PROJECT REPORT
ON
CAR PARKING MANAGEMENT SYSTEM

Carried Out at

CENTRE FOR DEVELOPMENT OF ADVANCED COMPUTING
ELECTRONIC CITY, BANGALORE.

UNDER THE SUPERVISION OF

Ms. Karuna Prasad
[Scientist E, C-DAC Bangalore]

Submitted By

Priyanshu Patidar	(250850120127)
Akshay Uparikar	(250850120020)
Anshul Lodhi	(250850120032)
Anurag Tiwari	(250850120035)
Manohar Dudhat	(250850120097)

PG DIPLOMA IN ADVANCED COMPUTING/PG DIPLOMA IN IT INFRASTRUCTURE, SYSTEMS & SECURITY
C-DAC, BANGALORE

Candidate's Declaration

We hereby certify that the work being presented in the report entitled **"Car Parking Management System"**, submitted in the partial fulfillment of the requirements for the award of **Post Graduate Diploma in Advanced Computing (PG-DAC)** and submitted in the department of **Advanced Computing** of the **C-DAC Bangalore**, is an authentic record of our work carried out during the period **21st August 2025– 04th February 2026** under the supervision **of Ms.Karuna Prasad (Scientist E), C-DAC Bangalore**.

The matter presented in the report has not been submitted by me for the award of any degree of this or any other Institute/University.

PRN & Student Name:

250850120127 - Priyanshu Patidar

250850120020 - Akshay Uparikar

250850120032 - Anshul Lodhi

250850120035 - Anurag Tiwari

250850120097 - Manohar Dudhat

Counter Signed By:

Name(s) of Supervisor(s)

Karuna

Karuna, Scientist E

ACKNOWLEDGMENT

We take this opportunity to express my gratitude to all those people who have been directly and indirectly with me during the competition of this project "Car Parking Management System".

We pay thank to Ms. Karuna Prasad who has given guidance and a light to me during this major project. His versatile knowledge about "Car Parking Management System" has eased us in the critical times during the span of this Final Project.

We acknowledge here out debt to those who contributed significantly to one or more steps. We take full responsibility for any remaining sins of omission and commission**.**

Date: 31-01-2026

SIGN:

Project Guide:
Karuna Prasad

Student Name:
Priyanshu Patidar(250850120127)
Akshay Uparikar(250850120020)
Anshul Lodhi(250850120032)
Anurag Tiwari(250850120035)
Manohar Dudhat(250850120097)
<PG-DAC Aug 2025>

ABSTRACT

The Car Parking Management System is a robust web-based application designed to automate, streamline, and optimize parking operations across commercial complexes, residential communities, and public infrastructures. Unlike traditional manual systems that relied on paper tickets and human attendants, the proposed solution leverages **modern full-stack technologies**, IoT sensors, and AI-powered cameras to provide real-time information on parking slot availability, thereby reducing vehicle search time, traffic congestion, and environmental pollution. The system supports online slot booking, vehicle entry and exit management through RFID tags, QR codes, and automatic license plate recognition, while ensuring automated billing and secure online payments via integrated payment gateways. Role-based access control separates functionalities for administrators, parking staff, and end-users, enabling efficient monitoring, reporting, and **operational transparency**. Cloud deployment ensures scalability, high availability, and seamless integration with smart city infrastructure, while database-driven architectures—both relational and NoSQL—facilitate efficient handling of large volumes of parking and user data. Advanced features such as dynamic pricing models, predictive analytics, and big data dashboards allow administrators to forecast demand, optimize space utilization, and conduct revenue analysis. Integration with intelligent transportation systems, and public transit hubs further enhances sustainability and user convenience. Emerging technologies such as blockchain-based payments, edge computing for low-latency sensor data processing, and augmented reality navigation for drivers are explored to strengthen security, efficiency, and user experience. By **eliminating manual processes** and embracing automation, the system not only improves operational efficiency and user satisfaction but also contributes to broader **urban mobility goals**, environmental sustainability, and the development of smart city ecosystems.

TABLE OF CONTENT

LIST OF DIAGRAMS / SNAPSHOTS

LIST OF TABLES

ABBREVIATIONS & ACRONYMS

- **AI:** Artificial Intelligence
- **QR:** Quick Response
- **NoSQL:** Not Only SQL
- **MERN:** MongoDB, Express, React, Node.js
- **Pcs:** Personal Computers
- **RFID:** Radio Frequency Identification
- **UI:** User Interface
- **React.js**: React+ JavaScript
- **API:** Application Programming Interface
- **REST:** Representational State Transfer
- **ODM:** Object Document Mapper
- **JWT:** JSON Web Token
- **JSON:** JavaScript Object Notation
- **e.g :** For Example
- **HTTP:** HyperText Transfer Protocol
- **INT:** Integer
- **CI:** Continuous Integration
- **CD:** Continuous Deployment
- **QA:** Quality Assurance
- **IEEE:** Institute of Electrical and Electronics Engineers
- **IJCRT:** International Journal of Creative Research Thoughts
- **IoT:** Internet of Things
- **ICTCS:** International Conference on Information and Communication Technology for Competitive Strategies

1. INTRODUCTION

The Car Parking Management System is a full-stack web application built using the MERN stack. It is designed to automate and streamline the management of parking spaces in commercial, residential, and public areas. The system enables real-time tracking of parking slot availability, vehicle entry/exit logging, billing, and user management.

A parking management system is a comprehensive solution that employs a combination of hardware and software components to efficiently manage and monitor parking areas. It involves a network of interconnected devices, sensors, cameras, and parking management software applications to automate various parking processes, including entry and exit control, payment collection, space allocation, and security monitoring. At its core, a parking management system aims to optimize the utilization of available parking spaces, reduce congestion, and enhance the overall parking experience for users. By digitizing and automating manual tasks, the system eliminates the need for traditional paper-based ticketing systems and manual monitoring, leading to increased efficiency and accuracy. The system's hardware components include access control devices such as barriers, gates, ticket dispensers, license plate recognition systems, and payment kiosks. These devices work together to control access to the parking area, issue tickets or digital passes, and facilitate payment transactions. The system is modular, secure, and optimized for performance, making it suitable for deployment in smart city infrastructure.

➢ Overall Description:

1.1 Problem Statement:

Traditional parking systems rely heavily on manual processes, which are prone to errors, inefficiencies, and delays. These systems often fail to provide real-time updates on slot availability, leading to congestion and user dissatisfaction.

1.2 Proposed Solution:

The Car Parking Management System offers a centralized digital platform that streamlines parking operations. It enables users to check slot availability, reserve spaces, make payments, and track their parking history. Administrators can manage zones, monitor occupancy, and generate reports.

➢ **Target Users:**

- General public (vehicle owners)
- Parking lot staff
- Facility administrators.

➢ **Assumptions and Dependencies:**

- Users have access to internet-enabled devices.
- Payment gateway integration is available.
- Parking lots are equipped with basic hardware (PCs, scanners).
- IoT sensors may be integrated for real-time slot tracking.

1.3 Functionalities of Car Parking Management System:

- View available parking slots.
- Register vehicles and assign slots.
- Track entry and exit times.
- Calculate parking charges based on duration.
- Online payment integration.
- Admin dashboard for slot and rate management.
- Generate reports on usage and revenue.
- SMS/email notifications for booking confirmations and due alerts.

1.4 Objectives and Scope:

- Authenticate users and provide personalized dashboards.

- Allow users to view parking history and payment records.
- Enable admins to add/remove slots, update rates, and monitor occupancy.
- Support multiple parking zones and vehicle types.
- Provide alerts for full occupancy and overdue vehicles.
- Ensure secure and encrypted data handling.

2. LITERATURE SURVEY

Researchers have identified parking congestion as a major urban problem caused by the rapid increase in vehicles and inefficient traditional parking systems. Early parking management systems were manual and paper-based, leading to errors, lack of real-time information, and poor space utilization. Many studies propose smart parking systems that use software platforms to provide real-time parking slot availability, reducing vehicle search time and traffic congestion. Several research papers highlight the use of IoT sensors and cameras to detect vehicle presence and automatically update parking slot status. Web-based parking management systems have been widely discussed, focusing on centralized control, user authentication, and automated billing. Mobile and web applications are commonly used in modern parking systems to allow users to search, reserve, and pay for parking slots online.

