A

Project Report

On

SMART PARKING MANAGEMENT SYSTEM

Submitted during SIX semester in partial fulfilment of the requirements for the award of degree of

Bachelor of Technology

in

Computer Engineering

by

SHAHJHAN ALAM

(22001003522)

Under supervision of

Dr .Atul Mishra

Department of Computer Engineering
FACULTY OF INFORMATICS & COMPUTING

J.C. Bose University of Science & Technology, YMCA
Faridabad – 121006

May 2024

CANDIDATE'S DECLARATION

I hereby certify that the work being presented in this Project titled **"Smart Parking Management System**" in partial fulfilment of the requirements for the degree of Bachelor of Technology in **Computer Engineering** and being submitted to J.C. Bose University of Science and Technology, YMCA, Faridabad , is an authentic record of my own work carried out under the supervision of **DR .ATUL MISHRA**.

The work contained in this project has not been submitted to any other University or Institute for the award of any degree or diploma by me.

Student's Signature
SHAHJHAN ALAM

Supervisor Signature
DR .ATUL MISHRA

CERTIFICATE

This is to certify that **Shahjhan Alam** of J.C. BOSE UNIVERSITY OF SCIENCE AND TECHNOLOGY (JCBUST), YMCA has successfully completed the project work titled "SMART PARKING MANAGEMENT SYSTEM " in partial fulfillment of the requirement for the completion of the UG course. The project is currently in progress, and the Some development is in pending. This project report is the record of authentic work carried out by them during the period from **January 2024 to May 2024**. They had worked under my guidance.

Signature :

Mentor Name: Dr .Atul Mishra

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Brief of the Project

The Smart Parking Management System project is a comprehensive solution designed to address the challenges of traditional parking systems by leveraging modern web technologies. Parking congestion and inefficient space utilization are common issues faced in urban environments. This project aims to provide an innovative approach to parking management using HTML, CSS, JavaScript, PHP, and XAMPP.

The primary objective of this system is to enhance the overall parking experience for users and administrators. By developing a user-friendly web interface, drivers can conveniently check real-time parking availability and make bookings as needed. Simultaneously, an administrative panel empowers parking authorities to efficiently manage parking slots and user data.

The project utilizes a full-stack development approach, encompassing front-end technologies like HTML for structure, CSS for styling, and JavaScript for interactivity. On the server-side, PHP is employed to handle business logic and communicate with a MySQL database managed through XAMPP. This integration enables dynamic updates, user authentication, and seamless database operations.

In summary, the Smart Parking Management System project endeavors to showcase the integration of various technologies to create a robust and efficient parking solution. This introduction sets the stage for exploring the project's objectives, features, technologies, and implementation in detail within the project report.

1.2 Problem Statement

The conventional parking management systems in urban areas often suffer from inefficiencies such as limited visibility into parking availability, lack of real-time updates, and manual management processes. These challenges lead to frustration among drivers due to time wasted searching for parking and contribute to congestion and environmental impacts from increased traffic.

Key issues include:

- **Limited Visibility:** Drivers often struggle to find available parking spots, leading to time wastage and increased traffic congestion.
- **Manual Management:** Parking administrators face difficulties in efficiently managing parking slots and user bookings without automated tools.
- **Real-time Updates:** The absence of real-time updates on parking availability makes it challenging for drivers to plan their parking needs effectively.
- **Resource Optimization:** Parking spaces are not optimally utilized, leading to inefficient use of valuable urban infrastructure.

The Smart Parking Management System aims to address these issues by implementing a modern, user-friendly solution that leverages web technologies. By providing real-time parking availability information and enabling online booking functionalities, this system seeks to enhance parking efficiency, reduce congestion, and improve the overall user experience for both drivers and parking administrators. This project seeks to bridge the gap between traditional parking systems and modern technological advancements to create a smarter and more sustainable solution.

1.3 Proposed Solution

The proposed Smart Parking Management System is a modern solution that integrates web technologies to optimize parking space utilization and enhance user convenience. This system aims to address the challenges faced by traditional parking management through the following key features:

- **Real-time Parking Availability:** Implement Computer Vision technology to detect and update parking slot availability in real-time, providing accurate information to users.
- **User-friendly Interface:** Develop a responsive web interface using HTML, CSS, and JavaScript to allow drivers to easily check parking availability and make bookings.
- **Online Booking System:** Enable users to reserve parking slots in advance through the web platform, reducing the time spent searching for parking.
- **Administrative Dashboard:** Provide parking administrators with a centralized dashboard to manage parking slots, monitor occupancy, and oversee user bookings.
- **Database Integration:** Utilize MySQL database management with PHP for efficient storage, retrieval, and management of parking-related data.

By integrating these features, the Smart Parking Management System aims to streamline the parking process, improve resource utilization, reduce congestion, and enhance the overall parking experience for both drivers and administrators. This project will demonstrate the feasibility and effectiveness of leveraging technology to address parking management challenges in urban environments.

1.4 Objective

The Smart Parking Management System project aims to achieve several key objectives focused on enhancing parking efficiency and user experience through innovative technology solutions:

- **Optimize Parking Utilization:** Implement a system that dynamically monitors and updates parking slot availability in real-time, enabling better utilization of parking spaces and reducing congestion.
- **Enhance User Convenience:** Develop a user-friendly web interface that allows drivers to easily check parking availability, make reservations, and receive real-time updates, thus minimizing the time spent searching for parking.
- **Enable Online Booking:** Implement an online booking system that allows users to reserve parking slots in advance, improving accessibility and reducing the uncertainty of finding parking during peak hours.
- **Centralized Management:** Provide parking administrators with a centralized dashboard to monitor parking occupancy, manage parking slots, and oversee user bookings efficiently.
- **Integrate Modern Technologies:** Utilize Computer Vision technology to automate the process of updating parking availability, ensuring accuracy and reliability of information presented to users.
- **Improve Traffic Flow:** By reducing the time spent searching for parking spots, the system aims to contribute to smoother traffic flow and reduce environmental impact from vehicle emissions.
- **Demonstrate Technological Feasibility:** Showcase the feasibility of integrating HTML, CSS, JavaScript, PHP, FLASK and MySQL technologies to create a comprehensive parking management system suitable for real-world deployment.

CHAPTER 2

LITERATURE SURVEY

A literature survey for the Smart Parking Management System project involves reviewing existing research and publications related to parking management systems, Computer Vision, web technologies, and similar projects. This survey provides insights into the current state-of-the-art solutions, technologies, and methodologies relevant to the development and implementation of a smart parking system.

Below are key areas to explore in the literature survey:

1. Smart Parking Systems Overview:

Review research papers and articles discussing smart parking systems, including their architectures, functionalities, and benefits in urban environments.

Understand different approaches to real-time parking monitoring, occupancy detection, and reservation systems.

2. Web Technologies for Parking Management:

Investigate literature on web-based parking management systems, focusing on frontend development using HTML, CSS, and JavaScript for user interfaces.

Explore backend technologies like PHP and database management (MySQL) for handling user requests, data storage, and system logic.

3. User Experience and Interface Design:

Study research on user experience (UX) design principles for parking applications, including interface design considerations to enhance usability and accessibility for drivers.

Review best practices for developing responsive and intuitive web interfaces for parking availability checking and booking.

4. **Case Studies and Implementations:**

Analyze case studies and project reports of similar smart parking systems implemented in different locations or institutions.

Identify challenges faced, lessons learned, and outcomes achieved in deploying smart parking solutions.

5. **Traffic Management and Urban Mobility:**

Explore literature on the impact of smart parking systems on traffic flow, congestion reduction, and overall urban mobility.

Understand how efficient parking management contributes to sustainable transportation and environmental goals.

6. **Security and Privacy Considerations:**

Investigate studies addressing security and privacy concerns related to smart parking systems, particularly in data transmission, user authentication, and system integrity.

By conducting a comprehensive literature survey in these areas, the Smart Parking Management System project can benefit from existing knowledge and insights to inform the design, implementation, and evaluation stages. This survey will also help identify gaps in current research and opportunities for innovation in smart parking technology.

CHAPTER 3

TECHNOLOGIES USED

The **Smart Parking Management System** project leverages a combination of frontend and backend technologies to create a robust and efficient parking solution. The technologies used in this project include:

Frontend Technologies:

HTML (HyperText Markup Language): Used for creating the structure and content of web pages.

HTML serves as the backbone of web development, providing the essential structure and content of web pages. It uses tags to define various elements such as headings, paragraphs, images, forms, and links. These elements form the building blocks of a web page, organizing content in a hierarchical and semantic manner. HTML is fundamental for creating accessible, SEO-friendly web pages that browsers can interpret and display correctly. It works in conjunction with CSS for styling and JavaScript for interactivity, forming the basis of modern web development.

