# **EarnVerseTable of Contents**

1. **Introduction**
   * 1.1. Document Purpose & Audience
   * 1.2. Project Vision & Value Proposition
   * 1.3. Target Audience & User Personas
   * 1.4. Key Terminology
2. **User Authentication & Onboarding**
   * 2.1. User Registration / Signup (with Social Login)
   * 2.2. User Login & Session Management
   * 2.3. Forgot Password & Secure Password Reset
   * 2.4. Email Verification & Account Activation
3. **Core Earning Modules**
   * 3.1. Spin the Wheel
   * 3.2. App Install & Survey Offers (Offerwalls)
   * 3.3. Quick-Win Missions & Engagement Streaks
   * 3.4. Premium-Only Offers (Pro Users)
4. **User Growth & Engagement**
   * 4.1. Referrals & Invite Link
5. **Financial & User Management**
   * 5.1. Wallet & Withdrawals
   * 5.2. Earnings Tracker & Analytics
6. **User Account Management**
   * 6.1. Profile & Settings
   * 6.2. Subscription Management
7. **Technical Specifications**
   * 7.1. Detailed Database Schema
   * 7.2. API Endpoint Specification
   * 7.3. Security, Compliance & Fraud Detection

## **1. Introduction**

### **1.1. Document Purpose & Audience**

This Product Requirements Document (PRD) provides a comprehensive specification for the "EarnVerse" application. It details the functional requirements, user flows, and technical considerations necessary for successful implementation. This document is intended to serve as the single source of truth for the project team, ensuring alignment between stakeholders, designers (UI/UX), developers (frontend/backend), and QA testers throughout the project lifecycle, from initial design concepts to final deployment and iteration.

### **1.2. Project Vision & Value Proposition**

**Vision:** To be the most engaging and trusted rewards platform for supplemental income earners in emerging markets.

**Value Proposition:** EarnVerse is a mobile-first web application that bridges the gap between users seeking simple, flexible earning opportunities and companies seeking to engage with a broad audience. Unlike other platforms, EarnVerse focuses on a gamified, user-centric experience with a clear path for progression from a Free User to a "Pro" member, unlocking significantly higher earning potential. Our dual focus ensures users feel rewarded for their time while providing tangible value and reach for our advertising partners.

### **1.3. Target Audience & User Personas**

The primary target audience includes tech-savvy individuals aged 18-35 looking for a simple, engaging way to earn supplemental income.

* **Persona 1: Priya, the College Student**
  + **Demographics:** 20 years old, studying in a tier-2 city.
  + **Goals:** Wants to earn money for daily expenses like mobile recharges and snacks without committing to a part-time job.
  + **Behavior:** Spends several hours a day on her smartphone. Is attracted to gamified experiences and quick rewards. She will be a frequent user of "Spin the Wheel" and "Quick-Win Missions."
* **Persona 2: Amit, the Young Professional**
  + **Demographics:** 28 years old, working an entry-level IT job in a major city.
  + **Goals:** Looking to supplement his income to save for a larger purchase. He is willing to invest a small amount if the return is significant.
  + **Behavior:** Values efficiency and high-return tasks. He is the ideal candidate for the "Pro User" tier, focusing on "Premium-Only Offers" and high-paying surveys.

### **1.4. Key Terminology**

* **Free/Pro User:** The two tiers of user accounts.
* **Task/Offer:** An action a user can complete to earn a reward.
* **Offerwall:** A third-party, in-app advertising unit that provides multiple offers for users to complete.
* **Postback:** A server-to-server communication method used by offerwall providers to notify EarnVerse of a successful task completion.
* **Clawback:** A reversal of a reward if a user's action is found to be fraudulent or incomplete (e.g., installing an app and uninstalling it immediately).
* **KYC (Know Your Customer):** A verification process, potentially required for very high-value withdrawals to comply with financial regulations.

## **2. User Authentication & Onboarding**

### **2.1. User Registration / Signup**

**Objective:** To offer a multi-faceted, secure, and frictionless method for new users to create an EarnVerse account.

**Visuals & UI Components:**

* **Primary Registration Form:**
  + Fields for Full Name, Email Address, Password, Confirm Password.
  + Real-time inline validation with clear, helpful messages (e.g., "✓ Passwords match", "✗ Email format is invalid").
  + Password strength meter.
  + Mandatory checkbox for Terms of Service and Privacy Policy.
  + Google reCAPTCHA v3 for bot prevention.
* **Social Login (V1.1 Feature):**
  + Prominent "Sign up with Google" and "Sign up with Facebook" buttons.
  + This provides a one-click registration path, significantly reducing friction.