Research shows that real-time parking guidance systems significantly reduce fuel consumption and environmental pollution by minimizing unnecessary driving. Some studies emphasize the importance of role-based access control, separating functionalities for administrators, parking staff, and users. Secure payment mechanisms using online payment gateways are recommended to improve transparency and reduce manual cash handling. Database-driven systems using NoSQL or relational databases are preferred for handling large volumes of parking and user data efficiently. Recent research explores cloud-based parking systems, which improve scalability, availability, and ease of deployment. A few studies investigate QR code and RFID-based access control, enabling faster vehicle entry and exit management.

Many studies propose smart parking systems that use software platforms to provide real-time parking slot availability, reducing vehicle search time and traffic congestion. Several research papers highlight the use of IoT sensors and cameras to detect vehicle presence and automatically update parking slot status. Web-based parking management systems have been widely discussed, focusing on centralized control, user authentication, and automated billing. Mobile and web applications are commonly used in modern parking systems to allow users to search, reserve, and pay for parking slots online.

Research shows that real-time parking guidance systems significantly reduce fuel consumption and environmental pollution by minimizing unnecessary driving. Some studies emphasize the importance of role-based access control, separating functionalities for administrators, parking staff, and users. Secure payment mechanisms using online payment gateways are recommended to improve transparency and reduce manual cash handling. Database-driven systems using NoSQL or relational databases are preferred for handling large volumes of parking and user data efficiently. Recent research explores cloud-based parking systems, which improve scalability, availability, and ease of deployment. A few studies investigate QR code and RFID-based access control, enabling faster vehicle entry and exit management.

Researchers have also proposed the use of data analytics to generate parking usage reports and revenue analysis for administrators. Literature highlights challenges such as system scalability, data security, and integration with existing infrastructure. Most studies conclude that integrating software-driven parking management systems with modern technologies improves operational efficiency and user satisfaction.

3. SOFTWARE REQUIREMENT SPECIFICATION

3.1 User Interface:

The user interface (UI) of the system is designed to be intuitive, responsive, and inclusive, ensuring seamless interaction across both desktop and mobile platforms. By prioritizing usability and accessibility, the interface caters to a diverse range of users, including general visitors, registered members, and administrative staff. The design emphasizes clarity, efficiency, and real-time responsiveness, enabling users to perform essential tasks such as registration, slot booking, and payment with minimal friction.

➢ **Key UI components include**:

- Login and registration pages.
- Dashboard for users and admins.
- Booking and payment screens.
- Accessibility features for differently abled users.

3.2 Hardware Interface:

- Operates on standard PCs, tablets.
- Compatible with barcode scanners, ticket printers, and payment terminals.
- RFID readers for vehicle identification.

3.3 Software Interface:

➢ **Frontend:**

- Framework: React.js
- State Management: Redux (centralized store for predictable state handling)
- Deployment: Vercel (optimized for frontend hosting, serverless functions, and edge caching)
- Flow: User → React UI → Redux store → API calls to backend

➢ **Backend:**

- Runtime: Node.js
- Framework: Express.js (RESTful API endpoints)
- Deployment: Render (backend hosting with autoscaling)
- Flow: Frontend requests → Express routes → Business logic → Database queries → Response to frontend

➢ **Database:**

- Type: NoSQL (MongoDB)
- ODM: Mongoose (schema modeling, validation, query building)
- Flow: Express.js → Mongoose → MongoDB collections

➢ **Authentication:**

- Method: JWT (JSON Web Tokens)
- Security: bcrypt for password hashing
- Flow:
    1. User login → bcrypt verifies password
    2. Server issues JWT → stored in client (localStorage/cookies)
    3. Subsequent requests → JWT verified via middleware

4. PROJECT ARCHITECTURE

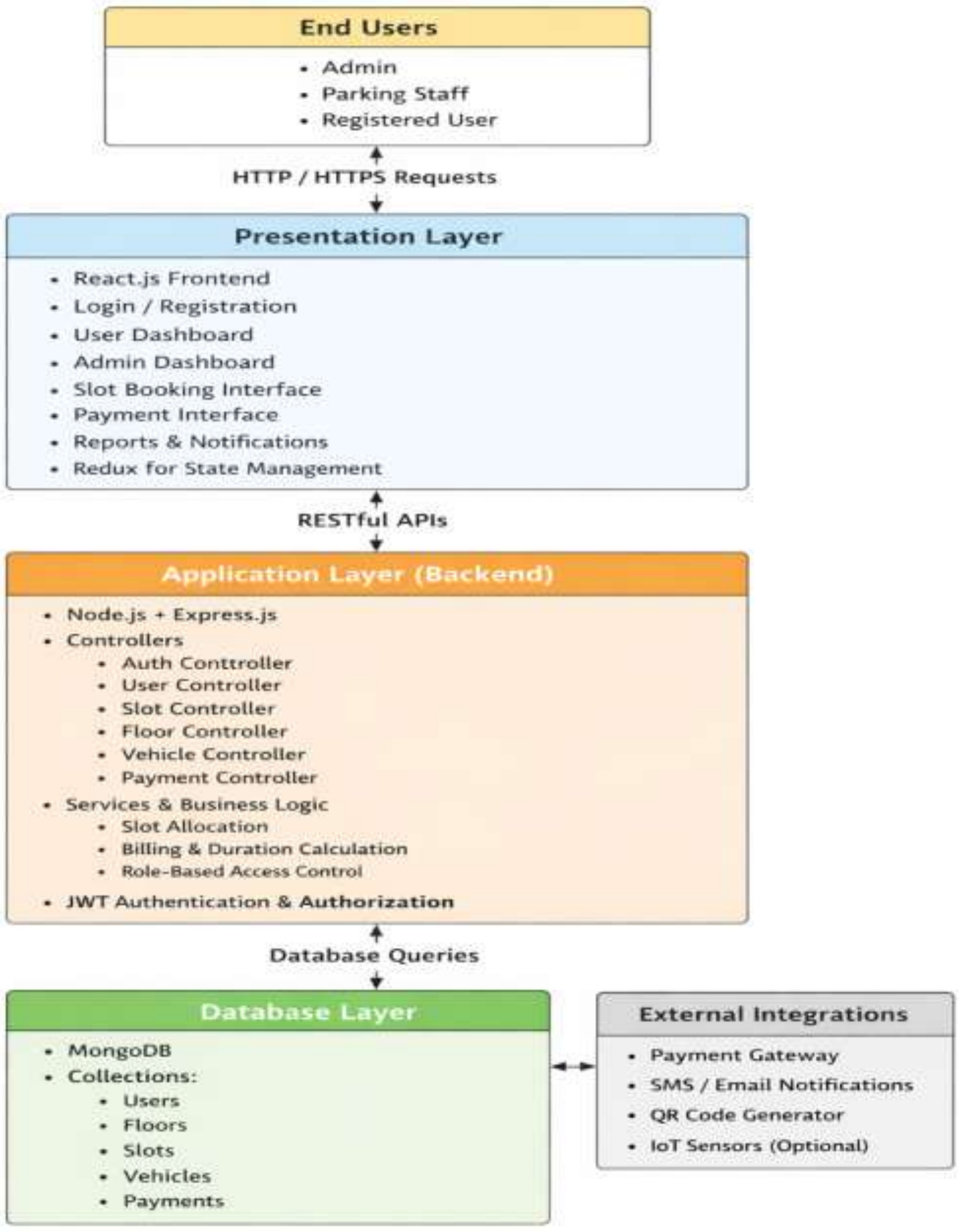


Fig.1 Project Architecture

5. SYSTEM DESIGN OF PROJECT

System design defines the architecture, components, data flow, and interactions that enable the Car Parking Management System to function efficiently. This project follows a modular, role-based, and scalable architecture, ensuring secure and seamless parking operations.

