PARKING SLOT REGISTRATION

CAPSTONE PROJECT

Submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF SCIENCE

IN

COMPUTER TECHNOLOGY

Submitted by

HAFIL A - [22BCT017]
TANISHA NAHATA – [22BCT054]
CHARUMATHI V S – [22BCT009]

Under the Guidance of

Dr. B. Radha M.C.A. Ph.D.
Associate Professor and Head of the Department
Department of Computer Technology and Data Science

SRI KRISHNA ARTS AND SCIENCE COLLEGE
Accredited by NAAC with 'A' grade
Coimbatore – 641008

OCTOBER 2023

CERTIFICATE

SRI KRISHNA ARTS AND SCIENCE COLLEGE

Accredited by NAAC with 'A' grade, Kuniamuthur, Coimbatore – 641008

CERTIFICATE

This is to certify that the Capstone Project report entitled **"PARKING SLOT REGISTRATION"** in partial fulfilment of requirements for the award of the degree of Bachelor of Science in Computer Technology is a record of bonafide work carried out by **HAFIL A - 22BCT017, TANISHA NAHATA - 22BCT054, CHARUMATHI V S – 22BCT009** and that no part of this has been submitted for the award of any other degree or diploma and the work has not been published in popular journal or magazine.

GUIDE **HOD** **DEAN**

This Capstone Report is submitted for the viva voce conducted on_____at Sri Krishna Arts and Science College.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

SRI KRISHNA ARTS AND SCIENCE COLLEGE

Accredited by NAAC with 'A' grade, Kuniamuthur, Coimbatore – 641008

DECLARATION

I hereby declare that the Capstone Project report entitled **"PARKING SLOT REGISTRATION"** submitted in partial fulfilment of the requirements for the award of the degree of **Bachelor of Science in Computer Technology** is an original work submitted and it has not been previously formed the basis for the award of any other Degree, Diploma, Associate ship, Fellowship or similar titles to any other university or body during the period of my study.

Place: Coimbatore

Date:

Signature of the Candidate

HAFIL A

TANISHA NAHATA

CHARUMATHI V S

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

I am ineffably indebted to **Dr. K. Sundararaman, M.Com., M.Phil., Ph.D., Chief Eexcutive Officer,** Sri Krishna Institutions, Coimbatore.

I convey my profound gratitude to **Dr. R. Jagajeevan MBA., Ph.D., Principal,** Sri Krishna Arts and Science College for giving me this opportunity to undergo this Capstone Project work.

It is my prime to solemnly express my sense of gratitude to **Dr. K. S. Jeen Marseline, M.C.A., M.Phil., Ph.D., HOD and Dean,** Department of ICT & Cognitive Systems, Sri Krishna Arts and Science College.

I would like to express sincere gratitude to **Dr. B. Radha M.C.A., Ph.D., Associate Professor and Head of the Department,** Department of Computer Technology & Data Science, Sri Krishna Arts and Science College.

I would like to extend my thanks and unbound sense for the timely help and assistance given by **Dr. B. Radha M.C.A., Ph.D., Associate Professor and Head of the Department,** Department of Computer Technology & Data Science, Sri Krishna Arts and Science College in completing the Capstone Project work. Her remarkable guidance at every stage of my work was coupled with suggestion and motivation.

I take this opportunity to thank my parents and friends for their constant support and encouragement throughout this training

HAFIL A

TANISHA NAHATA

CHARUMATHI V S

TABLE OF CONTENTS

TABLE OF CONTENTS

PROBLEM STATEMENT

Chapter 1

PROBLEM STATEMENT

Parking Slot Registration is a web – based software developed for the purpose of managing and booking slot for parking vehicle conveniently in our campus and also helps to utilize the total parking spaces. The main purpose of this website is to help students, faculties and other staffs of our college to book or register a parking space for their vehicles conveniently by providing information status about No. of spaces available, No. of vehicles registered, details and contact number of the registered vehicles, etc. Everyone know that our campus has a large No. of parking spaces for vehicles like car and motorcycles but they don't know whether there are parking spaces left over and even watch-man are struggling to find the No. of free spaces available in the parking. So, due to this there are many parking spaces are left over free causing congestion in the college. Initially all the system of our campus are unknown and difficult for everyone. By using this website students and faculty members can find the details of all the parking space available. So, this also helps in getting a parking on time instead of searching for it.

SYSTEM STUDY

Chapter 2
SYSTEM STUDY

2.1 REVIEW ON THE EXISTING WORK

Usually, we don't have an application or a software for the Parking Slot Registration. Due to this there are many parking spaces left over without being filled. We also know that many people come till the square parking but due to lack of parking spaces or congestion they are forced to go back to the bus parking.

2.1.1 LIMITATIONS

- **Security Vulnerabilities:** The code does not include robust security measures against common web vulnerabilities such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF). This could pose a significant security risk. Passwords are stored in plaintext in the database, which is a severe security risk. Passwords should be securely hashed and salted.

- **Database Handling:** There is no validation of user input before inserting data into the database, making it susceptible to SQL injection attacks. Database connection credentials are hardcoded in the PHP code, which is not recommended for security reasons. These credentials should be stored securely, and environment variables should be used.

- **User Experience:** The UI/UX could be improved for better user engagement. The design and layout are relatively basic. Error handling and validation for user inputs are minimal. Providing meaningful error messages to users can enhance the user experience.

- **Code Organization:** The code is somewhat fragmented across multiple files. Consider organizing your code into a more structured and modular format. The lack of comments and documentation in the code makes it challenging for others (and even yourself) to understand and maintain it in the future.

- **Scalability and Performance:** The code does not address potential scalability issues, such as handling a large number of registered vehicles or users. There's no consideration for optimizing database queries, which could lead to performance issues as the application scales.

- **Maintenance:** Updating and maintaining this codebase may become challenging over time due to the lack of clear structure and documentation.

- **Browser Compatibility:** The code may not be fully compatible with all web browsers. It's essential to test and ensure cross-browser compatibility.

- **Data Validation:** Data input is not thoroughly validated. Proper validation should be performed on both the client-side (JavaScript) and server-side (PHP) to prevent invalid or malicious data.

2.2 PROPOSED WORK

Parking Slot Registration is a Static Website – developed by using HTML, CSS, JavaScript and PHP languages. In this website, users can get information of their parking details. Admins can customize or edit the details of the respected parking spaces and users get a User-friendly Interface for booking parking spaces. This website can help everyone to park their vehicles in the registered slot so that they don't have to waste their time searching for a vehicle space.

2.2.1 ADVANTAGES OF THE PROPOSED WORK

- **Efficient Parking Management:** The system streamlines parking management, making it more efficient and organized. Users can reserve parking slots in advance, reducing congestion and ensuring a smoother parking experience.
- **User Convenience:** Users can easily register their vehicles and reserve parking spaces from the comfort of their homes or on-the-go through a user-friendly interface. This convenience saves time and effort.
- **Reduced Congestion:** By allowing users to reserve parking spaces in advance, the system helps reduce congestion in parking areas, leading to quicker entry and exit.
- **Improved Security:** The system can enhance security by keeping a record of vehicles and their owners. This information can be valuable in case of any security incidents or disputes.
- **Real-time Information:** Users can access real-time information about available parking slots, making it easier to plan their visits to the parking area.
- **Data Analytics:** The system can collect data on parking usage patterns, which can be analysed to optimize parking space allocation and pricing strategies.
- **User Notifications:** Users can receive notifications about their parking reservations, helping them remember their bookings and arrive on time.
- **Environmental Benefits:** By reducing the time spent searching for parking and minimizing idling, the system can contribute to lower fuel consumption and reduced emissions.
- **Administrative Efficiency:** For administrators, the system simplifies the management of parking data, making it easier to track and monitor usage, issue permits, and enforce regulations.
- **Scalability:** The system can be easily scaled to accommodate additional parking areas or facilities as needed.

2.3 LANGUAGES USED

- **HTML**

  HTML (Hypertext Markup Language) is used for creating the structure and content of the web pages in your application. It defines the layout, text, forms, and other elements that users interact with.

- **CSS**

  CSS (Cascading Style Sheets) is used to style and format the HTML elements, making your application visually appealing and user-friendly. It controls aspects like colours, fonts, spacing, and layout.

- **JavaScript**

  JavaScript adds interactivity and dynamic behaviour to your web application. It's essential for handling user actions, form validation, and making asynchronous requests to the server for real-time updates.

- **PHP**

  PHP is a server-side scripting language that can process user requests on the server, interact with the database, and generate dynamic content. In your case, it's likely used for server-side logic such as user authentication, handling form submissions, and communicating with the database.

- **MySQL**

  MySQL is a relational database management system (RDBMS) used to store and manage data for your application. It's used for tasks like storing user information, parking slot details, and handling data retrieval and manipulation.

2.4 SOFTWARE AND HARDWARE CONFIGURATION

HARDWARE SPECIFICATION

RAM(Memory): 8 GB

Internal Storage: 128 GB SSD

Processor: AMD Ryzen 3 3200G 10th gen

GPU: AMD Ryzen 7 5800

SOFTWARE SPECIFICATION

Operating System: Windows

Front-End: HTML, CSS, JavaScript

Back-End: PHP

Platform: Visual Studio Code

Visual Studio Code (VS Code) is a free, lightweight, open-source code editor developed by Microsoft. It has gained immense popularity among developers due to its versatility, powerful features, and extensive extension ecosystem. Here's an overview of Visual Studio Code and its key features.

- **Cross-Platform**

  Visual Studio Code is available for Windows, macOS, and Linux, making it accessible to a wide range of developers.

- **Intuitive User Interface**

  It features a clean and user-friendly interface with a sidebar for navigation, a central editor area, and a status bar at the bottom.

- **Code Editing**

  VS Code provides syntax highlighting, autocompletion, code navigation, and linting, making code editing efficient and error-free.

