

Admin Panel – Front-end Documentation

By: - Arjun (Front-end Lead)

Overview: The three components (AdminDashboard, UserManagementScreen, and ListenerDetailScreen) collectively form a comprehensive Tier-1 Administrative Suite. Together, they establish a clear "Drill-Down" architecture that allows an administrator to move from high-level business metrics to specific user-level moderation actions seamlessly.

2. Technical Architecture

The application is built using the following modern technology stack:

- **Framework:** React Native (via Expo)
- **Routing:** Expo Router (File-based routing)
- **Language:** TypeScript
- **Styling:** Stylesheet API with Custom Theme Constants
- **Icons:** Expo Vector Icons
- **Gradient Engine:** expo-linear-gradient

3. Design System & UI/UX

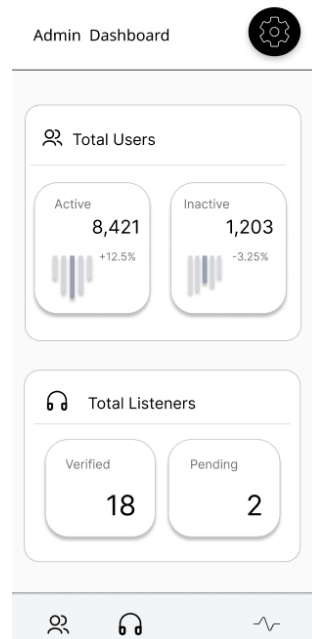
The application follows a strict "Neon/Dark" design language to ensure visual distinctiveness and reduce eye strain in low-light environments.

3.1 Color Palette

- **Primary Background:** #1A1225 (Deep Purple/Black)
- **Card Background:** #251D30 (Glassmorphic Dark)
- **Primary Accent:** #00FFFF (Neon Cyan)
- **Secondary Text:** #A0A0A0 (Light Grey)
- **Success State:** #00FF9D (Neon Green)

Screens worked on:

1. Admin Dashboard



A. Data Handling & State Management

The component utilizes standard React hooks for state management:

- **State:**
 - **data:** Stores the complex DashboardData object containing users, listeners, revenue, sessions, flagged items, and tickets.
 - **loading:** A boolean flag used to toggle the visualization of the initialization state.
- **Lifecycle:**
 - A `useEffect` hook triggers on mount to simulate an asynchronous API call.
 - A `setTimeout` of 1200ms is used to mock network latency before populating the state with static data.

B. Type Safety (TypeScript)

The code enforces strict typing through three core interfaces:

- **MetricData:** Defines the structure for simple counters (count, percentage change, trend direction).
- **ListEntry:** Defines the structure for list items (ID, title, subtitle, time, icon, priority).
- **DashboardData:** The master interface that aggregates all metrics into a single state object, ensuring data consistency across the dashboard.

3. Visual Design System

The UI implements a specific "Neon/Dark" theme defined in the COLORS constant:

- **Color Palette:**
 - **Background:** Deep Violet/Black (#1A1225).
 - **Accent:** Neon Cyan (#00FFFF) used for buttons, active charts, and icons.
 - **Status Colors:** Green (#4CD964) for positive trends and Red (#FF4B4B) for negative trends/alerts.
- **Styling Techniques:**
 - **Glassmorphism:** Components use semi-transparent backgrounds (rgba(255, 255, 255, 0.08)) with light borders to mimic frosted glass.
 - **Shadows:** Buttons utilize shadowColor and elevation to create a glowing neon effect.
 - **Safe Area:** The SafeAreaView ensures the UI does not overlap with physical device notches or status bars.

4. Functional Modules (Widgets)

The dashboard is divided into vertically stacked cards wrapped in a ScrollView:

1. **Header:** Contains the title and a navigation button to the Settings panel.
2. **User & Listener Metrics:**
 - Utilizes the reusable DataCard sub-component.
 - Displays split statistics (Active vs. Inactive Users; Verified vs. Pending Listeners).
3. **Revenue Analytics:**
 - Displays current monthly revenue and percentage growth.
 - **Visualization:** Renders a custom Bar Chart using mapped View components with dynamic heights and opacity logic to visualize trends without external charting libraries.
4. **Session Statistics:**
 - Similar visualization to Revenue, tracking Total Sessions and Average Duration.
 - Bar opacity changes based on data magnitude (highlighting peaks).
5. **Support Ticket Queue:**
 - Lists recent high-priority tickets using the ListItem sub-component.
 - Includes a "View All Tickets" button.
6. **Flagged Content:**
 - Lists moderation alerts (e.g., Inappropriate language).
 - Uses distinct iconography (Alert Octagon) and red color coding to denote urgency.