5.1. Design Goals

- Automate parking slot booking and management
- Enable secure, role-based access for users, staff, and admins
- Support real-time slot availability and payment processing
- Ensure scalability, maintainability, and integration with smart city infrastructure

5.2. Three-Tier Model

A. Presentation Layer

Technology: React.js + Redux

➢ **Responsibilities:**

- User interfaces for login, booking, dashboards, and payments
- Responsive design for desktop and mobile
- Real-time updates via Redux state management

B. Application Layer (Backend)

Technology: Node.js + Express.js

➢ **Responsibilities:**

- RESTful APIs for all operations
- Controllers for user, slot, floor, vehicle, and payment

- Business logic: slot allocation, billing, QR generation
- JWT-based authentication and role-based authorization

C. Data Layer

Technology: MongoDB + Mongoose ODM

- **Responsibilities:**

- Collections: Users, Slots, Floors, Vehicles, Bookings, Payments
- Schema validation and indexing

- **Integration with:**

- Payment Gateway (e.g., Razorpay)
- QR Code Generator
- SMS/Email Notification Service

5.3. Role-Based System Design

A. User (Driver)

- Search and book parking slots
- Make online payments
- View booking history and QR codes

B. Security/Parking Staff

- Verify QR codes at entry/exit
- Log vehicle movements
- Update slot occupancy status

C. Admin

- Manage users, slots, floors, and pricing
- Assign staff to zones
- View analytics and generate reports
- Configure system settings and integrations

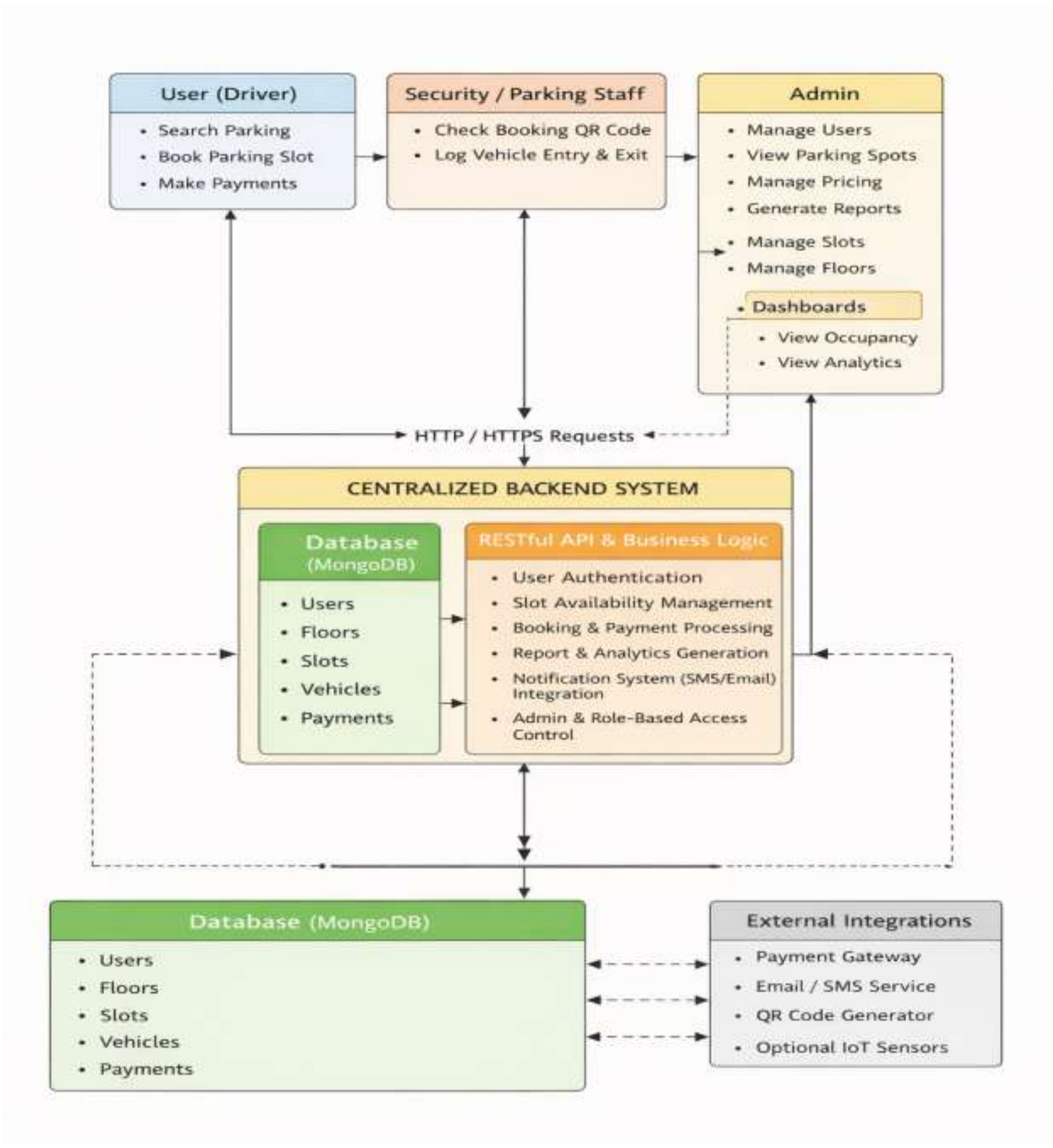


Fig.2 System Design Of Project

➢ Diagram For Car Parking Management System (User):

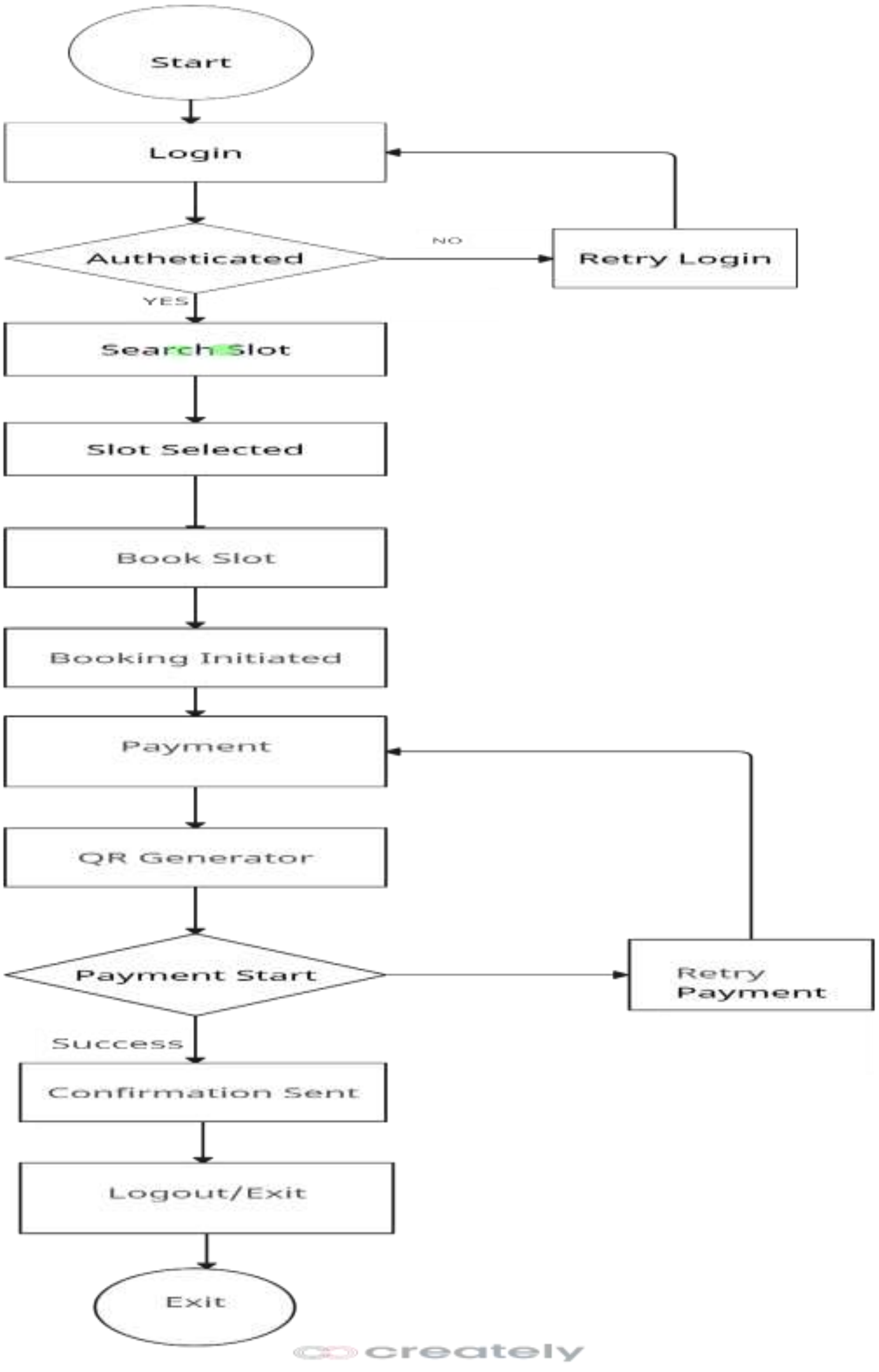


Fig.3 Flow Chart Diagram Of Car Parking Management System(User)

