

Etaxi Booking Management System



Project Submitted by
Mr. Sumeet Panigrahy

A
PROJECT REPORT ON
Etaxi

E-taxi Booking System



Submitted to the Department of Computer Science
As a part fulfilment of the requirements for the
Degree of

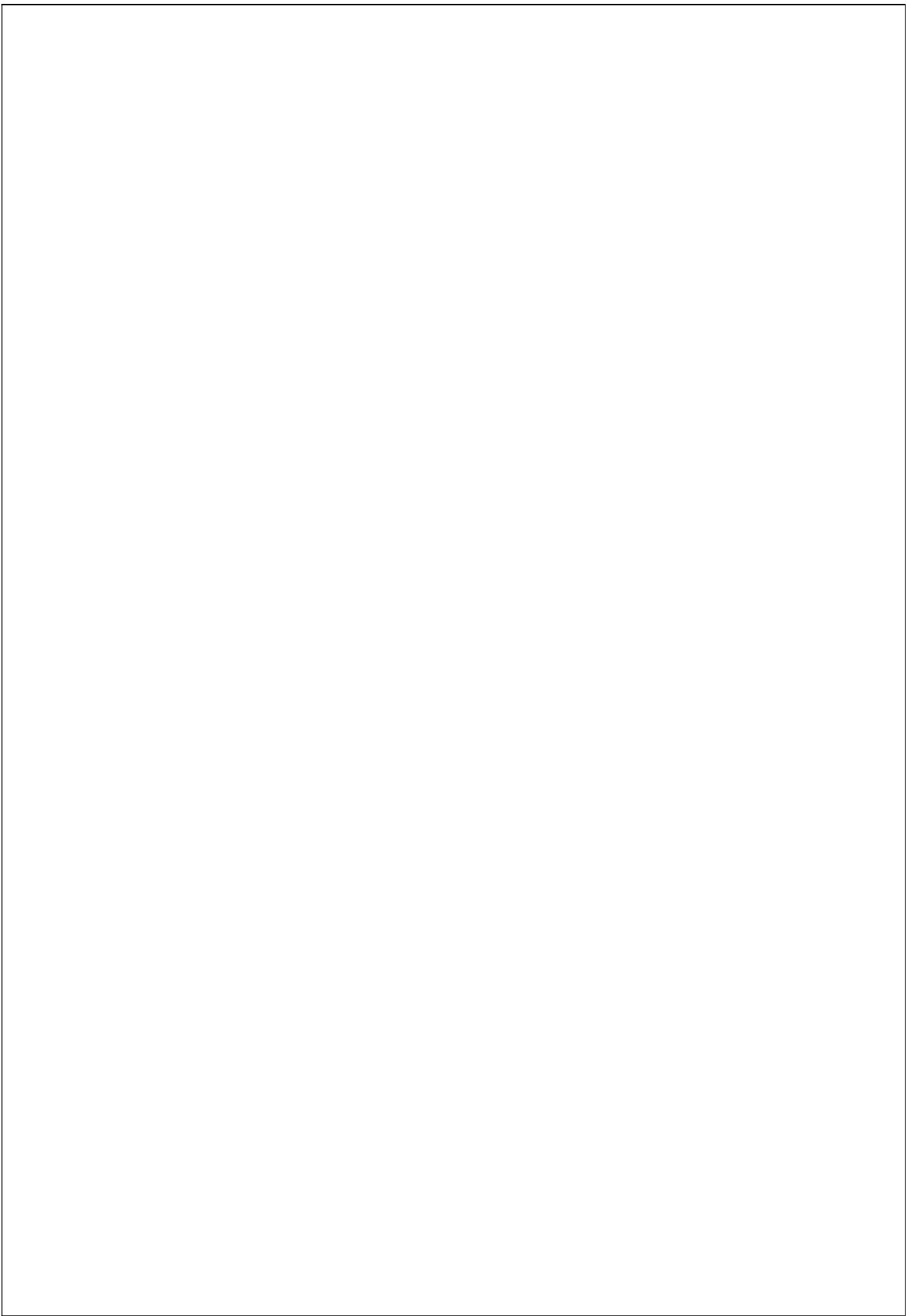
BACHELOR IN COMPUTER SCIENCE

SUBMITTED BY
Mr. Sumeet Panigrahy
ROLL NO: GCMH002623
REGD NO: 47274/2022

Guided by
Mr. G. Jogindra
Faculty, Dept. of Computer Science

**GAYATRI INSTITUTE OF COMPUTER AND
MANAGEMENT, HINJILICUT**

Index



E-taxi Booking System

It's 8 AM, and Sarah is running late for an important meeting. She rushes to the nearest street corner, hoping to find a taxi — but every cab that passes is either occupied or refuses to stop. After 15 frustrating minutes, she finally gets a ride, but the driver quotes an inflated price. With no other options, she accepts, knowing she's already late. Meanwhile, on the other side of town, Raj, a taxi driver, has been driving around for over an hour without finding a passenger. He's low on fuel and worried about making enough money to cover his daily expenses. This is a daily reality for many customers and drivers in a manual taxi system. Customers face difficulties finding taxis, experience inconsistent pricing, and lack safety assurances. Drivers, on the other hand, endure long waiting times, wasted fuel, and unpredictable income due to inefficient ride distribution. The E-Taxi Booking System aims to solve these problems by providing an easy-to-use, digital platform where customers can book rides instantly, track their taxis, and ensure secure payments. Drivers benefit from fair, optimized ride assignments, reducing downtime and improving their earnings. The system ensures a smoother, safer, and more reliable experience for both customers and drivers.

The E-Taxi Booking System is an online platform designed to simplify and streamline the process of booking taxis. Traditional taxi services often pose day-to-day challenges for customers, such as difficulty finding an available taxi, especially during peak hours or in remote areas. Customers may face inconsistent pricing, long waiting times, and unreliable service due to the lack of a structured booking system. Additionally, safety concerns arise when passengers board taxis without knowing driver details or ride routes.

For drivers, the manual system often leads to low earnings and inefficient ride distribution. Drivers may spend hours searching for passengers, leading to wasted fuel, time, and effort. This uneven demand-supply balance impacts their income and job satisfaction. The proposed E-Taxi Booking System addresses these issues by providing a seamless, digital platform where customers can book rides instantly, track their taxi in real-time, and ensure secure, fair transactions. Drivers benefit from optimized ride assignments, reducing idle time and increasing daily earnings. Admins can monitor operations to ensure smooth, reliable, and safer transportation services for all users.

Introduction and Objectives

The transportation industry plays a vital role in urban mobility, with taxi services serving as a convenient, on-demand mode of travel. However, traditional taxi booking methods, reliant on manual hailing or phone-based dispatch systems, present several inefficiencies for both customers and drivers. Customers often face challenges such as difficulty in finding available taxis, unpredictable pricing, and safety concerns due to the lack of verified driver information. Simultaneously, drivers experience reduced earnings due to time wasted searching for passengers and inconsistent ride distribution.

The E-Taxi Booking System addresses these challenges by leveraging modern technology to create a user-friendly, automated booking platform. It integrates features such as GPS-based ride matching, real-time tracking, fare estimation, and secure digital payments to streamline the booking process. The system optimizes ride assignments, minimizing driver idle time while ensuring faster, more reliable service for passengers. Additionally, an admin module oversees the entire system, ensuring operational efficiency, driver management, and data analytics to further enhance performance. This project aims to create a more accessible, efficient, and equitable transportation solution that benefits all stakeholders — customers, drivers, and service administrators alike.

1. Enhance Accessibility and Convenience for Customers:

The primary objective is to ensure customers can book taxis quickly and easily without the hassle of searching for an available ride. The system will feature an intuitive user interface, allowing customers to request rides, track the taxi's arrival in real-time, and receive driver details. This improves convenience and reduces wait times, especially in peak hours or remote areas.

2. Optimize Driver Efficiency and Increase Earnings:

For drivers, the system aims to reduce idle time and fuel wastage by providing smart ride assignments based on proximity, traffic conditions, and availability. This ensures that drivers get more consistent rides, leading to better daily earnings and resource utilization.

3. Ensure Fair and Transparent Pricing:

The system will incorporate an automated fare calculation mechanism that considers distance, time, and potential surcharges — ensuring customers receive fair pricing while drivers are compensated appropriately. This eliminates the negotiation struggles commonly associated with manual taxi booking.

4. Improve Safety and Security:

Safety is a key concern in traditional taxi services. This system will enhance security by providing customers with verified driver details (name, vehicle number, and ratings) before the ride begins. Additionally, real-time ride tracking allows customers to share their location with friends or family, ensuring peace of mind during the journey.

5. Streamline Administrative Operations:

An admin panel will oversee and manage drivers, monitor bookings, handle disputes, and ensure service quality. This backend control ensures the system runs smoothly, supports scalability, and maintains operational efficiency.

6. Implement Data-Driven Insights and Analytics:

The platform will gather data on ride demand, driver performance, peak hours, and more. This information can be analyzed to optimize operations, predict high-demand areas, and improve service delivery. Drivers can also access insights into their earnings and performance, promoting continuous improvement.

7. Promote Eco-Friendly Operations:

By minimizing unnecessary driving and idle time, the system indirectly supports fuel efficiency and reduces carbon emissions. The system could further be enhanced to integrate electric taxis or carpooling options in the future.

8. Support Cashless and Flexible Payments:

To enhance the user experience, the system will integrate secure digital payment gateways, supporting various methods like credit/debit cards, e-wallets, or UPI. This reduces cash dependency and ensures a faster, hassle-free transaction process for both parties.

Project Category

The **E-Taxi Booking System** falls under three key categories: **RDBMS**, **OOPs**, and **Expert System**.

Each category contributes to the system's functionality, performance, and intelligent behavior.

1. RDBMS (Relational Database Management System)

The system relies on a structured database to store and manage interconnected data — such as user profiles, driver details, bookings, payments, and ride history — ensuring data consistency, security, and fast retrieval. An RDBMS supports complex relationships between different entities, making it ideal for a multi-user, data-intensive application like this.

Core functions in the project:

- **Data Storage:** Stores customer, driver, and admin information, along with ride data and payment records.
- **Data Relationships:** Establishes connections between tables, e.g., a booking links to both a customer and a driver.
- **Query Processing:** Retrieves data efficiently using SQL queries (e.g., finding the nearest available driver).
- **Data Integrity:** Ensures rules like unique user IDs, valid bookings, and completed payments are enforced.
- **Backup and Recovery:** Supports data backup to prevent loss in case of system failure.

👉 **Example:** A ride booking links to both the customer's and driver's data through foreign keys, ensuring that only valid users and verified drivers can complete transactions.

2. OOPs (Object-Oriented Programming System)

The project models real-world entities (like customers, drivers, rides, and payments) as objects, each with its attributes (e.g., name, car type, location) and behaviors (e.g., book ride, calculate fare). OOPs principles make the system modular, reusable, and easy to maintain.

Core functions in the project:

- **Encapsulation:** Keeps data secure within objects — e.g., a driver's rating is only accessible through authorized functions.
- **Inheritance:** Allows a generic **User** class to serve both **Customer** and **Driver**, inheriting common features (e.g., login) while having unique ones (e.g., ride booking for customers, availability toggle for drivers).
- **Polymorphism:** Supports dynamic behavior — e.g., different types of rides (standard, premium, shared) can calculate fares differently using a common **Ride** class.
- **Abstraction:** Hides complex processes from the user — e.g., customers only see "estimated fare" without knowing how distance, time, and demand factor in.

👉 **Example:** The **Driver** class inherits from the **User** class and extends its features by adding behaviors like **toggle availability** and **receive booking**. This keeps the code clean and avoids duplication.

3. Expert System

The system simulates human decision-making to automate ride matching, fare calculations, and route suggestions — tasks that typically require a human dispatcher's judgment. By applying predefined rules and dynamic inputs (e.g., location, traffic, availability), the system behaves intelligently.

Core functions in the project:

- **Smart Ride Allocation:** Matches customers to nearby drivers based on GPS coordinates and driver availability.
- **Dynamic Pricing:** Adjusts fares based on distance, estimated duration, traffic, and peak-hour demand — ensuring fair pricing for both customers and drivers.
- **Route Optimization:** Suggests the fastest route for drivers to minimize travel time, enhancing service efficiency.
- **Safety Decision Rules:** Ensures only verified drivers are assigned to customers and alerts admin in case of abnormal behavior (e.g., prolonged idle status).

👉 **Example:** If two drivers are equidistant from a customer, the system prioritizes the one with better ratings or lower recent earnings — mimicking a dispatcher's logic to balance workload and service quality.

⌚ Why Combining These Categories Works Best

- **RDBMS** ensures efficient, reliable data handling.
- **OOPs** ensures clean, reusable, and scalable code.
- **Expert System** adds intelligence, improving efficiency and user satisfaction.

Analysis

Requirement Analysis

Functional Analysis

These are the **core features** that the system **must perform** — the "what it should do" part. Let's break them down:

1. User Module

- **User Registration/Login:** New users can create accounts, and existing users can log in securely.
- **Profile Management:** Users can update personal details.
- **Ride Booking:** Users enter pickup/drop locations, select ride type, and request a ride.
- **Fare Estimation:** The system calculates the estimated fare before confirming the ride.
- **Real-Time Ride Tracking:** Users track the driver's live location.
- **Payment Processing:** Supports multiple methods (cash, card, wallet).
- **Ride History:** Users can view past rides, fare breakdowns, and receipts.

2. Driver Module

- **Driver Registration/Login:** Drivers submit personal, license, and vehicle details.
- **Availability Toggle:** Drivers can switch between online and offline status.
- **Ride Request Handling:** Drivers receive notifications, accept/reject rides.
- **Ride Status Update:** Drivers update ride status (Accepted, In-Progress, Completed).
- **Earnings Summary:** Drivers view their daily/weekly/monthly earnings breakdown.

3. Admin Module

- **User/Driver Management:** Admin oversees user and driver accounts, handles disputes.
- **Ride Management:** Admin monitors ongoing rides and resolves issues.
- **Payment Oversight:** Admin ensures payments are logged and disputes managed.

Non-Functional Analysis

These define **how the system performs** rather than **what it does** — focusing on **performance, reliability, security,**

1. Performance Requirements

- **Fast Response Time:** Ride searches and fare calculations should complete within 2-3 seconds.
- **Concurrent Users:** System must handle **hundreds of simultaneous bookings** without lag.
- **Low Latency:** Real-time location tracking must update within **1-2 seconds**.

2. Security Requirements

- **User Data Protection:** Passwords are encrypted, payment data securely transmitted.
- **Role-Based Access:** Users, drivers, and admins have distinct permissions.
- **Fraud Prevention:** Block fake accounts, suspicious activities, and ensure ride tracking integrity.

3. Maintainability & Scalability

- **Modular Architecture:** Supports future feature upgrades without heavy rework.
- **Database Optimization:** Efficient indexing ensures performance even with millions of records.
- **Load Balancing:** Handles spikes during peak hours (e.g., weekends, festivals).

4. Usability Requirements

- **Intuitive Interface:** User-friendly design for both tech-savvy and non-tech users.
- **Multi-Platform Access:** Supports desktops, tablets, and smartphones.
- **Error Handling:** Clear error messages (e.g., "Invalid OTP," "Driver Unavailable").

5. Reliability Requirements

- **High Uptime:** System uptime should be **99.9%**.
- **Data Backup:** Automatic backups to prevent data loss.

Dataflow Diagrams

Level 0 DFD

Purpose of Level 0 DFD

- To **visualize the system's boundaries**.
- To **show external entities** (User, Driver, Admin) interacting with the system.
- To **illustrate data flow** — what information comes in, how the system processes it, and what outputs go back to the entities.

External Entities (Actors)

These are the people or systems interacting with the E-Taxi Booking System:

1. User (Customer)

- The user registers/logs in, books rides, tracks the ride, and makes payments.
- Data Flow: Provides user details, ride requests, payment information → Receives ride confirmations, driver details, fare info, and payment confirmation.

2. Driver

- The driver registers, accepts/rejects rides, and updates ride status (e.g., Arrived, Ride Started, Completed).
- Data Flow: Provides driver details, availability status, ride status updates → Receives ride requests and trip details.

3. Admin

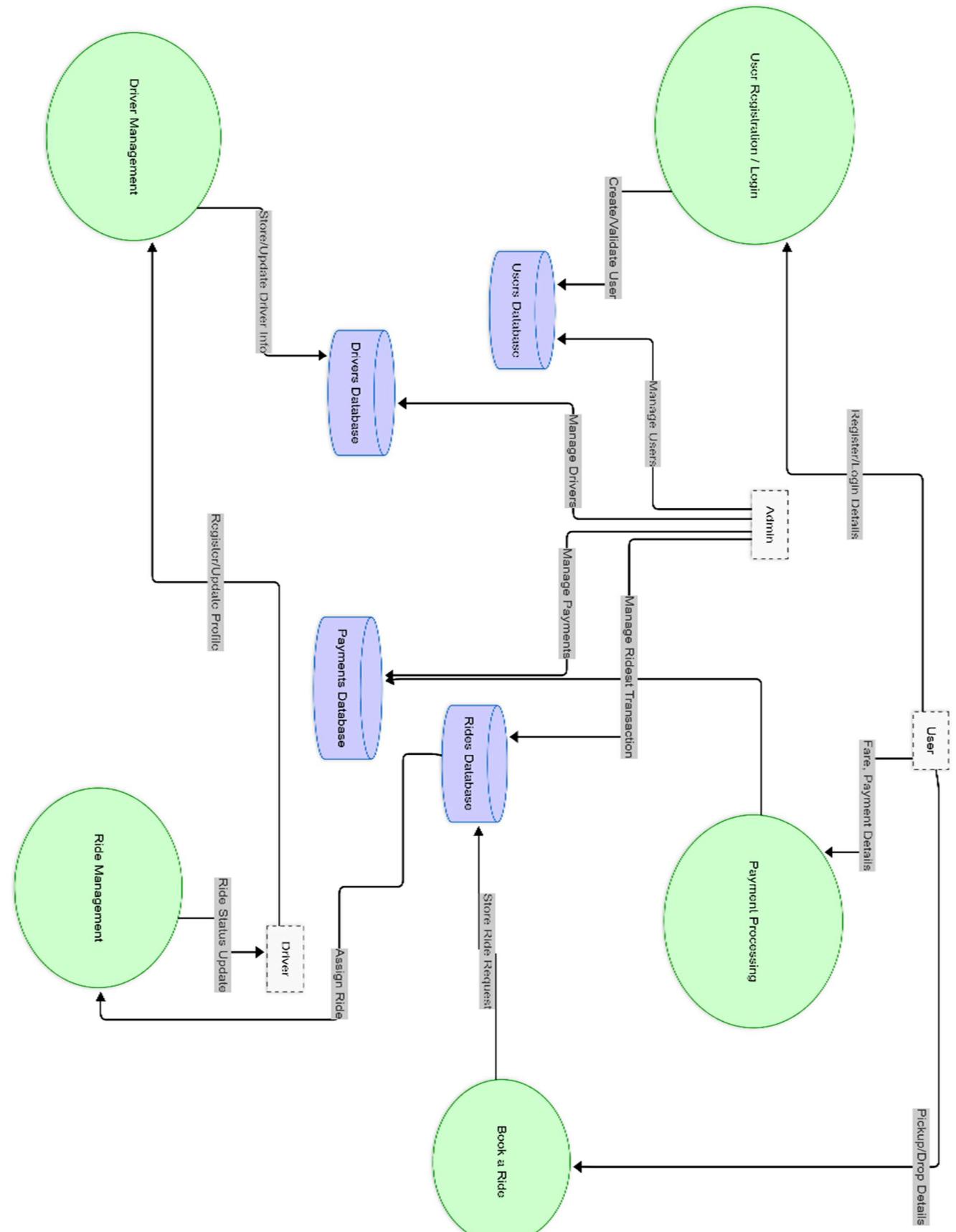
- Manages the system's backend — user accounts, driver profiles, rides, and payments.
- Data Flow: Provides system configurations, manages data, monitors rides → Receives reports on system performance, users, drivers, payments.

Main Process: E-Taxi Booking System

This is the **central process** — a single bubble (circle) representing the entire system's core functionality. It receives data from external entities, processes it, and produces results.

It handles:

- **User Registration & Authentication**
- **Ride Booking & Driver Assignment**



Level 1 DFD

⌚ Purpose of Level 1 DFD

- Decompose the **main system** into **smaller processes**.
- Show **detailed data interactions** within the system.
- Highlight **data stores** involved in each process.

♦ Key Subprocesses (Expanded Main System)

We take the **single main process** from Level 0 and expand it into multiple smaller functional units:

1. User Registration & Login

- **Input:** User submits details (name, phone, email, password, OTP).
- **Process:** The system verifies and stores user data in the **Users Database**.
- **Output:** User gets login confirmation or error message.

Flow:

User → Registration Details → System → User Validation → Users Database → Confirmation

2. Driver Registration & Management

- **Input:** Driver provides personal details, license, vehicle info.
- **Process:** The system verifies the driver's details, creating a profile in the **Drivers Database**.
- **Output:** Driver account activated, status set to **Available**.

Flow:

Driver → Registration Details → System → Driver Validation → Drivers Database → Profile Created

3. Booking a Ride

- **Input:** User provides pickup, drop location, ride type, and confirms.
- **Process:** The system creates a **ride request** and searches for an available driver.
- **Output:** User receives a ride confirmation with driver details.

Flow:

User → Ride Request → System → Rides Database → Driver Assignment → User (Ride Confirmed)

4. Driver Ride Handling

- **Input:** Driver receives ride request, accepts or rejects it.
- **Process:** If accepted, the system updates ride status to **In-progress**; if rejected, the system reassigns the ride.
- **Output:** User gets driver arrival update.

Flow:

Driver → Ride Acceptance/Rejection → System → Rides Database → User (Ride Status Update)

5. Payment Processing

- **Input:** User selects a payment method (cash, card, wallet), applies promo codes if available.
- **Process:** System calculates fare, applies discounts, and stores the transaction in **Payments Database**.
- **Output:** Payment confirmation, e-receipt generated.

Flow:

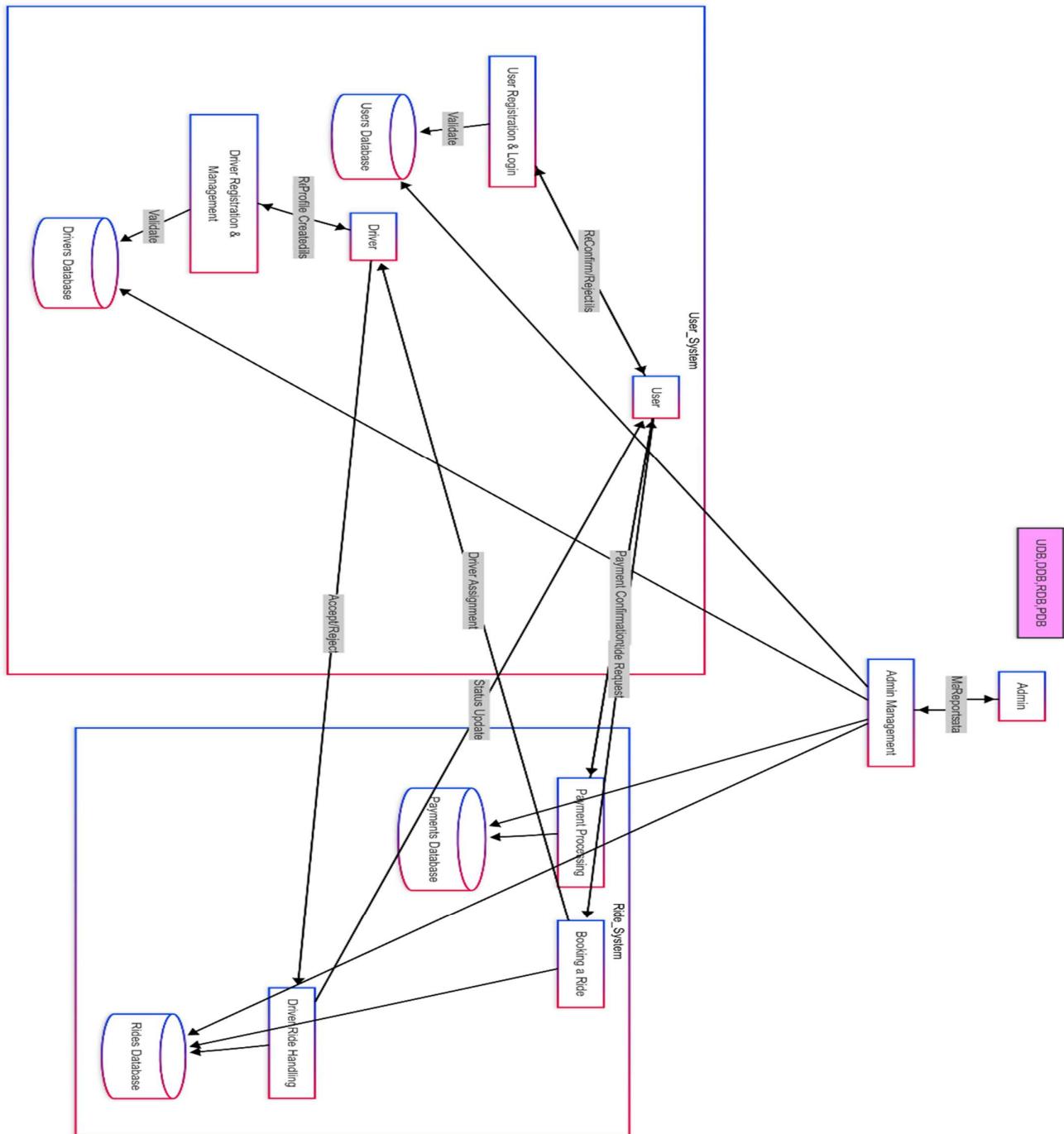
User → Payment Details → System → Payments Database → Payment Confirmation

6. Admin Management

- **Input:** Admin manages users, drivers, rides, and payments.
- **Process:** Admin modifies or monitors data, generates reports.
- **Output:** Admin receives system performance reports.

Flow:

Admin → Manage Data → System → All Databases → Reports



Level 2 DFD

⌚ Purpose of Level 2 DFD

- This is the **most detailed breakdown** of each **Level 1** subprocess.
- It **further decomposes** key operations, showing internal validations, calculations, and multiple data flows within each subprocess.

◆ Detailed Subprocesses in Level 2

Let's now **expand Level 1** into **smaller, detailed processes**:

1. User Registration & Login Process (Expanded)

- **Input:** User submits **name, email, phone, password, OTP**.
- **Subprocesses:**
 - Validate Input Data
 - Check for Existing User
 - OTP Verification
 - Create User Account
 - Start Session
- **Output:** User account created OR error message (e.g., "User already exists").

Flow:

User → Input Data → Validation → Check Existing User → OTP Verification → Users Database → Account Created / Error → Login Session

2. Driver Registration & Management (Expanded)

- **Input:** Driver provides **personal info, license, vehicle details**.
- **Subprocesses:**
 - Validate Driver Data
 - License Verification
 - Vehicle Details Check
 - Create Driver Profile
 - Set Availability to **Online**
- **Output:** Driver profile created or error (e.g., "Invalid license").

Flow:

Driver → Input Data → Validation → License & Vehicle Check → Drivers Database → Profile Created / Error → Availability Status Set

3. Ride Booking (Expanded)

- **Input:** User selects **pickup, drop, ride type (Economy, Premium)**.
- **Subprocesses:**
 - Validate Locations
 - Estimate Fare (Distance, Ride Type, Demand Surge)
 - Search for Available Drivers
 - Notify User (Waiting for Driver)
- **Output:** Ride request confirmed, driver details sent to user.

Flow:

User → Pickup/Drop Location → Location Validation → Estimate Fare → Search Driver → Rides Database
→ Notify User

4. Driver Ride Handling (Expanded)

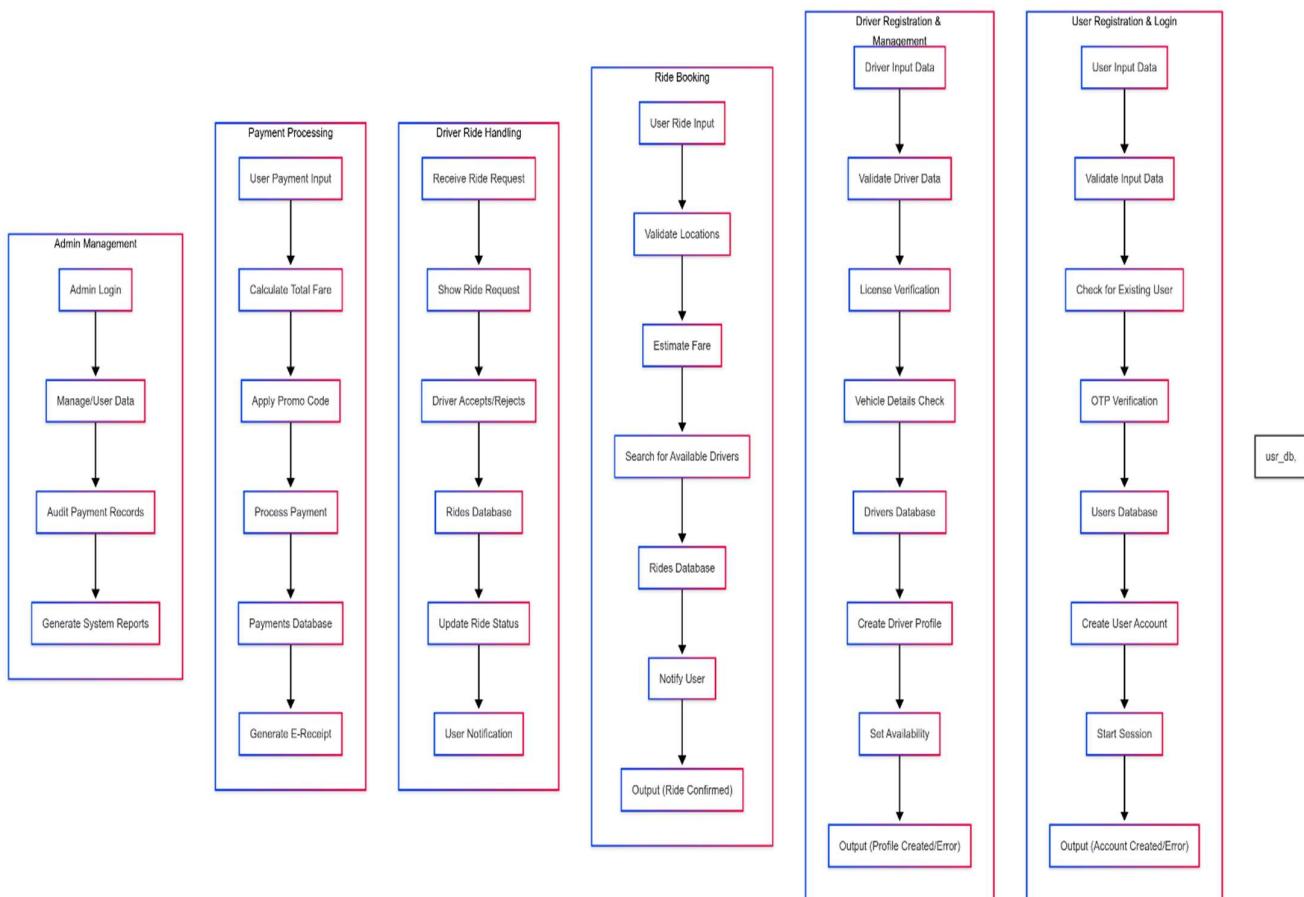
- **Input:** Driver receives ride request.
- **Subprocesses:**
 - Show Ride Request to Driver
 - Driver Accepts/Rejects
 - If Rejected → Reassign to Another Driver
 - Update Ride Status (Accepted, In-Progress, Completed)
- **Output:** Ride starts or user notified of driver reassignment.

5. Payment Processing (Expanded)

- **Input:** User selects payment method (cash, card, wallet).
- **Subprocesses:**
 - Calculate Total Fare (Base Fare + Distance + Ride Type + Surge Price)
 - Process Payment
 - Generate E-Receipt
- **Output:** Payment confirmation, receipt sent to user.

Flow:

User → Select Payment → Calculate Fare → Apply Promo Code → Payments Database → Payment Confirmation → Generate Receipt



Entity Relationship Diagram

Admins Table

- Stores admin user details who manage the system.
- Fields: id, name, email, password, phone, profile_image, created_at.
- **Purpose:** Allows system administrators to log in and manage drivers, customers, rides, payments.

Drivers Table

- Contains driver data, with essential fields like name, email, password, phone, and id_number for verification.
- status: Can be approved, pending, or rejected — reflecting the driver's application status.
- **Relationship:**
 - Connected to vehicles (each driver owns a vehicle).
 - Linked to rides (driver takes on multiple rides).

Vehicles Table

- Captures vehicle-related details like license_number, vehicle_type, model, number, and vehicle_image.
- Foreign Key: driver_id — links each vehicle to a specific driver.
- **Purpose:** Ensures every driver is associated with a registered vehicle.

Customers Table

- Holds customer data like name, email, password, phone, and profile_image.
- login: Supports multiple login methods — manual, google, or facebook.
- **Relationship:**
 - Connected to rides (customers book rides).
 - Linked to reviews (customers leave feedback).

Rides Table

- Central table that ties customers and drivers together.
- Fields: pickup_location, dropoff_location, fare, status (pending, accepted, completed, cancelled, paid).
- request_time and complete_time: Tracks ride duration.
- **Relationships:**
 - Linked to drivers and customers (who's riding with whom).
 - Connected to payments (each ride results in a payment).

Payments Table

- Manages transactions for rides.
- Fields: ride_id, customer_id, driver_id, payment_id, amount, payment_method, and payment_status.

- **Purpose:** Tracks all payments and ensures ride statuses like paid are updated.

Reviews Table

- Allows customers to rate and review drivers.
- Fields: ride_id, customer_id, driver_id, rating, review.
- **Purpose:** Supports a feedback system to improve service quality.

Notifications Table

- Handles system notifications for users (drivers/customers).
- Fields: user_id, message, status (unread, read).
- **Purpose:** Keeps users updated on ride status, promotions, or other alerts.

Promo Codes Table

- Supports discount codes and promotions.
- Fields: code, discount_percentage, valid_from, valid_to.
- **Purpose:** Allows customers to redeem discounts when booking rides.

Messages Table

- Facilitates communication between customers and drivers/admins.
- Fields: sender, receiver, content, status (seen, delivered).
- **Purpose:** Supports in-app messaging for smoother ride coordination.

Blogs Table

- A bonus feature! Supports content marketing or announcements.
- Fields: title, content, author, created_at.



UML class diagram

🎯 1. Overview of the System

Our **E-Taxi Booking System** allows **customers** to book rides, **drivers** to accept them, and **admins** to manage the platform. We're combining the core **MVC architecture** with **advanced features** like **notifications, reviews, and admin monitoring** for a fully scalable solution.

✓ Key Stakeholders:

- **Customers:** Book rides, get notifications, leave reviews.
- **Drivers:** Accept rides, update status, view earnings.
- **Admins:** Manage users, drivers, payments, rides, and monitor system performance.

✓ Technology Stack:

- **Backend:** Object-Oriented PHP (hidden from user)
- **Frontend:** HTML/CSS, JavaScript
- **Database:** MySQL
- **Architecture:** MVC (Model-View-Controller)

🛠 2. Detailed MVC Structure Breakdown

Let's dive into each MVC layer — with classes, methods, and relationships.

⌚ Models (Data Layer)

🧠 A) User Model

```
class User {  
    public $id, $name, $email, $password, $phone, $profile_image, $created_at;  
  
    public function register() {}  
    public function login() {}  
    public function updateProfile() {}  
}
```

✓ Relationships:

- A user has **multiple notifications**.
- A user **books** multiple rides.

🛠 B) Driver Model

```
class Driver extends User {
```

```
public $id_number, $vehicle_details, $status;
```

```
    public function acceptRide() {}  
    public function updateStatus() {}  
}
```

Relationships:

- A driver **handles** multiple rides.
- A driver **receives** multiple reviews.

C) Ride Model

```
class Ride {  
  
    public $id, $customer_id, $driver_id, $pickup, $dropoff, $fare, $status, $request_time;  
  
    public function createRide() {}  
    public function updateStatus() {}  
    public function completeRide() {}  
}
```

Relationships:

- Each ride **has one** review.
- A ride **triggers** a notification.

D) Payment Model

```
class Payment {  
  
    public $id, $ride_id, $method, $amount, $status;  
  
    public function processPayment() {}  
    public function applyPromoCode() {}  
}
```

Relationships:

- Payment **belongs to** one ride.

E) Notification Model

```
class Notification {  
  
    public $id, $user_id, $message, $status, $created_at;  
  
    public function sendNotification() {}
```

```
    public function markAsRead() {}  
}
```

Relationships:

- Notification **belongs to** one user (driver/customer).

★ F) Review Model

```
class Review {  
    public $id, $ride_id, $customer_id, $driver_id, $rating, $review_text, $created_at;  
  
    public function submitReview() {}  
}
```

🎯 Controllers (Business Logic Layer)

Here's a quick breakdown of our controllers:

Controller Key Functions

UserController	handleLogin(), handleRegistration(), handleProfileUpdate()
-----------------------	--

DriverController	handleRideAcceptance(), handleStatusUpdate()
-------------------------	--

RideController	initiateRide(), assignDriver(), completeRide()
-----------------------	--

PaymentController	handlePaymentProcessing(), applyPromoCode(), generateReceipt()
--------------------------	--

NotificationController	createNotification(), getNotifications(), markNotificationRead()
-------------------------------	--

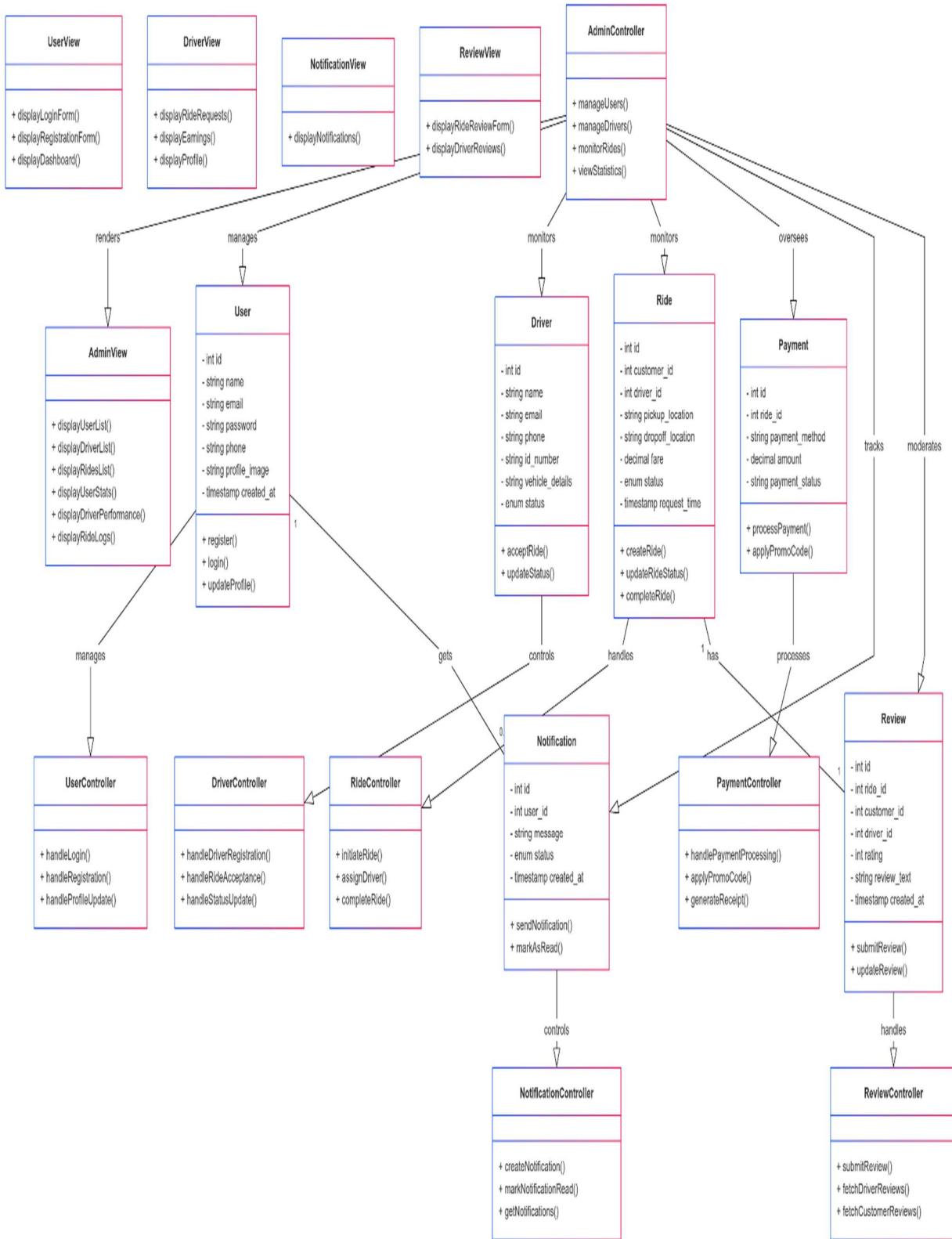
ReviewController	submitReview(), fetchDriverReviews()
-------------------------	--------------------------------------

AdminController	manageUsers(), manageDrivers(), monitorRides(), viewStatistics()
------------------------	--

🎯 Views (UI Layer)

- **CustomerView:** Dashboard, ride booking, reviews.
- **DriverView:** Ride requests, earnings, status.
- **AdminView:** User/driver lists, performance charts, logs.

