**TABLE OF CONTENTS**

**SL. NO. CHAPTERS PAGE NO.**

1. **Introduction** 
   1. Architecture of DBMS 1
   2. Overview of project 3
      1. Problem statement 3
      2. Overview 3
      3. Objectives 3
      4. Mission 4
2. **System Design and Methodology** 
   1. System Architecture 5
   2. Project Requirement 5
   3. ER Diagram 6
   4. Schema Diagram 8
   5. Algorithm 9
3. **System Implementation** 
   1. Module Description 10
4. **Results and Screenshots 13**

**Conclusion 17**

**Bibliography 18**

**LIST OF FIGURES AND TABLES**

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **Title of figure** | **Page no.** |
| 1 | Three Schema diagram | 2 |
| 2 | ER Diagram | 6 |
| 3 | Schema Diagram | 8 |
| 4 | Admin Login Page | 12 |
| 5 | User Login Page | 12 |
| 6 | User Registration Page | 13 |
| 7 | Destination Selection Page | 13 |
| 8 | Package Selection Page | 14 |
| 9 | Details Page | 14 |
| 10 | Profile1 Page | 15 |
| 11 | Profile2 Page | 15 |

**CHAPTER 1**

**INTRODUCTION**

**1.1 ARCHITECTURE**

Database is a collection of related data. DBMS came into existence in 1960 by Charles. Again in 1960 IBM brought IMS-Information management system. In 1970 Edgor Codd at IBM came with new database called RDBMS. In 1980 then came SQL Architecture- Structure Query Language. In 1980 to 1990 there ere advances in DBMS e.g. DB2, ORACLE. A database has the following implicit properties:

* A database represents some aspect of the real world, sometimes called the mini world or the universe of discourse (UoD). Changes to the mini world are reflected in the database.
* A database is a logically coherent collection of data with some inherent meaning. A random assortment of data cannot correctly be referred to as a database
* A database is designed, built, and populated with data for a specific purpose. It has an intended group of users and some preconceived applications in which these users are interested.

In other word, a database some source from which data is derived, some degree of interaction with events in the real world, and an audience that is actively interested in its contents.

Metadata (meta data, or sometimes meta information) is “data about data”, of any sort in any media. An item of metadata may describe a collection of data including multiple content items and hierarchical levels, for example a database schema. In data processing, metadata is definitional data that provides information about or documentation of other data managed within an application or environment. The term should be used with caution as all data is about something and is therefore metadata.

A database management system (DBMS) is a collection of programs that enables users to create and maintain database. The DBMS is a general-purpose software system that facilitates the process of defining, constructing, manipulating and sharing databases among various users and applications.

Defining a database specifying the database involves specifying the data types, constraints and structures of the data to be stored in the database. The descriptive information is also stored in the database in the form database catalogue or dictionary; it is called meta-data. Manipulating the data includes the querying the database to retrieve the specific data. An application program accesses the database by sending the queries or requests for data to DBMS. The important function provided by the DBMS includes protecting the database and maintain the database.

****

**Fig 1.1: Three Schema Architecture**

The figure 1.1 shows the Three schema architecture of Database Management System. The Three schema architecture consists of three levels of the architecture:

* **External Level:**

The external level is the view that the individual user of the database has. This view is often a restricted view of the database and the same database may provide a number of different views for different classes of users. In general, the end users and even the application programmers are only interested in a subset of the database.

* **Conceptual Level:**

The conceptual view is the information model of the enterprise and contains the view of the whole enterprise without any concern for the physical implementation. The conceptual view is the overall community view of the database and it includes all the information that is going to be represented in the database.

* **Internal Level:**

the internal view is the view about the actual physical storage of data. It describes what data is stored in database and how.

**1.2 OVERVIEW OF THE PROJECT**

**1.2.1 PROBLEM STATEMENT**

To maintain and manipulate the data of details about travel using travel agency database management system

**1.2.2 OVERVIEW**

The travel agency database management system creates software that stores and manages all the data needed for applying a tour packages across the world. It describes administration, application users and the structure of details about their packages, duration of the tour, total cost etc.