5. Navigation Logic

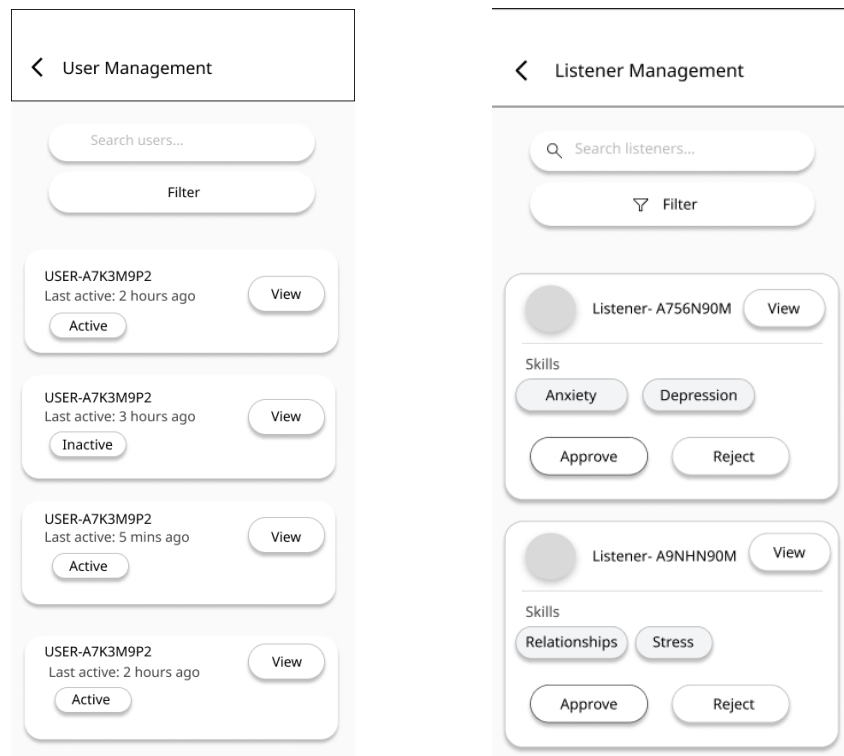
The component relies on expo-router for screen transitions. Three explicit routes are defined:

- **Settings:** ../(Dashboard)/Setting/Settings
- **Tickets:** ../(Dashboard)/Tickets/Ticket
- **Reports:** ../(Dashboard)/Reports/Flagged

6. Code Quality & Reusability

- **Modularization:** The DataCard and ListItem components are extracted from the main render function. This reduces code duplication and ensures visual consistency across different sections of the dashboard.
- **Responsiveness:** Flexbox is used throughout (flexDirection, justifyContent) to ensure the layout adapts to different screen widths.

2. User and Listener Management



A. Data Flow & State Logic

The component employs a "Fetch-then-Filter" architectural pattern:

1. **Initialization:** On mount, `useEffect` triggers a mock asynchronous fetch (1.2s delay).
2. **State Storage:** Raw user data is stored in the `users` state array.
3. **Computed View:** The `filteredUsers` constant is a derived value. It filters the raw `users` array in real-time based on the search state text input. This ensures that the UI updates instantly as the user types, without needing new API calls.

B. Dynamic Navigation (Expo Router)

The component utilizes Expo Router's object-based navigation method to handle dynamic routing:

- **Target Route:** `../Usr_Management/[id]`
- **Parameter Passing:** When the "View" button is pressed, the specific `userId` is passed as a route parameter (`params: { id: user.userId }`). This allows the destination screen to know which user profile to fetch and display.

C. Safe Area Handling

Unlike the previous dashboard which used a wrapper component, this screen uses the `useSafeAreaInsets` hook.

- **Logic:** `paddingTop: insets.top > 0 ? 10 : 20`
- **Purpose:** This provides granular control over the top padding, ensuring the search bar sits comfortably below the status bar on notched devices (like iPhone 14/15) while maintaining consistent spacing on older devices.

