**TOURISM MANAGEMENT SYSTEM**

**PROJECT REPORT**

Submitted by

**SHATHIYA SHRIDHAR B(20-BC-85)**

Under the guidance of

**Dr.B.AZHAGUSUNDARI MCA., M.Phil., Ph.D.**

Associate Professor, UG Department of Computer Applications

A Project report submitted to

**NALLAMUTHU GOUNDER MAHALINGAM COLLEGE**

Affilliated to Bharathiar University, towards partial fulfillment

of the requirements for the award of the degree of

**BACHELOR OF COMPUTER APPLICATIONS**



**DECEMBER 2022**

**UG DEPARTMENT OF COMPUTER APPLICATIONS**

**NALLAMUTHU GOUNDER MAHALINGAM COLLEGE**

**Pollachi-642001**.

(AUTONOMOUS)

(An ISO 9001:2015 Certified Institution)

(Re-Accredited by NAAC)

**CERTIFICATE Ⅰ**

We hereby declare that project work entitled by “**Tourism Management System”** is the original work done by us.

**Date :**

**Place :**  Pollachi

**SHATHIYA SHRIDHAR B(20-BC-85)**

**CERTIFICATE Ⅱ**

This is to certify that the project work entitled “**Tourism Management System**” is a record of the original project done by **Shathiya Shridhar B(20-BC-85)** under the supervision and guidance **Dr.B.Azhagusundari** of andit does not form the basis for the award of any degree / diploma in any other University / Institution.

**Date:**

**Place:** Pollachi

**Signature of the Guide**

**Dr.B.AZHAGU SUNDARI**

**CERTIFICATE Ⅲ**

This is to certify that the project work done by **Shathiya Shridhar B(20-BC-85)** pursing final year B.C.A., submitted for the **Nallamuthu Gounder Mahalingam College**, affiliated to Bharathiar University in partial fulfilment of requirements for the award of the degree of Bachelor of Computer Application and does not form the basis for the award of in any other University / Institution.

**GUIDE HEAD OF THE DEPARTMENT**

This is to certify that this is a project work entitled

**TOURISM MANAGEMENT SYSTEM**

Submitted by

**SHATHIYA SHRIDHAR B(20-BC-85)**

For External Viva-voce examinations held on

**EXTERNAL EXAMINER** **INTERNAL EXAMINER**

**PRINCIPAL**

**ACKNOWLEDGEMENT**

I express my sincere thanks to all the people who have a possible for the successful completion of this project.

My thanks to the Principal **Dr.Muthukumaran M.A., M.Phil., Ph.D. ,**for giving me the opportunity to learn in this prestigious institution.

My sincere thanks to **Dr. K. Haridas M.C.A., M.Phil., Ph.D.,** Head of the Department of BCA for granting me permission to take up this project.

I express my sincere thanks to my guide **Dr. B.Azhagusundari MCA., M.Phil., Ph.D.,** Associate Professor who gave us necessary information and valuable suggestions regarding the project without whom the project would not have been a success.

I am are grateful to the staff members of UG Department of Computer Application, NGM College for their valuable suggestions.

I also thank my parents and friends, who have supported me in completing the project.

|  |  |  |
| --- | --- | --- |
|  | **TABLE OF CONTENTS** |  |
| **CHAPTER** | **TITLE** | **PAGE NO** |
| **1** | **INTRODUCTION** | **1** |
|  | 1.1 ABOUT THE PROJECT | 1 |
|  | 1.2 MODULE DESCRIPTION | 2 |
| **2** | **SYSTEM STUDY** | **3** |
|  | 2.1 EXISTING SYSTEM | 3 |
|  | 2.1.1 DRAWBACKS OF EXISTING SYSTEM | 3 |
|  | 2.2 PROPOSED SYSTEM | 3 |
|  | 2.2.1 ADVANTAGES OF PROPOSED SYSTEM | 3 |
|  | 2.3 SOFTWARE FEATURES | 4 |
|  | 2.4 MYSQL FEATURES | 5 |
|  | 2.5 SYSTEM SPECIFICATION | 7 |
|  | 2.5.1 HARDWARE SPECIFICATION | 7 |
|  | 2.5.2 SOFTWARE SPECIFICATION | 7 |
| **3** | **SYSTEM DESIGN** | **8** |
|  | 3.1 FILE DESIGN | 8 |
|  | 3.2 INPUT DESIGN | 8 |
|  | 3.3 DATABASE DESIGN | 9 |
|  | 3.4 OUTPUT DESIGN | 9 |
|  | 3.5 TABLE DESIGN | 10 |
|  | 3.6 DATA FLOW DIAGRAM | 13 |
|  | 3.7 ENTITY RELATIONSHIP DIAGRAM | 15 |
| **4** | **SYSTEM TESTING** | **16** |
|  | 4.1 TESTING | 16 |
| **5** | **CONCLUSION & FUTURE ENHANCEMENT** | **25** |
|  | 5.1 CONCLUSION | 25 |
|  | 5.2 FUTURE ENHANCEMENT | 25 |
| **6** | **BIBLIOGRAPHY** | **26** |
| **7** | **APPENDIX** | **27** |
|  | A. SAMPLE SCREEN | 27 |
|  | B. SAMPLE CODING | 49 |

# 1. INTRODUCTION

## 

## 1.1 ABOUT THE PROJECT

The Project Entitled as “**Online Tourism Management System**“ was programmed using **PHP** as a frontend and **MYSQL** as backend tool for database. This project is very helpful when you are having a business of ticketing system via online with the place, destination, stay, and all packages.

The main focus of this system is to help the user to easily booking with generating of tickets for travel or vacation purposes. The customer can sign in/sign up and search for the tour packages. Customers can book easily through the site. And there are other features such as customers can check their Tour history and their issued tickets, and other services.

Travel and tourism management system is used to book a tour from anywhere in the world by a single dynamic website which will help the user to know all about the places and tour details in a single website. The admin can add packages to the website from a certain travel agents and hotels by create a tour page. Then the users can sign in and book each project, they can be confirmed by the admin in manage booking page. The user can see the confirmation in the booking page. It is an easiest platform for all travelers which can be easily booked and know the all details. Tour Management system is a dynamic website for tourism business. It is dynamic and responsive web design. It is also called travel technology solution for agencies & tour operation. Nearly everyone goes to a vacation for this ‘a Tourism management system’ would play a vital role in planning the perfect trip. The tourism management system allows the user of the system access all the details such as location, events, etc. The main purpose is to help tourism companies to manage customer and hotels etc. The system can also be used for both professional and business trips.

**1.2 Modules Description:**

**Admin**

Admin can create tour Package and also manage the packages details, Manage Users, Inquiries, Issues, Booking, Pages, Change Password, and Admin Dashboard

**Package Details**

This module allows categorizing the different products by their characteristics. Admin can add the different package in this module which like package id, package name, package description, category, subcategory, price etc.

**User registration**

User can register yourself and use with valid email and password. The User forgot the Password using the registered Email Id and Mobile no user can create a new password, Tour Booking, Manage Booking, Generate Ticket.

**User login**

In this module maintaining the user login details. The user are get unique user name and password after user registration form. After login user can view the product and shop details.

## 2. SYSTEM STUDY

## 2.1 EXISTING SYSTEM

In the existing system, each task is carried out manually and processing is also a tedious job. In previous system travelers were maintaining time table details manually in pen and paper, which was time taking and costly. The travelers are not 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. It is tedious for a customer to plan a particular journey and have it executed properly.

### **2.1.1 DRAWBACKS OF EXISTING SYSTEM**

* Increased transaction leads to increased source document and hence maintenance becomes difficult.
* If any admin, user entry is wrongly made then the maintenance becomes very difficult.
* Lot of paper work required.
* Man power was more.

## 2.2 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.

### **2.2.1 ADVANTAGES OF PROPOSED SYSTEM**

* This system maintains user’s personal detail such as name, address, room preferences,
* payment information and contact details.
* With the help of reach user interface, user can easily navigate and use the new system to

**2.3 ABOUT THE SOFTWARE SPECIFICATION**

# HTML (HYPERTEXT MARKUP LANGUAGE)

**HTML** stands for **Hypertext Markup Language**, and it is the language in which, until recently, virtually all Web pages were written. Now, don’t break out in hives when you hear the word “language.” You don’t need complex logical or mathematical formulas to work with HTML, and you don’t need to think like a programmer to use it.

Hypertext refers to the way in which Web pages (HTML documents) are linked together. When you click a link in a Web page, you are using hypertext. It is this system of linking documents that has made the World Wide Web the global phenomenon it has become. Markup Language describes how HTML works. With a markup language, you simply “**markup**” a text document with tags that tell a Web browser how to structure it. HTML originally was developed with the intent of defining the structure of documents (headings, paragraphs, lists, and so forth) to facilitate the sharing of scientific information between researchers.

# FOUR KEY CONCEPTS

The first step toward understanding and working with HTML is learning the basic terms that describe most of the functions of this language. You will come across these terms repeatedly as you use HTML and if you understand them, you will have progressed a long way toward comprehending HTML, not to mention XHTML.

# ELEMENTS

All HTML pages are made up of elements. Think of an element as a container in which a portion of a page is placed. Whatever is contained inside the element will take on the characteristics of that element. For example, to identify a heading on a page, you would enclose it in a heading element <h1> </h1>. If you want to create a table, you put the table information inside the table element <table> </table>. To construct a form, you need the form element <form> </form>.

