**SYNOPSIS**

**CHAPTER 1**

**SYNOPSIS**

* 1. **Title of the project**

**THE TRAVELLER’S GUIDE**

* 1. **Objectives of the project**
* Strength and strain of manual labor can be reduced
* To provide user friendly interface
* High Security
* Quick data retrieval and large amount of data storing
* To store the bus, flight, train, hotels, tour package records offered by companies, allowing customers to choose from wide range of services
* To store customer details
* Saving customers time as he can book the ticket online
* Reducing the amount the customer spends for travelling as he doesn’t have to pay commission
* To reduce data redundancy to some extent
  1. **Project Category**
* RDBMS (Relational Database Management System)
* Dynamic Web Based Applications
  1. **Languages used**

Front End: PHP 5.0

Back End: MySQL

* 1. **Structure of the project**

This system can be used to store the details of the bus, flight, hotels, trains, tour packages and it provides easy interface for updating the information about them. All the details are maintained in the database

* + 1. **Analysis**

In today’s environment, people love to travel, be it going to the nearest holiday spot or on a tour around the world. Each and every country has something special in it, which makes people want to visit. Travel agencies and operators know this fact and try to take advantage of it by increasing the travel charges. It is also very difficult to get some time to go out and book the tickets and check for its confirmation ourselves, due to our busy life schedule. In order to solve this, B2C websites have been started. Now unlike the olden days, where the customers had to go to the travel agency to book a ticket and to check it’s schedule, The Traveller’s Guide, allows its clients to book ticket without wasting their time(to go to the nearest travel agency).Customers here can book tickets and also check its confirmation online. This does not only saves time but also money as they don’t have to spend extra on commission. The system provides varied range of services for the clients allowing them to book flight tickets, bus tickets, train tickets and also reserving hotel accommodations and other holiday packages.

* + 1. **Advantage of the newly proposed system**
* Choose services faster and easier at one place.
* Saves time of travelling to the travel agency.
* Good/Trusted & Tension free services. Once the ticket is booked, customer can take its print out any time before the journey.
* Alerts and real time reporting through e-mails.
* Admin can track booking report, billing report etc.
* Reports generated can be saved for future references..
* It allows multiple bookings in flights, bus, train, tour packages and hotels.
  + 1. **Module Description**

This project consists of several modules. They are:-

* **Flight ticket booking:**

In this module the customer can search source-destination, check flight timings and book a ticket online.

* Search flight ticket.
* View flights and flight schedules.
* Flight information.
* Make a ticket booking online and print ticket.
* Cancellation.
* **Bus ticket reservation module:**

In bus ticket reservation module, there is a collection of buses; customer can book the ticket by selecting agency, date and departure time of the bus.

* Search bus type.
* Advance bus ticket booking.
* Check Departure time and arrival time.
* Print E- Ticket.
* Cancellation

.

* **Train ticket reservation module:**

In train reservation module customer can check train scheduled timings and can book tickets online by entering place, date and time.

* Check seat availability.
* Fare enquiry.
* Online reservation.
* Search Train/Fare availability details.
* Train schedule details.
* Train type information.
* Cancellation.
* **Hotel reservation:**

In this module, customers will be provided with an option to search hotels. Customer can reserve the hotels by selecting check-in date, checkout date and hotel name.

* Search hotels by entering city.
* Find hotels and room types.
* Add check-in date and check-out date.
* Book or reserve domestic and international hotels.
* Cancellation.
* **Holiday package :**

Here in this module, customer can search tour packages by entering destination, duration of travel, month of travel and number of person.

* Holiday packages based on vacation.
* View holiday package price details, and package details.
* Book domestic and International holiday packages.
* Cancellation.
* **Payment module:**

This module allows customers to make payment after the reservation. The customer will receive ticket receipt by mail after the payment.

* Make payment to selected service.
* Generate billing receipt.
* Update credit card details.
* View payment details.
* **My Account:**

This module is for customers where customer can access account module after login id and password. This module will display booked ticket, Billing receipt, ticket cancellation page, logout, etc.

* Display booked receipt.
* Change password.
* Update profile information.
* **Cancel booking:**

Customer can cancel flight ticket, bus ticket, train ticket, or hotel bookings. The system refunds paid amount the after deducting cancellation charges.

* Cancel booking details.
* Cancellation policy.
* Refundable and non-refundable booking amount.
* **Dashboard module:**

This module allows the administrator of this site to manage the different aspects of ticket reservation related activities. The administrator of the site can add or modify the records of flights, bus, train, holiday packages, etc.

* Add, edit, view flight records.
* Add, edit, and view bus agency and buses.
* Add, edit, view train details.
* Add, edit, view hotel information.
* Add, edit, view holiday packages.
* View bookings, View cancellation report.
* View payment report.
* Add users.
* Change password.
  1. **Future Scope and enhancement of the project**
* In future we can build a mobile application for smartphones.
* Currently the system supports only desktop pc and in future we can make it mobile browser compatible site.
* The project can be made more users friendly in future.

**LITERATURE SURVEY**

**CHAPTER 2**

**LITERATURE SURVEY**

* 1. **PHP**
* PHP stands for hypertext Preprocessor, a high-level server side scripting language used by a range of organizations and developers, specially designed for web applications.
* PHP code is embedded in HTML documents, as is the case with JavaScript.
* PHP is open source software and free to download and use.
* PHP files can contain text, HTML tags and scripts.
* PHP files can have a file extension of “.php”, “.php3”, or “.phtml”.
* PHP is installed on more than 20 million websites and 1 million web servers.
* In addition to PHP’s capability as a web scripting language, PHP also can be used as a shell scripting language. The means that you can use a single language to write web Applications and create shell scripts to manage your computers. You can even use PHP for creating desktop applications.
  1. **MySQL**

A good database design is crucial for a high performance application, just like an aerodynamic body is important to a race car. If the car doesn’t have smooth lines, it will produce drag and go slower. Without optimized relationship, database won’t perform as efficiently as possible. MySQL is a powerful and comprehensive relational database server, which was originally developed by David Axmark, Allan Larsson and Micheal “Montty” widenius. Although the MySQL software originated as an open source project, its creators were confident that they could run a business using the product as a base. MySQL has enjoyed enormous popularity, and its customers include Yahoo! Finance, MP3.com, Motorola, NASA, Silicon Graphics and Texas Instruments.

MySQL is a full-featured database and uses open stands, such as the ANSI SQL 99 standard, for communicating with databases with Structured Query Language (SQL). This standard provides a means to insert, update and query information in the database by using an industry standard language. This standard language is used across database product and like other products. MySQL supports a number of additional SQL statements. As well as being a standardized database, MySQL is also multi-platform. This means that in addition to Linux, MySQL also runs on other operating systems, such as Windows, Mac OS X or BSD and UNIX variants.

* 1. **AJAX**
* AJAX stands for Asynchronous JavaScript and XML.
* AJAX is not a new programming language, but a new way to use existing standards.
* AJAX is the art of exchanging data with a server, and updating parts of a web page - without reloading the whole page.
  1. **JavaScript**

JavaScript (JS) is a dynamic computer programming language. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. It is also being used in server-side programming, game development and the creation of desktop and mobile applications.

JavaScript is a prototype-based scripting language with dynamic typing and has first-class functions. Its syntax was influenced by C. JavaScript copies many names and naming conventions from Java, but the two languages are otherwise unrelated and have very different semantics. The key design principles within JavaScript are taken from the self and Scheme programming languages. It is a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles.

* 1. **Cascading Style Sheets**

CSS is a style sheet language used to describe the presentation semantics (that is, the look and formatting) of a document written in a markup language (Style defines how to display HTML elements). Its most common application is to style web page written in HTML and any kind of XML documents.

Cascading Style Sheets (CSS) are widely recognized for their contributions in building fact-loading, standards-compliant, easily modifiable web pages. External Style Sheets are stored in CSS files.

**2.6 HTML**

* + HTML stands for Hyper Text Markup Language
  + A markup language is a set of markup tags.
  + HTML markup tags are usually called as HTML tags.
  + HTML tags are keywords surrounded by angle brackets like <html>.
  + HTML tags are normally come in pairs like <b> and </b>.
  + HTML document describe webpage.
  + HTML document contain HTML tags and plain text.
  + HTML documents are also called as Web Pages.

**SOFTWARE REQUIREMENT**

**SPECIFICATION**

**CHAPTER 3**

**SOFTWARE REQUIREMENT SPECIFICATION**

* 1. **Introduction**

The client usually does not understand software or software development process, and the developer does not understand the client problem. This causes a communication gap.

The basic purpose of SRS is to bridge the communication gap between the client and the developer. SRS is a medium through which client and the user needs are accurately specified. The basic goal of requirement phase is to produce SRS, which describes complete external behavior of the proposed software. Important purpose of developing an SRS is helping the clients understand their own needs. A high quality SRS is prerequisite to high quality software. The SRS of “**The Traveller’s Guide**” is as follows.

* + 1. **Purpose**

Computer and computer application are finding its voyage to each and every field of mankind. Data and information are the primary necessities in the field of information technology. The manipulation and utilization of these data in a meaningful way becomes the need of each and every organization. Manual work is normally time consuming and prone to errors. Computer does it faster, which reduces the manual work thus helping speed up the work.

“**The Traveller’s Guide**” is a widely implemented strategy for managing a travel agency’s interaction with its customers. It involves technology to organize and automate the process. The overall goal is to computerize the administrators work.

The purpose of this document is to specify requirements and to give guidelines for the development of “**The Traveller’s Guide**”. This document aims to give a brief description about the “**The Traveller’s Guide**” Project. With the help of this document the needs of the travel agency and the solutions that will be provided to that needs will be clearly presented. In other words, this document will provide a basis for validation and verification

* + 1. **Product Scope**

Here is a list of features supported by the Online Travel Management System.

* The system provides a user friendly environment for the customers to maintain their profile information.
* The customers do not have to personally go to the counter or contact representatives through the telephone, but they have to only access this website to book tickets.
* Ticket reservation which include checking availability, booking tickets payment options.
* Customers who wish to travel across may reserve their flight tickets, bus tickets, train tickets operated by any company among all available airports, bus stops and railway stations through this website.
* By using this system, user can not only save time but also get much larger search space from which they have higher chance to find a suitable flight, bus, train, tour package, hotels.
* User friendly administrator dashboard to manage company details, hotels, room types, rooms, fight details, tour package details etc.
  + 1. **Definition, Acronyms, Abbreviation**

|  |  |
| --- | --- |
| Acronym | Definition |
| DFD | Data Flow Diagram |
| PHP | Hypertext Preprocessor |
| ER | Entity Relationship |
| GUI | Graphical User Interface |

* + 1. **Project Overview**

The project “The Traveller’s Guide” is related to Online Travel Management System, which makes it easier to maintain the records related to clients. The SRS is organized in a way that any user can easily understand and use “The Traveller’s Guide”. Since it is related to travel, the system helps in providing bus, flight, and train and tour package bookings for the customer by selecting from the listed services in our website.

* 1. **Overall Description**

The overall description of our project can be stated as creating and managing the database, developing a friendly user interface to manipulate the database, provide an authentication mechanism to safely accomplish tasks mentioned above.

* + 1. **Product Perspective**

There are 3 types of users. These types who have different roles can be stated as; Admin, customers, visitors. The admin and customer first enter the main authentication page and after that, system will grant them authorization. Visitors have no permissions and hence, no authorization is required. After being authorized according to their permissions (role types) admin will basically query and edit the database via The Traveller’s Guide.

* + 1. **Product Function**

The system allows its user to buy services (flight tickets, bus tickets, train tickets, hotel rooms, holiday packages) offered in this system across internet connection globally.

* + 1. **User Characteristics**

The system provides access to Visitors, Customers and Administrators.

* Visitor: Users who are currently not logged in and have not registered to the website are visitors. Visitors can search the details of the flight, bus and train. In order to book the ticket the visitor will need to login with the valid username and password else register with the system and create an account.
* Customer: Customers can check availability of the flight, bus and train reservations and also book tickets by making payment online. They can also view their booking details, payment details etc.
* Administrator: Administrator can manage company details, hotels, room types, rooms, fight details, tour package details, etc. Also admin can view and print transaction report, booking report, cancellation report, etc.
  + 1. **General Constraints**
* Internet facility is essential to use this web application.

## The server will stop working if there happens to be a high traffic.

* The developed system should run under any platform (UNIX, Linux, Mac, Windows etc.) that contains a web browser which supports PHP, JavaScript and AJAX.
  + 1. **Assumptions and Dependencies**
* The user should operate in computer or laptop that has internet facility.
* Roles and responsibilities are already established.
* Administrator is already created.
* User of this software must have basic computer knowledge.
  1. **Specific Requirements**
     1. **External Interface Requirement**

The following sections will introduce the numerous requirements of the system from the point of view of different users and will introduce a number of decisions that have been made regarding implementation. These sections also attempt to somewhat describe the role of each user group in the system, discussing their individual roles through the functions they can perform.

* + 1. **User Interface**

The user interface for this system will have to be simple and clear. Most importantly, the ages must be easy to read, easy to understand and accessible. The color scheme should be appropriate to provide familiarity with the travel site and there should be no contrast issues

* + 1. **Software Interface**
* Scripting languages: HTML, CSS, JavaScript
* Front End: PHP
* Back End: MySQL
* Software: XAMPP 1.8.2
* Server: Apache server
* Database server: MySQL Server
* Operating System: Windows XP or higher
  + 1. **Hardware Interface**
* Processor: Pentium IV & above.
* RAM: 1 GB (Recommended).
* Disk Space: 40 GB Hard Disk.
* Monitor: Color Monitor.
* Input Device: Keyboard, Mouse
* Output Device: Monitor, Printer
  + 1. **Communication Interface**

Internet connection is required.

* 1. **Functional Requirement**
* **Flight ticket booking:**

In this module the customer can search source-destination, check flight timings and book a ticket online.

* Search flight ticket.
* View flights and flight schedules.
* Flight information.
* Make a ticket booking online and print ticket.
* Cancellation.
* **Bus ticket reservation module:**

In bus ticket reservation module, there has been a collection of buses; customer can book the ticket by selecting agency, date and departure time of the bus.

* Search bus type.
* Advance bus ticket booking.
* Check Departure time and arrival time.
* Print ticket.
* Cancellation.
* **Train ticket reservation module:**

In train reservation module customer can check train scheduled timings and can book tickets online by entering place, date and time.

* Check seat availability.
* Fare enquiry.
* Online reservation.
* Search Train/Fare availability details.
* Train schedule details.
* Train type information.
* Cancellation.
* **Hotel reservation module:**

In this module, customers will be provided with an option to search hotels. Customer can reserve the hotels by selecting check-in date, checkout date and hotel name.

* Search hotels by entering city.
* Find hotels and room types.
* Add check-in date and check-out date.
* Book or reserve domestic and international hotels.
* Cancellation.
* **Holiday package module:**

Here in this module, customer can search tour packages by entering destination, duration of travel, month of travel and number of person.

* Holiday packages based on vacation.
* View holiday package price details, and package details.
* Book domestic and International holiday packages.
* Cancellation.
* **Payment module:**

This module allows customers to make payment after the reservation. The customer will receive ticket receipt by mail after the payment.

* Make payment to selected service.
* Generate billing receipt.
* Update credit card details.
* View payment details.
* **My Account module:**

This module is for customers where customer can access account module after login id and password. This module will display booked ticket, Billing receipt, ticket cancellation page, logout, etc.

* Display booked receipt.
* Change password.
* Update profile information.
* **Cancel booking module:**

Customer can cancel flight ticket, bus ticket, train ticket, or hotel bookings. The system refunds paid amount the after deducting cancellation charges.

* Cancel booking details.
* Cancellation policy.
* Refundable and non-refundable booking amount.
* **Dashboard module:**

This module allows the administrator of this site to manage the different aspects of ticket reservation related activities. The administrator of the site can add or modify the records of flights, bus, train, holiday packages, etc.

* Add, edit, view flight records.
* Add, edit, and view bus agency and buses.
* Add, edit, view train details.
* Add, edit, view hotel information.
* Add, edit, view holiday packages.
* View bookings, View cancellation report.
* View payment report.
* Add users.
* Change password.
  1. **Design and Implementation Constraints**
* Customer does not have any rights to edit or delete booking and payment records.
* The online payment gateway is virtual payment gateway for testing purpose.
* This system is not support distributed database Facility.
* System is limited to HTTP/HTTPS Protocols.
* Basic computer knowledge and internet connection required.

