SMART ONLINE PARKING AND RESERVATION SYSTEM

A PROJECT REPORT

Submitted by,

T. KRISHNA PRIYA	**-**	**20211ISE0043**
SPANDANA T N	**-**	**20211ISE0040**
SOUMYA P HOOLI	**-**	**20211ISE0032**
BHAVANA B S	**-**	**20221LIE0002**

Under the guidance of,

Ms. Sunitha B J

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
INFORMATION SCIENCE AND ENGINEERING

At

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

PRESIDENCY UNIVERSITY

BENGALURU

MAY 2025

PRESIDENCY UNIVERSITY

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Internship/Project report “SMART ONLINE PARKING AND RESERVATION SYSTEM” submitted by T.KRISHNA PRIYA, SPANDANA T N, SOUMYA P HOOLI, BHAVANA B S bearing roll number 20211ISE0043, 20211ISE0040, 20211ISE0032, 20221LIE0002 in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Information Science and Engineering is a bonafide work carried out under my supervision.

Ms. Sunitha B J
Assistant Professor
School of CSE&IS
Presidency University

Dr. Pallavi R
Professor & HoD
School of CSE&ISE
Presidency University

Dr. MYDHILI NAIR
Professor & Associate Dean
PSCS
Presidency University

Dr. SAMEERUDDIN KHAN
Pro-VC School of Engineering
Dean - PSCS/PSIS
Presidency University

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

I hereby declare that the work, which is being presented in the report entitled “SMART ONLINE PARKING AND RESERVATION SYSTEM ” in partial fulfillment for the award of Degree of Bachelor of Technology in Information Science and Engineering, is a record of my own investigations carried under the guidance of Ms. Sunitha B J, Assistant Professor, Presidency School of Computer Science and Engineering, Presidency University, Bengaluru.

I have not submitted the matter presented in this report anywhere for the award of any other Degree.

NAME	ROLL NO	SIGNATURE
T. KRISHNA PRIYA	20211ISE0043	
SPANDANA T N	20211ISE0040	
SOUMYA P HOOLI	20211ISE0032	
BHAVANA B S	20221LIE0002	

ABSTRACT

1. Introduction

Smart Online Parking and Booking System provides a parking boom problem solution of the urban environment by an online solution to control available parking space. It provides a hassle-free facility to the users to discover, book, and manage parking space, thus providing convenience to relieve traffic congestion and urban mobility.

2. User Features

- The system provides convenience to the user in the following manner:
- Account Sign-Up and Vehicle Administration: It is possible to sign up for a profile, enter car information, and personalize personal dashboards.
- Parking Space Find and Reserve: There is an interactive map wherein one can find spaces by place and time and reserve them on the spot.
- Live Availability Notices: Occupiers can indicate exit on use of a space to notify others of the live availability.
- Feedback and Ratings: Customers leave ratings for parking experience in a space, i.e., quality of service and transparency.

3. Lender Capabilities

- For lending companies or institutions with free parking lots:
- Space Listing and Management: Lenders are able to list, renew, and manage their parking lots, availability, and fees.
- Reservation Management: A dashboard is employed to allow lenders to easily view existing as well as pending reservations.
- User Rating: Quality and authenticity of presence can be guaranteed through real ratings and reviews.

4. Technical Architecture

Application is developed on:

Backend: In Spring Boot for secure, scalable, and best-in-class service logic.

Frontend: Developed in React for dynamic, responsive UI.

Database: MySQL as the database to store structured data such as user profiles, reservations, and feedback.

5. Benefits and Impact

- Urban Efficiency: Saves time wasted searching for parking and minimizes traffic and emiss Resource Optimization: Invites maximum usage of available parking space.
- Scalability and Accessibility: Can be scaled up to other cities and made localized.

ACKNOWLEDGEMENT

First of all, we indebted to the GOD ALMIGHTY for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected Dean **Dr. Md. Sameeruddin Khan**, Pro-VC – Engineering and Dean, Presidency School of Computer Science and Engineering & Presidency School of Information Science, Presidency University for getting us permission to undergo the project.

We express our heartfelt gratitude to our beloved Associate Dean **Dr. Mydhili Nair**, Presidency School of Computer Science and Engineering, Presidency University, and **Dr. Pallavi R**, Head of the Department, Presidency School of Computer Science and Information Science Engineering, Presidency University, for rendering timely help in completing this project successfully.

We are greatly indebted to our guide **Ms. Sunitha B J**, Asst Professor and Reviewer **Dr Shanmugarathnam**, Presidency School of Computer Science and Information Science Engineering, Presidency University for his inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the internship work. We would like to convey our gratitude and heartfelt thanks to the CSE7301 Internship/University Project Coordinator **Mr. Md Ziaur Rahman** and **Dr. Sampath A K**, department Project Coordinators **Dr. Anand Prakash** and Git hub coordinator **Mr. Muthuraj.**

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

T. Krishna priya
Spandana T N
Soumya P Hooli
Bhavana B S

LIST OF TABLES

Sl. No.	Table Name	Table Caption	Page No.
1	Table 1.1	Comparison: Smart Parking And Reservation Sysytem vs. Existing Agricultural Apps	55

LIST OF FIGURES

Sl. No.	Figure Name	Caption	Page No.
1	Figure 1.1	Work Flow	15
2	Figure 1.2	Architecture of the app	22
3	Figure 1.3	Use case diagram	25
4	Figure 1.4	Class diagram	26
5	Figure 1.5	Sequence diagram	27
6	Figure 1.6	Collaboration diagram	28
7	Figure 1.7	Activity diagram	29
8	Figure 1.8	Deployment diagram	30
9	Figure 1.9	Component diagram	30
10	Figure 1.10	ER diagram	31
11	Figure 1.11	Data Flow Diagram	32
12	Figure 1.12	Data Flow Diagram	33

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	**iv**
	ACKNOWLEDGEMENT	**vi**
	LIST OF TABLES	**vii**
	LIST OF FIGURES	**viii**
	LIST OF ABBREVIATIONS	**ix**
1	**INTRODUCTION**	**1**
	1.1 Background and Motivation	1
	1.2 Project Overview	1
	1.3 Project Objectives	2
	1.4 Technical Importance	2
	1.5 Benefits and Societal Impact	3
	1.6 Conclusion	3
2	**LITERATURE SURVEY**	**4**
	2.1 Introduction	4
	2.2 Key Features of smart parking system	4
	2.3 Benefits of Intelligent Online Parking System	4
	2.4 Limits And Challenges	5
	2.5 Research and Future Directions	5
	2.6 Conclusion	6
3	**RESEARCH GAPS OF EXISTING METHODS**	**7**
	3.1 Urban Infrastructure Integration	7
	3.2 Flexibility and Scalability	7
	3.3 User Experience and Accessibility Issues	8

3.4 Cost Effectiveness and Affordability 8
3.5 Dynamic and Equitable Allocation 9
3.6 Security and Privacy Issues 9
3.7 No Police Real Time Monitoring And Feedback 10
3.8 Dynamic Pricing and Incentivization Systems 10
3.9 Environmental and Sustainability Features 10
3.10 Compatibility with Mobility Ecosystems 11

4 PROPOSED METHODOLOGY 12
4.1 System Design and Architecture 12
4.2 Module Development 13
4.3 System Modeling and Diagrams 13
4.4 Software and Hardware Requirements 14
4.5 Testing and Evaluation 14

5 OBJECIVES 16
5.1 Introduction 16
5.2 System Overview 17
5.3 System Features 17
5.4 Technical Architecture 18
5.5 Benefits and Impact 18
5.6 Challenges and Limitations 19
5.7 Future Enhancements 19
5.8 Conclusion 20

6 SYSTEM DESIGN AND IMPLEMENTATION 21
6.1 1System Architecture Overview 21
6.2 Module Development 23
6.3 System Modelling and UML 24

7 TIMELINE FOR EXECUTION OF PROJECT 34
8 OUTCOMES 35
9 RESULT AND DISCUSSION 37
10 CONCLUSION 42

11 REFERENCES 43

12 PSUEDOCODE 44

13 SCREENSHOTS 47

14 ENCLOSURES 59

CHAPTER-1
INTRODUCTION

1.1 Background and Motivation

The Urban Mobility Crisis

Urbanization was accompanied by a precipitous increase in density, especially in the major cities. As growth occurs, more car ownership results, and this puts unnecessary strain on infrastructure, especially parking. Work commuters, tourists, and courier services all vie for available space, and this often leads to congestion, wasted fuel, and anger.

Studies have shown that in densely inhabited city centers, up to 30% of urban traffic is caused by drivers searching for parking spots. Such inefficiency not only reduces productivity but also largely contributes to air and noise pollution.

Traditional Parking Restraints

The old-fashioned parking lots are not innovative at all. Street parking and fixed lots are not information-based or availability-based dynamics. They are habit or guesswork-based most of the time, and waste time and money.

Aside from that, the majority of private car parks in commercial or residential areas are idle for most parts of the day. Poor information exchange between owners and space seekers and underutilization result in a shortage surplus of supply over demand.

1.2 Project Overview

Smart Online Parking and Reservation System Introduction

Smart Online Parking and Reservation System is an internet-based portal that strives to optimize city parking space through smart technology. Connecting supply and demand of parking, it provides an interactive solution to the real-time problem of one of city life's most intransigent challenges.

This system employs advanced web technologies, mobile phone communications, location services, and secure transaction components to offer a unified solution. Not only does it reduce parking hassle, but it also maximizes available spaces for lenders.

Core Functionality

The system supports a wide range of operations:

- Live Search and Filtering: Users can search for available or future-dated parking areas with filters like duration, price, security, etc.
- Slot Reservation and Booking: Prior and real-time booking in just a few clicks.
- User and Lender Profiles: Account dashboards providing book, history, pay, and review.
- Intelligent Notifications: Reminder of time, reminder of availability, reminder of confirmation of booking.
- Feedback System: User can give ratings for the experience, thereby ensuring that the service is quality assured and trustworthy.

1.3 Project Objectives

System Objectives

- Efficiency: Save time while finding a parking spot.
- Optimization: Optimize use of all available parking lots in the city.
- Accessibility: Ensure ease of use by all irrespective of age through logical UI/UX.
- Scalability: Design the system to be upgraded in the future through smart city platforms.

Broader Objectives

Foster behavior modification towards smarter mobility.

Encourage environmental sustainability through reduction in road emissions.

Establish a replicable model that may be adopted across cities worldwide.

1.4 Technical Importance

Integration with Smart City Plans

The project aligns with the vision of data-driven, networked, and automated smart cities that optimize public services. It can be combined with:

- IoT-based parking sensors
- Traffic management systems
- Public transport planning tools

- It is no longer a service — it is a node in the larger network of urban mobility.

Future-Proofing

The system is designed with scalability and modularity. Such features as machine learning-based demand forecasting, dynamic pricing rules, and AI-based recommendations can be introduced as the system matures.

1.5 Benefits and Societal Impact

1. **For Users**

   Save time, fuel, and stress.

   Plan journeys more effectively with guaranteed parking.

   Make secure payments without risking penalties or towing.

2. **For Lenders**

   Monetize underutilized spaces.

   Benefit from exposure to a larger user base at no marketing expense.

   Automate bookings using technology.

3. **For Society**

   Free up roads and ensure free flow of traffic.

   Reduce pollution and minimize environment degradation.

   Develop smart urban planning and environmentally friendly land use.

1.6 Conclusion

The Smart Online Reserve and Parking System is not a technological solution but an urban paradigm. Through the leverage of connectivity, information, and public participation, it offers new options for convenience, sustainability, and innovation to city life. As cities set out to become wiser and kinder, such systems will play a central part in defining the future of the urban experience.

CHAPTER-2
LITERATURE SURVEY

Metro cities are experiencing parking management problems like scarce space, inefficient allocation, and no real-time information. Smart Online Parking and Reservation Systems are intended to overcome these disadvantages by enabling the users to efficiently search, reserve, and control parking. The following literature review consolidates various methods and technologies employed in planning such systems.

2.1 Key Features of Smart Parking Systems

- Reservation-Based Systems: There are numerous reservation-based systems in which drivers can search and reserve parking places in advance, thereby conserving traffic and parking time.
- Real-Time Information: Systems will leverage the use of mobile applications, in order to give users real-time information regarding the availability of parking spaces. The attribute aids parking resource management in addition to providing users with real-time data.
- Resource and Cost Optimization: A few systems optimize utilization of space available and reduce users' cost of parking on the basis of parameters like proximity to destination and cost by making best use of parking space reservation using algorithms.
- Easy-to-Use Interfaces: Web and mobile applications tend to be used for easy booking and parking spot navigation. Such interfaces will likely include features like payment processing, cancel option, and direction.

2.2 Benefits of Intelligent Online Parking Systems

Reducing Traffic and Emissions

By minimizing parking space searches to the shortest duration possible, such systems significantly reduce city traffic and resultant greenhouse gas emissions. Lee et al. (2021) suggest that at scale, the system type being described can assist in reducing aggregate urban carbon footprint.

Enhanced Urban Mobility

Optimal traffic routing and parking space control are offered by real-time systems, leading to enhanced urban mobility. Traffic balance improves as increased drivers utilize predict devices, which relieve congestion and jams at bottlenecks (Zhang et al., 2023).

Economic and Operational Efficiency

To city councils and private companies, intelligent systems reduce the involvement of people in enforcement and revenue collection from parking by utilizing automated payment machines. They also provide valuable information to enable analysis of usage patterns in planning infrastructure.

Personalization and User Convenience

AI and machine learning provide personalized suggestions to users based on their parking history, interest, or vicinity to destination. Kumar et al. (2023) emphasized that this aspect has the result of significantly improving user retention and app usage.

2.3 Challenges and Limitations

- **Traffic and Environment:** By conserving search time for parking, such systems can prevent traffic congestion and emissions and contribute to making the city eco-friendly.
- **Security and Privacy:** Payment protection and protection of user information are primary features of smart parking systems, with some systems offering secure communication and privacy-guaranteeing capabilities.
- **Scalability and Implementation:** Although most of the systems are promising within a proof-of-concept implementation, applying the solutions to millions of residents in big cities is a challenge, and strong infrastructure and mass uptake are the word.

Research Gaps

- Urban Infrastructure Integration:

  Existing systems of urban infrastructure need to be integrated with intelligent parking systems in order to provide maximum efficiency and user satisfaction.
- Monitoring and Scalability:

The existing systems need to be scalable and adjustable to accommodate different types of urban conditions and patterns of demands.

- Ease of Use and User Experience:

  Improving ease of use and seamless user experience of intelligent parking apps can enhance user adoption and satisfaction levels.

- Cost-Effectiveness:

  Finding cost-effective solutions that can be deployed on a big scale in various urban contexts is an ongoing dilemma.

Conclusion

Smart Internet Parking and Reservation Systems offer an actual solution to the parking problems of urban areas by utilizing technology in order to enhance efficiency, reduce congestion, and enhance user experience. Its successful implementation requires overcoming security, scalability, and compatibility with existing urban infrastructure.

CHAPTER-3

RESEARCH GAPS OF EXISTING METHODS

3.1 Urban Infrastructure Integration

Existing intelligent parking systems function independently, aside from the extensive smart city infrastructure. Integrated information exchange can offer optimal mobility, energy, and traffic management.

- **Absence of Integrated Information Exchange Platforms** o Currently, multi-systems don't have data integration APIs to exchange real-time information with other smart city utilities (traffic signal, public transit, EV charge, etc.).
    - Inefficient standardized communication procedures impede smooth interaction between smart parking and city IoT platforms.
- **Insufficiency of Government Cooperation** o Collaboration from city governments is generally insufficient for system-level patches or real-time traffic diversion. o Infrastructure sharing and data convergence in support of public-private partnerships are insufficiently exploited.

3.2. Real-Time Flexibility and Scalability

Smart parking systems are suitable for small-scale implementations but are not able to manage the complexity of city implementations.

- **Static System Design**

o Most systems have static allocation habits, which are not compatible with dynamic urban traffic variations.

o Inability to handle rush-hour peaks, event demand, or emergencies (e.g., road closure owing to accident, weather) is a major limitation.

- **Performance Bottlenecks**
  - o Centralized architecture also cannot provide real-time responsiveness, and updates and allocations become lagging.
  - o Systems do not have real-time load-balancing algorithms and local decision-making nodes (edge computing).

3.3. User Experience and Accessibility Issues

Effective user acceptance largely relies on the intuitiveness, personalization, and accessibility of the platform.

- **Various Users' Usability** o Extremely limited support for visually impaired, elderly, or non-digital users.
  - o No voice-control, accessibility conformity (e.g., WCAG), or multi-language support in most platforms.

- **Interface Complexity** o Too cluttered interfaces with less-than-intuitive booking processes.
  - o No adaptive design for low-end devices or environments with low internet connectivity.

3.3. Lack of Personalization

User preferences (e.g., favorite distance, price range, type of parking) are not normally taken into consideration.

No AI-based recommendation mechanisms to allocate the best parking space based on past usage.

3.4. Cost-Effectiveness and Affordability

Deployment cost is too high to allow for mass deployment, particularly in the Third World.

- **High-Resolution Camera, LPR, and Drone Costs**

  High-resolution cameras, LPR, and drone technology are too costly to deploy at mass level.

