Parking Management System

N. Kishore[1], MR. D. Joseph Pushpaj[2]
[1]*M.E (Computer Science and Engineering), PSN College of Engineering and Technology (Autonomous) Melathediyoor,Tirunelveli 627 152*
[2]*Guide, M.E (Computer Science and Engineering), PSN College of Engineering and Technology (Autonomous) Melathediyoor,Tirunelveli 627 152*

***Abstract*—Parking management has become a significant challenge in urban areas due to the rapid increase in the number of vehicles and limited parking spaces. Efficient monitoring and management of parking facilities are essential to reduce traffic congestion, fuel consumption, and time wastage. This project focuses on the development of a smart Parking Management System using modern web technologies, specifically Angular 17 for the frontend. The proposed system provides a user-friendly interface that allows users to view real-time parking availability, reserve parking slots, and manage vehicle entries and exits efficiently. The system is designed to handle parking data such as slot availability, booking status, vehicle details, and time records. It incorporates features like dynamic slot allocation, automated updates, and responsive design to enhance user experience. The application is developed using Angular 17, along with supporting technologies such as TypeScript, HTML, CSS, and RESTful APIs for seamless data communication. Data handling techniques and efficient state management are implemented to ensure smooth performance and scalability. The system was tested under various scenarios to evaluate its reliability and responsiveness. The results demonstrate that the application effectively reduces manual effort, improves parking space utilization, and minimizes traffic issues within parking areas. The developed system can assist parking authorities, commercial complexes, and smart city initiatives in managing parking operations efficiently. Future enhancements may include integration with IoT sensors for real-time slot detection, mobile application support, and online payment systems for a fully automated parking experience.**

I. INTRODUCTION

Parking management has emerged as a critical issue in modern urban environments due to the rapid growth in the number of vehicles and the limited availability of parking infrastructure. As cities continue to expand and populations increase, the demand for efficient parking solutions has become more pressing than ever. Inadequate parking facilities not only cause inconvenience to drivers but also contribute significantly to traffic congestion, increased fuel consumption, and environmental pollution. Drivers often spend a considerable amount of time searching for available parking spaces, which leads to unnecessary delays and frustration. Traditional parking systems are largely manual and lack the integration of modern technologies. These systems do not provide real-time information about parking availability, resulting in inefficient utilization of space. Furthermore, the absence of automation makes the process time-consuming and prone to human error. Issues such as improper space allocation, lack of monitoring, and difficulty in tracking vehicle entry and exit further reduce the effectiveness of conventional parking systems. To address these challenges, the concept of a Smart Parking Management System has gained significant attention. A smart system leverages modern web technologies, real-time data processing, and user-friendly interfaces to provide an efficient and seamless parking experience. This project focuses on the development of a Smart Parking Management System using Angular 17, a powerful front-end framework that enables the creation of dynamic and responsive web applications. The proposed system aims to automate the parking process by providing real-time updates on parking space availability, enabling users to easily locate and reserve parking slots. It also incorporates features such as vehicle tracking, digital records management, and efficient space allocation. By utilizing a modern framework like Angular 17, the system ensures scalability,

maintainability, and an enhanced user experience through intuitive design and fast performance. This chapter introduces the fundamental concepts behind the Smart Parking Management System and emphasizes the need for adopting automated solutions in today's urban landscape. It outlines the key objectives of the project, including reducing traffic congestion, optimizing space utilization, minimizing human intervention, and improving overall efficiency. Additionally, the chapter presents the problem statement, describing the limitations of existing systems, and explains the proposed approach to overcome these issues using modern technologies. Finally, the chapter provides an overview of the project structure, giving insight into how the system is organized and implemented. It sets the foundation for the subsequent chapters, which will delve deeper into system design, development methodology, implementation details, and evaluation of the proposed solution.

