1. INTRODUCTION

1.1 Project Summary

The Car Pooling System is a web application designed to facilitate ride-sharing between drivers and passengers. It allows riders to search and book rides, while drivers can offer rides and manage bookings. The system also provides an administrative panel for managing users, vehicles, rides, and feedback.

1.2 Purpose

The purpose of this project is to reduce travel costs, minimize environmental pollution, and make commuting more efficient by enabling carpooling. It provides a platform for secure and user-friendly ride sharing.

1.3 Scope

The scope of this project includes modules for Guest, Rider, Driver, and Admin. Guests can browse rides, Riders can book and manage rides, Drivers can offer rides, and Admins can oversee the entire system.

1.4 Technology & Literature Review

The system is developed using modern web technologies, a relational database (MySQL), and follows the Software Development Life Cycle (SDLC). Existing transportation systems often lack proper ride-sharing features, which this project addresses.

2. SYSTEM REQUIREMENT STUDY

2.1 User Characteristics

There are four types of users:

- Guest – Can browse rides and contact admin.
- Rider – Registers, books rides, manages bookings, provides feedback.
- Driver – Registers, adds vehicles, offers rides, manages rides.
- Admin – Manages users, rides, bookings, vehicles, and generates reports.

2.2 Hardware & Software Requirements

Server: Core i5 processor, 8GB RAM, 20GB free disk, MySQL & PHP/Apache. Client: Any modern browser, 4GB RAM, dual-core processor. Software: MySQL, PHP, HTML, CSS, JavaScript.

2.3 Constraints

- Only registered Riders can book rides.
- Limited seats depend on driver vehicle capacity.
- Internet connection is mandatory.
- Security is ensured through authentication and role-based access.

3. SYSTEM ANALYSIS

3.1 Problem Definition

The current transportation system lacks a structured carpooling solution. Passengers often face high costs and traffic congestion, while drivers travel with empty seats.

3.2 Fact Finding Techniques

Techniques used: interviews, observation, studying existing apps, and questionnaires to understand user requirements and challenges.

3.3 Need for Computerization

Computerization enables automated ride matching, reduces manual efforts, and ensures efficient management of users, bookings, and payments.

3.4 Process Model

The system follows the Incremental Model of SDLC, with features delivered in stages including user registration, ride management, booking, and reporting.

3.5 Requirement Analysis

Key entities include Users, Vehicles, Rides, Bookings, and Feedback. Relationships are defined in ER diagrams and represented through structured database tables.

3.6 Feasibility Study

- Operational Feasibility – Users can easily access via web browsers.
- Technical Feasibility – Uses widely available technologies like PHP and MySQL.
- Schedule Feasibility – Project can be developed in 3-4 months.
- Economic Feasibility – Low-cost academic project requiring minimal infrastructure.

3.7 Requirement Validation

Requirements validated through prototype testing and user feedback. Ensures clarity, accuracy, and consistency of features.

3.8 Study of Current System

Currently, carpooling is managed informally through social groups without structured applications. This creates reliability and security concerns.

3.9 Problems & Weakness of Current System

- Lack of proper ride scheduling.
- No central database.
- Manual coordination between drivers and passengers.
- Security and trust issues.

3.10 Requirement of New System

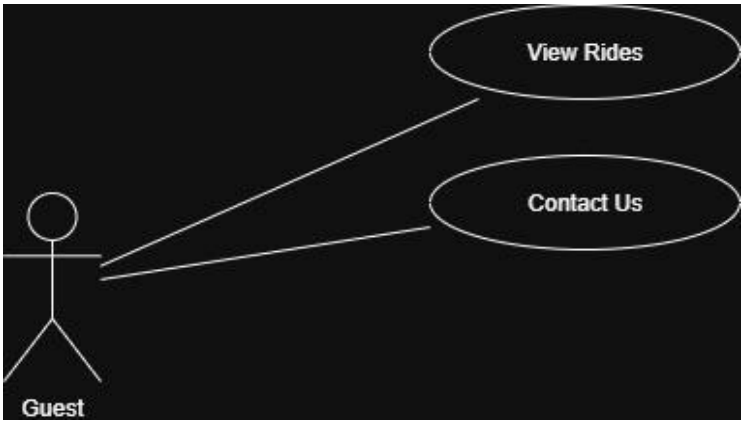
- Centralized carpooling platform.
- Secure authentication and role management.
- Easy ride booking and cancellation.
- Feedback and rating system.
- Admin management for system integrity.