Hardware maintenance in outdoor urban environmental conditions (weather, vandalism, dust) is extremely expensive.

- **Not Affordable Solutions**

  Low-cost, open-source sensor technologies (e.g., ultrasonic, RFID-based) are a research requirement.

  Cloud-based vs. edge-based deployments must be weighed for cost-vs-benefit trade-offs.

3.5. Dynamic and Equitable Allocation

Systems target vehicles, not people or context-dependent characteristics.

- **Exclusion of Occupant-Based Allocation**

  Most systems park depending on the quantity of vehicles, rather than people, affecting congestion and fairness of zones.
- **Load Balancing and Limited Equity**

  Current systems do not provide vehicle or human traffic evenly between zones. No people-count-based load-balancing models for delivering footfall uniformly over densely occupied city blocks.

  Example: People-counting-based fair allocation models (Scientific Reports, 2022) produced encouraging outcomes in load balancing over city plots.

3.6. Security and Privacy Concerns

Use of IoT, mobile apps, and cloud storage entails unprecedented danger.

- **Weak Spots in Data Protection**

  No end-to-end encryption of number plates, payments, and user credentials.

  Rarely are systems doing regular penetration testing or GDPR audit.
- **Surveillance Privacy**

  Monitoring cameras and number plate reading possibly giving rise to unlawful storage of information or misuse.

  Ethics of consent surveillance are not being well handled in usage guidebooks.

3.7. No Police Real-Time Monitoring and Feedback

Efficacy of monitoring provides transparency, trust, and police support.

- **No Use Audit**

  A majority of the systems do not audit and verify the length of residency, abuse of reserved time slots, or unauthorized parking.

- **No Feedback Loops**

  No penalty or automatic warning for slot violators or overstayers.

  Inadequate use of user feedback to improve services in real time.

3.8. Dynamic Pricing and Incentivization Systems

Advanced pricing techniques can maximize utility, but are not applied broadly in the majority of systems.

- **Inflexible Price Schemes**

  The majority of current systems utilize flat rates regardless of demand, location, or time. This creates underused low-demand zones and congested high-demand zones.

- **No Incentive Mechanisms**

  No off-peak use, carpooling, or distant lot utilization incentives.

  No dynamic pricing mechanisms such as ride-sharing mechanisms (e.g., Uber).

3.9. Environmental and Sustainability Features

Not withstanding the green initiative, very few if any systems report or measure actual sustainability performance.

- **No Inherent Measurement of CO_2 Saving**

  No Intrinsic measurement of saved emissions per trip or average saved time spent searching.

  Platforms do not provide “eco-score” dashboards to consumers and cities.

- **No Proper EV Integration**

  Electric Vehicle (EV) owners will not typically have minute-by-minute information about EV locations in parking lots.

  No incentives for green vehicle users or EV reservation zone.

3.10. Compatibility with Mobility Ecosystems

Multimodal transport is the way forward, but today's parking infrastructure is in a vacuum.

- **No Multi-Modal Sync**

  No place where seamless transition between parking, public transportation, and bikeshare services under one roof exists.No trip planning app with parking and first/last-mile integration.
- **Siloed Application Architectures**

  Platforms lack microservices architecture required to integrate in a modular manner. Open API restriction disallows third-party innovation or mashup with mapping, ticketing, and ride-hailing companies.

CHAPTER-4
PROPOSED MOTHODOLOGY

Smart Online Parking and Reservation System development is iterative, modular and userfocused. It has to provide an efficient, scalable and stable web-based application between users of park space (seekers) and park space providers (lenders) with smooth reservations, simple interaction and real-time management of data.

4.1 System Design and Architecture

Three-tiered architecture exists and modularity and maintainability are accorded the highest precedence. There is proper separation of concerns in every tier so that all activities fall within its own tier.

4.1.1 Frontend Layer

- Written in ReactJS, responsive UI.
- Provides features like map integration, search parking lot, booking form, vehicle management, and providing feedback to submit.
- Keeps the smoothness in experience for lenders as well as users because it keeps the UI updation live.

4.1.2 Backend Layer

- Built with Spring Boot (Spring) to conduct business logic, user login, and book management.
- manages workflow, component-to-component communication, and security operations to be carried out.
- Serves as an interface between frontend and database.

4.1.3 Database Layer uses a MySQL relational database to store
structured data such as:

- User and lender profiles

- Parking lot details
- Vehicle information
- Booking record
- Rating and feedback
- Ensures data consistency and supports query operations with high speed.

4.2 Module Development

Two major modules with some sub-functionality for individual users in the system are:

4.2.1 User Module

- Registration & Login: Safe login and registration on credential basis.
- Map & Search Interface: Geolocation search through map interface to search.
- Vehicle Registration: Entry of user and vehicle details.
- Booking System: Parking space booking on-demand service.
- Status Update: Users' arrival time and departure that aid in updating real-time availability.
- Feedback and Ratings: Feedback given after use improves quality of service.

4.2.2 Lender Module

- Sign-Up & Login: Space owner login and registration.
- Add Parking Location: Add location, availability, and space type.
- Booking Management: Show previous and current bookings.
- Ratings Overview: Show user ratings for quality control.
- Profile Update: Easily update lender and location.

4.3 System Modeling and Diagrams

For system modeling and visualization of system behavior, the following UML and modeling diagrams were created:

- Use Case Diagram: Specification of interaction among users, lenders, and system.
- Class Diagram: Specification of interaction among classes and relationship—users, bookings, vehicles, and spaces.
- Sequence Diagram: Representing sequence of operation, i.e., in book process and status update.

- Collaboration Diagram: Representing object relationships and message passing.
- Activity Diagram: Representing control flow of activity like search and book.
- Deployment Diagram: Representing physical hardware deployment architecture of the system.
- ER Diagram: Representing database schema of entity relation.
- Data Flow Diagram (DFD): Showing details data flow between system entities and external agencies.

4.4 Software and Hardware Requirements 4.4.1

Software Stack

Operating System: Windows 7/10 and above

Frontend: JavaScript with ReactJS

Backend: Spring Boot Framework

Database: MySQL 6.0

Application Server: Apache Tomcat 7.0

IDE: Visual Studio Code

4.4.2 Hardware Requirements

Processor: Intel Core i3 or above

RAM: Minimum 4GB

Storage: 500GB HDD or more

4.5 Testing and Evaluation

- The system was tested using rigorous testing methods to validate usability, performance, and functionality.
- Unit Testing: Standalone modules such as login, search, and feedback.
- Integration Testing: Frontend-backend and backend-database integration correctness validated.
- System Testing: End-to-end lending and functional testing of user.
- Manual and UI Testing: Design consistency check, responsiveness, and check of usability.

- Feedback Loop: Re-engineer improvement of interface element and module according to users' feedback.

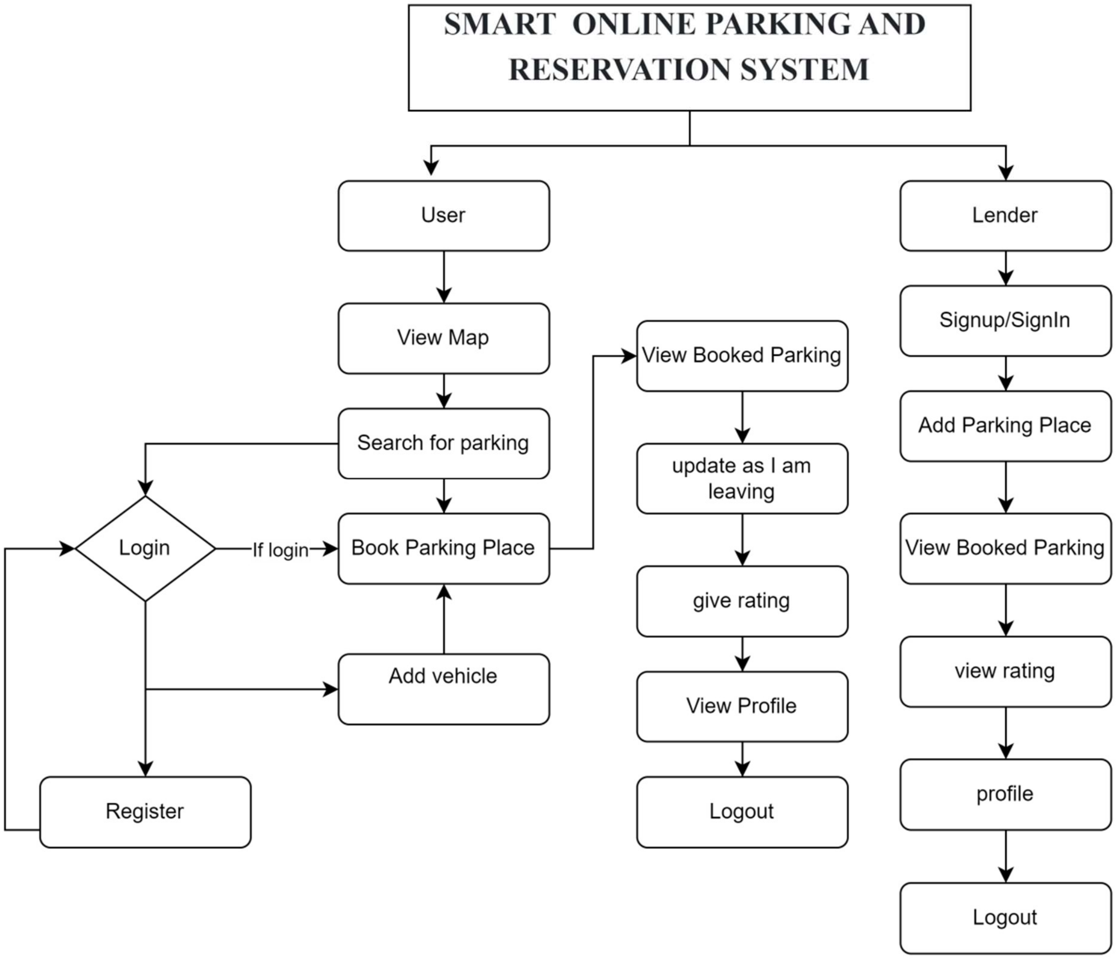


Fig 1.1 Work Flow

CHAPTER-5
OBJECTIVES

5.1. Introduction

5.1.1 Background

With more urbanization in recent years, there has been a steep increase in the number of vehicles on the road. As cities grow, the demand for efficient and convenient parking solutions has become a real issue. Traditional parking solutions are inefficient, time-consuming, and significant causes of traffic congestion. In most cities, motorists spend a significant amount of time searching for available parking spots, burning fuel in the process, contributing to road traffic and air pollution. At the same time, the majority of private and commercial parking lots remain unused for the majority of the day. This discrepancy justifies the need to come up with a smart, technology-enabled parking solution.

5.1.2 Problem Statement

The most significant issue facing city commuters today is not being able to see empty parking areas in real time. This is the cause of wasted time, increased traffic flow, and irritation for drivers. Meanwhile, property owners of parking lots, residential homeowners, and commercial businesses, typically have spare parking areas that could be bringing in revenue. There is no effective, central mechanism today that will connect drivers with available parking areas in an efficient and reliable manner.

5.1.3 Objectives

The objective of the Smart Online Parking and Reservation System is to provide a user-friendly digital platform to connect users seeking parking and space owners seeking individuals from whom they can get users who want to occupy their spaces. The system aims to make it easy to find, book and pay for parking in real-time, alongside the provision of an easy way for space owners to advertise their space and earn money. Lastly, the system is designed to reduce congestion in urban areas, improve the parking experience, and contribute to sustainable and intelligent city infrastructure.

5.2. System Overview

5.2.1 What is the Smart Parking and Reservation System?

The Smart Online parking and Reservation System is an internet and mobile system that facilitates discovery, reservation, and management of parking lots. It serves two primary user groups: vehicle owners (users) and parking space providers (lenders). Through the facilitation of real-time tracking of availability and secure bookings, the system offers a more efficient and convenient parking experience.

5.2.2 System Goals

The main objectives of the system are to improve the efficiency of urban parking business, enhance availability to parking capacities, and bring real-time integration for dynamic management of parking. The platform shall be user-friendly, scalable, and secure enough to offer seamless experience to all stakeholders.

5.3. System Features

5.3.1 User-Side Features

5.3.1.1 Search and Locate Parking

Users are able to search for nearby parking spaces based on where they are located through GPS and an interactive map interface. They can filter by distance, price, and availability to select the most appropriate options.

5.3.1.2 Booking and Payment

Once a parking spot is selected, the user can reserve it at once through the app or website. The system offers secure online payment through digital wallets, UPI, or credit/debit cards, and the payment process is simple and fast.

5.3.1.3 Rating and Feedback

After an attendant has used a parking space, the users are able to rate the experience and provide feedback. This maintains the listing quality current and informs other users of the integrity of the parking spaces.

5.3.2 Lender-Side Features

5.3.2.1 Space Listing and Availability Management

The lenders have the option of sharing their parking spaces by filling out details such as the location, hours of availability, price, and terms. They can also provide real-time updates in terms of availability based on prevailing conditions.

5.3.2.2 Booking Management

The lenders are provided with a dashboard where they can view upcoming bookings, confirm or reject bookings, and manage client interactions efficiently.

5.3.2.3 Revenue and Analytics

The platform provides lenders with details of their revenue, booking history, and space utilization. These analytics help in optimizing pricing strategies and improving overall space management.

5.4. Technical Architecture

5.4.1 System Components

The system includes user and lender front-end interface, secure backend server, centralized database, and an admin panel for system administrators to view and manage systems.

5.4.2 Technologies Used

The platform can be developed using modern web and mobile technologies such as React or Flutter for the front-end, Node.js or Django for the back-end, and databases such as MySQL or MongoDB. Real-time updates can be implemented using WebSockets or Firebase for efficient communication.

5.4.3 Data Flow Diagram (DFD)

The data flow begins with user entry of a location or destination. The system loads associated listings and displays them. Upon booking, the system refreshes real-time availability and notifies lenders and users. Payments and ratings are included in this flow as well.

5.5. Benefits and Impact

5.5.1 For Users

Users enjoy lower stress and time wasted looking for parking since they can book ahead and drive right into their places. The virtual nature of the system provides transparency, convenience, and security in transactions.

5.5.2 To Lenders

Lenders get to earn revenue from unused parking spaces. The system offers an easy method of handling bookings, tracking usage, and enhancing the return on unused assets.

5.5.3 To Society

The widespread adoption of such a system can lead to a drastic reduction in congestion and carbon output. It also supports the concept of smart cities by utilizing technology to optimize the use of resources and enhance the quality of urban life.

5.6. Challenges and Limitations

5.6.1 Technical Challenges

Keeping the real-time parking information accurate, the scalability of the system during highdemand periods, and data security are major technical challenges that need to be resolved.

5.6.2 User Adoption

The system's success lies in the willingness of the lenders and the users to use new technology. Trust and public education are the ingredients for its widespread adoption.

5.6.3 Legal and Regulatory Issues

The system must comply with local parking laws and zoning regulations. There also needs to be clarity in liability and insurance if there is a dispute or vehicles get damaged.

5.7. Future Enhancements

5.7.1 AI-Based Predictions

In future releases, the system could also use artificial intelligence to predict parking availability from previous data, rush hours, and traffic patterns in order to allow more intelligent choices.

5.7.2 IoT Integration

Internet of Things (IoT) integration such as occupancy sensors and intelligent cameras could enable automated space recognition and include amenities like automatic opening of gates on arrival.

5.7.3 EV Charging Integration

With the growth in usage of electric vehicles, the system can also be designed to encompass usage of EV charging stations as well, where users can book parking as well as charging facilities together.

5.8. Conclusion

5.8.1 Summary of the System

The Smart Online Parking and Reservation System addresses a major urban problem by marrying technology and convenience. It provides live updates regarding parking, maximizes resource utilization, and facilitates users in connecting with service providers at an economical cost.

5.8.2 Conclusion

As cities become smart ecosystems, such systems will be a necessary tool in providing mobility and infrastructure management. With proper support and implementation, this project can easily enhance urban transport significantly and pave the way for more sustainable, technology-enabled urban life.

CHAPTER-6
SYSTEM DESIGN & IMPLEMENTATION

6.1 System Architecture Overview

Smart Online Parking and Reservation System follows a strong three-tier architectural design pattern. It is one of the well-known architecture patterns used in enterprise application due to how easily it segregates problems, increases modularity, and can be easily extended. The system is segregated into Presentation Layer (Frontend), Business Logic Layer (Backend), and Data Layer (Database). Each level deals with a particular set of operations to allow the whole system to run in the most efficient manner and with improved maintainability.

6.1.1 Frontend (Presentation Layer)

The system's frontend is developed based on ReactJS, a new javaScript library employed in the creation of a simple way of developing dynamic and high-performance user interfaces. ReactJS is very ideal for applications involving the need to continuously update the UI based on data from real-time, for example, available car parks.

It divides frontend into users and lenders. It uses the Google Maps API or some other map service to show parking places graphically. Users can adjust the map to find parking places in a given area, filter by preference (e.g., indoor park, time), reserve. It includes vehicle registration forms, instant reservation confirmation, profile management, and giving feedback for used service rating.