CSS (Cascading Style Sheets): Employed to style the HTML elements and enhance the visual presentation of the web interface.

CSS plays a crucial role in web design by controlling the visual appearance of HTML elements. It allows developers to apply styles such as colors, fonts, spacing, layout, and responsiveness to enhance the presentation of web pages. By separating style from content, CSS enables consistent and efficient design across multiple web pages and devices. Selectors and rules define how HTML elements are styled, providing flexibility and modularity to web design. CSS can be applied inline, internally within HTML documents, or externally in separate style sheets, promoting maintainability and scalability of web projects.

JavaScript: JavaScript is a versatile programming language used extensively in web development to add interactivity and dynamic behavior to web pages. It runs on the client-side (in the user's browser) and can manipulate HTML elements, handle user events (like clicks and keystrokes), make asynchronous requests to servers (AJAX), and dynamically update content without reloading the entire page (DOM manipulation). JavaScript frameworks and libraries (e.g., React, Vue.js) further enhance its capabilities, allowing developers to build complex web applications efficiently. JavaScript is essential for implementing features like form validations, interactive maps, sliders, and animations, making websites more engaging and responsive to user actions.

Backend Technologies:

PHP (Hypertext Preprocessor): Used as the server-side scripting language to handle business logic, process user requests, and interact with the database.

PHP is a widely-used server-side scripting language specifically designed for web development. It is embedded within HTML and executes on the server, generating dynamic content that is then sent to the client's web browser. PHP is ideal for handling business logic, processing user requests (like form submissions), and interacting with databases to retrieve or manipulate data. In the context of the Smart Parking Management System, PHP can be used to authenticate users, manage parking slot bookings, update availability status based on sensor data, and perform other backend tasks. Its versatility and extensive community support make PHP a popular choice for web development.

MySQL:

MySQL is a powerful relational database management system (RDBMS) used to store and manage structured data. It is well-suited for applications requiring efficient data handling, indexing, querying, and transaction management. In the context of the parking management system, MySQL can store information about parking slots, user details, bookings, transaction records, and sensor data. PHP scripts can interact with MySQL databases using SQL queries to perform operations such as inserting new data, retrieving specific records, updating information, and generating reports. MySQL's scalability and reliability make it an excellent choice for handling the data-intensive requirements of the Smart Parking Management System.

Integrating our parking availability detection system with Flask): we create a user-friendly web interface, allowing users to access real-time parking information. Leveraging Flask's simplicity and flexibility, our system seamlessly serves predictions to users, enhancing their parking experience. With minimal setup and intuitive design, our Flask-powered web application provides instant access to parking availability data, optimizing user convenience.

Utilizing the power of Computer Vision): our parking availability detection system leverages cv2 for real-time video processing, enabling efficient monitoring of parking lots. By employing a pre-trained machine learning model serialized with pickle, we accurately classify parking spots as either occupied or vacant. Capturing live video feed from cameras installed at parking lot entrances, our system preprocesses each frame to enhance image quality before feeding it into the model. The model's predictions are visualized in real-time, providing users with immediate feedback on parking availability. With seamless integration into existing infrastructure, our system optimizes parking resource utilization and enhances user experience.

CHAPTER 4
REQUIREMENTS

The project statement or requirements document for a parking management system using HTML, CSS, PHP, and potentially incorporating a "HASHKEY."

Below is a sample project statement that outlines the requirements for a Smart Parking Management System.

Note that **"HASHKEY"** is a term that generally refers to a cryptographic hash function; however, its specific application in your project might need further clarification.

Technology Stack:

Frontend: HTML5, CSS3, JavaScript

Backend: PHP, MySQL ,Computer Vision

Web Application: Hybrid framework

Security: Encryption (HASHKEY), HTTPS

Requirements:

1. **User module**
2. **Admin module**
3. **Integration Module**
4. **Security and Privacy**
5. **Guard module**

6.Parking Space Detection

CHAPTER 5:

METHODOLOGY

To run the Smart Parking Management System project effectively, you'll need to ensure that certain requirements are met. Below are the key requirements for running this project:

Web Browser:

Use a modern web browser such as Google Chrome, Mozilla Firefox, Safari, or Microsoft Edge to access the web-based user interface.

Web Server Environment:

Install and set up a web server environment like Apache to host the PHP files. XAMPP can be used for local development, providing Apache and PHP out of the box.

Database Management System:

Set up MySQL database to store parking-related data. XAMPP includes MySQL, making it easy to manage the database locally.

Programming Languages:

Ensure PHP is installed and configured on the web server for server-side scripting.

JavaScript support is required in the web browser for client-side interactivity.

HTML/CSS/JavaScript Support:

Ensure the web browser supports HTML5, CSS3, and JavaScript for rendering the user interface and enabling dynamic features.

Dependencies and Libraries:

Install any necessary dependencies or libraries used in the project, such as jQuery for DOM manipulation or Bootstrap for responsive design (if applicable).

Computer Vision:

OpenCV (cv2) for real-time video processing, Flask for building the web interface, and a pre-trained machine learning model serialized with pickle for parking spot classification.

Internet Connectivity (for real-time updates):

While not strictly required for local testing, internet connectivity may be necessary for real-time updates or external API integrations (if applicable).

Security Measures:

Implement security best practices such as HTTPS protocol, input validation, and user authentication to secure the application and protect user data.

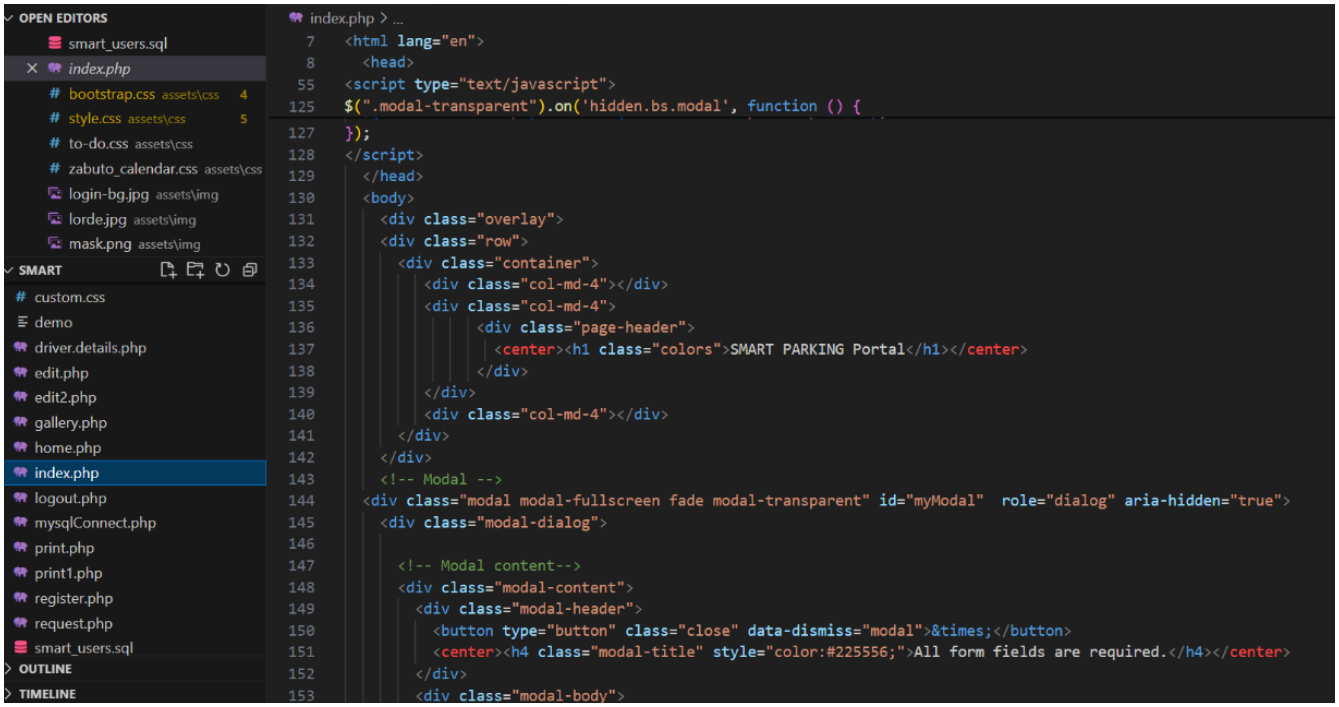
Testing and Debugging Tools:

Use development tools like browser developer tools (e.g., Chrome DevTools) for debugging and testing frontend code, and PHP debugging tools for server-side scripts.

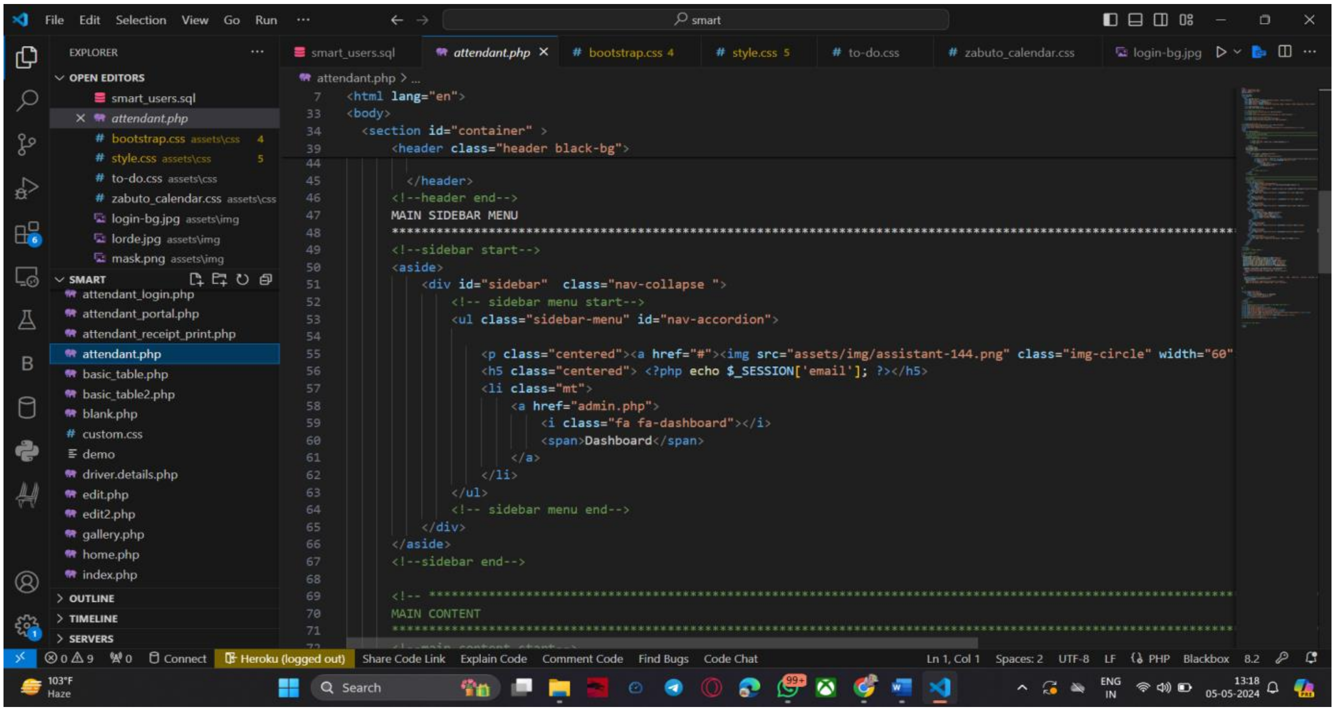
CHAPTER 6:

CODE SNAPSHOTS

Here are some example code snapshots demonstrating key functionalities of the Smart Parking Management System using HTML, CSS, JavaScript, PHP, Open CV, Pickle and MySQL:

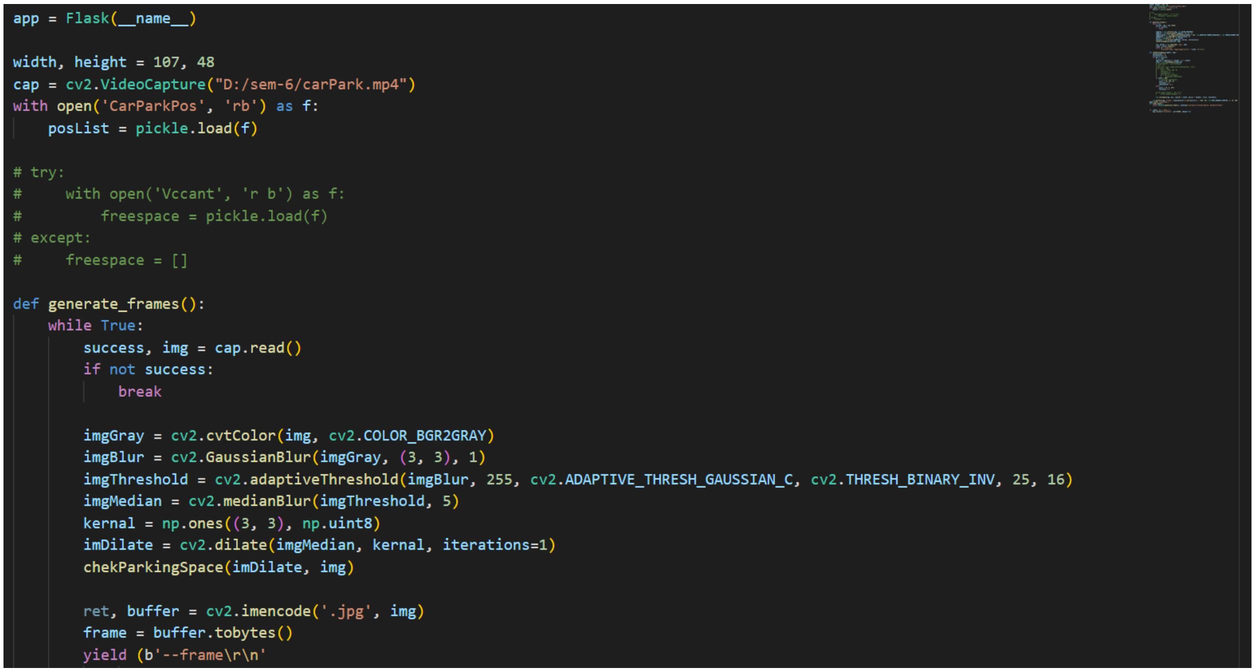


```html
<html lang="en">
  <head>
<script type="text/javascript">
$(".modal-transparent").on('hidden.bs.modal', function () {
});
</script>
  </head>
  <body>
    <div class="overlay">
    <div class="row">
      <div class="container">
        <div class="col-md-4"></div>
        <div class="col-md-4">
          <div class="page-header">
            <center><h1 class="colors">SMART PARKING Portal</h1></center>
          </div>
        </div>
        <div class="col-md-4"></div>
      </div>
    </div>
    <!-- Modal -->
  <div class="modal modal-fullscreen fade modal-transparent" id="myModal"  role="dialog" aria-hidden="true">
    <div class="modal-dialog">

      <!-- Modal content-->
      <div class="modal-content">
        <div class="modal-header">
          <button type="button" class="close" data-dismiss="modal">&times;</button>
          <center><h4 class="modal-title" style="color:#225556;">All form fields are required.</h4></center>
        </div>
        <div class="modal-body">
```

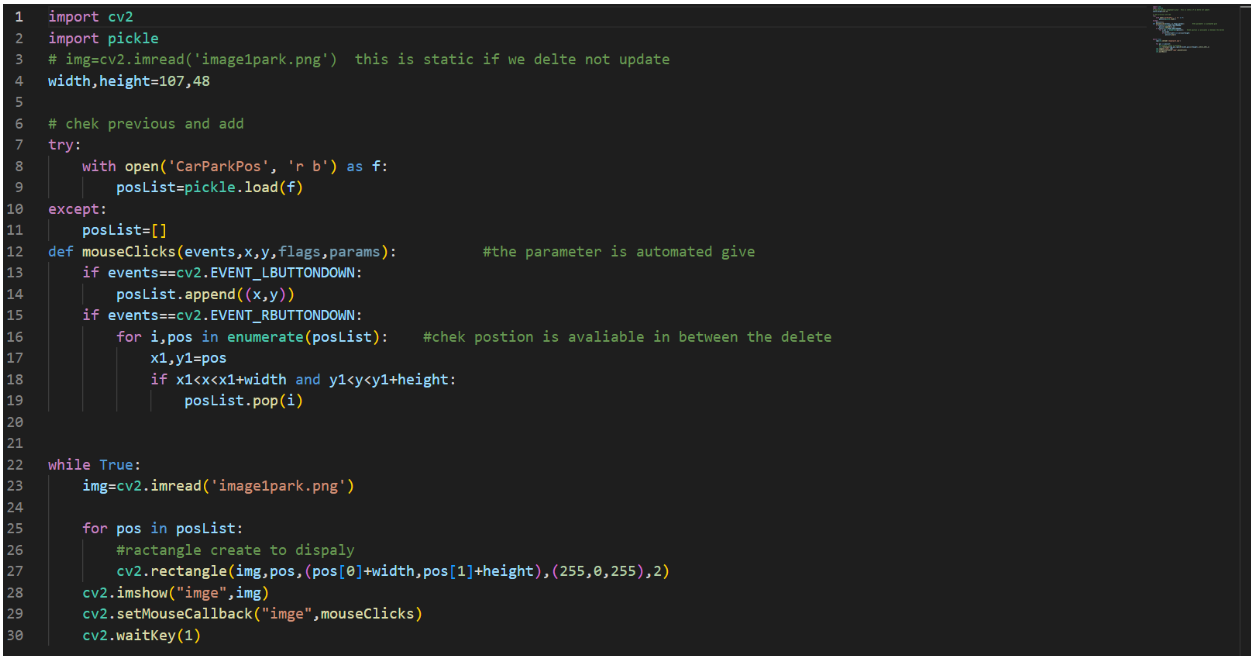


```html
<html lang="en">
<body>
  <section id="container" >
      <header class="header black-bg">

      </header>
    <!--header end-->
    MAIN SIDEBAR MENU
    ******************************************************************************************
    <!--sidebar start-->
    <aside>
        <div id="sidebar"  class="nav-collapse ">
            <!-- sidebar menu start-->
            <ul class="sidebar-menu" id="nav-accordion">

                <p class="centered"><a href="#"><img src="assets/img/assistant-144.png" class="img-circle" width="60"
                <h5 class="centered"> <?php echo $_SESSION['email']; ?></h5>
                <li class="mt">
                    <a href="admin.php">
                        <i class="fa fa-dashboard"></i>
                        <span>Dashboard</span>
                    </a>
                </li>
            </ul>
            <!-- sidebar menu end-->
        </div>
    </aside>
    <!--sidebar end-->

    <!-- ******************************************************************************************
    MAIN CONTENT
    ******************************************************************************************
    <!--main content start-->
```