It has a database administration that has access to the entire database, in regards with viewing and update of information. End users can view all their personal data. It can also edit their own data using authorization access. The data can be viewed, accessed, manipulated and retrieved very easily. The data is well protected for any further operation on them.

**1.2.3 OBJECTIVES**

TRAVELHILLS’s objectives for the users include:

* The creation of unique, innovative and customized tour packages that will differentiate from other sources
* The formation of company’s image that will bring people with diverse interests and backgrounds together in a common forum.
* To serve its customer with best possible quality and in all efficient way possible including providing them more things in price on one.

**1.2.4 MISSION**

We want to serve our customers with best possible services and provide them the kind of comfort they want. We would also want to customize our tours as per our customer’s needs without reducing joy or quality of tour. People of all ages and backgrounds will come to enjoy the unique, upscale, joyful, and refreshing environment that TRAVELLHILLS provides.

**CHAPTER 2**

**SYSTEM DESIGN AND METHODLOGY**

**2.1 SYSTEM ARCHITECTURE**

The main softwares used for this are HTML5, PHP and WampServer**.**

**PHP: Hypertext Preprocessor** is server language designed for web development and also used in a general purpose programming language. PHP code may be embedded into HTML code or it can be used in combination with various web templates system, web content management system, and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a common gateway interfaceexecutable. The web server combines the results of the interpreter and executed PHP code, which may be any type of data including images with the generated web pages. The PHP interpreter only executes PHP code within its delimiter. Anything outside its delimiter is processed by PHP, although non-PHP text is still subject to control structures described in PHP code. The most common delimiters are *<?php* to open and *?>* to close PHP section. The shortened from <? exists.

**HTML: Hypertext Markup Language** is a markup language for creating a webpage. Webpages are usually viewed in a web browser. They include writing, links, pictures, and even sound and videos. HTML is used to mark and describe each of these kinds of content so the web browser can display them correctly. HTML can also be used to add meta information to a webpage. Meta information is usually shown by web browsers and is data about the web page, e.g. the name of the person who created the page. Cascading style sheets CSS is used to style HTML elements while JavaScript is used to maintain HTML element and CSS styles.

**2.2 PROJECT REQUIREMENT**

**Hardware requirement:**

* Pentium i3, i5 or more processor
* 2GB or higher RAM
* 10GB Disk Space

**Software Requirement:**

* Windows 7,8,10 Operating System
* Internet Explorer 9 and above,
* Google Chrome Latest version

**2.3 ENTITY RELATIONSHIP DIAGRAM**

books

**BOOKING**

**LOGIN**

**LO**

**PACKAGEDETAILS**

pays

**FARES**

**ADMIN**

manages

Selects from

**Fig 2.1: ER diagram of travel agency**

An Entity-Relationship Diagram (ERD) is a data modelling technique that graphically illustrates an information system’s entities and the relationships between those entities. An ERD is a conceptual and representation model of data used to represent the entity framework infrastructure. Fig 2.1 shows the ER diagram of travel agency which includes the following attribute:

* The admin entity has adminid, password as attributes where adminid is primary key.
* The login entity has fname, lname, phcode, mobile, dob, email, type, pass as attributes where email is primary key.
* The booking entity has id, email, package, adultno, childno, dotravel, mode as attributes where id and email are primary key.
* The packagedetails entity has distance, package, places, days as attributes.
* The fares entity has package and cost as attributes.
* The comments have name, email, mobile, comment as attributes where email is primary key.

**2.4 SCHEMA DIAGRAM**

**ADMIN**

|  |  |
| --- | --- |
| **Admin id** | **password** |

**LOGIN**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **fname** | **lname** | **phcode** | **mobile** | **dob** | **email** | **type** | **pass** |

**BOOKING**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **id** | **email** | **package** | **adultno** | **childno** | **dotravel** | **mode** |

**PACKAGEDETAILS**

|  |  |  |  |
| --- | --- | --- | --- |
| **distance** | **package** | **places** | **days** |

**FARES**

|  |  |
| --- | --- |
| **package** | **cost** |

**COMMENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **email** | **mobile** | **comments** |

**Fig 2.2: Schema diagram of travel agency**

**2.5 ALGORITHM**

**TRIGGER:**

To store details of newly registered users and users who deleted their accounts.

