

“TRAVEL AND TOURISM MANAGEMENT SYSTEM”

TABLE OF CONTENT

Content	Page No.
Acknowledgement	3
Declaration	4
1. Introduction and objective of the project	
1.1 Introduction	7
1.2 Advantage	8
1.3 Existing Sysytem	9
1.4 Proposed System	10
2. System Study	
2.1. Preliminary Investigation	11
2.2. System Development Life Cycle	11
2.3. Economic feasibility	14
2.4. Technical feasibility	14
3. System Analysis	
3.1. Importance of Computerized System	15
3.2. System Analysis	17
4. System Design	
4.1. Phtsical Design	19
4.2. Data Flow Diagram	21
4.3. Description of the Modeling	21
4.4. ER Diagram	23

4.5. Context Level DFD	31
4.6. Data Modeling	41
5. System Development	
5.1. Source Code	44
5.2. Coding Samples	83
5.3. Optimization of Code	83
6. Testing	
6.1 Testing	84
6.2 Levels of Testing	85
6.3 Test cases	90
6.4 Verification & Validation	95
7. System Implementation	96
7.1 Hardware Specifications	97
7.2 Software Requirements	97
7.3 Conversion and Training	97
7.4 Training Needs	98
8. COST ESTIMATION OF THE PROJECT	99
9. Gant & Pert Chart	100
10. Scope of Future Application	103
1.1 Conclusion	104
11. Bibliography	104
12. Glossary	105

CHAPTER 1

INTRODUCTION AND OBJECTIVE OF THE PROJECT

1.1 INTRODUCTION

To make the system efficient and user friendly. “**Travel and Tourism Management System** ” has been designed to computerize the following functions that are performed by the Website. This site is about various travel agencies, their packages and information about the routes for different places in India. The person who visits the site can search for various travel agencies, routes that they follow and facilities provided by them. He can book the seats online for required class of buses. User can get confirmation through mail. Any travel agent can register himself to our site and only then he is authorised to give information. Since Tour and Travels is associated with the lives of common people and their day-to-day routines so I decided to work on this project.

The manual handling of the record is time consuming and highly prone to error. This project includes listing schedules of trains, flights, total seats reserved for a particular package, registration of tourists, hotels, guides, Arrival & departure transfers, Hotel bookings, Transport by road, train & domestic airlines, Sightseeing with guide, Monument entrances, Meals if requested. Updating and Modification system helps to add or delete record and modify the existing record set. Based on the nature of updating it provides appropriate access to the user. Performing these essential activities manually can be troublesome and time consuming at some instance. In the Proposed system, all these system activities can be performed quickly and efficiently.

This project is useful for

- ❑ The customers who want to reserve seats.
- ❑ The customers who want to get information about any train or flight scheduling.
- ❑ The travelers who want to book a package.
- ❑ Packages for fun travel.
- ❑ Packages for Holly trip.

- ❑ Packages for Cultural tours.
- ❑ Packages for Heritage tours
- ❑ Packages for Educational Tours.
- ❑ Hotel Bookings..
- ❑ Guide Booking.
- ❑ Ticket Reservations.
- ❑ Booking Information.
- ❑ Schedules of Trains and Flights (Arrivals and Departures).
- ❑ Information About different places.
- ❑ Register your agency(for agency owners)
- ❑ Register your self as a guide.
- ❑ Administrator module is second important module of this project.
- ❑ Administrator provides booking (Hotel, Guide, Transportation) informations to the users

OBJECTIVE:

- The objective of the project is to develop a system that automates the processes and activities of a travel and tourism agency.
- The purpose is to design a system using which one can perform all operations related to traveling and sight-seeing.

1.2 ADVANTAGE

I have designed the given proposed system in the JSP to automate the process of day to day activities of Travel Agency-Travel Mastii Pvt.Ltd.

The following steps that give the detailed information of the need of proposed system are:

Performance: During past several decades, the Travel and Tourism Management System is supposed to maintain manual handling of all the Travel Agency's daily activities. The manual handling of the record is time consuming and highly prone to error. To improve the performance of the Travel Agency, the online website is to be undertaken. The computerized Travel Agency project is fully computerized and user friendly.

Efficiency: The basic need of the project is efficiency. The project should be efficient so that whenever a user open this site he will get the knowledge about,

India Tours - with off the beaten track experiences.

Variety - in budget and deluxe tours, business visits, short stays, conferences and festivals.

Learning - with educational touch including lectures and unique exposures to Indian history and culture.

Expertise - in providing comprehensive travel services to FITs and Groups.

Attention - to the details that make travel arrangements mesh smoothly.

Value - helping you get the maximum mileage out of your available time and money.

Control: The complete control of the project is under the hands of authorized person who has the password to access this project and illegal access is not supposed to deal with. All the control is under the administrator and the other members have the rights to just see the records not to change any transaction or entry.

1.3 EXISTING SYSTEM:

- In the present system a customer has to approach various agencies to find details of places and to book tickets.
- This often requires a lot of time and effort.
- A customer may not get the desired information from these offices and often the customer may be misguided.
- It is tedious for a customer to plan a particular journey and have it executed properly.

1.4 PROPOSED SYSTEM:

- The proposed system is a web based application and maintains a centralized repository of all related information.
- The system allows one to easily access the relevant information and make necessary travel arrangements.
- Users can decide about places they want to visit and make bookings online for travel and accommodation.

Projecthelpline.com

CHAPTER 2 SYSTEM STUDY

2.1 PRELIMINARY INVESTIGATION

System development, a process consisting of two major steps of system analysis and design, start when management or sometimes system development personnel feel that a new system or an improvement in the existing system is required. The system development life cycle is classically thought of as the set of activities that analysts, designers and users carry out to develop and implement an information system.

- Preliminary investigation
- Determination of system requirements
- Design of system
- Development of software
- System testing
- Implementation, evaluation, and maintenance

A request to take assistance from information system can be made for many reasons, but in each case someone in the organization initiates the request is made, the first system activity the preliminary investigation begins. This activity has three parts:

- 1) Request clarification
- 2) Feasibility study
- 3) Request approval

2.2 SYSTEM DEVELOPMENT LIFE CYCLE

Systems are created to solve problems. One can think of the systems approach as an organized way of dealing with a problem. In this dynamic world, the subject System Analysis and Design (SAD) mainly deals with the software development activities.

Defining a System

A collection of components that work together to realize some objective forms a system. Basically there are three major components in every system, namely input, processing and output.

In a system the different components are connected with each other and they are interdependent. For example, human body represents a complete natural system. We are also bound by many national systems such as political system, economic system, educational system and so forth.

System Life Cycle

System life cycle is an organizational process of developing and maintaining systems. It helps in establishing a system project plan, because it gives overall list of processes and sub-processes required for developing a system.

System development life cycle means combination of various activities. In other words we can say that various activities put together are referred as system development life cycle.

Following are the different phases of software development cycle:

- System study
- Feasibility study
- System analysis
- System design
- Coding
- Testing
- Implementation
- Maintenance

The Different Phases Of Software Development Life Cycle Are Shown Below.

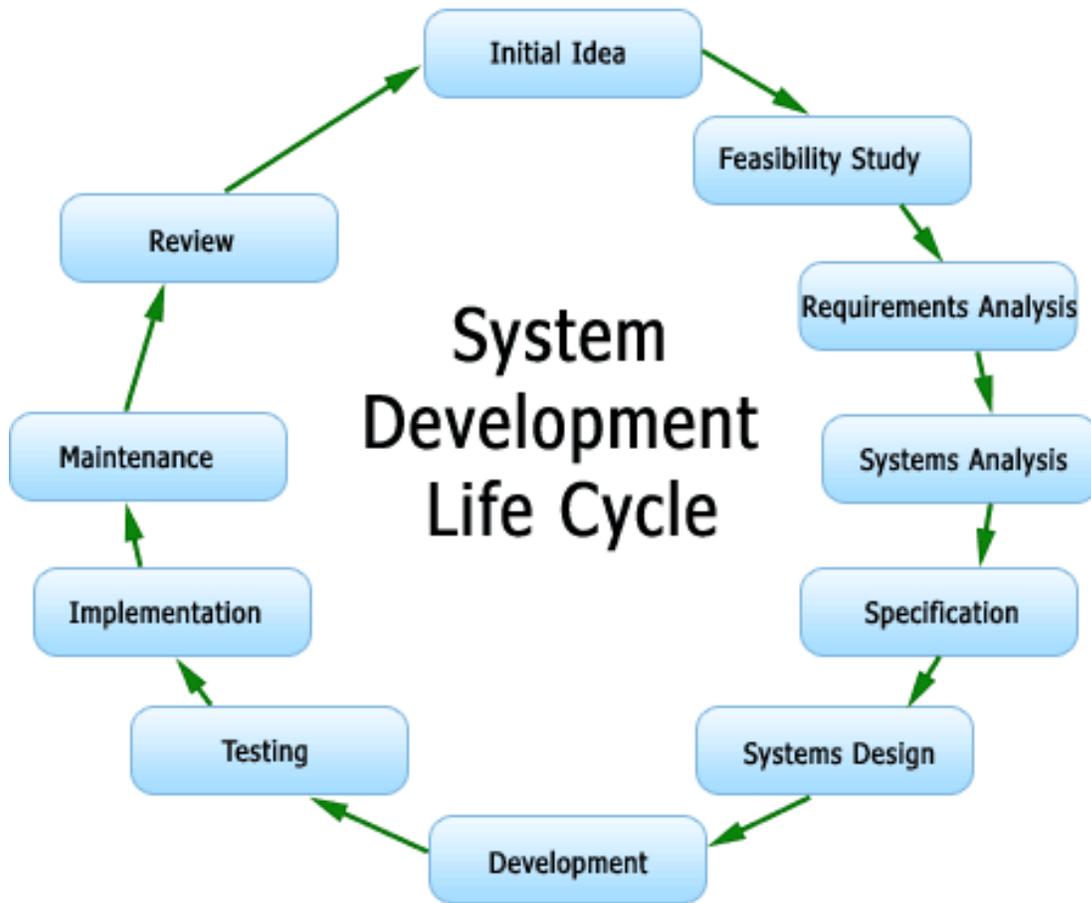


FIG: SHOWING GENERAL LIFE CYCLE PROCESS AND PERCENTAGE OF TIME DEVOTED

A system analysis is a separation of a substance into parts for study and their implementation and detailed examination.

Before designing any system it is important that the nature of the business and the way it currently operates are clearly understood. The detailed examination provides the specific data required during designing in order to ensure that all the client's requirements are fulfilled. The investigation or the study conducted during the analysis phase is largely based on the feasibility study. Rather it would not be wrong to say that the analysis and feasibility phases overlap. High-level analysis begins during the feasibility study. Though analysis is represented as one phase of the system development life cycle (SDLC), this is not true. Analysis begins with system initialization and continues until its maintenance.

2.3 ECONOMIC FEASIBILITY

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs. The system is economically feasible. It does not require any additional hardware or software.

2.4 TECHNICAL FEASIBILITY

- The technical issue usually raised during the feasibility stage of the investigation includes the following:
- Does the necessary technology exist to do what is suggested?
- Do the proposed equipments have the technical capacity to hold the data required to use the new system?
- Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- Can the system be upgraded if developed?

CHAPTER 3 SYSTEM ANALYSIS

3.1 IMPORTANCE OF COMPUTERIZED

Travel and Tourism Management System

There are several attributes in which the computer based information works. Broadly the working of computer system is divided into two main groups:

- ◆ Transaction System
- ◆ Decision Support System

Transaction System:

A transaction is a record of some well-defined single and usually small occurrence in a system. Transactions are input into the computer to update the database files. It checks the entering data for its accuracy. This means that numeric data appears in numeric field and character data in character field. Once all the checks are made, transaction is used to update the database. Transaction can be inputted in on-line mode or batch mode. In on-line mode, transactions are entered and updated into the database almost instantaneously. In batch mode, transactions are collected into batches, which may be held for a while and inputted later.

Decision Support System:

It assists the user to make analytical decision. It shows the various data in organized way called analysis. This analysis can be made to syrdy preferences and help in making decisions.

Type of Reports/Statements:

The report generated by the project includes listing schedules of trains, flights, total seats reserved for a particular package, list of cancellations for a particular date and time.

Updation and Modification system helps to add or delete record and modify the existing record set. Based on the nature of updation it provides appropriate access to the user. Performing these essential activities manually can be troublesome and time consuming at some instance. In the Proposed system, all these system activities can be performed quickly and efficiently.

This project is useful for

- ❑ The customers who want to reserve or cancel a seat.
- ❑ The customers who want to get information about any train or flight scheduling.
- ❑ The travelers who want to book a package.
- ❑ Packages for fun travel.
- ❑ Packages for Holly trip.
- ❑ Short or long packages

Last but not least, it is simple menu driven software, which anyone can operate, and also it doesn't require any special training for operation. Simply a few keystrokes and all these are performed instantly.

