PARKING SLOT BOOKING SYSTEM

CSE1005 - Software Engineering

FACULTY : K. RAJA SRAVAN KUMAR

TEAM NUMBER : 2

TEAM MEMBERS :

NAME :	REG.NO :
PALACHARLA MOHAN SUBHASH	20BCD7193
P. VARUN SAI	20BCD7190
B. SANTOSHI REDDY	20BCE7019
AYUSHI BAJPAI	20BCD7149

INDEX

S NO	TITLE	PAGE NO
1	INTRODUCTION	3-4
2	LITERATURE SURVEY	5
3	PROBLEM STATEMENT	6
4	OBJECTIVE	6
5	FEASIBILITY STUDY	7
6	FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS	8-11
7	METHODOLOGY AND DESIGN	12-18
8	SAMPLE CODE AND OUTPUTS	19-38
9	TESTING	39-43
10	RESULTS DISCUSSION	44
11	FUTURE SCOPE	44
12	REFERENCE	44

TABLE-1

1.INTRODUCTION

1.1 Purpose

1.2 Intended Audience

1.3 Intended Use

1.4 Analysis

1.5 Description of Individual Module

1.1 PURPOSE :

The purpose of this project is to provide a software solution that delivers an easy and reliable system to large buildings and multiplexes that manages the parking system optimally. This document will give information about the features of the Parking slot booking system and the requirements needed by the project in developing the software for the organizations.

1.2 INTENDED AUDIENCE :

This document can used by developers for understanding the requirements of the project. The intended audience of this document are project managers and developers for planning and scheduling. Testing team to generate test cases, document writers for preparation of user manual.

1.3 INTENDED USE :

The main use of our project Parking slot booking System is that people will definitely get benefited in terms of saving money and saving time because now a day's traffic is becoming the major issue. So, we want to create a portal so that all the people who have work near that parking space can book slots before starting at their home and they can come at that time and park their vehicles.

1.4.PROBLEM STATEMENT

The main aim of our project Parking Slot Booking System is to make a portal for the people to book the parking slots at a particular space. We are planning this project such a way that people will definitely get benefited in terms of saving money and saving time because now a day's traffic is becoming the major issue in mega cities because most of the people are parking their vehicles on the roads, so due to this traffic is becoming a huge problem.So, we want to create a portal so that all the people who have work near that

parking space canbook slots before starting at their home and then they can come at that time and park their vehicles. So, here people are saving time by booking their slots nearby a complex instead of searching the space on the roads. So, we are planning to make this project more efficient for the people use by dividing into some modules.

We are planning to create two portals one is user portal to book parking slots and another portal is admin portal to add parking slots, update the parking slot details and also to update the details of attendants at parking slots.

1.5 OBJECTIVE

The objective of the project helps the customers to find a parking slot very quickly and makes it easy for them to park in a large building. This project is focussed at managing the parking slots in a complex optimally for both the customers and the employees working there. Now-a-days people are busy with their different schedules. Parking on road side, waiting for parking slot to become free leading to waste of time. This software reduces a lot of paperwork and enables the customer to do everything mostly online in these tough times of the pandemic. People are not getting full detailed receipt of how much time they are spending and how much it will cost. So, we are planning to make this project more efficient for the people use by dividing into some modules. We are planning to create two portals one is user portal to book parking slots and another portal is admin portal to add parking slots, update the parking slot details and also to update the details of attendants at parking slots.

1.6 ANALYSIS :

Looking for parking is an activity which is usually consider a waste of time because now a day's traffic is becoming the major issue in mega cities because most of the people are parking their vehicles on the roads, so due to this traffic is becoming a huge problem. So, we are planning to make this project more efficient for the people use by dividing into some modules.

1.7 INFRASTRUCTURE:

HARDWARE: Processer i5-1065G7,

RAM: 8GB,

SSD:512GB, 64 bit operating system

SOFTWARE : Windows os,

xampp server,

php my admin

Visual Studio.Net Framework.

2.0 SOFTWARE REQUIREMENT ANALYSIS AND PLANNING:

2.1 DESCRIPTION OF INDIVIDUAL MODULE :

A.USER PORTAL MODULES

1.Login Module:

We are planning to create a login page which contains a username and password to sign for the people to book parking slots and also for the new users there will be a sign up option to create account in our portal and then book parking slots.

2.Slot booking Module:

There will be a parking slot booking module which is to book slots in different floors like ground floor, first floor, second floor etc…. and also this module consists of options like hours to be booked for a particular parking slot and also calculate the cost for number of hours booked and also we are planning to show how many remaining slots are still available at a particular floor to book for the users.

3.Logout Module:

This module will be useful to sign out from the web portal after booking parking slot.

B.ADMIN PORTAL MODULES

1.Login Module:

We are planning to create a login page which contains a username and password to sign for the admins to Check all the details like how many slots are booked in each floor, who booked parking slots for how many hours etc….

2.Parking details Module:

In this module, we are planning to include the details of parking slots like location, street name, adding new parking slots, adding more slots if space is available at a particular floor and we can appoint attendant for each floor, fix the cost of each parking slot per hour.

3.Attendant details Module :

In this module, we are planning to add all the details of attendants working at parking slots. So, in this module we can see name of attendant, mobile number of attendants, his working floor in the complex. So, here admin can see all the details of working attendants and also admin can add or delete or update the details of attendants.

4.Requests Module :

This is the module where admin can see all the requests and accept them. So, that the users request for parking slot will be booked and also in this module along with the username who have requested for parking slot we are planning to create to show how many hours the user requested and what will be the cost to be paid by customer and also in which floor did the user requested for parking slot and also we are planning to create a function which will download all the entries in pdf format.

5.FEASIBILITY STUDY

5.1 Technical Feasibility

5.2 Economic Feasibility

5.3 Operational Feasibility

5.1 Technical Feasibility :

Operating System	Windows
Programming Languages	HTML, CSS, PHP
IDE	XAMPP
Database	MYSQL
Web Browser	Chrome

TABLE-3

Design mechanisms are limited to the capabilities of Html, php and MySQL DB. The user interface must be viewable on a monitor with a 1024x768 resolution or larger. Safety and Security Considerations Personal information of parking details and attendant details information will be contained, so a separate portal is created for admins and users and so that personal and sensitive information are not accessed by unauthorized users.

These technologies are freely available. We can say our project is technically feasible.

5.2 Economic Feasibility :

PARKING SLOT BOOKING SYSTEM is to help society. So it doesn't require any extra charges like data transfer, bandwidth required for this application will costs very low cost.

PARKING SLOT BOOKING SYSTEM will have an associated hosting cost. Customers are not charged for subscription after login in this system. Hence, we can say SPMS is Economically Feasible.

5.3 Operational Feasibility :

The services for PARKING SLOT BOOKING SYSTEM are very flexible. This system will mostly target the people who want to book a parking slot. There will be no labour required as this is completely online. The most important thing is it saves time for people not to search for parking slot on roads.

Therefore, it is feasible.

6.FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

6.1 Functional Requirements

6.2 External Interface Requirements

6.3 Non-Functional Requirements

6.1 FUNCTIONAL REQUIREMENTS

A. ADMIN PORTAL

Login feature:

- Accept the admin user name password.
- A case insensitive comparison is done for the admin user name and a case sensitive comparison is done for a password.
- If the admin user id and password are correct then, Dashboard will be displayed.
- If admin enters an invalid user id or password is entered then the system will display error message “Incorrect username or password” and should quit the application.

Dashboard:

- Parking maintenance menu: Insert, update or view records from parking details database.
- Attendant maintenance menu: Add, delete, modify or view records in Attendant details database.
- Requests menu: Displays all the requests made by users regarding parking slot booking and generate reports to the attendants with all the details of user slot booking details from end users.

Add Parking details:

- All fields are mandatory.
- Location can contain only alphabets.
- Floor names will not be duplicated.
- Parking slot no will be automatically generated by the system.
- No of remaining slots will automatically get reduced and generated by system after customer slot booking.

- Update Parking details:
- Other Attendant can be updated at particular floor parking's if the allotted attendant is not available.
- Price per hour can be updated half-yearly or yearly.
- When the user finishes updating the price, attendant, floor name, no of slots etc.., the control should display the "view all parking details" menu and display all the parking details with updated details.

Delete Parking details:

- This feature will delete all parking details at a particular floor if that floor is in maintenance.

View all Parking details:

- This feature will display all the parking details in a tabular format.

Add Attendant details:

- Attendant first name and last name can be added.
- Attendant mobile number will be added without duplicated number.
- Attendant address can be added in this module.
- Attendant working location like his working floor no will be added in this menu.

Update Attendant details:

- Attendant mobile number can be updated in this module.
- Attendant address can be updated in this module.
- Attendant working location i.e floor can be updated according to availability of attendants.

Delete Attendant details:

- Attendant details will be deleted in this feature if he resigns to work.

View all Attendant details:

- This feature will display all the Attendants details in a tabular format.

View all Requests:

- This feature will display complete details regarding the user requests in parking floor, no of hours the user requested for a particular slot and the cost to be paid by customer will be displayed beside the username of customer in a tabular format.

B. USER PORTAL

Login feature:

- Accept the customer user name password.
- A case insensitive comparison is done for the customer user name and a case sensitive comparison is done for a password.
- If the customer user id and password are correct then, Slot booking menu will be displayed.
- If customer enters an invalid user id or password is entered then the system will display error message “Incorrect username or password” and should quit the application.

Slot booking menu:

- In this feature, we can book parking slot in a floor and we can select parking slot at a particular floor, no of hours required per slot and then this feature will automatically display the amount to be paid by the user and if the user confirms then the slot will be allotted under that user.
- In the same menu, we will show the no of remaining slots to the new user once the previous user confirms a parking slot.

6.2 EXTERNAL INTERFACE REQUIREMENTS

A. USER INTERFACES

Admin Portal Interface

Login Interface :

This admin interface prompts the admin to authenticate his/her credentials by asking username and password, so that only authorized admins can login to the portal. Also it displays “Incorrect username or password” if the given credentials are wrong.

Parking details Maintenance :

This interface provides the admin users with the option to add a new parking slots, update the existing parking details and cost for a slot per hour, and also view details of all the existing parking’s.

Attendant details Maintenance :

This interface provides the admin users with the option to add a new attendants, update the existing attendant details like updating attendant mobile number and attendant address , and also view details of all the existing Attendants.

Requests details Maintenance :

This interface provides the admin users with the option to view all the requests made by the customers in user portal. This interface can be used to display all the requests by customers with the floor no and no of hours a customer requested. This interface is also used to display a detailed report in pdf format with all the entries made by the customer in User portal.

User portal interface

Login Interface :

This user interface prompts the user to authenticate his/her credentials by asking username and password, so that only authorized users can log in to the portal. Also it displays “Incorrect username or password” if the given credentials are wrong.

Slot Booking Maintenance :

This interface provides the users with the option to book parking slots with no of hours a parking slot needed and the floor number where the customers wants slot can also be booked and then this feature is used to display the cost to be paid by customer for a slot according to the hours a customer requested.

B. HARDWARE INTERFACES

This software does not require any direct hardware interfaces.