- **NotificationView:** Shows new notifications.



Structure and Control Flows

Tech Stack

The E-Taxi project is built using a combination of frontend, backend, database, and third-party APIs to ensure a seamless and efficient ride-booking system. Below is a detailed breakdown of the **tech stack** used:

1. Frontend Technologies

These technologies handle the **user interface (UI)** and ensure a smooth experience for customers, drivers, and admins.

HTML (HyperText Markup Language)

- Used to structure the web pages.
- Provides the foundation for displaying content, forms, and tables.

CSS & Bootstrap

- **CSS** (Cascading Style Sheets) is used for styling the application.
- **Bootstrap** (CSS framework) is used for responsive design, providing pre-designed UI components like buttons, forms, grids, and navigation bars.

JavaScript & jQuery

- **JavaScript** is used to enhance the interactivity of the website.
- **jQuery** (a JavaScript library) simplifies DOM manipulation, AJAX calls, and event handling.

AJAX (Asynchronous JavaScript and XML)

- Used for real-time interactions without reloading the page.
- Helps in **OTP verification, ride booking, driver status updates, and payment status checks**.

2. Backend Technologies

The backend ensures **business logic execution, data processing, and API integration**.

PHP (Hypertext Preprocessor)

- **Core backend language** handling user authentication, ride booking, payments, and data processing.
- Used in **controllers, models, and API endpoints**.
- Framework used: **Custom PHP MVC pattern** (no Laravel or CodeIgniter mentioned).

MySQL Database

- **Relational database** used to store user data, ride details, payments, and reports.
- Queries are executed using **PHP MySQLi or PDO**.

3. Third-Party APIs & Services

Several third-party APIs are integrated for different functionalities:

SMTP (Simple Mail Transfer Protocol)

- Used to send **email notifications**, including OTP verification and ride confirmations.
- The `smtp` library in the `libs/` directory handles email sending.
- Works with **PHPMailer** or **third-party SMTP services** (like Gmail SMTP or SendGrid).

UltraMsg WhatsApp API

- Integrated for **WhatsApp notifications**.
- Sends messages for **ride confirmations, OTPs, and payment alerts**.
- Uses UltraMsg's API for quick **WhatsApp messaging automation**.

Google Maps Route API

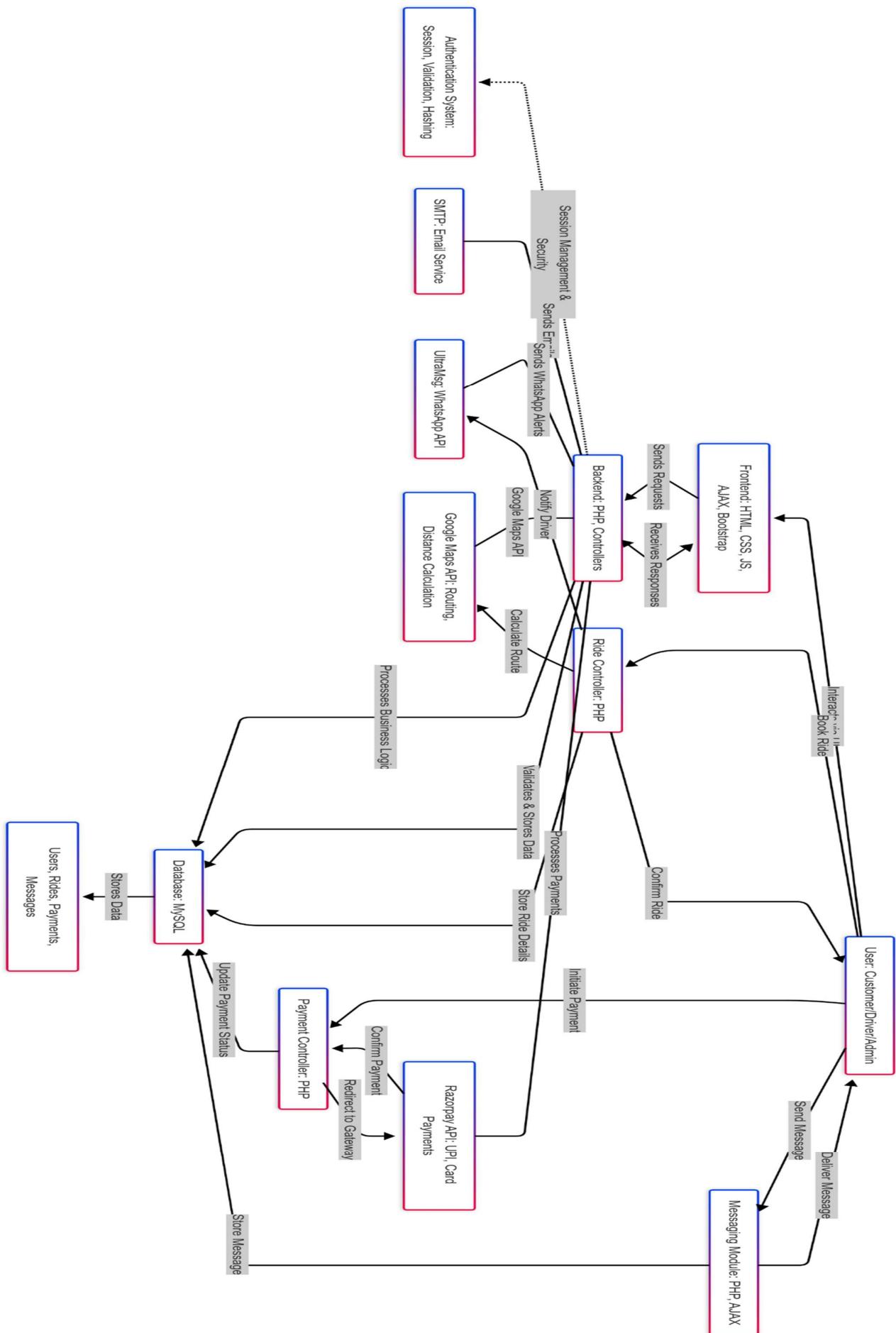
- Used for:
 - **Live route tracking** between customer & driver.
 - **Distance calculation** to estimate fares.
 - **Auto-suggestions** for locations in ride booking.
- Provides **latitude & longitude coordinates** for route planning.

Razorpay Payment Gateway API

- Manages **online payments** for ride bookings.
- Allows customers to pay using **UPI, debit/credit cards, and net banking**.
- Secure API with **callback functions** for success and failure transactions.

Summary of Tech Stack

Technology	Purpose
HTML	Page structure
CSS & Bootstrap	Styling and responsiveness
JavaScript & jQuery	User interaction and dynamic updates
AJAX	Asynchronous data exchange (e.g., real-time updates)
PHP	Backend processing and business logic
MySQL	Database storage and queries
SMTP	Email notifications (OTP, ride confirmation)
UltraMsg API	WhatsApp alerts and notifications
Google Maps API	Route tracking and distance calculation
Razorpay API	Online payments processing
Session Handling	User authentication and security



MVC (Model-View-Controller) Structure

The eTaxi project follows the **MVC (Model-View-Controller)** architectural pattern, which helps in maintaining a well-structured and scalable application.

💡 What is MVC?

MVC is a software design pattern that separates an application into **three main components**:

1. Model (M)

- Represents the **data and business logic**.
- Interacts with the **database** to fetch, insert, update, and delete data.
- Example: User authentication, ride details, payment status.

2. View (V)

- Handles the **UI (User Interface)**.
- Displays data received from the **Controller**.
- Example: Booking a ride page, driver dashboard, user profile.

3. Controller (C)

- Manages the **business logic and request handling**.
- Acts as a **bridge between Model & View**.
- Processes **user input**, interacts with the **Model**, and updates the **View**.
- Example: When a user books a ride, the **RideController** verifies data, updates the database, and returns a success message.

1. Models (Data & Business Logic)

Located in the /models directory:

- Admin.php – Handles admin-related data.
- Blog.php – Manages blog content.
- Customer.php – Stores customer information.
- Driver.php – Stores driver details.
- Messages.php – Manages messaging between users.
- Notification.php – Handles system notifications.
- Payment.php – Manages payment transactions.
- PromoCodes.php – Stores promo codes and discounts.
- Report.php – Generates ride/payment reports.
- Review.php – Stores user reviews and ratings.
- Ride.php – Manages ride details and tracking.

2. Views (User Interface & Templates)

Located in the /views directory:

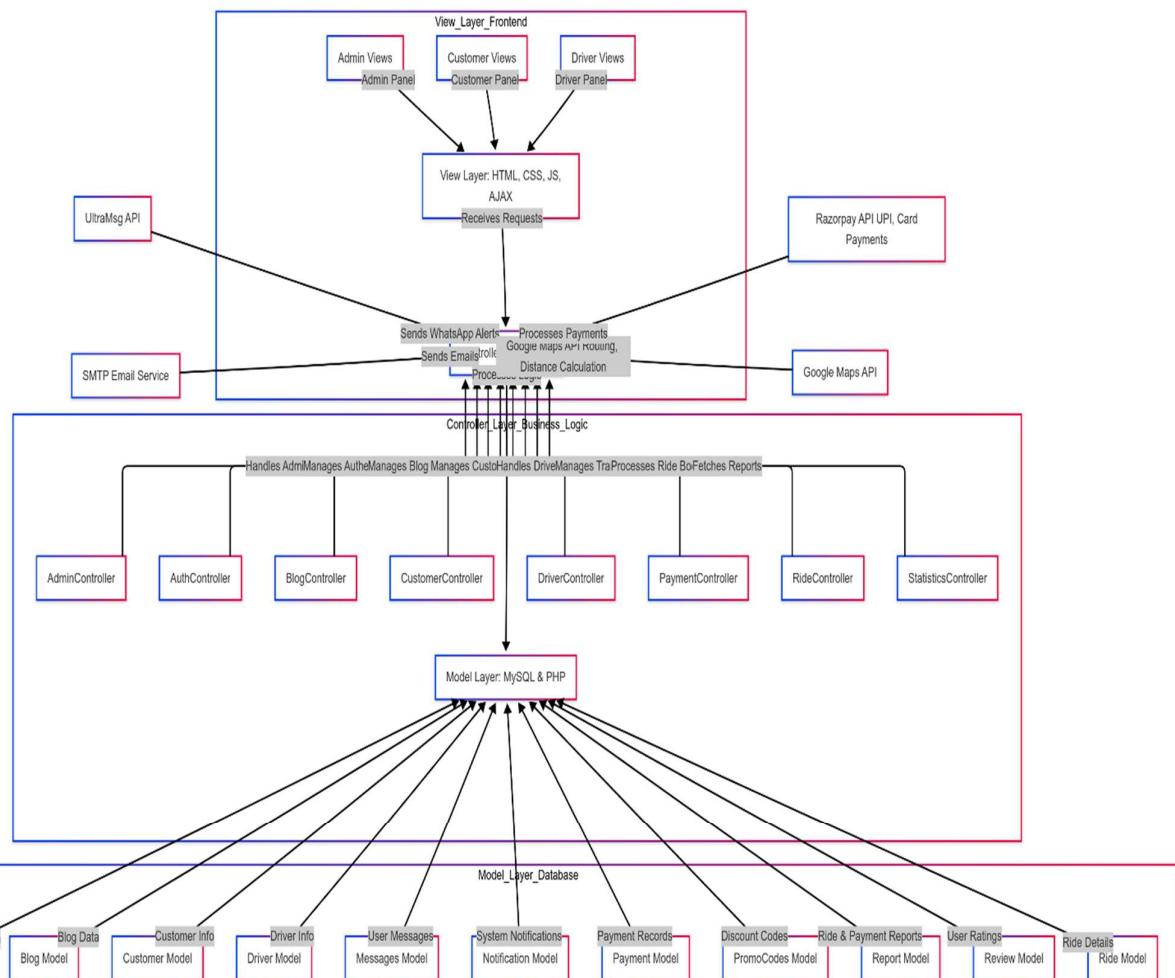
- **Admin Panel (/views/admin/)**
 - dashboard.php – Admin dashboard.
 - manage-drivers.php – Manage driver accounts.
 - manage-customers.php – Manage customer details.
 - manage-rides.php – View ride history.
- **Customer Panel (/views/customer/)**

- dashboard.php – Customer dashboard.
- book-ride.php – Ride booking page.
- my-rides.php – View past rides.
- payments.php – Manage payments.
- **Driver Panel (/views/_driver/)**
- dashboard.php – Driver dashboard.
- my-rides.php – View assigned rides.
- earnings.php – Track earnings.

3. Controllers (Logic & Request Handling)

Located in the /controllers directory:

- AdminController.php – Handles admin actions.
- AuthController.php – Manages user authentication.
- BlogController.php – Handles blog operations.
- CustomerController.php – Manages customer-related tasks.
- DriverController.php – Handles driver data.
- PaymentController.php – Manages payment transactions.
- RideController.php – Processes ride bookings.
- StatisticsController.php – Fetches reports & analytics.



Modules

The eTaxi system is designed using the MVC (Model-View-Controller) architecture, breaking the application into different modules for better scalability, maintainability, and performance. Each module has a specific function, ensuring a seamless ride-booking experience for customers, drivers, and administrators.

This project is designed with a **modular architecture**, ensuring scalability, maintainability, and efficient system management. Each module follows a structured **data model** with well-defined **database schemas**, ensuring smooth data flow across different system components. The **process logic** is implemented using the **MVC (Model-View-Controller) pattern**, separating the user interface, business logic, and data management layers for better performance and maintainability.

The **testing approach** includes **unit testing, integration testing, and user acceptance testing (UAT)** to ensure the system's stability, security, and responsiveness. This structured breakdown of modules and processes provides an **accurate estimation of the student's effort** in understanding and implementing the project effectively.

Login Module

1. Module Overview

Purpose

The Authentication Module is a comprehensive security system that manages user authentication and verification through both traditional login methods and OTP-based verification. This integrated module serves as the gateway for user access control in the E-taxi service platform, implementing a sophisticated multi-layered authentication approach.

Key objectives:

- Provide unified authentication services for multiple user roles
- Implement secure multi-factor authentication
- Ensure seamless user experience across different authentication methods
- Maintain high security standards while preserving accessibility
- Support both synchronous and asynchronous verification methods

2. Data Structures

Database Tables

1. Admins Table	2. Customers Table	3. Drivers Table
<pre>CREATE TABLE admins (id INT PRIMARY KEY AUTO_INCREMENT, name VARCHAR(255), email VARCHAR(255) UNIQUE, password VARCHAR(255), profile_image VARCHAR(255), created_at TIMESTAMP);</pre>	<pre>CREATE TABLE customers (id INT PRIMARY KEY AUTO_INCREMENT, name VARCHAR(255), email VARCHAR(255) UNIQUE, password VARCHAR(255), phone VARCHAR(20), profile_image VARCHAR(255), created_at TIMESTAMP, login VARCHAR(50));</pre>	<pre>CREATE TABLE drivers (id INT PRIMARY KEY name VARCHAR(255), email VARCHAR(255) UNIQUE, password VARCHAR(255), phone VARCHAR(20), id_number VARCHAR(50), status ENUM('pending', 'approved', 'rejected'), profile_image VARCHAR(255), created_at TIMESTAMP);</pre>

3. Process Logic

Implementation Process

1. Initial Authentication

```
// AuthController.  
switch ($role) {  
    case 'admin':  
        $response = $admin->login($email, $password);  
        break;  
    case 'customer':  
        $response = $this->initiateOTPAuth('email', $email);  
        break;  
    case 'driver':  
        $response = $this->initiateOTPAuth('mobile', $phone);  
        break;  
}
```

2. OTP Generation and Delivery

```
// OTP Generation  
function generateOTP() {  
    return rand(100000, 999999);}  
// Email OTP Delivery  
$mail = new Mailer(true);  
$mail->setFrom("system@etaxi.com", "ETaxi Service");  
$mail->addAddress($email);  
$mail->Subject = "Authentication OTP";  
$mail->Body = "Your OTP: $otp";  
$mail->send();  
// Mobile OTP Delivery  
$params = [  
    'token' => 'API_TOKEN',  
    'to' => $phone,  
    'body' => "Your ETaxi OTP: $otp"  
];  
curl_setopt($curl, CURLOPT_POSTFIELDS, http_build_query($params));
```

3. Verification Process

```
// OTP Verification  
function verifyOTP($type, $identifier, $otp) {  
    switch ($type) {  
        case 'email':  
            return $otp === $_SESSION['reset_otp'] &&  
                   $identifier === $_SESSION['reset_email'];  
        case 'mobile':  
            return $otp === $_SESSION['otp'] &&  
                   $identifier === $_SESSION['otp_phone'];  
    }
```

Registration Module

1. Module Overview

Purpose

The Registration Module provides a comprehensive user registration system for the E-taxi service platform, handling both customer and driver registrations. This module implements a secure, multi-step registration process with OTP verification to ensure data validity and user authenticity.

Scope

The Registration Module encompasses:

Customer Registration Features:

- Basic information collection (first name, last name, email, phone)
- Email and phone verification through OTP
- Profile image upload
- Password security with BCRYPT hashing
- Automatic phone number generation if not provided

Driver Registration Features:

- Personal information collection
- Document verification (Aadhaar, License)
- Vehicle information management
- Multiple image uploads (profile, ID, vehicle, license)
- OTP verification**
- Status tracking (pending/approved/rejected)
- Vehicle number uniqueness validation

2. Data Structures

Database Tables

1. Customers Table

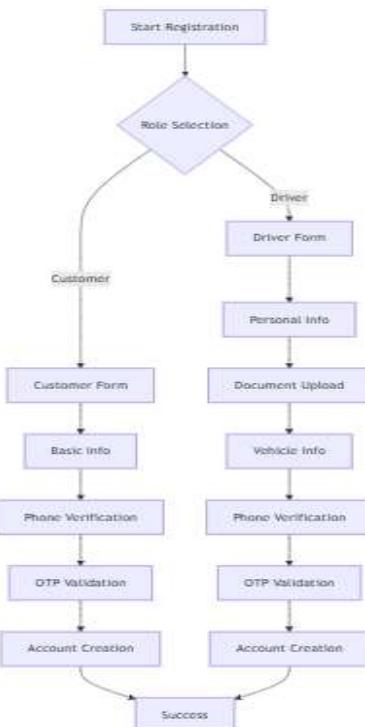
```
CREATE TABLE customers (
    id INT PRIMARY KEY
    AUTO_INCREMENT,
    name VARCHAR(255),
    email VARCHAR(255) UNIQUE,
    password VARCHAR(255),
    phone VARCHAR(20),
    profile_image VARCHAR(255),
    created_at TIMESTAMP,
    login VARCHAR(50)
);
```

2. Drivers Table

```
CREATE TABLE drivers (
    id INT PRIMARY KEY
    AUTO_INCREMENT,
    name VARCHAR(255),
    email VARCHAR(255) UNIQUE,
    password VARCHAR(255),
    phone VARCHAR(20),
    id_number VARCHAR(50),
    id_image VARCHAR(255),
    status ENUM('pending',
    approved', 'rejected'),
    profile_image VARCHAR(255),
    created_at TIMESTAMP
);
```

3. Vehicles Table

```
CREATE TABLE vehicles (
    id INT PRIMARY KEY
    AUTO_INCREMENT,
    driver_id INT,
    license_number VARCHAR(50),
    license_image VARCHAR(255),
    vehicle_type VARCHAR(50),
    vehicle_model VARCHAR(100),
    vehicle_number VARCHAR(20),
    vehicle_image VARCHAR(255),
    created_at TIMESTAMP,
    FOREIGN KEY (driver_id)
    REFERENCES drivers(id)
);
```



3. Process Logic

Implementation Process

1. Customer Registration

```
// user_register.  
<form id="registerForm" action="controllers/AuthController." method="POST"  
enctype="multipart/form-data">  
    <!-- Hidden Fields -->  
    <input type="hidden" name="action" value="register">  
    <input type="hidden" name="role" value="customer">  
    <!-- Personal Information -->  
    <div class="row">  
        <div class="col-lg-6 mb-3">  
            <label class="form-label">First Name</label>  
            <input type="text" name="firstname" class="form-control" required>  
        </div>  
        <div class="col-lg-6 mb-3">  
            <label class="form-label">Last Name</label>  
            <input type="text" name="lastname" class="form-control" required>  
        </div>  
        <div class="col-lg-12 mb-3">  
            <label class="form-label">Email Address</label>  
            <input type="email" name="email" class="form-control" required>  
        </div>  
        <div class="col-lg-12 mb-3">  
            <label class="form-label">Phone Number</label>  
            <div class="input-group">  
                <input type="tel" name="phone" class="form-control" required>  
                <button type="button" class="btn btn-secondary" id="sendOtp">Send OTP</button>  
            </div>  
            <!-- Password -->  
            <div class="col-lg-12 mb-3">  
                <label class="form-label">Password</label>  
                <input type="password" name="password" class="form-control" required>  
            </div>  
            <!-- Submit Button -->  
            <div class="col-lg-12 mb-3">  
                <button type="submit" class="btn btn-primary w-100">Create Account</button>  
            </div>  
    </div>  
</form>
```

2. Driver Registration

```
// driver_register.
```

```
<form id="registerForm" action="controllers/AuthController." method="POST"
enctype="multipart/form-data">
    <!-- Hidden Fields -->
    <input type="hidden" name="action" value="register">
    <input type="hidden" name="role" value="customer">
    <!-- Personal Information -->
    <div class="row">
        <!-- First Name -->
        <div class="col-lg-6 mb-3">
            <label class="form-label">First Name</label>
            <input type="text" name="firstname" class="form-control" required>
        </div>
        <!-- Last Name -->
        <div class="col-lg-6 mb-3">
            <label class="form-label">Last Name</label>
            <input type="text" name="lastname" class="form-control" required>
        </div>
        <!-- Email -->
        <div class="col-lg-12 mb-3">
            <label class="form-label">Email Address</label>
            <input type="email" name="email" class="form-control" required>
        </div>
        <input type="text" name="otp" class="form-control" required>
        <button type="button" class="btn btn-secondary" id="verifyOtp">Verify
OTP</button>
    </div>
    </div>
    <div class="col-lg-12 mb-3">
        <label class="form-label">Password</label>
        <input type="password" name="password" class="form-control" required>
    </div>
    <!-- Submit Button -->
    <div class="col-lg-12 mb-3">
        <button type="submit" class="btn btn-primary w-100">Create Account</button>
    </div>
</div>
</form>
```

User Module

1. Module Overview

Purpose

The User Module provides a comprehensive interface for customers to manage their taxi booking experience. This module handles user profile management, ride booking, payment tracking, and communication with drivers.

2. Data Structures

Database Tables

1. Customers Table	2. Rides Table	3. Payments Table
<pre>CREATE TABLE customers (id INT PRIMARY KEY AUTO_INCREMENT, name VARCHAR(255), email VARCHAR(255) UNIQUE, password VARCHAR(255), phone VARCHAR(20), profile_image VARCHAR(255), created_at TIMESTAMP, login VARCHAR(50));</pre>	<pre>CREATE TABLE rides (id INT PRIMARY KEY AUTO_INCREMENT, customer_id INT, driver_id INT, pickup_location VARCHAR(255), dropoff_location VARCHAR(255), ride_status ENUM('pending', accepted', 'ongoing', 'completed', cancelled'), fare DECIMAL(10,2), distance DECIMAL(10,2), rate DECIMAL(10,2), request_time TIMESTAMP, FOREIGN KEY (customer_id) REFERENCES customers(id), FOREIGN KEY (driver_id) REFERENCES drivers(id));</pre>	<pre>CREATE TABLE payments (id INT PRIMARY KEY AUTO_INCREMENT, ride_id INT, customer_id INT, amount DECIMAL(10,2), payment_status ENUM('pending', completed', 'failed'), payment_date TIMESTAMP, FOREIGN KEY (ride_id) REFERENCES rides(id), FOREIGN KEY (customer_id) REFERENCES customers(id));</pre>

3. Process Logic

Implementation Process

1. Dashboard Initialization

```
// CustomerController.  
if ($_SERVER['REQUEST_METHOD'] === 'GET') {  
    $customerData = $_SESSION['customer'];  
    $customerRides = $ride->getRidesByCustomer($customerData['id']);  
    $totalRides = count($customerRides);  
    $totalPayments = $payment->getTotalPaymentsByCustomer($customerData['id']);  
    $latestRides = $ride->getLatestRidesByCustomer($customerData['id'], 5);  
}
```

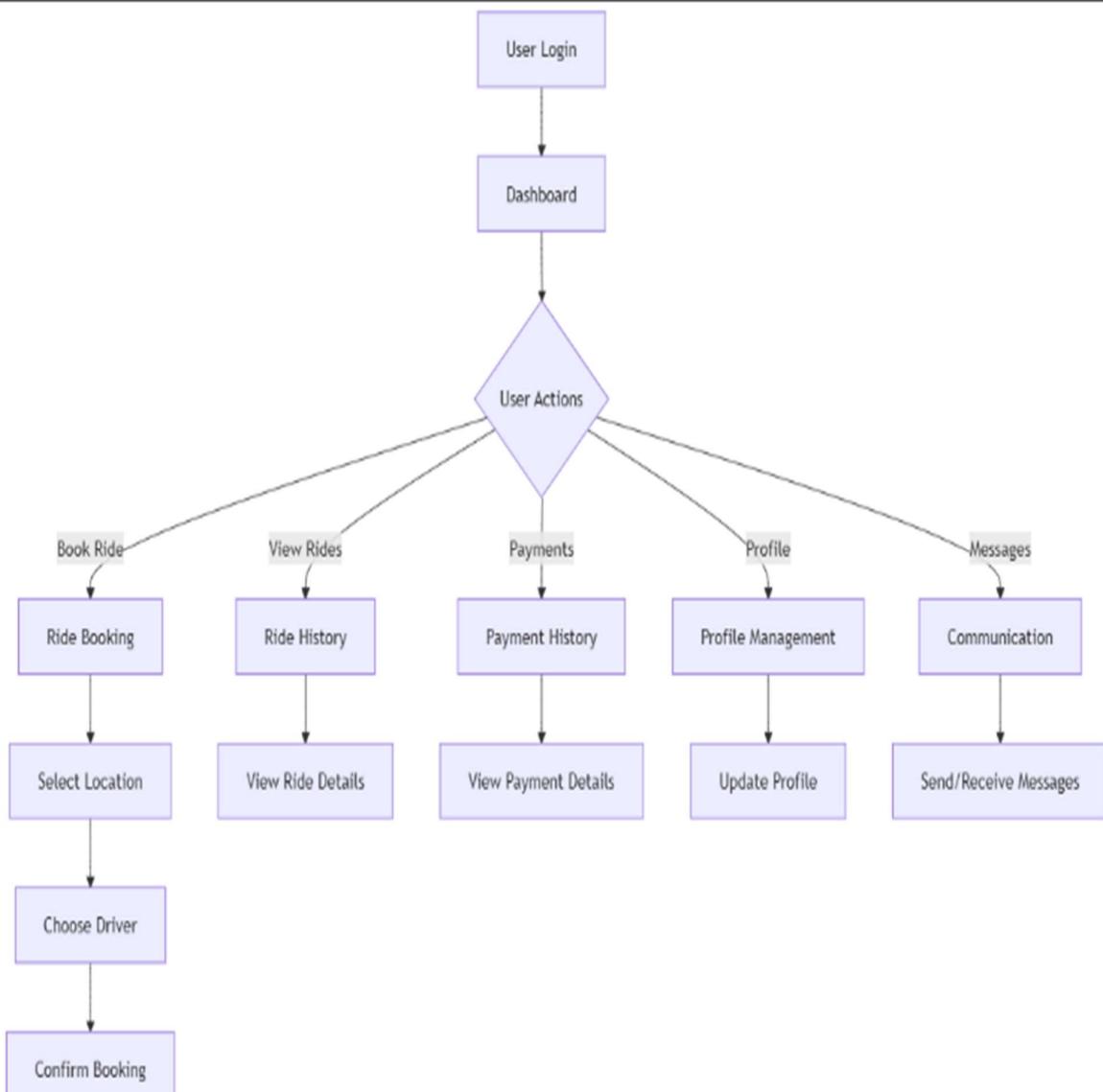
2. Profile Management

```
// CustomerController.  
if ($action === 'updateProfile') {  
    $data = [
```

```

'id' => $_SESSION['customer']['id'],
'name' => $_POST['name'],
'email' => $_POST['email'],
'phone' => $_POST['phone']
];
if (isset($_FILES['profile_image'])) {
    $data['profile_image'] = $_FILES['profile_image'];
}
$result = $customer->updateProfile($data);
}

```



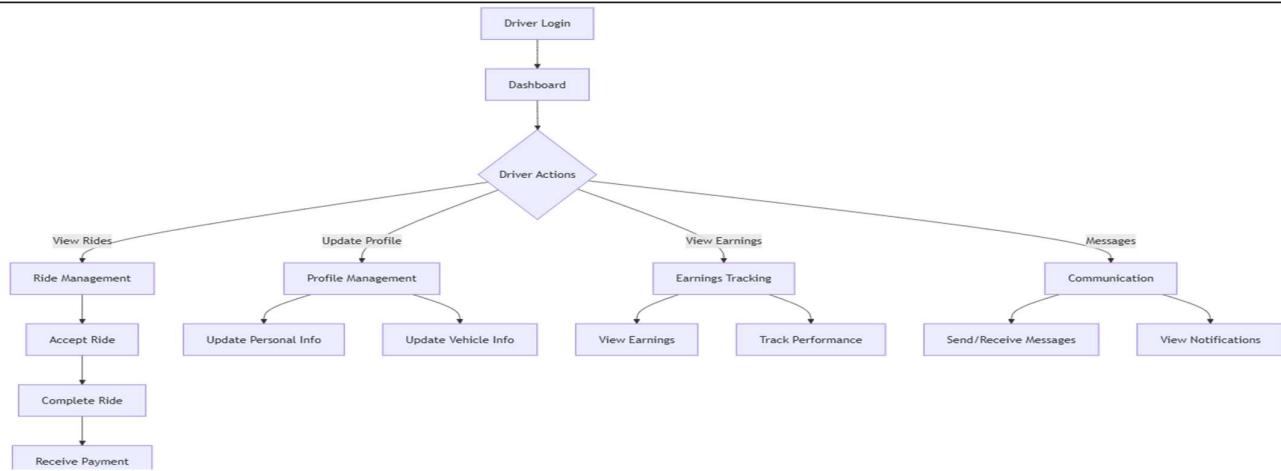
Driver Module

1. Module Overview

Purpose

The Driver Module manages the driver-side functionality of the taxi booking system, enabling drivers to:

- Manage their availability status
- Accept and complete rides
- Track earnings and performance



2. Data Structures

Database Tables

1. Drivers Table

```
CREATE TABLE drivers (
    id INT PRIMARY KEY
    AUTO_INCREMENT,
    name VARCHAR(255),
    email VARCHAR(255) UNIQUE,
    password VARCHAR(255),
    phone VARCHAR(20),
    id_number VARCHAR(50),
    id_image VARCHAR(255),
    status ENUM('pending',
    approved', 'rejected'),
    profile_image VARCHAR(255),
    created_at TIMESTAMP
);
```

2. Vehicles Table

```
CREATE TABLE vehicles (
    id INT PRIMARY KEY
    AUTO_INCREMENT,
    driver_id INT,
    license_number VARCHAR(50),
    license_image VARCHAR(255),
    vehicle_type VARCHAR(50),
    vehicle_model VARCHAR(100),
    vehicle_number VARCHAR(20),
    vehicle_image VARCHAR(255),
    created_at TIMESTAMP,
    FOREIGN KEY (driver_id)
    REFERENCES drivers(id)
);
```

3. Driver Availability Table

```
CREATE TABLE driver_availability (
    id INT PRIMARY KEY
    AUTO_INCREMENT,
    driver_id INT,
    is_available BOOLEAN DEFAULT
    TRUE,
    updated_at TIMESTAMP,
    FOREIGN KEY (driver_id)
    REFERENCES drivers(id)
);
```

3. Process Logic

Implementation Process

1. Driver Registration

```
public function register($data) {
    // Validate existing email/phone
    $check_driver_ = "SELECT email, phone FROM drivers WHERE email = ? OR phone = ?";
    // Upload profile and ID images
    $profile_image_name = $name . '_' . substr($phone, -4) . '_profile_' . time();
    $id_image_name = $name . '_' . substr($id_number, -4) . '_id_' . time();
    // Insert driver data
    $_driver = "INSERT INTO drivers (name, email, password, phone, id_number, id_image, status,
    profile_image) VALUES (?, ?, ?, ?, ?, ?, ?, ?)";
    // Insert vehicle data
    $_vehicle = "INSERT INTO vehicles (driver_id, license_number, vehicle_type, vehicle_model,
    vehicle_number) VALUES (?, ?, ?, ?, ?)";
}
```

2. Ride Management

```
// DriverController.
public function acceptRide($ride_id, $driver_id) {
    // Check availability
    $availability = $this->driver->getDriverAvailabilityStatus($driver_id);
    if (!$availability) {
        return ["success" => false, "error" => "Driver is currently unavailable"];
    }
    $this->driver->updateAvailabilityStatus($driver_id, 'unavailable');
    return $this->ride->acceptRide($ride_id, $driver_id);
}
```

3. Profile Management

```
public function updateProfile($driver_id, $data, $files) {
    try {
        $this->conn->begin_transaction();
        $_driver = "UPDATE drivers SET name = ?, email = ?, phone = ?, id_number = ? WHERE id = ?";
        if (!empty($files['profile_image']['name'])) {
            $profile_image = $this->uploadImage($files['profile_image'], 'profile', $data['name']);
        }
        $_vehicle = "UPDATE vehicles SET license_number = ?, vehicle_type = ?, vehicle_model = ?,
        vehicle_number = ? WHERE driver_id = ?";
        $this->conn->commit();
        return ["success" => true];
    } catch (Exception $e) {
        $this->conn->rollback();
        return ["success" => false, "error" => $e->getMessage()];
    }
}
```

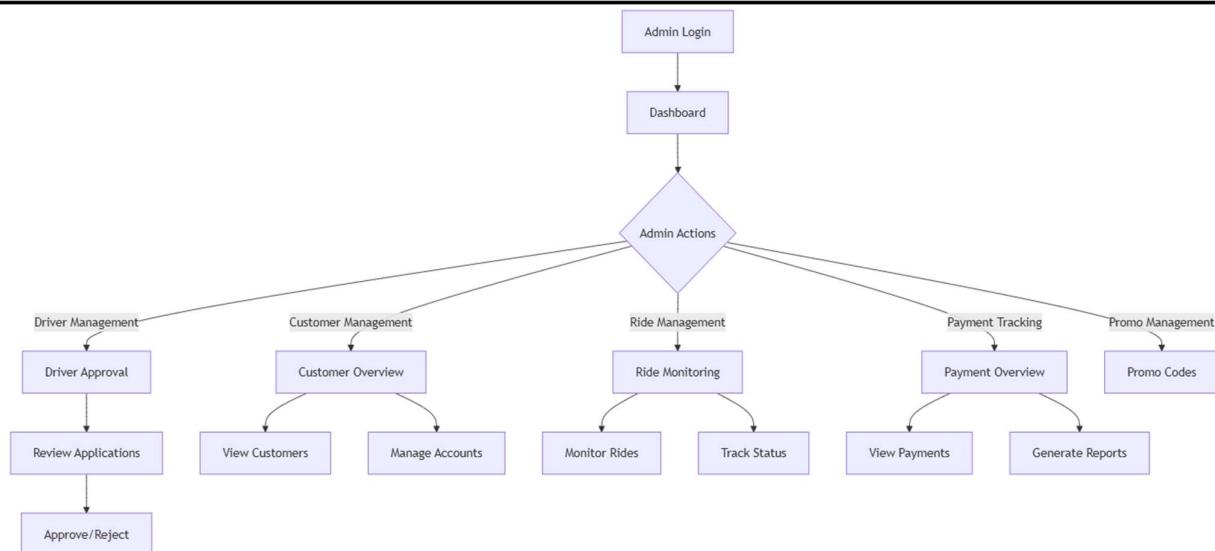
Admin Module

1. Module Overview

Purpose

The Admin Module serves as the central control system for the taxi booking platform, providing administrators with tools to:

- Manage and monitor all system users (drivers, customers)
- Oversee ride operations and payments
- Handle communication and notifications



2. Data Structures

Database Tables

1. Admins Table

```
CREATE TABLE admins (
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(255),
    email VARCHAR(255) UNIQUE,
    password VARCHAR(255),
    phone VARCHAR(20),
    profile_image VARCHAR(255),
    created_at TIMESTAMP
);
```

2. Driver Approval System

```
CREATE TABLE drivers (
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(255),
    email VARCHAR(255) UNIQUE,
    phone VARCHAR(20),
    id_number VARCHAR(50),
    id_image VARCHAR(255),
    status ENUM('pending', 'approved', 'rejected'),
    profile_image VARCHAR(255),
    created_at TIMESTAMP
);
```

3. Process Logic

Implementation Process

1. Dashboard Statistics

```
// AdminController.  
$promoCodesCount = count($promoCodes);  
$totalRides = count($rides);  
$totalEarnings = array_sum(array_column($payments, 'amount'));
```

```

$activeDrivers = count($drivers);
$registeredCustomers = count($customers);

2. Driver Approval System
case 'approve':
    $driverId = $data['driverId'];
    $approveResult = $driver->approveDriver($driverId);
    echo json_encode($approveResult);
    break;
case 'reject':
    $driverId = $data['driverId'];
    $rejectResult = $driver->rejectDriver($driverId);
    echo json_encode($rejectResult);
    break;

```

Messaging Module

1. Module Overview

Purpose

The Messaging Module facilitates real-time communication between different user types (Admin, Customer, and Driver) within the taxi booking system. It enables:

- Direct messaging between users
- Message status tracking (sent, delivered, read)
- Message history management
- Real-time notifications
- Multi-user communication

2. Data Structures

Database Tables

1. Messages Table

```

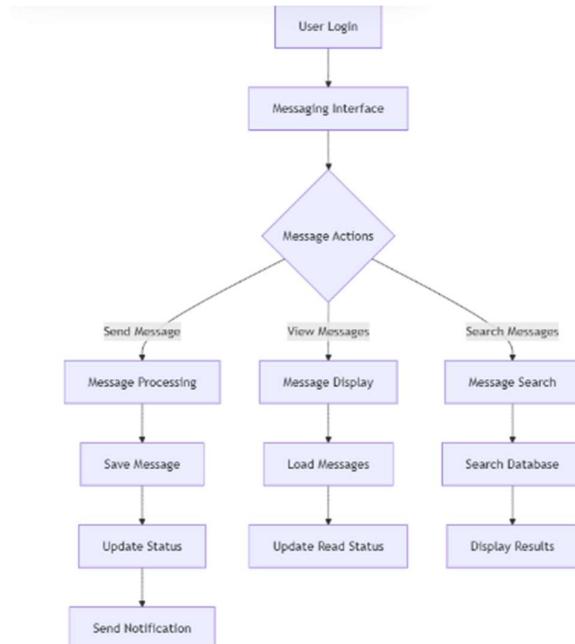
CREATE TABLE messages (
    id INT PRIMARY KEY AUTO_INCREMENT,
    sender_email VARCHAR(255),
    receiver_email VARCHAR(255),
    content TEXT,
    status ENUM('sent', 'delivered', 'read'),
    created_at TIMESTAMP DEFAULT
    CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT
    CURRENT_TIMESTAMP ON UPDATE
    CURRENT_TIMESTAMP,
    FOREIGN KEY (sender_email) REFERENCES users(email),
    FOREIGN KEY (receiver_email) REFERENCES users(email)
);

```

3. Process Logic

Implementation Process

1. Message Sending



```

public function sendMessage($senderEmail, $receiverEmail, $content) {
    $ = "INSERT INTO messages (sender_email, receiver_email, content, status)
        VALUES (?, ?, ?, 'sent')";
    $stmt = $this->conn->prepare($);
    $stmt->bind_param("sss", $senderEmail, $receiverEmail, $content);
    if ($stmt->execute()) {
        $this->sendNotification($receiverEmail);
        return ["success" => true, "message_id" => $this->conn->insert_id];
    }
    return ["success" => false, "error" => $stmt->error];
}

2. Message Retrieval
// Messages.
public function getMessagesByReceiver($receiverEmail) {
    $ = "SELECT m.,
        u1.name as sender_name, u1.profile_image as sender_image,
        u2.name as receiver_name, u2.profile_image as receiver_image
        FROM messages m
        JOIN users u1 ON m.sender_email = u1.email
        JOIN users u2 ON m.receiver_email = u2.email
        WHERE m.receiver_email = ?
        ORDER BY m.created_at DESC";
    $stmt = $this->conn->prepare($);
    $stmt->bind_param("s", $receiverEmail);
    $stmt->execute();
    return $stmt->get_result()->fetch_all(MYI_ASSOC);
}

```

Notification Module

1. Module Overview

Purpose

The Notification Module enables administrators to create, manage, and broadcast system-wide notifications to all users (customers and drivers). It provides:

- Centralized notification management

2. Data Structures

Database Tables

1. Notifications Table

```

CREATE TABLE notifications (
    id INT PRIMARY KEY AUTO_INCREMENT,
    user_id INT,
    message TEXT,
    status ENUM('unread', 'read'),

```

```

    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id)
);

```

3. Process Logic

Implementation Process

1. Notification Creation

```

// Notification.

public function addNotification($user_id, $message) {
    $query = "INSERT INTO " . $this->table_name . "(user_id, message, status, created_at) VALUES (?, ?, 'unread', NOW())";
    $stmt = $this->conn->prepare($query);
    if (!$stmt) {
        return ["success" => false, "error" => "Database error"];
    }
    $stmt->bind_param("is", $user_id, $message);
    if ($stmt->execute()) {
        return ["success" => true];
    }
    return ["success" => false, "error" => $stmt->error];
}

```

2. Notification Retrieval

```

public function getAllNotifications() {
    $query = "SELECT * FROM " . $this->table_name . " ORDER BY created_at DESC";
    $stmt = $this->conn->prepare($query);
    if (!$stmt) {
        return ["success" => false, "error" => "Database error"];
    }
    $stmt->execute();
    $result = $stmt->get_result();
    $notifications = $result->fetch_all(MYSQLI_ASSOC);
    return ["success" => true, "notifications" => $notifications];
}

```

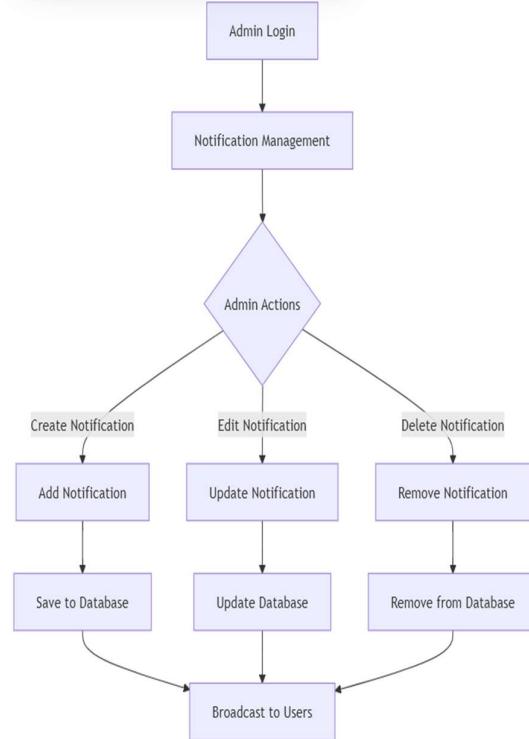
Ride Module

1. Module Overview

Purpose

The Ride Module manages the core taxi booking functionality, enabling customers to:

- Book rides with real-time fare calculation
- Track ride status
- Make payments
- Receive ride confirmations
- Verify ride completion with OTP



2. Data Structures

Database Tables

1. Rides Table

```
CREATE TABLE rides (
    id INT PRIMARY KEY AUTO_INCREMENT,
    customer_id INT,
    driver_id INT,
    pickup_location VARCHAR(255),
    dropoff_location VARCHAR(255),
    ride_status ENUM('pending', 'accepted', 'ongoing', 'completed', 'cancelled', 'paid'),
    fare DECIMAL(10,2),
    distance DECIMAL(10,2),
    rate DECIMAL(10,2),
    OTP INT,
    request_time TIMESTAMP,
    complete_time TIMESTAMP,
    FOREIGN KEY (customer_id) REFERENCES customers(id),
    FOREIGN KEY (driver_id) REFERENCES drivers(id)
);
```

3. Process Logic

Implementation Process

1. Location Selection and Route Calculation

```
function calculateRoute() {
    if (!marker1 || !marker2) return;
    const request = {
        origin: marker1.getPosition(),
        destination: marker2.getPosition(),
        travelMode: google.maps.TravelMode.DRIVING
    };
    directionsService.route(request, (result, status) => {
        if (status === 'OK') {
            const route = result.routes[0];
            const distance = route.legs[0].distance.value / 1000;
            const duration = route.legs[0].duration.text;
            document.getElementById('distance').value = distance.toFixed(2);
            document.getElementById('fare').value = (distance * 10).toFixed(2);
        }
    });
}
```

2. Ride Booking

```
public function bookRide($customer_id, $pickup_location, $dropoff_location, $fare, $distance, $rate) {
    $ = "INSERT INTO " . $this->table_name . "
        (customer_id, pickup_location, dropoff_location, ride_status, fare, request_time, distance, rate)
    VALUES (?, ?, ?, 'pending', ?, NOW(), ?, ?);
```

```

$stmt = $this->conn->prepare($);
$stmt->bind_param("issddi", $customer_id, $pickup_location, $dropoff_location, $fare, $distance,
$rate);
if ($stmt->execute()) {
    return ["success" => true, "ride_id" => $this->conn->insert_id];
}
return ["success" => false, "error" => "Failed to book ride"];
}

```

Payment Module

1. Module Overview

Purpose

The Payment Module manages all payment-related functionalities in the taxi booking system, including:

- Processing payments through Razorpay
- Managing promo code discounts
- Generating payment invoices

2. Data Structures

Database Tables

1. Payments Table	2. Promo Codes Table
CREATE TABLE payments (id INT PRIMARY KEY AUTO_INCREMENT, ride_id INT, driver_id INT, customer_id INT, amount DECIMAL(10,2), payment_id VARCHAR(255), payment_method VARCHAR(50), payment_status VARCHAR(50), payment_date TIMESTAMP, invoice VARCHAR(255), FOREIGN KEY (ride_id) REFERENCES rides(id), FOREIGN KEY (driver_id) REFERENCES drivers(id), FOREIGN KEY (customer_id) REFERENCES customers(id));	CREATE TABLE promo_codes (id INT PRIMARY KEY AUTO_INCREMENT, code VARCHAR(50) UNIQUE, discount_percentage DECIMAL(5,2), valid_from TIMESTAMP, valid_to TIMESTAMP, is_active BOOLEAN DEFAULT true, created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP);

3. Process Logic

Implementation Process

1. Payment Processing

```

// PaymentController.

public function savePayment($paymentData) {
    if (!$paymentData) {
        return ['success' => false, 'error' => 'No payment data provided'];
    }
    $result = $this->paymentModel->savePayment($paymentData);
    if ($result) {
        return ['success' => true, 'message' => 'Payment saved successfully'];
    }
}

```

```

    }
    return ['success' => false, 'error' => 'Failed to save payment'];
}

```

Blog Module

1. Module Overview

Purpose

The Blog Module enables users to share their experiences and insights about the taxi service, providing:

- User-generated content platform
- Experience sharing
- Community engagement
- Content discovery
- Search functionality

2. Data Structures

Database Tables

1. Blogs Table

```
CREATE TABLE blogs (
    id INT PRIMARY KEY AUTO_INCREMENT,
    title VARCHAR(255) NOT NULL,
    content TEXT NOT NULL,
    author VARCHAR(100) NOT NULL,
    created_at      TIMESTAMP      DEFAULT
    CURRENT_TIMESTAMP
);
```

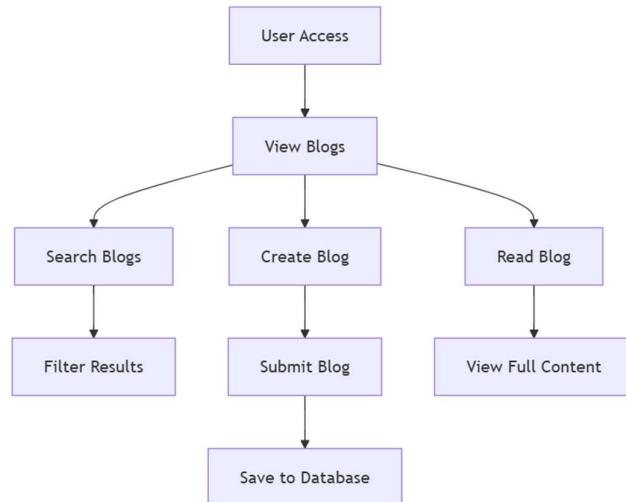
3. Process Logic

Implementation Process

1. Blog Creation

```
if ($action === 'createBlog') {
    $title = $_POST['title'];
    $content = $_POST['content'];
    $author = $_POST['author'];
    $result = $blog->createBlog($title,
    $content, $author);

    if ($result) {
        echo json_encode(['success' => true]);
    } else {
        echo json_encode(['success' => false, 'error' => 'Failed to create blog']);
    }
}
```



Source Code

Includes/header.php

```
<?php
session_start();
?>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>My Website</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet" crossorigin="anonymous">
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.3/font/bootstrap-
icons.min.css">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.5.1/css/all.min.css">
</head>
<body class="bg-light text-dark">
<header>
<div class="px-3 py-2 text-bg-dark border-bottom">
<div class="container">
<div class="d-flex flex-wrap align-items-center justify-content-between">
<a href="/" class="d-flex align-items-center my-2 my-lg-0 me-lg-auto text-white text-decoration-
none">
<i class="bi bi-bootstrap bi me-2" style="font-size: 24px;"></i>
<span>ETaxi</span>
</a>
<!-- Replace the existing login/signup and user section with this -->
<div class="d-flex align-items-center gap-3">
<!-- Theme Toggle Button -->
<button id="theme-toggle" class="btn btn-outline-light">
<i id="theme-icon" class="bi bi-moon"></i>
</button>
<?php if (!isset($_SESSION['role']) || empty($_SESSION['role'])): ?>
<div class="d-flex align-items-center">
<a href="login.php" class="btn btn-outline-light me-2">Login</a>
<a href="user_register.php" class="btn btn-outline-light">Sign Up</a>
</div>
<?php else: ?>
<div class="dropdown">
<button class="btn btn-outline-light dropdown-toggle d-flex align-items-center gap-2"
type="button">
```

```

        data-bs-toggle="dropdown"
        aria-expanded="false">
    <i class="bi bi-person-circle"></i>
    <span><?php echo isset($_SESSION['role']) ? ucfirst($_SESSION['role']) : 'User'; ?></span>
</button>
<ul class="dropdown-menu dropdown-menu-end shadow">
    <?php
    switch($_SESSION['role']) {
        case 'admin':
            echo '<li><a class="dropdown-item" href="views/admin/dashboard.php"><i class="bi bi-speedometer2 me-2"></i>Admin Dashboard</a></li>';
            break;
        case 'driver':
            echo '<li><a class="dropdown-item" href="views/driver/dashboard.php"><i class="bi bi-car-front me-2"></i>Driver Dashboard</a></li>';
            break;
        case 'customer':
            echo '<li><a class="dropdown-item" href="views/customer/dashboard.php"><i class="bi bi-person me-2"></i>Customer Dashboard</a></li>';
            break;
        default:
            echo '<li><a class="dropdown-item" href="index.php"><i class="bi bi-speedometer2 me-2"></i>Dashboard</a></li>';
    }
    ?>
    <li><hr class="dropdown-divider"></li>
    <li><a class="dropdown-item text-danger" href="views/logout.php"><i class="bi bi-box-arrow-right me-2"></i>Logout</a></li>
<div class="container">
    <header class="d-flex justify-content-center py-3">
        <ul class="nav nav-pills">
            <li class="nav-item"><a href="index.php" class="nav-link <?php echo basename($_SERVER['PHP_SELF']) == 'index.php' ? 'active' : '' ; ?>">Home</a></li>
            <li class="nav-item"><a href="earn.php" class="nav-link <?php echo basename($_SERVER['PHP_SELF']) == 'earn.php' ? 'active' : '' ; ?>">Earn</a></li>
            <li class="nav-item"><a href="stats.php" class="nav-link <?php echo basename($_SERVER['PHP_SELF']) == 'stats.php' ? 'active' : '' ; ?>">Stats</a></li>
            <li class="nav-item"><a href="about.php" class="nav-link <?php echo basename($_SERVER['PHP_SELF']) == 'about.php' ? 'active' : '' ; ?>">About</a></li>
            <li class="nav-item"><a href="help.php" class="nav-link <?php echo basename($_SERVER['PHP_SELF']) == 'help.php' ? 'active' : '' ; ?>">Help</a></li>
            <li class="nav-item"><a href="gallery.php" class="nav-link <?php echo basename($_SERVER['PHP_SELF']) == 'gallery.php' ? 'active' : '' ; ?>">Gallery</a></li>
            <li class="nav-item"><a href="blog.php" class="nav-link <?php echo basename($_SERVER['PHP_SELF']) == 'blog.php' ? 'active' : '' ; ?>">Blog</a></li>

```

```

<li class="nav-item"><a href="games.php" class="nav-link <?php echo
basename($_SERVER['PHP_SELF']) == 'blog.php' ? 'active' : ''; ?>">Blog</a></li>
</ul>
</header>
</div>
</header>

include/footer.php
<div class="container">
<footer class="d-flex flex-wrap justify-content-between align-items-center py-3 my-4 border-top">
<div class="col-md-4 d-flex align-items-center">
<a href="/" class="mb-3 me-2 mb-md-0 text-body-secondary text-decoration-none lh-1">
<svg class="bi" width="30" height="24"><use xlink:href="#bootstrap"></use></svg>
</a>
<span class="mb-3 mb-md-0 text-body-secondary">© 2024 Company, Inc</span>
</div>
<ul class="nav col-md-4 justify-content-end list-unstyled d-flex">
<li class="ms-3"><a class="text-body-secondary" href="#"><svg class="bi" width="24"
height="24"><use xlink:href="#twitter"></use></svg></a></li>
<li class="ms-3"><a class="text-body-secondary" href="#"><svg class="bi" width="24"
height="24"><use xlink:href="#instagram"></use></svg></a></li>
<li class="ms-3"><a class="text-body-secondary" href="#"><svg class="bi" width="24"
height="24"><use xlink:href="#facebook"></use></svg></a></li>
</ul>
</footer>
</div>
</body>
</html>

```

Index.php

```

<?php include 'includes/header.php'; ?>

<div class="container col-xxl-8 px-4 py-5">
<div class="row flex-lg-row-reverse align-items-center g-5 py-5">
<div class="col-10 col-sm-8 col-lg-6">
<div id="carouselExampleCaptions" class="carousel slide" data-bs-ride="carousel">
<div class="carousel-indicators">
<button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-to="0"
class="active" aria-current="true" aria-label="Slide 1"></button>
<button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-to="1" aria-
label="Slide 2"></button>
<button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-to="2" aria-
label="Slide 3"></button>
</div>
<div class="carousel-inner">

```

```
<div class="carousel-item active">

<div class="carousel-caption d-none d-md-block">
<h5>Professional Taxi Service</h5>
<p>Experience comfortable and reliable transportation with our professional drivers.</p>
</div>
</div>
<div class="carousel-item">

<div class="carousel-caption d-none d-md-block">
<h5>Modern Fleet</h5>
<p>Our fleet of modern vehicles ensures a safe and comfortable journey.</p>
</div>
</div>
<div class="carousel-item">

<div class="carousel-caption d-none d-md-block">
<h5>24/7 Availability</h5>
<p>Available round the clock for your transportation needs.</p>
</div>
</div>
</div>
<button class="carousel-control-prev" type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide="prev">
<span class="carousel-control-prev-icon" aria-hidden="true"></span>
<span class="visually-hidden">Previous</span>
</button>
<button class="carousel-control-next" type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide="next">
<span class="carousel-control-next-icon" aria-hidden="true"></span>
<span class="visually-hidden">Next</span>
</button>
</div>
<div class="col-lg-6">
<h1 class="display-5 fw-bold lh-1 mb-3">Your Trusted Taxi Service Partner</h1>
<p class="lead">Welcome to ETaxi, where reliability meets comfort. We provide professional taxi services with experienced drivers, modern vehicles, and competitive rates. Whether you need airport transfers, city tours, or corporate travel solutions, we've got you covered 24/7.</p>
<div class="d-grid gap-2 d-md-flex justify-content-md-start">
<a href="about.php" class="btn btn-primary btn-lg px-4 me-md-2">About Us</a>
<a href="#services" class="btn btn-outline-secondary btn-lg px-4">Our Services</a>
</div>
</div>
</div>
```

```
</div>
<div class="container px-4 py-5">
    <h2 class="pb-2 border-bottom">Why Choose Our Taxi Service</h2>

    <div class="row row-cols-1 row-cols-sm-2 g-4">
        <div class="col d-flex flex-column gap-2 text-center align-items-center justify-content-center">
            <div class="feature-icon-small d-flex align-items-center justify-content-center text-bg-primary bg-gradient"
                style="width: 60px; height: 60px; border-radius: 10px;">
                <i class="bi bi-clock-history flex-shrink-0" style="font-size: 28px; color: white;"></i>
            </div>
            <h4 class="fw-semibold mb-0 text-body-emphasis">24/7 Availability</h4>
            <p class="text-body-secondary">Round-the-clock service ensuring you can get a ride whenever you need it, day or night.</p>
        </div>

        <div class="col d-flex flex-column gap-2 text-center align-items-center justify-content-center">
            <div class="feature-icon-small d-flex align-items-center justify-content-center text-bg-primary bg-gradient"
                style="width: 60px; height: 60px; border-radius: 10px;">
                <i class="bi bi-shield-check flex-shrink-0" style="font-size: 28px; color: white;"></i>
            </div>
            <h4 class="fw-semibold mb-0 text-body-emphasis">Safe & Secure</h4>
            <p class="text-body-secondary">Professional licensed drivers, GPS tracking, and secure payment options for your peace of mind.</p>
        </div>

        <div class="col d-flex flex-column gap-2 text-center align-items-center justify-content-center">
            <div class="feature-icon-small d-flex align-items-center justify-content-center text-bg-primary bg-gradient"
                style="width: 60px; height: 60px; border-radius: 10px;">
                <i class="bi bi-cash-coin flex-shrink-0" style="font-size: 28px; color: white;"></i>
            </div>
            <h4 class="fw-semibold mb-0 text-body-emphasis">Competitive Rates</h4>
            <p class="text-body-secondary">Transparent pricing with no hidden fees. Get the best value for your money with our fair fare system.</p>
        </div>

        <div class="col d-flex flex-column gap-2 text-center align-items-center justify-content-center">
            <div class="feature-icon-small d-flex align-items-center justify-content-center text-bg-primary bg-gradient"
                style="width: 60px; height: 60px; border-radius: 10px;">
                <i class="bi bi-car-front-fill flex-shrink-0" style="font-size: 28px; color: white;"></i>
            </div>
            <h4 class="fw-semibold mb-0 text-body-emphasis">Modern Fleet</h4>
        </div>
    </div>
</div>
```

```
<p class="text-body-secondary">Well-maintained, comfortable vehicles equipped with  
modern amenities for a pleasant journey.</p>  
</div>  
</div>  
</div>  
  
<div class="container px-4 py-5" id="custom-cards">  
<h2 class="pb-2 border-bottom">Our Services Gallery</h2>  
  
<div class="row row-cols-1 row-cols-lg-3 align-items-stretch g-4 py-5">  
<div class="col">  
<div class="card card-cover h-100 overflow-hidden text-bg-dark rounded-4 shadow-lg"  
style="background-image: linear-gradient(rgba(0, 0, 0, 0.5), rgba(0, 0, 0, 0.5)),  
url('assets/gallery/im1.jpg'); background-size: cover; background-position: center;">  
<div class="d-flex flex-column h-100 p-5 pb-3 text-white text-shadow-1">  
<h3 class="pt-5 mt-5 mb-4 display-6 lh-1 fw-bold">Airport Transfers</h3>  
<ul class="d-flex list-unstyled mt-auto">  
<li class="me-auto">  
<div class="badge bg-primary">Premium Service</div>  
</li>  
<li class="d-flex align-items-center me-3">  
<i class="bi bi-geo-alt" style="font-size: 1em;"></i>  
<small class="ms-2">Airport</small>  
</li>  
<div class="col">  
<div class="card card-cover h-100 overflow-hidden text-bg-dark rounded-4 shadow-lg"  
style="background-image: linear-gradient(rgba(0, 0, 0, 0.5), rgba(0, 0, 0, 0.5)),  
url('assets/gallery/im17.jpg'); background-size: cover; background-position: center;">  
<div class="d-flex flex-column h-100 p-5 pb-3 text-white text-shadow-1">  
<h3 class="pt-5 mt-5 mb-4 display-6 lh-1 fw-bold">City Tours</h3>  
<ul class="d-flex list-unstyled mt-auto">  
<li class="me-auto">  
<div class="badge bg-success">Popular</div>  
</li>  
<li class="d-flex align-items-center me-3">  
<i class="bi bi-geo-alt" style="font-size: 1em;"></i>  
<small class="ms-2">City Wide</small>  
<div class="col">  
<div class="card card-cover h-100 overflow-hidden text-bg-dark rounded-4 shadow-lg"  
style="background-image: linear-gradient(rgba(0, 0, 0, 0.5), rgba(0, 0, 0, 0.5)),  
url('assets/gallery/im24.jpg'); background-size: cover; background-position: center;">  
<div class="d-flex flex-column h-100 p-5 pb-3 text-white text-shadow-1">  
<h3 class="pt-5 mt-5 mb-4 display-6 lh-1 fw-bold">Corporate Travel</h3>  
<ul class="d-flex list-unstyled mt-auto">  
<li class="me-auto">
```

```

        <div class="badge bg-info">Business Class</div>
    </li>
    <li class="d-flex align-items-center me-3">
        <i class="bi bi-geo-alt" style="font-size: 1em;"></i>
        <small class="ms-2">Business District</small>

<div class="container px-4 py-5">
    <h2 class="pb-2 border-bottom">Latest Blog Posts</h2>
    <?php
    require_once 'controllers/BlogController.php';

    if (isset($blogs) && $blogs['success']) {
        echo '<div class="row px-4 py-5">';
        foreach ($blogs['blogs'] as $blog) {
            ?>
            <div class="col-lg-4 mb-4">
                <div class="card h-100">
                    <div class="card-body">
                        <h2 class="card-title h4 fw-normal"><?php echo htmlspecialchars($blog['title']); ?></h2>
                        <p class="card-text"><?php echo substr(htmlspecialchars($blog['content']), 0, 150) . '...'; ?></p>
                        <div class="d-flex justify-content-between align-items-center">
                            <small class="text-muted">By <?php echo htmlspecialchars($blog['author']); ?></small>
                            <small class="text-muted"><?php echo date('M d, Y', strtotime($blog['created_at'])); ?></small>
                        </div>
                        <p class="mt-3"><a class="btn btn-primary" href="blog.php?id=<?php echo $blog['id']; ?>">Read More </a></p>
                    }
                    echo '</div>';
                } else {
                    echo '<div class="alert alert-info">No blog posts available at the moment.</div>';
                }
            ?>
        </div>

    <?php include 'includes/footer.php'; ?>

```

Login.php

```

<?php include 'includes/header.php'; ?>
    <!-- Role Selection Dropdown -->
    <div class="form-outline mb-4">
        <select class="form-select form-control-lg" name="role" id="roleSelect">
            <option value="customer">Login as Customer</option>

```

```

        <option value="driver">Login as Driver</option>
        <option value="admin" selected>Login as Admin</option>
    </select>
</div>
<!-- Email Input -->
<div class="form-outline mb-4" id="emailInput">
    <input type="email" name="email" id="typeEmailX-2" class="form-control form-control-lg" required />
    <label class="form-label" for="typeEmailX-2">Email</label>
    <button class="btn btn-primary btn-sm mt-2" type="button" id="sendOtpButtonEmail" style="display: none;">Send OTP</button>
</div>
<!-- Password Input -->
<div class="form-outline mb-3" id="passwordInput">
    <input type="password" name="password" id="typePasswordX-2" class="form-control form-control-lg" required />
    <label class="form-label" for="typePasswordX-2">Password</label>
</div>
<!-- Phone Input for Driver -->
<div class="form-outline mb-4" id="phoneInput" style="display: none;">
    <input type="text" name="phone" id="typePhoneX-2" class="form-control form-control-lg" />
    <label class="form-label" for="typePhoneX-2">Phone</label>
    <button class="btn btn-primary btn-sm mt-2" type="button" id="sendOtpButtonPhone" style="display: none;">Send OTP</button>
</div>
<!-- OTP Input -->
<div class="form-outline mb-3" id="otpInput" style="display: none;">
    <input type="text" name="otp" id="typeOtpX-2" class="form-control form-control-lg" />
    <label class="form-label" for="typeOtpX-2">OTP</label>
</div>
<!-- Forgot Password Link -->
<div class="mb-3 text-start">
    <a href="forgotpassword.php" class="text-muted">Forgot password?</a>
</div>
<!-- Create Account Link -->
<div class="mt-3">
    <p>Don't have an account? <a href="user_register.php">Create Account</a></p>
</div>
</section>
<script>
document.getElementById('sendOtpButtonEmail').addEventListener('click', function () {
    const email = document.getElementById('typeEmailX-2').value;
    if (email) {
        fetch('api/mail_otp.php', {
            method: 'POST',

```

```
        headers: { 'Content-Type': 'application/x-www-form-urlencoded' },
        body: new URLSearchParams({ email: email, role: 'customer' })
    }).then(response => response.json()).then(data => {
        if (data.success) {
            alert('OTP sent to your email.');
        } else {
            alert('Failed to send OTP. ' + data.message);
            location.reload();
        }
    });
} else {
    alert('Please enter a valid email.');
}
});

document.getElementById('sendOtpButtonPhone').addEventListener('click', function () {
    const phone = document.getElementById('typePhoneX-2').value;
    if (phone) {
        fetch('api/mobile_otp.php', {
            method: 'POST',
            headers: { 'Content-Type': 'application/x-www-form-urlencoded' },
            body: new URLSearchParams({ phone: phone })
        }).then(response => response.json()).then(data => {
            if (data.success) {
                alert('OTP sent to your phone.');
            } else {
                alert('Failed to send OTP. ' + data.message);
                location.reload();
            }
        });
    } else {
        alert('Please enter a valid phone number.');
    }
});
}

document.getElementById('verifyAndLoginButton').addEventListener('click', function () {
    const formData = new FormData(document.getElementById('loginForm'));
    fetch('controllers/AuthController.php', {
        method: 'POST',
        body: formData
    }).then(response => response.json()).then(data => {
        if (data.success) {
            alert('Login Successful');
            window.location.href = data.role === 'admin' ? 'views/admin/dashboard.php' : data.role ===
            'driver' ? 'views/driver/dashboard.php' : 'views/customer/dashboard.php';
        } else {
            alert('Login Failed: ' + data.error);
        }
    });
});
```

```

        location.reload();
    }
});
});
if (data.success) {
    new bootstrap.Modal(document.getElementById('successModal')).show();
    setTimeout(() => {
        let redirectUrl = {
            admin: 'views/admin/dashboard.php',
            driver: 'views/driver/dashboard.php',
            customer: 'views/customer/dashboard.php'
        }[data.role] || 'index.php';
        window.location.href = redirectUrl;
    }, 2000);
} else {
    showErrorModal(data.error);
}
} catch (error) {
    console.error('Error parsing JSON:', error);
    showErrorModal('An error occurred. Please try again later.');
}
})
.catch(error => {
    console.error('Fetch error:', error);
    showErrorModal('An error occurred. Please check your internet connection.');
});
});
function showErrorModal(message) {
    document.getElementById('errorModal').querySelector('.modal-body').textContent = message;
    new bootstrap.Modal(document.getElementById('errorModal')).show();
}
</script>
<?php include 'includes/footer.php'; ?>
User_register.php
<?php include 'includes/header.php'; ?>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
<div class="container mt-5 d-flex justify-content-center">
    <div class="card shadow-lg" style="width: 100%; max-width: 500px;">
        <div class="card-header bg-primary text-white text-center">
            <h2>User Registration</h2>
        </div>
        <div class="card-body">
            <form id="registrationForm" action="controllers/AuthController.php?action=register" method="POST">

```

```

<div class="mb-3">
    <label for="name" class="form-label">Name:</label>
    <input type="text" id="name" name="name" class="form-control" required>
</div>
<div class="mb-3">
    <label for="email" class="form-label">Email:</label>
    <input type="email" id="email" name="email" class="form-control" required>
</div>
<div class="mb-3">
    <label for="phone" class="form-label">Phone:</label>
    <div class="input-group">
        <input type="text" id="phone" name="phone" class="form-control" required>
        <button type="button" id="sendOtpButton" class="btn btn-primary">Send
OTP</button>
    <div class="mb-3">
        <label for="otp" class="form-label">OTP:</label>
        <div class="input-group">
            <input type="text" id="otp" name="otp" class="form-control" required>
            <button type="button" id="verifyOtpButton" class="btn btn-success">Verify
OTP</button>
        </div>
    </div>
    <div class="mb-3">
        <label for="password" class="form-label">Password:</label>
        <input type="password" id="password" name="password" class="form-control" required>
    </div>
    <button type="submit" id="registerButton" class="btn btn-primary w-100"
disabled>Register</button>
</form>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
<script>
let isOtpVerified = false;
document.getElementById('sendOtpButton').addEventListener('click', function() {
    const phone = document.getElementById('phone').value;
    if (!phone) {
        alert('Please enter your phone number.');
        return;
    }
    fetch('api/mobile_otp.php', {
        method: 'POST',
        headers: { 'Content-Type': 'application/x-www-form-urlencoded' },
        body: new URLSearchParams({ phone })
    })
    .then(response => response.json())
})
```

```
.then(data => {
    if (data.success) {
        alert('OTP sent successfully!');
    } else {
        showErrorModal('Failed to send OTP: ' + data.message);
    }
})
.catch(error => showErrorModal('Error: ' + error.message));
});

document.getElementById('verifyOtpButton').addEventListener('click', function() {
    const phone = document.getElementById('phone').value;
    const otp = document.getElementById('otp').value;
    if (!phone || !otp) {
        alert('Please enter your phone number and OTP.');
        return;
    }
    fetch('api/verify_mobile_otp.php', {
        method: 'POST',
        headers: { 'Content-Type': 'application/x-www-form-urlencoded' },
        body: new URLSearchParams({ phone, otp })
    })
    .then(response => response.json())
    .then(data => {
        if (data.success) {
            alert('OTP verified successfully!');
            isOtpVerified = true;
            document.getElementById('registerButton').disabled = false;
        } else {
            showErrorModal('Failed to verify OTP: ' + data.message);
            isOtpVerified = false;
            document.getElementById('registerButton').disabled = true;
        }
    })
    .catch(error => showErrorModal('Error: ' + error.message));
});

document.getElementById('registrationForm').addEventListener('submit', function(event) {
    event.preventDefault();
    if (!isOtpVerified) {
        showErrorModal('Please verify your OTP before submitting.');
        return;
    }
    const formData = new FormData(this);
    fetch(this.action, {
        method: 'POST',
        body: formData
    })
    .then(response => response.json())
    .then(data => {
        if (data.success) {
            alert('Registration successful!');
            window.location.href = 'success.html';
        } else {
            showErrorModal('Registration failed: ' + data.message);
        }
    })
    .catch(error => showErrorModal('Error: ' + error.message));
});
```

```

        })
        .then(response => response.json())
        .then(data => {
            if (data.success) {
                const successModal = new bootstrap.Modal(document.getElementById('successModal'));
                successModal.show();
                setTimeout(() => {
                    window.location.href = 'login.php';
                }, 3000);
            } else {
                showErrorModal(data.error || 'Registration failed. Please try again.');
            }
        })
        .catch(error => showErrorModal('Error: ' + error.message));
    });
}

function showErrorModal(message) {
    document.getElementById('errorMessage').textContent = message;
    const errorModal = new bootstrap.Modal(document.getElementById('errorModal'));
    errorModal.show();
}

```

</script>

<?php include 'includes/footer.php'; ?>

Gallery.php

```

<?php include 'includes/header.php'; ?>

<?php
// Add this PHP code at the top to get all images
$gallery_path = 'assets/gallery/';
$images = glob($gallery_path . "*.{jpg,jpeg,png,gif}", GLOB_BRACE);
$first_image = !empty($images) ? $images[0] : '';
?>

<div class="gallery-container">
    <!-- Main large image display -->
    <div class="main-image">
        
    </div>

    <!-- Horizontal slider -->
    <div class="horizontal-slider">
        <div class="slider-track">
            <?php foreach($images as $image); ?>
            <div class="slide">
                
            </div>
        </div>
    </div>
</div>

```

```
</div>
<?php endforeach; ?>
</div>
</div>

<!-- Vertical slider --&gt;
&lt;div class="vertical-slider"&gt;
    &lt;div class="slider-track-vertical"&gt;
        &lt;?php foreach($images as $image): ?&gt;
            &lt;div class="slide"&gt;
                &lt;img src="&lt;?php echo $image; ?&gt;" onclick="changelImage(this.src)" alt="Gallery Image"&gt;
            &lt;/div&gt;
        &lt;?php endforeach; ?&gt;
    &lt;/div&gt;
&lt;/div&gt;
&lt;script&gt;
function changelImage(src) {
    document.getElementById('featured-image').src = src;
}

// Modified script with auto-update for vertical slider
document.addEventListener('DOMContentLoaded', function() {
    const hTrack = document.querySelector('.slider-track');
    const vTrack = document.querySelector('.slider-track-vertical');

    const hClones = hTrack.innerHTML;
    const vClones = vTrack.innerHTML;

    hTrack.innerHTML += hClones;
    vTrack.innerHTML += vClones;

    // Auto update vertical slider every 3 seconds
    const verticalImages = document.querySelectorAll('.slider-track-vertical .slide img');
    let currentIndex = 0;

    setInterval(() =&gt; {
        currentIndex = (currentIndex + 1) % verticalImages.length;
        const nextImage = verticalImages[currentIndex].src;
        changelImage(nextImage);
    }, 3000);
});

&lt;/script&gt;

&lt;?php include 'includes/footer.php'; ?&gt;</pre>
```

Driver_register.php

```
<?php include 'includes/header.php'; ?>
<section class="h-100 h-custom gradient-custom-2">
    <div class="p-5">
        <h3 class="fw-normal mb-5" style="color: #4835d4;">Driver Registration</h3>
        <form id="registrationForm" action="controllers/AuthController.php" method="POST"
        enctype="multipart/form-data">
            <input type="hidden" name="action" value="register_driver">
            <div class="mb-4 pb-2">
                <div data-mdb-input-init class="form-outline">
                    <input type="text" id="name" name="name" class="form-control form-control-lg"
required />
                    <label class="form-label" for="name">Name</label>
                </div>
            </div>
            <div class="mb-4 pb-2">
                <div data-mdb-input-init class="form-outline">
                    <input type="email" id="email" name="email" class="form-control form-control-lg"
required />
                    <label class="form-label" for="email">Email</label>
                </div>
            </div>
            <div class="mb-4 pb-2">
                <div data-mdb-input-init class="form-outline">
                    <input type="password" id="password" name="password" class="form-control form-control-lg"
required />
                    <label class="form-label" for="password">Password</label>
                </div>
            <div class="mb-4 pb-2">
                <div data-mdb-input-init class="form-outline">
                    <input type="text" id="id_number" name="id_number" class="form-control form-control-lg"
required />
                    <label class="form-label" for="id_number">Aadhaar Number</label>
                </div>
            </div>
            <div class="mb-4 pb-2">
                <div data-mdb-input-init class="form-outline">
                    <input type="file" id="id_image" name="id_image" class="form-control form-control-
lg" required />
                    <label class="form-label" for="id_image">Aadhaar Proof</label>
                </div>
            </div>
            <div class="mb-4 pb-2">
                <div data-mdb-input-init class="form-outline">
```

```
<input type="file" id="profile_image" name="profile_image" class="form-control form-control-lg" required />
    <label class="form-label" for="profile_image">Profile Image</label>
    <div class="col-lg-6">
<input type="file" id="license_image" name="license_image" class="form-control form-control-lg" required />
    <label class="form-label" for="license_image">License Proof</label>
        <input type="text" id="otp" name="otp" class="form-control form-control-lg" required />
            <label class="form-label" for="otp">OTP</label>
        </div>
    </div>
    <button type="button" onclick="sendOTP()" class="btn btn-primary btn-lg">Send OTP</button>
    <button type="button" onclick="verifyOTP()" class="btn btn-secondary btn-lg">Verify OTP & Submit</button>
</form>
</section>
<?php include 'includes/footer.php'; ?>
<script>
function sendOTP() {
    const phone = document.getElementById('phone').value;
    fetch('api/mobile_otp.php', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/x-www-form-urlencoded'
        },
        body: `phone=${phone}`
    })
    .then(response => response.json())
    .then(data => {
        if (data.success) {
            alert('OTP sent to your phone');
        } else {
            alert('Failed to send OTP');
        }
    });
}
function verifyOTP() {
    const enteredOTP = document.getElementById('otp').value;
    const phone = document.getElementById('phone').value;
    fetch('api/verify_mobile_otp.php', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/x-www-form-urlencoded'
        }
    })
    .then(response => response.json())
    .then(data => {
        if (data.success) {
            alert('OTP verified successfully');
        } else {
            alert('Failed to verify OTP');
        }
    });
}
```

```

        },
        body: `otp=${enteredOTP}&phone=${phone}`
    })
    .then(response => response.json())
    .then(data => {
        if (data.success) {
            const form = document.getElementById('registrationForm');
            if (form.checkValidity()) {
                let formData = new FormData(this);
                let enteredOtp = formData.get('otp');
                let storedOtp = localStorage.getItem('otp');
                if (enteredOtp !== storedOtp) {
                    document.getElementById("errorMessage").innerText = "Invalid OTP. Please try again.";
                    let errorModal = new bootstrap.Modal(document.getElementById('errorModal'));
                    errorModal.show();
                    return;
                }
                fetch('controllers/AuthController.php', {
                    method: 'POST',
                    body: formData
                })
                .then(response => response.json())
                .then(data => {
                    if (data.success) {
                        let successModal = new bootstrap.Modal(document.getElementById('successModal'));
                        successModal.show();
                        setTimeout(() => { window.location.href = "login.php"; }, 2000);
                    } else {
                        document.getElementById("errorMessage").innerText = data.error;
                        let errorModal = new bootstrap.Modal(document.getElementById('errorModal'));
                        errorModal.show();
                    }
                })
            }
        }
    })
</script>

```

Report.php

```

<?php
class Review {
    private $conn;
    private $table_name = "reviews";

    public function __construct($db) {
        $this->conn = $db;
    }

    public function submitReview($data) {

```

```

$query = "INSERT INTO " . $this->table_name . " (ride_id, customer_id, driver_id, rating, review,
created_at) VALUES (?, ?, ?, ?, ?, ?)";

$stmt = $this->conn->prepare($query);

$stmt->bind_param('iiisss', $data['ride_id'], $data['customer_id'], $data['driver_id'],
$data['rating'], $data['review'], $data['created_at']);

if ($stmt->execute()) {
    return ['success' => true];
} else {
    return ['success' => false, 'error' => $stmt->error];
}
}

?>

```

Payment.php

```

<?php
class Payment {
    private $conn;
    private $table_name = "payments";

    public function __construct($db) {
        $this->conn = $db;
    }

    public function getAllPayments() {
        $sql = "SELECT p.id, p.ride_id, p.amount, p.payment_method, p.payment_status,
p.payment_date,
            c.name as customer_name, d.name as driver_name
        FROM " . $this->table_name . " p
        JOIN customers c ON p.customer_id = c.id
        JOIN rides r ON p.ride_id = r.id
        JOIN drivers d ON r.driver_id = d.id";
        $stmt = $this->conn->prepare($sql);
        if (!$stmt) {
            error_log("Prepare failed: " . $this->conn->error);
            return ["success" => false, "error" => "Database error"];
        }
        $stmt->execute();
        $result = $stmt->get_result();
        $payments = $result->fetch_all(MYSQLI_ASSOC);

        return ["success" => true, "payments" => $payments];
    }
}

```

```

public function getTotalEarningsByDriver($driver_id) {
    $sql = "SELECT SUM(amount) as totalEarnings FROM " . $this->table_name . " WHERE driver_id = ?";
    $stmt = $this->conn->prepare($sql);
    if (!$stmt) {
        error_log("Prepare failed: " . $this->conn->error);
        return ["success" => false, "error" => "Database error"];
    }
    $stmt->bind_param("i", $driver_id);
    $stmt->execute();
    $result = $stmt->get_result();
    $totalEarnings = $result->fetch_assoc()['totalEarnings'];

    return ["success" => true, "totalEarnings" => $totalEarnings];
}

public function getTotalPaymentsByCustomer($customer_id) {
    $sql = "SELECT SUM(amount) as totalPayments FROM " . $this->table_name . " WHERE customer_id = ?";
    $stmt = $this->conn->prepare($sql);
    if (!$stmt) {
        error_log("Prepare failed: " . $this->conn->error);
        return 0;
    }
    $stmt->bind_param("i", $customer_id);
    $stmt->execute();
    $result = $stmt->get_result();
    $row = $result->fetch_assoc();
    return $row['totalPayments'] ?? 0;
}

public function savePayment($paymentData) {
    $sql = "INSERT INTO " . $this->table_name . "
        (ride_id, driver_id, customer_id, amount, payment_id, payment_method, payment_status,
        payment_date)
        VALUES (?, ?, ?, ?, ?, ?, ?, ?)";

    $stmt = $this->conn->prepare($sql);
    if (!$stmt) {
        error_log("Prepare failed: " . $this->conn->error);
        return ["success" => false, "error" => "Prepare failed: " . $this->conn->error];
    }

    $stmt->bind_param("iiidssss",