# TAGS

Often, you’ll find the terms element and tag used interchangeably. It’s fairly common, but not strictly accurate. An element is made up of two tags: an opening tag and a closing tag. Although it might seem somewhat picky to make this distinction, when you begin to work with XHTML (Extensible Hypertext Markup Language), it will be a very important difference to remember.

All tags are constructed the same way. The tag begins with a “less than” sign (<), then the element name, followed by a “greater than” sign (>). For example, an opening tag for the paragraph element would look like this: <p>. The only difference in a closing tag is that the closing tag includes a slash (/) before the element name: </p>. Your content goes between the tags. A simple paragraph might look like this :<p>this is an HTML paragraph. </p>

Some elements do not use closing tags because they do not enclose content. These are called empty elements. For example, the line break element <br> does not require a closing tag. In the case of empty elements, add a closing slash after the element name, like this: <br />.

When a browser sees the slash, it will recognize the element as one that does not need a separate, closing tag.

**About PHP**

PHP is a powerful server-side scripting language for creating dynamic and interactive websites. PHP widely used; free and efficient alternative to competitors such as Microsoft’s

ASP. PHP is perfectly suited for Web development and can be embedded directly into the HTML code. The PHP syntax is similar to pearl and C.

PHP is open source that it is readily available and absolutely free. Stability, flexibility and speed are chief qualities that attract to choose PHP. PHP have multiple extensions and is extremely scalable. **Server-side scripting**

This server-side scripting is the most traditional and main target field for PHP. Programmer needs three things to make this work. Programmer need to run the web server, with a connected PHP installation. Programmer can access the PHP program output with a web browser, viewing the PHP page through the server. All these can run on your home machine if programmers are just experimenting with PHP programming.

**Command line scripting**

Programmer can make a PHP script to run it without any server or browser. Programmers only need the PHP parser to use it this way. This type of usage is ideal for scripts regularly executed using croon (on\*nix or Linux) or Task Scheduler (on Windows). These scripts can also be used for simple text processing tasks. **Features of PHP**

* PHP runs on different platforms (Windows, Linux, UNIX, etc.)  PHP is compatible with almost all servers used today.
* **PHP is free to download from the official PHP resource: www.php.net.**

## 2.4 MYSQL FEATURES

MYSQL is an open-source relational database management systems (RDBMS), is developed, distributed and supported by MYSQL AB. MYSQL is a popular choice of database for use in web applications MYSQL can be scaled by deploying it on more powerful hardware, such as a multi-processor server with gigabytes of memory. MYSQL is easy to use, yet extremely powerful, secure, and scalable. And because of its small size and speed, it is the ideal database solution for Web sites.

### **MYSQL is a database management system**

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amount of information in a corporation network. To add, access and process data stored in a computer database we need a database management system such as MYSQL server. Since computers are very good at handling large amount of data, database management system plays a central role in computing.

**MYSQL is a relational database management system**

A relational database stores separate data in separate tables rather than putting all the data in one big storeroom. This adds speed and flexibility. The SQL part of “MYSQL” stands for “Structured Query Language”. SQL is the most common standardize language used to access database and is defined by the ANSI/ISO SQL standard. The SQL standard has been evolving since 1986 and several versions exist.

**MYSQL software is open source**

Open source means that it is possible for anyone to use modify the software. Anybody can download the MYSQL software uses the GPL (GNU General Public License), to define what we may and may not use do with the software.

**MYSQL Server works in Client/ Server or embedded systems**

The MYSQL database software is a client/server system that consists of a multi-threaded SQL server that supports different backend, several different client programs and libraries, administrative tools and a wide range of Application Programming Interface (APIs). A large amount of contributed MYSQL software is available:

Modern day websites seem to be relying more and more on compel the Structured Query Language is a very popular database language, and its standardization makes it easy to store, update and access data. One of the most powerful SQL servers out there is called MYSQL and

surprisingly enough, it’s free.

Some of the features of MYSQL include: Handles large databases, in the area of 50,000,000+records. No memory leaks. Tested with a commercial memory leakage detector (purify). A privilege and password system which is very flexible and secure, and which allows host-based verification. Passwords are secure since all password traffic when connecting the server is encrypted.

**Features of MYSQL**

**Client/server Architecture:** MYSQL is a client/server system. There is a database server (MYSQL) and arbitrarily many clients (application programs), which communicate with the server. The clients can run on the same computer as the server or on another computer.

**SQL Compatibility:** As before said SQL is a standardized language for querying and updating data and for the administration of a database. Through the configuration setting solmode we can make the MYSQL server behave for the most part compatibly with various database systems.

**Stored procedures:** Stored procedures (SPs for short) are generally used to simplify steps such as inserting or deleting a data record.

**Triggers:** Triggers are SQL commands that are automatically executed by the server in certain database operations INSERT, UPDATE, and DELETE, MYSQL has supported triggers.

**Replication:** Replication allows the contents of a database to be copied (replicated) onto a number of computers to increase protection against system and to improve the speed of database queries.

**Platform independence:** MYSQL can be executed under a number of operating systems. The most important are Apple Macintosh OS X, Linux, Microsoft Windows, and the Unix.

**Speed: MYSQL is considered a very fast database program**

<?php

// DB credentials.

define('DB\_HOST','localhost');

define('DB\_USER','root');

define('DB\_PASS','');

define('DB\_NAME','tms');

// Establish database connection.

try

{

$dbh = new PDO("mysql:host=".DB\_HOST.";dbname=".DB\_NAME,DB\_USER, DB\_PASS,array(PDO::MYSQL\_ATTR\_INIT\_COMMAND => "SET NAMES 'utf8'"));

}

catch (PDOException $e)

{

exit("Error: " . $e->getMessage());

}

?>

**2.5 SYSTEM SPECIFICATION**

## 2.5.1 HARDWARE CONFIGURATION

## All the system with web browser can access this web application

**2.5.2 SOFTWARE SPECIFICATION**

**Web server :** XAMPP Server

**Backend :** MYSQL

**Server side scripting :** PHP

**Client side scripting :** HTML

# Designing Tools : VSCODE

# 3. SYSTEM DESIGN

## 3.1 FILE DESIGN

System design is the process of planning a new system to complement or altogether replace the old system. The purpose of the design phase is the first step in moving from the problem domain to the solution domain. System design is also called top-level design. The design phase translates the logical aspects of the system into physical aspects of the system.

# 3.2 INPUT DESIGN

The data, which is input to a computer – based information system, must be correct. If data is carelessly input and errors enter the system, it will lead to incorrect results whose consequences will be expensive and embarrassing to the designer. In data processing, the data entry operator often makes errors. This can be controlled by input design by using menu, interactive dialogue, consistent format etc.

In this system the users are provided with user friendly pages to give the input and if the user gives any wrong input validations are done and message boxes are provided in the necessary places. System is interactive dialogue, which simplifies the data entry or access, instead of remembering what to enter. User can choose from a list of options and type it in the cursor position. This will reduce the number of corrections while entering the data.

1. **Admin login -**  enter valid admin username and password to login to this screen (Refer fig – 2&3).
2. **User register & login –** enter the following details like username, password etc, to create a account and login using the registered username and password to this screen ( Refer fig –4&5 ).
3. **User details:** enter the name of the user, age, gender, number of adults & children, mode of transport
4. **Add package –** admin enter the package name to create new package in this screen (Refer fig – 12 ).

## 3.3 DATABASE DESIGN

The database design involves creation of tables that are represented in physical database as stored files. They have their own existence. Each table constitute of rows and columns where each row can be viewed as record that consists of related information and column can be viewed as field of data of same type. The table is also designed with some position can have a null value.

( Refer fig – 9 for screen shot of database design ).

## 3.4 OUTPUT DESIGN

The proposed system is a web oriented system and hence it does not provide any reports. The output results are viewed in the web pages itself. Outputs from the computer system are required primarily to communicate the result of processing to users. They are also used to override a permanent copy of the results for later consultation.The output reports and input documents should be documented in terms of data content and approximate layout; it is not necessary to define the methods of presentation. It is possible to work back for the output data items are derived by calculations or by logical deduction.

1. **view Packages–** user click the category menu to view the category (Refer fig –6).
2. **Package details –** user click the product to view the package details (Refer fig –10 ).
3. **Booked Details –** Admin can view the total number of booked users(ref fig -11 )

**3.5 TABLE DESIGN:**

**Table name :**  admin

**Primary key :** admin\_id

**Description :** This table is to store the admin login details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** |  | **Width** | **Constant** | **Description** |
| admin\_id | Varchar |  | 5 | Primary key | The unique id |
| username | Varchar |  | 15 | Not null | Username |
| password | Varchar |  | - | Not null | Login Password |
| updation\_date | Timestamp |  | - | Not null | Update date |

**Table name** : User

**Primary key :** User\_id

**Description** : This table is to store the user details

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Width** | **Constant** | **Description** |
| User\_id | Varchar | 6 | Primary key | User id |
| User\_name | Varchar | 6 | unique key | User name |
| User\_Email | Varchar | 20 | Secondary key | User Email |
| User\_mob | Int | 13 | Not null | Contact number |
| Password | Varcahr | - | Not null | Password |
| RegDate | Date/time | - | Not null | Registration Date |
| update\_date | Date/time | - | Not null | date |