**Core Logic & Process Flow:**

1. **User chooses a method:** Form or Social Login.
2. **Form Submission:**
   * Frontend performs real-time validation.
   * On POST /api/auth/register, the backend re-validates all data.
   * It checks for email uniqueness in the users table.
     + **If exists:** Returns 409 Conflict with a clear error message.
     + **If unique:** Hashes the password (bcrypt), creates a user record (status: 'unverified', role: 'free'), generates a verification token, and dispatches a verification email. The email template will be visually appealing, with a clear call-to-action button.
3. **Social Login Flow:**
   * User authenticates with the provider (Google/Facebook).
   * The provider returns an auth token to the frontend.
   * Frontend sends this token to POST /api/auth/social-login.
   * Backend verifies the token with the provider, retrieves the user's name and email, and creates a new EarnVerse account if one doesn't already exist. The account is created as 'active' since the email is already verified by the social provider.
4. **Redirection:** Form-based signups are redirected to a "Please Verify Your Email" page. Social signups are logged in and redirected directly to the dashboard.

### **2.2. User Login & Session Management**

**Objective:** To provide secure, persistent access to the platform.

**Visuals & UI Components:**

* Standard fields for Email and Password.
* "Forgot Password?" link.
* "Remember Me" checkbox.
* Social login options ("Log in with Google / Facebook").

**Core Logic & Process Flow:**

1. **API Call:** POST /api/auth/login with credentials.
2. **Backend Processing:**
   * Finds user by email.
   * Compares submitted password with the stored hash.
   * **On success:**
     + Checks status. If 'unverified', returns 403 Forbidden with EMAIL\_NOT\_VERIFIED code and a "Resend Verification" option.
     + If 'active', generates a secure JWT (JSON Web Token).
     + **JWT Payload:** { "userId": "...", "role": "pro", "exp": ... }.
     + The JWT is returned to the client to be stored securely (e.g., HttpOnly cookie).
3. **Session Persistence:** If "Remember Me" is checked, a long-lived refresh token is also issued and stored securely. This allows the user to get a new JWT without re-entering their password.

### **2.3. Forgot Password & Secure Password Reset**

**Objective:** To provide a secure, multi-step, self-service account recovery process.

**Process Flow:**

1. **Step 1: Request Reset:**
   * User enters their email on the "Reset Password" page.
   * POST /api/auth/forgot-password.
   * Backend generates a single-use, 15-minute expiry token, stores it in auth\_tokens, and emails a reset link.
   * The system always returns a generic success message to prevent email enumeration.
2. **Step 2: Perform Reset:**
   * User clicks the link, landing on the "Create New Password" page.
   * POST /api/auth/reset-password with the token and new password.
   * Backend validates the token's existence, type, and expiry.
   * **If valid:** Hashes the new password, updates the users table, and invalidates the reset token.
   * **Security Measure:** For enhanced security, all other active sessions for this user are invalidated by changing a salt in the JWT signature.
3. **Confirmation:** The user is automatically logged in and redirected to their dashboard with a success message.

### **2.4. Email Verification & Account Activation**

**Objective:** To confirm email ownership, a critical step for communication and account security.

**Logic & Process Flow:**

1. **Trigger:** User clicks the verification link in their registration email.
2. **API Call:** The link leads to GET /verify-email?token=..., which makes a POST request to /api/auth/verify-email.
3. **Backend Processing:**
   * The backend validates the token.
   * **If invalid:** The user is redirected to an error page explaining the link has expired and offering a button to resend the verification email.
   * **If valid:** The user's status is updated to 'active', and the token is invalidated.
4. **Redirection:** The user is redirected to the Login page with a success message, priming them to log in for the first time.

## **3. Core Earning Modules**

### **3.1. Spin the Wheel**

**Objective:** To provide a gamified daily earning opportunity that encourages user retention.

**Visuals & UI:**

* **Animations & Sound:** The wheel spin will be accompanied by an exciting animation and sound effects. A "win" will trigger a celebratory sound and confetti animation.
* **Bonus Task Logic:** Landing on "Bonus Task" will trigger a modal: "Congratulations! You've unlocked a bonus mission. Complete it now to earn an extra ₹5!". A "Start Mission" button will lead directly to a simple, pre-selected Quick-Win Mission.