4. SYSTEM DESIGN

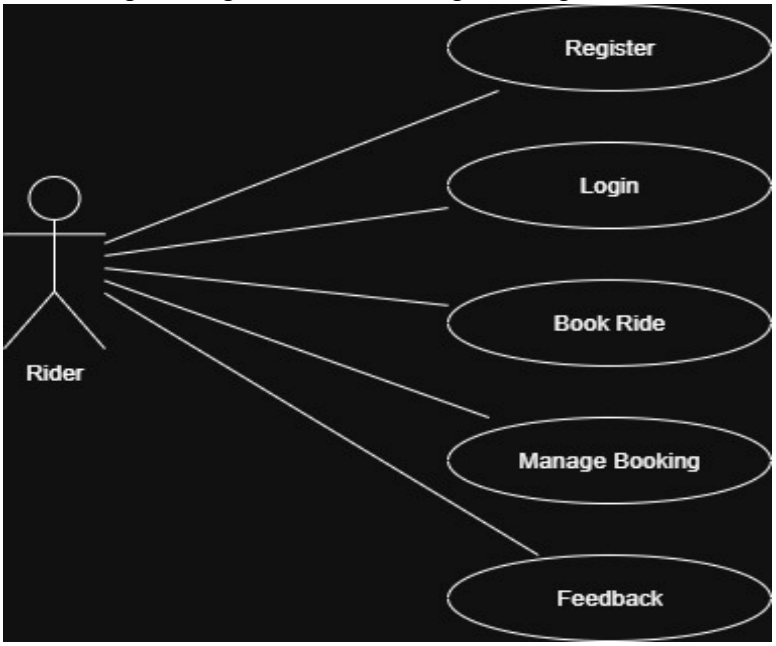
System design includes Data Flow Diagrams (DFD), Entity-Relationship (ER) diagrams, Use Case diagrams, Activity diagrams, Sequence diagrams, and Flowcharts. The database schema is detailed in the Data Dictionary.

Use Case Diagram (Text Explanation):