**Table name :** Booking

**Primary key :** Booking\_id

**Foreign key :** User\_id

**Description :** This table is used to store the booking details

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Width** | **Constant** | **Description** |
| Booking\_id | Varchar | 6 | Primary key | Booking id |
| Package\_id | Varchar | 6 | Not null | Package id |
| userEmail | Varchar | 20 | Secondary  key | User Email |
| User\_id | Varchar | 6 | Foreign key | User id |
| From date | Date/Time | - | Not null | From date |
| To Date | Date/time | - | Not null | To date |
| Name | Varchar | 15 | Not Null | Name of the user |
| Age | Varchar | 2 | Not Null | Age of the user |
| Gender | Varchar | 5 | Not Null | Gender of the user |
| Adultcount | Varchar | 6 | Not Null | Count of adults |
| Childcount | Varchar | 4 | Not Null | Count of the children |
| Transportmode | varchar | 5 | Not Null | Mode of transport |
| Command | Varchar | 60 | Not null | Command any query |
| Reg\_date | Date/Time | - | Not null | regDate |

**Table name :** Review

**Primary key :** enquiry\_id

**Foreign Key** : user\_id

**Description** : This table is to store the user Details

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Width** | **Constant** | **Description** |
| Enq\_id | Int | 5 | Primarykey | Enquiry id |
| User\_id | Varchar | 6 | Foreign key | User id |
| Mobileno | Number | 13 | Not null | Contact No |
| UserEmail | Varchar | 20 | Not null | Email Id |
| Subject | Varchar | 10 | Not null | Subject details |
| Description | Text | 100 | Not null | Description |
| Post Date | Date/Time | - | Not null | Posting Date |
| Status | Varchar | 3 | Not null | Status |

**Table name** : tblpackage

**Primary key :** package\_id

**Description**  : This table is to store the package Details

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Width** | **Constant** | **Description** |
| package\_Id | Varchar | 6 | Primary key | Package id |
| Package\_Name | Varchar | 15 | Not null | Package Name |
| PackageType | Varchar | 20 | Not null | Package Type |
| PackageLoc | Text | 15 | Not null | Package Location |
| PackagePrice | Int | 10,2 | Not null | Package Price |
| PackageFeat | Varchar | 15 | Not null | Package Features |
| PackageDetail | Varchar | 10 | Not null | Package Details |
| CretionDate | Date/Time | - | Not null | Creation Date |
| UpdateDate | Date/Time | - | Not null | Update Date |

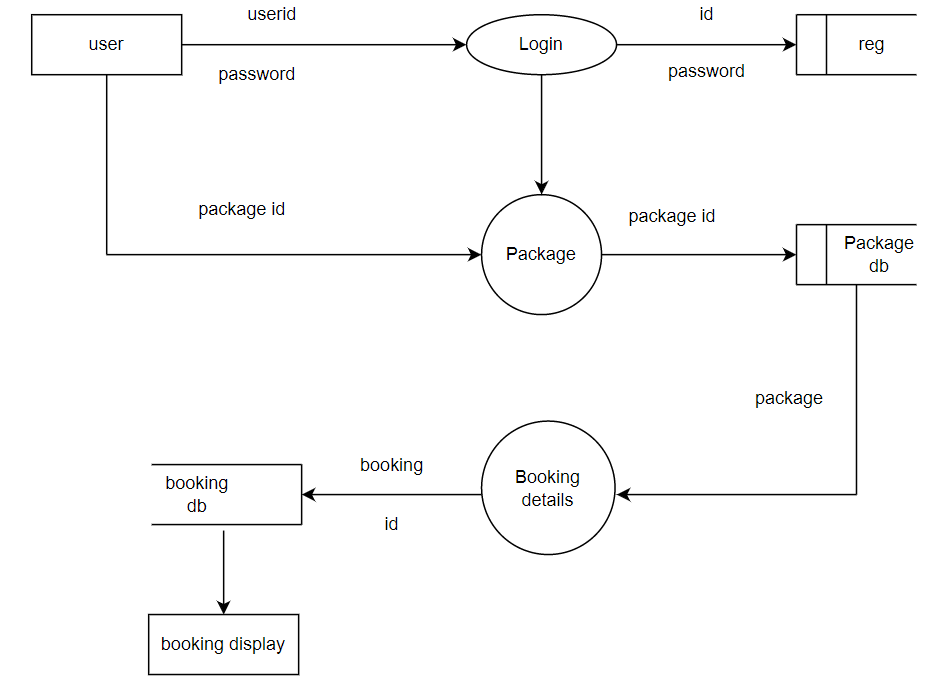
**3.6 DATA FLOW DIAGRAM**

# LEVEL 0

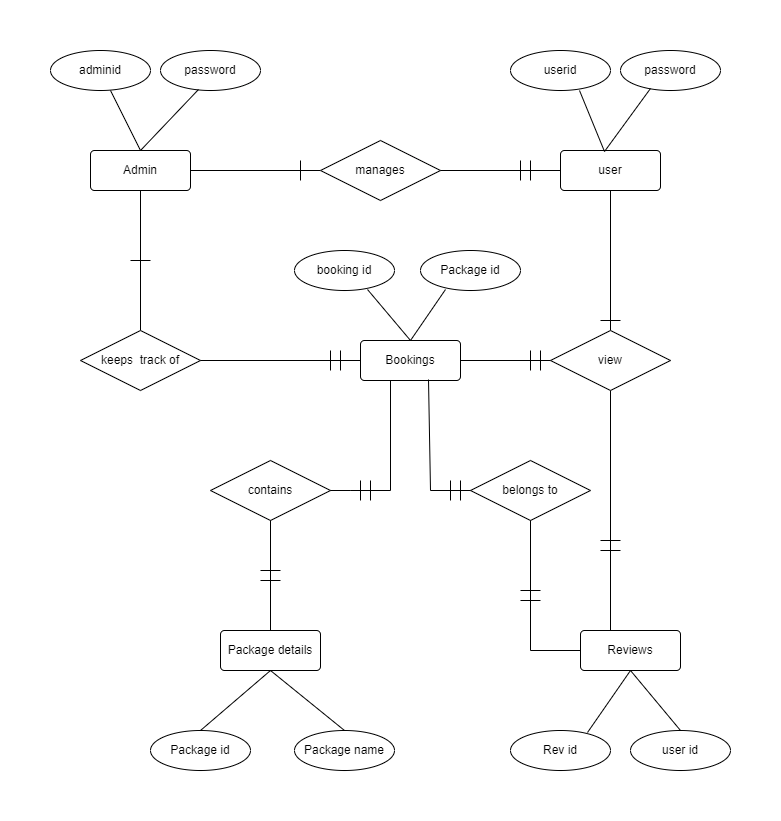
# LEVEL 1



# LEVEL 2



**3.5 ENTITY RELATIONSHIP :**



# 4. SYSTEM TESTING

## 4.1 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. However the data are created with the intent of determining whether the system will process them correctly without any errors to produce the required output.

**TESTING PERFORMED**

## ALPHA TESTING

Alpha testing is carried out by the developers to identify various types of issues or bugs before publishing. The developers focus remain on the task which a general user might want or experience**.** The alpha test results involving content experts indicated the prototype met the needs of the users. Majority of the testers give the feedback positively.

## BETA TESTING

Beta test follows revisions and utilizes the full product for testing. For this project the beta testing was performed by the students. This is the formal process to determine the usability of the prototype. Indirectly the developer were able to identify any weaknesses or problems from the perspective of the student in relation to the use of the developed software. Feedback of the beta testing, the students responded positively.

## INTERFACE TESTING

User interaction aspects are tested for proper operation during this testing. This section lists the functional requirements used for creating the test-case table, the test cases that were used to verify the interface table, and the results for the test-cases table.

**Table 1** lists the functional requirements for the interface built for this application, along with a short description of each requirement.

**Table 1.** list of functional requirements

|  |  |
| --- | --- |
| **Functional Requirement Number** | **Functional Requirement Description** |
| FR01 | The application shall have two types of authentications.  User authentication and Admin authentication. |
| FR02 | The application shall be accessible to all the users to browse all the categories and the items. |
| FR03 | The users shall be able to view the items they added to the shopping cart. |
| FR06 | The Admin shall be able to upload new/revised products as well as to add/modify the categories. |
| FR07 | The Admin shall be able to view all the users registered in the system. |
| FR08 | The Admin shall be able to view all the information about users who booked a package. |

## TEST CASES

**Table 2** shows the functional requirements used to write the test cases along with the testcase numbers for each test case and a short description of the test cases.

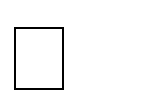
**Table 2.** List of test cases

|  |  |  |
| --- | --- | --- |
| **Functional**  **Requirement Number** | **Test Case Number** | **Test Case Description** |
| FR01 | TC01 | To test the Login/Authentication interface for the Admin. |
|  | TC02 | To test the Login/Authentication interface for the users |
| FR03 | TC03 | To test, users can view the packages.. |
| FR06 | TC04 | To test, Admin can upload new/revised package To test, Admin can upload new/revised packages |
| FR07 | TC05 | To test, Admin can view all the users registered in the system. |

The following list includes the steps that should be taken by the user, the conditions that should be met for the successful execution of the test case, and the end result that should be met for the test cases to pass.

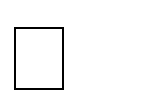
1. TC01: To test the Login/Authentication interface
   * Input: Username and Password
   * Output: Valid Destination Page
   * Valid Range : Admin Name Alphanumeric, Password Alphanumeric
   * End Messages/Result

* + 1. If (Admin == Valid Admin), an order form appears to complete the checkout process.
    2. ii. If ( Admin != Valid Admin), an error message is displayed on the Login interface.

1. TC02: To test the Login/Authentication interface
   * Input: Username and Password
   * Output: Valid Destination Page
   * Valid Range: User Name Alphanumeric, Password Alphanumeric
   * End Messages/Result
   1. If (User == Valid User), an order form appears to complete the checkout process.
   2. If (User != Valid User), an error message is displayed on the Login interface.

