Shri B.V.V. Sangha's

BASAVESHWAR ENGINEERING COLLEGE, BAGALKOTE

[AICTE Approved, Government Aided Institution Affiliated To VTU, Belagavi]

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Mini-Project Report on

"PARKING MANGEMENT SYSTEM"

Submitted in partial fulfillment of requirement for the award of the degree

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

2024-2025

Project Guide	Coordinator	HOD
Prof. P. B. Madhavanavar	Prof. Smitha. K	Dr . V. B. Pagi

Submitted by:

Tejaswi. B. Kiranagi	2BA22CS110
Vinayaka	2BA22CS121
Vishnu	2BA22CS122
Yamanoorappa	2BA22CS124

Shri B.V.V. Sangha's

BASAVESHWAR ENGINEERING COLLEGE, BAGALKOTE

[AICTE Approved, Government Aided Institution Affiliated To VTU, Belagavi]

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that **Mr. Tejaswi. B. Kiranagi (2BA22CS110), Mr. Vinayaka (2BA22CS121), Mr. Vishnu (2BA22CS122), Mr. Yamanoorappa (2BA22CS124),** has satisfactorily completed the mini-project on **"Parking Mangement System",** for partial fulfillment of their course in B.E. Computer Science and Engineering prescribed by Basaveshwar Engineering College, Bagalkote during the academic year 2024-2025.

Project Guide	**Coordinator**	**HOD**
Prof. P. B. Madhavanavar	**prof. Smitha. K**	**Dr. V. B. Pagi**

Name of Examiners	**Signature of Examiner**
1……………………	1………………………
2…………………....	2………………………
3……………………	3………………………

ABSTRACT

PARKING MANGEMENT SYSTEM

The **Parking Management System** is a comprehensive software solution designed to streamline and optimize the operations of modern parking facilities. This system addresses key challenges such as slot allocation, payment processing, and user management through automation and user-friendly interfaces.

Built using Python's Flask framework and integrated with a MySQL database, the system supports functionalities like user registration, slot booking, dynamic pricing, secure payments, and bill generation. Admin users benefit from real-time dashboards for monitoring and managing parking resources efficiently, while end-users experience a seamless booking and payment process.

The system employs robust security measures, such as password hashing and session management, to ensure data safety and user privacy. It also incorporates automated bill generation in PDF format for enhanced convenience and transparency. Furthermore, the system is scalable and adaptable, making it suitable for various types of parking facilities, from small lots to large multi-location operations.

This project contributes to reducing traffic congestion and carbon emissions by minimizing the time vehicles spend searching for parking. It also introduces potential enhancements, including IoT integration, AI-powered analytics, and mobile app support for real-time updates and personalized services.

In summary, the Parking Management System is a versatile, secure, and sustainable solution that meets the growing demands of smart urban infrastructure, delivering value to both administrators and end-users alike.

TABLE OF CONTENTS:

Chapter 1

Introduction

1.1. Overview

The Parking Management System is a digital solution designed to automate and streamline parking operations. It efficiently handles parking slot allocation, vehicle tracking, and payment processing, ensuring optimal utilization of resources. The system allows users to check real-time slot availability, facilitates quick entry and exit, and provides secure payment options. Administrators can monitor slot usage, manage payments, and generate reports for analysis. Built with Python Flask and integrated with a database like SQLite or MySQL, the system ensures seamless operation, reduces manual errors, and enhances user experience. It is scalable and customizable for diverse parking facility needs.

1.2. Role in Modern Parking Mangement System

The role of a **Modern Parking Management System** is to revolutionize traditional parking operations through automation, efficiency, and user-centric functionality. It leverages technology to address parking challenges, ensuring smooth, secure, and hassle-free management. Its key roles include:

1. **Slot Optimization**: Efficiently manages parking space to maximize utilization and minimize idle slots.
2. **Real-Time Monitoring**: Tracks vehicle entries, exits, and slot availability in real-time.
3. **Revenue Management**: Automates payment processing, tracks revenue, and reduces fraud.
4. **User Convenience**: Enhances user experience through pre-booking, quick access, and transparent billing.
5. **Scalability and Sustainability**: Supports advanced features like IoT sensors, EV charging integration, and eco-friendly practices, catering to modern urban needs.

1.3 Features

- **Real-Time Slot Availability :** Displays current availability of parking slots to users and administrators.
- **Automated Slot Allocation:** Automatically assigns the nearest or preferred slot to incoming vehicles.
- **Vehicle Tracking:** Logs entry and exit times with vehicle details for accurate tracking.
- **Payment Integration:** Supports multiple payment methods (cashless options like UPI, credit/debit cards, wallets).
- **User-Friendly Interface:** Intuitive web or mobile app for users to book slots, check availability, and make payments.
- **Data Security:** Ensures secure storage and processing of user and transaction data
- **Notification System:** Sends alerts to users for slot confirmation, payment receipts, or reminders.
- **Customizable Pricing:** Allows flexible pricing based on time, duration, or special events.
- **Admin Dashboard:** Provides administrators with tools to monitor slots, manage users, and view revenue statistics.
- **Report Generation:** Generates reports on parking usage, revenue, and system performance for analysis.

Chapter 2

Literature Survey

2.1.Existing System

Parking Management Systems have evolved significantly over the years, with various approaches and technologies being implemented to address the challenges of traditional parking operations. Here is a review of existing systems and their key characteristics:

2.1.1. Efficient Parking Management

Parking has become a significant challenge in daily life, with issues such as availability, emissions, slot allocation, and charges. Understanding patterns and timelines is crucial to addressing these problems. This project aims to design a parking management system that optimizes space utilization while reducing emissions and enhancing user satisfaction. Using priority-based algorithms like First Come First Serve (FCFS), priority, and distance, combined with innovative approaches, the system ensures efficient parking allocation. It minimizes power consumption by managing ventilation systems to control CO and CO_2 levels, reducing emissions inside parking lots. Additionally, it shortens the time and effort required for customers to park, saving fuel and improving their experience. This indirect relationship between optimization and user satisfaction ensures maximum efficiency, benefiting both customers and parking owners by simplifying management and enhancing parking operations.

Keywords: IoT, Cloud-based parking management, Parking space optimization, Smart Parking, Parking Guidance System.

2.1.2. A SMART ANDROID PARKING SYSTEM

An intelligent parking system controlled by an Android application and a parking spot assignment model was created using the Internet of Things (IoT). On a crowded campus, staff and students scramble for parking spaces close to the classrooms, adding to traffic congestion and aggravating drivers who can't find any open spots. The created application, model, and hardware are used to simplify parking on a college campus because it always takes a significant amount of time to look for open spaces. Microprocessors with WiFi modules, ultrasonic sensors, motors, and gates were all part of the hardware. Students and faculty can use the application to reserve parking spaces or make online payments for parking. This procedure saves faculty and staff time while reducing traffic around the campus. The system is affordable, simple to create, adaptable, and usable in any parking application.

Keywords: Internet of Things, Android, Industry 4.0, parking model, Smart Parking

2.1.3. The Smart Parking Management System

With growing, Car parking increases with the number of car users. With the increased use of smartphones and their applications, users prefer mobile phone-based solutions. This paper proposes the Smart Parking Management System (SPMS) that depends on Arduino parts, Android applications, and based on IoT. This gave the client the ability to check available parking spaces and reserve a parking spot. IR sensors are utilized to know if a car park space is allowed. Its area data are transmitted using the WI-FI module to the server and are recovered by the mobile application which offers many options attractively and with no cost to users and lets the user check reservation details. With IoT technology, the smart parking system can be connected wirelessly to easily track available locations.

Keywords: Internet of Things, Cloud Computing, Smart Parking, Smart City, Mobile Application

2.1.4. Automatic Car Parking Management System

One of the most important issue in the modern world is issue of car parking. Normally parking space is available in every society but the number of people who are having more than one car are very large and in the number most of the builders normally tell the specific parking space before the formation of society to the member who go and purchase the flat soon. A big city like Mumbai, Delhi people have to pay more than 5 lakh for parking but it is illegal for the rule of the court. In urban area if someone has to go to the market or shopping Mall than owner also have to pay for parking at the rate of fixed rate for per hour parking if by mistake someone park your car at no parking then be ready to pay a big fine to the parking authority at so many market there is not any facility of parking many people have to park there car so far or in no parking by their own risk. This paper which my team has prepared with this my team are trying to make parking easier. There are also many new technology in which some are very helpful mean to say that some are successful and some are unsuccessful. Car parking issue is a very big deal, it cannot solve easily people have to take some steps and by this paper people can take a little steps to improve the car parking issue. Especially in urban area. It will be also very beneficial for our environment and streets. In this paper my team is trying to solve the issue of car parking.

Keywords: IoT, Arduino, Servo Motor, IR Sensor, Car Parking

Chapter 3

Problem Statement

Parking is a growing challenge in urban areas due to increased vehicle ownership and inefficient management. Traditional systems struggle with issues like limited slot availability, improper space utilization, long search times, and high emissions in enclosed parking areas. These problems lead to user frustration, wasted fuel, air pollution, and operational inefficiencies for parking owners.

A smart parking management system is needed to address these issues by optimizing space utilization, reducing emissions, and improving user satisfaction. By using a HTML CSS And Javascript for frontend and MYSQL,FLASK(Python) for backend, the system can provide real-time slot availability, efficient allocation, and environmental monitoring. Such a solution ensures sustainable, time-saving, and user-friendly parking, benefiting both customers and parking operators in modern urban settings.

3.1. Problem definition

The increasing number of vehicles in urban areas has led to significant challenges in managing parking spaces efficiently. Traditional parking systems lack the ability to optimize parking slot utilization, resulting in unorganized parking, wasted time searching for available spaces, increased fuel consumption, and higher carbon emissions. Additionally, inadequate ventilation in enclosed parking areas contributes to unhealthy levels of CO and CO_2, affecting air quality.

Parking operators face difficulties in monitoring and managing spaces effectively, leading to revenue loss and customer dissatisfaction. There is a need for a smart, automated solution that incorporates real-time slot tracking, efficient space allocation, and environmental monitoring. Such a system should reduce emissions, save time, and enhance both customer and operator experiences.

3.2 Issues and Challenges

1. **Limited Parking Space Availability**
   - Difficulty in accommodating the growing number of vehicles in urban areas.
   - Inefficient allocation leading to underutilized or overcrowded spaces.
2. **Time Consumption**
   - Users spend excessive time searching for available parking slots, especially in busy areas.
3. **Environmental Concerns**
   - High fuel consumption and increased emissions (CO, CO_2) due to prolonged search times.
   - Poor ventilation in enclosed spaces, leading to unhealthy air quality.

