# CAR PARKING MANAGEMENT SYSTEM

**A Project in Software Development Lab is submitted in partial fulfillment of the requirements for the Award of the degree of**

## MASTER OF COMPUTER SCIENCE AND APPLICATION

**SUBMITTED BY**

PAVITHRA .M
112105029

HARIDHARSHINI.C
112105016

## GUIDED BY

**Mr.M.KRISHNAMOORTHY.M.C.A.,M.Phil.,**
ASSISTANT PROFESSOR

**Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya**
**(SCSVMV)**
**(Deemed to be University U/S 3 of the UGC Act 1956)**
**Accredited with "A" Grade by NAAC**
**Enathur, Kanchipuram – 631 561**

**DECEMBER 2022**


## BONAFIDE CERTIFICATE

Certified that this project work entitled car parking management system is the bonafide work carried out by Mr./Ms. M.PAVITHRA Reg.No:112105029 in the MCA203P80- Mini Project of MCA during the III[rd] Semester of the academic year 2022.

**Mr.M.KRISHNAMOORTHY**
Assistant Professor
Department of Computer Science & Applications
SCSVMV

**DR.M.KANNAN**
Head of Department
Department of Computer Science & Applications
SCSVMV.

Submitted for the project work viva-voce examination held on _____

Internal Examiner

External Examiner

## ACKNOWLEDGMENT

"Project is the product out of experience that goes a long way in shaping up a person's caliber. The experience and success one attains is no by oneself but with a group of kind hearts behind".

First and foremost, we express our sincere thanks to our respected **Vice Chancellor** and our beloved **Registrar, Dean (Faculty of Science)** and **HOD, Dept of CSA**, for providing us adequate infrastructure and congenial academic environment.

We express our gratitude to **Project lab incharge**, **Dr.M.Kannan, Assistant Professor** whose guidance and encouragement has helped us in completing this project work.

We extend our sincere thanks to our internal guide, **Mr.M.Krishnamoorthy, Assistant Professor,** Dept of CSA for giving the confidence to complete the project successfully by providing the valuable suggestions and interest at every stage of the project.

Lastly, but not certainly the least, I express my warm thanks to my parents, family members, friends and well-wishers who helped me directly or indirectly in completing the project.

We would be failing in our duty if we don't mention the wholehearted support and technical assistance extended to us by our staff members and lab assistants of our department.

## CONTENTS

**Page No.**

# Chapter – I – Introduction

## 1.1 INTRODUCTION

- Nowadays parking has become an expensive resource in almost all majorities in the world, and its limited availability is the concurrent cause of urban traffic congestion and air pollution. The common method of ending a parking space is manual where the driver usually and a space on the street through luck and experience. The problem has been further exacerbated by the fact that nowadays even people from the low-income group are able to own cars.The user requests the Parking Control Unit to check the status of available parking slots. As soon as the user request, all the available free slots are displayed to the user. If the availability of parking space is confirmed, the user can book the parking slot .

- The vehicle follows its path towards the starting of the parking area. The user fixes his slots by showing his confirmation details to the concerned person at parking area.The main responsibility of the car parking management system is to help the user to find an area where parking is available and total number of slots free in that area.

## 1.2.ABSTRACT


- The main objective of this project is to avoid the congestion in the car parking area by implementing a parking management system.Normally at public places such as multiplex theaters,market areas, hospitals, function-halls, offices and shopping malls, one experiences the discomfort in looking out for a vacant parking slot, though it's a paid facility with an attendant/ security guard.Advanced online parking system is a project developed to provide an easy way in finding the parking space for vehicles.This project helps users by analyzing the areas where parking is available and details about number of slots free in that area.

- Advanced online parking system enables users to book before four hours prior to his expected arrival, the user can pre-book a slot in the place he desires if it is available.This will help reduce the load on the administrator as his physical work reduces drastically and user can search the parking slot easily.This system aims at providing the control system of the number of the cars in it, monitoring the movement of the parking lot, checking the space availability for new cars and the dwell time of the cars, thus ensuring the precision and the effectiveness of the system.The parking charges are automatically deducted from the user's account after checkout the slot.

## EXISTING

- In the existing system the number of personal vehicles usage is increasing manifold.Finding a parking space in most apartments, colleges especially during the rush hours, is difficult for drivers.No service provider are available, shopping mall and customer need to work as a unit to make the parking which takes a lot of time in searching for a parking slot.It includes the man power and expensive devices that results in high cost for maintenance.

## DRAWBACKS:

### The high cost of construction or installation

- The cost of having a sound, working parking management system is usually high. This is because of the various components that go into making the system work.Components such as the statistical feature, automated ticketing, and statistical reports, and many others make it all expensive. Some organizations may not be able to afford such.

### Regular maintenance

- The system is automated; however, it still requires several regular maintenances from the company.This is to ensure that the system is working perfectly and that nothing has gone wrong. The maintenance could be once in months.

### Operation

- A lot of people are not used to the parking management system. As a result, it may be difficult for them to make use of, thereby causing further complications during parking.

### Breakdown

- As a machine, the system could inevitably breakdown at some point. When this occurs, vehicles may not be able to have access to buildings, and cars parked inside might not be able to move.In another way, it could malfunction and lead cars to park in the wrong places.

# PROPOSED SYSTEM:

- To solve existing system drawbacks, we develop a time efficient application named as “car parking management system”.By using this application users like Admin can login into the system and they can perform their desired action in this application.The proposed system provides high security to the user details information including user and vehicle.This proposed Smart parking system consists of the deployed web development module which delivers real-time output and monitors the flow of the parking of vehicles in and out of that particular parking lot. The methodology provides the optimal solution for the parking space.

- System needs to store information about new Car Entry. System needs to update and delete the record. System also needs a search area.
- System needs to keep the record of the Parking vehicles.
- System needs to provide the information to the management staff.

- Creating the Parking Lot Management System web application using:

1. HTML
2. CSS
3. SQL
4.Dot net

## MODULE DESCRIPTION:

1) **USER MODULE:**

- The user module allows users to register, log in, and log out. Users benefit from being able to sign on because this associates content they create with their account and allows various permissions to be set for their roles.The user module supports user roles, which can be set up with fine-grained permissions allowing each role to do only what the administrator permits. Each user is assigned one or more roles.

**BOOKING MODULE:**

- This booking module is about parking your car in the available slot. This reservation system increases your online booking rates.Set your own booking rules, automate the reservation process, and provide your customers with an easy and reliable online service.

**ADMINISTRATOR MODULE :**

- This is the operative module of the application. It works in the backend for managing the database and performs various operations on it.The administrator stores all the user's data in the database as soon as he gets registered with the application. Administrator maintains the details of all parking slots ( both empty and reserved), their price for booking , user details in database and the modification on these data is only can be done by the administrator.