**SYSTEM DESIGN**

**CHAPTER 4**

**SYSTEM DESIGN**

* 1. **Introduction**

Design is a process through which requirements are translated into a representation of the software. The purpose of the designing phase is to plan a solution for the problem specified by the requirement document. This phase moves from the problem domain to the solution domain i.e. the requirements are translated into software. The design activity often results in three separate outputs:

1. Architecture design
2. High level design
3. Detailed design

In architecture design the focus is on identifying components or subsystems and how they interact to each other. The high level design identifies the modules that should build for developing the system. In case of detailed design the focus is on how the modules are implemented in software.

System design is a solution, a “How to” approach to the creation of a new system. The important phase is composed of several steps. It provides the understanding of procedural details necessary for implementing the system recommended in the feasibility study. Emphasis is on translating the performance requirements into design specifications.

* 1. **Functional Decomposition**

The system can be decomposed into functional components as follows.

The Components are –

* Registration for the customers i.e. Login component.
* Flight ticket booking for customers.
* Bus ticket booking for customers.
* Railway ticket booking for customers.
* Hotel booking for customers.
* Tour packages booking for customers.
* Billing Component.
  1. **Description of program**
     1. **Context Flow Diagram**

Context flow diagram is a top level data flow diagram. It only contains one process node that generalizes the function of the entire system in relationship to external entities. In context diagram the entire system is treated as a single process and all its inputs, outputs, sinks and sources are identified and shown.



* + 1. **Data Flow Diagram**

A data flow diagram is a graphical representation of the flow of data through an information system. A data flow diagram can also be used for the visualization of the data processing. It is common practice for a designer to draw a context level DFD. It shows the interaction between the system and the outside entities. This context level DFD, is then exploded to show more detail of the system being modeled.

A DFD represents flow of data through a system. Data flow diagrams are commonly used during problem analysis. It views a system as a function that performs the input into the desired output. A DFD shows movement of data through the different transformations or processes in the system.

Data Flow diagrams can be used to provide the end users with the physical idea of where the data they input ultimately has an effect upon the structure of whole system from order to dispatch to restock how any system is developed can be determined through data flow diagram. The appropriate register saved in database and maintained by appropriate authorities.

**Notations in the DFD**

|  |  |
| --- | --- |
| **Symbol** | **Description** |
|  | The circle or bubble represents a process. A process is named and each process is represented by a named circle. |
|  | The source or sink is represented as a rectangular box. The source or sink is the net originator or the consumer of the data that flows in the system. |
|  | The arrow represents the flow of data through the system. The labeled arrows enter or leave the bubbles. |
|  | The database is represented with the open box symbol. |

**DFD**  **Level-1**(Top Level DFD)

****

**DFD Level-2**: Registration



**Description Of Components :**

* **Input:** Customer fills registration details by entering customerid, customername, gender, dob, password.
* **Process:** The system checks the entered details are valid or not.
* **Output** : After the registration ,customer can login to system by entering loginid and password

**DFD Level-3** : Flight



**Description Of Components :**

* **Input** : Admin enters flight details by entering source and destination location ,company information ,arrival time and departure time
* **Process :** Customer searches for available flights
* **Output:** The system generates billing records after customer books the flight ticket.

**DFD Level-4: Bus**



**Description Of Components :**

* **Inputs:** Admin entersbus details by selecting source and destination location , company details , departure date and time and no of seats
* **Process:** Customer will search bus information by entering date, source and destination location.
* **Output:** After making payment confirmation booking ticket will be sent to the customer

**DFD Level-5**: Train



**Description Of Components :**

* **Inputs:** Admin enters train schedule timings and train types by entering source and destination location
* **Process:** Customers will search train details by entering source and destination location
* **Output:** The customer will receive billing receipt after making the payment

**DFD Level-6**: Hotel



**Description Of Components :**

* **Inputs :** Administrator will enter hotel details and its room types and number of rooms and upload its images
* **Process**: Customer will search room types from the available hotel
* **Output:** The customer books rooms after making the payment.

**DFD Level-7:** Tour package

****

**Description Of Components :**

* **Inputs:** Adminenters tour package details, tourpackage place, tourpackage travel options and tour package itinerary.
* **Process:** Customers will search tour package details by entering tour package type.
* **Output:** The customers will receive tour package booking after confirmation.

**DFD Level-8:** Payment



**Description Of Components:**

* **Inputs:** Customers will enter payment details, payment credentials.
* **Process:** The system checks entered payment details valid or not.
* **Output:** The payment details will be the sent to the customer.

**DATABASE DESIGN**

**CHAPTER 5**

**DATABASE DESIGN**

* 1. **Introduction**

Database Design maintains the data required by the system. One of the key design issues involved in the database to design is the distribution of data in a way that minimizes transaction traffic. Another key design issue is the choice of the database management system. Database Tables used are described in the following sections.

A database is a collection of data organized to allow easy access of retrievals, additions, modifications and deletions.

A typical database consists of different parts as shown in the following figure. The Database server (MySQL Server) in the top level is a software server that can be accessed by multiple users. Within the database server, number of databases can be stored.

Each database stored data in the series of tables that can be related to each other in different ways. The most widely used approach for structuring the data is called Relational database system (RDBMS). A RDBMS is a collection of tables of data.

Database design is a process is a process of organizing the data in an orderly manner so as to provide easy access to the required information.

It is difficult to maintain the database if lot of repetitive data is stored in the table. If one instance of the data undergoes a change, that change has to be made for all occurrences of the data. To eliminate duplication and easy maintenance of data, it is recommended to create a table of repeated values in one table called master table. The reference remains same throughout the other tables.

Normalization is the process of organizing the data in order to minimize duplication and inconsistency.

* 1. **Schema Description**

# Table Name: admin

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Index** | **Description** |
| **adminid** | int(10) | Primary Key | Admin ID |
| adminname | varchar(25) | No | Administrator Name |
| loginid | varchar(25) | No | Adminstrator Login ID |
| password | varchar(50) | No | Administrator password |
| status | varchar(10) | No | Administrator status |

## Table Name: amenities

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Index** | **Description** |
| **amenitiesid** | int(10) | Primary Key | Amenities ID |
| busid | int(10) | Foreign Key | Bus ID |
| amenities | varchar(25) | No | Amenities Name |
| icon | varchar(100) | No | Icon |
| status | varchar(10) | No | Amenities Status |

## Table Name: billing

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Index** | **Description** |
| **billingid** | int(10) | Primary Key | Billing ID |
| customerid | int(10) | Foreign Key | Customer ID |
| billingcost | float(10,2) | No | Billing Cost |
| tax | float(10,2) | No | Billing Tax |
| date | Date | No | Billing Date |
| payment\_type | varchar(20) | No | Payment type |
| cardno | varchar(25) | No | Card Number |
| status | varchar(20) | No | Billing Status |

## 

## 

## Table Name: boardingpoints

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Index** | **Description** |
| **boardingpointid** | int(10) | Primary Key | Boarding Point ID |
| busid | int(10) | Foreign Key | Bus ID |
| boardingpointno | int(10) | No | Boarding Point Number |
| slocation | varchar(50) | Foreign Key | Source Location ID |
| dlocation | varchar(50) | Foreign Key | Destination Location ID |
| arrivaltime | datetime | No | Arrival Time |
| departuretime | datetime | No | Departure Time |
| seatercost | float(10,2) | No | Seater Seats Cost |
| sleepercost | float(10,2) | No | Sleeper Seats Cost |
| status | varchar(10) | No | Boarding Points Status |

## Table Name: booking

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Index** | **Description** |
| **bookingid** | int(10) | Primary Key | Booking ID |
| billingid | int(10) | Foreign Key | Billing ID |
| bookingtype | varchar(20) | No | Booking Type |
| ticketed | int(10) | No | Ticket ID |
| seatno | varchar(25) | No | Seat Number |
| passengername | varchar(25) | No | Passenger Name |
| gender | varchar(10) | No | Gender |
| contact | varchar(15) | No | Contact |
| status | varchar(10) | No | Booking status |

## Table Name: bus

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Index** | **Description** |
| **busid** | int(10) | Primary Key | Bus id |
| companyid | int(10) | Foreign Key | Company ID |
| bustype | varchar(25) | No | Type of the Bus |
| sleeperseats | int(10) | No | Seats of Sleeper Type |
| seaterseats | int(10) | No | Seats of Seater Type |
| status | varchar(10) | No | Bus Status |

## Table Name: cancellation

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Index** | **Description** |
| **cancellation\_id** | int(10) | Primary Key | Cancellation ID |
| billingid | int(10) | Foreign Key | Billing ID |
| refundamount | float(10,2) | No | Refund Amount |
| cancellationdate | date | No | Cancellation Date |
| status | varchar(10) | No | Cancellation Status |

## Table Name: company

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Index** | **Description** |
| **companyid** | int(10) | Primary Key | Company ID |
| companytype | varchar(20) | No | Company Type |
| companyname | varchar(25) | No | Company Name |
| companylogo | varchar(100) | No | Company Logo |
| description | text | No | Description |
| status | varchar(10) | No | Company Status |

## Table Name: customer

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Index** | **Description** |
| **customerid** | int(10) | Primary Key | Customer ID |
| customername | varchar(25) | No | Customer Name |
| dob | date | No | Date of Birth |
| gender | varchar(10) | No | Gender |
| address | varchar(500) | No | Customer Address |
| contact | varchar(20) | No | Contact |
| mobile | varchar(20) | No | Mobile |
| state | varchar(20) | No | State |
| country | varchar(25) | No | Country |
| pincode | varchar(10) | No | Pincode |
| **emailed** | varchar(20) | No | Email ID |
| **loginid** | varchar(20) | No | Login ID |
| password | varchar(50) | No | Password |
| status | varchar(10) | No | Customer Status |

## Table Name: flight

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Index** | **Description** |
| **flightid** | int(10) | Primary Key | Flight ID |
| companyid | int(10) | Foreign Key | Company ID |
| dlocationid | int(10) | Foreign Key | Departure Location ID |
| alocationid | int(10) | Foreign Key | Arrival Location ID |
| duration | varchar(25) | No | Duration |
| departtime | datetime | No | Departure Time |
| arrivetime | datetime | No | Arrival Time |
| economyseats | int(10) | No | Economy Seats |
| businessseats | int(10) | No | Business seats |
| firstclassseats | int(10) | No | First Class seats |
| premiumeconomyseats | int(10) | No | Premium Economy seats |
| econ\_tariff | float(10,2) | No | Economy Tariff |
| business\_tariff | float(10,2) | No | Business Tariff |
| firstclass\_tariff | float(10,2) | No | First Class Tariff |
| premium\_tariff | float(10,2) | No | Premium Economy Tariff |
| description | Text | No | Description |
| status | varchar(10) | No | Flight Status |

## Table Name: hotel

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Index** | **Description** |
| **hoteid** | int(10) | Primary Key | Hotel ID |
| companyid | int(10) | Foreign Key | Company ID |
| locationid | int(10) | Foreign Key | Location ID |
| hotelname | varchar(25) | No | Hotel Name |
| hoteladdress | varchar(250) | No | Hotel Address |
| pincode | varchar(50) | No | Pincode |
| landmark | varchar(50) | No | Landmark |
| hoteldescription | Text | No | Hotel Description |

## Table Name: hotel\_image

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Index** | **Description** |
| **hotel\_image\_id** | int(10) | Primary Key | Hotel Image ID |
| hotelid | int(10) | Foreign Key | Hotel ID |
| hotel\_image | varchar(50) | No | Hotel Image |
| hotel\_image\_description | Text | No | Hotel Image Description |

## Table Name: location

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Index** | **Description** |
| **locationid** | int(10) | Primary Key | Location ID |
| location | varchar(25) | No | Location |
| locationtype | varchar(25) | No | Location Type |
| locationtitle | varchar(100) | No | Location Title |
| description | Text | No | Description |
| status | varchar(10) | No | Location Status |

## Table Name: rating

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Index** | **Description** |
| **ratingid** | int(10) | Primary Key | Rating ID |
| ratingtype | varchar(25) | No | Rating Type |
| customerid | int(10) | Foreign Key | Customer ID |
| companyid | int(10) | Foreign Key | Company ID |
| rating | int(10) | No | Rating |
| commenttitle | varchar(100) | No | Comment Title |
| comments | Text | No | Comments |
| ratingdate | Date | No | Rating Date |
| status | varchar(10) | No | Rating Status |

## Table Name: room

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Index** | **Description** |
| **roomid** | int(10) | Primary Key | Room ID |
| roomno | varchar(30) | No | Room Number |
| roomtypeid | int(10) | Foreign Key | Room Type ID |
| status | varchar(10) | No | Room Status |

## Table Name: roomtype

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Index** | **Description** |
| **roomtypeid** | int(10) | Primary Key | Room Type ID |
| hoteid | int(10) | Foreign Key | Hotel ID |
| roomtype | varchar(50) | No | RoomType |
| note | Text | No | Note |
| roomtariff | float(10,2) | No | Room Tariff |
| status | varchar(10) | No | Status |

## Table Name: room\_booking

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Index** | **Description** |
| **room\_booking\_id** | int(10) | Primary Key | Room Booking ID |
| billingid | int(10) | Foreign Key | Billing ID |
| customerid | int(10) | Foreign Key | Customer ID |
| roomid | int(10) | No | Room ID |
| checkindate | datetime | No | Check in Date |
| checkoutdate | datetime | No | Check out Date |
| noofadults | int(10) | No | Number of Adults |
| noofchildren | int(10) | No | Number of Childrens |
| status | varchar(10) | No | Status |

## Table Name: service

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Index** | **Description** |
| **serviceid** | int(10) | Primary Key | Service ID |
| hoteid | int(10) | Foreign Key | Hotel ID |
| servicetype | varchar(50) | No | Service Type |
| servicetitle | varchar(100) | No | Service Title |
| description | text | No | Description |
| image | varchar(100) | No | Image |
| cost | float(10,2) | No | Cost |
| tariff\_type | varchar(25) | No | Tariff Type |
| unit | varchar(25) | No | Unit |
| status | varchar(10) | No | Status |

## Table Name: tourpackage

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Index** | **Description** |
| **tour\_package\_id** | int(10) | Primary Key | Tour Package ID |
| companyid | int(10) | Foreign Key | Company ID |
| tourpackagetype | varchar(25) | No | Tour Package Type |
| packagetitle | varchar(100) | No | Package Title |
| videolink | varchar(100) | No | Video Link |
| imagelink | varchar(100) | No | Image Link |
| overview | text | No | Overview |
| inclusion | text | No | Inclusion |
| no\_of\_days | varchar(50) | No | Number of Days |
| minprice | float(10,2) | No | Minimum Price |

## Table Name: tourpackage\_itinerary

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Index** | **Description** |
| **itineraryid** | int(10) | Primary Key | Itinerary ID |
| tour\_package\_id | int(10) | Foreign Key | Tour Package ID |
| day\_no | int(10) | No | Day Number |
| itinerary\_title | varchar(100) | No | Itinerary Title |
| itinerary\_description | text | No | Itinerary Description |
| itinerary\_image | varchar(100) | No | Itinerary Image |

## Table Name: tourpackage\_place

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Index** | **Description** |
| **tourpackage\_place\_id** | int(10) | Primary Key | Tour Package place ID |
| tourpackage\_id | int(10) | Foreign Key | Tour Package ID |
| about\_place | text | No | About Place |
| place\_map | text | No | Place Map |
| contact | varchar(25) | No | Contact |
| status | varchar(10) | No | Status |

## Table Name: tourpackage\_travel\_options

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Index** | **Description** |
| **travel\_option\_id** | int(10) | Primary Key | Travel option ID |
| tour\_package\_id | int(10) | Foreign Key | Tour Package ID |
| travel\_source\_loc | varchar(25) | No | Travel Source Location |
| travel\_destination\_loc | varchar(25) | No | Travel Destination Location |
| travel\_type | varchar(25) | No | Travel Type |
| note | text | No | Note |

## Table Name: train

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Index** | **Description** |
| **trainid** | int(10) | Primary Key | Train ID |
| trainname | varchar(25) | No | Train Name |
| traintype | varchar(25) | No | Train Type |
| sleeperseats | int(10) | No | Sleeper Seats |
| seaterseats | int(10) | No | Seater Seats |
| status | varchar(25) | No | Status |

