Detailed Requirements Document for Driver Log & Payment App

## 1. Introduction

This document outlines the requirements for a mobile application designed to manage driver logs (odometer readings, work times) and automate salary and allowance calculations. The primary goal is to replace the current manual process involving WhatsApp messages and Google Sheets with a more efficient, accurate, and automated system.

## 2. Scope

The application will include:

- Driver-facing functionality for inputting daily log data.
- Admin-facing functionality for viewing logs, managing settings, and generating reports.
- Automated calculation of regular hours, overtime, and fuel allowances.
- Generation of monthly payment summaries.

## 3. Functional Requirements

### 3.1 Driver Module

- **FR-D-001: Clock-in/Clock-out:** The driver shall be able to "clock in" at the start of their shift and "clock out" at the end of their shift. This action should automatically record the current time and location (optional, for verification).
- **FR-D-002: Odometer Input - Start:** Upon clocking in, the driver shall be prompted to enter the current odometer reading.
- **FR-D-003: Odometer Input - End:** Upon clocking out, the driver shall be prompted to enter the current odometer reading.
- **FR-D-004: Multiple Shifts per Day:** The system shall allow a driver to clock in and out multiple times within the same day to accommodate split shifts. Each clock-in/out pair will constitute a distinct shift.
- **FR-D-005: View Daily Logs:** The driver shall be able to view their own recorded log entries for the current day and past days.
- **FR-D-006: Data Correction (Limited):** The driver should have a limited window (e.g., 24 hours) to edit their own clock-in/out times or odometer readings, with all changes being logged for auditing.
- **FR-D-007: Offline Capability:** The app should allow drivers to record entries even without an internet connection and sync data once connectivity is restored.

### 3.2 Admin Module

- **FR-A-001: View Driver Logs:** The admin shall be able to view all driver log entries for all drivers, filtered by date range and driver.

- **FR-A-002: Edit Driver Logs:** The admin shall be able to edit any driver's log entry (times, odometer readings). All edits must be logged with timestamp and user.
- **FR-A-003: Calculate Monthly Pay:** The system shall automatically calculate the monthly salary for each driver based on:
  - Base salary (₹27,000/month).
  - Overtime hours (before 8 AM, after 8 PM, or any time on Sunday) at ₹100/hour.
  - Fuel allowance (₹33.30 per day worked).
- **FR-A-004: Generate Payment Summary:** The system shall generate a detailed monthly payment summary for each driver, showing:
  - Total regular hours.
  - Total overtime hours.
  - Total fuel allowance.
  - Total gross pay.
  - Breakdown of calculations.
- **FR-A-005: Export Data:** The admin shall be able to export log data and payment summaries (e.g., CSV, PDF).
- **FR-A-006: Manage Driver Accounts:** The admin shall be able to add, modify, or deactivate driver accounts.
- **FR-A-007: Configure Rules:** The admin shall be able to configure core parameters (e.g., standard work hours, overtime rate, fuel allowance, monthly salary).

## 4. Non-Functional Requirements

- **NFR-S-001: Security:**
  - User authentication (driver login, admin login).
  - Data encryption (in transit and at rest).
  - Access control based on user roles (driver vs. admin).
- **NFR-P-001: Performance:**
  - App response times should be minimal (e.g., < 2 seconds for common operations).
  - Scalable to accommodate multiple drivers and years of data.
- **NFR-U-001: Usability:**
  - Intuitive and easy-to-use interface for both drivers and admins.
  - Clear feedback messages for user actions.
- **NFR-R-001: Reliability:**
  - High availability and uptime.
  - Data integrity and consistency.
- **NFR-M-001: Maintainability:**
  - Well-documented code.
  - Modular architecture to facilitate future enhancements.
- **NFR-C-001: Compatibility:**
  - Mobile app compatible with common Android and iOS versions.
  - Admin web interface compatible with modern web browsers.
- **NFR-A-001: Auditability:** All data modifications (especially log entries) must be logged with user, timestamp, and old/new values.

**5. Technical Requirements**

- **TR-T-001: Technology Stack:** To be determined, but should support rapid development and deployment (e.g., React Native for mobile, Node.js/Python for backend, PostgreSQL/MongoDB for database).
- **TR-API-001: API:** A robust API will be developed for communication between the mobile app and the backend, and for potential third-party integrations.
- **TR-DB-001: Database:** A relational database (e.g., PostgreSQL) is preferred for structured data and transactional integrity.
- **TR-C-001: Cloud Platform:** A cloud platform (e.g., AWS, GCP, Azure) will be used for hosting the backend and database.

-----User Stories

**As a Driver...**

1. **US-D-001: Clock In**
    - I want to easily record my start time and starting odometer reading when I begin work
    - So that my work hours can be accurately tracked.
    - *Acceptance Criteria:*
        - Can enter odometer reading.
        - System automatically records current timestamp.
        - Confirms successful clock-in.
        - Allows multiple clock-ins per day for separate shifts.
2. **US-D-002: Clock Out**
    - I want to easily record my end time and ending odometer reading when I finish work
    - So that my work hours and distance driven can be accurately tracked.
    - *Acceptance Criteria:*
        - Can enter odometer reading.
        - System automatically records current timestamp.
        - Confirms successful clock-out.
3. **US-D-003: View My Daily Logs**
    - I want to see my recorded start/end times and odometer readings for the current day
    - So I can verify my entries.
    - *Acceptance Criteria:*
        - Displays a list of my shifts for the selected day.
        - Shows start time, end time, start odometer, end odometer, and calculated duration for each shift.
4. **US-D-004: Correct a Recent Entry**
    - I want to be able to correct a mistake in my clock-in/out or odometer reading shortly after I've made it (e.g., within 24 hours)
    - So that my records are accurate without needing admin intervention.

    - *Acceptance Criteria:*
        - Option to edit recent entries.
        - Changes are clearly marked as edited.
        - Admin is notified of the edit. (Optional, but good for audit)

**As an Admin (or AVR Venkatesa)...**

1. **US-A-001: View All Driver Logs**
    - I want to view all driver log entries for specific dates or date ranges
    - So I can oversee driver activity and prepare for payroll.
    - *Acceptance Criteria:*
        - Can filter logs by driver and date range.
        - Displays start time, end time, odometer readings, and calculated shift duration for each entry.
2. **US-A-002: Edit Driver Logs**
    - I want to be able to correct any driver's log entry (time or odometer)
    - So that I can fix errors or adjust records as needed.
    - *Acceptance Criteria:*
        - Ability to modify any field in a driver's log.
        - All modifications are logged (who, what, when).
3. **US-A-003: Calculate Monthly Payroll**
    - I want the system to automatically calculate each driver's monthly salary, including overtime and allowances
    - So I don't have to manually calculate it, reducing errors and saving time.
    - *Acceptance Criteria:*
        - Correctly calculates regular pay (based on ₹27,000/month).
        -