# REQUIREMENTS SPECIFICATION

## SOFTWARE REQUIREMENTS:

- Operating system : Microsoft Windows 11,64bit (operating system).
- Front End : Microsoft visual studio 2010 professional, 64 bit
- Back End : SQL Server

## HARDWARE REQUIREMENTS:

- System : Windows 11
- Speed : 2.42 GHz
- Hard Disk : 500GB
- RAM : 16GB

# Chapter – III – System Design

## 3.1 Database Design

### Register Table:

| COLUMN NAME | DATATYPE | DESCRIPTION |
|---|---|---|
| Name | Varchar(20) | Used to store the user name |
| Email | Varchar (20) | Used to store the Email id |
| Password | Varchar(20 | Used to store password |
| Vehicle_No | nvarchar(20) | Used to store the vehicle no of the user |

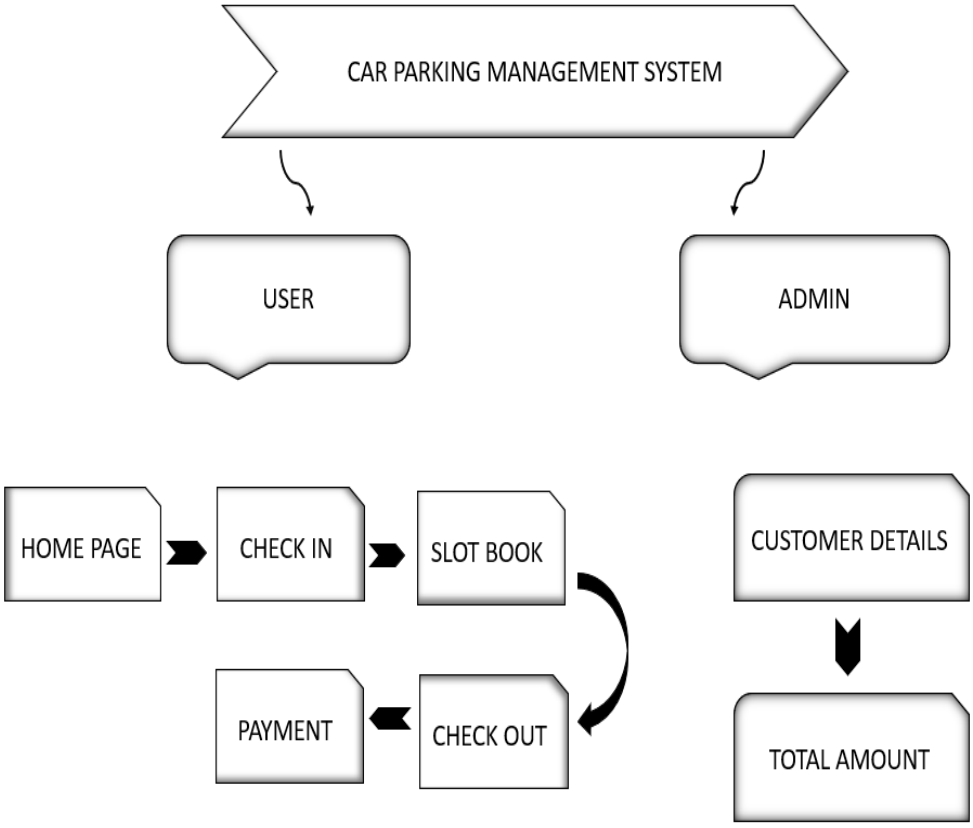
### BOOKSLOT TABLE:

| COLUMN NAME | DATATYPE | DESCRIPTION |
|---|---|---|
| Slotno | nvarchar(20) | Used to store the slot no |
| Vehicleno | nvarchar(20) | Used to store the vehicleno |
| Intime | datetime | Used to store the slot the booking time |

## Checkout slot:

| COLUMNE NAME | DATATYPE | DESRCIPTION |
|---|---|---|
| Slotno | Varchar(20) | Used to store the slotno |
| Vehicleno | nvarchar(20) | Used to the vehicleno |
| Intime | Datetime | Used to store the checkout time |
| Outtime | datetime | Used to store the checkout time |
| Amount | Numeric(18,2) | Used to store the total amount |

## 3.2. Overall System Design Structure




**MODULE DESCRIPTION :**

- Customer registeration
- Login
- Home page
- Checkin
- Checkout
- payment

**Chapter – IV – System Implementation**

## 4.1 Methodology used for Testing

## Unit Testing:

- Unit testing verification efforts on the smallest unit of software design, module. This is known as “Module Testing”. The modules are tested separately. This testing is carried out during programming stage itself. In these testing steps, each module is found to be working satisfactorily as regard to the expected output from the module.

- Module testing-Registration, Login, Admin, Doctor and Patients-all these modules went through unit testing. Each module independently is able to give output.

| | |
|---|---|
| **Test Case Description** | This test case deals with the creation of user information the creation program takes many inputs .the test should check for proper inputs and verify whether the creation function called properly with the correct input parameters. |
| **Expected Inputs** | Doctor details provided by admin. |
| **Expected Outputs** | Provide the patient details to Doctor. |
| **Actual Test Results** | An alert window was shown whenever the user give incorrect data such as entering numbers in the name field entering characters in numeric fields. |

## Black Box Testing:

- **Black box testing,** also known as Behavioral Testing, is a software testing method in which the internal structure/ design/ implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional.

## White-Box Testing:

- **White-box testing** (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing).and developers.

## User Interface Testing:

- In this testing strategy the developer will uncover errors related to specific interface mechanism and also uncover the errors such that the interface implements the semantic of navigation ,web app functionality ,or enter display.Front end application is giving appropriate output as desired.The front end application is user-friendly.

## Integration Testing:

- Integration testing is a systematic technique for constructing tests to uncover error associated within the interface. In the project, all the modules are combined and then the entire programmer is tested as a whole. In the integration-testing step, all the error uncovered is corrected for the next testing steps. Integration testing carried out to check if database connectivity is established throughout the module execution.

## System Testing:

- Here the developer performed testing on the complete ,integrated system. All modules rarely login ,activity ,relationship ,request invitation and enhanced modules are combined and tested together to evaluate the system's compliance with the specified requirements

## Validation Testing:

- The process of evaluating software during the development process or at the end of the development process to determine whether it satisfies specified business requirements. Validation Testing ensures that the product actually meets the client's needs. It can also be defined as to demonstrate that the product fulfil its intended use when deployed on appropriate environment.
- Validation testing on the project modules:
- Validation for checking proper email id format.
- Validation for checking whether mobile number is only number (10).

## Configuration Testing:

- Here the developer tested on different versions of web browsers and other screen resolutions.
- The web application was deployed on different versions of browsers and it worked properly.
- The application is flexible with different screen resolutions.