II. OBJECTIVES

The main objectives of this project are to design and develop a smart parking system using Angular 17 that delivers an efficient and modern solution for parking management. The system aims to provide real-time information about parking availability, enabling users to quickly identify and reserve available slots. By doing so, it seeks to reduce traffic congestion and minimize the time wasted in searching for parking spaces. Another important objective is to optimize the utilization of available parking areas through intelligent allocation and management. Additionally, the project focuses on creating a scalable and user-friendly web application that ensures ease of use, accessibility, and adaptability to future enhancements.

This project focuses on developing a smart Parking Management System that simplifies and modernizes parking operations through automation and real-time monitoring. The system is built using Angular 17 for the frontend, providing a dynamic and responsive user interface, and it integrates seamlessly with backend services for efficient data storage, processing, and management. By combining these technologies, the system ensures accurate and up-to-date information is always available to both users and administrators. Users can interact with the system through an intuitive web interface that allows them to check parking slot availability in real time, book parking spaces in advance, and manage their parking activities such as viewing booking history or canceling reservations. The system also supports automated processes like slot allocation and vehicle entry and exit tracking, reducing the need for manual intervention and minimizing errors. This leads to improved operational efficiency and a more convenient user experience. Furthermore, the project highlights the practical application of modern web technologies in addressing real-world urban challenges. By reducing the time spent searching for parking and optimizing space utilization, the system contributes to decreased traffic congestion and fuel consumption. Overall, the project demonstrates how intelligent, technology-driven solutions can play a vital role in the development of smart cities, enhancing both infrastructure efficiency and quality of life for users.

III. LITERATURE SURVEY

1. The high cost of free parking Authors: Donald Shoup

This study discusses the major drawbacks of traditional parking systems in urban environments. It highlights how poorly managed parking spaces lead to excessive traffic congestion, increased fuel consumption, and environmental pollution. The author explains that drivers spend a significant amount of time searching for parking, which contributes to inefficiency in transportation systems. The paper emphasizes the need for structured parking policies and better utilization of available resources.

2. Parking Management and Supply Optimization in Urban Areas Authors: Arnott, R., & Inci, E.

This paper focuses on optimizing parking supply in cities. It explains that traditional parking systems fail to balance demand and supply effectively, leading to overcrowding in certain areas and underutilization in others. The study suggests dynamic pricing and better allocation strategies to improve parking efficiency.

3. A New Smart Parking System Based on Optimal Resource Allocation Authors: Y. Geng and C. G. Cassandras

This research introduces a smart parking system that uses real-time data and reservation mechanisms. The system allows users to reserve parking slots in advance, reducing uncertainty and waiting time. The study shows that optimal allocation algorithms improve overall efficiency and reduce congestion in busy areas.

4. Smart Parking Systems and Sensors: A Survey Authors: G. Revathi and V. R. S. Dhulipala

This survey provides an overview of technologies used in smart parking systems, such as IoT sensors, wireless communication, and cloud computing. It explains how sensors detect vehicle presence and update the system in real time. The paper also discusses how these technologies improve accuracy and reduce human intervention.

5. Smart Parking Service Based on Wireless Sensor Networks Authors: J. Yang, J. Portilla, and T. Riesgo

This paper presents a system that uses wireless sensor networks to monitor parking spaces. Sensors installed in parking slots send occupancy data to a central server, which is then accessed by users through applications. The system improves reliability and provides real-time updates, making parking management more efficient.

6. IoT-Based Smart Parking System Authors: A. Khanna and R. Anand

This study explores the use of Internet of Things (IoT) in parking systems. It explains how interconnected devices collect and share data to provide real-time parking information. The system improves user convenience by allowing remote access to parking data and supports automation in parking operations.

7. Real-Time Smart Parking System Using IoT Authors: Y. Zhang et al.

This paper describes a real-time smart parking system that integrates IoT with web technologies. It demonstrates how users can view parking availability and reserve slots through a web interface. The study shows that real-time systems significantly reduce traffic congestion and improve space utilization.

