Project Synopsis

On

FIRST PARK THEN PAY

Submitted by

Udhayarajan J

MCA I (Semester: I)

Guided by
Nidhi Damle

Assistant Professor, MCA Department

P.E. Society's
Modern Institute of Business Studies (Autonomous),
Nigdi, Pune - 411044

2025 - 2026

❖ **Student Name: Udhayarajan J**

❖ **Title Of the Project: First Park Then Pay**

❖ **Name of the Company/Client: Academic Project**

❖ **Abstract:**

The project **"First Park Then Pay"** is a smart parking management system designed to simplify and digitalize the vehicle parking process. It enables users to locate available parking spaces, reserve them, and make payments seamlessly after parking, rather than before. The system aims to minimize congestion, optimize parking space utilization, and improve user convenience. It integrates **Spring Boot RESTful APIs**, **MySQL database**, and **Keycloak** for secure user authentication and authorization. The entire system is containerized using **Docker** for deployment flexibility and scalability.

❖ **Introduction and Objectives of the Project:**

The **First Park Then Pay** application addresses the common challenges faced in urban parking management by providing a digital solution that automates space tracking, booking, and payment.

Objectives:

1. To provide real-time parking availability information to users.
2. To ensure secure and role-based access using **Keycloak authentication**.
3. To allow users to make payments after parking, ensuring transparency.
4. To maintain a centralized database of parking slots, users, and transactions.
5. To containerize the application using **Docker** for easy deployment and maintenance.

6. To develop a scalable **REST API** backend using **Spring Boot** that can integrate with web and mobile clients.

❖ **Project Category: Web Application**

❖ **Tools / Platform, Hardware and Software Requirement specifications:**

   ✓ **Operating System Platform**:
      - Ubuntu 22.04 LTS (Linux Environment)
   ✓ **Hardware Requirements:**
      - **Processor**: Intel Core i3 or equivalent
      - **Hard Disk**: Minimum 50 GB
      - **RAM**: 2 GB or above

   ✓ **Software Requirements:**
      - **Front-End (4GL Programming Language):** HTML5, CSS3, JavaScript (or Angular if applicable)
      - **Middle Layer**: Java 21 with Spring Boot Framework
      - **Back End**: MySQL 8.0
      - **Other Tool**:
         - **Keycloak** – for identity and access management
         - **Docker** – for containerization and deployment
         - **Postman** – for API testing
         - **Git & GitHub** – for version control
         - **Gradle** – for project build and dependency management

❖ **Scope of the Project**

The **"First Park Then Pay"** project aims to provide a complete digital solution for managing parking spaces efficiently in urban and semi-urban areas. The system covers the entire parking workflow — from locating available spaces to managing payments after the parking session — ensuring transparency and user convenience.

The project is designed as a **web-based application** with a modular and scalable architecture, allowing future integration with mobile apps or IoT-based sensors for real-time parking detection.

In Scope:

1. **User Management:** Secure registration, login, and authentication using **Keycloak**.
2. **Parking Slot Management:** Real-time tracking of available, booked, and occupied slots.
3. **Payment Management:** Post-parking payment feature with transaction history.
4. **Admin Dashboard:** Role-based access for administrators to manage users, parking slots, and transactions.
5. **RESTful APIs:** Backend APIs built using **Spring Boot** to handle all core operations.
6. **Database Management:** Storage and retrieval of data using **MySQL**.
7. **Deployment:** Containerized setup using **Docker** for consistent development and production environments.

Out of Scope (Future Enhancements):

1. Integration with **IoT sensors** for automated slot detection.
2. Support for **mobile applications** (Android/iOS).
3. Integration with **UPI or payment gateways** for live online transactions.
4. **Smart parking analytics** and predictive suggestions using AI/ML.

Overall, the system is developed to be **secure, maintainable, and scalable**, providing a strong foundation for future expansion into a complete smart parking ecosystem.

❖ Proposed system

The **proposed system**, *First Park Then Pay*, is a modern, automated parking management solution that addresses the inefficiencies of traditional parking systems. Unlike manual methods or prepaid parking solutions, this system allows users to **park first and pay later**, ensuring transparency, convenience, and efficient resource utilization.

The system is designed as a **web-based application** powered by **Java 21** and **Spring Boot RESTful APIs**, providing secure communication between users and administrators. User authentication and role management are handled using **Keycloak**, ensuring a robust and centralized identity management mechanism.

The backend uses **MySQL** for data storage, maintaining details of users, vehicles, parking slots, and transactions. To ensure easy deployment and scalability, the entire application is containerized using **Docker**, enabling consistent performance across different environments.

