HABITFORGE

WWW.HABITFORGE.CA

CONTENT

OUR TEAM

Abhi Patel Scrum Master +Developer	Ahmed Yaser Product Owner +Developer	Ameen Khawaja Project Manager +Developer
Nico McFarlane Developer	Zakir Raza Developer	Rafael Bocsa Developer

INTRODUCTION

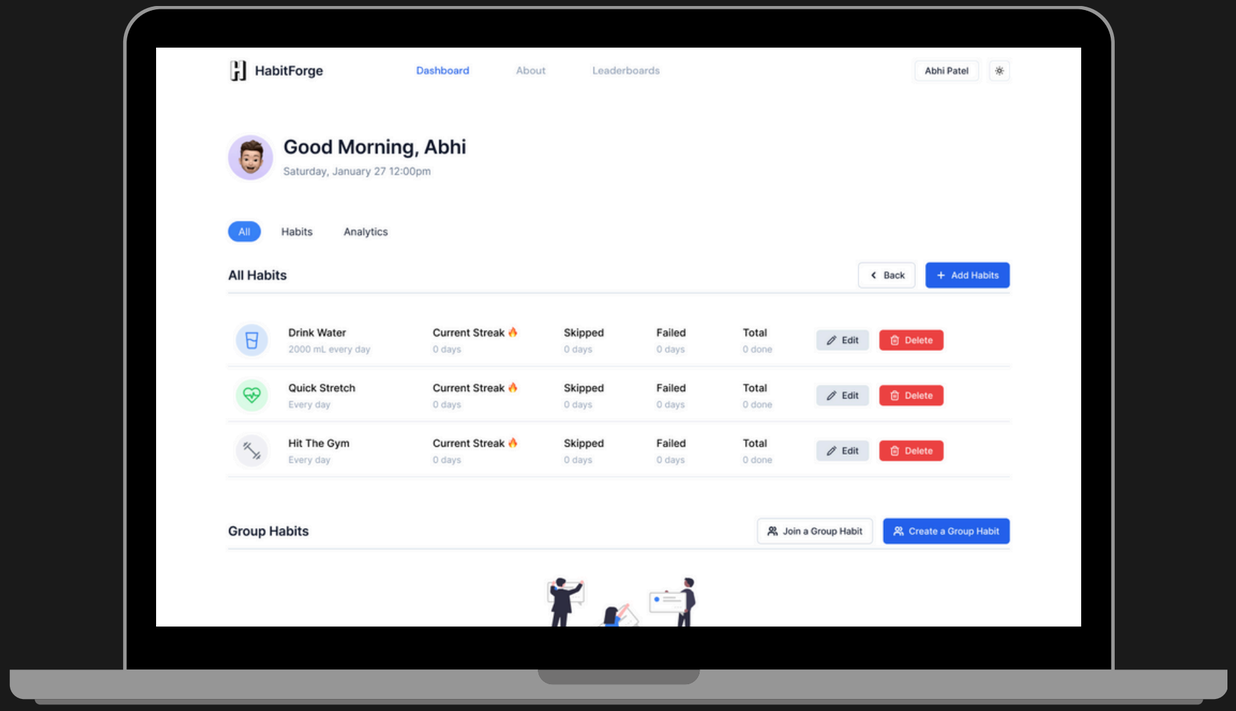
Signifigance

- A Habit Tracker is useful application that can be helpful to many individuals.
- It helps maintaining consistency and achieving goals by reducing procrastination and increasing motivation to complete a task in day to day lives.

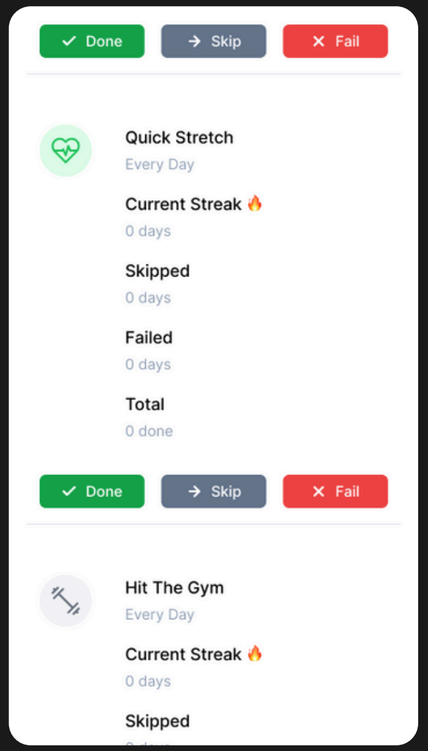
- It required months of weekly careful research, countless brainstorming meetings and meticulous attention to detail at every step.
- Each team member contributing their unique expertise after research to bring HabitForge to life.