“Travel and Tourism Management System” has been designed to computerize or online the following functions that are performed by the system:

1. Customer Details

- a) Booked holly packages.
- b) Booked fun/huny moon packages.
- c) Packages for 2-3 days
- d) Long distance packages.

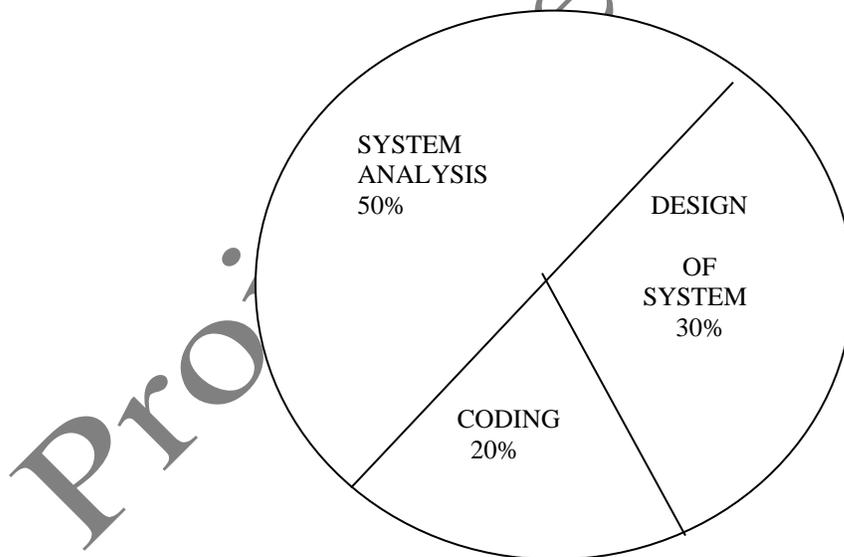
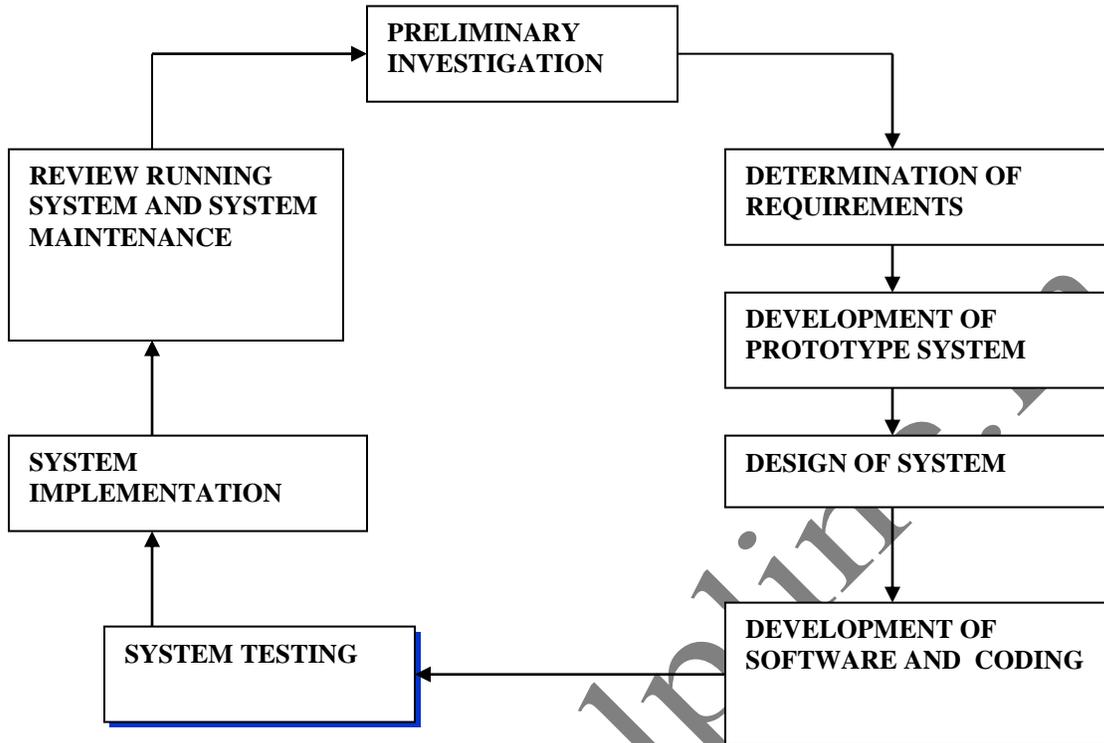
2. Railways

- a) Scheduling
- b) Reservation
- c) Cancellation

3. Airways

- a) Scheduling
- b) Reservation
- c) Cancellation

3.2 SYSTEM ANALYSIS



A system analysis is a separation of a substance into parts for study and their implementation and detailed examination.

Before designing any system it is important that the nature of the business and the way it currently operates are clearly understood. The detailed examination provides the specific data required during designing in order to ensure that all the client's

requirements are fulfilled. The investigation or the study conducted during the analysis phase is largely based on the feasibility study. Rather it would not be wrong to say that the analysis and feasibility phases overlap. High-level analysis begins during the feasibility study. Though analysis is represented as one phase of the system development life cycle (SDLC), this is not true. Analysis begins with system initialization and continues until its maintenance. Even after successful implementation of the system, analysis may play its role for periodic maintenance and up gradation of the system.

One of the main causes of project failures is inadequate understanding, and one of the main causes of inadequate understanding of the requirements is the poor planning of system analysis.

Analysis requires us to recall the objectives of the project and consider following three questions:

- What type of information is required?
- What are the constraints on the investigation?
- What are the potential problems that may make the task more difficult?

Keeping the above questions in mind and considering the survey conducted to determine the need of the system, the total system was designed and can be described as under:

The three major parts of the system are:

Providing Information:

The system is effectively used to provide large variety of information to the interested customer. The major purpose of the site is to easily provide transaction details, account details with quick update to latest modifications in the records

Constraints: After the objectives were clear during the analysis phase, it was essential to understand the constraints in order to plan and avoid problems arising during detailed analysis.

Technology - the customer may be committed to a particular hardware or software solution. The software required in this case is: complete Java developer kit, Microsoft windows environment for MS - access.

CHAPTER 4

SYSTEM DESIGN

The design document that we will develop during this phase is the blueprint of the software. It describes how the solution to the customer problem is to be built. Since solution to complex problems isn't usually found in the first try, iterations are most likely required. This is true for software design as well. For this reason, any design strategy, design method, or design language must be flexible and must easily accommodate changes due to iterations in the design. Any technique or design needs to support and guide the partitioning process in such a way that the resulting sub-problems are as independent as possible from each other and can be combined easily for the solution to the overall problem. Sub-problem independence and easy combination of their solutions reduces the complexity of the problem. This is the objective of the partitioning process. Partitioning or decomposition during design involves three types of decisions: -

Define the boundaries along which to break;

Determine into how many pieces to break; and

4.1 PHYSICAL DESIGN

The systems objectives outlined during the feasibility study serve as the basis from which the work of system design is initiated. Much of the activities involved at this stage is of technical nature requiring a certain degree of experience in designing systems, sound knowledge of computer related technology and thorough understanding of computers available in the market and the various facilities provided by the vendors. Nevertheless, a system cannot be designed in isolation without the active involvement of the user. The user has a vital role to play at this stage too. As we know that data collected during feasibility study will be utilized systematically during the system design.

System Design Considerations

The system design process is not a step-by-step adherence of clear procedures and guidelines. Though, certain clear procedures and guidelines have emerged in recent days, but still much of design work depends on knowledge and experience of the designer.

When designer starts working on system design, he will face different type of problems. Many of these will be due to constraints imposed by the user or limitations of the hardware and software available in the market. Sometimes, it is difficult to enumerate the complexity of the problems and solutions thereof since the variety of likely problems is so great and no solutions are exactly similar. However, following considerations should be kept in mind during the system-designing phase:

Design Methodology:

Design Methodology is a way to transform the "art" of system analysis and design into an "engineering - type" discipline. It explains the relationship amongst various modules and programs within the system. It standardizes the approach to analysis and design, simplifies design by segmentation, improves documentation and subsequent maintenance and enhancements. The following structured diagram can appropriately represent the relationship between various modules.

b. Design Overview:

In analyzing the present system a great deal of information was collected during the investigation and feasibility phases through list of problems and requirements, interview reports, questionnaires, onsite observations, manuals and determining potential solutions.

It is important to record this information in an unambiguous, concise manner which will be clear and accessible to others, and which can be used by other analysts and designers involved in developing the system. Structured techniques help us to record the information in this way, using diagrams and minimum amount of the text.

c. Process Modeling:

System design goes through two phases of development: logical and physical. Logical implementation represented by Data Flow Diagram shows the logical flow of a system and defines the boundaries of the system it describes the input (source), outputs

(destinations), data bases (data stores), and procedures (data flows) - all in the format that meets the user's requirements.

4.2 DATA FLOW DIAGRAM

Data flow diagrams are the most commonly used way of documenting the processing of the candidate system. As their name suggest they are a pictorial way of representing the flow of data into, around, and out of the system.

- **External entities** - represents the sources of the data that enter the system or the recipients of the system that leave the system.

for example - passenger is the usual receiver of information and supplier of data during form filling.

- **Data stores** - represent the stores of the data within the system example: computer files, databases or in the manual system files, etc. data stores can not be linked directly by data flows either to each other or to external entities without an intervening process to transform them.

- **Processes** - represent activities in which data is manipulated by being stored or retrieved or transformed in some way.

Process names are generally unambiguous and convey as much meaning as possible without being too long. Example: verify data, acquired time schedule etc.

4.3 DESCRIPTION OF THE MODULES

Number of Modules

After careful analysis the system has been identified to have the following modules:

1. **Administrator module**
2. **Travels module**
3. **Routes module**
4. **Reservations module**
5. **Testimonials module.**

1. ADMINISTRATOR MODULE:

This module provides administrator related functionality. Administrator manages all information and has access rights to add, delete, edit and view the data related to places, travels, routes, bookings, restaurants etc.

2. TRAVELS MODULE:

This module provides the details of various travel agencies. A user can select the appropriate agency depending on convenience and accessibility.

3. ROUTES MODULE:

This module provides information related to various routes connecting sources and destinations. For each route, information such as source, destination, fare, reservation details, pick up points etc are provides. Only administrator can add , delete, edit and manage the data. Users can only view the information.

4. RESERVATIONS MODULE:

This module provides functionalities that allow a user to book tickets or cancel previously booked tickets. The module maintains the details of all reservations made so far and allows administrator to either confirm or reject the bookings.

5. TESTIMONIALS MODULE:

Users of this application can post their opinions, complaints and suggestions regarding this portal and services to the administrator. Accordingly, the administrator can take various steps to act on the complaints and suggestions.

Pro

4.4 ER DIAGRAM

Projecthelpline.in

INPUT DESIGN:

Input design is a part of overall system design. The main objective during the input design is as given below:

- To produce a cost-effective method of input.
- To achieve the highest possible level of accuracy.
- To ensure that the input is acceptable and understood by the user.

INPUT STAGES:

The main input stages can be listed as below:

- Data recording
- Data transcription
- Data conversion
- Data verification
- Data control
- Data transmission
- Data validation
- Data correction

INPUT TYPES:

It is necessary to determine the various types of inputs. Inputs can be categorized as follows:

- External inputs, which are prime inputs for the system.
- Internal inputs, which are user communications with the system.

- Operational, which are computer department's communications to the system?
- Interactive, which are inputs entered during a dialogue.

INPUT MEDIA:

At this stage choice has to be made about the input media. To conclude about the input media consideration has to be given to;

- Type of input
- Flexibility of format
- Speed
- Accuracy
- Verification methods
- Rejection rates
- Ease of correction
- Storage and handling requirements
- Security
- Easy to use
- Portability

Keeping in view the above description of the input types and input media, it can be said that most of the inputs are of the form of internal and interactive. As

Input data is to be the directly keyed in by the user, the keyboard can be considered to be the most suitable input device.

OUTPUT DESIGN:

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provide a permanent copy of the results for later consultation. The various types of outputs in general are:

- External Outputs whose destination is outside the organization.
- Internal Outputs whose destination is within organization and they are the User's main interface with the computer.
- Operational outputs whose use is purely within the computer department.
- Interface outputs, which involve the user in communicating directly with the system.

OUTPUT DEFINITION

The outputs should be defined in terms of the following points:

- Type of the output
- Content of the output
- Format of the output
- Location of the output
- Frequency of the output
- Volume of the output
- Sequence of the output

It is not always desirable to print or display data as it is held on a computer. It should be decided as which form of the output is the most suitable.

For Example

- Will decimal points need to be inserted
- Should leading zeros be suppressed.

OUTPUT MEDIA:

In the next stage it is to be decided that which medium is the most appropriate for the output. The main considerations when deciding about the output media are:

- The suitability for the device to the particular application.
- The need for a hard copy.
- The response time required.
- The location of the users
- The software and hardware available.