```

```

$paymentData['ride_id'],
$paymentData['driver_id'],
$paymentData['customer_id'],
$paymentData['amount'],
$paymentData['payment_id'],
$paymentData['payment_method'],
$paymentData['payment_status'],
$paymentData['payment_date']
);

if ($stmt->execute()) {
    return ["success" => true];
} else {
    return ["success" => false, "error" => "Execute failed: " . $stmt->error];
}
}

public function getAllPaymentsByCustomer($customer_id) {
    $sql = "SELECT * FROM " . $this->table_name . " WHERE customer_id = ?";
    $stmt = $this->conn->prepare($sql);
    if (!$stmt) {
        error_log("Prepare failed: " . $this->conn->error);
        return ["success" => false, "error" => "Database error"];
    }
    $stmt->bind_param("i", $customer_id);
    $stmt->execute();
    $result = $stmt->get_result();
    $payments = $result->fetch_all(MYSQLI_ASSOC);
    return ["success" => true, "payments" => $payments];
}

public function getPaymentByRideId($rideId) {
    $sql = "SELECT * FROM " . $this->table_name . " WHERE ride_id = ?";
    $stmt = $this->conn->prepare($sql);
    if (!$stmt) {
        error_log("Prepare failed: " . $this->conn->error);
        return ["success" => false, "error" => "Database error"];
    }
    $stmt->bind_param("i", $rideId);
    $stmt->execute();
    $result = $stmt->get_result();
    $payment = $result->fetch_assoc();

    if (!$payment) {
        return ["success" => false, "error" => "Payment not found"];
    }
}

```

```

    }

    return ["success" => true, "payment" => $payment];
}

public function updateInvoicePath($paymentId, $invoicePath) {
    $sql = "UPDATE " . $this->table_name . " SET invoice = ? WHERE id = ?";
    $stmt = $this->conn->prepare($sql);
    if (!$stmt) {
        error_log("Prepare failed: " . $this->conn->error);
        return ["success" => false, "error" => "Database error"];
    }
    $stmt->bind_param("si", $invoicePath, $paymentId);
    if ($stmt->execute()) {
        return ["success" => true];
    } else {
        return ["success" => false, "error" => $stmt->error];
    }
}

public function getDriverEarnings($driver_id) {
    try {
        $sql = "SELECT p.id, p.ride_id, p.driver_id, p.customer_id, p.payment_id,
                p.amount, p.payment_method, p.payment_status, p.payment_date, p.invoice
                FROM payments p
                WHERE p.driver_id = ?
                ORDER BY p.payment_date DESC";

        $stmt = $this->conn->prepare($sql);
        if (!$stmt) {
            error_log("Prepare failed: " . $this->conn->error);
            return ["success" => false, "error" => "Database error"];
        }

        $stmt->bind_param("i", $driver_id);
        $stmt->execute();
        $result = $stmt->get_result();
        $earnings = $result->fetch_all(MYSQLI_ASSOC);

        return [
            "success" => true,
            "earnings" => $earnings
        ];
    } catch (Exception $e) {

```

```

        error_log("Error fetching driver earnings: " . $e->getMessage());
        return [
            "success" => false,
            "error" => "Failed to fetch earnings"
        ];
    }
}
?>

```

Blog.php

```

<?php
require_once 'controllers/BlogController.php';
include 'includes/header.php';
?>
<div class="container">
    <h1 class="mt-4">Blogs</h1>
    <input type="text" id="searchKeyword" class="form-control mt-3" placeholder="Search blogs...">
    <?php
    $blogs = $blog->getAllBlogs();
    $items_per_page = 6;
    $current_page = isset($_GET['page']) ? (int)$_GET['page'] : 1;
    $total_items = count($blogs['blogs']);
    $total_pages = ceil($total_items / $items_per_page);
    $offset = ($current_page - 1) * $items_per_page;
    $current_blogs = array_slice($blogs['blogs'], $offset, $items_per_page);
    if (!empty($blogs['blogs'])): ?>
        <div class="row mt-4" id="blogsContainer">
            <?php foreach ($current_blogs as $blog): ?>
                <div class="col-md-4 mb-4 blog-card">
                    <div class="card h-100 shadow-sm">
                        <div class="card-body">
                            <div class="d-flex justify-content-center align-items-center rounded-circle bg-primary
text-white mb-3" style="width: 50px; height: 50px; font-size: 24px;">
                                <?php echo strtoupper(substr($blog['author'], 0, 1)); ?>
                            </div>
                            <h5 class="card-title"><?php echo htmlspecialchars($blog['title']); ?></h5>
                            <p class="card-text text-truncate" style="max-height: 100px; overflow:
hidden;"><?php echo htmlspecialchars(substr($blog['content'], 0, 150)) . '...'; ?></p>
                            <div class="card-footer bg-transparent border-0 text-muted small">
                                <div>By <?php echo htmlspecialchars($blog['author']); ?></div>
                                <div><?php echo htmlspecialchars($blog['created_at']); ?></div>
                            <?php endforeach; ?>
                        </div>
                    <!-- Pagination -->

```

```

<nav aria-label="Blog pagination" class="mt-4">
    <ul class="pagination justify-content-center">
        <li class="page-item <?php echo $current_page <= 1 ? 'disabled' : ''; ?>">
            <a class="page-link" href="?page=<?php echo $current_page - 1; ?>" tabindex="-1">Previous</a>
        </li>
        <?php for($i = 1; $i <= $total_pages; $i++): ?>
            <li class="page-item <?php echo $current_page == $i ? 'active' : ''; ?>">
                <a class="page-link" href="?page=<?php echo $i; ?>><?php echo $i; ?></a>
            </li>
        <?php endfor; ?>
        <li class="page-item <?php echo $current_page >= $total_pages ? 'disabled' : ''; ?>">
            <a class="page-link" href="?page=<?php echo $current_page + 1; ?>>Next</a>
        <?php else: ?>
            <p class="text-center mt-4">No blogs found.</p>
        <?php endif; ?>
    <?php include 'includes/footer.php'; ?>

```

Forgotpassword.php

```

<?php include 'includes/header.php'; ?>
    <form id="forgotPasswordForm" action="api/send_reset_otp.php" method="POST">
        <div class="row gy-3 overflow-hidden">
            <div class="col-12">
                <div class="form-floating mb-3">
                    <input type="email" class="form-control" name="email" id="email"
                        placeholder="name@example.com" required>
                    <label for="email" class="form-label">Email</label>
                </div>
            </div>
            <div class="col-12">
                <div class="form-floating mb-3">
                    <select class="form-select" name="role" id="role" required>
                        <option value="customer">Customer</option>
                        <option value="driver">Driver</option>
                        <option value="admin">Admin</option>
                    </select>
                    <a href="login.php" class="link-secondary text-decoration-none">Login</a>
                    <a href="register.php" class="link-secondary text-decoration-none">Register</a>
                </div>
            </div>
        </div>
    </form>
    <?php include 'includes/footer.php'; ?>

```

Verify_reset_otp.php

```

<?php include 'includes/header.php'; ?>
    <h2 class="h3">Verify OTP</h2>
    <h3 class="fs-6 fw-normal text-secondary m-0">
        Enter the OTP sent to your email to reset your password.
    </h3>

```

```

        </div>
        <form id="verifyOtpForm" action="api/verify_reset_otp.php" method="POST">
            <div class="row gy-3 overflow-hidden">
                <div class="col-12">
                    <div class="form-floating mb-3">
                        <input type="text" class="form-control" name="otp" id="otp" placeholder="Enter OTP" required>
                        <label for="otp" class="form-label">OTP</label>
                    <div class="col-12">
                        <div class="form-floating mb-3">
                            <input type="password" class="form-control" name="password" id="password" placeholder="New Password" required>
                            <label for="password" class="form-label">New Password</label>
                        <div class="col-12">
                            <div class="form-floating mb-3">
                                <input type="password" class="form-control" name="confirm_password" id="confirm_password" placeholder="Confirm Password" required>
                                <label for="confirm_password" class="form-label">Confirm Password</label>
                            <div class="col-12">
                                <div class="d-grid">
                                    <button class="btn btn-primary btn-lg" type="submit">Reset Password</button>
                                </div>
                            </div>
                            <a href="login.php" class="link-secondary text-decoration-none">Login</a>
                            <a href="register.php" class="link-secondary text-decoration-none">Register</a>
                        </div>
                    </div>
                </div>
            </div>
        </form>
        <a href="#" class="link-secondary text-decoration-none">Forgot Password?</a>
    </div>
</section>
<?php include 'includes/footer.php'; ?>

```

Stats.php

```

<?php include 'includes/header.php'; ?>
<?php
include 'controllers/StatisticsController.php';
$controller = new StatisticsController();
$stats = $controller->showStats();
?>
<div class="container mt-5">
    <div class="row">
        <div class="col-md-3">
            <div class="card text-white bg-primary mb-3">
                <div class="card-header">Total Drivers</div>
                <div class="card-body">
                    <h5 class="card-title"><?php echo $stats['totalDrivers']; ?></h5>
                <div class="col-md-3">
                    <div class="card text-white bg-success mb-3">
                        <div class="card-header">Total Payments</div>
                        <div class="card-body">
                            <h5 class="card-title"><?php echo $stats['totalPayments']; ?></h5>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>

```

```

        </div>
    <div class="card text-white bg-danger mb-3">
        <div class="card-header">Total Kilometers</div>
        <div class="card-body">
            <h5 class="card-title"><?php echo $stats['totalKilometers']; ?></h5>
        <div class="col-md-3">
            <div class="card text-white bg-warning mb-3">
        <?php include 'includes/footer.php'; ?>
View/dashboard_layout.php:
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Dashboard</title>
    <!-- Bootstrap CSS -->
    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet">

    <link href="https://cdnjs.cloudflare.com/ajax/libs/bootstrap-icons/1.10.5/font/bootstrap-
icons.min.css" rel="stylesheet">
</style>
</head>
<body class="light-mode">
    <div id="welcomePopup" class="welcome-popup">
        <h4>Welcome <?php echo htmlspecialchars($adminData['name']); ?></h4>
        <p>Managing your application made easy.</p>
    </div>
    <!-- Navbar -->
    <nav class="navbar navbar-expand-lg">
        <div class="container-fluid">
            <button id="openBtn" class="btn btn-outline-primary me-2" onclick="openNav()">
                <i class="bi bi-list"></i>
            </button>
            <a class="navbar-brand mx-auto" href="#"><?php echo htmlspecialchars($adminData['name']); ?> Dashboard</a>
            <div class="d-flex align-items-center">
                <a class="nav-link text-white ms-4" href="#" data-link="messages.php"><i class="bi bi-
envelope"></i></a>
                <a class="nav-link text-white ms-4" href="#" data-link="notifications.php"><i class="bi bi-
bell"></i></a>
                <div class="dropdown">
                    <a class="nav-link dropdown-toggle text-white ms-4" href="#" id="profileDropdown"
role="button" data-toggle="dropdown">

```

```

        
    </a>
    <ul class="dropdown-menu dropdown-menu-right">
        <li><a class="dropdown-item" href="#" data-link="edit-profile.php">Edit Profile</a></li>
        <li><a class="dropdown-item" href="../logout.php">Logout</a></li>
    </ul>
</div>
<button id="theme-toggle" class="btn btn-outline-primary ms-2">
    <i id="theme-icon" class="bi bi-moon"></i>
</button>
<button id="fullscreen-toggle" class="btn btn-outline-light ms-2">
    <i class="bi bi-arrows-fullscreen"></i>
</nav>
<div id="mySidenav" class="sidenav">
    <span class="closebtn" onclick="closeNav()">&times;</span>
    <?php
        foreach ($options as $item) {
            if (empty($item['subItems'])) {
                echo '<a href="#" data-link="" . $item['link'] . "">' . $item['name'] . '</a>';
            } else {
                $submenuld = "submenu-" . preg_replace('/\s+/', '-', strtolower($item['name'])); // Unique ID for submenu
                echo '<a href="#" . $submenuld . "" class="dropdown-toggle" data-toggle="collapse">' .
                $item['name'] . '</a>';
                echo '<div id="" . $submenuld . "" class="collapse pl-3">';
                foreach ($item['subItems'] as $subItem) {
                    echo '<a class="dropdown-item text-light bg-dark" href="#" data-link="" . $subItem['link'] . "">' .
                    $subItem['name'] . '</a>';
                }
                echo '</div>'}
            ?>
        </div>
<div id="main">
    <div class="container-fluid p-0">
        <iframe id="contentFrame" src="home.php"></iframe>
    </div>
</div>
<script>
    function openNav() {
        document.getElementById("mySidenav").style.width = "250px";
    }
    function closeNav() {
        document.getElementById("mySidenav").style.width = "0";
    }

```

```

window.onload = function() {
    let popup = document.getElementById("welcomePopup");
    popup.style.display = "block";
    setTimeout(() => { popup.style.display = "none"; }, 5000);
    const links = document.querySelectorAll('a[data-link]');
    links.forEach(link => {
        link.addEventListener('click', function(event) {
            event.preventDefault();
            const url = this.getAttribute('data-link');
            const iframe = document.getElementById('contentFrame');
            if (iframe.src.endsWith(url)) {
                iframe.contentWindow.location.reload();
            } else {
                loadContent(url);
            }
        });
    });
}

const themeToggleBtn = document.getElementById("theme-toggle");
const themeIcon = document.getElementById("theme-icon");
const body = document.body;
const iframe = document.getElementById("contentFrame");
document.getElementById("fullscreen-toggle").addEventListener("click", () => {
    if (!document.fullscreenElement) {
        document.documentElement.requestFullscreen();
    } else {
        document.exitFullscreen();
    }
});
</script>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</body>
</html>

```

View/admin:

dashboard.php

```

<?php
session_start();
$adminData = $_SESSION['admin'];

```



```

<div class="card text-white bg-warning mb-3">
  <div class="card-header">Active Drivers</div>
  <div class="card-body">
    <h4 class="card-title"><?= $activeDrivers ?></h4>
<div class="row">
  <div class="col-md-4">
    <div class="card text-white bg-info mb-3">
      <div class="card-header">Registered Customers</div>
      <div class="card-body">
        <h4 class="card-title"><?= $registeredCustomers ?></h4>
      <!-- Recent Rides -->
      <div class="card mt-4 scrollable-table">
        <div class="card-header">
          <h4>Recent Rides</h4>
        </div>
        <div class="card-body">
          <div class="table-responsive">
            <table class="table table-striped ">
              <thead>
                <tr>
                  <th>Ride ID</th>
                  <th>Driver</th>
                  <th>Customer</th>
                  <th>Pickup</th>
                  <th>Drop-off</th>
                  <th>Status</th>
                  <th>Earnings</th>
                </tr>
              </thead>
              <tbody>
                <?php foreach ($rides as $ride): ?>
                <tr>
                  <td>#<?= $ride['id'] ?></td>
                  <td><?= $ride['driver_id'] ?></td>
                  <td><?= $ride['customer_id'] ?></td>
                  <td><?= $ride['pickup_location'] ?></td>
                  <td><?= $ride['dropoff_location'] ?></td>
                  <td><span class="badge bg-<?= $ride['ride_status'] == 'completed' ? 'success' : ($ride['ride_status'] == 'ongoing' ? 'primary' : 'danger') ?>"><?= ucfirst($ride['ride_status']) ?></span></td>
                  <td>$<?= $ride['fare'] ?></td>
                </tr>
                <?php endforeach; ?>
              </tbody>
            </table>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>

```

```

<!-- Earnings Chart -->
<div class="card mt-4">
    <div class="card-header">
        <h4>Earnings Overview</h4>
    </div>
    <div class="card-body">
        <canvas id="earningsChart"></canvas>
    </div>
</div>
<!-- Google Maps Live Tracking (Placeholder) -->
<div class="card mt-4">
    <div class="card-header">
        <h4>Live Driver Tracking</h4>
    </div>
    <div class="card-body">
        <iframe
src="https://maps.google.com/maps?q=Hinjilicut&t=&z=13&ie=UTF8&iwloc=&output=embed"
width="100%" height="300px" frameborder="0"></iframe>
    </div>
</div>
</body>
</html>

```

manage-blogs.php

```

<?php
require_once __DIR__ . '/../controllers/AdminController.php';
if (!isset($_SESSION['admin'])) {
    header('Location: ../../login.php');
    exit;
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Manage Blogs</title>
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
</head>
<body>
    <div class="container mt-5">
        <div class="d-flex justify-content-between align-items-center mb-4">
            <h1>Manage Blogs</h1>
            <button class="btn btn-primary" data-toggle="modal" data-target="#createBlogModal">
                Create Blog
            </button>
        </div>
</div>

```

```

        </div>
        <input type="text" id="search" class="form-control mb-3" placeholder="Search by title or
author" onkeyup="searchBlogs()">
        <div class="row" id="blogs-container">
            <?php foreach ($blogs as $blog): ?>
            <div class="col-md-4 mb-3 blog-card" data-title=<?= $blog['title'] ?>" data-author=<?=
$blog['author'] ?>">
                <div class="card">
                    <div class="card-body">
                        <h5 class="card-title"><?= $blog['title'] ?></h5>
                        <p class="card-text">by <?= $blog['author'] ?></p>
                        <button class="btn btn-primary" onclick="viewBlog(<?= $blog['id'] ?>)">View</button>
                        <button class="btn btn-danger" onclick="deleteBlog(<?= $blog['id'] ?>)">Delete</button>
                    <?php endforeach; ?>
                <!-- View Blog Modal -->
                <div class="modal fade" id="blogModal" tabindex="-1" aria-labelledby="blogModalLabel" aria-
hidden="true">
                    <div class="modal-dialog">
                        <div class="modal-content">
                            <div class="modal-header">
                                <h5 class="modal-title" id="blogModalLabel">Blog Details</h5>
                                <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                                    <span aria-hidden="true">&times;</span>
                                </button>
                            </div>
                            <div class="modal-body">
                                <h5 id="blogTitle"></h5>
                                <p id="blogAuthor"></p>
                                <p id="blogContent"></p>
                            </div>
                            <div class="modal-footer">
                                <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
                            </div>
                        </div>
                    </div>
                <script>
                    function searchBlogs() {
                        const searchInput = document.getElementById('search').value.toLowerCase();
                        const blogCards = document.querySelectorAll('.blog-card');
                        blogCards.forEach(card => {
                            const title = card.getAttribute('data-title').toLowerCase();
                            const author = card.getAttribute('data-author').toLowerCase();
                            if (title.includes(searchInput) || author.includes(searchInput)) {
                                card.style.display = 'block';
                            } else {
                                card.style.display = 'none';
                            }
                        });
                    }
                </script>
            
```

```

        }
    });
}

document.getElementById('createBlogForm').addEventListener('submit', function(e) {
    e.preventDefault();
    const formData = {
        action: 'createBlog',
        title: document.getElementById('title').value,
        content: document.getElementById('content').value,
        author: document.getElementById('author').value
    };
    fetch('../controllers/BlogController.php', {
        method: 'POST',
        headers: { 'Content-Type': 'application/json' },
        body: JSON.stringify(formData)
    })
    .then(response => response.json())
    .then(data => {
        if (data.success) {
            alert('Blog created successfully!');
            $('#createBlogModal').modal('hide');
            location.reload();
        } else {
            alert('Failed to create blog: ' + (data.error || 'Unknown error'));
        }
    })
    .catch(error => {
        console.error('Error:', error);
        alert('An error occurred while creating the blog');
    });
});

function viewBlog(id) {
    fetch('../controllers/AdminController.php', {
        method: 'POST',
        headers: { 'Content-Type': 'application/json' },
        body: JSON.stringify({ action: 'viewBlog', blogId: id })
    })
    .then(response => response.json())
    .then(data => {
        if (data) {
            document.getElementById('blogTitle').innerText = data.title;
            document.getElementById('blogAuthor').innerText = `by ${data.author}`;
            document.getElementById('blogContent').innerText = data.content;
            $('#blogModal').modal('show');
        } else {
    
```

```

        alert('Failed to fetch blog details');
    }
})
.catch(error => {
    console.error('Error:', error);
    alert('An error occurred while fetching blog details');
});
}

function deleteBlog(id) {
    if (confirm('Are you sure you want to delete this blog?')) {
        fetch('../controllers/AdminController.php', {
            method: 'POST',
            headers: { 'Content-Type': 'application/json' },
            body: JSON.stringify({ action: 'deleteBlog', blogId: id })
        })
        .then(response => response.json())
    }
}

</script>
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.4/dist/umd/popper.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</body>
</html>

edit-profile.php
<?php
require_once __DIR__ . '/../controllers/AdminController.php';
$adminId = $_SESSION['admin']['id'];
$adminDetails = $admin->getAdminById($adminId);
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Edit Profile</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body class="bg-light">
    <div class="container mt-5">
        <h1 class="text-center mb-4">Edit Profile</h1>
        <div class="card shadow p-4">
            <form id="profileForm" action="../controllers/AdminController.php" method="POST"
enctype="multipart/form-data">

```

```

<input type="hidden" name="action" value="updateProfile">
<div class="row">
    <!-- Details Section -->
    <div class="col-md-6">
        <div class="mb-3">
            <label for="name" class="form-label">Name:</label>
            <input type="text" id="name" name="name" class="form-control editable" value="<?php echo $adminDetails['name']; ?>" readonly required>
        </div>
        <div class="mb-3">
            <label for="email" class="form-label">Email:</label>
            <input type="email" id="email" name="email" class="form-control editable" value="<?php echo $adminDetails['email']; ?>" readonly required>
        </div>
        <div class="mb-3">
            <label for="password" class="form-label">Password:</label>
            <input type="password" id="password" name="password" class="form-control editable" placeholder="Enter new password" readonly>
        </div>
        <div class="mb-3">
            <label for="phone" class="form-label">Phone:</label>
            <input type="text" id="phone" name="phone" class="form-control editable" value="<?php echo $adminDetails['phone']; ?>" readonly required>
        </div>
        <div class="mb-3">
            <label for="created_at" class="form-label">Created At:</label>
            <input type="text" id="created_at" name="created_at" class="form-control" value="<?php echo $adminDetails['created_at']; ?>" readonly>
        <!-- Image Section -->
        <div class="col-md-6 text-center">
            <div class="mb-3">
                <label for="profile_image" class="form-label">Profile Image:</label>
                <input type="file" id="profile_image" name="profile_image" class="form-control editable" disabled>
            </div>
            <?php if (!empty($adminDetails['profile_image'])): ?>
                
            <?php endif; ?>
        <!-- Button Group -->
        <div id="buttonGroup">
            <button type="button" id="editBtn" class="btn btn-secondary btn-sm">Edit</button>
            <div id="editActions" style="display: none;">
                <button type="submit" class="btn btn-primary btn-sm mt-2">Update Profile</button>
            </div>
        </div>
    </div>
</div>

```

```

        <button type="button" id="cancelBtn" class="btn btn-danger btn-sm mt-2">Cancel</button>
    </form>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
<script>
    function enableEdit() {
        document.querySelectorAll('.editable').forEach(function(field) {
            if(field.type === 'file'){
                field.disabled = false;
            } else {
                field.removeAttribute('readonly');
            }
        });
        document.getElementById('editBtn').style.display = 'none';
        document.getElementById('editActions').style.display = 'block';
    }
    function disableEdit() {
        window.location.reload();
    }
    document.getElementById('editBtn').addEventListener('click', enableEdit);
    document.getElementById('cancelBtn').addEventListener('click', disableEdit);
</script>
</body>
</html>

```

manage-customers.php

```

<?php
require_once __DIR__ . '/../controllers/AdminController.php';
if (!isset($_SESSION['admin'])) {
    header('Location: ../../login.php');
    exit;
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Manage Customers</title>
    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body>
    <div class="container mt-5">
        <h1>Manage Customers</h1>

```

```

<div id="alertContainer"></div>
<input class="form-control mb-3" id="searchInput" type="text" placeholder="Search by
name...">
<div class="scrollable-table">
<table class="table table-striped">
<thead class="thead-dark">
<tr>
<th>Name</th>
<th>Profile Image</th>
<th>Actions</th>
</tr>
</thead>
<tbody id="customerTable">
<?php if (isset($customers) && !empty($customers)): ?>
<?php foreach ($customers as $customer): ?>
<tr>
<td><?php echo htmlspecialchars($customer['name']); ?></td>
<td></td>
<td>
<button class="btn btn-info btn-sm view-details" data-customer='<?php echo
json_encode($customer); ?>'>View Details</button>
<button class="btn btn-danger btn-sm delete-customer" data-customer-
id="<?php echo htmlspecialchars($customer['id']); ?>">Delete</button>
<!-- Bootstrap Modal -->
<div class="modal fade" id="customerModal" tabindex="-1" aria-
labelledby="customerModalLabel" aria-hidden="true">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title" id="customerModalLabel">Customer Details</h5>
<button type="button" class="close" data-dismiss="modal" aria-label="Close">
<span aria-hidden="true">&times;</span>
</button>
</div>
<div class="modal-body">
<p><strong>Name:</strong> <span id="modalCustomerName"></span></p>
<p><strong>Email:</strong> <span id="modalCustomerEmail"></span></p>
<p><strong>Phone:</strong> <span id="modalCustomerPhone"></span></p>
<p><strong>Profile Image:</strong> <img id="modalCustomerImage" src="" alt="Profile
Image" width="50"></p>
<p><strong>Created At:</strong> <span id="modalCustomerCreatedAt"></span></p>
</div>
<div class="modal-footer">
<button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>

```

```
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.4/dist/umd/popper.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
<script>
    document.getElementById('searchInput').addEventListener('keyup', function() {
        var input = this.value.toLowerCase();
        var rows = document.querySelectorAll('#customerTable tr');
        rows.forEach(function(row) {
            var name = row.querySelector('td').textContent.toLowerCase();
            if (name.includes(input)) {
                row.style.display = '';
            } else {
                row.style.display = 'none';
            }
        });
    });
    document.querySelectorAll('.view-details').forEach(function(button) {
        button.addEventListener('click', function() {
            var customer = JSON.parse(this.getAttribute('data-customer'));
            document.getElementById('modalCustomerName').textContent = customer.name;
            document.getElementById('modalCustomerEmail').textContent = customer.email;
            document.getElementById('modalCustomerPhone').textContent = customer.phone;
            document.getElementById('modalCustomerImage').src = '../../../../../uploads/users/profile/' +
customer.profile_image;
            document.getElementById('modalCustomerCreatedAt').textContent = customer.created_at;
            $('#customerModal').modal('show');
        });
    });
    document.querySelectorAll('.delete-customer').forEach(function(button) {
        button.addEventListener('click', function() {
            var customerId = this.getAttribute('data-customer-id');
            if (confirm("Are you sure you want to delete this customer?")) {
                fetch '../../../../../controllers/AdminController.php', {
                    method: 'POST',
                    headers: {
                        'Content-Type': 'application/json'
                    },
                    body: JSON.stringify({
                        action: 'deleteCustomer',
                        customerId: customerId
                    })
                })
            .then(response => response.json())
            .then(data => {
```

```

        var alertContainer = document.getElementById('alertContainer');
        if (data.success) {
            alertContainer.innerHTML = '<div class="alert alert-success" role="alert">Customer
deleted successfully.</div>';
            setTimeout(function() {
                location.reload();
            }, 2000);
        } else {
            alertContainer.innerHTML = '<div class="alert alert-danger" role="alert">Error: ' +
data.error + '</div>';
        }
    })
    .catch(error => {
        console.error('Error:', error);
        var alertContainer = document.getElementById('alertContainer');
        alertContainer.innerHTML = '<div class="alert alert-danger" role="alert">An error
occurred while deleting the customer.</div>';
    });
</script></body></html>

```

manage-drivers.php

```

<?php
require_once __DIR__ . '/../controllers/AdminController.php';
if (!isset($_SESSION['admin'])) {
    header('Location: ../../login.php');
    exit;
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Manage Drivers</title>
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
    <script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.4/dist/umd/popper.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/toastr@0.9.2/dist/toastr.min.js"></script>
</head>
<body>
    <div class="container">
        <h1 class="mt-5">Manage Drivers</h1>
        <?php if (isset($error)): ?>
            <div class="alert alert-danger"><?php echo $error; ?></div>

```

```

<?php else: ?>
    <div class="row mb-4">
        <div class="col-md-6">
<input type="text" id="searchApproved" class="form-control" placeholder="Search Approved Drivers by ID or Name">
        </div>
            <table class="table table-bordered" id="approvedDriversTable">
                <thead>
                    <tr>
                        <th>SI No</th>
                        <th>Profile</th>
                        <th>Name</th>
                        <th>Actions</th>
                    </tr>
                </thead>
                <tbody>
                    <?php $sl_no = 1; ?>
                    <?php foreach ($drivers as $driver): ?>
                        <?php if ($driver['status'] === 'approved'): ?>
                            <tr>
                                <td><?php echo $sl_no++; ?></td>
<td></td>
                                <td><?php echo $driver['name']; ?></td>
                                <td>
button class="btn btn-primary" onclick="viewDetails(<?php echo $driver['id']; ?>)">View Details</button>
                                <div class="col-md-6">
                                    <h2 class="mt-4">Pending Drivers</h2>
                                    <div class="scrollable-table">
                                        <table class="table table-bordered" id="pendingDriversTable">
                                            <thead>
                                                <tr>
                                                    <th>SI No</th>
                                                    <th>Profile</th>
                                                    <th>Name</th>
                                                    <th>Actions</th>
                                                </tr>
                                            </thead>
                                            <tbody>
                                                <?php $sl_no = 1; ?>
                                                <?php foreach ($drivers as $driver): ?>
                                                    <?php if ($driver['status'] === 'pending'): ?>
                                                        <tr>
                                                            <td><?php echo $sl_no++; ?></td>

```

```
<td></td>
<td><?php echo $driver['name']; ?></td>
<td>
<button class="btn btn-primary" onclick="viewDetails(<?php echo $driver['id']; ?>)">View Details</button>
<button class="btn btn-success" onclick="approveDriver(<?php echo $driver['id']; ?>)">Approve</button>
<button class="btn btn-danger" onclick="rejectDriver(<?php echo $driver['id']; ?>)">Reject</button>
<!-- Popup Modal for Viewing Details -->
<div class="modal fade" id="detailsModal" tabindex="-1" role="dialog" aria-labelledby="detailsModalLabel" aria-hidden="true">
<div class="modal-dialog modal-xl" role="document">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title" id="detailsModalLabel">Driver & Vehicle Details</h5>
<button type="button" class="close" data-dismiss="modal" aria-label="Close">
<span aria-hidden="true">&times;</span>
</button>
</div>
<div class="modal-body">
<div class="row" id="driverDetails">
<!-- Content will be loaded here via AJAX -->
</div>
</div>
<div class="modal-footer">
<button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
<button type="button" class="btn btn-primary" id="updateButton" onclick="enableUpdate()">Update</button>
<button type="button" class="btn btn-danger" id="deleteButton" onclick="deleteDriver()">Delete</button>
<!-- Add Confirmation Modal -->
<div class="modal fade" id="confirmationModal" tabindex="-1" role="dialog">
<div class="modal-dialog" role="document">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title">Confirm Action</h5>
<button type="button" class="close" data-dismiss="modal">&times;</button>
</div>
<div class="modal-body">
<p id="confirmationMessage"></p>
</div>
<div class="modal-footer">
```

```
        <button type="button" class="btn btn-secondary" data-
dismiss="modal">Cancel</button>
        <button type="button" class="btn btn-danger"
id="confirmActionBtn">Confirm</button>
<script>
let sortColumn = 'name';
let sortDirection = 'asc';
toastr.options = {
  closeButton: true,
  progressBar: true,
  positionClass: "toast-top-right",
  timeOut: 3000
};
const validationRules = {
  name: /^[a-zA-Z\s]{2,50}$/,
  email: /^[^@\s]+@[^\s@]+\.[^\s@]+\$/,
  phone: /^+[+]?[\d\s-]{10,15}$/,
  id_number: /^[w-]{5,20}$
};
function validateInput(field, value) {
  if (!validationRules[field].test(value)) {
    throw new Error(`Invalid ${field} format`);
  }
  return true;
}
function showConfirmation(message, callback) {
  $('#confirmationMessage').text(message);
  $('#confirmActionBtn').off('click').on('click', callback);
  $('#confirmationModal').modal('show');
}
function handleAjaxError(xhr, status, error) {
  toastr.error(xhr.responseText || 'An error occurred while processing your request');
}
function sortTable(table, column) {
  const rows = Array.from(table.find('tbody tr'));
  const index = table.find('th').index(column);
  const isAsc = column.hasClass('asc');
  rows.sort((a, b) => {
    const aValue = $(a).find('td').eq(index).text();
    const bValue = $(b).find('td').eq(index).text();
    return isAsc ? bValue.localeCompare(aValue) : aValue.localeCompare(bValue);
  });
  table.find('tbody').empty().append(rows);
  table.find('th').removeClass('asc desc');
  column.addClass(isAsc ? 'desc' : 'asc');
```

```
}

function viewDetails(driverId) {
    $.ajax({
        url: '../../../../../controllers/AdminController.php',
        type: 'POST',
        contentType: 'application/json',
        data: JSON.stringify({ action: 'viewDetails', driverId: driverId }),
        success: function(response) {
            try {
                const data = JSON.parse(response);
                const driver = data.driver;
                const vehicle = data.vehicle;

                let driverHtml = `
                    <div class="col-md-6">
                        <div class="card">
                            <div class="card-header bg-primary text-white">
                                <h5 class="mb-0">Driver Information</h5>
                            </div>
                            <div class="card-body">
                                <form>
                                    <input type="hidden" id="driverId" value="${driver.id}">
                                    <div class="form-group">
                                        <label>Name</label>
                                        <input type="text" id="driverName" value="${driver.name}" class="form-control" disabled>
                                    </div>
                                    <div class="form-group">
                                        <label>Email</label>
                                        <input type="email" id="driverEmail" value="${driver.email}" class="form-control" disabled>
                                    </div>
                                    <div class="form-group">
                                        <label>Phone</label>
                                        <input type="tel" id="driverPhone" value="${driver.phone}" class="form-control" disabled>
                                    </div>
                                    <div class="form-group">
                                        <label>ID Number</label>
                                        <input type="text" id="driverIdNumber" value="${driver.id_number}" class="form-control" disabled>
                                    </div>
                                    <div class="form-group">
                                        <label>Status</label>
                                    </div>
                                </form>
                            </div>
                        </div>
                    </div>
                `;
            }
        }
    });
}
```

```

        <input type="text" id="driverStatus" value="${driver.status}" class="form-control" disabled>


        </form>
    </div>';
let vehicleHtml = '';
if (vehicle) {
    vehicleHtml = `
        <div class="col-md-6">
            <div class="card">
                <div class="card-header bg-success text-white">
                    <h5 class="mb-0">Vehicle Information</h5>
                    <label>License Number</label>
<input type="text" id="licenseNumber" value="${vehicle.license_number}" class="form-control" disabled>
                    </div>
                    <div class="form-group">
                        <label>Vehicle Type</label>
<input type="text" id="vehicleType" value="${vehicle.vehicle_type}" class="form-control" disabled>
                    </div>
                    <div class="form-group">
                        <label>Vehicle Model</label>
<input type="text" id="vehicleModel" value="${vehicle.vehicle_model}" class="form-control" disabled>
                    </div>
                    <div class="form-group">
                        <label>Vehicle Number</label>
<input type="text" id="vehicleNumber" value="${vehicle.vehicle_number}" class="form-control" disabled>
                    </div>

<div class="col-md-6">
                    <div class="form-group">
                        <label>Vehicle Image</label>
                        <div class="mt-2">
                            
                        </div>
                    </div>
                </div>
            </div>
        </div>
    `;
}
else {
    vehicleHtml = `
        <div class="col-md-6">
            <div class="card">
                <div class="card-header bg-success text-white">
                    <h5 class="mb-0">Vehicle Information</h5>
                    <label>License Number</label>
<input type="text" id="licenseNumber" value="${vehicle.license_number}" class="form-control" disabled>
                    </div>
                    <div class="form-group">
                        <label>Vehicle Type</label>
<input type="text" id="vehicleType" value="${vehicle.vehicle_type}" class="form-control" disabled>
                    </div>
                    <div class="form-group">
                        <label>Vehicle Model</label>
<input type="text" id="vehicleModel" value="${vehicle.vehicle_model}" class="form-control" disabled>
                    </div>
                    <div class="form-group">
                        <label>Vehicle Number</label>
<input type="text" id="vehicleNumber" value="${vehicle.vehicle_number}" class="form-control" disabled>
                    </div>

<div class="col-md-6">
                    <div class="form-group">
                        <label>Vehicle Image</label>
                        <div class="mt-2">
                            
                        </div>
                    </div>
                </div>
            </div>
        </div>
    `;
}

```

```

        <div class="card">
            <div class="card-header bg-success text-white">
                <h5 class="mb-0">Vehicle Information</h5>
            </div>
            <div class="card-body">
                <div class="alert alert-info">
                    No vehicle information available for this driver.
                </div>
                <form>
                    <div class="form-group mt-3">
                        <label>License Number</label>
                        <input type="text" id="licenseNumber" class="form-control" disabled>
                    </div>
                    <div class="form-group">
                        <label>Vehicle Type</label>
                        <input type="text" id="vehicleType" class="form-control" disabled>
                    </div>
                    <div class="form-group">
                        <label>Vehicle Model</label>
                        <input type="text" id="vehicleModel" class="form-control" disabled>
                    </div>
                    <div class="form-group">
                        <label>Vehicle Number</label>
                        <input type="text" id="vehicleNumber" class="form-control" disabled>
                    </div>
                </form>
            </div>';
        }
        $('#driverDetails').html(driverHtml + vehicleHtml);
        $('#detailsModal').modal('show');
    } catch (error) {
        toastr.error('Error parsing driver details');
        console.error(error);
    }
},
error: handleAjaxError
});
}
function approveDriver(driverId) {
    showConfirmation('Are you sure you want to approve this driver?', function() {
        $('#confirmationModal').modal('hide');
        $.ajax({
            url: '../../../../../controllers/AdminController.php',
            type: 'POST',
            contentType: 'application/json',
            data: JSON.stringify({ action: 'approve', driverId: driverId }),

```

```

        success: function(response) {
            toastr.success('Driver approved successfully');
            setTimeout(() => location.reload(), 1000);
        },
        error: handleAjaxError
    });
});
}

function rejectDriver(driverId) {
    showConfirmation('Are you sure you want to reject this driver?', function() {
        $('#confirmationModal').modal('hide');
        $.ajax({
            url: '../../../../../controllers/AdminController.php',
            type: 'POST',
            contentType: 'application/json',
            data: JSON.stringify({ action: 'reject', driverId: driverId }),
            success: function(response) {
                toastr.success('Driver rejected successfully');
                setTimeout(() => location.reload(), 1000);
            },
            error: handleAjaxError
        });
    });
}

function deleteDriver() {
    const driverId = $('#driverId').val();
    showConfirmation('Are you sure you want to delete this driver? This action cannot be undone.', function() {
        $('#confirmationModal').modal('hide');
        $.ajax({
            url: '../../../../../controllers/AdminController.php',
            type: 'POST',
            contentType: 'application/json',
            data: JSON.stringify({ action: 'delete', driverId: driverId }),
            success: function(response) {
                toastr.success('Driver deleted successfully');
                setTimeout(() => location.reload(), 1000);
            },
            error: handleAjaxError
        });
    });
}

function enableUpdate() {
    $('#driverName, #driverEmail, #driverPhone, #driverIdNumber, #driverStatus, #licenseNumber,
#vehicleType, #vehicleModel, #vehicleNumber').prop('disabled', false);
}

```

```

        $('#updateButton').text('Save').attr('onclick', 'saveDriver()');
    }
}

function saveDriver() {
    const driverId = $('#driverId').val();
    try {
        const driverData = {
            name: $('#driverName').val(),
            email: $('#driverEmail').val(),
            phone: $('#driverPhone').val(),
            id_number: $('#driverIdNumber').val(),
            status: $('#driverStatus').val()
        };
        const vehicleData = {
            license_number: $('#licenseNumber').val(),
            vehicle_type: $('#vehicleType').val(),
            vehicle_model: $('#vehicleModel').val(),
            vehicle_number: $('#vehicleNumber').val()
        };
        Object.keys(driverData).forEach(field => {
            if (validationRules[field]) {
                validateInput(field, driverData[field]);
            }
        });
        const vehicleValidationRules = {
            license_number: /^[A-Z0-9-]{5,20}$/,
            vehicle_number: /^[A-Z0-9-]{5,15}$
        };
        Object.keys(vehicleData).forEach(field => {
            if (vehicleValidationRules[field] && vehicleData[field]) {
                if (!vehicleValidationRules[field].test(vehicleData[field])) {
                    throw new Error(`Invalid ${field.replace('_', ' ')} format`);
                }
            }
        });
        $.ajax({
            url: '../../../../../controllers/AdminController.php',
            type: 'POST',
            contentType: 'application/json',
            data: JSON.stringify({
                action: 'update',
                driverId: driverId,
                driverData: driverData,
                vehicleData: vehicleData
            }),
            success: function(response) {

```

```

        toastr.success('Information updated successfully');
        setTimeout(() => location.reload(), 1000);
    },
    error: handleAjaxError
});
} catch (error) {
    toastr.error(error.message);
}
}

</body>
</html>