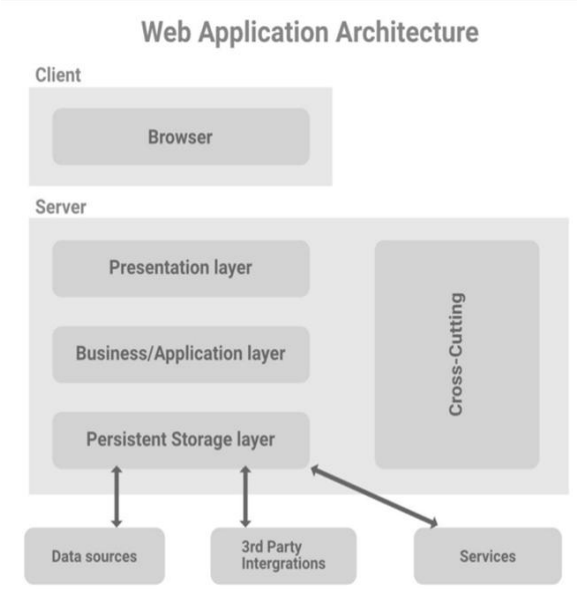
8. A Survey of Smart Parking Solutions Authors: T. Lin, H. Rivano, and F. Le Mouël

This paper evaluates different smart parking systems and identifies their limitations. It highlights issues such as high installation costs, dependency on sensors, and maintenance challenges. The study also points out that system failures can occur due to network disruptions.

IV. SYSTEM DESIGN

Architectural Design

The Smart Parking Management System is designed using a three-tier architecture, which ensures clear separation of concerns, improved scalability, and ease of maintenance. This architecture is divided into three main layers: the presentation layer, the application layer, and the data layer. The presentation layer is developed using Angular 17 and is responsible for handling user interactions through a dynamic and responsive interface. It allows users to perform actions such as checking parking availability, booking slots, and managing their reservations.



