SMART PARKING SYSTEM USING IOT

OBJECTIVES:

- Efficient Parking Management: Ensure efficient utilization of parking spaces and reduce the time spent searching for a parking spot.
- Real-time Monitoring: Enable real-time monitoring of parking space occupancy to improve user experience.
- Environment Impact: Reduce carbon emissions by minimizing the time spent searching for parking, which can contribute to traffic congestion.
- Revenue Generation: Integrate payment options and reservation systems to generate revenue for parking operators.
- User Convenience: Enhance the user experience with mobile apps and notifications.

IOT DEVICES SETUP:

Parking Sensors: Deploy IoT sensors (e.g., ultrasonic, magnetic, or infrared)in each parking space to detect vehicle presence. These sensors are connected to a microcontroller (e.g., Arduino, Raspberry Pi) to collect and transmit data.

IOT Gateways: Install IoT gateways (e.g., Lora WAN, Wi-Fi, or cellular) that collect data from the parking sensors and transmit it to the central server or cloud platform.

Central Server or Cloud Platform: Set up a central server or cloud platform that receives data from the gateways, processes it, and provides real-time information to users.

PLATFORM DEVELOPMENT:

Cloud Platform

Platform Development:

- User Interface (Mobile/Web App): Create user-friendly mobile or web applications for both parking operators and customers.

- Users can check space availability, make reservations, and pay for parking.

Mobile App:

Data Processing: Develop algorithms to process parking space occupancy data, detect available spaces, and calculate occupancy rates. This information is updated in real-time on the platform.

Reservation System: Implement a reservation system that allows users to reserve parking spots in advance, specifying their preferred time and location.

Payment Integration: Integrate payment gateways to enable users to pay for parking via the app, using methods such as credit cards, digital wallets, or RFID cards.

Notifications: Implement notification systems to inform users about available parking spaces, reservation confirmations, and payment receipts.

DIAGRAMS:

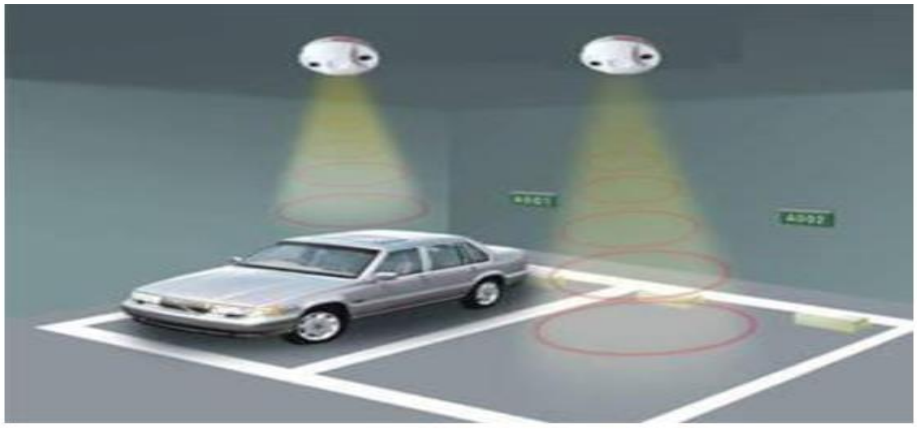


Figure 4. Ultrasonic Sensor Detection Area

PARKING SENSOR CODE FOR IOT DEVICES:

```
# Import required libraries
import RPi.GPIO as GPIO
import time
import requests
# Set GPIO mode and pin
```

```
    GPIO.setmode(GPIO.BCM)
    sensor_pin = 17
    # Configure the sensor pin as input
    GPIO.setup(sensor_pin, GPIO.IN)
    # Set the API endpoint for the central server
    server_url = https://your-central-server.com/api/parking
while True:
   try:
      # Read sensor data (1 for occupied, 0 for vacant)
      sensor_data = GPIO.input(sensor_pin)
      # Send sensor data to the central server
      payload = {"sensor_data": sensor_data}
      response = requests.post(server_url, json=payload)
      if response.status_code == 200:
         print("Data sent successfully")
      else:
         print("Failed to send data")
         time.sleep(5)  # Send data every 5 seconds
         except KeyboardInterrupt:
         GPIO.cleanup()
         break
```

In this code, we use the RPi.GPIO library to interface with the GPIO pins of a Raspberry Pi (assuming you're using a Raspberry Pi for your IoT device). The code reads the sensor data (1 for occupied and 0 for vacant) and sends it to the central server via an HTTP POST request.