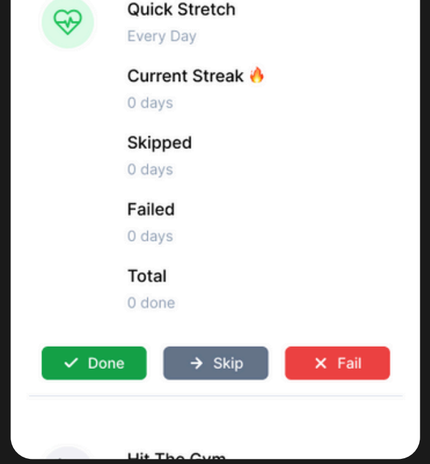
Developing

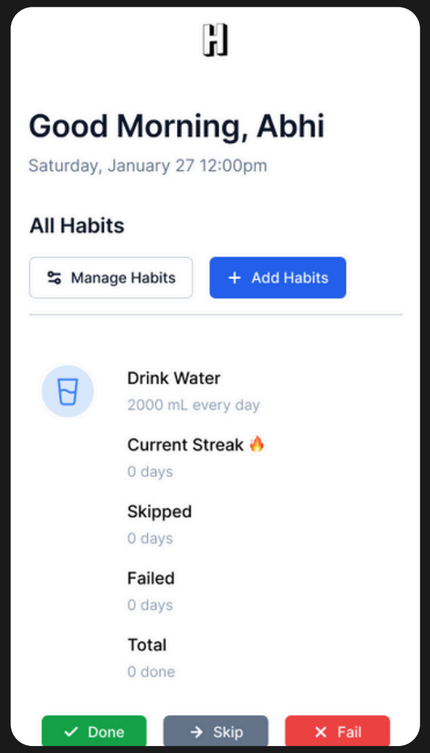
UI DESIGNS

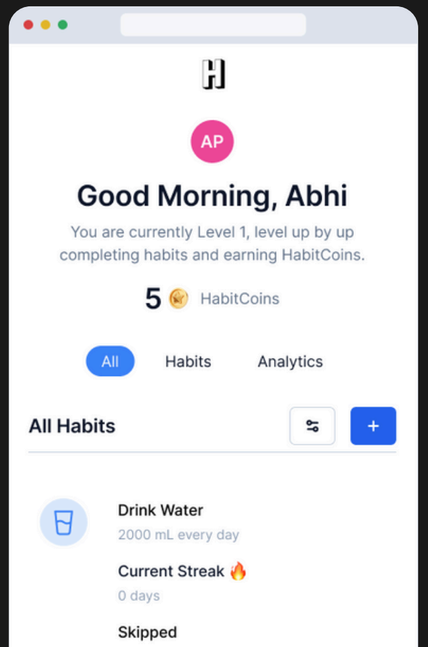


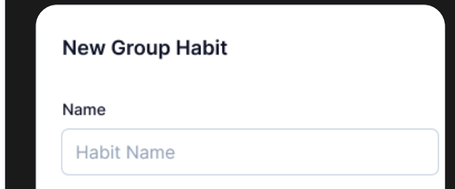
Figma Link











PROJECT PROCESS

PROCESS MODEL

Agile (Scrum) Methodology

Why we chose Agile

- We had a vision of how we wanted to develop HabitForge, but we were not certain on all the features we wanted to implement and how we were going to implement them

How we implemented Agile methodology?

- Every week, we revised our ideas and changed certain features throughout the lifecycle of the project
- Some weeks certain teammates would spend time researching particular implementations rather than developing
- HabitForge was developed as a series of versions/increments, where each sprint we would deploy the new changes live and test the features out
- All required documentation to keep the team and the TA well informed about what work is being done was completed.

PROJECT PROCESS

Abhi (Scrum Master)

- Set up the codebase to ensure smooth usage for all. This included creating the firebase project, Next.JS app, and GitHub repo for version control.
- Clear and descriptive sprints planned for the entire 4 months.
- Facilitated the sprint meetings.
- Informed all progress to the project owner and took any feedback to the development team.