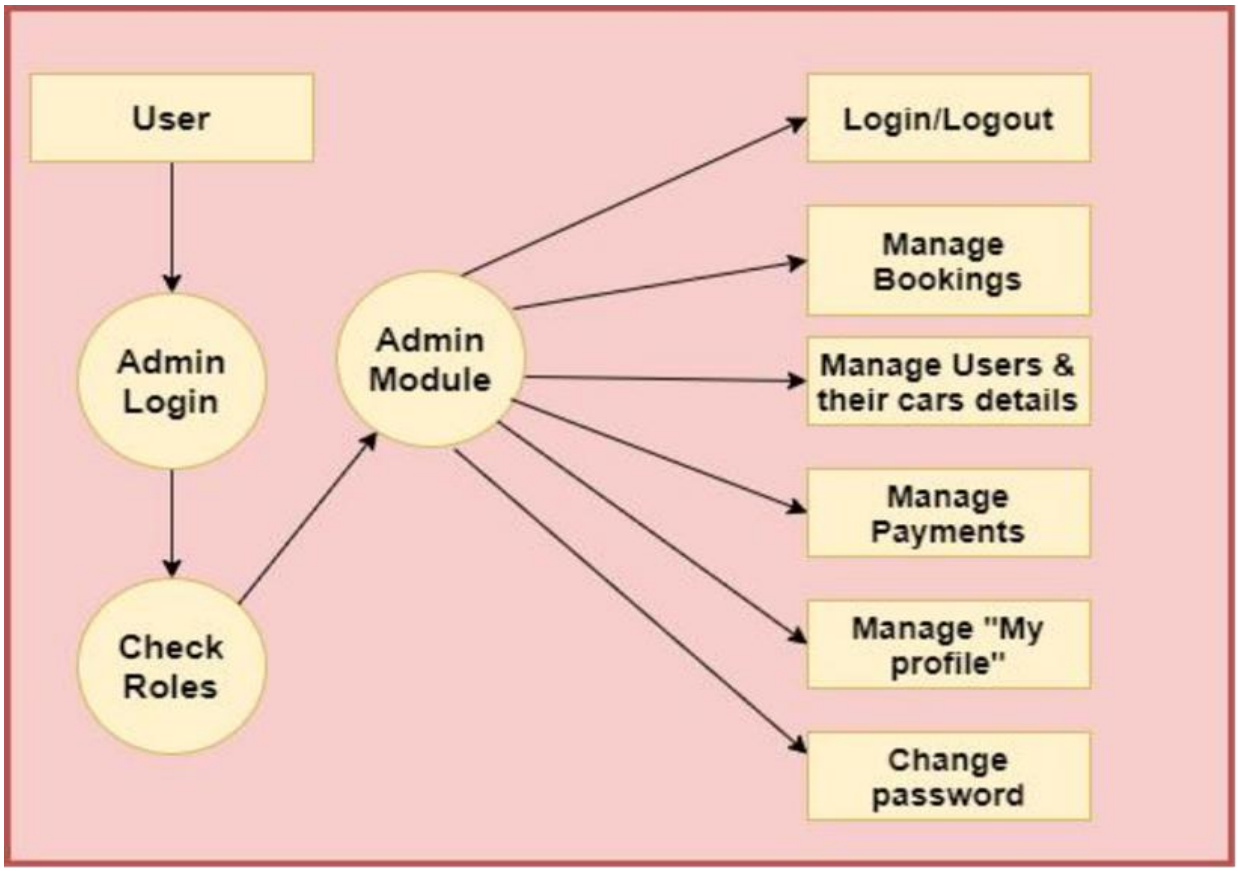
4.1 Architectural Design

The application layer acts as the core of the system, where all business logic is implemented. It processes user requests, validates data, and communicates with the database through RESTful APIs. This layer ensures that all operations, such as booking and

slot allocation, are handled efficiently and securely. The data layer is responsible for storing all system-related information, including user details, parking slot data, and booking records, using databases such as MySQL or MongoDB. This layered approach enhances system performance, enables independent development of components, and ensures secure and efficient data handling.

Data Flow Diagram (DFD)

The Data Flow Diagram (DFD) represents how data flows within the Smart Parking Management System and how different components interact with each other. At the highest level, known as the Level 0 or context diagram, the system is represented as a single process that interacts with external entities such as users and administrators.