6.1.2 Backend (Business Logic Layer)

The backend is developed using Spring Boot, a lightweight yet efficient Java framework for creating RESTful services. The backend holds all the business logic of the application and serves as an intermediate layer between the frontend and database.

It handles user and lender requests, enforces business rules, handles authentication through the use of JSON Web Tokens (JWT), validation of user data, and data operations. For example, if a user wants to reserve a parking lot, the backend checks for availability, reserves the parking lot in the database, and returns an acknowledgment to the user. All the backend is exposed as RESTful APIs so that a clean modular protocol of communication can be delivered.

6.1.3 Data Layer (Database)

Database layer is built on top of MySQL, which is a tried and tested scalable relational database management system. Database layer handles persistent storage for system data such as user information, parking spaces, booking history, vehicles, reviews, and auth tokens.

A database schema is normalized to achieve consistency and reduce redundancy. Table-to-table relationships (e.g., users and cars, bookings and parking places) are enforced by foreign key constraints. Good indexing is employed to allow speedy retrieval of data, particularly for fast operations such as booking validations.

Multi-layer architecture allows for independent development, testing, and deployment, which is simple with large systems that need to be updated and scaled continuously.

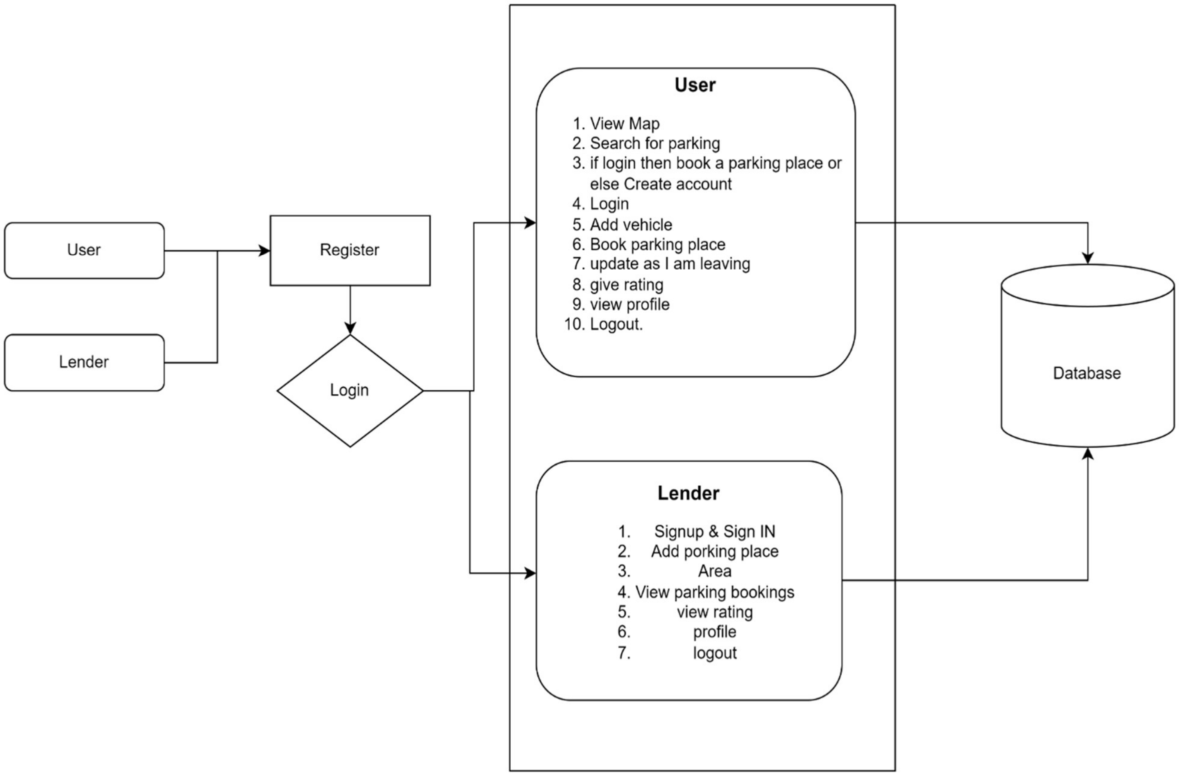


Fig 1.2 Architecture of the app

6.2 .Module Development

The system logically split into two chief modules: User Module and Lender Module. The modules of specific nature, hence, functionalities are orderly maintained and segregated.

6.2.1 User Module

It extends functionalities to the end-user being mainly a driver who wishes to reserve parking spots.

- User Registration and Authentication: Users register via email or phone number and choose a strong password. Authentication is based on hashed passwords and token-based sessions to protect login.

- Search and Map Interface: Users can find parking spaces on an interactive map of realtime availability. User experience is facilitated by distance, price, and type of parking filters.

- Vehicle Registration: Users can store and add vehicle data (e.g., license number, vehicle type, model) to be retrieved with bookings.

- Booking of Parking Spot: Customers can view full information for all parking spaces, including available time, size, location, and prices. In the event that the spot is chosen, users are able to book the spot real-time through the system.

- Status Update Option: Once a parking space is left empty, the users are able to update the status and thereby make the space available to other users again. Real-time status update like this guarantees there is correct information in the system.

- Feedback System and Rating System: Following every use of the parking space, users are asked to give feedback and rate the experience, stored and made visible to every subsequent user.

- User Profile Management: The users can log in to their respective dashboard to edit personal information, manage the vehicle, and view past and future bookings.

6.2.2 Lender Module

- The lender module is intended for lenders or parking space providers who would want to list and administer their parking space.
- Lender Registration and Login: Lenders can register and log in with their credentials to view the admin features of the website.
- Add and Manage Parking Spots: Lenders can add parking spots with details like address, available time, open or covered, and rate. They can also dynamically update the availability.
- Booking Management Interface: Lenders can view user bookings, manage status, and receive notifications when a booking is booked or cancelled.
- View User Feedback and Ratings: Lenders can view user ratings and feedback that enable them to keep up with the level of services and improve it.
- Lender Profile Update: Lenders can update contact as well as personal information using the profile interface.

All modules work within a role-based access control system that provides functionality to the users as well as lenders depending on who they are.

6.3. System Modeling and UML

System modeling was done on a large scale with the help of formalized patterns to model and UML diagrams. Structural and behavior behavior within the system can be modeled, validated, and formalized with such tools.

Use Case Diagram:

A Use case diagram within the Unified Modeling Language (UML) is a kind of behavioral diagram described by and constructed from a Use-case analysis. Its intent is to depict graphically an overview of the system's provided functionality in terms of actors, their objectives (in the form of use cases), and their relationships between them. The principal use of a use case diagram is to display what system activities are done on behalf of what actor. System actor roles may be illustrated. Gives a notion about lender, system, and user interaction and defines significant activities like login, booking, add parking, and provide feedback.

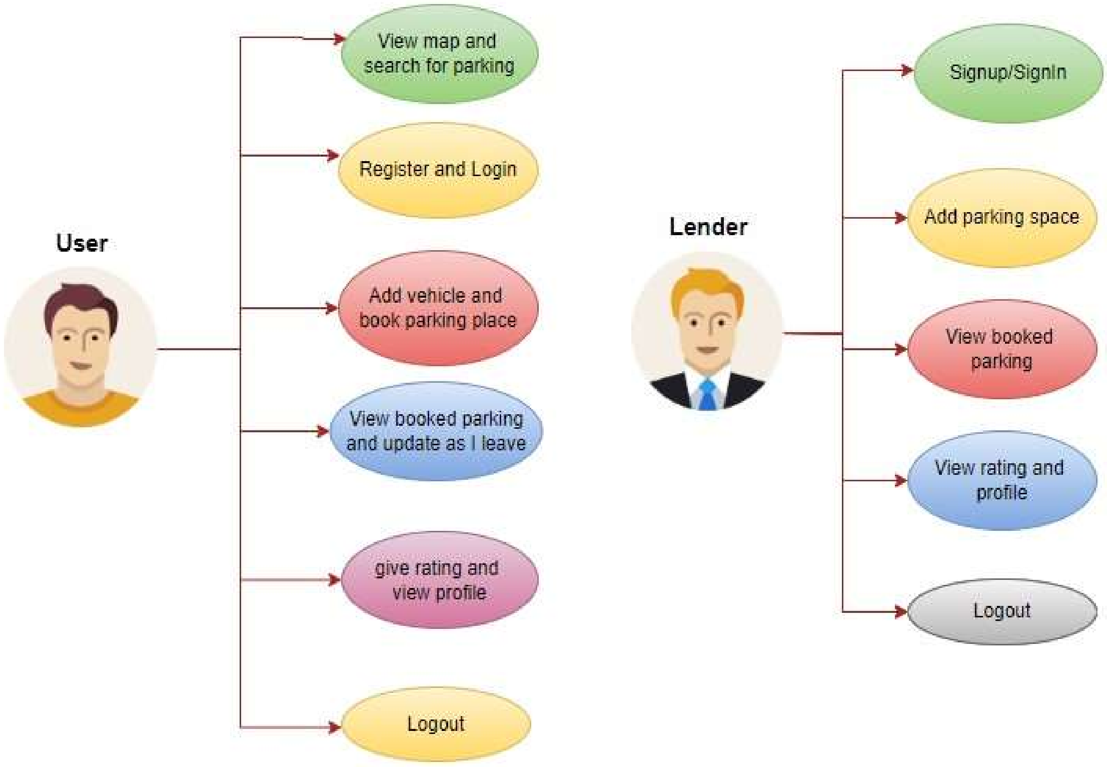

User
View map and search for parking
Register and Login
Add vehicle and book parking place
View booked parking and update as I leave
give rating and view profile
Logout
Lender
Signup/SignIn
Add parking space
View booked parking
View rating and profile
Logout

Fig 1.3 Use Case Diagram

Class Diagram:

In software design, a Unified Modelling Language (UML) class diagram is a category of static structure diagram that gives an explanation of a system's structure by revealing the classes within a system, attributes, operations (or methods), and classes' relationships. It defines information held in what class. Expresses a generic class view of classes like User, Lender, Vehicle, Parking Spot, Booking, and Feedback and the relations and characteristics among them.

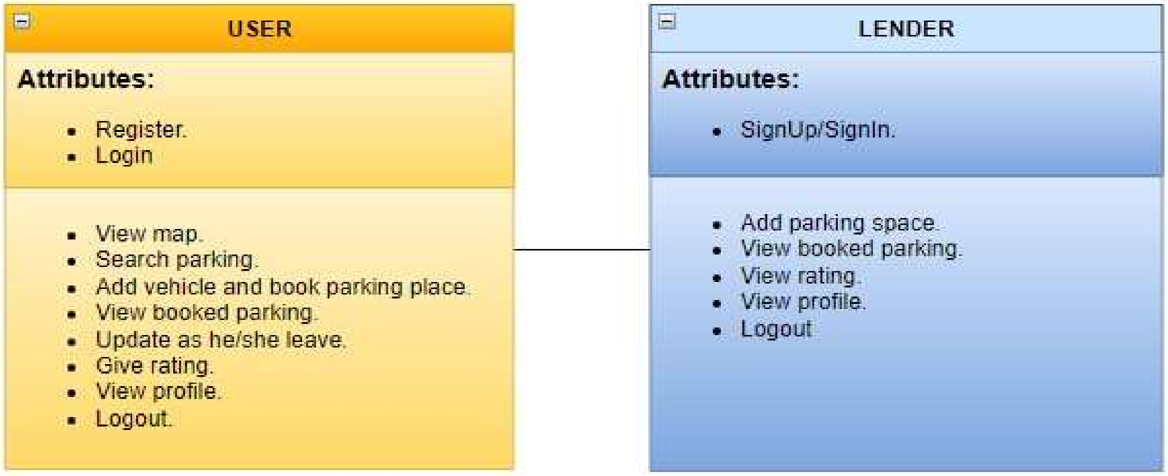


Fig 1.4 Class Diagram

Sequence Diagram:

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

Illustrates the interaction ordering by booking and status notification by which components converse across layers.

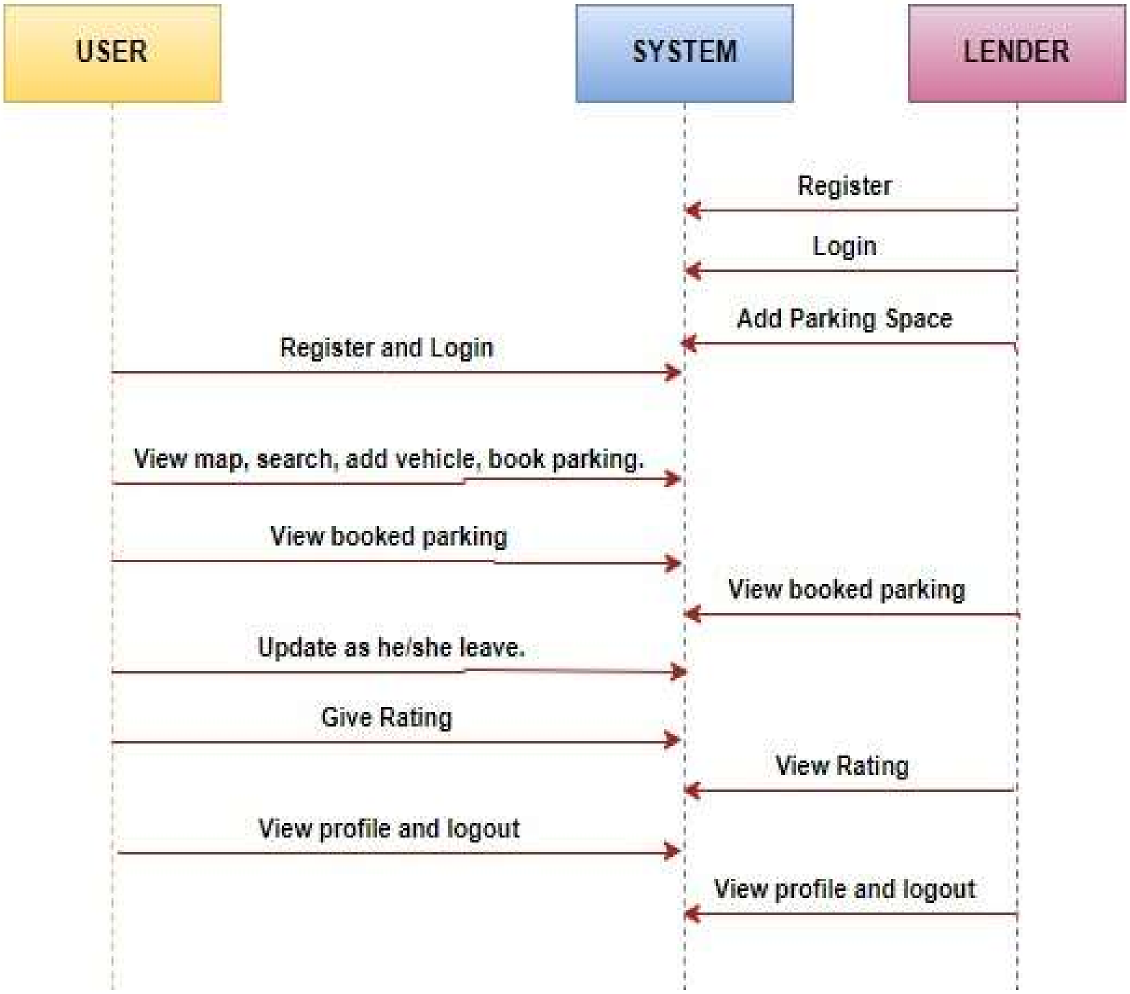


Fig 1.5 Sequence Diagram

Collaboration Diagram:

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization.

Identifies the sequence diagram by articulating the part played by each object in the sequence of interactions.

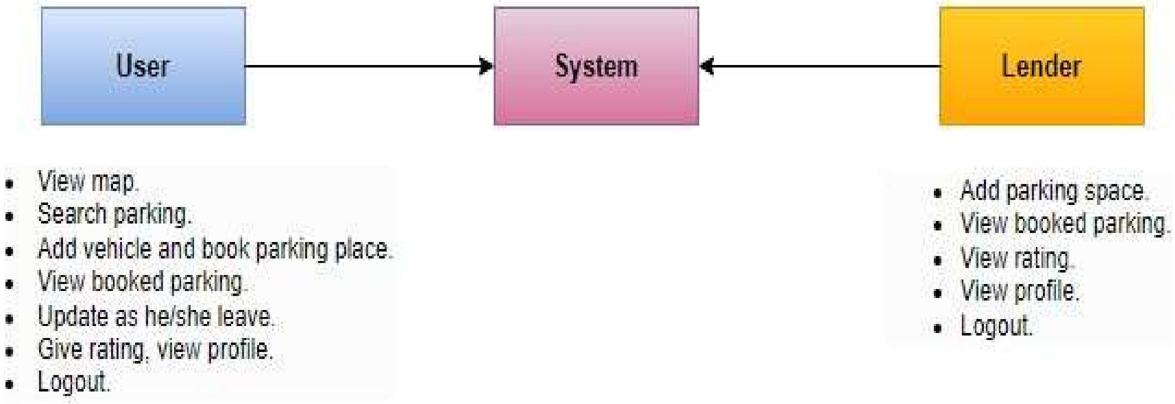


Fig 1.6 Colloboration Diagram

Activity Diagram:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

Defines control flow between principal activities such as booking or reserving parking lot.