Ameen (Project Manager)

- Wrote progress reports
- Setup project instructions and documentation
- Managed resource allocation
- Facilitated risk management
  - Including branch structure strategies
- Ensured all features requested by the product owner are delivered in the final app
- Sprint notes
  - Gather notes from weekly meetings

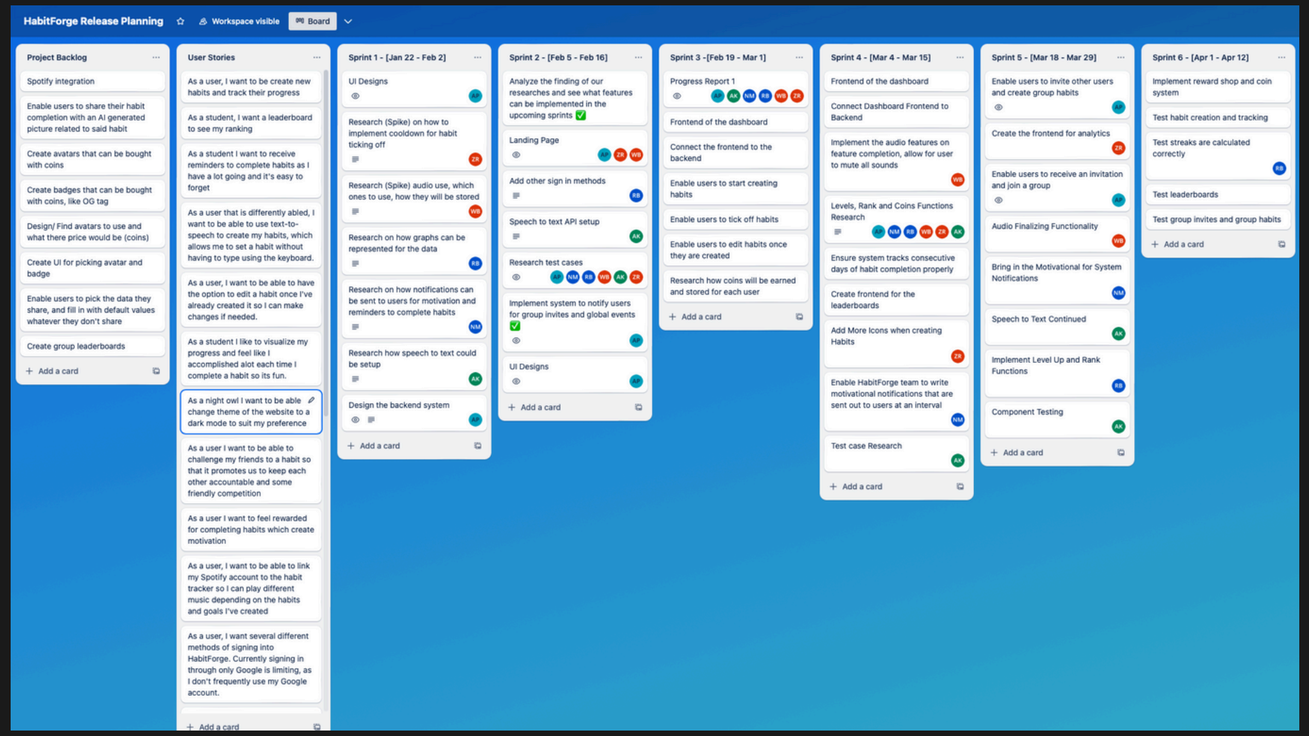
Ahmed (Product Owner)

- Defined project objectives and requirements
- Strategic Planning
  - Defined and communicated app vision
  - Goals and roadmap
- Facilitated user stories and feedback
- Enforced a user-centric design

PROJECT PROCESS

Sprint Breakdown

- Sprint 1 - [Research & UI Designs]
- Sprint 2 -[UI Designs, Set up code, Landing Page, Speech-to-text]
- Sprint 3 - [Creating Habits]
- Sprint 4 - [Ranks, Motivational Notifications, Stability]
- Sprint 5 - [Group Habits, Analytics, Component Testing]
- Sprint 6 - [Reward shop, clean up code]



Trello Board

FEATURES

FEATURES

Culmination of months of research and collaborative effort, the features are designed to empower the users on their journey on HabitForge.