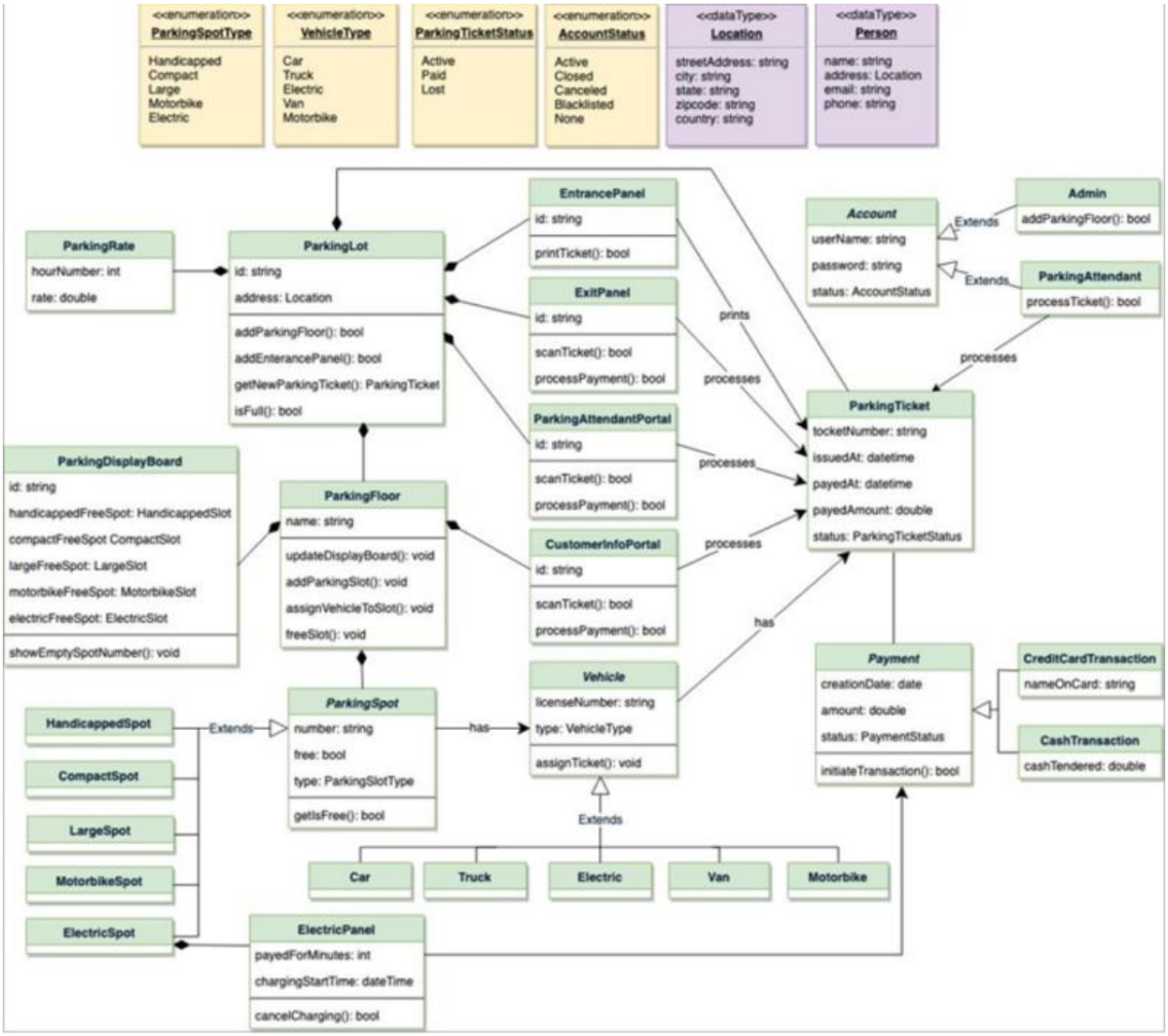


4.2 DFD diagram

Users provide inputs such as login credentials and booking requests, while the system responds with outputs such as parking availability and booking confirmations. At Level 1, the system is broken down into multiple processes, including user authentication, parking slot management, booking operations, and database interaction. These processes handle tasks such as validating user credentials, updating slot availability, and storing booking information. Data flows between these processes and data stores, ensuring smooth communication and efficient operation. The DFD helps in understanding how information is processed, stored, and transferred within the system, making it an essential tool for system design and analysis.

Class Diagram

The Class Diagram provides a structural representation of the Smart Parking Management System by illustrating the various classes, their attributes, methods, and relationships. The system consists of key classes such as User, ParkingSlot, Booking, and Admin.