1. TC03: To test, the users can view the Products they add to the shopping cart.
   * Description : The system shows all the saved tour packages for a particular user.
   * Input: The user book packages from any of the available categories.
   * Output: The booking page pops up, showing the tour package that is added by the user.
   * End messages/Result.
     1. If (Selection == Packages == exists), the user is able to book the package.
2. TC04: To test, the Admin can upload new/revised categories and Packages.
   * Description: The Admin can add or upload more Package to a category or can add a completely new category. The Admin can also modify the price, information and shipping taxes, etc. for the existing packages and categories.
   * Input:
3. User = Admin
4. Selection = Package

* 1. If (User type = “Admin” &Selection = (Package|| Category)&& package =existing), then display the modified packages in the application.
  2. If (User type = “Admin” &Selection = packages || Category) & packages =existing), then display newly added Packages in the application.

1. TC05: To test, the Admin can view all the users registered in the system.
   * Description: The Admin can view all the users who are registered in the system in the database.
   * Input
     1. User Name Alphanumeric, Password Alphanumeric ii. User==Admin

iii. Selection==View Database

* + Output: User List
  + End messages/Result

1. If( login type == “Admin” & Database.clicked = ‘true’ and list.clicked=true and userlist.exists=true), then display users.
2. If (login type == “Admin” &Database.clicked = ‘true’ and list.clicked=true and userlist.exists=false), then display the empty database.

## RESULTS

This section lists the results that were produced by running the test cases. **Table 3** lists the test cases that were used while testing the interface along with the expected result and the actual results for each test case.

**Table 3.** List of test case result

|  |  |  |
| --- | --- | --- |
| **Test Case Number** | **Expected Result** | **Actual Result** |
| TC01 | PASS | PASS |
| TC02 | PASS | PASS |
| TC03 | PASS | PASS |
| TC04 | PASS | PASS |
| TC05 | PASS | PASS |

## PATH TESTING

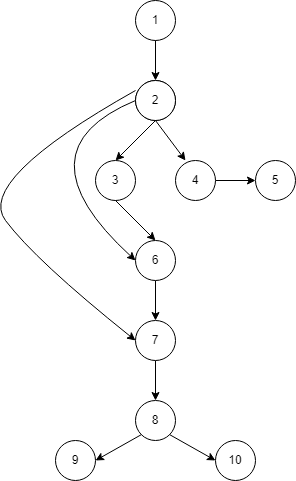
In this testing , cyclomatic complexity is implemented by using flow graph to measure the complexity of the program.

## PROGRAM

### **//user login program**

1. if(isset($\_POST['submit']))
2. if($query){
3. echo alert('You are successfully registered');}
4. else{
5. echo alert('Not registered something went worng');}
6. if(isset($\_POST['login']))
7. if($num>0){
8. $\_SESSION['login']
9. else{
10. $\_SESSION['errmsg']="Invalid email id or Password";}

### FLOW GRAPH



### **M = E – N + 2P**

where,

**E** = the number of edges in the control flow graph.

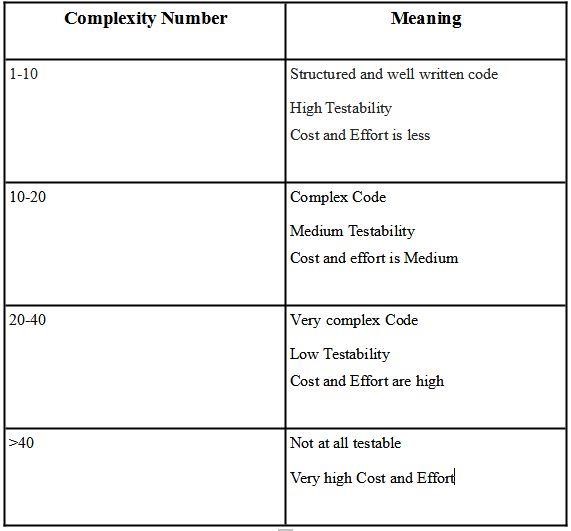
**N** = the number of nodes in the control flow graph.

**P** = the number of connected components.

#### **CYCLOMATIC COMPLEXITY – E = 11 , N = 10 , P = 1**

M = 11 – 10 + 2

= 3



#### RESULT

The result of complexity measurement is between 0-10, So the code is structured and well written. It has high testability.

# 5. CONCLUSION AND FUTURE ENHANCEMENTS

## 5.1 CONCLUSION

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

# 5.2 FUTURE ENHANCEMENTS

Due to the lack of time, the design part is not done so attractive. Further enhancements can be made in designing the screens. Some more forms can also be added so as to better retrieve the feedback details. Various other options can also be added for the better usability of project.

# 6 BIBLIOGRAPHY BOOK REFERENCES

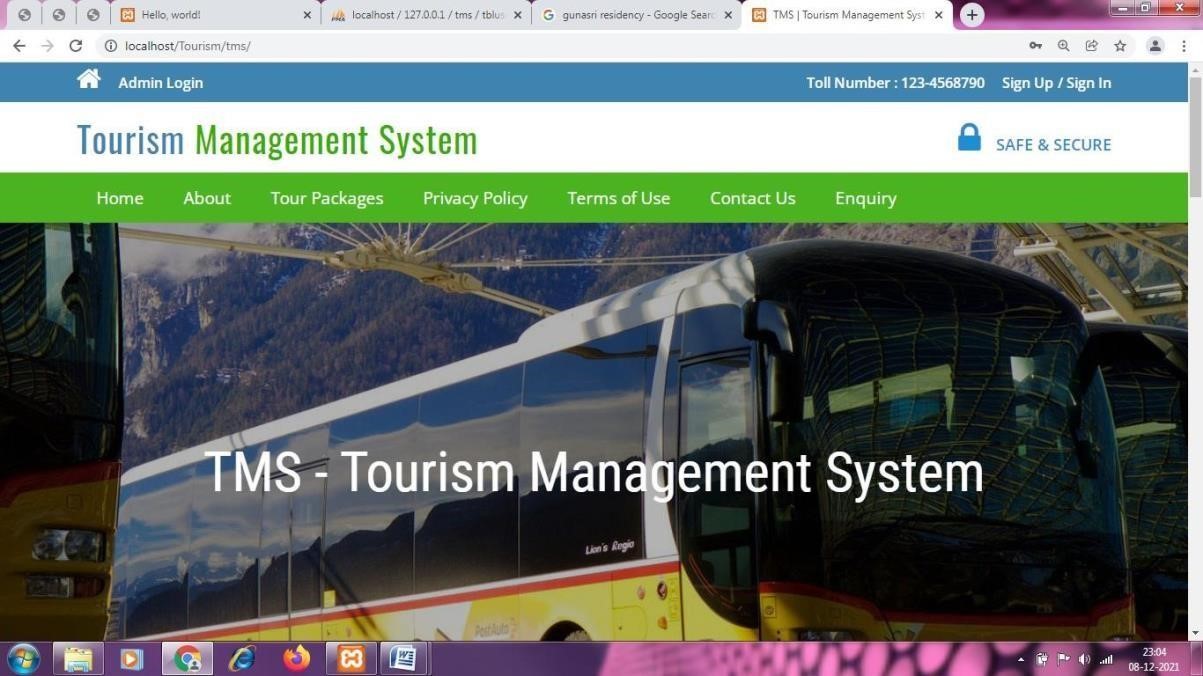
* Jesus Castagnetto, Sascha Schumann, **“Proffesional Php Programming”**, Addison wosley Publication, Fifth Edition.
* Jay Greenspan, Brad Bulgar, **“Mysql/Php Database Applications”**, Tata McGrawHill Publishing Company, Third Edition.
* William Stallings, **“Cryptography And Network Security”**, Tata McGraw-Hill Publishing Company, Third Edition.
* Bruce Schneier, **“Applied Cryptography”**, Pearson Education, Second Edition.
* Rogers Pressman, **“Software Engineering and Applications”**, Galgotie Publication, Sixth Edition.
* Matt Doyle, **Beginning PHP 5.3**, Published by Wiley Publishing, Inc. 10475 Cross point Boulevard, Indianapolis, IN 46256.
* Larry Ullman, **PHP and MySQL for Dynamic Web Sites** Fourth Edition , Peach pit Press, 1249 Eighth Street, Berkeley, CA 94710.