```

report.php

```

<?php
require_once '../controllers/StatisticsController.php';
$controller = new StatisticsController();
$data = $controller->getReportData();
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Monthly Report - ETaxi</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.7.2/font/bootstrap-icons.css"
rel="stylesheet">
    transform: translateY(-5px);
</head>
<body>
    <div class="container-fluid py-4">
        <div class="row mb-4">
            <div class="col">
                <h2>Monthly Report</h2>
                <p class="text-muted">Generated on <?php echo $generatedDate; ?></p>
            </div>
            <div class="col text-end no-print">
                <button class="btn btn-primary" onclick="window.print()">
                    <i class="bi bi-printer"></i> Print Report
                </button>
            <!-- Filter Section -->
            <div class="card mb-4 no-print">
                <div class="card-body">
                    <form method="GET" class="row g-3">
                        <div class="col-md-4">
                            <label class="form-label">Year</label>
                            <select name="year" class="form-select">

```



```

        <h5 class="card-title">Monthly Rides Trend</h5>
        <div class="chart-container">
            <div class="col-md-6">
                <div class="card">
                    <div class="card-body">
                        <h5 class="card-title">Monthly Earnings Trend</h5>
                        <div class="chart-container">
                            <canvas id="earningsChart"></canvas>
                        <div class="card-body">
                            <h5 class="card-title">Monthly Breakdown</h5>
                            <div class="table-responsive">
                                <table class="table table-striped">
                                    <thead>
                                        <tr>
                                            <th>Month</th>
                                            <th>Total Rides</th>
                                            <th>Total Distance (km)</th>
                                            <th>Total Earnings</th>
                                            <th>Total Customers</th>
                                        </tr>
                                    </thead>
                                    <tbody>
                                        <?php foreach($yearlyStats as $monthNum => $monthData): ?>
                                        <tr>
                                            <td><?php echo $months[$monthNum]; ?></td>
                                            <td><?php echo number_format($monthData['rides']); ?></td>
                                            <td><?php echo number_format($monthData['distance'], 2); ?></td>
                                            <td>₹<?php echo number_format($monthData['amount'], 2); ?></td>
                                            <td><?php echo number_format($monthData['customers']); ?></td>
                                        </tr>
                                        <?php endforeach; ?>
                                    </tbody>
                                </table>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </body>
</html>

```

Views/customer:

dashboard.php

```

<?php
session_start();
if (!isset($_SESSION['customer'])) {
    header('Location: ../../login.php');
}

```

```

        exit;
    }
$adminData = $_SESSION['customer'];
$adminImage = '../..uploads/users/profile/' . (!empty($adminData['profile_image'])) ? $adminData['profile_image'] : 'default.jpg'; // Default image if not set
$options = [
    ['name' => 'Home', 'link' => 'home.php', 'subItems' => []],
    ['name' => 'Book a Ride', 'link' => 'book-ride.php', 'subItems' => []],
    ['name' => 'My Rides', 'link' => 'my-rides.php', 'subItems' => []],
    ['name' => 'Payments', 'link' => 'payments.php', 'subItems' => []],
    ['name' => 'Blogs', 'link' => 'blogs.php', 'subItems' => []],
];
include '../dashboard_layout.php';
?>

home.php
<?php
require_once '../controllers/CustomerController.php';
if (!isset($_SESSION['customer'])) {
    header('Location: ../login.php');
    exit;
}
$customerData = $_SESSION['customer'];
$customerRides = $ride->getRidesByCustomer($customerData['id']);
$totalRides = count($customerRides);
$totalPayments = $payment->getTotalPaymentsByCustomer($customerData['id']);
$latestRides = $ride->getLatestRidesByCustomer($customerData['id'], 5);
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>User Dashboard</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
<div class="container mt-4">
    <!-- Dashboard Summary -->
    <div class="row">
        <div class="col-md-6">
            <div class="card text-white bg-primary mb-3">
                <div class="card-header">Total Rides</div>
                <div class="card-body">
                    <h4 class="card-title"><?= $totalRides ?></h4>
                </div>
            </div>
        </div>
    </div>
</div>

```

```

        </div>
    <div class="col-md-6">
        <div class="card text-white bg-success mb-3">
            <div class="card-header">Total Payments</div>
            <div class="card-body">
                <h4 class="card-title">$<?= number_format($totalPayments, 2) ?></h4>
            <!-- Recent Rides -->
            <div class="card mt-4">
                <div class="card-header">
                    <h4>Recent Rides</h4>
                </div>
                <div class="card-body">
                    <table class="table table-striped">
                        <thead>
<?php
function getRideStatusColor($status) {
    switch ($status) {
        case 'completed':
            return 'success';
        case 'ongoing':
            return 'primary';
        case 'cancelled':
            return 'danger';
        case 'pending':
            return 'warning';
        default:
            return 'secondary';
    }
?>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
<script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyAhC0pX3qvfsOUFqvh0iztqgdA52kXZW4"></script>
<script>
function initMap() {
    const map = new google.maps.Map(document.getElementById('map'), {
        zoom: 15,
        center: { lat: 0, lng: 0 }
    });
    if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(
            (position) => {
                const pos = {
                    lat: position.coords.latitude,
                    lng: position.coords.longitude
                };

```

```

        map.setCenter(pos);
        new google.maps.Marker({
            position: pos,
            map: map,
            title: 'Your Location'
        });
    },
}
</script>
</body>
</html>
edit-profile.php
<?php
require_once '../controllers/CustomerController.php';
if (!isset($_SESSION['customer'])) {
    header('Location: ../../login.php');
    exit;
}
$customerData = $_SESSION['customer'];
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Edit Profile</title>
    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">
<script>
    function previewImage(input) {
        if (input.files && input.files[0]) {
            const reader = new FileReader();
            reader.onload = function(e) {
                document.getElementById('preview-image').src = e.target.result;
            }
            reader.readAsDataURL(input.files[0]);
        }
    }
</script>
</head>
<body>
    <div class="container mt-5">
        <h2>Edit Profile</h2>
        <?php if (isset($_SESSION['success_message'])): ?>
            <div class="alert alert-success">

```

```

        <?php echo $_SESSION['success_message']; unset($_SESSION['success_message']); ?>
    </div>
<?php endif; ?>
<?php if (isset($_SESSION['error_message'])): ?>
    <div class="alert alert-danger">
        <?php echo $_SESSION['error_message']; unset($_SESSION['error_message']); ?>
    </div>
<?php endif; ?>
        <form      action="../../controllers/CustomerController.php"      method="POST"
enctype="multipart/form-data">
    <input type="hidden" name="action" value="updateProfile">
    <div class="row">
        <div class="col-md-4 text-center">
            <div class="profile-image-container">
                <?php
                    $profile_image = !empty($customerData['profile_image']) ?
                        htmlspecialchars($customerData['profile_image']) :
                        'default-profile.jpg';
                ?>
                
            </div>
            <div class="form-group">
                <label for="profile_image" class="profile-upload-label btn btn-outline-primary">
                    <i class="fas fa-camera"></i> Change Profile Picture
                </label>
                <input type="file" id="profile_image" name="profile_image" class="form-control d-none" accept="image/*"
                    onchange="previewImage(this);">
            <div class="col-md-8">
                <div class="form-group">
                    <label for="name">Name</label>
                <input type="text" name="name" class="form-control" value="<?php echo
                    htmlspecialchars($customerData['name']); ?>" required>
                </div>
                <div class="form-group">
                    <label for="email">Email</label>
                <input type="email" name="email" class="form-control" value="<?php echo
                    htmlspecialchars($customerData['email']); ?>" required>
                </div>
                <div class="form-group">
                    <label for="phone">Phone</label>
                <input type="text" name="phone" class="form-control" value="<?php echo
                    htmlspecialchars($customerData['phone']); ?>" required>
                </div>

```

```

        <button type="submit" class="btn btn-primary">Update Profile</button>
    </form>
</div>
<link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css" rel="stylesheet">
</body>
</html>

```

book-ride.php

```

<?php
session_start();
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Book a Ride</title>
    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">
</head>
<body class="bg-light">
    <div class="container-fluid py-4">
        <?php if (isset($_GET['success'])): ?>
            <div class="alert alert-success" role="alert">
                Ride booked successfully! Ride ID: <?php echo htmlspecialchars($_GET['ride_id']); ?>, Fare: ₹ <?php echo htmlspecialchars($_GET['fare']); ?>
            </div>
        <?php elseif (isset($_GET['error'])): ?>
            <div class="alert alert-danger" role="alert">
                Error: <?php echo htmlspecialchars($_GET['error']); ?>
            </div>
        <?php endif; ?>
        <form action="../../controllers/RideController.php" method="POST">
            <div class="form-group">
                <label for="customer_id">Customer ID:</label>
                <input type="text" class="form-control bg-light" id="customer_id" name="customer_id" value="<?php echo htmlspecialchars($_SESSION['customer_id']); ?>" readonly>
            </div>
            <div class="form-group">
                <label for="pickup_location">Pickup Location:</label>
                <input type="text" class="form-control" id="pickup_location" name="pickup_location" required>
            </div>
            <div class="form-group">

```

```

        <label for="dropoff_location">Dropoff Location:</label>
        <input type="text" class="form-control" id="dropoff_location"
name="dropoff_location" required>
        </div>
        <div id="route-info" class="alert alert-info d-none mb-4">
            <div id="distance-time"></div>
        </div>
        <div class="row">
            <div class="col-md-6">
                <div class="form-group">
                    <label for="distance">Distance (km):</label>
<input type="text" class="form-control bg-light" id="distance" name="distance" readonly required>
                    </div>
                </div>
                <div class="col-md-6">
                    <div class="form-group">
                        <label for="fare">Fare (₹):</label>
                        <input type="text" class="form-control bg-light" id="fare" name="fare" readonly
required>
                    <button type="submit" class="btn btn-success btn-lg btn-block mt-4" id="bookRideBtn"
disabled>
                        Book Ride
                    </button>
                </div>
            </div>
        <!-- Map Column -->
        <div class="col-md-6 pl-md-2">
            <div class="card shadow-sm h-100">
                <div class="card-body p-0">
                    <div id="map" class="rounded"></div>
                <script async defer
src="https://maps.gomaps.pro/maps/api/js?key=AlzaSyNtmxXnhshZB8JTxGTK4tluneh1xHF3zuq
&libraries=geometry,places&callback=initMap">
                </script>
                <script>
                    let map, directionsService, directionsRenderer;
                    let autocomplete1, autocomplete2;
                    let marker1, marker2;
                    function initMap() {
                        const ganjamBounds = {
                            north: 20.017,
                            south: 19.059,
                            east: 85.090,
                            west: 84.108
                        };
                        const ganjamCenter = {

```

```
        lat: 19.588,
        lng: 84.578
    );
    map = new google.maps.Map(document.getElementById('map'), {
        center: ganjamCenter,
        zoom: 10,
        restriction: {
            latLngBounds: ganjamBounds,
            strictBounds: false
        },
        mapTypeControl: true,
        mapTypeControlOptions: {
            style: google.maps.MapTypeControlStyle.DROPDOWN_MENU
        }
    });
    const ganjamRectangle = new google.maps.Rectangle({
        bounds: ganjamBounds,
        strokeColor: '#FF0000',
        strokeOpacity: 0.8,
        strokeWeight: 2,
        fillColor: '#FF0000',
        fillOpacity: 0.1
    });
    ganjamRectangle.setMap(map);
    directionsService = new google.maps.DirectionsService();
    directionsRenderer = new google.maps.DirectionsRenderer({
        map: map,
        suppressMarkers: true
    });
    const input1 = document.getElementById('pickup_location');
    const input2 = document.getElementById('dropoff_location');
    const options = {
        bounds: new google.maps.LatLngBounds(
            new google.maps.LatLng(ganjamBounds.south, ganjamBounds.west),
            new google.maps.LatLng(ganjamBounds.north, ganjamBounds.east)
        ),
        strictBounds: true,
        componentRestrictions: { country: 'IN' }
    };
    autocomplete1 = new google.maps.places.Autocomplete(input1, options);
    autocomplete2 = new google.maps.places.Autocomplete(input2, options);
    autocomplete1.addListener('place_changed', () => {
        const place = autocomplete1.getPlace();
        setMarker(place, 'marker1');
        calculateRoute();
    });
}
```

```

    });
    autocomplete2.addListener('place_changed', () => {
      const place = autocomplete2.getPlace();
      setMarker(place, 'marker2');
      calculateRoute();
    });
  }
  function setMarker(place, markerType) {
    if (!place.geometry) {
      console.log("No details available for: " + place.name);
      return;
    }
    const position = place.geometry.location;
    const ganjamBounds = {
      north: 20.017,
      south: 19.059,
      east: 85.090,
      west: 84.108
    };
    if (!isLocationInGanjam(position, ganjamBounds)) {
      const locationType = markerType === 'marker1' ? 'pickup' : 'dropoff';
      document.getElementById(locationType === 'pickup' ? 'pickup_location' : 'dropoff_location').value = "";

      const alertDiv = document.createElement('div');
      alertDiv.className = 'alert alert-danger alert-dismissible fade show mt-2';
      alertDiv.innerHTML = `
        <strong>Invalid Location!</strong> Selected ${locationType} location is outside Ganjam
district.
        <button type="button" class="close" data-dismiss="alert" aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      `;
      const existingAlert = document.querySelector('.alert-danger:not([role="alert"])');
      if (existingAlert) {
        existingAlert.remove();
      }
      const inputField = document.getElementById(locationType === 'pickup' ? 'pickup_location' : 'dropoff_location');
      inputField.parentNode.insertBefore(alertDiv, inputField.nextSibling);
      resetFormState();
      return;
    }
    const icon = {

```

```
url: markerType === 'marker1' ? 'https://maps.google.com/mapfiles/ms/icons/green-dot.png'
: 'https://maps.google.com/mapfiles/ms/icons/red-dot.png',
scaledSize: new google.maps.Size(40, 40)
};

if (markerType === 'marker1') {
  if (marker1) marker1.setMap(null);
  marker1 = new google.maps.Marker({
    position,
    map,
    icon: icon,
    title: 'Pickup Location'
  });
} else {
  if (marker2) marker2.setMap(null);
  marker2 = new google.maps.Marker({
    position,
    map,
    icon: icon,
    title: 'Dropoff Location'
  });
}

function isLocationInGanjam(position, bounds) {
  return position.lat() >= bounds.south &&
    position.lat() <= bounds.north &&
    position.lng() >= bounds.west &&
    position.lng() <= bounds.east;
}

function resetFormState() {
  document.getElementById('bookRideBtn').disabled = true;
  document.getElementById('bookRideBtn').innerHTML = 'Book Ride';
  document.getElementById('route-info').classList.add('d-none');
  document.getElementById('distance').value = '';
  document.getElementById('fare').value = '';
  if (directionsRenderer) {
    directionsRenderer.setDirections({routes: []});
  }
  if (marker1) marker1.setMap(null);
  if (marker2) marker2.setMap(null);
}

function removeLocationAlerts() {
  const alerts = document.querySelectorAll('.alert-danger:not([role="alert"])');
  alerts.forEach(alert => alert.remove());
}

document.getElementById('pickup_location').addEventListener('input', () => {
```

```

        resetFormState();
        removeLocationAlerts();
    });
document.getElementById('dropoff_location').addEventListener('input', () => {
    resetFormState();
    removeLocationAlerts();
});
</script>
</body>
</html>
my-rides.php
<?php
require_once '../controllers/CustomerController.php';
if (!isset($_SESSION['customer'])) {
    header('Location: ../login.php');
    exit;
}
$customerData = $_SESSION['customer'];
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</head>
<body>
    <div class="container mt-5">
        <h1 class="mb-4">My Rides</h1>
        <div class="table-responsive">
            <table class="table table-striped table-bordered">
                <thead class="thead-dark">
                    <tr>
                        <th>ID</th>
                        <th>Driver</th>
                        <th>Pickup Location</th>
                        <th>Dropoff Location</th>
                        <th>Status</th>
                        <th>Action</th>
                    </tr>
                </thead>
                <tbody>
                    <?php foreach ($rides as $ride): ?>
                    <tr>
                        <td><?= $ride['id'] ?></td>

```

```

<td><?= $ride['driver_name'] ?></td>
<td><?= $ride['pickup_location'] ?></td>
<td><?= $ride['dropoff_location'] ?></td>
<td><?= $ride['ride_status'] ?></td>
<td>
    <?php if ($ride['ride_status'] === 'pending'): ?>
        <button class="btn btn-danger btn-sm" onclick="cancelRide(<?= $ride['id'] ?>)">Cancel</button>
    <?php elseif ($ride['ride_status'] === 'accepted'): ?>
        <button class="btn btn-primary btn-sm" onclick="openReviewModal(<?= $ride['id'] ?>, <?= $ride['driver_id'] ?>)">Review</button>
        <button class="btn btn-success btn-sm" onclick="makePayment(<?= $ride['id'] ?>, <?= $ride['fare'] ?>)">Pay</button>
    <?php endif; ?>
<div id="reviewModal" class="modal" tabindex="-1" role="dialog">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title">Review Ride</h5>
                <button type="button" class="close" onclick="closeReviewModal()" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <div class="modal-body">
                <form id="reviewForm" onsubmit="submitReview(event)">
                    <input type="hidden" id="ridelId" name="ridelId">
                    <input type="hidden" id="driverId" name="driverId">
                    <div class="form-group">
                        <label for="rating">Rating:</label>
                        <input type="number" id="rating" name="rating" class="form-control" min="1" max="5" required>
                    </div>
                    <div class="form-group">
                        <label for="review">Review:</label>
                        <textarea id="review" name="review" class="form-control" required></textarea>
                    </div>
                    <button type="submit" class="btn btn-primary">Submit</button>
                    <button type="button" class="btn btn-secondary" onclick="closeReviewModal()">Cancel</button>
                </form>
            </div>
        </div>
    </div>
<script>
    function cancelRide(ridelId) {
        if (!confirm('Are you sure you want to cancel this ride?')) {
            return;
        }
    }

```

```
const formData = new FormData();
formData.append('action', 'cancelRide');
formData.append('ride_id', ridId);
fetch('../controllers/RideController.php', {
    method: 'POST',
    body: formData
})
.then(response => response.json())
.then(data => {
    if (data.success) {
        alert('Ride cancelled successfully.');
        location.reload();
    } else {
        alert('Failed to cancel ride: ' + (data.error || 'Unknown error'));
    }
})
.catch(error => {
    console.error('Error:', error);
    alert('An error occurred while cancelling the ride. Please try again.');
});
}

function openReviewModal(ridId, driverId) {
    document.getElementById('ridId').value = ridId;
    document.getElementById('driverId').value = driverId;
    $('#reviewModal').modal('show');
}

function closeReviewModal() {
    $('#reviewModal').modal('hide');
}

function submitReview(event) {
    event.preventDefault();
    const formData = new FormData(document.getElementById('reviewForm'));
    formData.append('action', 'submitReview');
    fetch('../controllers/CustomerController.php', {
        method: 'POST',
        body: formData
    })
    .then(response => response.json())
    .then(data => {
        if (data.success) {
            alert('Review submitted successfully.');
            closeReviewModal();
            location.reload();
        } else {
            console.error('Failed to submit review:', data.error);
        }
    })
}
```

```

        alert('Failed to submit review: ' + data.error);
    }
})
.catch(error => {
    console.error('An error occurred:', error);
    alert('An error occurred. Please try again.');
});
}

function makePayment(rideId, rideFare) {
    window.location.href = '/ETaxi/payment/pay.php?ride_id=' + rideId + '&ride_fare=' + rideFare;
}
</script>
</body>
</html>
messages.php
notifications.php
blogs.php
<?php
session_start();
if (!isset($_SESSION['customer'])) {
    header('Location: ../../login.php');
    exit;
}
require_once '../../controllers/BlogController.php';
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Blogs</title>
    <link href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</head>
<body>
    <div class="container">
        <h1>Blogs</h1>
        <button type="button" class="btn btn-primary" data-toggle="modal" data-target="#createBlogModal">
            Create Blog
        </button>
        <div class="modal fade" id="createBlogModal" tabindex="-1" role="dialog" aria-labelledby="createBlogModalLabel" aria-hidden="true">

```

```

<div class="modal-dialog" role="document">
  <div class="modal-content">
    <div class="modal-header">
      <h5 class="modal-title" id="createBlogModalLabel">Create Blog</h5>
      <button type="button" class="close" data-dismiss="modal" aria-label="Close">
        <span aria-hidden="true">&times;</span>
      </button>
    </div>
    <div class="modal-body">
      <form id="createBlogForm">
        <div class="form-group">
          <label for="title">Title</label>
          <input type="text" class="form-control" id="title" placeholder="Title" required>
        </div>
        <div class="form-group">
          <label for="content">Content</label>
          <textarea class="form-control" id="content" placeholder="Content" required></textarea>
        </div>
        <div class="form-group">
          <label for="author">Author</label>
          <input type="text" class="form-control" id="author" placeholder="Author" required>
        </div>
        <button type="submit" class="btn btn-primary">Create Blog</button>
      </form>
    <div class="table-responsive" style="max-height: 400px; overflow-y: auto;">
      <table class="table mt-3">
        <thead>
          <tr>
            <th>ID</th>
            <th>Title</th>
            <th>Content</th>
            <th>Author</th>
            <th>Created At</th>
          </tr>
        </thead>
        <tbody id="blogsTable">
          <?php foreach ($blogs['blogs'] as $blog): ?>
          <tr id="blog-<?php echo htmlspecialchars($blog['id']); ?>">
            <td><?php echo htmlspecialchars($blog['id']); ?></td>
            <td><?php echo htmlspecialchars($blog['title']); ?></td>
            <td><?php echo htmlspecialchars($blog['content']); ?></td>
            <td><?php echo htmlspecialchars($blog['author']); ?></td>
            <td><?php echo htmlspecialchars($blog['created_at']); ?></td>
          </tr>
        </tbody>
      </table>
    </div>
  </div>
</div>

```

```

        <?php endforeach; ?>
    </tbody>
</table>
</div>
<script>
    document.addEventListener('DOMContentLoaded', function() {
        document.getElementById('createBlogForm').addEventListener('submit', createBlog);
        function createBlog(event) {
            event.preventDefault();
            const title = document.getElementById('title').value.trim();
            const content = document.getElementById('content').value.trim();
            const author = document.getElementById('author').value.trim();
            if (!title || !content || !author) {
                alert('Please fill in all fields');
                return;
            }
            fetch('../controllers/BlogController.php', {
                method: 'POST',
                headers: {
                    'Content-Type': 'application/json'
                },
                body: JSON.stringify({
                    action: 'createBlog',
                    title: title,
                    content: content,
                    author: author
                })
            })
            .then(response => response.json())
            .then(data => {
                if (data.success) {
                    alert('Blog created successfully!');
                    $('#createBlogModal').modal('hide');
                    location.reload();
                } else {
                    alert('Failed to create blog: ' + (data.error || 'Unknown error'));
                }
            })
            .catch(error => {
                console.error('Error:', error);
                alert('An error occurred while creating the blog');
            });
        }
        function loadAllBlogs() {
            fetch('../controllers/BlogController.php?action=all') // Fetch all blogs

```

```

        .then(response => response.json())
        .then(data => {
            if (data.success) {
                renderBlogs(data.blogs);
            } else {
                alert('Failed to load blogs');
            }
        })
        .catch(error => {
            console.error('Error:', error);
            alert('An error occurred while loading blogs');
        });
    }

    function renderBlogs(blogs) {
        const blogsTable = document.getElementById('blogsTable');
        blogsTable.innerHTML = '';
        blogs.forEach(blog => {
            const row = document.createElement('tr');
            row.innerHTML = `
                <td>${blog.id}</td>
                <td>${blog.title}</td>
                <td>${blog.content}</td>
                <td>${blog.author}</td>
                <td>${blog.created_at}</td>
            `;
            blogsTable.appendChild(row);
        });
    }

    loadAllBlogs();
});

</script>
</body>
</html>
Views/driver:
dashboard.php
<?php
session_start();
if (!isset($_SESSION['driver'])) {
    header('Location: ../../login.php');
    exit;
}
$adminData = $_SESSION['driver'];
$adminImage = '../../uploads/drivers/profile/' . (!empty($adminData['profile_image'])) ? $adminData['profile_image'] : 'default.jpg'; // Default image if not set
$options = [
    'name' => 'Home', 'link' => 'home.php', 'subItems' => []
],

```



```

        <input type="email" class="form-control" name="email" value="<?=
htmlspecialchars($profile['email']) ?>" required>
    </div>
    <div class="form-group">
        <label>Phone</label>
        <input type="text" class="form-control" name="phone" value="<?=
htmlspecialchars($profile['phone']) ?>" required>
    </div>
    <div class="form-group">
        <label>ID Number</label>
        <input type="text" class="form-control" name="id_number" value="<?=
htmlspecialchars($profile['id_number']) ?>" required>
    </div>
    <div class="form-group">
        <label>Profile Image</label>
        
        <input type="file" class="form-control" name="profile_image" accept="image/*">
    </div>
    <button type="submit" class="btn btn-success btn-block">Save Driver Info</button>
</form>
<!-- Vehicle Information Form --&gt;
&lt;form id="vehicleForm" enctype="multipart/form-data"&gt;
    &lt;input type="hidden" name="action" value="updateVehicleProfile"&gt;
    &lt;div class="card mb-4"&gt;
        &lt;div class="card-header bg-primary text-white"&gt;Vehicle Information&lt;/div&gt;
        &lt;div class="card-body"&gt;
            &lt;div class="form-group"&gt;
                &lt;label&gt;License Number&lt;/label&gt;
                &lt;input type="text" class="form-control" name="license_number" value="&lt;?=
htmlspecialchars($profile['license_number']) ?&gt;" required&gt;
            &lt;/div&gt;
            &lt;div class="form-group"&gt;
                &lt;label&gt;Vehicle Type&lt;/label&gt;
                &lt;input type="text" class="form-control" name="vehicle_type" value="&lt;?=
htmlspecialchars($profile['vehicle_type']) ?&gt;" required&gt;
            &lt;/div&gt;
            &lt;div class="form-group"&gt;
                &lt;label&gt;Vehicle Model&lt;/label&gt;
                &lt;input type="text" class="form-control" name="vehicle_model" value="&lt;?=
htmlspecialchars($profile['vehicle_model']) ?&gt;" required&gt;
            &lt;/div&gt;
            &lt;div class="form-group"&gt;
                &lt;label&gt;Vehicle Number&lt;/label&gt;
</pre>

```

```
        <input type="text" class="form-control" name="vehicle_number" value="<?=
htmlspecialchars($profile['vehicle_number']) ?>" required>
    </div>
    <div class="form-group">
        <label>Vehicle Image</label>
        
        <input type="file" class="form-control" name="vehicle_image" accept="image/*">
    </div>
    <button type="submit" class="btn btn-success btn-block">Save Vehicle Info</button>
</form>
</div>
<script>
    document.getElementById('driverForm').addEventListener('submit', function(e) {
        e.preventDefault();
        const formData = new FormData(this);
        fetch('../controllers/DriverController.php', {
            method: 'POST',
            body: formData,
        })
        .then(response => response.json())
        .then(data => {
            if (data.success) {
                alert('Driver information updated successfully!');
                location.reload();
            } else {
                alert('Failed to update driver information: ' + (data.error || 'Unknown error'));
            }
        })
        .catch(error => console.error('Error:', error));
    });
    document.getElementById('vehicleForm').addEventListener('submit', function(e) {
        e.preventDefault();
        const formData = new FormData(this);
        fetch('../controllers/DriverController.php', {
            method: 'POST',
            body: formData,
        })
        .then(response => response.json())
        .then(data => {
            if (data.success) {
                alert('Vehicle information updated successfully!');
                location.reload();
            } else {
                alert('Failed to update vehicle information: ' + (data.error || 'Unknown error'));
            }
        })
    });

```

```

        }
    })
    .catch(error => console.error('Error:', error));
});
</script>
</body>
</html>
home.php
<?php
require_once '../controllers/DriverController.php';
if (!isset($_SESSION['driver'])) {
    header('Location: ../login.php');
    exit;
}
$driverId = $_SESSION['driver']['id'];
$controller = new DriverController();
$data = $controller->getHomePageData($driverId);
$dashboardData = $data['dashboardData'];
$isAvailable = $data['availability'];
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['accept_ride_id'])) {
    $ridId = $_POST['accept_ride_id'];
    $result = $controller->handleAcceptRide($ridId, $driverId);
    if (!$result['success']) {
        $_SESSION['error'] = $result['error'];
    } else {
        $_SESSION['success'] = $result['message'];
    }
    header('Location: home.php');
    exit;
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Driver Dashboard</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
</head>
<body>
<div class="container mt-4">
    <!-- Dashboard Summary -->
    <div class="row">

```

```

<div class="col-md-4">
    <div class="card text-white bg-primary mb-3">
        <div class="card-header">Total Rides</div>
        <div class="card-body">
            <h4 class="card-title"><?= $dashboardData['totalRides'] ?></h4>
<div class="col-md-4">
    <div class="card text-white bg-success mb-3">
        <div class="card-header">Total Earnings</div>
        <div class="card-body">
            <h4 class="card-title">$<?= number_format($dashboardData['totalEarnings'], 2) ?></h4>
<div class="col-md-4">
    <div class="card <?= $isAvailable ? 'bg-success' : 'bg-danger' ?> text-white">
        <div class="card-header">Availability Status</div>
        <div class="card-body">
            <h4 class="card-title"><?= $isAvailable ? 'Available' : 'Unavailable' ?></h4>
<?php if (isset($_SESSION['error'])): ?>
    <div class="alert alert-danger">
        <?= htmlspecialchars($_SESSION['error']) ?>
        <?php unset($_SESSION['error']); ?>
    </div>
<?php endif; ?>
<?php if (isset($_SESSION['success'])): ?>
    <div class="alert alert-success">
        <?= htmlspecialchars($_SESSION['success']) ?>
        <?php unset($_SESSION['success']); ?>
    </div>
<?php endif; ?>
<!-- Pending Rides -->
<div class="card mt-4">
    <div class="card-header">
        <h4>Pending Rides</h4>
    </div>
    <div class="card-body">
        <table class="table table-striped">
            <thead>
                <tr>
                    <th>Ride ID</th>
                    <th>Customer</th>
                    <th>Pickup</th>
                    <th>Drop-off</th>
                    <th>Action</th>
                </tr>
            </thead>
            <tbody>
                <?php foreach ($dashboardData['pendingRides'] as $ride): ?>

```

```

<tr>
    <td><?= htmlspecialchars($ride['id']) ?></td>
    <td><?= htmlspecialchars($ride['customer_id']) ?></td>
    <td><?= htmlspecialchars($ride['pickup_location']) ?></td>
    <td><?= htmlspecialchars($ride['dropoff_location']) ?></td>
    <td>
        <form method="POST">
            <input type="hidden" name="action" value="acceptRide">
            <input type="hidden" name="ride_id" value="<?= htmlspecialchars($ride['id']) ?>">
            <input type="hidden" name="driver_id" value="<?= htmlspecialchars($driverId) ?>">
            <button type="submit" class="btn btn-success">Accept</button>
        </form></td> </tr>
    <?php endforeach; ?>
</tbody>
</table>
<div class="card mt-4">
    <div class="card-header">
        <h4>Earnings Overview</h4>
    </div>
    <div class="card-body">
        <canvas id="earningsChart"></canvas>
    </div>
</div>
<!-- Google Maps Live Tracking (Placeholder) -->
<div class="card mt-4">
    <div class="card-header">
        <h4>Live Driver Tracking</h4>
    </div>
    <div class="card-body">
        <iframe
src="https://maps.google.com/maps?q=Dhabalpur&t=&z=13&ie=UTF8&iwloc=&output=embed"
width="100%" height="300px" frameborder="0"></iframe>
    </div>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
earnings.php
<?php
require_once '../controllers/DriverController.php';
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">

```

```

<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Driver Earnings - ETaxi</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">
</head>
<body>
    <div class="container mt-4">
        <div class="card shadow-sm">
            <div class="card-header bg-primary text-white">
                <div class="d-flex justify-content-between align-items-center">
                    <h4 class="mb-0">My Earnings</h4>
                    <span class="badge bg-light text-primary">
                        Total Earnings: <?php
                            $total = array_sum(array_column($earnings, 'amount'));
                            echo '₹' . number_format($total, 2);
                        ?>
                    </span>
                </div>
            <div class="card-body">
                <?php if (count($earnings) > 0): ?>
                    <div class="table-responsive">
                        <table class="table table-hover table-bordered">
                            <thead class="table-light">
                                <tr>
                                    <th class="text-center">Payment ID</th>
                                    <th class="text-center">Ride ID</th>
                                    <th class="text-center">Amount</th>
                                    <th class="text-center">Payment Method</th>
                                    <th class="text-center">Status</th>
                                    <th class="text-center">Date</th>
                                </tr>
                            </thead>
                            <tbody>
                                <?php foreach ($earnings as $payment): ?>
                                    <tr>
                                        <td class="text-center"><?php echo htmlspecialchars($payment['payment_id']); ?></td>
                                        <td class="text-center"><?php echo htmlspecialchars($payment['ride_id']); ?></td>
                                        <td class="text-center text-success fw-bold">
                                            <?php echo '₹' . number_format($payment['amount'], 2); ?>
                                        </td>
                                        <td class="text-center">
                                            <span class="badge bg-info">
                                                <?php echo htmlspecialchars($payment['payment_method']); ?>
                                            </span>
                                        </td>
                                    </tr>
                                <?php endforeach; ?>
                            </tbody>
                        </table>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```

        </span>
    </td>
    <td class="text-center">
        <span class="badge <?php
            echo $payment['payment_status'] === 'completed' ? 'bg-success' :
            ($payment['payment_status'] === 'pending' ? 'bg-warning' : 'bg-danger');
        ?>">
            <?php echo htmlspecialchars($payment['payment_status']); ?>
        </span>
    </td>
    <td class="text-center">
        <?php
            $date = new DateTime($payment['payment_date']);
        <div class="alert alert-info text-center">
            <i class="fas fa-info-circle me-2"></i>No earnings found.
        </div>
        <?php endif; ?>
    <?php include '../includes/footer.php'; ?>
    <!-- Bootstrap JS Bundle with Popper -->

```

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>

```

</body>
</html>

my-rides.php
<?php
session_start();
if (!isset($_SESSION['driver'])) {
    header('Location: ../login.php');
    exit;
}
require_once '../controllers/RideController.php';
$controller = new RideController();
$driver_id = $_SESSION['driver']['id'];
$accepted_rides = $controller->getDriverRides($driver_id, 'accepted');
$paid_rides = $controller->getDriverRides($driver_id, 'paid');
$completed_rides = $controller->getDriverRides($driver_id, 'completed');
if (!$accepted_rides['success'] || !$completed_rides['success']) {
    $_SESSION['error'] = "Failed to fetch rides data";
    header('Location: dashboard.php');
    exit;
}
?>
<!DOCTYPE html>
<html lang="en">
<head>

```

```
<meta charset="UTF-8">
<title>My Rides</title>
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
<link rel="stylesheet" href="https://cdn.datatables.net/1.10.22/css/dataTables.bootstrap4.min.css">
</head>
<body>
<div class="container mt-5">
    <!-- Error/Success Messages -->
    <?php if (isset($_SESSION['error'])): ?>
        <div class="alert alert-danger">
            <?= htmlspecialchars($_SESSION['error']) ?>
            <?php unset($_SESSION['error']); ?>
        </div>
    <?php endif; ?>
    <?php if (isset($_SESSION['success'])): ?>
        <div class="alert alert-success">
            <?= htmlspecialchars($_SESSION['success']) ?>
            <?php unset($_SESSION['success']); ?>
        </div>
    <?php endif; ?>
    <div class="card mb-4">
        <div class="card-header">
            <h3>Paid Rides</h3>
        </div>
        <div class="card-body">
            <table class="table table-striped" id="acceptedRidesTable">
                <thead>
                    <tr>
                        <th>ID</th>
                        <th>Customer</th>
                        <th>Phone</th>
                        <th>Pickup</th>
                        <th>Dropoff</th>
                        <th>Fare</th>
                        <th>Distance</th>
                        <th>Request Time</th>
                        <th>Action</th>
                    </tr>
                </thead>
                <tbody>
                    <?php foreach ($paid_rides['rides'] as $ride): ?>
                    <tr>
                        <td><?= htmlspecialchars($ride['id']) ?></td>
```

```

<td><?= htmlspecialchars($ride['customer_name']) ?></td>
<td><?= htmlspecialchars($ride['customer_phone']) ?></td>
<td><?= htmlspecialchars($ride['pickup_location']) ?></td>
<td><?= htmlspecialchars($ride['dropoff_location']) ?></td>
<td>₹<?= htmlspecialchars($ride['fare']) ?></td>
<td><?= htmlspecialchars($ride['distance']) ?> km</td>
<td><?= htmlspecialchars($ride['request_time']) ?></td>
<td>
    <button class="btn btn-success btn-sm"
        onclick="completeRide(<?= $ride['id'] ?>)">Complete Ride</button>
</td>/tr> <?php endforeach; ?></tbody></table>
<div class="card mb-4">
    <div class="card-header">
        <h3>Accepted Rides</h3>
    </div>
    <div class="card-body">
        <table class="table table-striped" id="acceptedRidesTable">
            <thead>
                <tr>
                    <th>ID</th>
                    <th>Customer</th>
                    <th>Phone</th>
                    <th>Pickup</th>
                    <th>Dropoff</th>
                    <th>Fare</th>
                    <th>Distance</th>
                    <th>Request Time</th>
                </tr>
            </thead>
            <tbody>
                <?php foreach ($accepted_rides['rides'] as $ride): ?>
                <tr>
                    <td><?= htmlspecialchars($ride['id']) ?></td>
                    <td><?= htmlspecialchars($ride['customer_name']) ?></td>
                    <td><?= htmlspecialchars($ride['customer_phone']) ?></td>
                    <td><?= htmlspecialchars($ride['pickup_location']) ?></td>
                    <td><?= htmlspecialchars($ride['dropoff_location']) ?></td>
                    <td>₹<?= htmlspecialchars($ride['fare']) ?></td>
                    <td><?= htmlspecialchars($ride['distance']) ?> km</td>
                    <td><?= htmlspecialchars($ride['request_time']) ?></td>
                </tr>
            <?php endforeach; ?>
            </tbody>
        </table>
        <div class="card">

```

```

<div class="card-header">
    <h3>Completed Rides</h3>
</div>
<div class="card-body">
    <table class="table table-striped" id="completedRidesTable">
        <thead>
            <tr>
                <th>ID</th>
                <th>Customer</th>
                <th>Phone</th>
                <th>Pickup</th>
                <th>Dropoff</th>
                <th>Fare</th>
                <th>Distance</th>
                <th>Completed Time</th>
            </tr>
        </thead>
        <tbody>
            <?php foreach ($completed_rides['rides'] as $ride): ?>
            <tr>
                <td><?= htmlspecialchars($ride['id']) ?></td>
                <td><?= htmlspecialchars($ride['customer_name']) ?></td>
                <td><?= htmlspecialchars($ride['customer_phone']) ?></td>
                <td><?= htmlspecialchars($ride['pickup_location']) ?></td>
                <td><?= htmlspecialchars($ride['dropoff_location']) ?></td>
                <td>₹<?= htmlspecialchars($ride['fare']) ?></td>
                <td><?= htmlspecialchars($ride['distance']) ?> km</td>
                <td><?= htmlspecialchars($ride['complete_time']) ?></td>
            </tr>
            <?php endforeach; ?>
        </tbody>
    </table>
<div class="modal fade" id="otpModal" tabindex="-1">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title">Enter OTP to Complete Ride</h5>
                <button type="button" class="close" data-dismiss="modal">
                    <span>&times;</span>
                </button>
            </div>
            <div class="modal-body">
                <div class="form-group">
                    <label for="otpInput">Enter OTP received from customer</label>
                    <input type="text" id="otpInput" class="form-control">
                </div>
            </div>
        </div>
    </div>

```

```

        placeholder="Enter 6-digit OTP" maxlength="6" pattern="\d{6}>
    <div class="invalid-feedback">Please enter a valid 6-digit OTP</div>
    </div>
    <input type="hidden" id="rideld">
    <div id="otpError" class="alert alert-danger d-none"></div>
    </div>
    <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-dismiss="modal">Cancel</button>
        <button type="button" class="btn btn-primary" onclick="verifyOTP()">Verify &
Complete</button>
    <script>
$(document).ready(function() {
    $('#acceptedRidesTable').DataTable({
        order: [[7, 'desc']]
    });
    $('#completedRidesTable').DataTable({
        order: [[7, 'desc']]
    });
});
function completeRide(rideld) {
    $('#rideld').val(rideld);
    $('#otpInput').val("");
    $('#otpError').addClass('d-none');
    $('#otpInput').removeClass('is-invalid');
    $('#otpModal').modal('show');
}
function verifyOTP() {
    const rideld = $('#rideld').val();
    const otp = $('#otpInput').val();
    const errorDiv = $('#otpError');
    $('#otpInput').removeClass('is-invalid');
    errorDiv.addClass('d-none');
    if (!otp || !/\d{6}/.test(otp)) {
        $('#otpInput').addClass('is-invalid');
        return;
    }
    const verifyBtn = $('.modal-footer .btn-primary');
    verifyBtn.prop('disabled', true).html('<span class="spinner-border spinner-border-sm" role="status" aria-hidden="true"></span> Verifying...');

    fetch('../controllers/RideController.php', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/x-www-form-urlencoded',
        },
        body: `action=completeRide&ride_id=${rideld}&driver_id=<?= $driver_id ?>&otp=${otp}`
    })
}