**STEP 1**: BEGIN

**STEP 2**: After INSERT on LOGIN table

**STEP 3**: If Registration successful OR user account DELETED

INSERT INTO LOGINLOG VALUES(NEW.EMAIL,NEW.FNAME,”OPERATION”, NOW())

**STEP 4**: END

\*OPERATION=REGISTERED/DELETED

**STORED PROCEDURE:**

To Display existing Booking details from booking table in Admin Page.

**STEP 1**: BEGIN

**STEP 2**: Create Procedure selection()

**STEP 3**: SELECT \* from booking;

**STEP 4**: Display Booking details in admin page

**STEP 5**:END

**CHAPTER 3**

**SYSTEM IMPLEMENTATION**

**3.1 MODULE DESCRIPTION**

This project includes following modules:

**HOMEPAGE**

**INPUT:**

The user can select user login or admin login provided on home page.

**OUTPUT:**

On selecting

* user login, the user is provided with user login page
* admin login, the user is provided with admin login page

**DESCRIPTION:**

Front end is designed using HTML and CSS. The user is taken to selected login page. The user can now enter the details that are being proposed for further actions on application and on successful operation he can perform more operations subsequently. If the user does not possess any account, he or she can create their particular account by choosing sign up option provided on page. There, the user can fill their personal details and can have freedom to avail the services by becoming a member of this prestige service.

**ADMIN LOGIN**

**INPUT:**

Username and password

**OUTPUT:**

A successful loginshall take the admin to his respective page. On entering wrong details, it displays an error message.

**DESCRIPTION:**

Front end is designed using HTML and CSS. On logging in to the admin page, the admin can perform operations which are authorized to the admin. They are as follows:

* Admin can view all the booking details regarding a person’s plan.
* Admin can also edit or include several plans with some benefits.

**USER LOGIN**

**INPUT:**

User Id and password

**OUTPUT:**

Upon entering correct user id and password, the user will be successfully login in to the application. In case the user has entered either wrong user id or password, it would provide an error message stating wrong password or id.

**DESCRIPTION:**

Front end is designed using HTML and CSS. On logging in to the user page successfully, the user can perform operations which are authorized to the particular user. They are as follows:

* He can view all available packages for different places across the country.
* He can also comment about their experience using the services.
* He can apply for his suitable plan for tour.
* He can edit the details regarding the tour in case of change in plan.
* He can also delete the previously booked plan.

**USER SIGNUP**

**INPUT:**

First name, last name, mobile no, date of birth, email, password and gender

**OUTPUT:**

Upon entering all the required fields in sign up page, the user can submit the details. Once the registration is successful, the user can enjoy all the services provided by the application.

**DESCRIPTION:**

Front end is designed using HTML and CSS. On successful registration, the user can login in user login page using personal account and can apply for packages, edit and delete the details.

**BOOKING**

**INPUT:**

Package name, email, number of adults, number of children, date of booking and transport mode