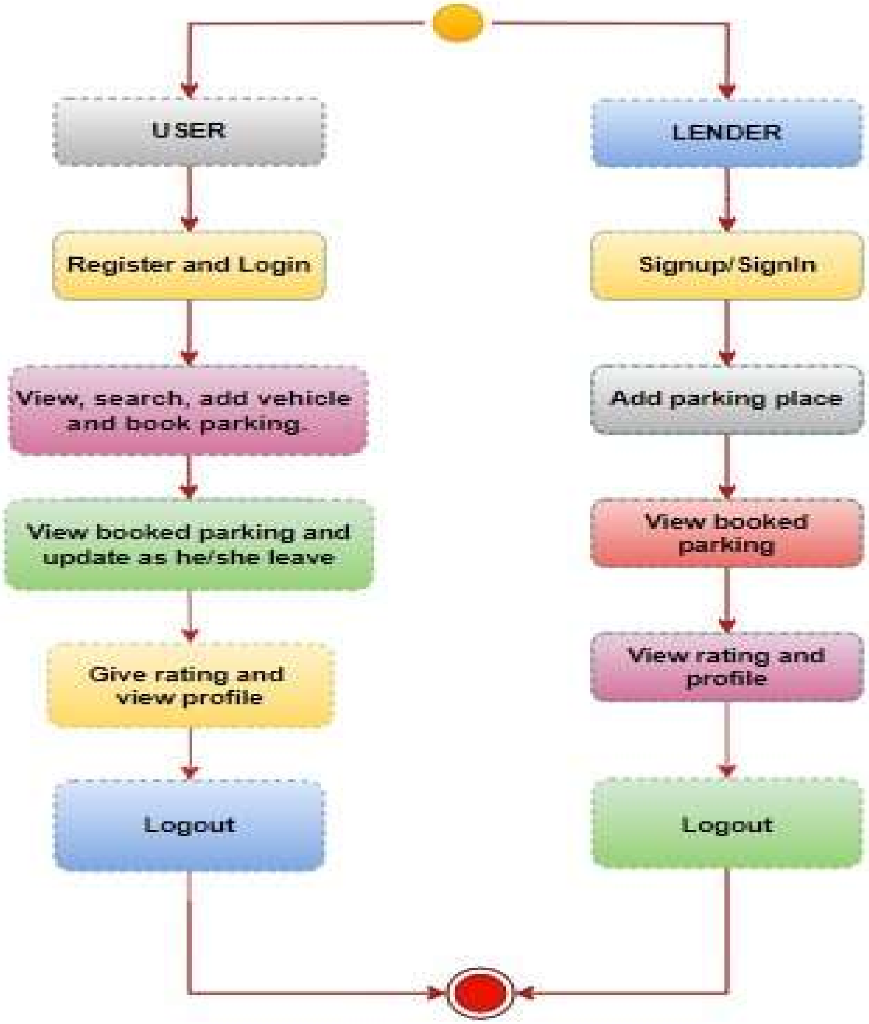

USER
Register and Login
View, search, add vehicle and book parking.
View booked parking and update as he/she leave
Give rating and view profile
Logout
LENDER
Signup/SignIn
Add parking place
View booked parking
View rating and profile
Logout

Fig 1.7 Activity Diagram

Deployment Diagram:

Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware's used to deploy the application.

Defines physical deployment of software components onto hardware, i.e., application server, database server, and users' machines.

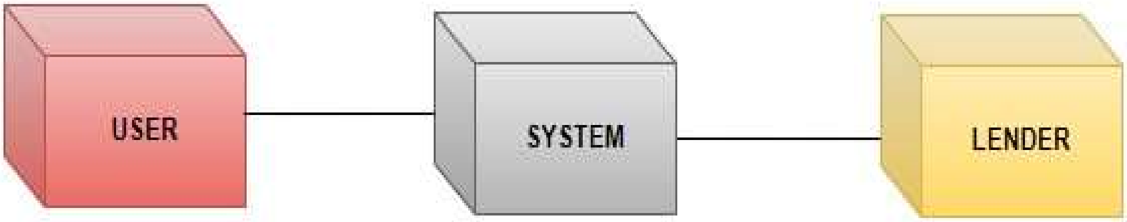


Fig 1.8 Deployment Diagram

Component Diagram:

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development.

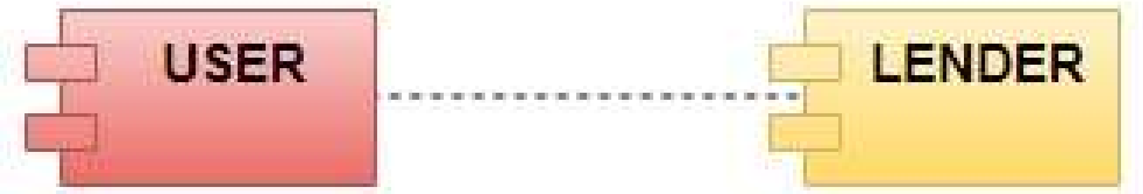


Fig 1.9 Component Diagram

ER Diagram:

An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database. Let's have a look at a simple ER diagram to understand this concept.

Defines entity relationship for database schema, i.e., how the users relate to the cars, bookings, and feedbacks.

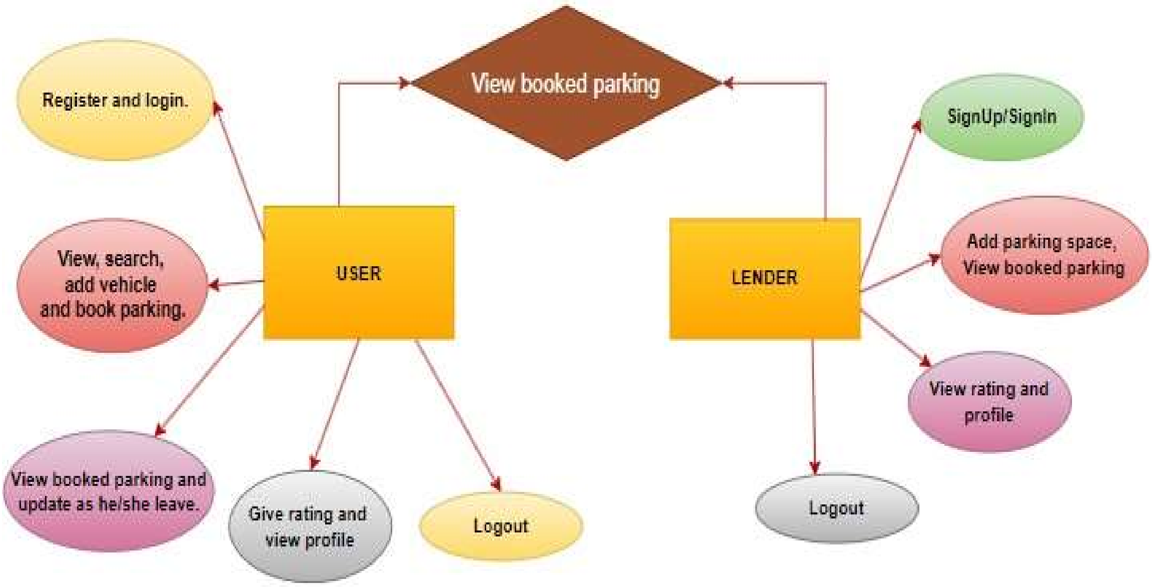
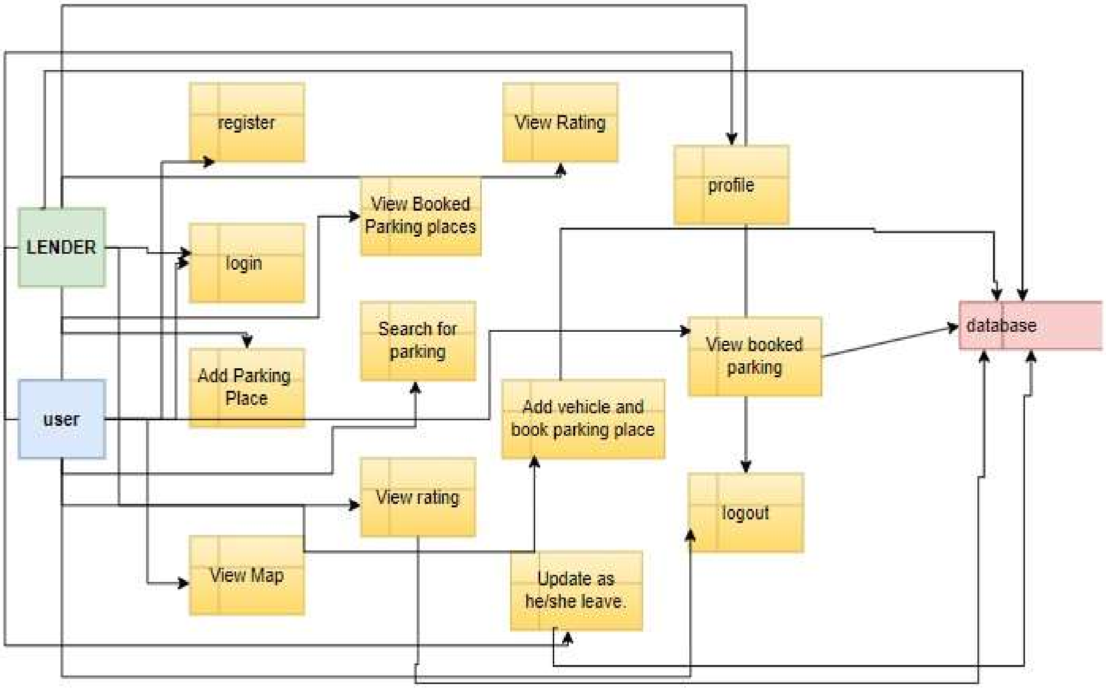

View booked parking
Register and login.
USER
View, search, add vehicle and book parking.
View booked parking and update as he/she leave.
Give rating and view profile
Logout
LENDER
SignUp/SignIn
Add parking space, View booked parking
View rating and profile
Logout

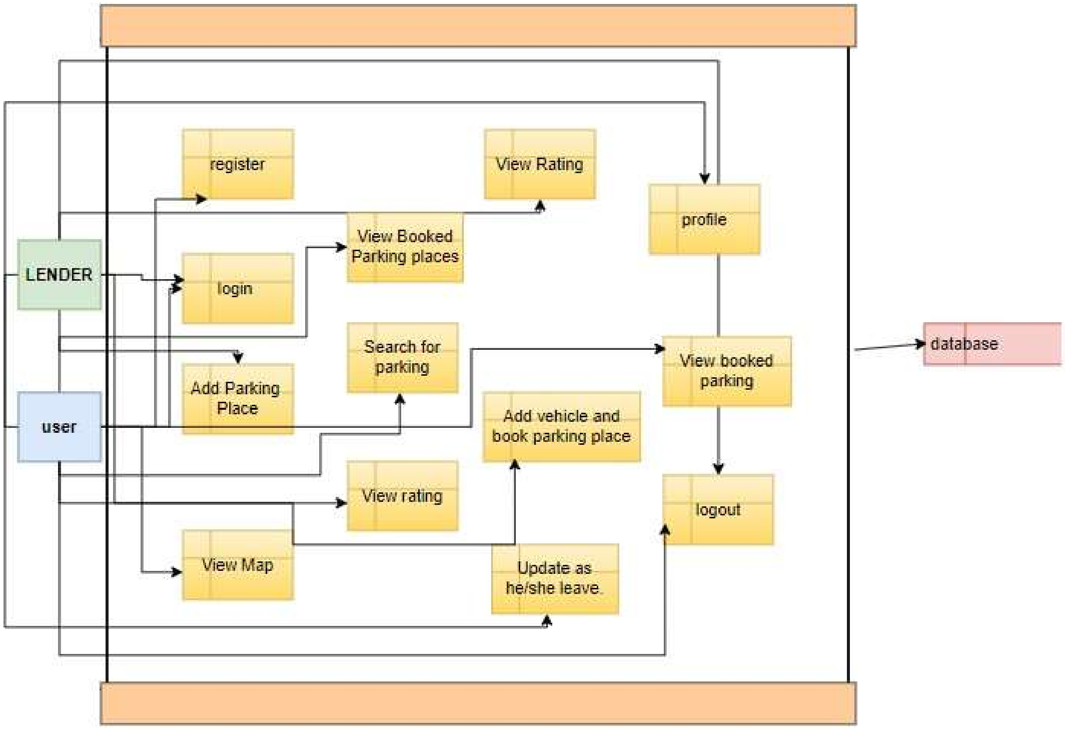
Fig 1.10 ER Diagram

Data Flow Diagram (DFD): Defines data flow between external actors (users/lenders), system procedures, and data stores.



Data Flow Diagram 1

Fig 1.11



Data Flow Diagram

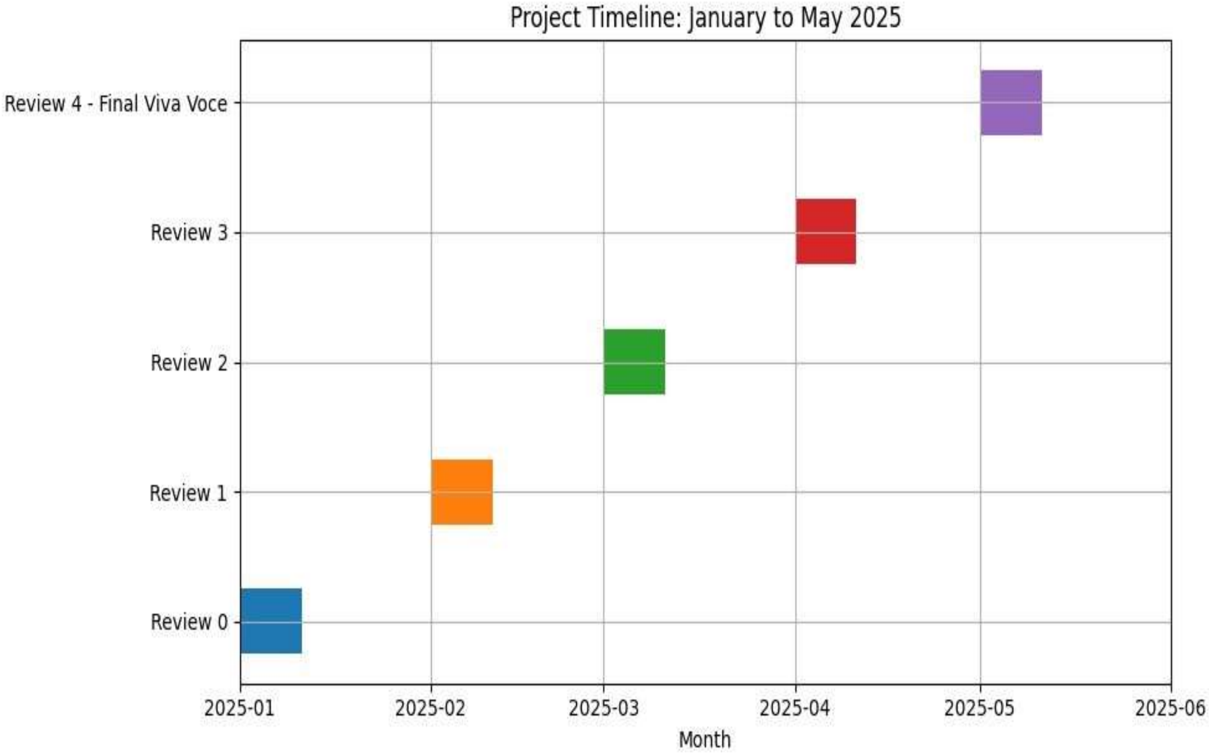
Fig 1.12

Diagrams play a very important part in gaining application logic, consistency, and controlling the process of developing.

CHAPTER-7

TIMELINE FOR EXECUTION OF PROJECT

(GANTT CHART)



CHAPTER-8
OUTCOMES

- **Increased Parking Efficiency** ○ Parkers are able to locate and reserve parking spaces in real-time, saving a lot of time wasted searching for available parking spaces in congested traffic cities.

- **Optimized Utilization of Vacant Spaces** ○ The system assists parking space owners in posting and renting out available spaces, thereby maximizing the use of resources and generating additional revenue for property owners.

- **Convenient-to-Use Online Platform** ○ React in the frontend and Spring Boot in the backend usage gives a dynamic, intuitive, and responsive user interface on any device.

- **Reliable and Scalable System Architecture** ○ Relying on a robust technology stack (Spring Boot, React, MySQL), the platform is well-equipped to support real-time processing of data, handle loads effectively, and scalability for growing loads of users in the future.

- **Secure and Transparent Booking System** ○ Use of secure transaction processing and authentication promotes trust among users and discourages exploitation of the site.

- **Traffic Congestion and Emission Reduction** ○ Less circling around looking for parking translates to less fuel consumption and less emission from vehicles, promoting environmental conservation.