4. **Manual Errors**
    - Dependence on traditional manual systems causes errors in slot allocation and billing.
5. **Lack of Real-Time Data**
    - Absence of real-time updates on slot availability frustrates users and reduces efficiency.
6. **High Operational Costs**
    - Inefficient resource management increases costs for parking operators.
7. **Customer Dissatisfaction**
    - Long wait times, unclear instructions, and unoptimized slot allocation reduce user satisfaction.
8. **Scalability Issues**
    - Traditional systems struggle to handle increasing vehicle volumes and multi-location operations.
9. **Security Concerns**
    - Poor monitoring can lead to vehicle thefts or unauthorized parking.
10. **Integration with Advanced Technologies**
    - Challenges in implementing IoT, cloud-based systems, or machine learning for real-time data and automation.

3.3Limitations:

1. **High Initial Setup Costs**

    - Advanced parking management systems, especially those incorporating IoT, sensors, and real-time tracking, require significant initial investment for hardware, software, and infrastructure setup.

2. **Technology Dependence**

    - Reliance on internet connectivity and power sources may lead to system downtime or failures if the infrastructure is unstable or experiences technical issues.

3. **Complexity in Integration**

    - Integrating newer technologies like IoT, AI, and vehicle recognition systems with existing parking infrastructure can be complex and require customization.

4. **Maintenance Costs**

- Regular maintenance of sensors, cameras, and other hardware components increases operational costs over time.

5. **Data Privacy and Security**

   - Managing sensitive user data, including payment information, requires robust security protocols to prevent data breaches or cyberattacks.

6. **User Adoption**

   - Resistance from users who are unfamiliar or uncomfortable with technology, especially mobile apps or automated parking solutions, may limit system effectiveness.

7. **Scalability Limitations**

   - While systems can be designed for individual parking lots, scaling them to multi-location operations or across cities can be challenging due to variations in infrastructure and user requirements.

8. **Environmental Factors**

   - Sensors and cameras may have reduced effectiveness in extreme weather conditions, such as heavy rain or fog, impacting the system's accuracy.

9. **Legal and Regulatory Barriers**

   - Local regulations, especially concerning data privacy, surveillance, and urban planning, may pose challenges for implementing advanced systems in certain regions.

3.4Applications:

1. Urban Parking Solutions

- **Application**: Optimizes parking space usage in busy city areas, helping reduce traffic congestion and improve the overall urban mobility experience.

- **Example**: Smart city parking systems that provide real-time data on available spaces, enabling drivers to find parking quickly.

2. Commercial Parking Facilities

- **Application**: Used in malls, office complexes, and airports to manage high volumes of vehicles while ensuring efficient space utilization and customer satisfaction.

- **Example**: Automated systems in shopping malls that guide drivers to available parking slots and handle payment seamlessly.

3. Residential Complexes

- **Application**: Manages parking spaces in gated communities or apartment complexes, offering residents reserved parking and reducing conflicts over parking spaces.

- **Example**: A system that assigns parking spots to residents and monitors their usage to prevent unauthorized parking.

4. Electric Vehicle (EV) Parking

- **Application**: Specialized parking for electric vehicles with integrated charging stations, allowing users to find and reserve spots for charging their EVs.

- **Example**: Smart parking systems that identify available EV charging spots and automatically manage charging sessions.

5.Airport Parking

- **Application**: Helps passengers book parking spaces in advance, reducing the time spent searching for spots and providing seamless entry and exit processes.

- **Example**: Systems that reserve and guide drivers to available spaces based on their flight schedules, reducing congestion at airport parking areas.

6.On-Street Parking

- **Application**: Real-time monitoring of on-street parking availability, helping drivers find open spaces quickly and pay electronically.

- **Example**: A mobile app that allows users to check availability and pay for on-street parking.

7.Event Parking

- **Application**: Manages parking at venues hosting large events such as concerts, conferences, or sports games, optimizing space usage and reducing waiting times.

- **Example**: A parking system that provides event-goers with pre-booked spaces, ensuring smooth entry and exit during busy event days.

8. Fleet Management

- **Application**: Helps manage the parking of company fleets by tracking the availability and usage of vehicle slots for businesses operating with multiple vehicles.

- **Example**: A company uses a smart parking system to monitor fleet vehicles, ensuring that they are parked efficiently and minimizing downtime.

9.Car Sharing & Ride-Hailing Services

- **Application**: Manages parking spaces for shared cars or ride-hailing services, ensuring efficient vehicle pickup and drop-off operations.

- **Example**: A parking system that identifies optimal drop-off and pick-up points for ride-sharing services, optimizing traffic flow and reducing congestion.

10.Public Transport Hubs

- **Application**: Manages parking at transport stations such as train stations or bus terminals, offering spaces for commuters to park their cars before transferring to public transport.

- **Example**: A smart parking system at a railway station that allows commuters to easily find parking and track availability.

Chapter 4

Software Requirements

4.1. User Requirements:

Functional Requirements

1. **User Module**
    - Register and log in to the system.
    - View real-time parking slot availability.
    - Book, cancel, or modify parking reservations.
    - Make payments through multiple methods (UPI, credit/debit cards, wallets).
2. **Admin Module**
    - Monitor and manage parking slots.
    - View revenue reports and analytics.
    - Handle user accounts and issue notifications.
3. **Parking Slot Management**
    - Automated slot allocation based on algorithms (FCFS, priority, distance).
    - Real-time tracking of slot usage.
    - Notification of slot availability.
4. **Payment Integration**

- Secure and seamless online payment processing.
- Generate and send digital receipts to users.

5. **Environmental Monitoring**
   - Track and manage CO and CO_2 levels in enclosed parking areas.

6. **Reporting and Analytics**
   - Generate usage reports for administrators.
   - Provide insights on slot utilization, peak hours, and revenue.

Non-functional Requirements:

1. **Performance**
   - The system must handle up to 500 concurrent users with minimal latency.
2. **Scalability**
   - Must support multi-location parking facilities and scale with increasing users and vehicles.
3. **Reliability**
   - Ensure system uptime of 99.9% to avoid disruptions.
4. **Usability**
   - Provide an intuitive and user-friendly interface for all stakeholders.
5. **Security**
   - Encrypt sensitive user data and payment information.
   - Implement authentication mechanisms (e.g., OTP, two-factor authentication).

6. **Compatibility**

   - Compatible with web browsers and mobile devices (iOS and Android).

7. **Maintainability**

   - Modular design for easy updates and bug fixes.

4.2 System Requirements

Functional Requirements

1. **User Management**

   - Allow users to register, log in, and manage their profiles.
   - Enable administrators to manage user accounts.

2. **Slot Management**

   - Display real-time availability of parking slots.
   - Provide automated slot allocation using algorithms like First Come First Serve (FCFS) or priority- based allocation.
   - Enable manual slot allocation and adjustments by the administrator.

3. **Booking and Reservations**

   - Allow users to book, modify, or cancel parking slots.
   - Maintain a record of current and past bookings.

4. **Payment Integration**

   - Support secure online payment methods (e.g., UPI, credit/debit cards, wallets).
   - Generate and send digital receipts for all transactions. .

5. **Notifications**

   - Send real-time updates and alerts for booking confirmations, cancellations, and slot availability.

6. **Navigation Assistance**
   - Integrate mapping APIs (e.g., Google Maps) to guide users to their designated parking slots.
7. **Environmental Monitoring**
   - Monitor air quality (CO and CO_2 levels) in enclosed parking areas.
   - Notify the administrator if thresholds exceed safe limits.
8. **Reports and Analytics**
   - Generate usage and revenue reports for administrators.
   - Provide analytics on slot utilization, peak hours, and user behavior.

Non-functional Requirements

1. **Performance**
   - The system should handle at least 500 concurrent users with minimal latency.
2. **Scalability**
   - Should support multi-location parking facilities and scale with increased user demand.
3. **Reliability**
   - Ensure 99.9% system uptime to avoid disruptions.
4. **Security**
   - Protect sensitive user data (e.g., personal information and payment details) through encryption.
   - Implement multi-factor authentication for secure access.
5. **Usability**
   - Provide a user-friendly interface for both customers and administrators.
   - Ensure that the mobile app and web platform are intuitive and easy to navigate.

6. **Compatibility**

   - The system should work on multiple platforms, including desktop browsers, Android, and iOS devices.

7. **Maintainability**

   - Ensure the system is modular for easy updates and debugging**.**

8. **Data Integrity**

   - Maintain consistent and accurate data for bookings, payments, and slot usage.

9. **Environmental Resilience**

   - Sensors and hardware should operate effectively in various environmental conditions, such as heat, cold, and humidity.

10. **Compliance**

    - Adhere to local data privacy and payment regulations.

4.3 Domain Requirements

1. **Parking Slot Management**
   - Real-time tracking and monitoring of parking slot availability.
   - Automated and manual allocation of parking spaces based on predefined algorithms (e.g., First Come First Serve, priority-based, or proximity).
   - Support for reserved, temporary, and VIP parking slots.
   - Integration with IoT-enabled sensors for accurate slot detection.

2. **User Interaction**
   - User-friendly interfaces for drivers to check availability, book, modify, or cancel slots.
   - Notification system for slot allocation, availability updates, and payment confirmations.
   - Support for multilingual interfaces to cater to diverse user bases.
3. **Payment and Billing**
   - Seamless integration with payment gateways for cashless transactions (UPI, wallets, cards).
   - Flexible billing options based on hourly, daily, or subscription-based models.
   - Automated invoice and receipt generation.

4. **Environmental Monitoring**
   - Continuous tracking of CO and CO_2 levels in closed parking environments.
   - Integration with ventilation systems to maintain air quality and alert users in case of unsafe levels.
5. **Reporting and Analytics**
   - Generate reports on parking usage, revenue, peak hours, and slot utilization.
   - Analyze user behavior and provide insights for optimizing operations.
   - Track and monitor emissions for compliance with environmental regulations.
6. **Security and Safety**
   - Provide video surveillance integration for parking lot monitoring.
   - Maintain logs of vehicle entry and exit for security purposes.
   - Restrict unauthorized access to parking slots.
7. **Legal and Regulatory Compliance**
   - Adhere to local laws regarding parking charges, emissions, and data privacy.
   - Comply with financial regulations for online transactions.
8. **System Scalability**
   - Support for multi-location parking facilities with centralized control.
   - Expandable to include additional features, such as EV charging stations or valet parking.
9. **Integration with External Systems**
   - Use of mapping APIs for navigation assistance.
   - Compatibility with smart city platforms and third-party parking aggregators.
10. **User Categories**
    - **General Users (Drivers):** Require booking, navigation, and payment options.
    - **Parking Operators (Admins):** Need tools for slot management, reporting, and customer interaction.
    - **Government/Regulatory Bodies:** May require access to usage reports and environmental compliance data.

Chapter 5

Proposed Methodology

5.1. Methodology

The **Parking Management System** is a **web-based platform** designed to optimize parking operations, enhance user convenience, and streamline slot management. The proposed methodology follows a structured, iterative approach to ensure functionality, reliability, and user satisfaction.