**OUTPUT:**

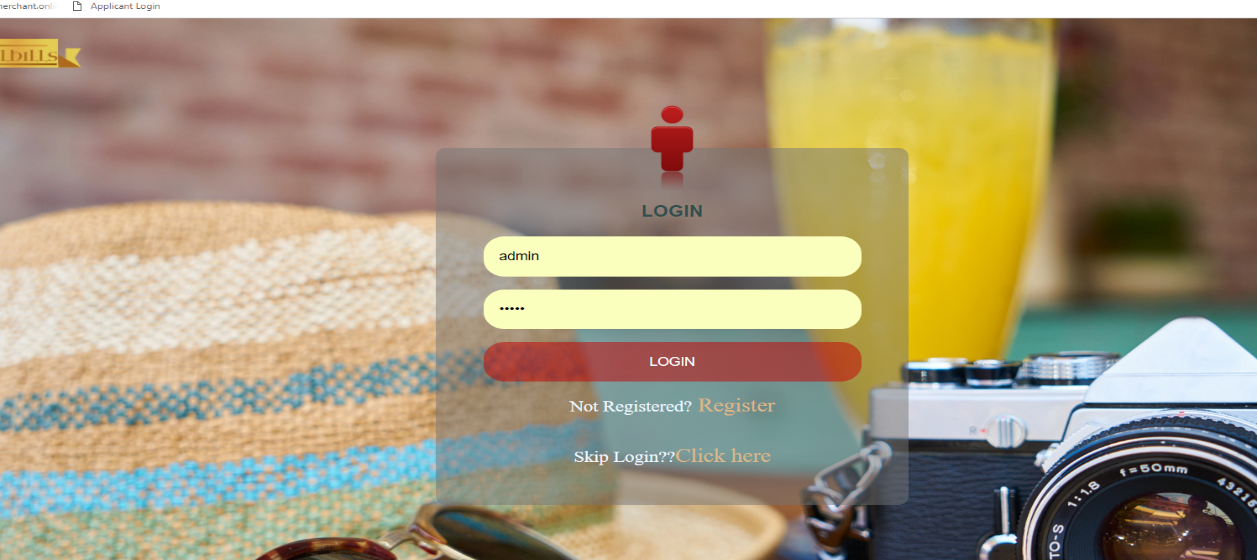
Upon entering all the above mention details such as package name, email, number of adults, number of children, date of booking and transport mode, it would give a message stating that booking is confirmed. Later the user can view all their booking details and can edit or delete too.

**DESCRIPTION:**

Front end is designed using HTML and CSS. The user can go to their profile page and check their booking details. If the user wants a change in their booking, they can do so in their profile using edit option provided. The user can also remove a tour plan from their account using delete option in their profile.