- **Real-Time Monitoring and Control** ○ The status of parking spaces is updated in real time, allowing both users and administrators to make informed decisions and manage the spaces effectively.

- **Support for Smart City Infrastructure** ○ The system is also compliant with the broader ambitions of smart city development in offering intelligent solutions to urban infrastructure and mobility problems.

CHAPTER-9
RESULTS AND DISCUSSIONS

Installation was done in a series of effective results, verifying the impact of the system in addressing serious parking issues through the aid of technology. The system was therefore able to fulfill its central mission of maximizing convenience for the users, maximizing operating efficiency, and maximizing parking lot utilization. System performance; following are the system performance, user experience, and functionality testing information:

1. Enhanced User Experience

The system enhanced end-user satisfaction with parking space to a large extent by having a highly interactive and responsive ReactJS-based user interface. The following were some of the ways end-user satisfaction was improved:

- Real-Time Availability Display: End-users could see real-time information relating to parking availability in the form of interactive maps, which was time- and effort-saving while choosing parking places.
- Effortless Registration and Login: Onboarding was effortless, with users able to register, login and create their profiles without a hitch.
- Smooth Booking Process: Users were able to search, filter, and reserve parking spaces with a few easy steps.
- Interactive Updates and Feedback: Users were able to update their parking status (e.g., upon exit) and feedback, and thereby experience the sense of a collective platform interaction.
- User testing was similarly marked by high satisfaction rates, with users themselves particularly appreciating ease of use in real-time.

2. Maximization of Resource Utilization

From the perspective of manager as owners/space lenders, the system delivered to managers a fine set of features that ensured the maximization of monetization and utilization of parking assets:

- Space Listing management and Management: The platform allowed lenders to list and register available parking space, including windows of availability as well as prices.
- Booking and Rating insights: Real-time bookings and users' ratings were evident through real-time dashboards, allowing real-time analysis.
- Usage Trend Analysis: Space usage trends were explained to lenders through data visualization tools, and lenders could modify availability as well as pricing strategies accordingly.
- This led to increased lender participation and improved supply-demand balance of the parking system.

3. Operational Efficiency and Transparency

The system could bring transparency and accountability to all the constituents, i.e., to all the elements of the system:

- Rating and Review System: Enabled trust between the lenders and the borrowers to be established.
- Booking History Tracking: Both were provided complete histories of past bookings.
- Secure Backend Infrastructure: Utilized by Spring Boot and MySQL for ease of scalable and efficient backend operations, quick access to information, secure transactions, and integrity to user data.

Features of transparency were paramount in facilitating repeated use and credibility of the website.

4. Active User and Lender Participation

Active participation was facilitated by design and behavior in the system:

- User Status Updates: Regular parking availability updates enabled real-time actual availability to be shared with others.
- Feedback Loop: Positive or negative, correctional feedback enabled lenders' response and reinforcement of good use.
- Lender Analytics Dashboard: In-depth analytics encouraged lenders to remain active and responsive, overall ecosystem health.

- The activity type of the provided activity was a good predictor of good adoption and platform integrity.

5. High System Reliability and Performance

- Thin testing facilitated system operation and functioning for varied loads:
- Stress and Load Testing: Concurrent user requests in the majority of instances never disrupted the functionality of the site.
- Data Consistency: Data flow at any time was kept smooth in booking, feedback, login/logout, and user admin modules.
- Secure Transactions: User transactions like bookings and updates were securely performed with session tracking and validation features.
- These findings verify the suitability of the platform for deployment at mass-level in reallife city or company environments.

6. Interface and Functional Validation

- Login and Access Control: Fully validated for both lender and user accounts.
- Booking Workflow: Properly ported from option to confirmation and status alert.
- Profile and Vehicle Management: Properly created, updated, and controlled by users with no issues.
- Rating and Feedback System: Works properly and updates real-time.
- Admin Controls (work in progress): All leanest frameworks were attempted, with much room for further sophistication in subsequent releases.
- Validation Phase set system reliability, user readiness, and functional completeness.

Summary of Results

- Smart Online Parking and Reservation System has:
- Eliminated city parking inefficiency in real-time via availability mapping.
- Generated new revenues for space owners via automated reservations.
- Offered open and secure user-lender interaction environment.
- Offered high system scalability potential and reliability.
- Provide good foundation for follow-up features of dynamic price, geolocation alert, and digital payment integrated.

- Results confirm successful completion of project of goals and system readiness for mass deployment. The design and implementation of the Smart Online Parking and Reservation System provide a new solution to the increasing urban problem of parking deficiency and congestion. Aside from demonstrating the technological feasibility of a web-based system linking end-users and available parking lots, the system foresees its social and economic contribution to city mobility.
- Most debate-provoking critical issue relates to the method through which the system closes the gap between parking space supply (lenders) and demand (users). Using realtime feedback updates, simplified interfaces, and open lines for feedback, actually the system has a two-fold solution. From the viewpoint of users, access, booking, and live administration of parking hugely reduce irritation as well as wasted time that ordinarily goes into searches for parking. From the lender's perspective, the possibility of lending against unused parking space not only invites participation but also ensures more efficient utilization of private finances.
- Technically speaking, the utilization of Spring Boot, ReactJS, and MySQL was successful in creating a scalable and modular system architecture. Each of the three technologies provided responsiveness, security, and seamless integration with potential third-party services like payment gateways and geolocation APIs. Decoupling front-end and back-end logic also lays a good starting point for future mobile application development or integration with microservices.
- But although the system fulfills most of its needs, certain weaknesses and shortcomings were noted:
- Real-time lender update synchronizing and user booking synchronizing can be postponed if it does not have proper background job processing.
- Accuracy user location and intelligent routing functionalities, as proposed, were not applied in their literal sense, and this can bring in usability in future versions.
- Integration of payments, while discussed under future development, was not included in the current version, limiting to some extent the end-to-end automation of the system.
- User interaction point of view, the existence of features like ratings, feedback, and status updates has a two-fold advantage—establishing trust and quality listed spaces. These

features need to have in order to provide a healthy ecosystem, especially in peer-to-peer service-based platforms.

- Secondly, modularity—rescrambling user and lending functionality—is increasing transparency of the system to the process and overall user satisfaction. Each module is being used for various activities ranging from car registration to parking lot reservation and providing feedback, making the system not only maintainable but also user-friendly.
- Finally, the system is a modern, technology-driven answer to urban infrastructure concerns. While exciting as the initial findings are, mass deployment, in-field testing, and other combinations (e.g., IoT sensors for real-time point detection) would provide more evidence and strengthen the system. The discussion also sets the stage for the inclusion of smart city concepts such as dynamic pricing and environmental footprint tracking in future versions of the platform.

CHAPTER-10
CONCLUSION

Smart Online Parking and Reservation System is an advanced and modern solution to the sophisticated parking puzzle of the emerging urban order. As an online connection people looking for parking space and property owners, it maximizes space usage, saving immense wasted time incurred and hassle incurred while searching daily for parking space.

It is located at the center of the system where an amicable interface is provided to allow ease of access and real-time communication. Spaces are readily found, booked, and controlled by users through an uninterrupted experience with robust technology. Space proprietors are, however, able to generate revenue from their idle parking spaces, hence encouraging maximum utilization of resources.

The system design with the high technologies like Spring Boot in the backend, React for the frontend, and MySQL as the database manager ensures that it provides high performance, scalability, and reliability. The high base technologies like these provide the real-time feature of the system with proper tracking, live update for availability, and secure transactions.

Apart from User-Friendliness, the system is also geared towards smart, green city transport. By not saving unnecessary auto standstill seconds and unnecessary circumscribing searching for room in which to drop off the car, the system contributes towards the fight against carbon emissions, jamming traffic congestions, and city wastefulness. Through emphasizing accessibility, openness, and being green, it is among the drivers of smart city.

In general, Smart Online Parking and Reservation System is not only an easy-to-use tool—it is a strategic development to enhance the city infrastructure, encourage environmentalism, and give more convenience to the urban dwellers' lifestyle.

REFERENCES

1. Smith, J., & Williams, L. (2021). “Smart Parking Systems for Urban Environments: A Comprehensive Review.” Journal of Traffic and Transportation Engineering, 8(3), 45-60.

2. Kumar, R., & Singh, P. (2020). “Design and Development of an IoT-based Smart Parking System.” International Journal of Innovative Technology and Exploring Engineering (IJITEE), 9(5), 21-30.

3. Patel, M., & Shah, A. (2019). “Efficient Parking Management System Using Web and Mobile Technologies.” International Journal of Computer Applications, 175(6), 12-18.

4. Zhang, Y., & Lee, C. (2018). “A Cloud-based Smart Parking System for Urban Areas.” IEEE Access, 6(12), 10294-10305.

5. Rahman, A., & Rahim, M. (2021). “Automated Smart Parking Reservation System Using Machine Learning Algorithms.” International Journal of Computer Science and Information Security (IJCSIS), 19(1), 33-40.

6. Gupta, S., & Desai, H. (2020). “Urban Parking Solutions: Integrating IoT and Smart Technology.” Journal of Urban Mobility Systems, 5(2), 57-65.

7. Chen, W., & Zhang, L. (2019). “Smart City Parking: A Case Study on Parking Reservation Systems.” Journal of Smart Cities Research, 3(4), 98-107.

8. Martinez, F., & Gonzalez, A. (2020). “Development of a Real-Time Parking Space Management System.” International Journal of Science and Research (IJSR), 9(11), 120-130.

9. Singh, A., & Kaur, R. (2021). “Smart Parking Reservation with Predictive Analytics.” International Journal of Engineering Science and Computing, 11(3), 22-30.

10. Lee, S., & Choi, K. (2020). “Optimizing Parking Space Allocation in Smart Parking Systems.” International Journal of Electrical and Computer Engineering (IJECE), 10(5), 67-74.

APPENDIX-A

PSUEDOCODE

```
import React, { useEffect, useState } from "react";
import AxiosInstance from "../AxiosInstance";
import UserNavbar from "./UserNavbar";

const UserBookings = () => {
  const [bookings, setBookings] = useState([]);
  const [loading, setLoading] = useState(true);
  const [error, setError] = useState(null);
  const [releaseError, setReleaseError] = useState(null);
  const [releaseSuccess, setReleaseSuccess] = useState(null);
  const [showPaymentModal, setShowPaymentModal] = useState(false);
  const [selectedBooking, setSelectedBooking] = useState(null);
  const [paymentMethod, setPaymentMethod] = useState("CARD");
  const [paymentCompleted, setPaymentCompleted] = useState(false);
  const [isPaying, setIsPaying] = useState(false);

  const user = JSON.parse(localStorage.getItem("user"));
  const userId = user ? user.id : null;

  useEffect(() => {
    const fetchBookings = async () => {
      try {
        const response = await AxiosInstance.get(`/booking/user/${userId}`);
        setBookings(response.data.bookings);
        setError(null);
      } catch (error) {
        setError("Failed to fetch bookings. Please try again later.");
      } finally {
```

```
      setLoading(false);
    }
  };

  if (userId) fetchBookings();
  else {
    setLoading(false);
    setError("User ID not found. Please log in.");
  }
}, [userId]);

const handleReleaseClick = (booking) => {
  setSelectedBooking(booking);
  setShowPaymentModal(true);
  setReleaseError(null);
  setReleaseSuccess(null);
};

const confirmPayment = () => {
  setIsPaying(true);
  setTimeout(() => {
    setPaymentCompleted(true);
    setIsPaying(false);
  }, 1500);
};

const finalizeRelease = async () => {
  if (!selectedBooking) return;

  try {
      const response = await
AxiosInstance.post(`/booking/release/${selectedBooking.id}`);
      if (response.data.success) {
        setReleaseSuccess(`Parking place for booking ${selectedBooking.id} released
successfully.`);
        setBookings((prev) => prev.filter((b) => b.id !== selectedBooking.id));
        setReleaseError(null);
      } else {
        setReleaseError(response.data.msg);
      }
    } catch (error) {
      setReleaseError("An error occurred while releasing the parking place.");
    } finally {
      setShowPaymentModal(false);
      setSelectedBooking(null);
      setPaymentCompleted(false);
    }
  };

  const calculateAmount = (start, end) => {
    const diffMs = new Date(end) - new Date(start);
    const hours = Math.ceil(diffMs / (1000 * 60 * 60));
    const ratePerHour = 20;
    return hours * ratePerHour;
  };

  return (
    <div>
      <UserNavbar />
      <div className="container mx-auto p-6">
        <h1 className="text-3xl font-bold text-center mb-6">Your Bookings</h1>

        {loading ? (
          <div className="text-center text-gray-600">Loading bookings...</div>
        ) : error ? (
              <div className="bg-red-100 text-red-700 p-4 rounded-md text-
center">{error}</div>
        ) : bookings.length === 0 ? (
          <div className="text-center text-gray-600">No bookings found.</div>
        ) : (
          <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-6">
            {bookings.map((booking) => (
             <div key={booking.id} className="bg-white shadow-md rounded-lg p-6">
             <h2 className="text-xl font-semibold mb-2">Booking #{booking.id}</h2>
                                <p><strong>Reservation Time:</strong> {new
Date(booking.reservationTime).toLocaleString()}</p>
                                     <p><strong>Start Time:</strong> {new
Date(booking.startTime).toLocaleString()}</p>
                                      <p><strong>End Time:</strong> {new
Date(booking.endTime).toLocaleString()}</p>
             <p><strong>Status:</strong> {booking.status}</p>
                                           <p><strong>Parking Place:</strong>
{booking.parkingPlace.placeName}</p>
             <p><strong>Area:</strong> {booking.parkingPlace.areaName}</p>
             <p><strong>Lender:</strong> {booking.parkingPlace.lender.name}</p>

             {booking.status === "Accepted" && (
              <button
                     className="mt-4 w-full bg-blue-500 text-white py-2 rounded-lg
hover:bg-blue-600"
```

```
            onClick={() => handleReleaseClick(booking)}
          >
            Release Parking Place
          </button>
        )}
      </div>
    ))}
  </div>
)}

{releaseSuccess && (
  <div className="mt-4 bg-green-100 text-green-700 p-4 rounded-md text-center">
    {releaseSuccess}
  </div>
)}
{releaseError && (
  <div className="mt-4 bg-red-100 text-red-700 p-4 rounded-md text-center">
    {releaseError}
  </div>
)}

{/* Dummy Payment Modal */}
{showPaymentModal && selectedBooking && (
  <div className="fixed inset-0 bg-black bg-opacity-50 flex justify-center items-center z-50">
    <div className="bg-white w-full max-w-md p-6 rounded-xl shadow-lg text-center">
      {!paymentCompleted ? (
        <>
              <h2 className="text-xl font-bold mb-4">Payment for Booking #{selectedBooking.id}</h2>
          <p className="mb-2 text-gray-800">
            <strong>Parking Duration:</strong>{" "}
              {Math.ceil((new Date(selectedBooking.endTime) - new Date(selectedBooking.startTime)) / (1000 * 60 * 60))} hours
          </p>
          <p className="mb-4 text-gray-800">
            <strong>Total Amount:</strong> ₹{calculateAmount(selectedBooking.startTime, selectedBooking.endTime)}
          </p>

              <label className="block mb-2 font-medium">Choose Payment Method:</label>
          <select
            className="w-full mb-4 p-2 border rounded"
            value={paymentMethod}
            onChange={(e) => setPaymentMethod(e.target.value)}
          >
            <option value="CARD">Credit/Debit Card</option>
            <option value="UPI">UPI</option>
          </select>

          {paymentMethod === "CARD" ? (
            <>
               <input type="text" placeholder="Card Number" className="w-full p-2 border mb-2 rounded" />
              <input type="text" placeholder="Expiry (MM/YY)" className="w-full p-2 border mb-2 rounded" />
```

```
              <input type="text" placeholder="CVV" className="w-full p-2 border mb-4 rounded" />
            </>
          ) : (
             <input type="text" placeholder="Enter UPI ID" className="w-full p-2 border mb-4 rounded" />
          )}

          <div className="flex justify-between">
            <button
              className="bg-gray-300 px-4 py-2 rounded hover:bg-gray-400"
              onClick={() => setShowPaymentModal(false)}
            >
              Cancel
            </button>
            <button
              className={`bg-green-500 text-white px-4 py-2 rounded hover:bg-green-600 ${isPaying ? 'opacity-50 cursor-not-allowed' : ''}`}
              onClick={confirmPayment}
              disabled={isPaying}
            >
              {isPaying ? "Processing..." : "Confirm Payment"}
            </button>
          </div>
```

Comparison: SMART ONLINE PARKING AND RESERVATION SYSTEM vs Existing Apps

Category	Feature	SOPRS	Existing Apps (e.g., Park+, SpotHero, JustPark)
Core Functionalities	Real-time Parking Availability	Yes, via interactive map	Yes, typically with IoT or thirdparty integration
	Online Reservation	Yes	Yes
	Vehicle Registration	Yes	Sometimes (varies)
	User Feedback & Ratings	Yes	Yes
	Lender Space Listing	Yes (individual property owners)	Some apps (e.g., JustPark, Spacer) support this
	Booking Status Update	Yes (user indicates when they leave)	Rare
	Profile Management	Yes (for both users and lenders)	Basic user profiles; lender management varies
Technical Architecture	Frontend	ReactJS	React Native, Angular, Native Android/iOS

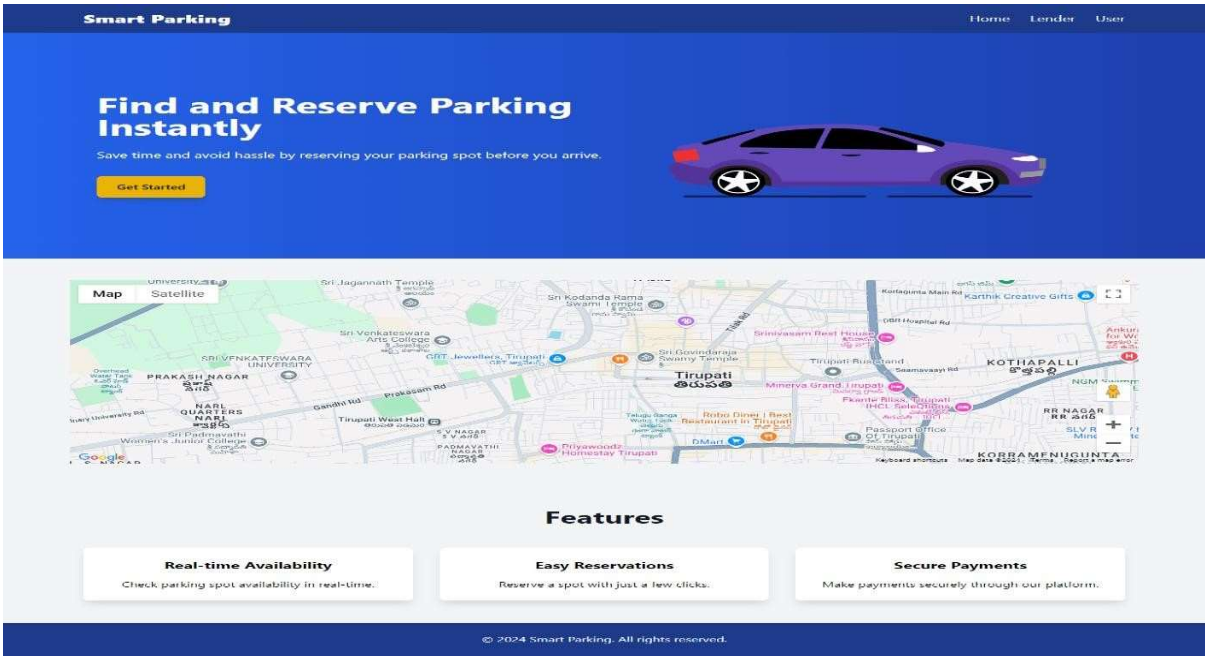
	Backend	Spring Boot (Java)	Node.js, Python, Java, Ruby (varies)
	Database	MySQL	PostgreSQL, MongoDB, cloud DBs (e.g., AWS RDS)
	Architecture Style	Modular (User/Lender separation)	Mostly monolithic or microservices
	Deployment	Web application	Web and fullfeatured mobile apps
Unique Strengths of SOPRS	Manual Departure Notification	Yes	Mostly not included
	Structured UML/System Design	Yes (Use Case, Class, ER, DFD diagrams)	Rarely disclosed publicly in commercial apps
	Transparent Ratings System	Yes (visible to both users and lenders)	Yes
Limitations/Enhancements Needed	Mobile App	Not yet available	Fully operational on Android/iOS
	Payment Integration	Planned for future	Fully integrated (credit card, PayPal, wallets, etc.)
	IoT Sensor Support	Not implemented	Common in premium apps

	Smart Navigation/Geolocation	Basic map-based; enhanced routing	Often includes live navigation
	Accuracy	proposed	with APIs like Google Maps
	Dynamic Pricing / AI	Not included	Some apps use predictive pricing or AI-based spot suggestion

APPENDIX-B

SCREENSHOTS

Homepage: default-landing page here about the project and services components are there



Find and Reserve Parking Instantly

Save time and avoid hassle by reserving your parking spot before you arrive.

Features

Real-time Availability

Check parking spot availability in real-time.

Easy Reservations

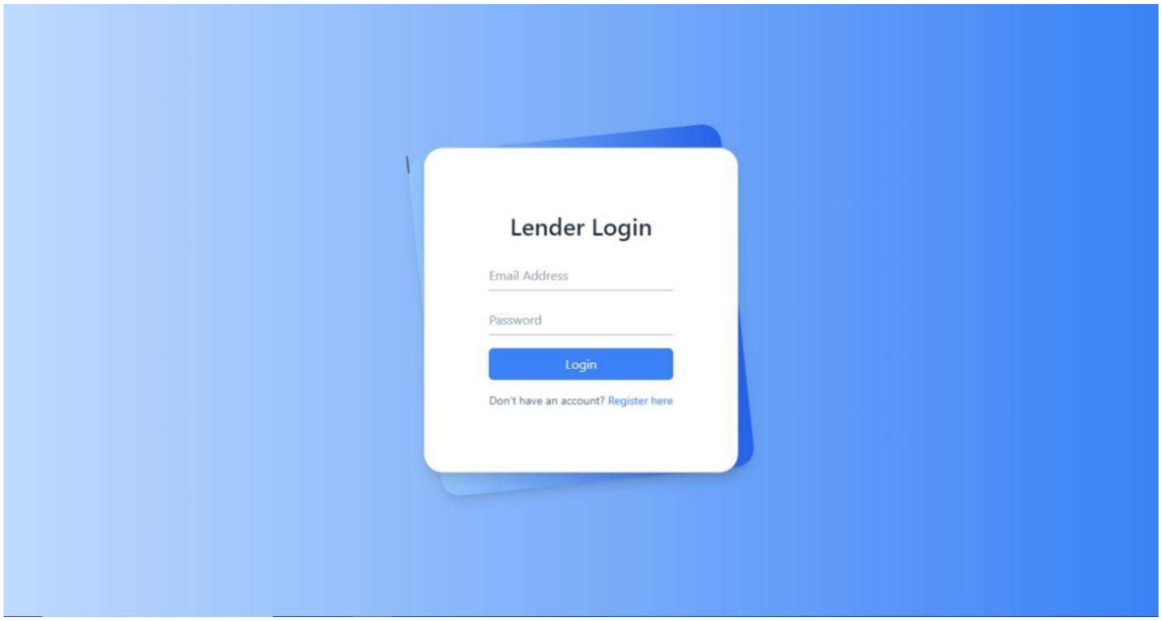
Reserve a spot with just a few clicks.

Secure Payments

Make payments securely through our platform.

© 2024 Smart Parking. All rights reserved.

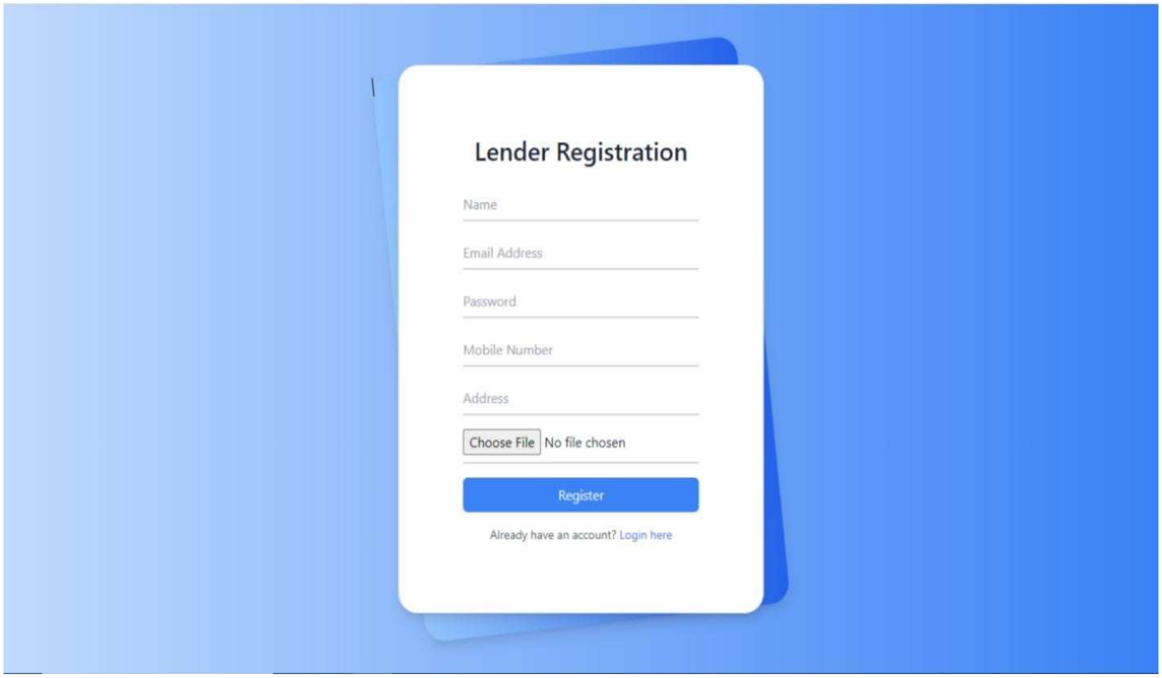
Lender Login: Lender will login to the application by entering credentials.



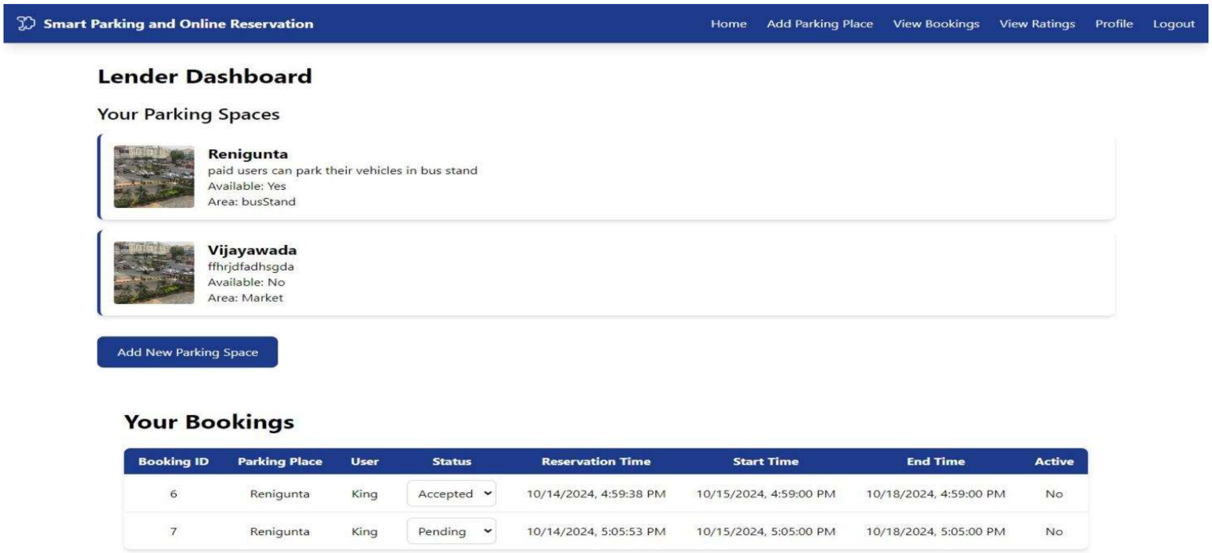
Lender Login

Don't have an account? Register here

Lender Register: Lender will register into application by entering his details



Lender Home: After successful login, lender will redirect to his homepage



Lender Dashboard

Your Parking Spaces

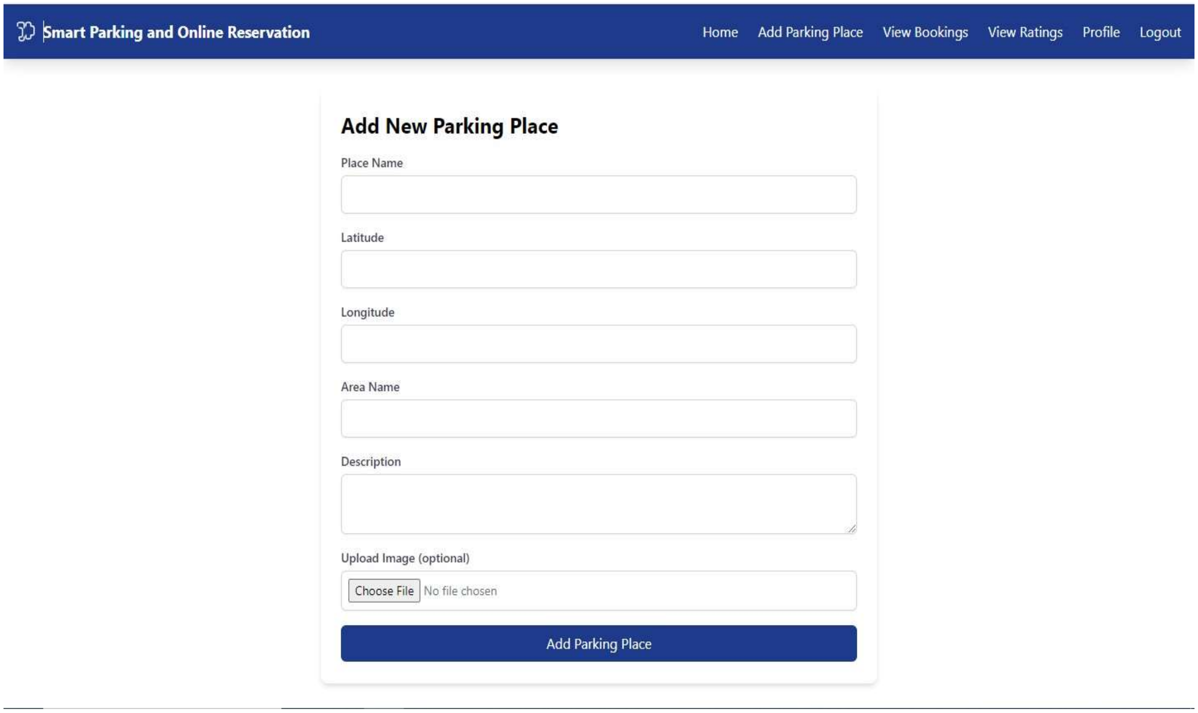
Renigunta
paid users can park their vehicles in bus stand
Available: Yes
Area: busStand

Vijayawada
ffhrjdfadhsgda
Available: No
Area: Market

Your Bookings

Booking ID	Parking Place	User	Status	Reservation Time	Start Time	End Time	Active
6	Renigunta	King	Accepted	10/14/2024, 4:59:38 PM	10/15/2024, 4:59:00 PM	10/18/2024, 4:59:00 PM	No
7	Renigunta	King	Pending	10/14/2024, 5:05:53 PM	10/15/2024, 5:05:00 PM	10/18/2024, 5:05:00 PM	No

Add Parking Place: Lender will add the parking places.



Add New Parking Place

Place Name

Latitude

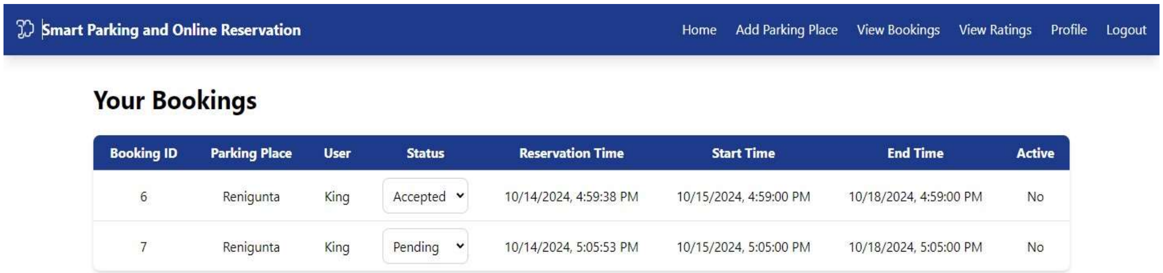
Longitude

Area Name

Description

Upload Image (optional)

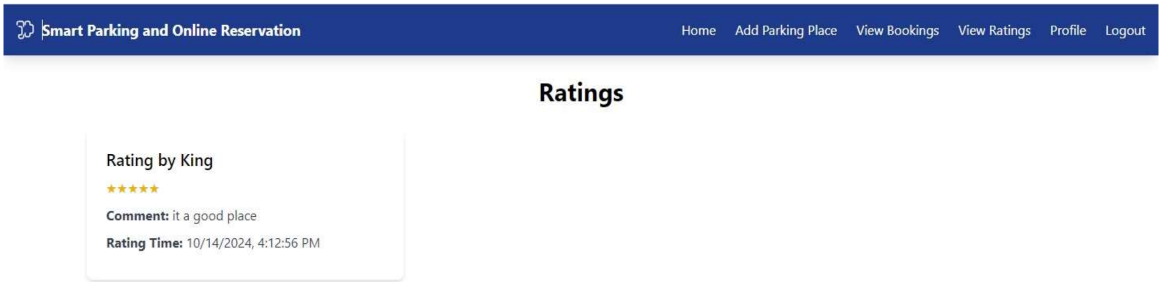
View Bookings: lender will view his bookings.