DESIGN

Chapter 3
DESIGN

3.1 DATA FLOW DIAGRAM

Level 0

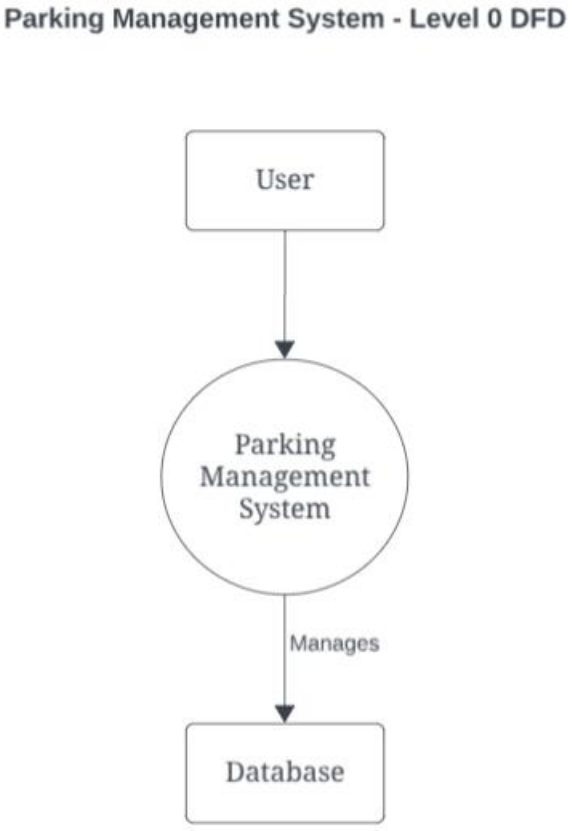


Figure-1

Level 1

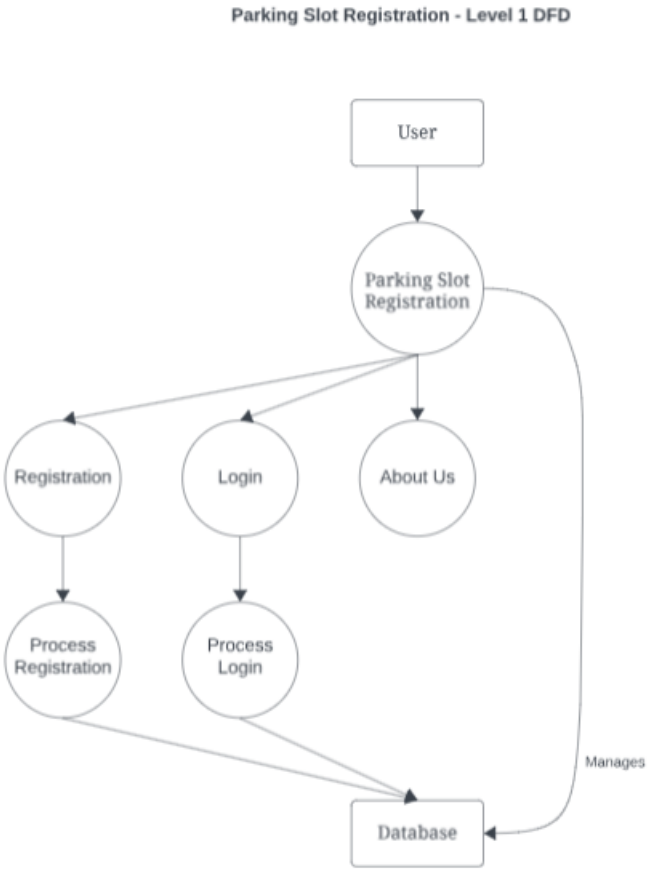


Figure-2

Level 2 (Process Registration)

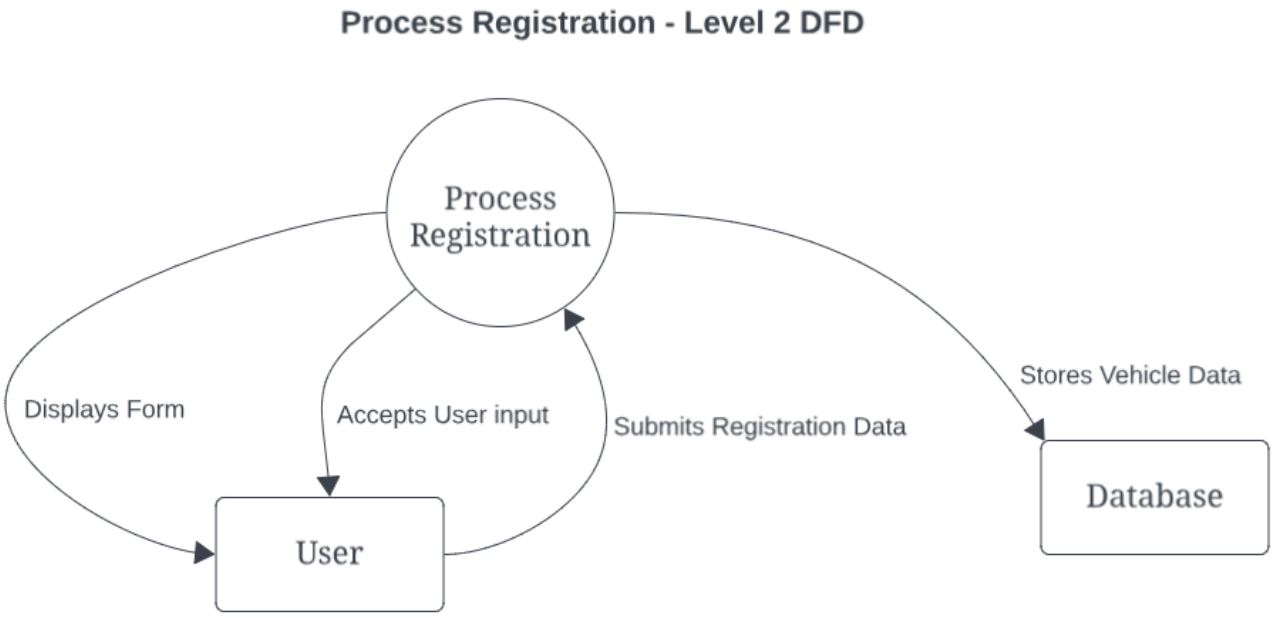


Figure-3

Level 2 (Process Login)

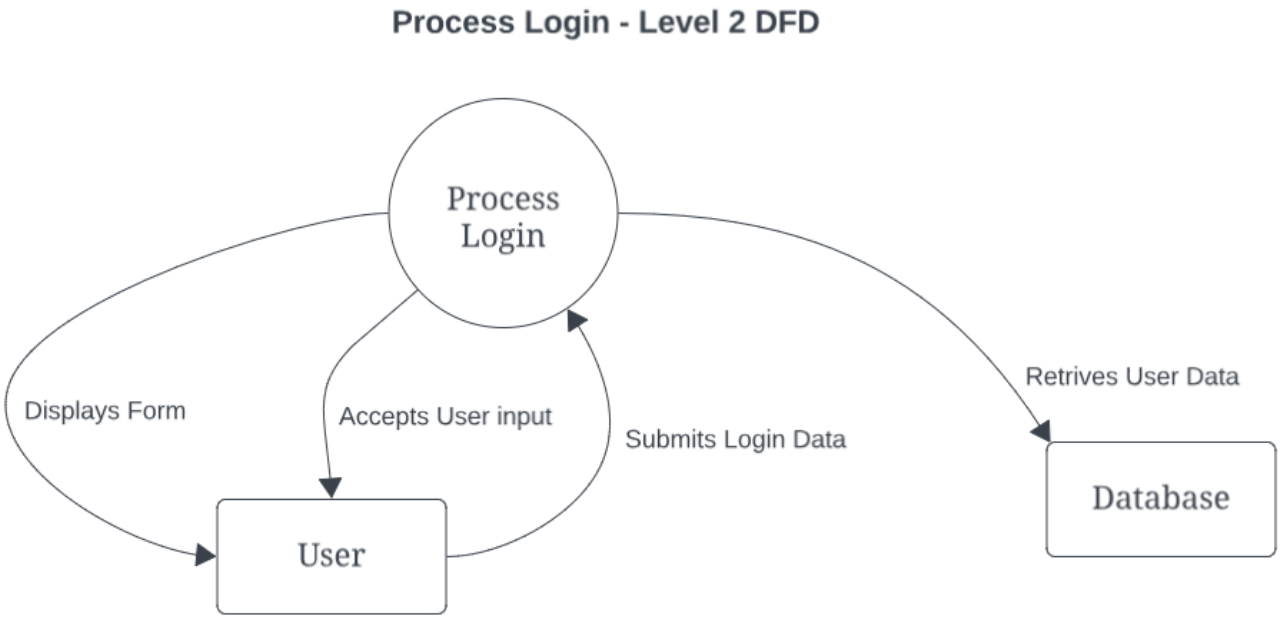


Figure-4

3.2 Entity Relationship Diagram (ERD)

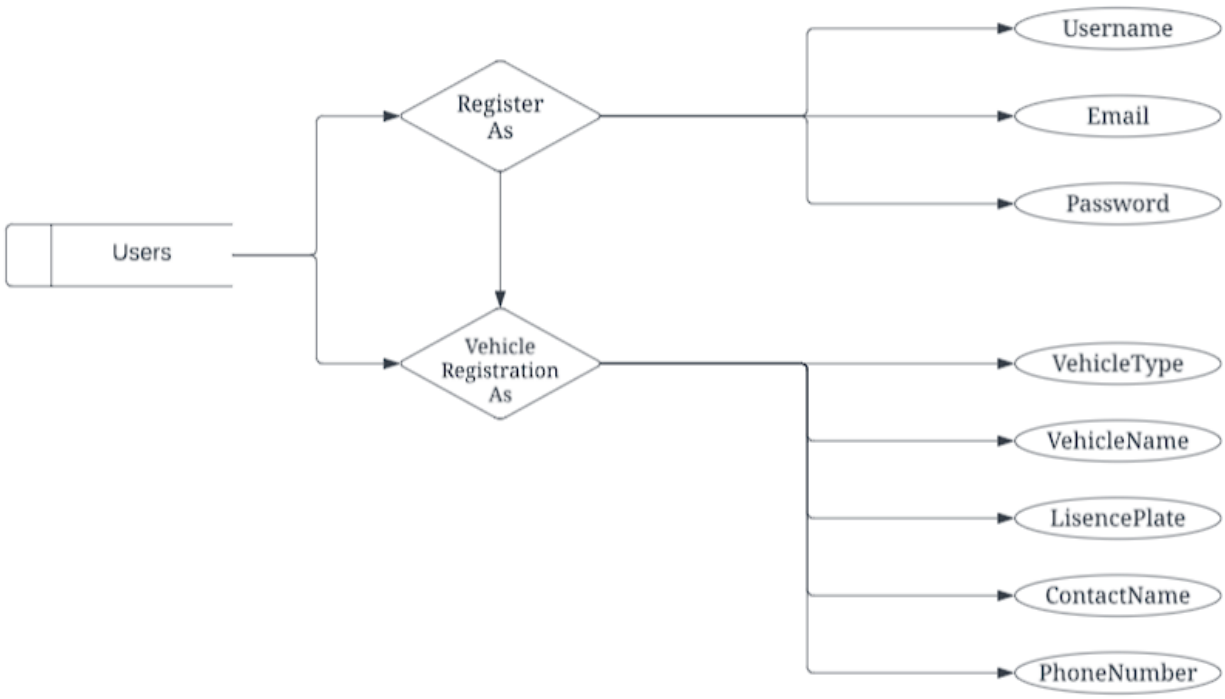


Figure-5

3.3 TABLE DESIGN

Parking Slot Registration is a software application that allows users to book their parking slot before as to avoid searching of parking in colleges to avoid congestions. Here is a sample database design for Parking Slot Registration system in SQL:

User Registration Table

The registration table will have the user registration details.

This table will contain the following fields:

Field name	Type	Attributes	Description
id	Int	Primary key	Unique identification for each user
username	Varchar	Null	Name of the user
email	Varchar	Null	Email of the user
password	Varchar	Null	Password for the account created

Vehicle Details Table

The Vehicle Details table will contain information about the vehicles registered.

This table will contain the following fields:

Field name	Type	Attributes	Description
id	int	Primary key	Unique identifier each registration
vehicleType	enum	Null	The type of vehicle as car or bike
vehicleName	varchar	Null	The name or model of the vehicle
lisencePlate	varchar	Null	The license plate number of the vehicle
contactName	varchar	Null	The name of the user of vehicle
phoneNumber	Int	Null	the price of product per unit

MODULE AND DESCRIPTION

Chapter 4
MODULES & DESCRIPTION

I. HOME

In this module, there are 4 buttons in this page. There is a home button, About-Us button, and Contact-Us button. The home button will redirect to 'homepage'. And the other two buttons redirect to the 'about-page'. Also there will be a Sign-in/Register button. This button will redirect to the Sign-in/Register page.

