**ParkEase: A Smart Parking Web Application.**

**1. Introduction**

**1.1 Overview**

In modern urban environments, one of the most significant challenges faced by drivers is finding a suitable parking spot efficiently. Traffic congestion, lack of real-time parking information, and inefficient reservation systems often lead to increased fuel consumption, wasted time, and environmental pollution. To address these concerns, **ParkEase** is developed as an innovative **smart parking web application** that simplifies the process of parking spot reservation and management.

The system is designed to provide **real-time updates on parking availability**, offer **location-based services**, and incorporate a **secure and intuitive reservation system** that ensures seamless user experience. Furthermore, the integration of **real-time session management**, **email notifications**, and **cloud-based deployment** makes ParkEase a scalable and highly efficient solution.

**1.2 Motivation**

The primary motivations behind the development of ParkEase include:

* The increasing number of vehicles in urban areas leading to parking challenges.
* The need for a **technology-driven solution** that minimizes human intervention.
* Enhancing user convenience by providing an advanced **reservation system**.
* Implementing an **eco-friendly** approach by optimizing parking lot usage.

**1.3 Objectives**

The main objectives of ParkEase are as follows:

* **User-Friendly Interface:** Develop an intuitive and responsive front-end application.
* **Real-Time Availability Updates:** Provide dynamic updates on parking spot availability.
* **Efficient Parking Reservation:** Allow users to book spots in advance with minimal effort.
* **Email & Notification Alerts:** Enhance user engagement through instant notifications.
* **Secure Authentication:** Implement **JWT-based authentication** for improved security.
* **Scalable Deployment:** Utilize **AWS cloud computing** for efficient data handling.

**2. System Design & Architecture**

**2.1 System Architecture Overview**

The **three-tier architecture** of ParkEase consists of:

1. **Presentation Layer (Frontend):** Developed using **React.js** for an interactive user experience.
2. **Business Logic Layer (Backend):** Implemented with **J2EE and .NET Core** for handling business operations and API endpoints.
3. **Data Layer (Database):** Managed using **MySQL** to store parking records, user data, and reservations.

**2.1.1 Frontend (React.js)**

The frontend is built using **React.js**, ensuring an optimized and responsive user experience. Features include:

* **Dynamic Dashboard** with real-time parking status updates.
* **Interactive** slot selection.
* **User Authentication** using JWT.

**2.1.2 Backend (J2EE & .NET Core)**

The backend is developed using **J2EE** for API handling and **.NET Core** for microservices architecture. Functionalities include:

* **User Management:** Registration, authentication, and role-based access.
* **Parking Slot Management:** Real-time updates, additions, and deletions.
* **Booking System:** Processing reservations and cancellations.
* **Email Notification Service:** Sending automated booking confirmations.
* **Session Management:** Secure handling of active user sessions.

**2.1.3 Database (MySQL)**

The **MySQL database** stores all application-related data, including:

* **User Information:** Profile, contact details, and authentication tokens.
* **Parking Slot Data:** Slot availability, reservations, and pricing.
* **Reservation History:** Reservation details for users and administrators.

**3. Features and Functionalities**

**3.1 User Features**

**3.1.1 Real-Time Parking Spot Availability**

* Users can view live updates on vacant and occupied parking spaces.
* Color-coded indicators show slot status (Green for available, Red for occupied).

**3.1.2 Parking Spot Reservation System**

* Users can **pre-book** a parking slot based on availability.
* Secure booking confirmation sent via email .

**3.1.3 Secure User Authentication**

* Implemented **JWT-based authentication** for login and session security.
* Role-based access control (User, Admin, Parking Employee).

**3.2 Admin Features**

**3.2.1 Parking Lot Management**

* Add, update, or remove parking spots dynamically.