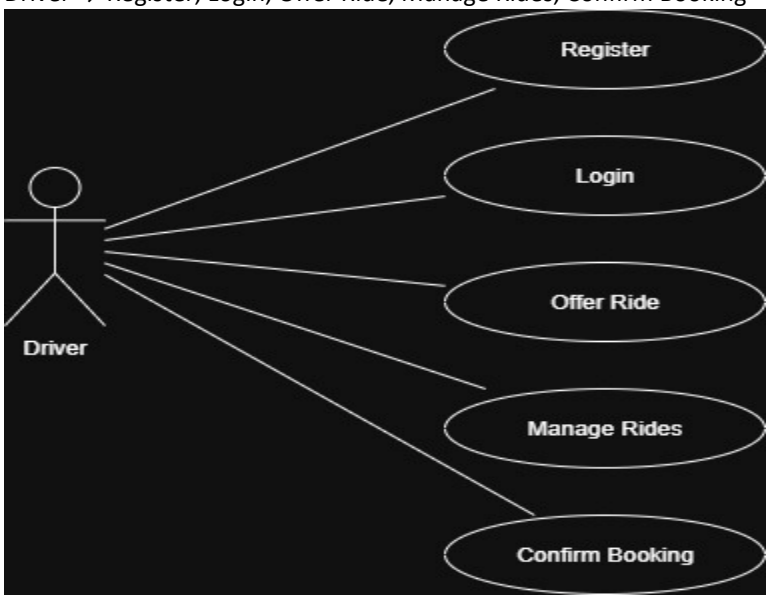
- Guest → View Rides, Contact Us



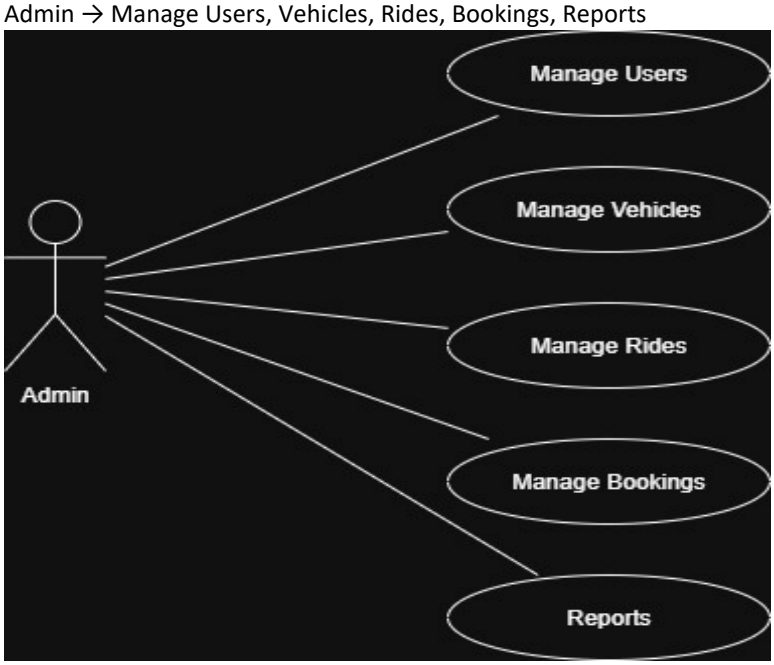
- Rider → Register, Login, Book Ride, Manage Booking, Feedback