## Table Name: train\_type

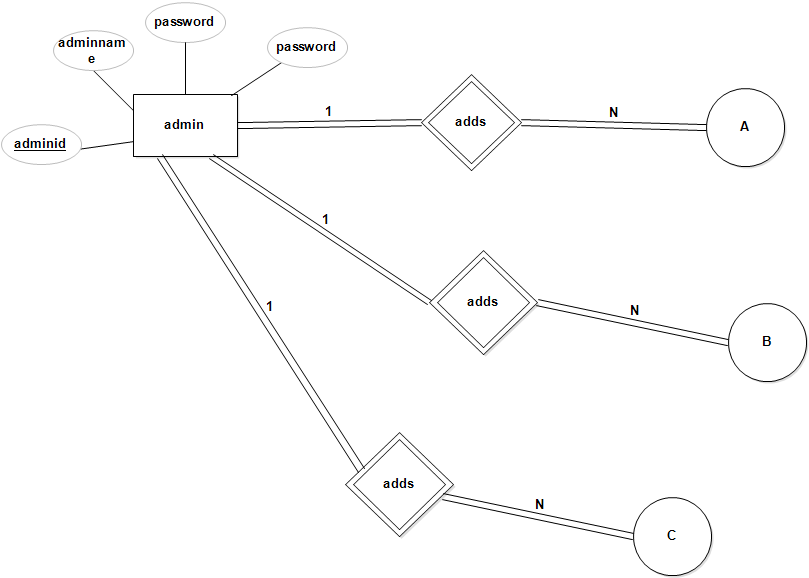
|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Index** | **Description** |
| **train\_type\_id** | int(10) | Primary Key | Train Type ID |
| trainid | int(10) | Foreign Key | Train ID |
| boardingpointno | int(10) | No | Boarding point number |
| slocation | varchar(50) | Foreign Key | Starting point of travel |
| dlocation | varchar(50) | Foreign Key | Ending point of travel |
| arrivaltime | datetime | No | Arrival Time |
| departuretime | datetime | No | Departure Time |
| seatercost | float(10,2) | No | Seater Cost |
| sleepercost | float(10,2) | No | Sleeper Cost |
| status | varchar(10) | No | Status |

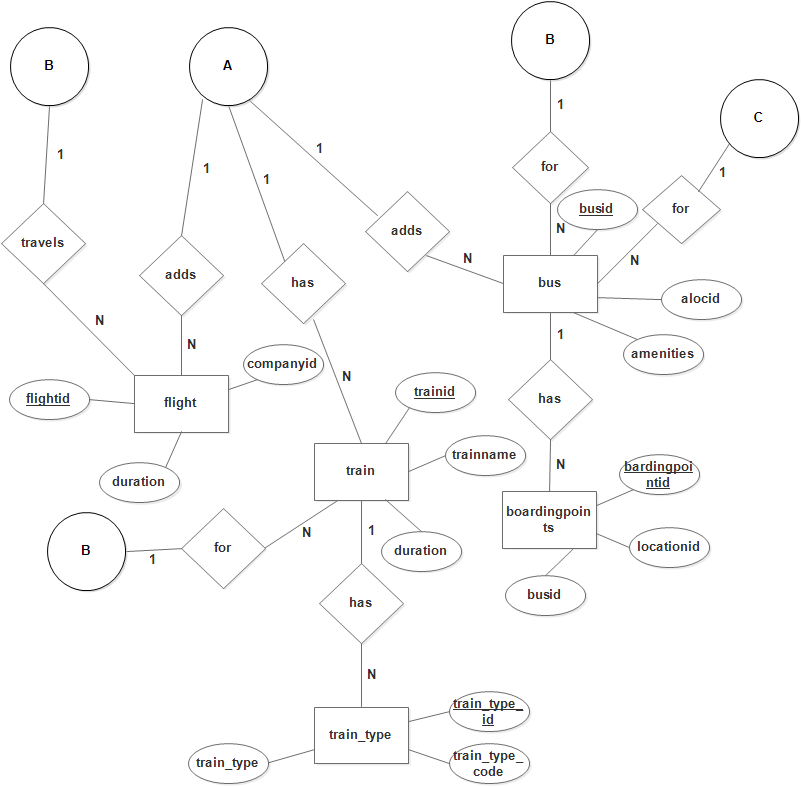
* 1. **Entity Relationship Model**

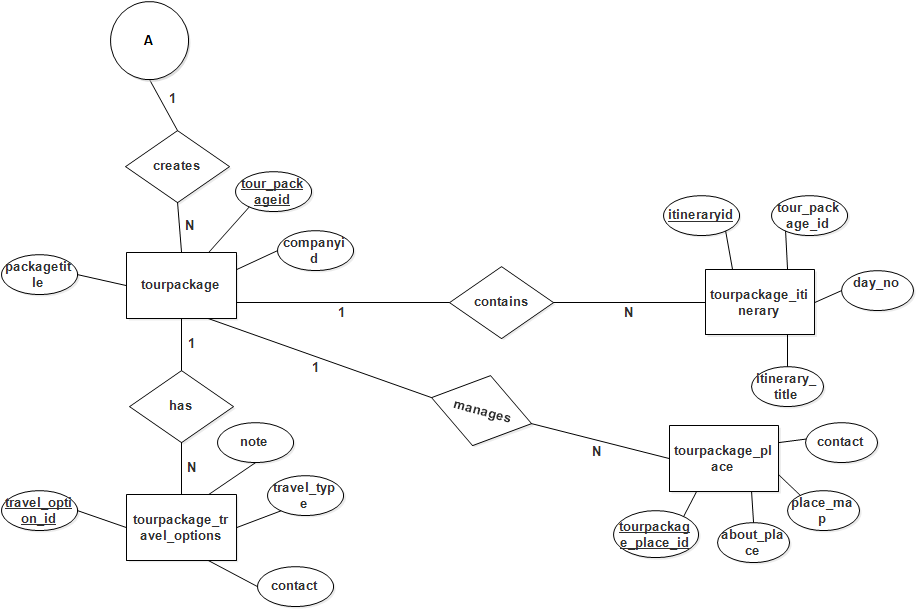
An entity-relationship (ER) diagram is a specialized graphic that illustrates the [relationships between entities in a database](http://databases.about.com/od/specificproducts/a/Database-Relationships-An-Introduction-To-Foreign-Keys-Joins-And-E-R-Diagrams.htm). ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes.

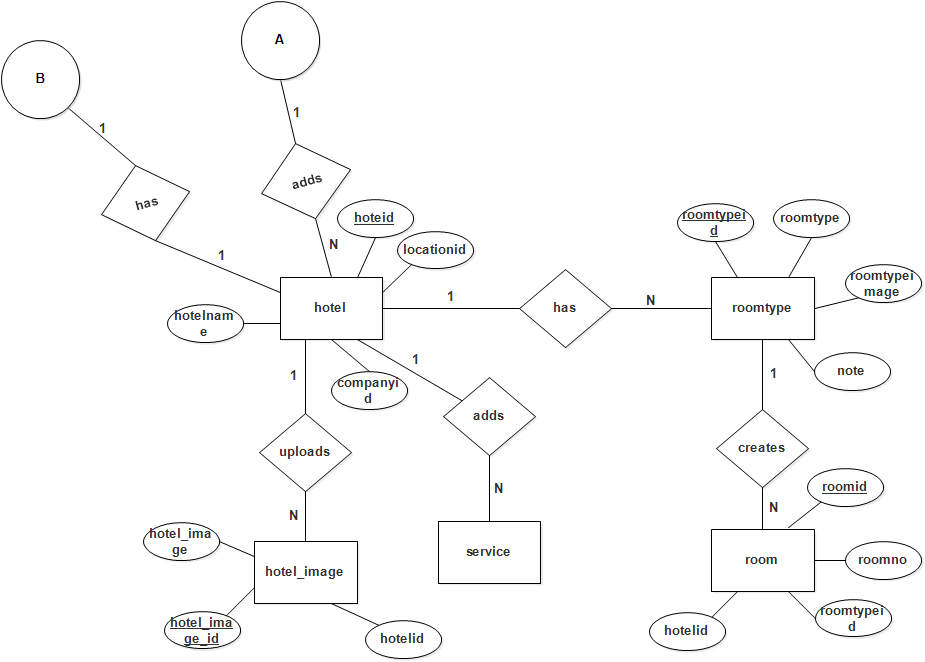
The Symbols are shown in table below:

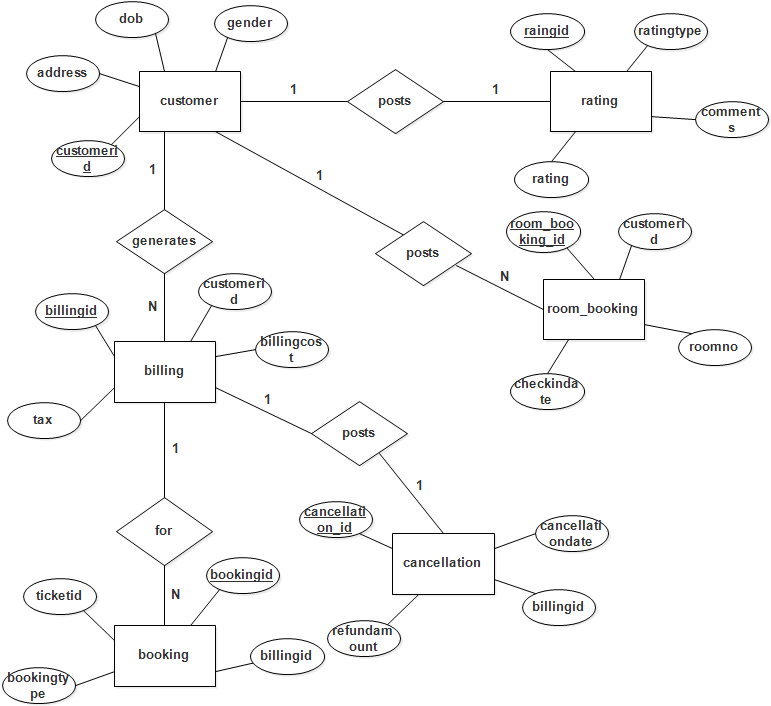
|  |  |  |
| --- | --- | --- |
| **Name** | **Notation** | **Description** |
| Entity |  | Entity is represented by a box within the ERD. Entities are abstract concepts, each representing one or more instances of the concept in question. An entity might be considered a container that holds all of the instances of a particular thing in a system. Entities are equivalent to database tables in a relational database, with each row of the table representing an instance of that entity. |
| Relationship |  | Relationships are represented by Diamonds. A relationship is a named collection or association between entities or used to relate to two or more entities with some common attributes or meaningful interaction between the objects. |
| Attributes |  | Attributes are represented by Oval. An attribute is a single data item related to a database object. The database schema associates one or more attributes with each database entity. |

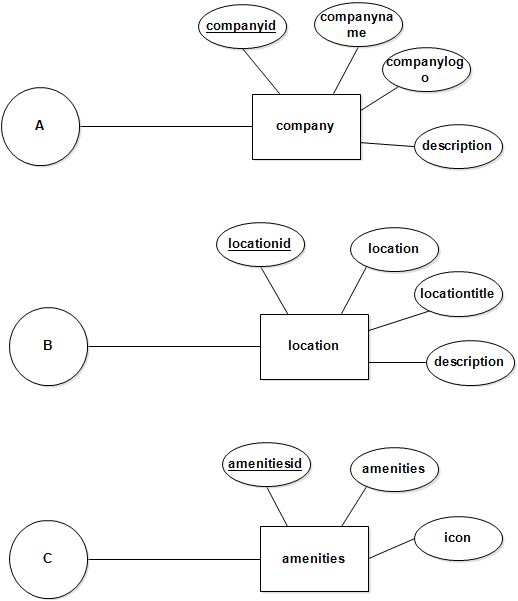












**DETAILED DESIGN**

**CHAPTER 6**

**DETAILED DESIGN**

* 1. **Introduction**

The details design we decide the internal logic for the module, which implement the given specification. Detailed design is one of the levels of the Design process for software items detailed design, which is also known as logic design, involves the internal sign of the module and how the specification of the module can be satisfied. Detailed design is an extensor of system design.

The first step before the detailed design or code for a module can be developed is that the specification of the module be given precisely. Once the module is precisely specified, the external logic for the module that will implement the given identification can be decided.

The basic logic in detailed design is to specify the logic for the different modules that been specified during system design. Specifying the logic will require developing an algorithm that will implement the given specifications.

This phase includes the specification of table structures, which makes us understand the internal logic of the system. The tables are described with their field names, data type, table constraints, description of each field etc.

* 1. **Structured English**

**AdminLogin Form:**

Begin

If (click on login) then

Check the username and password, of the user exists then redirect to appropriate panel

Else

Display a message “failed to login”

End if

End

**Admin Form:**

BEGIN

IF (click on add admin) THEN

Admin form is loaded and administrator can add new admin

ELSE IF (click on edit admin) THEN

Admin form is loaded with appropriate admin details, and

administrator updates it

END IF

IF (click on submit) THEN

Admin record is inserted / updated and Success message is displayed

ELSE

Failure message is displayed

END IF

END

**Bus Form:**

begin

IF (click on add bus) THEN

Bus form is loaded and administrator can add new bus details

ELSE IF (click on edit bus) THEN

Bus form is loaded with appropriate bus details, and administrator

Updates it

END IF

IF (click on submit) THEN

Bus record is inserted / updated and Success message is displayed

ELSE

Failure message is displayed

END IF

END

**Company Form:**

Begin

if (click on add company) then

Company form is loaded and administrator can add new company details

Else if (click on edit company) then

Company form is loaded with appropriate company details, and administrator updates it

End if

If (click on submit) then

Company record is inserted / updated and Success message is displayed

Else

Failure message is displayed End if

End

**Customer Form:**

Begin

if (click on signup) then

Customer form is loaded and administrator can add new customer details

Else if (click on edit customer) then

Customer form is loaded with appropriate customer details, and administrator updates it

End if

If (click on submit) then

Customer record is inserted / updated and Success message is displayed

Else

Failure message is displayed

End if

End

**Flight Form:**

Begin

if (click on add flight) then

Flight form is loaded and administrator can add new flight details

Else if (click on edit flight) then

Flight form is loaded with appropriate flight details, and administrator updates it

End if

If (click on submit) then

Flight record is inserted / updated and Success message is displayed

Else

Failure message is displayed

End if

End

**Hotel Form:**

Begin

if (click on add hotel) then

Hotel form is loaded and administrator can add new hotel details

Else if (click on edit hotel) then

Hotel form is loaded with appropriate hotel details, and administrator updates it

End if

If (click on submit) then

Hotel record is inserted / updated and Success message is displayed

Else

Failure message is displayed

End if

End

**Location Form:**

Begin

if (click on add location) then

Location form is loaded and administrator can add new location details

Else if (click on edit location) then

Location form is loaded with appropriate location details, and administrator updates it

End if

If (click on submit) then

Location record is inserted / updated and Success message is displayed

Else

Failure message is displayed

End if

End

**Tourpackage Form:**

Begin

if (click on add tourpackage) then

Tourpackage form is loaded and administrator can add new tourpackage details

Else if (click on edit tourpackage) then

Tourpackage form is loaded with appropriate tourpackage details, and administrator updates it

End if

If (click on submit) then

Tourpackage record is inserted / updated and Success message is displayed

Else

Failure message is displayed

End if

End

**Train Form:**

Begin

if (click on add train) then

Train form is loaded and administrator can add new train details

Else if (click on edit train) then

Train form is loaded with appropriate train details, and administrator updates it

End if

If (click on submit) then

Train record is inserted / updated and Success message is displayed

Else

Failure message is displayed

End if

End

**CODING**

**CHAPTER 7**

**CODING**

* 1. **Introduction**

The main goal of the coding or programming phase is to translate the design of the system produced during the design phase into code in a given programming language. It is then executed on a computer to verify whether the design is correct or not. In coding phase the output document is code can be extremely useful in enhancing the understandability. Internal documentation of code is done using comments in the program. Comments are textual statements that are meant for the program reader and are not executed.