# WEBSITE

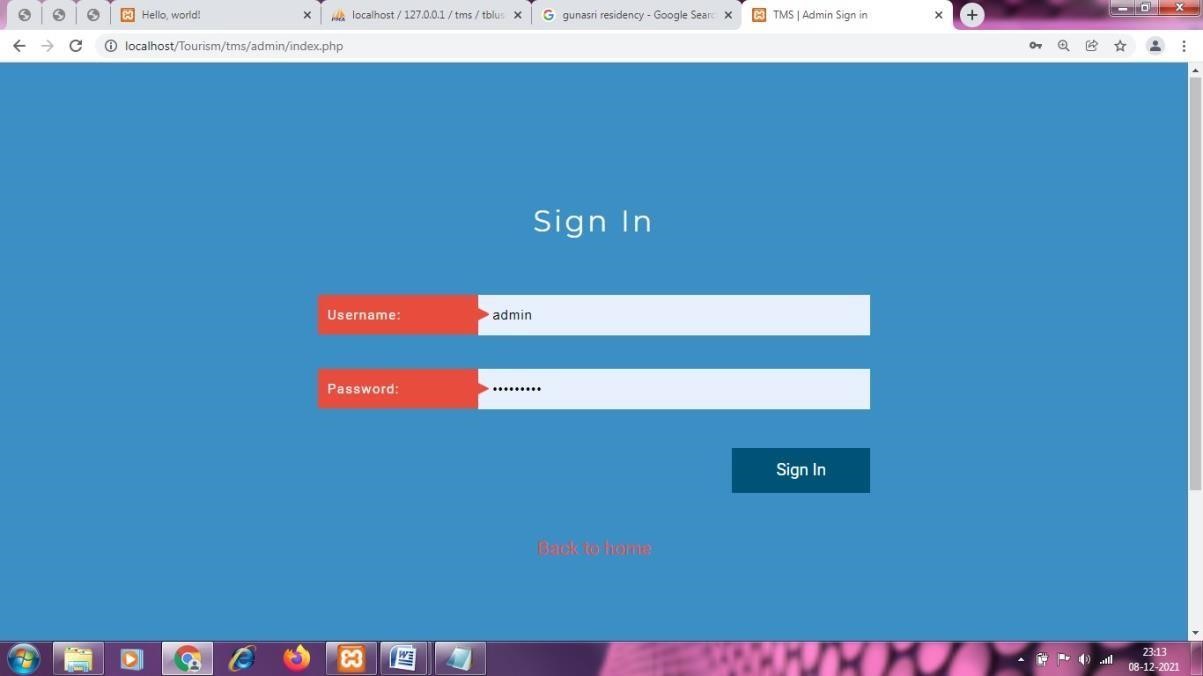
* www.msdn.microsoft.com
* www.phpcity.com
* www.phpgurukul.com
* www.myproject.com
* [www.youtube.com](http://www.youtube.com)
* [](http://) [www.w3schools.com](http://www.w3schools.com/)

# 7. APPENDIX

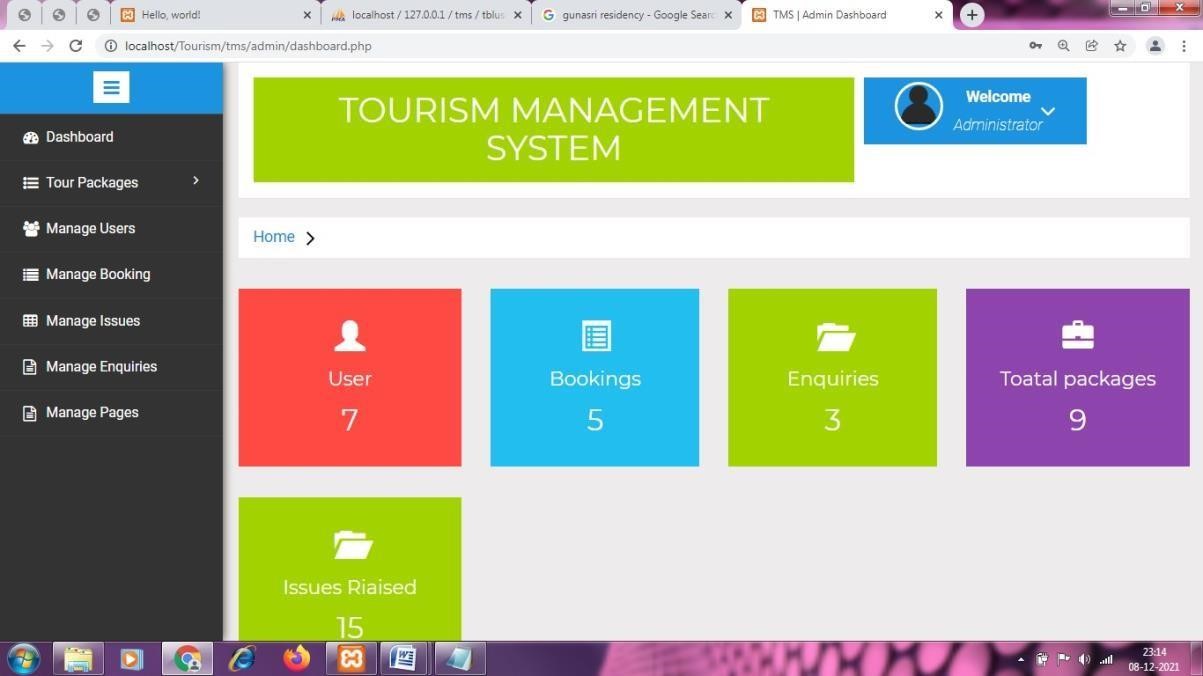
## SAMPLE SCREEN



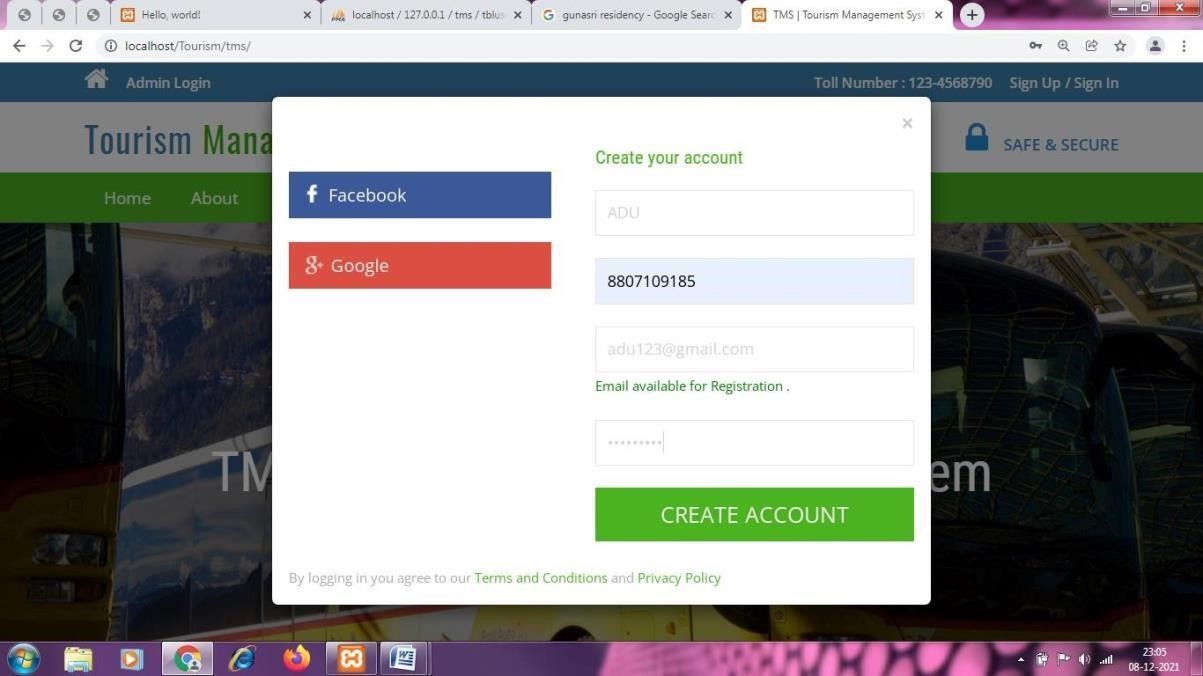
**fig – 1 ( Home page)**



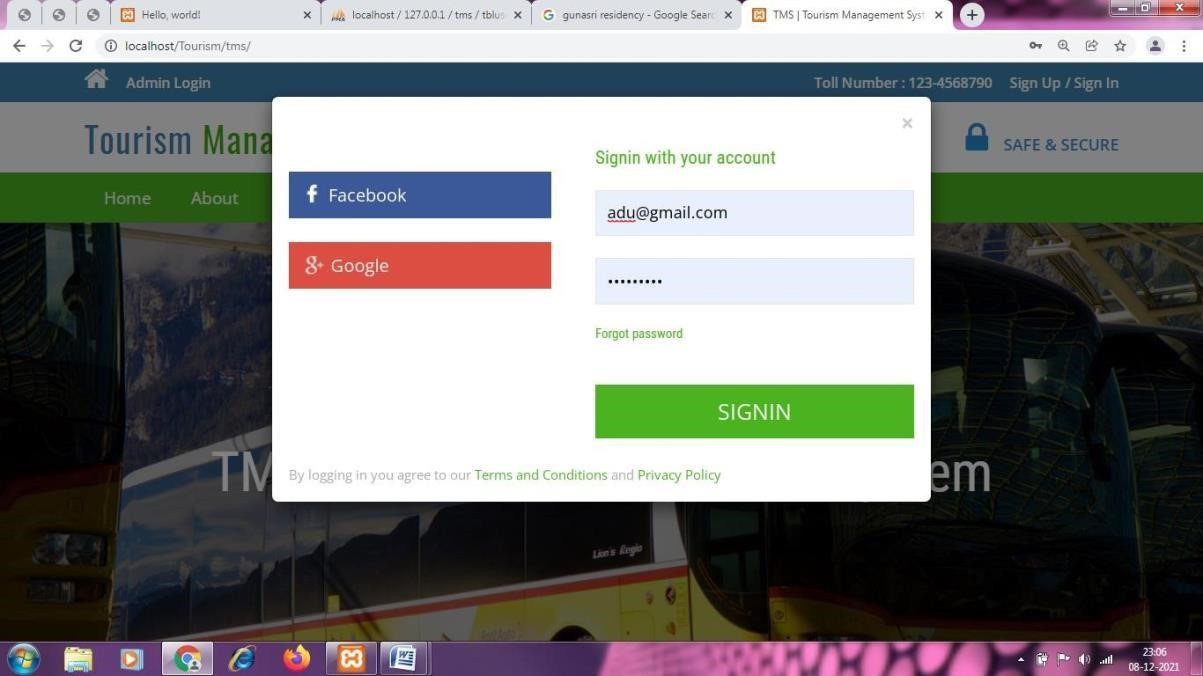
**fig – 2(admin login page )**



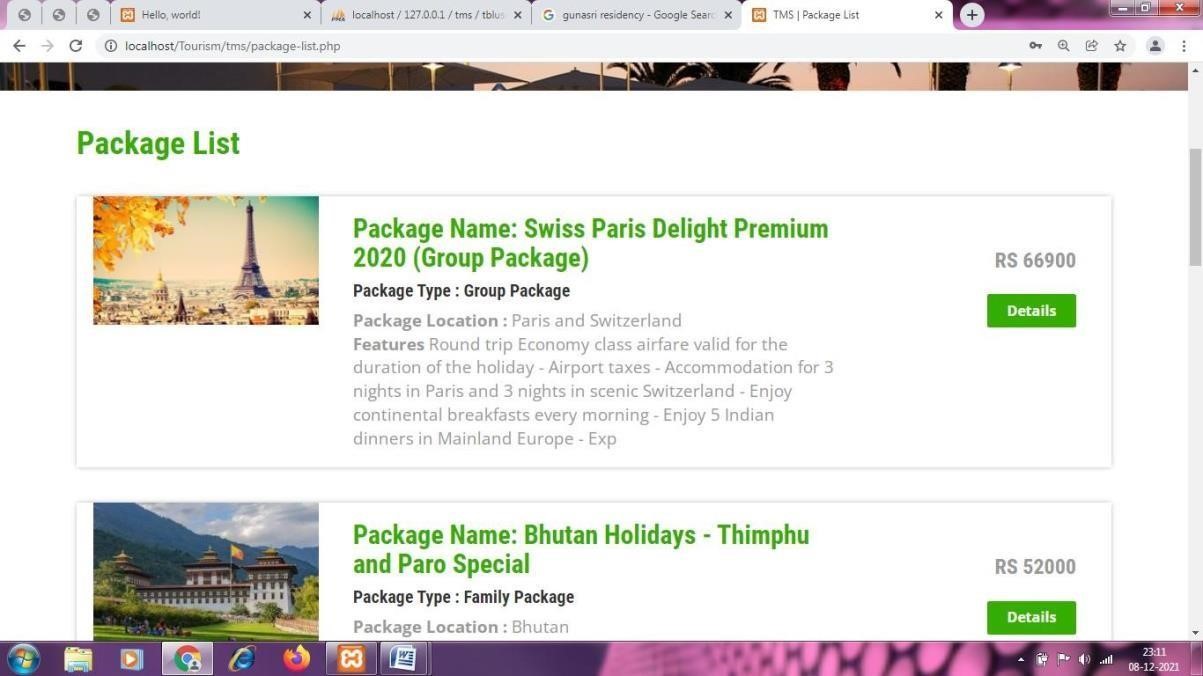
**Fig - 3(DASHBOARD )**



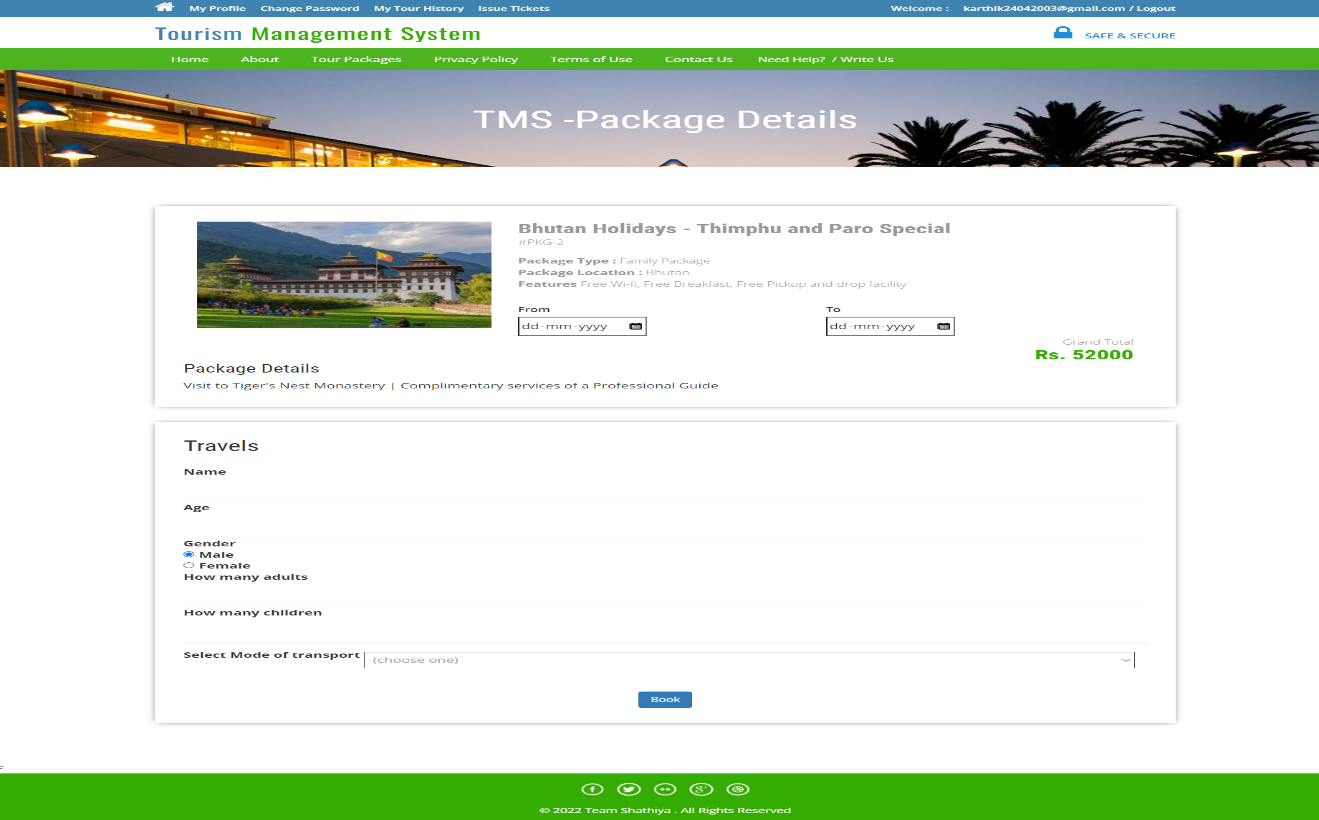
# Fig - 4(USER REGISTRATION PAGE )



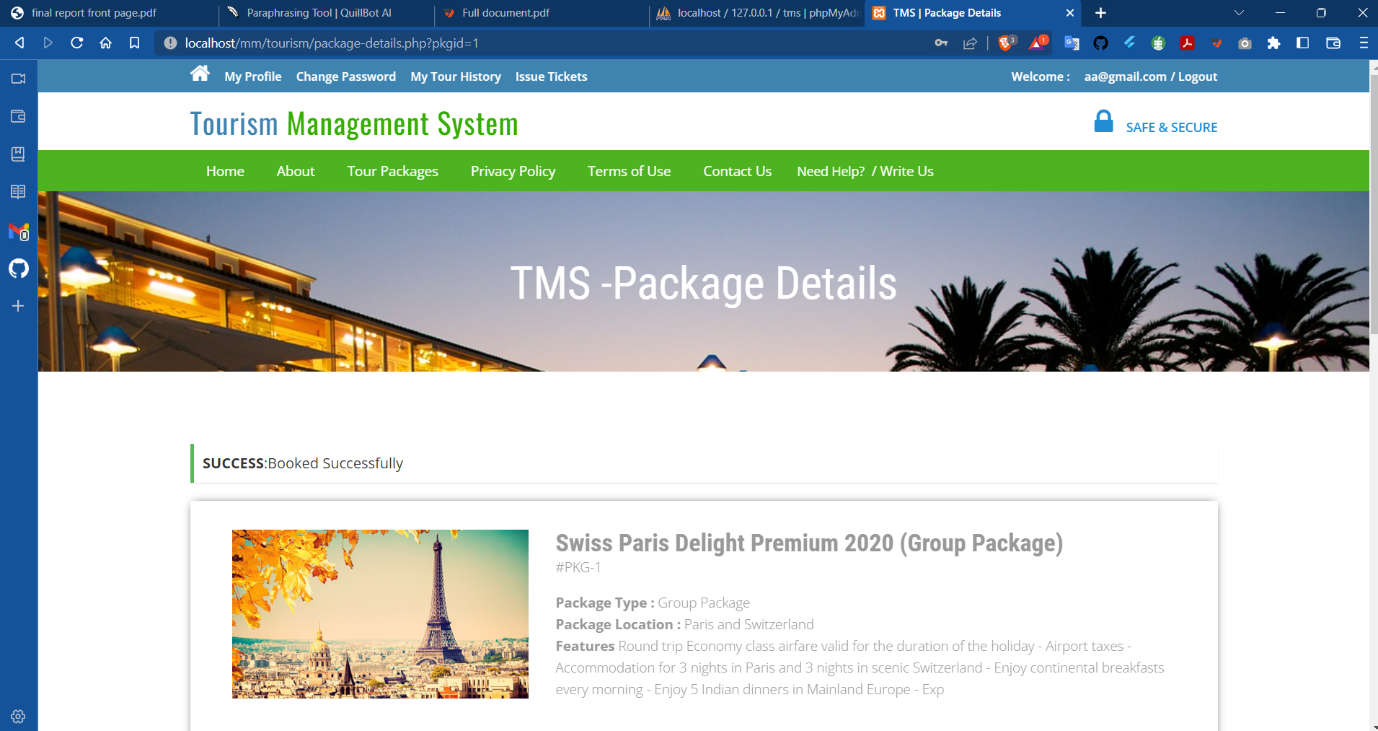
**Fig - 5(USER LOGIN PAGE)**



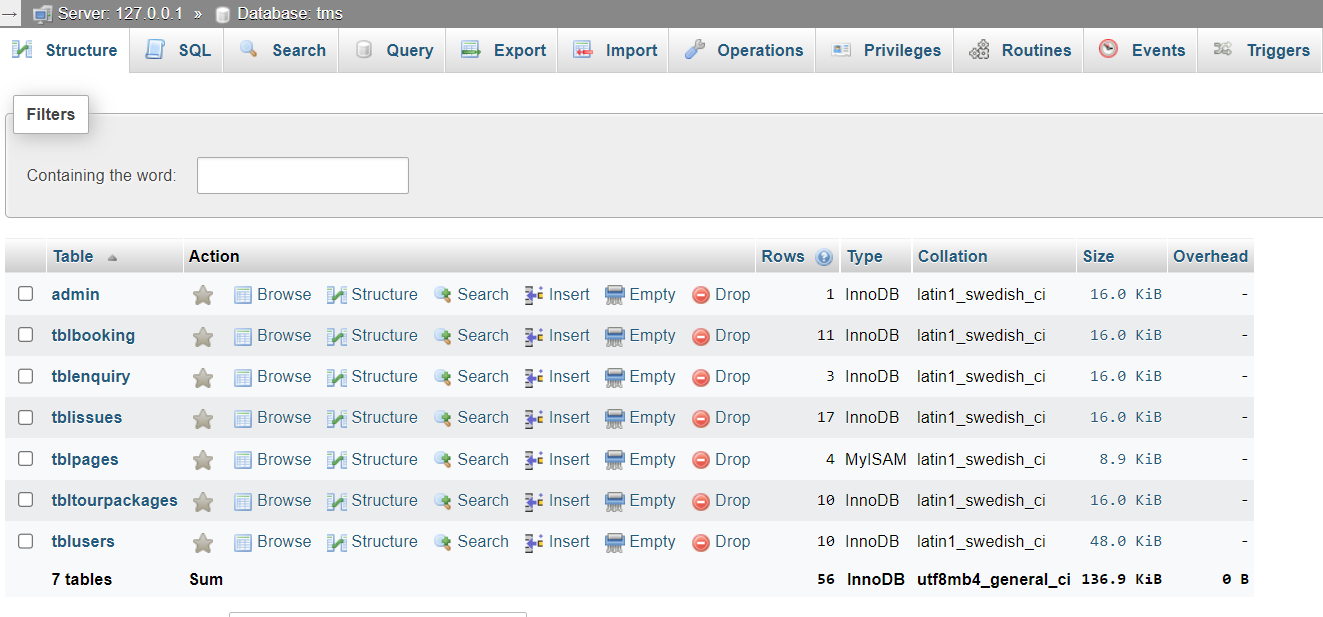
**Fig - 6(PACKAGE LIST)**



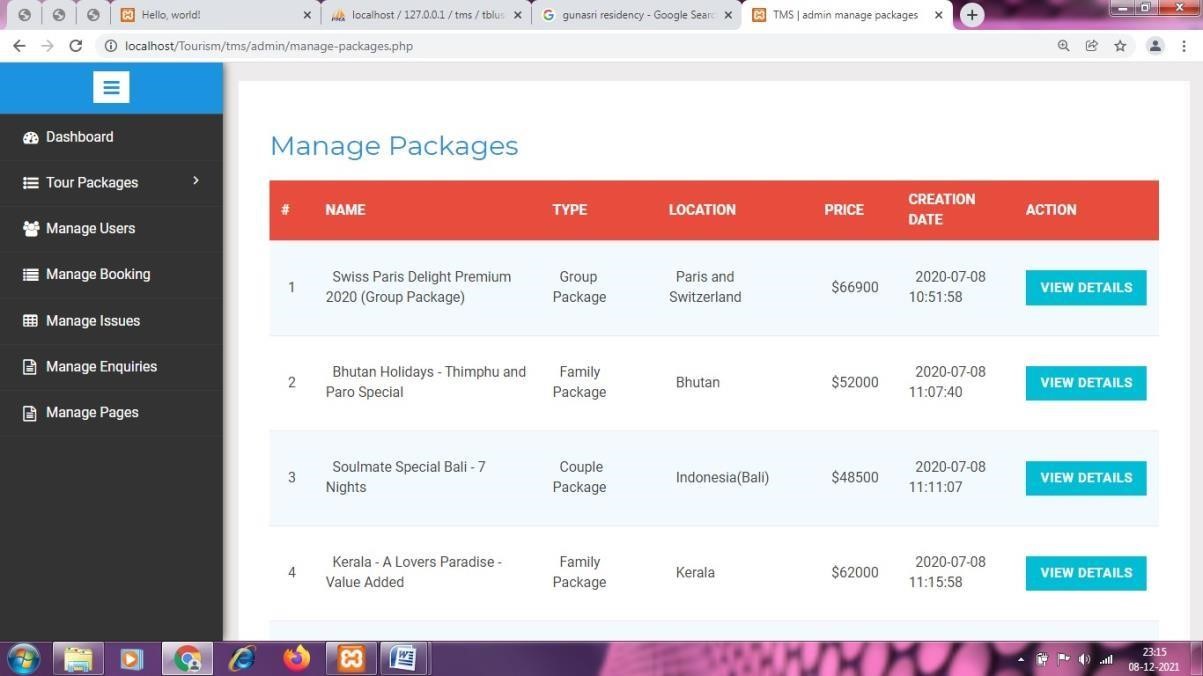
**Fig - 7(PACKAGE BOOKING PAGE)**



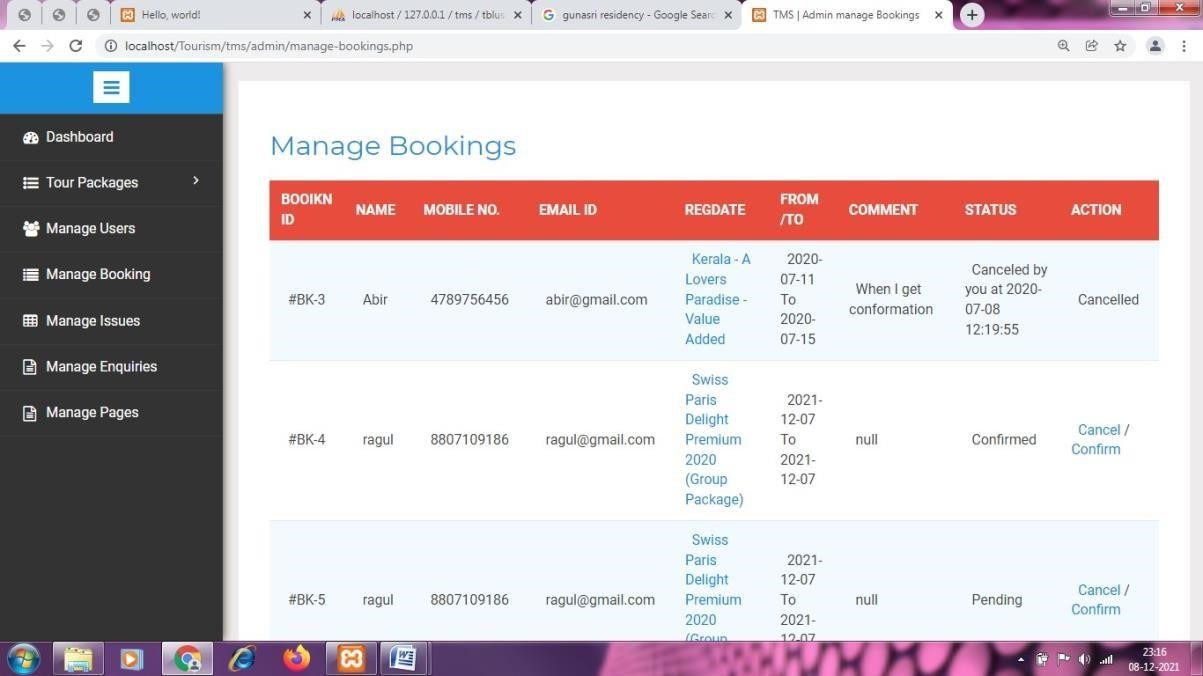
**Fig - 8(BOOKING CONFIRMATION PAGE)**

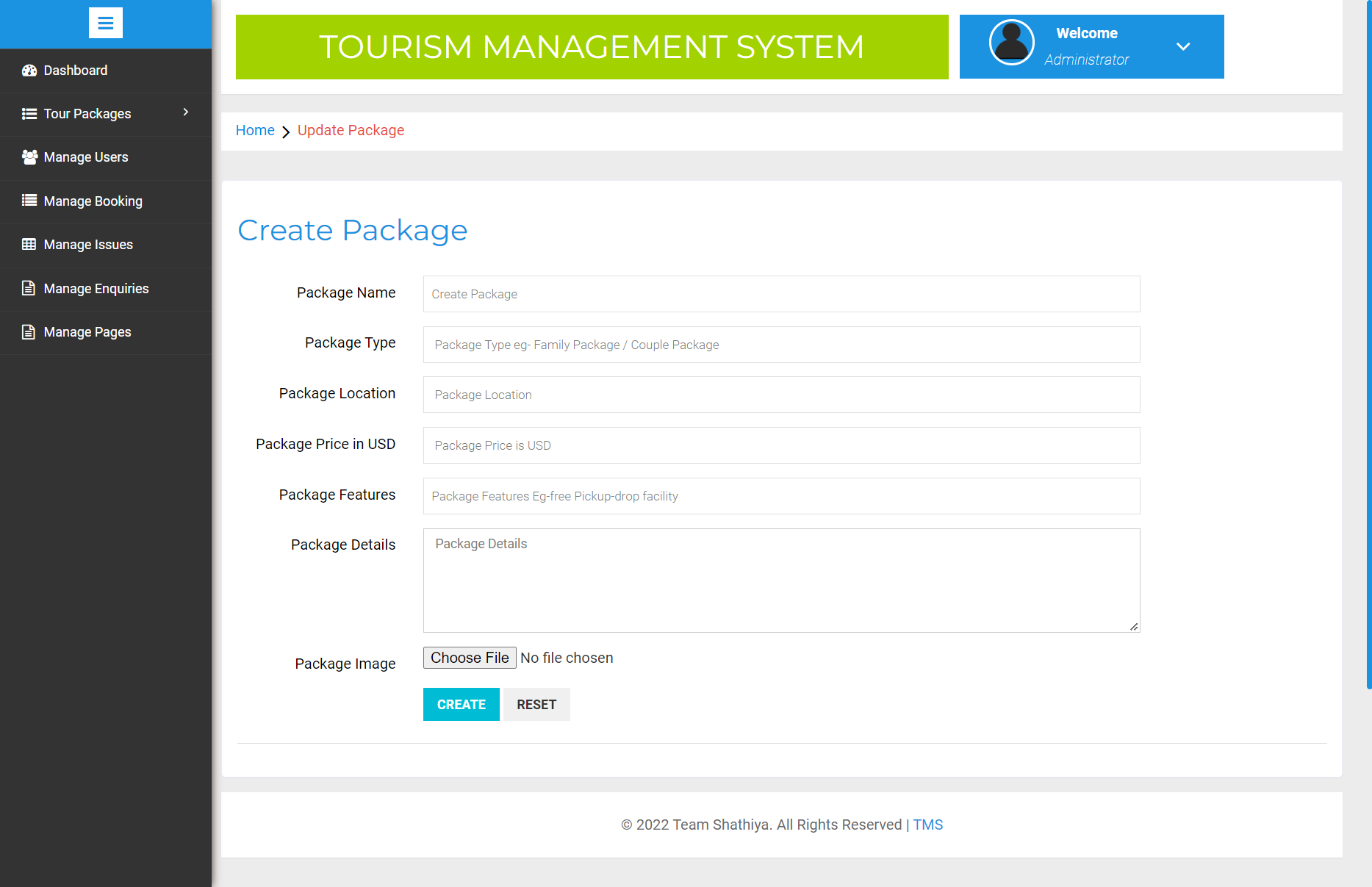


**Fig - 9(Database Design)**



**Fig- 10(MANAGE PACKAGES )**

  
**Fig - 11(MANAGE BOOKING** )

****

**Fig – 12(PACKAGE ADDING PAGE**)

# 5.4 APPENDIX

## MY INDEX

<?php

session\_start();

error\_reporting(0);

include('includes/config.php');

?>

<!DOCTYPE HTML>

<html>

<head>

<title>TMS | Tourism Management System</title>

<meta name="viewport" content="width=device-width, initial-scale=1">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<script type="applijewelleryion/x-javascript"> addEventListener("load", function() { setTimeout(hideURLbar, 0); }, false); function hideURLbar(){ window.scrollTo(0,1); } </script>

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

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

<link href='//fonts.googleapis.com/css?family=Open+Sans:400,700,600' rel='stylesheet' type='text/css'>

<link href='//fonts.googleapis.com/css?family=Roboto+Condensed:400,700,300' rel='stylesheet' type='text/css'>

<link href='//fonts.googleapis.com/css?family=Oswald' rel='stylesheet' type='text/css'>

<link href="css/font-awesome.css" rel="stylesheet">

<!-- Custom Theme files -->

<script src="js/jquery-1.12.0.min.js"></script>

<script src="js/bootstrap.min.js"></script>

<!--animate-->

<link href="css/animate.css" rel="stylesheet" type="text/css" media="all">

<script src="js/wow.min.js"></script>

<script>

new WOW().init();

</script>

<!--//end-animate-->

</head>

<body>

<?php include('includes/header.php');?>

<div class="banner">

<div class="container">

<h1 class="wow zoomIn animated animated" data-wow-delay=".5s" style="visibility: visible; animation-delay: 0.5s; animation-name: zoomIn;"> TMS - Tourism Management System</h1>

</div>

</div>

<!--- rupes ---->

<div class="container">

<div class="rupes">

<div class="col-md-4 rupes-left wow fadeInDown animated animated" data-wow-delay=".5s" style="visibility: visible; animation-delay: 0.5s; animation-name: fadeInDown;">

<div class="rup-left">

<a href="offers.html"><i class="fa fa-RS"></i></a>

</div>

<div class="rup-rgt">

<h3>UP TO 1500 OFF</h3>

<h4><a href="offers.html">TRAVEL SMART</a></h4>

</div>

<div class="clearfix"></div>

</div>

<div class="col-md-4 rupes-left wow fadeInDown animated animated" data-wow-delay=".5s" style="visibility: visible; animation-delay: 0.5s; animation-name: fadeInDown;">

<div class="rup-left">

<a href="offers.html"><i class="fa fa-h-square"></i></a>

</div>

<div class="rup-rgt">

<h3>UP TO 25% OFF</h3>

<h4><a href="offers.html">ON HOTELS ACROSS WORLD</a></h4>

</div>

<div class="clearfix"></div>

</div>