Central Server Code (Python):

The central server code will depend on your chosen technology stack, such as Flask or Django for web services and a database for data storage. Here's a simplified example using Flask:

```
from flask import Flask, request, jsonify
from flask_cors import CORS
import sqlite3
app = Flask(_name_)
CORS(app)  # Enable Cross-Origin Resource Sharing
# Set up a SQLite database for storing parking data
conn = sqlite3.connect('parking_data.db')
cursor = conn.cursor()
cursor.execute('''CREATE TABLE IF NOT EXISTS parking (id INTEGER PRIMARY KEY, sensor_data INT)''')
conn.commit()
# API endpoint to receive sensor data
@app.route('/api/parking', methods=['POST'])
def receive_parking_data():
data = request.get_json()
sensor_data = data.get('sensor_data')
if sensor_data is not None:
# Store data in the database
cursor.execute("INSERT INTO parking (sensor_data) VALUES (?)", (sensor_data,))
conn.commit()
return "Data received", 200
else:
return "Invalid data", 400
```

If _name_ == '_main_':

app.run(debug=True)

This is a simple Flask-based API that receives sensor data and stores it in an SQLite database. In a real-world scenario, you would implement more features like data processing, user management, and security measures.

Arduino Code for Parking System

Basically, there will be two systems, one controlling the parking system and other the authentication system at the gate. Here we will discuss the prior part.The IR sensors will be connected in different pins and will give the status of the area (1 for occupied and 0 for empty). This data will be uploaded to 'ThingSpeak' server. There you will see the proper visualization of the inputs given by the sensor. This will be uploaded byusing the ESP8266 wifi module. Below is the code.