**User Logic & Gating:**

* **Frequency:** One spin per 24 hours for Free Users, two for Pro Users.
* **Backend Logic:**
  + POST /api/rewards/spin.
  + The backend uses a weighted probability system to determine the outcome server-side, preventing client-side manipulation. (e.g., "Try Again": 40%, ₹1: 30%, Bonus Task: 15%, ₹5: 10%, ₹10: 5%).
  + The result is logged in earnings\_log and the user's wallet is updated in a single database transaction to ensure consistency.

### **3.2. App Install & Survey Offers (Offerwalls)**

**Objective:** To serve as the primary earning channel through third-party offerwall integrations.

**Visuals & UI:**

* **Task Details Modal:** Clicking a task card opens a detailed modal with:
  + Full instructions and rules.
  + Reward amount and estimated completion time.
  + Clear disclosure of any potential "clawback" conditions.
* **Offerwall Provider Branding:** The source of the offer (e.g., "from Tapjoy") will be displayed.

**User Logic & Gating:**

* **Free Users:** Access to tasks up to ₹30.
* **Pro Users:** Exclusive access to tasks from ₹50 to ₹200.
* **Task Completion Flow (Postback System):**
  1. User clicks an offer, and is redirected via a tracked affiliate link.
  2. The third-party provider tracks the user's progress.
  3. Upon completion, the provider sends a server-to-server postback to a dedicated endpoint: POST /api/webhooks/offer-completion.
  4. The EarnVerse backend validates the postback's authenticity (using a shared secret or hash), checks for duplicate transaction IDs, and credits the user's **pending balance**.
  5. Earnings move from "pending" to "available" after a confirmation period (e.g., 24 hours) to account for potential fraud or clawbacks.

### **3.3. Quick-Win Missions & Engagement Streaks**

**Objective:** To offer simple, in-house tasks for instant gratification and to build daily habits.

**Visuals & UI:**

* **Checklist Interface:** With satisfying animations and sound effects on completion.
* **Engagement Streak:** A dedicated UI element shows the user's current daily streak (e.g., "🔥 5-Day Streak!").

**User Logic & Gating:**

* **Daily Reset:** Missions reset at midnight (user's local time).
* **Free Users:** Access to standard missions (₹1–₹3).
* **Pro Users:** Access to exclusive, higher-paying missions (₹5–₹10).
* **Streak Bonus:** Completing all daily missions for 7 consecutive days unlocks a significant bonus reward (e.g., ₹25 for Free Users, ₹50 for Pro Users), which is credited automatically. The backend runs a daily cron job to check for and award streak bonuses.

### **3.4. Premium-Only Offers**

**Objective:** To create a high-value, exclusive earning section that is the primary driver for Pro upgrades.

**Visuals & UI:**

* **Exclusive Section:** Labeled "Pro Offers" with a distinct, premium look and feel.
* **Upgrade CTA Modal:** If a Free User attempts to access it, a modal appears that clearly lists all benefits of upgrading:
  + "Unlock offers up to ₹500"
  + "Lower withdrawal limit (₹20)"
  + "Double daily spins"
  + "Higher referral bonuses"

**User Logic & Gating:**

* **100% Gated:** The API endpoint GET /api/tasks/pro will return a 403 Forbidden error if the user's JWT role is not 'pro'.
* **KYC for High-Value Offers:** For offers exceeding a certain threshold (e.g., ₹300), a one-time, simplified KYC process (e.g., uploading an ID) may be required to prevent large-scale fraud, handled via a secure third-party service.

## **4. User Growth & Engagement**

### **4.1. Referrals & Invite Link**

**Objective:** To leverage the user base for organic growth through a structured referral program.

**Visuals & UI:**

* **Sharing Options:** Native share sheet integration, plus direct buttons for WhatsApp, Telegram, and Facebook.
* **Referral Dashboard:** An enhanced dashboard with a timeline view of referral activity and clear status labels: "Signed Up," "Email Verified," "First Task Completed," "Reward Credited."

**User Logic & Gating:**

* **Successful Referral Definition:** A referral is successful when the new user signs up, verifies their email, and completes their first non-spin/non-bonus task (i.e., an offerwall or quick-win mission) worth at least ₹1. This ensures genuine engagement.
* **Free Users:** Earn **₹5** per successful referral.
* **Pro Users:** Earn **₹10** per successful referral.
* **Anti-Fraud:** The backend will implement checks for duplicate IP addresses, device fingerprints, and referral patterns from a single user to prevent self-referral and farming. Suspicious referrals will be flagged for manual review.