➢ **Class Diagram Admin Section:**

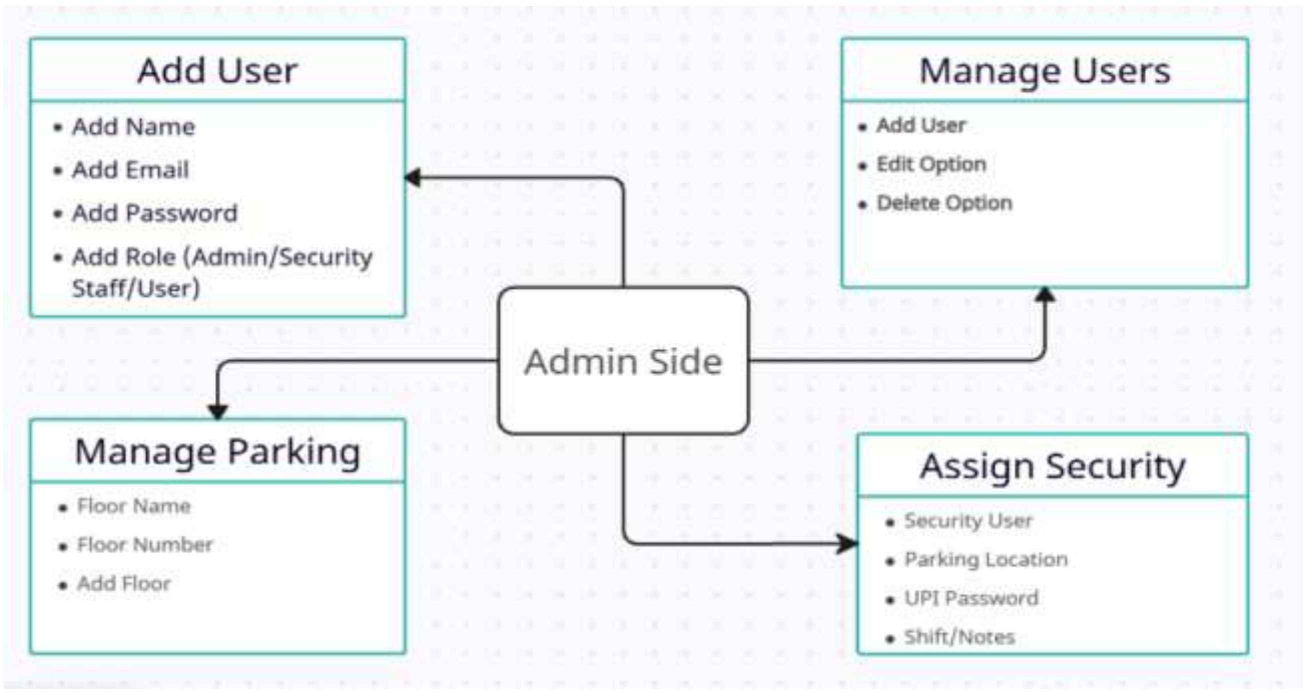


Fig.4 Class Diagram Admin Section

➢ **Class Diagram Security Section:**

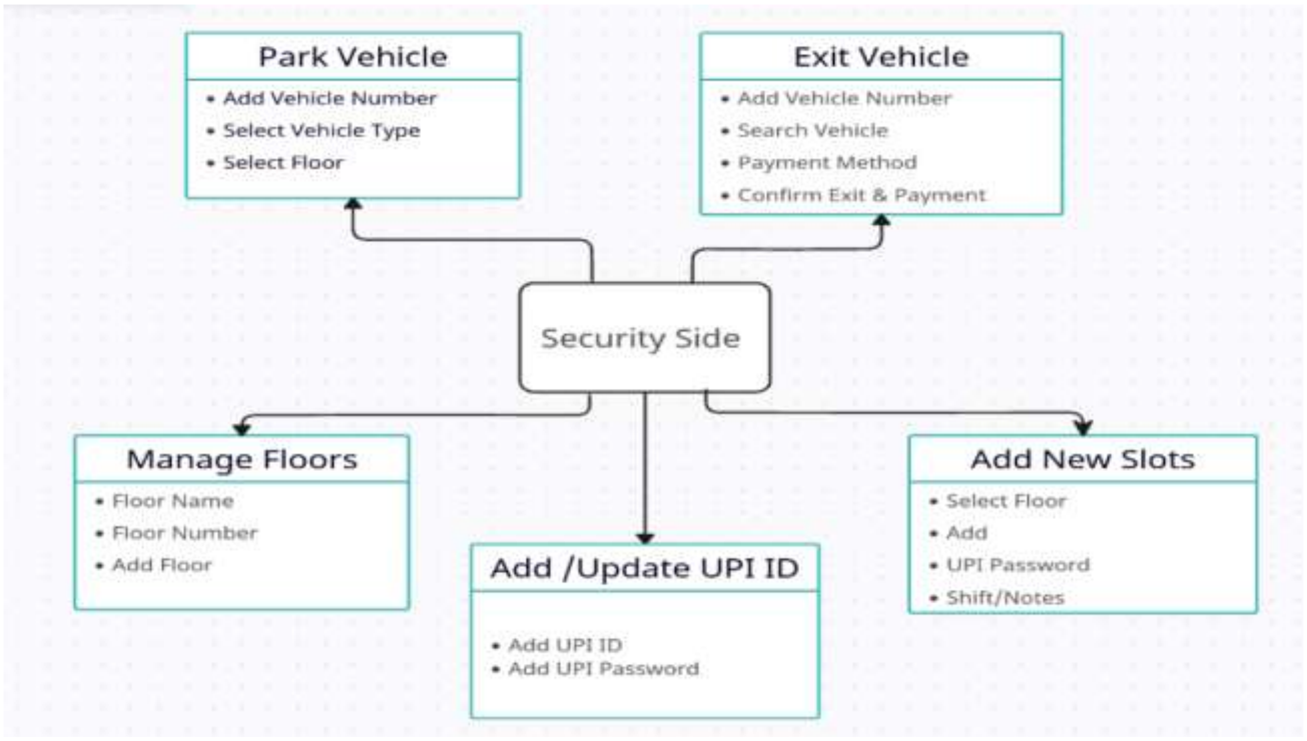


Fig.5 Class Diagram Security Section

5.4 Database Design:

Table 1: USERS

FIELD NAME	DATATYPE	CONSTRAINTS
userId	BIGINT	PRIMARY KEY, AUTO INCREMENT
name	VARCHAR(255)	NOT NULL
email	VARCHAR(255)	UNIQUE
password	VARCHAR(255)	NOT NULL
role	VARCHAR(255)	NOT NULL
createdAt	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP
updatedAt	TIMESTAMP	AUTO UPDATE

Table 2: SLOTS

FIELD NAME	DATATYPE	CONSTRAINTS
slotId	BIGINT	PRIMARY KEY, AUTO INCREMENT
floor	INT	UNIQUE, NOT NULL
slotNumber	INT	UNIQUE, NOT NULL
type	VARCHAR(255)	NOT NULL
isOccupied	BOOLEAN	DEFAULT FALSE
vehicle	VARCHAR(255)	FOREIGN KEY(VEHICLE)
createdAt	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP
updatedAt	TIMESTAMP	AUTO UPDATE

TABLE 3: FLOORS

FIELD NAME	DATATYPE	CONSTRAINTS
floorId	BIGINT	PRIMARY KEY
floorName	VARCHAR(255)	NOT NULL
floorNumber	INT	UNIQUE, NOT NULL
parking	BIGINT	FOREIGN KEY(VEHICLE)
createdAt	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP
updatedAt	TIMESTAMP	AUTO UPDATE

TABLE 4: VEHICLES

FIELD NAME	DATATYPE	CONSTRAINTS
vehicleId	BIGINT	PRIMARY KEY, AUTO INCREMENT
userId	BIGINT	FOREIGN KEY(USERS)
parkingId	BIGINT	FOREIGN KEY(PARKING)
slotId	BIGINT	FOREIGN KEY(SLOT)
number	VARCHAR(255)	UNIQUE, NOT NULL
type	VARCHAR(255)	NOT NULL
entryTime	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP
exitTime	TIMESTAMP	AUTO UPDATE
createdAt	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP
updatedAt	TIMESTAMP	AUTO UPDATE