Keeping in view the above description the project is to have outputs mainly coming under the category of internal outputs. The main outputs desired according to the requirement specification are:

The outputs were needed to be generated as a hard copy and as well as queries to be viewed on the screen. Keeping in view these outputs, the format for the output is taken from the outputs, which are currently being obtained after manual processing. The standard printer is to be used as output media for hard copies.

SDLC Methodology:

This document play a vital role in the development of life cycle (SDLC) as it describes the complete requirement of the system. It means for use by developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

SPIRAL MODEL was defined by Barry Boehm in his 1988 article, "A spiral Model of Software Development and Enhancement. This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration models.

As originally envisioned, the iterations were typically 6 months to 2 years long. Each phase starts with a design goal and ends with a client reviewing the progress thus far. Analysis and engineering efforts are applied at each phase of the project, with an eye toward the end goal of the project.

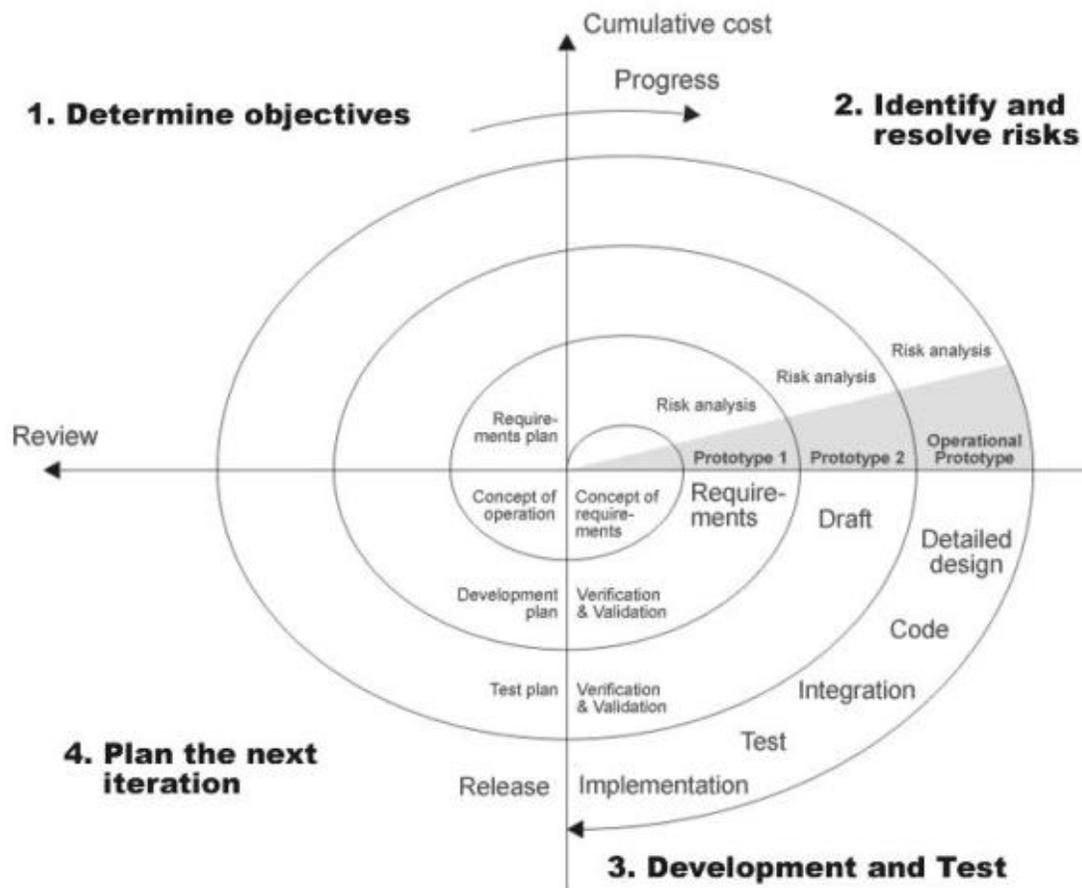
The steps for Spiral Model can be generalized as follows:

- The new system requirements are defined in as much details as possible. This usually involves interviewing a number of users representing all the external or internal users and other aspects of the existing system.
- A preliminary design is created for the new system.
- A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.
- A second prototype is evolved by a fourfold procedure:
 - Evaluating the first prototype in terms of its strengths, weakness, and risks.
 - Defining the requirements of the second prototype.
 - Planning and designing the second prototype.
 - Constructing and testing the second prototype.
- At the customer option, the entire project can be aborted if the risk is deemed too great. Risk factors might involved development cost overruns, operating-cost miscalculation, or any other factor that could, in the customer's judgment, result in a less-than-satisfactory final product.
- The existing prototype is evaluated in the same manner as was the previous prototype, and if necessary, another prototype is developed from it according to the fourfold procedure outlined above.

- The preceding steps are iterated until the customer is satisfied that the refined prototype represents the final product desired.
- The final system is constructed, based on the refined prototype.
- The final system is thoroughly evaluated and tested. Routine maintenance is carried on a continuing basis to prevent large scale failures and to minimize down time.



The following diagram shows how a spiral model acts like:



Advantages:

- Estimates(i.e. budget, schedule etc .) become more realistic as work progresses, because important issues discovered earlier.

- It is more able to cope with the changes that are software development generally entails.
- Software engineers can get their hands in and start working on the core of a project earlier

Performance Requirements:

- Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely with the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

- The requirement specification for any system can be broadly stated as given below:



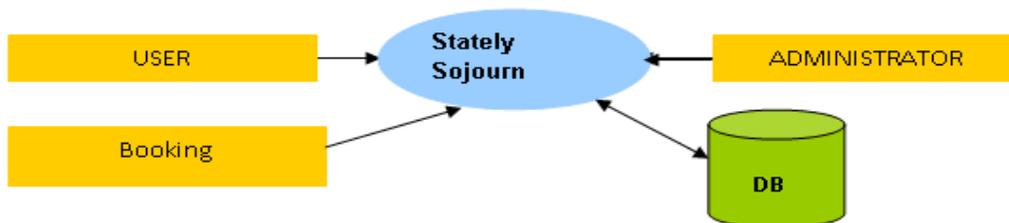
- The system should be able to interface with the existing system
- The system should be accurate
- The system should be better than the existing system



- The existing system is completely dependent on the user to perform all the duties.

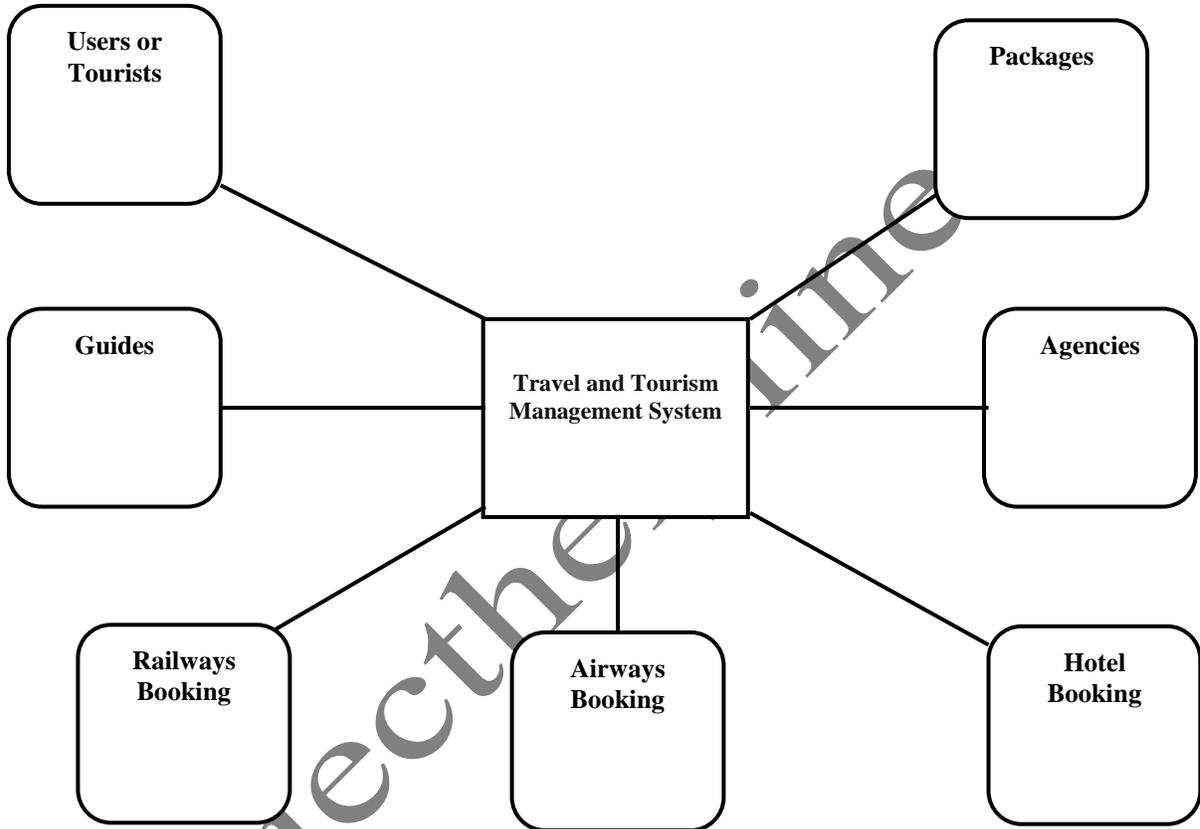


DFD (Context level Diagram):