Researched speech-to-text API's, graph representations, habit cooldown, notifications, and sounds.

SPRINT 1

Implemented a system to notify users for group invites and global events.

SPRINT 2

Enable users to create, tick-off, and edit habits. Research how coins will be earned.

SPRINT 3

FEATURES

Implement audio features with a mute option. Enable HabitForge to send motivational notifications to users.

SPRINT 4

Allow users to invite others and create group habits. Implement Level Up and Rank Functions.

SPRINT 5

Implement reward shop and coin system.

SPRINT 6

RANK SYSTEM

Research

- Formulate a scoring formula based off habit tracking data
- Rewarded players who have been tracking and completing habits on time

How?

- Wrote a cloud method that calculates each users rank via habit data
- Deployed cloud function to run every midnight

Importance

- A metric that users can use to see habit tracking behavior
- Used to order leaderboard to view where each user stands.

SPEECH-TO-TEXT

Research

- Google Cloud Speech-to-Text API
- AssemblyAI
- Mozilla SpeechRecognition API
- Settled on Mozilla despite its cons

How?

- React's integration with Mozilla’s Web Speech API simplifies speech-to-text functionality
- The API manages user speech, featuring start and stop listening functions.
- A recognition component was developed and seamlessly added to Abhi's "Add Habit" UI.

Importance

- Convenience
- Increases ease-of-use of website
- Improves user experience
- Increases accessibility for all users
- An additional feature that a user can opt-in to use if they want to

HABIT COOLDOWN

Research

- Limit the number of habits you can completed depending on certain factors
- Selecting icons based on the habits

How?

- Every level you can create 2 times the level of habits
- For example:
  - Level 1 its 1x2 = 2 max habit, +1 for group habits
  - Level 2 its 2*2 = 4 max habit, +2 for group habits
- Level 5 is max

Importance

- Functionality to avoid users exploiting our system
- Enables a fair and level playing field for habit tracking
- Users can track habit completions

INPUT SOUNDS

Research	How?	Importance
• Different usage of sound indicate different achievements and moods. • Audio libraries (SoundBible and FreeSound)	• Storing ambient sounds in firebase • Enabled sounds to play when user complets certain tasks.	• Provides audio feedback to users, enhancing user experience • Distinct sounds to different actions • Maintaining Simplicity (not overwhelming)

SYSTEM NOTIFICATIONS

Research

- Notification API
- Permissions on web API/system
- Cloud functions

How?

- Send out all notifications to all users within the system at a specified time everyday
- FCM Registration Token and Payload Creating/sending

Importance

- Engagement
- Personalized motivational quotes
- Automation (efficient)
- Enhanced user experience

UNIT TESTING

How It Works

- Testing was done using Jest, an automated testing framework for JavaScript
- Major components for HabitForge is tested by writing a test class for the component, followed by several test cases
- Works by creating mock data, and then simulating a scenario

Completion

```
PASS  __tests__/GroupHabitsDialog.test.jsx
PASS  __tests__/AudioHome.test.jsx
PASS  __tests__/ModeToggle.test.jsx
PASS  __tests__/Podium.test.jsx
PASS  __tests__/AnalyticsSec.test.jsx
PASS  __tests__/MuteProvider.test.jsx
PASS  __tests__/Edit.test.jsx
PASS  __tests__/TimeElapsed.test.jsx
PASS  __tests__/UseSpeechRecognitionHook.test.jsx

Test Suites: 12 passed, 12 total
Tests:       60 passed, 60 total
Snapshots:   0 total
Time:        1.965 s
```

Importance

- A metric to test and validate every component
- Ensure that every component works even after modifications are done to it

GROUP HABITS

What?

- Group habits enable users to feel part of a community.
- What the strucutre of notifications and groups will look like?

How?

- Front-end to allow users to create group habits and invite members by email.
- Back-end that handles the creation and sends an invite and acceptance of invite.

Importance

- Community engagement
- Collaborative Goals
- Prioritizes user experience
- Accountability for habits
- Group Notifications engage more users

OTHER FEATURES

- Sign in methods
- Analytics
- Dashboard
- Leaderboards
- Light/Dark/ System mode
- Changing profile avatars/Coin shop
- Browser Notifications

DEMO

VIDEO LINK

THANKS FOR LISTENING

HABITFORGE