Key Features of the Proposed System:

1. **User Authentication and Authorization:**
   - Secure login and role-based access using **Keycloak**.
2. **Parking Slot Management:**
   - Real-time tracking of available and occupied parking slots.
3. **Smart Payment System:**
   - Users are billed after their parking duration, enhancing fairness and flexibility.
4. **Admin Dashboard:**
   - Allows administrators to manage users, slots, and payments efficiently.
5. **REST API Integration:**
   - Backend services developed using **Spring Boot** can be easily integrated with web or mobile clients.
6. **Containerized Deployment:**

- Application and database run as **Docker containers** for consistent and portable environments.

**Advantages of the Proposed System:**

- Eliminates manual errors and delays in slot allocation.
- Ensures security through token-based authentication and role management.
- Reduces congestion by providing real-time slot availability.
- Improves user satisfaction with a post-parking payment model.
- Scalable and adaptable for future IoT or mobile integrations.

❖ **Module Specification:** [No. of Modules & their Specification]

The **"First Park Then Pay"** application is organized into **four main modules**, each focusing on a specific functionality within the system. It follows a **SaaS (Software as a Service)** model where each customer (parking provider) can create an account and manage their own parking setup, charges, and billing through a secure web application.

1. User Management Module
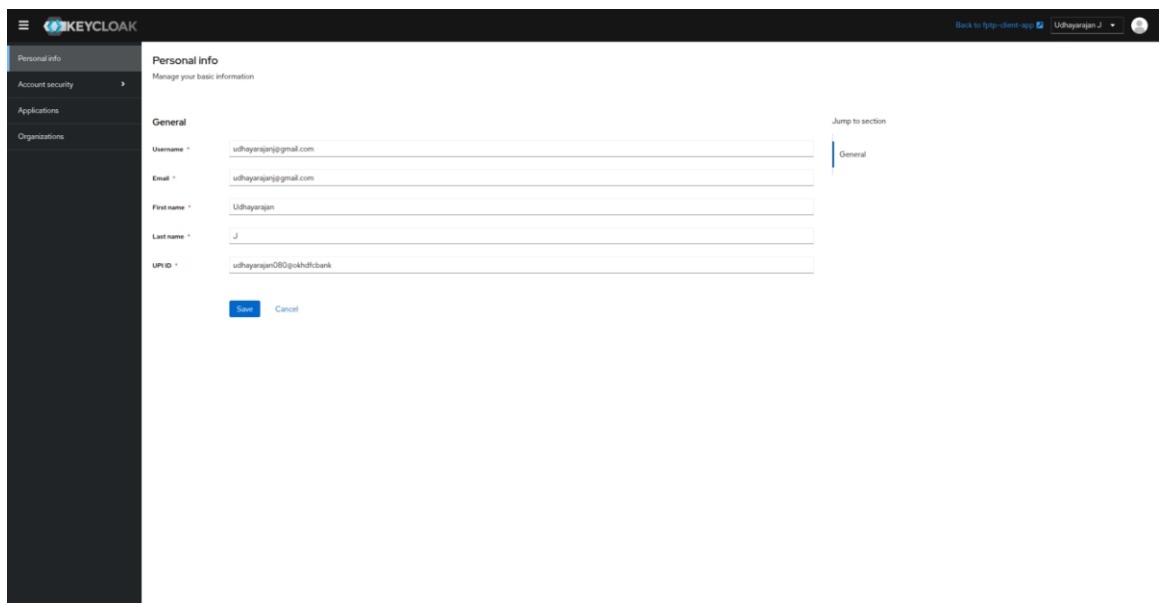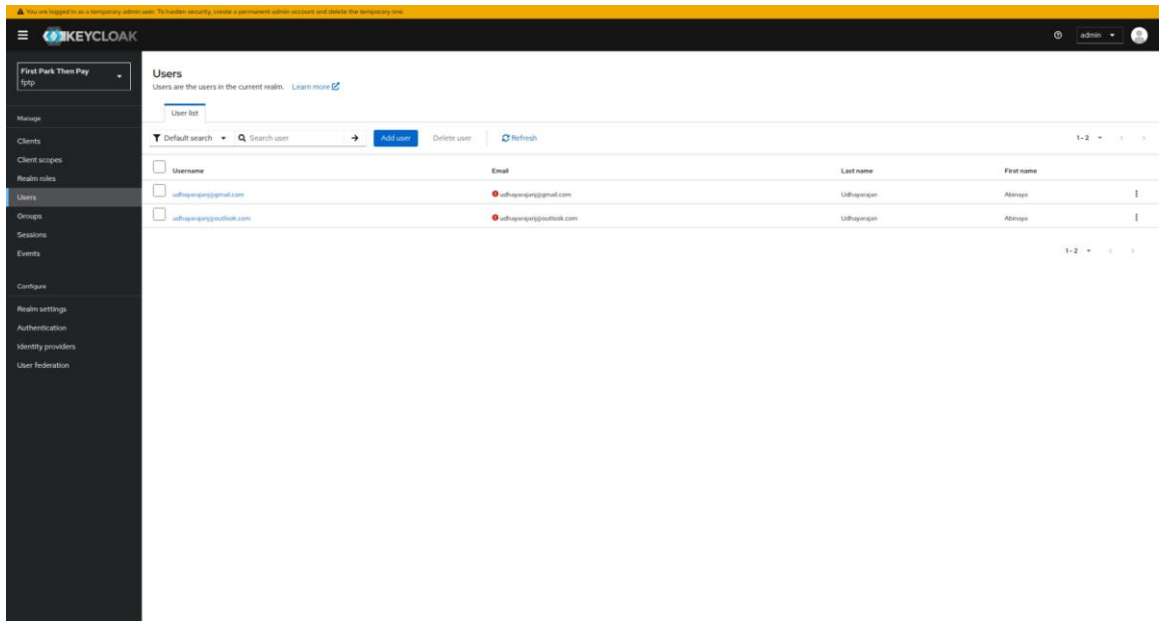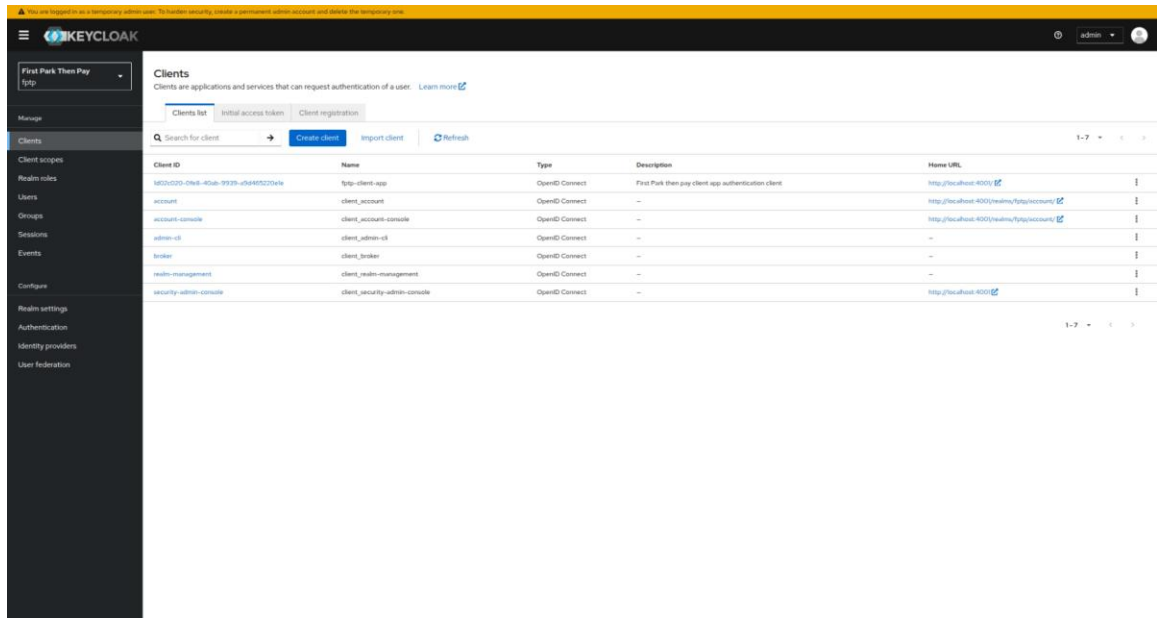
Purpose:
Manages all user accounts, authentication, and authorization through **Keycloak**.

Specifications:

- User authentication and authorization handled by **Keycloak**.
- Role-based access for **Customer (Parking Owner)** and **Admin**.
- Each customer can register and maintain their own parking business account.
- Secure login, token-based access, and session handling.
- Profile details and credentials synced from Keycloak.

2. Vehicle Management Module

Purpose:

Maintains vehicle type configurations and associates parking slots with those types.

Specifications:

- Customers can create and manage vehicle types (e.g., Car, Bike, Van, EV, etc.).
- Each vehicle type includes configuration details such as:
    - **Block Name**
    - **Slot Number**
    - **Per-Hour Parking Charges**
- Vehicle type data is used for billing and parking allocation.
- Dropdown-based vehicle type selection ensures consistency.
- Allows expansion for new vehicle types as needed.