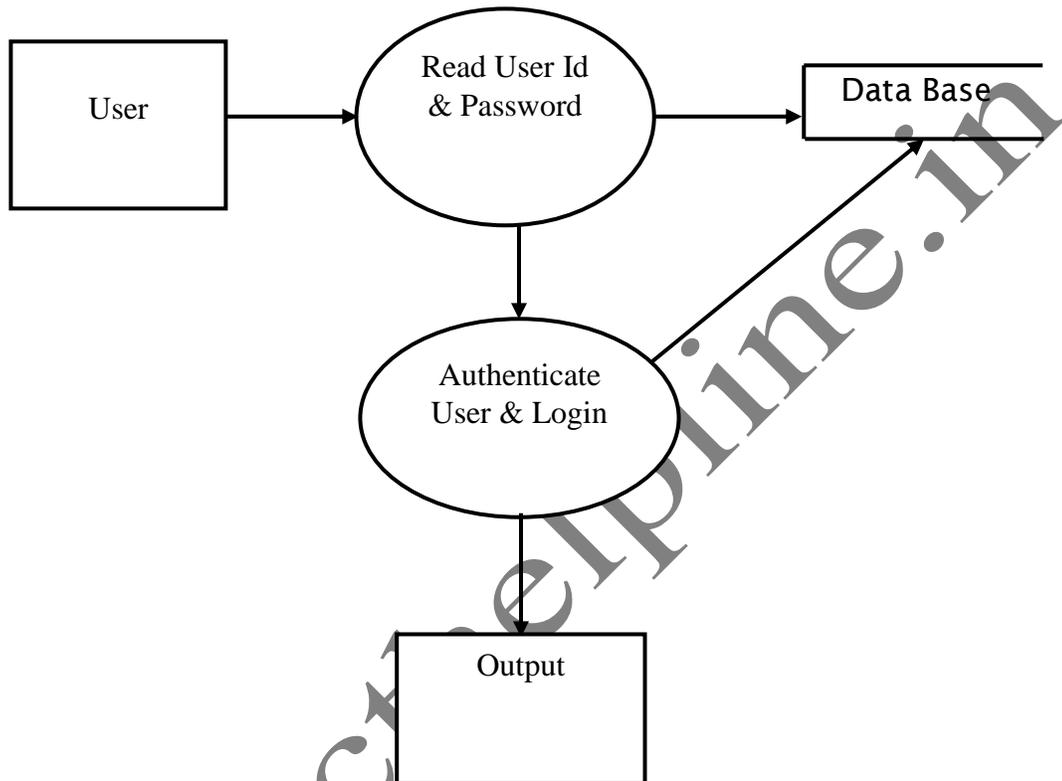


4.5 CONTEXT LEVEL DFD

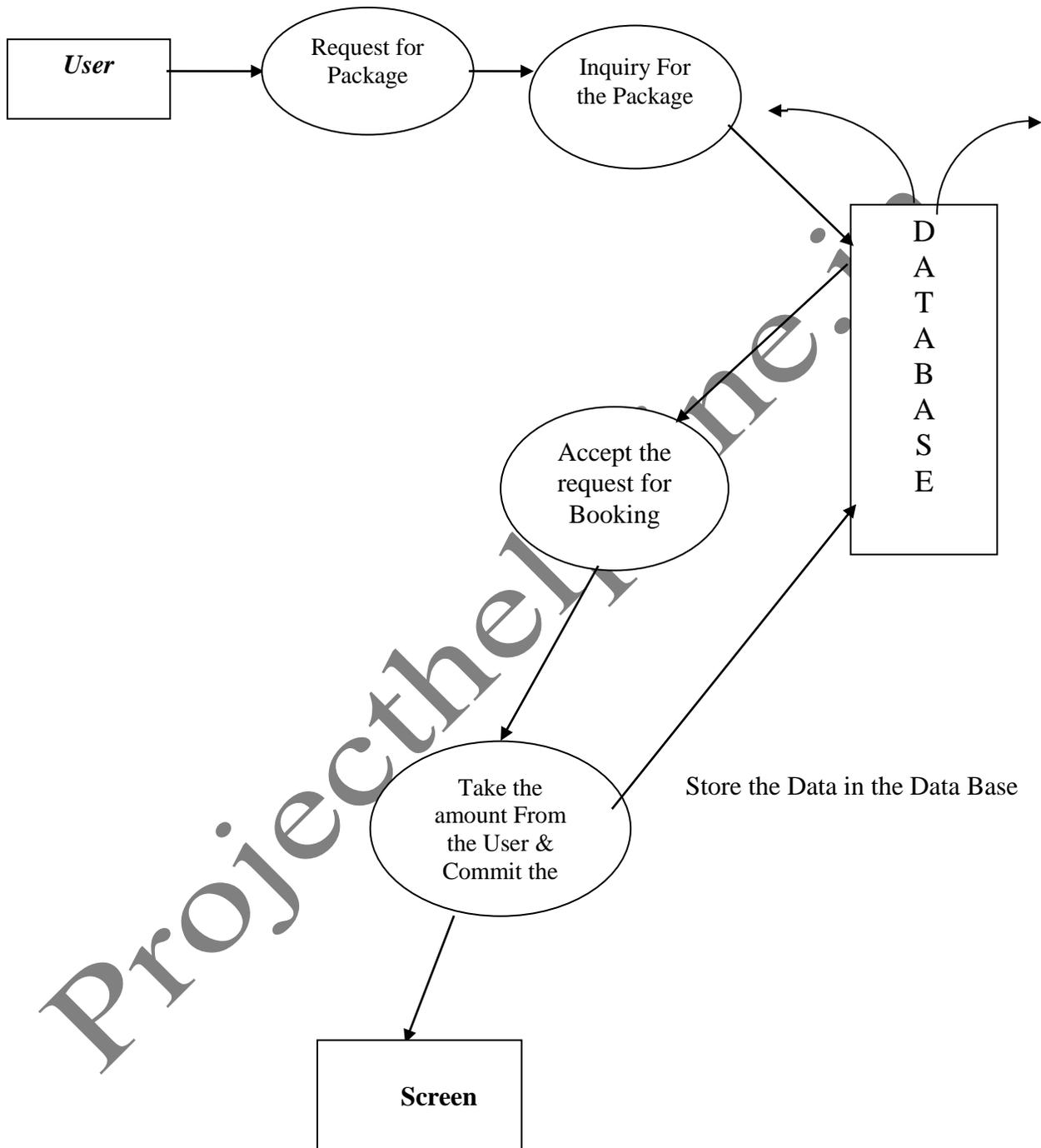
Travel and Tourism Management System



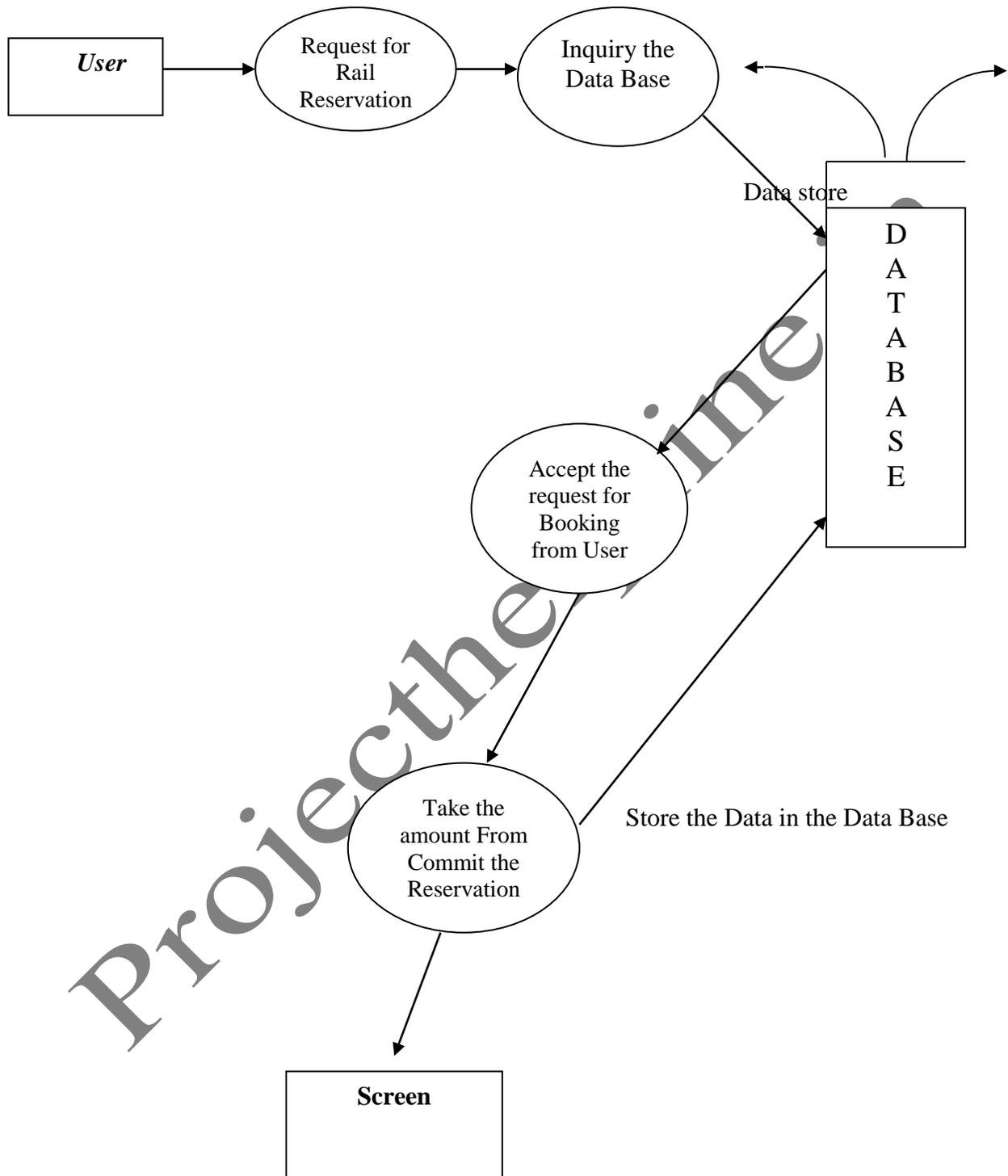
DFD for Login Of User



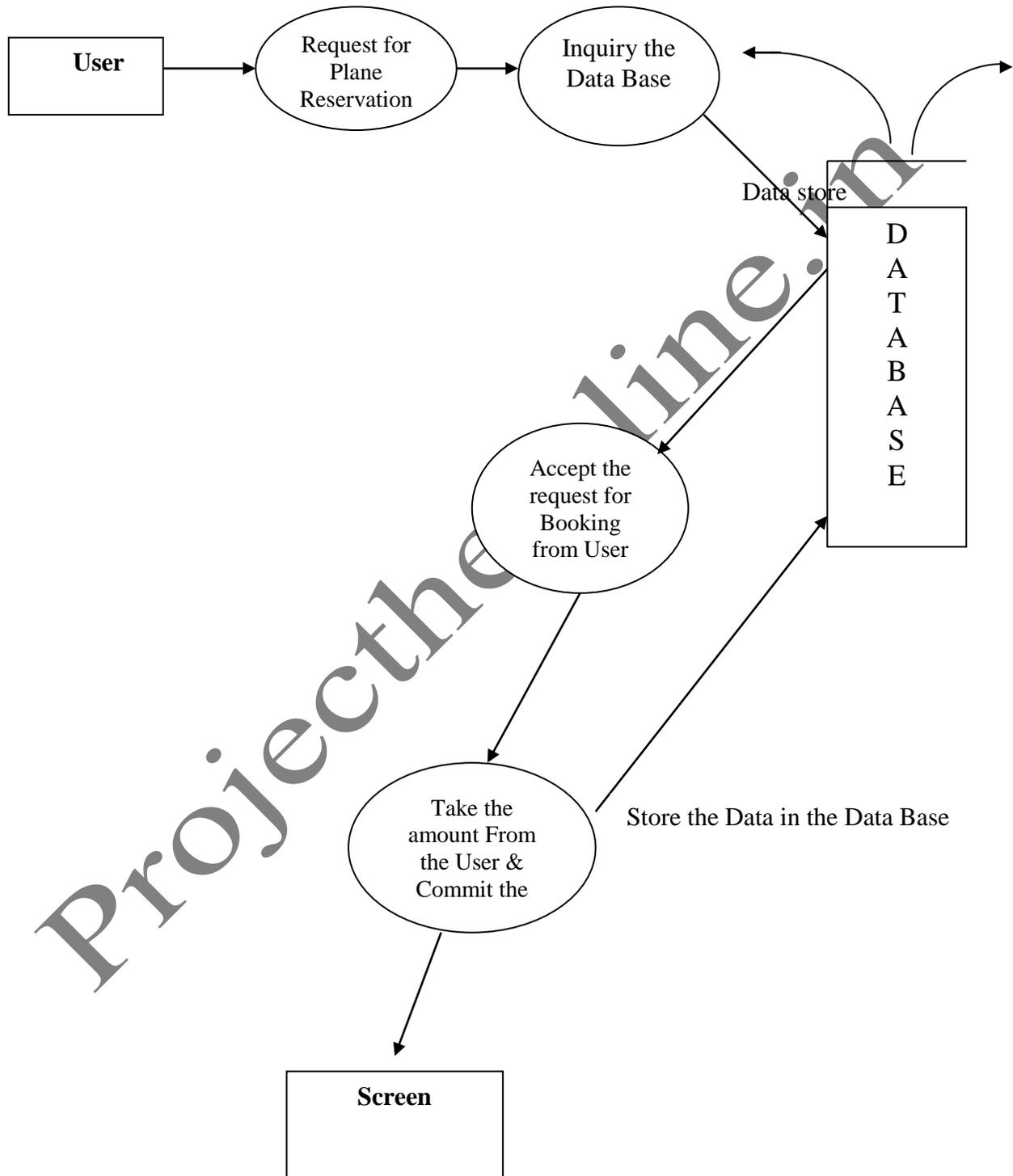
DFD for Package



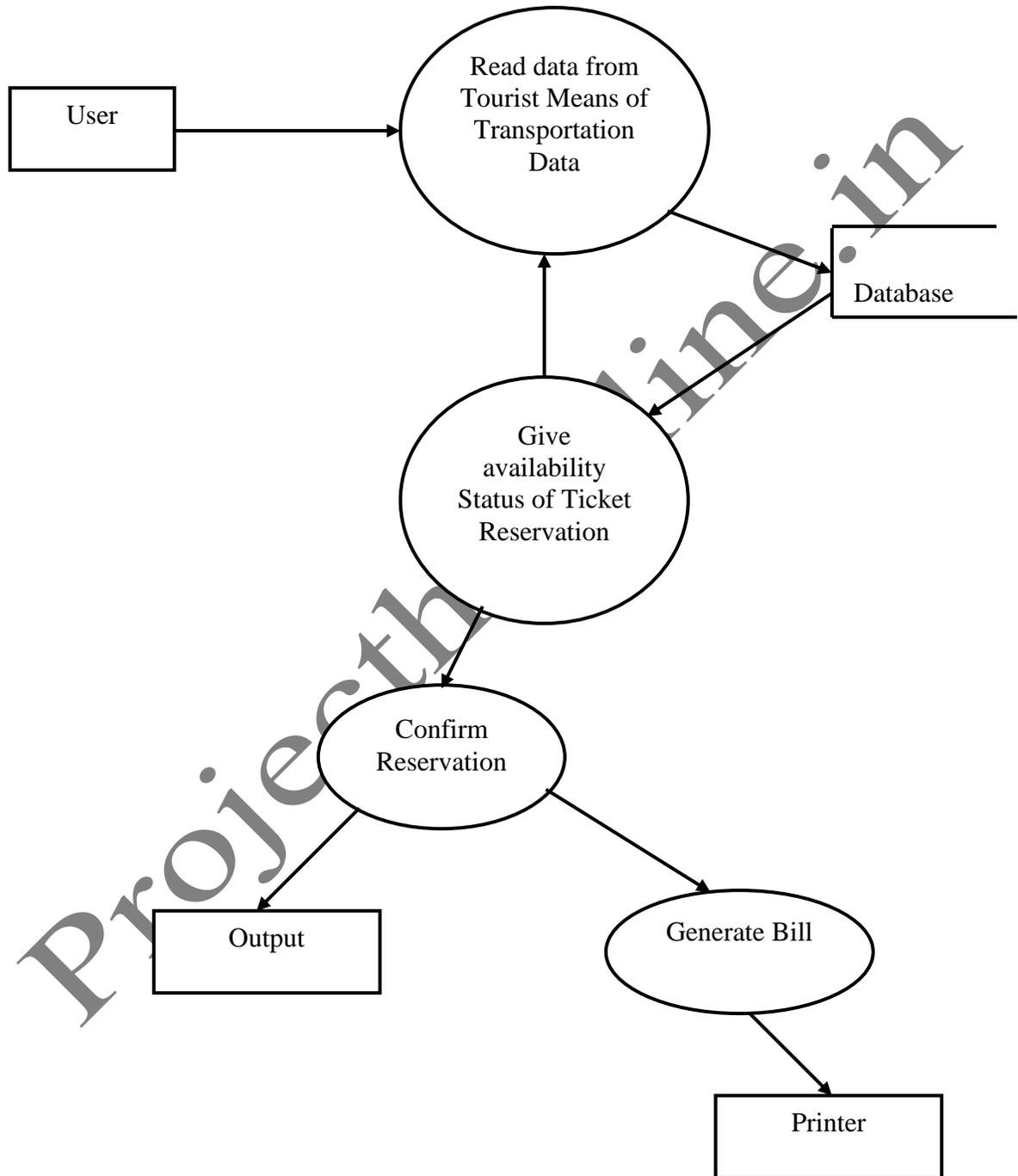
DFD for Railways



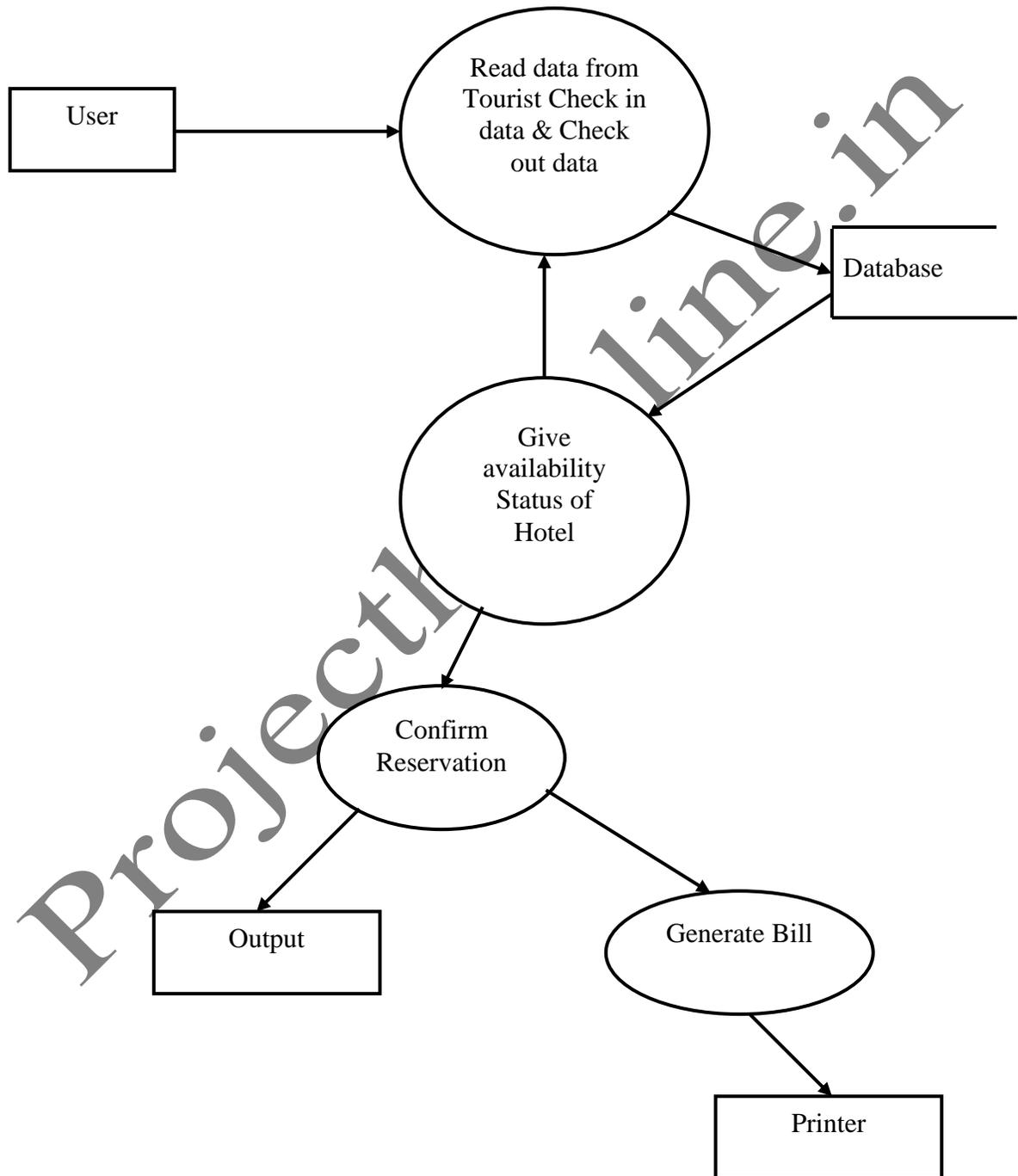
DFD for Airways



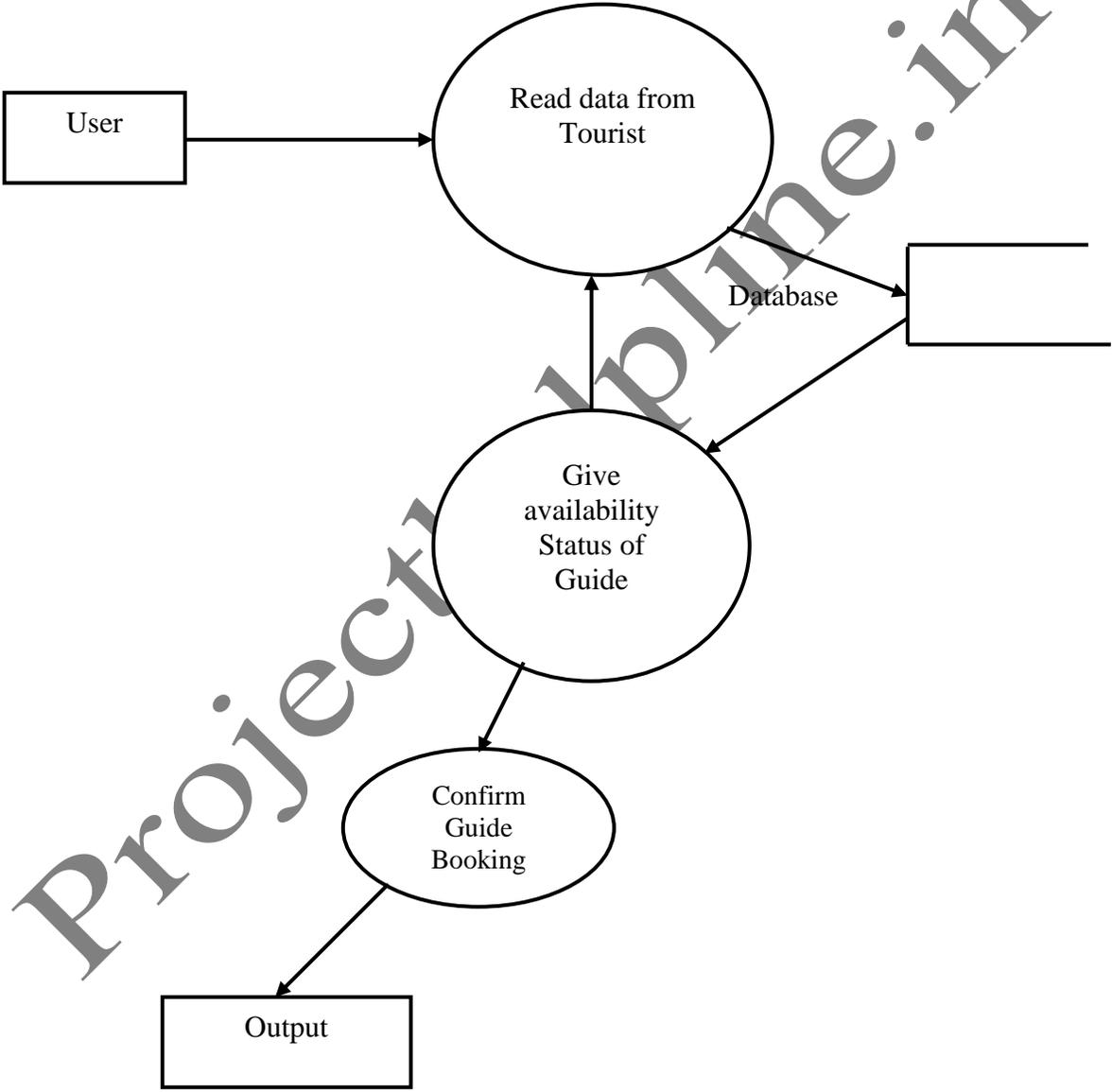
DFD for Ticket Reservation



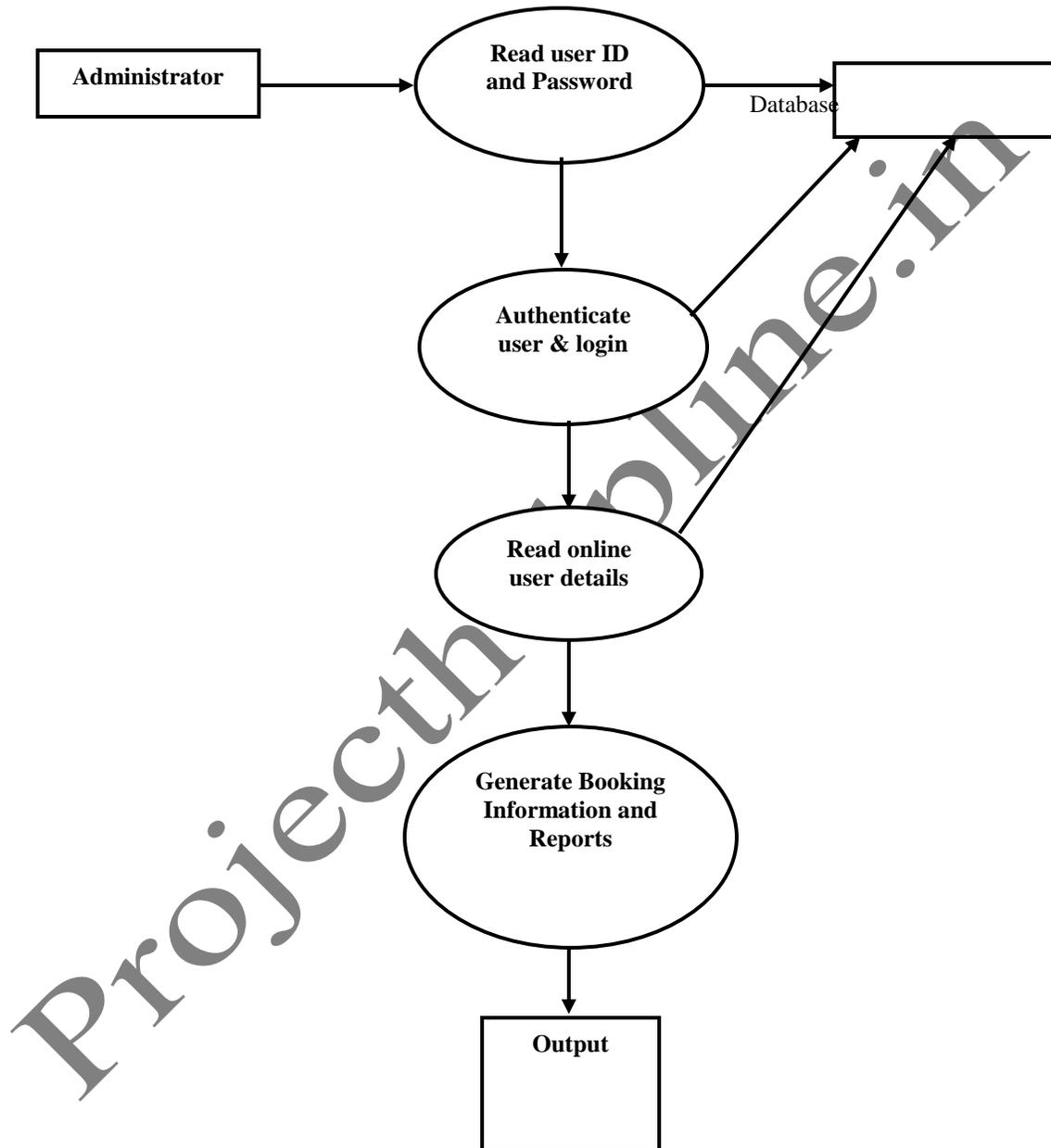
DFD for Hotel Reservation



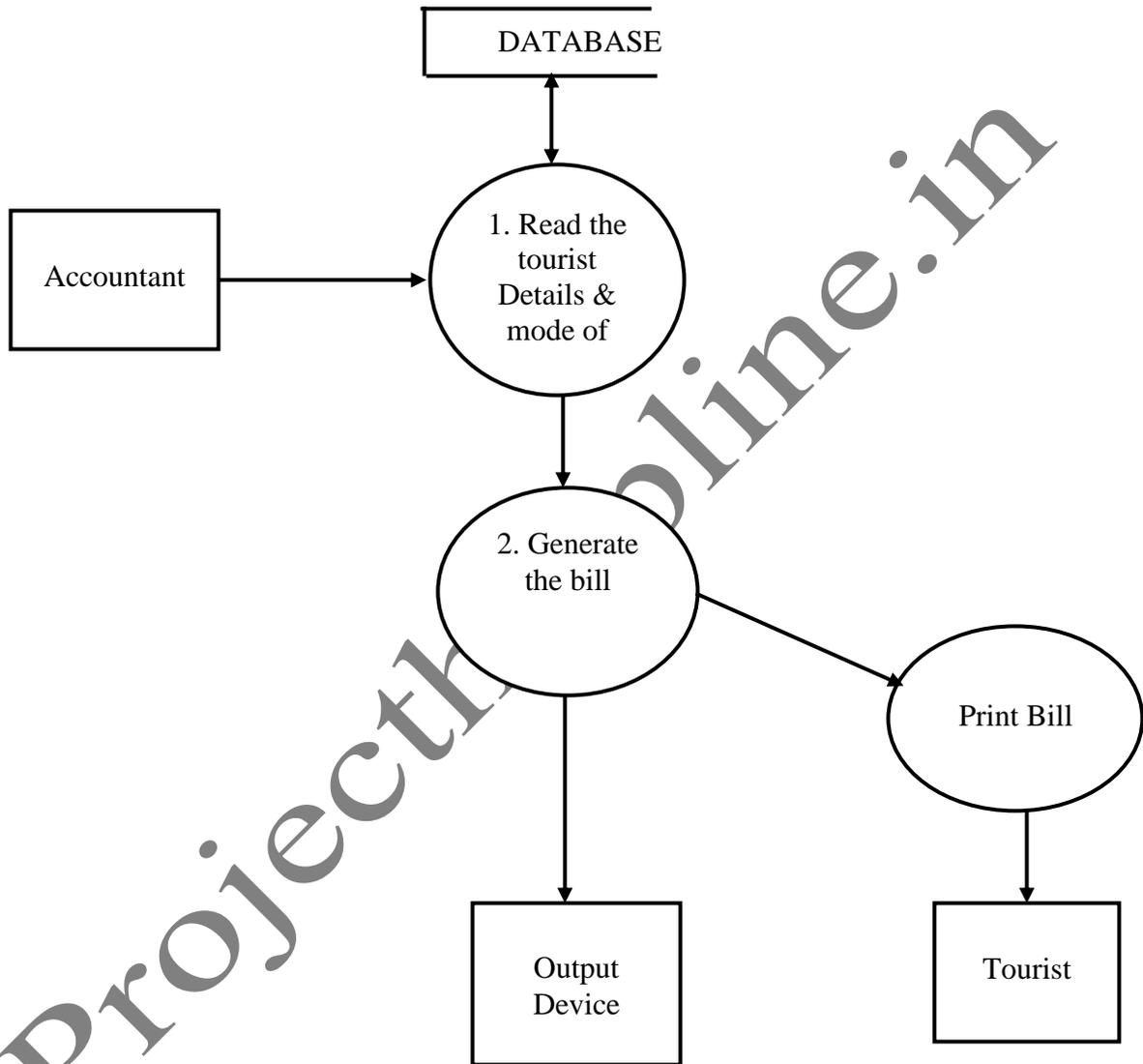
DFD for Guide Booking



DFD for Administrator



DFD for Bill Payment



4.6 DATA MODELING

Password Table

Field	Type	Constraint
Username	Char (30)	Not null
Password	Char (30)	Not null

Administrator password Table

Field	Type	Constraint
Username	Char (30)	Not null
Password	Char (30)	Not null

Agency Table

Field	Type	Constraint
Name	Char(20)	Not null
Address	Char(30)	Not null
Phone No	Number(15)	Not null
Mobile	Number(15)	Not null
Fax	Number(30)	Not null
Email	Char(30)	Not null
State	Char(30)	Not null
City	Char(30)	Not null
Pin	Number(12)	Not null
Contact Person	Char(30)	Not null
Contact Person No	Number(12)	Not null

Hotel Booking Table

Field	Type	Constraint
Name	Char(15)	Not null
Address	Char(30)	Not null
City	Char(15)	Not null
State	Char(15)	Not null

Pin	Number(15)	Not null
Email	Char(30)	Not null
Phone	Number(15)	Not null
Fax	Number	Not null
Resort_Name	Char(30)	Not null
Resort_Address	Char(30)	Not null
Resort_City	Char(15)	Not null
Resort_State	Char(15)	Not null
Resort_person	Char(30)	Not null
Resort_Person	Number(15)	Not null

Guide Table

Field	Type	Constraint
Name	Char(15)	Not null
Address	Char(20)	Not null
Phone	Number(15)	Not null
Email	Char(30)	Not null