3. Visual Design System

The "Neon/Dark" theme is applied consistently with the Admin Dashboard:

- **Glassmorphism:** The Search Bar, Filter Button, and User Cards all use a highly transparent white background (`rgba(255, 255, 255, 0.08)`) with a subtle border to create depth against the deep purple background.
- **Dynamic Status Indicators (Pills):**
 - **Active:** Renders with a Cyan tint (`rgba(0, 255, 255, 0.1)`) and Neon Cyan text.
 - **Inactive:** Renders with a faded white tint and dimmed text.

- **Implementation:** The code uses conditional styling arrays within the UserCard component to switch styles based on the user.status prop.

4. Functional Modules

A. Search & Filter Section

- **Input:** A standardized text input styled to blend with the glass theme.
- **Action:** A placeholder "Filter" button is included for future implementation (e.g., filtering by "Active only" or "Inactive only").

B. User Card (Sub-Component)

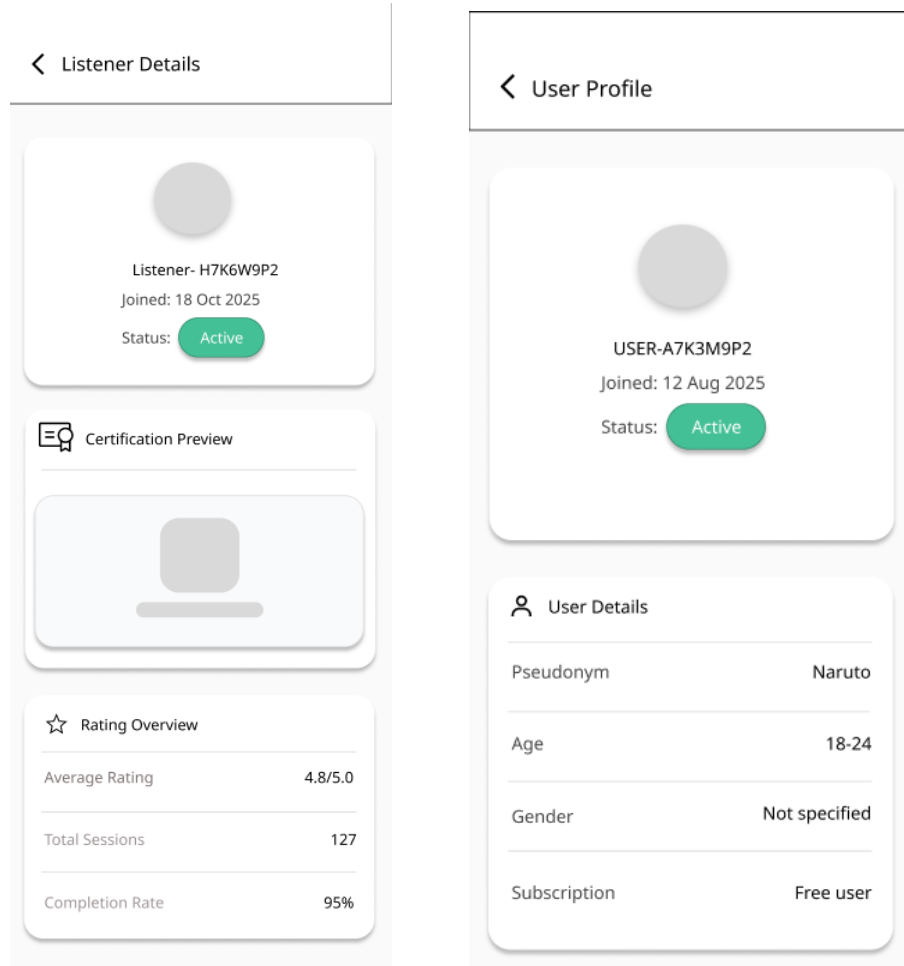
To maintain clean code, the individual list items are extracted into a UserCard component. This component handles:

- **Information Display:** Formats the User ID and "Last active" timestamp.
- **Visual Feedback:** Provides activeOpacity on the "View" button for tactile feedback during interaction.

5. Type definitions

```
interface UserItem {  
    id: string;  
    userId: string;  
    lastActive: string;  
    status: "Active" | "Inactive";  
}
```

4. Listener and User Profile



A. Navigation & Parameter Handling

The component acts as a dynamic route receiver.