The coding phase effects both testing and maintenance phases. Well written code can reduce the testing and maintenance teffort. The goal of coding should be to reduce the testing and maintenance effort

* 1. **Source Code**

**Database Connection Page**

<?php

$con = mysqli\_connect("localhost","root","technology","travel");

if(mysqli\_connect\_errno($con))

{

echo "Error in MySQL connection. ". mysqli\_connect\_error($con);

}

?>

**AdminLogin Page**

<?php

session\_start();

if(isset($\_SESSION[adminid]))

{

header("Location: dashboard.php");

}

ob\_start();

include("header.php");

include("dbconnection.php");

if(isset($\_POST[submit]))

{

$sql="SELECT \* FROM admin WHERE loginid='$\_POST[loginid]' AND password='$\_POST[password]'";

$qsql = mysqli\_query($con,$sql);

if(mysqli\_num\_rows($qsql) == 1)

{

$rs = mysqli\_fetch\_array($qsql);

$\_SESSION[adminid] = $rs[adminid];

header("Location: dashboard.php");

}

else

{

echo "<script>alert('Failed to login. Kindly enter again...');</script>";

}

}

?>

<div class="page-title-container">

<div class="container">

<div class="page-title pull-left">

<h2 class="entry-title">Admin Login Panel</h2>

</div>

</div>

</div>

<section id="content" class="gray-area">

<div class="container">

<div class="row">

<div id="main" class="col-sm-8 col-md-9">

<div class="booking-section travelo-box">

<form class="cruise-booking-form" method="post" action="" >

<div class="person-information">

<h4>Kindly enter login ID and Password</h4>

<div class="row">

<div class="form-group col-sm-6 col-md-5">

<label>Login ID</label>

<input type="text" name="loginid" class="input-text full-width" value="" placeholder="" />

</div>

</div>

<div class="row">

<div class="form-group col-sm-6 col-md-5">

<label>Password</label>

<input type="password" name="password" class="input-text full-width" value="" placeholder="" />

</div>

</div>

<div class="row">

<div class="form-group col-sm-6 col-md-5">

<input type="submit" name="submit" value="submit" />

</div>

</div>

</div>

</form>

</div>

</div>

<?php

include("rightsidebar.php");

?>

</div>

</div>

</section>

<?php

include("footer.php");

?>

**Customer Page**

<?php

include("header.php");

include("dbconnection.php");

if(isset($\_POST["submit"]))

{

$sqlbilling= "SELECT \* FROM customer WHERE loginid='$\_GET[loginid]' OR emailid='$\_POST[emailid]'";

$qsqlbilling = mysqli\_query($con,$sqlbilling);

if(mysqli\_num\_rows($qsqlbilling) >= 1)

{

echo "<script>alert('Login ID or Email ID already exists.. ')</script>";

echo "<script>window.location.assign('customer.php');</script>";

}

else

{

$sql = "INSERT INTO customer(customername ,dob,gender,address,contact,mobile ,state,country,pincode,emailid,loginid,password,status)values('$\_POST[firstname] $\_POST[lastname] ','$\_POST[dob]','$\_POST[gender]' ,'$\_POST[address]' ,'$\_POST[contact]','$\_POST[mobile]','$\_POST[state]','$\_POST[country]','$\_POST[pincode]','$\_POST[emailid]','$\_POST[loginid]','$\_POST[password]','Active')";

if(!mysqli\_query($con,$sql))

{

echo "Error in insert statement" .mysqli\_error($con);

}

else

{

echo "<script>alert('Customer record inserted successfully..');</script>";

}

}

}

?>

<div class="page-title-container">

<div class="container">

<div class="page-title pull-left">

<h2 class="entry-title">Customer Registration Form</h2>

</div>

</div>

</div>

<section id="content" class="gray-area">

<div class="container">

<div class="row">

<div id="main" class="col-sm-8 col-md-9">

<div class="booking-section travelo-box">

<form class="cruise-booking-form" method="post" action="" name="frmcustomer" onsubmit="return validatecustomer();">

<div class="person-information">

<div class="row">

<div class="form-group col-sm-6 col-md-5">First Name<br />

<input type="text" name="firstname" class="input-text full-width" value="" placeholder="" />

</div>

</div>

<div class="row">

<div class="form-group col-sm-6 col-md-5">Last Name<br />

<input type="text" name="lastname" class="input-text full-width" value="" placeholder="" />

</div>

</div>

<div class="row">

<div class="form-group col-sm-6 col-md-5">DOB<br/>

<input type="date" name="dob" class="input-date full-width" value="" placeholder="" />

</div>

</div>

<div class="row">

<div class="form-group col-sm-6 col-md-5">Gender<br/>

<input type="radio" name="gender" id="male" value="Male" > Male <br />

<input type="radio" name="gender" id="female" value="Female" >Female

</div>

</div>

<div class="row">

<div class="form-group col-sm-6 col-md-5">Address <br/>

<textarea name="address" id="address"></textarea>

</div>

</div>

<div class="row">

<div class="form-group col-sm-6 col-md-5">State

<p> <select name="state" ><?php

$arr= array("Select","Andhra pradesh","Karnataka","Kerala");

foreach($arr as $val)

{

if($val == $rs[state])

{

echo "<option value='$val' selected>$val</option>";

}

else

{

echo "<option value='$val'>$val</option>";

}

}

?></select></p>

</div>

</div>

<div class="row">

<div class="form-group col-sm-6 col-md-5">Country

<p> <select name="country" ><?php

$arr= array("Select","India","China","USA","Australia");

foreach($arr as $val)

{

if($val == $rs[country])

{

echo "<option value='$val' selected>$val</option>";

}

else

{

echo "<option value='$val'>$val</option>";

}

}

?></select></p>

</div>

</div>

<div class="row">

<div class="form-group col-sm-6 col-md-5">

<label>PIN Code</label>

<input type="text" name="pincode" placeholder="Please enter pin code" class="input-text full-width" />

</div>

</div>

<div class="row">

<div class="form-group col-sm-6 col-md-5">

<label>Contact</label>

<input type="text" name="contact" placeholder="Please enter contact" class="input-text full-width" />

</div>

</div>

<div class="row">

<div class="form-group col-sm-6 col-md-5">

<label>Mobile No</label>

<input type="text" name="mobile" placeholder="Please enter mobile no" class="input-text full-width" />

</div>

</div>

<div class="row">

<div class="form-group col-sm-6 col-md-5">

<label>E-mail Id</label>

<input type="email" name="emailid" placeholder="Please enter emailid" class="input-text full-width" />

</div>

</div>

<div class="row">

<div class="form-group col-sm-6 col-md-5">

<label>Login Id</label>

<input type="text" name="loginid" placeholder="Please enter login id" class="input-text full-width" />

</div>

</div>

<div class="row">

<div class="form-group col-sm-6 col-md-5">

<label>password</label>

<input type="password" name="password" placeholder="Please enter password" class="input-text full-width" />

</div>

</div>

<div class="row">

<div class="form-group col-sm-6 col-md-5">

<label>Confirm password</label>

<input type="password" name="confirmpassword" placeholder="Please confirm password" class="input-text full-width" />

</div>

</div>

<div class="row">

<div class="form-group col-sm-6 col-md-5">

<input type="submit" name="submit" value="Register" />

</div>

</div>

</div>

</form>

</div>

</div>

<?php

include("rightsidebar.php");

?>

</div>

</div>

</section>

<?php

include("footer.php");

?>

<script type="application/javascript">

function validatecustomer()

{

var numericExpression = /^[0-9]+$/;

var alphaExp = /^[a-zA-Z\s.]+$/;

if(document.frmcustomer.firstname.value=="")

{

alert("Kindly select Firstname ...");

document.frmcustomer.firstname.focus();

return false;

}

else if(!document.frmcustomer.firstname.value.match(alphaExp))

{

alert("Kindly enter only characters in customer field..");

document.frmcustomer.firstname.focus();

return false;

}

else if(document.frmcustomer.lastname.value=="")

{

alert("Kindly select Lastname...");

document.frmcustomer.lastname.focus();

return false;

}

else if(!document.frmcustomer.lastname.value.match(alphaExp))

{

alert("Kindly enter only characters in customer field..");

document.frmcustomer.lastname.focus();

return false;

}

else if(document.frmcustomer.dob.value=="")

{

alert("Kindly select Date of Birth...");

document.frmcustomer.dob.focus();

return false;

}

else if(document.frmcustomer.gender.value=="")

{

alert("Kindly select Gender...");

document.frmcustomer.gender.focus();

return false;

}

else if(document.frmcustomer.address.value=="")

{

alert("Kindly select Address...");

document.frmcustomer.address.focus();

return false;

}

else if(document.frmcustomer.state.value=="Select")

{

alert("Kindly select State...");

document.frmcustomer.state.focus();

return false;

}

else if(document.frmcustomer.country.value=="Select")

{

alert("Kindly select Country...");

document.frmcustomer.country.focus();

return false;

}

else if(document.frmcustomer.pincode.value=="")

{

alert("Kindly select Pincode...");

document.frmcustomer.pincode.focus();

return false;

}

else if(!document.frmcustomer.pincode.value.match(numericExpression))

{

alert("Kindly enter numeric value in the PIN code..");

document.frmcustomer.pincode.focus();

return false;

}

else if(document.frmcustomer.contact.value=="")

{

alert("Kindly select Contact...");

document.frmcustomer.contact.focus();

return false;

}

else if(!document.frmcustomer.contact.value.match(numericExpression))

{

alert("Kindly enter numeric value in the contact number..");

document.frmcustomer.contact.focus();

return false;

}

else if(document.frmcustomer.mobile.value=="")

{

alert("Kindly select Mobile...");

document.frmcustomer.mobile.focus();

return false;

}

else if(!document.frmcustomer.mobile.value.match(numericExpression))

{

alert("Kindly enter numeric value in the mobile number..");

document.frmcustomer.mobile.focus();

return false;

}

else if(document.frmcustomer.emailid.value=="")

{

alert("Kindly select Email id...");

document.frmcustomer.emailid.focus();

return false;

}

else if(document.frmcustomer.loginid.value=="")

{

alert("Kindly select Login id...");

document.frmcustomer.loginid.focus();

return false;

}

else if(document.frmcustomer.password.value=="")

{

alert("Kindly select Password..");

document.frmcustomer.password.focus();

return false;

}

else if(document.frmcustomer.password.value.length < 6)

{

alert("Password should be more than 6 characters...");

document.frmcustomer.password.focus();

return false;

}

else if(document.frmcustomer.confirmpassword.value!=document.frmcustomer.password.value)

{

alert("Password and Confirm password not matching...");

document.frmcustomer.confirmpassword.focus();

return false;

}

else

{

return true;

}

}

</script>

**View Hotel Page**

<?php

session\_start();

if(!isset($\_SESSION[adminid]))

{

header("Location: adminlogin.php");

}

include("header.php");

include("dbconnection.php");

if(isset($\_GET[delid]))

{

$sql = "DELETE FROM hotel WHERE hoteid='$\_GET[delid]'";

$qsql = mysqli\_query($con,$sql);

if(mysqli\_affected\_rows($con) ==1)

{

echo "<script>alert('Hotel record deleted successfully..');</script>";

}

}

?>

<div class="page-title-container">

<div class="container">

<div class="page-title pull-left">

<h2 class="entry-title">View Hotel Records</h2>

</div>

</div>

</div>

<section id="content" class="gray-area">

<div class="container">

<div class="row">

<div id="main" class="col-sm-8 col-md-9">

<div class="booking-section travelo-box">

<div style='overflow:auto; width:825px;height:400px;'>

<table width="826" border="1" class="hoverTable">

<tr>

<th scope="col" >Company/Location/Hotel</th>

<th scope="col">Hotel Address</th>

<th scope="col" width="500">Hotel Description</th>

<th scope="col" width="100">Action</th>

</tr>

<?php

$sql = "SELECT \* FROM hotel";

$resdisplay = mysqli\_query($con,$sql);

while($rs = mysqli\_fetch\_array($resdisplay))

{

?>

<tr>

<td>&nbsp;<strong>Company:</strong> <?php

$sql1 = "SELECT \* FROM company WHERE companyid='$rs[companyid]'";

$resdisplay1 = mysqli\_query($con,$sql1);

$rs1 = mysqli\_fetch\_array($resdisplay1);

echo $rs1[companyname];

?> <br />

&nbsp;<strong>Location:</strong> <?php

$sql3 = "SELECT \* FROM location WHERE locationid='$rs[locationid]'";

$resdisplay3 = mysqli\_query($con,$sql3);

$rs3= mysqli\_fetch\_array($resdisplay3);

echo $rs3[location];

?><br />

&nbsp;<strong>Hotel:</strong> <?php echo $rs[hotelname] ?> </td>

<td>&nbsp;

<?php echo $rs[hoteladdress]; ?> <br />

&nbsp;&nbsp;<?php echo "PIN : ".$rs[pincode]; ?><br />

&nbsp;&nbsp;<?php echo "Landmark : ".$rs[landmark]; ?> </td>

<td>&nbsp;<?php echo $rs[hoteldescription]; ?> </td>

<td valign="top" align="center">

&nbsp;<a href='hotel.php?editid=<?php echo $rs[hoteid]; ?>'>Edit</a> <br />

&nbsp;<a href='viewhotel.php?delid=<?php echo $rs[hoteid]; ?>'>Delete</a> <br />

&nbsp; <a href="hotelimage.php?viewid=<?php echo $rs[hoteid]; ?>">View Images</a>

<br />&nbsp; <a href="viewroomtype.php?viewid=<?php echo $rs[hoteid]; ?>">View Room types</a>

<br />&nbsp; <a href="viewroom.php?viewid=<?php echo $rs[hoteid]; ?>">Add/View Rooms</a>

</td>

</tr>

<?php

}

?>

</table>

</div>

</div>

</div>

<?php

include("rightsidebar.php");

?>

</div>

</div>

</section>

<?php

include("footer.php");

?>

**TESTING PHASE**

**CHAPTER 8**

**TESTING PHASE**

* 1. **Introduction**

Testing is the process of detecting errors. Testing performs a very special role for quantity assurance and for ensuring the reliability of software. The results of testing are used later on during maintenance also.

* 1. **Psychology of testing**

The aim of testing is often to demonstrate that a program works by showing that it has no errors. The basic purpose of testing phase is to detect the errors that may be present in the program. Hence one should not start testing with the intent to show that a program doesn’t work. Testing is the process of executing a program with the intent of finding errors.

* 1. **Testing Objectives**

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Starting formally, we can say, testing is a process of executing a program with the intent of finding error.

* A successful test is one that uncovers an as yet undiscovered error.
* A good test case is one that has a high probability of finding error, if it exists.
* The tests are inadequate to detect possibly present errors.
* The software more or less confirms to quality and reliable standards.
  1. **System Testing**

Software testing is a critical element of software quantity assurance and represents the ultimate reviews of specifications, design and coding. The testing phase involves the testing of system using various test data. Preparation of test data plays a vital role in the system testing. After preparation the test data, the system under study is tested.

Those test data, errors were found and corrected following testing steps and corrections are recorded for future references. Thus a series testing is performed on the system before it is ready for implementation.

* 1. **Levels of testing**
* Unit testing
* Integrated testing
* Validation testing
* Output testing
* User Acceptance testing

**Unit Testing**

Unit testing focuses verification effort on the smallest unit of software unit of software i.e. the module. Using detailed design and the process specification testing is done to uncover errors within the boundary of the module. All modules must be successful in the unit test before the start of the integration testing begins.

In this project each service can be thought of as a module. There are basic modules. Giving different sets of inputs has tested each module. When developing the module as well as finishing the development so that each module works without any error. The inputs are validated when accepting from the user.

In this application developer tests the programs up as system. Software units in a system are the modules and routines that are assembled and integrate, to form a specific function. Unit testing is first done on modules, independent of one another to locate errors. This enables to detect errors.

**Integrated Testing**

After the unit testing we have to perform integration testing. The goal here is to see if modules can be integrated properly, the emphasis being on testing interfaces between modules. This testing activity can be considered as testing the design and hence the emphasis on testing module interactions. In this project integrating all the modules, I have checked whether the integration effects working of any of the services by giving different combinations of inputs.

**Validation Testing**

At the culmination of the integration testing, the software was completely assembled as a package, interfacing errors have been uncovered and corrected and a final series of software validation testing began. Here we test the system in a manner that can be reasonable expected by the customer; the system was tested against system requirement specification.

**Output Testing**

After performing validation test the next phase is output test of the system, since no system could be useful if it does not produce the desired output in desired format. By considering the format of the report/output, output/report is generated or displayed and is tested. Here output format is considered in two ways: one is the screen and other is on printed form.

**User Acceptance Testing**

Acceptance test is performed with realistic data of the client to demonstrate that the software is working satisfactory. Testing here is focused on external behavior of the system; the internal logic of program is not emphasized. Test cases should be selected so that the largest number of attributes of an equivalence class is exercised at once.

The testing phase is an important part of the software developed. It is the process of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied.