City	Char(20)	Not null

Booking Table

Field	Type	Constraint
Book Id	Char(15)	Primary Key
Book Date	Char(30)	Not null
No of seat	Char(15)	Not null
Payment	Char(15)	Not null
City	Number(15)	Not null
Phone	Char(30)	Not null
Fax	Number(15)	Not null
Mobile	Number	Not null
Remarks	Char(30)	Not null
Journey	Char(30)	Not null

Tourist Table

Field	Type	Constraint
Id	Char(10)	Primary Key
Name	Char(15)	Not null
Address	Char(30)	Not null
City	Char(15)	Not null
State	Char(15)	Not null
Pin	Number(15)	Not null
Email	Char(30)	Not null
Phone	Number(15)	Not null
Fax	Number	Not null

Booking_info Table

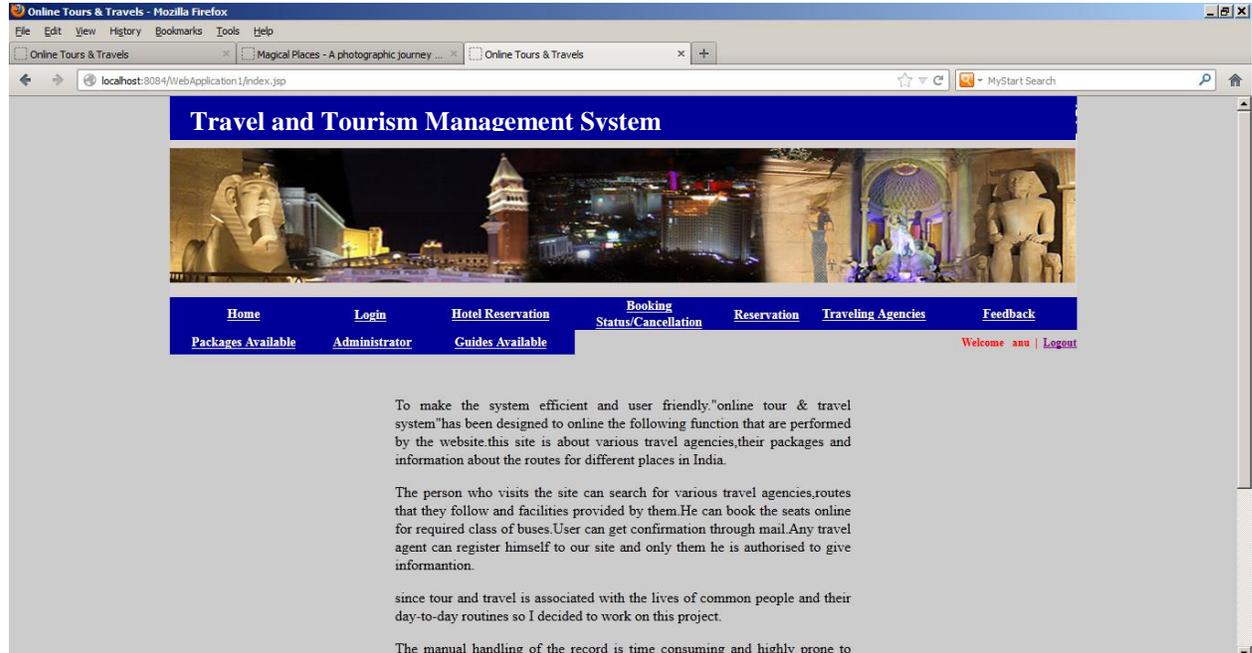
Field	Type	Constraint
Tourist Id	Char(15)	Primary Key
Information	Char(20)	Not null

Package Table

Field	Type	Constraint
Tourist Id	Char(15)	Primary Key
Package_name	Char(20)	Not null
Date	Date	Not null

CHAPTER 5 SYSTEM DEVELOPMENT

5.1 SOURCE CODE INDEX.JSP



```
<% @ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Online Tours & Travels</title>
<style type="text/css">
<!--
body {
    margin-left: 0px;
    margin-top: 0px;
    margin-right: 0px;
    margin-bottom: 0px;
    background-color: #CCCCCC;
    background-image: url();
}
.style1 {
    color: #0000FF;
    font-weight: bold;
}
.style3 {color: #0000FF; font-weight: bold; font-size: 36px; }
.style6 {color: #FF0000}
.style7 {font-weight: bold; font-size: 14px;}
.style8 {
    font-size: 14px;
    color: #990000;
```

```

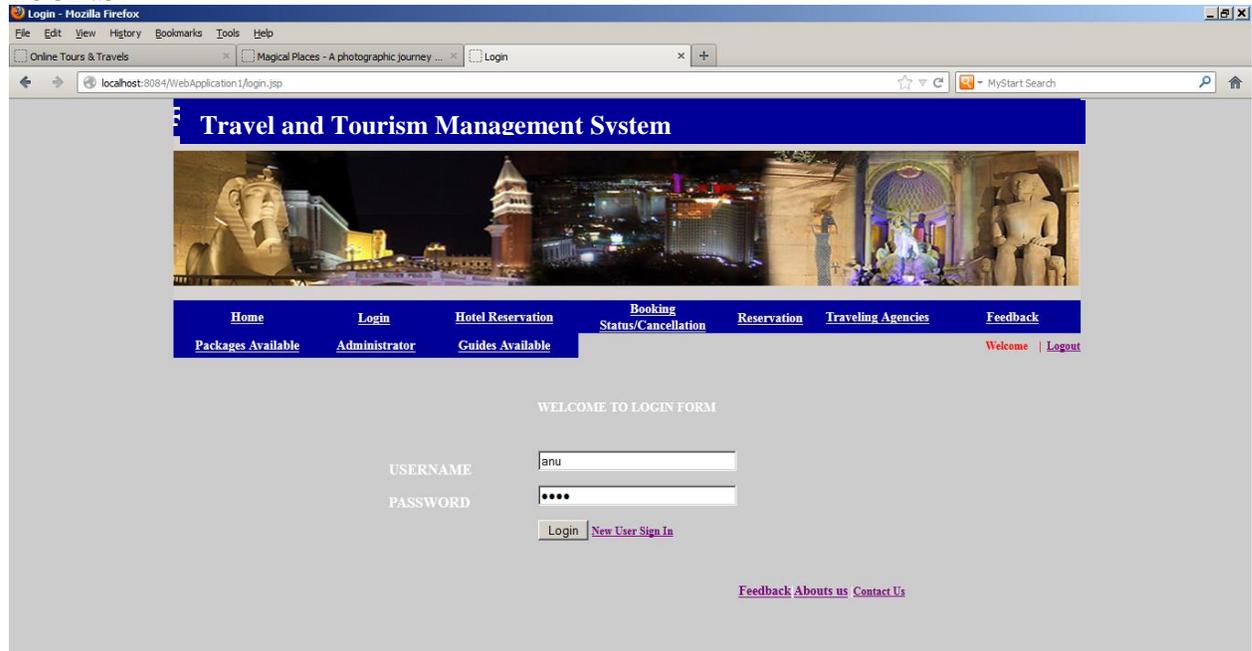
}
.style10 {font-weight: bold; font-size: 14px; color: #990000; }
.style11 {color: #FF0000; font-weight: bold; }
.style12 {color: #990000}
-->
</style></head>

<body>
<form id="form1" name="form1" method="post" action="">
<%@include file="header.jsp"%>
<table width="989">
  <tr>
    <td width="979">&nbsp;</td>