```

```

        })
        .then(response => response.json())
        .then(data => {
            if (data.success) {
                location.reload();
            } else {
                errorDiv.removeClass('d-none').text(data.error || 'Failed to complete ride');
            }
        })
    </script>
</body>
</html>

```

Includes/database.php

```

<?php
class Database {
    private $host = "localhost";
    private $db_name = "etaxi";
    private $username = "root";
    private $password = "";
    private $conn;
    public function getConnection() {
        if ($this->conn !== null) {
            return $this->conn; // Return existing connection if already established
        }
        try {
            $this->conn = new mysqli($this->host, $this->username, $this->password, $this->db_name);
            if ($this->conn->connect_error) {
                throw new Exception("Connection failed: " . $this->conn->connect_error);
            }
        } catch (Exception $e) {
            error_log("Database Connection Error: " . $e->getMessage());
            return null; // Return null on failure
        }
        return $this->conn;
    }
}
?>

```

Models:

Admin.php

```

<?php
class Admin {
    private $conn;
    public function __construct($db) {
        $this->conn = $db;
    }
}

```

```

}

public function login($email, $password) {
    if (!$this->conn) { // Debugging: Check if connection exists
        error_log("Database connection is null in Admin class.");
        return ["success" => false, "error" => "Database connection error"];
    }
    $sql = "SELECT * FROM admins WHERE email = ?";
    $stmt = $this->conn->prepare($sql);
    if (!$stmt) {
        error_log("Prepare failed: " . $this->conn->error);
        return ["success" => false, "error" => "Database error"];
    }
    $stmt->bind_param("s", $email);
    $stmt->execute();
    $result = $stmt->get_result();
    $admin = $result->fetch_assoc();
    if ($admin && $password === $admin['password']) {
        return ["success" => true, "admin" => $admin, "role" => "admin"];
    } else {
        return ["success" => false, "error" => "Invalid email or password"];
    }
}

public function verifyPassword($adminPassword) {
    $admin = $_SESSION['admin']; // Assuming the entire admin data is stored in session
    if ($admin && $adminPassword === $admin['password']) {
        return true;
    } else {
        return false;
    }
}

public function getAdminById($id) {
    $sql = "SELECT id, name, email, phone, profile_image, created_at FROM admins WHERE id = ?";
    $stmt = $this->conn->prepare($sql);
    $stmt->bind_param("i", $id);
    $stmt->execute();
    $result = $stmt->get_result();
    return $result->fetch_assoc();
}

public function updateAdmin($id, $data) {
    $fields = [];
    $params = [];
    $types = "";
    foreach ($data as $key => $value) {
        $fields[] = "`$key` = ?";
        $params[] = $value;
        $types .= is_int($value) ? "i" : "s";
    }
}

```

```

    }
    $sql = "UPDATE admins SET " . implode(", ", $fields) . " WHERE id = ?";
    $params[] = $id;
    $types .= "i";
    $stmt = $this->conn->prepare($sql);
    if (!$stmt) {
        return ["success" => false, "error" => $this->conn->error];
    }
    $stmt->bind_param($types, ...$params);
    if ($stmt->execute()) {
        return ["success" => true];
    } else {
        return ["success" => false, "error" => $stmt->error];
    }
}
?>

```

Customer.php

```

<?php
class Customer {
    private $conn;
    private $table_name = "customers";
    public function __construct($db) {
        $this->conn = $db;
    }
    public function register($data) {
        $name = $data['name'];
        $email = $data['email'];
        $phone = $data['phone'];
        $password = $data['password'];
        $created_at = $data['created_at'];
        $check_sql = "SELECT id FROM " . $this->table_name . " WHERE email = ? OR phone = ?";
        $stmt = $this->conn->prepare($check_sql);
        if (!$stmt) {
            return ["success" => false, "error" => "Database error: " . $this->conn->error];
        }
        $stmt->bind_param("ss", $email, $phone);
        $stmt->execute();
        $stmt->store_result();
        if ($stmt->num_rows > 0) {
            return ["success" => false, "error" => "Email or phone already exists!"];
        }
        $sql = "INSERT INTO " . $this->table_name . " (name, email, phone, password, created_at) VALUES
        (?, ?, ?, ?, ?)";
        $stmt = $this->conn->prepare($sql);

```

```

if (!$stmt) {
    return ["success" => false, "error" => "Database error: " . $this->conn->error];
}
$stmt->bind_param("sssss", $name, $email, $phone, $password, $created_at);
if ($stmt->execute()) {
    return ["success" => true, "message" => "Registration successful"];
} else {
    return ["success" => false, "error" => "Error: " . $stmt->error];
}
}

public function login($email, $password) {
    $sql = "SELECT * FROM customers WHERE email = ?";
    $stmt = $this->conn->prepare($sql);
    if (!$stmt) {
        error_log("Prepare failed: " . $this->conn->error);
        return ["success" => false, "error" => "Database error"];
    }
    $stmt->bind_param("s", $email);
    $stmt->execute();
    $result = $stmt->get_result();
    $user = $result->fetch_assoc();
    if ($password === $user['password']) {
        return ["success" => true, "id" => $user['id'], "user" => $user, "role" => "customer"];
    } else {
        return ["success" => false, "error" => "Invalid email or password"];
    }
}

public function loginWithOtp($email, $otp) {
    $sql = "SELECT * FROM customers WHERE email = ?";
    $stmt = $this->conn->prepare($sql);
    if (!$stmt) {
        error_log("Prepare failed: " . $this->conn->error);
        return ["success" => false, "error" => "Database error"];
    }
    $stmt->bind_param("s", $email);
    $stmt->execute();
    $result = $stmt->get_result();
    $user = $result->fetch_assoc();
    if ($user && $_SESSION['reset_otp'] == $otp && $_SESSION['reset_email'] == $email) {
        return ["success" => true, "id" => $user['id'], "customer" => $user, "role" => "customer"];
    } else {
        return ["success" => false, "error" => "Invalid OTP"];
    }
}

public function getAllCustomers() {

```

```

$sql = "SELECT id, name, email, phone, profile_image, created_at FROM " . $this->table_name;
$stmt = $this->conn->prepare($sql);
if (!$stmt) {
    error_log("Prepare failed: " . $this->conn->error);
    return ["success" => false, "error" => "Database error"];
}
$stmt->execute();
$result = $stmt->get_result();
$customers = $result->fetch_all(MYSQLI_ASSOC);
return ["success" => true, "customers" => $customers];
}

public function deleteCustomer($customerId) {
    $sql = "DELETE FROM " . $this->table_name . " WHERE id = ?";
    $stmt = $this->conn->prepare($sql);
    if (!$stmt) {
        error_log("Prepare failed: " . $this->conn->error);
        return ["success" => false, "error" => "Database error"];
    }
    $stmt->bind_param("i", $customerId);
    if ($stmt->execute()) {
        return ["success" => true];
    } else {
        error_log("Execute failed: " . $stmt->error);
        return ["success" => false, "error" => "Error: " . $stmt->error];
    }
}

public function updateProfile($customerId, $name, $email, $phone, $profileImage) {
    $query = "UPDATE " . $this->table_name . "
        SET name = ?, email = ?, phone = ?, profile_image = ?
        WHERE id = ?";
    $stmt = $this->conn->prepare($query);
    $stmt->bind_param("ssssi", $name, $email, $phone, $profileImage, $customerId);
    if($stmt->execute()){
        return ["success" => true];
    } else {
        return ["success" => false, "error" => $stmt->error];
    }
}
?>

```

Driver.php

```

<?php
class Driver {
    private $conn;
    public function __construct($db) {

```

```

        $this->conn = $db;
    }

    public function register($data) {
        $name = $data['name'];
        $email = $data['email'];
        $password = password_hash($data['password'], PASSWORD_BCRYPT);
        $phone = $data['phone'];
        $id_number = $data['id_number'];
        $status = 'pending';
        $created_at = date('Y-m-d H:i:s');

        $check_driver_sql = "SELECT email, phone FROM drivers WHERE email = ? OR phone = ?";
        $stmt = $this->conn->prepare($check_driver_sql);
        if (!$stmt) {
            error_log("Prepare failed: " . $this->conn->error);
            return ["success" => false, "error" => "Database error"];
        }

        $stmt->bind_param("ss", $email, $phone);
        $stmt->execute();
        $stmt->store_result();
        if ($stmt->num_rows > 0) {
            return ["success" => false, "error" => "Email or phone number already exists!"];
        }

        $profile_image_name = $name . '_' . substr($phone, -4) . '_profile_' . time() . '.' .
pathinfo($_FILES['profile_image']['name'], PATHINFO_EXTENSION);

        $id_image_name = $name . '_' . substr($id_number, -4) . '_id_' . time() . '.' .
pathinfo($_FILES['id_image']['name'], PATHINFO_EXTENSION);

        $profile_image_tmp = $_FILES['profile_image']['tmp_name'];
        $profile_image_folder = './uploads/drivers/profile/' . $profile_image_name;
        move_uploaded_file($profile_image_tmp, $profile_image_folder);

        $id_image_tmp = $_FILES['id_image']['tmp_name'];
        $id_image_folder = './uploads/drivers/id/' . $id_image_name;
        move_uploaded_file($id_image_tmp, $id_image_folder);

        $sql_driver = "INSERT INTO drivers (name, email, password, phone, id_number, id_image, status,
profile_image, created_at)      VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)";
        $stmt = $this->conn->prepare($sql_driver);
        if (!$stmt) {
            error_log("Prepare failed: " . $this->conn->error);
            return ["success" => false, "error" => "Database error"];
        }

        $stmt->bind_param("ssssssss", $name, $email, $password, $phone, $id_number,
$id_image_name, $status, $profile_image_name, $created_at);
        if ($stmt->execute()) {
            $driver_id = $stmt->insert_id;
            $vehicle_type = $data['vehicle_type'];
            $vehicle_model = $data['vehicle_model'];
        }
    }
}

```

```

$vehicle_number = $data['vehicle_number'];
$license_number = $data['license_number'];
$check_vehicle_sql = "SELECT vehicle_number FROM vehicles WHERE vehicle_number = ?";
$stmt = $this->conn->prepare($check_vehicle_sql);
if (!$stmt) {
    error_log("Prepare failed: " . $this->conn->error);
    return ["success" => false, "error" => "Database error"];
}
$stmt->bind_param("s", $vehicle_number);
$stmt->execute();
$stmt->store_result();
if ($stmt->num_rows > 0) {
    return ["success" => false, "error" => "Vehicle number already exists!"];
}
$vehicle_image_name = $vehicle_number . '_vehicle_' . time() . '.' .
pathinfo($_FILES['vehicle_image']['name'], PATHINFO_EXTENSION);
$license_image_name = $license_number . '_license_' . time() . '.' .
pathinfo($_FILES['license_image']['name'], PATHINFO_EXTENSION);
$vehicle_image_tmp = $_FILES['vehicle_image']['tmp_name'];
$vehicle_image_folder = './uploads/drivers/vehicle/' . $vehicle_image_name;
move_uploaded_file($vehicle_image_tmp, $vehicle_image_folder);
$license_image_tmp = $_FILES['license_image']['tmp_name'];
$license_image_folder = './uploads/drivers/license/' . $license_image_name;
move_uploaded_file($license_image_tmp, $license_image_folder);
$sql_vehicle = "INSERT INTO vehicles (driver_id, license_number, license_image, vehicle_type,
vehicle_model, vehicle_number, vehicle_image, created_at)
VALUES (?, ?, ?, ?, ?, ?, ?, ?)";
$stmt = $this->conn->prepare($sql_vehicle);
if (!$stmt) {
    error_log("Prepare failed: " . $this->conn->error);
    return ["success" => false, "error" => "Database error"];
}
$stmt->bind_param("issssss", $driver_id, $license_number, $license_image_name,
$vehicle_type, $vehicle_model, $vehicle_number, $vehicle_image_name, $created_at);
if ($stmt->execute()) {
    return ["success" => true];
} else {
    error_log("Execute failed: " . $stmt->error);
    return ["success" => false, "error" => "Error: " . $stmt->error];
}
} else {
    error_log("Execute failed: " . $stmt->error);
    return ["success" => false, "error" => "Error: " . $stmt->error];
}
}
}

```

```

public function login($email, $password) {
    $sql = "SELECT * FROM drivers WHERE email = ?";
    $stmt = $this->conn->prepare($sql);
    if (!$stmt) {
        error_log("Prepare failed: " . $this->conn->error);
        return ["success" => false, "error" => "Database error"];
    }
    $stmt->bind_param("s", $email);
    $stmt->execute();
    $result = $stmt->get_result();
    $driver = $result->fetch_assoc();
    if ($driver && password_verify($password, $driver['password'])) {
        return ["success" => true, "driver" => $driver, "role" => "driver"];
    } else {
        return ["success" => false, "error" => "Invalid email or password"];
    }
}

public function loginWithOtp($phone, $otp) {
    $sql = "SELECT * FROM drivers WHERE phone = ?";
    $stmt = $this->conn->prepare($sql);
    if (!$stmt) {
        error_log("Prepare failed: " . $this->conn->error);
        return ["success" => false, "error" => "Database error"];
    }
    $stmt->bind_param("s", $phone);
    $stmt->execute();
    $result = $stmt->get_result();
    $driver = $result->fetch_assoc();
    if ($driver && isset($_SESSION['otp']) && isset($_SESSION['otp_phone']) && $_SESSION['otp'] == $otp && $_SESSION['otp_phone'] == $phone) {
        return ["success" => true, "id" => $driver['id'], "driver" => $driver, "role" => "driver"];
    } else {
        return ["success" => false, "error" => "Invalid OTP"];
    }
}

public function getAllDrivers() {
    $sql = "SELECT * FROM drivers";
    $stmt = $this->conn->prepare($sql);
    if (!$stmt) {
        error_log("Prepare failed: " . $this->conn->error);
        return ["success" => false, "error" => "Database error"];
    }
    $stmt->execute();
    $result = $stmt->get_result();
    $drivers = $result->fetch_all(MYSQLI_ASSOC);
    return ["success" => true, "drivers" => $drivers];
}

```

```

    }

    public function getDriverById($driver_id) {
        $sql = "SELECT * FROM drivers WHERE id = ?";
        $stmt = $this->conn->prepare($sql);
        if (!$stmt) {
            error_log("Prepare failed (getDriverById): " . $this->conn->error);
            return ["success" => false, "error" => "Database error"];
        }
        $stmt->bind_param("i", $driver_id);
        $stmt->execute();
        $result = $stmt->get_result();
        $driver = $result->fetch_assoc();
        if (!$driver) {
            error_log("No driver found for ID: $driver_id");
            return ["success" => false, "error" => "Driver not found"];
        }
        return ["success" => true, "driver" => $driver];
    }

    public function getDriverVehicles($driver_id) {
        $sql = "SELECT * FROM vehicles WHERE driver_id = ?";
        $stmt = $this->conn->prepare($sql);
        if (!$stmt) {
            error_log("Prepare failed (getDriverVehicles): " . $this->conn->error);
            return ["success" => false, "error" => "Database error"];
        }
        $stmt->bind_param("i", $driver_id);
        $stmt->execute();
        $result = $stmt->get_result();
        $vehicles = $result->fetch_all(MYSQLI_ASSOC);
        if (!$vehicles) {
            error_log("No vehicles found for driver ID: $driver_id");
        }
        return ["success" => true, "vehicles" => $vehicles];
    }

    $stmt->bind_param("sssssi", $driverData['name'], $driverData['email'], $driverData['phone'],
$driverData['id_number'], $driverData['status'], $driverId);
    if ($stmt->execute()) {
        return ["success" => true];
    } else {
        error_log("Execute failed (updateDriver): " . $stmt->error);
        return ["success" => false, "error" => "Error: " . $stmt->error];
    }
}

public function deleteDriver($driverId) {
    $sql = "DELETE FROM drivers WHERE id = ?";

```

```

$stmt = $this->conn->prepare($sql);
if (!$stmt) {
    error_log("Prepare failed (deleteDriver): " . $this->conn->error);
    return ["success" => false, "error" => "Database error"];
}
$stmt->bind_param("i", $driverId);
if ($stmt->execute()) {
    return ["success" => true];
} else {
    error_log("Execute failed (deleteDriver): " . $stmt->error);
    return ["success" => false, "error" => "Error: " . $stmt->error];
}
}

public function approveDriver($driverId) {
try {
    $this->conn->begin_transaction();
    $sql = "UPDATE drivers SET status = 'approved' WHERE id = ?";
    $stmt = $this->conn->prepare($sql);
    if (!$stmt) {
        throw new Exception("Prepare failed (approveDriver): " . $this->conn->error);
    }
    $stmt->bind_param("i", $driverId);
    $stmt->execute();
    $check_sql = "SELECT id FROM driver_availability WHERE driver_id = ?";
    $stmt = $this->conn->prepare($check_sql);
    if (!$stmt) {
        throw new Exception("Prepare failed (check availability): " . $this->conn->error);
    }
    $stmt->bind_param("i", $driverId);
    $stmt->execute();
    $result = $stmt->get_result();
    if ($result->num_rows === 0) {
        $insert_sql = "INSERT INTO driver_availability (driver_id, is_available, updated_at) VALUES (?, 1, NOW())";
        $stmt = $this->conn->prepare($insert_sql);
        if (!$stmt) {
            throw new Exception("Prepare failed (insert availability): " . $this->conn->error);
        }
        $stmt->bind_param("i", $driverId);
        $stmt->execute();
    }
    $this->conn->commit();
    return ["success" => true];
} catch (Exception $e) {
    $this->conn->rollback();
}
}

```

```

        error_log("Approve driver failed: " . $e->getMessage());
        return ["success" => false, "error" => $e->getMessage()];
    }
}

public function rejectDriver($driverId) {
    $sql = "UPDATE drivers SET status = 'rejected' WHERE id = ?";
    $stmt = $this->conn->prepare($sql);
    if (!$stmt) {
        error_log("Prepare failed (rejectDriver): " . $this->conn->error);
        return ["success" => false, "error" => "Database error"];
    }
    $stmt->bind_param("i", $driverId);
    if ($stmt->execute()) {
        return ["success" => true];
    } else {
        error_log("Execute failed (rejectDriver): " . $stmt->error);
        return ["success" => false, "error" => "Error: " . $stmt->error];
    }
}

public function getDriverProfile($driver_id) {
    $sql = "SELECT d.*, v.* FROM drivers d
        LEFT JOIN vehicles v ON d.id = v.driver_id
        WHERE d.id = ?";
    $stmt = $this->conn->prepare($sql);

    if (!$stmt) {
        error_log("Prepare failed: " . $this->conn->error);
        return ["success" => false, "error" => "Database error"];
    }
    $stmt->bind_param("i", $driver_id);
    $stmt->execute();
    $result = $stmt->get_result();
    $profile = $result->fetch_assoc();

    return ["success" => true, "profile" => $profile];
}

public function updateProfile($driver_id, $data, $files) {
    try {
        $this->conn->begin_transaction();
        $sql_driver = "UPDATE drivers SET name = ?, email = ?, phone = ?, id_number =
? WHERE id = ?";
        $stmt = $this->conn->prepare($sql_driver);
        if (!$stmt) {
            throw new Exception("Prepare failed: " . $this->conn->error);
        }
    }
}

```

```

$stmt->bind_param("ssssi",
    $data['name'],
    $data['email'],
    $data['phone'],
    $data['id_number'],
    $driver_id
);
$stmt->execute();
if (!empty($files['profile_image']['name'])) {
    $profile_image = $this->uploadImage($files['profile_image'], 'profile', $data['name']);
    $sql = "UPDATE drivers SET profile_image = ? WHERE id = ?";
    $stmt = $this->conn->prepare($sql);
    if (!$stmt) {
        throw new Exception("Prepare failed: " . $this->conn->error);
    }
    $stmt->bind_param("si", $profile_image, $driver_id);
    $stmt->execute();
}
if (!empty($files['id_image']['name'])) {
    $id_image = $this->uploadImage($files['id_image'], 'id', $data['id_number']);
    $sql = "UPDATE drivers SET id_image = ? WHERE id = ?";
    $stmt = $this->conn->prepare($sql);
    if (!$stmt) {
        throw new Exception("Prepare failed: " . $this->conn->error);
    }
    $stmt->bind_param("si", $id_image, $driver_id);
    $stmt->execute();
}
$this->conn->commit();
return ["success" => true];
} catch (Exception $e) {
    $this->conn->rollback();
    error_log("Profile update failed: " . $e->getMessage());
    return ["success" => false, "error" => $e->getMessage()];
}
}

private function uploadImage($file, $type, $prefix) {
    $upload_dir = '../uploads/drivers/' . $type . '/';
    $extension = pathinfo($file['name'], PATHINFO_EXTENSION);
    $filename = $prefix . '_' . $type . '_' . time() . '.' . $extension;
    $target_path = $upload_dir . $filename;
    if (!move_uploaded_file($file['tmp_name'], $target_path)) {
        throw new Exception("Failed to upload " . $type . " image");
    }
    return $filename;
}

```

```

    }

    public function updateVehicle($driver_id, $data, $files) {
        try {
            $this->conn->begin_transaction();
            $check_sql = "SELECT id FROM vehicles WHERE driver_id = ?";
            $stmt = $this->conn->prepare($check_sql);
            if (!$stmt) {
                throw new Exception("Prepare failed: " . $this->conn->error);
            }
            $stmt->bind_param("i", $driver_id);
            $stmt->execute();
            $result = $stmt->get_result();
            if ($result->num_rows > 0) {
                $sql = "UPDATE vehicles SET license_number = ?, vehicle_type = ?, vehicle_model = ?, vehicle_number = ?"
                WHERE driver_id = ?";
                $stmt = $this->conn->prepare($sql);
                if (!$stmt) {
                    throw new Exception("Prepare failed: " . $this->conn->error);
                }
                $stmt->bind_param("ssssi",
                    $data['license_number'],
                    $data['vehicle_type'],
                    $data['vehicle_model'],
                    $data['vehicle_number'],
                    $driver_id
                );
            } else {
                $sql = "INSERT INTO vehicles (driver_id, license_number, vehicle_type, vehicle_model, vehicle_number)
VALUES (?, ?, ?, ?, ?)";
                $stmt = $this->conn->prepare($sql);
                if (!$stmt) {
                    throw new Exception("Prepare failed: " . $this->conn->error);
                }
                $stmt->bind_param("issss",
                    $driver_id,
                    $data['license_number'],
                    $data['vehicle_type'],
                    $data['vehicle_model'],
                    $data['vehicle_number']
                );
            }
            $stmt->execute();
            if (!empty($files['vehicle_image']['name'])) {

```

```

        $vehicle_image = $this->uploadImage($files['vehicle_image'], 'vehicle',
$data['vehicle_number']);
        $sql = "UPDATE vehicles SET vehicle_image = ? WHERE driver_id = ?";
$stmt = $this->conn->prepare($sql);
if (!$stmt) {
    throw new Exception("Prepare failed: " . $this->conn->error);
}
$stmt->bind_param("si", $vehicle_image, $driver_id);
$stmt->execute();
}
if (!empty($files['license_image'])['name'])) {
    $license_image = $this->uploadImage($files['license_image'], 'license',
$data['license_number']);
    $sql = "UPDATE vehicles SET license_image = ? WHERE driver_id = ?";
$stmt = $this->conn->prepare($sql);
if (!$stmt) {
    throw new Exception("Prepare failed: " . $this->conn->error);
}
$stmt->bind_param("si", $license_image, $driver_id);
$stmt->execute();
}
$this->conn->commit();
return ["success" => true];
}
}
?>

```

Ride.php

```

<?php
require_once __DIR__ . '/Driver.php';
class Ride {
    private $conn;
    private $table_name = "rides";
    public function __construct($db) {
        $this->conn = $db;
    }
    public function getAllRides() {
        $sql = "SELECT * FROM " . $this->table_name . " ORDER BY id DESC";
        $stmt = $this->conn->prepare($sql);
        if (!$stmt) {
            error_log("Prepare failed: " . $this->conn->error);
            return ["success" => false, "error" => "Database error"];
        }
        $stmt->execute();
        $result = $stmt->get_result();
        $rides = $result->fetch_all(MYSQLI_ASSOC);
    }
}

```

```

        return ["success" => true, "rides" => $rides];
    }
    public function getRideDetails($ride_id) {
        $sql = "SELECT r.*,
                    c.name AS customer_name, c.email AS customer_email, c.phone AS customer_phone,
                    c.profile_image AS customer_profile,
                    d.name AS driver_name, d.email AS driver_email, d.phone AS driver_phone,
                    d.profile_image AS driver_profile
                FROM " . $this->table_name . " r
                JOIN customers c ON r.customer_id = c.id
                JOIN drivers d ON r.driver_id = d.id
                WHERE r.id = ?";
        $stmt = $this->conn->prepare($sql);
        if (!$stmt) {
            error_log("Prepare failed: " . $this->conn->error);
            return ["success" => false, "error" => "Database error"];
        }
        $stmt->bind_param("i", $ride_id);
        $stmt->execute();
        $result = $stmt->get_result();
        $ride = $result->fetch_assoc();
        if (!$ride) {
            return ["success" => false, "error" => "Ride not found"];
        }
        $reviews = $this->getRideReviews($ride_id);
        return ["success" => true, "ride" => $ride, "reviews" => $reviews];
    }
    private function getRideReviews($ride_id) {
        $sql = "SELECT id, ride_id, customer_id, driver_id, rating, review, created_at FROM reviews WHERE
ride_id = ?";
        $stmt = $this->conn->prepare($sql);
        if (!$stmt) {
            error_log("Prepare failed: " . $this->conn->error);
            return [];
        }
        $stmt->bind_param("i", $ride_id);
        $stmt->execute();
        $result = $stmt->get_result();
        return $result->fetch_all(MYSQLI_ASSOC);
    }
    public function bookRide($customer_id, $pickup_location, $dropoff_location, $fare, $distance,
$rate) {
        $sql = "INSERT INTO " . $this->table_name . "
                (customer_id, pickup_location, dropoff_location, ride_status, fare, request_time, distance,
                rate)

```

```

        VALUES (?, ?, ?, 'pending', ?, NOW(), ?, ?)");
$stmt = $this->conn->prepare($sql);
if (!$stmt) {
    error_log("Prepare failed: " . $this->conn->error);
    return ["success" => false, "error" => "Database error"];
}
$stmt->bind_param("issddi", $customer_id, $pickup_location, $dropoff_location, $fare,
$distance, $rate);
if ($stmt->execute()) {
    return ["success" => true, "ride_id" => $this->conn->insert_id];
} else {
    return ["success" => false, "error" => "Failed to book ride"];
}
}

public function getRidesByCustomer($customerId) {
$query = "SELECT r.id, r.driver_id, r.pickup_location, r.dropoff_location, r.ride_status, r.fare,
r.request_time, r.distance, r.rate,
d.name AS driver_name
FROM " . $this->table_name . " r
LEFT JOIN drivers d ON r.driver_id = d.id
WHERE r.customer_id = ?";
$stmt = $this->conn->prepare($query);
$stmt->bind_param("i", $customerId);
$stmt->execute();
$result = $stmt->get_result();
$rides = $result->fetch_all(MYSQLI_ASSOC);
return $rides;
}

public function getLatestRidesByCustomer($customerId, $limit = 5) {
$query = "SELECT r.id, r.driver_id, r.pickup_location, r.dropoff_location, r.ride_status, r.fare,
r.request_time,
d.name AS driver_name
FROM " . $this->table_name . " r
LEFT JOIN drivers d ON r.driver_id = d.id
WHERE r.customer_id = ?
ORDER BY r.request_time DESC
LIMIT ?";
$stmt = $this->conn->prepare($query);
$stmt->bind_param("ii", $customerId, $limit);
$stmt->execute();
$result = $stmt->get_result();
return $result->fetch_all(MYSQLI_ASSOC);
}

public function cancelRide($rideId) {
$query = "UPDATE " . $this->table_name . " SET ride_status = 'cancelled' WHERE id = ?";

```

```

$stmt = $this->conn->prepare($query);
$stmt->bind_param("i", $rideld);
if ($stmt->execute()) {
    return ["success" => true];
} else {
    return ["success" => false, "error" => $stmt->error];
}
}

public function getTotalRidesByDriver($driver_id) {
    $sql = "SELECT COUNT(*) as totalRides FROM " . $this->table_name . " WHERE driver_id = ?";
    $stmt = $this->conn->prepare($sql);
    if (!$stmt) {
        error_log("Prepare failed: " . $this->conn->error);
        return ["success" => false, "error" => "Database error"];
    }
    $stmt->bind_param("i", $driver_id);
    $stmt->execute();
    $result = $stmt->get_result();
    $totalRides = $result->fetch_assoc()['totalRides'];
    return ["success" => true, "totalRides" => $totalRides];
}

public function getPendingRidesByDriver($driver_id) {
    $sql = "SELECT * FROM " . $this->table_name . " WHERE driver_id = ? AND ride_status = 'pending'";
    $stmt = $this->conn->prepare($sql);
    if (!$stmt) {
        error_log("Prepare failed: " . $this->conn->error);
        return ["success" => false, "error" => "Database error"];
    }
    $stmt->bind_param("i", $driver_id);
    $stmt->execute();
    $result = $stmt->get_result();
    $pendingRides = $result->fetch_all(MYSQLI_ASSOC);
    return ["success" => true, "pendingRides" => $pendingRides];
}

public function getAllPendingRides() {
    $sql = "SELECT * FROM " . $this->table_name . " WHERE ride_status = 'pending'";
    $stmt = $this->conn->prepare($sql);
    if (!$stmt) {
        error_log("Prepare failed: " . $this->conn->error);
        return ["success" => false, "error" => "Database error"];
    }
    $stmt->execute();
    $result = $stmt->get_result();
    $pendingRides = $result->fetch_all(MYSQLI_ASSOC);
    return ["success" => true, "pendingRides" => $pendingRides];
}

```

```

    }

    public function acceptRide($ride_id, $driver_id) {
        try {
            $sql = "UPDATE " . $this->table_name . " SET ride_status = 'accepted', driver_id = ? WHERE id = ?";
            $stmt = $this->conn->prepare($sql);
            if (!$stmt) {
                throw new Exception("Prepare failed: " . $this->conn->error);
            }
            $stmt->bind_param("ii", $driver_id, $ride_id);
            if (!$stmt->execute()) {
                throw new Exception("Failed to update ride status: " . $stmt->error);
            }
            $otp = rand(100000, 999999);
            $updateOtpSql = "UPDATE " . $this->table_name . " SET OTP = ? WHERE id = ?";
            $updateOtpStmt = $this->conn->prepare($updateOtpSql);
            if (!$updateOtpStmt) {
                throw new Exception("Prepare failed for OTP update: " . $this->conn->error);
            }
            $updateOtpStmt->bind_param("ii", $otp, $ride_id);
            if (!$updateOtpStmt->execute()) {
                throw new Exception("Failed to update OTP: " . $updateOtpStmt->error);
            }
            $driverModel = new Driver($this->conn);
            $availabilityResult = $driverModel->updateDriverAvailability($driver_id, 0);
            if (!$availabilityResult['success']) {
                throw new Exception("Failed to update driver availability: " . $availabilityResult['error']);
            }
            if ($rideDetails['success']) {
                $this->sendMessageToCustomer($rideDetails['ride'], $otp);
            }
            return ["success" => true];
        } catch (Exception $e) {
            error_log("Accept ride failed: " . $e->getMessage());
            return ["success" => false, "error" => $e->getMessage()];
        }
    }

    public function getDriverRides($driver_id, $status) {
        $sql = "SELECT r.*, c.name as customer_name, c.phone as customer_phone
                FROM " . $this->table_name . " r
                JOIN customers c ON r.customer_id = c.id
                WHERE r.driver_id = ? AND r.ride_status = ?
                ORDER BY r.request_time DESC";
        $stmt = $this->conn->prepare($sql);
        if (!$stmt) {

```

```

        error_log("Prepare failed: " . $this->conn->error);
        return ["success" => false, "error" => "Database error"];
    }
    $stmt->bind_param("is", $driver_id, $status);
    $stmt->execute();
    $result = $stmt->get_result();
    $rides = $result->fetch_all(MYSQLI_ASSOC);
    return ["success" => true, "rides" => $rides];
}

public function completeRide($ride_id, $driver_id, $otp) {
    if (!is_numeric($ride_id) || !is_numeric($driver_id) || !is_numeric($otp)) {
        return ["success" => false, "error" => "Invalid input parameters"];
    }
    $verify_sql = "SELECT OTP, ride_status FROM " . $this->table_name . "
                  WHERE id = ? AND driver_id = ?";
    $stmt = $this->conn->prepare($verify_sql);
    if (!$stmt) {
        error_log("Prepare failed: " . $this->conn->error);
        return ["success" => false, "error" => "Database error: " . $this->conn->error];
    }
    $stmt->bind_param("ii", $ride_id, $driver_id);
    if (!$stmt->execute()) {
        error_log("Execute failed: " . $stmt->error);
        return ["success" => false, "error" => "Database error: " . $stmt->error];
    }
    $result = $stmt->get_result();
    $ride = $result->fetch_assoc();
    if (!$ride) {
        return ["success" => false, "error" => "Ride not found"];
    }
    if ($ride['ride_status'] !== 'paid') {
        return ["success" => false, "error" => "Ride is not in paid state"];
    }
    if ($ride['OTP'] != $otp) {
        return ["success" => false, "error" => "Invalid OTP"];
    }
    $this->conn->begin_transaction();
    try {
        $update_sql = "UPDATE " . $this->table_name . "
                      SET ride_status = 'completed', complete_time = NOW(), OTP = NULL
                      WHERE id = ?";
        $stmt = $this->conn->prepare($update_sql);
        if (!$stmt) {
            throw new Exception("Failed to prepare ride update statement: " . $this->conn->error);
        }
    }
}

```

```

$stmt->bind_param("i", $ride_id);
if (!$stmt->execute()) {
    throw new Exception("Failed to update ride status: " . $stmt->error);
}
$driver = new Driver($this->conn);
$driverUpdateResult = $driver->updateDriverAvailability($driver_id, 1);
if (!$driverUpdateResult['success']) {
    throw new Exception("Failed to update driver availability: " . ($driverUpdateResult['error'] ?? 'Unknown error'));
}
$this->conn->commit();
return ["success" => true];
} catch (Exception $e) {
    $this->conn->rollback();
    error_log("Complete ride failed: " . $e->getMessage());
    error_log("Stack trace: " . $e->getTraceAsString());
    return ["success" => false, "error" => "Failed to complete ride: " . $e->getMessage()];
}
}

public function markRideAsPaid($ride_id) {
    $sql = "UPDATE " . $this->table_name . " SET ride_status = 'paid' WHERE id = ?";
    $stmt = $this->conn->prepare($sql);
    if (!$stmt) {
        error_log("Prepare failed: " . $this->conn->error);
        return ["success" => false, "error" => "Database error"];
    }
    $stmt->bind_param("i", $ride_id);
    if ($stmt->execute()) {
        return ["success" => true];
    } else {
        return ["success" => false, "error" => $stmt->error];
    }
}
}

?>

Report.php
<?php
class Statistics {
    private $db;
    public function __construct($dbConnection) {
        $this->db = $dbConnection;
    }
    public function getTotalDrivers() {
        $sql = "SELECT COUNT(*) as total FROM drivers";
        $result = $this->db->query($sql);

```

```

        $row = $result->fetch_assoc();
        return $row['total'];
    }

    public function getTotalPayments() {
        $sql = "SELECT SUM(amount) as total FROM payments";
        $result = $this->db->query($sql);
        $row = $result->fetch_assoc();
        return $row['total'];
    }

    public function getTotalKilometers() {
        $sql = "SELECT SUM(distance) as total FROM rides";
        $result = $this->db->query($sql);
        $row = $result->fetch_assoc();
        return $row['total'];
    }

    public function getTotalCustomers() {
        $sql = "SELECT COUNT(*) as total FROM customers";
        $result = $this->db->query($sql);
        $row = $result->fetch_assoc();
        return $row['total'];
    }

    public function getMonthlyRides($year, $month) {
        $sql = "SELECT COUNT(*) as total_rides, SUM(distance) as total_distance
                FROM rides
                WHERE MONTH(request_time) = ? AND YEAR(request_time) = ?";
        $stmt = $this->db->prepare($sql);
        $stmt->bind_param("ii", $month, $year);
        $stmt->execute();
        $result = $stmt->get_result();
        return $result->fetch_assoc();
    }

    public function getMonthlyEarnings($year, $month) {
        $sql = "SELECT COUNT(*) as total_payments, SUM(amount) as total_amount
                FROM payments
                WHERE MONTH(payment_date) = ? AND YEAR(payment_date) = ?";
        $stmt = $this->db->prepare($sql);
        $stmt->bind_param("ii", $month, $year);
        $stmt->execute();
        $result = $stmt->get_result();
        return $result->fetch_assoc();
    }

    public function getMonthlyCustomers($year, $month) {
        $sql = "SELECT COUNT(DISTINCT customer_id) as total_customers
                FROM rides
                WHERE MONTH(request_time) = ? AND YEAR(request_time) = ?";
    }
}

```

```

$stmt = $this->db->prepare($sql);
$stmt->bind_param("ii", $month, $year);
$stmt->execute();
$result = $stmt->get_result();
return $result->fetch_assoc();
}

public function getMonthlyStats($year, $month) {
    $rides = $this->getMonthlyRides($year, $month);
    $earnings = $this->getMonthlyEarnings($year, $month);
    $customers = $this->getMonthlyCustomers($year, $month);
    return [
        'rides' => $rides['total_rides'] ?? 0,
        'distance' => $rides['total_distance'] ?? 0,
        'payments' => $earnings['total_payments'] ?? 0,
        'amount' => $earnings['total_amount'] ?? 0,
        'customers' => $customers['total_customers'] ?? 0
    ];
}

public function getYearlyStats($year) {
    $sql = "SELECT
        MONTH(request_time) as month,
        COUNT(*) as total_rides,
        SUM(distance) as total_distance,
        COUNT(DISTINCT customer_id) as total_customers
    FROM rides
    WHERE YEAR(request_time) = ?
    GROUP BY MONTH(request_time)
    ORDER BY month";
    $stmt = $this->db->prepare($sql);
    $stmt->bind_param("i", $year);
    $stmt->execute();
    $result = $stmt->get_result();
    $rides = [];
    while ($row = $result->fetch_assoc()) {
        $rides[$row['month']] = $row;
    }
    $sql = "SELECT
        MONTH(payment_date) as month,
        COUNT(*) as total_payments,
        SUM(amount) as total_amount
    FROM payments
    WHERE YEAR(payment_date) = ?
    GROUP BY MONTH(payment_date)
    ORDER BY month";
    $stmt = $this->db->prepare($sql);
}

```

```

$stmt->bind_param("i", $year);
$stmt->execute();
$result = $stmt->get_result();
$payments = [];
while ($row = $result->fetch_assoc()) {
    $payments[$row['month']] = $row;
}
$yearlyStats = [];
for ($month = 1; $month <= 12; $month++) {
    $yearlyStats[$month] = [
        'rides' => $rides[$month]['total_rides'] ?? 0,
        'distance' => $rides[$month]['total_distance'] ?? 0,
        'customers' => $rides[$month]['total_customers'] ?? 0,
        'payments' => $payments[$month]['total_payments'] ?? 0,
        'amount' => $payments[$month]['total_amount'] ?? 0
    ];
}
return $yearlyStats;
}
}
?>

```

Payment.php

```

<?php
class Payment {
    private $conn;
    private $table_name = "payments";
    public function __construct($db) {
        $this->conn = $db;
    }
    public function getAllPayments() {
        $sql = "SELECT p.id, p.ride_id, p.amount, p.payment_method, p.payment_status, p.payment_date,
                c.name as customer_name, d.name as driver_name
                FROM " . $this->table_name . " p
                JOIN customers c ON p.customer_id = c.id
                JOIN rides r ON p.ride_id = r.id
                JOIN drivers d ON r.driver_id = d.id";
        $stmt = $this->conn->prepare($sql);
        if (!$stmt) {
            error_log("Prepare failed: " . $this->conn->error);
            return ["success" => false, "error" => "Database error"];
        }
        $stmt->execute();
        $result = $stmt->get_result();
        $payments = $result->fetch_all(MYSQLI_ASSOC);
        return ["success" => true, "payments" => $payments];
    }
}

```

```

    }

    public function getTotalEarningsByDriver($driver_id) {
        $sql = "SELECT SUM(amount) as totalEarnings FROM " . $this->table_name . " WHERE driver_id = ?";
        $stmt = $this->conn->prepare($sql);
        if (!$stmt) {
            error_log("Prepare failed: " . $this->conn->error);
            return ["success" => false, "error" => "Database error"];
        }
        $stmt->bind_param("i", $driver_id);
        $stmt->execute();
        $result = $stmt->get_result();
        $totalEarnings = $result->fetch_assoc()['totalEarnings'];
        return ["success" => true, "totalEarnings" => $totalEarnings];
    }

    public function getTotalPaymentsByCustomer($customer_id) {
        $sql = "SELECT SUM(amount) as totalPayments FROM " . $this->table_name . " WHERE customer_id = ?";
        $stmt = $this->conn->prepare($sql);
        if (!$stmt) {
            error_log("Prepare failed: " . $this->conn->error);
            return 0;
        }
        $stmt->bind_param("i", $customer_id);
        $stmt->execute();
        $result = $stmt->get_result();
        $row = $result->fetch_assoc();
        return $row['totalPayments'] ?? 0;
    }

    public function savePayment($paymentData) {
        $sql = "INSERT INTO " . $this->table_name . "
            (ride_id, driver_id, customer_id, amount, payment_id, payment_method, payment_status,
            payment_date)
            VALUES (?, ?, ?, ?, ?, ?, ?, ?)";

        $stmt = $this->conn->prepare($sql);
        if (!$stmt) {
            error_log("Prepare failed: " . $this->conn->error);
            return ["success" => false, "error" => "Prepare failed: " . $this->conn->error];
        }
        $stmt->bind_param("iiidssss",
            $paymentData['ride_id'],
            $paymentData['driver_id'],
            $paymentData['customer_id'],
            $paymentData['amount'],