- Driver → Register, Login, Offer Ride, Manage Rides, Confirm Booking

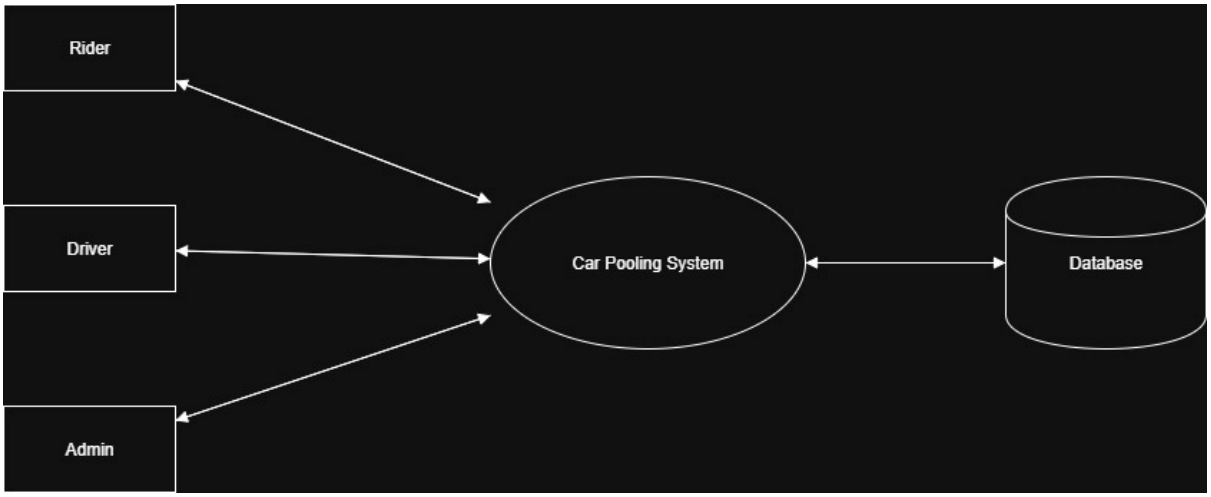


- Admin → Manage Users, Vehicles, Rides, Bookings, Reports

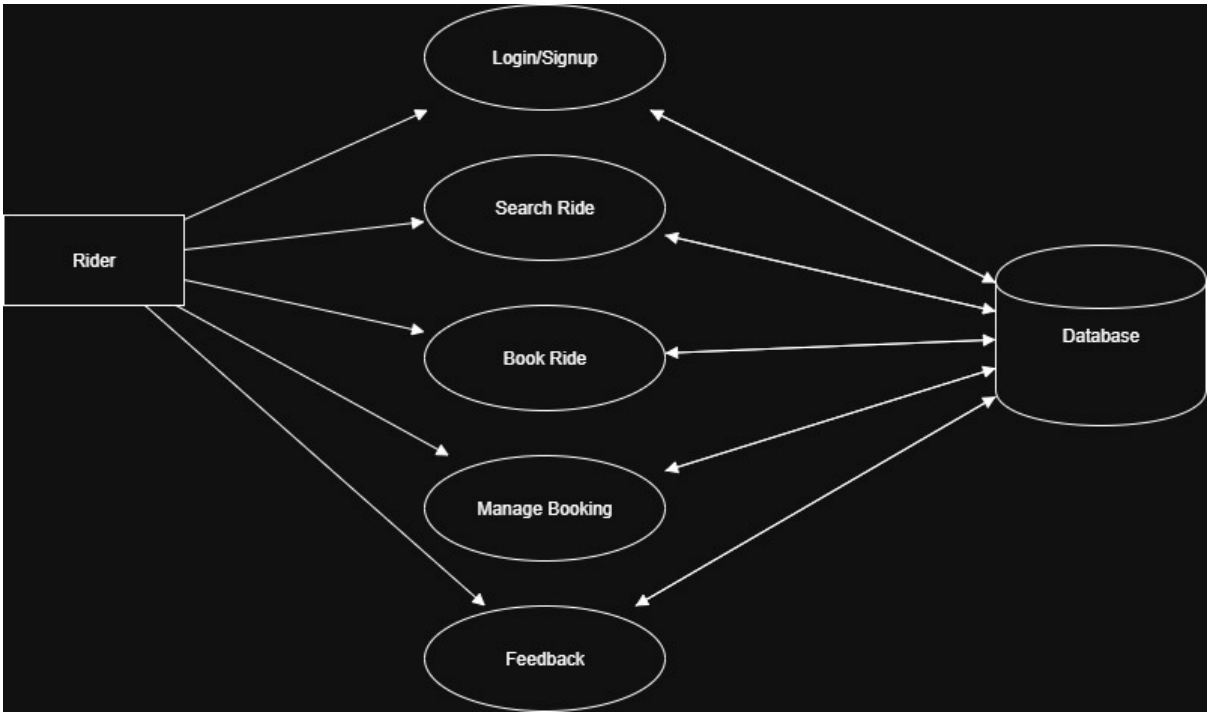


DFD – Data Flow Diagram (Explanation):

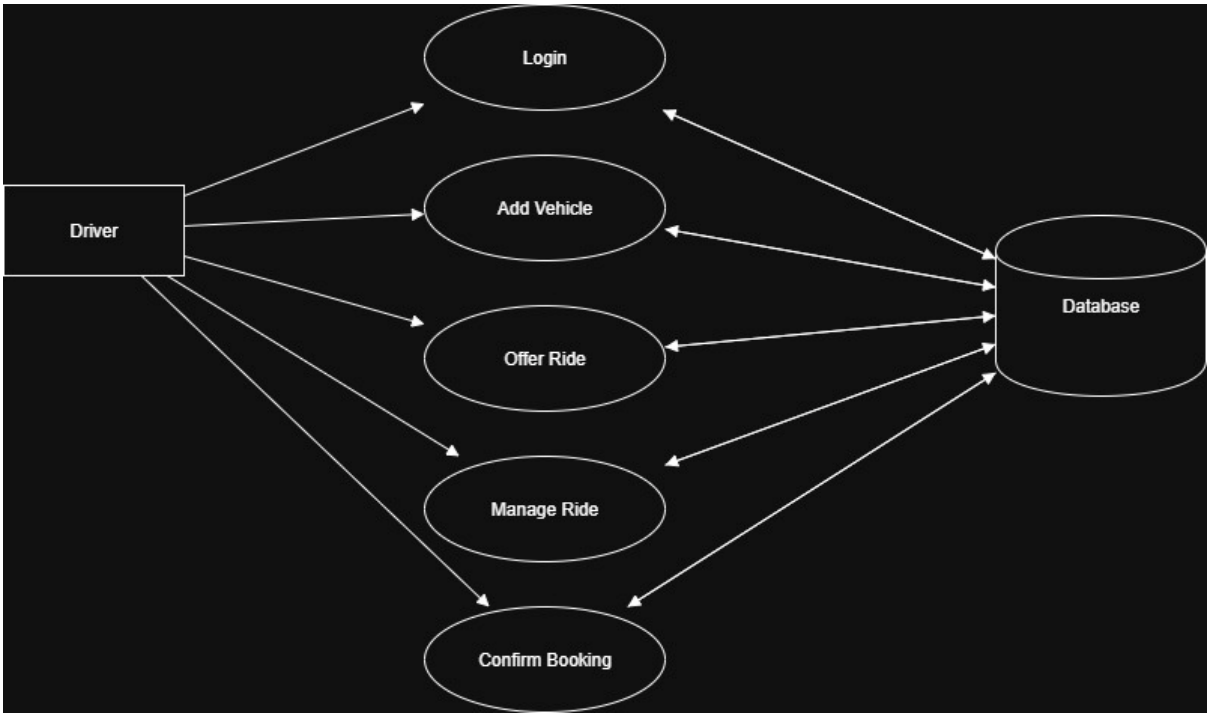
- **Level 0 (Context Diagram):** Users (Driver, Rider, Admin) interact with Car Pooling System → Database