  </tr>
</table>
<table width="986" height="412">
  <tr>
    <td width="502" height="278" valign="top" bgcolor="#CCCCCC"><p>To make the system efficient and user friendly.&quot;online tour &amp; travel system&quot;;has been designed to online the following function that are performed by the website.this site is about various travel agencies,their packages and information about the routes for different places in India.</p>
    <p>The person who visits the site can search for various travel agencies,routes that they follow and facilities provided by them.He can book the seats online for required class of buses.User can get confirmation through mail.Any travel agent can register himself to our site and only them he is authorised to give informantion.</p>
    <p>since tour and travel is associated with the lives of common people and their day-to-day routines so I decided to work on this project.</p>
    <p>The manual handling of the record is time consuming and highly prone to error.This project includes listing schedules of trains,flieghts,total seats reserved for a particular package,registration of tourists,hotels,guides,arrival &amp; departure transfers,Hotel booking ,Transport by road, train &amp; domestic airlines,Sightseeing with guide.</p>
    <p>Updation and Modification system helps to add or delete record and modify the existing record set.Based on the nature of updation it provides appropriate access to the user.Performing these essential activities manually can be troublesome and time consuming at instance .In the proposed system,all these system activities can be performed quickly and efficiently.</p></td>
    <td width="468" valign="top"></td>
  </tr>
</table>

<%@include file="foot.jsp"%>
</form>
</body>
</html>