- **Hook:** `useLocalSearchParams()` is used to extract the id parameter from the URL.
- **Fallback Logic:** If no ID is provided (e.g., during direct preview), it defaults to a hardcoded placeholder "H7K6W9P2".

B. Component Composition (Atomic Design)

To maintain maintainability and visual consistency, the UI is built using granular, reusable sub-components:

- **GlassCard:** A wrapper component that applies the standard "glassmorphism" container style (background opacity, border, rounded corners) and optionally renders a standard header with an icon.
- **InfoRow:** A standardized key-value pair display used in the "User Details" section to ensure alignment.
- **AdminActionBtn:** A smart button component that accepts a color prop, allowing it to dynamically style itself for different contexts (e.g., Red for "Suspend", Green for "Release Payment") without duplicating style code.

C. Layout & Safe Area

- **SafeAreaView:** The layout is wrapped in a SafeAreaView with specific edge props (edges=[{"bottom", "left", "right"}]). This ensures the content doesn't get cut off by the home indicator on iOS devices, while the top is likely handled by a header or parent layout.
- **ScrollView:** Content is scrollable to accommodate varying lengths of activity logs or notes on smaller screens.

3. Visual Design System

The component strictly adheres to the "Neon/Dark" theme:

- **Color Semantics:**
 - **Identity:** Neon Cyan (#00FFFF) is used for primary icons and active elements.
 - **Risk/Action Levels:**
 - **Success (Green):** Used for "Release Payment" and "Active" status.
 - **Warning (Yellow):** Used for "Hold Payment" and "Send Warning".
 - **Danger (Red):** Used for "Suspend Listener".
- **Visual Hierarchy:**
 - The **Profile Header** is the most prominent element, centered with a large avatar placeholder.
 - **Skeleton Loading:** The "Notes" section simulates a loading state or empty state using skeletonLine styles, providing a visual cue for where text will appear.

4. Functional Modules

1. **Profile Header:** Displays the user's core identity (Avatar, ID, Join Date) and current status (Active/Suspended).
2. **User Details:** A read-only section for demographic and account info (Pseudonym, Age, Gender, Subscription Tier).
3. **Activity Log:** A chronological list of system events (e.g., "Joined platform", "Updated profile"). It uses a standard mapping function to render rows.
4. **Admin Actions:** A control panel for moderation. This is the interactive core of the screen, allowing admins to perform business-critical operations.
5. **Notes & Flags:** An internal communication section for admins to leave context about the user.