TABLE 5: PARKINGS

FIELD NAME	DATATYPE	CONSTRAINTS
parkingId	BIGINT	PRIMARY KEY, AUTO INCREMENT
name	VARCHAR(255)	NOT NULL
address	VARCHAR(255)	NOT NULL
location	VARCHAR(255)	UNIQUE
pricing	INT	NOT NULL
Status	BOOLEAN	DEFAULT FALSE
upiID	VARCHAR(255)	NOT NULL
createdAt	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP
updatedAt	TIMESTAMP	AUTO UPDATE

TABLE 6: PAYMENT HISTORIES

FIELD NAME	DATATYPE	CONSTRAINTS
paymenthistoriesId	BIGINT	PRIMARY KEY
userId	BIGINT	FOREIGN KEY
vehicleId	BIGINT	FOREIGN KEY
parkingId	BIGINT	FOREIGN KEY
slotId	BIGINT	FOREIGN KEY(SLOT)
amount	INT	NOT NULL
payementMethod	VARCHAR(255)	NOT NULL
paidAt	TIMESTAMP	CURRENT_TIMESTAMP
createdAt	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP
updatedAt	TIMESTAMP	AUTO UPDATE

Table 7: BOOKINGS

FIELD NAME	DATATYPE	CONSTRAINTS
userId	BIGINT	FOREIGN KEY (USERS)
parkingId	BIGINT	FOREIGN KEY (PARKING)
slotId	BIGINT	FOREIGN KEY (SLOTS)
startTime	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP
endTime	TIMESTAMP	NOT NULL
status	ENUM	DEFAULT NOT RESERVED
totalAmount	INT	NOT NULL
isPreBooked	BOOLEAN	DEFAULT FALSE
paymentStatus	ENUM	DEFAULT PENDING
prebookingFee	INT	NOT NULL
arrivalWindow	INT	NOT NULL
gracePeriod	INT	NOT NULL
createdAt	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP
updatedAt	TIMESTAMP	AUTO UPDATE

6. SYSTEM FEATURES OF PROJECT

- **Authentication & Authorization:**

- JWT-based login and register.
- Role-based access (Admin, Staff/Security, User).

- **Slot Management:**

- Add, update, delete slots.
- Real-time availability status.
- Categorization by zone and vehicle type.
- Slot reservation.

- **Booking & Check-in:**

- Book slots in advance.
- QR code/ticket-based check-in.
- Entry/exit time logging.
- Grace period handling.

- **Billing & Payments:**

- Dynamic rate calculation as per hours/day.
- Online payment via Razorpay.
- Invoice generation.

7. IMPLEMENTATION

The project will culminate in a fully functional **MERN-based web application**, integrating user and admin dashboards, secure authentication, and real-time slot booking capabilities. The implementation phase ensures that all functional and non-functional requirements are met, supported by robust documentation, testing, and deployment strategies.

7.1 Core Application:

- **Responsive Full-Stack Application:** Built using MongoDB, Express.js, React.js, and Node.js, enabling users to register, book slots, make payments, and manage parking sessions seamlessly.
- **Authentication Module:** Secure JWT-based login system with bcrypt encryption, managing user verification and role-based access permissions.

➢ **Dashboards:**

- **User Dashboard:** Slot booking, payment history, and session management.
- **Admin Dashboard:** Add/remove user, add/remove security staff, assign role to security staff.
- **Security Staff Dashboard:** Can Park on spot vehicle user vehicle manually, manage floor, manage slots, payment from user, exit the vehicle.

7.2 Documentation:

- **RESTful API Documentation (Swagger/Postman):** Complete API reference for backend endpoints including authentication, slot management, booking, and payment integration.
- **MongoDB Schema Design:** Optimized schemas for users, slots, bookings, payments, and QR codes, ensuring scalability and performance.
- **User Manual & Installation Guide:** Step-by-step documentation for end users and developers, covering system usage, setup instructions, and troubleshooting tips.

- **Third-Party API Documentation:** Technical references for integrating external services such as Google Maps API and payment gateways.

7.3 Deployment & DevOps:

- **Docker-Based Deployment Scripts:** Containerized environments for frontend, backend, and database services.
- **CI/CD Pipelines:** Automated build, test, and deployment workflows using GitHub Actions or GitLab CI.

➢ **Hosting:**

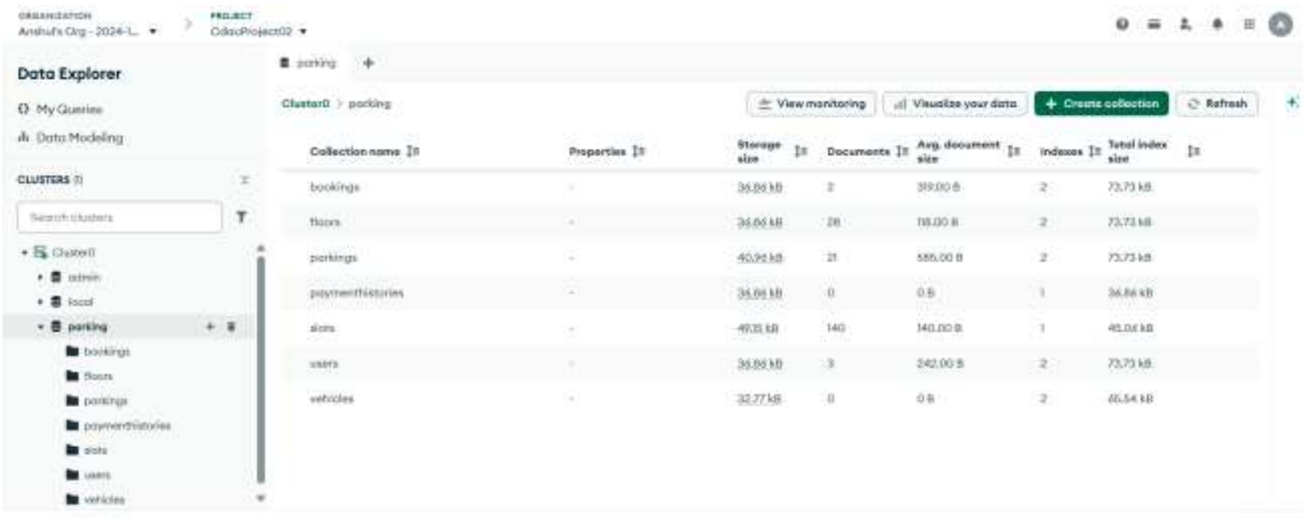
- **Frontend:** Vercel
- **Backend:** Render
- **Database:** MongoDB Atlas

7.4 Deliverables:

- **Source Code Repository:** Git-based repository with version control, branches, commits, and history for collaborative development.
- **Test Cases & QA Reports:** Comprehensive testing suite ensuring reliability, performance, and security.
- **Presentation Assets:** Polished slide deck and demo video showcasing system features, architecture, and use cases for academic or stakeholder review.
- **Detailed Documentation:** User manuals, design specifications, and installation guides to facilitate deployment and user training.