Computer Vision(Parking slot availability):

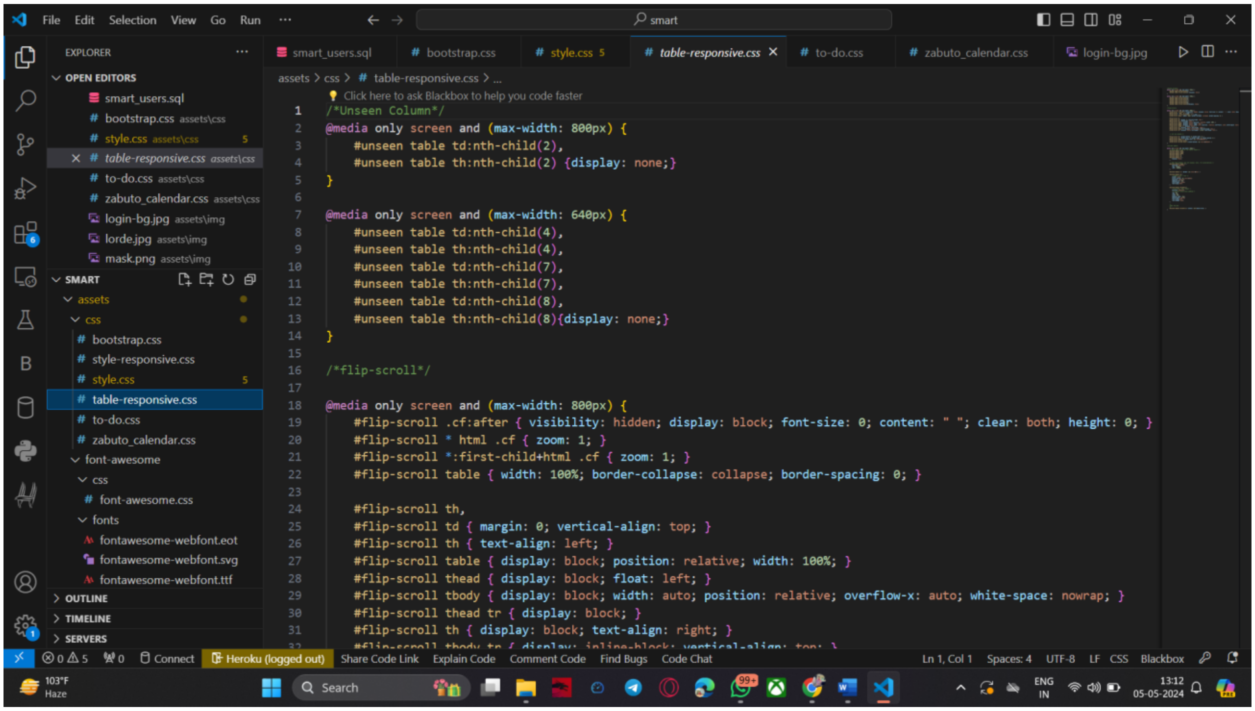


```python
app = Flask(__name__)

width, height = 107, 48
cap = cv2.VideoCapture("D:/sem-6/carPark.mp4")
with open('CarParkPos', 'rb') as f:
    posList = pickle.load(f)

# try:
#     with open('Vccant', 'r b') as f:
#         freespace = pickle.load(f)
# except:
#     freespace = []

def generate_frames():
    while True:
        success, img = cap.read()
        if not success:
            break

        imgGray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        imgBlur = cv2.GaussianBlur(imgGray, (3, 3), 1)
        imgThreshold = cv2.adaptiveThreshold(imgBlur, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY_INV, 25, 16)
        imgMedian = cv2.medianBlur(imgThreshold, 5)
        kernal = np.ones((3, 3), np.uint8)
        imDilate = cv2.dilate(imgMedian, kernal, iterations=1)
        chekParkingSpace(imDilate, img)

        ret, buffer = cv2.imencode('.jpg', img)
        frame = buffer.tobytes()
        yield (b'--frame\r\n'
```

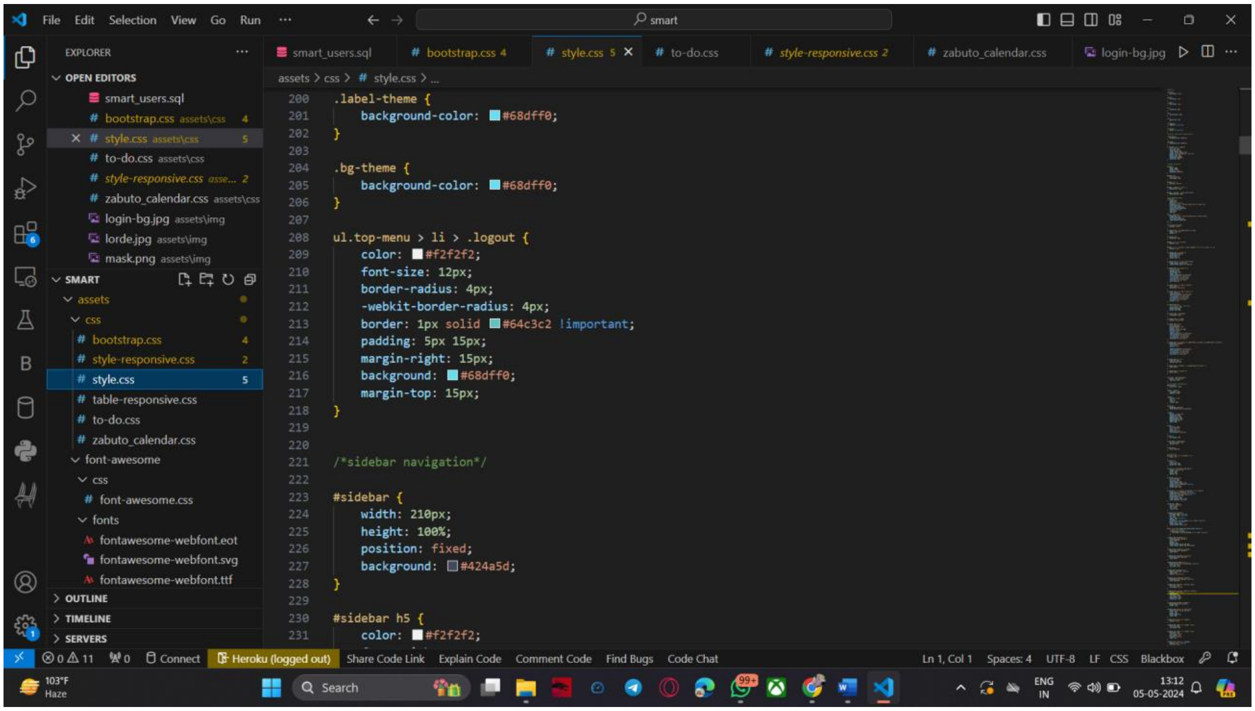


```python
import cv2
import pickle
# img=cv2.imread('image1park.png')  this is static if we delte not update
width,height=107,48

# chek previous and add
try:
    with open('CarParkPos', 'r b') as f:
        posList=pickle.load(f)
except:
    posList=[]
def mouseClicks(events,x,y,flags,params):          #the parameter is automated give
    if events==cv2.EVENT_LBUTTONDOWN:
        posList.append((x,y))
    if events==cv2.EVENT_RBUTTONDOWN:
        for i,pos in enumerate(posList):    #chek postion is avaliable in between the delete
            x1,y1=pos
            if x1<x<x1+width and y1<y<y1+height:
                posList.pop(i)


while True:
    img=cv2.imread('image1park.png')

    for pos in posList:
        #ractangle create to dispaly
        cv2.rectangle(img,pos,(pos[0]+width,pos[1]+height),(255,0,255),2)
    cv2.imshow("imge",img)
    cv2.setMouseCallback("imge",mouseClicks)
    cv2.waitKey(1)
```

CSS for Styling Form Elements :

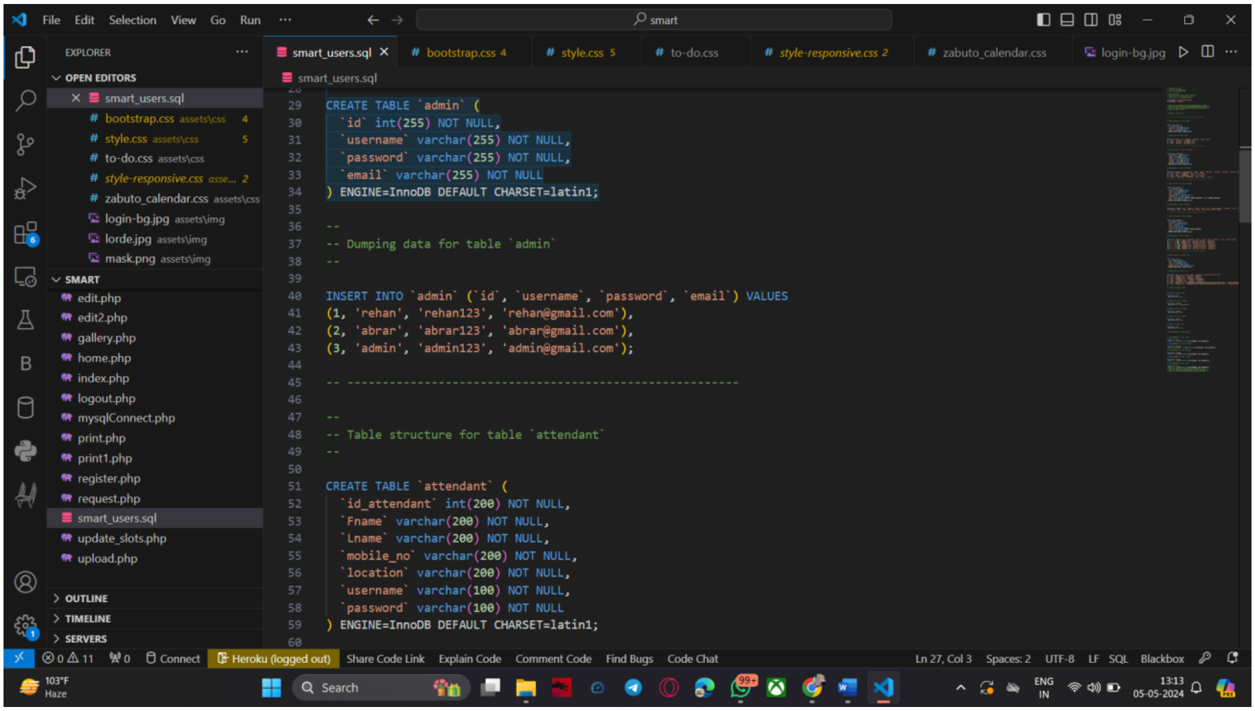


```css
/*Unseen Column*/
@media only screen and (max-width: 800px) {
    #unseen table td:nth-child(2),
    #unseen table th:nth-child(2) {display: none;}
}

@media only screen and (max-width: 640px) {
    #unseen table td:nth-child(4),
    #unseen table th:nth-child(4),
    #unseen table td:nth-child(7),
    #unseen table th:nth-child(7),
    #unseen table td:nth-child(8),
    #unseen table th:nth-child(8){display: none;}
}

/*flip-scroll*/

@media only screen and (max-width: 800px) {
    #flip-scroll .cf:after { visibility: hidden; display: block; font-size: 0; content: " "; clear: both; height: 0; }
    #flip-scroll * html .cf { zoom: 1; }
    #flip-scroll *:first-child+html .cf { zoom: 1; }
    #flip-scroll table { width: 100%; border-collapse: collapse; border-spacing: 0; }

    #flip-scroll th,
    #flip-scroll td { margin: 0; vertical-align: top; }
    #flip-scroll th { text-align: left; }
    #flip-scroll table { display: block; position: relative; width: 100%; }
    #flip-scroll thead { display: block; float: left; }
    #flip-scroll tbody { display: block; width: auto; position: relative; overflow-x: auto; white-space: nowrap; }
    #flip-scroll thead tr { display: block; }
    #flip-scroll th { display: block; text-align: right; }
    #flip-scroll tbody tr { display: inline-block; vertical-align: top; }
```

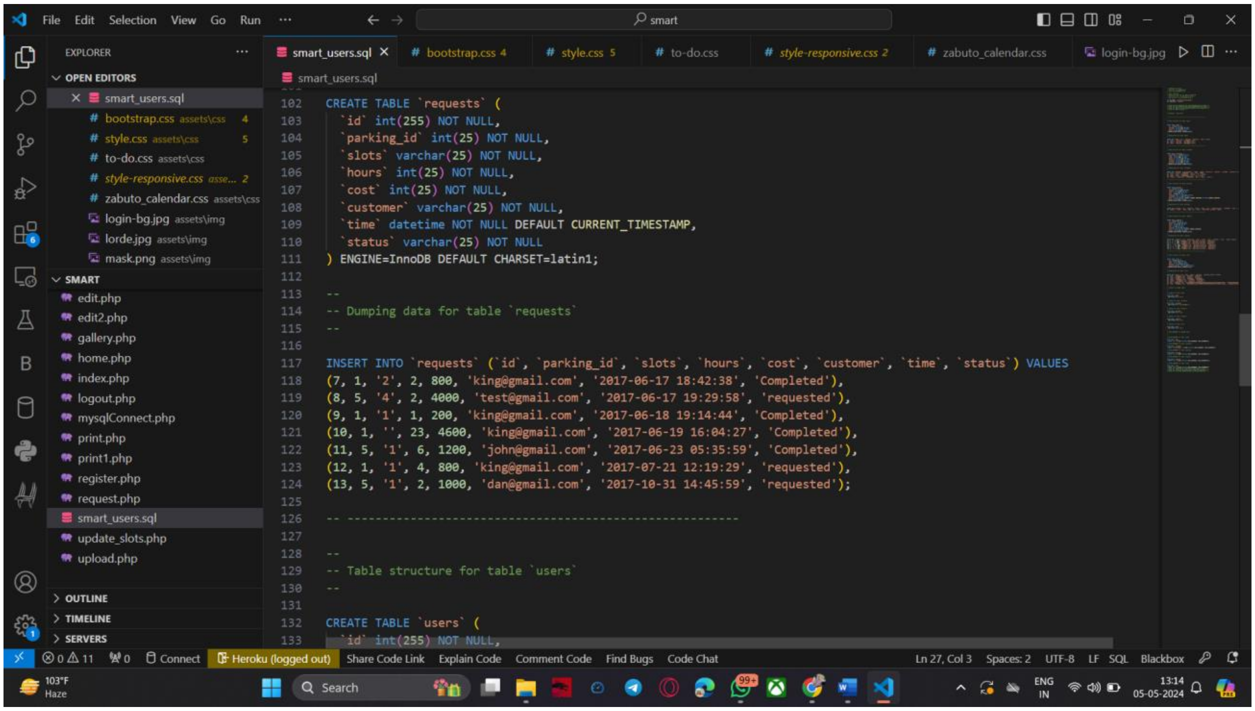


```css
.label-theme {
    background-color: #68dff0;
}

.bg-theme {
    background-color: #68dff0;
}

ul.top-menu > li > .logout {
    color: #f2f2f2;
    font-size: 12px;
    border-radius: 4px;
    -webkit-border-radius: 4px;
    border: 1px solid #64c3c2 !important;
    padding: 5px 15px;
    margin-right: 15px;
    background: #68dff0;
    margin-top: 15px;
}


/*sidebar navigation*/

#sidebar {
    width: 210px;
    height: 100%;
    position: fixed;
    background: #424a5d;
}

#sidebar h5 {
    color: #f2f2f2;
```

MySQL Table Schema:



```sql
CREATE TABLE `admin` (
  `id` int(255) NOT NULL,
  `username` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `admin`
--

INSERT INTO `admin` (`id`, `username`, `password`, `email`) VALUES
(1, 'rehan', 'rehan123', 'rehan@gmail.com'),
(2, 'abrar', 'abrar123', 'abrar@gmail.com'),
(3, 'admin', 'admin123', 'admin@gmail.com');

-- --------------------------------------------------------

--
-- Table structure for table `attendant`
--

CREATE TABLE `attendant` (
  `id_attendant` int(200) NOT NULL,
  `Fname` varchar(200) NOT NULL,
  `Lname` varchar(200) NOT NULL,
  `mobile_no` varchar(200) NOT NULL,
  `location` varchar(200) NOT NULL,
  `username` varchar(100) NOT NULL,
  `password` varchar(100) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```



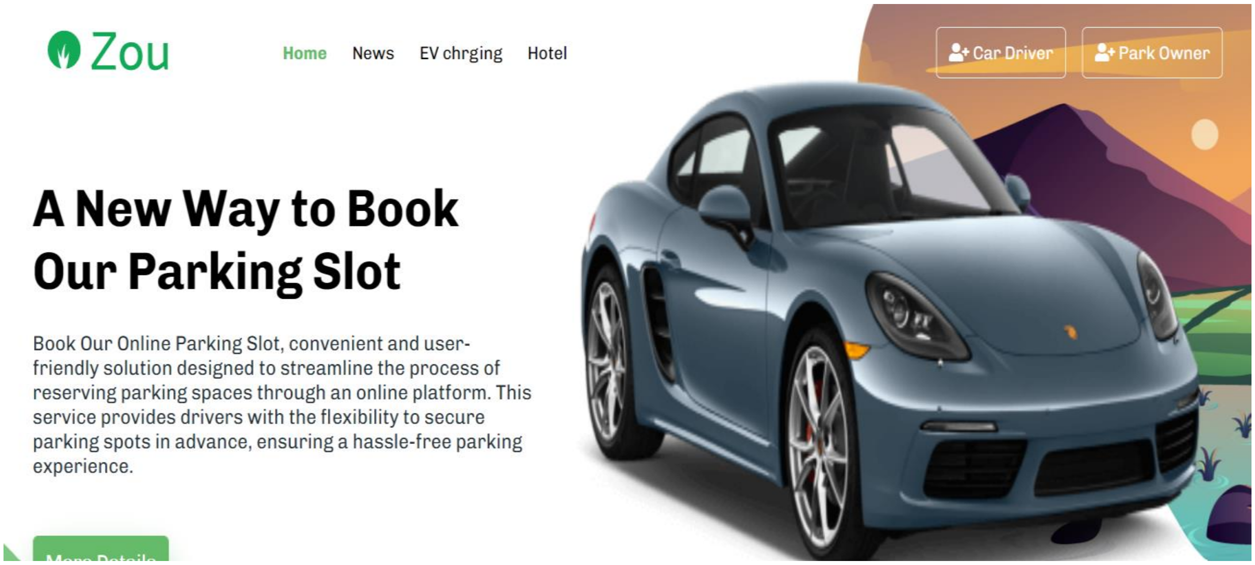
```sql
CREATE TABLE `requests` (
  `id` int(255) NOT NULL,
  `parking_id` int(25) NOT NULL,
  `slots` varchar(25) NOT NULL,
  `hours` int(25) NOT NULL,
  `cost` int(25) NOT NULL,
  `customer` varchar(25) NOT NULL,
  `time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `status` varchar(25) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `requests`
--

INSERT INTO `requests` (`id`, `parking_id`, `slots`, `hours`, `cost`, `customer`, `time`, `status`) VALUES
(7, 1, '2', 2, 800, 'king@gmail.com', '2017-06-17 18:42:38', 'Completed'),
(8, 5, '4', 2, 4000, 'test@gmail.com', '2017-06-17 19:29:58', 'requested'),
(9, 1, '1', 1, 200, 'king@gmail.com', '2017-06-18 19:14:44', 'Completed'),
(10, 1, '', 23, 4600, 'king@gmail.com', '2017-06-19 16:04:27', 'Completed'),
(11, 5, '1', 6, 1200, 'john@gmail.com', '2017-06-23 05:35:59', 'Completed'),
(12, 1, '1', 4, 800, 'king@gmail.com', '2017-07-21 12:19:29', 'requested'),
(13, 5, '1', 2, 1000, 'dan@gmail.com', '2017-10-31 14:45:59', 'requested');

-- --------------------------------------------------------

--
-- Table structure for table `users`
--

CREATE TABLE `users` (
  `id` int(255) NOT NULL,
```

These code snippets demonstrate how HTML, CSS, JavaScript, PHP, and MySQL can be used together to implement various functionalities of the Smart Parking Management System, such as displaying a booking form, styling form elements, checking parking slot availability dynamically, processing booking requests, and storing booking information in a MySQL database.

CHAPTER 7

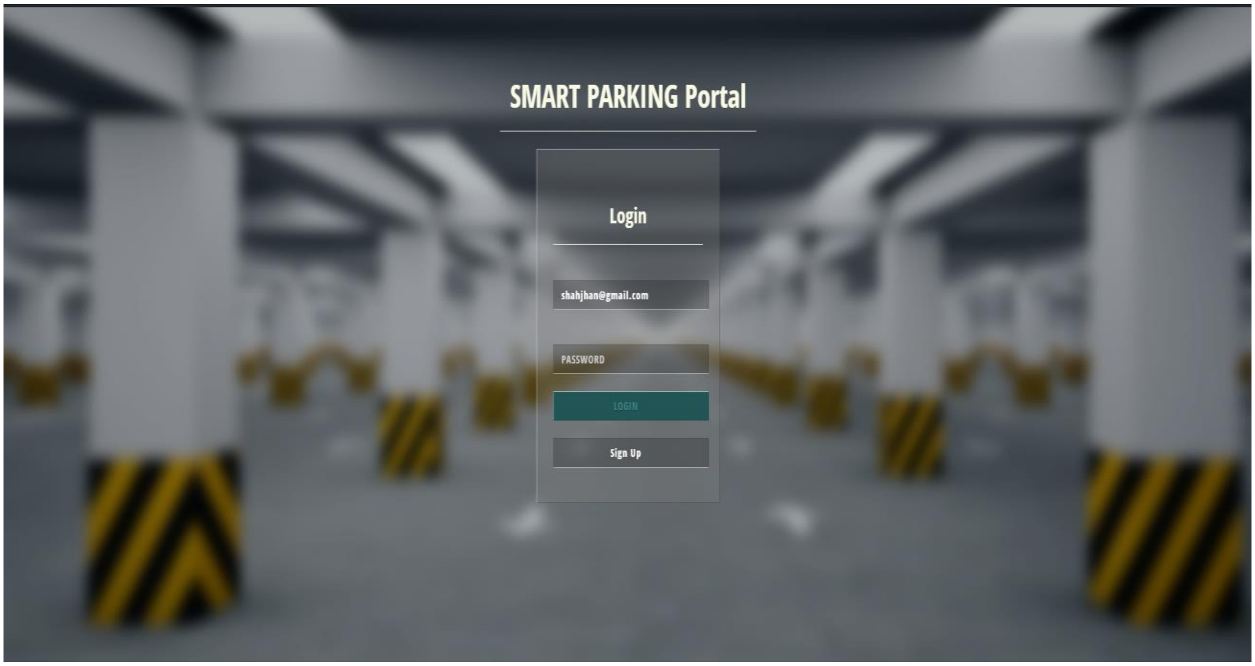
OUTPUT SNAPSHOTS

To provide output snapshots for the Smart Parking Management System, I'll describe the expected user interface and interactions based on the functionalities implemented in the project. Here are example descriptions of potential output snapshots:



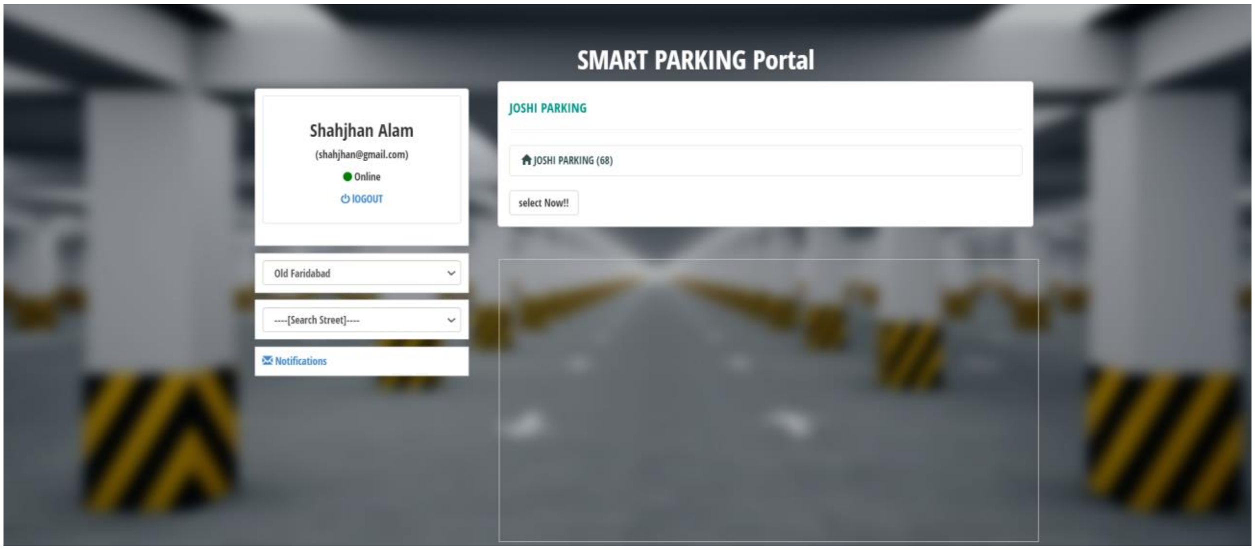
A New Way to Book Our Parking Slot

Book Our Online Parking Slot, convenient and user-friendly solution designed to streamline the process of reserving parking spaces through an online platform. This service provides drivers with the flexibility to secure parking spots in advance, ensuring a hassle-free parking experience.


SMART PARKING Portal

Login

shahjhan@gmail.com



SMART PARKING Portal

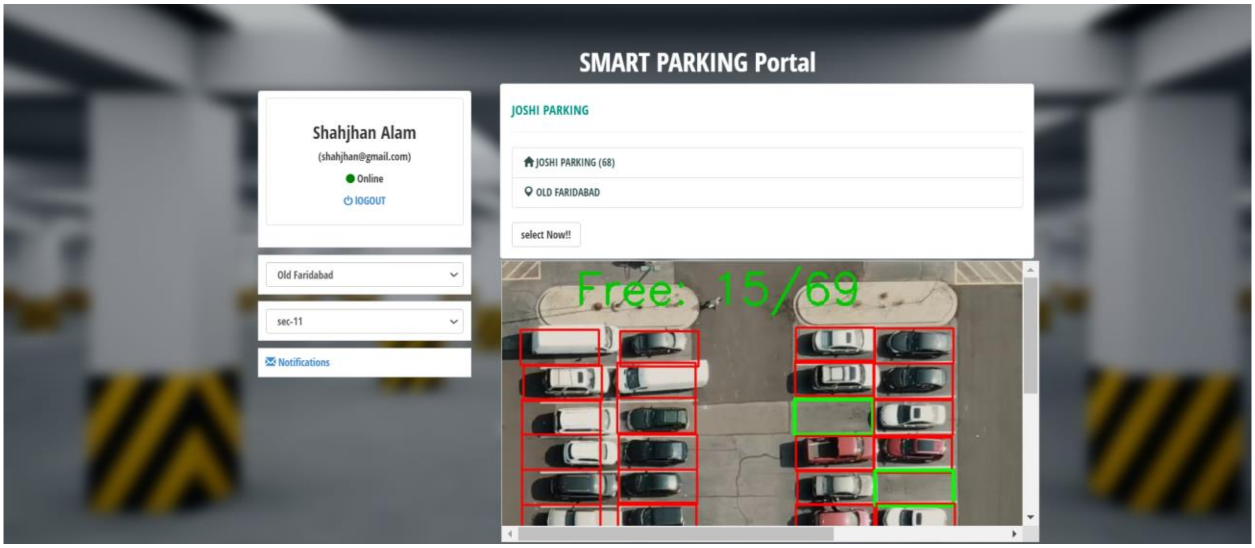
Shahjhan Alam
(shahjhan@gmail.com)
Online

JOSHI PARKING

JOSHI PARKING (68)

Old Faridabad

----[Search Street]----



SMART PARKING Portal

Shahjhan Alam
(shahjhan@gmail.com)
Online

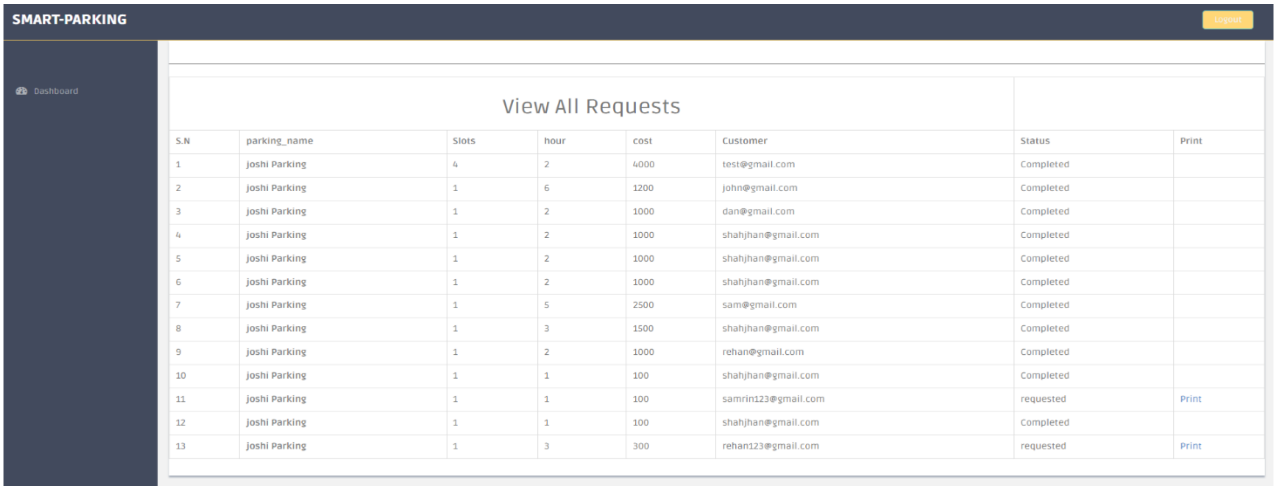
JOSHI PARKING

JOSHI PARKING (68)

OLD FARIDABAD

Old Faridabad

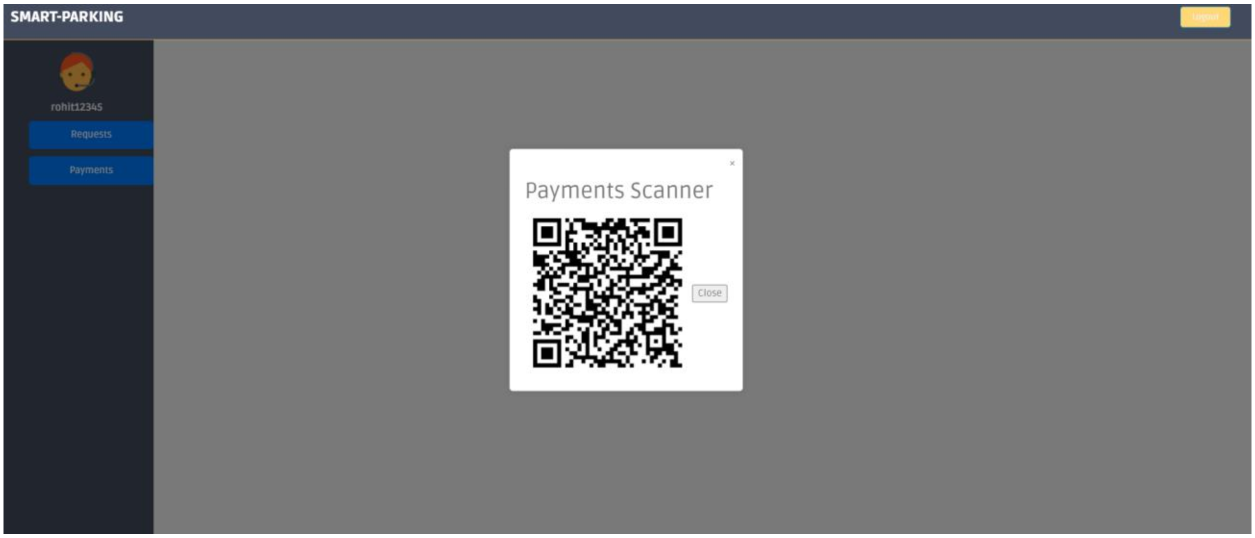
sec-11

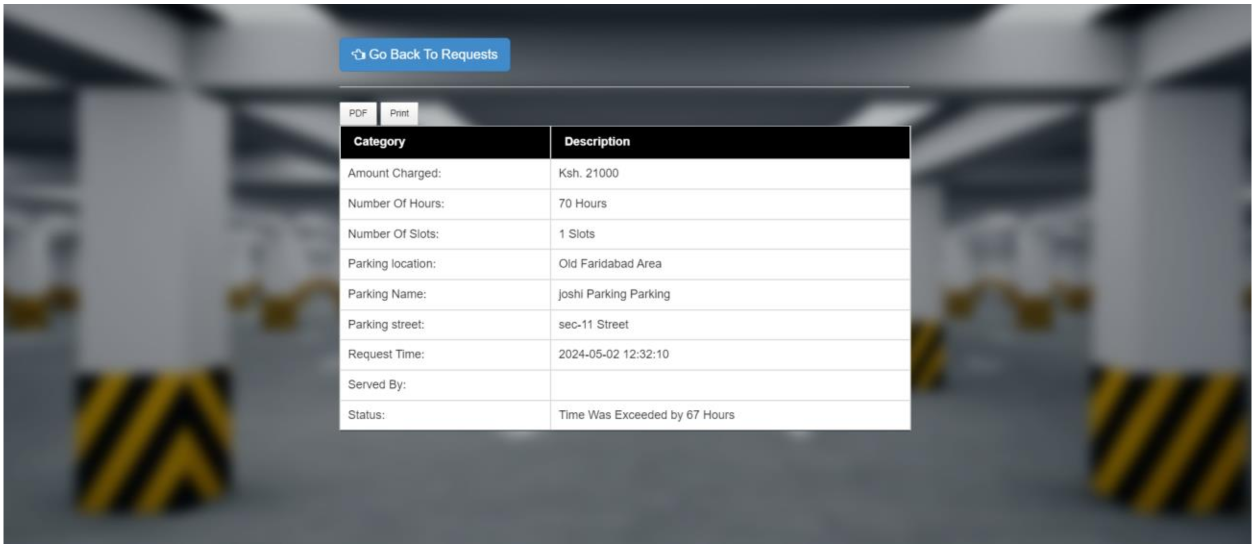


SMART-PARKING

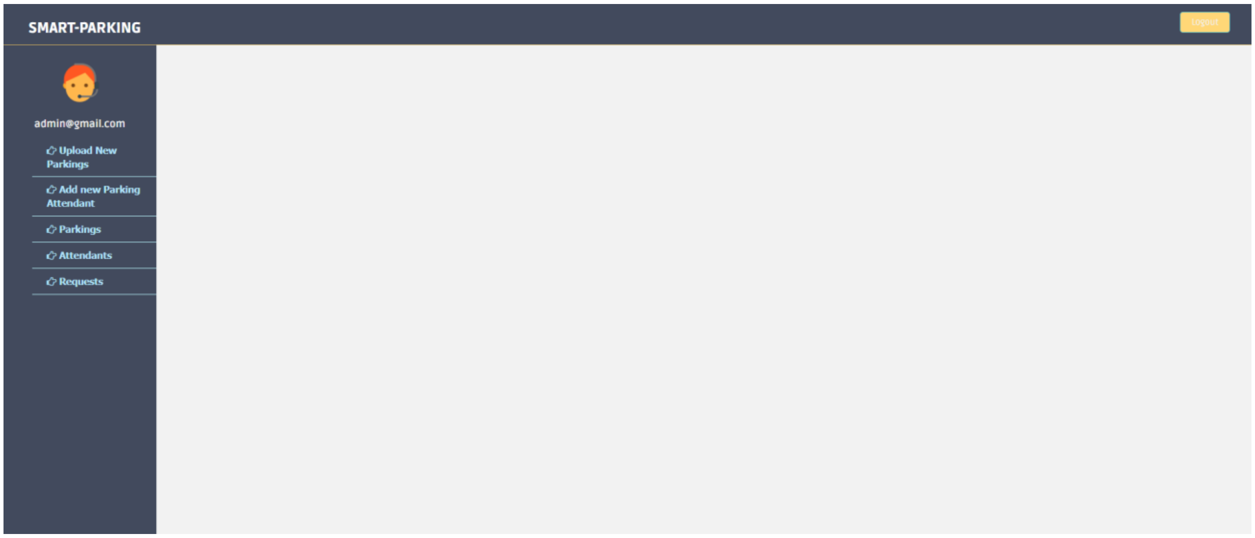
View All Requests

S.N	parking_name	Slots	hour	cost	Customer	Status	Print
1	joshi Parking	4	2	4000	test@gmail.com	Completed	
2	joshi Parking	1	6	1200	john@gmail.com	Completed	
3	joshi Parking	1	2	1000	dan@gmail.com	Completed	
4	joshi Parking	1	2	1000	shahjhan@gmail.com	Completed	
5	joshi Parking	1	2	1000	shahjhan@gmail.com	Completed	
6	joshi Parking	1	2	1000	shahjhan@gmail.com	Completed	
7	joshi Parking	1	5	2500	sam@gmail.com	Completed	
8	joshi Parking	1	3	1500	shahjhan@gmail.com	Completed	
9	joshi Parking	1	2	1000	rehan@gmail.com	Completed	
10	joshi Parking	1	1	100	shahjhan@gmail.com	Completed	
11	joshi Parking	1	1	100	samrin123@gmail.com	requested	Print
12	joshi Parking	1	1	100	shahjhan@gmail.com	Completed	
13	joshi Parking	1	3	300	rehan123@gmail.com	requested	Print





Category	Description
Amount Charged:	Ksh. 21000
Number Of Hours:	70 Hours
Number Of Slots:	1 Slots
Parking location:	Old Faridabad Area
Parking Name:	joshi Parking Parking
Parking street:	sec-11 Street
Request Time:	2024-05-02 12:32:10
Served By:	
Status:	Time Was Exceeded by 67 Hours



CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENT

Future Enhancements:

Integrating dynamic pricing into our smart parking system enhances user experience by offering flexible rates that adapt to demand and supply. Utilizing real-time parking availability data, the system dynamically adjusts pricing based on factors like occupancy levels, time of day, and special events, optimizing revenue for parking operators. Additionally, incorporating an advanced booking feature allows users to reserve parking spots in advance, ensuring guaranteed parking availability at their preferred location and time. This feature streamlines the parking process, reduces congestion, and provides users with convenience and peace of mind. Together, these enhancements transform our smart parking system into a comprehensive solution that maximizes efficiency, revenue, and user satisfaction.

Mobile Application Development: Develop a companion mobile application for iOS and Android platforms, allowing users to access parking information and make bookings on the go

Smart Navigation and Guidance: Implement smart navigation features within the application to guide users to available parking slots based on their location**.**

Accessibility Features: Enhance accessibility features to accommodate users with disabilities, ensuring inclusivity and usability for all individuals.

Security and Privacy Enhancements: Strengthen security measures such as encryption of sensitive data, user authentication, and regular security audits to protect user information and maintain trust

Conclusion:

In conclusion, our smart parking system offers a comprehensive solution to modern parking challenges, enhancing convenience, efficiency, and revenue generation. Leveraging real-time parking availability data and cutting-edge technologies like computer vision and machine learning, our system optimizes parking space utilization and reduces congestion.

The integration of real-time parking availability information allows users to make informed decisions about where to park, reducing time spent searching for parking spots and alleviating traffic congestion. Additionally, the system provides parking operators with valuable insights into parking usage patterns, enabling them to better manage their parking facilities and optimize revenue generation.

Overall, our smart parking system represents a significant advancement in urban infrastructure, demonstrating the power of technology to improve everyday experiences and address complex urban challenges. With its user-centric design and seamless integration, our system sets a new standard for parking management, paving the way for smarter, more efficient cities.

REFERENCE

1. L. Mainetti, L. Palano, L. Patrono, M. L. Stefanizzi, and R. Vergallo, Integration of RFID and WSN technologies in a smart parking system," in Proc. 22nd Int. Conf. Softw., Telecommun. Comput. Netw. (SoftCOM), 2014, pp. 104110.

2. Hamada R. H.Al-Absi, Justin Dinesh Daniel Devaraj, Patrick Sebastian, Yap Vooi Voon "A vision based car parking system"

3. Hanif NHHM, Badiozaman MH, Daud H (2010) Smart parking reservation system using Short Message Services (SMS). In: International Conference on Intelligent and Advanced Systems (ICIAS), pp 1–5

4. Khang SC, Hong TJ, Chin TS, Wang S (2010) Wireless Mobile-Based Shopping Mall Car Parking System (WMCPS). In: IEEE AsiaPacific Services Computing Conferences (APSCC)