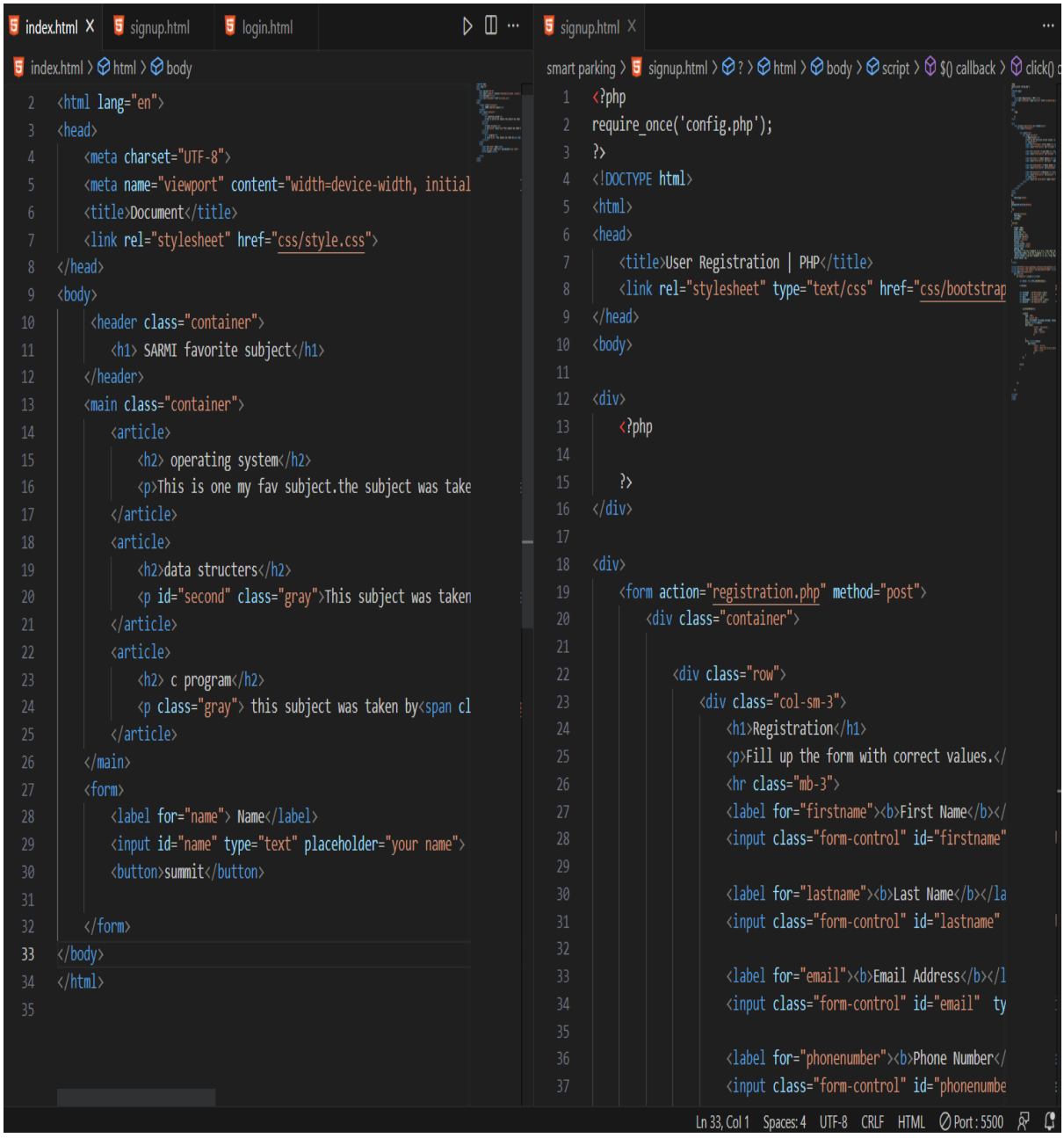
```
#include <SoftwareSerial.h>    //Software Serial library
const int ProxSensor1=5;
const int ProxSensor2=6;
const int ProxSensor3=7;

int inputVal = 0;
SoftwareSerial mySerial(12, 13);
SoftwareSerial espSerial(2, 3);   //Pin 2 and 3 act as
RX and TX. Connect them to TX and RX of ESP8266
#define DEBUG true
String mySSID = "ram";       // WiFi SSID
String myPWD = "ramnath123$"; // WiFi Password
String myAPI = "I3SMDXSR4EMUK7CR";   // API Key
String myHOST = "api.thingspeak.com";
String myPORT = "80";
String myFIELD1 = "field1";
String myFIELD2 = "field2";
String myFIELD3 = "field3";
int sendVal1; int sendVal2; int sendVal3;
int k=0;
```

```
void setup()
{
  Serial.begin(9600);
  mySerial.begin(115200);
  espSerial.begin(115200);
  pinMode(ProxSensor1,INPUT);    //Pin 2 is connected to
the output of proximity sensor
  pinMode(ProxSensor2,INPUT);
  pinMode(ProxSensor3,INPUT);

  espData("AT+RST", 1000,
DEBUG);                      //Reset the ESP8266 module
  espData("AT+CWMODE=1", 1000,
DEBUG);                 //Set the ESP mode as station
mode
  espData("AT+CWJAP=\""+ mySSID +"\",\""+ myPWD +"\"",
1000, DEBUG);   //Connect to WiFi network
  /*while(!esp.find("OK"))
  {
      //Wait for connection
  }*/
  delay(1000);

}

  void loop()
  {


    sendVal1=digitalRead(ProxSensor1);
    sendVal2=digitalRead(ProxSensor2);
    sendVal3=digitalRead(ProxSensor3);
    String sendData1 = "GET /update?api_key="+ myAPI
+"&"+ myFIELD1 +"="+String(sendVal1);
    String sendData2 = "GET /update?api_key="+ myAPI
+"&"+ myFIELD2 +"="+String(sendVal2);
    String sendData3 = "GET /update?api_key="+ myAPI
+"&"+ myFIELD3 +"="+String(sendVal3);
    espData("AT+CIPMUX=1", 1000, DEBUG);       //Allow