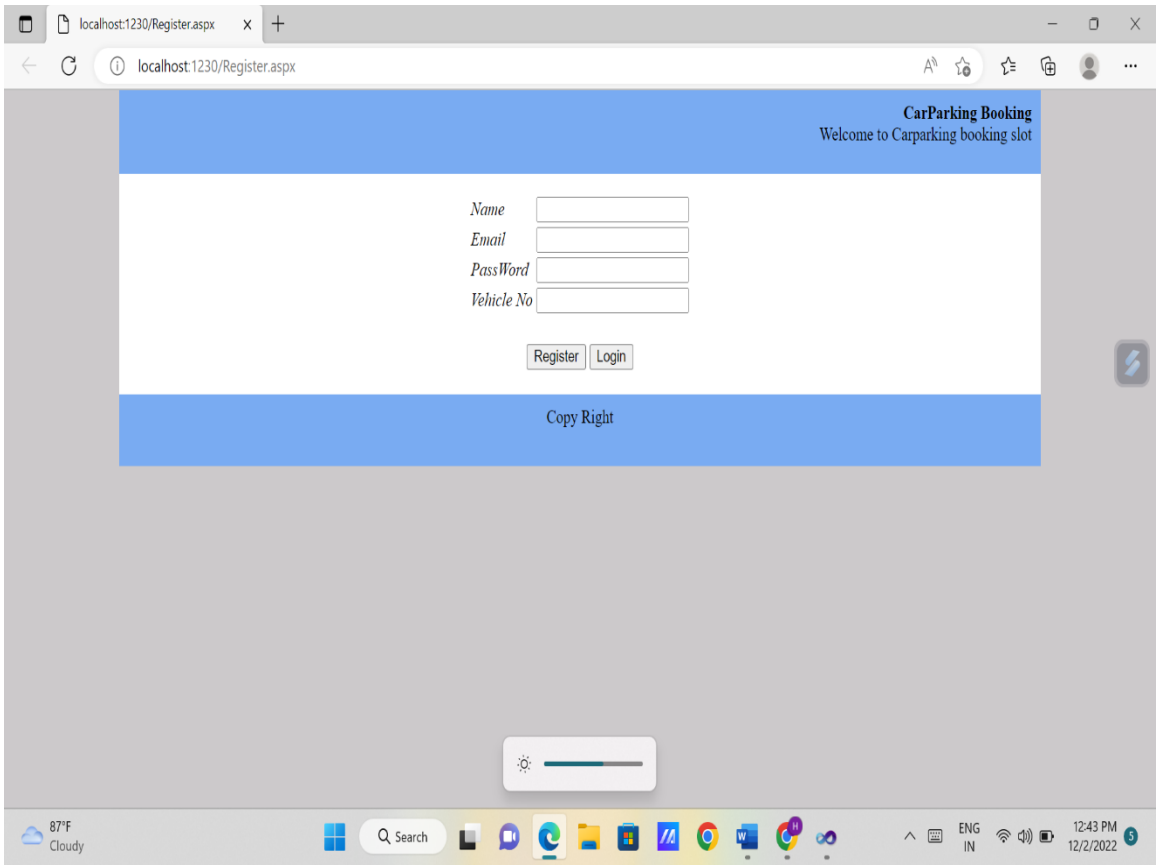
## 4.2 System Implementation

- **Systems implementation** is a set of procedures performed to complete the design (as necessary) contained in the *approved systems design document* and to test, install, and begin to use the new or revised Information System. Depicts systems implementation as the fifth major step in the development of an Information System.

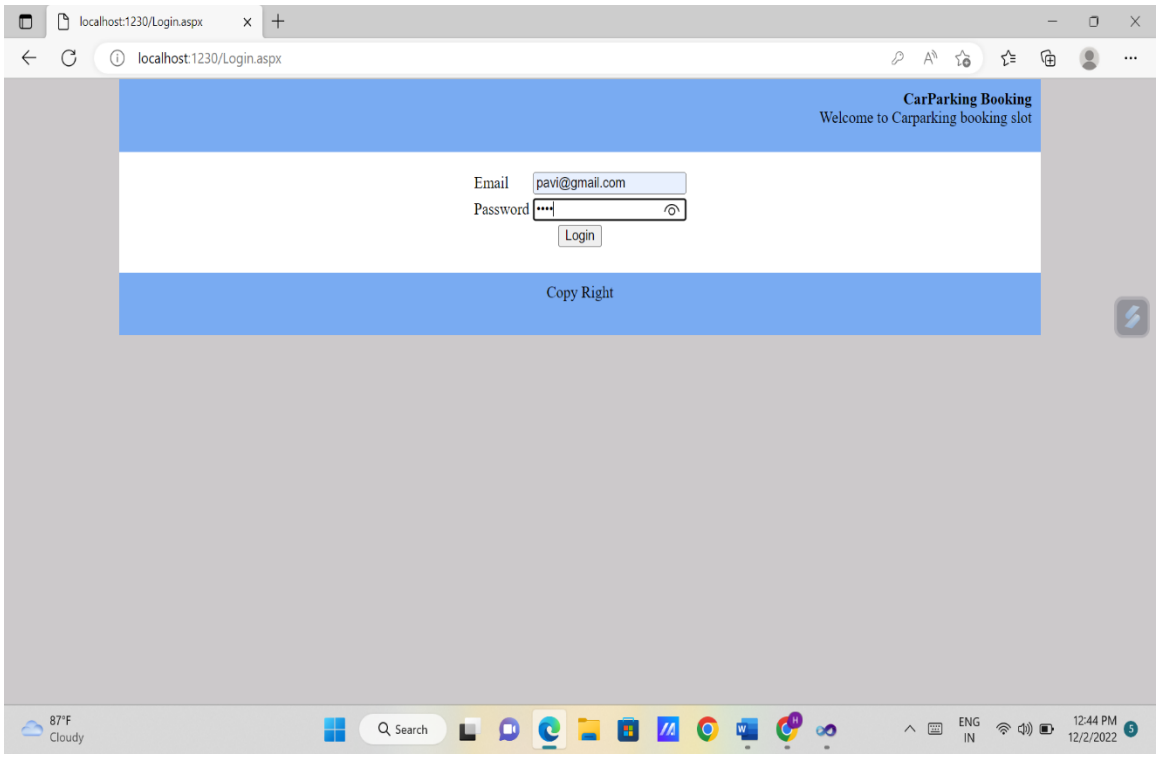
# Chapter – V – User Manual

## 5.1 Screen Shots

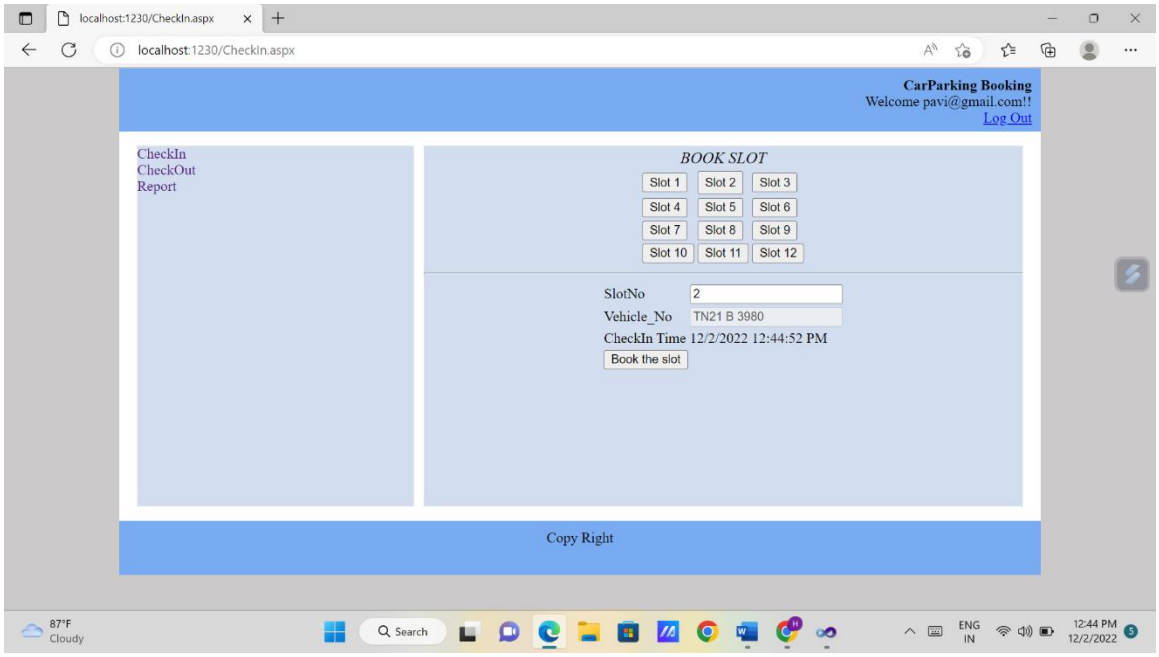
## Register Page :



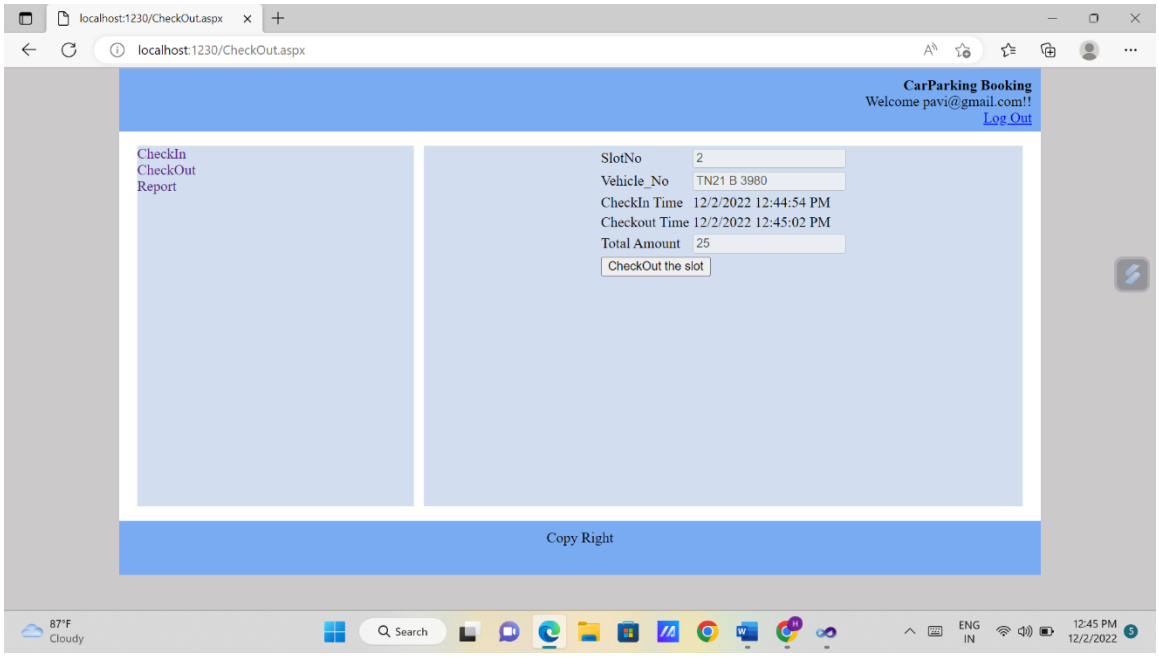
## Login Page :


localhost:1230/Login.aspx
CarParking Booking
Welcome to Carparking booking slot
Email
pavi@gmail.com
Password
Login
Copy Right
87°F
Cloudy
Search
ENG
IN
12:44 PM
12/2/2022

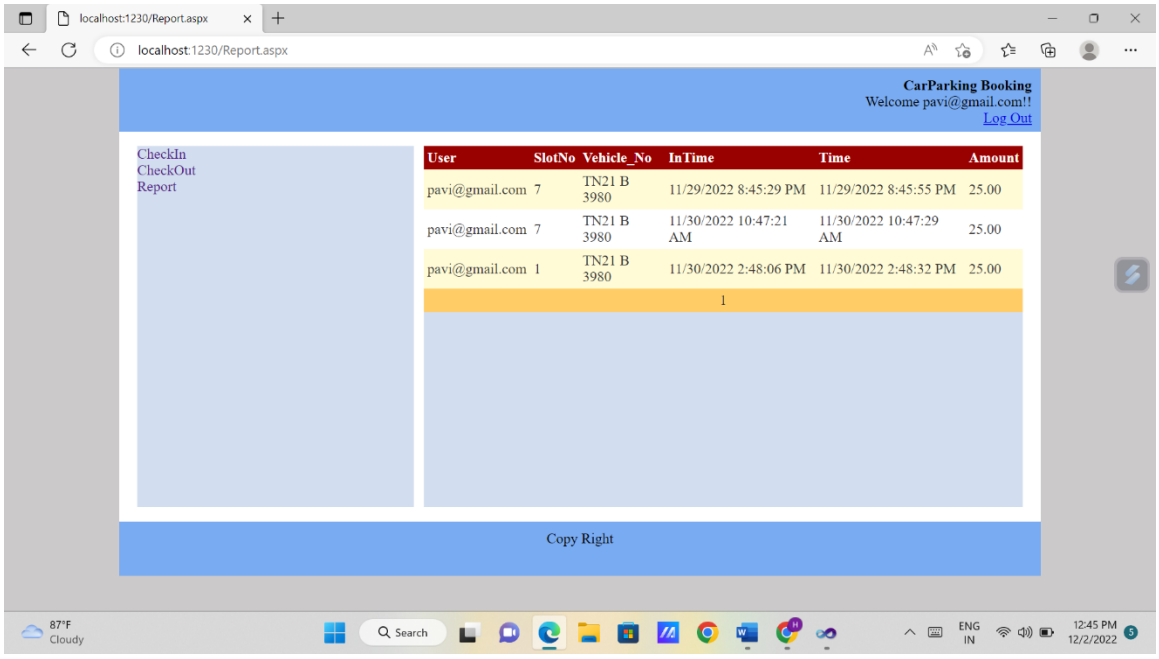
## Slot Book :


localhost:1230/CheckIn.aspx
CarParking Booking
Welcome pavi@gmail.com!!
Log Out
CheckIn
CheckOut
Report
BOOK SLOT
Slot 1
Slot 2
Slot 3
Slot 4
Slot 5
Slot 6
Slot 7
Slot 8
Slot 9
Slot 10
Slot 11
Slot 12
SlotNo
2
Vehicle_No
TN21 B 3980
CheckIn Time 12/2/2022 12:44:52 PM
Book the slot
Copy Right

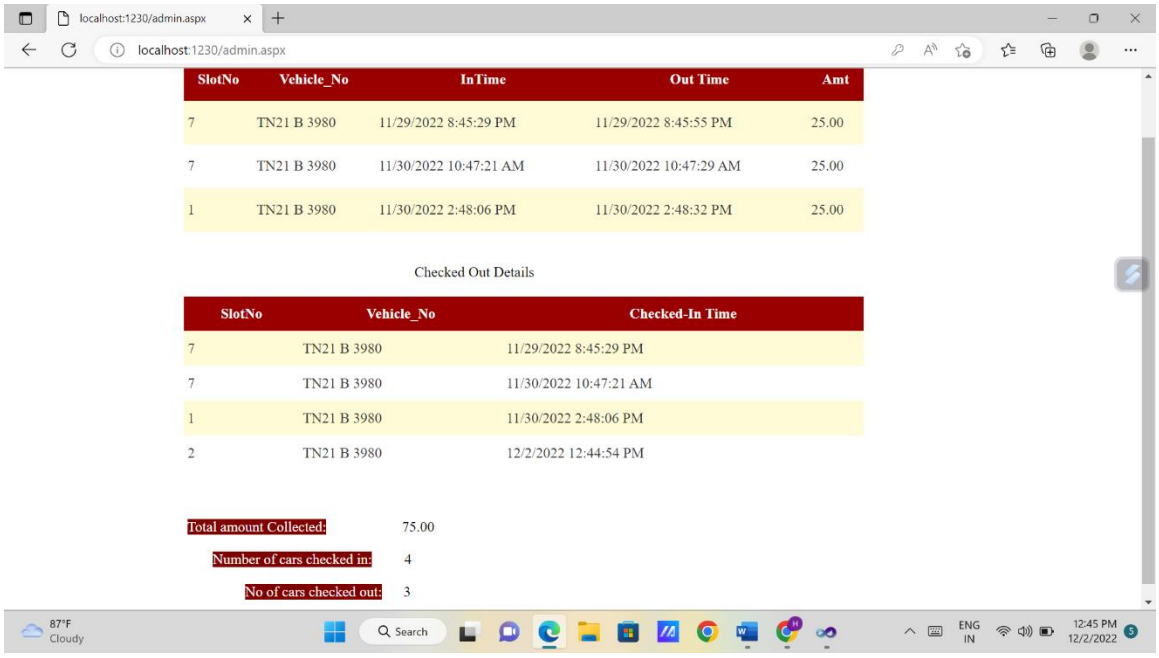
# CHECK OUT:


localhost:1230/CheckOut.aspx
CarParking Booking
Welcome pavi@gmail.com!!
Log Out
CheckIn
CheckOut
Report
SlotNo 2
Vehicle_No TN21 B 3980
CheckIn Time 12/2/2022 12:44:54 PM
Checkout Time 12/2/2022 12:45:02 PM
Total Amount 25
CheckOut the slot
Copy Right

# REPORT PAGE:


localhost:1230/Report.aspx
CarParking Booking
Welcome pavi@gmail.com!!
Log Out
CheckIn
CheckOut
Report
User SlotNo Vehicle_No InTime Time Amount
pavi@gmail.com 7 TN21 B 3980 11/29/2022 8:45:29 PM 11/29/2022 8:45:55 PM 25.00
pavi@gmail.com 7 TN21 B 3980 11/30/2022 10:47:21 AM 11/30/2022 10:47:29 AM 25.00
pavi@gmail.com 1 TN21 B 3980 11/30/2022 2:48:06 PM 11/30/2022 2:48:32 PM 25.00
1
Copy Right

# ADMIN PAGE:


localhost:1230/admin.aspx
localhost:1230/admin.aspx
SlotNo Vehicle_No InTime Out Time Amt
7 TN21 B 3980 11/29/2022 8:45:29 PM 11/29/2022 8:45:55 PM 25.00
7 TN21 B 3980 11/30/2022 10:47:21 AM 11/30/2022 10:47:29 AM 25.00
1 TN21 B 3980 11/30/2022 2:48:06 PM 11/30/2022 2:48:32 PM 25.00
Checked Out Details
SlotNo Vehicle_No Checked-In Time
7 TN21 B 3980 11/29/2022 8:45:29 PM
7 TN21 B 3980 11/30/2022 10:47:21 AM
1 TN21 B 3980 11/30/2022 2:48:06 PM
2 TN21 B 3980 12/2/2022 12:44:54 PM
Total amount Collected: 75.00
Number of cars checked in: 4
No of cars checked out: 3
87°F Cloudy
Search
ENG IN
12:45 PM 12/2/2022

# Chapter – V1 – Conclusion

## 6.1 Conclusion

- Our project successfully reduces the parking problem in places of entertainment such as theatres and shopping malls. Our project helps in finding out the availability of a parking slot , get the availability confirmed, and reach the place within the time slot allotted.It helps the administration to allocate the vacant slot to the next person in queue. Our project saves the time of visitors in booking a parking slot.

# Chapter – VII – Bibliography

## Appendix:

**BOOK:** ".NET and .NET Core official support policy"; supports qualifier: version type; language of work or name: American English; retrieved: 8 November 2022; last update: 8 November 2022.

**WEBSITE**: *"core/LICENSE.TXT"*. *GitHub*. *Retrieved June 4, 2018.*

# Register page

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Data;

namespace CarParkingBooking
{
   public partial class Register : System.Web.UI.Page
   {
      Connection conobj = new Connection();
      protected void Page_Load(object sender, EventArgs e)
      {
         if (!IsPostBack)
         {
            conobj.connect();
          //lblmsg.Visible = false;
         }
      }

      protected void TextBox5_TextChanged(object sender, EventArgs e)
      {

  SqlCommand cmd;
         String s;
```

```
            s = "SELECT COUNT(1) FROM[carpark] WHERE Email=@Email";
            cmd = new SqlCommand(s, Connection.con);
            cmd.Parameters.AddWithValue("@Email", TextBox2.Text);

            //Connection.con.Open();
            int count = Convert.ToInt32(cmd.ExecuteScalar());
            if (count == 0)
            {
                s = "insert into [carpark]
(Name,Email,Password,Vehicle_No)values(@Name,@Email,@Password,@Vehicle_No)
;";
                cmd = new SqlCommand(s, Connection.con);
                cmd.Parameters.AddWithValue("@Name", TextBox1.Text);
                cmd.Parameters.AddWithValue("@Email", TextBox2.Text);
                cmd.Parameters.AddWithValue("@Password", TextBox3.Text);
                cmd.Parameters.AddWithValue("@Vehicle_No", TextBox4.Text);
                //cmd.Parameters.AddWithValue("@Userid", "");
                //Connection.con.Open();
                cmd.ExecuteNonQuery();
                Response.Redirect("Login.aspx");
                //Connection.con.Close();
            }
            else
            {
              //lblmsg.Text = "User Already Exist";
               //lblmsg.Visible = true;
            }
        }
 protected void Button2_Click(object sender, EventArgs e)
        {
            Response.Redirect("Login.aspx"); }
```

# Login page

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;

namespace CarParkingBooking
{
   public partial class Login : System.Web.UI.Page
   {

      String s;
      SqlCommand cmd;
      protected void Page_Load(object sender, EventArgs e)
      {
         lblinfo.Visible = false;
         lblinfo.Text="";
      }

      protected void Button1_Click(object sender, EventArgs e)
      {

         //SqlConnection con = new SqlConnection(Connection.con.ConnectionString);
         s = "SELECT COUNT(1) FROM[carpark] WHERE Email=@Email AND
Password=@Password;";
```

```
    cmd = new SqlCommand(s, Connection.con);
    cmd.Parameters.AddWithValue("@Email", TextBox1.Text);
    cmd.Parameters.AddWithValue("@Password", TextBox2.Text);
    //Connection.con.Open();
    int count = Convert.ToInt32(cmd.ExecuteScalar());
    if (count == 1)
    {
        Session["Name"] = TextBox1.Text.Trim();
        Response.Redirect("Homepage.aspx");
        //Connection.con.Close();
    }
    else if(TextBox1.Text=="admin" && TextBox2.Text=="haridharshini")
    {
        Session["Name"] = TextBox1.Text.Trim();
        Response.Redirect("admin.aspx");
    }
    else
    {
        lblinfo.Visible = true;
        lblinfo.Text = "Invalid User Name!!!!";
    }
}

protected void TextBox1_TextChanged(object sender, EventArgs e)
{
    lblinfo.Visible = false;
    lblinfo.Text = "";
}

protected void TextBox2_TextChanged(object sender, EventArgs e)
 lblinfo.Visible = false;
```

```
 lblinfo.Text = "";
        }
    }
}
```

homepage

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace CarParkingBooking
{
    public partial class WebForm1 : System.Web.UI.Page
```

```
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (Session["Name"] == null)
            {
                Response.Redirect("Login.aspx");
            }
        }
    }
}
```

Checkin page

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Data.SqlClient;
using System.Data;
```

```
using System.Web.UI.WebControls;

namespace CarParkingBooking
{
   public partial class CheckIn : System.Web.UI.Page
   {
      Connection conobj = new Connection();
      protected void Page_Load(object sender, EventArgs e)
      {
         if (Session["Name"] == null)
         {
            Response.Redirect("Login.aspx");
         }

         lblmsg.Visible = false;

         SqlCommand cmdd;
         SqlDataReader rd, rd1, rd2;
         string r1, r2, r3, r4, r5, r6, r7, r8,r9,r10,r11,r12;
         string s1, s2, s3, s4, s5, s6, s7, s8, s9, s10, s11, s12, s13;

         // string[] s=new string[7];

         if (IsPostBack != true)
         {
            Label6.Text = DateTime.Now.ToString();
            {
               r1 = "select Status from slotstatus where SlotNo = 1 ;";
               cmdd = new SqlCommand(r1, Connection.con);
               if (Connection.con.State == ConnectionState.Closed)
                  Connection.con.Open();
               rd = cmdd.ExecuteReader();
               rd.Read();
               s1 = rd.GetString(0);
               if (s1 == "booked")
               {
                  Button1.Enabled = false;
               }
               if (Connection.con.State == ConnectionState.Open)
                  Connection.con.Close();
            }
            {
               r2 = "select Status from slotstatus where SlotNo = 2 ;";
               SqlCommand cmd1 = new SqlCommand(r2, Connection.con);
               Connection.con.Open();
               rd1 = cmd1.ExecuteReader();
```

```
    rd1.Read();
    s2 = rd1.GetString(0);
    //Response.Write(s2);
    if (s2 == "booked")
    {
        Button2.Enabled = false;
    }
    Connection.con.Close();
}

{
    r3 = "select Status from slotstatus where SlotNo = 3 ;";
    cmdd = new SqlCommand(r3, Connection.con);
    Connection.con.Open();
    rd2 = cmdd.ExecuteReader();
    rd2.Read();
    s3 = rd2.GetString(0);
    //Response.Write(s3);
    if (s3 == "booked")
    {
        Button3.Enabled = false;
    }
    Connection.con.Close();
}
{
    r4 = "select Status from slotstatus where SlotNo = 4;";
    cmdd = new SqlCommand(r4, Connection.con);
    Connection.con.Open();
    rd = cmdd.ExecuteReader();
    rd.Read();
    s4 = rd.GetString(0);
    //Response.Write(s4);
    if (s4 == "booked")
    {
        Button4.Enabled = false;
    }
    Connection.con.Close();
}
{
    r5 = "select Status from slotstatus where SlotNo = 5 ;";
    cmdd = new SqlCommand(r5, Connection.con);
    Connection.con.Open();
    rd = cmdd.ExecuteReader();
    rd.Read();
    s5 = rd.GetString(0);
    //Response.Write(s5);
```

```
    if (s5 == "booked")
    {
        Button5.Enabled = false;
    }
    Connection.con.Close();
}
{
    r6 = "select Status from slotstatus where SlotNo = 6 ;";
    cmdd = new SqlCommand(r6, Connection.con);
    Connection.con.Open();
    rd = cmdd.ExecuteReader();
    rd.Read();
    s6 = rd.GetString(0);
    //Response.Write(s1);
    if (s6 == "booked")
    {
        Button6.Enabled = false;
    }
    Connection.con.Close();
}
{
    r7 = "select Status from slotstatus where SlotNo = 7 ;";
    cmdd = new SqlCommand(r7, Connection.con);
    Connection.con.Open();
    rd = cmdd.ExecuteReader();
    rd.Read();
    s7 = rd.GetString(0);
    if (s7 == "booked")
    {
        Button7.Enabled = false;
    }
    Connection.con.Close();
}
{
    r8 = "select Status from slotstatus where SlotNo = 8 ;";
    cmdd = new SqlCommand(r8, Connection.con);
    Connection.con.Open();

    rd = cmdd.ExecuteReader();
    rd.Read();
    s8 = rd.GetString(0);
    if (s8 == "booked")
    {
        Button8.Enabled = false;
    }
    Connection.con.Close();
```

```
}
{
   r9 = "select Status from slotstatus where SlotNo =9 ;";
   cmdd = new SqlCommand(r9, Connection.con);
   Connection.con.Open();

   rd = cmdd.ExecuteReader();
   rd.Read();
   s9 = rd.GetString(0);
   if (s9 == "booked")
   {
      Button9.Enabled = false;
   }
   Connection.con.Close();

}
{
   r10 = "select Status from slotstatus where SlotNo =10 ;";
   cmdd = new SqlCommand(r10, Connection.con);
   Connection.con.Open();

   rd = cmdd.ExecuteReader();
   rd.Read();
   s10 = rd.GetString(0);
   if (s10 == "booked")
   {
      Button10.Enabled = false;
   }
   Connection.con.Close();

}
{
   r11 = "select Status from slotstatus where SlotNo =11 ;";
   cmdd = new SqlCommand(r11, Connection.con);
   Connection.con.Open();

   rd = cmdd.ExecuteReader();
   rd.Read();
   s11 = rd.GetString(0);
   if (s11 == "booked")
   {
      Button11.Enabled = false;
   }
   Connection.con.Close();
```

```
            }
            {

                r12 = "select Status from slotstatus where SlotNo =12 ;";
                cmdd = new SqlCommand(r12, Connection.con);
                Connection.con.Open();
                rd = cmdd.ExecuteReader();
                rd.Read();
                s12 = rd.GetString(0);
                if (s12 == "booked")
                {
                    Button12.Enabled = false;
                }
                Connection.con.Close();
            }
            {

                string s = "select Vehicle_No from carpark where Email = '" +
Session["Name"].ToString() + "' ";
                cmdd = new SqlCommand(s, Connection.con);
                Connection.con.Open();
                rd = cmdd.ExecuteReader();
                rd.Read();
                s13 = rd.GetString(0);
                TextBox2.Text = s13.ToString();
                Connection.con.Close();

            }


        }
    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        TextBox1.Text = "1";

    }

    protected void Button1_Click1(object sender, EventArgs e)
    {
        TextBox1.Text = "1";
    }

    protected void Timer1_Tick(object sender, EventArgs e)
    {
```

```
    Label6.Text = DateTime.Now.ToString();
}

protected void Button2_Click(object sender, EventArgs e)
{
    TextBox1.Text = "2";

}

protected void Button3_Click(object sender, EventArgs e)
{
    TextBox1.Text = "3";
}

protected void Button4_Click(object sender, EventArgs e)
{
    TextBox1.Text = "4";
}

protected void Button5_Click(object sender, EventArgs e)
{
    TextBox1.Text = "5";
}

protected void Button6_Click(object sender, EventArgs e)
{
    TextBox1.Text = "6";
}

protected void Button7_Click(object sender, EventArgs e)
{
    TextBox1.Text = "7";
}

protected void Button8_Click(object sender, EventArgs e)
{
    TextBox1.Text = "8";
}

protected void Button9_Click(object sender, EventArgs e)
{
    TextBox1.Text = "9";
}

protected void Button10_Click(object sender, EventArgs e)
{
```

```
        TextBox1.Text = "10";
    }

    protected void Button11_Click(object sender, EventArgs e)
    {
        TextBox1.Text = "11";
    }

    protected void Button12_Click(object sender, EventArgs e)
    {
        TextBox1.Text = "12";
    }

    protected void Button13_Click(object sender, EventArgs e)
    {
        SqlCommand cmd;
        string s1 = "SELECT COUNT(1) FROM bookslot b inner join  slotstatus s on
b.SlotNo=s.SlotNo WHERE b.Vehicle_No='" + TextBox2.Text + "' AND
s.Status='booked';";
        cmd = new SqlCommand(s1, Connection.con);
        Connection.con.Open();
        int count = Convert.ToInt32(cmd.ExecuteScalar());

        if (count == 0)
        {
            string s = "insert into
bookslot(SlotNo,Vehicle_No,Time)values(@t_SlotNo,@t_Vehicle_No,@t_Time);";
            cmd = new SqlCommand(s, Connection.con);
            cmd.Parameters.AddWithValue("@t_SlotNo", TextBox1.Text);
            cmd.Parameters.AddWithValue("@t_Vehicle_No", TextBox2.Text);
            cmd.Parameters.AddWithValue("@t_Time", Label6.Text);
            //Connection.con.Open();
            cmd.ExecuteNonQuery();
            //Response.Write("Inserted");
            lblmsg.Text = "Inserted";
            lblmsg.Visible = true;
            //Connection.con.Close();
            //Button1.Enabled = false;
            //Button1.BackColor = Color.Red;
            string slot = "update slotstatus set Status='booked' where SlotNo=@slotno;";
            cmd = new SqlCommand(slot, Connection.con);
            cmd.Parameters.AddWithValue("@slotno", TextBox1.Text);
            //Connection.con.Open();
            cmd.ExecuteNonQuery();
            Connection.con.Close();
        }
```

```
            else
            {
                lblmsg.Text = "Already slot booked for this Vehicle. Kindly do checkout first";
                lblmsg.Visible = true;
            }
        }

        protected void UpdateTimer_Tick(object sender, EventArgs e)
        {
            Label6.Text = DateTime.Now.ToString();
        }

        protected void TextBox2_TextChanged(object sender, EventArgs e)
        {

        }
    }
}
```

Checkout:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Data;
```

```
namespace CarParkingBooking
{
   public partial class CheckOut : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
         if (Session["Name"] == null)
         {
            Response.Redirect("Login.aspx");
         }

         lblmsg.Visible = false;

         SqlCommand cmdd;
         SqlDataReader rd;
         if (!IsPostBack)
         {
            string s = "select TOP (1) c.Vehicle_No,b.SlotNo,b.Time from carpark c inner
join bookslot b on c.Vehicle_No=b.Vehicle_No inner join slotstatus s on b.slotNo =
s.slotno where c.Email = '" + Session["Name"].ToString() + "' AND s.Status='booked'
ORDER BY b.Time DESC ";
            cmdd = new SqlCommand(s, Connection.con);
            if (Connection.con.State == ConnectionState.Closed)
               Connection.con.Open();
            rd = cmdd.ExecuteReader();
            DataTable dt = new DataTable();
            dt.Load(rd);
            if (dt.Rows.Count > 0)
            {
               TextBox1.Text = dt.Rows[0][1].ToString();
               TextBox2.Text = dt.Rows[0][0].ToString();
               DateTime intime = Convert.ToDateTime(dt.Rows[0][2].ToString());
               Label7.Text = dt.Rows[0][2].ToString();
               DateTime outtime = DateTime.Now;
               TimeSpan interval = outtime - intime;
               if (interval.TotalHours < 1)
                  TextBox3.Text = "25";
               else if (interval.TotalHours > 1)
                  TextBox3.Text = (interval.TotalHours * 25).ToString("0.00");
            }
            else
            {
               Button13.Enabled = false;
               lblmsg.Text = "ThankYou";
               lblmsg.Visible = true;
            }
```

```
            if (Connection.con.State == ConnectionState.Open)
                Connection.con.Close();

        }
    }

    protected void Button13_Click(object sender, EventArgs e)
    {
        string s = "insert into
CheckoutSlot(SlotNo,Vehicle_No,Time,Amt,InTime)values(@t_SlotNo,@t_Vehicle_No
,@t_Time,@amt,@o_Time);";
        SqlCommand cmd = new SqlCommand(s, Connection.con);
        cmd.Parameters.AddWithValue("@t_SlotNo", TextBox1.Text);
        cmd.Parameters.AddWithValue("@t_Vehicle_No", TextBox2.Text);
        cmd.Parameters.AddWithValue("@t_Time", Label6.Text);
        cmd.Parameters.AddWithValue("@amt", TextBox3.Text);
        cmd.Parameters.AddWithValue("@o_Time", Label7.Text);
        if (Connection.con.State == ConnectionState.Closed)
            Connection.con.Open();
        cmd.ExecuteNonQuery();
        //Response.Write("Inserted");
        lblmsg.Text = "Checkedout Sucessfully";
        lblmsg.Visible = true;
        //Connection.con.Close();
        //Button1.Enabled = false;
        //Button1.BackColor = Color.Red;
        string slot = "update slotstatus set Status='UnBooked' where SlotNo=@slotno;";
        cmd = new SqlCommand(slot, Connection.con);
        cmd.Parameters.AddWithValue("@slotno", TextBox1.Text);
        //Connection.con.Open();
        cmd.ExecuteNonQuery();
        if (Connection.con.State == ConnectionState.Open)
            Connection.con.Close();
        checkAvlSlot();
    }

    protected void UpdateTimer_Tick(object sender, EventArgs e)
    {
        Label6.Text = DateTime.Now.ToString();
    }
    void checkAvlSlot()
    {
        string s = "select TOP (1) c.Vehicle_No,b.SlotNo,b.Time from carpark c inner
join bookslot b on c.Vehicle_No=b.Vehicle_No inner join slotstatus s on b.slotNo =
s.slotno where c.Email = '" + Session["Name"].ToString() + "' AND s.Status='booked'
ORDER BY b.Time DESC ";
```

```
            SqlCommand cmdd = new SqlCommand(s, Connection.con);
            if (Connection.con.State == ConnectionState.Closed)
                Connection.con.Open();
            SqlDataReader rd = cmdd.ExecuteReader();
            DataTable dt = new DataTable();
            dt.Load(rd);
            if (dt.Rows.Count > 0)
            {
                TextBox1.Text = dt.Rows[0][1].ToString();
                TextBox2.Text = dt.Rows[0][0].ToString();
                DateTime intime = Convert.ToDateTime(dt.Rows[0][2].ToString());
                Label7.Text = dt.Rows[0][2].ToString();
                DateTime outtime = DateTime.Now;
                TimeSpan interval = outtime - intime;
                if (interval.TotalHours < 1)
                    TextBox3.Text = "25";
                else if (interval.TotalHours > 1)
                    TextBox3.Text = (interval.TotalHours * 25).ToString("0.00");
            }
            else
            {
                Label7.Text = "";
                TextBox1.Text = "";
                TextBox2.Text = "";
                TextBox3.Text = "";
                Button13.Enabled = false;
                lblmsg.Text = "Please book your slot first!!";
                lblmsg.Visible = true;
            }
        }
}
        }
```

Report:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Data;
```

```
namespace CarParkingBooking
{
   public partial class Report : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
         BindData1();
      }
      public void BindData1()
      {
         SqlConnection con = new SqlConnection( Connection.con.ConnectionString);
         SqlCommand cmd = new SqlCommand();
         DataSet ds = new DataSet();
         cmd.CommandText = "Select
p.Email,c.SlotNo,c.Vehicle_No,c.InTime,c.Time,c.Amt from checkoutSlot c inner join
carpark p on c.vehicle_No=p.vehicle_No where p.Email='" +
Session["Name"].ToString() + "'";
         cmd.Connection = con;
         SqlDataAdapter da = new SqlDataAdapter(cmd);
         da.Fill(ds);
         con.Open();
         cmd.ExecuteNonQuery();
         Grid1.DataSource = ds;
         Grid1.DataBind();
         con.Close();
      }
   }
}
```

Admin:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Data;
namespace CarParkingBooking
```

```
{
   public partial class admin : System.Web.UI.Page
   {
      //Connection con = new Connection("Data
Source=.\\SQLEXPRESS;AttachDbFilename=\"C:\\Users\\dhars\\Documents\\Visual
Studio 2010\\Projects\\WebSite1\\carpark.mdf\";Integrated Security=True;Connect
Timeout=30;User Instance=True");
      //Connection con = new Connection("Data
Source=.\\SQLEXPRESS;AttachDbFilename=C:\\Users\\dhars\\Documents\\Visual
Studio 2010\\Projects\\WebSite1\\carpark.mdf;Integrated Security=True;Connect
Timeout=30;User Instance=True");

      protected void Page_Load(object sender, EventArgs e)
      {

         decimal amt = 0 ;
          for(int i = 0 ; i<GridView1.Rows.Count;i++)
          {
             amt += Convert.ToDecimal(GridView1.Rows[i].Cells[4].Text);
          }
          Label2.Text = amt.ToString();//total amount
          Label7.Text = GridView1.Rows.Count.ToString();//checked out no
          Label6.Text = GridView2.Rows.Count.ToString();


      }

   }
}
```