C. SOFTWARE INTERFACES

- Windows operating system can be used as a platform for development and deployment.
- XAMPP version 3.2.4 can be used as interface to connect to frontend and backend.
- Html, CSS can be used for frontend development.
- MySQL can be used for backend development.

D. COMMUNICATION NTERFACES

This software is a standalone system hence does not require any communication interfaces.

6.3 NON-FUNCTIONAL REQUIREMENTS

1.PERFORMANCE REQUIREMENTS

- The Parking Slot Booking system can be able to run on a standalone computer.
- As the database grows the intended performance with respect to report generation and accessing of data is not going to be affected.

2.AVAILABILITY REQUIREMENTS

- This software is completely replacing the existing system which some of the parking areas are still in manual process. So, the system must be able to provide accurate data to all the users with valid credentials.
- The scheduled maintenance time will be taken for maximum of 3hrs per week during midnights where the server will be free at that time and make the software unavailable.

3.TRAINING REQUIREMENTS

- The GUI should be user friendly and provide self-learning with a training time of 3-6 hrs.

7.SYSTEM ARCHITECTURE AND DESIGN

7.1 Data Flow Diagram

7.2 Usecase Diagram

7.3 Class Diagram

7.4 Activity Diagram

7.5 Sequential Diagram

7.1 DATA FLOW DIAGRAM

LEVEL 0 :

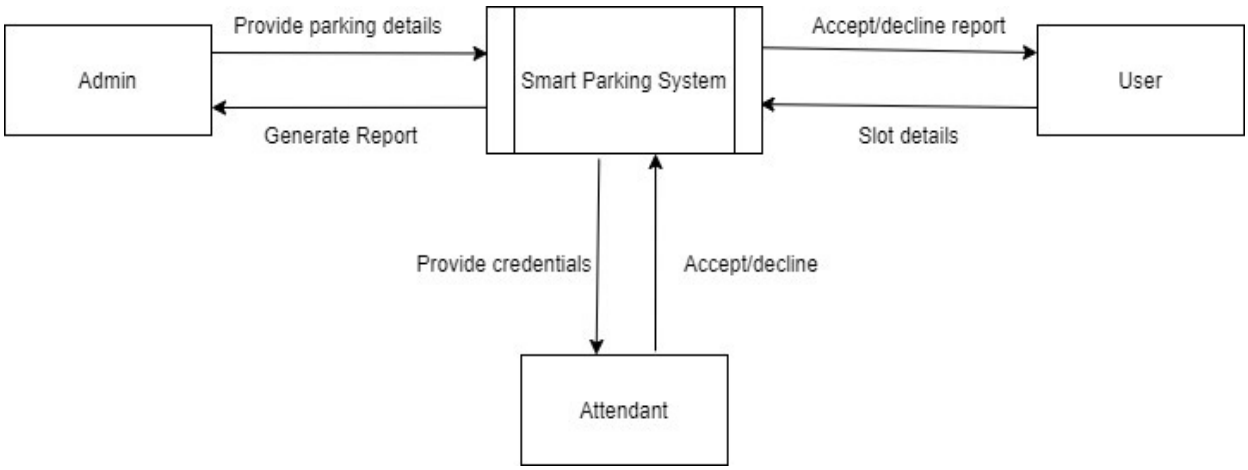

Figure-1

LEVEL 1 :

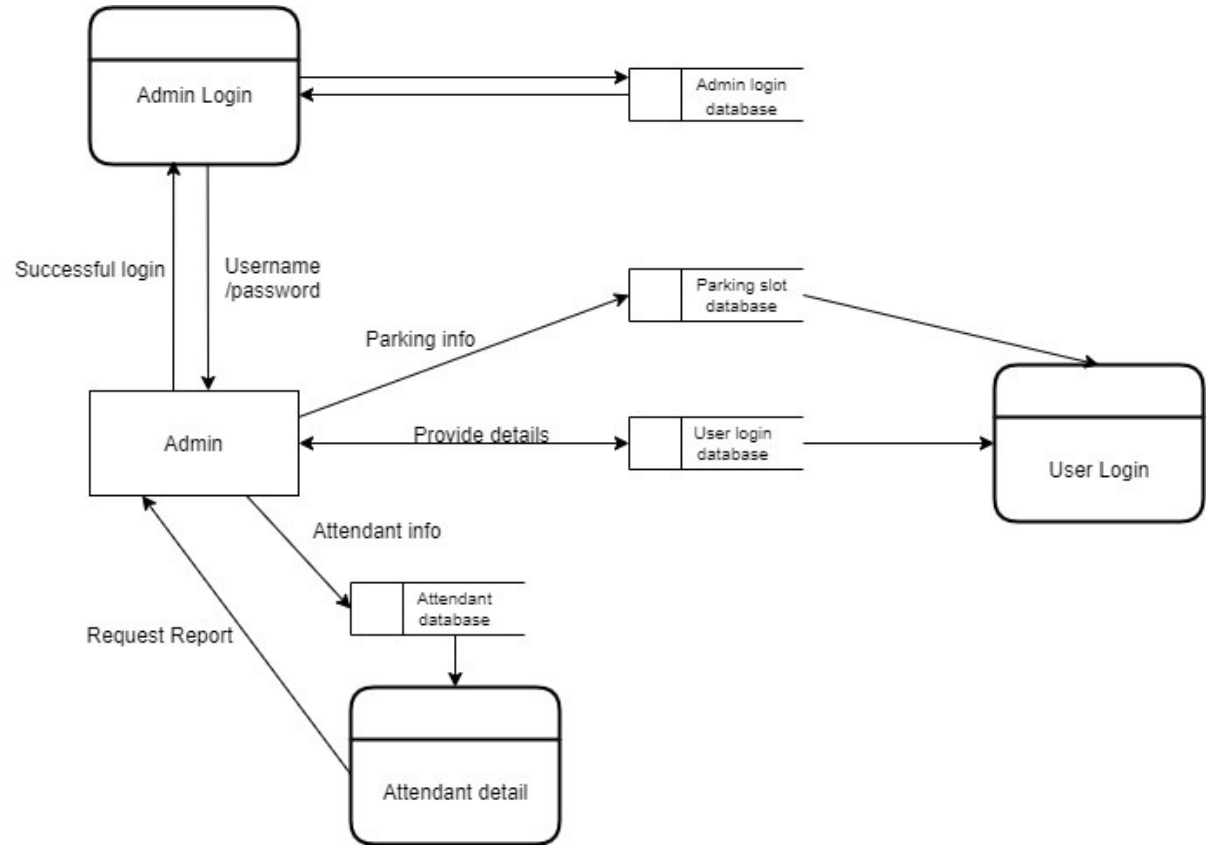

Figure-2

LEVEL 2 :