multiple connections
```

```
    espData("AT+CIPSTART=0,\"TCP\",\""+ myHOST +"\","+
myPORT, 1000, DEBUG);
    espData("AT+CIPSEND=0,"
+String(sendData1.length()+4),1000,DEBUG);
    espSerial.find(">");
    espSerial.println(sendData1);
    espData("AT+CIPMUX=1", 1000, DEBUG);       //Allow
multiple connections
    espData("AT+CIPSTART=0,\"TCP\",\""+ myHOST +"\","+
myPORT, 1000, DEBUG);
    espData("AT+CIPSEND=0,"
+String(sendData2.length()+4),1000,DEBUG);
    espSerial.find(">");
    espSerial.println(sendData2);
    espData("AT+CIPMUX=1", 1000, DEBUG);       //Allow
multiple connections
    espData("AT+CIPSTART=0,\"TCP\",\""+ myHOST +"\","+
myPORT, 1000, DEBUG);
    espData("AT+CIPSEND=0,"
+String(sendData3.length()+4),1000,DEBUG);
    espSerial.find(">");
    espSerial.println(sendData3);
    Serial.print("Value to be sent: ");
    Serial.println(sendVal1);
    Serial.print("Value to be sent: ");
    Serial.println(sendVal2);
    Serial.print("Value to be sent: ");
    Serial.println(sendVal3);
    if(k==0&&sendVal1==1&&sendVal2==1&&sendVal3==1)
    {
      SendMessage();
      k=1;
    }

 if (mySerial.available()>0)
   Serial.write(mySerial.read());
    espData("AT+CIPCLOSE=0",1000,DEBUG);
    //delay(100);
  }
```

```
  String espData(String command, const int timeout,
boolean debug)
{
  Serial.print("AT Command ==> ");
  Serial.print(command);
  Serial.println("     ");

  String response = "";
  espSerial.println(command);
  long int time = millis();
  while ( (time + timeout) > millis())
  {
    while (espSerial.available())
    {
      char c = espSerial.read();
      response += c;
    }
  }
  if (debug)
  {
    //Serial.print(response);
  }
  return response;
}
 }
```