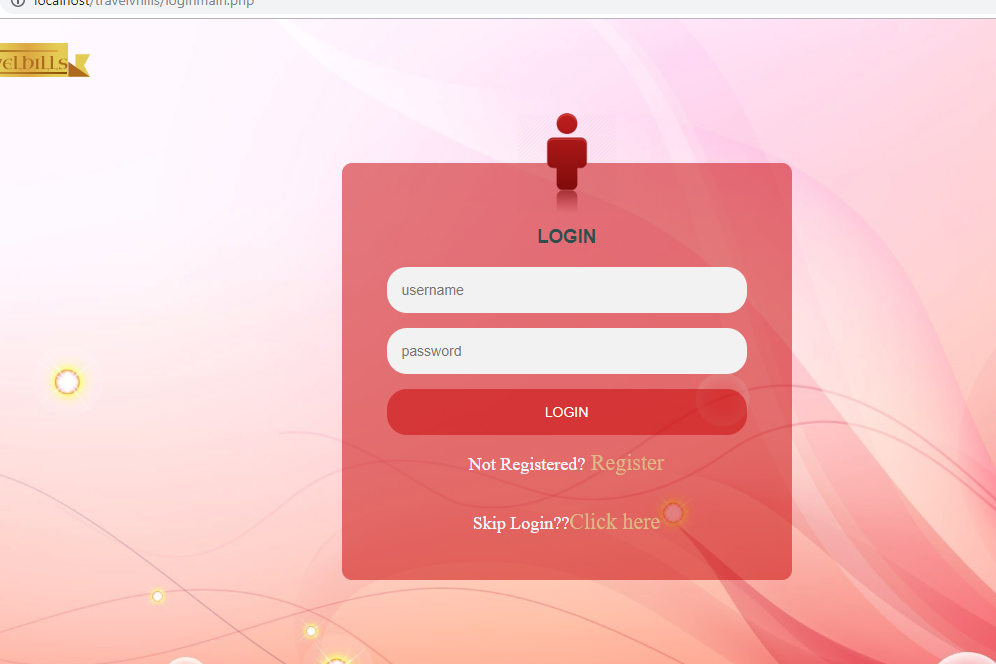
**CHAPTER 4**

**RESULTS AND SCREENSHOTS**

****

**Fig 4.1: Admin Login Page**

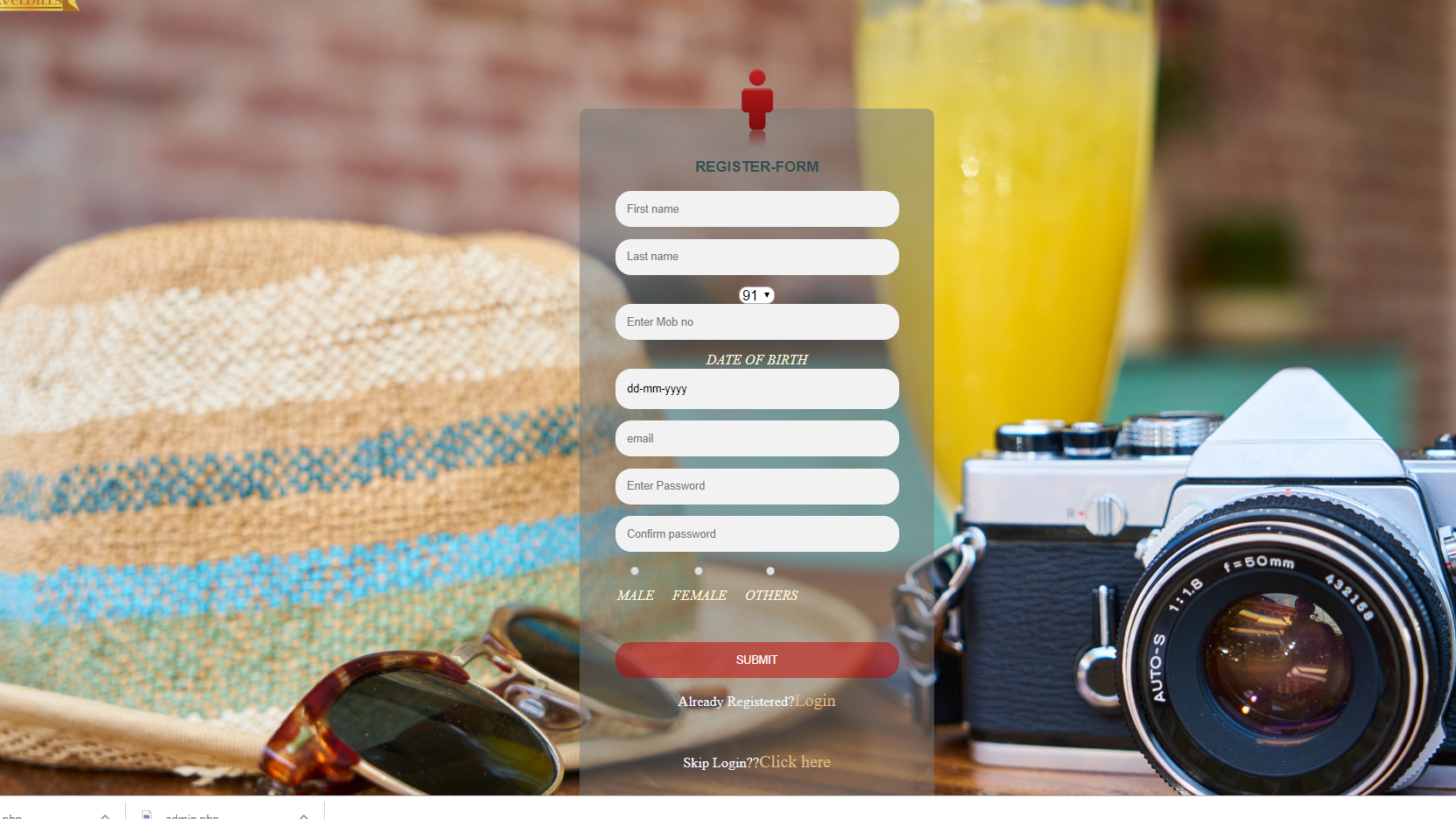
Admin page: This page is displayed when users select admin login in homepage.



**Fig 4.2: User Login Page**

User page: This page is used by users to login to their particular account.

.



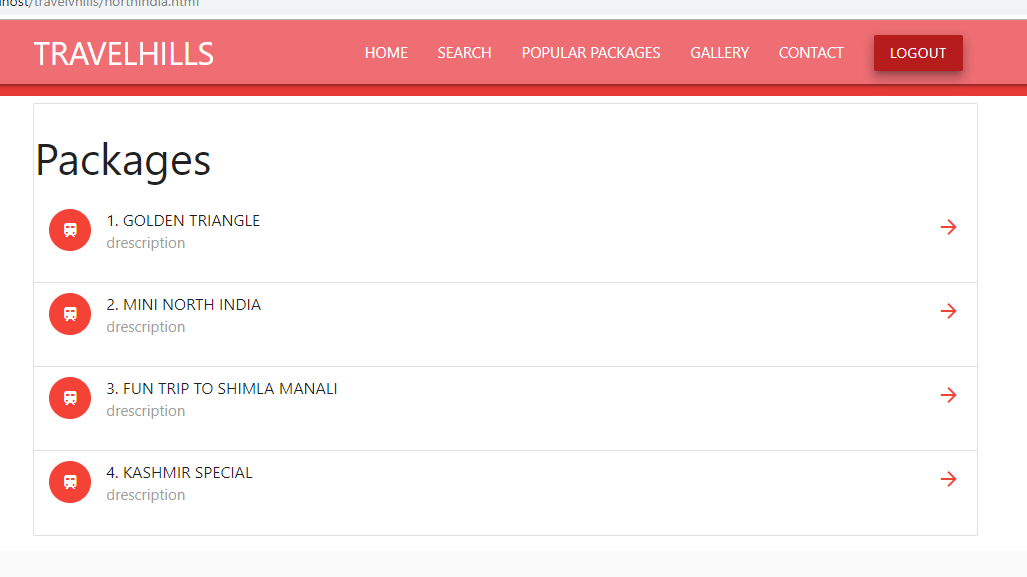
**Fig 4.3: Users Registration page**

Registration Page: This page allows the users to create their own account by filling their personal details.



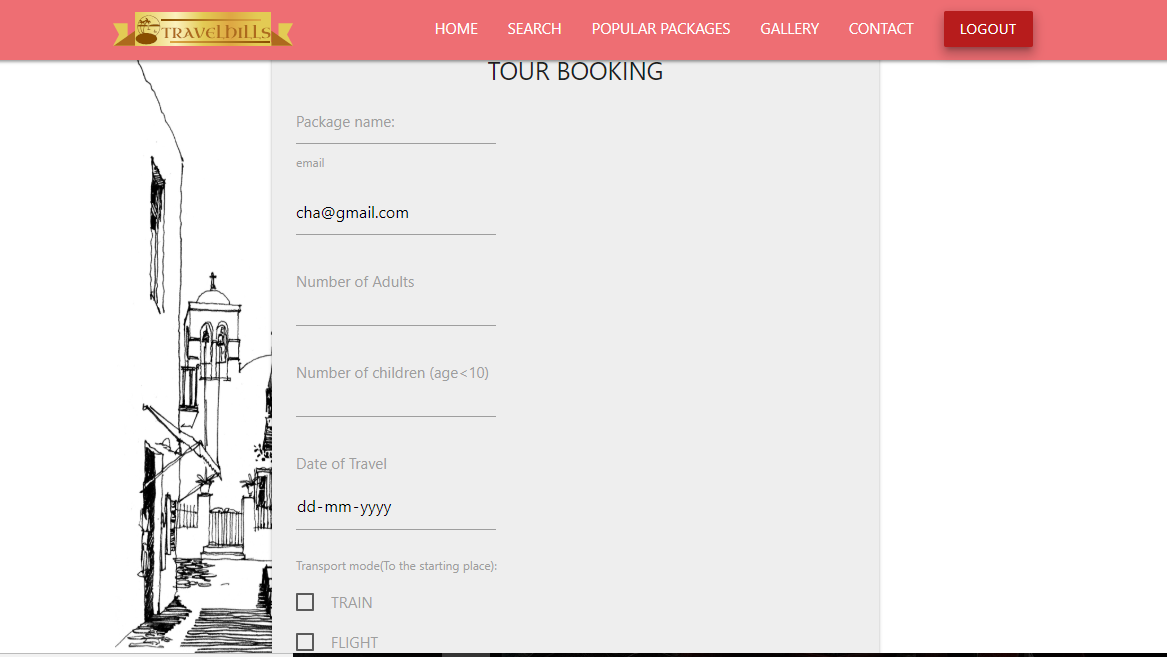
**Fig 4.4:** **Destination Selection Page**