Steps in the Proposed Methodology:

1. **Requirement Analysis:**
    - Gather and analyze user and administrator needs, focusing on features like slot availability, booking, payments, and reports.
    - Identify the primary challenges, such as inefficient parking space usage, booking conflicts, and manual record-keeping.
    - Define functional and non-functional requirements to guide development.
2. **System Design:**
    - Develop a scalable, modular architecture that supports future enhancements.
    - Design **HTML CSS wireframes** for a user-friendly web interface.
    - Create an **Entity-Relationship (ER) diagram** to define the database schema for managing users, parking slots, bookings, and payments.
    - Establish workflows for core processes such as booking, slot allocation, and payment.
3. **Algorithm Development:**
    - Implement efficient algorithms for parking slot allocation:

    - **First Come First Serve (FCFS):** Allocate slots based on booking order.
    - **Priority-Based Allocation:** Assign slots to VIP or reserved users first.
  - Optimize algorithms to minimize user search time and maximize space utilization.

4. **Prototyping:**
   - Develop a prototype of key features, such as slot booking, availability display, and payment integration.
   - Present the prototype to stakeholders for feedback and make necessary adjustments.

5. **System Development:**
   - **Frontend Development:** Use technologies like **HTML**, **CSS**, and **JavaScript** for an interactive and responsive user interface.
   - **Backend Development:** Use **Flask** to handle server-side operations, including user authentication, slot allocation, and booking management.
   - **Database Integration:** Use **MySQL** to store and retrieve user profiles, parking slot data, and payment records.
   - **Payment Integration:** Integrate secure payment gateways (e.g., Razorpay, PayPal) for seamless transactions.

6. **Testing:**
   - Conduct thorough testing to ensure system reliability and performance:
     - **Functional Testing:** Verify that all features work as intended.
     - **Load Testing:** Test the system’s ability to handle multiple users simultaneously.

- **Security Testing:** Ensure secure handling of user data and payment information.

7. **Deployment:**

- Deploy the web-based platform on cloud services like **AWS** or **Heroku** for scalability and accessibility.
- Ensure compatibility with major web browsers for seamless user experience.

8. **Monitoring and Maintenance:**

- Monitor system performance and resolve any technical issues promptly.
- Update the platform regularly to enhance features and address user feedback.
- Maintain system logs for error tracking and analytics.

5.2 Module Design:

The **Parking Management System** is divided into several interconnected modules to ensure modularity, scalability, and ease of development. Each module has a specific responsibility and interacts seamlessly with other modules.:

1. **User Management Module**

- **Purpose:** Handle user accounts and roles (e.g., admin, regular user).
- **Features:**
  - User registration and login (with role-based access control).
  - Profile management (update personal details, view history).
  - Authentication and authorization mechanisms.

2. **Parking Slot Management Module**

- **Purpose:** Manage the availability and allocation of parking slots.
- **Features:**
  - Display real-time parking slot availability.
  - Slot booking, allocation, and deallocation.
  - Support for different slot types (reserved, general, VIP).
  - Admin control for manual slot adjustments.

3. **Booking and Reservation Module**

- **Purpose:** Facilitate online booking and reservation of parking slots.
- **Features:**

    - Search for available slots based on date, time, and location.
    - Make, modify, or cancel bookings.
    - Maintain a booking history for users and admins.

4. **Payment Module**

- **Purpose:** Enable secure and efficient transaction processing.
- **Features:**
    - Integration with payment gateways (e.g., Razorpay, PayPal).
    - Flexible payment options (hourly, daily, subscription-based).
    - Generate and email payment receipts.
    - Support for discounts and promotional codes (if applicable).

5. **Navigation Module**

- **Purpose:** Guide users to their allocated parking slot.
- **Features:**
    - Integration with third-party mapping APIs (e.g., Google Maps).
    - Provide directions to parking lots and allocated slots.
    - Update routes in case of slot reassignment.

6. **Reports and Analytics Module**

- **Purpose:** Provide insights and reports for administrators.
- **Features:**

  - Generate reports on slot utilization, peak hours, and revenue.
  - Analyze booking trends and user behavior.
  - Provide environmental metrics if relevant (e.g., emissions data).

7. **Notification Module**

- **Purpose:** Keep users informed about their bookings and updates.
- **Features:**
  - Send real-time notifications for booking confirmations, cancellations, and slot availability.
  - Notify users about payment status and receipts.
  - Alert admins of any anomalies or operational issues.

8. **Admin Control Panel Module**

- **Purpose:** Enable administrators to manage the system effectively.
- **Features:**
  - View and manage user accounts.
  - Monitor and manage parking slots.
  - Generate and export detailed reports.
  - Oversee payment records and resolve disputes.

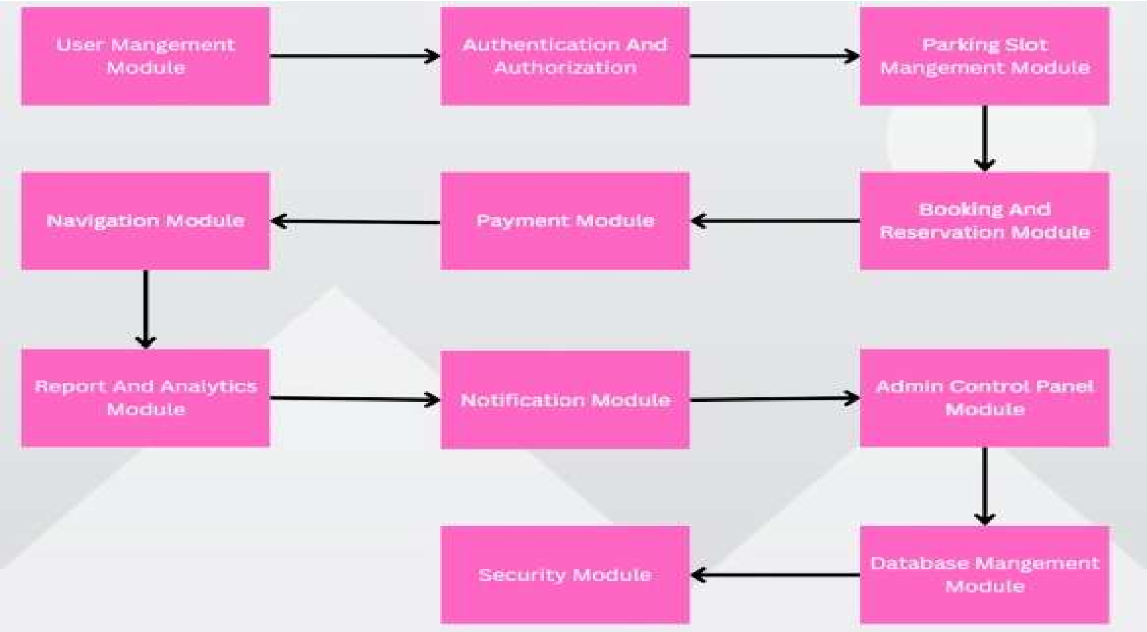
9. **Database Management Module**

- **Purpose:** Handle all data storage and retrieval operations.

- **Features:**
  - Store user information, booking details, payment history, and slot data.
  - Maintain data integrity and consistency.
  - Optimize queries for performance and scalability.

10. Security Module

- **Purpose:** Ensure secure operation and data handling.
- **Features:**
  - Encrypt sensitive data (e.g., passwords, payment information).
  - Implement multi-factor authentication for secure access.
  - Regular monitoring for unauthorized access or vulnerabilities.


User Mangement Module
Authentication And Authorization
Parking Slot Mangement Module
Navigation Module
Payment Module
Booking And Reservation Module
Report And Analytics Module
Notification Module
Admin Control Panel Module
Security Module
Database Mangement Module

Chapter 6
Implementation

6.1. System Development

The development of the Parking Management System is divided into distinct phases to ensure seamless progress and high-quality output.

1. **Planning and Requirement Analysis**

    - Identify user and system requirements based on stakeholder discussions.
    - Define project scope, features, and deliverables.
    - Prioritize key functionalities like slot booking, payment integration, and user management.

2. **System Design**

    - Create architectural designs and workflows for each module.
    - Use tools like UML diagrams to define relationships between system components.
    - Design an efficient database schema for storing user profiles, parking slots, and transaction records.

3. **Frontend Development**

    - Develop a user-friendly and responsive interface using **HTML**, **CSS**, and **JavaScript**.
    - Use frameworks like **React** or **Angular** to create an interactive web interface.
    - Ensure compatibility across multiple devices (PC, tablet, and mobile).

4. **Backend Development**

    - Use **Flask** as the backend framework to handle server-side operations, such as:

    - User authentication and role-based access.
    - Slot management and booking processing.
    - Integration with payment gateways for secure transactions.
- Develop RESTful APIs to enable communication between frontend and backend.

5. **Database Development**

- Use **MySQL** for database management.
- Create tables for:
    - User information (e.g., name, email, role).
    - Parking slot details (e.g., slot ID, status, type).
    - Booking records (e.g., booking ID, time, user ID).
    - Payment history.
- Optimize queries for fast data retrieval and updates.

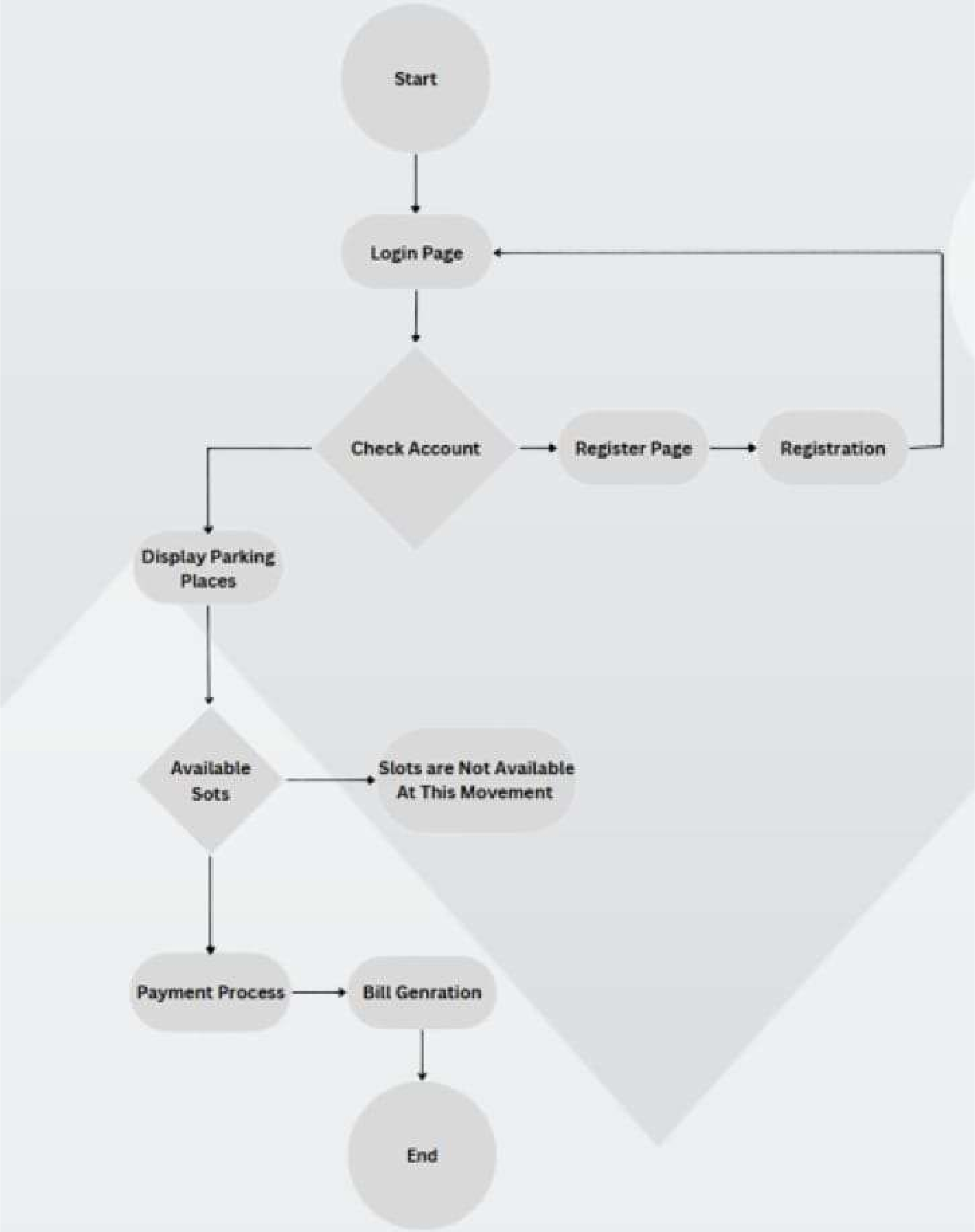
6. **Testing**

- **Unit Testing:** Verify individual modules (e.g., login, slot booking) for correctness.
- **Integration Testing:** Ensure smooth communication between modules (e.g., booking and payment).
- **System Testing:** Validate overall system functionality against requirements.
- **User Acceptance Testing (UAT):** Gather feedback from end-users and stakeholders to fine-tune the system.

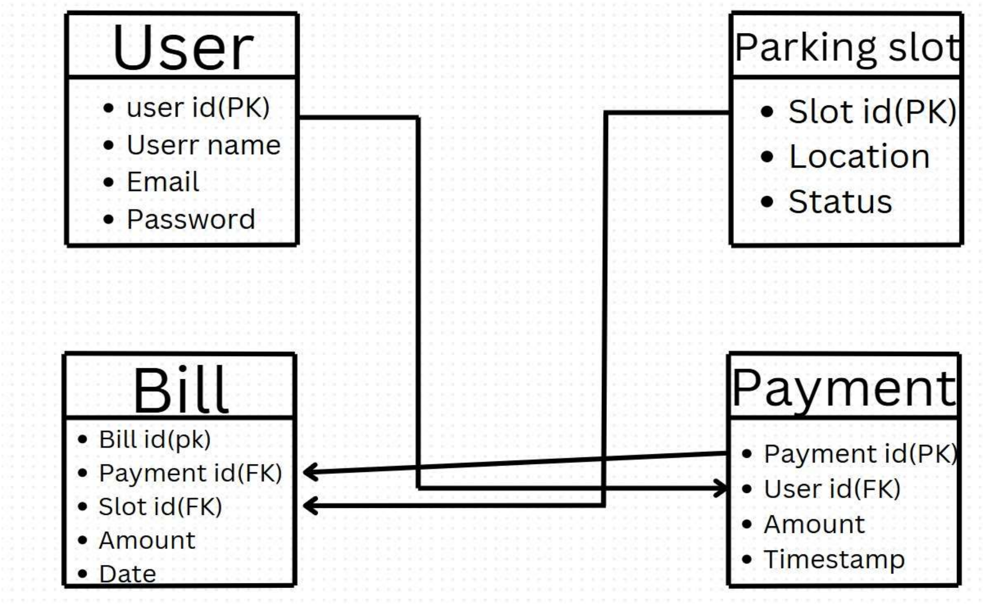
7. Deployment

- Host the system on a reliable web server or cloud platform like **AWS**, **Heroku**, or **Google Cloud**.
- Set up a domain and ensure HTTPS support for secure communication.
- Ensure cross-browser compatibility for better accessibility.

6.2 Work flow of the system:



6.3 ER-Diagram:



1. User:

- Represents the system's users.
- Attributes:
    - user_id (PK): A unique identifier for each user (Primary Key).
    - User name: The name of the user.
    - Email: The email address for communication and login purposes.
    - Password: The login credential for secure access.
- **Purpose:** Users interact with the system to book parking slots, make payments, and receive bills.

2. Parking Slot:

- Represents the parking spaces managed in the system.
- Attributes:
    - Slot_id (PK): A unique identifier for each parking slot (Primary Key).
    - Location: The physical location of the parking slot.
    - Status: Indicates whether the slot is occupied, available, or under maintenance.
- **Purpose:** To store and track the availability of parking slots.

3. Payment:

- Represents transactions made by users.
- Attributes:
    - Payment_id (PK): A unique identifier for each payment (Primary Key).
    - User_id (FK): A foreign key linking the payment to the user who made it.
    - Amount: The amount paid by the user.
    - Timestamp: The date and time when the payment was made.
- **Purpose:** To log and track financial transactions.

4. Bill:

- Represents a record of charges for parking services.
- Attributes:
    - Bill_id (PK): A unique identifier for each bill (Primary Key).

- Payment_id (FK): A foreign key linking the bill to the corresponding payment.
- Slot_id (FK): A foreign key linking the bill to the corresponding parking slot.
- Amount: The total charge for the parking service.
- Date: The date when the bill was generated.

- **Purpose:** To provide a detailed record of transactions and parking usage.

Relationships:

1. **User → Parking Slot:**

- **Relationship Type: One-to-Many.**
- Explanation: A single user can book multiple parking slots over time, but each parking slot is associated with only one user at a given time.

2. **User → Payment:**

- **Relationship Type: One-to-Many.**
- Explanation: A single user can make multiple payments, but each payment is linked to one user.

3. **Payment → Bill:**

- **Relationship Type: One-to-One.**
- Explanation: Each payment generates one bill, and each bill corresponds to a single payment.

4. **Parking Slot → Bill:**

- **Relationship Type: Many-to-One.**
- Explanation: Multiple parking slots can be billed in one transaction, but each bill is linked to one parking slot in this context.

6.3 System Workflow Explaination:

1. **Start:** The process begins.

2. **Login Page:** Users are prompted to login using their credentials. If they do not have an account, they are directed to the Register Page.

3. **Registration:** New users can create an account by providing necessary details. After successful registration, they are redirected to the login Page.

4. **Check Account:** The system verifies the user`s credentials. Upon successful login, user proceed to view options.

5. **Display Parking Places:** The system displays available parking places based on user location and preferences.

6. **Check Slot Availability:** If Available slots exist, the user can proceed to the payment process. If no slots are available, the system notifies the user that no slots are currently available.

7. **Payment Process:** Users pay for their reserved parking slot using integrated payment gateways.

8. **Bill Generation:** The system generates a bill or receipt for the transaction, which can be downloaded or emailed to the user.

9. **End:** The workflow completes, and the user successfully reserves their parling slot.

6.4 System connection Explaination:

The system connection in a Parking Management System outlines how different components interact with each other to ensure smooth functionality. Below is an overview of the system's architecture and connections:

1. Frontend and User Interaction

- **User Interface:**
  - Built using web technologies like HTML, CSS, and JavaScript.
  - Allows users to interact with the system for actions such as login, registration, parking slot selection, and payments.
- **Connection:**
  - Communicates with the backend via REST APIs for data retrieval and updates.

2. Backend and Application Logic

- **Server:**
  - Developed using Python Flask for handling user requests and processing system logic.
  - Manages user sessions, slot availability, and payment transactions.
- **Connection:**
  - Interfaces with the database to store and fetch data.
  - Sends responses to the frontend for display.

3. Database

- Database Management System (DBMS):
    - Uses MySQL for structured data storage.
    - **Tables include:**
        - User data (e.g., account details, preferences).
        - Parking slot data (e.g., availability, location).
        - Booking records (e.g., booking time, user ID).
        - Payment records (e.g., transaction ID, amount).
  - **Connection:**
    - The backend executes SQL queries to fetch or update information in the database.

4. Payment Gateway Integration

- **Third-Party API:**
  - Secure payment gateways like Phonepe, Gpay, Paytm are used for handling online payments.
- **Connection:**
  - Backend connects with the payment gateway using APIs to process payments securely and update transaction status.

5. Admin Panel

- **Administrator Interface:**
  - Allows administrators to manage parking slots, monitor usage, and generate reports.
- **Connection:**

- Interfaces with backend for updating data and generating analytics reports.

6.Notification System

- **Email and SMS Alerts:**
  - Used to notify users about bookings, payments, or slot availability.
- **Connection:**
  - Backend integrates with SMIP servers or third-party SMS APIs to send notifications

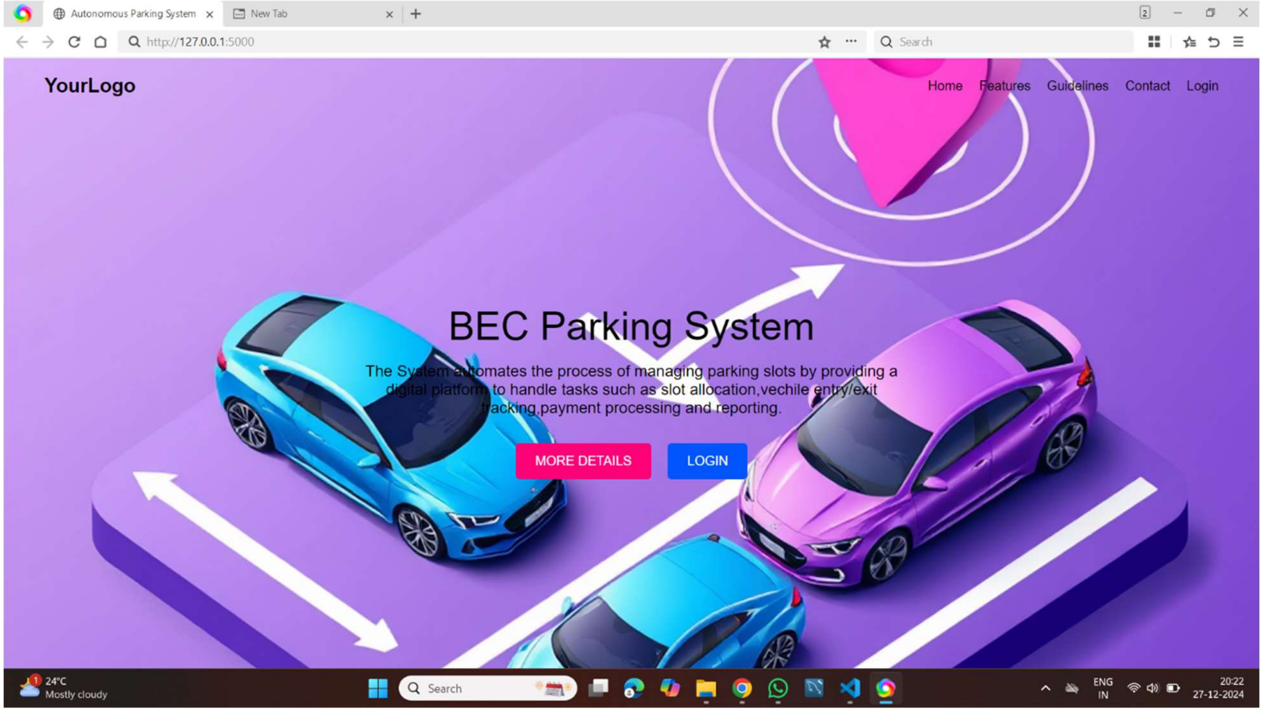
7.Security System

- **Authentication and Authorization:**
  - Role-based access control ensures secure login for users and administrators.
- **Connection:**
  - Uses secure protocols like HTTPS and encryption for data transmission.

➢ System Flow:

1. Frontend sends user requests to the backend.
2. Backend processes the request and queries the database.
3. Database retrieves or updates data and sends results to the backend.
4. Backend sends the response to the frontend or communicates with external systems (e.g., payment gateways, notification services).
5. Frontend displays results to the user.

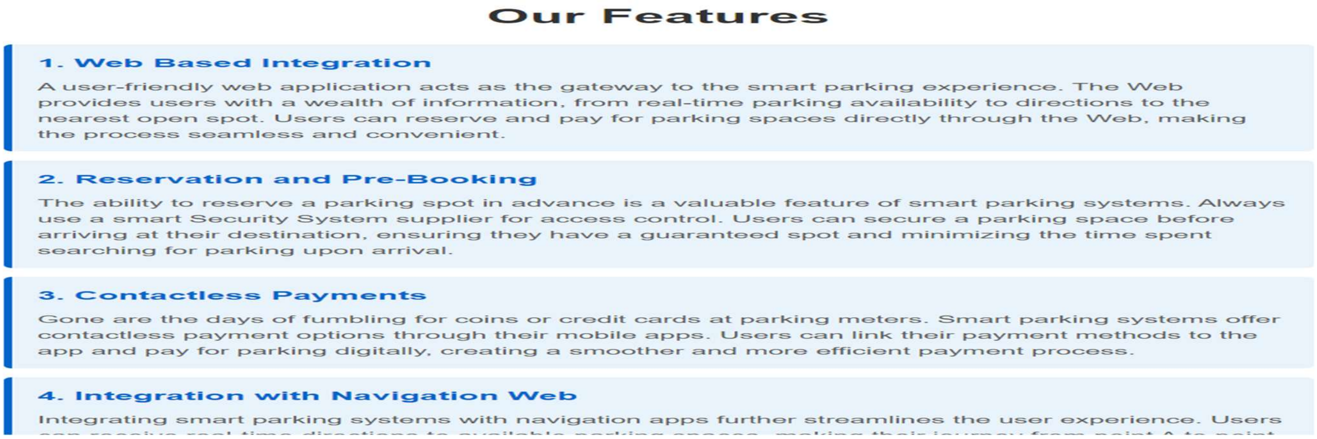
Chapter 7
Results



BEC Parking System

The System automates the process of managing parking slots by providing a digital platform to handle tasks such as slot allocation,vechile entry/exit tracking,payment processing and reporting.

Figure 7.1 Home Page



Our Features

1. Web Based Integration

A user-friendly web application acts as the gateway to the smart parking experience. The Web provides users with a wealth of information, from real-time parking availability to directions to the nearest open spot. Users can reserve and pay for parking spaces directly through the Web, making the process seamless and convenient.

2. Reservation and Pre-Booking

The ability to reserve a parking spot in advance is a valuable feature of smart parking systems. Always use a smart Security System supplier for access control. Users can secure a parking space before arriving at their destination, ensuring they have a guaranteed spot and minimizing the time spent searching for parking upon arrival.

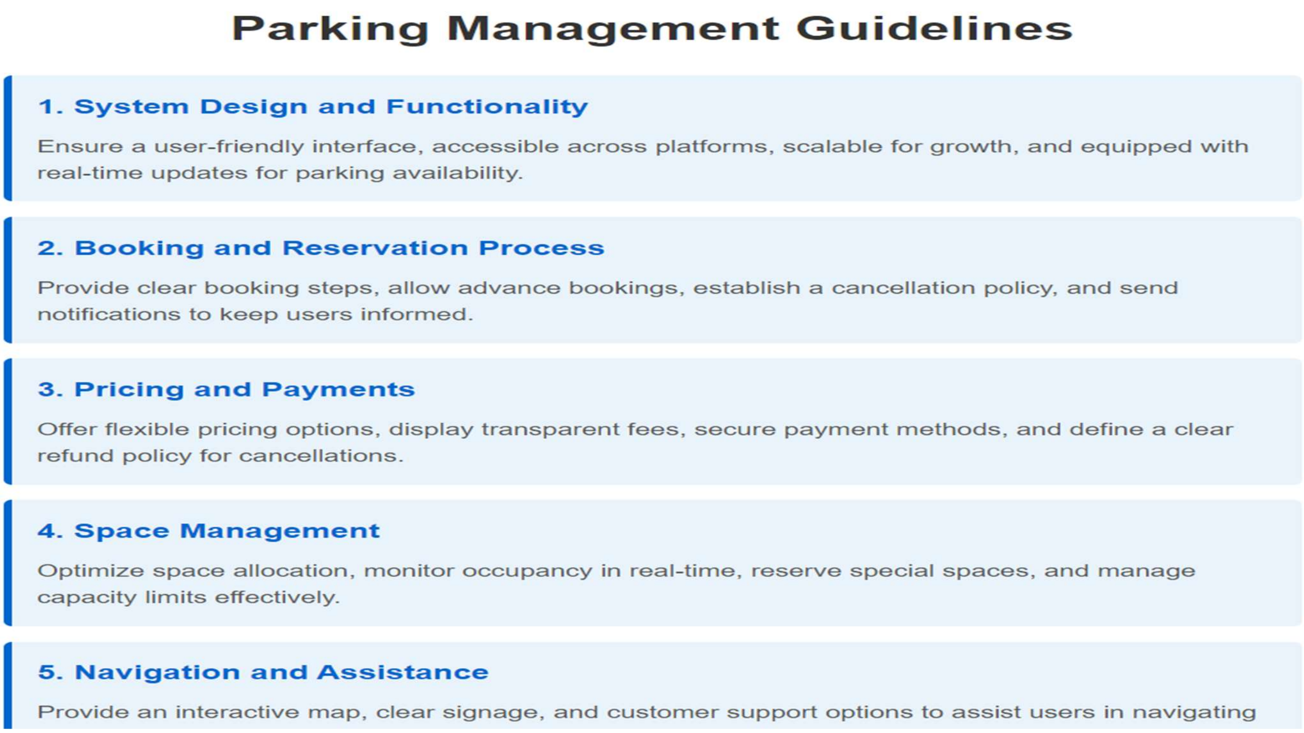
3. Contactless Payments

Gone are the days of fumbling for coins or credit cards at parking meters. Smart parking systems offer contactless payment options through their mobile apps. Users can link their payment methods to the app and pay for parking digitally, creating a smoother and more efficient payment process.

4. Integration with Navigation Web

Integrating smart parking systems with navigation apps further streamlines the user experience. Users

Figure 7.2 Features



Parking Management Guidelines

1. System Design and Functionality

Ensure a user-friendly interface, accessible across platforms, scalable for growth, and equipped with real-time updates for parking availability.

2. Booking and Reservation Process

Provide clear booking steps, allow advance bookings, establish a cancellation policy, and send notifications to keep users informed.

3. Pricing and Payments

Offer flexible pricing options, display transparent fees, secure payment methods, and define a clear refund policy for cancellations.

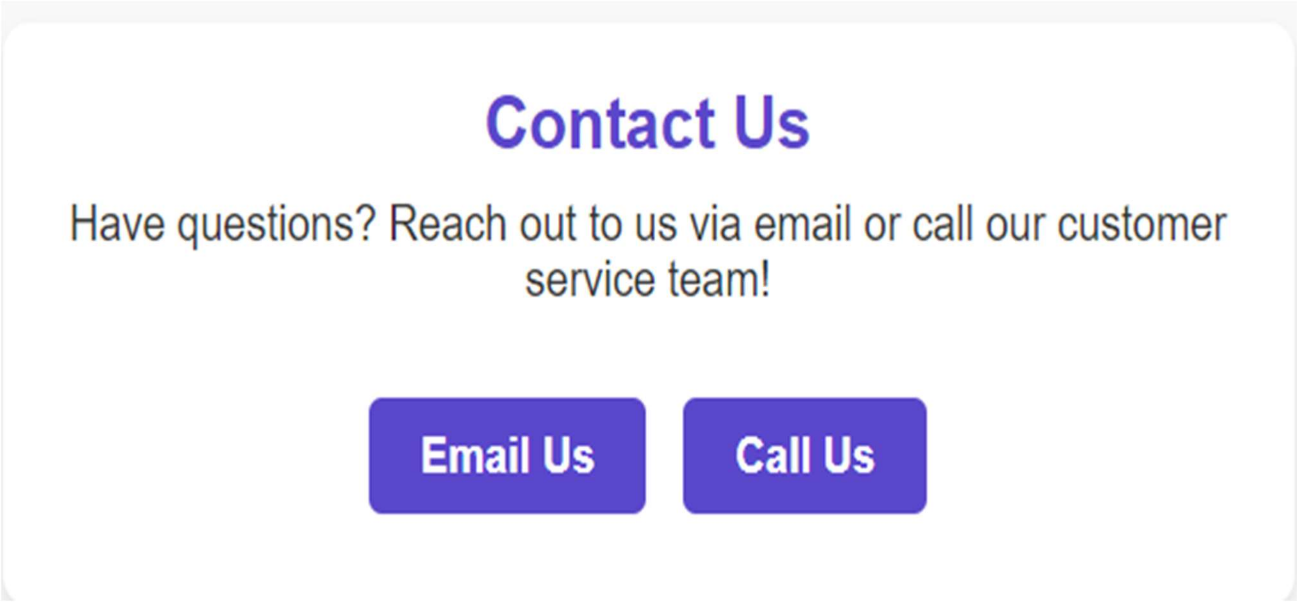
4. Space Management

Optimize space allocation, monitor occupancy in real-time, reserve special spaces, and manage capacity limits effectively.

5. Navigation and Assistance

Provide an interactive map, clear signage, and customer support options to assist users in navigating

Figure 7.3 Guidelines



Contact Us

Have questions? Reach out to us via email or call our customer service team!

Figure 7.4 Contact Us

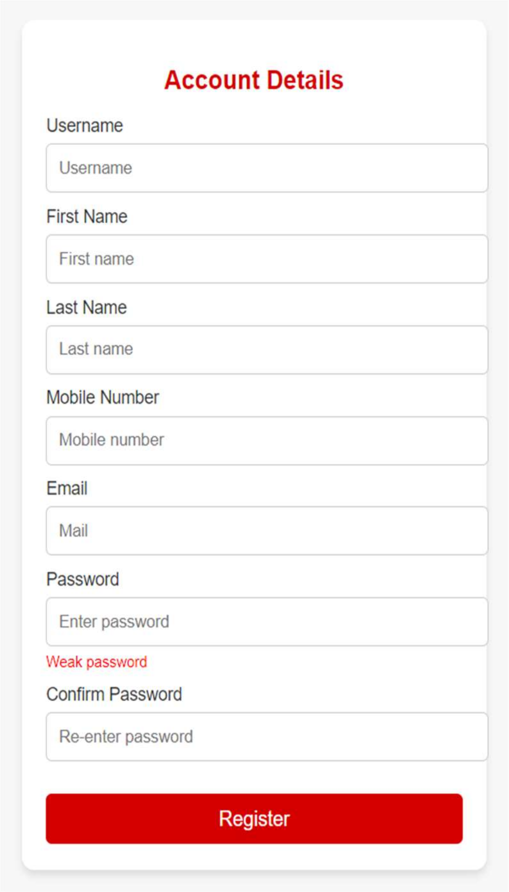


Figure 7.5 Registration

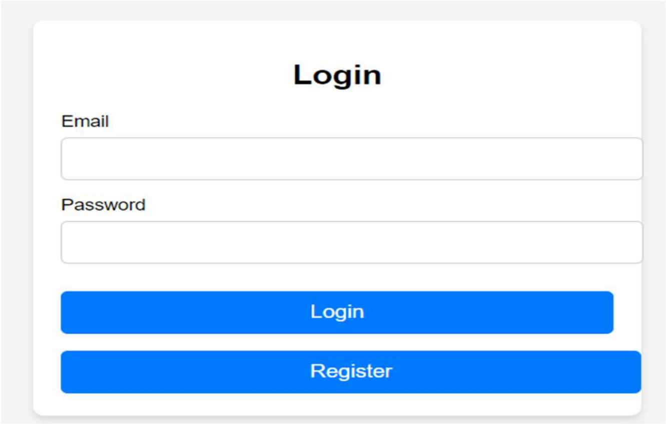
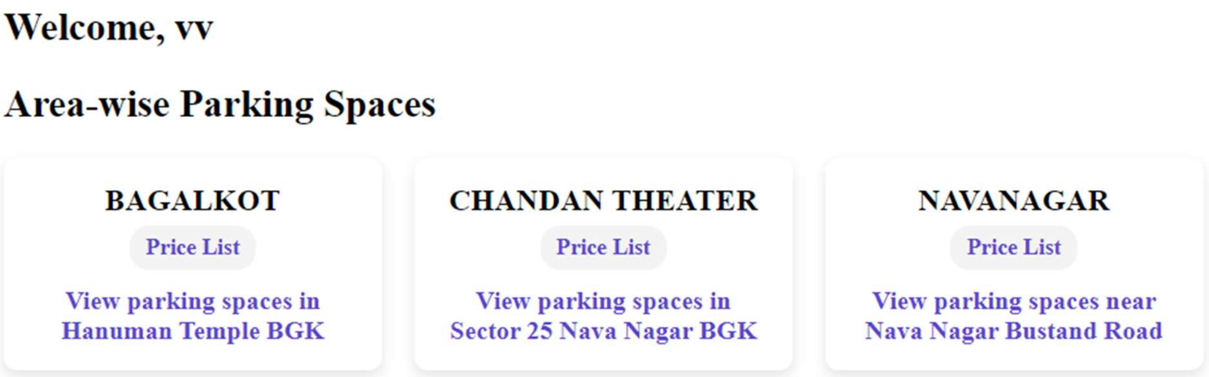


Figure 7.6 Login



Welcome, vv

Area-wise Parking Spaces

BAGALKOT

Price List

View parking spaces in Hanuman Temple BGK

CHANDAN THEATER

Price List

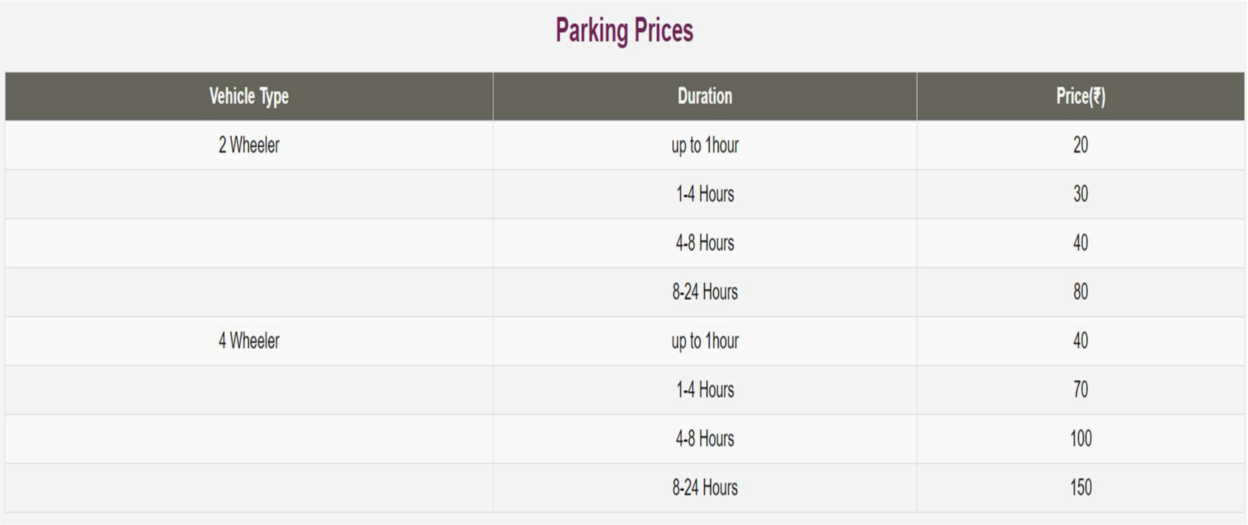
View parking spaces in Sector 25 Nava Nagar BGK

NAVANAGAR

Price List

View parking spaces near Nava Nagar Bustand Road

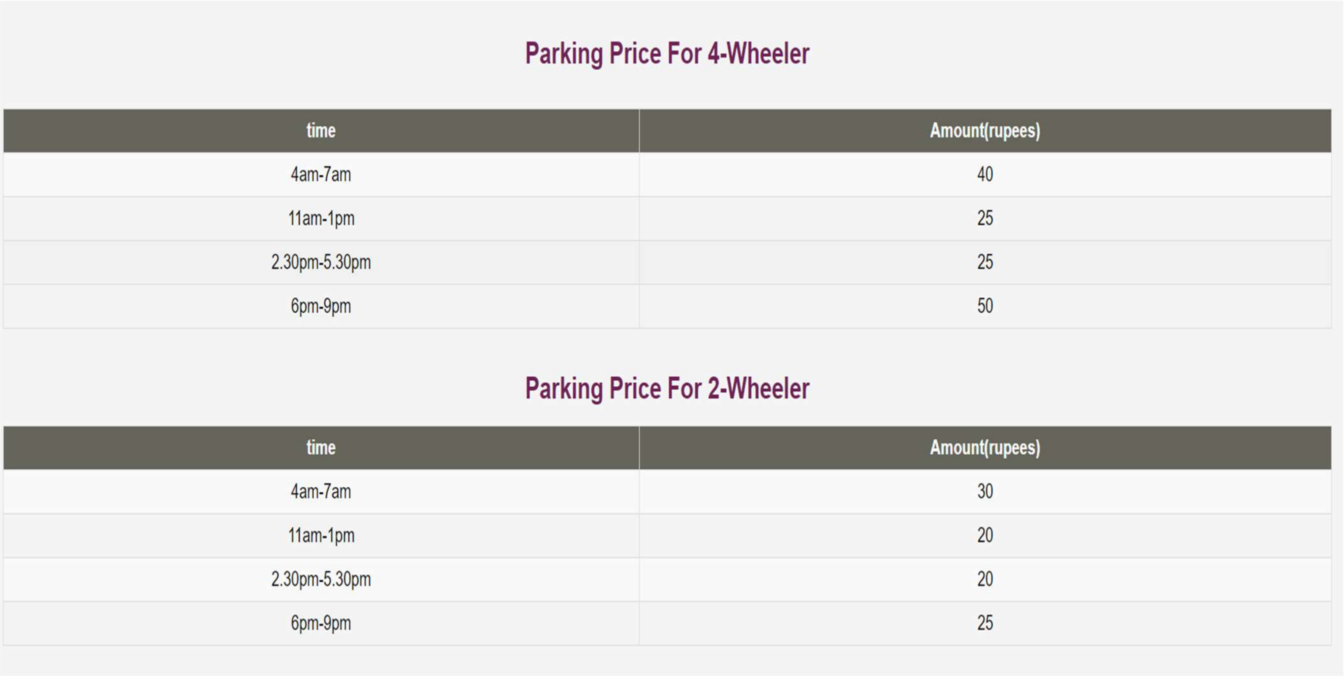
Figure 7.7 Area-wise parking spaces



Parking Prices

Vehicle Type	Duration	Price(₹)
2 Wheeler	up to 1hour	20
	1-4 Hours	30
	4-8 Hours	40
	8-24 Hours	80
4 Wheeler	up to 1hour	40
	1-4 Hours	70
	4-8 Hours	100
	8-24 Hours	150

Figure 7.8 Parking prices list for Bagalkot locations



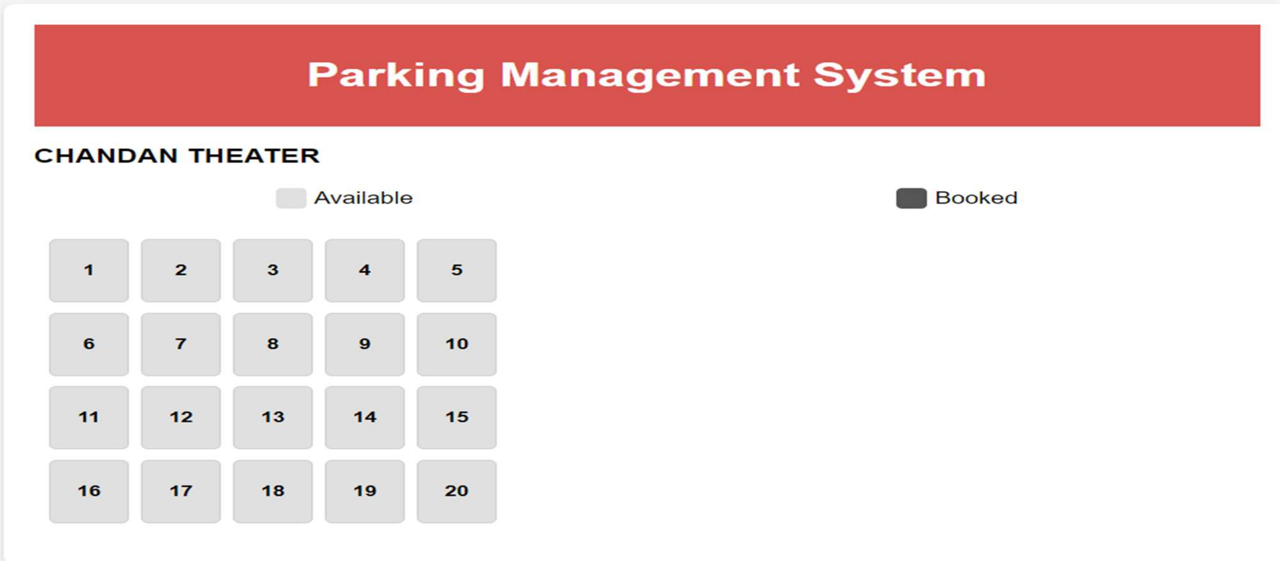
Parking Price For 4-Wheeler

time	Amount(rupees)
4am-7am	40
11am-1pm	25
2.30pm-5.30pm	25
6pm-9pm	50

Parking Price For 2-Wheeler

time	Amount(rupees)
4am-7am	30
11am-1pm	20
2.30pm-5.30pm	20
6pm-9pm	25

Figure 7.9 Parking prices list for Chandan Theater locations



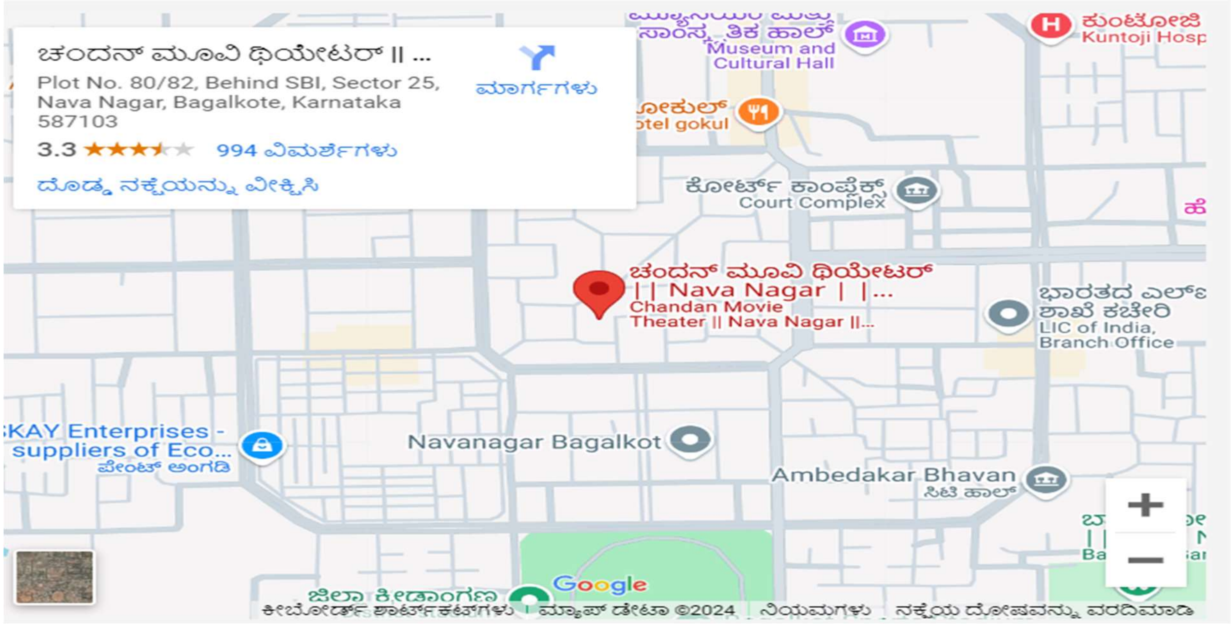
Parking Management System

CHANDAN THEATER

□ Available ■ Booked

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20

Figure 7.10 parking slots



ಚಂದನ್ ಮೂವಿ ಥಿಯೇಟರ್ || ...

Plot No. 80/82, Behind SBI, Sector 25, Nava Nagar, Bagalkote, Karnataka 587103

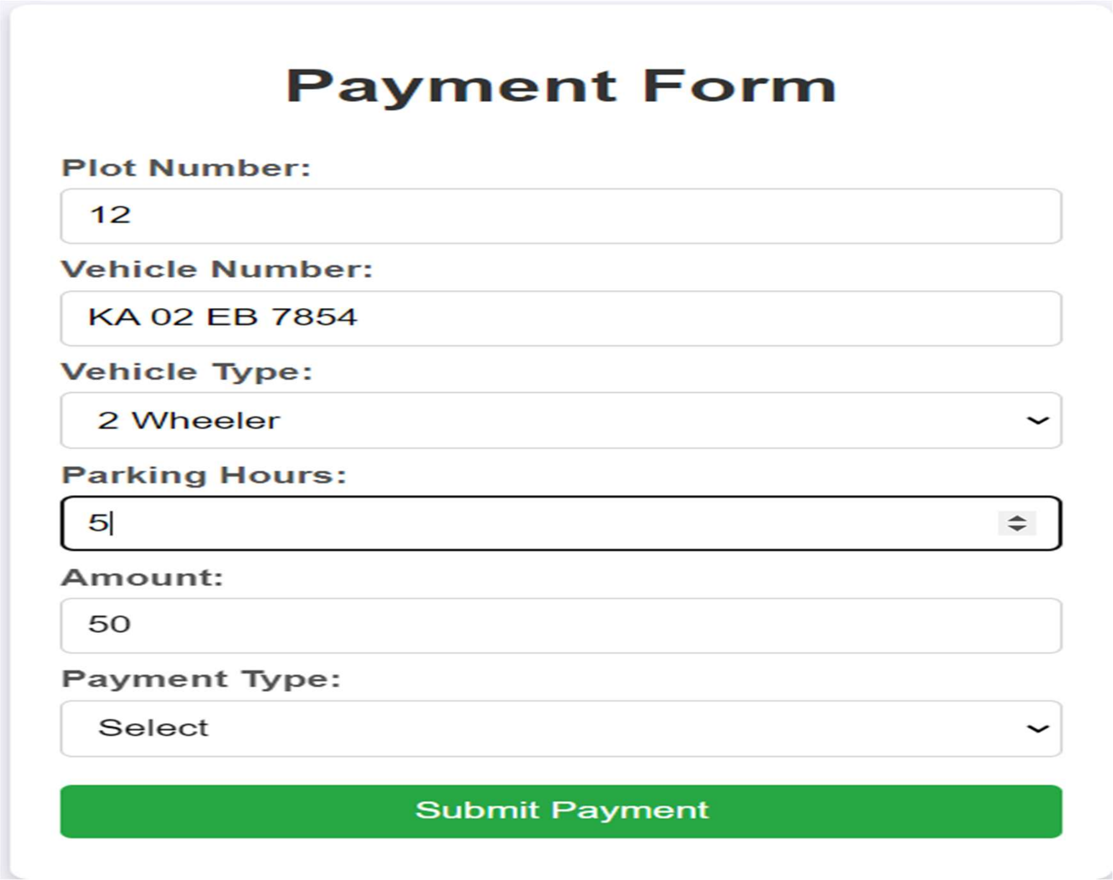
3.3 ★★★★☆ 994 ವಿಮರ್ಶೆಗಳು

ದೊಡ್ಡ ನಕ್ಷೆಯನ್ನು ವೀಕ್ಷಿಸಿ

ಮಾರ್ಗಗಳು

Museum and Cultural Hall
Kuntoji Hosp
otel gokul
ಕೋರ್ಟ್ ಕಾಂಪ್ಲೆಕ್ಸ್
Court Complex
ಚಂದನ್ ಮೂವಿ ಥಿಯೇಟರ್ || Nava Nagar | |...
Chandan Movie Theater || Nava Nagar ||...
ಭಾರತದ ಎಲ್‌ಐ ಶಾಖೆ ಕಚೇರಿ
LIC of India, Branch Office
KAY Enterprises - suppliers of Eco...
ಪೇಂಟ್ ಅಂಗಡಿ
Navanagar Bagalkot
Ambedakar Bhavan
ಸಿಟಿ ಹಾಲ್
ಜಿಲ್ಲಾ ಕ್ರೀಡಾಂಗಣ
Google
ಕೀಬೋರ್ಡ್ ಶಾರ್ಟ್‌ಕಟ್‌ಗಳು | ಮ್ಯಾಪ್ ಡೇಟಾ ©2024 | ನಿಯಮಗಳು | ನಕ್ಷೆಯ ದೋಷವನ್ನು ವರದಿಮಾಡಿ
+
−

Figure 7.11 Map For A Particular Locations



Payment Form

Plot Number: 12

Vehicle Number: KA 02 EB 7854

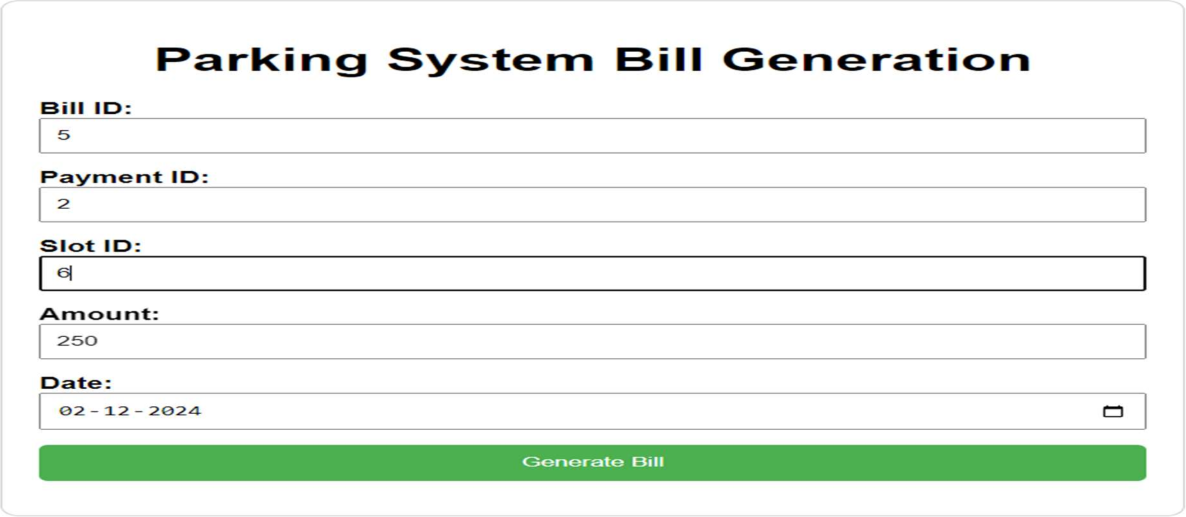
Vehicle Type: 2 Wheeler

Parking Hours: 5

Amount: 50

Payment Type: Select

Figure 7.12 Payment Form



Parking System Bill Generation

Bill ID: 5

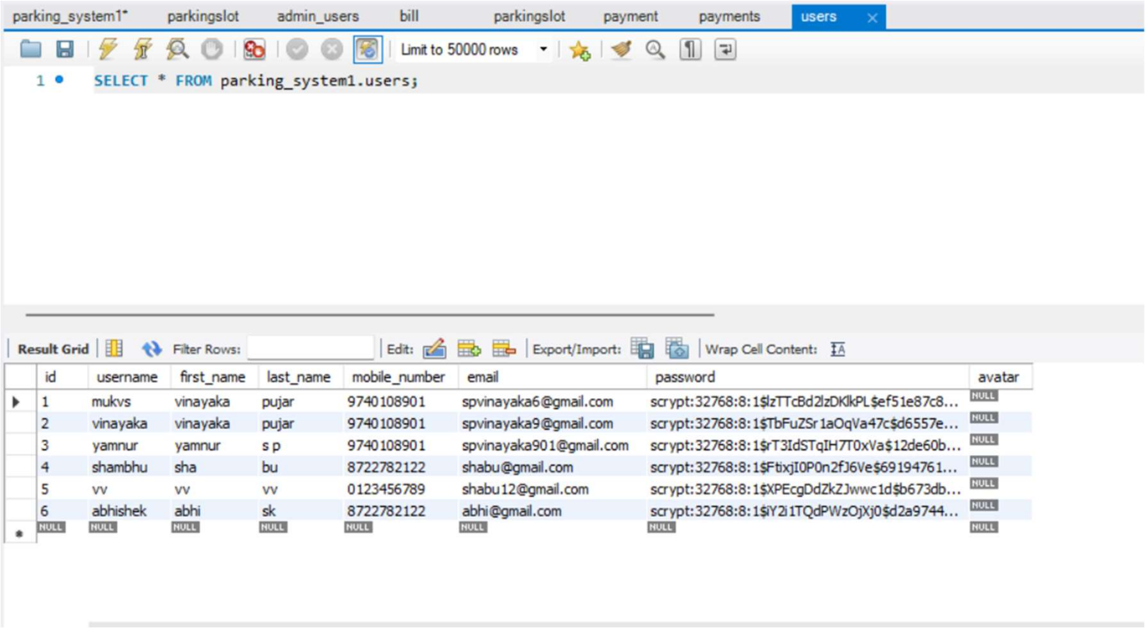
Payment ID: 2

Slot ID: 6

Amount: 250

Date: 02-12-2024

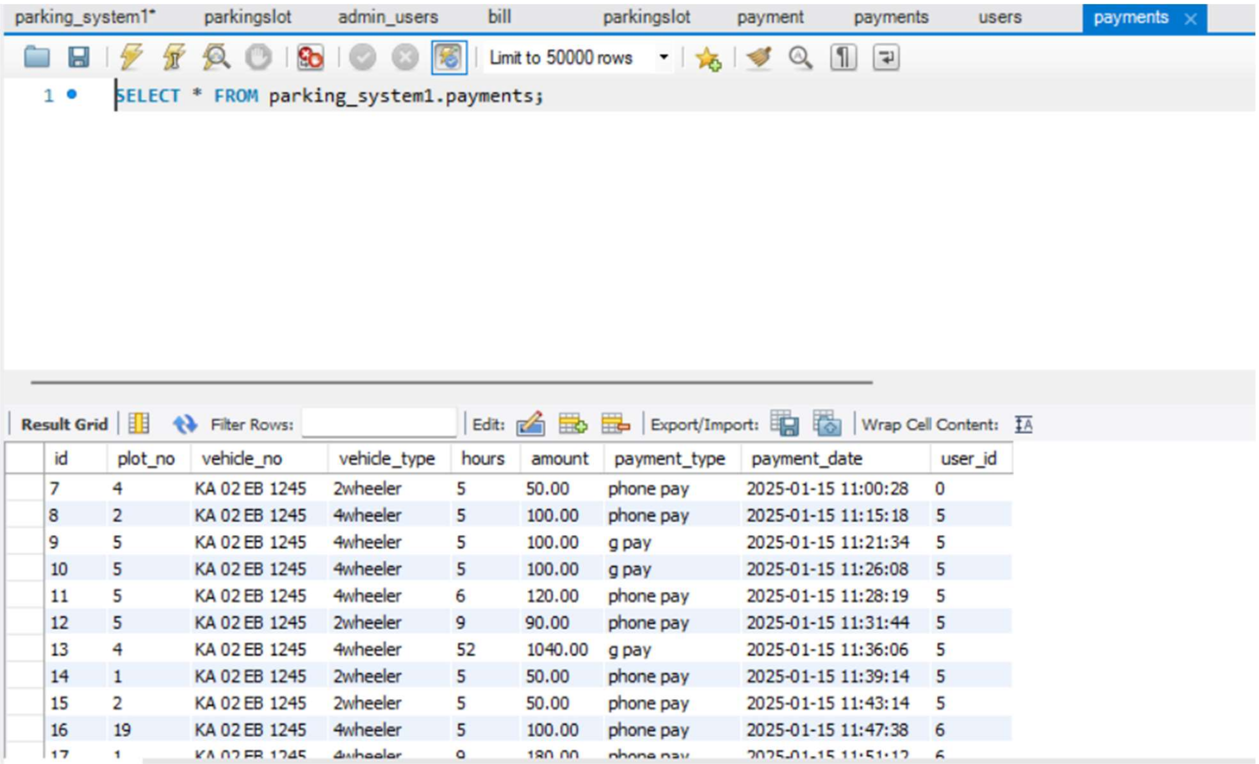
Figure 7.13 Bill Generation



```
1 • SELECT * FROM parking_system1.users;
```

id	username	first_name	last_name	mobile_number	email	password	avatar
1	mukvs	vinayaka	pujar	9740108901	spvinayaka6@gmail.com	scrypt:32768:8:1$lzTTcBd2lzDKlkPL$ef51e87c8...	NULL
2	vinayaka	vinayaka	pujar	9740108901	spvinayaka9@gmail.com	scrypt:32768:8:1$TbFuZSr1aOqVa47c$d6557e...	NULL
3	yamnur	yamnur	s p	9740108901	spvinayaka901@gmail.com	scrypt:32768:8:1$rT3IdSTqIH7T0xVa$12de60b...	NULL
4	shambhu	sha	bu	8722782122	shabu@gmail.com	scrypt:32768:8:1$FtixjI0P0n2fJ6Ve$69194761...	NULL
5	vv	vv	vv	0123456789	shabu12@gmail.com	scrypt:32768:8:1$XPEcgDdZkZJwwc1d$b673db...	NULL
6	abhishek	abhi	sk	8722782122	abhi@gmail.com	scrypt:32768:8:1$iY2i1TQdPWzOjXj0$d2a9744...	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 7.14 Users Store Data In MYSQLDB



```
1 • SELECT * FROM parking_system1.payments;
```

id	plot_no	vehicle_no	vehicle_type	hours	amount	payment_type	payment_date	user_id
7	4	KA 02 EB 1245	2wheeler	5	50.00	phone pay	2025-01-15 11:00:28	0
8	2	KA 02 EB 1245	4wheeler	5	100.00	phone pay	2025-01-15 11:15:18	5
9	5	KA 02 EB 1245	4wheeler	5	100.00	g pay	2025-01-15 11:21:34	5
10	5	KA 02 EB 1245	4wheeler	5	100.00	g pay	2025-01-15 11:26:08	5
11	5	KA 02 EB 1245	4wheeler	6	120.00	phone pay	2025-01-15 11:28:19	5
12	5	KA 02 EB 1245	2wheeler	9	90.00	phone pay	2025-01-15 11:31:44	5
13	4	KA 02 EB 1245	4wheeler	52	1040.00	g pay	2025-01-15 11:36:06	5
14	1	KA 02 EB 1245	2wheeler	5	50.00	phone pay	2025-01-15 11:39:14	5
15	2	KA 02 EB 1245	2wheeler	5	50.00	phone pay	2025-01-15 11:43:14	5
16	19	KA 02 EB 1245	4wheeler	5	100.00	phone pay	2025-01-15 11:47:38	6
17	1	KA 02 EB 1245	4wheeler	9	180.00	phone pay	2025-01-15 11:51:12	6

Figure 7.14 Payment Details Store Data In MYSQLDB

Chapter 8

Conclusion

The **Parking Management System** serves as a comprehensive platform to address the challenges of modern parking facilities. It combines efficiency, user convenience, and technological advancements to streamline parking operations and enhance user satisfaction. Below are the key takeaways:

1. **Operational Efficiency:** The system simplifies and optimizes the management of parking slots, reducing manual errors and ensuring better utilization of resources.
2. **User-Friendly Design:** With its intuitive interface, the system provides a seamless experience for both administrators and users, enabling quick registration, slot booking, and payment processing.
3. **Automation and Transparency:** Automated processes such as slot allocation, dynamic pricing, and bill generation increase transparency and save time for users and administrators alike.
4. **Secure Transactions:** The integration of secure payment methods ensures data safety and fosters trust among users.
5. **Scalability and Flexibility:** The system is scalable to accommodate various types of parking lots and is adaptable for expansion across multiple locations.
6. **Environmental Impact:** By reducing the time spent searching for parking, the system contributes to decreasing traffic congestion and vehicle emissions, aligning with sustainability goals.
7. **Administrative Control:** The admin dashboard provides real-time insights, detailed reports, and effective user and slot management capabilities, empowering administrators to make informed decisions.
8. **Error Handling and Robustness:** The system is equipped with robust error-handling mechanisms and logging capabilities to ensure high reliability and easy troubleshooting.

Future Scope

The Parking Management System has immense potential for future enhancements, such as:

- **Mobile App Development** for easier user access.
- **IoT and AI Integration** to enable real-time parking updates and predictive analytics.
- **Vehicle Authentication Systems** for improved security.
- **Advanced Reporting Features** for data-driven decision-making.

References:

Research Papers and Journals:

1). **Rishabh Vantagudi**, **Samaya Pillai**, **Dr Pankaj Pathak** and **Vikas Yadav** Efficient *Parking Management*. 04 Jun 2024

2). **Mohamed Khamis Hassan** . A SMART ANDROID *PARKING SYSTEM*. 28 Aug 2024

3). **Amira A. Elsonbaty.** The Smart *Parking Management System*. 17 Nov 2020

4). **Deepak Divakar, Saurav Kumar Singh** and **Fedina Dkhar. Automatic** Car *Parking Management System*. 13 Jul 2022