```

LOGIN.JP



```
<% @ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
```

```
<title>Login</title>
```

```
<style type="text/css">
```

```
<!--
```

```
.style1 {
```

```
    color: #0000FF;
```

```
    font-weight: bold;
```

```
}
```

```
.style4 {color: #339966; font-weight: bold; }
```

```
body {
```

```
    background-color: #CCCCCC;
```

```
    margin-left: 0px;
```

```
    margin-top: 0px;
```

```
}
```

```
.style5 {color: #FF0000; font-weight: bold; }
```

```
.style6 {
```

```
    color: #CC0099;
```

```
    font-weight: bold;
```

```
    font-size: 36px;
```

```
}
```

```
.style7 {
```

```
    font-size: 24px;
```

```
    color: #0000FF;
```

```
}
```

```
.style8 {color: #FFFFFF; font-weight: bold; }
```

```
.style10 {font-size: 12px }
```

```
.style11 {
```

```
    color: #0000FF;
```

```

        font-size: 36px;
    }
-->
</style>
</head>

<script language="javascript" type="text/javascript">
    window.history.forward(1);
</script>

<script>

function validate(form)
{
if (form.name.value=="")
    {
        alert("Name should not be blank. Please enter it.");
        form.name.focus(); return false;
    }

if (form.password.value=="")
    {
        alert("password should not be blank. Please enter it.");
        form.password.focus(); return false;
    }
}
</script>
<body>
    <form id="form1" name="form1" method="post" action="login" onsubmit="return
validate(this)">
    <%@include file="header.jsp"%>
    <%
if(session.getAttribute("ins1")!=null)
{

    response.sendRedirect("logout1.jsp");
}%>
<table width="200" border="0" align="center">
<tr>
<td colspan="3" align="center"></td>
</tr>
<tr>
<td colspan="3">&nbsp;</td>
</tr>
<tr>
<td colspan="3"><p align="center" class="style6 style7 style11">WELCOME TO LOGIN FORM </p>
<table width="523" height="163" border="0" cellpadding="0" cellspacing="0">
<tr>
<td width="164" height="163"><p class="style8">USERNAME</p>
<p class="style8">PASSWORD</p>
<p class="style4">&nbsp;</p></td>
<td width="359">
<label>
<input name="name" type="text" size="33" />
</label>
<p>

```

```

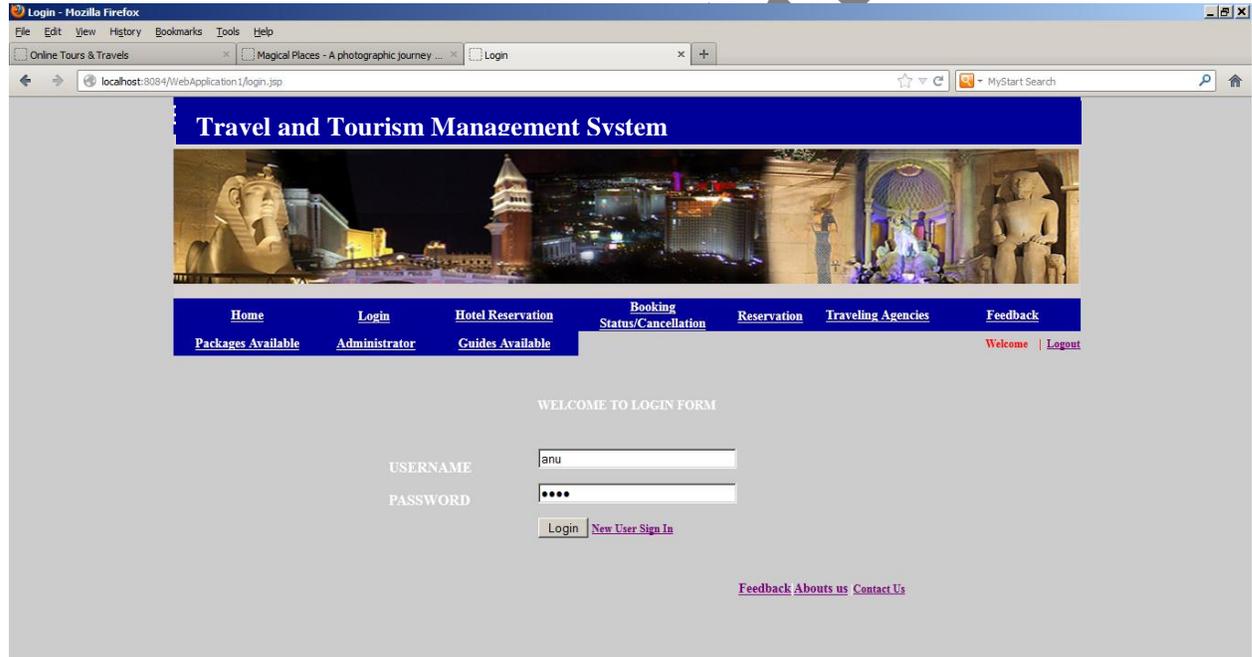
<label>
<input name="password" type="password" size="33" />
</label>
</p>
<p>
<label>
<input type="submit" name="Submit" value="Login" />
</label>
<span class="style10"><a href="nuser.jsp">New User Sign In </a></span></p>
</tr>
</table>
</td>
</tr>
</table>

<% @include file="foot.jsp"%>
</form>
</body>
</html>

```

ine.in

HOTEL.JSP



```

<% @ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title></title>
<style type="text/css">
<!--
body {
margin-left: 0px;
margin-top: 0px;

```

```

        margin-right: 0px;
        margin-bottom: 0px;
        background-color: #CCCCCC;
    }
    .style1 {
        color: #0000FF;
        font-weight: bold;
    }
    .style2 {
        color: #0000FF;
        font-weight: bold;
    }
    .style3 {color: #0000FF}
    .style7 {font-size: 24px}
    .style9 {color: #990000; font-weight: bold; font-size: 18px; }
    .style10 {font-size: 18px; color: #990000;}
    .style12 {
        font-size: 24px;
        font-weight: bold;
        color: #0000FF;
    }
-->
</style></head>

<script>

function validate(form)
{
if (form.name.value=="")
    {
        alert("Name should not be blank. Please enter it.");
        form.name.focus(); return false;
    }

if (form.address.value=="")
    {
        alert("address should not be blank. Please enter it.");
        form.address.focus(); return false;
    }

if (form.city.value=="")
    {
        alert("city should not be blank. Please enter it.");
        form.city.focus(); return false;
    }

if (form.pin.value=="")
    {
        alert("pin should not be blank. Please enter it.");
        form.pin.focus(); return false;
    }
    if(isNaN(form.pin.value))
    {
        alert("pin Should be a number ");
    }
form.pin.focus();
return false;
}

if (form.email.value=="")
    {
        alert("email should not be blank. Please enter it.");
        form.email.focus(); return false;
    }
}

```

```

var x=form.email.value
var atpos=x.indexOf("@");
var dotpos=x.lastIndexOf(".");
if (atpos<1 || dotpos<atpos+2 || dotpos+2>=x.length)
{
    alert("Not a valid e-mail address");
    form.email.focus();
    return false;
}

if (form.phone.value=="")
{
    alert("phone should not be blank. Please enter it.");
    form.phone.focus(); return false;
}
if(isNaN(form.phone.value))
{
    alert("phone Should be a number ");
    form.phone.focus();
    return false;
}
if((form.phone.value).length < 10)
{
    alert(" Your mobile no. must be atleast 10 digits ");
    return false;
}

if (form.t1.value=="")
{
    alert("Hotel Name should not be blank. Please enter it.");
    form.t1.focus(); return false;
}
if (form.t2.value=="")
{
    alert("Hotel Address should not be blank. Please enter it.");
    form.t2.focus(); return false;
}

if (form.t3.value=="")
{
    alert("Type of room should not be blank. Please enter it.");
    form.t3.focus(); return false;
}

if (form.t7.value=="")
{
    alert("Date should not be blank. Please enter it.");
    form.t7.focus(); return false;
}
if (form.t7.value=="")
{
    alert("Booking Date should not be blank. Please enter it.");
    form.t7.focus(); return false;
}if (form.t8.value=="")
{
    alert("Checkin Date should not be blank. Please enter it.");
    form.t8.focus(); return false;
}
}
if (form.t9.value=="")
{
    alert("Checkout Date should not be blank. Please enter it.");
    form.t9.focus(); return false;
}

```

```

    }

    function isValidDate(sText) {
var reDate = /^(?:0[1-9]|[12][0-9]|3[01])\.(?:0[1-9]|1[0-2])\.(?:19|20\d{2})$/;
    return reDate.test(sText);
    }
    if (isValidDate(form.t7.value))
    {

    }
    else
    {
        alert("Invalid Booking Date !");
        form.t7.value=""
        form.t7.focus(); return false;
    }

    if (isValidDate(form.t8.value))
    {

    }
    else
    {
        alert("Checkin Invalid Date !");
        form.t8.value=""
        form.t8.focus(); return false;
    }

    }
</script>

<body>
    <form id="form1" name="form1" method="post" action="hotel" onsubmit="return validate
(this)">
    <% @include file="header.jsp"%>
<table width="1002" height="117" border="0" cellpadding="0" cellspacing="0">
<tr>
    <td colspan="2" align="center" bgcolor="#CCCCCC"><span class="style1">
        <marquee>
</marquee></span>
<table width="200" border="0" align="center">
<tr>
    <td colspan="3" align="center"></td>
</tr>
</table>
<span class="style1">
<marquee>
<span class="style7">Booking Information About New Visitor </span>
</marquee>
</span></td>
</tr>
<tr>
    <td width="237" align="center" valign="top"><p align="right" class="style1">Name

```

```

<input type="text" name="name" />
</p>
<p align="right" class="style1">Address
  <input type="text" name="address" />
</p>
<p align="right" class="style1">State
  <select name="state">
    <option>--select--</option>
    <option>Delhi</option>
    <option>Mumbai</option>
    <option>kolkata</option>
    <option>Chennai</option>
    <option>Lucknow</option>
    <option>Goa</option>
    <option>Simla</option>
  </select>
</p>
<p align="right" class="style1">City
  <input type="text" name="city" />
</p>
<p align="right" class="style1">Pin
  <input type="text" name="pin" />
</p>
<p align="right" class="style1">E-mail
  <input type="text" name="email" />
</p>
<p align="right" class="style1">Phone
  <input type="text" name="phone" />
</p>
<p align="right" class="style1">Hotel Name
  <input name="t1" type="text" id="t1" />
</p>
<p align="right" class="style1">Hotel Address
  <input name="t2" type="text" id="t2" />
</p>
<p align="right" class="style1">Type of Rooms
  <input name="t3" type="text" id="t3" />
</p>
<p align="right" class="style1">No. of Adults
  <select name="t4" id="t4">
    <option value="1">1</option>
    <option value="2">2</option>
    <option value="3">3</option>
    <option value="4">4</option>
    <option value="5">5</option>
    <option value="6">6</option>
  </select>
</p>
<p align="right" class="style1">No. of Chlid<span class="style2">
  <select name="t5" id="t5">
    <option value="1">1</option>
    <option value="2">2</option>
    <option value="3">3</option>
    <option value="4">4</option>
    <option value="5">5</option>
    <option value="6">6</option>
  </select>

