labels

feature, data-visualization, enhancement, time-series

Estimate

M

markdown

Title

Add Timestamp Grouping Option for Aggregated Usage Views

Description

Introduce the ability to group inhalation data by hour, day, week, or month to better visualize patterns and usage trends in the AshTrail (Smoke-Log) app.

Tasks

- [] Add dropdown or segmented control for selecting grouping interval (hour/day/week/month)
- [] Modify data query logic to aggregate based on selected grouping
- [] Update charting logic to display grouped bars or lines appropriately
- [] Ensure grouping works seamlessly with date range and rolling window filters

Acceptance Criteria

- [] Grouped data displays correctly on charts and adjusts dynamically with filters
- [] Aggregation logic handles edge cases (e.g., partial weeks, daylight savings)
- [] UI provides clear labels for grouped time intervals

Labels

feature, enhancement, data-aggregation, ui/ux

Estimate

M

markdown

Title

Implement Summary Stats Section Above Charts

Description

Add a summary section above the charts to show quick stats based on the selected date range and filters. This section should display total sessions, total inhalation time, average session duration, and longest session.

Tasks

- [] Create a UI component for displaying summary statistics
- [] Calculate stats from filtered and grouped data
- [] Integrate the component into the dashboard above the charts
- [] Style the component for clarity and readability

Acceptance Criteria

- [] Stats update dynamically with date range or filter changes
- [] Display values are accurate and match charted data
- [] Layout is responsive and visually consistent with the app

Labels

feature, enhancement, stats, ui/ux

Estimate

S

markdown

Title

Implement Calendar-Based Session View

Description

Introduce a calendar-style view where users can select a day

and see the sessions recorded for that date. This feature aims to make browsing historical sessions more intuitive and visual.

Tasks

- [] Add a calendar UI component to the dashboard
- [] Connect calendar selections to existing filtering logic
- [] Display a list or chart of sessions below the calendar for the selected day
- [] Highlight days with recorded sessions

Acceptance Criteria

- [] User can view session data by selecting dates
- [] Days with sessions are visually marked
- [] Calendar is responsive and integrates with existing filters

Labels

feature, enhancement, calendar, ui/ux

Estimate

M

markdown

Title

Add Visual Summary for Weekly Smoking Trends

Description

Create a new dashboard component that summarizes smoking activity over the past 7 days. This should include a bar graph or line chart displaying total inhalation duration per day, with visual indicators for heavy vs. light days.

Tasks

- [] Design and implement a chart component for weekly data
- [] Aggregate session data by day for the past 7 days
- [] Apply color coding or markers for significant variations in usage

- [] Ensure responsiveness and proper rendering on mobile

Acceptance Criteria

- [] Chart displays the last 7 days with daily usage
- [] Visual indicators differentiate usage intensity
- [] Component loads dynamically with available session data

Labels

feature, dashboard, analytics, ui/ux

Estimate

M

markdown

Title

Implement "Longest Session" Highlight Card on Dashboard

Description

Add a UI card to the main dashboard that highlights the longest recorded inhalation session. This card should display the duration, timestamp, and optional label if present. It should be visually distinct to draw user attention.

Tasks

- [] Query the dataset for the session with the longest inhalation length
- [] Design a card UI component to feature the session prominently
- [] Display session duration, date/time, and label if applicable
- [] Add fallback messaging for users with no data

Acceptance Criteria

- [] Longest session is calculated accurately
- [] Card is visually distinct and integrates with the dashboard layout
- [] Responsive design for both desktop and mobile

Labels

feature, dashboard, data-visualization, ui/ux

Estimate

M

markdown

Title

Display Average Inhalation Length Over Selected Time Range

Description

Introduce a dashboard widget that calculates and displays the average inhalation length over the user-selected time range. This metric helps users track patterns and compare usage over time.

Tasks

- [] Implement logic to compute average inhalation length for a given range
- [] Add a visual card component to the dashboard to present the value
- [] Format output for readability (e.g., seconds with one decimal)
- [] Update dynamically when range selector changes

Acceptance Criteria

- [] Average value updates correctly with range selection
- [] Responsive and styled consistently with existing dashboard UI
- [] No errors when no data is available in the selected range

Labels

feature, dashboard, statistics, enhancement

Estimate

S

markdown

Title

Add Daily Usage Count Chart to Dashboard

Description

Implement a chart on the dashboard that shows the number of smoking sessions per day over a user-defined date range. This will help users visualize frequency trends and identify high-usage days.

Tasks

- [] Create Firestore query to group and count logs per day
- [] Build a new line or bar chart component for daily counts
- [] Integrate the chart into the dashboard layout
- [] Ensure compatibility with the existing range selector

Acceptance Criteria

- [] Chart displays daily counts correctly for any selected date range
- [] Visual style matches other dashboard charts
- [] Proper labels and tooltips appear on hover

Labels

feature, chart, dashboard, analytics

Estimate

M

markdown

Title

Add Cumulative Usage Time Chart with Rolling Average

Description

Implement a line chart on the dashboard that shows cumulative smoking session duration over time. Include an optional rolling average (e.g., 7-day) to help users visualize long-

term usage trends and compare their behavior over different time spans.

Tasks

- [] Extend data processing to calculate cumulative usage duration over time
- [] Implement optional rolling average calculation (e.g., 7-day)
- [] Create chart component to display cumulative and rolling average lines
- [] Add toggle to enable/disable rolling average display

Acceptance Criteria

- [] Chart displays accurate cumulative usage with optional rolling average
- [] Performance remains smooth for large data sets
- [] Matches dashboard styling and layout conventions

Labels

feature, chart, analytics, enhancement

Estimate

M

markdown

Title

Implement Time Range Selector with Presets for Chart Filtering

Description

Add a time range selector to the dashboard that allows users to filter session charts by predefined ranges such as Last 7 Days, Last 30 Days, This Month, and Custom Range. This will enable more flexible exploration of usage patterns across different time spans.

Tasks

- [] Create time range selector component with preset options
- [] Implement custom date range input

- [] Connect selector to session data filtering logic
- [] Ensure all dashboard charts update dynamically based on selected range

Acceptance Criteria

- [] Users can select a time range from preset options or input a custom date range
- [] All charts update to reflect the selected time range without reload
- [] Selector matches visual style of other dashboard controls

Labels

feature, UI, filters, dashboard

Estimate

M

markdown

Title

Design and Implement Responsive Layout for Mobile Devices

Description

Adapt the AshTrail (Smoke-Log) dashboard interface for smaller screens to ensure usability on mobile devices. Elements should stack or resize appropriately, and interactions must remain intuitive and accessible.

Tasks

- [] Audit current UI components for responsiveness
- [] Define mobile breakpoints and layout behavior
- [] Refactor layout and components to support smaller screens
- [] Test design on various devices and orientations

Acceptance Criteria

- [] Dashboard is fully usable on screens down to 375px width
- [] No horizontal scrolling required

- [] Touch targets are appropriately sized and spaced
- [] Visual hierarchy remains clear and readable

Labels

enhancement, UI, responsive, mobile

Estimate

M

markdown

Title

Implement Persistent State for User Settings and Filters

Description

Allow users to persist their selected filters, chart preferences, and viewing ranges between sessions in the AshTrail (Smoke-Log) dashboard. This will improve user experience by maintaining customization.

Tasks

- [] Define which settings and filters should be persisted
- [] Implement state storage using localStorage or Firestore user-specific documents
- [] Load persisted settings on app start
- [] Add toggle/reset option for users to clear or disable persistence

Acceptance Criteria

- [] Settings are restored on page reload or user return
- [] Toggle/reset function is easily accessible and functional
- [] No performance degradation from persistence implementation

Labels

enhancement, UX, settings

Estimate

M

markdown

Title

Implement Timeline Mode for Cumulative Usage View

Description

Add a timeline mode to the AshTrail (Smoke-Log) dashboard that visualizes cumulative usage over time. This view should support zooming, panning, and optionally displaying annotations or notes tied to specific dates or events.