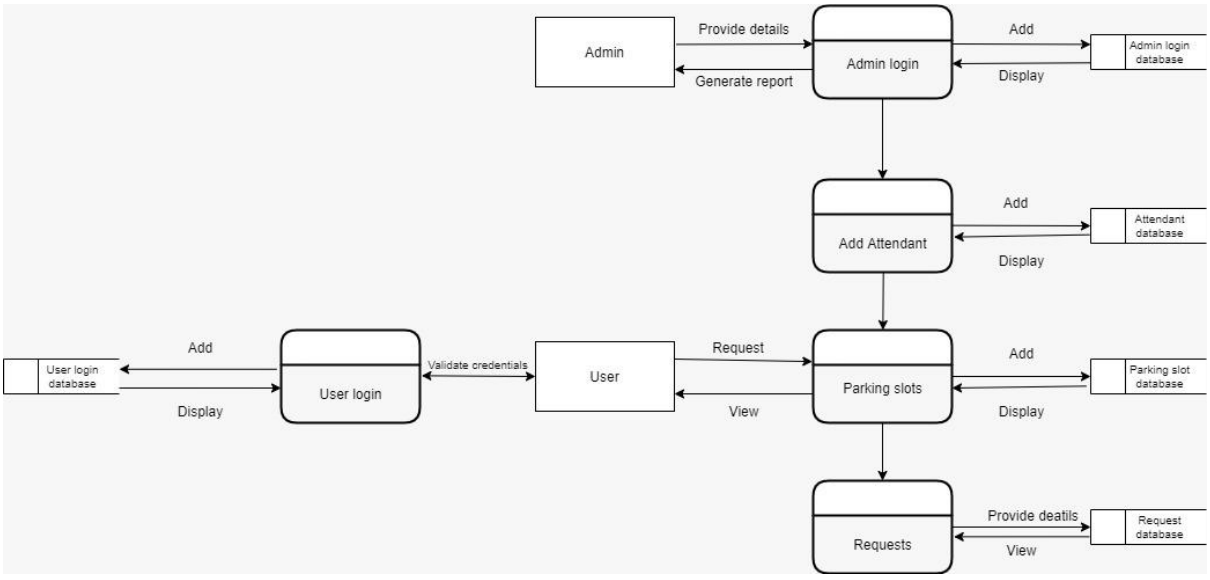

Figure-3

7.2 USECASE DIAGRAM :

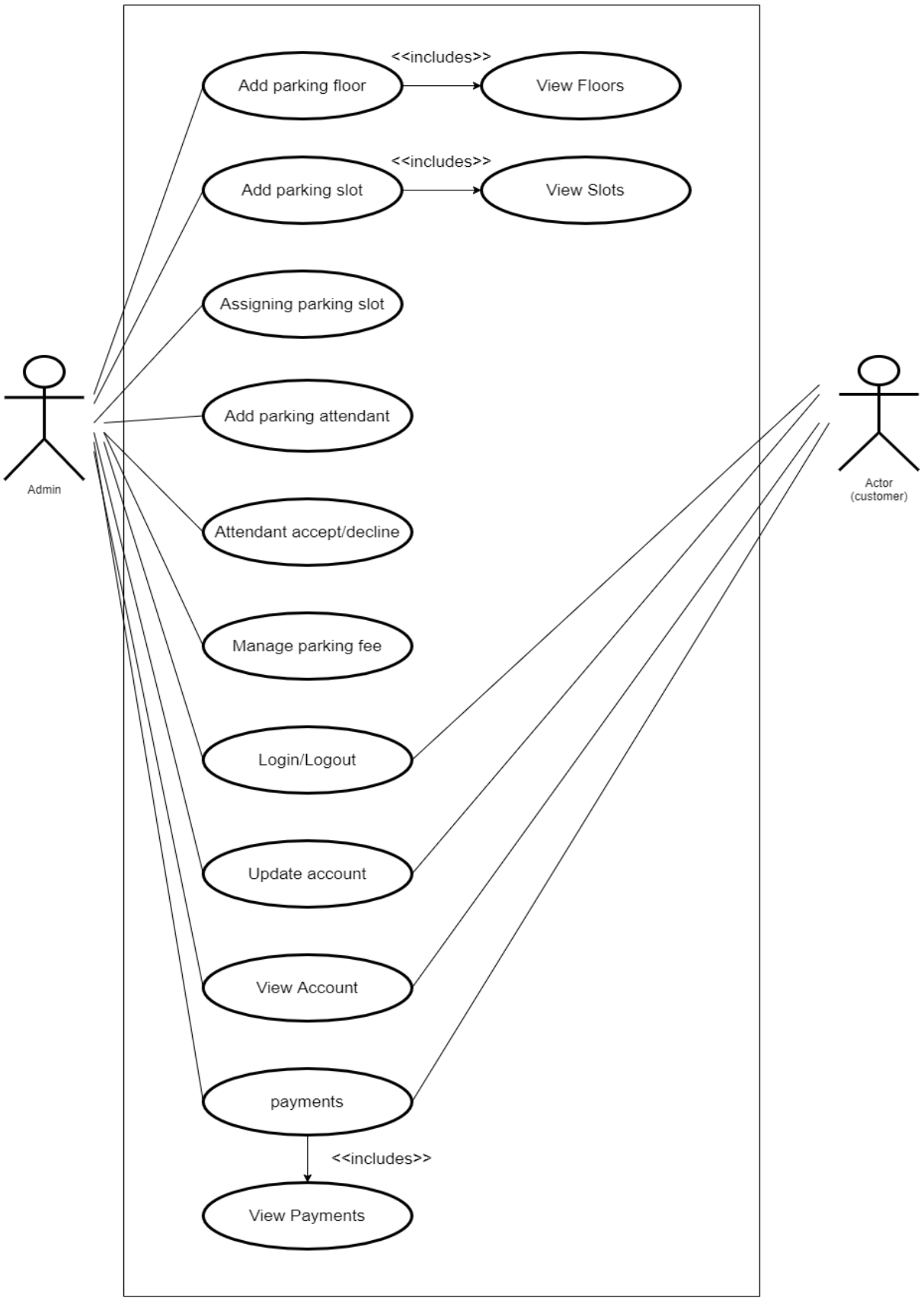

Figure-4

7.3 CLASS DIAGRAM

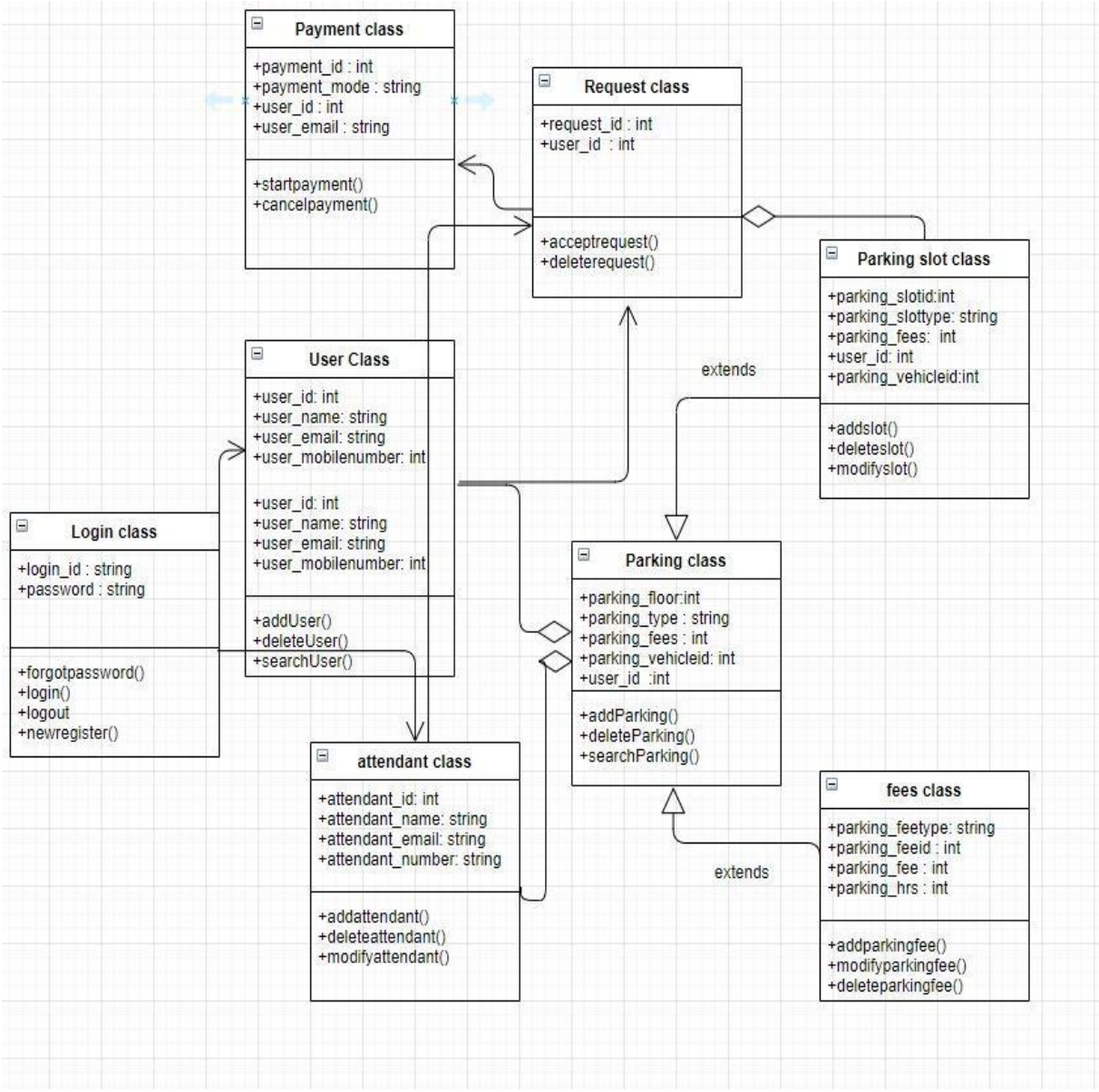

Figure-5

7.4 ACTIVITY DIAGRAM

User Portal :

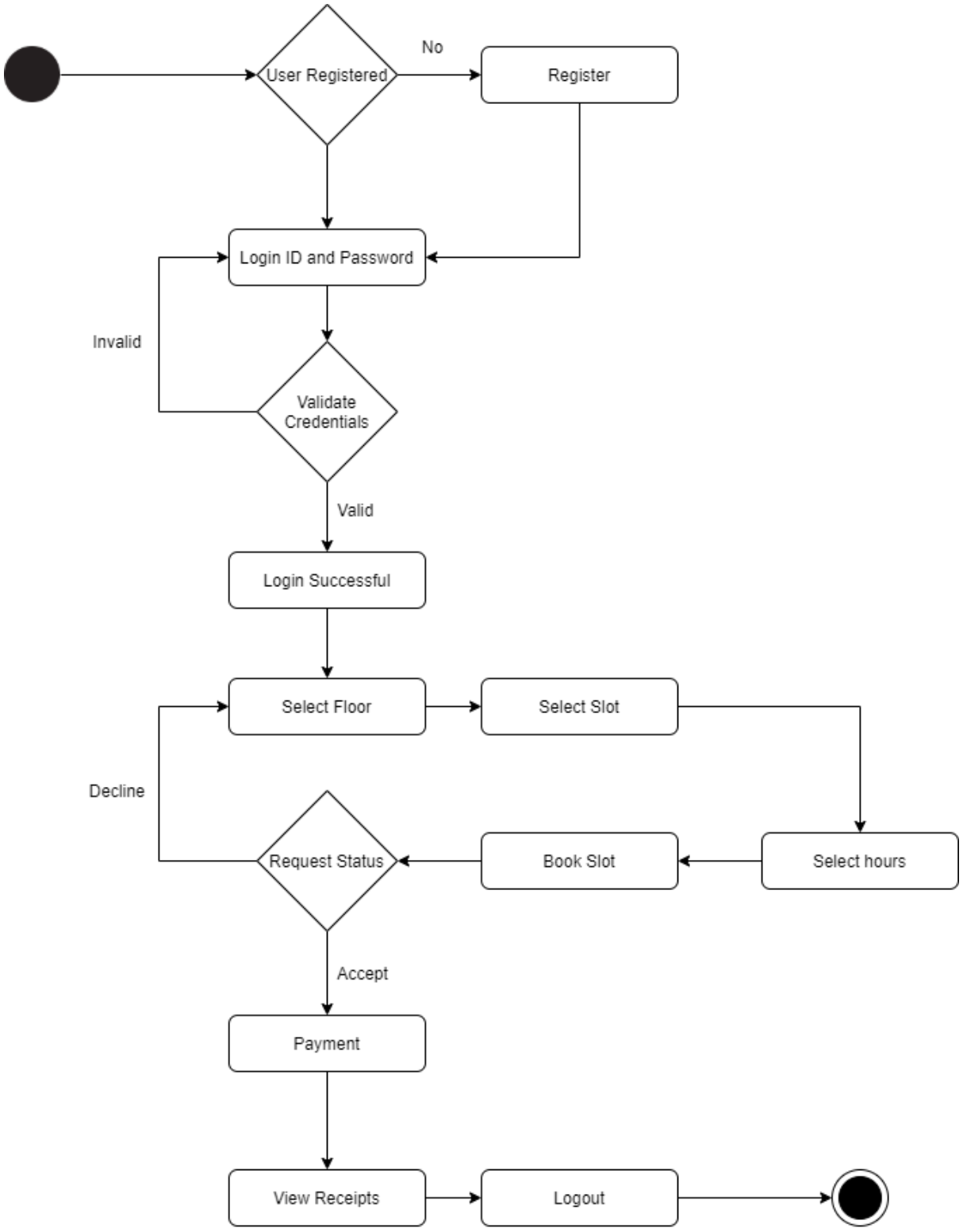

Figure-6

Admin Portal :

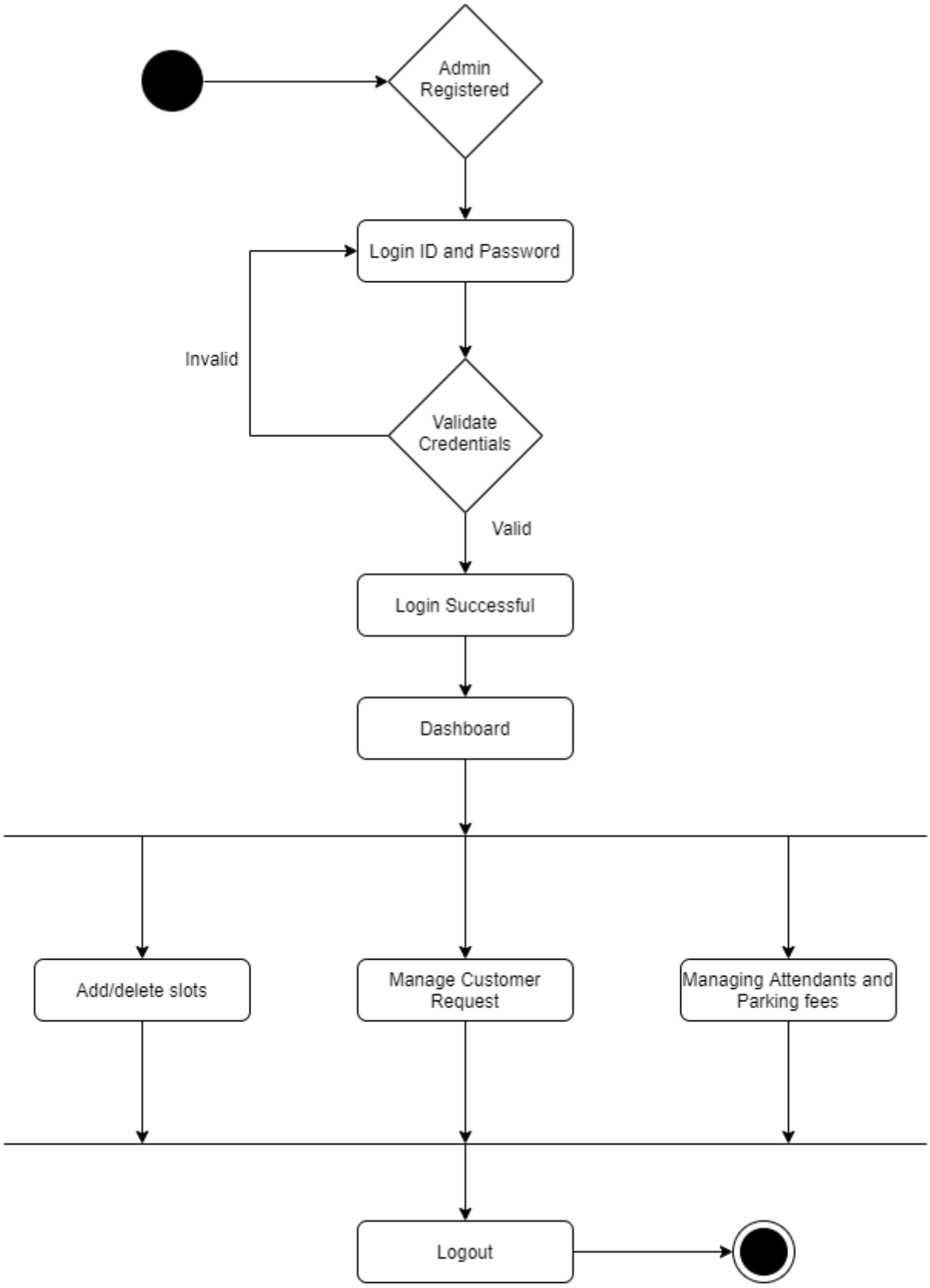

Figure-7

7.5 SEQUENTIAL DIAGRAM

User Portal :

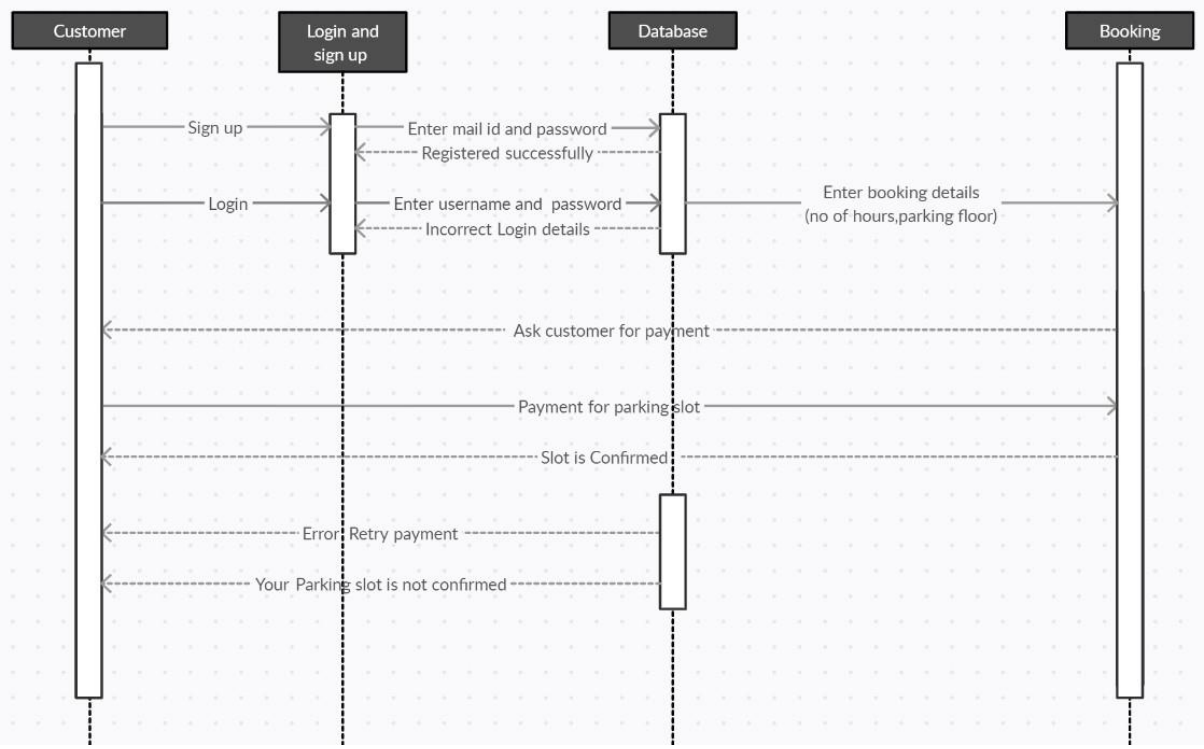

Figure-8

Admin Portal :

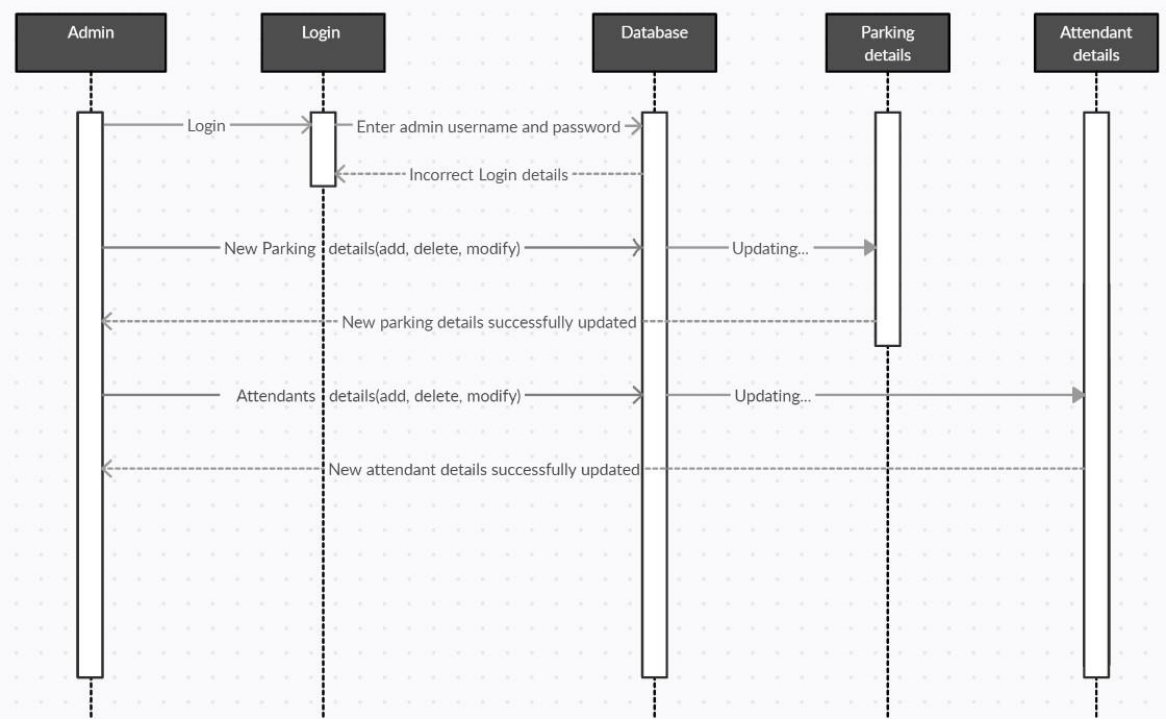

Figure-9

DATABASE STRUCTURE:

Admin login page:

Sign in page:

EMAIL	Password
king@gmail.com	king123

S.no	Parking_name	Slots	Hour	Cost	Customer	status
1	Dmart	1	6	1200	ntr@gmal.com	completed
2	Dmart	1	2	1000	king@gmail.com	Completed
3	Dmart	1	5	2500	chiru@gmail.com	Completed
4	Dmart	1	6	3000	rashi@gmail.com	Completed
5	Dmart	1	4	2000	keethu@gmail.com	completed

Attendent login page:

Sign in page

Username	Password
rashmi@gmail.com	Apduwmnb123

Category	Description
Amount charged	Rs 38000
Number if hours	19 hrs
Number of slots	1 slot
Parking location	Guntur area
Parking name	Dmart parking
Parking street	Dmart street
Request time	2021-11-13. 17:25:06
Served by	
status	Time was exceeded by 15 hours

User login page:

Email	Password
nani@gmail.com	14929usksln

8.SAMPLE CODE AND OUTPUTS

ADMIN LOGIN PORTAL :

```
<?php
session_start();
require 'update_slots.php';
require 'mysqlConnect.php';
?>
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta name="description" content="">
  <meta name="author" content="Dashboard">
  <title>Admin</title>
  <!-- Bootstrap core CSS -->
  <link href="assets/css/bootstrap.css" rel="stylesheet">
  <!--external css-->
  <link href="assets/font-awesome/css/font-awesome.css" rel="stylesheet" />
  <!-- Custom styles for this template -->
  <link href="assets/css/style.css" rel="stylesheet">
  <link href="assets/css/style-responsive.css" rel="stylesheet">
 </head>
 <body>
        <div id="login-page">
              <div class="container">
                  <form class="form-login" action="admin_login.php" method="post">
                   <h2 class="form-login-heading">sign in</h2>
                   <div class="login-wrap">
                      <input type="text" name="email" class="form-control"
placeholder="email" autofocus>
                      <br>
                      <input type="password" name="password"  class="form-control"
placeholder="Password">
        </br>
       </br>
                      <button class="btn btn-theme btn-block" href="user_login.php"
name='admin_login'  type="submit"><i class="fa fa-lock"></i> SIGN IN</button>
                     <!-- Modal -->
                     <div aria-hidden="true" aria-labelledby="myModalLabel" role="dialog"
tabindex="-1" id="myModal" class="modal fade">
                       <div class="modal-dialog">
```

```
                                <div class="modal-content">
                                    <div class="modal-header">
                                        <button type="button" class="close" data-dismiss="modal"
aria-hidden="true">&times;</button>
                                    </div>
                                    <div class="modal-footer">
                                        <button data-dismiss="modal" class="btn btn-default"
type="button">Cancel</button>
                                        <button class="btn btn-theme"
type="button">Submit</button>
                                    </div>
                                </div>
                            </div>
                        </div>
                        <!-- modal -->
                    </form>
                </div>
            </div>
    <!-- js placed at the end of the document so the pages load faster -->
    <script src="assets/js/jquery.js"></script>
    <script src="assets/js/bootstrap.min.js"></script>
    <!--BACKSTRETCH-->
    <!-- You can use an image of whatever size. This script will stretch to fit in any screen size.-
->
    <script type="text/javascript" src="assets/js/jquery.backstretch.min.js"></script>
    <script>
        $.backstretch("assets/img/admin0000.jpg", {speed: 500});
    </script>
    <?php
  if(isset($_POST['admin_login'])){
  $password=mysqli_real_escape_string($con,$_POST['password']);
```

```
 $email=mysqli_real_escape_string($con,$_POST['email']);
 $sel="select * from admin where email='$email' AND password='$password'";
 $run=mysqli_query($con,$sel);
 $check=mysqli_num_rows($run);
 if($check==0)
 {
        echo"<script>alert('password or email is not correct,try again!')</script>";
        exit();
 }
 else{
        $_SESSION['email']=$email;
        echo"<script>window.open('admin.php','_self')</script>";
 }
 }
 ?>
 </body>
</html>
```

ATTENDANT LOGIN PORTAL :

```
<?php
session_start();
require 'mysqlConnect.php';
?>
<!DOCTYPE html>
<html lang="en">
 <head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta name="description" content="">
```

```
  <meta name="author" content="Dashboard">
  <meta name="keyword" content="Dashboard, Bootstrap, Admin, Template, Theme, Responsive, Fluid, Retina">

  <title>Attendant</title>

  <!-- Bootstrap core CSS -->
  <link href="assets/css/bootstrap.css" rel="stylesheet">
  <!--external css-->
  <link href="assets/font-awesome/css/font-awesome.css" rel="stylesheet" />

  <!-- Custom styles for this template -->
  <link href="assets/css/style.css" rel="stylesheet">
  <link href="assets/css/style-responsive.css" rel="stylesheet">

 </head>

 <body>

      <div id="login-page">
          <div class="container">

              <form class="form-login" action="attendant_login.php" method="post">
                <h2 class="form-login-heading">sign in now</h2>
                <div class="login-wrap">
                  <input type="text" name="username" class="form-control" placeholder="Username" autofocus>
                  <br>
                  <input type="password" name="password"  class="form-control" placeholder="Password">
```

```
          </br>
         </br>
                        <button class="btn btn-theme btn-block" href="user_login.php"
name='attendant_login'  type="submit"><i class="fa fa-lock"></i> SIGN IN</button>
                      <!-- Modal -->
                      <div aria-hidden="true" aria-labelledby="myModalLabel" role="dialog"
tabindex="-1" id="myModal" class="modal fade">
                        <div class="modal-dialog">
                          <div class="modal-content">
                            <div class="modal-header">
                              <button type="button" class="close" data-dismiss="modal"
aria-hidden="true">&times;</button>

                            </div>
                            <div class="modal-footer">
                              <button data-dismiss="modal" class="btn btn-default"
type="button">Cancel</button>
                              <button class="btn btn-theme"
type="button">Submit</button>
                            </div>
                          </div>
                        </div>
                      </div>
                      <!-- modal -->

                    </form>

                  </div>
          </div>

    <!-- js placed at the end of the document so the pages load faster -->
```

```
    <script src="assets/js/jquery.js"></script>
    <script src="assets/js/bootstrap.min.js"></script>

    <!--BACKSTRETCH-->
    <!-- You can use an image of whatever size. This script will stretch to fit in any screen size.-
->
    <script type="text/javascript" src="assets/js/jquery.backstretch.min.js"></script>
    <script>
        $.backstretch("assets/img/atten000.jpg", {speed: 500});
    </script>

  <?php
if(isset($_POST['attendant_login'])){
$password=mysqli_real_escape_string($con,$_POST['password']);
$username=mysqli_real_escape_string($con,$_POST['username']);
$sel="select * from attendant where username='$username'";
$result=mysqli_query($con,$sel);
if(mysqli_num_rows($result) > 0)
{
  while($row = mysqli_fetch_array($result))
  {
    if(password_verify($password, $row["password"]))
    {
      //return true
      $_SESSION['username']=$username;
      echo"<script>window.open('attendant_portal.php','_self')</script>";
    }
    else
    {
      //return false
```

```
      echo"<script>alert('wrong user details,try again!')</script>";

      echo"<script>window.open('attendant_login.php','_self')</script>";
      exit();
    }
  }
}

else{
   echo"<script>alert('wrong user details,try again!')</script>";
   echo"<script>window.open('attendant_login.php','_self')</script>";
   exit();
}
}
?>

 </body>
</html>
```

USER LOGIN PORTAL :

```
<?php
session_start();
require 'mysqlConnect.php';
require 'update_slots.php';
?>
<!DOCTYPE html>
<html lang="en">
 <head>
   <meta charset="utf-8">
```

```
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- The above 3 meta tags *must* come first in the head; any other head content must come *after* these tags -->
    <title>Smart Parking Web Portal</title>
    <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
    <link href="assets/css/bootstrap.css" rel="stylesheet">
    <link href="custom.css" rel="stylesheet">


<style>
/* .modal-fullscreen size: we use Bootstrap media query breakpoints */

.modal-fullscreen .modal-dialog {
  margin: 0;
  margin-right: auto;
  margin-left: auto;
  width: 100%;
}
@media (min-width: 768px) {
  .modal-fullscreen .modal-dialog {
    width: 750px;
  }
}
@media (min-width: 992px) {
```

```
  .modal-fullscreen .modal-dialog {
    width: 970px;
  }
}
@media (min-width: 1200px) {
  .modal-fullscreen .modal-dialog {
     width: 1170px;
  }
}

/* .modal-transparent */

#regForm label{
color: #000 !important; /* makes the text-black */
}



</style>
<script type="text/javascript">
function checkPass()
{


    //Store the password field objects into variables ...
    var pass1 = document.getElementById('password');
    var pass2 = document.getElementById('password_confirm');
    //Store the Confimation Message Object ...
    var message = document.getElementById('confirmMessage');
    //Set the colors we will be using ...
```

```
    var goodColor = "#66cc66";
    var badColor = "#ff6666";
    //Compare the values in the password field
    //and the confirmation field
    if(pass1.value == pass2.value){
        //The passwords match.
        //Set the color to the good color and inform
        //the user that they have entered the correct password
        pass2.style.backgroundColor = goodColor;

        $('#regBtn').prop('disabled', false);
        $('#regOwner').prop('disabled', false);
    }else{
        //The passwords do not match.
        //Set the color to the bad color and
        //notify the user.
        $('#regBtn').prop('disabled', true);
        $('#regOwner').prop('disabled', true);
        pass2.style.backgroundColor = badColor;
        message.style.color = badColor;
        message.innerHTML = "Passwords Do Not Match!"
    }
}

// validate email
function email_validate(email)
{
var regMail = /^([_a-zA-Z0-9-]+)(\.[_a-zA-Z0-9-]+)*@([a-zA-Z0-9-]+\.)+([a-zA-Z]{2,3})$/;
    var status = document.getElementById("emailstatus");
```

```
   if(regMail.test(email) == false)
   {
   document.getElementById("emailstatus").innerHTML    = "<span class='warning'>Email address is not valid.</span>";
       status.style.color = "#f44336";
       $('#regBtn').prop('disabled', true);
       $('#regOwner').prop('disabled', true);
   }
   else
   {
   document.getElementById("emailstatus").innerHTML  = "<span class='valid'>Email address is Valid!</span>";
       status.style.color = "#00838f ";
       $('#regBtn').prop('disabled', false);
       $('#regOwner').prop('disabled', false);
   }
}

$(".modal-fullscreen").on('show.bs.modal', function () {
  setTimeout( function() {
    $(".modal-backdrop").addClass("modal-backdrop-fullscreen");
  }, 0);
});
$(".modal-fullscreen").on('hidden.bs.modal', function () {
  $(".modal-backdrop").addClass("modal-backdrop-fullscreen");
});

$(".modal-transparent").on('show.bs.modal', function () {
  setTimeout( function() {
    $(".modal-backdrop").addClass("modal-backdrop-transparent");
```

```
  }, 0);
});
$(".modal-transparent").on('hidden.bs.modal', function () {
  $(".modal-backdrop").addClass("modal-backdrop-transparent");
});
</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">
       <form  id="regForm" action="register.php" method="POST" enctype="multipart/form-
data">

 <div class="form-group">
    <label for="name" >Name:</label>
    <input type="text" name="name" id="name"  class="text ui-widget-content ui-corner-
all">
 </div>

 <div class="form-group">
    <label for="email">Email</label>
    <input type="email" id="email" name="email" placeholder="" class="text ui-widget-
content ui-corner-all"  onchange="email_validate(this.value);" required>
        <p id="emailstatus"></p>
 </div>

  <div class="form-group">
    <label for="password">Password</label>
   <input type="password" id="password" name="password" placeholder="" class="text ui-
widget-content ui-corner-all" required>
 </div>

  <div class="form-group">
    <label for="password">Confirm Password</label>
    <input type="password" class="text ui-widget-content ui-corner-all"
id="password_confirm" name="password_confirm" placeholder=""  onkeyup="checkPass();
return false;" required>
 </div>
```

```
<button type="submit" id="regBtn" class="form-control" name="register">create
account</button>
      </div>
      </form>
    </div>
  </div>
 </div>
  <div class="row">
    <div class="container">
      <div class="col-md-4"></div>
      <div class="col-md-4">
        <form enctype="multipart/form" action="user_login.php" method="POST" id="">
            <div class="page-header">
              <center><h3 class="colors">Login</h3></center>
            </div>
          <label for=""></label>
          <input type="text" name="email" id="" placeholder="email" class="email">

          <label for=""></label>
          <input type="password" name="password" id="" placeholder="password"
class="pass">

          <button type="submit" name="login">login</button>
           <label for=""></label>
           <button type="button" data-toggle="modal" data-target="#myModal">Sign
Up</button>

        </form>
```

```
        </div>
        <div class="col-md-4"> </div>
    </div>
  </div>
</div>

<?php
if(isset($_POST['login'])){
$password=mysqli_real_escape_string($con,$_POST['password']);
$email=mysqli_real_escape_string($con,$_POST['email']);

$sel="select * from users where email='$email'";
$result=mysqli_query($con,$sel);
if(mysqli_num_rows($result) > 0)
{

  while($row = mysqli_fetch_array($result))
  {
    if(password_verify($password, $row["password"]))
    {
      //return true
      $_SESSION['driver_email']=$email;
      echo"<script>window.open('home.php','_self')</script>";
    }
    else
    {
      //return false
       echo"<script>alert('wrong user details,try again!')</script>";
```

```
      echo"<script>window.open('user_login.php','_self')</script>";
      exit();
    }
  }
}
else{

   echo"<script>alert('wrong user details,try again!')</script>";
   echo"<script>window.open('user_login.php','_self')</script>";
   exit();
}
}
?>
   <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
   <script src="assets/js/jquery-1.8.3.min.js"></script>
   <!-- Include all compiled plugins (below), or include individual files as needed -->
   <script src="assets/js/bootstrap.min.js"></script>
  </body>
</html>
```

OUTPUTS :

Admin Login Page :

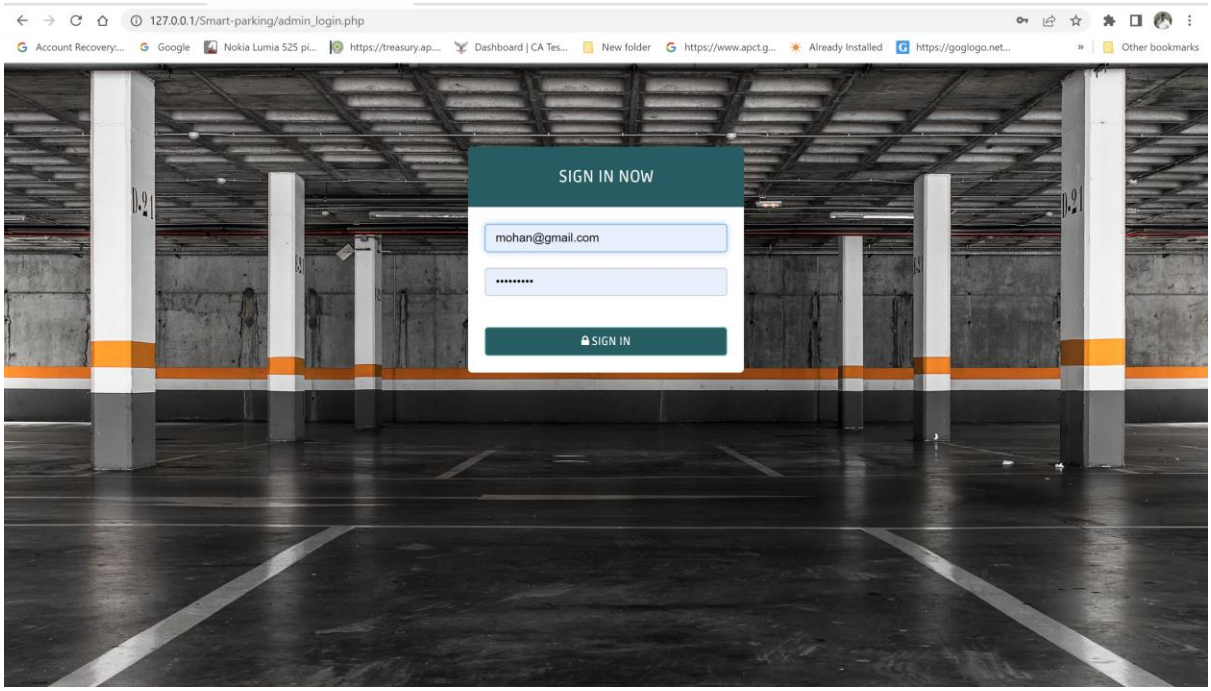

Figure-10

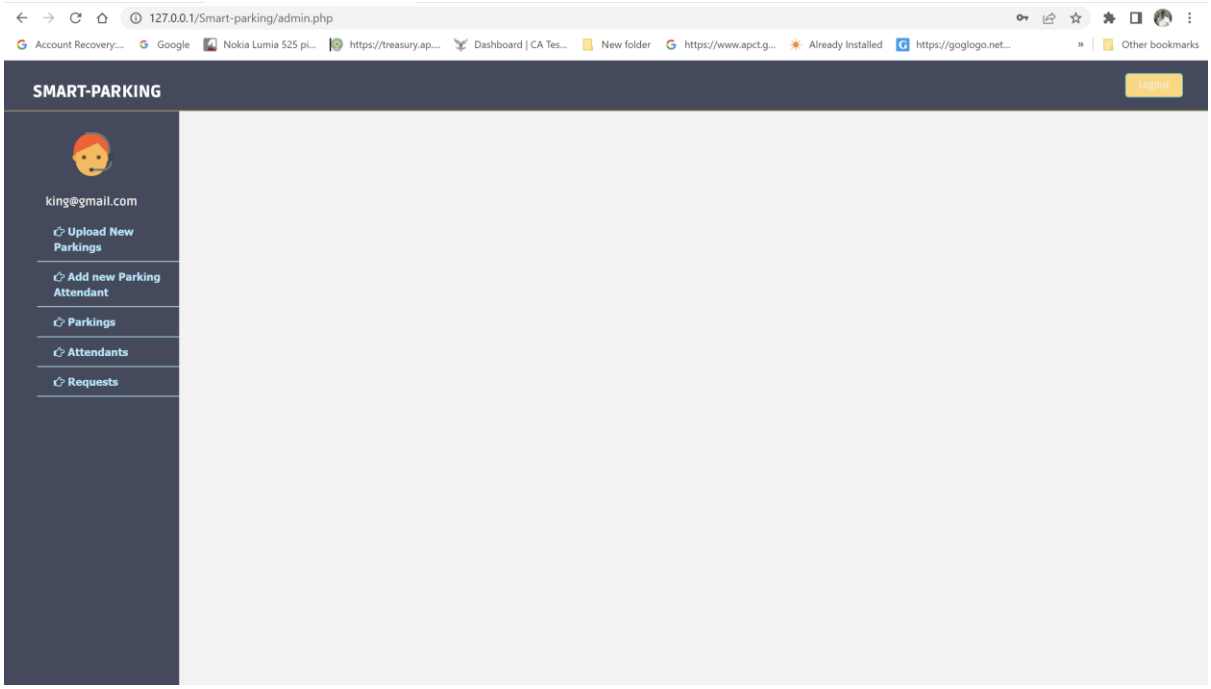

Figure-11

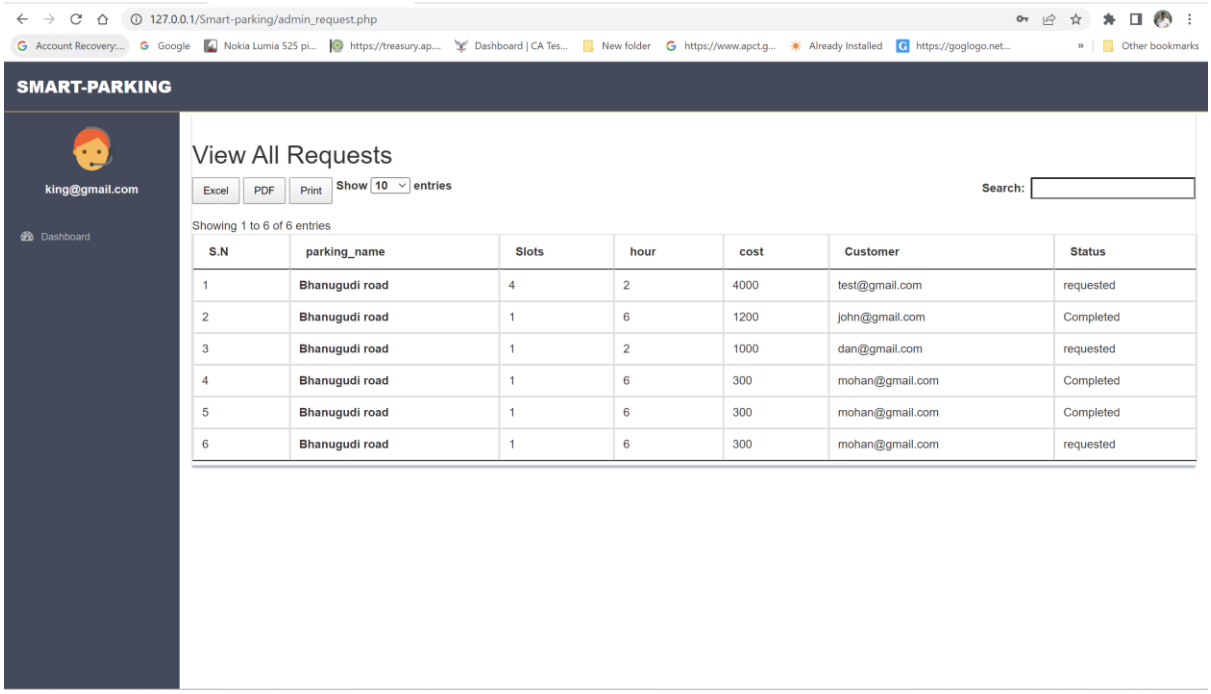

Figure-12

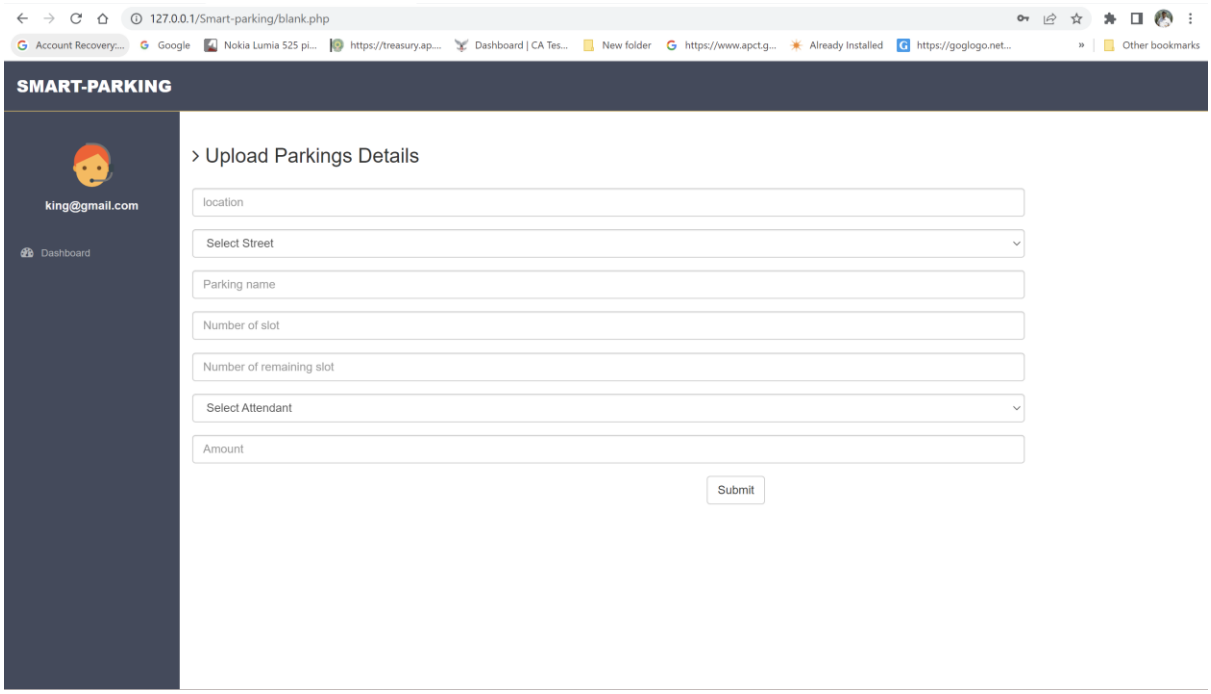

Figure-13

Attendant Login Page :