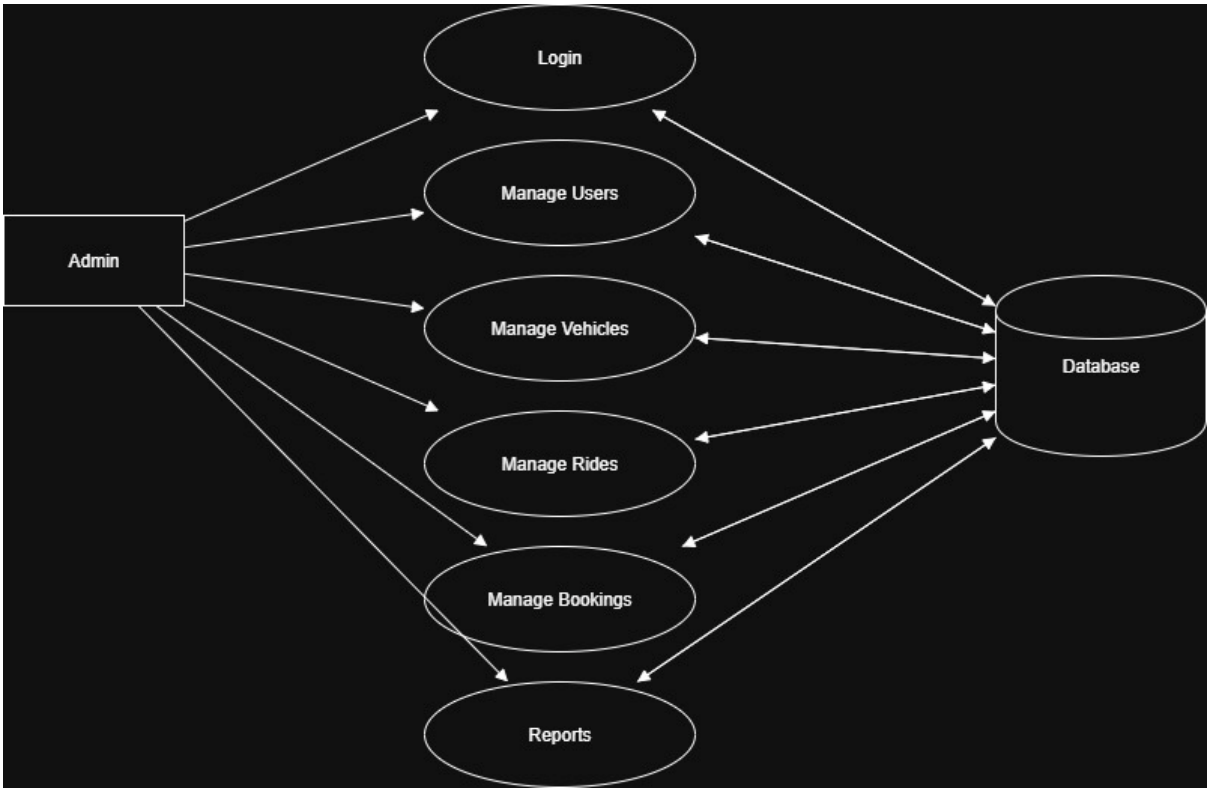
- **Level 1 (Rider side):** Rider → (Login/Signup, Search Ride, Book Ride, Manage Booking, Feedback) → Database