5. Code Quality & Extensibility

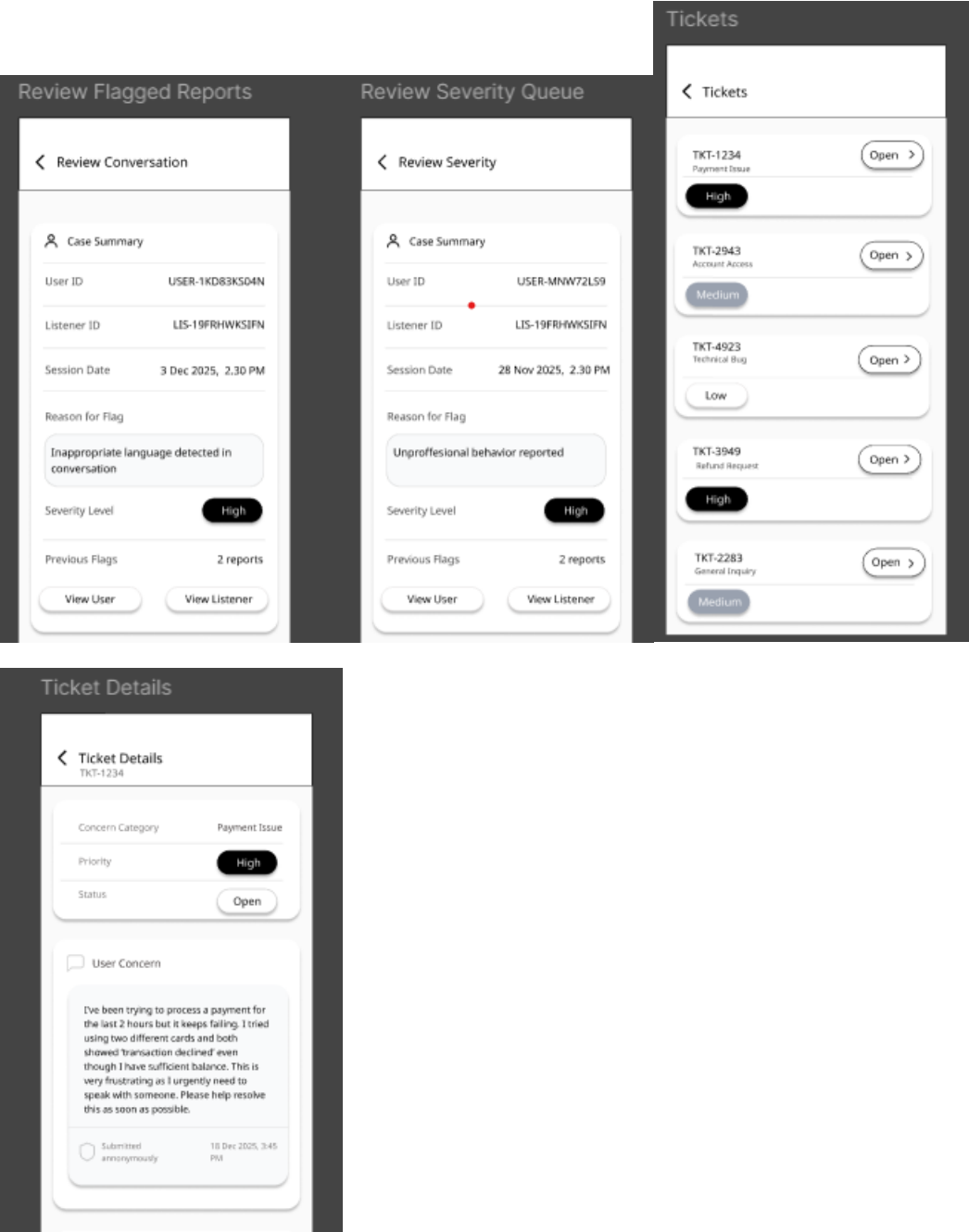
- **Modular Styling:** The styles object is well-organized, with specific sections for "Glass Cards," "Admin Actions," and "Activity Log." This makes tweaking the design system easy.
- **Icon Integration:** The code leverages multiple icon sets (Feather, MaterialCommunityIcons, Octicons) from @expo/vector-icons to provide semantic context (e.g., a shield for admin actions, a clock for history).

Conclusion

I have built a visually polished and structurally sound frontend for an admin application. The UI logic handles loading states, empty states, and user interactions effectively. The modular component structure (extracting GlassCard, UserCard, etc.) proves that the codebase is ready for scalability.

ADMIN PANEL – UI/UX FUNCTIONAL SCREENS REPORT

Designed by: Vijay T S



1. Overview

The Admin Panel serves as the central governance, moderation, and safety management interface of the Anonymous Listening Platform. It is designed to support administrators in handling sensitive user interactions, enforcing platform guidelines, and resolving support issues efficiently and fairly.

Given the high responsibility associated with moderation and safety decisions, the Admin Panel emphasizes clarity, contextual awareness, and structured workflows. The interface ensures administrators can review cases thoroughly, prioritize urgency, and take informed actions without unnecessary cognitive overload.

The overall experience focuses on:

- Safety-first decision making
 - Clear visibility of critical information
 - Severity-based prioritization of cases
 - Efficient handling of support tickets
 - Maintaining fairness, transparency, and platform trust
-

2. Design System Reference

Typography

- Font Family: System default for consistency and readability
- 16px – Page titles and key headings
- 14px – Section headers and primary actions
- 12px – Body text, metadata, and supporting information

Color Palette (Implemented Theme)

The Admin Panel follows the same dark-mode design system used across the platform to maintain visual consistency and reduce visual fatigue during prolonged administrative tasks.

- **Primary Background:** Deep indigo (#18152A), used to minimize eye strain during long review sessions
- **Card / Surface Background:** Dark surface (#221F3A), used to separate information blocks clearly
- **Accent / Primary Action:** Cyan (#00D5FF), used for interactive elements and navigation cues
- **Primary Text:** White (FFFFFF), ensuring clear readability

- **Secondary Text:** Muted blue-grey (#A5B0D0), used for metadata and contextual details
- **Severity / Critical Actions:** Red (#FF4B6E), reserved strictly for high-risk indicators and critical actions

The controlled use of red ensures urgency is communicated effectively without overwhelming the interface.