3. Billing and Payment Module

Purpose:

Handles billing, payment tracking, and transaction verification for parked vehicles.

Specifications:

- Automatically calculates charges based on **parking duration** and **vehicle type**.
- Payment mode options: **Cash** or **Online**.
- Validates whether the bill is paid or pending before releasing the slot.
- Maintains detailed payment history for both admin and users.
- Integrated with vehicle data for accurate charge mapping.

4. Dashboard Module
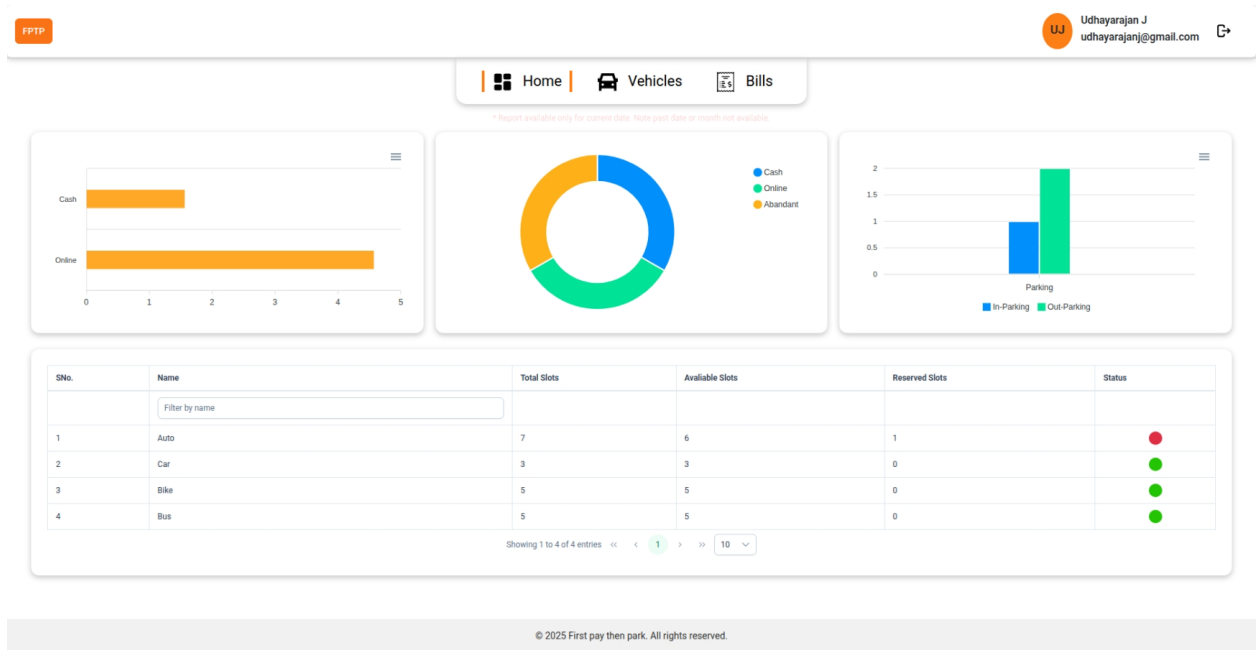
Purpose:

Acts as the central interface for both users and admins to monitor system activities and performance.

Specifications:

- **User Dashboard:** Displays registered vehicle types, created blocks/slots, and transaction history.
- **Admin Dashboard:** Summarizes customer accounts, total payments, and parking activities.
- Real-time statistics such as total vehicles parked, earnings, and slot usage.
- Data visualizations through charts or tables for better analysis.

❖ **Bibliography:**

- **Angular Framework Documentation**
  https://angular.io/docs
- **Node.js Official Documentation**
  https://nodejs.org/en/docs
- **Java 21 Documentation (Oracle)**
  https://docs.oracle.com/en/java/
- **Spring Boot Framework Documentation**
  https://docs.spring.io/spring-boot/docs/current/reference/html/
- **Keycloak Documentation**
  https://www.keycloak.org/documentation
- **MySQL Database Documentation**
  https://dev.mysql.com/doc/
- **Docker Documentation**
  https://docs.docker.com/
- **Visual Studio Code Documentation**
  https://code.visualstudio.com/docs
- **IntelliJ IDEA Community Edition Documentation**
  https://www.jetbrains.com/idea/documentation/
- **Project Source Code (GitHub Repository)**
  https://github.com/UdhayarajanJ/2202561-MCA-2025-2027.git

Student Email: udhayarajanj@gmail.com

Student Contact No: 8248748016

Date: 00-00-0000   **_____**

Place: Pune   **Signature of Student**