* 1. **Test Cases**

**Login page**

|  |  |  |  |
| --- | --- | --- | --- |
| SERIAL NO | TEST CONDITION | EXPECTED RESULT | RESULT |
| 1. | When the customer clicks “Log in” | The Custom Home Page to be displayed | Successful |
| 2. | When the customer clicks “Log in” without entering valid loginid or password or without entering any record | An error message to be displayed | Successful |

**Signup page**

|  |  |  |  |
| --- | --- | --- | --- |
| SERIAL NO | TEST CONDITION | EXPECTED RESULT | RESULT |
| 1. | When the user clicks “Register” | The Customer record is inserted | Successful |
| 2. | When the user clicks “Register” without entering first name | Kindly select Firstname | Successful |
| 3. | When the user clicks “Register” without entering characters in the first name | Kindly enter only characters in customer field | Successful |
| 4. | When the user clicks “Register” without entering last name | Kindly select Lastname | Successful |
| 5. | When the user clicks “Register” without entering characters in the last name | Kindly enter only characters in customer field | Successful |
| 6. | When the user clicks “Register” without entering date of birth | Kindly select Date of Birth | Successful |
| 7. | When the user clicks “Register” without entering address. | Kindly select address | Successful |
| 8. | When the user clicks “Register” without entering state. | Kindly select state | Successful |
| 9. | When the user clicks “Register” without entering country. | Kindly select country | Successful |
| 10. | When the user clicks “Register” without entering pincode. | Kindly select pincode | Successful |
| 11. | When the user clicks “Register” without entering numeric value for pincode. | Kindly enter numeric value in the PIN code | Successful |
| 12. | When the user clicks “Register” without entering contact | Kindly select contact | Successful |
| 13. | When the user clicks “Register” without entering numeric value for contact | Kindly enter numeric value in the contact number | Successful |
| 14. | When the user clicks “Register” without entering mobile | Kindly select mobile | Successful |
| 15. | When the user clicks “Register” without entering numeric value for mobile | Kindly enter numeric value in the mobile number | Successful |
| 16. | When the user clicks “Register” without entering email id | Kindly select email id | Successful |
| 17. | When the user clicks “Register” without entering email id with @ and domain name following it | Kindly enter a complete email id | Successful |
| 18. | When the user clicks “Register” without entering login id | Kindly select Login id. | Successful |
| 19. | When the user clicks “Register” without entering password | Kindly select Password | Successful |
| 20. | When the user clicks “Register” with password shorter than 6 characters | Password should be more than 6 characters | Successful |
| 21. | When the user clicks “Register” when the password and confirm password are not matching | Password and Confirm password not matching. | Successful |

**Admin login page**

|  |  |  |  |
| --- | --- | --- | --- |
| SERIAL NO | TEST CONDITION | EXPECTED RESULT | RESULT |
| 1 | When the admin clicks “submit” | Login is successful and the page redirects to dashboard | Successful |
| 2 | When the admin clicks “submit” without entering any valid login id or password | Failed to login. Kindly enter again | Successful |

**Admin page**

|  |  |  |  |
| --- | --- | --- | --- |
| SERIAL NO | TEST CONDITION | EXPECTED RESULT | RESULT |
| 1 | When the admin clicks “submit | The admin record is inserted | Successful |
| 2 | When the admin clicks “submit” without entering admin name | Please enter Admin name | Successful |
| 3. | When the admin clicks “submit” without entering characters in admin name | Kindly enter only characters in admin field | Successful |
| 3 | When the admin clicks “submit” without entering login id | Please enter Login ID | Successful |
| 4. | When the admin clicks “submit” without entering password | Please enter the password. | Successful |
| 5. | When the admin clicks “submit” with password shorter than 6 characters | Password should be more than 6 characters. | Successful |
| 6. | When the user clicks “submit” when the password and confirm password are not matching | Password and Confirm password not matching. | Successful |

**Company page**

|  |  |  |  |
| --- | --- | --- | --- |
| SERIAL NO | TEST CONDITION | EXPECTED RESULT | RESULT |
| 1 | When the admin clicks “Submit” | The company record is inserted | Successful |
| 2 | When the admin clicks “submit” without entering company type | Please enter Company type. | Successful |
| 3 | When the admin clicks “submit” without entering company name | Please enter Company name. | Successful |
| 4. | When the admin clicks “submit” without entering characters in company name | Kindly enter only characters in company field | Successful |
| 5. | When the admin clicks “submit” without entering company logo | Please provide company logo | Successful |
| 6. | When the admin clicks “submit” without selecting status | Please select the Status. | Successful |

**Location page**

|  |  |  |  |
| --- | --- | --- | --- |
| SERIAL NO | TEST CONDITION | EXPECTED RESULT | RESULT |
| 1. | When the admin clicks “submit” | The location record is inserted | Successful |
| 2 . | When the admin clicks “submit” without entering location | Kindly provide location. | Successful |
| 3. | When the admin clicks “submit” without entering characters in location | Kindly enter only characters in location field. | Successful |
| 4. | When the admin clicks “submit” without entering location type | Kindly provide location type. | Successful |
| 5. | When the admin clicks “submit” without entering location title | Kindly provide location title. | Successful |
| 6. | When the admin clicks “submit” without entering characters location title | Kindly enter only characters in location field. | Successful |
| 7. | When the admin clicks “submit” without entering status | Kindly select status.. | Successful |

**Service page**

|  |  |  |  |
| --- | --- | --- | --- |
| SERIAL NO | TEST CONDITION | EXPECTED RESULT | RESULT |
| 1. | When the admin clicks “submit” | The service record will be inserted | Successful |
| 2. | When the admin clicks “submit” without entering service type | Kindly specify service type. | Successful |
| 3. | When the admin clicks “submit” without entering characters in service type | Kindly enter only characters in service field. | Successful |
| 4. | When the admin clicks “submit” without entering service type | Kindly specify service title. | Successful |
| 5. | When the admin clicks “submit” without entering characters in service type | Kindly enter only characters in service field. | Successful |
| 6. | When the admin clicks “submit” without entering image | Kindly provide image. | Successful |
| 7. | When the admin clicks “submit” without entering cost | Cost should not be empty. | Successful |
| 8. | When the admin clicks “submit” without selecting tariff type | Tariff type should not be empty. | Successful |
| 9. | When the admin clicks “submit” without selecting units | Unit should not be empty. | Successful |
| 10. | When the admin clicks “submit” without selecting status | Select status | Successful |

**Hotel page**

|  |  |  |  |
| --- | --- | --- | --- |
| SERIAL NO | TEST CONDITION | EXPECTED RESULT | RESULT |
| 1. | When the admin clicks “submit” | The hotel record will be inserted | Successful |
| 2. | When the admin clicks “submit” without entering company | Kindly select company type. | Successful |
| 3. | When the admin clicks “submit” without entering location | Kindly select the location. | Successful |
| 4. | When the admin clicks “submit” without entering hotel name | Kindly provide the hotel name. | Successful |
| 5. | When the admin clicks “submit” without entering characters in hotel name | Kindly enter only characters in hotel name field. | Successful |
| 6. | When the admin clicks “submit” without entering hotel address | Hotel address should not be empty. | Successful |
| 7. | When the admin clicks “submit” without entering pin code | Pin code should not be empty. | Successful |
| 8. | When the admin clicks “submit” without entering numbers in Pin code. | Kindly enter numeric value in the Pin code. | Successful |
| 9. | When the admin clicks “submit” without selecting landmark | Landmark should not be empty. | Successful |

**Tourpackage travel options page**

|  |  |  |  |
| --- | --- | --- | --- |
| SERIAL NO | TEST CONDITION | EXPECTED RESULT | RESULT |
| 1. | When the admin clicks “submit” | The Tourpackage travel options record will be inserted | Successful |
| 2. | When the admin clicks “submit” without entering the type of tourpackage | Kindly select tour package. | Successful |
| 3. | When the admin clicks “submit” without entering the source location | Kindly select source location. | Successful |
| 4. | When the admin clicks “submit” without entering the destination location | Kindly select destination location. | Successful |
| 5. | When the admin clicks “submit” without entering the travel type | Kindly select travel type. | Successful |

**Tourpackage place page**

|  |  |  |  |
| --- | --- | --- | --- |
| SERIAL NO | TEST CONDITION | EXPECTED RESULT | RESULT |
| 1. | When the admin clicks “submit” | The Tourpackage place record will be inserted | Successful |
| 2. | When the admin clicks “submit ”without selecting the tourpackage | Kindly select tour package. | Successful |
| 3. | When the admin clicks “submit” without entering the contact | Contact should not be empty. | Successful |
| 4. | When the admin clicks “submit” without entering numeric value contact | Kindly enter numeric value in the contact. | Successful |
| 5. | When the admin clicks “submit” without selecting the status | Select Status. | Successful |

**Tourpackage itinerary page**

|  |  |  |  |
| --- | --- | --- | --- |
| SERIAL NO | TEST CONDITION | EXPECTED RESULT | RESULT |
| 1. | When the admin clicks “submit” | The Tourpackage itinerary record will be inserted | Successful |
| 2. | When the admin clicks “submit” without selecting the tourpackage | Kindly select tour package. | Successful |
| 3. | When the admin clicks “submit” without entering the no of days | Kindly select day\_no | Successful |
| 4. | When the admin clicks “submit” without entering the itinerary title | Kindly select itinerary\_title | Successful |
| 5. | When the admin clicks “submit” without entering characters in the itinerary title | Kindly enter only characters in tourpackage itinerary field. | Successful |

**Tourpackage page**

|  |  |  |  |
| --- | --- | --- | --- |
| SERIAL NO | TEST CONDITION | EXPECTED RESULT | RESULT |
| 1. | When the admin clicks “submit” | The tourpackage record will be inserted | Successful |
| 2. | When the admin clicks “submit” without entering company type | Kindly select company type. | Successful |
| 3. | When the admin clicks “submit” without entering tourpackage type | Kindly select tourpackage type. | Successful |
| 4. | When the admin clicks “submit” without entering package title. | Kindly specify package title. | Successful |
| 5. | When the admin clicks “submit” without entering characters in package title. | Kindly enter only characters in tourpackage title field. | Successful |
| 6. | When the admin clicks “submit” without entering overview | Overview should not be empty. | Successful |
| 7. | When the admin clicks “submit” without entering inclusion | Inclusion should not be empty. | Successful |
| 8. | When the admin clicks “submit” without entering no of days | No\_of\_days should not be empty. | Successful |

**Train type page**

|  |  |  |  |
| --- | --- | --- | --- |
| SERIAL NO | TEST CONDITION | EXPECTED RESULT | RESULT |
| 1. | When the admin clicks “submit” | The train type record will be inserted | Successful |
| 2. | When the admin clicks “submit” without selecting train name | Please select train name | Successful |
| 3. | When the admin clicks “submit” without selecting departure location | Please enter Departure Location | Successful |
| 4. | When the admin clicks “submit” without selecting arrival location | Please enter Arrival Location | Successful |
| 5. | When the admin clicks “submit” without selecting train type | Please enter Train type | Successful |
| 6. | When the admin clicks “submit” without entering the train type code. | Please enter train type code. | Successful |
| 7. | When the admin clicks “submit” without entering number of seats | Please enter Number of seats | Successful |
| 8. | When the admin clicks “submit” without entering the train tariff | Please enter train\_tariff | Successful |
| 9. | When the admin clicks “submit” without entering the numeric value for train tariff | Kindly enter numeric value in the train tariff. | Successful |
| 10. | When the admin clicks “submit” without selecting the status | Please select status. | Successful |

**Train page**

|  |  |  |  |
| --- | --- | --- | --- |
| SERIAL NO | TEST CONDITION | EXPECTED RESULT | RESULT |
| 1. | When the admin clicks “submit” | The train record will be inserted | Successful |
| 2. | When the admin clicks “submit” without entering train name | Kindly select Train name | Successful |
| 3. | When the admin clicks “submit” without entering characters in train name | Kindly enter only characters in train field | Successful |
| 4. | When the admin clicks “submit” without selecting departure location | Kindly select Departure location. | Successful |
| 5. | When the admin clicks “submit” without selecting arrival location | Kindly select Arrival location. | Successful |
| 6. | When the admin clicks “submit” without entering the duration | Kindly select Duration. | Successful |
| 7. | When the admin clicks “submit” without entering the travel date | Kindly select Travel date. | Successful |
| 8. | When the admin clicks “submit” without entering the travel time | Kindly select Travel time. | Successful |

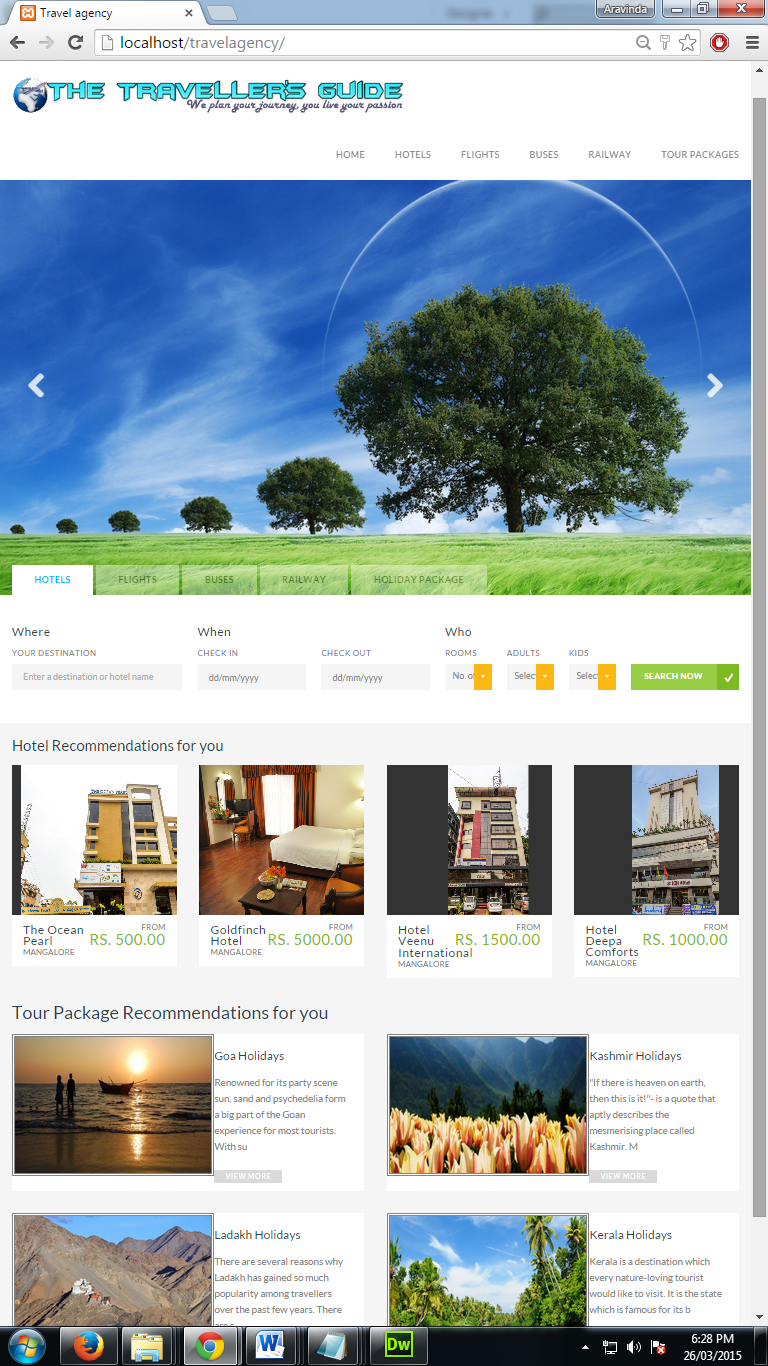
**Bus page**

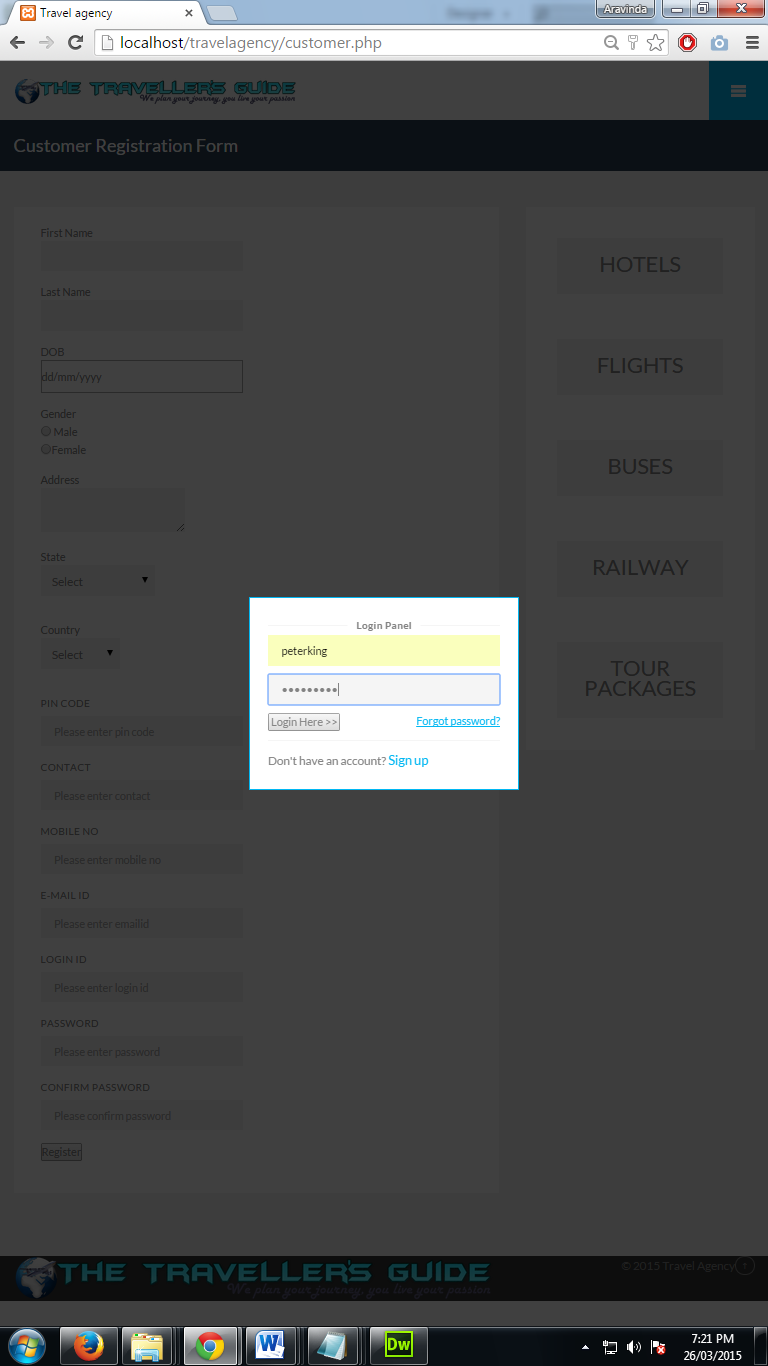
|  |  |  |  |
| --- | --- | --- | --- |
| SERIAL NO | TEST CONDITION | EXPECTED RESULT | RESULT |
| 1. | When the admin clicks “Submit” | The Bus record is inserted | Successful |
| 2. | When the admin clicks “Submit” without selecting company | Kindly select company. | Successful |
| 3. | When the admin clicks “Submit” without selecting the bus type | Kindly select the bus type | Successful |
| 4. | When the admin clicks “Submit” without entering the number of sleeper seats | Please specify the number of sleeper seats. | Successful |
| 5. | When the admin clicks “Submit” without entering the number of seater seats | Please specify the number of seater seats. | Successful |
| 6. | When the admin clicks “Submit” without selecting the status | Kindly select status | Successful |

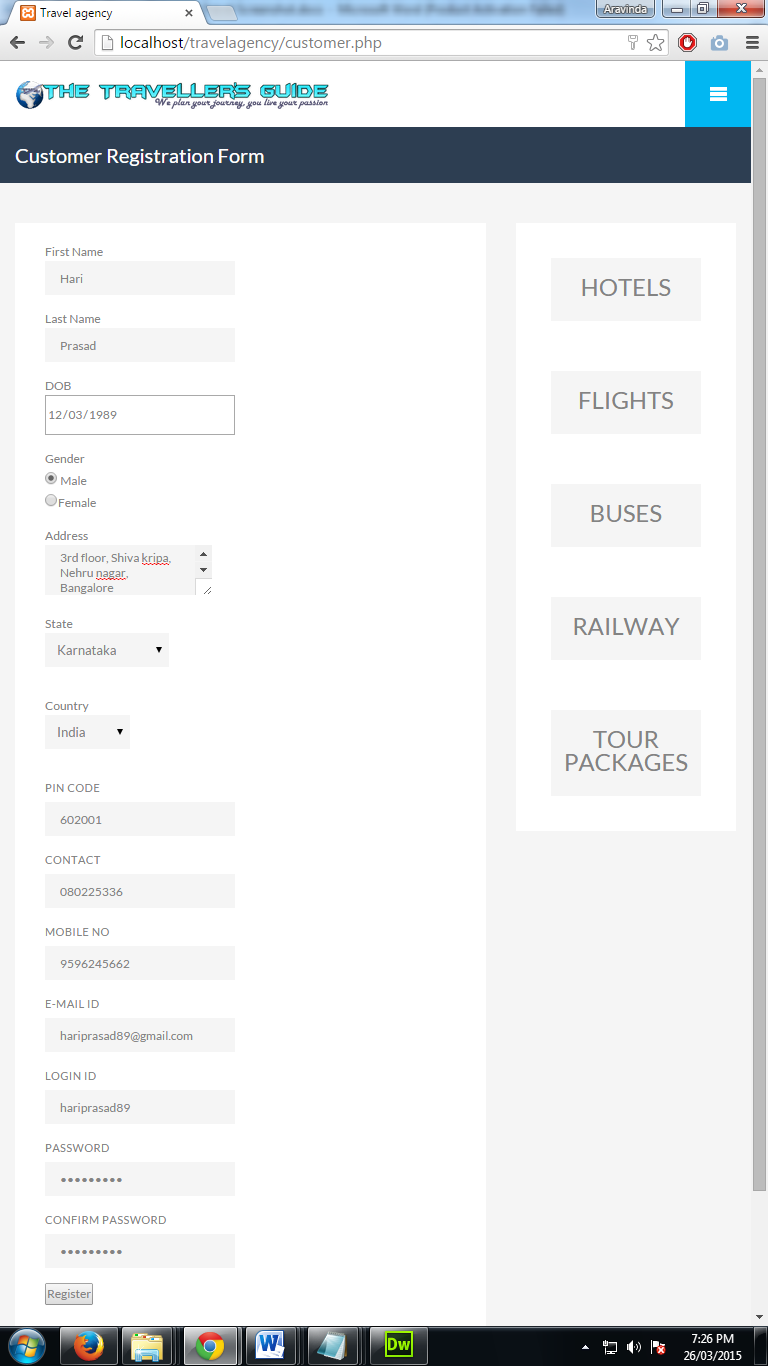
**Flight page**

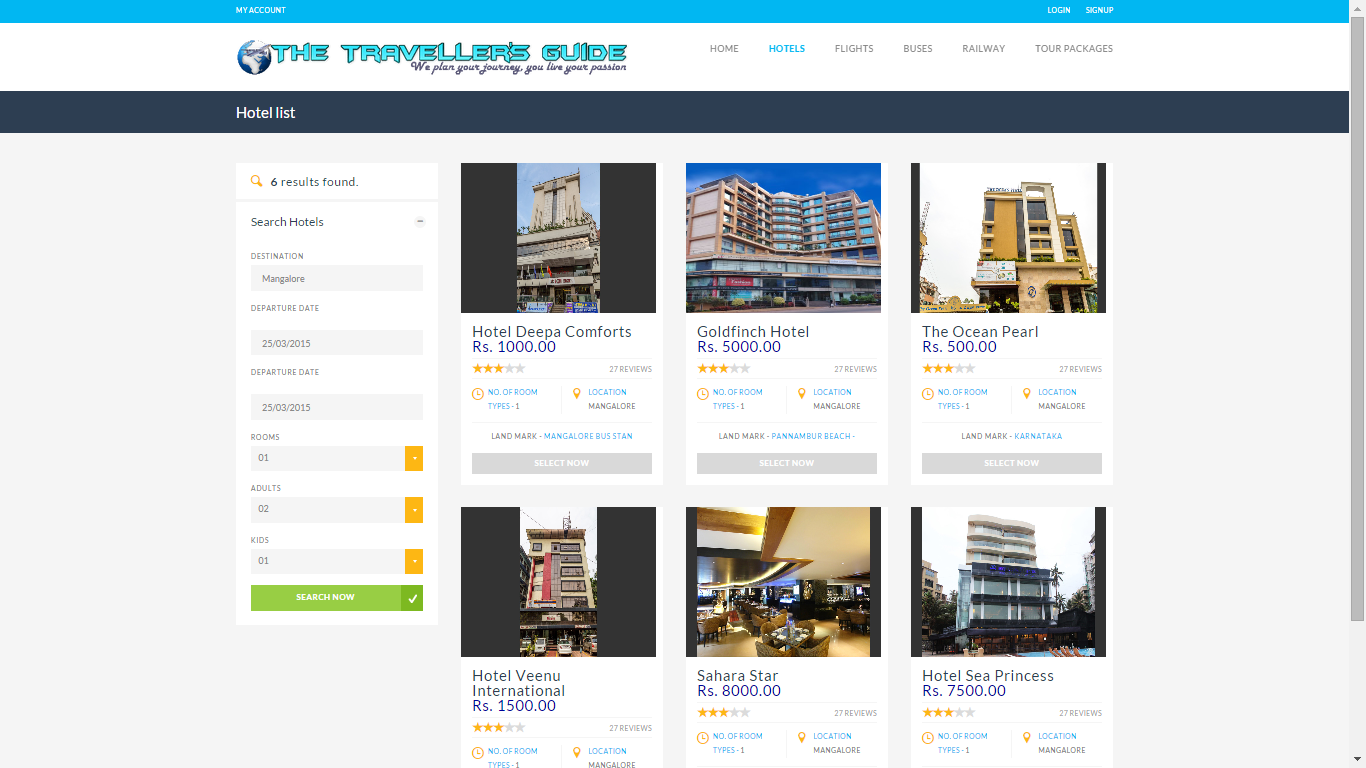
|  |  |  |  |
| --- | --- | --- | --- |
| SERIAL NO | TEST CONDITION | EXPECTED RESULT | RESULT |
| 1. | When the admin clicks “Submit” | The Flight record is inserted | Successful |
| 2. | When the admin clicks “Submit” without selecting company | Kindly select company. | Successful |
| 3. | When the admin clicks “Submit” without selecting departure location | Kindly select Departure location. | Successful |
| 4. | When the admin clicks “Submit” without selecting arrival location. | Kindly select arrival location. | Successful |
| 5. | When the admin clicks “Submit” without entering duration | Duration should not be empty. | Successful |
| 6. | When the admin clicks “Submit” without entering departure date | Departure date should not be empty. | Successful |
| 7. | When the admin clicks “Submit” without entering departure time | Departure time should not be empty. | Successful |
| 8. | When the admin clicks “Submit” without entering arrival date. | Arrival date should not be empty. | Successful |
| 9. | When the admin clicks “Submit” without entering arrival time. | Arrival time should not be empty. | Successful |
| 10. | When the admin clicks “Submit” without entering number of economy seats | Economy seats should not be empty. | Successful |
| 11. | When the admin clicks “Submit” without entering number of business class seats. | Business class seats should not be empty. | Successful |
| 12. | When the admin clicks “Submit” without entering first class seats | First class seats record should not be empty. | Successful |
| 13. | When the admin clicks “Submit” without entering premium economy seats | Premium economy seats should not be empty. | Successful |
| 14. | When the admin clicks “Submit” without entering economy tariff | Economy tariff should not be empty. | Successful |
| 15. | When the admin clicks “Submit” without entering business class tariff | Business class tariff should not be empty | Successful |
| 16. | When the admin clicks “Submit” without entering first class tariff | First class tariff should not be empty | Successful |
| 17. | When the admin clicks “Submit” without entering premium tariff | Premium tariff should not be empty. | Successful |
| 18. | When the admin clicks “Submit” without selecting the status | Kindly select status | Successful |