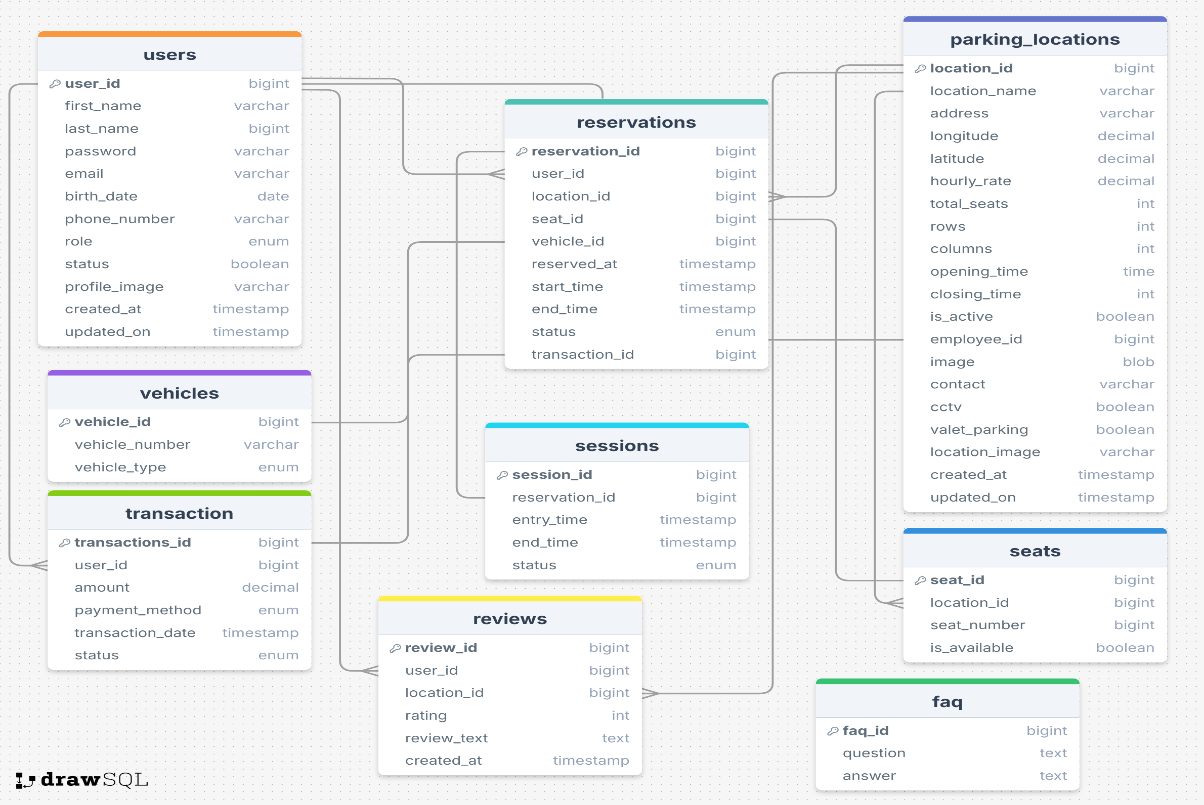
**3.2.2 User Management**

* Monitor registered users.

**3.2.3 Reservation Monitoring**

* View ongoing reservations and real-time parking lot status.
* Generate automated reports for analytics and decision-making.

1. **Diagrams**
2. **ER Diagram**



2. **Deployment Strategy**

**5.1 Cloud Deployment on AWS**

* Hosted on **AWS EC2 instances** for scalable backend services(email service).

**5.2 Security Measures**

* **Token Expiry:** JWT tokens expire after a defined period to prevent unauthorized access.

**6. Challenges & Solutions**

| **Challenge** | **Solution Implemented** |
| --- | --- |
|  |  |
| Real-time Slot Updates | Implemented WebSockets & polling mechanisms. |
|  |  |
| Cross-Platform Compatibility | Developed a responsive React UI for mobile-friendliness. |

**7. Future Enhancements**

**7.1 Location navigation**

* **Based on co-ordinates of location**

**7.2 Dynamic pricing**

* **Dynamic seat pricing according to facilities**

**7.3 Buffer based slot Management**

* **Allow to booking in advanced for a week.**

**8. Conclusion**

ParkEase successfully streamlines the parking experience by leveraging modern web technologies. The system provides real-time availability tracking, an intuitive reservation process, and enhanced user engagement via notifications and security measures. With future integration, ParkEase aims to revolutionize urban parking management for greater efficiency and convenience.

**Prepared By:**

1. Mayur Bhole
2. Parag Chaudhari
3. Pranjal Deshmukh
4. Pankaj Pidurkar

**Project Guide:**

Priyanka Rangole