II. SIGN-IN/REGISTER PAGE

In this module, users are asked for login credentials if not they are redirected to the user register page where users have to provide details like username, email and password. This page has 3 button, Sign-in button redirects to 'vehicle register page', Register button redirects to 'Sign-in page' and home button redirects to 'home page'.

III. VEHICLE REGISTRATION

In this module, there are 5 message bars and a button. In the bars users are required to fill in the details of the vehicles like their type, model, license plate number also with the user's name and contact number. The Register button at the bottom redirects to the next module that is 'vehicle information' page.

IV. VEHICLE INFORMATION

In this module, there will be a table and some lines. The lines have details about the No. of vehicles registered, No. of available spaces, etc. The table contains the details of each and every vehicle's that has been registered like Type, Model, License plate Number and the contact details of the person like his/her Name and Contact Number.

TESTING USING SELENIUM

Chapter 5
TESTING USING SELENIUM

SELENIUM

Selenium is a free (open-source) automated testing framework used to validate web applications across different browsers and platforms. You can use multiple programming languages like Java, C#, Python, etc to create Selenium Test Scripts. Testing done using the Selenium testing tool is usually referred to as Selenium Testing.

TESTING

Test No	Task	Expected Output	Actual Output	Status
1.	Click on the "Sign in/Register" button	Sign in/Register button should be visible and clickable	Sign in/Register button was visible and clickable	Pass
2.	Click on the "Sign in" button	Sign in button should be visible and clickable	Login button was visible and clickable	Pass
3.	Enter a valid username and password	Username and password fields should be visible and clickable	Username and password fields was visible and clickable	Pass
4.	Verify that the "Sign-in" button is clickable.	The "Sign-in" button should be clickable and openable.	The "Sign-in" button was clickable and openable.	Pass

5.	Verify that the registration form is fillable.	The user should be able to fill his/her details in the registration form.	The user was able to fill his/her details in the registration form.	Pass
6.	Verify that the “Register” button is clickable.	Register button should be visible and clickable	Register button was visible and clickable	Pass
7.	The Database should be Visible.	The user should be able to see the Registered Vehicles and its details.	The user was able to see the Registered Vehicles and its details.	Pass
8.	Verify that the user can “Logout”	The user should be able to logout of their account without errors	The user was able to logout of their account without any errors	Pass

SCREENSHOT:

Figure-1: Home Page

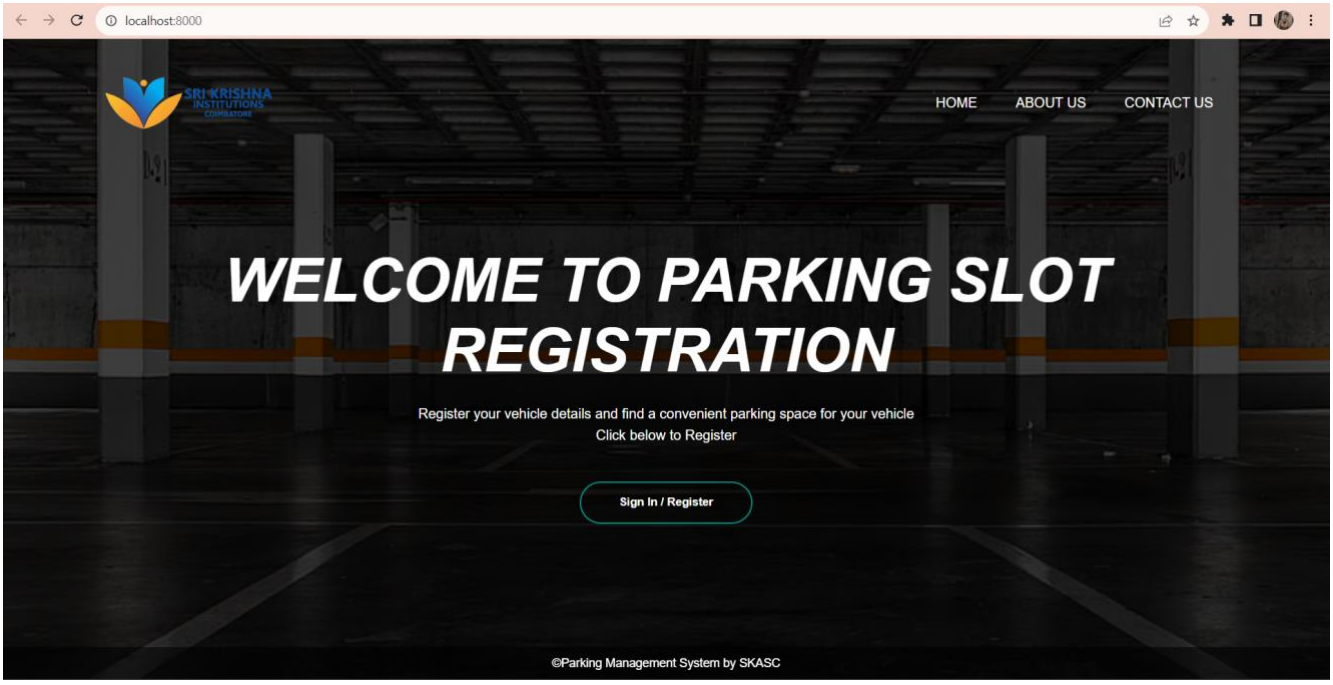
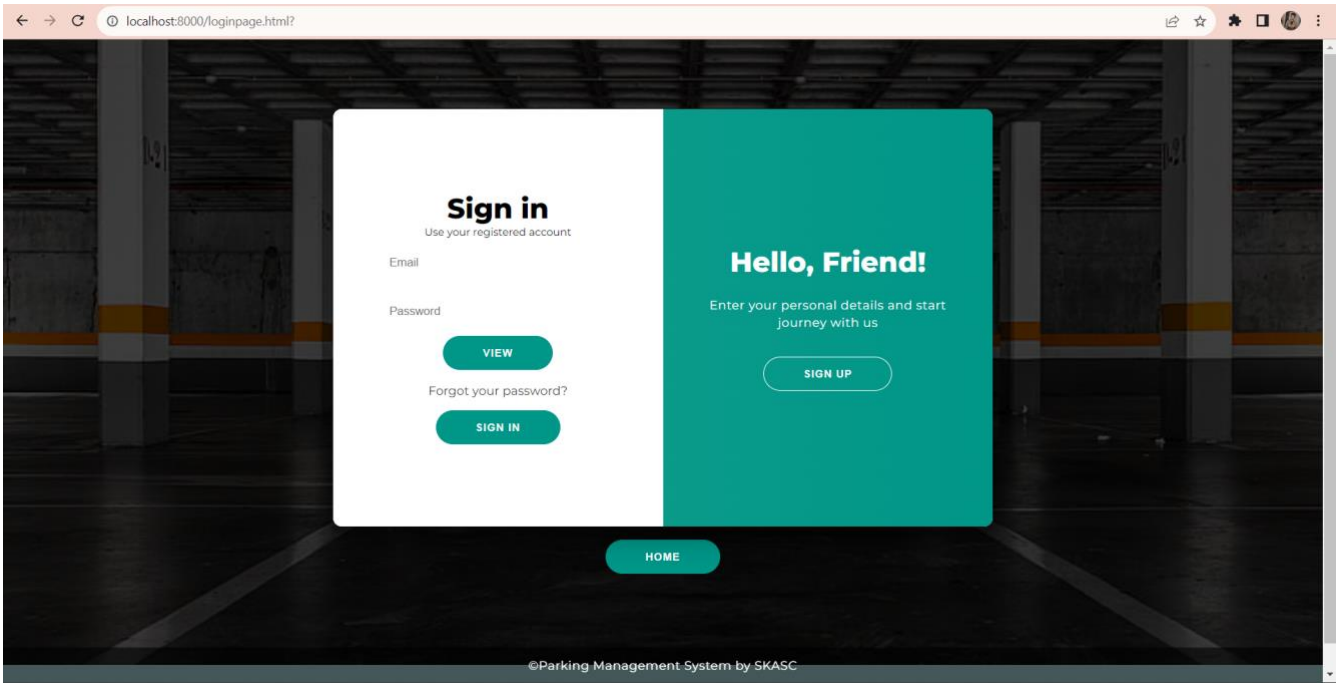


Figure-2: Login Page



RESULT AND IMPLEMENTATION

Chapter 6
RESULT & IMPLEMENTATION

RESULT

Finally, the website is successfully created and opened properly. And also testing result of the website is done and successful. This HTML (static) website will helpful for many students and users in a very efficient manner to park their vehicles based on their needs. An implementation of the website also was successful too.

IMPLEMENTATION

The first step is creating a home page with details about the website along with the Sign in/Register button, also homepage has been included with the About-Us button, Home and Contact-Us button. Second step is creating a Login\Sign-up page for the website. Third step is creating a registration page for filling the vehicle details and inserting buttons to redirect into next page for viewing the vehicle details. Fourth step is displaying details of the registered vehicles, No. of spaces available, etc. to the registered users.

Figure-1: index.html(homepage)

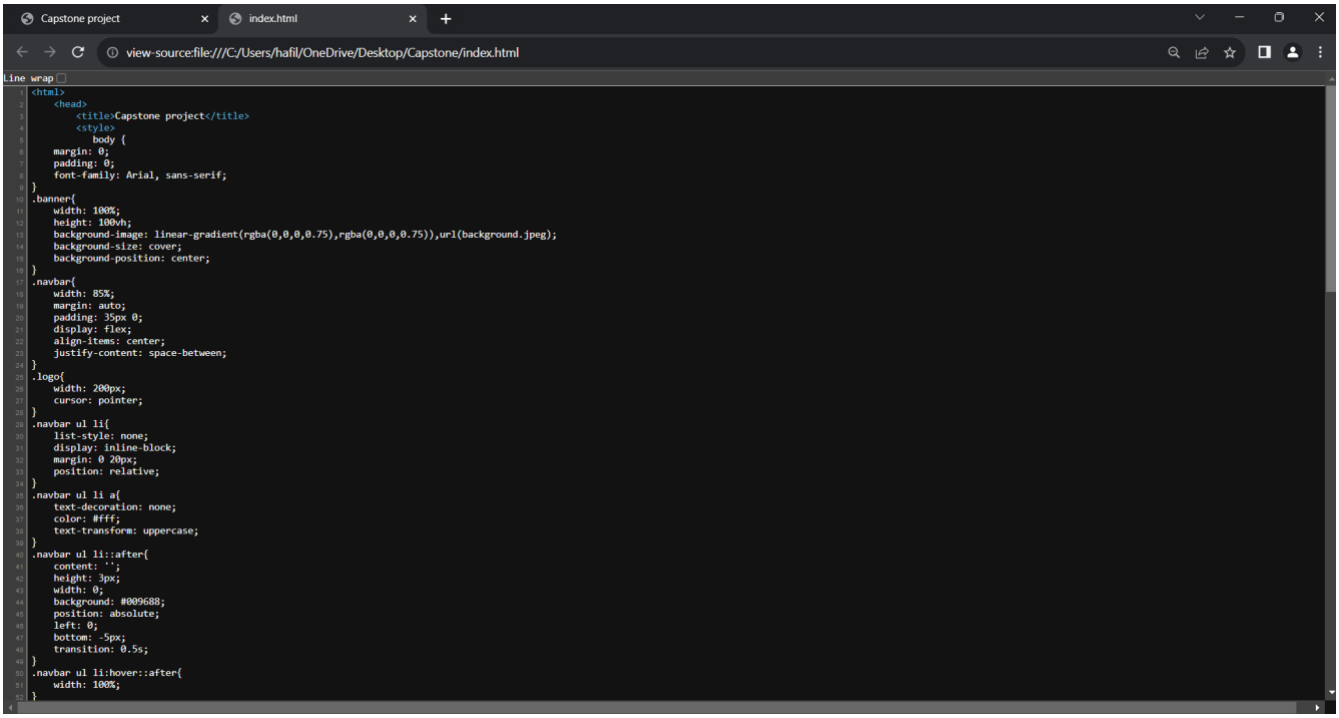


Figure-2: index.html(homepage)