## **5. Financial & User Management**

### **5.1. Wallet & Withdrawals**

**Objective:** To provide a transparent, reliable system for managing earnings and payouts.

**Visuals & UI:**

* **Withdrawal Form:** With real-time validation of the amount entered and the UPI ID/PayPal email format.
* **History Table Statuses:**
  + Pending: Request received.
  + In-Process: Sent to the payment processor.
  + Completed: Funds successfully transferred.
  + Failed: Transfer failed (e.g., invalid details). The amount is returned to the user's wallet.
  + Refunded: A clawback occurred after payout.

**User Logic & Gating:**

* **Minimums:** ₹50 for Free Users, ₹20 for Pro Users.
* **Withdrawal Failure:** If a withdrawal fails, an automated email is sent to the user explaining the likely reason and prompting them to check their payment details.

### **5.2. Earnings Tracker & Analytics**

**Objective:** To provide users with powerful insights into their earning habits.

**Visuals & UI:**

* **Interactive Charts:** A line chart for daily earnings and a pie chart showing the breakdown of earnings by source (Tasks, Spins, Referrals, Bonuses).
* **Date Range Filters:** "Last 7 Days," "Last 30 Days," "This Month," "All Time."

**Future Integration: Personalized Income Roadmap (Gemini API)**

* **Pro Feature:** This will be an exclusive feature for Pro Users.
* **Prompt Engineering:** The prompt to the Gemini API will be structured and anonymized: {"user\_role": "pro", "avg\_daily\_earnings": 15, "preferred\_task\_types": ["surveys", "app\_installs"], "completion\_rate": 0.8, "target\_goal": 1000, "timeframe": "month"}.
* **Actionable Output:** The API will return a structured JSON object that the frontend can render into a personalized, step-by-step plan, creating significant value and a "premium" feel.

## **6. User Account Management**

### **6.1. Profile & Settings**

**Objective:** To give users full control over their account details and preferences.

**Visuals & UI:**

* **Change Email Process:** This will be a secure flow. Clicking "Change Email" will require the user to enter their password, then a verification link will be sent to the *new* email address. The change only takes effect after the new link is clicked.
* **Delete Account Process:** This will be a two-step process. After clicking "Delete Account," a modal will appear requiring the user to type "DELETE" to confirm, explaining that all data and earnings will be permanently lost.

## **7. Technical Specifications**

### **7.1. Detailed Database Schema**

* **users**: userId, fullName, email, passwordHash, role, status, referralCode, referredBy, lastLoginAt, createdAt, updatedAt.
* **user\_profiles**: userId, country, dob, kycStatus.
* **wallet**: walletId, userId, balance, pendingBalance, lifetimeEarnings.
* **earnings\_log**: logId, userId, amount, sourceType, sourceId, description, timestamp.
* **withdrawals**: withdrawalId, userId, amount, method, destination, status, processorTransactionId, createdAt, processedAt.
* **auth\_tokens**: tokenId, userId, tokenHash, type, expiresAt.
* **notifications**: notificationId, userId, title, body, isRead, createdAt.

### **7.2. API Endpoint Specification**

(In addition to the previous list)

* POST /api/auth/social-login
* POST /api/missions/complete
* GET /api/streaks/status
* POST /api/webhooks/offer-completion
* POST /api/webhooks/payment-status
* POST /api/profile/change-email-request
* POST /api/profile/change-email-confirm
* DELETE /api/profile

### **7.3. Security, Compliance & Fraud Detection**

* **Password Hashing:** Use bcrypt with a sufficient cost factor.
* **Session Management:** Use JWTs with short expiry (15 mins) and secure refresh tokens (7 days).
* **Input Validation & ORM:** Use a trusted ORM to prevent SQL injection and validate all inputs.
* **Rate Limiting:** Apply strict rate limiting on auth endpoints (e.g., 5 login attempts/min, 1 password reset request/5 mins).
* **Fraud Detection:** Implement a rule-based system to flag suspicious activity:
  + Multiple accounts from a single IP address.
  + Anomalously high task completion rates.
  + Use of VPNs or proxies.
  + Suspicious referral patterns.
* **Compliance:** Adhere to data privacy regulations by offering clear data export and deletion options. Implement KYC for withdrawals over a certain threshold to comply with AML (Anti-Money Laundering) regulations.