Destination Selection: This page allows users to select a place.



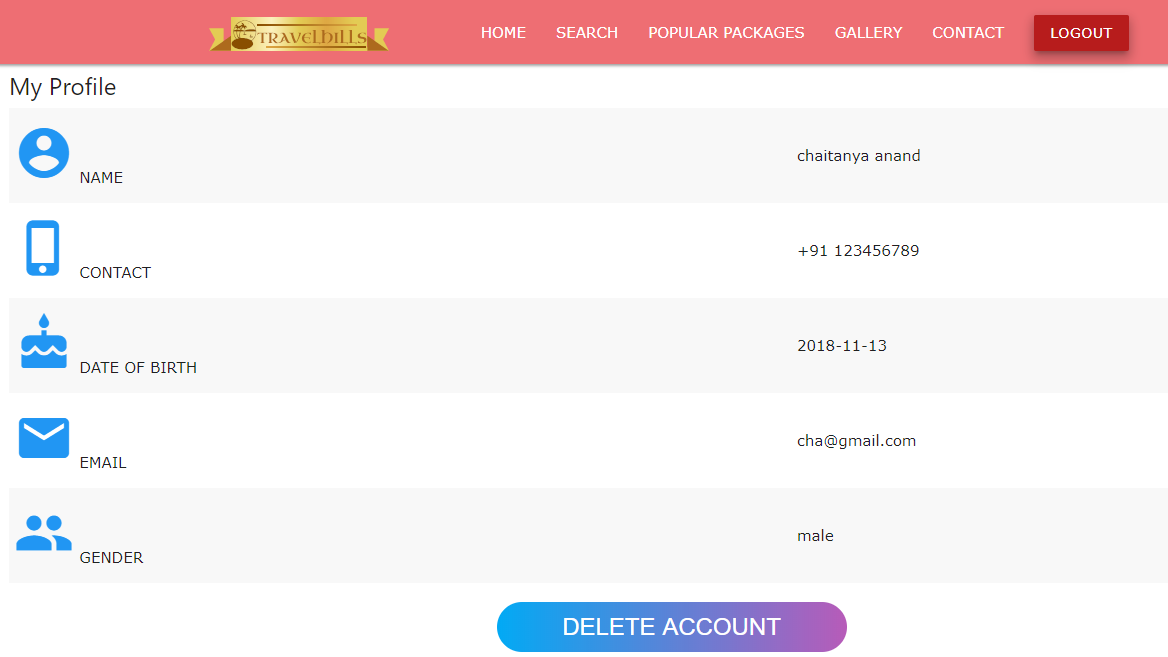
**Fig. 4.5: Package Selection Page**

Package Selection: This allows users to select a particular package in selected place.