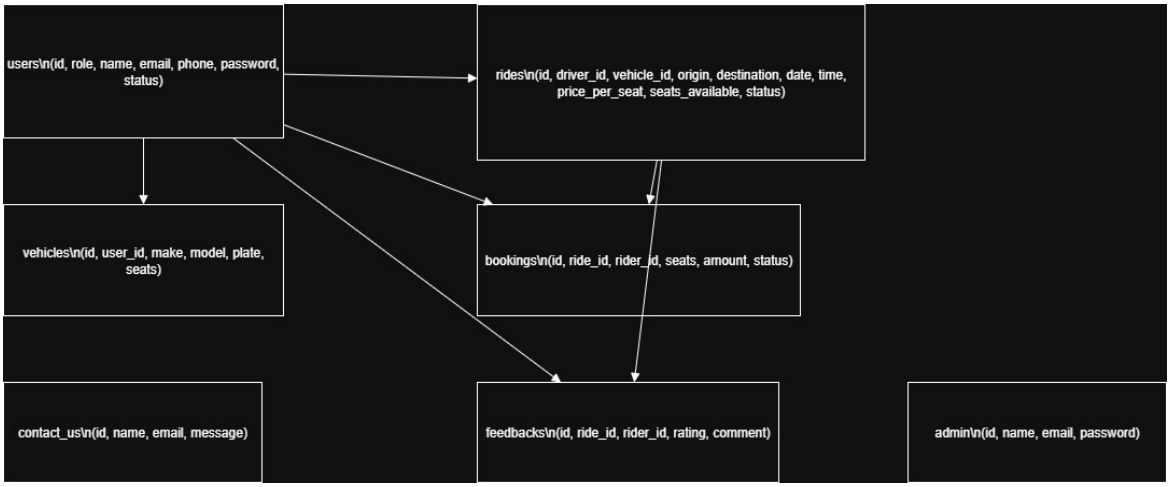
- **Level 1 (Driver side):** Driver → (Login, Add Vehicle, Offer Ride, Manage Ride, Confirm Booking) → Database



- **Level 1 (Admin side):** Admin → (Login, Manage Users, Vehicles, Rides, Bookings, Reports) → Database



- ER Diagram:



4.1 Data Dictionary

Tables include: users, vehicles, rides, bookings, feedbacks, contact_us, and admin. Each table includes primary keys, foreign keys, and attributes for efficient storage.

Total Number Of Table:

- users.tbl
- vehicles.tbl
- rides.tbl
- admin.tbl
- bookings.tbl
- feedbacks.tbl
- contact_us.tbl

Database & table create :

CREATE DATABASE carpooling;

CREATE TABLE users(id INT PRIMARY KEY AUTO_INCREMENT,role ENUM('rider','driver','admin') DEFAULT 'rider', name VARCHAR(255), email VARCHAR(255) UNIQUE, phone VARCHAR(20), password VARCHAR(255), status TINYINT DEFAULT 1);

CREATE TABLE admin(id INT PRIMARY KEY AUTO_INCREMENT, name VARCHAR(255), email VARCHAR(255), password VARCHAR(255));

CREATE TABLE vehicles(id INT PRIMARY KEY AUTO_INCREMENT, user_id INT, make VARCHAR(255), model VARCHAR(255), plate VARCHAR(50),seats INT, FOREIGN KEY(user_id) REFERENCES users(id));

CREATE TABLE rides(id INT PRIMARY KEY AUTO_INCREMENT,driver_id INT, vehicle_id INT, origin VARCHAR(255), destination VARCHAR(255), date DATE, time TIME, price_per_seat INT, seats_available INT, status ENUM('active','completed','cancelled') DEFAULT 'active', FOREIGN KEY(driver_id) REFERENCES users(id), FOREIGN KEY(vehicle_id) REFERENCES vehicles(id));

CREATE TABLE bookings(id INT PRIMARY KEY AUTO_INCREMENT,ride_id INT, rider_id INT,seats INT,amount INT,status ENUM('pending','confirmed','cancelled') DEFAULT 'pending',FOREIGN KEY(ride_id) REFERENCES rides(id),FOREIGN KEY(rider_id) REFERENCES users(id));

CREATE TABLE feedbacks(id INT PRIMARY KEY AUTO_INCREMENT,ride_id INT, rider_id INT, rating INT, comment VARCHAR(255), FOREIGN KEY(ride_id) REFERENCES rides(id),FOREIGN KEY(rider_id) REFERENCES users(id));

CREATE TABLE contact_us(id INT PRIMARY KEY AUTO_INCREMENT,name VARCHAR(255), email VARCHAR(255),message VARCHAR(500));

5. CONCLUSION & DISCUSSION

The Car Pooling System successfully addresses the issues of transportation cost, environmental impact, and ride availability. It is user-friendly, scalable, and secure, with clear modules for Rider, Driver, and Admin. The project demonstrates the application of SDLC principles in real-world software development.