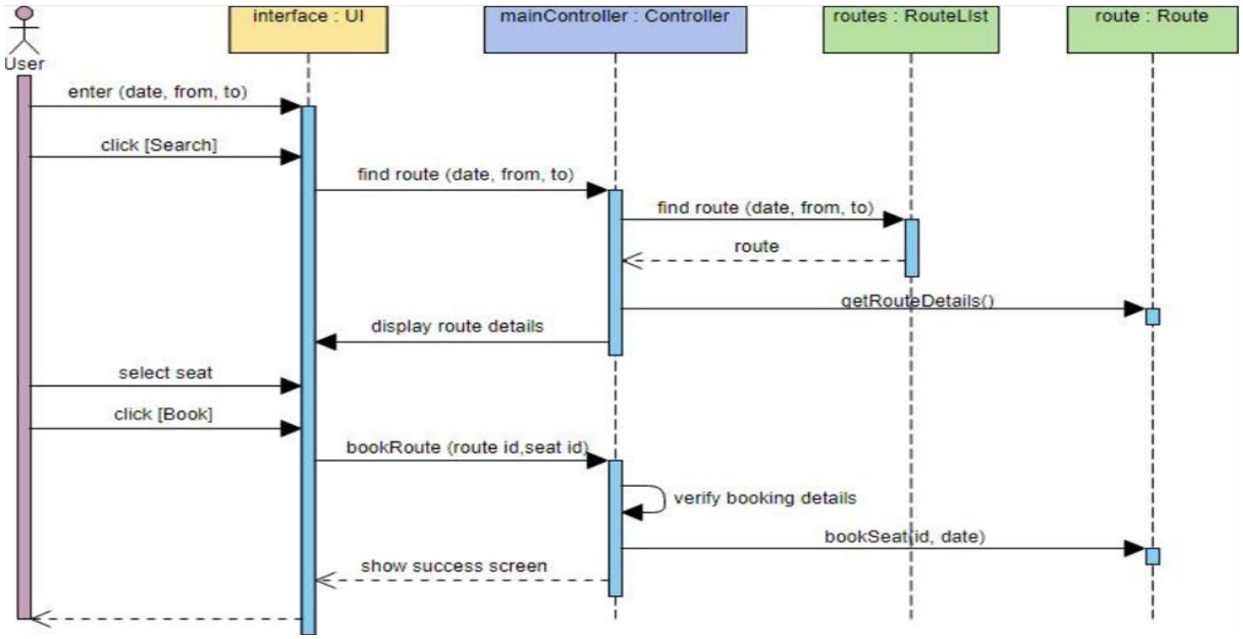


4.3 Class Diagram

The User class contains attributes like user ID, name, email, and password, along with methods for login, registration, and booking management. The ParkingSlot class includes details such as slot ID, location, and status, and provides methods for allocating and releasing slots. The Booking class manages reservation details, including booking ID, time, and status, and includes methods for creating and canceling bookings. The Admin class is responsible for managing parking slots and monitoring system operations. Relationships between these classes include associations, where a user can have multiple bookings, and each booking is linked to a specific parking slot. This diagram is essential for understanding the object-oriented structure of the system and serves as a blueprint for implementation.

Sequence Diagram

The Sequence Diagram illustrates the interaction between different components of the system over time, showing how messages are exchanged to complete a specific task. In the Smart Parking Management System, the sequence begins when a user initiates a login request through the frontend interface.



4.4 Sequence Diagram

This request is sent to the backend server, which validates the user credentials by communicating with the database. Once authenticated, the user can request parking availability, and the backend retrieves the relevant data from the database and sends it back to the frontend. When the user selects a parking slot and initiates a booking request, the backend processes the request, updates the database with the booking information, and confirms the reservation. The confirmation is then sent back to the user through the frontend. This step-by-step interaction helps in understanding the dynamic behavior of the system and ensures proper communication between components.