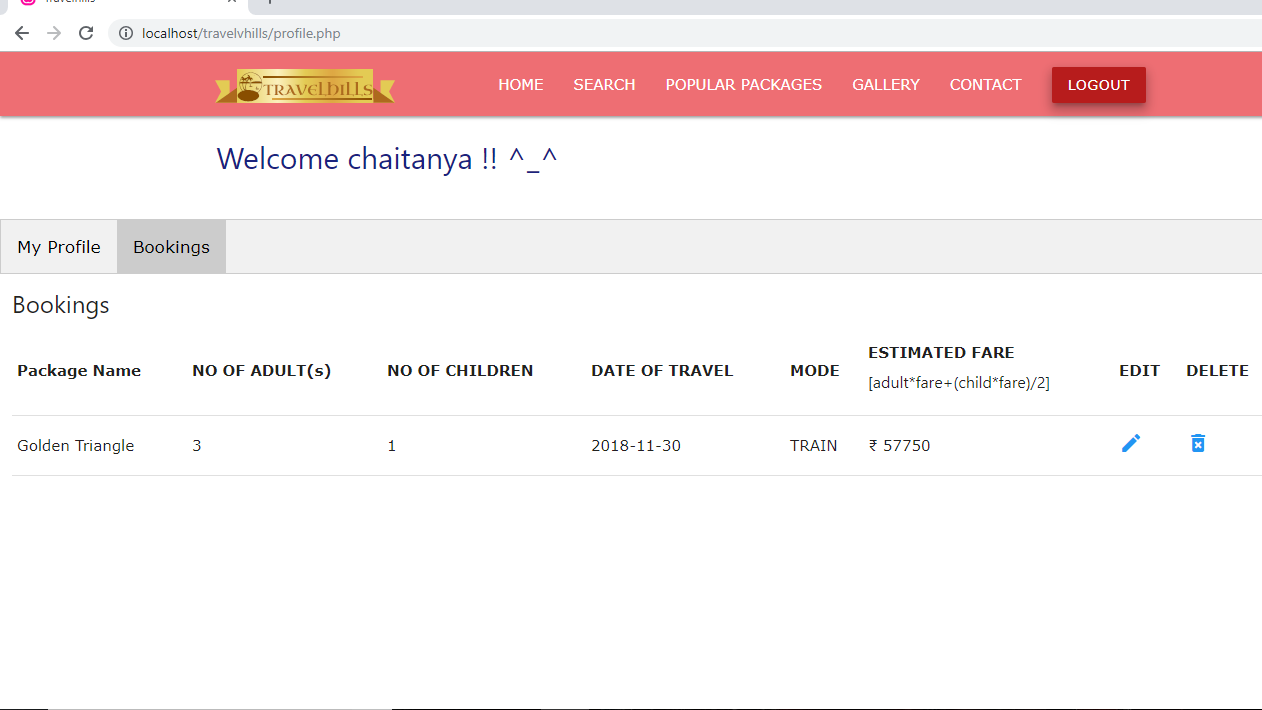
**Fig 4.6: Details page**

Details: This page is for users to fill the details regarding his or her tour booking.



**Fig 4.7: Profile1 page**

Profile1: This page gives the details about account holder and it also allows users to delete the account.



**Fig 4.8: Profile2 page**

Profile2: This page gives the details regarding all the booking done by the user. Here, the user can also edit or delete the booking.

**CONCLUSION**

An application has been developed using HTML and CSS for front end and Wamp server for back end where front end and backend has been connected using PHP. It is to provide user friendly services to meet the requirements of users. Here, the data can be accessed, manipulated and retrieved very easily.

It also calculates the total fare for a tour package based on number of adults and number of child which is to be paid by users. Hence, this application provides user friendly interface.

**FUTURE WORKS**

* The project can include packages for other parts of country along with overseas.
* The project can include different payment mode.
* The project can include the confirmation messages to user’s registered mobile number.
* The project can also include the function of search bar which can search and list the required places.

**BIBLIOGRAPHY**

* [www.stackoveflow.com](http://www.stackoveflow.com)
* [www.youtube.com](http://www.youtube.com)
* [www.scribd.com](http://www.scribd.com)
* [www.csmaterialze.com](http://www.csmaterialze.com)
* www.w3schools.com