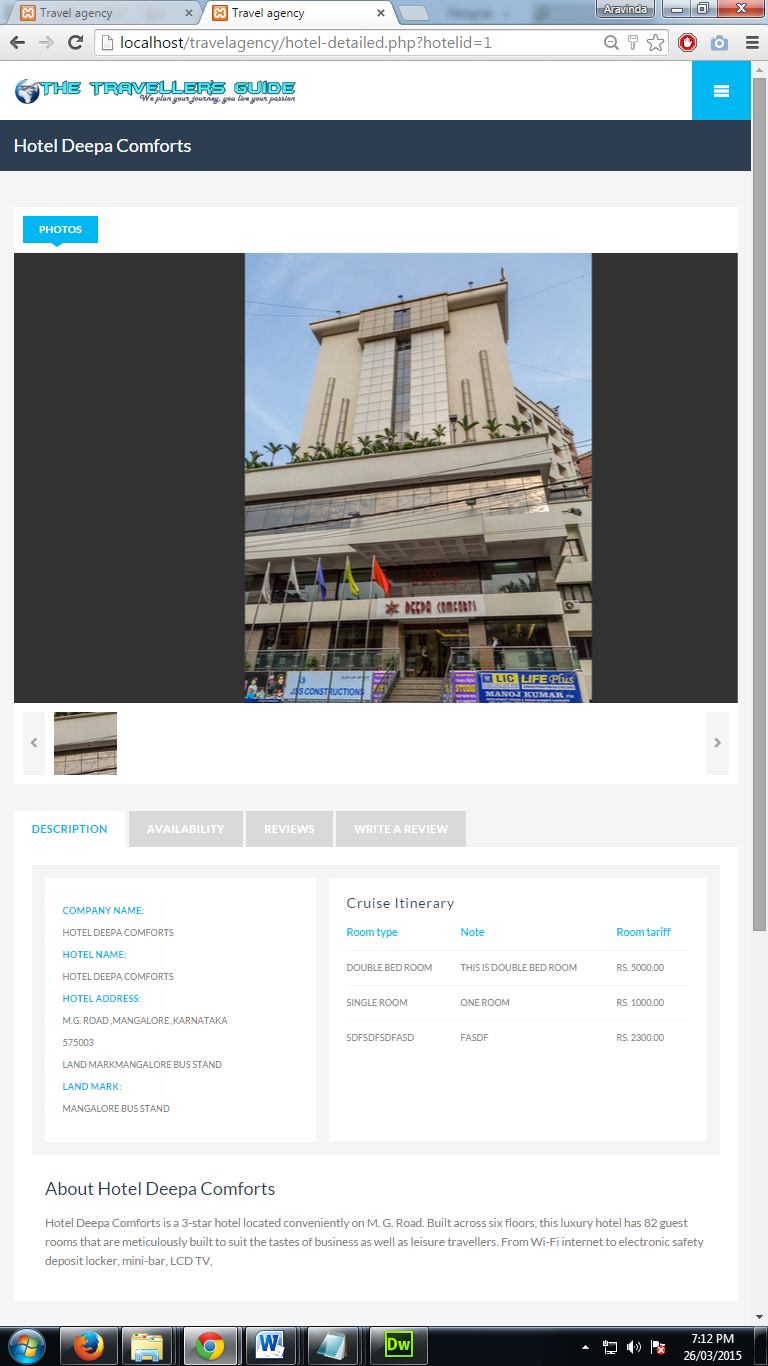
**SNAPSHOTS**

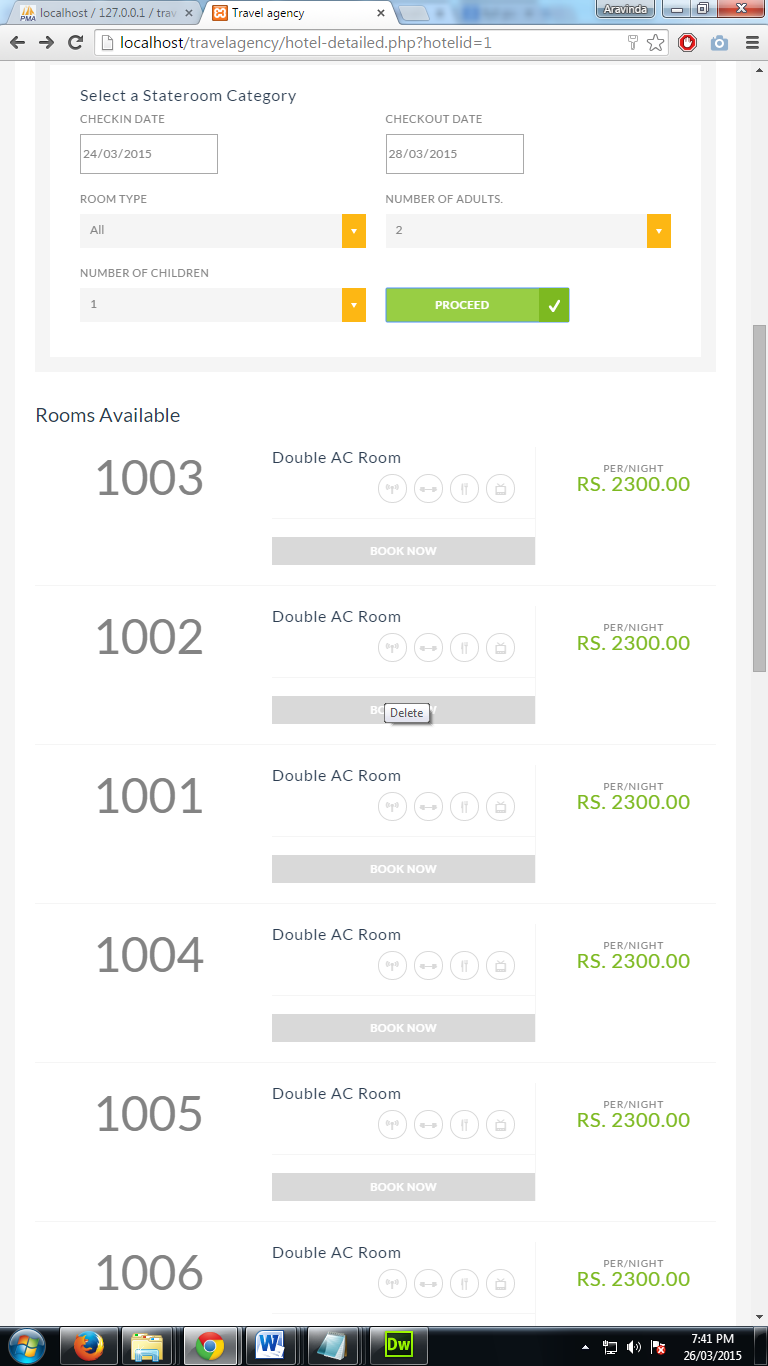
**Home Page:**  


**Login Page:**

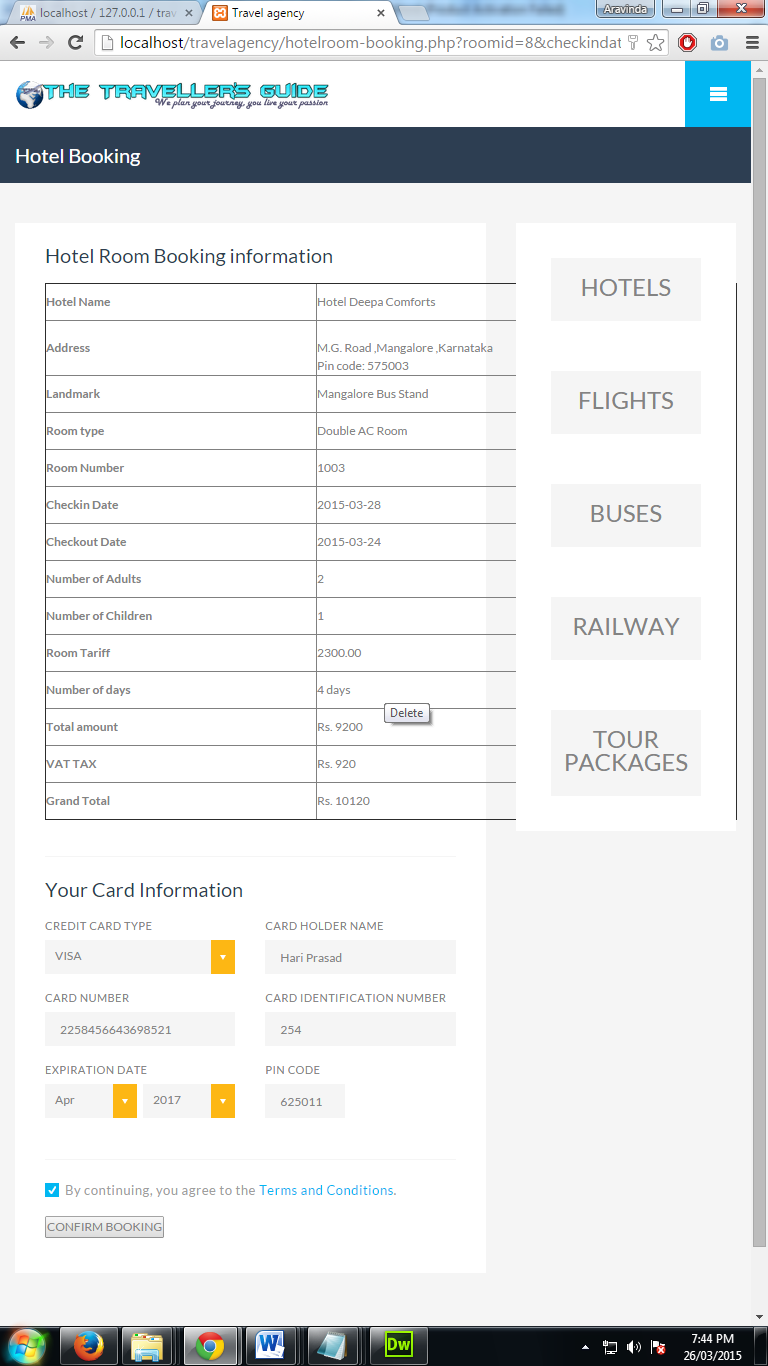
**Registration Page:**

**Hotel List:**

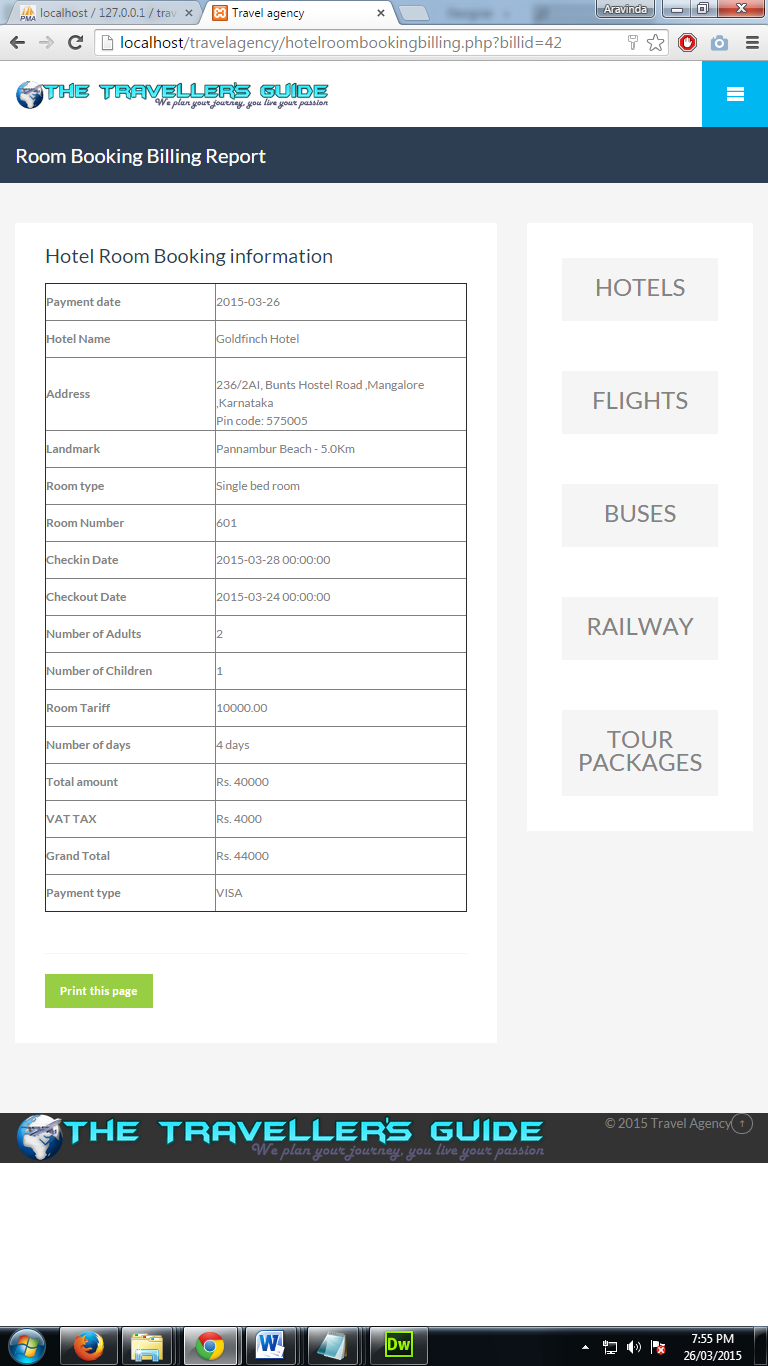
**Hotel details:**

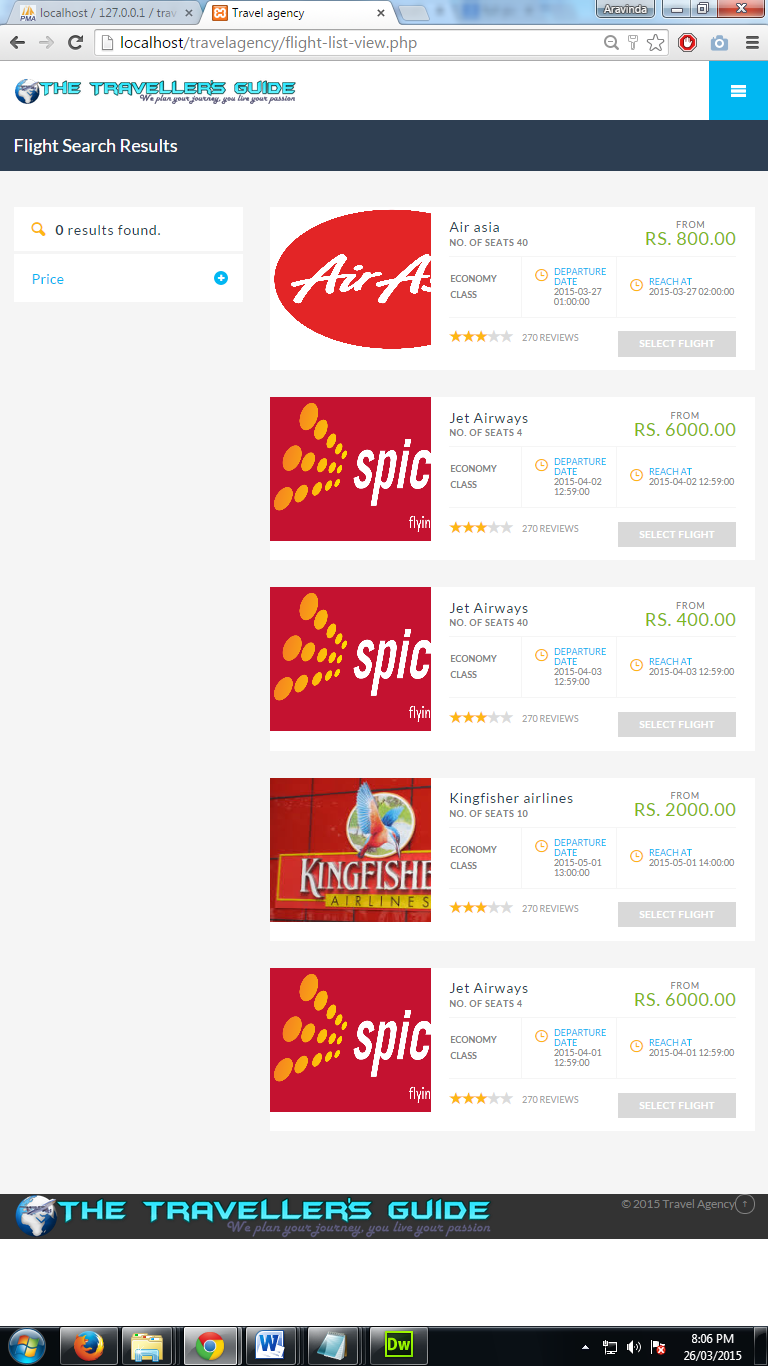
**Room availability search:**

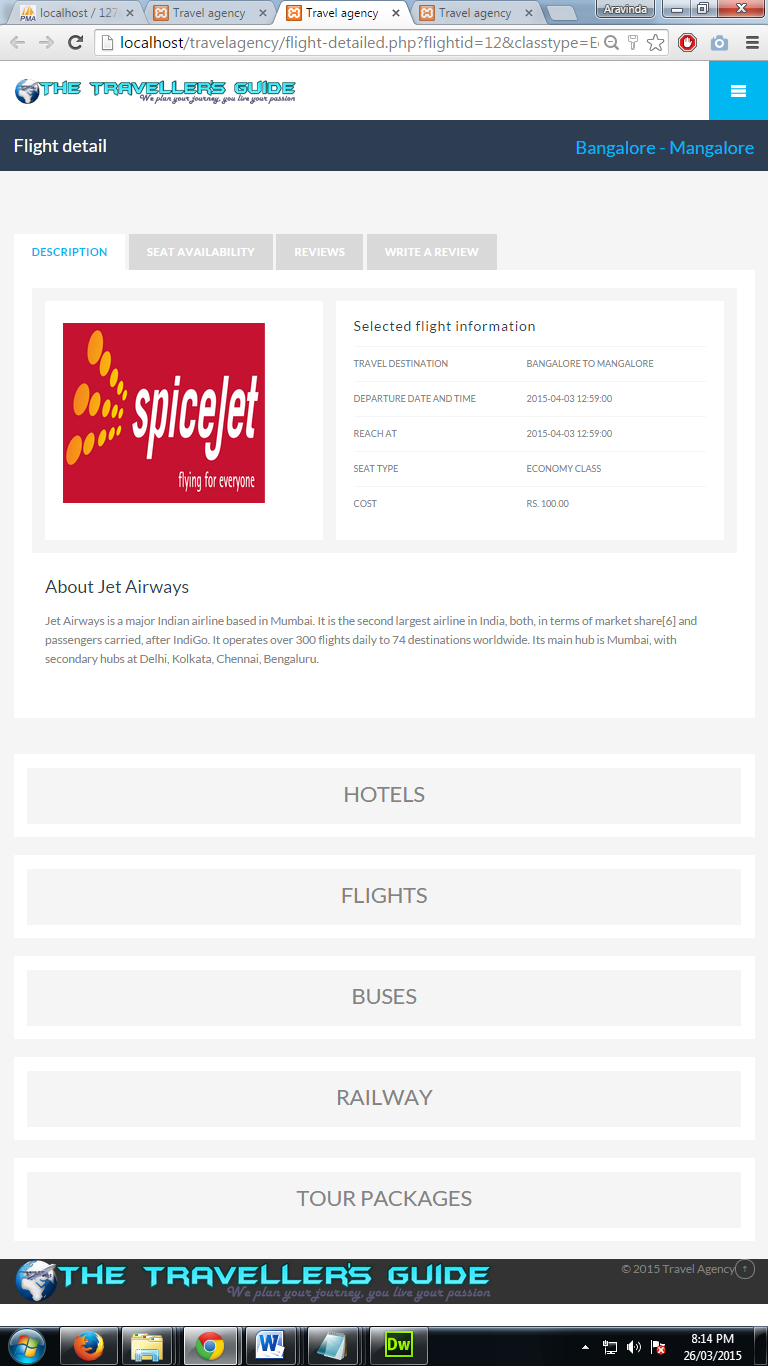
**Room Booking Panel:**

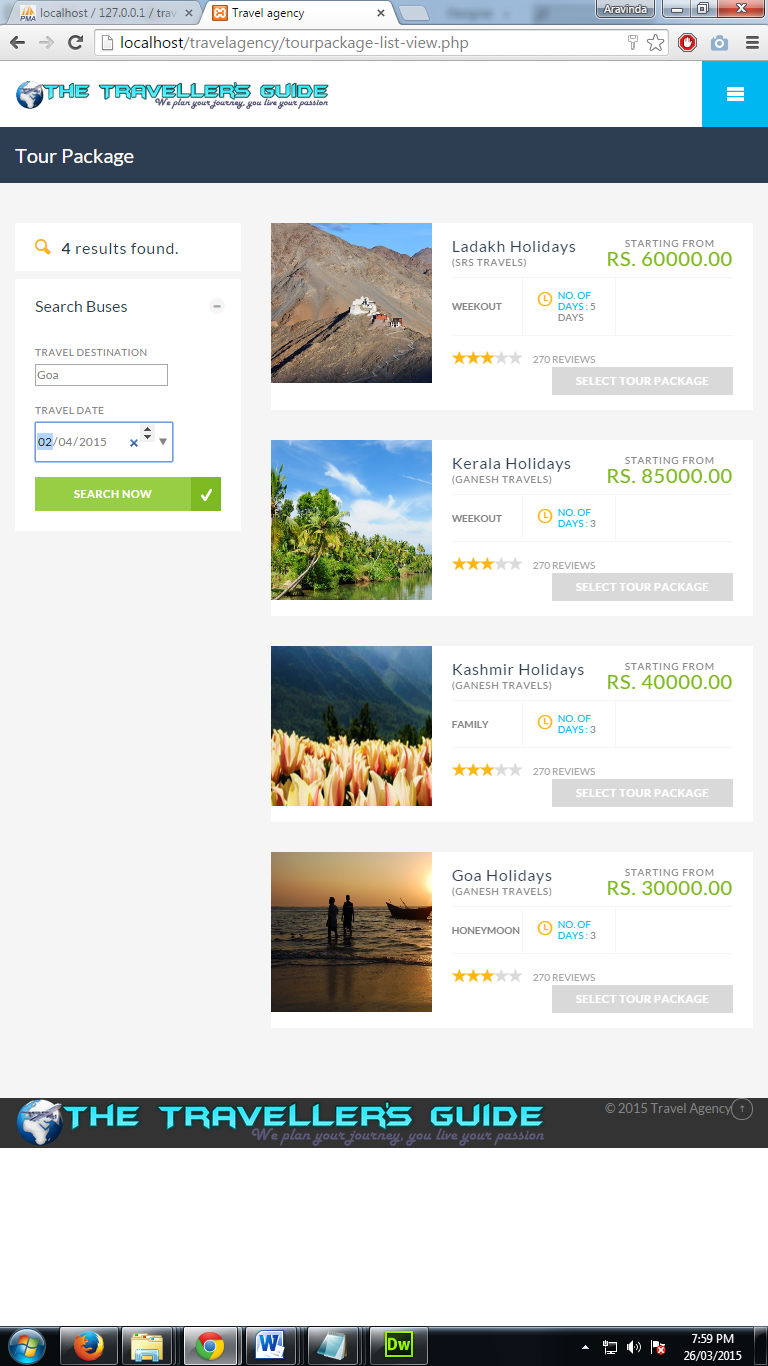


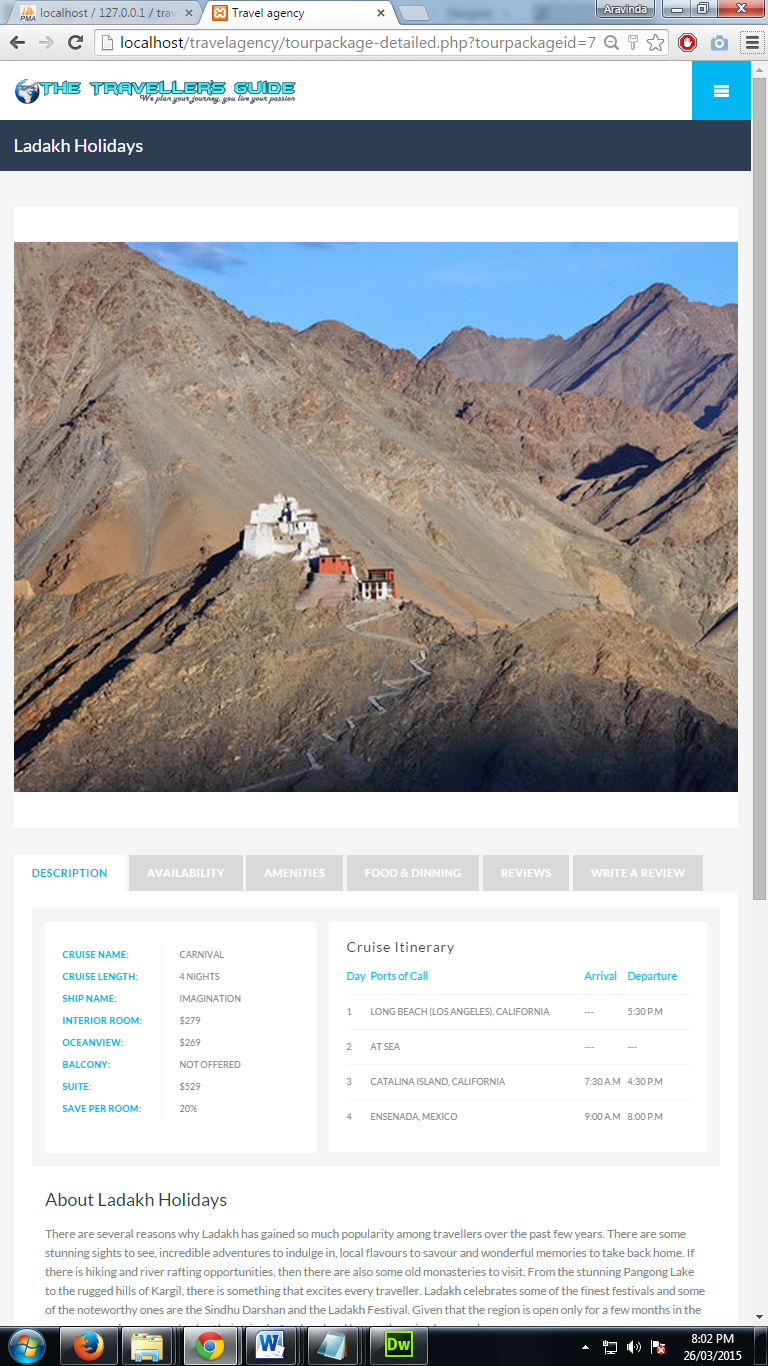
**Room Booking Payment report:**

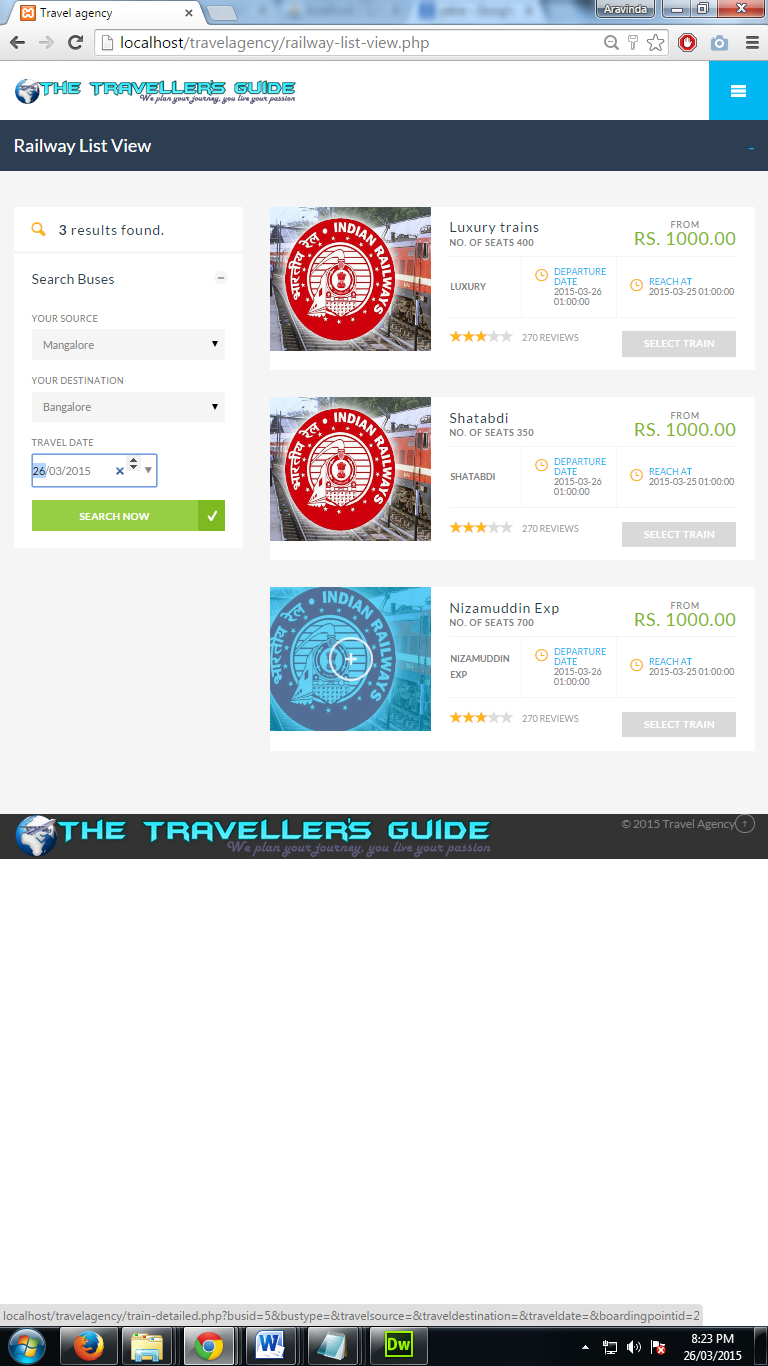


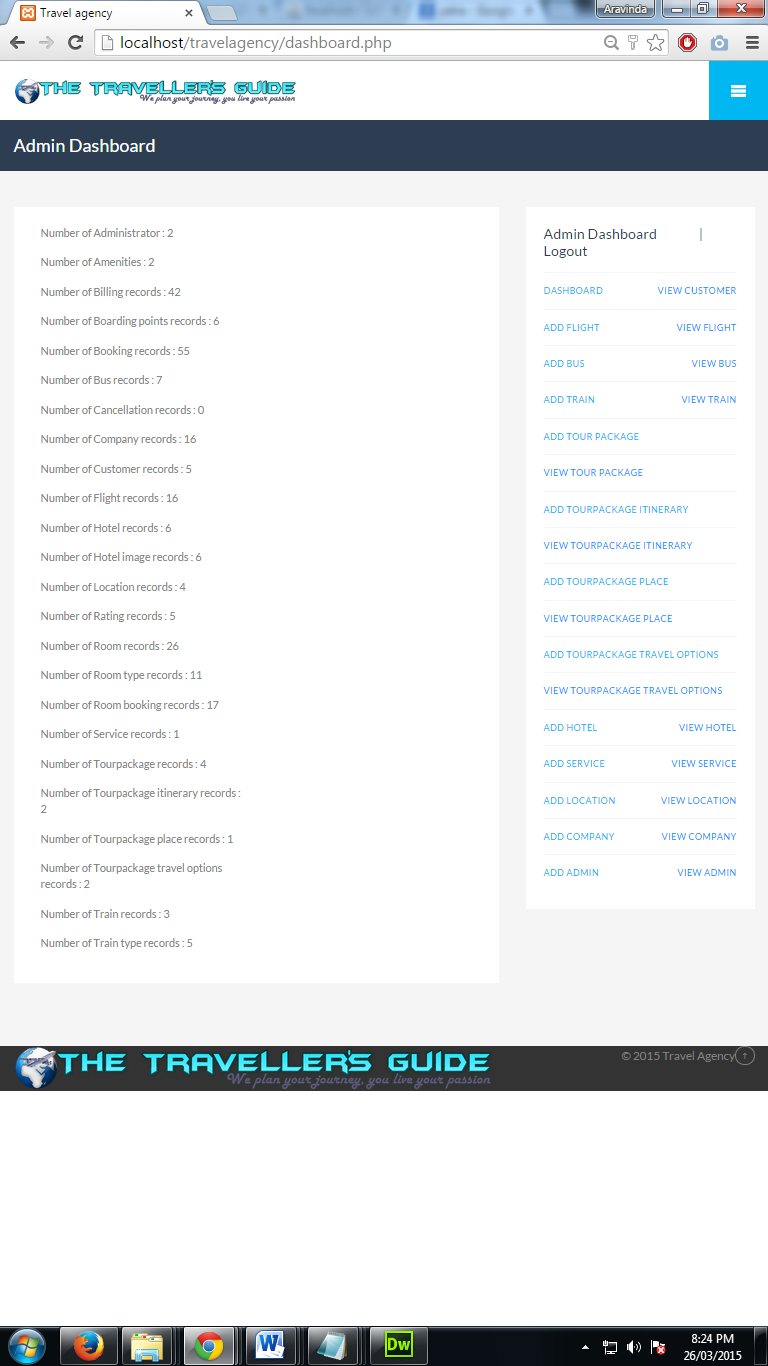
**Flight Search result:**   


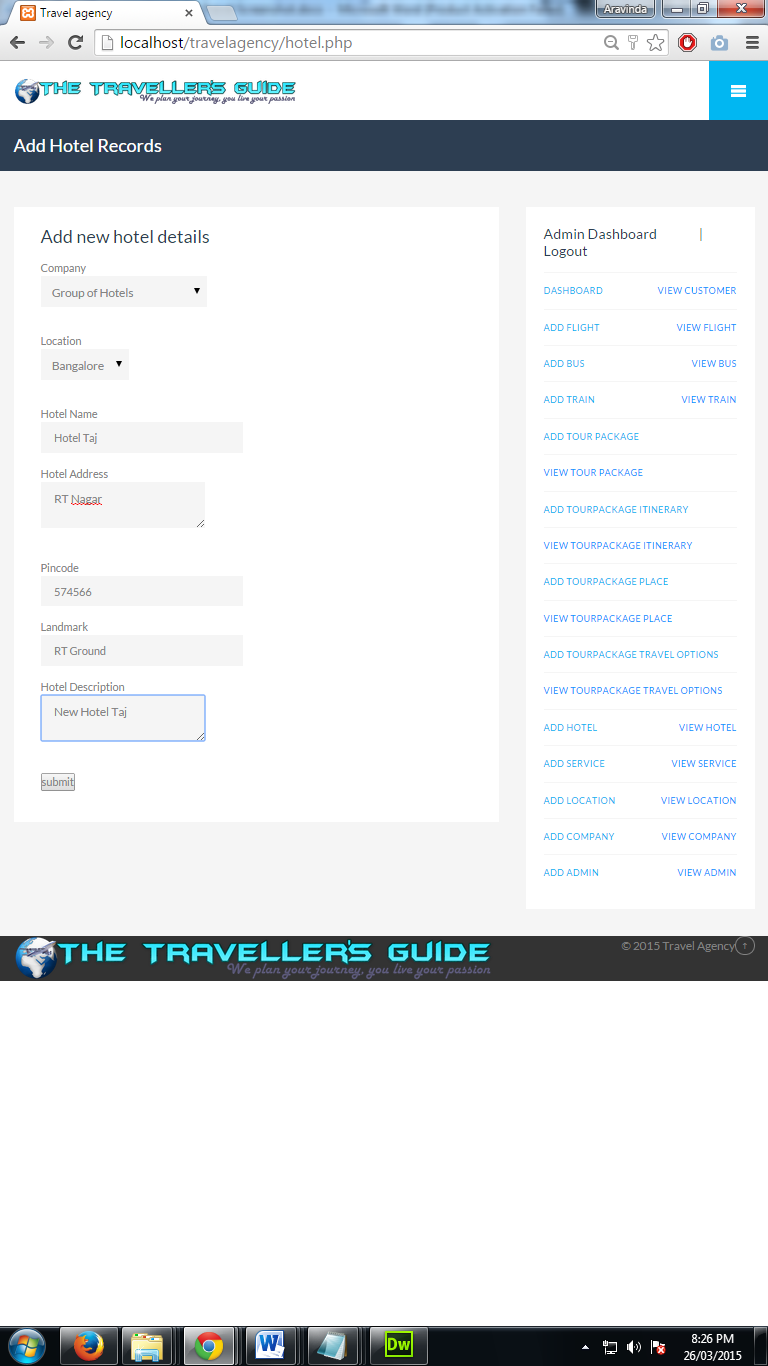
**Flight detailed view:**

**Tour Package list:**

**Tour Package detailed view:**

**Railway list view:**  


**Administrator Dashboard:**

**Add hotel records:**

**CONCLUSION**

**CHAPTER 10**

**CONCLUSION**

The project work titled **“The Traveller’s Guide”** has been designed using Hypertext Pre Processor (PHP) wherein, many user friendly form controls have been added in order to make it a user interactive application. The system is developed in such a way that the user with common knowledge of computers can handle it easily. The system forms a general platform for building most advanced travel systems.

The future enhancements to the system can be made as technology changes.

**FUTURE ENHANCEMENTS**

**CHAPTER 11**

**FUTURE ENHANCEMENTS**

The future scope and enhancement of the project are:-

* In future we can build a mobile application for smartphones.
* Currently the system supports only desktop pc and in future we can make it mobile browser compatible site
* The project can be made more users friendly in future.

**BIBLIOGRAPHY**

**CHAPTER 12**

**BIBLIOGRAPHY**

**Book references :**

* Pankaj Jalote, An Integrated Approach to Software Engineering, Third Edition, Narosa Publications.
* Learning php, mysql, javascript and css by Robin Nixon.

**Web references :**

* [www.w3schools.com](http://www.w3schools.com/)
* [www.tutorialspoint.com/**php**](http://www.tutorialspoint.com/php)
* [www.php.net](http://www.php.net/) – PHP Official site
* [www.mysql.com](http://www.mysql.com/) – MySQL Database official site