3. Admin Panel – Moderation & Support Screens

Screens Covered

- Review Flagged Reports
 - Review Severity Queue
 - Tickets List
 - Ticket Details
-

4. Moderation & Safety Workflow Overview

The Admin Panel is structured around a layered moderation workflow that enables administrators to move from **high-level severity awareness** to **detailed case review** and finally to **resolution or escalation**.

This layered approach ensures:

- Urgent cases are surfaced immediately
 - Context is preserved during reviews
 - Administrative decisions are consistent and well-informed
-

5. Review Flagged Reports

Page Purpose

The Review Flagged Reports screen allows administrators to examine conversations flagged for inappropriate behavior, policy violations, or safety concerns. This screen acts as the first step in the moderation process.

Key Components

- Case summary displaying User ID, Listener ID, and session date

- Reason for flag clearly stated for context
- Severity level badge to indicate urgency
- Previous flags count to identify recurring behavior
- View User and View Listener actions for deeper investigation

Design Considerations

- Information is grouped logically to reduce scanning time
 - Severity indicators provide immediate risk awareness
 - Historical data supports fair and consistent moderation decisions
-

6. Review Severity Queue

Page Purpose

The Review Severity Queue helps administrators prioritize flagged cases based on severity level. This screen ensures high-risk cases receive immediate attention while lower-risk cases remain visible but secondary.

Key Components

- Severity indicators (High priority cases highlighted clearly)
- Case metadata for quick assessment
- Navigation to detailed review screens

Priority Handling

- High-severity cases are surfaced prominently
 - Queue structure prevents neglect of urgent issues
 - Enables structured workload distribution among admins
-

7. Tickets List

Page Purpose

The Tickets List screen displays all user-submitted support tickets, allowing administrators to manage operational and account-related concerns efficiently.

Key Components

- Ticket ID for reference and tracking

- Concern category (Payment, Account, Technical, etc.)
- Priority badge (Low, Medium, High)
- Status indicator showing whether the ticket is open or resolved

Ticket Management Logic

- Priority badges help admins allocate attention effectively
 - Clear categorization improves resolution speed
 - Status indicators prevent duplicate or missed actions
-

8. Ticket Details

Page Purpose

The Ticket Details screen provides complete context for a selected support ticket, enabling administrators to understand the issue thoroughly before taking action.

Key Components

- Ticket metadata including category, priority, and status
- User concern description displayed clearly
- Submission timestamp for reference
- Anonymous submission indicator to preserve user privacy

Contextual Review Benefits

- Reduces back-and-forth communication
 - Enables informed and accurate resolutions
 - Maintains transparency and traceability
-

9. UX Considerations

- Severity-based visual hierarchy supports fast decision-making
- Information density is balanced to avoid overload
- Controlled use of color prevents visual fatigue
- Consistent layout improves long-term usability
- Deliberate action flows reduce the risk of accidental decisions

10. Security & Trust Considerations

- Anonymous identities are preserved throughout all admin views
- Access is role-restricted to prevent misuse
- Sensitive data is presented only when required
- Historical records support accountability and auditing

11. Scalability & Maintainability

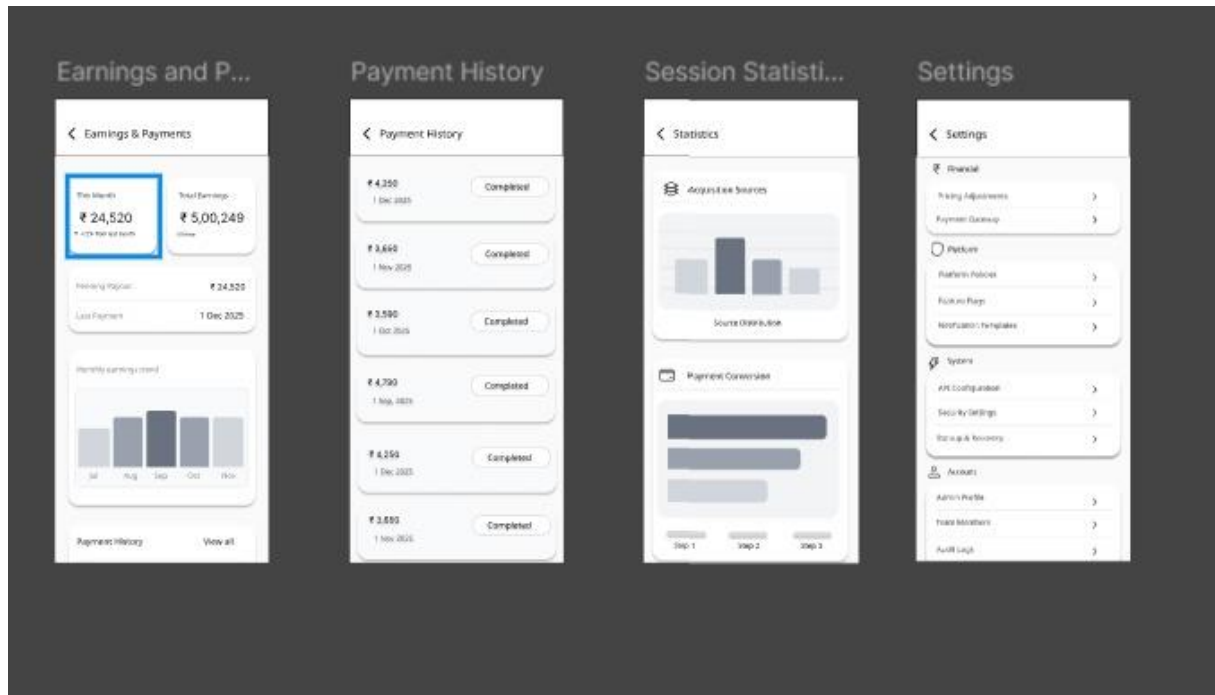
- Modular screen design supports future feature expansion
- Queue-based workflows allow handling increased volume
- Clear separation between moderation and ticketing tasks
- Design supports multi-admin collaboration

12. Conclusion

The Admin Panel functions as the safety, moderation, and operational backbone of the Anonymous Listening Platform. Through structured workflows, severity-based prioritization, and context-rich interfaces, the panel enables administrators to act responsibly, efficiently, and fairly. The design balances urgency with clarity, ensuring platform safety while maintaining trust and transparency at scale.

ADMIN PANEL – UI/UX FUNCTIONAL SCREENS REPORT

Designed by: ASHWIN K



1. Overview

The **Admin Panel** serves as the central **governance, moderation, and operational management interface** of the Anonymous Listening Platform. It is designed to support administrators in managing sensitive user interactions, enforcing platform policies, reviewing safety incidents, and resolving support issues in a structured and accountable manner.

Given the critical responsibility associated with moderation, financial oversight, and safety-related decisions, the Admin Panel emphasizes **clarity, contextual awareness, and controlled workflows**. The interface ensures administrators can review cases comprehensively, prioritize urgency accurately, and take informed actions without unnecessary cognitive overload.

The overall administrative experience focuses on:

- Safety-first and policy-driven decision making
- Clear visibility of critical and contextual information
- Severity- and priority-based case handling
- Efficient management of support tickets and payments
- Maintaining fairness, transparency, and platform trust

2. Design System Reference

Typography

- Font Family: System default for consistency and readability
- 16px – Page titles and key headings
- 14px – Section headers and primary actions
- 12px – Body text, metadata, and supporting information

Color Palette (Implemented Theme)

The Admin Panel follows the same dark-mode design system used across the platform to maintain visual consistency and reduce visual fatigue during prolonged administrative tasks.

- **Primary Background:** Deep indigo (#18152A), used to minimize eye strain during long review sessions
- **Card / Surface Background:** Dark surface (#221F3A), used to separate information blocks clearly
- **Accent / Primary Action:** Cyan (#00D5FF), used for interactive elements and navigation cues
- **Primary Text:** White (FFFFFF), ensuring clear readability
- **Secondary Text:** Muted blue-grey (#A5B0D0), used for metadata and contextual details
- **Severity / Critical Actions:** Red (#FF4B6E), reserved strictly for high-risk indicators and critical actions

The controlled use of red ensures urgency is communicated effectively without overwhelming the interface

3. Admin Panel – Moderation & Support Screens

Screens Covered

- Review Flagged Reports
- Review Severity Queue
- Tickets List
- Ticket Details
- Earnings & Payment
- Session Statistics
- Settings

4. Moderation & Safety Workflow Overview

The Admin Panel is structured around a layered moderation workflow that enables administrators to move from **high-level severity awareness** to **detailed case review** and finally to **resolution or escalation**.