❖ **Project Application Screenshot:**

➢ **Backend Database MongoDB Connection Dashboard:**

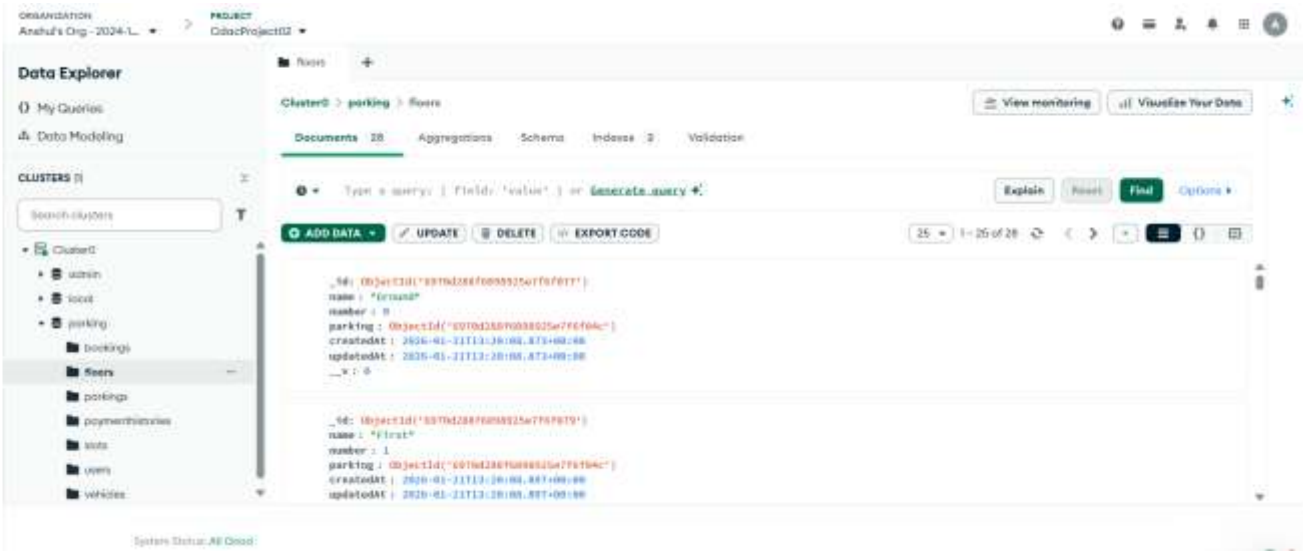


Cluster0 > parking

Collection name	Properties	Storage size	Documents	Avg. document size	Indexes	Total index size
bookings	-	36.86 kB	2	319.00 B	2	73.73 kB
floors	-	36.86 kB	28	118.00 B	2	73.73 kB
parkings	-	40.96 kB	21	585.00 B	2	73.73 kB
paymenthistories	-	36.86 kB	0	0 B	1	36.86 kB
slots	-	49.15 kB	140	140.00 B	1	45.06 kB
users	-	36.86 kB	3	242.00 B	2	73.73 kB
vehicles	-	32.77 kB	0	0 B	2	65.54 kB

Fig.6 Backend Database MongoDB Connection Dashboard

➢ **Backend Database:(Floor)**



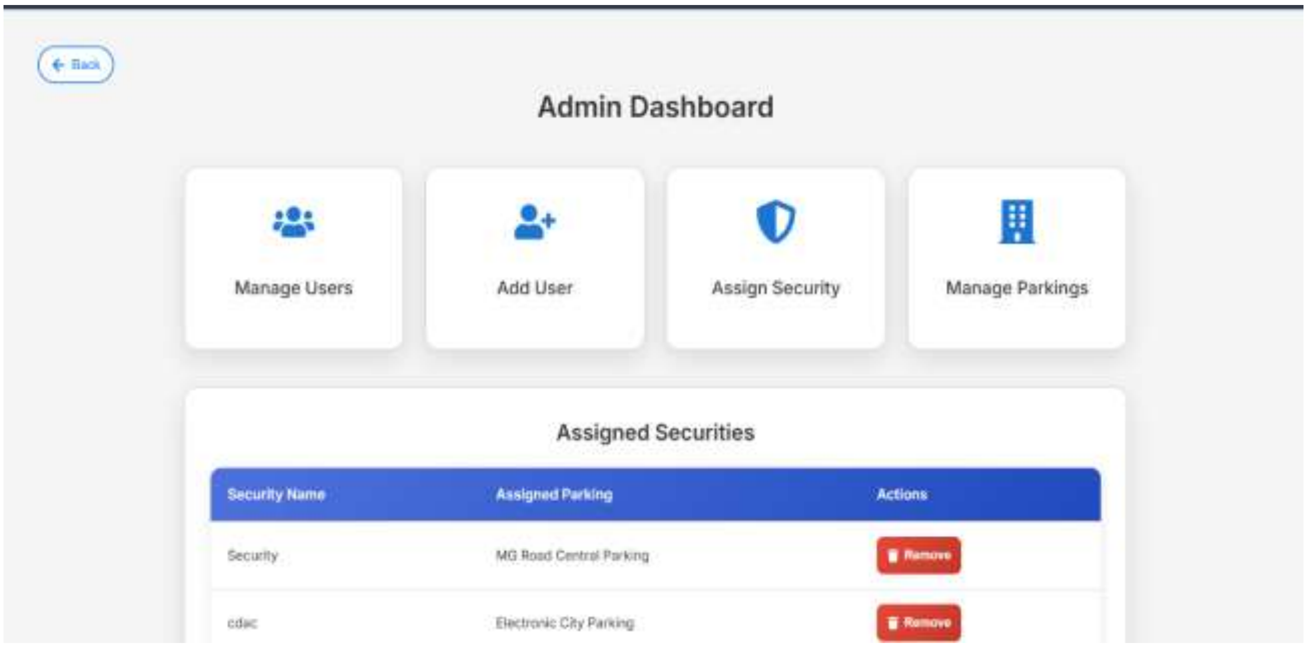
Cluster0 > parking > floors

```
_id: ObjectId('[illegible]')
name : "Ground"
number : 0
parking : ObjectId('[illegible]')
createdAt : 2026-01-21T13:20:08.873+00:00
updatedAt : 2026-01-21T13:20:08.873+00:00
__v : 0

_id: ObjectId('[illegible]')
name : "First"
number : 1
parking : ObjectId('[illegible]')
createdAt : 2026-01-21T13:20:08.887+00:00
updatedAt : 2026-01-21T13:20:08.887+00:00
```

Fig.7 Backend Database(Floor)

➢ Admin Dashboard:



Security Name	Assigned Parking	Actions
Security	MG Road Central Parking	Remove
cdac	Electronic City Parking	Remove

Fig.8 Admin Dashboard

➢ Add New User:

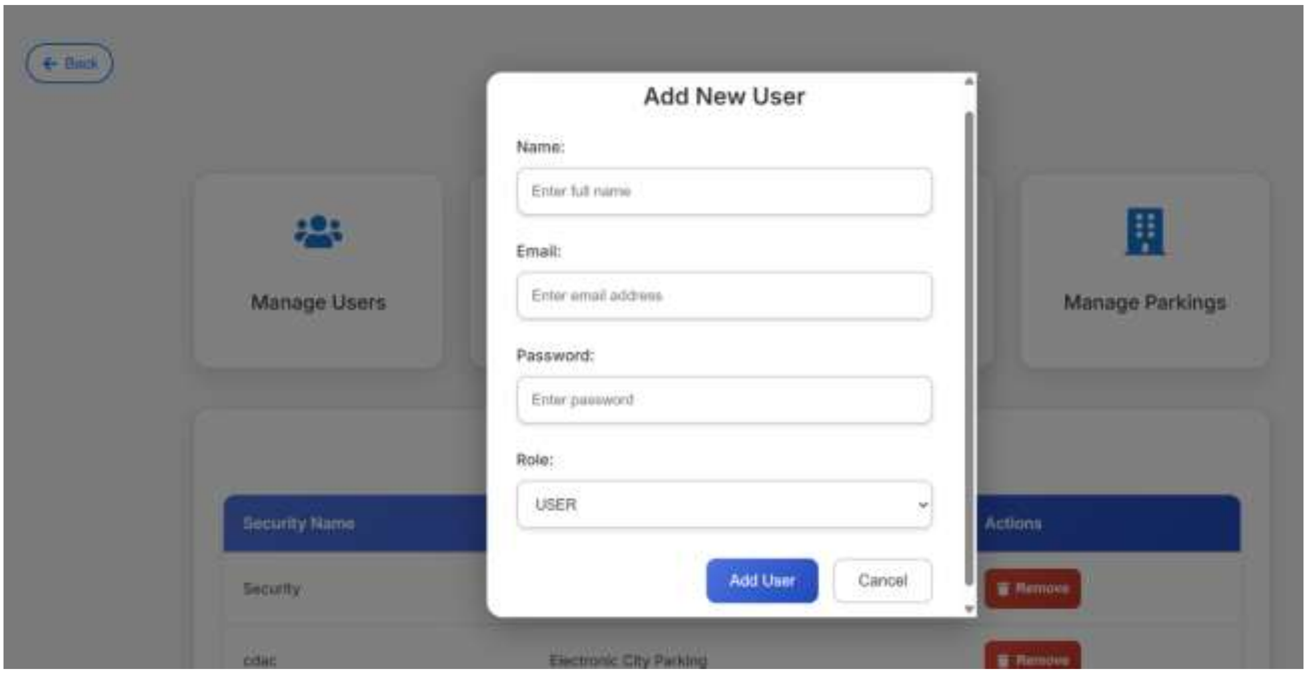


Fig.9 Admin (Add New User)

➢ User Dashboard:

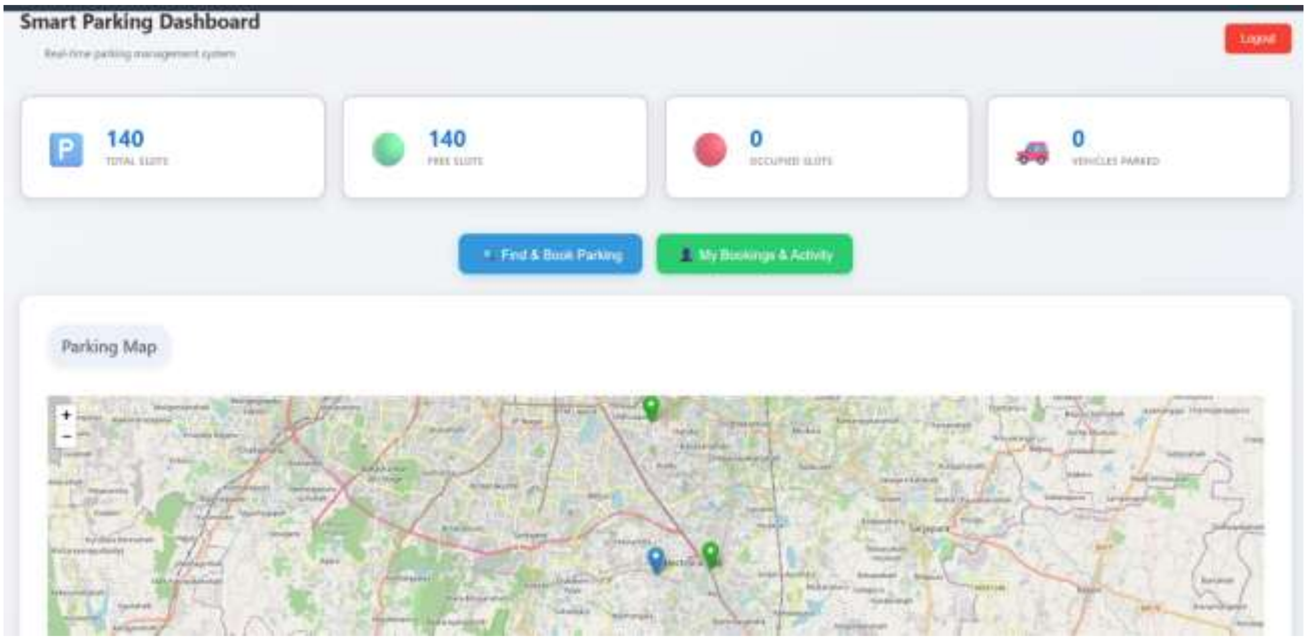
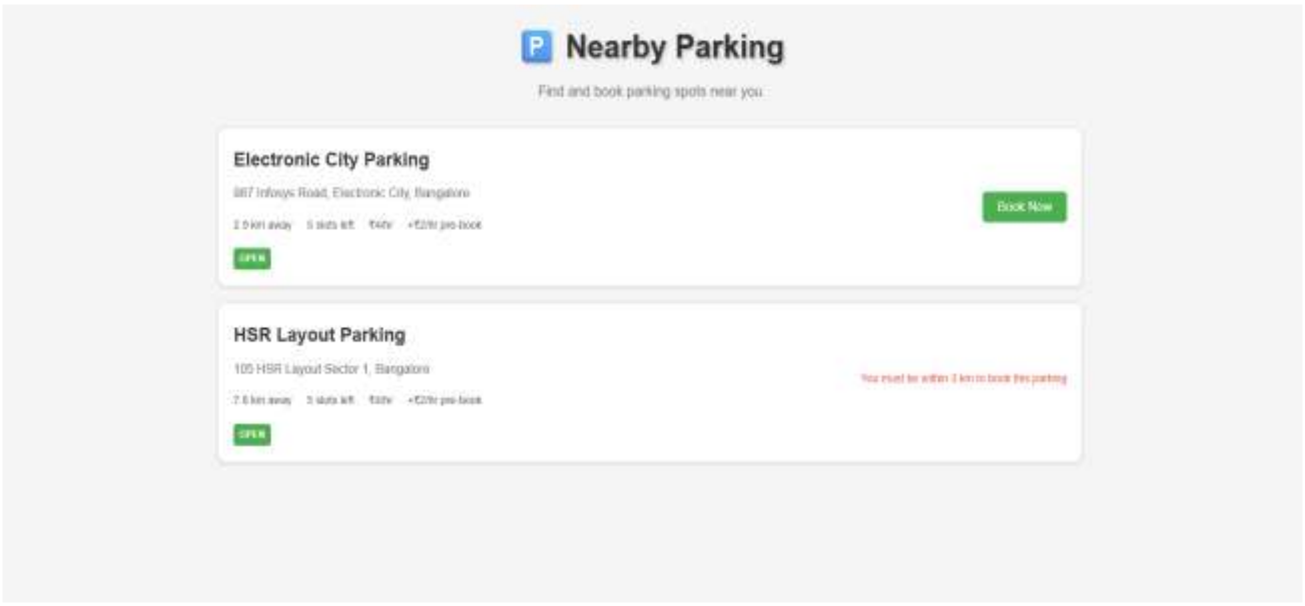


Fig.10 User Dashboard

➢ Find & Book Parking Interface:



Nearby Parking

Find and book parking spots near you

Electronic City Parking

887 Infosys Road, Electronic City, Bangalore

2.0 km away 5 slots left ₹4/hr +₹2/hr pre-book

OPEN

Book Now

HSR Layout Parking

105 HSR Layout Sector 1, Bangalore

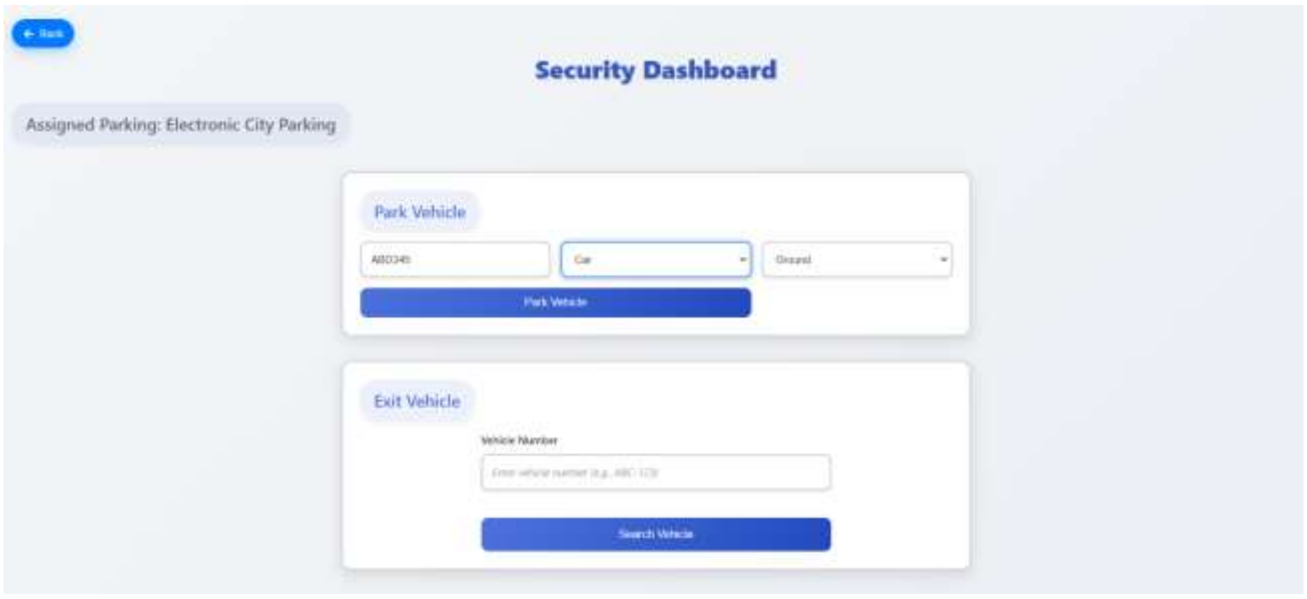
7.6 km away 5 slots left ₹3/hr +₹2/hr pre-book

OPEN

You must be within 3 km to book this parking

Fig.11 Find & Book Parking Interface

➢ Security/Parking Staff Dashboard:



Security Dashboard

Assigned Parking: Electronic City Parking

Park Vehicle

ABD345 | Car | Ground

Park Vehicle

Exit Vehicle

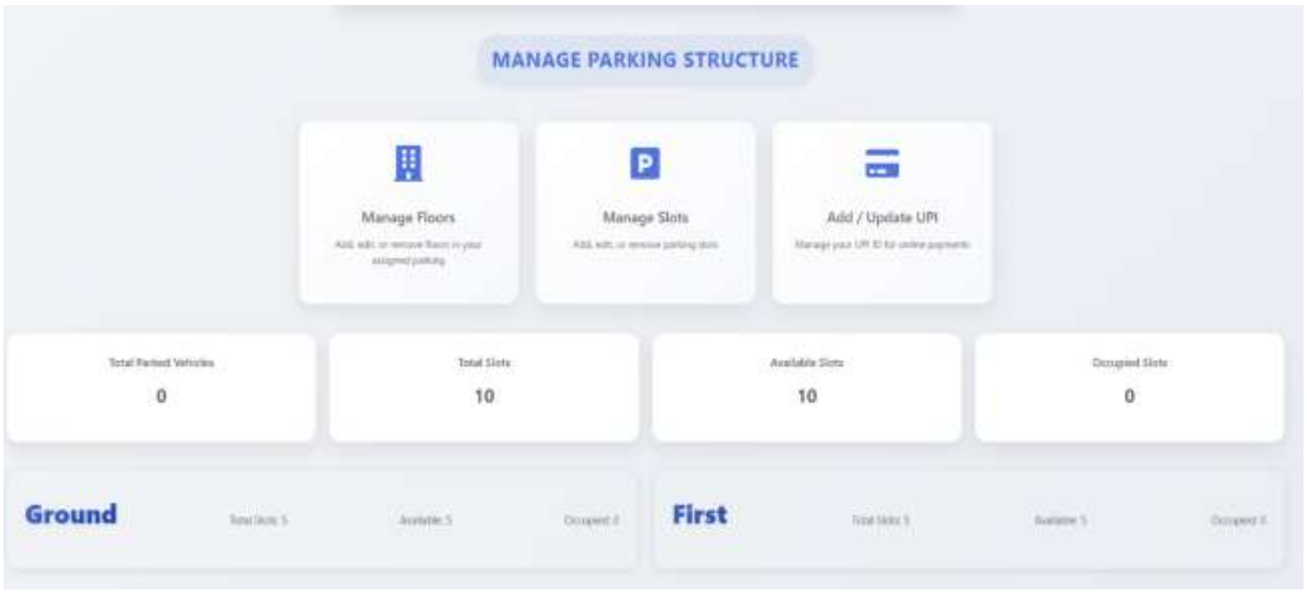
Vehicle Number

Enter vehicle number (e.g., ABC 123)

Search Vehicle

Fig.12 Security/Parking Staff Dashboard

➢ Parking Staff Management Interface:



MANAGE PARKING STRUCTURE

Manage Floors
Add, edit, or remove floors in your assigned parking

Manage Slots
Add, edit, or remove parking slots

Add / Update UPI
Manage your UPI ID for online payments

Total Parked Vehicles	Total Slots	Available Slots	Occupied Slots
0	10	10	0

Ground Total Slots: 5 Available: 5 Occupied: 0

First Total Slots: 5 Available: 5 Occupied: 0

Fig.13 Parking Staff Management Interface

➢ Slot Management Interface:

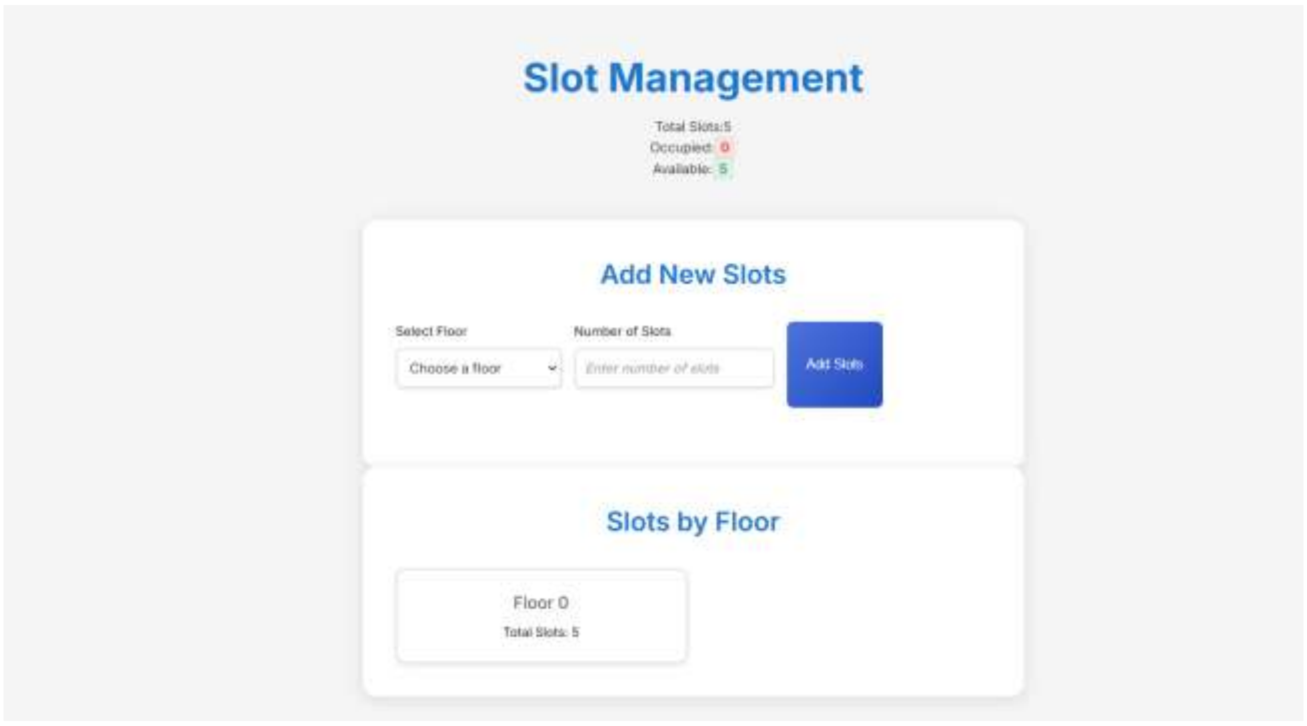


Fig.14 Slot Management Interface

CONCULSION

Car Parking Management System is a web-based application developed to solve the problems of traditional parking systems. This system focuses on automation, accuracy, and ease of use. Users can easily register, log in, add their vehicle details, and book parking slots in advance, which helps in reducing waiting time and traffic congestion. One of the main features of our project is the online payment gateway, which allows users to make secure and cashless payments. After successful payment, the system generates a unique QR code for each booking. This QR code is used at the entry and exit points for quick verification, which reduces manual checking and improves security.

From the admin side, the system provides a centralized dashboard where parking floors, slots, users, bookings, and payment details can be managed efficiently. By converting the entire parking process into a digital system, our project reduces manual work, avoids errors, and improves overall efficiency. Through this project, we have shown how technologies like QR codes, database management, and online payments can be used effectively to solve real-world parking problems in a simple and reliable way.

REFERENCES

1. A Study on Smart Parking Management System in India, IJCRT, 2023 – Discusses the need for intelligent parking solutions in urban environments and proposes a system architecture for efficient slot allocation.
2. Real-Time Car Parking Management System, IEEE Xplore, 2023 – Presents a microcontroller-based system for automating vehicle entry and exit using real-time data processing.
3. Parking problems in Abu Dhabi, UAE toward an intelligent parking management system "ADIP: Abu Dhabi Intelligent Parking"- SA Alkheder, MM Al Rajab, K Alzoubi - Alexandria Engineering Journal, 2016 – Elsevier.
4. Vehicle Parking Management System, IJNRD, 2025 – Explores software-driven parking automation with emphasis on user authentication, slot booking, and QR code-based access control.
5. Smart Parking Systems: A Survey, ScienceDirect, 2020 – Reviews various technologies used in smart parking, including IoT sensors, mobile apps, and cloud-based data management.
6. R. S. Rajesh, S. S. Kumar, and R. S. Balaji, "Smart Parking System Based on IoT," International Journal of Advanced Research in Computer Science, vol. 9, no. 5, pp. 45–50, 2018.
7. M. Al-Turjman and S. Alturjman , "Context-sensitive access in smart parking system," in Proc. Int. Conf. New Trends in Computing Sciences (ICTCS), 2017, pp. 241–246.
8. A. Khanna and R. Anand, "IoT based smart parking system," International Journal of Science and Technology Research, vol. 9, no. 2, pp. 108–113, 2020.