



# CHAPTER-1

## **INTRODUCTION**

Travel and Tourism Management System is a web-based application. The main purpose of Travel and Tourism management system is to provide a convenient way for a customer to book hotels, flight, train and bus for tour purposes. The objective of this project is to develop a system that automates the processes and activities of a travel agency. In this project. We will make an easier task of searching places and for booking train, flight or bus. In the present system a customer has to approach various agencies to find details of places and to book tickets. This often requires a lot of time and effort. We provide approach skills to critically examine how a tourist visits and its ability to operate in an appropriate way when dealing with the consequences of tourism, locally, regionally, and nationally including visitor security and ecological influences. It is tedious for a customer to plan a particular journey and have it executed properly. The project „Travel and Tourism Management System“ is developed to replace the currently existing system, which helps in keeping records of the customer details.

Tourism has turned out to be an economic booster contributing to the economic development of many countries over the last few decades. People see holidays as a necessity, and not as luxury in the present scenario. Tourism calls for coordination and cooperation between travel agents, tour operators, and tourists. Tourism has a few major elements – destinations, attractions, sites, accommodation, and all ancillary services.

It involves the management of multitude of activities such as studying tour destination, planning the tour, making travel arrangements and providing accommodation. It also involves marketing efforts to attract tourists to travel to particular destinations.

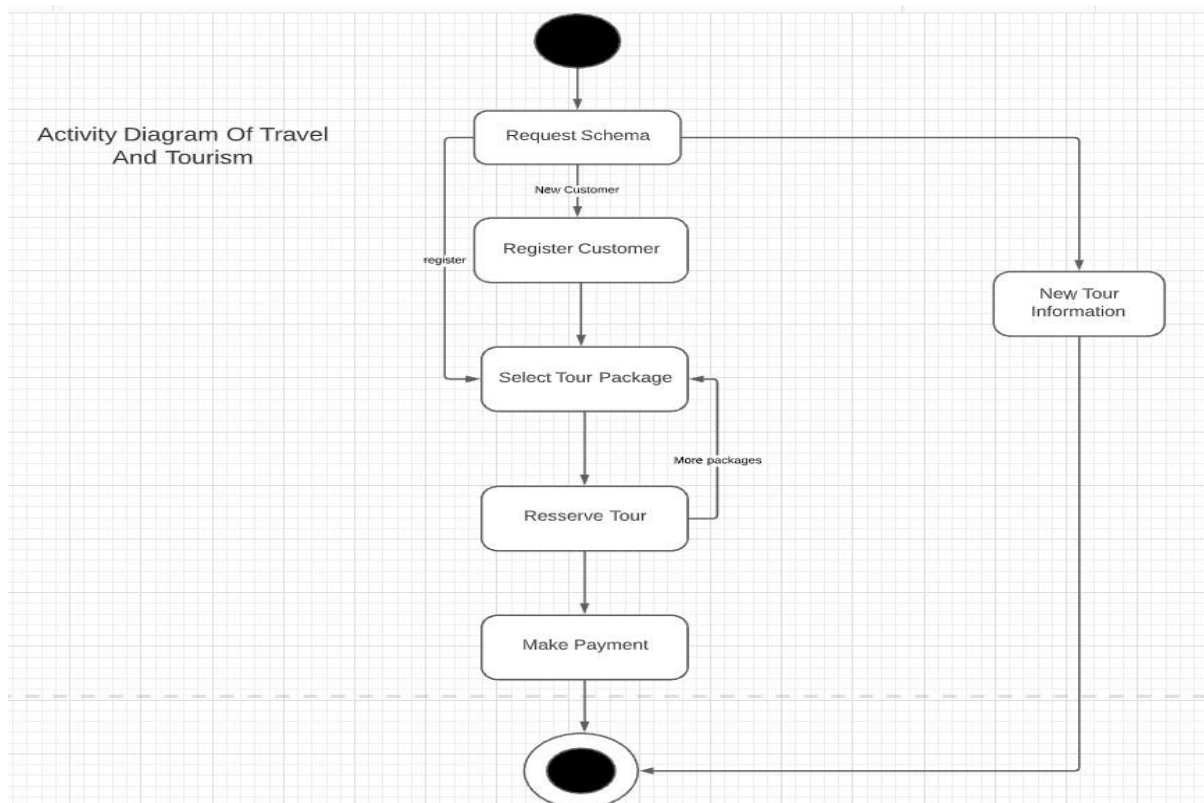
Travelling is going from the place of residence or work to another distant or a neighbouring place by any means of transport. Routine commutation can be termed as travelling.

Tourism is travelling with an objective. All tourism necessarily include travel but all travel does not necessarily include tourism. We can say, travelling is a subset of tourism.

## 1.1 Problem Statement

1. The aim of the problem statement was to design a model.
2. The model must be user-friendly.
3. It should not allow viewing others data.
4. The data privileges should be easily viewed.

## 1.2 System Architecture



**Fig 1.2.1 System Architecture**



## CHAPTER-2

## LITERATURE SURVEY / EXISTING SYSTEM

In the existing system, each task is carried out manually and processing is also a tedious job. In previous system travellers were maintaining time table details manually in pen and paper, which was time taking and costly. The travellers are able to achieve its need in time and also the results may not accurate. Because of the manual maintenance there are number of difficulties and drawbacks exist in the system.

### Drawbacks of the Existing System:

- Increased transaction leads to increased source document and hence maintenance becomes difficult.
- If any admin, user enter wrongly made then the maintenance becomes very difficult.
- If any user forgets his registration number, then it is very difficult to maintain it by admin.

### EXISTING SYSTEM:



**Fig 2.1 Existing System**



## CHAPTER-3

# SOFTWARE REQUIREMENT ANALYSIS

## 3.1 Functional Requirements

- Creation of new record this function creates a record for a new user.
- Deletion of record this function is used to delete the existing record of any user.
- Updating in record this function updates the information in a Data base of any user.
- Display of data in record This Function displays the destination of the user.
- The system asks for user's information such as name, password.
- The system stores user's record.
- The system stores college day to day financial status.

## 3.2 Non-Functional Requirements

- **System performance** All system must perform all the mentioned operations at a reasonable speed so that the user experience is not affected.
- **Security** The personal information of the users along with other sensitive information must be hidden from other users and should only be visible to the system admin. Hence this data must remain in an encrypted form to increase system security.
- **Usability** as the system is implemented through a universal app the user can log into the system both from his mobile and desktop, anywhere.
- **Capacity** The system must have the capacity to handle large amount of user's data and should not slow down as the amount of data increases.
- **Maintainability** The system must be able to modify the stored records so that any change in the user's information that occur can be accommodated easily in the user's database.



## CHAPTER-4



## PROPOSED SYSTEM

The proposed system is designed to be more efficient than the manual system. It invokes all base tasks that are now carried out manually, such as the forms transactions and reports which is added advantage. The proposed System is completely computer-based application. Thousands of records can search and displayed without taking any significant time

### Advantages of the Proposed System:

- Gives accurate information
- Simplifies the manual work
- It minimizes the documentation related work
- Provides up to date information
- Friendly Environment by providing warning messages.
- Traveller's details can be provided ·
- Booking confirmation notification



**Fig 4.1 Proposed System**



# CHAPTER-5

## **PROCEDURE**

### **5.1 Modules of travel and tourism management**

#### **5.1.1 User Management**

- a) Login
- b) User Profile
- c) Update Information
- d) Role Based Rights

#### **5.1.2 Administrator module:**

In This module user provides administrator related functionality. This module use can add route information, bus information, train information, flight information, tour packages, travel packages, bus seat details, etc. From this module Admin can view daily, weekly and monthly report. This module is developed for admin of the website and admin can add, delete, edit and view the data related to places, travels, routes, bookings from this module

- a) Manage User Information
- b) Update Information
- c) Manage Trips
- d) Manage Transportation
- e) Manage Hotel, Bookings
- f) Hotel details

Details of the hotels in which the accommodation of the customer will be done during the tours. Details like availability of meals, station-pickup, drop facility and contact no. of the hotels are also provide to the customers on special request

- a) Registration (as user)
- b) Registration (as hotel)
- c) Search

### **5.1.3 Transportation Module**

- a) Flight
- b) Train
- c) Bus

### **5.1.4 Package Module:**

User can view different tour packages available for tourist. User can select any packages from this module he can also check the details of various travel agencies. A user can select any travel agency from this module.

### **5.1.5 Testimonial's module:**

In this testimonial module passenger can post feedback after the journey and they can share their experience. Users of this application can post their opinions, complaints and suggestions.

- a) Photos
- b) Videos

### **5.1.6 Payment Modules:**

Payment through PayPal, Remit2India etc;

### **5.1.7 Search Module:**

Search City wise hotels, Flights, Packages, Bus and Railways.

### **5.1.8 Routes Modules:**

This will display the route information of Source location and destination location. User can also check best routes for his destination. User can check the best route train tour and car route for his journey and can select any route packages from the available tour packages. From this module user can also get information related to various routes connecting sources and destinations. For each route, information such as source, destination, fare, reservation details, pick up points etc are provided.

### **5.1.9 Reservation Modules:**

This module is for passengers where passenger can reserve the seats by making payment. Using this module user can book tickets. From this module user can also book tickets or cancel previously booked tickets. The module maintains the details of all reservations made so far and allows administrator to either confirm or reject the bookings. this portal services to the administrator. Accordingly, the administrator can take various steps to act on the complaints and suggestions.

### **5.1.10 Tour Details:**

Details of different types of tours which includes tours like family tours, couple tours, general tours, date and time of departure and the fair of the tours etc are maintained. As the customer ask for the details of a particular tour, the tours and travel management system give the details of the related place where tourist wants to go, the date, time of the tour and number of seats.



# CHAPTER-6

# **HARDWARE AND SOFTWARE REQUIREMENTS**

## **Recommended Operating Systems:**

- Windows: 7 or newer
- MAC: OS X v10.7 or higher ➤ Linux: Ubuntu

## **6.1 Hardware Requirements**

- Processor: Minimum 1 GHz; Recommended 2GHz or more
- Ethernet connection (LAN) OR a wireless adapter (Wi-Fi)
- Hard Drive: Minimum 32 GB; Recommended 64 GB or more
- Memory (RAM): Minimum 1 GB; Recommended 4 GB or above
- Sound card w/speakers
- Some classes require a camera and microphone

## **6.2 SOFTWARE REQUIREMENTS**

### **6.2.1 Operating System:**

Windows7 An Operating System (OS) is an interface between a computer user and computer hardware. An operating system is a software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers. Functions of an operating system:

- Memory Management
- Processor Management
- Device Management
- File Management
- Security
- Control over system performance
- Job accounting
- Error detecting aids
- Coordination between other software and users 11

### 6.2.2 XAMPP:

XAMPP is a source platform web solution package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible. XAMPP's ease of deployment means a WAMP or LAMP stack can be installed quickly and simply on an operating system by a developer.

## 6.3 HTML & CSS

### 6.3.1 HTML

HTML is the standard markup language for creating Web pages.

HTML stands for Hyper Text Markup Language

- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content
- HTML elements are represented by tags
- HTML files are saved using .html extension
- HTML tags label pieces of content such as "heading", "paragraph", "table", and so on

Browsers do not display the HTML tags, but use them to render the content of the page.

**Structure of a HTML page:**

```
<html>
<head><title></title>
</head>
<body>
</body>
</html>
```



### 6.3.2 CSS

CSS stands for Cascading Style Sheets.

- CSS is a style sheet language used for describing the presentation of a Webpage
- CSS provides more flexibility and control in the specification of presentation.
- CSS enables multiple webpages to share formatting.
- CSS files are saved using .CSS extension.
- CSS provides Site-wide consistency

## 6.4 Scripting Languages

### 6.4.1 PHP

PHP stands for Hypertext Pre-processor.

PHP is server scripting language.

At Present we are using PHP 7

PHP 8 is in beta

PHP files are saved using .php extension

**Example PHP code for printing “Hello, World!”**

```
<?php
    Echo 'Hello, World!';
?>
```

### 6.4.2 JAVA SCRIPT

**JavaScript** is a lightweight, cross-platform and interpreted scripting language. It is well-known for the development of web pages; many non-browser environments also use it. JavaScript can be used for Client-side developments as well as Server-side developments. JavaScript contains a standard library of objects, like Array, Date, and Math, and a core set of language elements like operators, control structures, and statements.



# CHAPTER-7

# UML DIAGRAMS

UML is a way of visualizing a software program using a collection of diagrams. The notation has evolved from the work of Grady Booch, James Rumbaugh, Ivar Jacobson, and the Rational Software Corporation to be used for object-oriented design, but it has since been extended to cover a wider variety of software engineering projects. Today, UML is accepted by the Object Management Group (OMG) as the standard for modelling software development.

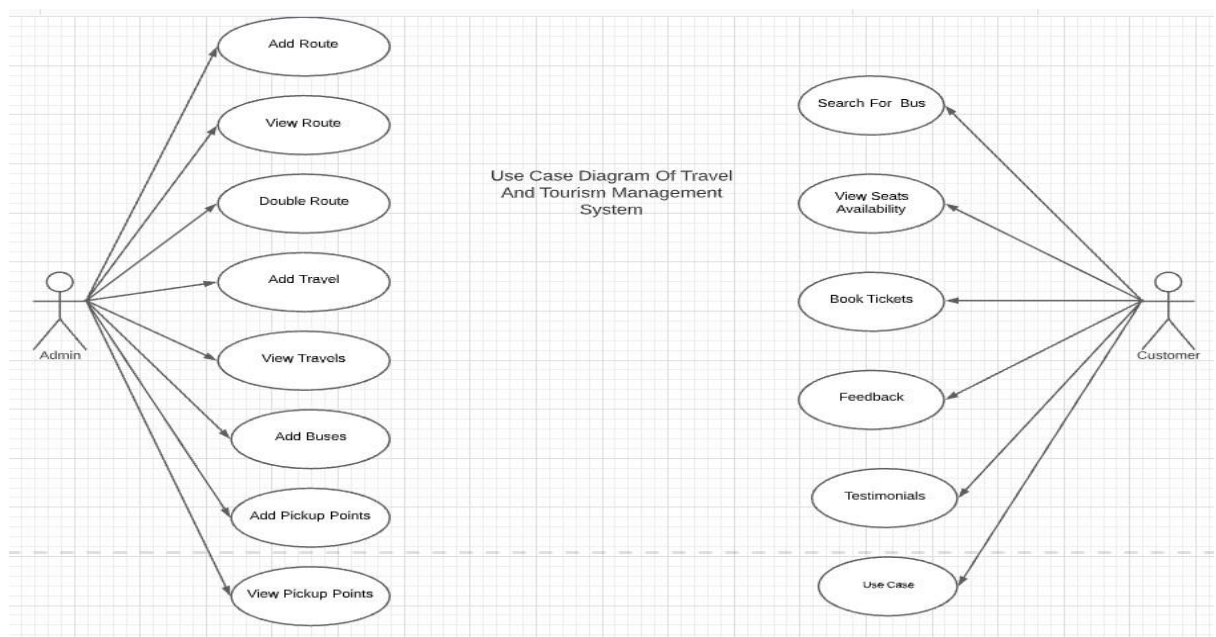
## Structural UML Diagrams

- Class diagram

## Behavioural UML Diagrams

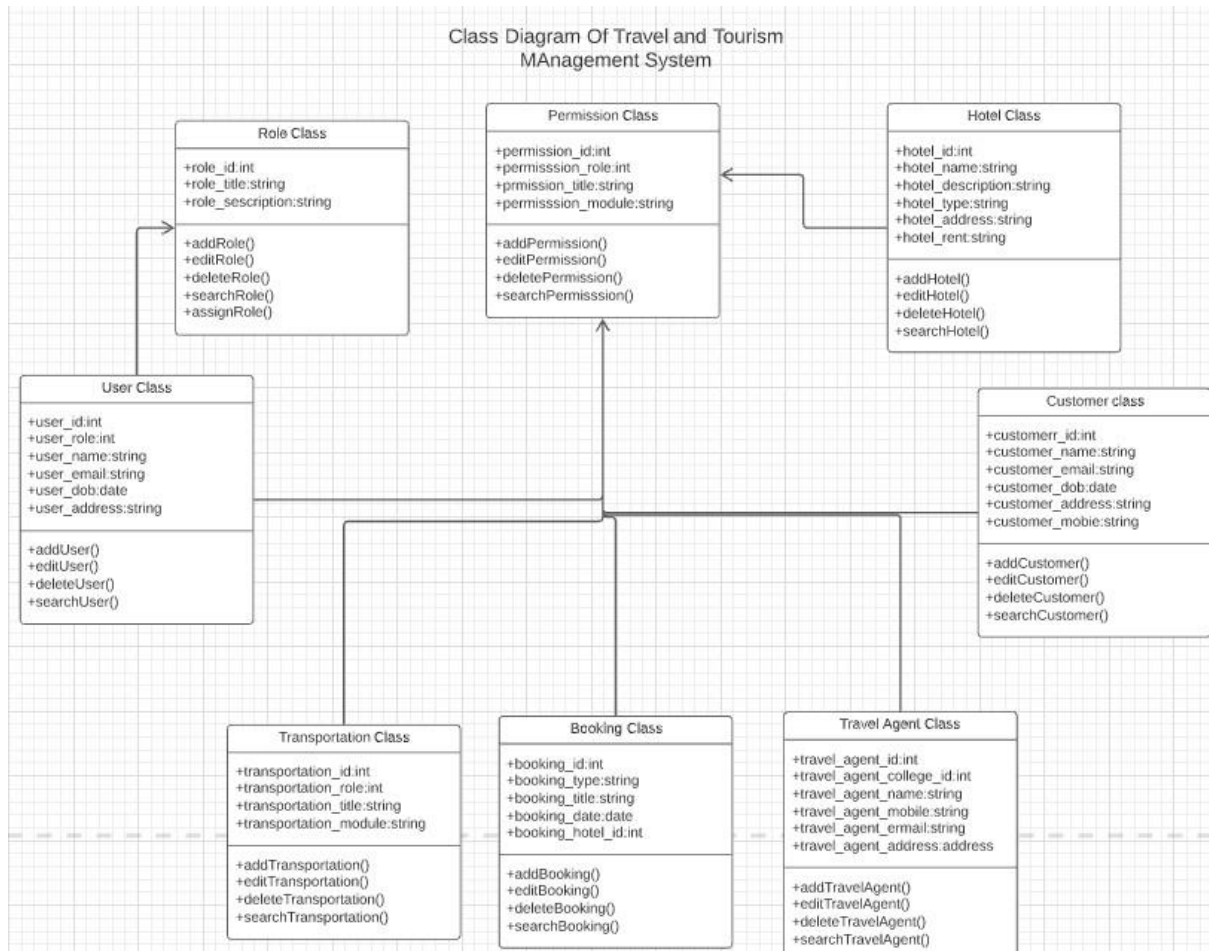
- Activity diagram
- Sequence diagram
- Use case diagram
- State diagram

## USECASE DIAGRAM:



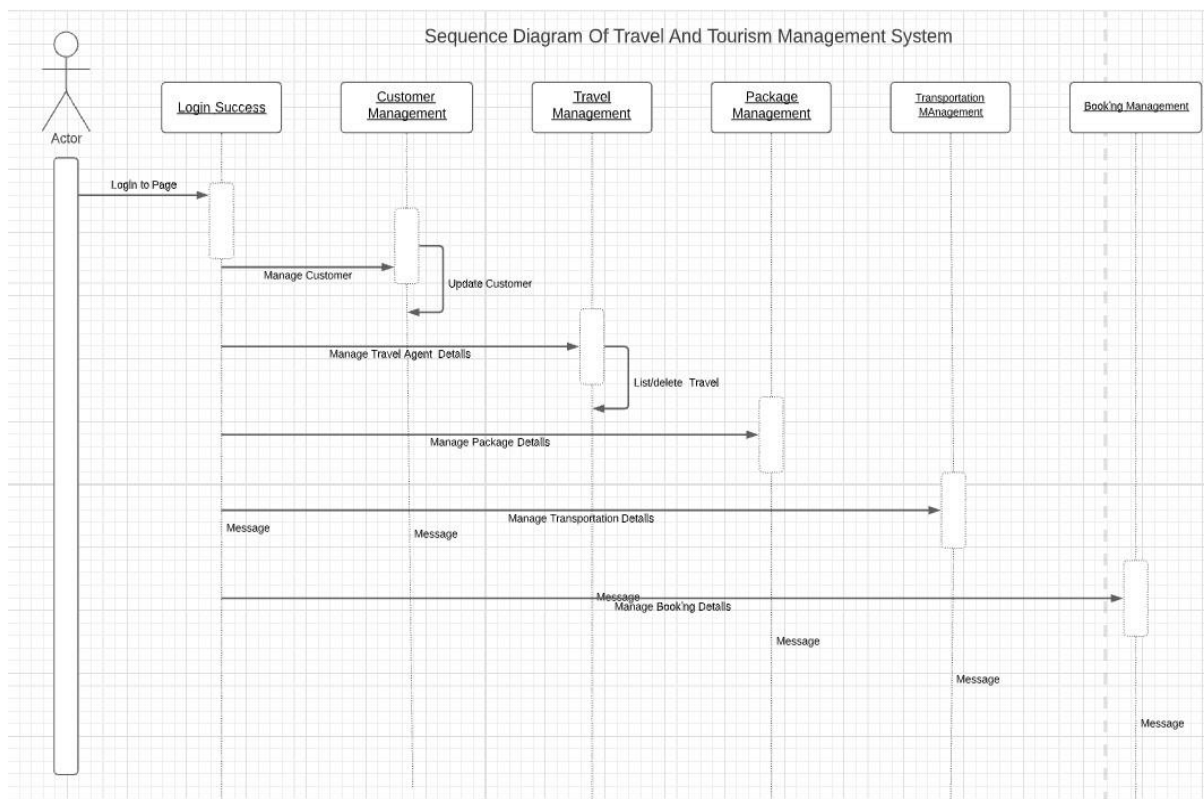
**Fig 7.1 Use case Diagram**

## CLASS DIAGRAM:



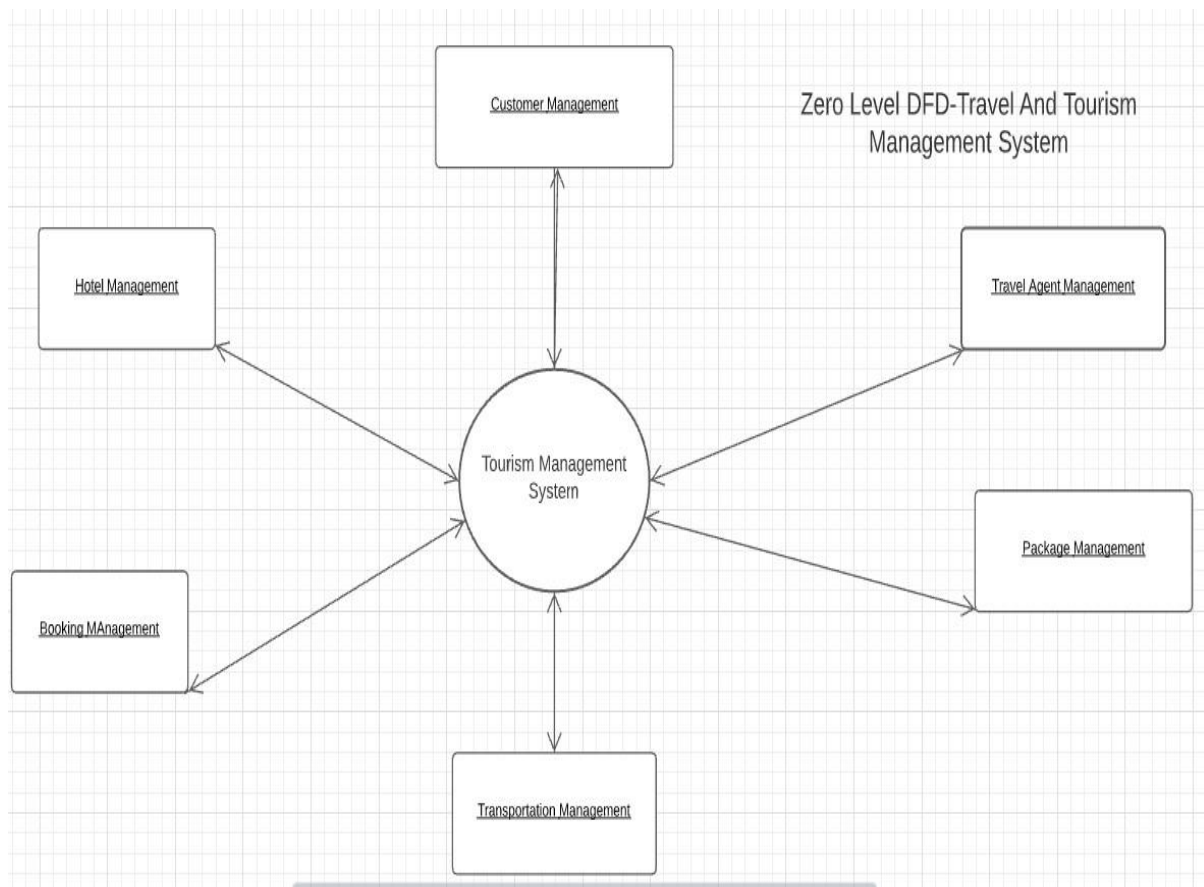
**Fig 7.2 Class diagram**

## SEQUENCE DIAGRAM:



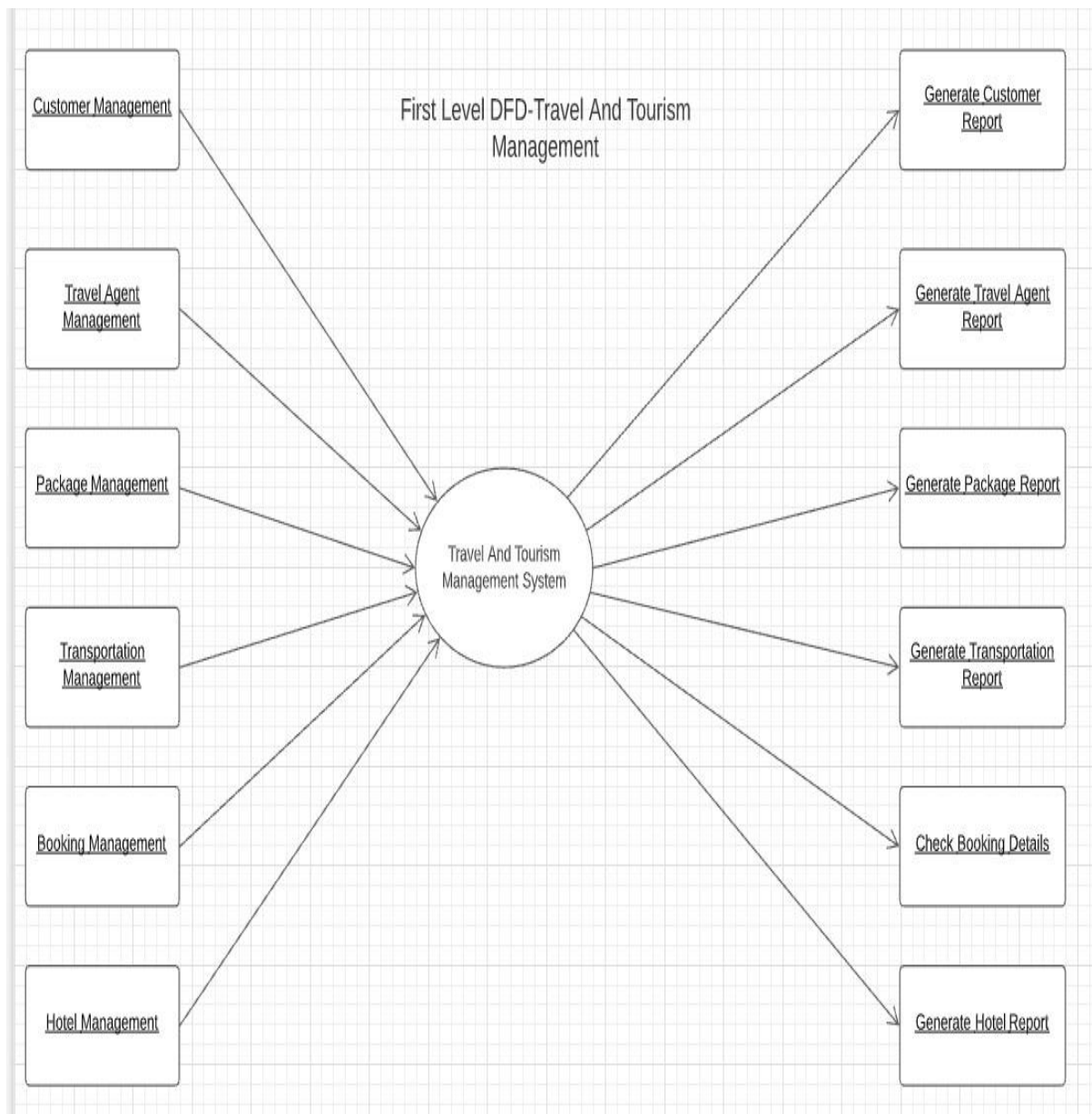
**Fig 7.3 Sequence diagram**

## Zero Level DFD:



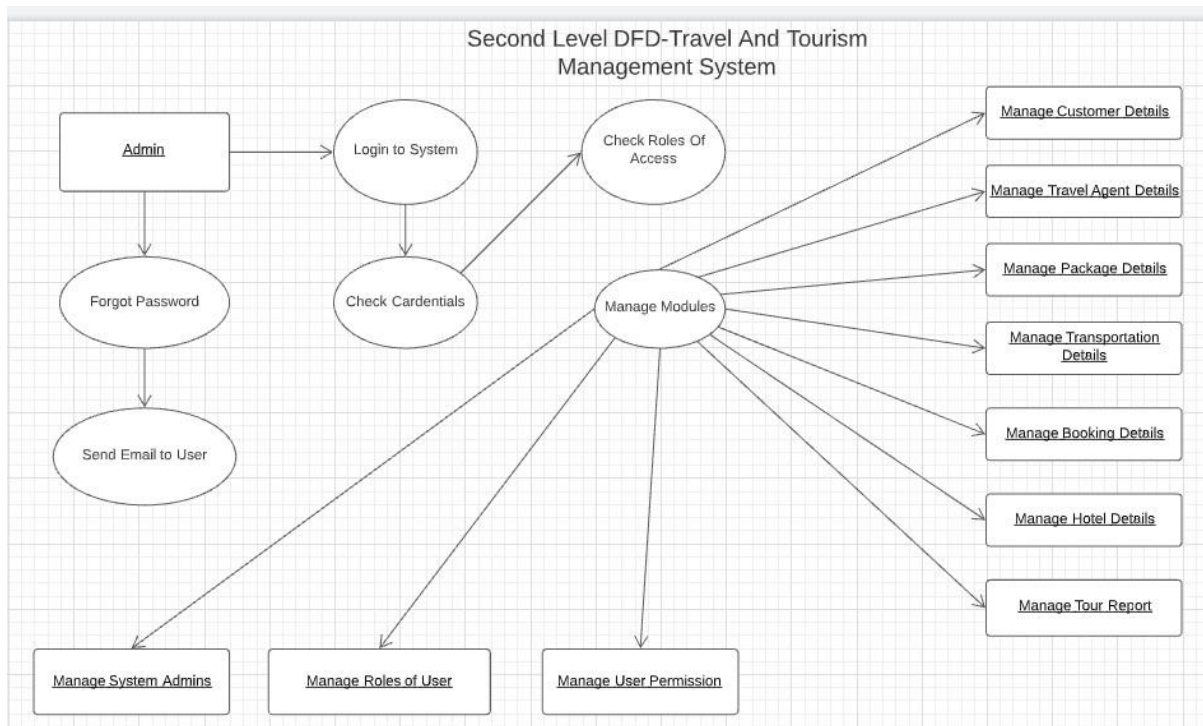
**Fig 7.4 Zero Level DFD**

## FIRST LEVEL DFD:



**Fig 7.5 First Level DFD**

## SECOND LEVEL DFD:



**Fig 7.6 Second Level DFD**





# CHAPTER-8

# CODING

## Home Page:

```
<!DOCTYPE html>

<html>

<head>

    <title>TRAVEL AND TOURISM</title>

    <link rel="stylesheet" type="text/css" href="css/style.css">

</head>

<body>

    <header>

        <div class="main">

            <div class="logo">

            </div>

            <ul>

                <li class><a href="index.html">Home</a></li>

                <li class><a href="index2.html">Registration</a></li>

                <li class><a href="index4.html">Gallery</a></li>

                <li class><a href="index5.html">About</a></li>

                <li class><a href="index3.html">Contact</a></li>

            </ul>

        </div>

        <div class="title">

            <h1>TRAVEL AND TOURISM</h1>

        </div>

        <div class="button">

            <a href="index6.html" class="btn">Watch Video</a>

            <a href="index7.html" class="btn">Login Form</a>

        </div>

    </header>
```

</body>

</html>

## Contact Page:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Contact-us</title>

<link rel="stylesheet" href="css/style3.css">

<link  
href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;600;800&display=swap"  
rel="stylesheet">

</head>

<body>

<div class="container">

<h1>Contact With Us</h1>

<p>We would love to respond to your queries and help you succeed.<br>Feel free to  
get in touch with us.</p>

<div class="contact-box">

<div class="contact-left">

<h3>Sent Your Request</h3>

<form>

<div class="input-row">

<div class="input-group">

<label>Name</label>

<input type="text" placeholder="Donald  
Trump">

</div><div class="input-group">

```

                    <label>Phno</label>
                    <input type="text" placeholder="+1 412
5203231">

                </div>
            </div>
            <div class="input-row">
                <div class="input-group">
                    <label>Email</label>
                    <input type="email"
placeholder="abc@gmail.com">

                </div>
                <div class="input-group">
                    <label>Subject</label>
                    <input type="text" placeholder="Demo">

                </div>
            </div>

            <label>Message</label>
            <textarea rows="5" placeholder="Your
Message"></textarea>

            <button type="submit">SEND</button>
        </form>
    </div>
    <div class="contact-right">
        <h3>Reach Us</h3>

        <table>
            <tr>
                <td>Email</td>
                <td>contactus@example.com</td>
            </tr></tr>

```

```

        <td>Phone</td>
        <td>+1 012 345 6789</td>
    </tr>
    <tr>
        <td>Address</td>
        <td>#7-19B, Chinna Ramyalam vedhi<br>
        Mylavaram,krishna Dt,<br>
        Andhra Pradesh 521230
    </td>
    </tr>
</table>

</div>

</div>

</div>
</body>
</html>

```

## Home Page CSS:

```

*{
    margin: 0;
    padding: 0;
    font-family: Century Gothic;
}

header{
    background-image: url(../1.jpg);
    height: 100vh;
    background-size: cover;
    background-position: center;
}

```

```

ul{
    float: right;
    list-style-type: none;
    margin-top: 25px;
}

ul li{
    display: inline-block;
}

ul li a{
    text-decoration: none;
    color: #fff;
    padding: 5px 20px;
    border: 1px solid transparent;
    transition: 0.65 ease;
}

ul li a:hover{
    background-color: #fff;
    color: #000;
}

ul li.active a{
    background-color: #fff;
    color: #000;
}

.logo img{
    float: left;
    width: 100px;

```

```
        height: auto;
    }

    .main{
        max-width: 1200px;
        margin: auto
    }

    .title{
        position: absolute;
        top: 50%;
        left: 50%;
        transform: translate(-50%,-50%);
    }

    .title h1 {
        color: #fff;
        font-size: 70px;
    }

    .button{
        position: absolute;
        top: 62%;
        left: 50%;
        transform: translate(-50%,-50%);
    }

    .btn{
        border: 1px solid #fff;
        padding: 10px 30px;
        color: #fff;
        text-decoration: none;
```

```
        transition: 0.65 ease;
    }

    .btn:hover{
        background-color: #fff;
        color: #000;
    }
```

## Contact Page CSS:

```
*{
    margin: 0;
    padding: 0;
}

body{
    background: #55ffe5;
    font-size: 14px;
    font-family: 'Poppins', sans-serif;
}

.container{
    width: 80%;
    margin: 50px auto;
}

.contact-box{
    background: #fff;
    display: flex;
}

.contact-left{
    flex-basis: 60%;
```



```
padding: 40px 60px;
}
```

```
.contact-right{
    flex-basis: 60%;
    padding: 40px 60px;
    background: #1c00b5;
    color: #fff;
}
```

```
h1{
    margin-bottom: 10px;
}
```

```
.container p{
    margin-bottom: 10px;
}
```

```
.input-row{
    display: flex;
    justify-content: space-between;
    margin-bottom: 20px;
}
```

```
.input-row .input-group{
    flex-basis: 45%;
}
```

```
input{
    width: 100%;
    border: none;
    border-bottom: 1px solid #ccc;
```

```
        outline: none;
        padding-bottom: 5px;
    }
```

```
textarea{
    width: 100%;
    border: 1px solid #ccc;
    outline: none;
    padding: 10px;
    box-sizing: border-box;

}
```

```
label{
    margin-bottom: 6px;
    display: block;
    color: #1c00b5;
}
```

```
button{
    background: #1c00b5;
    width: 100px;
    border: none;
    outline: none;
    color: #fff;
    height: 35px;
    border-radius: 30px;
    margin-top: 20px;
    box-shadow: 0px 5px 15px 0px rgba(28,0,181,0.3);

}
```

```
.contact-left h3{
    color: #1c00b5;
    font-weight: 600;
    margin-bottom: 30px;
}
```

```
.contact-right h3{

    font-weight: 600;
    margin-bottom: 30px;
}
```

```
tr td:first-child{
    padding-right: 20px;

}
```

```
tr td{
    padding-top: 20px;
}
```

## **Registration Page Using PHP:**

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8"/>
    <title>Registration</title>
    <link rel="stylesheet" href="style.css"/>
</head>
<body>
<?php
    require('db.php');
```

```

// When form submitted, insert values into the database.
if (isset($_REQUEST['username'])) {
    // removes backslashes
    $username = stripslashes($_REQUEST['username']);
    //escapes special characters in a string
    $username = mysqli_real_escape_string($con, $username);
    $email = stripslashes($_REQUEST['email']);
    $email = mysqli_real_escape_string($con, $email);
    $password = stripslashes($_REQUEST['password']);
    $password = mysqli_real_escape_string($con, $password);
    $create_datetime = date("Y-m-d H:i:s");
    $query = "INSERT into `users` (username, password, email, create_datetime)
        VALUES ('$username', '" . md5($password) . "', '$email', '$create_datetime)";
    $result = mysqli_query($con, $query);
    if ($result) {
        echo "<div class='form'>
            <h3>You are registered successfully.</h3><br/>
            <p class='link'>Click here to <a href='login.php'>Login</a></p>
            </div>";
    } else {
        echo "<div class='form'>
            <h3>Required fields are missing.</h3><br/>
            <p class='link'>Click here to <a href='registration.php'>registration</a> again.</p>
            </div>" }
    } else {
        ?>
        <form class="form" action="" method="post">
            <h1 class="login-title">Registration</h1>
            <input type="text" class="login-input" name="username" placeholder="Username" required />
            <input type="text" class="login-input" name="email" placeholder="Email Adress">
            <input type="password" class="login-input" name="password" placeholder="Password">
            <input type="submit" name="submit" value="Register" class="login-button">

```

```

        <p class="link"><a href="login.php">Click to Login</a></p>
    </form>
<?php
    }
?>
</body>
</html>

```

## Login Page Using PHP:

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8"/>
    <title>Login</title>
    <link rel="stylesheet" href="style.css"/>
</head>
<body>
<?php
    require('db.php');
    session_start();

    // When form submitted, check and create user session.
    if (isset($_POST['username'])) {
        $username = stripslashes($_REQUEST['username']); // removes backslashes
        $username = mysqli_real_escape_string($con, $username);
        $password = stripslashes($_REQUEST['password']);
        $password = mysqli_real_escape_string($con, $password);
        // Check user is exist in the database
        $query  = "SELECT * FROM `users` WHERE username='$username'
                    AND password='" . md5($password) . "'";
        $result = mysqli_query($con, $query) or die(mysqli_error());
        $rows = mysqli_num_rows($result);
        if ($rows == 1) {
            $_SESSION['username'] = $username;

```

```

        // Redirect to user dashboard page
        header("Location: dashboard.php");
    } else {
        echo "<div class='form'>
            <h3>Incorrect Username/password.</h3><br/>
            <p class='link'>Click here to <a href='login.php'>Login</a> again.</p>
            </div>";
    }
} else {
?>
<form class="form" method="post" name="login">
    <h1 class="login-title">Login</h1>
    <input type="text" class="login-input" name="username" placeholder="Username"
autofocus="true"/>
    <input type="password" class="login-input" name="password" placeholder="Password"/>
    <input type="submit" value="Login" name="submit" class="login-button"/>
    <p class="link"><a href="registration.php">New Registration</a></p>
</form>
<?php
    }
?>
</body>
</html>

```



# CHAPTER -9

# TESTING

Testing is a process of executing a program with the aim of finding error. To make our software perform well it should be error free. If testing is done successfully it will remove all the errors from the software. Software testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is defect free. It involves execution of a software component or system component to evaluate one or more properties of interest. Software testing also helps to identify errors, gaps or missing requirements in contrary to the actual requirements. It can be either done manually by using automated tools.

## Principles of Testing

1. All the test should meet the customer requirements.
2. To make our software testing should be performed by third party.
3. Exhaustive testing is not possible.
4. Start testing with small parts and extend it to large parts.
5. All the test to be conducted should be planned before implementing it(v) It follows pareto rule(80/20 rule) which states that 80% of errors comes from 20% of implementation.

## 9.1 SOFTWARE TESTING

Software testing is a critical element of software quality assurance and represents the ultimate reuse of specification. Design and Code testing represents interesting anomaly for the software during earlier definition and development phase, it was attempted to build software from an abstract concept to tangible implementation. The testing phase involves, testing of the development of the system using various techniques.

## 9.2 TESTING TECHNIQUES

### 9.2.1 White Box Testing

White box testing is a test case design method that uses control structure of the procedural design to derive test cases. The software tester has knowledge of their inner



Workings, structure and language of the software, or at least its purpose. After performing white box testing it was identified that The Leave Recording System (LRS) software guarantees that all independent paths within the modules have been exercised at least once. It has been exercised all logical decisions on their true and false sides. It was tested to execute all loops at their boundaries and within their Operational bounds. It was tested for the internal data structures to ensure their validity.

### **9.2.2 Black Box Testing**

Black box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. This tests provides inputs and responds to outputs without considering how the software works.

## **9.3 TESTING STRATEGIES**

### **9.3.1 Unit Testing**

Unit testing is a software testing technique by means of which individual units of software i.e. group of computer program modules, usage procedures and operating procedures are tested to determine whether they are suitable for use or not. It is a testing method using which every independent modules tested to determine if there are any issue by the developer himself. It is correlated with functional correctness of the independent modules.

### **9.3.2 Integration Testing**

Integration Testing is a level of software testing where individual units are combined and tested to verify if they are working as they intend to when integrated. The main aim here is to test the interface between the modules. Just unit testing is not enough for multiple reasons like A module/unit is usually designed by an individual software developer whose techniques and programming logic differs from that of other programmers Often at the time of module development, user requirements change and these new requirements may not be unit tested. This instigates Issues like data formatting, error trapping, hardware interfaces, and third-party service interfaces are sometimes missed during unit testing So, no matter how efficiently each

module/unit is running, if they aren't properly integrated, it will affect the functionality of the software program. As a solution integration testing is implemented. This article 'What is Integration Testing?' further lists out the advantages of integration testing.

### **9.3.3 System Testing**

Testing the entire system as a whole and checking for its correctness is system testing. The system is tested for discrepancies between the system and its original objectives. The project was effective and efficient.



# CHAPTER - 10

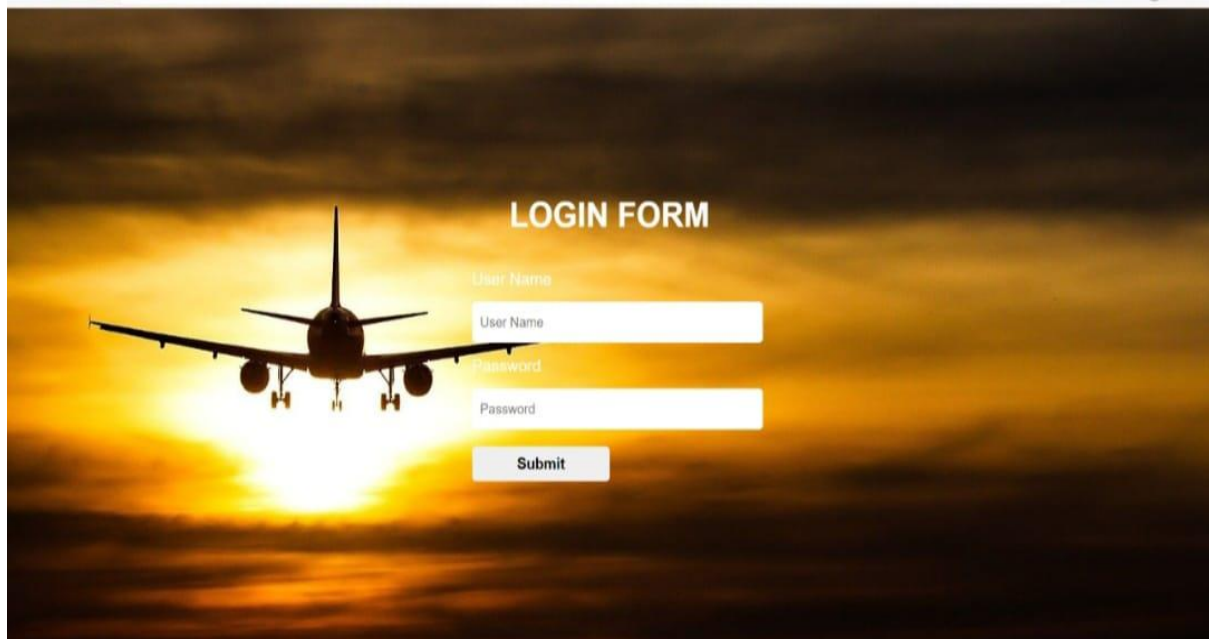
## SCREENS



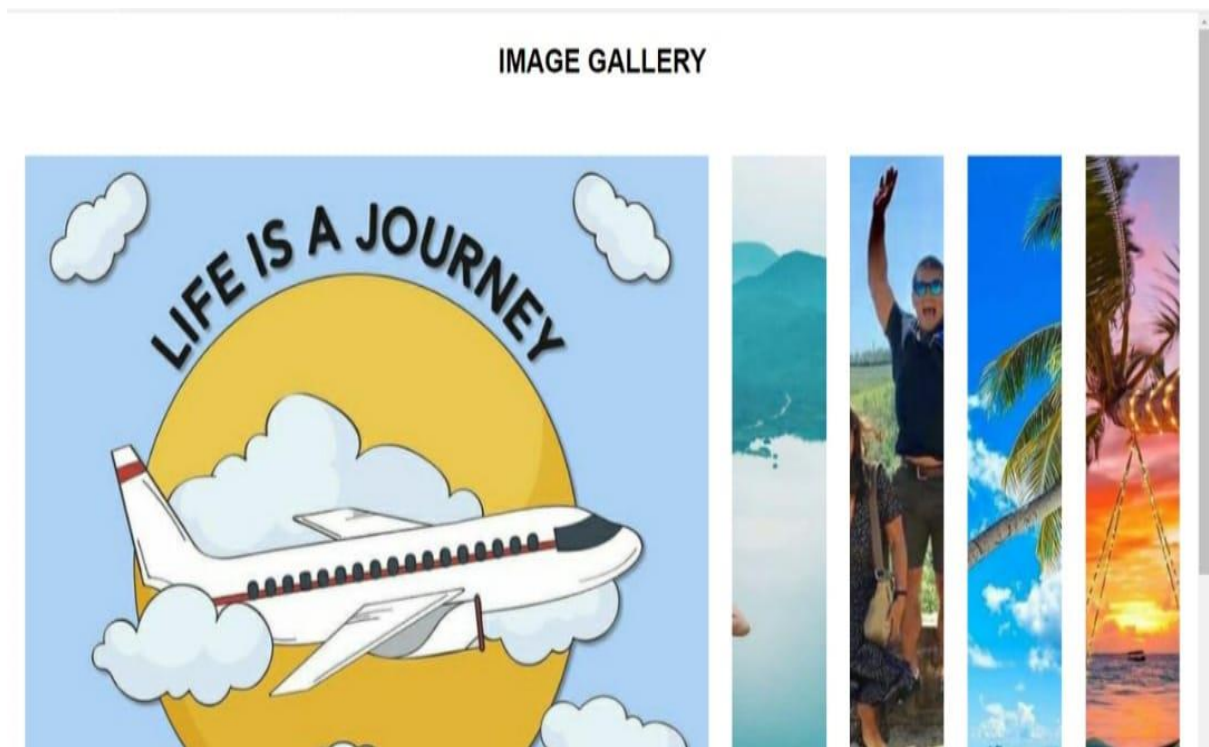
Screen 10.1 Home Page

The screenshot shows a registration form titled "REGISTRATION FORM" in white, bold, sans-serif capital letters. The background is a dark, dramatic image of an airplane flying towards the viewer against a bright, orange, and yellow sunset sky. The form consists of several white input fields and a submit button. The labels for the fields are in white text: "Full Name:", "User Name:", "Email:", and "Password:". The input fields are white with a subtle shadow. The "Submit" button is white with black text. The overall layout is clean and modern, with a focus on the travel theme.

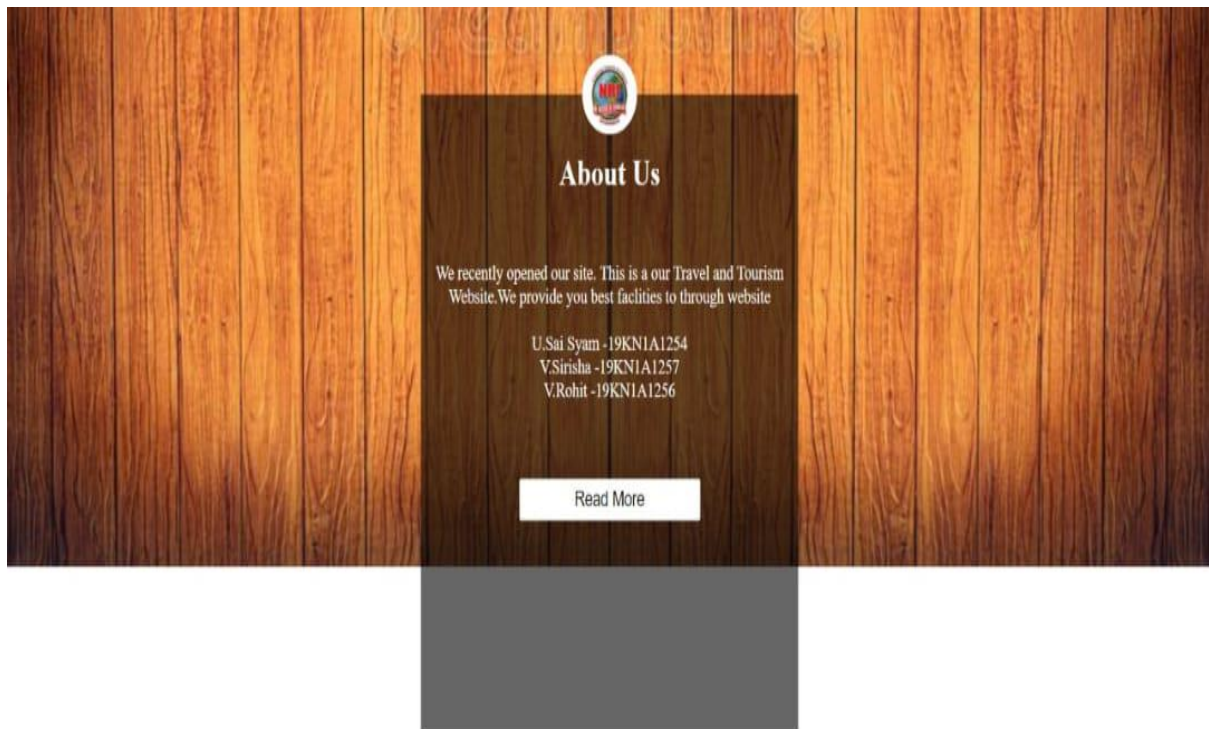
Screen 10.2 Registration Page



**Screen 10.3 Login Page**



**Screen 10.4 Gallery Page**



**Screen 10.5 About Page**

### Contact With Us

We would love to respond to your queries and help you succeed. Feel free to get in touch with us.

#### Sent Your Request

Name

Donald Trump

Phone

+1 412 5203231

Email

abc@gmail.com

Subject

Demo

Message

Your Message

SEND

#### Reach Us

Email

contactus@example.com

Phone

+1 012 345 6789

Address

#7-19B, Chinna Ramyalam vedhi  
Mylavaram,krishna Dt,  
Andhra Pradesh 521230

**Screen 10.6 Contact Page**



# CHAPTER-11

## **CONCLUSION**

This web application was successfully created and stored all the travel admin tourism package booking creating and managing tour details into the database using this application. The application was tested very well and the errors were properly debugged. Testing concluded that the performance of the system is satisfactory. All the necessary output is generated. This system provides an easy way to automate all the functionalities of consumption. If this application is implemented in few consumptions, it will be helpful. Further enhancements can be made to the project, so that the website works in a very attractive and useful manner than the present one. It is concluded that the application works well and satisfy the needs. It also acts as the sharing of files to the valuable resources.





## CHAPTER -12

## REFERENCES

Wang Y&Pizam, A. (2011). Destination marketing and management: theories and applications. London: CABI. Wanhill, S. R. C. (1980).

Tackling seasonality:

a technical note. *International Journal of Tourism Management*, 1, 243–245. WCED. (1987).

Our common future Oxford:

World Commission on Environment and Development and Oxford University Press.

Weitzman, M. L. (1998).

Recombinant growth *Quarterly Journal of Economics*, 113, 331– 360. Wensveen, J. G. (2007). Air transportation: a management perspective. Aldershot: Ashgate. Werther, H., & Klein, S. (1999). Information technology and tourism – A challenging relationship. Vienna: Springer. Weston, R. (1986).

The ubiquity of room taxes. *Tourism Management*, 4, 194–198. Wheeler, M. (1994).

The emergence of ethics in tourism and hospitality. *Progress in Tourism, Recreation and Hospitality Management*, 6, 647–654. White, P. R. (2008).