```

```

$paymentData['payment_id'],
$paymentData['payment_method'],
$paymentData['payment_status'],
$paymentData['payment_date']
);
if ($stmt->execute()) {
    return ["success" => true];
} else {
    return ["success" => false, "error" => "Execute failed: " . $stmt->error];
}
}

public function getAllPaymentsByCustomer($customer_id) {
    $sql = "SELECT * FROM " . $this->table_name . " WHERE customer_id = ?";
    $stmt = $this->conn->prepare($sql);
    if (!$stmt) {
        error_log("Prepare failed: " . $this->conn->error);
        return ["success" => false, "error" => "Database error"];
    }
    $stmt->bind_param("i", $customer_id);
    $stmt->execute();
    $result = $stmt->get_result();
    $payments = $result->fetch_all(MYSQLI_ASSOC);
    return ["success" => true, "payments" => $payments];
}

public function getPaymentByRideId($rideId) {
    $sql = "SELECT * FROM " . $this->table_name . " WHERE ride_id = ?";
    $stmt = $this->conn->prepare($sql);
    if (!$stmt) {
        error_log("Prepare failed: " . $this->conn->error);
        return ["success" => false, "error" => "Database error"];
    }
    $stmt->bind_param("i", $rideId);
    $stmt->execute();
    $result = $stmt->get_result();
    $payment = $result->fetch_assoc();
    if (!$payment) {
        return ["success" => false, "error" => "Payment not found"];
    }
    return ["success" => true, "payment" => $payment];
}

public function updateInvoicePath($paymentId, $invoicePath) {
    $sql = "UPDATE " . $this->table_name . " SET invoice = ? WHERE id = ?";
    $stmt = $this->conn->prepare($sql);
    if (!$stmt) {
        error_log("Prepare failed: " . $this->conn->error);
    }
}

```

```

        return ["success" => false, "error" => "Database error"];
    }
    $stmt->bind_param("si", $invoicePath, $paymentId);
    if ($stmt->execute()) {
        return ["success" => true];
    } else {
        return ["success" => false, "error" => $stmt->error];
    }
}
?>

```

Notification.php

```

<?php
class Notification {
    private $conn;
    private $table_name = "notifications";
    public function __construct($db) {
        $this->conn = $db;
    }
    public function getAllNotifications() {
        $query = "SELECT * FROM " . $this->table_name . " ORDER BY created_at DESC";
        $stmt = $this->conn->prepare($query);

        if (!$stmt) {
            error_log("Prepare failed: " . $this->conn->error);
            return ["success" => false, "error" => "Database error"];
        }
        $stmt->execute();
        $result = $stmt->get_result();
        $notifications = $result->fetch_all(MYSQLI_ASSOC);
        return ["success" => true, "notifications" => $notifications];
    }
    $stmt->bind_param("is", $created_by, $message);
    if ($stmt->execute()) {
        return ["success" => true];
    } else {
        return ["success" => false, "error" => $stmt->error];
    }
}
public function deleteNotification($id) {
    $query = "DELETE FROM " . $this->table_name . " WHERE id = ?";
    $stmt = $this->conn->prepare($query);
    $stmt->bind_param("i", $id);
    if ($stmt->execute()) {
        return ["success" => true];
    } else {

```

```

        return ["success" => false, "error" => $stmt->error];
    }
}

?>

Message.php
<?php

class Messages {
    private $conn;
    private $table_name = "messages";
    public function __construct($db) {
        $this->conn = $db;
    }
    public function getMessagesBySender($senderEmail) {
        $query = "SELECT id, sender, receiver, content, created_at, status FROM " . $this->table_name . "
WHERE sender = ?";
        $stmt = $this->conn->prepare($query);
        $stmt->bind_param("s", $senderEmail);
        $stmt->execute();
        $result = $stmt->get_result();
        $messages = $result->fetch_all(MYSQLI_ASSOC);
        return $messages;
    }
    public function getMessagesByReceiver($receiverEmail) {
        $query = "SELECT id, sender, receiver, content, created_at, status FROM " . $this->table_name . "
WHERE receiver = ?";
        $stmt = $this->conn->prepare($query);
        $stmt->bind_param("s", $receiverEmail);
        $stmt->execute();
        $result = $stmt->get_result();
        $messages = $result->fetch_all(MYSQLI_ASSOC);
        return $messages;
    }
    public function sendMessage($senderEmail, $receiverEmail, $content) {
        $query = "INSERT INTO " . $this->table_name . " (sender, receiver, content, created_at, status)
VALUES (?, ?, ?, NOW(), 'delivered')";
        $stmt = $this->conn->prepare($query);
        $stmt->bind_param("sss", $senderEmail, $receiverEmail, $content);
        if ($stmt->execute()) {
            return ["success" => true];
        } else {
            return ["success" => false, "error" => $stmt->error];
        }
    }
    public function updateMessageStatus($messageId, $status) {
        $query = "UPDATE " . $this->table_name . " SET status = ? WHERE id = ?";
        $stmt = $this->conn->prepare($query);

```

```

$stmt->bind_param("si", $status, $messageId);
if ($stmt->execute()) {
    return ["success" => true];
} else {
    return ["success" => false, "error" => $stmt->error];
}
?>

```

Payment/pay.php

```

<?php
require_once '../libs/razorpay/Razorpay.php';
require_once '../controllers/PaymentController.php';
use Razorpay\Api\Api;
$ride_id = $_GET['ride_id'] ?? null;
$ride_fare = $_GET['ride_fare'] ?? null; // Fetching the ride fare from the query string
if (!$ride_id || !$ride_fare) {
    die('Ride ID and fare are required');
}
$promoCode = $_GET['promo_code'] ?? null;
$finalAmount = $ride_fare;
$ride_fare_paise = $ride_fare * 100;
$promoMessage = "";
if ($promoCode) {
    $paymentController = new PaymentController();
    $promoResult = $paymentController->applyPromoCode($promoCode, $ride_fare);
    if ($promoResult['success']) {
        $finalAmount = $promoResult['finalAmount'];
        $ride_fare_paise = $finalAmount * 100;
        $promoMessage = "Promo code applied! Discount: ₹" . number_format($promoResult['discount'], 2);
    } else {
        $promoMessage = "Error: " . ($promoResult['error'] ?? 'Invalid promo code');
    }
}
$api = new Api("rzp_test_Tcoz7vjPf8FW4k", "cXXXKPU05Pzh6js0wtFQJUKSb");
$order = $api->order->create([
    'amount' => $ride_fare_paise, // Amount in paise
    'currency' => 'INR',
    'payment_capture' => 1
]);
$order_id = $order->id;
?>
<!DOCTYPE html>
<html class="h-100">
<head>

```

```

<title>Razorpay Payment</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
<script src="https://checkout.razorpay.com/v1/checkout.js"></script>
</head>
<body class="h-100 d-flex align-items-center justify-content-center bg-light py-4">
    <h2 class="card-title text-center mb-4">Complete Your Payment</h2>
    <button class="btn btn-primary w-100 mb-2" type="button">Apply
        </button>
    <div class="mb-4">
        <div class="d-flex justify-content-between mb-2">
            <span>Original Amount:</span>
            <span>₹<?= number_format($ride_fare, 2) ?></span>
        </div>
        <?php if ($promoCode && isset($promoResult['success']) && $promoResult['success']):?
        ?>
        <div class="d-flex justify-content-between mb-2 text-success">
            <span>Discount:</span>
            <span>-₹<?= number_format($promoResult['discount'], 2) ?></span>
        </div>
        <hr class="my-2">
        <span>Final Amount:</span>
        <span>₹<?= number_format($finalAmount, 2) ?></span>
        </div>
        <?php endif; ?>
    </div>
    <!-- Payment Button -->
    <div class="text-center">
        <button id="pay-button" class="btn btn-success btn-lg px-5">
            Pay Now
        </button>
    </div>
<script>
    function applyPromoCode() {
        const promoCode = document.getElementById('promo-code').value;
        if (!promoCode) {
            showMessage('Please enter a promo code', 'text-danger');
            return;
        }
        window.location.href = window.location.pathname +
            '?ride_id=<?= $ride_id ?>&ride_fare=<?= $ride_fare ?>&promo_code=' +
        encodeURIComponent(promoCode);
    }
    function showMessage(message, className = 'text-success') {
        const messageElement = document.getElementById('promo-message');
        messageElement.className = `mt-2 small ${className}`;
        messageElement.textContent = message;
    }
</script>

```

```

    }
    var options = {
        "key": "rzp_test_Tcoz7vjPf8FW4k",
        "amount": "<?= $ride_fare_paise ?>",
        "currency": "INR",
        "name": "Your Company Name",
        "description": "Payment for your order",
        "order_id": "<?= $order_id ?>",
        "theme": { "color": "#198754" },
        "handler": function(response) {
            showMessage('Payment successful! Redirecting...');

            window.location.href = 'success.php?payment_id=' + response.razorpay_payment_id +
                '&ride_id=' + <?= $ride_id ?> +
                '&ride_fare=' + <?= $finalAmount ?>;
        },
        "modal": {
            "ondismiss": function() {
                showMessage('Payment cancelled or failed. Please try again.', 'text-danger');
            }
        }
    };
    var rzp = new Razorpay(options);
    document.addEventListener('DOMContentLoaded', function() {
        document.getElementById('apply-promo-btn').addEventListener('click', applyPromoCode);
        document.getElementById('pay-button').addEventListener('click', function(e) {
            rzp.open();
            e.preventDefault();
        });
        document.getElementById('pay-button').innerText = 'Pay ₹' + <?= $finalAmount ?> + ' Now';
        <?php if (!empty($promoMessage)): ?>
        showMessage(<?= json_encode($promoMessage) ?>, 
            <?= isset($promoResult['success']) && $promoResult['success'] ? '"text-success"' : '"text-danger"' ?>);
        <?php endif; ?>
    });
    </script>
</body>
</html>

```

Success.php

```

<?php
require_once '../libs/razorpay/Razorpay.php';
use Razorpay\Api\Api;
require_once __DIR__ . '/../includes/database.php';
require_once '../controllers/RideController.php';
require_once '../controllers/PaymentController.php';

$api = new Api("rzp_test_Tcoz7vjPf8FW4k", "cXXXKPU05Pzh6js0wtFQJUKSb");
if (!isset($_GET['payment_id'])) {

```

```

echo "Payment failed or canceled.";
exit;
}

$payment_id = $_GET['payment_id'];
$ride_id = $_GET['ride_id'] ?? null;
$ride_fare = $_GET['ride_fare'] ?? null;
$rideController = new RideController();
$paymentController = new PaymentController();

try {
    // Fetch the payment details
    $payment = $api->payment->fetch($payment_id);

    if ($payment['status'] == 'captured') {
        // Example: Fetching ride details for a specific ride ID
        $rideDetails = $rideController->getRideDetails($ride_id);

        if ($rideDetails['success']) {
            $ride = $rideDetails['ride'];
            $paymentData = [
                'ride_id' => $ride['id'],
                'driver_id' => $ride['driver_id'],
                'customer_id' => $ride['customer_id'],
                'payment_id' => $payment['id'],
                'amount' => $ride['fare'],
                'payment_method' => 'Razorpay',
                'payment_status' => $payment['status'],
                'payment_date' => date('Y-m-d H:i:s')
            ];
        }

        // Save payment details using the controller
        $saveResult = $paymentController->savePayment($paymentData);
        if ($saveResult['success']) {
            // Display success message
            echo "<h1>Payment Successful!</h1>";
            echo "<p>Payment ID: " . $payment['id'] . "</p>";
            echo "<p>Amount: " . ($payment['amount'] / 100) . " INR</p>";
            echo "<p>Status: " . $payment['status'] . "</p>";

            // Call to mark the ride as paid
            $markPaidResult = $rideController->markRideAsPaid($ride_id);
            if ($markPaidResult['success']) {
                echo "<p>Ride marked as paid successfully.</p>";
            } else {

```

```

        echo "<p>Error marking ride as paid: " . $markPaidResult['error'] . "</p>";
    }
} else {
    echo "Error saving payment: " . $saveResult['error'];
}
} else {
    echo "Error fetching ride details: " . $rideDetails['error'];
}
} else {
    echo "<h1>Payment Failed or Pending!</h1>";
    echo "<p>Status: " . $payment['status'] . "</p>";
}
} catch (Exception $e) {
    echo "Error: " . $e->getMessage();
}

?>
<head>
    <style>
        h1 { color: green; }
        p { font-size: 16px; }
    </style>
</head>
<script>
    setTimeout(function() {
        window.location.href = '/ETaxi/views/customer/my-rides.php';
    }, 5000); // Redirect after 5 seconds
</script>

```

CONTROLLERS:

AuthController.php

```

<?php
session_start();
header('Content-Type: application/json');
include_once __DIR__ . '/../includes/database.php';
include_once __DIR__ . '/../models/Customer.php';
include_once __DIR__ . '/../models/Driver.php';
include_once __DIR__ . '/../models/Admin.php';
$database = new Database();
$db = $database->getConnection();
$response = ["success" => false, "error" => "Invalid email or password"];
// Validate required fields

```

```

if (!isset($_POST['action']) && !isset($_GET['action'])) {
    echo json_encode(["success" => false, "error" => "Missing required fields"]);
    exit;
}
$action = isset($_POST['action']) ? $_POST['action'] : $_GET['action'];
if ($action === 'register') {
    $data = [
        'name' => trim($_POST['name']),
        'email' => trim($_POST['email']),
        'phone' => trim($_POST['phone']),
        'otp' => trim($_POST['otp']),
        'password' => password_hash(trim($_POST['password']), PASSWORD_BCRYPT),
        'created_at' => date('Y-m-d H:i:s')
    ];
    if (!isset($_SESSION['otp']) || $_SESSION['otp'] != $data['otp'] || $_SESSION['otp_phone'] != $data['phone']) {
        echo json_encode(["success" => false, "error" => "Invalid or unverified OTP"]);
        exit;
    }
    $customer = new Customer($db);
    $response = $customer->register($data);
    echo json_encode($response); exit;
}
if ($action === 'register_driver') {
    $data = [
        'name' => trim($_POST['name']),
        'email' => trim($_POST['email']),
        'password' => trim($_POST['password']),
        'phone' => trim($_POST['phone']),
        'id_number' => trim($_POST['id_number']),
        'vehicle_type' => trim($_POST['vehicle_type']),
        'vehicle_model' => trim($_POST['vehicle_model']),
        'vehicle_number' => trim($_POST['vehicle_number']),
        'license_number' => trim($_POST['license_number']),
        'profile_image' => $_FILES['profile_image'],
        'id_image' => $_FILES['id_image'],
        'vehicle_image' => $_FILES['vehicle_image'],
        'license_image' => $_FILES['license_image']
    ];
    $driver = new Driver($db);
    $response = $driver->register($data);
    echo json_encode($response);
    exit;
}
$email = trim($_POST['email']);
$password = trim($_POST['password']);
$role = trim($_POST['role']);
}
if ($action === 'login_driver_otp') {
    $phone = trim($_POST['phone']);
    $otp = trim($_POST['otp']);
}

```

```

$driver = new Driver($db);
$response = $driver->loginWithOtp($phone, $otp);
if ($response['success']) {
    $_SESSION['driver_id'] = $response['id'];
    $_SESSION['driver'] = $response['driver'];
}
echo json_encode($response);
exit;
}

if ($action === 'login_customer_otp') {
    $email = trim($_POST['email']);
    $otp = trim($_POST['otp']);
    $customer = new Customer($db);
    $response = $customer->loginWithOtp($email, $otp);
    if ($response['success']) {
        $_SESSION['customer_id'] = $response['id'];
        $_SESSION['customer'] = $response['customer'];
    }
}
echo json_encode($response);
exit;
?>

```

adminController.php

```

<?php
require_once __DIR__ . '/../includes/database.php';
require_once __DIR__ . '/../models/Admin.php';
$database = new Database();
$db = $database->getConnection();
$admin = new Admin($db);
session_start();
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    if (isset($_POST['action']) && $_POST['action'] === 'updateProfile') {
        $adminId = $_SESSION['admin']['id']; // Fetch admin ID from session
        $name = $_POST['name'];
        $email = $_POST['email'];
        $password = !empty($_POST['password']) ? password_hash($_POST['password'],
PASSWORD_BCRYPT) : null;
        $phone = $_POST['phone'];
        $profileImage = $_FILES['profile_image']['name'];
        // Handle profile image upload
        if (!empty($profileImage)) {
            $targetDir = __DIR__ . "/../assets/images/";
            $targetFile = $targetDir . basename($profileImage);
            move_uploaded_file($_FILES['profile_image']['tmp_name'], $targetFile);
        }
    }
}

```

```

} else {
    $adminDetails = $admin->getAdminById($adminId);
    $profileImage = $adminDetails['profile_image'];
}
$updateData = [
    'name' => $name,
    'email' => $email,
    'phone' => $phone,
    'profile_image' => $profileImage
];
if ($password) {
    $updateData['password'] = $password;
}
$updateResult = $admin->updateAdmin($adminId, $updateData);
$data = json_decode(file_get_contents("php://input"), true);
if (isset($data['action'])) {
    switch ($data['action']) {
        case 'update':
            $driverId = $data['driverId'];
            $driverData = $data['driverData'];
            $vehicleData = $data['vehicleData'];
            $updateResult = $driver->updateDriver($driverId, $driverData);
            if ($updateResult['success']) {
                $vehicleResult = $driver->updateVehicle($driverId, $vehicleData);
                if ($vehicleResult['success']) {
                    echo json_encode(['success' => true]);
                } else {
                    echo json_encode(['error' => $vehicleResult['error']]);
                }
            } else {
                echo json_encode(['error' => $updateResult['error']]);
            }
            exit();
            break;
        case 'delete':
            $driverId = $data['driverId'];
            $deleteResult = $driver->deleteDriver($driverId);
            echo json_encode($deleteResult);
            exit();
            break;
        case 'deleteCustomer':
            $customerId = $data['customerId'];
            $deleteResult = $customer->deleteCustomer($customerId);
            echo json_encode($deleteResult);
            exit();
    }
}

```

```
        break;
    case 'approve':
        $driverId = $data['driverId'];
        $approveResult = $driver->approveDriver($driverId);
        echo json_encode($approveResult);
        exit();
        break;
    case 'reject':
        $driverId = $data['driverId'];
        $rejectResult = $driver->rejectDriver($driverId);
        echo json_encode($rejectResult);
        exit();
        break;
    case 'viewDetails':
        $driverId = $data['driverId'];
        $driverDetails = $driver->getDriverById($driverId);
        if ($driverDetails['success']) {
            $vehicleDetails = $driver->getVehicleById($driverId);
            if ($vehicleDetails['success']) {
                echo json_encode([
                    'driver' => $driverDetails['driver'],
                    'vehicle' => $vehicleDetails['vehicle']
                ]);
            } else {
                echo json_encode(["success" => false, "error" => $vehicleDetails['error']]);
            }
        } else {
            echo json_encode(["success" => false, "error" => $driverDetails['error']]);
        }
        exit();
        break;
    case 'deleteBlog':
        $blogId = $data['blogId'];
        $deleteResult = $blog->deleteBlog($blogId);
        echo json_encode(["success" => $deleteResult]);
        exit();
        break;
    case 'viewBlog':
        $blogId = $data['blogId'];
        $blogDetails = $blog->getBlogById($blogId);
        echo json_encode($blogDetails);
        exit();
        break;
    case 'getAllRides':
        $ridesResult = $ride->getAllRides();
```

```

        echo json_encode($ridesResult);
        exit();
        break;
    case 'viewDetails':
        $messageId = $data['messageId'];
        $updateResult = $messages->updateMessageStatus($messageId, 'seen');
        echo json_encode($updateResult);
        exit();
        break;
    case 'addNotification':
        $message = $data['message'];
        $created_by = $_SESSION['admin']['id']; // Assuming admin ID is stored in the session
        $addResult = $notification->addNotification($created_by, $message);
        echo json_encode($addResult);
        exit();
        break;
    default:
        echo json_encode(["success" => false, "error" => "Invalid action"]);
        exit();
        break;
    }
}
}
}

$driversResult = $driver->getAllDrivers();
if ($driversResult['success']) {
    $drivers = $driversResult['drivers'];
} else {
    $error = $driversResult['error'];
}
$blogsResult = $blog->getAllBlogs();
if ($blogsResult['success']) {
    $blogs = $blogsResult['blogs'];
} else {
    $error = $blogsResult['error'];
}
$customersResult = $customer->getAllCustomers();
if ($customersResult['success']) {
    $customers = $customersResult['customers'];
} else {
    $error = $customersResult['error'];
}
$RidesResult = $ride->getAllRides();
if ($RidesResult['success']) {
    $rides = $RidesResult['rides'];
} else {

```

```

        $error = $RidesResult['error'];
    }
    $paymentsResult = $payment->getAllPayments();
    if ($paymentsResult['success']) {
        $payments = $paymentsResult['payments'];
    } else {
        $error = $paymentsResult['error'];
    }
    $promoCodesResult = $promoCodes->getAllPromoCodes();
    if ($promoCodesResult['success']) {
        $promoCodes = $promoCodesResult['promoCodes'];
    } else {
        $error = $promoCodesResult['error'];
    }
    $notificationsResult = $notification->getAllNotifications();
    if ($notificationsResult['success']) {
        $notifications = $notificationsResult['notifications'];
    } else {
        $error = $notificationsResult['error'];
    }
    $adminData = getAdminData();
    function getAdminEmail() {
        global $admin;
        $adminData = $_SESSION['admin'];
        $adminDetails = $admin->getAdminById($adminData['id']);
        return $adminDetails['email'];
    }
    function getMessages() {
        global $messages;
        $adminEmail = getAdminEmail();
        $sentMessages = $messages->getMessagesBySender($adminEmail);
        $receivedMessages = $messages->getMessagesByReceiver($adminEmail);
        return ['sentMessages' => $sentMessages, 'receivedMessages' => $receivedMessages];
    }
    $messagesData = getMessages();
    $sentMessages = $messagesData['sentMessages'];
    $receivedMessages = $messagesData['receivedMessages'];
?>
```

DriverController.php

```
<?php
session_start(); // Add this at the top of the file
if (!isset($_SESSION['driver'])) {
    header('Location: ../../login.php');
    exit;
```

```

}

include_once __DIR__ . '/../includes/database.php';
include_once __DIR__ . '/../models/Ride.php';
include_once __DIR__ . '/../models/Payment.php';
include_once __DIR__ . '/../models/Messages.php';
include_once __DIR__ . '/../models/Notification.php';
include_once __DIR__ . '/../models/Driver.php';
class DriverController {
    private $db;
    private $ride;
    private $payment;
    private $messages;
    private $notification;
    private $driver;
    public function __construct() {
        $database = new Database();
        $this->db = $database->getConnection();
        $this->ride = new Ride($this->db);
        $this->payment = new Payment($this->db);
        $this->messages = new Messages($this->db);
        $this->notification = new Notification($this->db);
        $this->driver = new Driver($this->db);
    }
    public function getDashboardData($driver_id) {
        $totalRides = $this->ride->getTotalRidesByDriver($driver_id);
        $totalEarnings = $this->payment->getTotalEarningsByDriver($driver_id);
        $pendingRides = $this->ride->getAllPendingRides();
        return [
            "totalRides" => $totalRides['totalRides'],
            "totalEarnings" => $totalEarnings['totalEarnings'],
            "pendingRides" => $pendingRides['pendingRides']
        ];
    }
    public function acceptRide($ride_id, $driver_id) {
        $availability = $this->driver->getDriverAvailabilityStatus($driver_id);
        if (!$availability) {
            return [
                "success" => false,
                "error" => "Driver is currently unavailable for rides"
            ];
        }
        return [
            "sentMessages" => $sentMessages,
            "receivedMessages" => $receivedMessages
        ];
    }
}

```

```

    }

    public function sendMessage($data) {
        $senderEmail = $data['senderEmail'];
        $receiverEmail = $data['receiverEmail'];
        $content = $data['content'];
        return $this->messages->sendMessage($senderEmail, $receiverEmail, $content);
    }

    public function updateMessageStatus($messageId) {
        $result = $this->messages->updateMessageStatus($messageId, 'read');
        if ($result['success']) {
            return [
                'success' => true,
            ];
        }
    }

    public function getDriverNotifications() {
        return $this->notification->getAllNotifications();
    }

    public function displayEditProfile() {
        if (!isset($_SESSION['driver'])) {
            header('Location: ../../login.php');
            exit;
        }
        $driver_id = $_SESSION['driver']['id'];
        $result = $this->driver->getDriverProfile($driver_id);

        if ($result['success']) {
            $profile = $result['profile'];
            include '../../views/_driver/edit_profile_content.php';
        } else {
            $_SESSION['error'] = "Failed to load profile data";
            header('Location: dashboard.php');
            exit;
        }
    }

    public function handleProfileUpdate() {
        if (!isset($_SESSION['driver'])) {
            error_log("Driver not authenticated");
            echo json_encode(["success" => false, "error" => "Not authenticated"]);
            exit;
        }
        $driver_id = $_SESSION['driver']['id'];
        error_log("Updating profile for driver ID: $driver_id");
        try {
            $result = $this->driver->updateProfile($driver_id, $_POST, $_FILES);
            header('Content-Type: application/json');
            echo json_encode($result);
        } catch (Exception $e) {
            error_log("Error in handleProfileUpdate: " . $e->getMessage());
        }
    }
}

```

```
        echo json_encode(["success" => false, "error" => "An unexpected error occurred"]);
    }
    exit;
}
public function getDriverProfile($driver_id) {
    return $this->driver->getDriverProfile($driver_id);
}
public function getDriverEarnings($driver_id) {
    return $this->payment->getDriverEarnings($driver_id);
}
public function getEditProfileData() {
    if (!isset($_SESSION['driver'])) {
        header('Location: ../../login.php');
        exit;
    }
    $driver_id = $_SESSION['driver']['id'];
    $result = $this->driver->getDriverProfile($driver_id);
    if (!$result['success']) {
        $_SESSION['error'] = "Failed to load profile data";
        header('Location: dashboard.php');
        exit;
    }
    return $result['profile'];
}
public function getHomePageData($driver_id) {
    $dashboardData = $this->getDashboardData($driver_id);
    $availability = $this->getDriverAvailability($driver_id);
    return [
        'dashboardData' => $dashboardData,
        'availability' => $availability
    ];
}
public function handleAcceptRide($rideId, $driverId) {
    $rideModel = new Ride($this->db);
    $result = $rideModel->acceptRide($rideId, $driverId);
    if ($result['success']) {
        header('Location: ../../views/ride/my-rides.php');
        exit;
    } else {
        $_SESSION['error'] = $result['error'];
        header('Location: home.php');
        exit;
    }
}
```

```

$controller = new DriverController();
$driver_id = $_SESSION['driver']['id'];
$result = $controller->getDriverEarnings($driver_id);
$profile = $controller->getEditProfileData();
if (!$result['success']) {
    $_SESSION['error'] = "Failed to load earnings data";
    header('Location: dashboard.php');
    exit;
}
$earnings = $result['earnings'];
?>

```

CustomerController.php

```

<?php
require_once __DIR__ . '/../models/Customer.php';
require_once __DIR__ . '/../models/Messages.php';
require_once __DIR__ . '/../models/Notification.php';
require_once __DIR__ . '/../models/Ride.php';
require_once __DIR__ . '/../models/Review.php';
require_once __DIR__ . '/../models/Payment.php';
require_once __DIR__ . '/../includes/Database.php';
session_start();
$database = new Database();
$db = $database->getConnection();
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['action']) && $_POST['action'] ===
'updateProfile') {
    $customerId = $_SESSION['customer']['id'];
    $name = $_POST['name'];
    $email = $_POST['email'];
    $phone = $_POST['phone'];
    $profileImage = $_SESSION['customer']['profile_image'];
    if (isset($_FILES['profile_image']) && $_FILES['profile_image']['error'] === UPLOAD_ERR_OK) {
        $uploadDir = __DIR__ . '/../uploads/users/profile/';
        $fileName = basename($_FILES['profile_image']['name']);
        $targetFilePath = $uploadDir . $fileName;
        if (move_uploaded_file($_FILES['profile_image']['tmp_name'], $targetFilePath)) {
            $profileImage = $fileName;
        }
    }
    $customerModel = new Customer($db);
    $result = $customerModel->updateProfile($customerId, $name, $email, $phone, $profileImage);
    if ($result['success']) {
        $_SESSION['customer'] = array_merge($_SESSION['customer'], [
            'name' => $name,
            'email' => $email,
        ]);
    }
}

```

```

        'phone' => $phone,
        'profile_image' => $profileImage
    ]);
    $_SESSION['success_message'] = "Profile updated successfully.";
} else {
    $_SESSION['error_message'] = $result['error'];
}
header("Location: ../views/customer/edit-profile.php");
exit;
}

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $json = file_get_contents('php://input');
    $data = json_decode($json, true);
    if ($data === null) {
        header('Content-Type: application/json');
        echo json_encode(['success' => false, 'message' => 'Invalid JSON data']);
        exit;
    }
    $action = $data['action'] ?? "";

    if ($action === 'sendMessage') {
        $senderEmail = $data['senderEmail'];
        $receiverEmail = $data['receiverEmail'];
        $content = $data['content'];
        $result = $messages->sendMessage($senderEmail, $receiverEmail, $content);
        header('Content-Type: application/json');
        echo json_encode($result);
        exit;
    } elseif ($action === 'viewDetails') {
        $messageId = $data['messageId'];
        $result = $messages->updateMessageStatus($messageId, 'seen');
        header('Content-Type: application/json');
        echo json_encode($result);
        exit;
    } elseif ($action === 'cancelRide') {
        $rideId = $data['rideId'];
        $result = $ride->cancelRide($rideId);
        header('Content-Type: application/json');
        echo json_encode($result);
        exit;
    } elseif ($action === 'submitReview') {
        $reviewData = [
            'ride_id' => $data['rideId'],
            'customer_id' => $_SESSION['customer']['id'],
            'driver_id' => $data['driverId'],
        ];
    }
}

```

```

        'rating' => $data['rating'],
        'review' => $data['review'],
        'created_at' => date('Y-m-d H:i:s')
    ];
    $result = $review->submitReview($reviewData);
    header('Content-Type: application/json');
    echo json_encode($result);
    exit;
} else {
    header('Content-Type: application/json');
    echo json_encode(['success' => false, 'message' => 'Invalid action']);
    exit;
}
} elseif ($_SERVER['REQUEST_METHOD'] === 'GET') {
    if (!isset($_SESSION['customer'])) {
        header('Location: ../../login.php');
        exit;
    }
    $rides = $ride->getRidesByCustomer($customerData['id']);
    $sentMessages = $messages->getMessagesBySender($customerData['email']);
    $receivedMessages = $messages->getMessagesByReceiver($customerData['email']);
    $customerPayments = $payment->getAllPaymentsByCustomer($customerData['id']);
}
?>
```

RideController.php

```

<?php
require_once __DIR__ . '/../models/Ride.php';
require_once __DIR__ . '/../includes/database.php';
class RideController {
    private $rideModel;
    public function __construct() {
        $database = new Database();
        $db = $database->getConnection();
        $this->rideModel = new Ride($db);
    }
    public function bookRide() {
        if ($_SERVER['REQUEST_METHOD'] === 'POST') {
            $customer_id = $_POST['customer_id'];
            $pickup_location = $_POST['pickup_location'];
            $dropoff_location = $_POST['dropoff_location'];
            $distance = $_POST['distance'];
            $per_km_rate = 10;
            $per_minute_rate = 2;
            if ($distance <= 5) {
                $surcharges = 5;
            }
        }
    }
}
```

```

        } elseif ($distance <= 10) {
            $surcharges = 3;
        } else {
            $surcharges = 2;
        }
        $total_fare = ($distance * $per_km_rate) + ($time * $per_minute_rate) + $surcharges;
        $result = $this->rideModel->bookRide($customer_id, $pickup_location, $dropoff_location,
        $total_fare, $distance, $per_km_rate);
        if ($result['success']) {
            header("Location: ../views/customer/book-ride.php?success=1&ride_id=" . $result['ride_id']
            . "&fare=" . $total_fare);
        } else {
            public function getDriverRides($driver_id, $status) {
                return $this->rideModel->getDriverRides($driver_id, $status);
            }
            public function completeRide($ride_id, $driver_id, $otp) {
                return $this->rideModel->completeRide($ride_id, $driver_id, $otp);
            }
            public function cancelRide($ride_id) {
                return $this->rideModel->cancelRide($ride_id);
            }
            public function markRideAsPaid($ride_id) {
                return $this->rideModel->markRideAsPaid($ride_id);
            }
            public function getRideDetails($ridelId) {
                return $this->rideModel->getRideDetails($ridelId);
            }
        }
    }
    if ($_SERVER['REQUEST_METHOD'] === 'POST') {
        $controller = new RideController();

        if (isset($_POST['action'])) {
            header('Content-Type: application/json');
            switch ($_POST['action']) {
                case 'completeRide':
                    echo json_encode($controller->completeRide(
                        $_POST['ride_id'],
                        $_POST['driver_id'],
                        $_POST['otp']
                    ));
                    break;
                case 'cancelRide':
                    echo json_encode($controller->cancelRide($_POST['ride_id']));
                    break;
                case 'markRideAsPaid':

```

```

        echo json_encode($controller->markRideAsPaid($_POST['ride_id']));
        break;
    default:
        echo json_encode(["success" => false, "error" => "Invalid action"]);
    }
    exit;
}
$controller->bookRide();
}
?>

```

Blogcontroller.php

```

<?php
require_once __DIR__ . '/../models/Blog.php';
require_once __DIR__ . '/../includes/database.php';

$database = new Database();
$db = $database->getConnection();
$blog = new Blog($db);

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // Ensure content type is set
    if (isset($_SERVER["CONTENT_TYPE"]) && strpos($_SERVER["CONTENT_TYPE"], "application/json")
!== false) {
        $_POST = json_decode(file_get_contents("php://input"), true);
    }

    $action = $_POST['action'] ?? "";

    if ($action === 'createBlog') {
        $title = trim($_POST['title'] ?? "");
        $content = trim($_POST['content'] ?? "");
        $author = trim($_POST['author'] ?? "");

        // Call the createBlog method
        $result = $blog->createBlog($title, $content, $author);

        // Return JSON response
        header('Content-Type: application/json');
        echo json_encode($result);
        exit;
    }
} elseif ($_SERVER['REQUEST_METHOD'] === 'GET') {
    $action = $_GET['action'] ?? 'random'; // Default to random for homepage
}

```

```

if ($action === 'all') {
    // Fetch all blogs without pagination
    $blogs = $blog->getAllBlogs();
    header('Content-Type: application/json');
    echo json_encode($blogs);
    exit;
} else {
    $blogs = $blog->getRandomBlogs();
}
}

?>

```

Paymentcontroller.php

```

<?php
require_once __DIR__ . '/../models/Payment.php';
require_once __DIR__ . '/../models/PromoCodes.php';
require_once __DIR__ . '/../includes/database.php';

class PaymentController {
    private $paymentModel;
    private $promoCodeModel;

    public function __construct() {
        $database = new Database();
        $db = $database->getConnection();
        $this->paymentModel = new Payment($db);
        $this->promoCodeModel = new PromoCodes($db);
    }

    public function savePayment($paymentData) {
        if (!$paymentData) {
            return ['success' => false, 'error' => 'No payment data provided'];
        }

        $result = $this->paymentModel->savePayment($paymentData);
        if ($result) {
            return ['success' => true, 'message' => 'Payment saved successfully'];
        } else {
            return ['success' => false, 'error' => 'Failed to save payment'];
        }
    }

    public function applyPromoCode($code, $amount) {
        try {

```

```

        if (empty($code) || empty($amount)) {
            return ['success' => false, 'error' => 'Invalid promo code or amount'];
        }

        $result = $this->promoCodeModel->validatePromoCode($code);

        if ($result['success']) {
            $discountPercentage = $result['promoCode']['discount_percentage'];
            $discount = ($amount * $discountPercentage) / 100;
            $finalAmount = $amount - $discount;

            return [
                'success' => true,
                'originalAmount' => $amount,
                'discount' => $discount,
                'finalAmount' => max(0, $finalAmount), // Ensure amount doesn't go below 0
                'promoCode' => $result['promoCode']
            ];
        }
    }

    return $result;
} catch (Exception $e) {
    error_log("Error in applyPromoCode: " . $e->getMessage());
    return ['success' => false, 'error' => 'Error applying promo code'];
}
}

// Add other payment-related methods as needed
}
?>

```

Api

Mail_otp.php

```

<?php
require '../libs/smtp/smtp/PHPMailerAutoload.php';
require '../includes/database.php';
session_start();
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $userExists = false;
    if ($role === 'customer') {
        $stmt = $db->prepare("SELECT email FROM customers WHERE email = ?");
    } elseif ($role === 'driver') {

```

```

$stmt = $db->prepare("SELECT email FROM drivers WHERE email = ?");
} elseif ($role === 'admin') {
    $stmt = $db->prepare("SELECT email FROM admins WHERE email = ?");
}
if ($stmt) {
    $stmt->bind_param('s', $email);
    $stmt->execute();
    $stmt->store_result();
    if ($stmt->num_rows > 0) {
        $userExists = true;
    }
}
if ($userExists) {
    $mail = new PHPMailer(true);
    try {
        $mail->isSMTP();
        $mail->SMTPAuth = true;
        $mail->SMTPSecure = 'tls';
        $mail->Host = "smtp.gmail.com";
        $mail->Port = 587;
        $mail->isHTML(true);
        $mail->CharSet = 'UTF-8';
        $mail->Username = "sumeetpanigrahy494@gmail.com";
        $mail->Password = "wlsl gnsq sbbw tldz";
        $mail->SetFrom("sumeetpanigrahy494@gmail.com", "Etaxi Service");
        $mail->addAddress($email);
        $mail->Subject = 'Password Reset OTP';
        $mail->Body = "Your OTP code is $otp";
        $mail->send();
        echo json_encode(['success' => true, 'message' => 'OTP sent to your email.']);
    } catch (Exception $e) {
        echo json_encode(['success' => false, 'message' => "Message could not be sent. Mailer Error: {$mail->ErrorInfo}"]);
    }
} else {
    echo json_encode(['success' => false, 'message' => 'Email not found.']);
}
} else {
    echo json_encode(['success' => false, 'message' => 'Invalid request']);
}
?>

```

Mobile_otp.php

```
<?php
session_start();
```

```

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $phone = $_POST['phone'];
    $otp = rand(100000, 999999);
    $_SESSION['otp'] = $otp;
    $_SESSION['otp_phone'] = $phone;
    $params = array(
        'token' => 'gri3ic1y633mzmfu',
        'to' => $phone,
        'body' => "Your OTP code is $otp"
    );
    $curl = curl_init();
    curl_setopt_array($curl, array(
        CURLOPT_URL => "https://api.ultramsg.com/instance109742/messages/chat",
        CURLOPT_HTTPHEADER => array(
            "content-type: application/x-www-form-urlencoded"
        ),
    ));
    $response = curl_exec($curl);
    $err = curl_error($curl);
    curl_close($curl);
    if ($err) {
        echo json_encode(['success' => false, 'message' => "cURL Error #: " . $err]);
    } else {
        echo json_encode(['success' => true, 'response' => $response]);
    }
} else {
    echo json_encode(['success' => false, 'message' => 'Invalid request']);
}
?>

```

Verify_otp.php

```

<?php
session_start();
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $enteredOtp = $_POST['otp'];
    $storedOtp = isset($_SESSION['otp']) ? $_SESSION['otp'] : null;
    $storedPhone = isset($_SESSION['otp_phone']) ? $_SESSION['otp_phone'] : null;
    $enteredPhone = isset($_POST['phone']) ? $_POST['phone'] : null;
    if ($enteredOtp == $storedOtp && $enteredPhone == $storedPhone) {
        echo json_encode(['success' => true]);
    } else {
        echo json_encode(['success' => false, 'message' => 'Invalid OTP or phone number']);
    }
} else {

```

```
    echo json_encode(['success' => false, 'message' => 'Invalid request']);
}
```

Sendmessage.php

```
<?php
session_start();
header('Content-Type: application/json');

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $phone = $_POST['phone'];

    // Use custom message if provided, otherwise generate OTP
    if (isset($_POST['message'])) {
        $message = $_POST['message'];
    } else {
        $otp = rand(100000, 999999);
        $_SESSION['otp'] = $otp;
        $_SESSION['otp_phone'] = $phone;
        $message = "Your OTP code is $otp";
    }

    $params = array(
        'token' => 'gri3ic1y633mzmfu',
        'to' => $phone,
        'body' => $message
    );

    $curl = curl_init();
    curl_setopt_array($curl, array(
        CURLOPT_URL => "https://api.ultramsg.com/instance109742/messages/chat",
        CURLOPT_RETURNTRANSFER => true,
        CURLOPT_ENCODING => "",
        CURLOPT_MAXREDIRS => 10,
        CURLOPT_TIMEOUT => 30,
        CURLOPT_SSL_VERIFYHOST => 0,
        CURLOPT_SSL_VERIFYPEER => 0,
        CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
        CURLOPT_CUSTOMREQUEST => "POST",
        CURLOPT_POSTFIELDS => http_build_query($params),
        CURLOPT_HTTPHEADER => array(
            "content-type: application/x-www-form-urlencoded"
        ),
    ));

    $response = curl_exec($curl);
}
```

```
$err = curl_error($curl);

curl_close($curl);

if ($err) {
    echo json_encode(['success' => false, 'message' => "cURL Error #: " . $err]);
} else {
    echo json_encode(['success' => true, 'response' => $response]);
}
} else {
    echo json_encode(['success' => false, 'message' => 'Invalid request']);
}
exit;
?>
```

Demo Snapshot

index.php

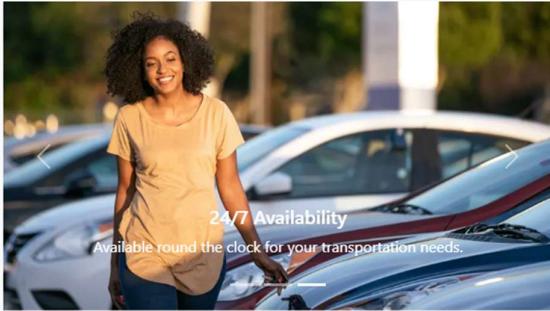
Screenshot of the index.php page showing the homepage of ETaxi.