Your Bookings

Booking ID	Parking Place	User	Status	Reservation Time	Start Time	End Time	Active
6	Renigunta	King	Accepted	10/14/2024, 4:59:38 PM	10/15/2024, 4:59:00 PM	10/18/2024, 4:59:00 PM	No
7	Renigunta	King	Pending	10/14/2024, 5:05:53 PM	10/15/2024, 5:05:00 PM	10/18/2024, 5:05:00 PM	No

View Ratings: Lender will view his ratings



Ratings

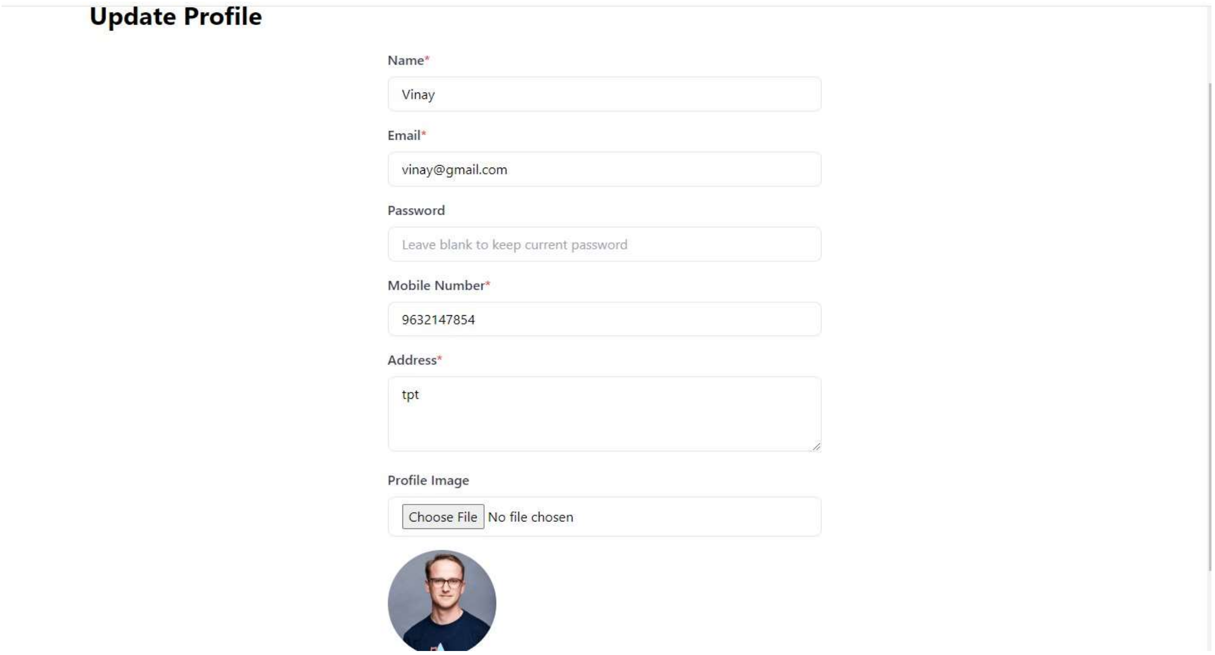
Rating by King

★★★★★

Comment: it a good place

Rating Time: 10/14/2024, 4:12:56 PM

Update Profile: Lender can update his profile.


Update Profile

Name*: Vinay

Email*: vinay@gmail.com

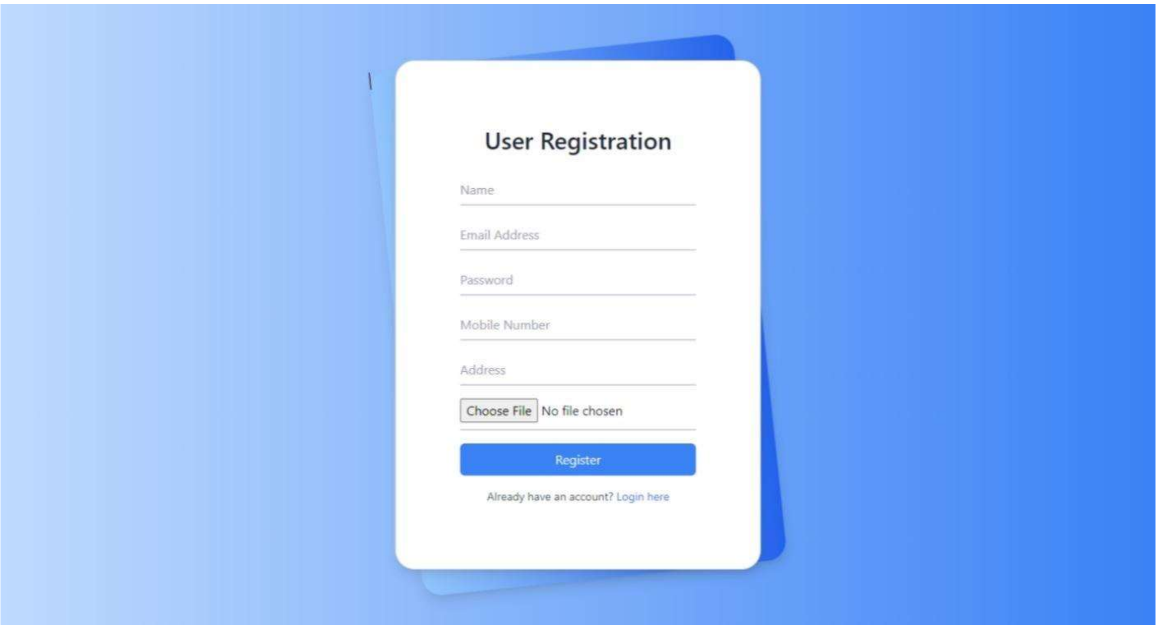
Password: Leave blank to keep current password

Mobile Number*: 9632147854

Address*: tpt

Profile Image: No file chosen

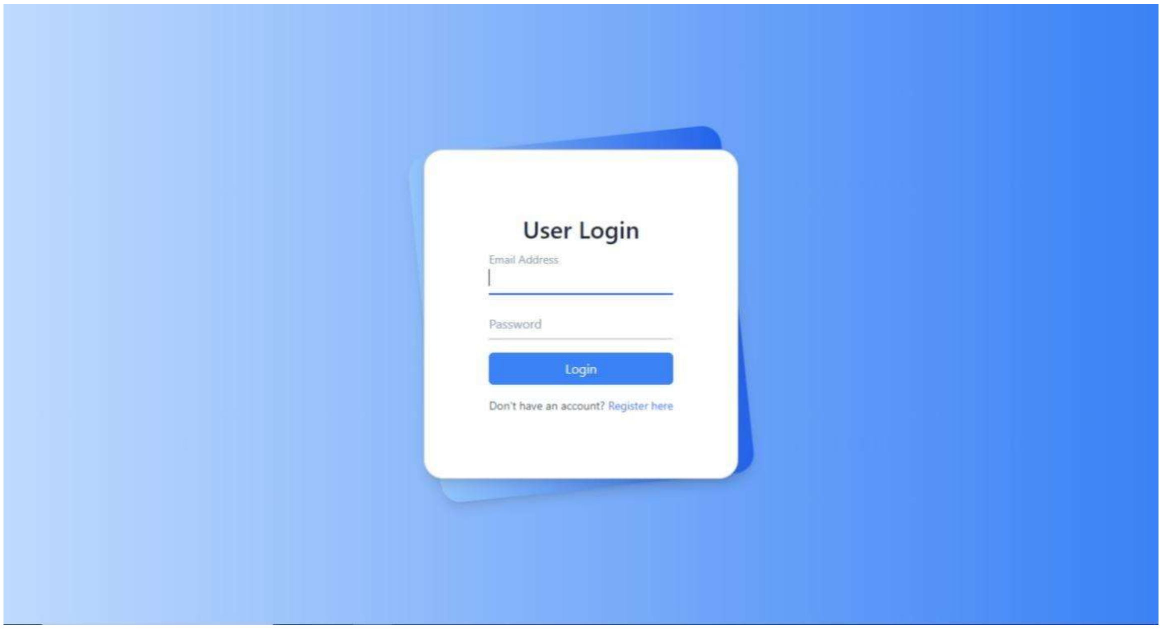
User Register:



User Registration

Already have an account? Login here

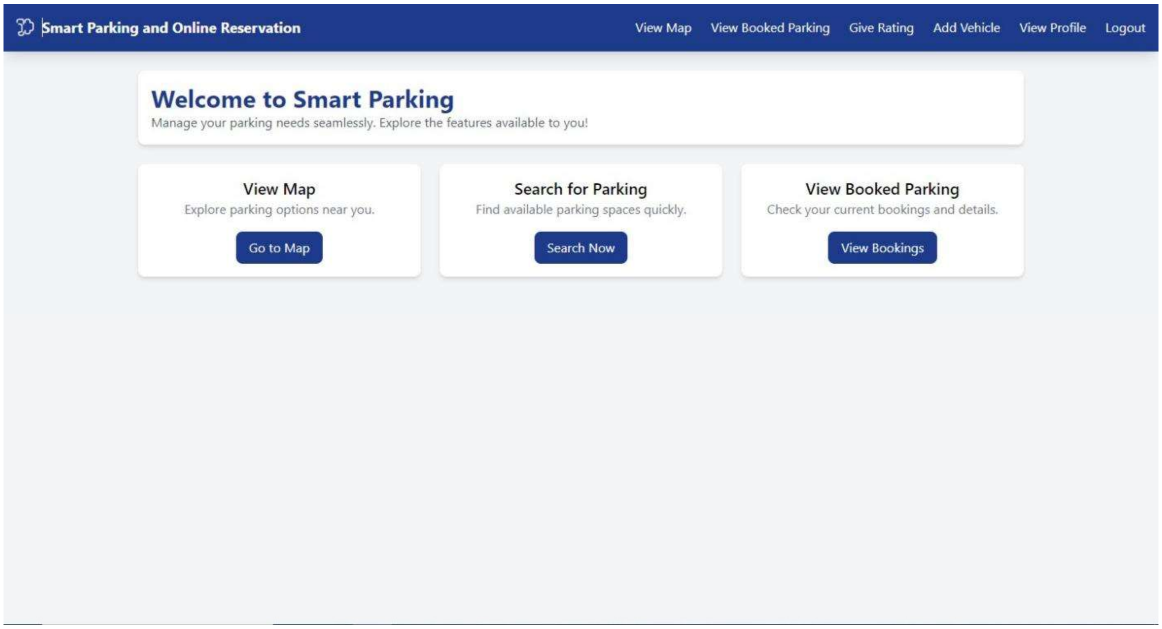
User Login:



User Login

Don't have an account? Register here

User Home:



Smart Parking and Online Reservation

View Map View Booked Parking Give Rating Add Vehicle View Profile Logout

Welcome to Smart Parking

Manage your parking needs seamlessly. Explore the features available to you!

View Map

Explore parking options near you.

Go to Map

Search for Parking

Find available parking spaces quickly.

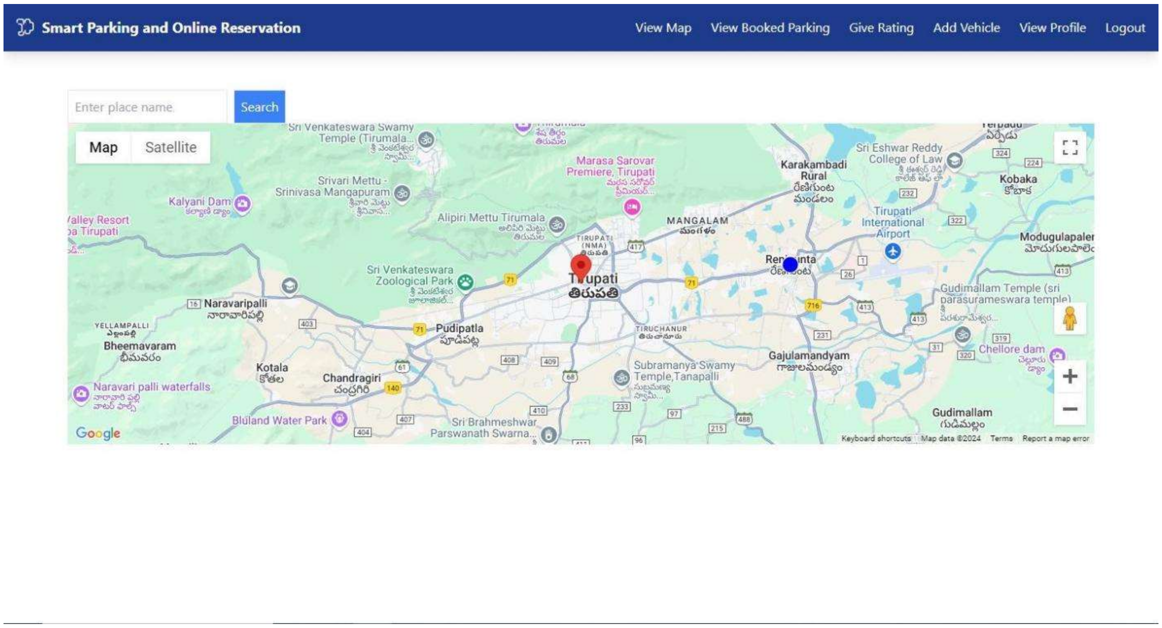
Search Now

View Booked Parking

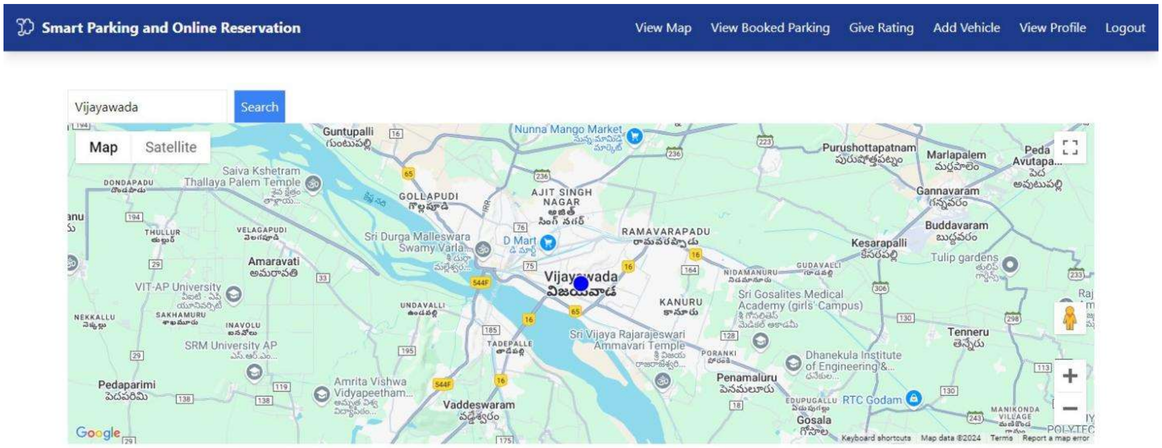
Check your current bookings and details.

View Bookings

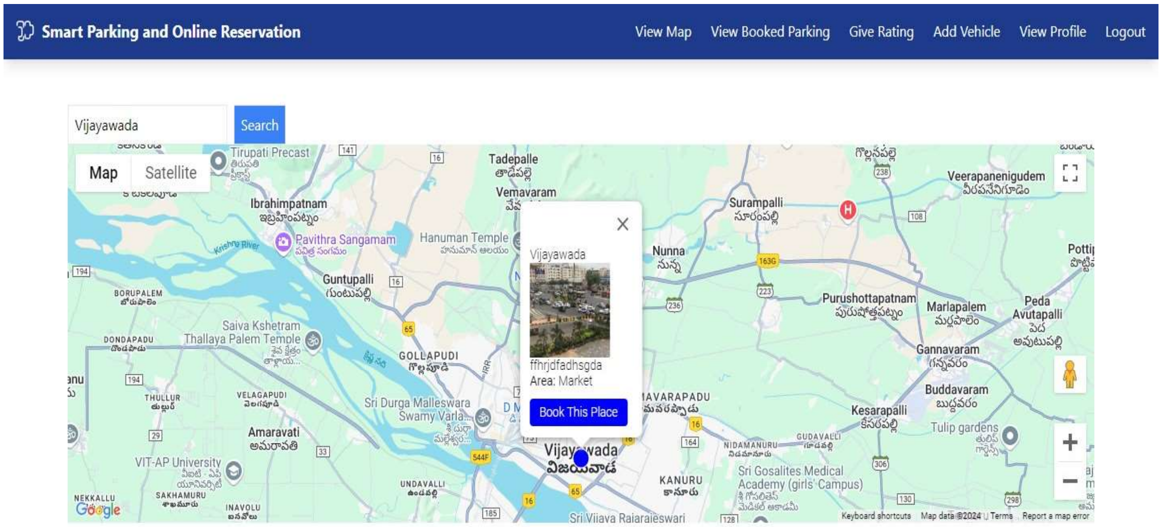
View Map:



Search Parking Place



User View Parking Place:

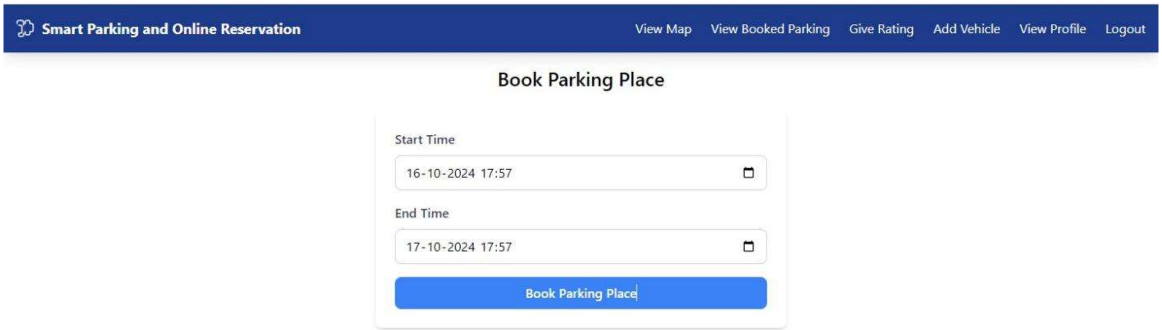


Vijayawada

ffhrjdfadhsgda

Area: Market

Book His Parking Place:



Book Parking Place

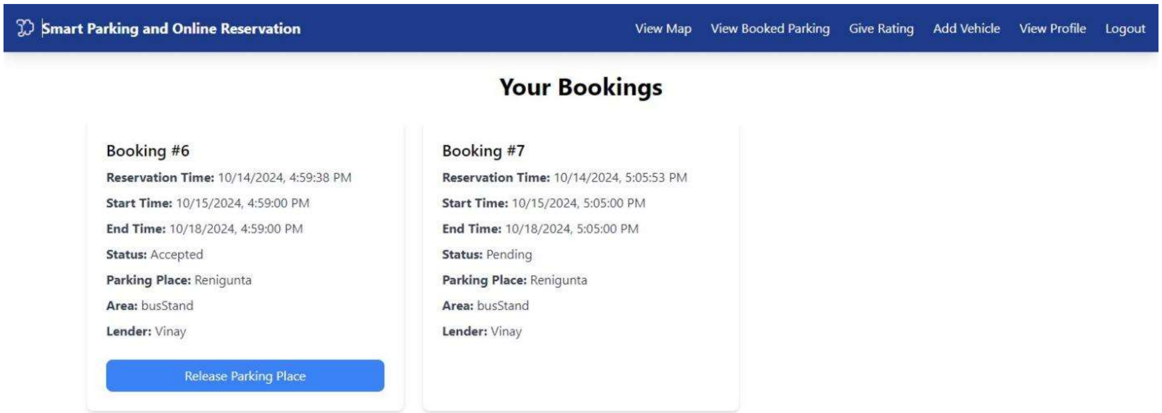
Start Time

16-10-2024 17:57

End Time

17-10-2024 17:57

View His Booking and Status:



Your Bookings

Booking #6

Reservation Time: 10/14/2024, 4:59:38 PM

Start Time: 10/15/2024, 4:59:00 PM

End Time: 10/18/2024, 4:59:00 PM

Status: Accepted

Parking Place: Renigunta

Area: busStand

Lender: Vinay

Booking #7

Reservation Time: 10/14/2024, 5:05:53 PM

Start Time: 10/15/2024, 5:05:00 PM

End Time: 10/18/2024, 5:05:00 PM

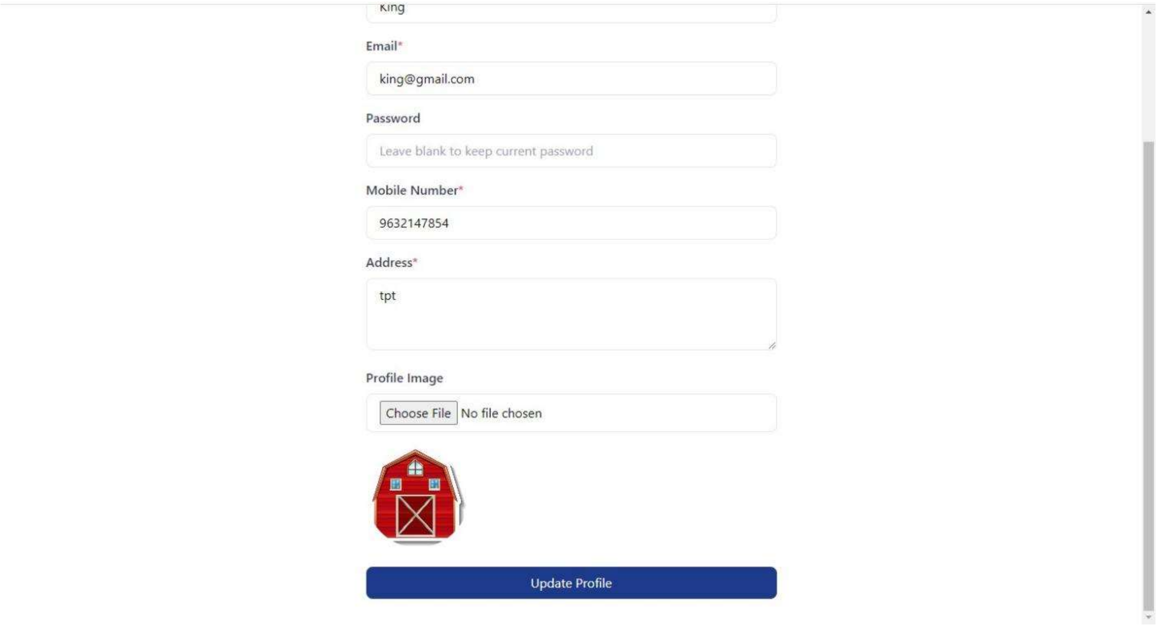
Status: Pending

Parking Place: Renigunta

Area: busStand

Lender: Vinay

Update Profile:



King

Email*

king@gmail.com

Password

Leave blank to keep current password

Mobile Number*

9632147854

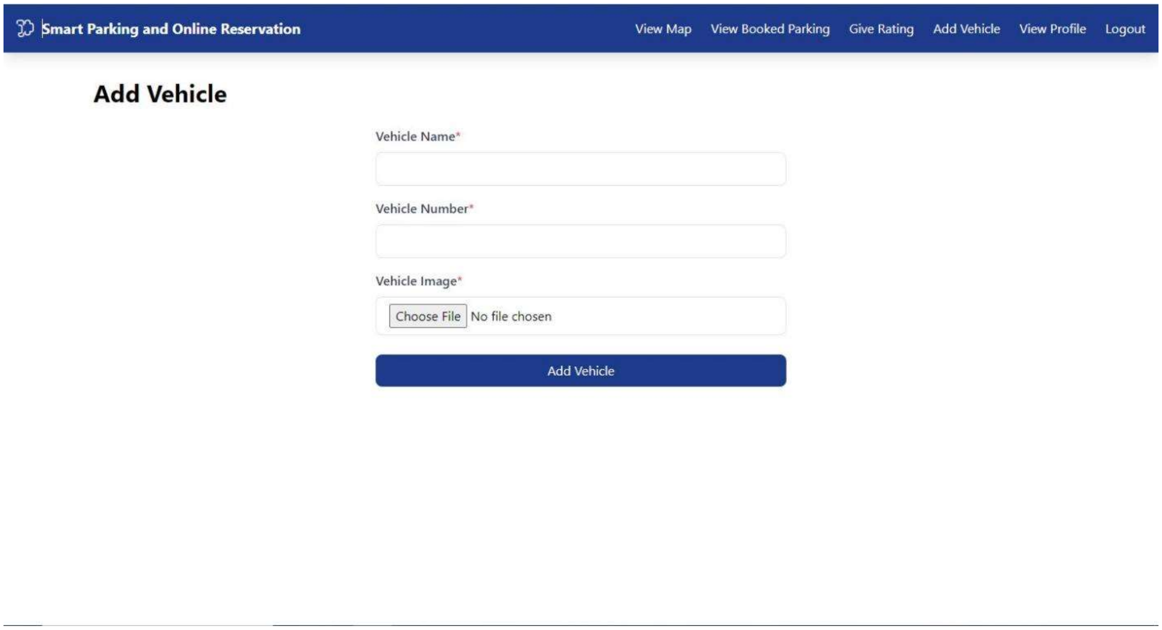
Address*

tpt

Profile Image

Choose File No file chosen

Add Vehicles:



Add Vehicle

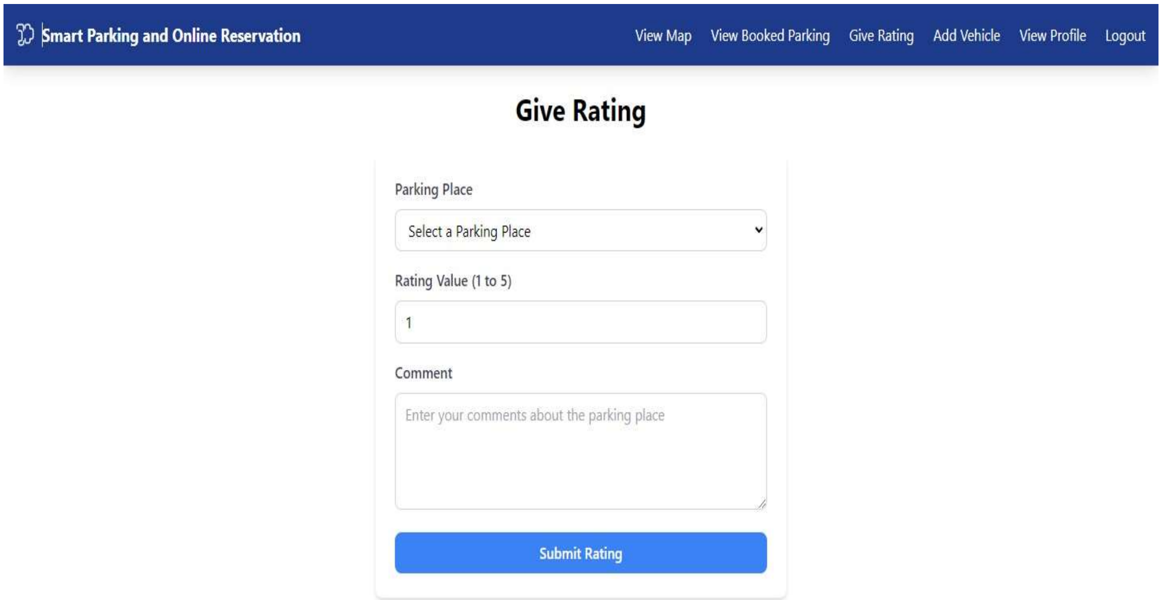
Vehicle Name*

Vehicle Number*

Vehicle Image*

Choose File No file chosen

Give Rating:



Give Rating

Parking Place

Select a Parking Place

Rating Value (1 to 5)

1

Comment

Enter your comments about the parking place

APPENDIX-C

ENCLOSURES

Details of mapping the project with the Sustainable Development Goals (SDGs).

SUSTAINABLE DEVELOPMENT GOALS
1 NO POVERTY
2 ZERO HUNGER
3 GOOD HEALTH AND WELL-BEING
4 QUALITY EDUCATION
5 GENDER EQUALITY
6 CLEAN WATER AND SANITATION
7 AFFORDABLE AND CLEAN ENERGY
8 DECENT WORK AND ECONOMIC GROWTH
9 INDUSTRY, INNOVATION AND INFRASTRUCTURE
10 REDUCED INEQUALITIES
11 SUSTAINABLE CITIES AND COMMUNITIES
12 RESPONSIBLE CONSUMPTION AND PRODUCTION
13 CLIMATE ACTION
14 LIFE BELOW WATER
15 LIFE ON LAND
16 PEACE, JUSTICE AND STRONG INSTITUTIONS
17 PARTNERSHIPS FOR THE GOALS

Goal 9: Industry, Innovation and Infrastructure

Target: Build resilient infrastructure, promote inclusive and sustainable industrialization, and foster innovation.

Elaboration:

1. Promoting Digital Innovation

A Smart Parking System is rooted in technological innovation, using:

- Internet of Things (IoT): Sensors detect parking space availability in real time.
- Mobile Applications: Allow users to find and reserve spots in advance.

- Cloud Computing & Big Data: Handle large-scale data on traffic patterns and parking usage.

This promotes a culture of innovation in public services and urban planning.

---

2. Building Smart Urban Infrastructure

Traditional parking systems often lead to inefficiencies like:

- Congestion from cars circling for spots.
- Poor use of urban space.
- Pollution from idling vehicles.

A smart parking system integrates digital infrastructure (sensors, real-time data) with physical infrastructure (parking lots, roads), leading to:

- Better space utilization.
- Reduced traffic.
- Efficient management of city infrastructure.

---

3. Supporting Sustainable Industrialization

Developing and maintaining these systems creates:

- New tech-oriented businesses (startups, platforms, hardware providers).
- Jobs in software development, system maintenance, and data analytics.

This boosts economic growth in smart city technology sectors, which are part of a sustainable industrial landscape.

---

4. Enhancing Connectivity

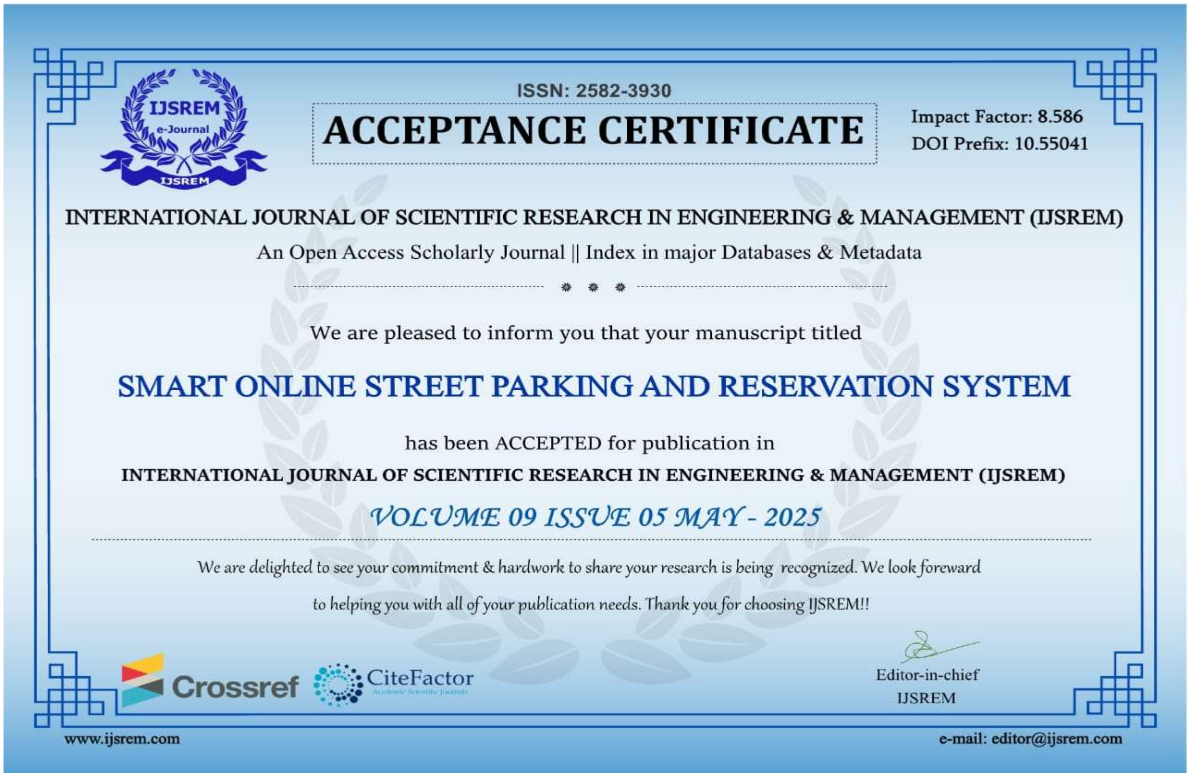
Modern infrastructure isn't just physical — it's digitally connected. Smart parking systems are part of larger smart city ecosystems that communicate with:

- Traffic lights.
- Public transportation.
- Electric vehicle charging stations.

This leads to a resilient and interconnected urban transport system.

ENCLOSURES

Journal publication/Conference Paper Presented Certificates (if any).



ISSN: 2582-3930

ACCEPTANCE CERTIFICATE

Impact Factor: 8.586
DOI Prefix: 10.55041

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT (IJSREM)

An Open Access Scholarly Journal || Index in major Databases & Metadata

We are pleased to inform you that your manuscript titled

SMART ONLINE STREET PARKING AND RESERVATION SYSTEM

has been ACCEPTED for publication in

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT (IJSREM)

VOLUME 09 ISSUE 05 MAY - 2025

We are delighted to see your commitment & hardwork to share your research is being recognized. We look foreward to helping you with all of your publication needs. Thank you for choosing IJSREM!!

Editor-in-chief
IJSREM

www.ijsrem.com

e-mail: editor@ijsrem.com