SCREENSHORT FOR REGISTRATION AND LOGIN IMAGES



index.html X | signup.html | login.html

index.html > html > body

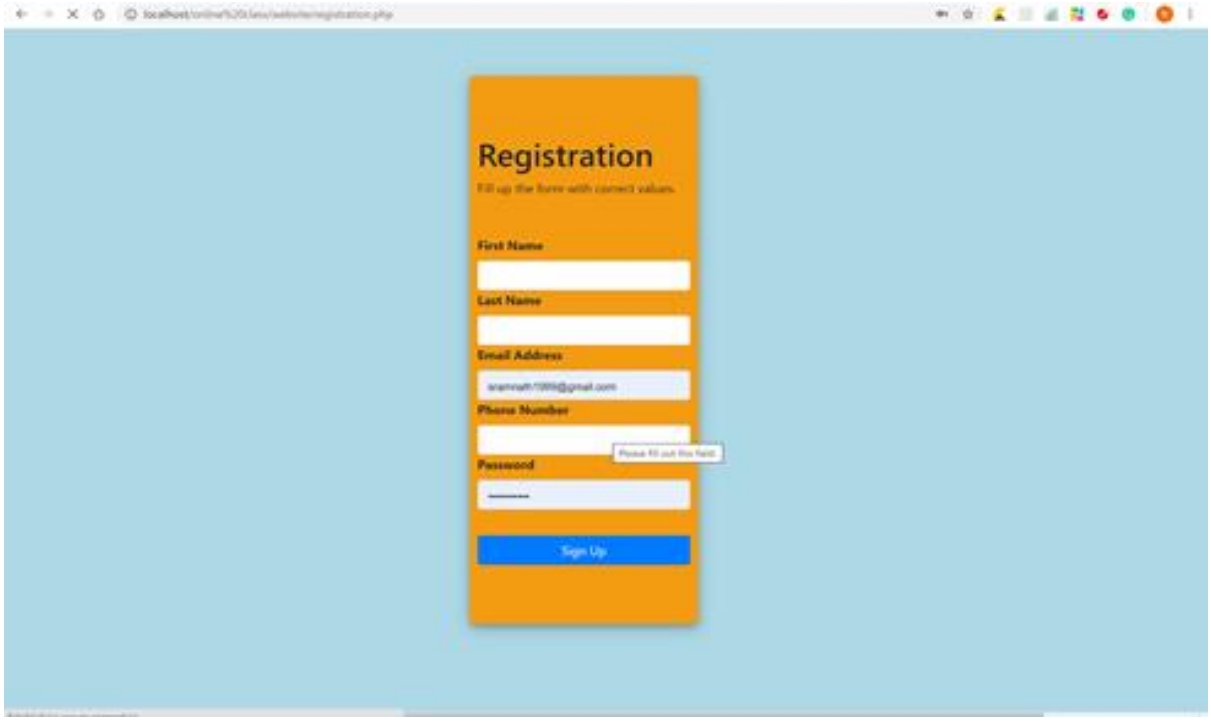
```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial
    <title>Document</title>
    <link rel="stylesheet" href="css/style.css">
</head>
<body>
    <header class="container">
        <h1> SARMI favorite subject</h1>
    </header>
    <main class="container">
        <article>
            <h2> operating system</h2>
            <p>This is one my fav subject.the subject was take
        </article>
        <article>
            <h2>data structers</h2>
            <p id="second" class="gray">This subject was taken
        </article>
        <article>
            <h2> c program</h2>
            <p class="gray"> this subject was taken by<span cl
        </article>
    </main>
    <form>
        <label for="name"> Name</label>
        <input id="name" type="text" placeholder="your name">
        <button>summit</button>

    </form>
</body>
</html>