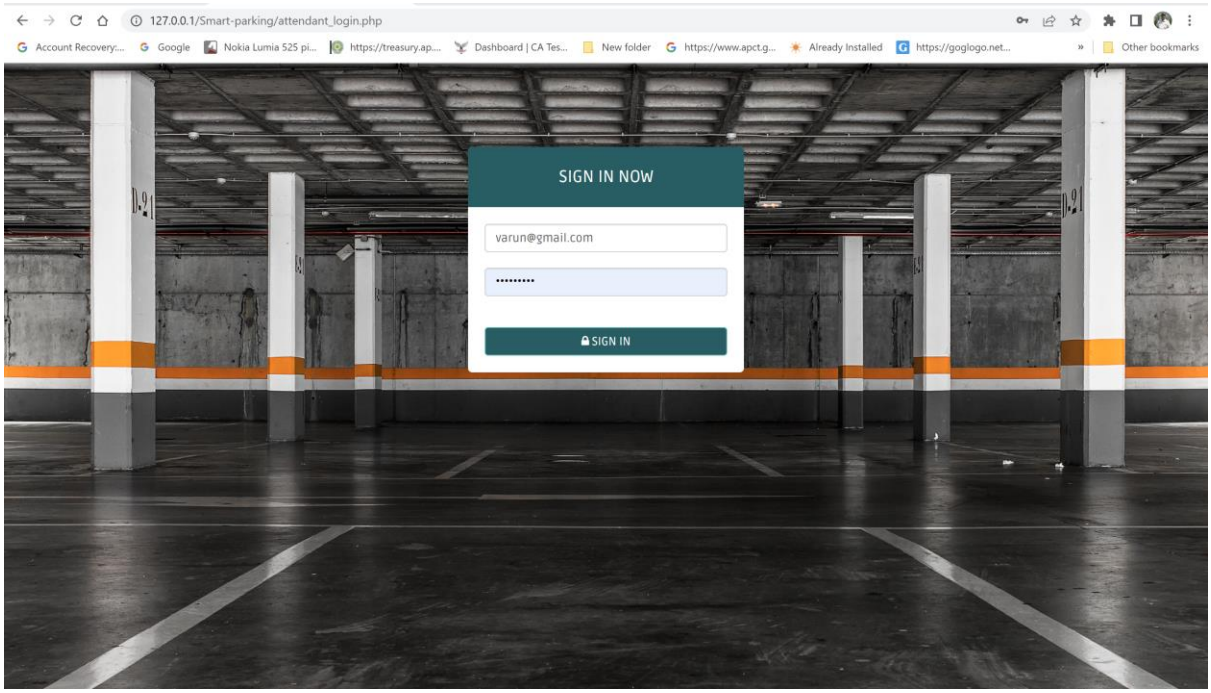

Figure-14

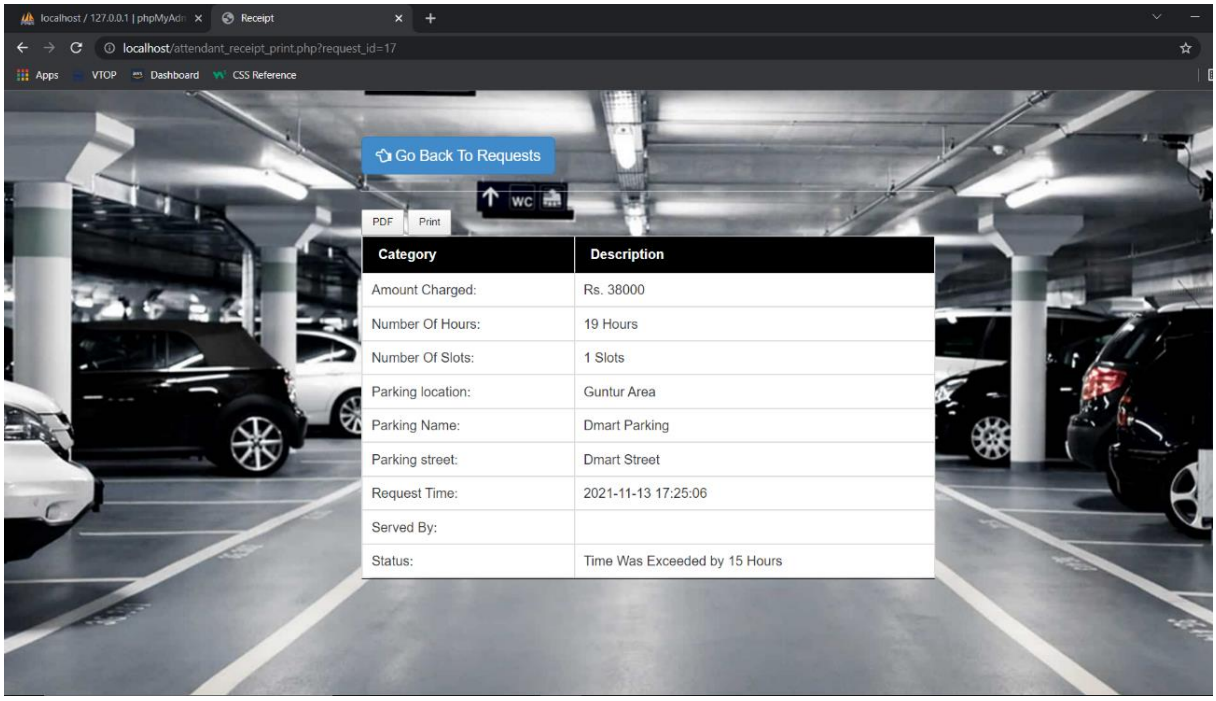

Figure-15

User Login Page :

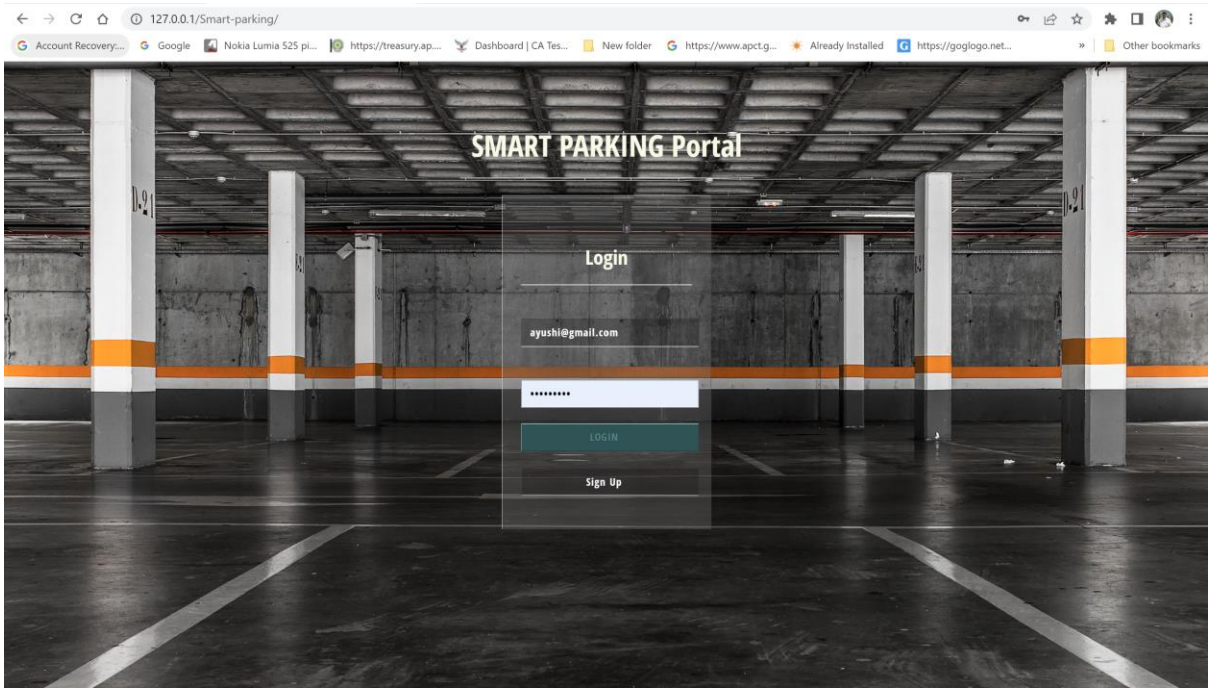

Figure-16

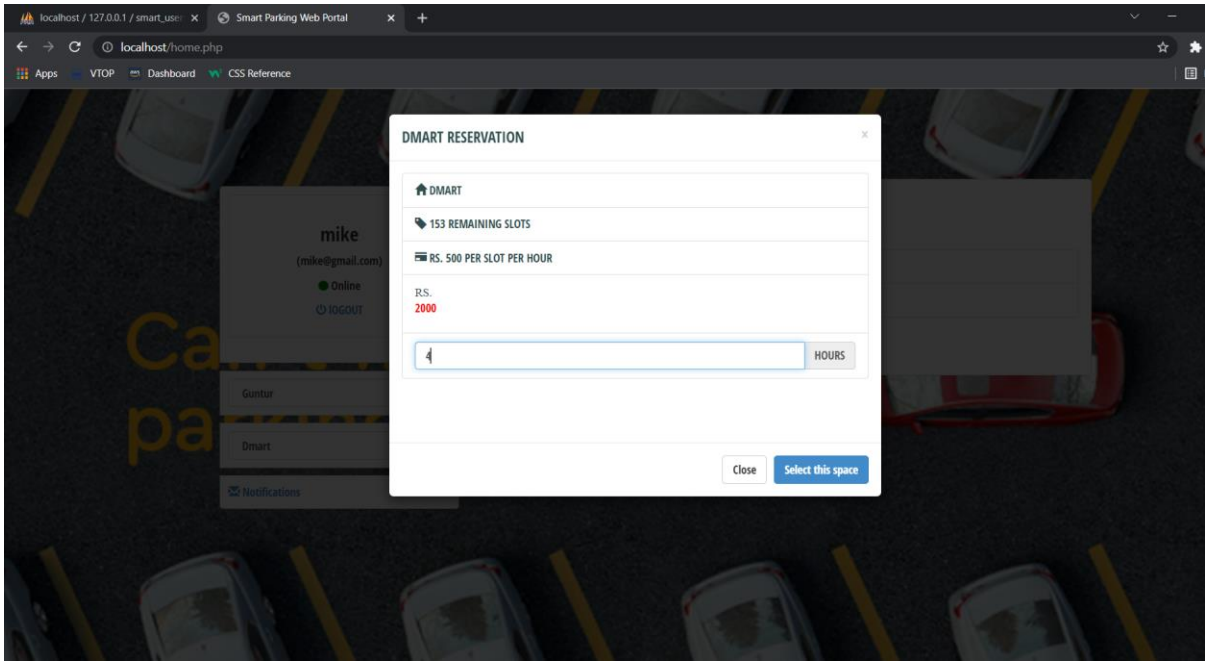

Figure-17

9.TESTING

USER :

```
<?php
if(isset($_POST['admin_login'])){
$password=mysqli_real_escape_string($con,$_POST['password']);
 $email=mysqli_real_escape_string($con,$_POST['email']);

 $sel="select * from admin where email='$email' AND password='$password'";
 $run=mysqli_query($con,$sel);
      $check=mysqli_num_
      rows($run);
      if($check==0)
{
echo"<script>alert('password or email is not correct,try again!')</script>";
exit();
 }
else{
$_SESSION['email']=$email;
echo"<script>window.open('admin.php','_self')</script>";
}
}
?>
```