Tasks

- [] Design the UI/UX for timeline mode toggle and display
- [] Create logic to calculate and display cumulative usage
- [] Implement zoom and pan functionality
- [] Optional: Allow annotations or events to be added and displayed on the timeline

Acceptance Criteria

- [] Cumulative usage is displayed clearly and updates with filters
- [] Timeline is interactive with zoom/pan support
- [] Timeline integrates with existing data structure without performance issues

Labels

feature, visualization, UX

Estimate

L

markdown

Title

Add Real-Time Logging Indicator and Status Display

Description

Introduce a visual indicator to show when new data is being logged in real-time to the AshTrail (Smoke-Log) app. This should help users know when activity is currently being recorded or if logging is paused/stopped.

Tasks

- [] Design an unobtrusive but noticeable indicator (e.g., pulsing dot, blinking icon)
- [] Implement logic to detect and reflect live data submission
- [] Display status message (e.g., "Logging active", "No recent activity")
- [] Ensure this component is present across all primary dashboard views

Acceptance Criteria

- [] Indicator correctly reflects the logging state in real-time
- [] Status message updates dynamically based on activity
- [] UI updates do not affect performance or usability

Labels

feature, UX, real-time

Estimate

M

markdown

Title

Implement Smoke Type Classification System

Description

Enhance AshTrail by allowing users to classify the type of smoke (e.g., cigarette, vape, cigar, joint) when logging data. This feature will support deeper analytics and filtering in charts and reports.

Tasks

- [] Design UI for selecting smoke type during logging
- [] Update Firestore data schema to include `smokeType` field
- [] Add support for filtering and grouping by smoke type in charts
- [] Update test data and mock modes accordingly

Acceptance Criteria

- [] Users can select a smoke type before submitting a log
- [] Charts and reports reflect data segmented by smoke type
- [] Feature is functional in both development and production modes

Labels

feature, data, charts, enhancement

Estimate

M

markdown

Title

Create Quick-Add Smoke Button with Default Values

Description

To streamline the logging process, implement a "Quick Add" button that logs a smoke session using default or most recent values. This feature is aimed at users who smoke consistently and want to minimize interaction.

Tasks

- [] Add a "Quick Add" button to the main UI
- [] Determine logic for default values (most recent or configurable preset)
- [] Ensure offline and online compatibility
- [] Add visual confirmation of successful log entry

Acceptance Criteria

- [] Button is clearly labeled and accessible
- [] Logs are submitted using default or last used values
- [] Log appears immediately in the UI and is stored properly

Labels

feature, UX, logging, enhancement

Estimate

S

markdown

Title

Integrate Visual Timeline Chart for Daily Smoke Patterns

Description

Implement a horizontal timeline chart on the dashboard that visually represents the user's smoking sessions throughout the day. This helps users recognize patterns in their smoking behavior and provides a more intuitive understanding of their habits.

Tasks

- [] Add a chart area on the dashboard
- [] Implement a horizontal timeline chart using a suitable charting library (e.g., Chart.js, D3.js)
- [] Populate timeline with smoke session start times and durations
- [] Enable dynamic scaling for different days

Acceptance Criteria

- [] Chart renders daily smoke sessions as visual blocks on a timeline
- [] Users can toggle between today and previous days
- [] Tooltip displays start time, duration, and session type on hover

Labels

feature, UI, visualization, analytics

Estimate

M

markdown

Title

Implement Session Notes Input Field and Display

Description

Add the ability for users to write and save short notes or reflections about individual smoke sessions. These notes should appear alongside each entry in the session list and be editable.

Tasks

- [] Add an optional text input to the session logging form
- [] Store notes in the Firestore document under a `note` field
- [] Display saved notes below each session in the session history list
- [] Add edit/delete functionality for session notes

Acceptance Criteria

- [] Users can add a note when submitting a session
- [] Notes are stored and persist across sessions
- [] Notes are displayed clearly with the corresponding session entry
- [] Users can edit or delete their notes

Labels

feature, UX, firestore, input

Estimate

S