<div class="col-md-4 rupes-left wow fadeInDown animated animated" data-wow-delay=".5s" style="visibility: visible; animation-delay: 0.5s; animation-name: fadeInDown;">

<div class="rup-left">

<a href="offers.html"><i class="fa fa-mobile"></i></a>

</div>

<div class="rup-rgt">

<h3>FLAT 15% off on</h3>

<h4><a href="offers.html">US APP </a></h4>

</div>

<div class="clearfix"></div>

</div>

</div>

</div>

<!--- /rupes ---->

<!---holiday---->

<div class="container">

<div class="holiday">

<h3>Package List</h3>

<?php $sql = "SELECT \* from tbltourpackages order by rand() limit 4";

$query = $dbh->prepare($sql);

$query->execute();

$results=$query->fetchAll(PDO::FETCH\_OBJ);

$cnt=1;

if($query->rowCount() > 0)

{

foreach($results as $result)

{ ?>

<div class="rom-btm">

<div class="col-md-3 room-left wow fadeInLeft animated" data-wow-delay=".5s">

<img src="admin/pacakgeimages/<?php echo htmlentities($result->PackageImage);?>" class="img-responsive" alt="">

</div>

<div class="col-md-6 room-midle wow fadeInUp animated" data-wow-delay=".5s">

<h4>Package Name: <?php echo htmlentities($result->PackageName);?></h4>

<h6>Package Type : <?php echo htmlentities($result->PackageType);?></h6>

<p><b>Package Location :</b> <?php echo htmlentities($result->PackageLocation);?></p>

<p><b>Features</b> <?php echo htmlentities($result->PackageFetures);?></p>

</div>

<div class="col-md-3 room-right wow fadeInRight animated" data-wow-delay=".5s">

<h5>RS <?php echo htmlentities($result->PackagePrice);?></h5>

<a href="package-details.php?pkgid=<?php echo htmlentities($result->PackageId);?>" class="view">Details</a>

</div>

<div class="clearfix"></div>

</div>

<?php }} ?>

<div><a href="package-list.php" class="view">View More Packages</a></div>

</div>

<div class="clearfix"></div>

</div>

<!--- routes ---->

<div class="routes">

<div class="container">

<div class="col-md-4 routes-left wow fadeInRight animated" data-wow-delay=".5s">

<div class="rou-left">

<a href="#"><i class="glyphicon glyphicon-list-alt"></i></a>

</div>

<div class="rou-rgt wow fadeInDown animated" data-wow-delay=".5s">

<h3>80000</h3>

<p>Enquiries</p>

</div>

<div class="clearfix"></div>

</div>

<div class="col-md-4 routes-left">

<div class="rou-left">

<a href="#"><i class="fa fa-user"></i></a>

</div>

<div class="rou-rgt">

<h3>1900</h3>

<p>Registered users</p>

</div>

<div class="clearfix"></div>

</div>

<div class="col-md-4 routes-left wow fadeInRight animated" data-wow-delay=".5s">

<div class="rou-left">

<a href="#"><i class="fa fa-ticket"></i></a>

</div>

<div class="rou-rgt">

<h3>7,00,00,000+</h3>

<p>Bookings</p>

</div>

<div class="clearfix"></div>

</div>

<div class="clearfix"></div>

</div>

</div>

<?php include('includes/footer.php');?>

<!-- signup -->

<?php include('includes/signup.php');?>

<!-- //signu -->

<!-- signin -->

<?php include('includes/signin.php');?>

<!-- //signin -->

<!-- write us -->

<?php include('includes/write-us.php');?>

<!-- //write us -->

</body>

</html>