Cyclomatic Complexity is E=6 N=6

E-N+2=6-6+2=2

Test Cases

Equivalence Partitioning

Input	Valid equivalence class	Invalid equivalence class
Username(string) a,b,c A,B,C 1,2,3 Special characters	[a,b,c] [A,B,C] [1,4,2]	No Invalid Equivalence class
Password(String) A,B,C 1,2,3 Special characters	[a,b,c] [A,B,C] [1,4,2]	No Invalid equivalence cases

Table-4

Upload slot :

```
<?php
if(isset($_POST[‘sub’])){
$location=mysqli_real_escape_string($con,$_POST[‘location’]);
$street = mysqli_real_escape_string($con,$_POST[‘street’]);
$name=mysqli_real_escape_string($con,$_POST[‘name’]);
$slot=mysqli_real_escape_string($con,$_POST[‘slot’]);
$remaining_slots=mysqli_real_escape_string($con,$_POST[‘remaining_slots’]);
$price=mysqli_real_escape_string($con,$_POST[‘price’]);
$attendant=mysqli_real_escape_string($con,$_POST[‘attendant’]);
if($location==’’&& $street==’’&& $name==’’ && $slot==’’ && $price==’’ &&
$remaining_slots==’’)
{
        echo”<script>alert(‘please fill all field’)</script>”;
echo”<script>window.open(‘blank.php’,’_self’)</script>”; exit();
}
```

```
else{

        $insert="INSERT INTO `parkings` (`id`, `location`, `street`, `name`, `slot` ,
`price`,`remaining_slots`,'attendant') VALUES (NULL, '$location', '$street', '$name', '$slot' ,
'$price','$remaining_slots','$attendant');";
$run_insert=mysqli_query($con,$insert);

if($run_insert){
        echo"<script>alert('Successful added!')</script>";
  echo"<script>window.open('blank.php','_self')</script>";
}
else{
echo"<script>alert('Error please try again')</script>";
echo"<script>window.open('blank.php','_self')</script>";
}
}}
?>
```

Cyclomatic Complexity E=10

N=10

E-N+2=10-10+2=2

Input	Boundary Cases
Integer N (no of slots) 1<=N<=100	0,1,2 99,100,101

Table-5

User Sign Up Successful :

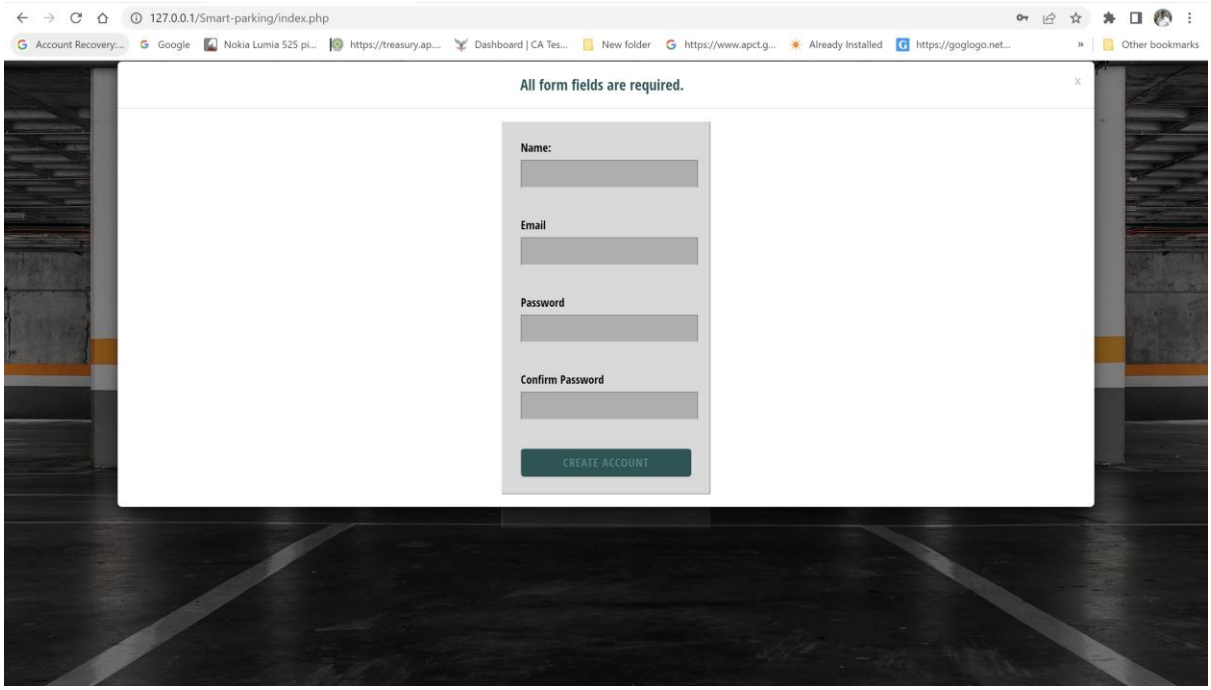

Figure-18

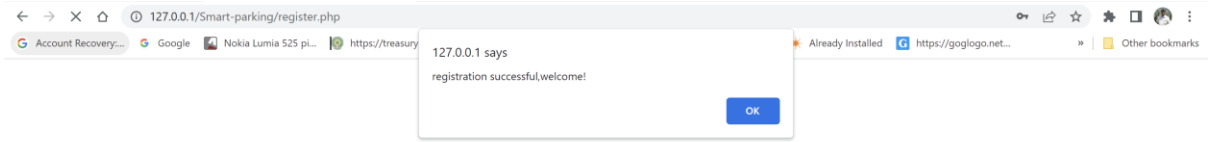

Figure-19

Attendant Adding Successful :

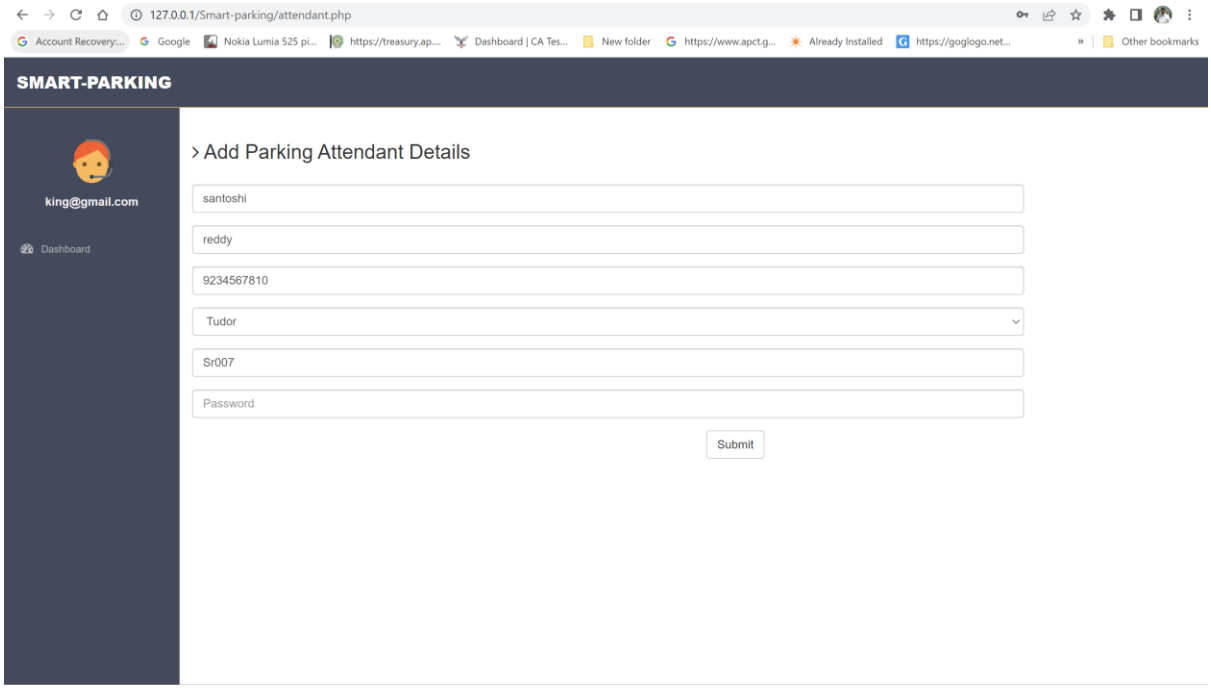

Figure-20

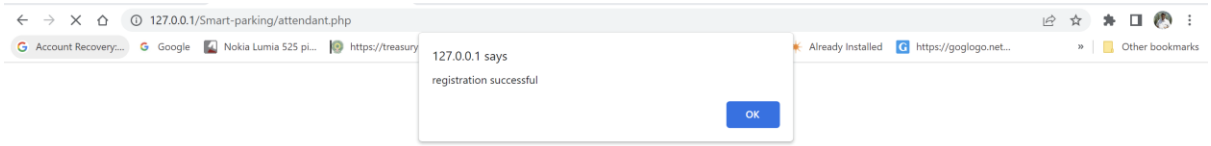

Figure-21

10.RESULTS DISCUSSION

Now a days, we can see that there are a greater number of vehicles and the space to park the vehicles are less in cities. So, there are many parking areas coming in most of cities like parking areas in a complex and this project "SMART PARKING MANAGEMENT SYSTEM" can be used there where it will save lot of time to customers like they can book their parking slot by selecting number of hours and then later they can come to parking area and park their vehicles which saves lot of time as the customers need not search on roads for parking place. Not only for customers this management system is very useful to the parking area admins where this management system will automatically allot parking places based on hours required for a customer and this will generate automatic receipt to customer and also it reduces a lot of paper work where there is no need of any attender to write all the details of vehicles coming in and going out. In this way this project is very useful for parking area complex if they use this in efficient way.

11.FUTURE SCOPE

This project is focussed at managing the parking slots in a complex optimally for both the customers and the employees working there. The objective of the project helps the customers to find a parking slot very quickly and makes it easy for them to park in a large building. This software reduces a lot of paperwork and enables the customer to do everything mostly online in these tough times of the pandemic.

12.REFERENCE

- Anusha, Arshitha M, S, Anushri, Geetanjali Bishtannavar "Review Paper on Smart Parking System," International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, Volume 7, Issue 08, Special Issue – 2019.
- M. Y. I. Idris, Y. Y. Leon, E. M. Tamil, N. M. Noor, and Z. Razak, "Car parking system: A review of the smart parking system and its technology," Information Technology Journal, pp. 101-113.], 2009.
- Amir O. Kotb, Yao-Chunsheng, and Yi Huang "Smart parking Guidance, Monitoring and Reservation: A Review," IEEE-ITSM, pp.6-16. Apr-2017.
- http://www.apache.org/docs/2.0/misc/tutorials.html
- www.w3schools.com