Listener panel – Front-end Documentation

By: - Arjun (Front-end Lead)

Overview: The combination of the Dashboard Screen and Withdraw Funds Screen creates a complete "Financial Loop" for the service provider. This module addresses the two most critical questions for a gig-worker on the platform: *"**How am I performing**?"* and *"**How do I get paid**?"*

1. 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:** Lucide React Native / Expo Vector Icons
- **Gradient Engine:** expo-linear-gradient

2. Design System & UI/UX

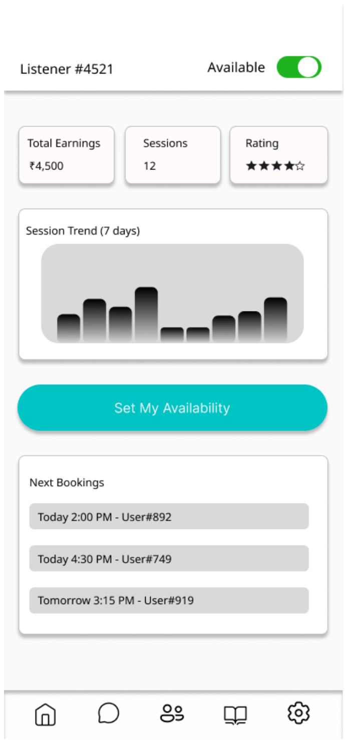
The application follows a strict "Neon/Dark" design language to ensure visual distinctiveness and reduce eye strain in low-light environments.

2.1 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. **Listener Dashboard and Splash Screen**


Listener #4521

Available

Total Earnings ₹4,500

Sessions 12

Rating ★★★★☆

Session Trend (7 days)

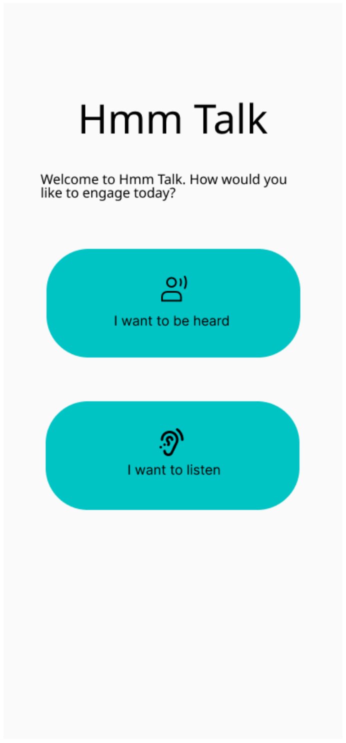
Set My Availability

Next Bookings

Today 2:00 PM - User#892

Today 4:30 PM - User#749

Tomorrow 3:15 PM - User#919


Hmm Talk

Welcome to Hmm Talk. How would you like to engage today?

A. State Management

- **Availability Toggle:** Uses useState(true) to track if the listener is online. This boolean directly controls the styling of the "Available" text (turning it from gray to Neon Cyan) and the toggle switch state.

B. Data Visualization (Custom Charting)

- **Implementation:** Instead of importing a heavy charting library, the component implements a lightweight **CSS-in-JS Bar Chart**.
- **Logic:** The chartData array [30, 50, ...] maps directly to View components where the height style property is dynamically set to the data value.
- **Styling:** Each bar has a shadow (shadowColor: COLORS.accent) to create a "glowing neon tube" effect, matching the app's cyberpunk aesthetic.

C. Navigation

- **Routing:** The component uses router.push() to navigate to deeper detail screens:
    - /home/earnings (clicked via Earnings Card)
    - /home/availability (clicked via "Set My Availability" button)

3. Visual Design System

- **Header:** Features a minimalistic design with the user's ID and the critical "Availability Switch."
- **Stats Grid:** A 3-column layout using Flexbox (justifyContent: 'space-between') to display Earnings, Session Count, and Rating.
- **Booking Cards:** Uses a subtle transparency (rgba(255, 255, 255, 0.05)) to differentiate list items from the background without using heavy borders.

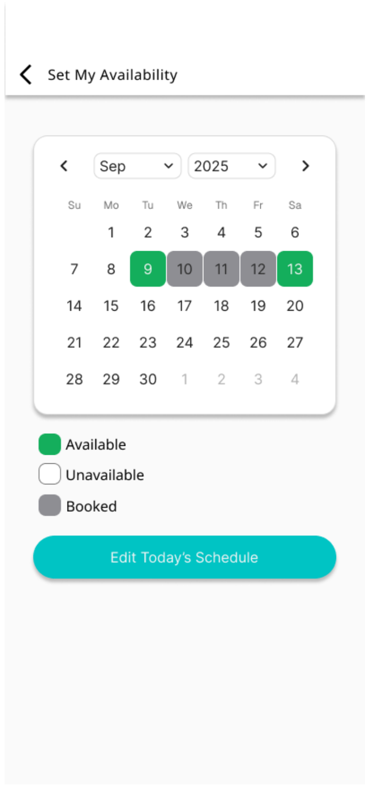
4. Functional Modules

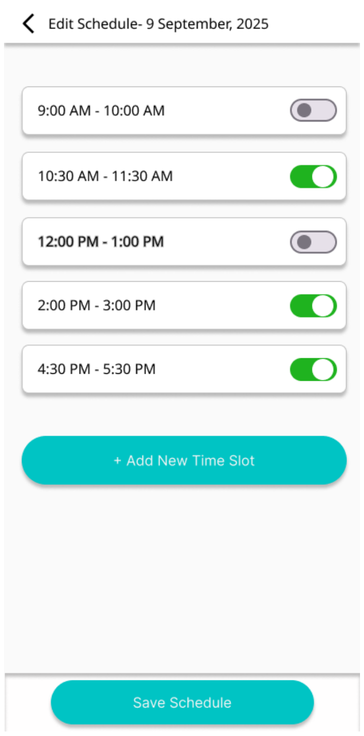
1. **Status Control:** A header switch that instantly broadcasts the user's availability to the network.
2. **Performance Snapshots:** Quick-glance metrics (Earnings, Volume, Quality Rating).
3. **Trend Analysis:** A visual representation of session volume over the last 7 days.
4. **Schedule Management:** A chronological list of upcoming bookings with distinct color coding (COLORS.accent) to highlight the client's ID.

5. Code Quality

- **Performance:** The chart relies on simple Flexbox views rather than SVG or Canvas, ensuring 60fps performance even on low-end devices.
- **Safe Area:** The edges={["top", "left", "right"]} prop ensures the status bar and notch are respected, while allowing the bottom content to flow naturally.

3. Availability and Schedule





A. Custom Grid Engine

Instead of relying on a third-party calendar library, this component implements a lightweight **Custom Grid System**:

- **Math-Based Layout:** It uses Dimensions.get("window") to calculate the exact width of every day cell: (Screen Width - Padding) / 7. This ensures perfect alignment across different device sizes (iPhone vs. Android).
- **Flex Wrapping:** The daysGrid container uses flexDirection: "row" with flexWrap: "wrap". This allows the stream of day objects to automatically break into new rows every 7 items, mimicking a calendar month without complex table logic.

B. Data Simulation & Mapping

- **Data Structure:** The helper function generateCalendarData creates a flat array representing the month. It handles "empty" filler days (at the start of the month) to ensure the 1st of the month aligns with the correct day of the week.
- **State Derivation:** The renderCalendarDay function acts as a view controller. It inspects the status property (available | booked | none) of each data item and conditionally applies styles.

3. Visual Design System

The component extends the "Neon/Dark" theme with a specific color-coded legend:

- **Neon Cyan (#00FFFF):** Represents **"Available"** slots. These are high-contrast to encourage the user to add more availability.
- **Deep Purple (#7B61FF):** Represents **"Booked"** slots. This distinct color (different from the background but less urgent than Cyan) indicates revenue-generating days.
- **Transparent/Bordered:** Represents **"Unavailable"** days. These blend into the glassmorphism background, reducing visual noise.

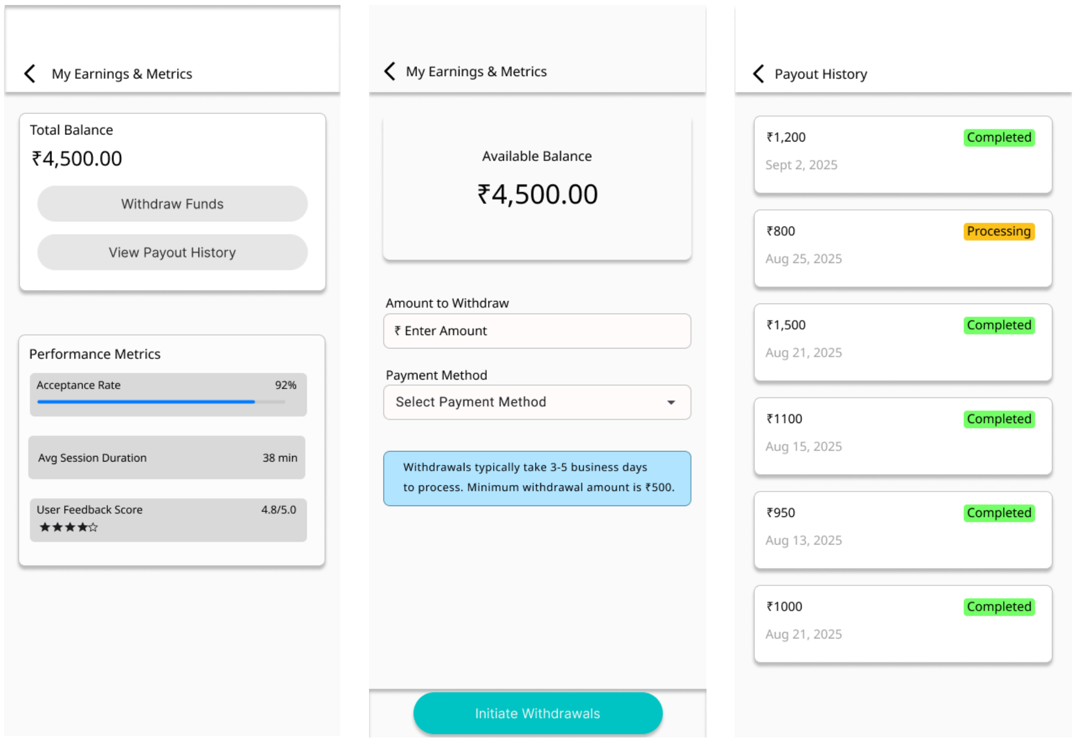
4. Functional Modules

1. **Month Navigator:** A glass-morphic header containing the Month/Year label and arrow controls for navigation (currently UI-only).
2. **The Grid:** The core interactive area. Days are rendered as TouchableOpacity elements, allowing future implementation of "tap-to-toggle" functionality.
3. **The Legend:** A static instructional section that explains the color coding system to the user.
4. **Quick Action:** A prominent floating button ("Edit Today's Schedule") allows the user to bypass the calendar view and jump straight to granular hourly management.

5. Code Quality Assessment

- **Maintainability:** The separation of generateCalendarData from the render logic is good practice. It allows the data source to be swapped for an API response without breaking the UI.
- **Edge Cases:** The code handles empty start slots (id: "empty-1") effectively, preventing the calendar dates from shifting incorrectly.

2. Earnings Withdrawal and Payout History



Withdrawals typically take 3-5 business days to process. Minimum withdrawal amount is ₹500.

₹1,200	Sept 2, 2025	Completed
₹800	Aug 25, 2025	Processing
₹1,500	Aug 21, 2025	Completed
₹1100	Aug 15, 2025	Completed
₹950	Aug 13, 2025	Completed
₹1000	Aug 21, 2025	Completed

A. Navigation & Data Continuity

The architecture implies a seamless transition where context is preserved:

- **Route:** Dashboard $\rightarrow$ router.push("/home/Withdrawals")
- **Visual Continuity:** Both screens feature a dominant "Total Balance" card at the top. This repetition is intentional UX; it reassures the user that they are transacting on the correct account and reminds them of their limit immediately before they type an amount.

B. Custom UI Engines (No External Libraries)

The code demonstrates a "Built-from-Scratch" philosophy to maintain lightweight performance:

- **Progress Bar Engine:** Instead of a heavy package like react-native-progress, the Acceptance Rate metric uses nested View components. The width is calculated dynamically (width: "92%") using inline styles, ensuring zero rendering lag.
- **Star Rating System:** The metrics row manually constructs the 5-star rating using text glyphs (★). This avoids the overhead of vector icon libraries for simple shapes and allows for precise coloring of "half stars" via text opacity/color manipulation.

C. Input State Logic

The WithdrawFundsScreen implements controlled component logic:

- **State:** const [isFocused, setIsFocused]
- **Behavior:** This boolean toggles the border color of the input container. When true, the border switches to COLORS.accent (Neon Cyan) and the background tint changes, providing clear visual feedback in a dark-mode environment.

3. Visual Design System

The "Neon Glass" design language is mature and consistent here:

- **Hierarchy of Action:**
  - **Primary Actions** (Withdraw Funds): Rendered in solid Neon Cyan (#00FFFF) with a drop shadow, demanding attention.
  - **Secondary Actions** (View Payout History): Rendered with a transparent background and glass border, offering functionality without competing for the user's focus.
- **Semantic Color Mapping:**
  - **Cyan:** Used for money, success (high ratings), and active inputs.
  - **White/Grey:** Used for static labels and passive information.
- **Input Masking:** The currency symbol (₹) is rendered as a separate text node *outside* the TextInput but *inside* the container. This "Prefix Masking" prevents the user from accidentally deleting the currency symbol while typing.

4. Functional Modules

1. **The Wallet Card (Shared):** A reusable visual anchor displaying the current financial standing.
2. **Performance Matrix:** A breakdown of *why* the user is earning money. It links behavior (Acceptance Rate, Duration) to results (Balance).
3. **Transaction Form:**
   - **Amount Input:** Numeric keyboard trigger.
   - **Method Selector:** A "Dummy" dropdown UI element ready for modal integration.

    - **Advisory Banner:** A dedicated InfoBanner component using a distinct background color (rgba(0, 255, 255, 0.05)) to separate legal/process information from the interactive form fields.

5. Code Quality & Scalability

- **Responsive Layout:** The ScrollView + SafeAreaView combination ensures the form is accessible on all device sizes, preventing the keyboard from covering the "Withdraw" button on smaller screens.
- **Modular Styling:** The styles are decoupled effectively. For instance, the progressBarBackground and progressBarFill classes are generic enough to be extracted into a separate <ProgressBar /> component later.

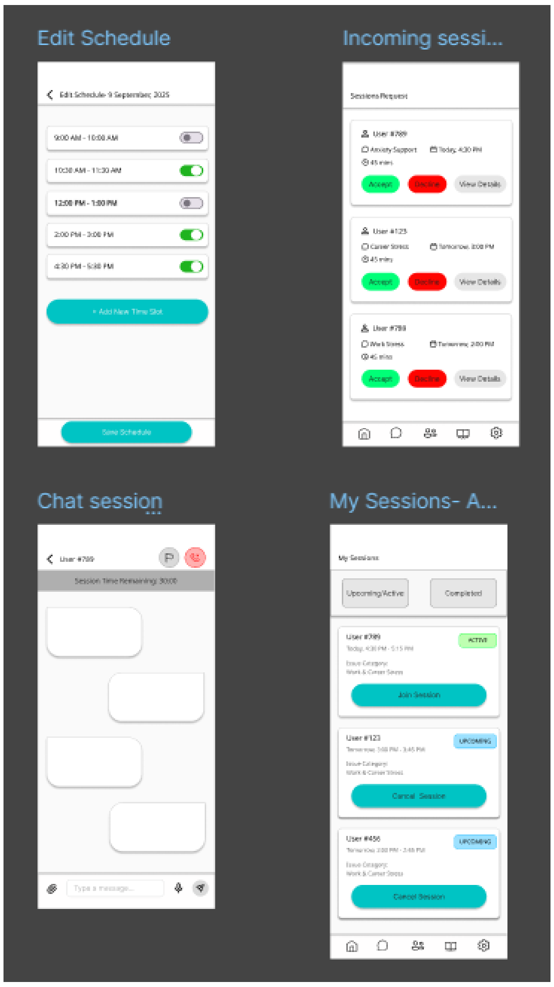
This module constitutes the **"Listener Economy"** engine of the application. It addresses the service provider's primary extrinsic motivation: financial tracking and retrieval.

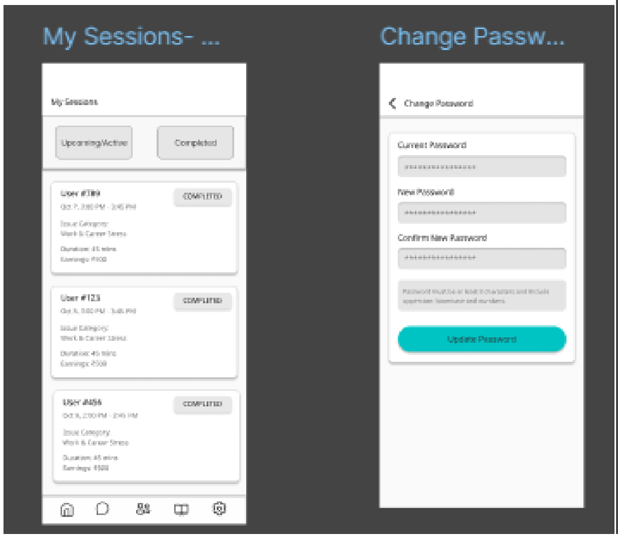
- **The Dashboard** acts as the *Monitor*, visualizing the correlation between performance metrics (quality) and total balance (reward).
- **The Withdrawal Screen** acts as the *Utility*, providing a secure, friction-free interface for converting digital credits into real-world currency.

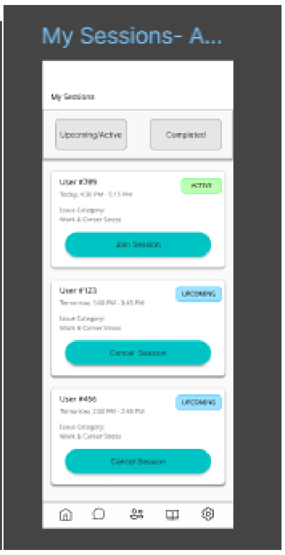
The two screens function as a cohesive unit: the Dashboard creates the intent (showing the balance), and the Withdrawal screen executes the action.

LISTENER PANEL – UI/UX FUNCTIONAL SCREENS REPORT

Designed by: Vijay T S







1. Overview

The Listener Panel is a role-specific interface designed to help listeners manage availability, respond to session requests, conduct live sessions, and track session history efficiently. The design prioritizes focus, clarity, and quick decision-making.

---

2. Design System Reference

Color Palette (Implemented Theme)

The Listener Panel follows the same dark-mode design system for consistency.

- Primary Background: #18152A
- Card Background: #221F3A
- Accent Color: #00D5FF
- Primary Text: #FFFFFF
- Secondary Text: #A5B0D0
- Danger Actions: #FF4B6E

---

3. Listener Panel – Session Management Screens

Screens Covered

- Edit Schedule
- Incoming Session Request
- Active Session (Chat)
- My Sessions – Active
- My Sessions – Completed
- Change Password

3.1 Edit Schedule

Page Purpose
Allows listeners to manage availability by enabling or disabling time slots.

Key Components

- Date header with back navigation

- Time-slot toggle switches
- Add New Time Slot button
- Save Schedule CTA

3.2 Incoming Session Request

Page Purpose
Enables listeners to review and respond to incoming session requests.

Key Components

- Request details (User ID, issue category, duration)
- Accept and Reject buttons
- Available time slots summary

3.3 Active Session (Chat Screen)

Page Purpose
Provides a real-time anonymous chat interface during an active session.

Key Components

- Session header with remaining time
- Message bubbles
- Input field with send and attachment options

3.4 My Sessions – Active

Page Purpose
Displays upcoming and ongoing sessions.

Key Components

- Upcoming / Active tabs
- Session cards with status badges
- Join Session and Cancel Session actions

3.5 My Sessions – Completed

Page Purpose
Provides a record of completed sessions for transparency.

Key Components

- Completed session cards
- Session duration and earnings

3.6 Change Password

Page Purpose
Allows listeners to securely update their account credentials.

---

4. UX Considerations

- Clear action prioritization
- Minimal distractions during live sessions
- Consistent component behavior
- Secure account management

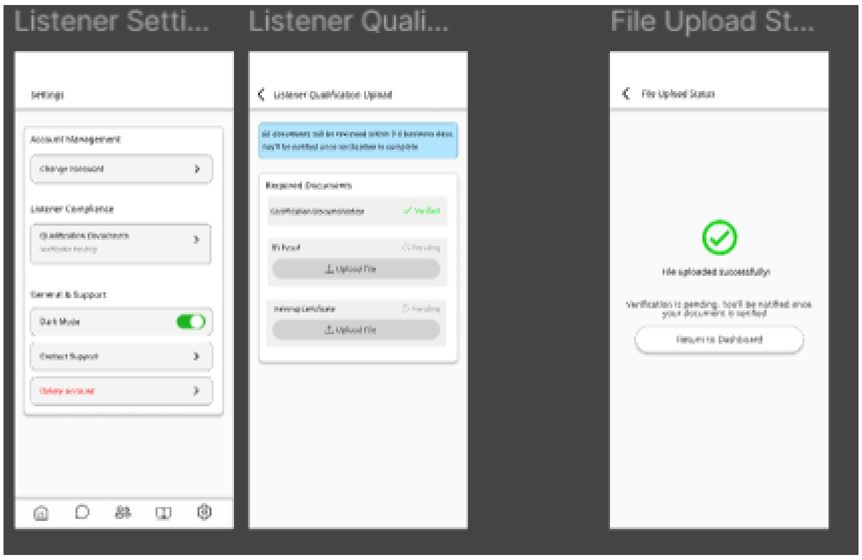
---

5. Conclusion

The Listener Panel offers a focused and efficient workspace that supports listeners in delivering high-quality support while maintaining clarity and security.

LISTENER PANEL – UI/UX FUNCTIONAL SCREENS REPORT

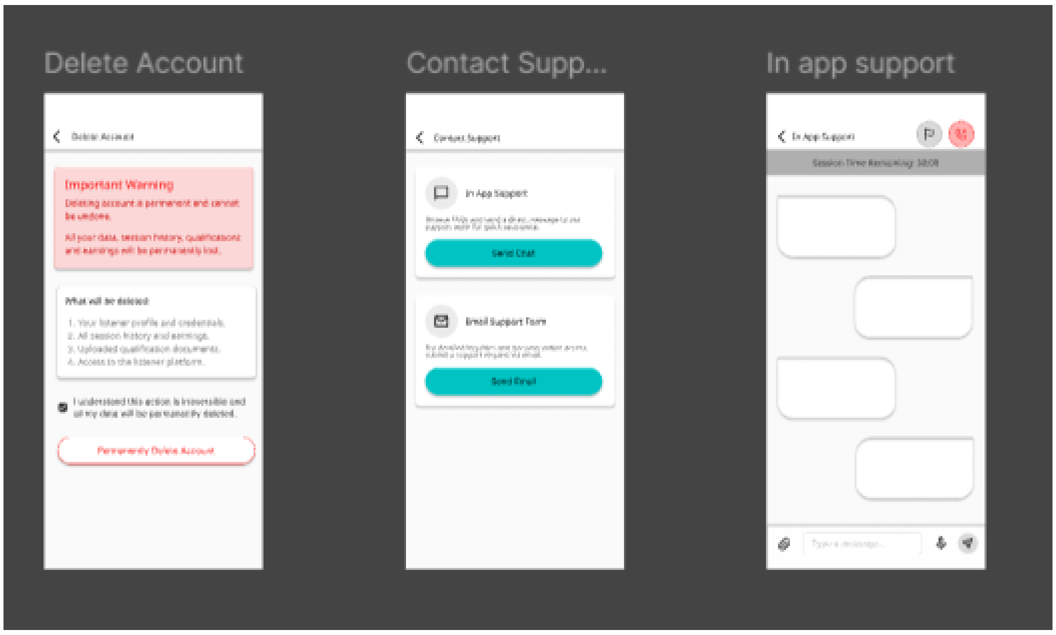
Designed by: Ashwin K



All documents will be reviewed within 2-3 business days. You'll be notified once verification is complete.

File uploaded successfully!

Verification is pending. You'll be notified once your document is verified.



Deleting account is permanent and cannot be undone.

All your data, session history, qualifications and earnings will be permanently lost.

What will be deleted:

1. Your listener profile and credentials.
2. All session history and earnings.
3. Uploaded qualification documents.
4. Access to the listener platform.

[✓] I understand this action is irreversible and all my data will be permanently deleted.

[illegible]

[illegible]

1. Overview

The **Listener Panel** is a role-specific interface designed to enable listeners to manage availability, accept and conduct sessions, track earnings, and review performance metrics efficiently. The panel supports listeners throughout their lifecycle—from login and availability management to live sessions and payout processing—while maintaining anonymity and operational clarity.

The design prioritizes:

- Focused workflows for real-time interactions
- Clear visibility of session and earnings information
- Quick decision-making during incoming requests
- Secure account and credential management

---

2. Design System Reference

Color Palette (Implemented Theme)

The Listener Panel follows the same dark-mode design system for consistency.

- Primary Background: #18152A
- Card Background: #221F3A
- Accent Color: #00D5FF
- Primary Text: #FFFFFF
- Secondary Text: #A5B0D0
- Danger Actions: #FF4B6E

---

3. Listener Panel – Session Management Screens

- Listener Login
- Listener Dashboard
- File Upload Status
- Listener Qualification Upload
- Delete Account Screen – Special Handling
- Listener Settings & Account Management
- In-App Support Chat
- Contact Support

3.1 Listener Login
Page Purpose

Allows listeners to securely access the platform using their unique Listener ID and password.

Key Components

- Listener ID input field
- Password input field
- Login CTA
- Forgot ID / Password option

3.2 Listener Dashboard
Page Purpose

Serves as the primary operational overview for listeners.

Key Components

- Listener status toggle (Available / Unavailable)
- Summary cards:
    - Total Earnings
    - Sessions Count
    - Listener Rating
- Session trend chart (last 7 days)
- Next bookings list
- Set Availability CTA

3.3 File Upload Status

Page Purpose

Confirms successful document upload and informs the listener about the verification process status.

This screen provides immediate reassurance after file submission and guides the listener back to the main workflow.

Key Components

- Success indicator (green check icon)
- Confirmation message:
    - *"File uploaded successfully!"*
- Verification status message:
    - *"Verification is pending. You'll be notified once your document is verified."*
- **Return to Dashboard** CTA to resume normal listener activities

---

3.4 Listener Qualification Upload

Page Purpose

Enables listeners to upload and manage mandatory qualification documents required for verification and platform eligibility.

This screen ensures that all required credentials are collected in a structured manner and clearly communicates verification timelines and status.

Key Components

- Informational banner indicating review timeline
    - *“All documents will be reviewed within 3–5 business days. You’ll be notified once verification is complete.”*
- **Required Documents** section containing:
    - **Certification Documentation**
        - Status indicator: **Verified** (green checkmark)
    - **ID Proof**
        - Status: **Pending**
        - Upload File action
    - **Training Certificate**
        - Status: **Pending**
        - Upload File action
- Each document row includes:
    - Document name
    - Current verification status (Verified / Pending)
    - Upload File CTA (disabled once verified, if applicable)

3.5 Delete Account Screen – Special Handling

Purpose:
Allows listeners to permanently delete their account after acknowledging consequences.

Key Components

- Important warning message
- List of data and access that will be deleted
- Confirmation checkbox
- **Permanently Delete Account** CTA

3.6 Listener Settings & Account Management

Page Purpose

Provides access to account controls, compliance requirements, and support options.

Key Components

Account Management

- Change Password
- Listener Qualification Upload
- File Upload Status

Care & Support

- Contact Support
- In-App Support Chat

Privacy & Security

- Delete Account (with irreversible action warning)

3.7 In-App Support Chat

Purpose:
Allows listeners to interact with support agents in real time.

Key Components

- Chat header with session time indicator
- Message thread
- Text input field
- Send action

3.8 Contact Support

Purpose:
Provides formal support escalation options.

Key Components

- App Help Support
- Email Support Form
- Clear CTA buttons

---

4. UX Considerations

- Clear action prioritization
- Minimal distractions during live sessions
- Consistent component behavior
- Secure account management

---

5. Conclusion

The Listener Panel offers a focused and efficient workspace that supports listeners in delivering high-quality support while maintaining clarity and security.