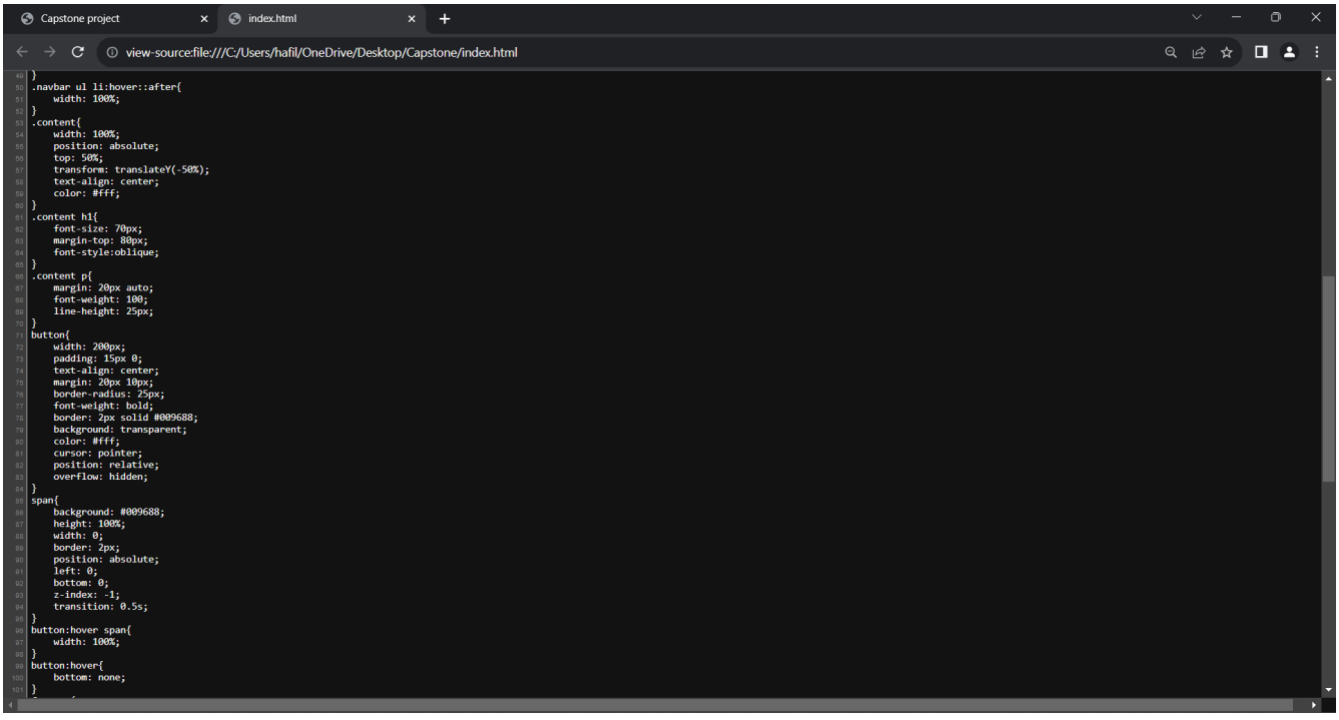
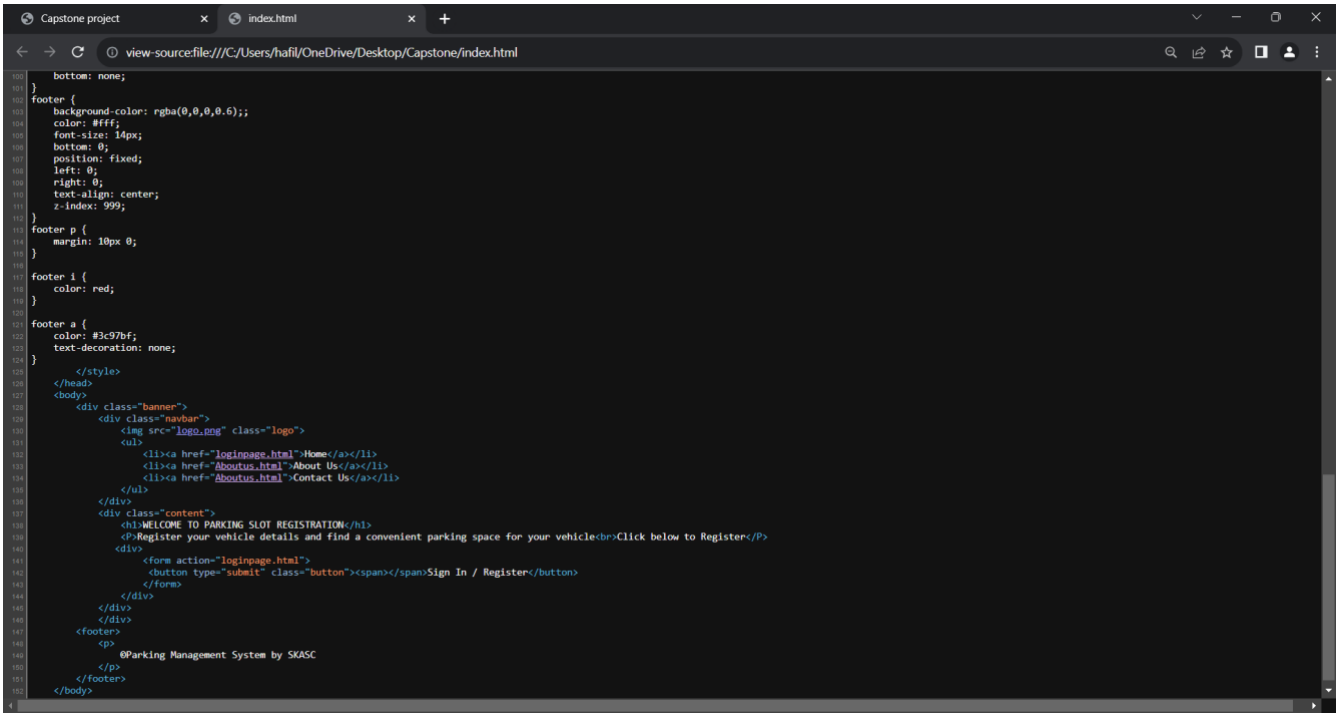
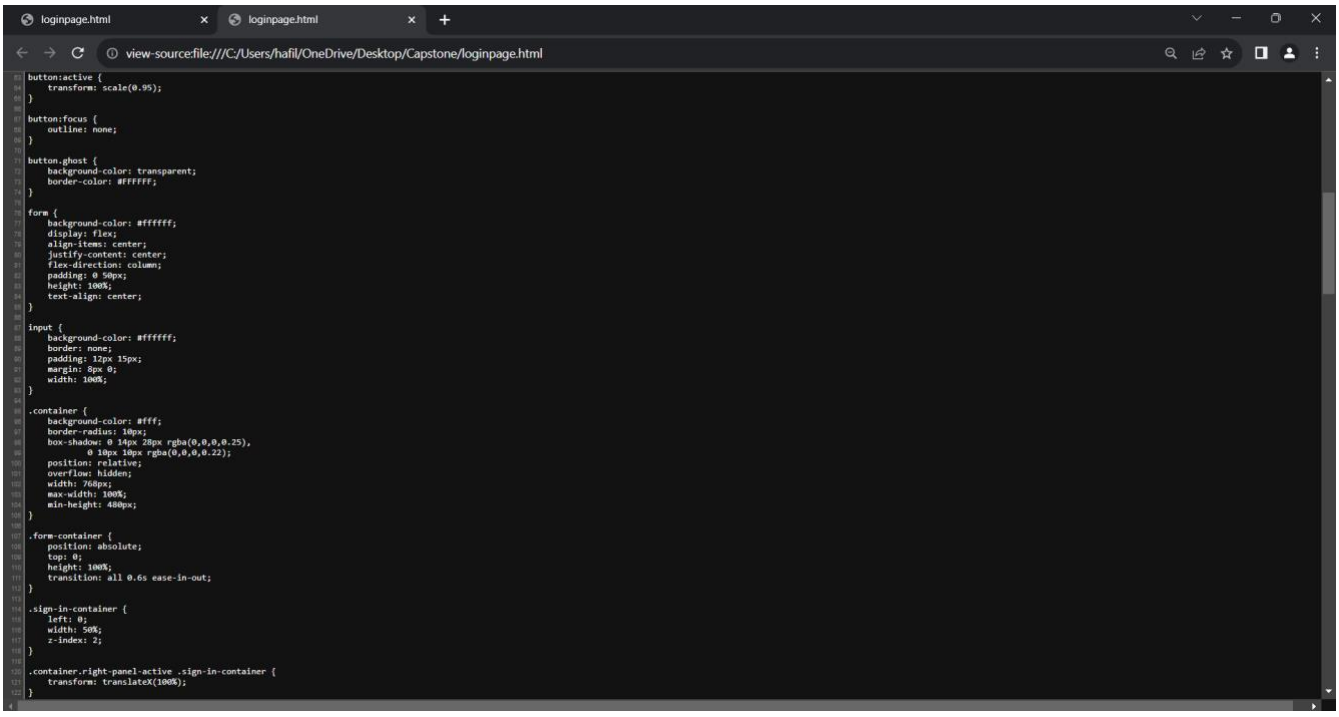


Figure-3: index.html(homepage)




```
    bottom: none;
}
footer {
    background-color: rgba(0,0,0,0.6);;
    color: #fff;
    font-size: 14px;
    bottom: 0;
    position: fixed;
    left: 0;
    right: 0;
    text-align: center;
    z-index: 999;
}
footer p {
    margin: 10px 0;
}

footer i {
    color: red;
}

footer a {
    color: #3c97bf;
    text-decoration: none;
}
        </style>
    </head>
    <body>
        <div class="banner">
            <div class="navbar">
                <img src="logo.png" class="logo">
                <ul>
                    <li><a href="loginpage.html">Home</a></li>
                    <li><a href="Aboutus.html">About Us</a></li>
                    <li><a href="Aboutus.html">Contact Us</a></li>
                </ul>
            </div>
            <div class="content">
                <h1>WELCOME TO PARKING SLOT REGISTRATION</h1>
                <P>Register your vehicle details and find a convenient parking space for your vehicle<br>Click below to Register</P>
               <div>
                    <form action="loginpage.html">
                     <button type="submit" class="button"><span></span>Sign In / Register</button>
                    </form>
                </div>
            </div>
            </div>
        <footer>
            <p>
                @Parking Management System by SKASC
            </p>
        </footer>
    </body>
```