```

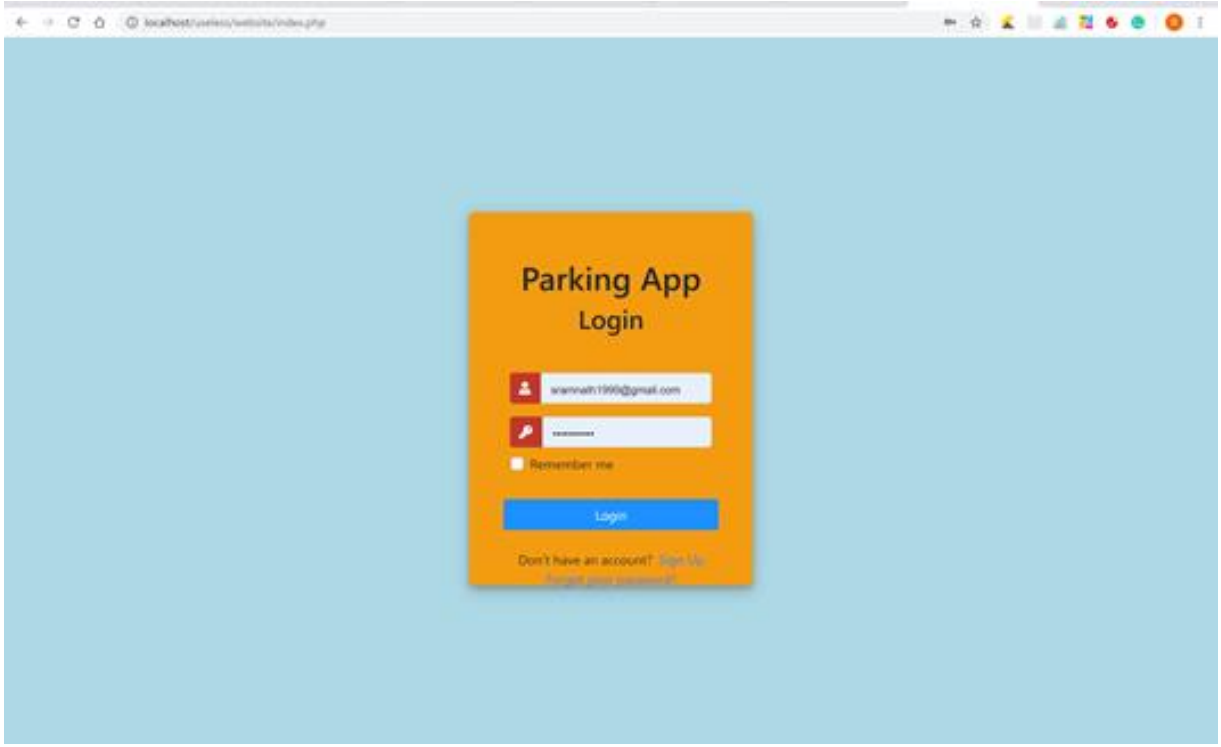
signup.html X

smart parking > signup.html > ? > html > body > script > $() callback > click() c

```
<?php
require_once('config.php');
?>
<!DOCTYPE html>
<html>
<head>
    <title>User Registration | PHP</title>
    <link rel="stylesheet" type="text/css" href="css/bootstrap
</head>
<body>

<div>
    <?php

    ?>
</div>

<div>
    <form action="registration.php" method="post">
        <div class="container">

            <div class="row">
                <div class="col-sm-3">
                    <h1>Registration</h1>
                    <p>Fill up the form with correct values.</
                    <hr class="mb-3">
                    <label for="firstname"><b>First Name</b></
                    <input class="form-control" id="firstname"

                    <label for="lastname"><b>Last Name</b></la
                    <input class="form-control" id="lastname"

                    <label for="email"><b>Email Address</b></l
                    <input class="form-control" id="email"  ty

                    <label for="phonenumber"><b>Phone Number</
                    <input class="form-control" id="phonenumbe
```

Ln 33, Col 1 Spaces: 4 UTF-8 CRLF HTML Port : 5500


Registration
First Name
Last Name
Email Address
Phone Number
Password
Sign Up


Parking App
Login
Remember me
Login
Don't have an account? Sign Up

Screenshort for welcome and booking:

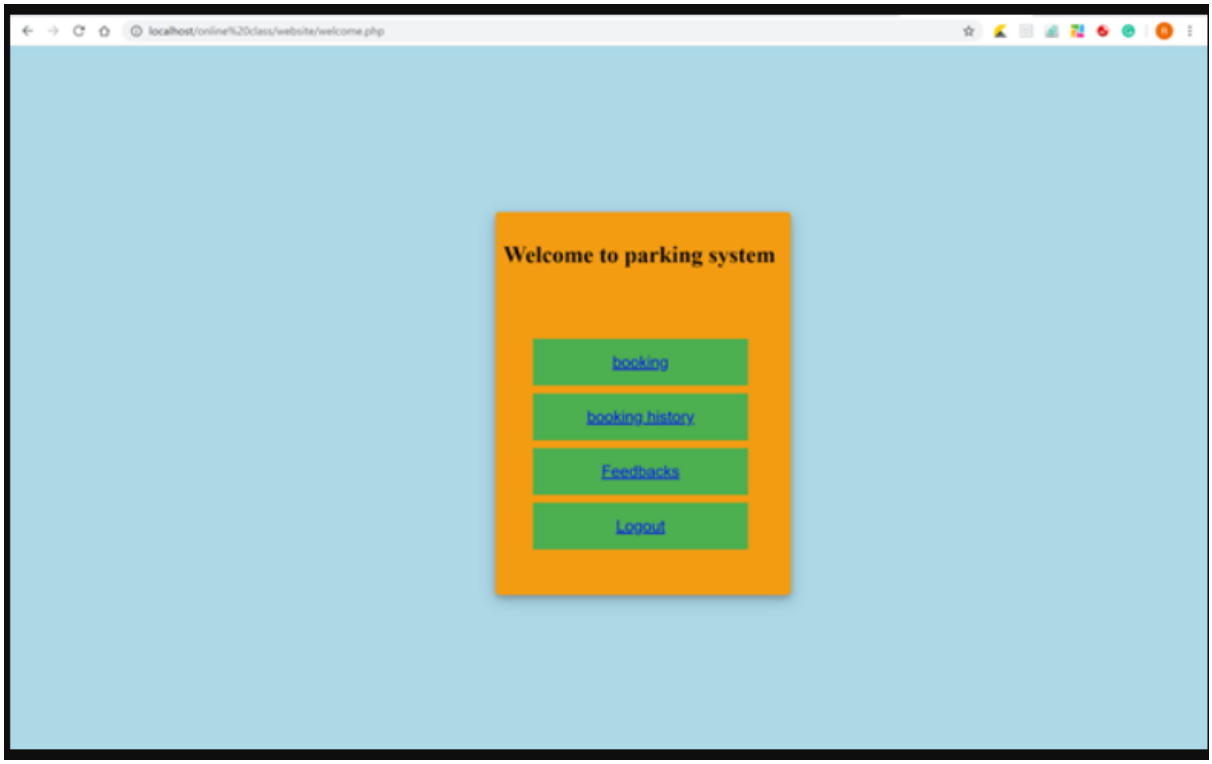
WELCOME.html > ? > style > .button2

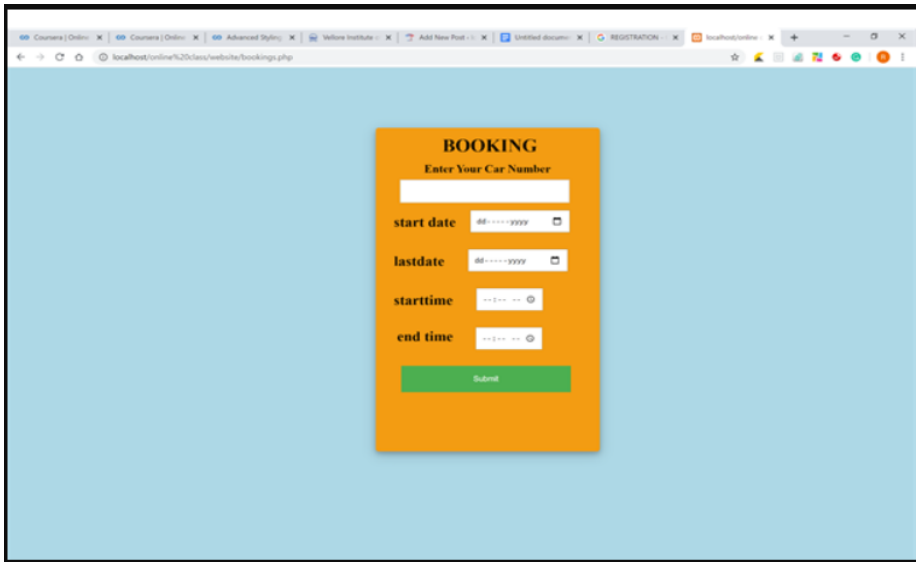
```php
<?php

session_start();
    if(!isset($_SESSION['userlogin'])){
        header("Location: bookings.php");
    }

    if(isset($_GET['logout'])){
        session_destroy();
        unset($_SESSION);
        header("Location: login.php");
    }

?>
<html>
<body>
<div class="first">
<p>Welcome to parking system</p>
<button class="button1"><a href="bookings.php">booking</a></bu


<button class="button2"><a href="booking1.php" >booking histor
<button class="button4"><a href="feedback1.php" >Feedbacks</a>
<button class="button3"><a href="index.php?logout=true">Logout
</div>
</body>
</html>
<style>
.first{
    height: 460px;
    width: 350px;
    margin-top: auto;
    margin-bottom: auto;
    background: #f39c12;
    position: relative;
    display: flex;
    justify-content: center;
```

booking.html > ?

```php
<?php
$db_user = "root";
$db_pass = "";
$db_name = "booking";
$db = new PDO('mysql:host=localhost;dbname=' . $db_name . ';ch
$db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

if(isset($_POST['search']))
{
    $carnumber  = $_POST['carnumber'];
    $txtstartdate =$_POST['txtstartdate'];
    $txtlastdate =$_POST['txtlastdate'];

    $stime =$_POST['stime'];
    $etime=$_POST['etime'];
    $sql="INSERT INTO bookings(carnumber,txtstartdate,txtlastd
    $stmtinsert =$db->prepare($sql);
    $result=$stmtinsert->execute([$carnumber,$txtstartdate,$tx

}



?>

<!DOCTYPE html>
<html>


<div class="first">
<form action="bookings.php" method="post">
<h1>BOOKING</h1>


<div class="row">
<label for="carnumber"><b>Enter Your Car Number </b></label>
<input class="form-control" id="carnumber" type="text" name="c
```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF HTML Port : 5500





Registration and booking: