CHAPTER I

INTRODUCTION

1.1ABOUT THE PROJECT

This project tourist guide provides the tourist with city map depending on its Current location entered by the website phone user. This information helps the tourists to find the desired locations to visit. Well it consists of entire details of those locations or how to reach the location as well as other emergency amenities like hospitals, institutes, bus stops etc but it provides the basic information to decide the places to visit. This project is mainly beneficial for the tourist's having no idea about the places they want to visit. By providing geographic based information system the tourists and people shifting to new cities can get a better guidance of the places they want to visit .This proposed application does not require any internet access and thus eliminates the disadvantage of single point failure. By making the application GIS based, it includes many advantages as the user can view the required location in map and accordingly estimate the time that will be required to reach the final destination. The system gives the basic details that will be required such as an image of that place along with basic details like the address, contact no etc. m. Even if a visitor can find new destinations via the internet, obtaining necessary information about the places to visit is time consuming when the tourist is unfamiliar with the area. This article discusses a website that uses mobile software technology to improve the tourism experience. Mobile tour management services will help a developer create a mobile application based on real time data values to improve the vacation experience. It will assist one in obtaining information on places to visit, lodges to stay at, restaurants to dine at, and checking out images and current weather that narrates the area description that they are now visiting without the use of a traditional guide.

CHAPTER II

SYSTEM STUDY

2.1 EXISTING SYSTEM

In existing system a normal manual working system is only generated where the users need to meet the admin directly and collect details about the travel packages. Normal travel package booking system is done in the proposed system where the implementation can be normally processed with the system. The implementation can be maintained with normal web based interface generation approach.

2.2 DISADVANTAGES

- Manual Working is only done
- Analysis of the interaction cannot be known

2.3 PROPOSED SYSTEM

In our proposed system a website developed for the customer to know the details about tourism spot as a computerized form. The customers can create an account and use this website to know about the amounts, travel details, source, destination etc. The website also enhanced with review system where the customer can also book their tour package on analyzing the reviews. Separate back end developed for storage where the data can be retrieved at any time when the admin needs.

2.4 ADVANTAGES

- The system reduces the manual work of both admin and customer
- Easily stored and retrieved any time
- The amount will be calculated per by person

2.5 PROBLEM DEFINITION AND DESCRIPTION

In the problem statement, the tourist places cannot be processed in the existing system where the complete verification system. The implementation can be done with the tourist spot managing system. The analysis of the system can be done with this implemented maintenance approach can be done.

CHAPTER III

SYSTEM ANALYSIS

3.1 Packages Selected

Front End	: ASP.NET
Back End	: SQL SERVER 2005

3.2 Resources Required

3.2.1 Hardware Requirements

Processor	: Dual core processor 2.6.0 GHz
RAM	: 1GB
Hard disk	: 160 GB
Compact Disk	: 650 MB
Keyboard	: Standard keyboard
Monitor	: 15 inch color monitor

3.2.2 Software Requirements

Front End	: ASP.NET
Back End	: SQL SERVER 2005
Operating System	:Windows 11

3.3 Data flow Diagram

Level 0

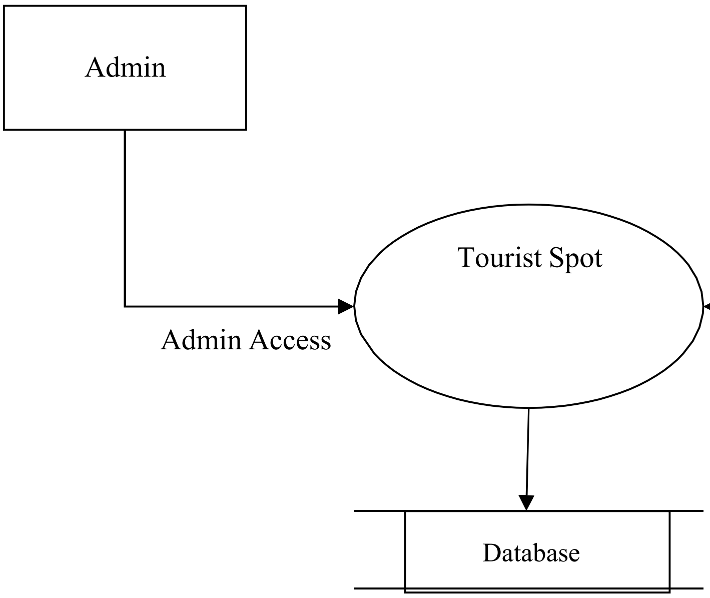


Fig no: 3.1 data flow level 0

Level 1

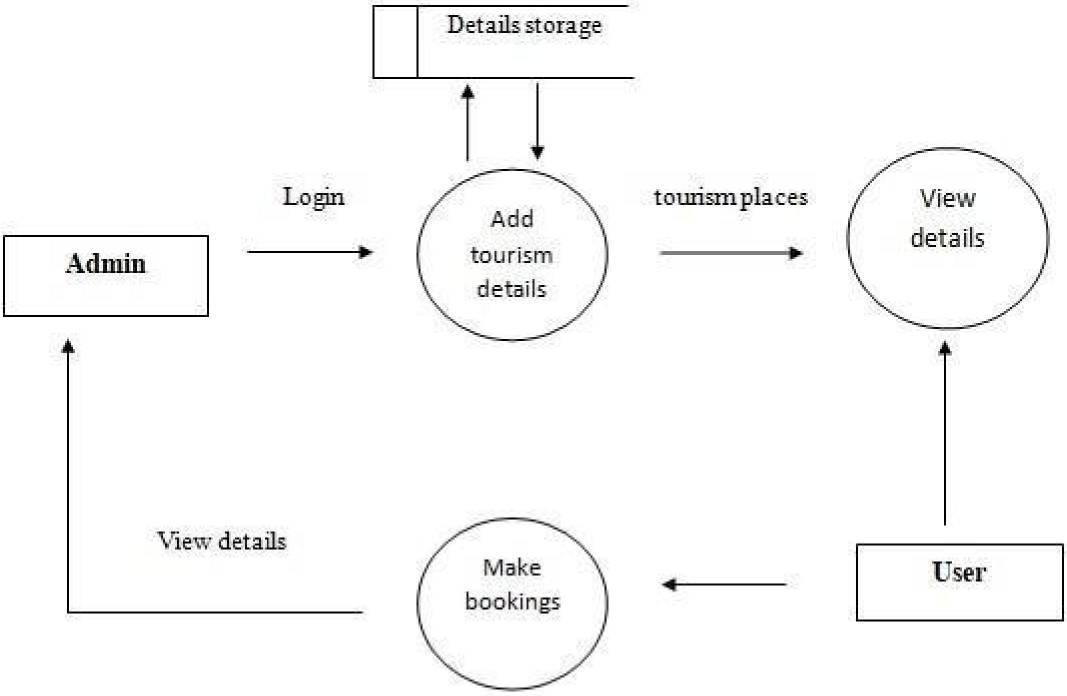


Fig no: 3.2 data flow level 1

CHAPTER IV

SYSTEM DESIGN

4.1 ARCHITECTURAL DESIGN

A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. System architecture can comprise system components, the externally visible properties of those components, the relationships (e.g. the behavior) between them. It can provide a plan from which products can be procured, and systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture; collectively these are called architecture description languages (ADLs).

Various organizations define systems architecture in different ways, including:

- An allocated arrangement of physical elements which provides the design solution for a consumer product or life-cycle process intended to satisfy the requirements of the functional architecture and the requirements baseline.
- Architecture comprises the most important, pervasive, top-level, strategic inventions, decisions, and their associated rationales about the overall structure (i.e., essential elements and their relationships) and associated characteristics and behavior.
- If documented, it may include information such as a detailed inventory of current hardware, software and networking capabilities; a description of long-range plans and priorities for future purchases, and a plan for upgrading and/or replacing dated equipment and software.

An architecture diagram is a graphical representation of a set of concepts that are part of architecture, including their principles, elements and components. Architecture diagram can help system designers and developers visualize the high-level, overall structure of their system or application, in order to ensure the system meets their users' needs. Using architecture diagram, you can also describe patterns that are used throughout the design. It's somewhat like a blueprint that you use as a guide, so that you and your colleagues can discuss, improve and follow.

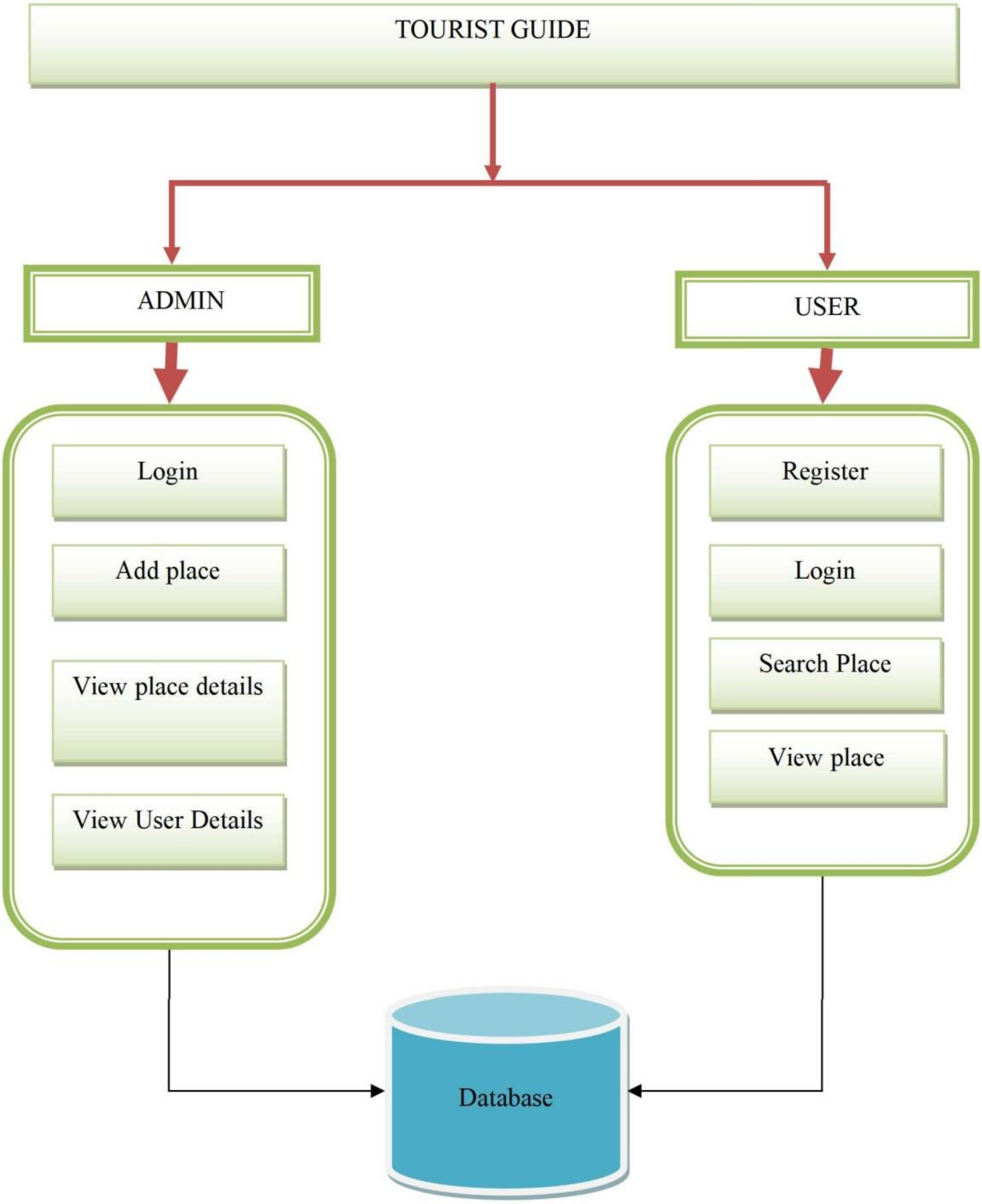


Fig no: 4.1 Architectural diagrams

4.2.1 I/O Form design

Form Design: Input Form

Form Name: User Registration

Description: The user can use this form and make a registration on creating account

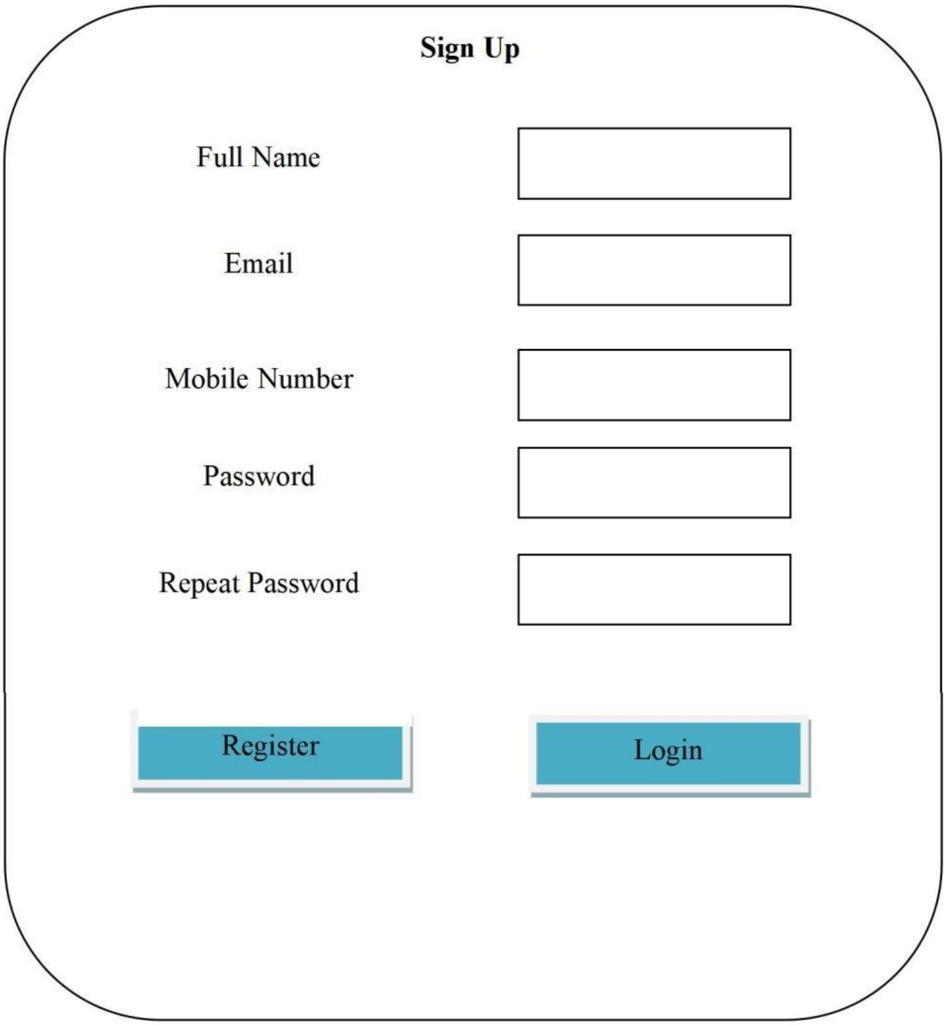
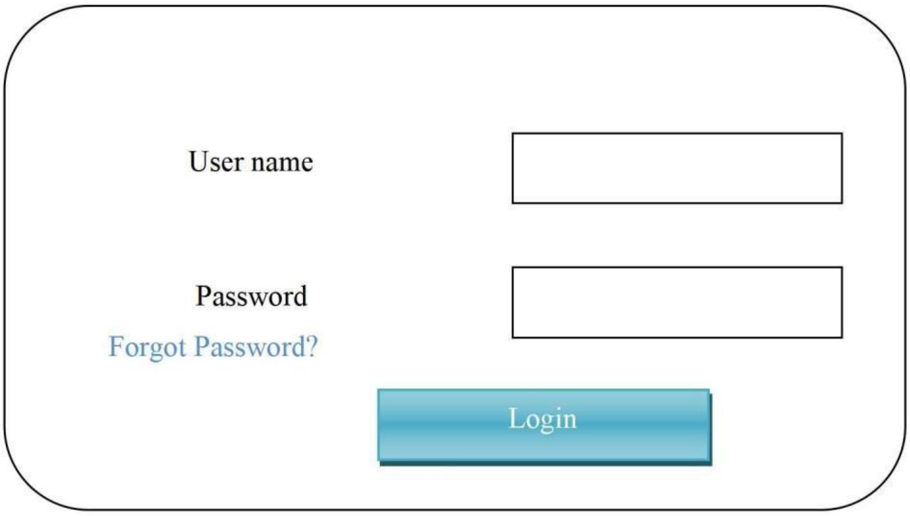


Fig 4.2 Input form of user registration

Form Design: Output Form

Form Name: User Login

Description: The user can make login with the created E-mail and password



User name

Password

Fig 4.4 Output Form

Form Design: Input Form

Form Name: Add place

Description: The place information can be added with the system

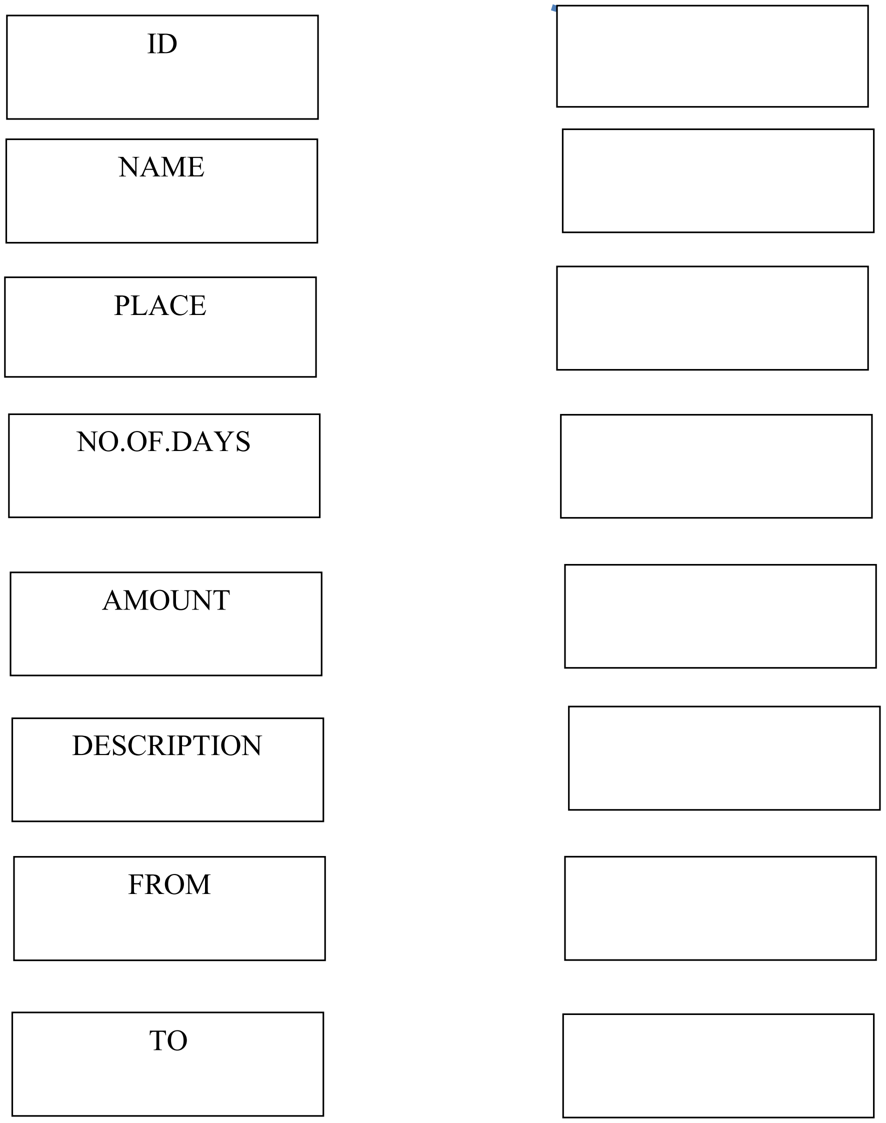


Fig 4.5 Input form of Interaction

4.3 TABLES

Tourist table

User details

Field	Type	Null	Default
id	int(100)	Yes	NULL
name	varchar(100)	Yes	NULL
contact	varchar(100)	Yes	NULL
email	varchar(100)	Yes	NULL
address	varchar(100)	Yes	NULL
username	varchar(100)	Yes	NULL
password	varchar(100)	Yes	NULL
status	varchar(100)	Yes	NULL
report	varchar(100)	Yes	NULL

4.4 ENTITY RELATIONSHIP DIAGRAM

An entity–relationship model (ER model for short) describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between entities (instances of those entity types). In software engineering, an ER model is commonly formed to represent things a business needs to remember in order to perform business processes. Consequently, the ER model becomes an abstract data model that defines data or information structure which can be implemented in a database, typically a relational database

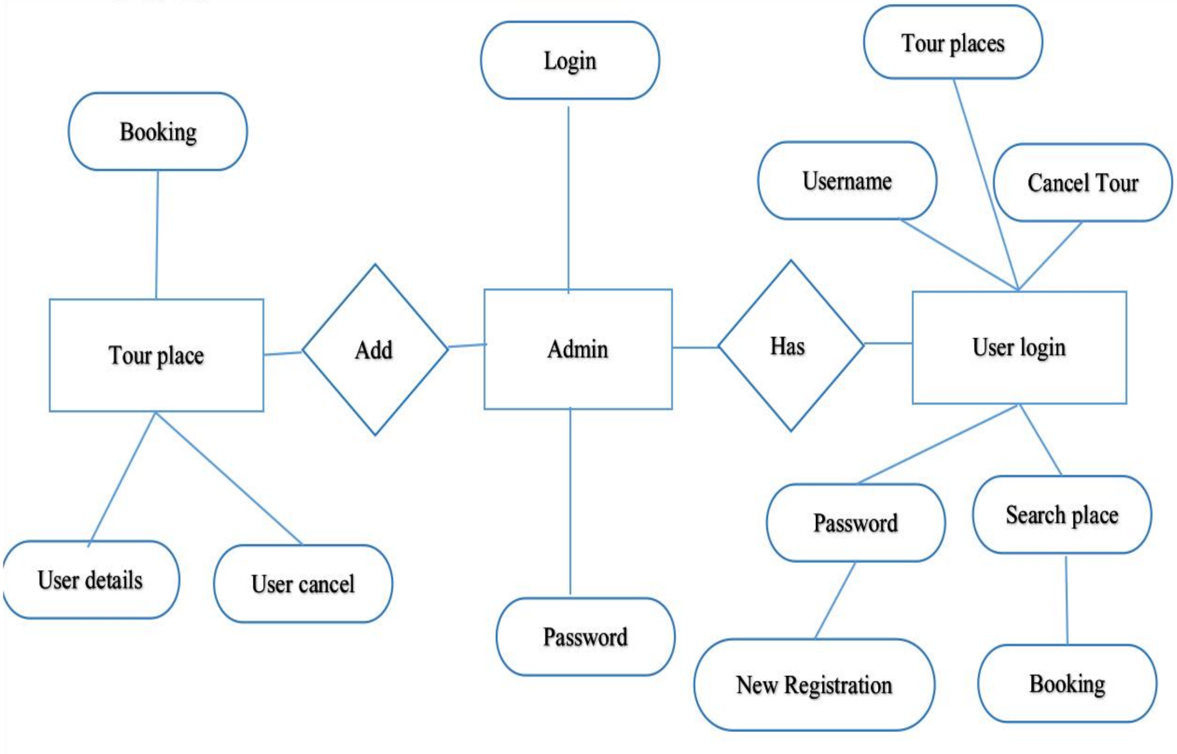


Fig no: 4.6 ER diagram

4.5 DATA DICTIONARY

Field	**Type**	Description	Example
Name	varchar(100)	name of the user	Akash
Email	varchar(100)	email id of the user	akash@gmail.com
Confirm password	varchar(100)	confirm password of the user	123
Mobile number	varchar(100)	mobile number of the uset	7485961330

Repeat password	varchar(100)	password of the user	123
id	varchar(100)	id of the expense	13
place	varchar(100)	place of the city	trichy
spot	varchar(100)	spotname	butterfly park

4.6 NORMALIZATION

Database Normalization is a technique of organizing the data in the database. Normalization is a systematic approach of decomposing tables to eliminate data redundancy (repetition) and undesirable characteristics like Insertion, Update and Deletion Anomalies. It is a multi-step process that puts data into tabular form, removing duplicated data from the relation tables.

Normalization is used for mainly two purposes,

- Eliminating redundant (useless) data.
- Ensuring data dependencies make sense i.e data is logically stored.

Normalization Rule

Normalization rules are divided into the following normal forms:

1. First Normal Form
2. Second Normal Form
3. Third Normal Form

Before Normalization

Place ID	Name	spot
1	trichy	malaikottai
2	trichy	butterfly
1	trichy	malaikottai
4	trichy	malaikottai

First Normal Form (3NF)

For a table to be in the First Normal Form, it should follow the following 4 rules:

1. It should only have single (atomic) valued attributes/columns.
2. Values stored in a column should be of the same domain
3. All the columns in a table should have unique names.
4. And the order in which data is stored, does not matter.

Place ID	Name	spot
1	trichy	malaikottai
2	trichy	butterfly
4	trichy	malaikottai

Second Normal Form (2NF)

For a table to be in the Second Normal Form,

1. It should be in the First Normal form.
2. And, it should not have Partial Dependency.

Register Details

Name	Mobile	Address	User Name	Password
mani	9009638527	Trichy	mani	1345
madhu	8527419632	Mathurai	madhu	1234
ravi	7534218695	Chennai	ravi	1236

Name	Mobile
mani	9009638527
madhu	8527419632
ravi	7534218695

Third Normal Form (3NF)

Name	Search
Sasi	trichy
ragu	butterfly park

CHAPTER V

SYSTEM DEVELOPMENT

5.1 FUNCTIONAL DOCUMENTATION

ADMIN

Admin Login

Admin will make a login with the created system. Admin will have a default username and password system.

Add place

The admin will add the place for the tourist places with its information system. In this module the tourist information will be noted

Add District

The admin in this module will add the details of the districts available in this system, where the tourist places can be verified

USER

User Register and login

The user needs to create a new account to make a registration. The personnel details of the user will be added. After that the login will be made with the system.

Search place

The user will search the details of the place with the district information’s, the admin added details will be retrieved in this modules

View Places

The user in this module view the places details with the booking process if the wish to do so in this system

5.2 SPECIAL FEATURES OF LANGUAGE / UTILITY

5.2.1 Front end (ASP.NET)

Asp.NET

The .NET platform

.NET is a developer platform made up of tools, programming languages, and libraries for building many different types of applications. The base platform provides components that apply to all different types of apps. Additional frameworks, such as ASP.NET, extend .NET with components for building specific types of apps.

Here are some things included in the .NET platform:

- The C#, F#, and Visual Basic programming languages
- Base libraries for working with strings, dates, files/IO, and more
- Editors and tools for Windows, Linux, mac OS, and Docker

ASP.NET extends .NET

ASP.NET extends the .NET platform with tools and libraries specifically for building web apps .These are some things that ASP.NET adds to the .NET platform. Base framework for processing web requests in C# or F#. Web-page tempting syntax, known as Razor, for building dynamic web pages using C#. Libraries for common web patterns, such as Model View Controller (MVC). Authentication system that includes libraries, a database, and template pages for handling logins, including multi-factor authentication and external authentication with Google, X, and more. Editor extensions to provide syntax highlighting, code completion, and other functionality specifically for developing web pages.

Back-end code

When using ASP.NET your back-end code, such as business logic and data access, is written using C#, F#, or Visual Basic. Because ASP.NET extends .NET, you can use the large ecosystem of packages and libraries available to all .NET developers. You can also author your own libraries that are shared between any applications written on the .NET platform.

SQL SERVER

- SQL Server 2005 is a relational database management system (RDBMS) developed by Microsoft. It was released in 2005 and was a significant upgrade from its predecessor, SQL Server 2000. SQL Server 2005 introduced several new features and enhancements, including:

- Integration Services (SSIS): This is a platform for building high-performance data integration and workflow solutions. SSIS provides tools for extracting, transforming, and loading (ETL) data.

- Analysis Services (SSAS): SSAS enables online analytical processing (OLAP) and data mining functionalities. It allows users to create multidimensional data models for analysis and reporting.

- Reporting Services (SSRS): SSRS is a server-based reporting platform that enables the creation, management, and delivery of interactive and paginated reports.

- Service Broker: This is a messaging framework built into SQL Server for building scalable and reliable applications. It enables asynchronous, bidirectional communication between database components.

- Database Mirroring: SQL Server 2005 introduced the database mirroring feature, which provides high availability and disaster recovery capabilities by maintaining redundant copies of a database.

- Database Snapshots: This feature allows you to create read-only, point-in-time views of a database. It's useful for reporting purposes and for reverting databases to a previous state.

- XML Support: SQL Server 2005 enhanced its support for XML, allowing XML data to be stored, queried, and manipulated directly within the database.

- Common Language Runtime (CLR) Integration: SQL Server 2005 allows developers to write database objects such as stored procedures, functions, and triggers using .NET languages like C# and VB.NET, leveraging the CLR.

- Native XML Web Services: SQL Server 2005 can expose stored procedures as web services, enabling them to be called remotely over HTTP.

- Partitioning: SQL Server 2005 introduced partitioning support, allowing large tables and indexes to be partitioned into smaller, more manageable chunks for improved performance and manageability.

- SQL Server 2005 reached its end of extended support on April 12, 2016. It's recommended to upgrade to a newer version of SQL Server to ensure continued support and access to the latest features and security updates.

5.4 PSEUDO CODE

USER REGISTRATION DETAILS

```
Step 1: Start
Step 2: Enter the Name
Step 3: Enter the contact
Step 4: Enter the password
Step 5: Enter the conform password
Step 6: Stop
```

ADD INTERACTIONS

```
Step 1: Start
Step 2: Enter the place
Step 3: Enter the spot
Step 4: Enter the amount
Step 5: Enter the id
Step 6: Enter the description
Step 7: Stop
```

CHAPTER VI

TESTING

6.1 TESTING

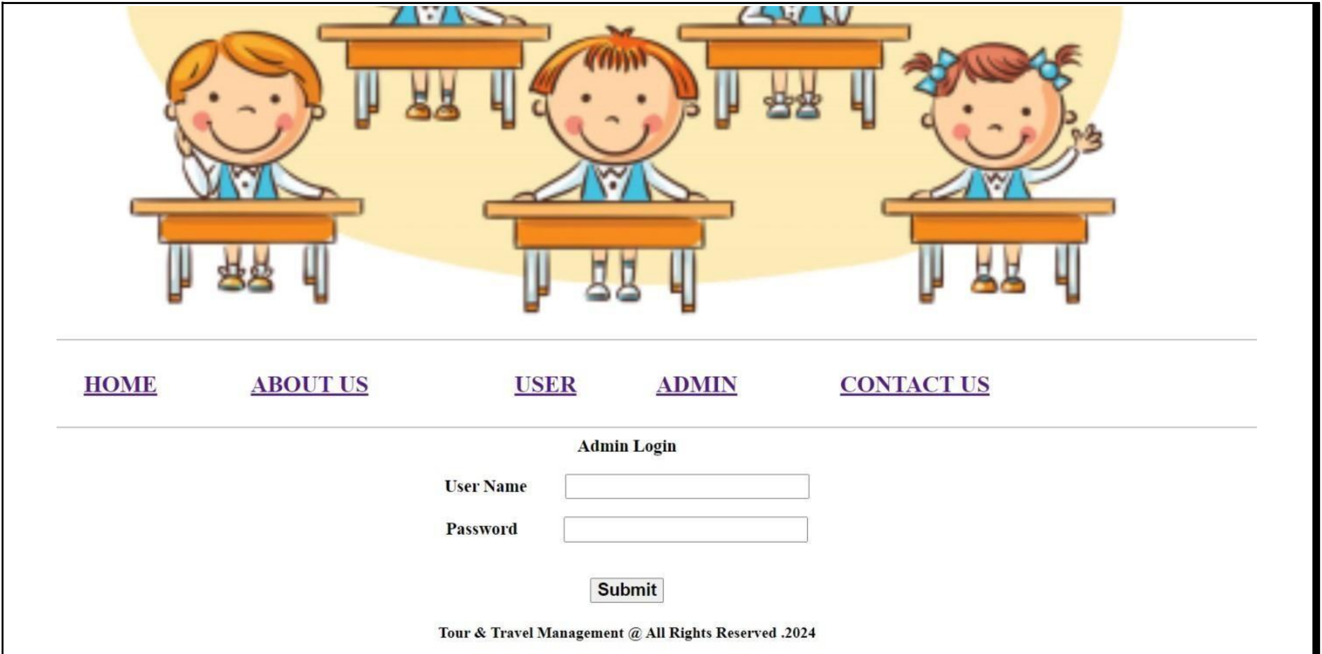
Testing is a series of different tests that whose primary purpose is to fully exercise the computer based system. Although each test has a different purpose, all work should verify that all system element have been properly integrated and performed allocated function. Testing is the process of checking whether the developed system works according to the actual requirement and objectives of the system. The philosophy behind testing is to find the errors. A good test is one that has a high probability of finding an undiscovered error. A successful test is one that uncovers the undiscovered error. Test cases are devised with this purpose in mind. A test case is a set of data that the system will process as an input.

6.2 TYPES OF TESTING DONE

- Unit Testing
- Integration Testing
- System Testing

6.2.1 UNIT TESTING

Unit testing refers testing of all the individual programs. This is sometimes called as program testing. This test should be carried out during programming stage in order to find the errors in coding and logic for each program in each module. Unit test focuses verification effort on the smallest unit of software design module. In this project, the user must fill each field otherwise the user to enter values.



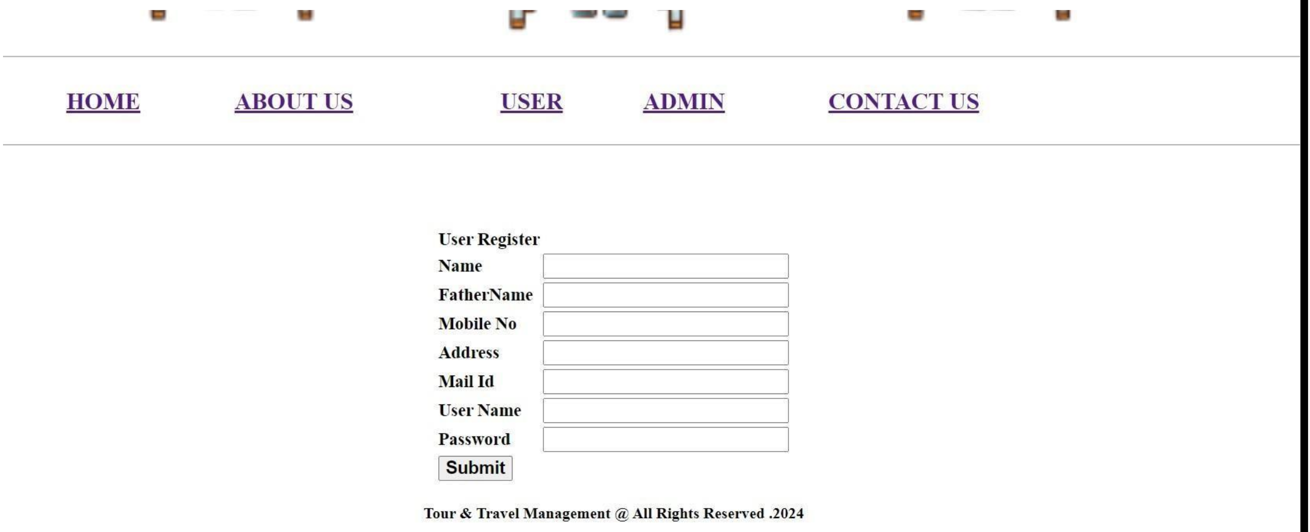
Admin Login

Tour & Travel Management @ All Rights Reserved .2024

Fig 6.1 unit testing

6.2.2 Validation Testing

Valid and invalid data should be created and the program should be made to process this data to catch errors. When the user of each module wants to enter into the page by the login page using the use rid and password .If the user gives the wrong password or use rid then the information is provided to the user like “you must enter user id and password”. Here the inputs given by the user are validated. That is password validation, format of date are correct, textbox validation. Changes that need to be done after result of this testing.



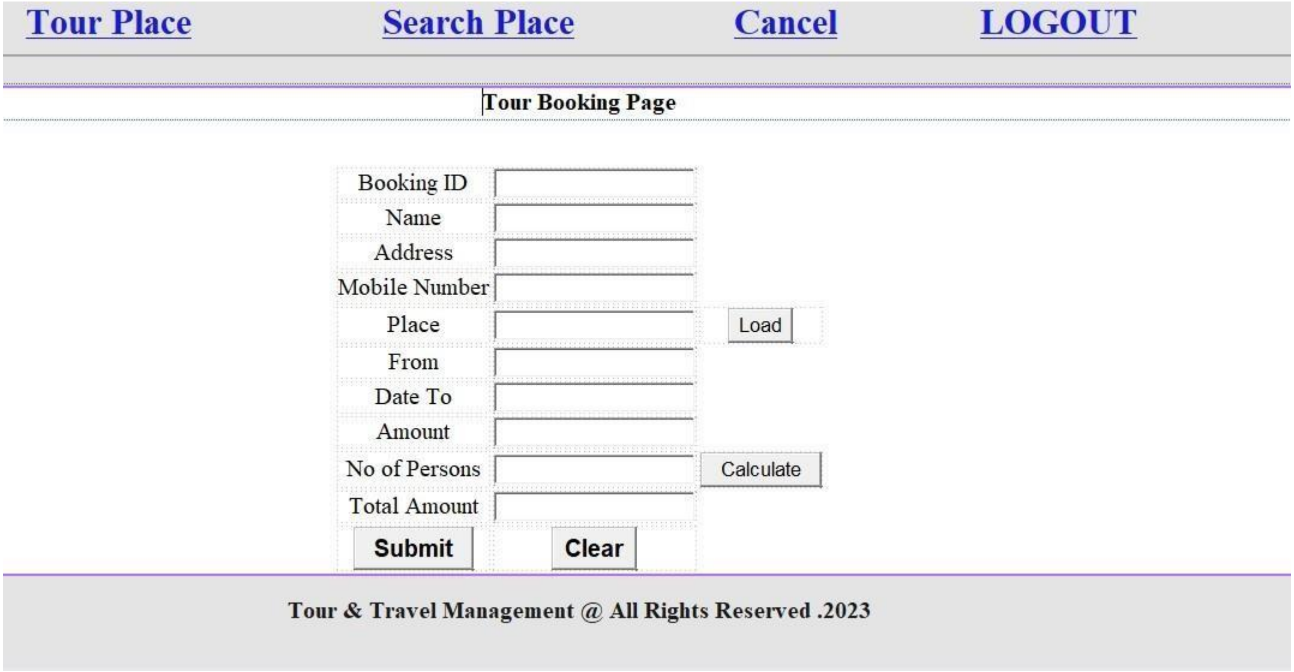
User Register

Tour & Travel Management @ All Rights Reserved .2024

Fig 6.2 Validation Testing

6.2.3 System Testing

System testing is used to test the entire system (Integration of all the modules). It also tests to find the discrepancies between the system and the original objective, current specification and system documentation. The entire system is checked to correct deviation to achieve correctness.



Tour & Travel Management @ All Rights Reserved .2023

Fig 6.3 System Testing

6.2.4 Integration Testing

Testing is which modules are combined and tested as a group. Modules are typically code modules, individual applications, source and destination applications on a network, etc. Integration Testing follows unit testing and precedes system testing. Testing after the product is code complete. Betas are often widely distributed or even distributed to the public at large in hopes that they will buy the final product when it is release.

CHAPTER VIII

CONCLUSION

7.1 SUMMARY OF THE PROJECT

This application combines website and Internet functionality. The tour management website application helps users organize their vacations by providing thorough information about tourist attractions, including descriptions, images, and maps. The system offers a variety of features and services, including personalized packages, distance between current and desired locations, Google maps, and online ticket buying, among others. Real-time data is used to help the system achieve its core aim. This website was successfully established, and this application was used to store all of the travel admin, place booking, creation management, and tour details into the database. It was thoroughly tested, and all errors were identified and corrected. The system's performance was also found to be satisfactory throughout testing. All of the required output has been created. As a result, this technology makes it simple to automate all consumption functions. This application will be beneficial if it is implemented in a limited number of instances

7.2 FUTURE POSSIBILITIES

The project can be improved further so that the website performs in a more appealing and useful manner than it does now. It is concluded that the application is functional and meets the requirements. The application has been thoroughly tested, and any errors have been identified and corrected. It also serves as a means of exchanging files with valuable resources.

8. BIBLIOGRAPHY

1. ASP.NET MVC 5 (WROX)" by Jon Galloway and Brad Wilson

2.ASP.NET Core and Angular 2" by Valerio de Sanctis

Web reference:

1. http://www.mjret.in/V2I3/M11-2-3-7-2015.pdf

2. https://www.irjet.net/archives/V5/i3/IRJET-V5I3501.pdf

3. https://erdplus.com/edit-diagram/e56adaca-1494-4687-af14-9b2eb3f1be3b

9. APPENDIX

A) Sample code

```
<%@ Master Language="VB" CodeFile="MasterPage.master.vb" Inherits="MasterPage" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
    <asp:ContentPlaceHolder id="head" runat="server">
    </asp:ContentPlaceHolder>
    <style type="text/css">
        .auto-style1
            {width:
            70%;
            height: 498px;
        }
    </style
       >
</head>
<body>
    <form id="form1" runat="server">
    <div>
      <center>
    <img src="01.jpg" alt="top" class="auto-style1"/>
    <hr>

    <table width="75%">
    <tr>
    <td> <h2><a href="Home.aspx">HOME</a></h2> </td>
     <td><h2> <a href="About.aspx">ABOUT US</a> </h2></td>
     <td><h2> <a href="UserLogin.aspx">USER</a> </h2></td>
     <td> <h2><a href="AdminLogin.aspx">ADMIN</a></h2></td>
     <td><h2><a href="Contact.aspx">CONTACT US</a> </h2> </td>
    </tr>

    </table>

           <hr />

        <asp:ContentPlaceHolder   id="ContentPlaceHolder1" runat="server">

        </asp:ContentPlaceHolder>
        <h4> Tour & Travel Management @ All Rights Reserved .2024 </h4>
        </center>
    </div>
```

```
        </form>
    </body>
        </html>
<%@ Master Language="VB" CodeFile="MasterPage.master.vb" Inherits="MasterPage" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
    <asp:ContentPlaceHolder id="head" runat="server">
    </asp:ContentPlaceHolder>
</head>
<body>
    <form id="form1" runat="server">
    <div>
     <center>
    <img src="001.png" alt="top" height="5%" width="75%"/>
    <hr>

    <table width="75%">
    <tr>
    <td> <h2><a href="Home.aspx">HOME</a></h2> </td>
     <td><h2> <a href="About.aspx">ABOUT US</a> </h2></td>
     <td><h2> <a href="UserLogin.aspx">USER</a> </h2></td>
     <td> <h2><a href="AdminLogin.aspx">ADMIN</a></h2></td>
     <td><h2><a href="Contact.aspx">CONTACT US</a> </h2> </td>
    </tr>

    </table>

          <hr />

        <asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">

        </asp:ContentPlaceHolder>
        <h4>  Tour & Travel Management @ All Rights Reserved .2023 </h4>
        </center>
    </div>
    </form>
</body>
</html>

Imports System.Data.SqlClient

Partial Class UserLogin
    Inherits System.Web.UI.Page
    Dim db As New TravelDB

    Protected Sub Button1_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Button1.Click
        Dim dr As SqlDataReader
        Dim cmdstr As String
        Dim cmd As SqlCommand
        cmdstr = " select uname,pwd from UserRegisterTable where uname ='" & Me.TextBox1.Text & "' and pwd='" &
Me.TextBox2.Text & "'"
        db.DataBaseConnect()
        cmd = New SqlCommand(cmdstr, db.con)
        dr = cmd.ExecuteReader
        If dr.Read Then
            If dr.HasRows Then
```

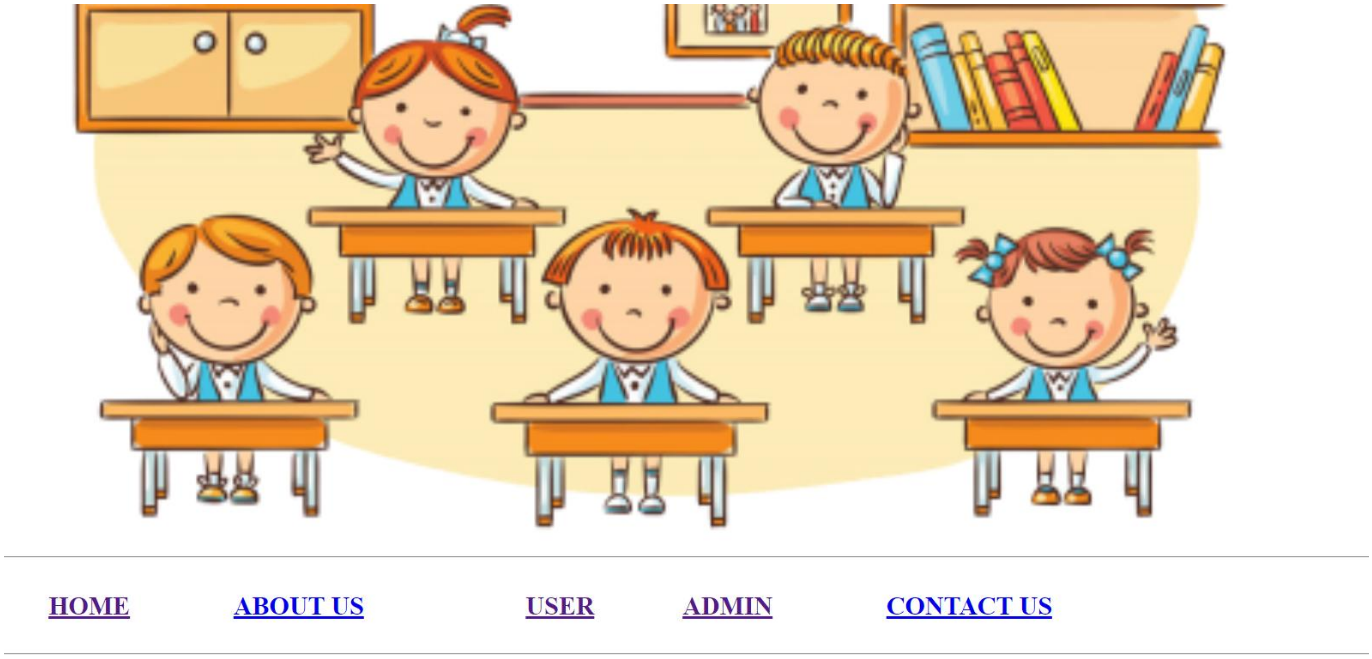
```
                If Me.TextBox1.Text = dr(0) And Me.TextBox2.Text = dr(1) Then
                    MsgBox("Login Successfull ")
                    Response.Redirect("user/userhome.aspx")
                End If
            Else
                MsgBox("Please Enter Correct UserName and Password")
            End If
        End If
    End Sub

    Protected Sub LinkButton1_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles LinkButton1.Click
        Response.Redirect("Userreg.aspx")
    End Sub
End Class

Imports System.Data.SqlClient
Partial Class Userreg
    Inherits System.Web.UI.Page
    Dim db As New TravelDB
    Sub clear()
        Me.TextBox1.Text = ""
        Me.TextBox2.Text = ""
        Me.TextBox3.Text = ""
        Me.TextBox4.Text = ""
        Me.TextBox5.Text = ""
        Me.TextBox6.Text = ""
        Me.TextBox7.Text = ""
    End Sub


    Protected Sub Button1_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Button1.Click
        Dim cmdstr As String
        Dim cmd As SqlCommand
        cmdstr = "insert into UserRegisterTable values('" & Me.TextBox1.Text & "','" & Me.TextBox2.Text & "','" &
Me.TextBox3.Text & "','" & Me.TextBox4.Text & "','" & Me.TextBox5.Text & "','" & Me.TextBox6.Text & "','" &
Me.TextBox7.Text & "')"
        If Me.TextBox1.Text = "" Or Me.TextBox2.Text = "" Then
            MsgBox("Values Needed..")
        Else
            db.DataBaseConnect()
            cmd = New SqlCommand(cmdstr, db.con)
            cmd.ExecuteNonQuery()
            MsgBox("New User Registered Successfully..")
            clear()
        End If

    End Sub
End Class


Partial Class AdminLogin
    Inherits System.Web.UI.Page

    Protected Sub Button1_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Button1.Click
        If Me.TextBox1.Text = "admin" And Me.TextBox2.Text = "admin" Then
            MsgBox("Login Successfull ..")
            Response.Redirect("admin/adminhome.aspx")
        Else
            MsgBox("Try Again ..")
        End If
    End Sub
End Class
```

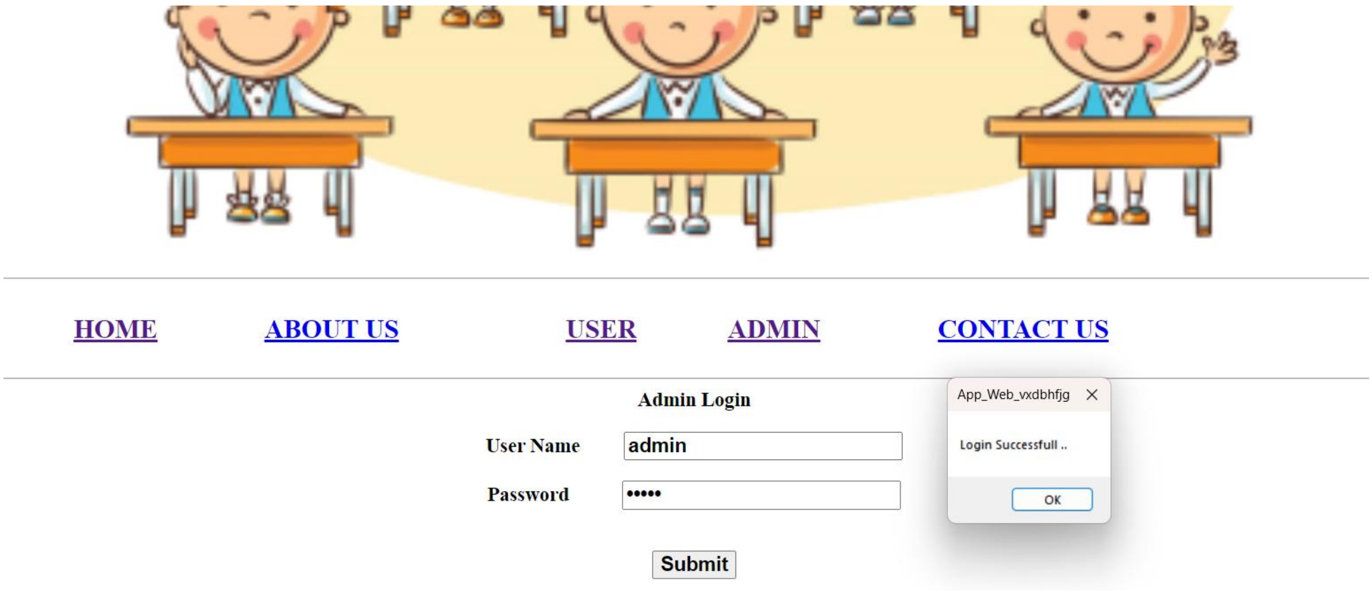
```
Imports System.Data.SqlClient
Imports System.Data
Partial Class admin_view
   Inherits System.Web.UI.Page
   Dim db As New TravelDB
   Public adap As New SqlDataAdapter
   Public ds As New DataSet
   Dim str As String
   Dim dr As SqlDataReader
   Dim cmd As SqlCommand
   Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
      db.DataBaseConnect()
      Dim cmd As New SqlCommand("select BookingID,    CName, Addres  ,MobileNumber,  NoofPersons    ,Place,
           DateFrom,     DateTo,  Amount,TotalAmount from bookingtable where flag='1' ", db.con)
      Dim da As New SqlDataAdapter(cmd)
      Dim ds As New DataSet()
      da.Fill(ds)
      GridView1.DataSource = ds
      GridView1.DataBind()
   End Sub
End Class

Imports System.Data.SqlClient
Imports System.Data
Partial Class admin_view
   Inherits System.Web.UI.Page
   Dim db As New TravelDB
   Public adap As New SqlDataAdapter
   Public ds As New DataSet
   Dim str As String
   Dim dr As SqlDataReader
   Dim cmd As SqlCommand
   Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
      db.DataBaseConnect()
      Dim cmd As New SqlCommand("select BookingID,    CName, Addres  ,MobileNumber,  NoofPersons    ,Place,
           DateFrom,     DateTo,  Amount,TotalAmount from bookingtable where flag='0' ", db.con)
      Dim da As New SqlDataAdapter(cmd)
      Dim ds As New DataSet()
      da.Fill(ds)
      GridView1.DataSource = ds
      GridView1.DataBind()
   End Sub
End Class

Imports System.Data.SqlClient
Partial Class admin_tour
   Inherits System.Web.UI.Page
   Dim db As New TravelDB
   Sub clear()
      Me.TextBox1.Text = ""
      Me.TextBox2.Text = ""
      Me.TextBox3.Text = ""
      Me.TextBox4.Text = ""
      Me.TextBox5.Text = ""
      Me.TextBox6.Text = ""
      Me.TextBox7.Text = ""
      Me.TextBox8.Text = ""
   End Sub

   Protected Sub Button1_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Button1.Click
      Dim cmdstr As String
```

```
        Dim cmd As SqlCommand
        cmdstr = "insert into tourplace values('" & Me.TextBox1.Text & "','" & Me.TextBox2.Text & "','" & Me.TextBox3.Text &
"','" & Me.TextBox4.Text & "','" & Me.TextBox5.Text & "','" & Me.TextBox6.Text & "','" & Me.TextBox7.Text & "','" &
Me.TextBox8.Text & "')"
        If Me.TextBox1.Text = "" Or Me.TextBox2.Text = "" Or Me.TextBox3.Text = "" Or Me.TextBox4.Text = "" Or
Me.TextBox5.Text = "" Or Me.TextBox6.Text = "" Or Me.TextBox7.Text = "" Or Me.TextBox8.Text = "" Then
            MsgBox("Values Needed..")
        Else
            db.DataBaseConnect()
            cmd = New SqlCommand(cmdstr, db.con)
            cmd.ExecuteNonQuery()
            MsgBox("Registered Successfully..")
            clear()
        End If
    End Sub


    Protected Sub Calendar1_SelectionChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Calendar1.SelectionChanged
        Me.TextBox4.Text = Me.Calendar1.SelectedDate.ToShortDateString
        Me.TextBox5.Text = Me.Calendar1.SelectedDate.ToShortDateString
    End Sub
End Class

Imports System.Data.SqlClient
Imports System.Data
Partial Class admin_view
    Inherits System.Web.UI.Page
    Dim db As New TravelDB
    Public adap As New SqlDataAdapter
    Public ds As New DataSet
    Dim str As String
    Dim dr As SqlDataReader
    Dim cmd As SqlCommand
    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
        db.DataBaseConnect()
        Dim cmd As New SqlCommand("select * from tourplace", db.con)
        Dim da As New SqlDataAdapter(cmd)
        Dim ds As New DataSet()
        da.Fill(ds)
        GridView1.DataSource = ds
        GridView1.DataBind()
    End Sub
End Class

Imports Microsoft.VisualBasic
Imports System.Data.SqlClient
Public Class TravelDB
    Public con As SqlConnection
    Public Sub DataBaseConnect()
        Dim constr As String
        constr = "Data Source=AJAYESRN;Initial Catalog=TravelDB;Integrated Security=True"
        con = New SqlConnection(constr)
        con.Open()
    End Sub
End Class

create database TravelDB
use TravelDB
create Table UserRegisterTable(usname varchar(25),faname varchar(25),mobileno varchar(25),addr varchar(25),mail
varchar(25),uname varchar(25),pwd varchar(25))
select *  from UserRegisterTable
```

```
Create Table tourplace(id varchar(25),tname varchar(25),place varchar(25),fromm varchar(25),too varchar(25),noday
varchar(25),amd varchar(25),Descript varchar(120))
select *  from tourplace

create Table bookingtable
(BookingID        varchar(25),
CName varchar(25),
Addres varchar(45),
MobileNumber varchar(15),
NoofPersons       varchar(5),
Place varchar(40),
DateFrom varchar(25),
DateTo  varchar(25),
Amount  varchar(25),
TotalAmount varchar(25),flag varchar(3))


select * from bookingtable
```

SCREENSHOTS





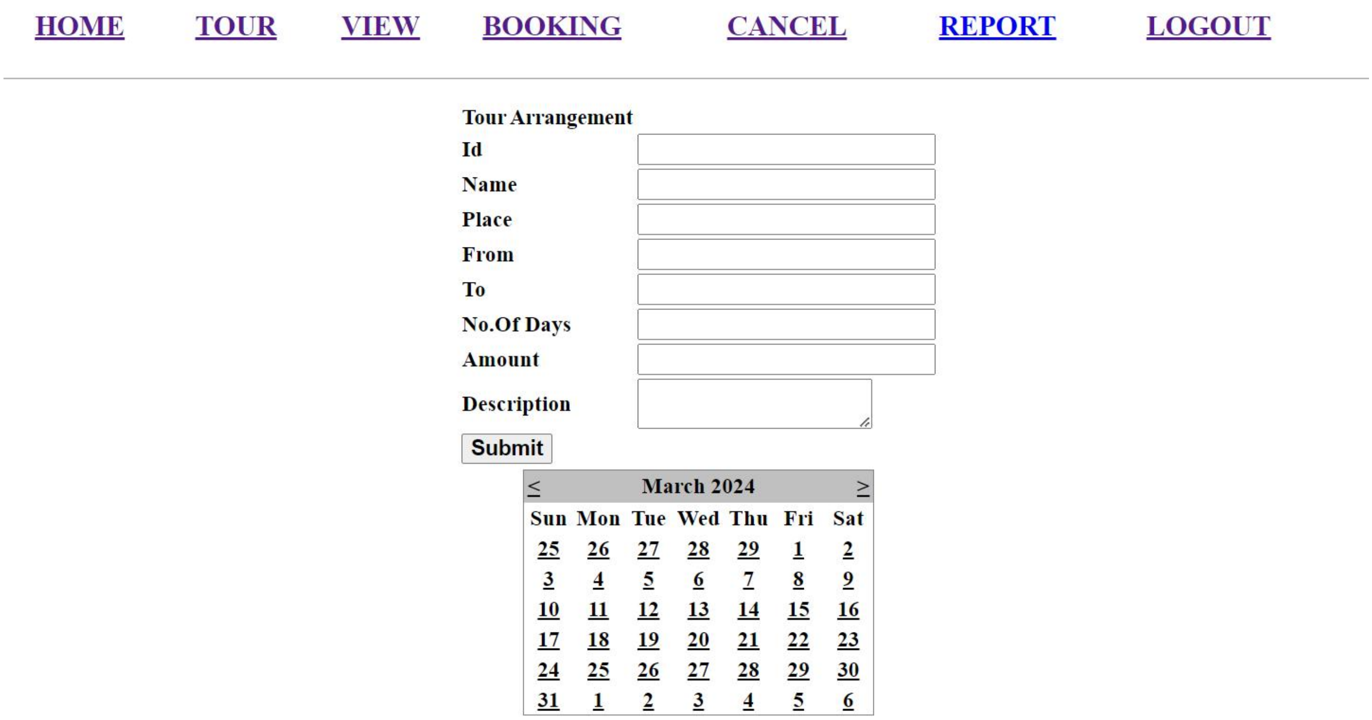
Admin Login

User Name admin

Password •••••

App_Web_vxdbhfjg

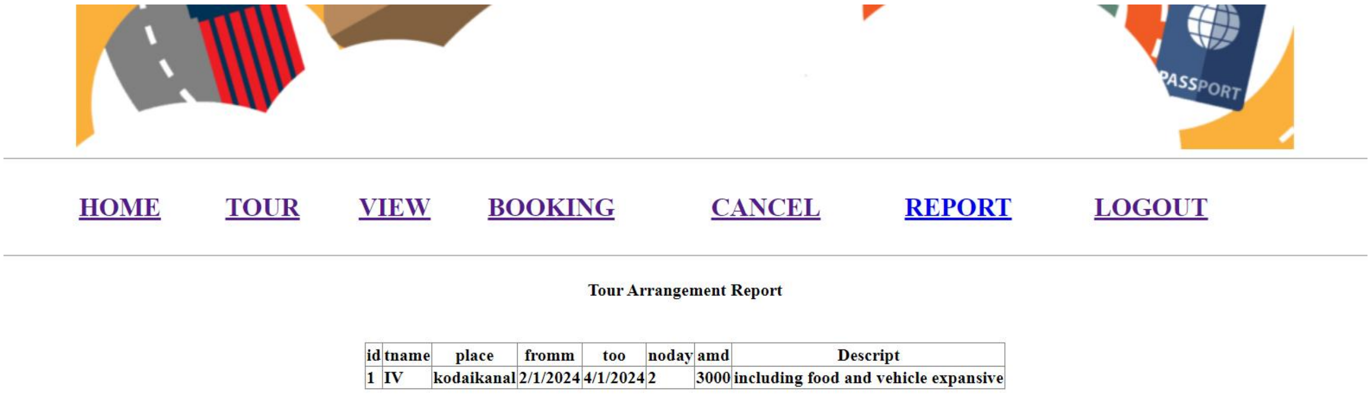
Login Successfull ..



Tour Arrangement

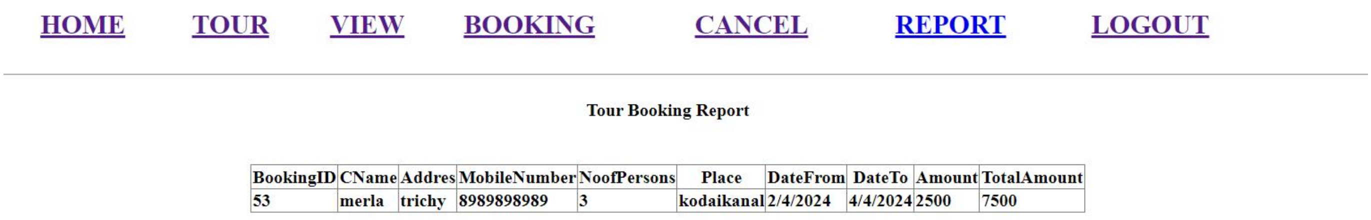
Id	
Name	
Place	
From	
To	
No.Of Days	
Amount	
Description	

<			March 2024			>
Sun	Mon	Tue	Wed	Thu	Fri	Sat
25	26	27	28	29	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6



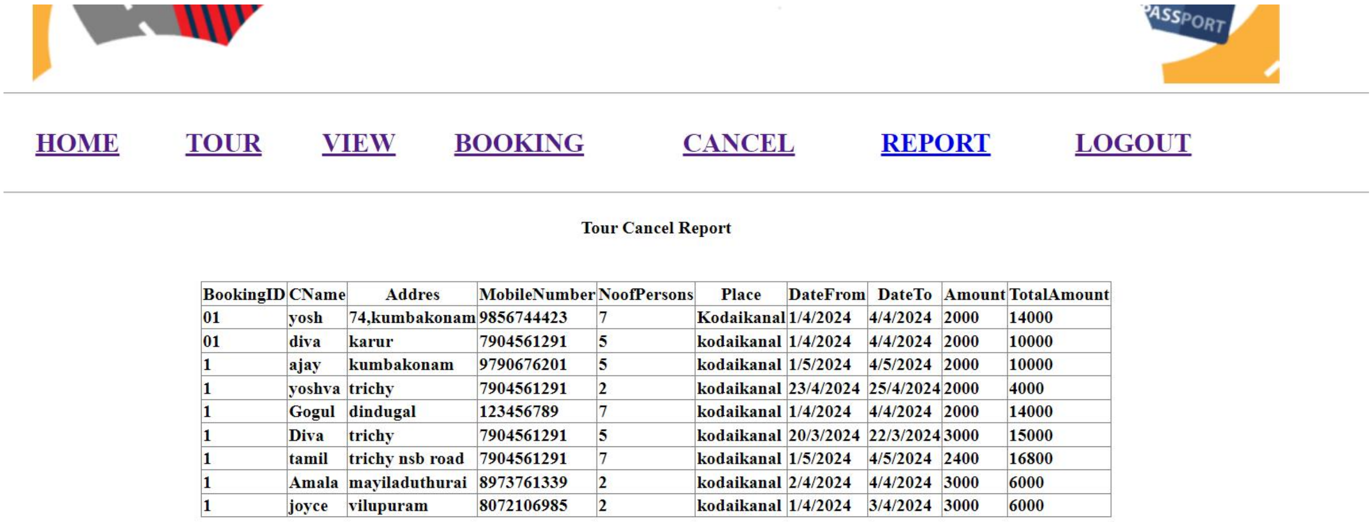
Tour Arrangement Report

id	tname	place	fromm	too	noday	amd	Descript
1	IV	kodaikanal	2/1/2024	4/1/2024	2	3000	including food and vehicle expansive



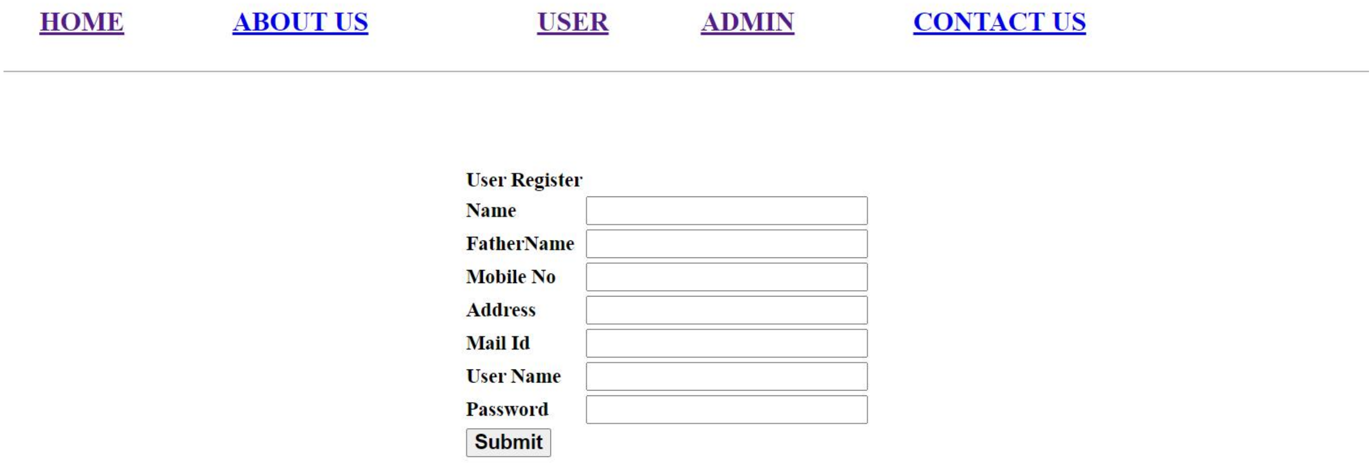
Tour Booking Report

BookingID	CName	Addres	MobileNumber	NoofPersons	Place	DateFrom	DateTo	Amount	TotalAmount
53	merla	trichy	8989898989	3	kodaikanal	2/4/2024	4/4/2024	2500	7500



Tour Cancel Report

BookingID	CName	Addres	MobileNumber	NoofPersons	Place	DateFrom	DateTo	Amount	TotalAmount
01	yosh	74,kumbakonam	9856744423	7	Kodaikanal	1/4/2024	4/4/2024	2000	14000
01	diva	karur	7904561291	5	kodaikanal	1/4/2024	4/4/2024	2000	10000
1	ajay	kumbakonam	9790676201	5	kodaikanal	1/5/2024	4/5/2024	2000	10000
1	yoshva	trichy	7904561291	2	kodaikanal	23/4/2024	25/4/2024	2000	4000
1	Gogul	dindugal	123456789	7	kodaikanal	1/4/2024	4/4/2024	2000	14000
1	Diva	trichy	7904561291	5	kodaikanal	20/3/2024	22/3/2024	3000	15000
1	tamil	trichy nsb road	7904561291	7	kodaikanal	1/5/2024	4/5/2024	2400	16800
1	Amala	mayiladuthurai	8973761339	2	kodaikanal	2/4/2024	4/4/2024	3000	6000
1	joyce	vilupuram	8072106985	2	kodaikanal	1/4/2024	3/4/2024	3000	6000



User Register

Name

FatherName

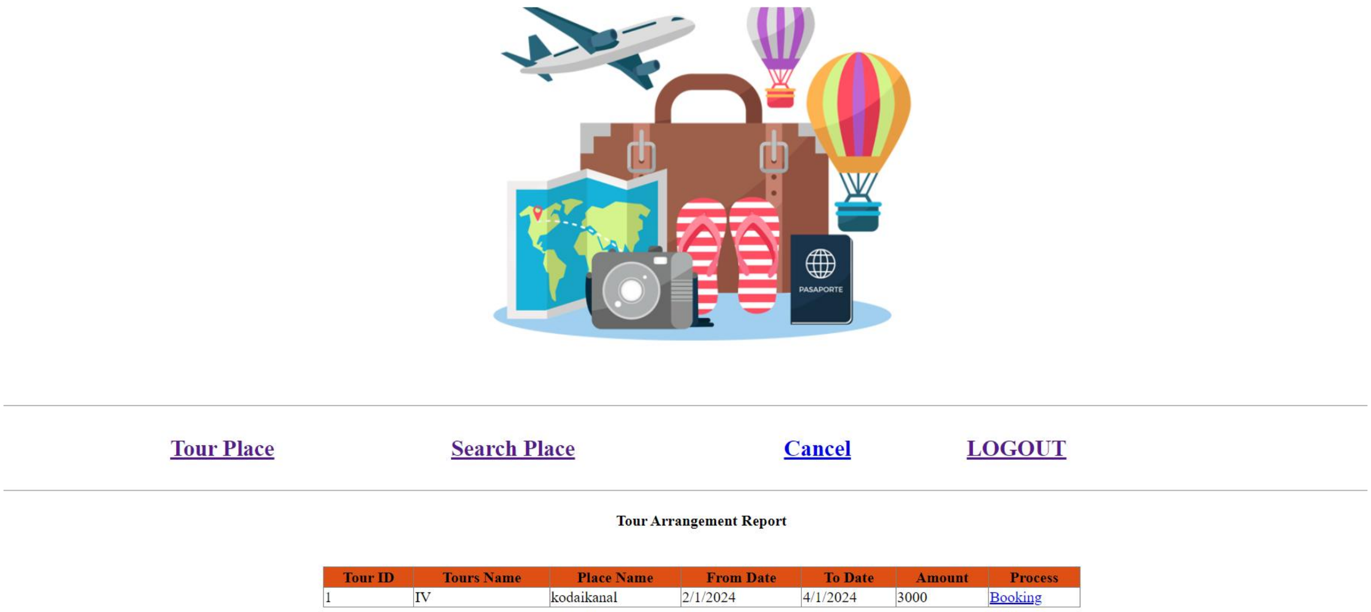
Mobile No

Address

Mail Id

User Name

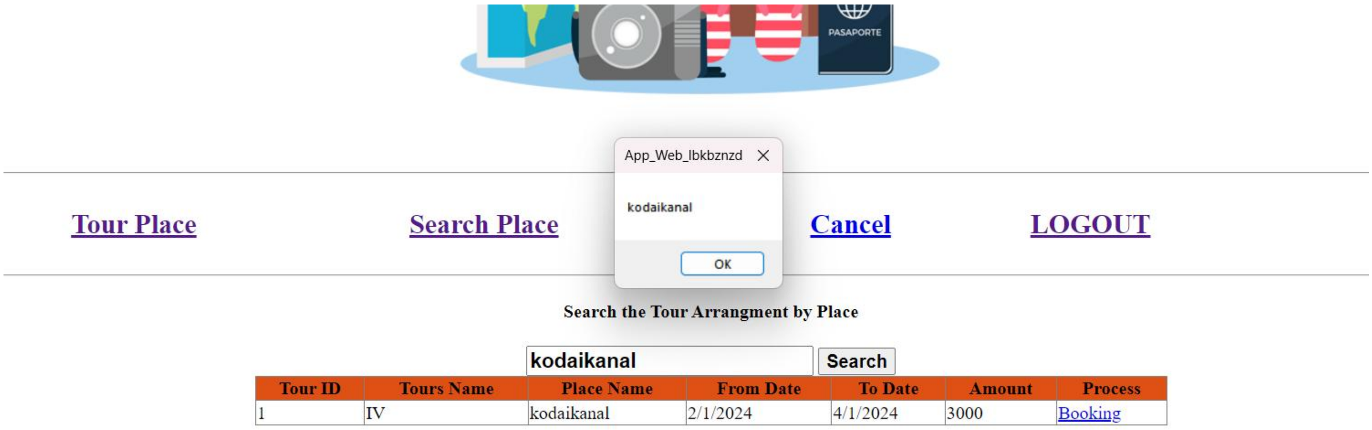
Password



Tour Place Search Place Cancel LOGOUT

Tour Arrangement Report

Tour ID	Tours Name	Place Name	From Date	To Date	Amount	Process
1	IV	kodaikanal	2/1/2024	4/1/2024	3000	Booking



Tour Place Search Place Cancel LOGOUT

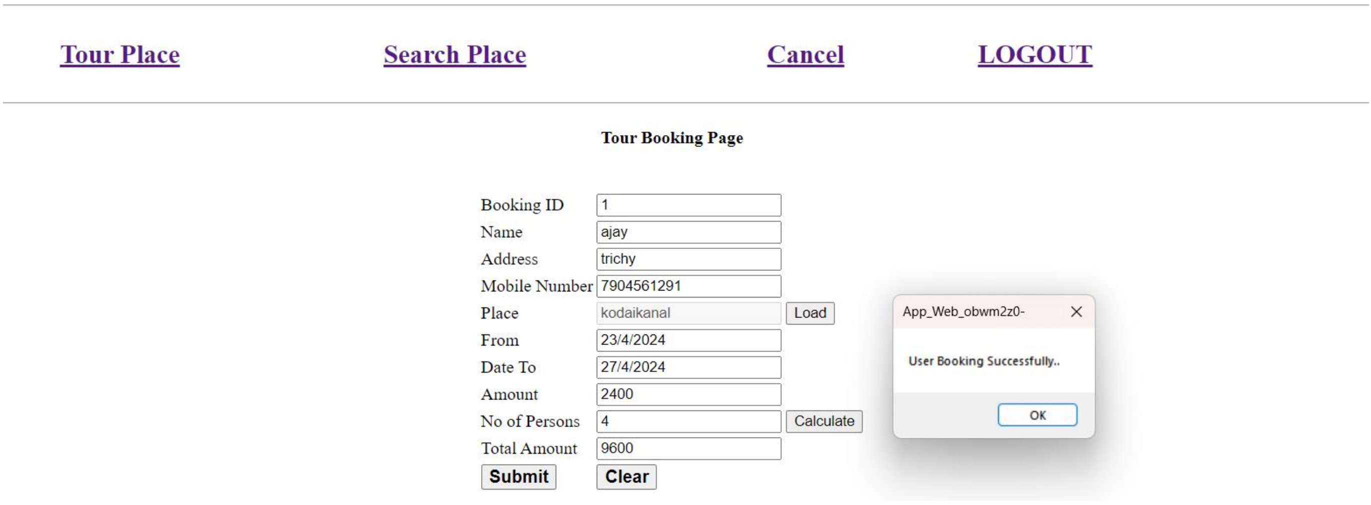
App_Web_lbkbznzd

kodaikanal

Search the Tour Arrangment by Place

kodaikanal

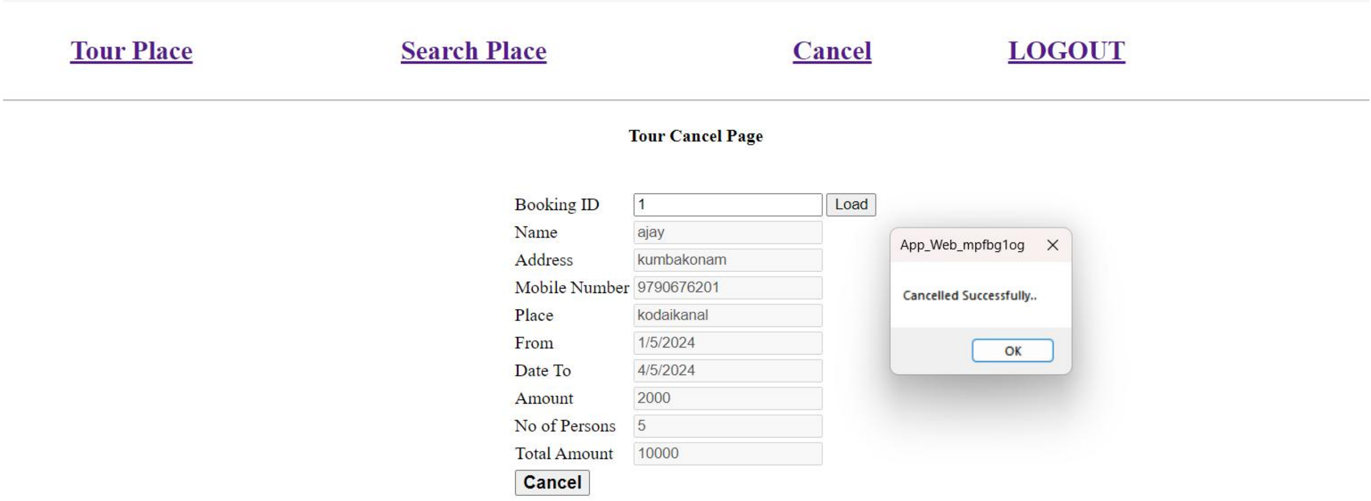
Tour ID	Tours Name	Place Name	From Date	To Date	Amount	Process
1	IV	kodaikanal	2/1/2024	4/1/2024	3000	Booking



Tour Booking Page

Booking ID	1
Name	ajay
Address	trichy
Mobile Number	7904561291
Place	kodaikanal
From	23/4/2024
Date To	27/4/2024
Amount	2400
No of Persons	4
Total Amount	9600

User Booking Successfully..



Tour Cancel Page

Booking ID	1
Name	ajay
Address	kumbakonam
Mobile Number	9790676201
Place	kodaikanal
From	1/5/2024
Date To	4/5/2024
Amount	2000
No of Persons	5
Total Amount	10000

Cancelled Successfully..