Figure-4: loginpage.html(Loginpage)




```
button:active {
    transform: scale(0.95);
}

button:focus {
    outline: none;
}

button.ghost {
    background-color: transparent;
    border-color: #FFFFFF;
}

form {
    background-color: #ffffff;
    display: flex;
    align-items: center;
    justify-content: center;
    flex-direction: column;
    padding: 0 50px;
    height: 100%;
    text-align: center;
}

input {
    background-color: #ffffff;
    border: none;
    padding: 12px 15px;
    margin: 8px 0;
    width: 100%;
}

.container {
    background-color: #fff;
    border-radius: 10px;
    box-shadow: 0 14px 28px rgba(0,0,0,0.25),
            0 10px 10px rgba(0,0,0,0.22);
    position: relative;
    overflow: hidden;
    width: 768px;
    max-width: 100%;
    min-height: 480px;
}

.form-container {
    position: absolute;
    top: 0;
    height: 100%;
    transition: all 0.6s ease-in-out;
}

.sign-in-container {
    left: 0;
    width: 50%;
    z-index: 2;
}

.container.right-panel-active .sign-in-container {
    transform: translateX(100%);
}
```

Figure-5: loginpage.html(Loginpage)

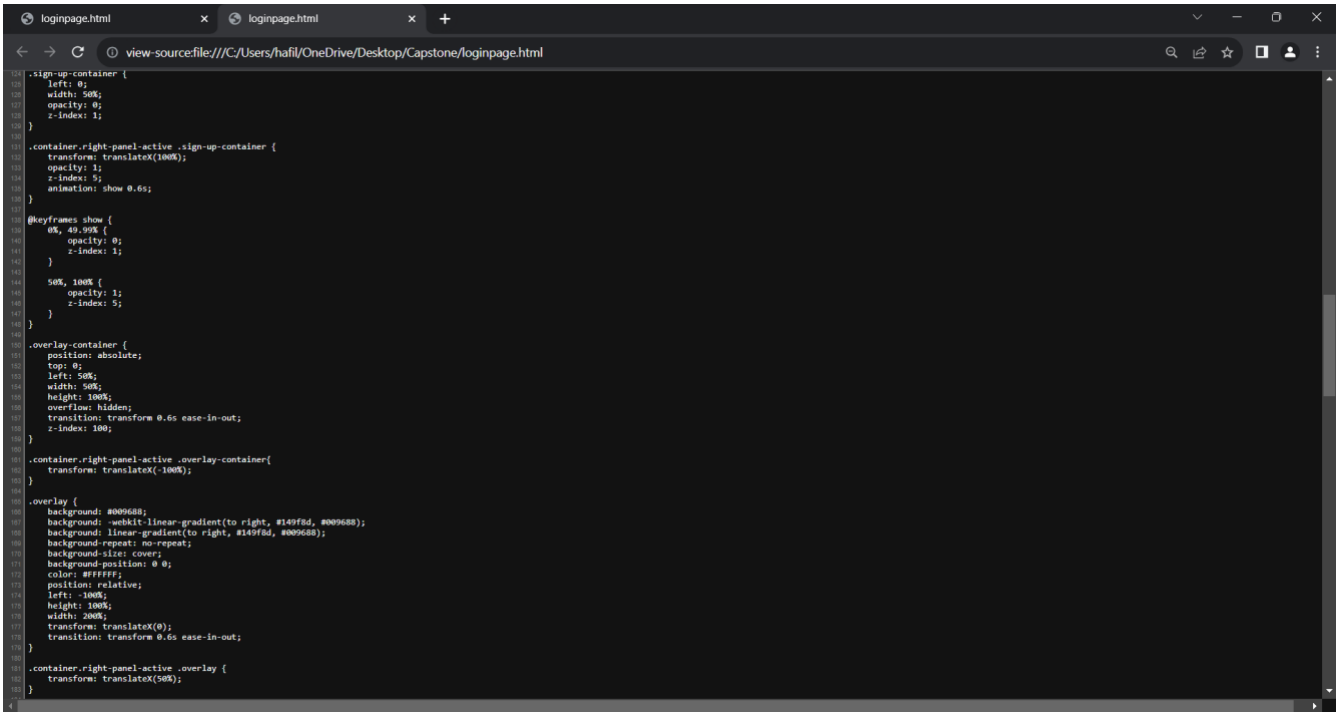


Figure-6: loginpage.html(Loginpage)

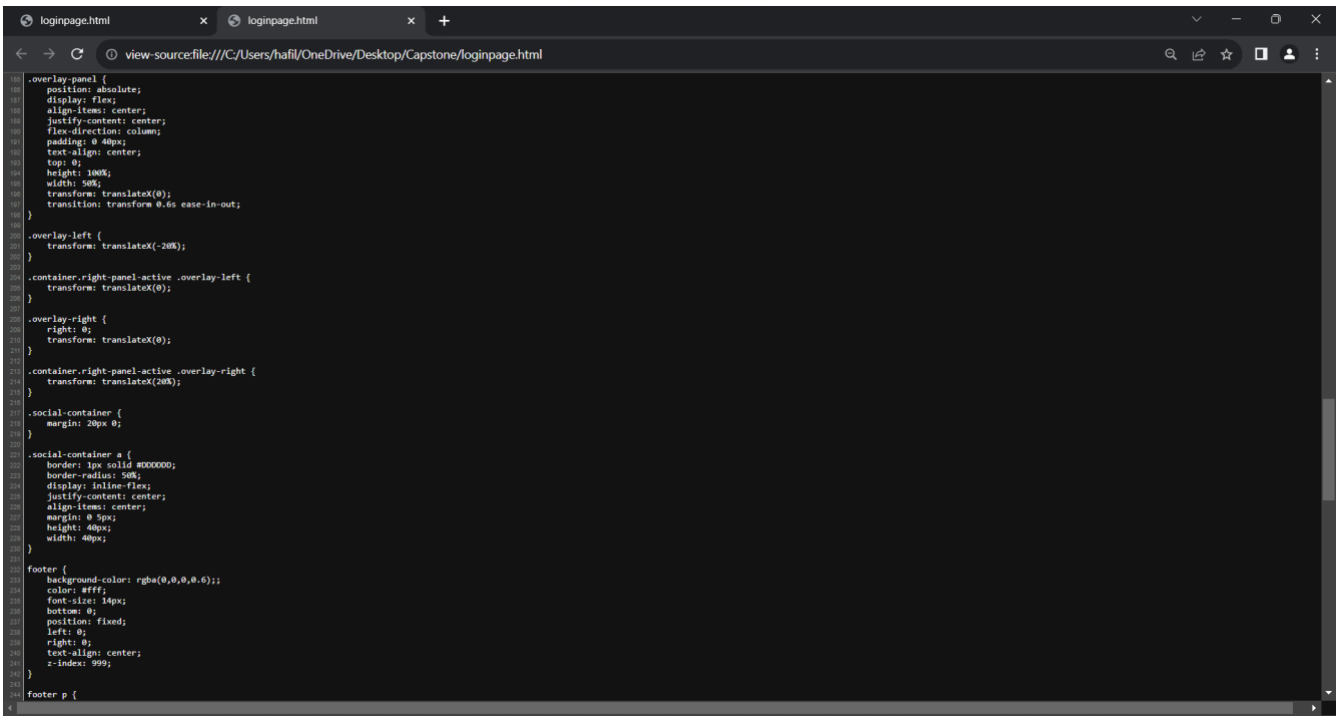
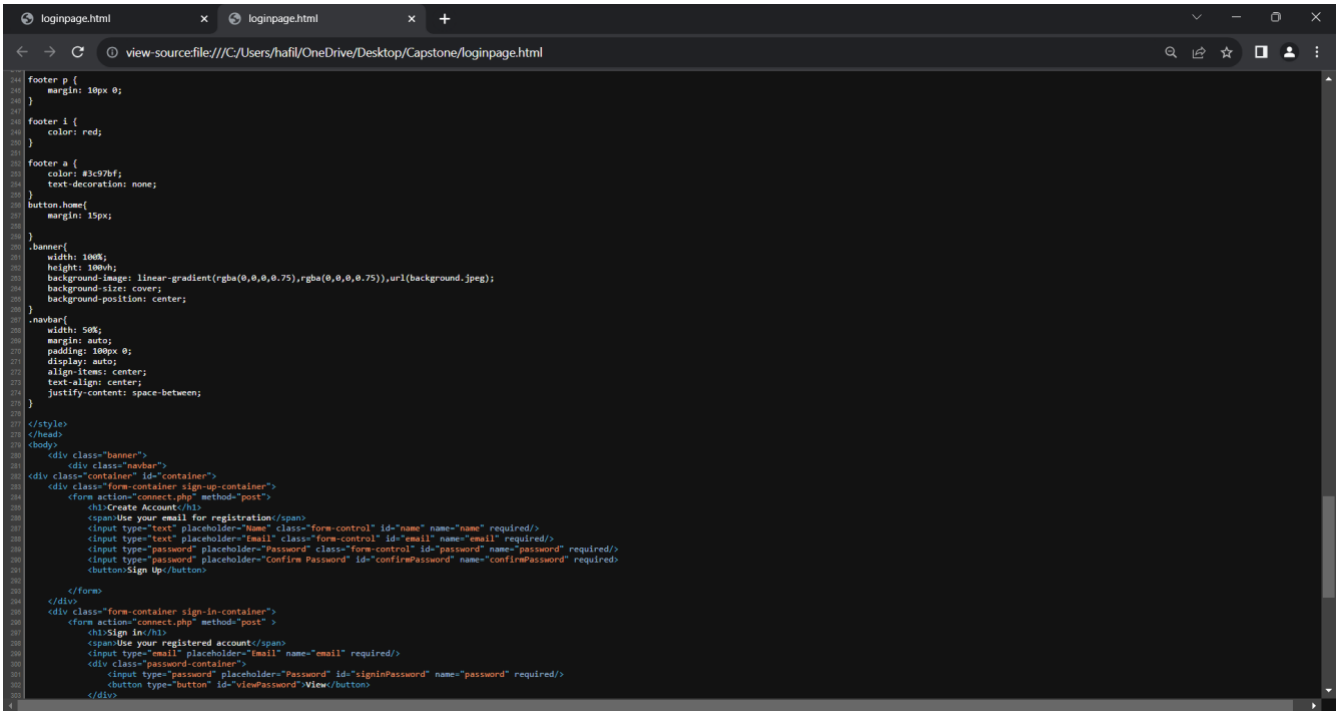
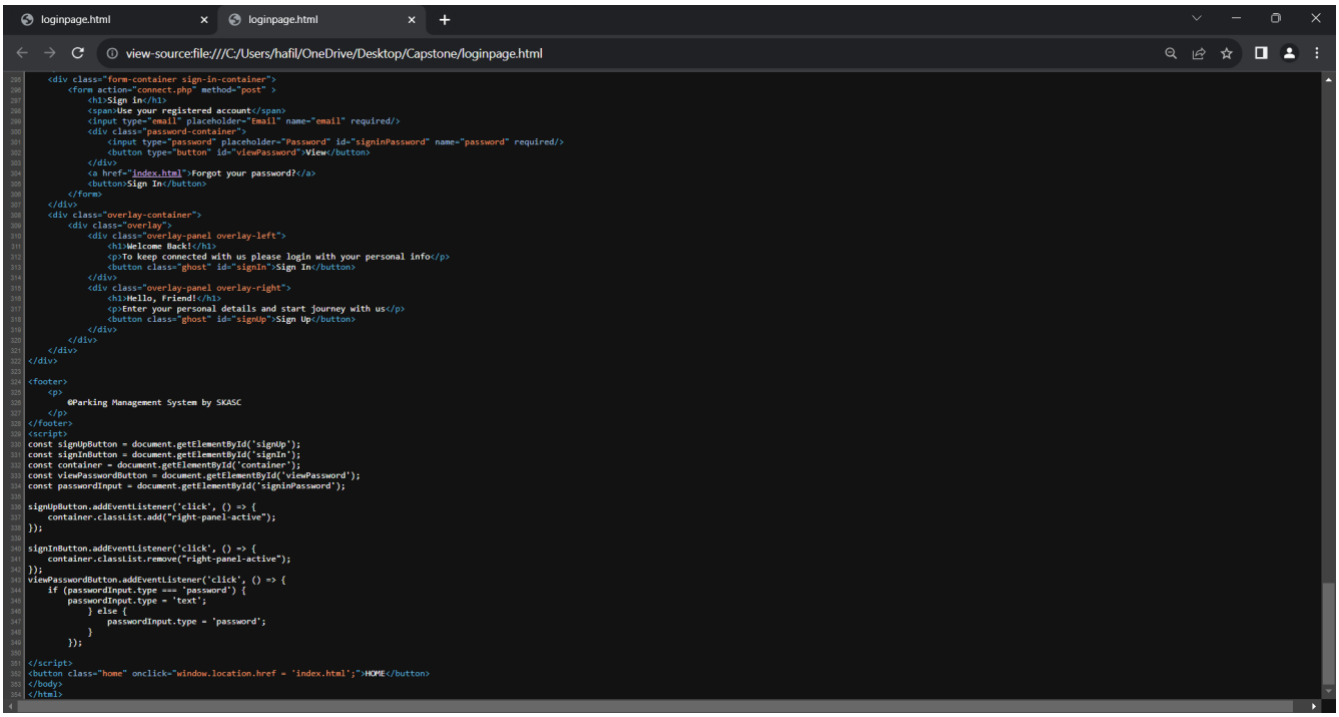


Figure-7: loginpage.html(Loginpage)




```
footer p {
    margin: 10px 0;
}

footer i {
    color: red;
}

footer a {
    color: #3c97bf;
    text-decoration: none;
}
button.home{
    margin: 15px;

}
.banner{
    width: 100%;
    height: 100vh;
    background-image: linear-gradient(rgba(0,0,0,0.75),rgba(0,0,0,0.75)),url(background.jpeg);
    background-size: cover;
    background-position: center;
}
.navbar{
    width: 50%;
    margin: auto;
    padding: 100px 0;
    display: auto;
    align-items: center;
    text-align: center;
    justify-content: space-between;
}

</style>
</head>
<body>
    <div class="banner">
        <div class="navbar">
<div class="container" id="container">
    <div class="form-container sign-up-container">
        <form action="connect.php" method="post">
            <h1>Create Account</h1>
            <span>Use your email for registration</span>
            <input type="text" placeholder="Name" class="form-control" id="name" name="name" required/>
            <input type="text" placeholder="Email" class="form-control" id="email" name="email" required/>
            <input type="password" placeholder="Password" class="form-control" id="password" name="password" required/>
            <input type="password" placeholder="Confirm Password" id="confirmPassword" name="confirmPassword" required>
            <button>Sign Up</button>

        </form>
    </div>
    <div class="form-container sign-in-container">
        <form action="connect.php" method="post" >
            <h1>Sign in</h1>
            <span>Use your registered account</span>
            <input type="email" placeholder="Email" name="email" required/>
            <div class="password-container">
                <input type="password" placeholder="Password" id="signinPassword" name="password" required/>
                <button type="button" id="viewPassword">View</button>
            </div>
```

Figure-8: loginpage.html(Loginpage)




```
    <div class="form-container sign-in-container">
        <form action="connect.php" method="post" >
            <h1>Sign in</h1>
            <span>Use your registered account</span>
            <input type="email" placeholder="Email" name="email" required/>
            <div class="password-container">
                <input type="password" placeholder="Password" id="signinPassword" name="password" required/>
                <button type="button" id="viewPassword">View</button>
            </div>
            <a href="index.html">Forgot your password?</a>
            <button>Sign In</button>
        </form>
    </div>
    <div class="overlay-container">
        <div class="overlay">
            <div class="overlay-panel overlay-left">
                <h1>Welcome Back!</h1>
                <p>To keep connected with us please login with your personal info</p>
                <button class="ghost" id="signIn">Sign In</button>
            </div>
            <div class="overlay-panel overlay-right">
                <h1>Hello, Friend!</h1>
                <p>Enter your personal details and start journey with us</p>
                <button class="ghost" id="signUp">Sign Up</button>
            </div>
        </div>
    </div>
</div>

<footer>
    <p>
        @Parking Management System by SKASC
    </p>
</footer>
<script>
const signUpButton = document.getElementById('signUp');
const signInButton = document.getElementById('signIn');
const container = document.getElementById('container');
const viewPasswordButton = document.getElementById('viewPassword');
const passwordInput = document.getElementById('signinPassword');

signUpButton.addEventListener('click', () => {
    container.classList.add("right-panel-active");
});

signInButton.addEventListener('click', () => {
    container.classList.remove("right-panel-active");
});
viewPasswordButton.addEventListener('click', () => {
    if (passwordInput.type === 'password') {
        passwordInput.type = 'text';
        } else {
            passwordInput.type = 'password';
        }
    });

</script>
<button class="home" onclick="window.location.href = 'index.html';">HOME</button>
</body>
</html>
```

CONCLUSION

Chapter 7
CONCLUSION

The primary objective of this project was to see how HTML, CSS, JavaScript and PHP can be used to make a static website. This project is just an example, because this idea is most useful in public places like mainly Hospitals, Airports, Railway stations, Museums, Apartments, Schools and many other important public places. It makes the person to get information about parking's in many places easily. So, it reduces the search of parking spaces and helps in time management.

In conclusion, it provides an efficient solution for managing parking spaces in a campus setting. The project's user-friendly interface and robust functionalities make it a valuable tool for both administrators and users.

SAMPLE CODE

Chapter 8
APPENDIX

SAMPLE CODE

loginpage.html

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="loginpage.css">
</head>
<body>
<div class="banner">
<div class="navbar">
<div class="container" id="container">
<div class="form-container sign-up-container">
<form action="connect.php" method="post">
<h1>Create Account</h1>
<span>Use your email for registration</span>
<input type="text" placeholder="Name" class="form-control" id="name" name="name"
required/>
<input type="text" placeholder="Email" class="form-control" id="email" name="email"
required/>
<input type="password" placeholder="Password" class="form-control" id="password"
name="password" required/>
<input type="password" placeholder="Confirm Password" id="confirmPassword"
name="confirmPassword" required>
<button>Sign Up</button>
</form>
</div>
<div class="form-container sign-in-container">
<form action="connect.php" method="post" >
<h1>Sign in</h1>
<span>Use your registered account</span>
<input type="email" placeholder="Email" name="email" required/>
<div class="password-container">
<input type="password" placeholder="Password" id="signinPassword" name="password"
required/>
<button type="button" id="viewPassword">View</button>
</div>
<a href="index.html">Forgot your password?</a>
<button>Sign In</button>
</form>
</div>
<div class="overlay-container">
<div class="overlay">
<div class="overlay-panel overlay-left">
<h1>Welcome Back!</h1>
<p>To keep connected with us please login with your personal info</p>
```

```
<button class="ghost" id="signIn">Sign In</button>
</div>
<div class="overlay-panel overlay-right">
<h1>Hello, Friend!</h1>
<p>Enter your personal details and start journey with us</p>
<button class="ghost" id="signUp">Sign Up</button>
</div>
</div>
</div>
</div>
<footer>
<p>
©Parking Management System by SKASC
</p>
</footer>
<script src="loginpage.js"></script>
<button class="home" onclick="window.location.href = 'index.html';">HOME</button>
</body>
</html>
```

loginpage.css

```
@import url('https://fonts.googleapis.com/css?family=Montserrat:400,800');
* {
box-sizing: border-box;
}
body {
background: #a8dddc;
display: flex;
justify-content: center;
align-items: center;
flex-direction: column;
font-family: 'Montserrat', sans-serif;
height: 100vh;
margin: -20px 0 50px;
}
h1 {
font-weight: bold;
margin: 0;
}
h2 {
text-align: center;
}
p {
font-size: 14px;
font-weight: 100;
line-height: 20px;
letter-spacing: 0.5px;
margin: 20px 0 30px;
}
span {
```

```
font-size: 12px;
}
a {
color: #333;
font-size: 14px;
text-decoration: none;
margin: 15px 0;
}
button {
border-radius: 20px;
border: 1px solid #009688;
background-color: #009688;
color: #FFFFFF;
font-size: 12px;
font-weight: bold;
padding: 12px 45px;
letter-spacing: 1px;
text-transform: uppercase;
transition: transform 80ms ease-in;
}
button:active {
transform: scale(0.95);
}
button:focus {
outline: none;
}
button.ghost {
background-color: transparent;
border-color: #FFFFFF;
}
form {
background-color: #ffffff;
display: flex;
align-items: center;
justify-content: center;
flex-direction: column;
padding: 0 50px;
height: 100%;
text-align: center;
}
input {
background-color: #ffffff;
border: none;
padding: 12px 15px;
margin: 8px 0;
width: 100%;
}
.container {
background-color: #fff;
border-radius: 10px;
box-shadow: 0 14px 28px rgba(0,0,0,0.25),
```

```
0 10px 10px rgba(0,0,0,0.22);
position: relative;
overflow: hidden;
width: 768px;
max-width: 100%;
min-height: 480px;
}
.form-container {
position: absolute;
top: 0;
height: 100%;
transition: all 0.6s ease-in-out;
}
.sign-in-container {
left: 0;
width: 50%;
z-index: 2;
}
.container.right-panel-active .sign-in-container {
transform: translateX(100%);
}
.sign-up-container {
left: 0;
width: 50%;
opacity: 0;
z-index: 1;
}
.container.right-panel-active .sign-up-container {
transform: translateX(100%);
opacity: 1;
z-index: 5;
animation: show 0.6s;
}
@keyframes show {
0%, 49.99% {
opacity: 0;
z-index: 1;
}
50%, 100% {
opacity: 1;
z-index: 5;
}
}
.overlay-container {
position: absolute;
top: 0;
left: 50%;
width: 50%;
height: 100%;
overflow: hidden;
transition: transform 0.6s ease-in-out;
```

```
z-index: 100;
}
.container.right-panel-active .overlay-container{
transform: translateX(-100%);
}

.overlay {
background: #009688;
background: -webkit-linear-gradient(to right, #149f8d, #009688);
background: linear-gradient(to right, #149f8d, #009688);
background-repeat: no-repeat;
background-size: cover;
background-position: 0 0;
color: #FFFFFF;
position: relative;
left: -100%;
height: 100%;
width: 200%;
transform: translateX(0);
transition: transform 0.6s ease-in-out;
}

.container.right-panel-active .overlay {
transform: translateX(50%);
}

.overlay-panel {
position: absolute;
display: flex;
align-items: center;
justify-content: center;
flex-direction: column;
padding: 0 40px;
text-align: center;
top: 0;
height: 100%;
width: 50%;
transform: translateX(0);
transition: transform 0.6s ease-in-out;
}

.overlay-left {
transform: translateX(-20%);
}

.container.right-panel-active .overlay-left {
transform: translateX(0);
}

.overlay-right {
right: 0;
```

```
transform: translateX(0);
}

.container.right-panel-active .overlay-right {
transform: translateX(20%);
}

.social-container {
margin: 20px 0;
}

.social-container a {
border: 1px solid #DDDDDD;
border-radius: 50%;
display: inline-flex;
justify-content: center;
align-items: center;
margin: 0 5px;
height: 40px;
width: 40px;
}

footer {
background-color: rgba(0,0,0,0.6);;
color: #fff;
font-size: 14px;
bottom: 0;
position: fixed;
left: 0;
right: 0;
text-align: center;
z-index: 999;
}

footer p {
margin: 10px 0;
}

footer i {
color: red;
}

footer a {
color: #3c97bf;
text-decoration: none;
}
button.home{
margin: 15px;

}
.banner{
```

```
width: 100%;
height: 100vh;
background-image: linear-gradient(rgba(0,0,0,0.75),rgba(0,0,0,0.75)),url(background.jpeg);
background-size: cover;
background-position: center;
}
.navbar{
width: 50%;
margin: auto;
padding: 100px 0;
display: auto;
align-items: center;
text-align: center;
justify-content: space-between;
}
```

loginpage.js

```
const signUpButton = document.getElementById('signUp');
const signInButton = document.getElementById('signIn');
const container = document.getElementById('container');
const viewPasswordButton = document.getElementById('viewPassword');
const passwordInput = document.getElementById('signinPassword');

signUpButton.addEventListener('click', () => {
container.classList.add("right-panel-active");
});

signInButton.addEventListener('click', () => {
container.classList.remove("right-panel-active");
});
viewPasswordButton.addEventListener('click', () => {
if (passwordInput.type === 'password') {
passwordInput.type = 'text';
} else {
passwordInput.type = 'password';
}
});
```

connect.php

```
<?php
error_reporting(E_ALL);
ini_set('display_errors', 1);

if ($_SERVER["REQUEST_METHOD"] == "POST") {
if (isset($_POST["name"]) && isset($_POST["email"]) && isset($_POST["password"])) {
// Registration logic
$name = $_POST['name'];
$email = $_POST['email'];
$password = $_POST['password'];
```

```
// Hash the password (use a secure password hashing library like password_hash)
$hashed_password = password_hash($password, PASSWORD_DEFAULT);

$conn = new mysqli('localhost', 'root', '', 'logindetails');
if ($conn->connect_error) {
die('Connection Failed : ' . $conn->connect_error);
}

$stmt = $conn->prepare("INSERT INTO users (name, email, password) VALUES (?, ?, ?)");
if (!$stmt) {
die('Prepare Failed : ' . $conn->error);
}

$stmt->bind_param("sss", $name, $email, $password);
if (!$stmt->execute()) {
die('Execute Failed : ' . $stmt->error);
}
$stmt->close();
$conn->close();
// Registration successful, redirect to login page
header("Location: loginpage.html");
exit;
}
}
$email = $_POST["email"];
$password = $_POST["password"];
// Database connection
$con = new mysqli("localhost","root","","logindetails");
if($con->connect_error) {
die("Failed to connect :".$con->connect_error);
}else{
$stmt = $con->prepare("select * from users where email = ?");
$stmt->bind_param("s", $email);
$stmt->execute();
$stmt_result = $stmt->get_result();
if($stmt_result->num_rows > 0){
$data = $stmt_result->fetch_assoc();
if($data['password'] === $password){
echo"<h2>Login Successful</h2>";
header("Location: veh_reg.html");
exit;
}else{
echo "<h2>Invalid Email or Password</h2>";
}
} else {
echo "<h2>Invalid Email or Password</h2>";
}
}
?>
```

SCREENSHOTS

SCREENSHOTS

Figure-1: Home Page

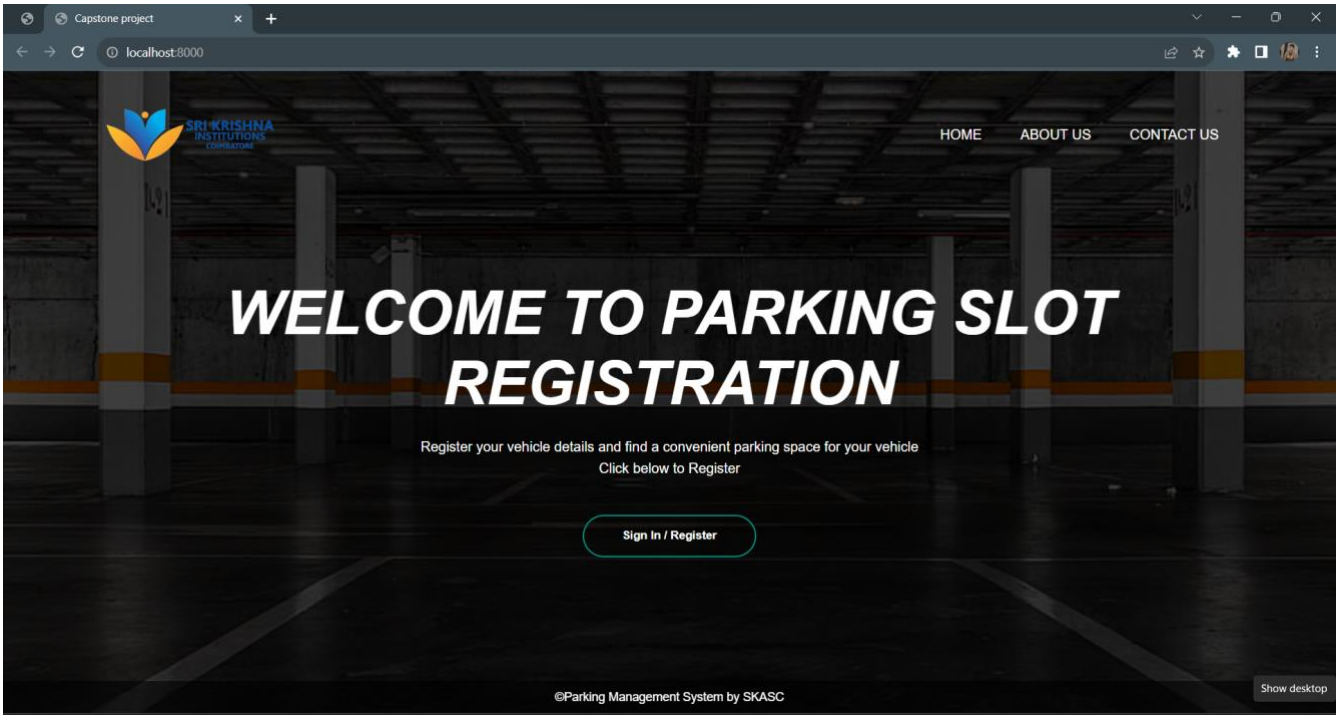


Figure-2: About Us Page

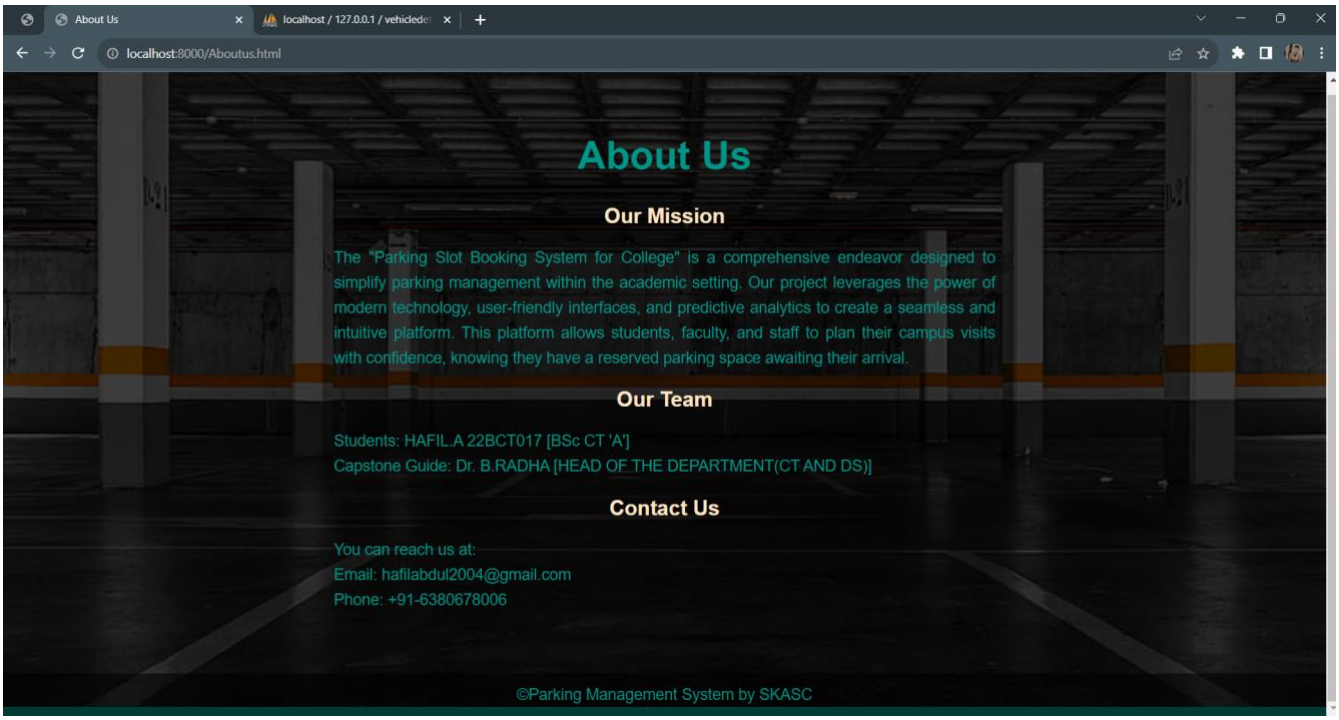


Figure-3: Login Page

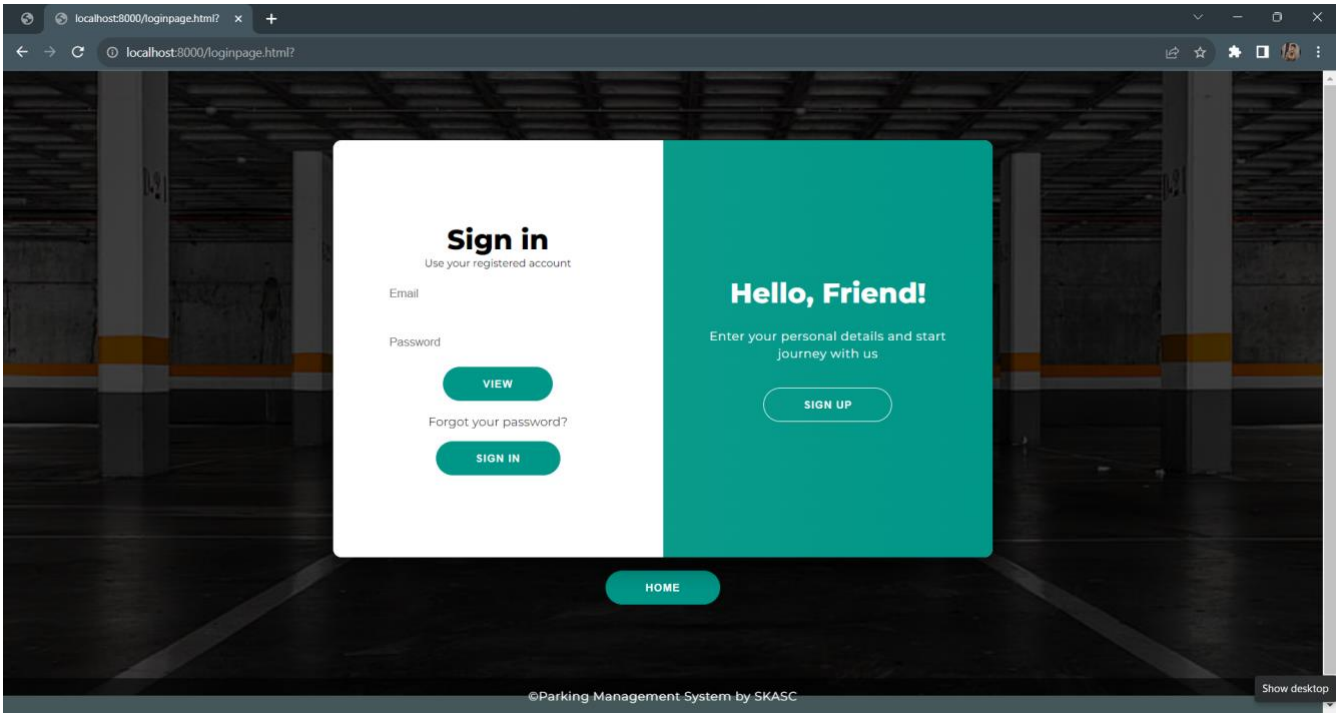


Figure-4: User Register Page

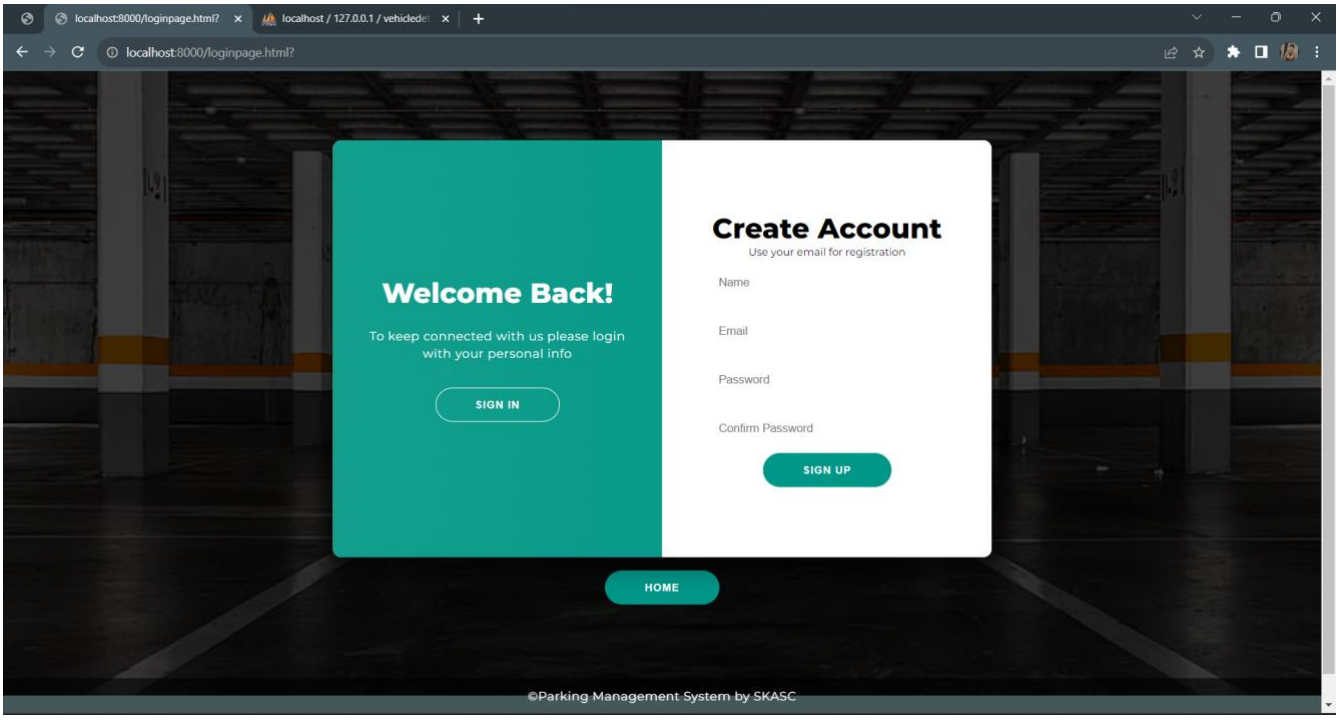


Figure-5: Parking Registration Page

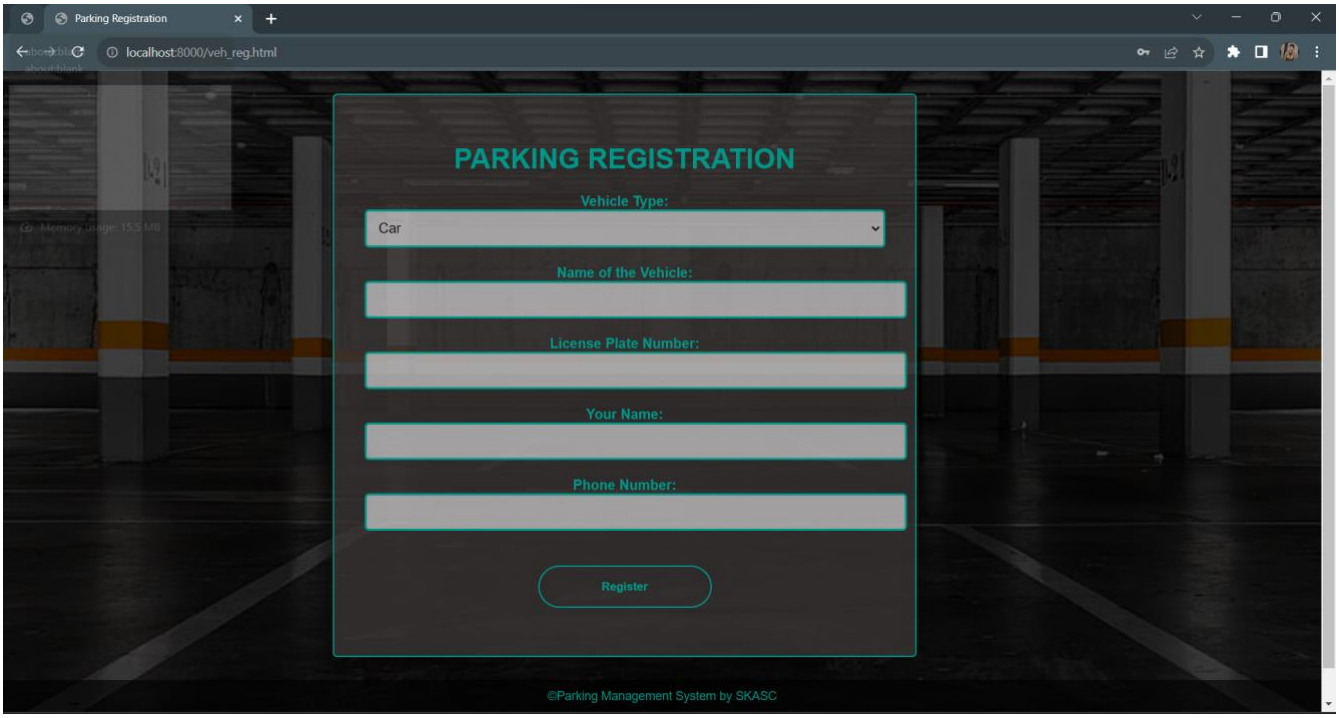
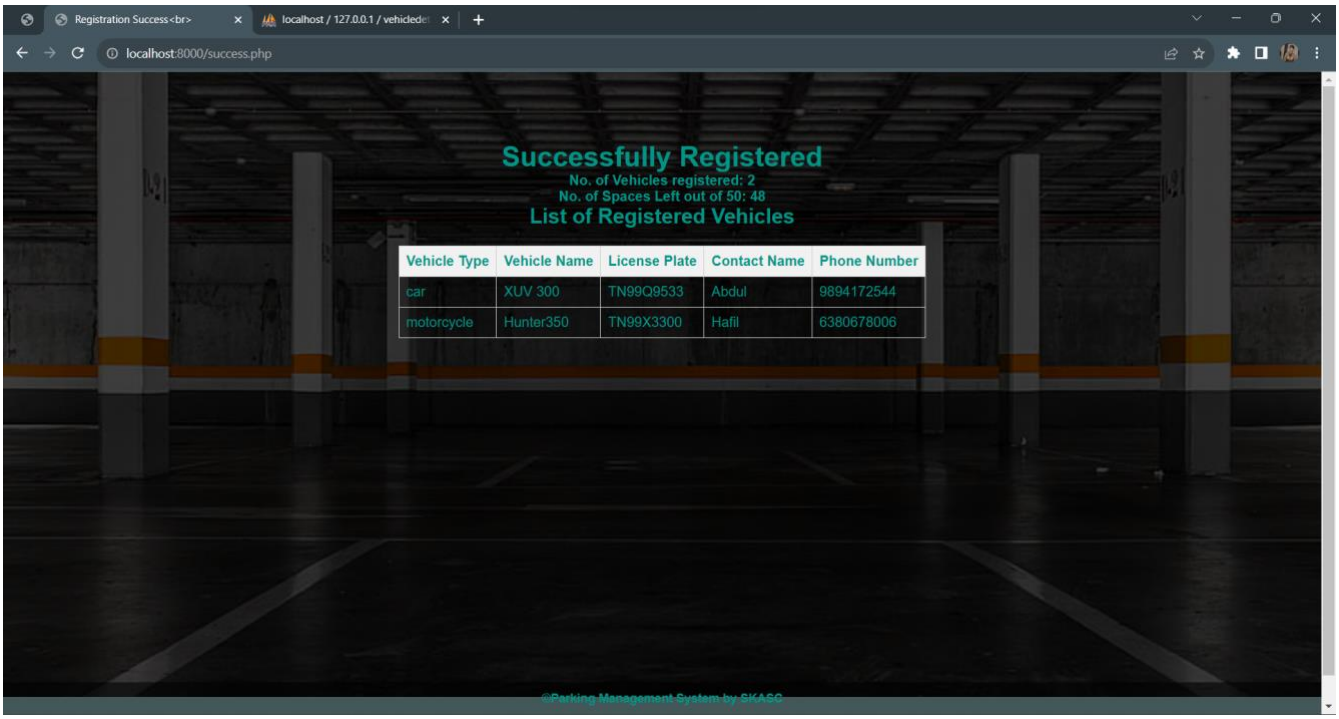


Figure-6: Vehicle Details Page



BIBLIOGRAPHY

Chapter 9
BIBLIOGRAPHY

- Books

1. 'PHP and MySQL for Dummies' by Paul DuBois
2. 'HTML, CSS, and JavaScript All in One' by Jon Duckett
3. 'MySQL in a Nutshell' by Russell J.T. Dyer

- Websites

1. PHP.net: https://www.php.net/
2. MySQL.com: https://www.mysql.com/
3. W3Schools: https://www.w3schools.com/
4. GeeksforGeeks: https://www.geeksforgeeks.org/

- Articles

1. How to Build a Parking Slot Management System with PHP and MySQL by TutorialFreak: https://m.youtube.com/watch?v=Dn9SG3zzFZI

2. Developing a Smart Parking System Using PHP and MySQL by GeeksforGeeks: https://phpgurukul.com/vehicle-parking-management-system-using-php-and-mysql/

3. Building a Parking Management System with HTML, CSS, MySQL, and PHP by CodeProject: https://www.sourcecodester.com/php-codeigniter-simple-parking-management-system-source-code