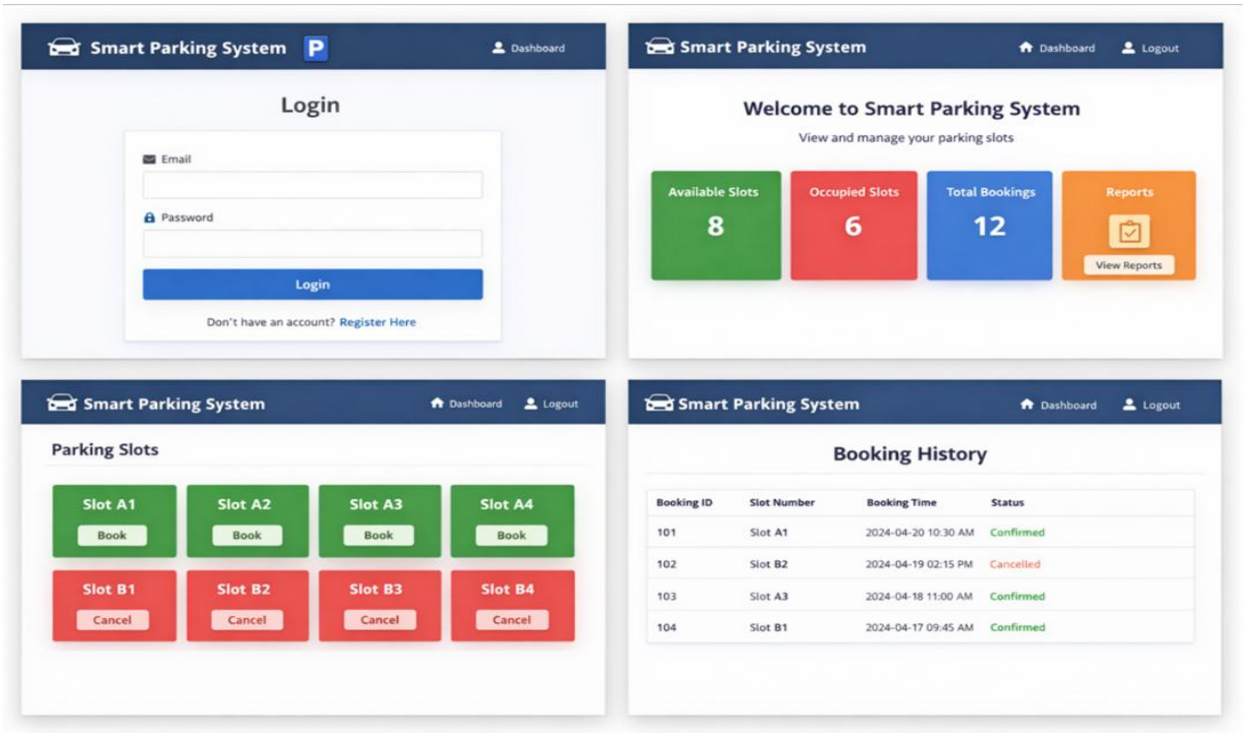
IMPLEMENTATION

```
import { Component, OnInit } from '@angular/core';
  import { ParkingService, ParkingSlot } from '../services/parking.service';

  @Component({
   selector: 'app-parking',
   templateUrl: './parking.component.html',
   styleUrls: ['./parking.component.css']
  })
  export class ParkingComponent implements OnInit {

   slots: ParkingSlot[] = [];

   constructor(private parkingService: ParkingService) {}

   ngOnInit(): void {
    this.loadSlots();
   }
```

```
  loadSlots() {
    this.parkingService.getSlots().subscribe(data => {
      this.slots = data;
    });
  }

  book(slotId: number) {
    this.parkingService.bookSlot(slotId).subscribe(() => {
      alert('Slot booked successfully');
      this.loadSlots();
    });
  }

  cancel(slotId: number) {
    this.parkingService.cancelSlot(slotId).subscribe(() => {
      alert('Booking cancelled');
      this.loadSlots();
    });
  }
}
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { HttpClientModule } from '@angular/common/http';

  import { AppComponent } from './app.component';
  import { ParkingComponent } from './parking/parking.component';

  @NgModule({
    declarations: [
      AppComponent,
      ParkingComponent
    ],
    imports: [
      BrowserModule,
      HttpClientModule
    ],
    bootstrap: [AppComponent]
  })
  export class AppModule {}
```

V. RESULT AND SNAPSHOTS

RESULTS



9.1 Result and Snapshots

VI. CONCLUSION

The Smart Parking Management System developed in this project successfully demonstrates how modern web technologies can be utilized to address real-world urban challenges. By using Angular 17 for the frontend and integrating it with backend services through REST APIs, the system provides a dynamic and efficient platform for managing parking operations. The application enables users to view real-time parking availability, book slots, and manage their reservations seamlessly through an intuitive web interface. This reduces the time spent searching for parking, minimizes traffic congestion, and enhances overall user convenience.The system also ensures better utilization of available parking spaces by automating the allocation process and reducing manual intervention. Through proper system design, including modular architecture and structured data flow, the application achieves scalability, reliability, and maintainability. Testing at various levels, such as unit, integration, and system testing, ensures that the system performs efficiently and meets the required specifications. Overall, the project highlights the importance of digital solutions in smart city development and provides a strong foundation for further advancements in intelligent parking systems.

Future work:

Although the current system provides an effective solution for parking management, there are several areas where it can be further enhanced to improve functionality and user experience. One potential improvement is the integration of Internet of Things (IoT) sensors to automatically detect the availability of parking slots in real time, eliminating the need for manual updates. This would make the system more accurate and reliable. Additionally, the development of a mobile application can extend accessibility, allowing users to book parking slots conveniently from their smartphones.

Another enhancement could involve implementing a payment gateway to support online payments for parking reservations, making the system more complete and user-friendly. Advanced features such as GPS-based navigation can guide users to the nearest available parking slot, further reducing search time.

The system can also incorporate machine learning techniques to predict parking demand based on historical data, helping in better resource planning and management. Furthermore, security can be improved by integrating authentication mechanisms such as biometric or two-factor authentication. Expanding the system to support multiple parking locations and integrating it with smart city infrastructure can make it more scalable and widely applicable. In conclusion, with these future enhancements, the Smart Parking Management System can evolve into a fully automated, intelligent, and scalable solution capable of addressing the growing challenges of urban parking management.

REFERENCES

[1] Sommerville, Ian. Software Engineering, 10th Edition, Pearson Education, 2016.

[2] Pressman, Roger S. Software Engineering: A Practitioner's Approach, 7th Edition, McGraw-Hill Education, 2014.

[3] Silberschatz, Abraham, Korth, Henry F., and Sudarshan, S. Database System Concepts, 6th Edition, McGraw-Hill, 2011.

[4] Elmasri, Ramez and Navathe, Shamkant B. Fundamentals of Database Systems, 7th Edition, Pearson, 2016.

[5] Gamma, Erich, Helm, Richard, Johnson, Ralph, and Vlissides, John. Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1994.

[6] Sommerville, Ian. Software Engineering, 10th Edition, Pearson Education, 2016.

[7] Pressman, Roger S. Software Engineering: A Practitioner's Approach, 7th Edition, McGraw-Hill Education, 2014.

[8] Silberschatz, Abraham, Korth, Henry F., and Sudarshan, S. Database System Concepts, 6th Edition, McGraw-Hill, 2011.

[9] Elmasri, Ramez and Navathe, Shamkant B. Fundamentals of Database Systems, 7th Edition, Pearson, 2016.

[10] Gamma, Erich, Helm, Richard, Johnson, Ralph, and Vlissides, John. Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1994.

[11] Shaheen, S. A., Cohen, A. P., and Zohdy, I. H. Smart Parking Management Systems: A Review, Transportation Research Board, 2016.

[12] Lin, T., Rivano, H., and Le Mouël, F. A Survey of Smart Parking Solutions, IEEE Transactions on Intelligent Transportation Systems, 2017.

[13] Geng, Y., and Cassandras, C. G. A New Smart Parking System Based on Optimal Resource Allocation and Reservations, IEEE Transactions on Intelligent Transportation Systems, 2012.

[14] Barone, R. E., Giuffrè, T., Siniscalchi, S. M., Morgano, M. A., and Tesoriere, G. Architecture for Parking Management in Smart Cities, IET Intelligent Transport Systems, 2014.

[15] Rajabioun, T., and Ioannou, P. A. On-Street and Off-Street Parking Availability Prediction Using Multivariate Spatio-Temporal Models, IEEE Transactions on Intelligent Transportation Systems, 2015.