This layered approach ensures:

- Urgent cases are surfaced immediately
- Context is preserved during reviews
- Administrative decisions are consistent and well-informed

5. Review Flagged Reports

Page Purpose

The Review Flagged Reports screen allows administrators to examine conversations flagged for inappropriate behavior, policy violations, or safety concerns. This screen acts as the first step in the moderation process.

Key Components

- Case summary displaying User ID, Listener ID, and session date
- Clearly stated reason for flagging
- Severity level badge indicating urgency
- Previous flags count to identify recurring behavior
- **View User** and **View Listener** actions for deeper investigation

Design Consideration

- Information grouped logically to reduce scanning time
- Severity indicators provide immediate risk awareness
- Historical data supports fair and consistent moderation decisions

6. Review Severity Queue

Page Purpose

The Review Severity Queue helps administrators prioritize flagged cases based on severity level. This screen ensures high-risk cases receive immediate attention while lower-risk cases remain visible but secondary.

Key Components

- Severity indicators (High priority cases highlighted clearly)
- Case metadata for quick assessment
- Navigation to detailed review screens

Priority Handling

- High-severity cases are surfaced prominently
 - Queue structure prevents neglect of urgent issues
 - Enables structured workload distribution among admins
-

7. Tickets List

Page Purpose

The Tickets List screen displays all user-submitted support tickets, allowing administrators to manage operational and account-related concerns efficiently.

Key Components

- Ticket ID for reference and tracking
- Concern category (Payment, Account, Technical, etc.)
- Priority badge (Low, Medium, High)
- Status indicator showing whether the ticket is open or resolved

Ticket Management Logic

- Priority badges help admins allocate attention effectively
 - Clear categorization improves resolution speed
 - Status indicators prevent duplicate or missed actions
-

8. Ticket Details

Page Purpose

The Ticket Details screen provides complete context for a selected support ticket, enabling administrators to understand the issue thoroughly before taking action.

Key Components

- Ticket metadata including category, priority, and status
- User concern description displayed clearly
- Submission timestamp for reference
- Anonymous submission indicator to preserve user privacy

Contextual Review Benefits

- Reduces back-and-forth communication
 - Enables informed and accurate resolutions
 - Maintains transparency and traceability
-

9. Earnings, Payment & Statistics

In line with the implemented screens shown in the image and PDF, the Admin Panel includes **financial and performance visibility modules**.

Functional Coverage

- Earnings overview with summary cards
- Payment history with completed and pending statuses
- Session statistics and conversion insights
- Clear card-based layouts for fast interpretation

These views support **financial transparency, auditing, and operational monitoring**.

10. Earnings, Payment & Statistics

In line with the implemented screens shown in the image and PDF, the Admin Panel includes **financial and performance visibility modules**.

Functional Coverage

- Earnings overview with summary cards
- Payment history with completed and pending statuses
- Session statistics and conversion insights
- Clear card-based layouts for fast interpretation

These views support **financial transparency, auditing, and operational monitoring**.

11. UX Considerations

- Severity-based visual hierarchy supports fast decision-making
 - Information density is balanced to avoid overload
 - Controlled use of color prevents visual fatigue
 - Consistent layout improves long-term usability
 - Deliberate action flows reduce the risk of accidental decisions
-

12. Security & Trust Considerations

- Anonymous identities are preserved throughout all admin views
 - Access is role-restricted to prevent misuse
 - Sensitive data is presented only when required
 - Historical records support accountability and auditing
-

13. Scalability & Maintainability

- Modular screen design supports future feature expansion
 - Queue-based workflows allow handling increased volume
 - Clear separation between moderation and ticketing tasks
 - Design supports multi-admin collaboration
-

14. Conclusion

The Admin Panel functions as the safety, moderation, and operational backbone of the Anonymous Listening Platform. Through structured workflows, severity-based prioritization, and context-rich interfaces, the panel enables administrators to act responsibly, efficiently, and fairly. The design balances urgency with clarity, ensuring platform safety while maintaining trust and transparency at scale.