The top navigation bar includes links for Home, Earn, Stats, About, Help, Gallery, Blog, and another Blog. A dropdown menu for 'Admin' is open, showing Admin Dashboard and Logout options.

Your Trusted Taxi Service Partner

Welcome to ETaxi, where reliability meets comfort. We provide professional taxi services with experienced drivers, modern vehicles, and competitive rates. Whether you need airport transfers, city tours, or corporate travel solutions, we've got you covered 24/7.

[About Us](#) [Our Services](#)



24/7 Availability
Available round the clock for your transportation needs.

Why Choose Our Taxi Service



24/7 Availability
Round-the-clock service ensuring you can get a ride whenever you need it, day or night.



Safe & Secure
Professional licensed drivers, GPS tracking, and secure payment options for your peace of mind.

user_register.php

User Registration

Name:

Email:

Phone:

OTP: Send OTP

Password: Verify OTP

Register

driver_register.php

<p>Driver Registration</p> <p><input type="text"/> Name</p> <p><input type="text"/> Email</p> <p><input type="password"/> Password</p> <p><input type="text"/> Phone</p> <p><input type="text"/> Aadhaar Number</p> <p><input type="file"/> Choose file No file chosen Aadhaar Proof</p> <p><input type="file"/> Choose file No file chosen Profile Image</p>	<p>Vehicle Details</p> <p><input type="text"/> Vehicle Type</p> <p><input type="text"/> Vehicle Model</p> <p><input type="text"/> Vehicle Number</p> <p><input type="file"/> Choose file No file chosen Vehicle Image</p> <p><input type="text"/> License Number</p> <p><input type="file"/> Choose file No file chosen License Proof</p> <p><input type="text"/> OTP</p>
<input type="button" value="Send OTP"/> <input type="button" value="Verify OTP & Submit"/>	

login.php

Sign in

Email

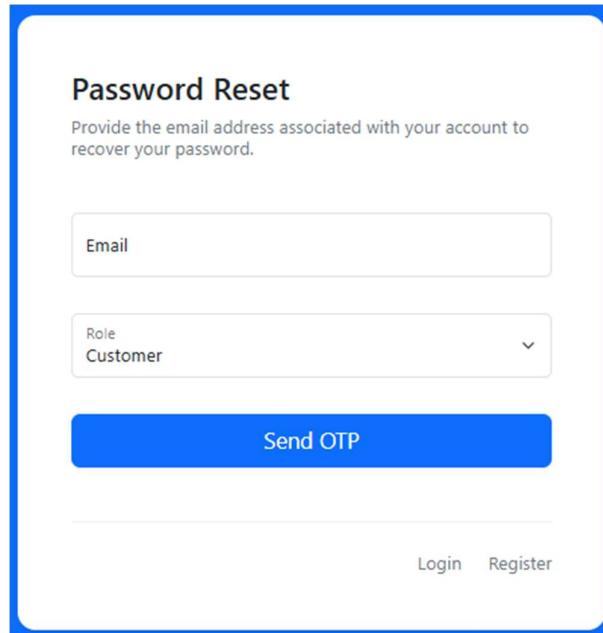
OTP

[Forgot password?](#)

Remember password

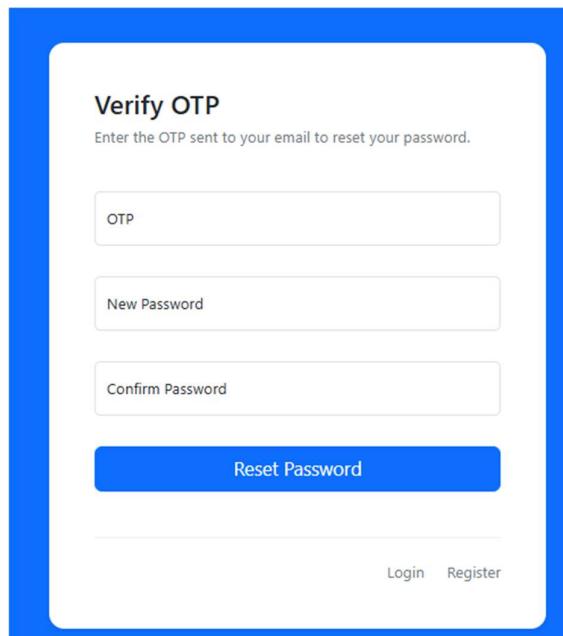
Don't have an account? [Create Account](#)

fogot_password.php



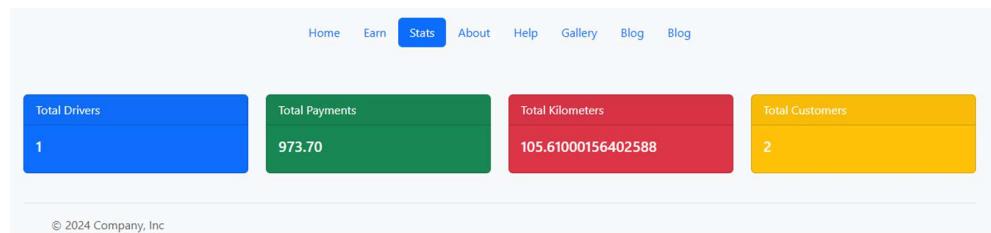
A screenshot of a 'Password Reset' form. The title 'Password Reset' is at the top. Below it is a sub-instruction: 'Provide the email address associated with your account to recover your password.' There are two input fields: 'Email' and 'Role'. The 'Role' field has a dropdown menu showing 'Customer'. A large blue button labeled 'Send OTP' is centered below the inputs. At the bottom right of the form area are 'Login' and 'Register' links.

verify_reset_otp.php



A screenshot of a 'Verify OTP' form. The title 'Verify OTP' is at the top. Below it is a sub-instruction: 'Enter the OTP sent to your email to reset your password.' There are three input fields: 'OTP', 'New Password', and 'Confirm Password'. A large blue button labeled 'Reset Password' is centered below the inputs. At the bottom right of the form area are 'Login' and 'Register' links.

stats.php



| blog.php

Blogs

Search blogs...



Blog Title 1

This is the content of blog number 1....

By custom_author_1
2025-03-03 20:29:17



Blog Title 5

This is the content of blog number 5....

By custom_author_5
2025-03-03 20:29:17



Blog Title 6

This is the content of blog number 6....

By custom_author_6
2025-03-03 20:29:17



Blog Title 7

This is the content of blog number 7....

By custom_author_7
2025-03-03 20:29:17



Blog Title 8

This is the content of blog number 8....

By custom_author_8
2025-03-03 20:29:17



Blog Title 9

This is the content of blog number 9....

By custom_author_9
2025-03-03 20:29:17

Previous 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 Next

admin

home.php

The screenshot shows the Admin Dashboard interface. On the left, a sidebar lists navigation options: Home, Manage Drivers, Manage Customers, Ride Management, Payments & Earnings, Promo Codes, Reports & Analytics, and Manage Blogs. The main area is titled "Admin Dashboard". It features three summary boxes: "Total Rides" (3), "Total Earnings" (\$973.70), and "Active Drivers" (1). Below these are sections for "Recent Rides" (listing three completed rides) and "Earnings Overview" (a line chart showing earnings increasing from approximately \$16,000 to \$18,000 over time). A top navigation bar includes links for "Edit Profile" and "Logout".

Ride ID	Driver	Customer	Pickup	Drop-off	Status	Earnings
#3	1	1	Gangadahani, Odisha, India	Rangunipalli - New Makarajhola Road, Rangan Palli, Odisha, India	Completed	\$169.90
#2	1	1	Berhampur, Odisha, India	Bhanjanagar, Odisha, India	Completed	\$803.80
#1		1	Bakilkona, Odisha, India	Dhabalpur Maa Kali Temple, Dhalapur, Odisha, India	Cancelled	\$89.40

Earnings Overview

The chart displays a single data series representing earnings over time. The Y-axis is labeled "Earnings \$" and has tick marks at 16,000 and 18,000. The X-axis represents time, with a single data point plotted at an intermediate point between two major grid lines.

edit-profile.php

Edit Profile

Name:

Profile Image: No file chosen

Email:

Password:

Phone:

Created At:

Edit

7.11 MODEL QUESTIONS

- Describe the relationship between average cost and marginal cost with the help of a diagram.
- Distinguish between fixed cost and variable cost.
- Explain why there is no fixed cost in the long run.
- What is the main reason behind the U shape of LAC curve?

~ the Break even Analysis and its applications

manage-customers.php

Manage Customers

Search by name...

Name	Profile Image	Actions
Sumit Panigrahy		<button>View Details</button> <button>Delete</button>
Sumeet Panigrahy		<button>View Details</button> <button>Delete</button>

manage-drivers.php

Manage Drivers

Search Approved Drivers by ID or Name

Search Pending Drivers by ID or Name

Approved Drivers

SI No	Profile	Name	Actions
1		Sumeet Panigrahy	<button>View Details</button>

Pending Drivers

SI No	Profile	Name	Actions
-------	---------	------	---------

manage-blogs.php

Manage Blogs

[Create Blog](#)

Blog Title 1

by custom_author_1

[View](#) [Delete](#)

Blog Title 5

by custom_author_5

[View](#) [Delete](#)

Blog Title 6

by custom_author_6

[View](#) [Delete](#)

Blog Title 7

by custom_author_7

[View](#) [Delete](#)

Blog Title 8

by custom_author_8

[View](#) [Delete](#)

Blog Title 9

by custom_author_9

[View](#) [Delete](#)

Blog Title 10

by custom_author_10

[View](#) [Delete](#)

Blog Title 11

by custom_author_11

[View](#) [Delete](#)

Blog Title 12

by custom_author_12

[View](#) [Delete](#)

Blog Title 13

by custom_author_13

[View](#) [Delete](#)

Blog Title 14

by custom_author_14

[View](#) [Delete](#)

Blog Title 15

by custom_author_15

[View](#) [Delete](#)

Blog Title 16

by custom_author_16

[View](#) [Delete](#)

Blog Title 17

by custom_author_17

[View](#) [Delete](#)

Blog Title 18

by custom_author_18

[View](#) [Delete](#)

manage-payments.php

Manage Payments

ID	Ride ID	Customer Name	Driver Name	Amount	Payment Method	Payment Status	Payment Date
1	2	Sumit Panigrahy	Sumeet Panigrahy	803.80	Razorpay	captured	2025-04-10 11:32:50
2	3	Sumit Panigrahy	Sumeet Panigrahy	169.90	Razorpay	captured	2025-04-10 11:36:52

manage-rides.php

Manage Rides

Filter by Status: All Search:

ID	Customer ID	Pickup Location	Dropoff Location	Status	Fare	Driver ID	Request Time	Distance	Rate	Actions
3	1	Gangadahani, Odisha, India	Rangunipalli - New Makarajhola Road, Rangan Palli, Odisha, India	completed	169.90	1	2025-04-10 15:05:18	16.79	10	<input type="button" value="View Details"/>
2	1	Berhampur, Odisha, India	Bhanjanagar, Odisha, India	completed	803.80	1	2025-04-10 14:27:10	80.18	10	<input type="button" value="View Details"/>
1	1	Bakilikona, Odisha, India	Dhabalpur Maa Kali Temple, Dhalapur, Odisha, India	cancelled	89.40		2025-04-10 08:10:02	8.64	10	<input type="button" value="View Details"/>

Ride Details

Ride Details

ID: 3

Pickup Location: Gangadahani, Odisha, India

Dropoff Location: Rangunipalli - New Makarajhola Road, Rangan Palli, Odisha, India

Status: completed

Fare: 169.90

Request Time: 2025-04-10 15:05:18

Distance: 16.79

Rate: 10

Customer Details

Name: Sumit Panigrahy

Email: sumeetpanigrahy494@gmail.com

Phone: 7894473241

Driver Details

Name: Sumeet Panigrahy

Email: sumeetpanigrahy494@gmail.com

Phone: 7894473241

Reviews

messages.php

Messages

Search for messages..

Sent Messages

ID	Receiver	Content	Created At	Status	Action
1	sumeetpanigrahy494@gmail.com	asdf	2025-04-09 20:14:53	seen	View Details

Received Messages

ID	Sender	Content	Created At	Status	Action
2	sumeetpanigrahy494@gmail.com	asdfghjk	2025-04-10 07:53:53	seen	View Details

[Send Message](#)

notifications.php

Notifications

Search for notifications..

[Add Notification](#)

ID	Message	Created At	Action
2	surajq	2025-04-09 21:06:57	Edit Delete

report.php

admin1et Dashboard

Monthly Report
Generated on April 17, 2025

[Print Report](#)

Year: 2025 Month: April [Generate Report](#)

Total Rides: 3 Total Distance: 105.61 km

Total Earnings: ₹973.70 Total Payments: 2

Total Customers: 1 Active Users

Average Fare: ₹324.57 Per Ride

Monthly Rides Trend

Monthly Earnings Trend

Month	Total Rides	Total Distance (km)	Total Earnings	Total Customers
January	0	0.00	₹0.00	0
February	0	0.00	₹0.00	0
March	0	0.00	₹0.00	0

customer home.php

Sumit Panigrahy Dashboard

Total Rides
3

Total Payments
\$973.70

Recent Rides

Ride ID	Driver	Pickup	Drop-off	Status	Fare	Date
#3	Sumeet Panigrahy	Gangadahani, Odisha, India	Rangunipalli - New Makarajhola Road, Rangan Palli, Odisha, India	Completed	\$169.90	Apr 10, 2025 15:05
#2	Sumeet Panigrahy	Berhampur, Odisha, India	Bhanjanagar, Odisha, India	Completed	\$803.80	Apr 10, 2025 14:27
#1	Not Assigned	Bakilkona, Odisha, India	Dhabalpur Maa Kali Temple, Dhalapur, Odisha, India	Cancelled	\$89.40	Apr 10, 2025 08:10

Your Current Location

A screenshot of a Google Maps interface showing a map of a city area. A red marker indicates the user's current location. Key landmarks visible include SHYAMSUNDARPUR NATURAL RESORT, The Amsterdam Cafe, Mad Mugs By Ricky, Bangalore Express, Jyoti Fire And Safety Academy, PATHARAGADIA, Nilachal Polytechnic, ION.Digital Zone (IOZ) 1, and PRASANTI VIHAR. The map also shows several roads and green spaces.

edit-profile.php

Edit Profile



Name

Sumit Panigrahy

Email

sumeetpanigrahy494@gmail.com

Phone

7894473241

Update Profile

book-ride.php

Sumit Panigrahy Dashboard

Book a Ride

Customer ID:
1

Pickup Location:
Berhampur, Odisha, India

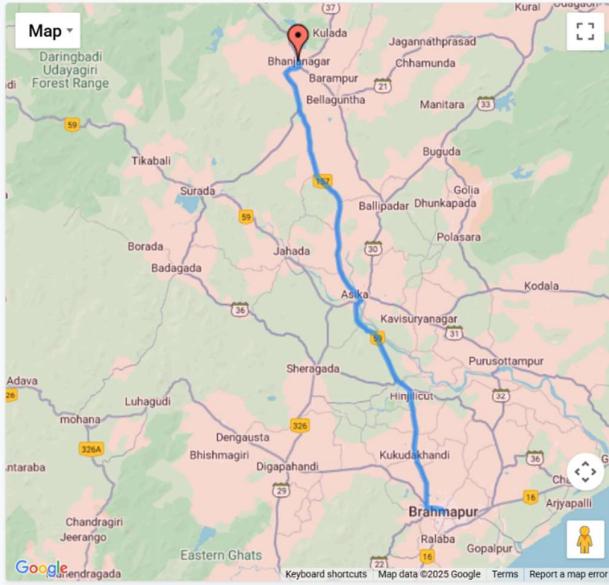
Dropoff Location:
Bhanjanagar, Odisha, India

Distance: 80.2 km
Estimated Time: 1 hour 55 mins

Distance (km):
80.18

Fare (₹):
801.82

Book Ride



my-rides.php

My Rides

ID	Driver	Pickup Location	Dropoff Location	Status	Action
1		Bakilikona, Odisha, India	Dhabalpur Maa Kali Temple, Dhalapur, Odisha, India	cancelled	
2	Sumeet Panigrahy	Berhampur, Odisha, India	Bhanjanagar, Odisha, India	completed	
3	Sumeet Panigrahy	Gangadahani, Odisha, India	Rangunipalli - New Makarajhola Road, Rangan Palli, Odisha, India	completed	

payments.php

My Payments

Payment ID	Ride ID	Amount	Payment Method	Payment Status	Payment Date
pay_QHJ2E00E6Cy9St	2	803.80	Razorpay	captured	2025-04-10 11:32:50
pay_QHJ6Scs3VBhATy	3	169.90	Razorpay	captured	2025-04-10 11:36:52

driver home.php

The screenshot shows the Admin Dashboard with the following statistics:

- Total Rides: 3
- Total Earnings: \$973.70
- Active Drivers: 1
- Registered Customers: 2

Recent Rides table:

Ride ID	Driver	Customer	Pickup	Drop-off	Status	Earnings
#3	1	1	Gangadahani, Odisha, India	Rangunipalli - New Makarajhola Road, Rangan Palli, Odisha, India	Completed	\$169.90
#2	1	1	Berhampur, Odisha, India	Ranjanagar, Odisha, India	Completed	\$803.80
#1	1	1	Balkikona, Odisha, India	Dhabalpur Mata Kali Temple, Dhalapur, Odisha, India	Canceled	\$89.40

Earnings Overview chart (line graph):

The graph shows a single data series labeled "Earnings" with three data points.

edit-profile.php

Driver & Vehicle Details

Driver Information

Name	Sumeet Panigrahy
Email	sumeetpanigrahy494@gmail.com
Phone	7894473241
ID Number	577263452967
Status	approved

Vehicle Information

License Number	546286729725
Vehicle Type	Bike
Vehicle Model	Jupiter
Vehicle Number	142536

Profile Image and **ID Image** sections are present, each containing a placeholder image and a question box.

Close | **Update** | **Delete**

earnings.php

My Earnings

Payment ID	Ride ID	Amount	Payment Method	Status	Date
pay_QHJ6Scs3VBhATy	3	₹169.90	Razorpay	captured	10 Apr 2025, 11:36 AM
pay_QHJ2E00E6Cy9St	2	₹803.80	Razorpay	captured	10 Apr 2025, 11:32 AM

my-rides.php

Paid Rides

Show 10 entries

Search:

ID Customer Phone Pickup Dropoff Fare Distance Request Time Action

No data available in table

Showing 0 to 0 of 0 entries

[Previous](#) [Next](#)

Accepted Rides

ID Customer Phone Pickup Dropoff Fare Distance Request Time

Completed Rides

Show 10 entries

Search:

ID Customer Phone Pickup Dropoff Fare Distance Completed Time

3	Sumit Panigrahy	7894473241	Gangadahani, Odisha, India	Rangunipalli - New Makarajhola Road, Rangan Palli, Odisha, India	₹169.90	16.79 km	2025-04-10 15:08:04
---	-----------------	------------	----------------------------	--	---------	----------	---------------------

2	Sumit Panigrahy	7894473241	Berhampur, Odisha, India	Bhanjanagar, Odisha, India	₹803.80	80.18 km	2025-04-10 15:03:27
---	-----------------	------------	--------------------------	----------------------------	---------	----------	---------------------

Showing 1 to 2 of 2 entries

[Previous](#) [1](#) [Next](#)

messages.php

Messages

Search for messages..

Sent Messages

ID	To Customer	Content	Created At	Status	Action
2	admin@example.com	asdfghjk	2025-04-10 07:53:53	seen	<button>View Details</button>

Received Messages

ID	From Customer	Content	Created At	Status	Action
1	admin@example.com	asdf	2025-04-09 20:14:53	seen	<button>View Details</button>

Send Message

Notification.php

Notifications

Search for notifications..

Add Notification

ID	Message	Created At	Action
2	surajq	2025-04-09 21:06:57	<button>Edit</button> <button>Delete</button>

Testing

Testing is the process of evaluating a system or its components to verify that it meets the specified requirements and works as expected. It involves identifying defects, ensuring quality, and validating that the software performs its intended functions.

Types of Testing

Manual Testing

Testing performed manually by a tester without using automation tools.

- **Exploratory Testing:** Testing without predefined test cases.
- **Ad-hoc Testing:** Informal testing without planning or documentation.

Functional Testing

Verifies that the software functions as expected based on requirements.

For the **E-Taxi project**, the following types of testing can be applied to verify that the software functions as expected based on its requirements:

1. Unit Testing

- **Definition:** Testing individual components or modules in isolation.
- **Example in E-Taxi:**
 - Testing the bookRide() method in RideController.php to ensure it calculates the fare correctly and saves the ride details in the database.
 - Testing the acceptRide() method in DriverController.php to verify that it updates the ride status and driver availability.

2. Integration Testing

- **Definition:** Testing combined modules to ensure they work together as expected.
- **Example in E-Taxi:**
 - Verifying the interaction between the RideController and PaymentController to ensure that ride booking triggers the correct payment process.
 - Testing the integration of the Google Maps API in book-ride.php to ensure pickup and drop-off locations are correctly processed.

3. System Testing

- **Definition:** Testing the complete system as a whole to ensure it meets the specified requirements.
- **Example in E-Taxi:**

- Testing the entire ride booking flow, from customer login to ride completion, including notifications, payment, and driver assignment.
- Verifying that the admin dashboard in manage-drivers.php displays accurate data for drivers, rides, and payments.

4. User Acceptance Testing (UAT)

- **Definition:** Testing by end-users to validate the software against business requirements.
- **Example in E-Taxi:**
 - Allowing customers to test the ride booking and payment process to ensure it is user-friendly and meets their expectations.
 - Drivers testing the ride assignment and earnings tracking features in home.php to confirm usability and functionality.

Black Box Testing

Black box testing is a software testing method where the tester evaluates the functionality of an application without having any knowledge of its internal code, structure, or implementation. The focus is on inputs and expected outputs, ensuring the software behaves as intended.

Example Test Cases for Black Box Testing

1. Login Functionality

- **Test Case Name:** Valid Login
 - **Input:** Correct username and password.
 - **Expected Output:** User is redirected to the dashboard.
- **Test Case Name:** Invalid Login
 - **Input:** Incorrect username or password.
 - **Expected Output:** Error message "Invalid credentials" is displayed.

2. Ride Booking

- **Test Case Name:** Successful Ride Booking
 - **Input:** Valid pickup and drop-off locations, payment method selected.
 - **Expected Output:** Ride is booked, and confirmation notification is sent.
- **Test Case Name:** Invalid Location
 - **Input:** Empty or invalid pickup/drop-off location.

- **Expected Output:** Error message "Invalid location" is displayed.

3. Payment Processing

- **Test Case Name:** Successful Payment
 - **Input:** Valid card details or UPI ID.
 - **Expected Output:** Payment is processed, and a success message is displayed.
- **Test Case Name:** Failed Payment
 - **Input:** Expired card or insufficient funds.
 - **Expected Output:** Error message "Payment failed" is displayed.

4. Registration

- **Test Case Name:** Valid User Registration
 - **Input:** All required fields filled with valid data.
 - **Expected Output:** User account is created, and a confirmation email is sent.
- **Test Case Name:** Missing Required Fields
 - **Input:** Leave one or more required fields empty.
 - **Expected Output:** Error message "Please fill all required fields" is displayed.

5. Search Functionality

- **Test Case Name:** Valid Search Query
 - **Input:** Enter a valid keyword in the search bar.
 - **Expected Output:** Relevant search results are displayed.
- **Test Case Name:** Invalid Search Query
 - **Input:** Enter a random string or special characters.
 - **Expected Output:** Message "No results found" is displayed.

White Box Testing

White box testing is a software testing method where the tester has knowledge of the internal code, structure, and logic of the application. The focus is on testing the internal workings of the system, such as code paths, loops, conditions, and data flow.

Example Test Cases for White Box Testing

1. Login Functionality

- **Test Case Name:** Validate Password Hashing
 - **Code Tested:** Password hashing function in AuthController.php.

- **Test Input:** Plain text password "password123".
- **Expected Output:** The hashed password is stored in the database and matches when verified.
- **Test Case Name:** Check Login Condition
 - **Code Tested:** if condition for verifying username and password in AuthController.php.
 - **Test Input:** Correct username and incorrect password.
 - **Expected Output:** The condition fails, and the user is not logged in.

2. Ride Booking

- **Test Case Name:** Validate Fare Calculation Logic
 - **Code Tested:** calculateFare() method in RideController.php.
 - **Test Input:** Distance = 10 km, Base Fare = \$5, Per km rate = \$2.
 - **Expected Output:** Total fare = \$25.
- **Test Case Name:** Test Ride Assignment Loop
 - **Code Tested:** Loop that assigns the nearest driver in RideController.php.
 - **Test Input:** List of drivers with locations, user location.
 - **Expected Output:** The nearest driver is assigned.

3. Payment Processing

- **Test Case Name:** Validate Payment Gateway API Call
 - **Code Tested:** API integration logic in PaymentController.php.
 - **Test Input:** Valid payment details (card number, expiry, CVV).
 - **Expected Output:** API returns a success response, and the payment is marked as completed.
- **Test Case Name:** Test Refund Logic
 - **Code Tested:** Refund function in PaymentController.php.
 - **Test Input:** Ride ID with a failed payment.
 - **Expected Output:** Refund is initiated, and the status is updated in the database.

4. Registration

- **Test Case Name:** Validate Email Verification Logic
 - **Code Tested:** Email OTP generation and verification in AuthController.php.

- **Test Input:** User email = "test@example.com".
- **Expected Output:** OTP is generated, stored in the database, and verified correctly.
- **Test Case Name:** Test Input Validation
 - **Code Tested:** Validation logic for required fields in AuthController.php.
 - **Test Input:** Missing email field.
 - **Expected Output:** Validation fails, and an error message is returned.

5. Search Functionality

- **Test Case Name:** Validate Search Query Execution
 - **Code Tested:** SQL query execution in SearchController.php.
 - **Test Input:** Search keyword = "Taxi".
 - **Expected Output:** SQL query returns matching results from the database.
- **Test Case Name:** Test Pagination Logic
 - **Code Tested:** Pagination function in SearchController.php.
 - **Test Input:** Total results = 50, Results per page = 10.
 - **Expected Output:** Pagination correctly divides results into 5 pages.

Let us now gone though some test cases to test all functionality like email validation login register otp sending

Test Case ID	Module	Test Case Description	Input Data / Steps	Expected Output	Actual Output	Status
TC01	User Registration	Register a new user	Name, Email: Phone: 9876543210, Password: ETaxi@123	User saved, OTP sent to phone	OTP sent and redirected	<input checked="" type="checkbox"/> Pass
TC02	User Registration	Duplicate email during registration	Email: again	Error message: Email already exists	Error shown correctly	<input checked="" type="checkbox"/> Pass
TC03	OTP Verification	Validate OTP	OTP entered: Correct	User verified and redirected to login	Redirect successful	<input checked="" type="checkbox"/> Pass

TC04	OTP Verification	Invalid OTP	OTP entered: Incorrect	Error message: Invalid OTP	Error shown	<input checked="" type="checkbox"/> Pass
TC05	Login	Valid login	Email Password: ETaxi@123	User logged in and redirected to dashboard	Redirect successful	<input checked="" type="checkbox"/> Pass
TC06	Login	Invalid login	Wrong Password	Error shown: Invalid credentials	Error shown	<input checked="" type="checkbox"/> Pass
TC07	Ride Booking	Book a ride successfully	Select pickup and drop-off locations → Confirm	Ride booked, Booking ID shown	Booking confirmed	<input checked="" type="checkbox"/> Pass
TC08	Ride Booking	Book a ride with invalid location	Invalid pickup location	Error: Invalid location	Error shown	<input checked="" type="checkbox"/> Pass
TC09	Ride Booking	Cancel a ride	Select an active ride → Click cancel	Ride canceled successfully	Ride canceled	<input checked="" type="checkbox"/> Pass
TC10	Payment	Complete payment via Razorpay	Choose payment method → Pay via Razorpay	Success screen + payment details shown	Payment successful	<input checked="" type="checkbox"/> Pass
TC	Payment	Failed payment	Insufficient funds in card	Error message: Payment failed	Error shown	<input checked="" type="checkbox"/> Pass
TC12	Ride History	View ride history	Logged-in user → Open "My Rides" tab	List of past rides shown	Rides displayed correctly	<input checked="" type="checkbox"/> Pass
TC13	Notifications	Send ride confirmation notification	Ride ID: R123 → Send notification	Notification delivered to user	Notification received	<input checked="" type="checkbox"/> Pass
TC14	Notifications	Send payment failure notification	Payment failed for Ride ID: R123	Notification delivered to user	Notification received	<input checked="" type="checkbox"/> Pass
TC15	Profile Management	Edit user profile and save changes	Change phone number and profile image	Profile updated	Changes saved	<input checked="" type="checkbox"/> Pass

TC16	Driver Module	View assigned rides	Logged-in driver → Open "My Rides" tab	List of assigned rides shown	Rides displayed correctly	<input checked="" type="checkbox"/> Pass
TC17	Driver Module	Update ride status	Driver marks ride as "Completed"	Ride status updated to "Completed"	Status updated	<input checked="" type="checkbox"/> Pass
TC18	Admin Dashboard	Approve a pending driver	Click on Approve for driver ID D102	Driver status = "Approved"	Updated successfully	<input checked="" type="checkbox"/> Pass
TC19	Admin Dashboard	Block a user	Click on Block for user ID U123	User status = "Blocked"	Updated successfully	<input checked="" type="checkbox"/> Pass
TC20	Admin Dashboard	View revenue report	Open "Reports" tab	Shows revenue generated from rides and subscriptions	Correct data shown	<input checked="" type="checkbox"/> Pass
TC21	Messaging Module	Send a message to the driver	Logged-in user → Type message → Send	Message delivered to driver	Message received	<input checked="" type="checkbox"/> Pass
TC22	Messaging Module	Receive a message from the driver	Driver sends a message to the user	Message delivered to user	Message received	<input checked="" type="checkbox"/> Pass
TC23	Subscription Module	Subscribe to a plan	Choose plan → Pay via Razorpay	Success screen + subscription details shown	Subscription successful	<input checked="" type="checkbox"/> Pass
TC24	Subscription Module	Cancel a subscription	Click on Cancel for active subscription	Subscription canceled successfully	Subscription canceled	<input checked="" type="checkbox"/> Pass
TC25	Search Functionality	Search for a ride	Enter ride ID or date in the search bar	Matching rides displayed	Rides displayed correctly	<input checked="" type="checkbox"/> Pass

Requirements Specifications

1. Platform Requirements

1.1 Web Platform	1.2 Mobile Platform (Future Implementation)
- Frontend Framework: HTML5, CSS3, JavaScript	- iOS: iOS 13.0 or higher
- Backend Framework: PHP 7.4 or higher	- Android: Android 8.0 (API level 26) or higher
- Database: MySQL 5.7 or higher	- Cross-platform Framework: React Native (recommended)
- Web Server: Apache 2.4 or higher	
- Responsive Design: Support for all modern browsers and devices	

2. Hardware Requirements

2.1 Server Requirements	2.2 Client Requirements
- CPU: Minimum 4 cores, 2.5 GHz or higher	- Desktop/Laptop:
- RAM: Minimum 8GB, 16GB recommended	- Minimum 4GB RAM
- Storage:	- Stable internet connection (minimum 5Mbps)
- SSD: Minimum 256GB	- GPS capability
- HDD: Minimum 1TB for backup	- Camera (for document verification)
- Network:	- Minimum 2GB RAM
- Minimum 100Mbps bandwidth	- Stable internet connection (4G/5G recommended)
- Static IP address	
- Redundant internet connection (recommended)	

3. Software Requirements

3.1 Development Tools	3.2 Server Software
- IDE: Visual Studio Code, PHPStorm, or similar	- Operating System:
- Version Control: Git	- Linux (Ubuntu 20.04 LTS or higher)
- Database Management: phpMyAdmin	- Windows Server 2019 or higher
- API Testing: Postman	- Web Server: Apache 2.4+
- Code Quality Tools:	- Database Server: MySQL 5.7+
- PHP_CodeSniffer	- PHP Extensions:
- ESLint	- cUrl
- Prettier	- MySQLi

4. Development Environment Setup

4.1 Local Development	4.2 Third-Party Integrations
- XAMPP/WAMP/MAMP	- Payment Gateway: RazorPay
- Composer (PHP package manager)	- Maps API: Google Maps API
- Git for version control	- SMS Gateway: Ultramsg
	- Email Service: SMTP

Future Scope and Enhancements

1. Mobile Application Development

Native Mobile Applications

- **iOS and Android Development:**
 - Develop native applications using React Native for cross-platform compatibility
 - Implement platform-specific UI/UX guidelines for optimal user experience
 - Utilize native device capabilities for enhanced performance
 - Ensure compatibility with latest OS versions and security requirements

Real-time Location Tracking

- **Enhanced GPS Integration:**
 - Implement high-precision location tracking with accuracy up to 5 meters
 - Utilize geofencing for precise pickup and drop-off locations
 - Implement background location tracking for continuous updates
 - Add route optimization based on real-time traffic data
 - Integrate with multiple map providers for redundancy

Push Notifications

- **Firebase Cloud Messaging Implementation:**
 - Real-time ride status updates (driver assigned, arrived, started, completed)
 - Payment confirmations and receipts
 - Promotional offers and discounts
 - Emergency alerts and safety notifications
 - Custom notification preferences for users

Offline Mode

- **Limited Functionality Without Internet:**
 - Cache essential user data and ride history
 - Allow viewing past rides and receipts
 - Enable basic profile management
 - Store favorite locations and routes
 - Queue actions for synchronization when online

Biometric Authentication

- **Advanced Security Features:**

- Fingerprint recognition for quick login
- Face ID integration for iOS devices
- Iris scanning for Android devices
- Voice recognition as an alternative authentication method
- Multi-factor authentication options

2. Advanced Features

AI-Powered Features

- **Smart Fare Prediction:**
 - Machine learning algorithms for dynamic pricing
 - Historical data analysis for demand forecasting
 - Weather condition integration for price adjustments
 - Special event detection and pricing optimization
 - Seasonal pricing patterns
- **Driver-Passenger Matching:**
 - AI-based matching algorithm considering multiple factors
 - Driver rating and performance history
 - Passenger preferences and past interactions
 - Vehicle type and comfort requirements
 - Route optimization for driver efficiency
- **Fraud Detection System:**
 - Behavioral analysis for suspicious activities
 - Payment pattern recognition
 - Location anomaly detection
 - Account sharing detection
 - Automated fraud reporting and blocking
- **Dynamic Pricing Engine:**
 - Real-time demand-supply analysis
 - Traffic condition integration
 - Weather impact consideration
 - Special event detection
 - Competitive pricing analysis

3. Business Expansion

Multiple Vehicle Types

- **Expanded Fleet Options:**
 - **Economy:** Standard sedans and compact cars
 - **Premium:** Luxury sedans and SUVs
 - **Luxury:** High-end vehicles with professional chauffeurs
 - **Electric Vehicles:** Eco-friendly transportation options
 - **Specialized Vehicles:** Wheelchair-accessible vehicles, vans for groups

Corporate Accounts

- **Business Solutions:**
 - Dedicated business dashboard
 - Employee ride management
 - Expense tracking and reporting
 - Custom billing and invoicing
 - Priority booking and dedicated fleet

Subscription Plans

- **Flexible Payment Options:**
 - Monthly subscription with fixed ride credits
 - Family plans with multiple user accounts
 - Premium plans with priority booking
 - Corporate subscription packages
 - Student and senior citizen discounts

Loyalty Program

- **Rewards System:**
 - Points earned per ride based on fare
 - Tiered membership levels (Bronze, Silver, Gold, Platinum)
 - Redeemable rewards (free rides, upgrades, partner discounts)
 - Special member-only promotions
 - Birthday and anniversary bonuses

Partner Integration

- **Ecosystem Expansion:**
 - Hotel booking platforms integration
 - Restaurant reservation systems
 - Event ticketing platforms
 - Travel agency partnerships
 - Corporate travel management systems

Conclusion

The **E-Taxi Project** is a comprehensive and user-friendly platform designed to streamline ride-hailing services for customers, drivers, and administrators. By integrating robust modules such as user authentication, ride booking, payment processing, and real-time notifications, the system ensures a seamless and secure experience for all stakeholders.

The project adheres to modern development standards, leveraging technologies like PHP, MySQL, and third-party APIs (e.g., Google Maps, Razorpay) to deliver high performance and scalability. With features like role-based access control, OTP verification, and responsive design, the platform is both secure and accessible across devices.

The **E-Taxi Project** is a comprehensive and innovative solution designed to address the growing demand for efficient and reliable ride-hailing services. Through its modular architecture, the platform integrates key functionalities such as user registration, ride booking, payment processing, driver management, and administrative oversight. Each module has been carefully developed to ensure seamless interaction, scalability, and user satisfaction.

The project emphasizes security, performance, and usability by incorporating advanced technologies like secure authentication mechanisms, encrypted data storage, and responsive design. Third-party integrations, such as Google Maps for location services and Razorpay for payment processing, enhance the platform's functionality and ensure a smooth user experience.

Extensive testing, including unit, integration, system, and user acceptance testing, has validated the platform's reliability and adherence to requirements. The system is designed to handle real-world scenarios efficiently, ensuring that users, drivers, and administrators can interact with the platform without disruptions.

In conclusion, the E-Taxi Project is a robust, secure, and user-friendly platform that meets the functional and non-functional requirements of a modern ride-hailing system. It serves as a reliable solution for connecting users and drivers while providing administrators with the tools needed to manage operations effectively. This project not only fulfills current requirements but also lays the groundwork for future advancements in the transportation industry.

Through rigorous testing, including unit, integration, system, and user acceptance testing, the application has been validated to meet functional and non-functional requirements. The inclusion of monitoring tools and a maintenance plan ensures long-term reliability and adaptability to future enhancements.

In conclusion, the E-Taxi Project is a well-structured and scalable solution that addresses the needs of the ride-hailing industry, providing a solid foundation for future growth and innovation.

Bibliography / References

The development of E-Taxi: A Ride-Hailing and Management System involved referencing multiple online resources, tools, and official documentation to ensure proper coding practices, secure payment integration, API usage, and deployment. Below are the key references:

1 Websites and Documentation

Backend Development (PHP & MySQL)

PHP Manual – <https://www.php.net/manual/en/>

W3Schools – PHP MySQL Tutorial – <https://www.w3schools.com/php/>

Frontend Development (HTML/CSS/JavaScript)

MDN Web Docs (Mozilla) – <https://developer.mozilla.org/>

W3Schools – HTML, CSS, JS Tutorials – <https://www.w3schools.com/>

Payment Gateway Integration

Razorpay Documentation – <https://razorpay.com/docs/>

WhatsApp API

UltraMsg WhatsApp API Documentation – <https://docs.ultramsg.com/>

SMTP Integration

PHPMailer Documentation – <https://github.com/PHPMailer/PHPMailer>

UI Framework

Bootstrap Official Documentation – <https://getbootstrap.com/docs/>

2 YouTube Channels and Tutorials

Code With Harry – PHP, MySQL, Login Systems

Razorpay – For setting up Razorpay payment gateway

The Net Ninja – JavaScript and frontend UI tips

WsCube Tech – Bootstrap implementation and responsive design

3 Books

"PHP and MySQL For Dummies" by Janet Valade

"Fundamentals of Web Development" by Randy Connolly and Ricardo Hoar

"Clean Code: A Handbook of Agile Software Craftsmanship" by Robert C. Martin

4 Academic & College Material

Class notes and resources shared by faculty for final year project development.

University project structure guidelines provided by the department.

This comprehensive list of references highlights the resources used to ensure the successful development and deployment of the E-Taxi project.