```

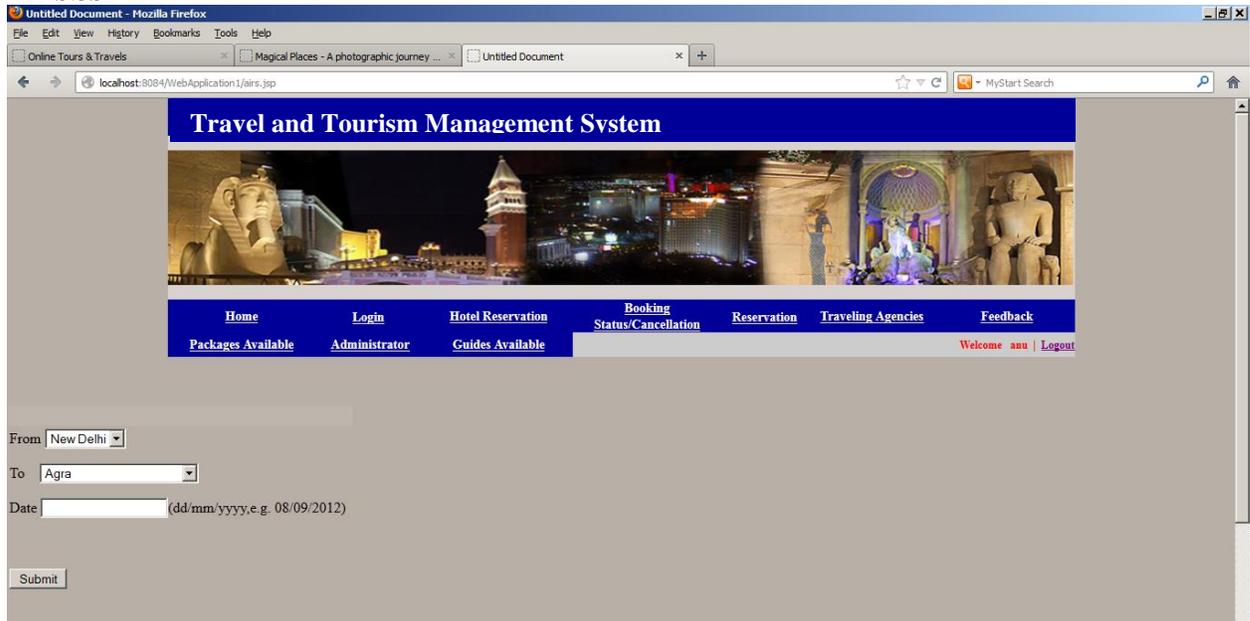
```

</select>
</span></p>
<p align="right" class="style1">Date of Booking
  <input name="t6" type="text" id="t6" /> (dd/mm/yyyy,e.g. 08/09/2012)
</p>
<p align="right" class="style1">Date of CheckIn
  <input name="t7" type="text" id="t7" />
</p>
<p align="right" class="style1">Date of CheckOut
  <input name="t8" type="text" id="t8" />
</p>
<p align="right" class="style1">Other Details
  <input name="t9" type="text" id="t9" />
</p>
<p align="right" class="style1">
  <input name="Submit" type="submit" class="style2" value="Submit" />
</p></td>
<td width="621"><table width="393" height="219" border="0" cellpadding="0" cellspacing="0">
  <tr>
    <td></td>
  </tr>
</table></td>
</tr>
</table>
<p>&nbsp;</p>
<%@include file="foot.jsp"%>
</form>
</body>
</html>

```

Projecthelpline.in

AIRS.JSP



```
<% @ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C/DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
<style type="text/css">
<!--
body {
    margin-left: 00px;
    margin-top: 00px;
    margin-right: 00px;
    margin-bottom: 00px;
    background-color: #B8AFA6;
}
-->
</style></head>

<body><script>

function validate(form)
{

function isValidDate(sText) {
    var reDate = /^(?:0[1-9]|[12][0-9]|3[01])\.(?:0[1-9]|1[0-2])\.(?:19|20\d{2})$/;
    return reDate.test(sText);
}
    if (isValidDate(form.journ.value))
    {

    }
    else
```



```
<td width="267">&nbsp;</td>
</tr>
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
</tr>
</table>
<p>
<label></label>
</p>
</p>
```

Projecthelpline.in

```

    <p>
      <label></label>
    </p>
    <p>
      <label></label>
    </p>
    <td width="483" bgcolor="#B8AFA6"></td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
</table>
<% @include file="foot.jsp"%>
</form>
</body>
</html>

```

Airdet.jsp

```

<% @ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
<style type="text/css">
<!--
body {
    margin-left: 00px;
    margin-top: 00px;
    margin-right: 00px;
    margin-bottom: 00px;
    background-color: #B8AFA6;
}
-->
</style></head>

<body>
<form id="form1" name="form1" method="post" action="air.jsp" onsubmit="return validate
(this)">
<%@include file="header.jsp"%>

<% @ include file="connect.jsp"%>
    <% String journ=request.getParameter("journ");

        String f=request.getParameter("from");
        String t=request.getParameter("to");out.print(f+" - "+t);
Statement smt=con.createStatement();
String query="select * from airsc where fro='"+f+"' and too='"+t+"'";
int c=0;
try{
ResultSet rs1=smt.executeQuery("select sum(no) from air where too='"+t+"' and journ='"+journ+"'");

```

```

while(rs1.next())
{
    c=Integer.parseInt(rs1.getString(1));
}
}
catch(Exception obj)
{

}
int seat=30-c;

```

```

ResultSet rs=smt.executeQuery(query);

```

```

%>

```

```

<p>&nbsp;</p>

```

```

<table width="932" height="197" border="0" cellpadding="0" cellspacing="0">

```

```

<tr>

```

```

    <table width="637" border="1" align="center" cellspacing="0" textcolor="red">

```

```

<tr>

```

```

<td width="65"><strong>From</strong></td>

```

```

<td width="74"><strong>To</strong></td>

```

```

<td width="65"><strong>Airlines Name </strong></td>

```

```

<td width="74"><strong>Flight No </strong></td>

```

```

<td width="74"><strong>Timings </strong></td>

```

```

<td width="74"><strong>Charges </strong></td>

```

```

</tr>

```

```

<%

```

```

while(rs.next())

```

```

{

```

```

%>

```

```

<tr>

```

```

<td><%=rs.getString(1)%></td>

```

```

<td><%=rs.getString(2)%></td>

```

```

<td><%=rs.getString(3)%></td>

```

```

<td><%=rs.getString(4)%></td>

```

```

<td><%=rs.getString(5)%></td>

```

```

<td><%=rs.getString(6)%><br> <b>Seats Available<%=seat%> per seat</b></td>

```

```

<td width="74"><strong><a href='air.jsp?id=<%=rs.getString(4)%>'>Booking</a> </strong></td>

```

```

</tr>

```

```

<% }%>

```

```

</table></td>

```

```

    </tr>

```

```

</tr>

```

```

</table>

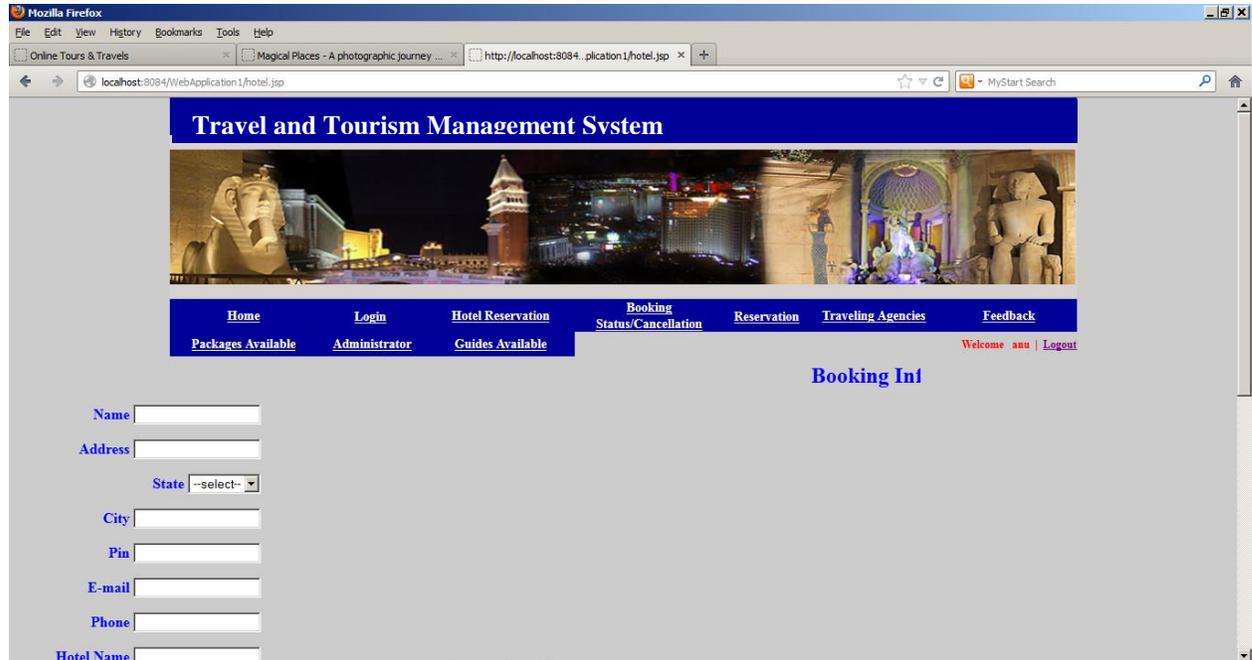
```

```

<% @include file="foot.jsp"%>
</form>
</body>
</html>

```

HOTEL.JSP



```

<% @ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title></title>
<style type="text/css">
<!--
body {
margin-left: 0px;
margin-top: 0px;
margin-right: 0px;
margin-bottom: 0px;
background-color: #CCCCCC;
}
.style1 {
color: #0000FF;
font-weight: bold;
}
.style2 {
color: #0000FF;
font-weight: bold;
}
.style3 {color: #0000FF}
.style7 {font-size: 24px}
.style9 {color: #990000; font-weight: bold; font-size: 18px; }

```

```

.style10 {font-size: 18px; color: #990000;}
.style12 {
    font-size: 24px;
    font-weight: bold;
    color: #0000FF;
}
-->
</style></head>

<script>

function validate(form)
{
if (form.name.value=="")
    {
        alert("Name should not be blank. Please enter it.");
        form.name.focus(); return false;
    }

if (form.address.value=="")
    {
        alert("address should not be blank. Please enter it.");
        form.address.focus(); return false;
    }

if (form.city.value=="")
    {
        alert("city should not be blank. Please enter it.");
        form.city.focus(); return false;
    }

if (form.pin.value=="")
    {
        alert("pin should not be blank. Please enter it.");
        form.pin.focus(); return false;
    }
    if(isNaN(form.pin.value))
    {
        alert("pin Should be a number ");
        form.pin.focus();
        return false;
    }
if (form.email.value=="")
    {
        alert("email should not be blank. Please enter it.");
        form.email.focus(); return false;
    }
    var x=form.email.value
    var atpos=x.indexOf("@");
    var dotpos=x.lastIndexOf(".");
    if (atpos<1 || dotpos<atpos+2 || dotpos+2>=x.length)
    {
        alert("Not a valid e-mail address");
        form.email.focus();
        return false;
    }

if (form.phone.value=="")
    {
        alert("phone should not be blank. Please enter it.");
        form.phone.focus(); return false;
    }
}

```

```

        if(isNaN(form.phone.value))
        {
            alert("phone Should be a number ");
form.phone.focus();
            return false;
        }
if((form.phone.value).length < 10)
{
    alert(" Your mobile no. must be atleast 10 digits ");
    return false;
}

if (form.t1.value=="")
{
    alert("Hotel Name should not be blank. Please enter it.");
    form.t1.focus(); return false;
}
if (form.t2.value=="")
{
    alert("Hotel Address should not be blank. Please enter it.");
    form.t2.focus(); return false;
}

if (form.t3.value=="")
{
    alert("Type of room should not be blank. Please enter it.");
    form.t3.focus(); return false;
}

if (form.t7.value=="")
{
    alert("Date should not be blank. Please enter it.");
    form.t7.focus(); return false;
}
if (form.t7.value=="")
{
    alert("Booking Date should not be blank. Please enter it.");
    form.t7.focus(); return false;
}if (form.t8.value=="")
{
    alert("Checkin Date should not be blank. Please enter it.");
    form.t8.focus(); return false;
}
if (form.t9.value=="")
{
    alert("Checkout Date should not be blank. Please enter it.");
    form.t9.focus(); return false;
}

function isValidDate(sText) {
var reDate = /^(?:0[1-9]|[12][0-9]|3[01])\.(?:0[1-9]|1[0-2])\.(?:19|20\d{2})/;
    return reDate.test(sText);
}
if (isValidDate(form.t7.value))
{
}
else
{
    alert("Invalid Booking Date !");
    form.t7.value=""
    form.t7.focus(); return false;
}

```

```

    }

    if (isValidDate(form.t8.value))
    {

    }
    else
    {
        alert("Checkin Invalid Date !");
        form.t8.value=""
        form.t8.focus(); return false;
    }

}
</script>

<body>
    <form id="form1" name="form1" method="post" action="hotel" onsubmit="return validate
(this)">
    <% @include file="header.jsp"%>
<table width="1002" height="117" border="0" cellpadding="0" cellspacing="0">
<tr>
<td colspan="2" align="center" bgcolor="#CCCCCC"><span class="style1">
<marquee>
</marquee></span>
<table width="200" border="0" align="center">
<tr>
<td colspan="3" align="center"></td>
</tr>
</table>
<span class="style1">
<marquee>
<span class="style7">Booking Information About New Visitor </span>
</marquee>
</span></td>
</tr>
<tr>
<td width="237" align="center" valign="top"><p align="right" class="style1">Name
<input type="text" name="name" />
</p>
<p align="right" class="style1">Address
<input type="text" name="address" />
</p>
<p align="right" class="style1">State
<select name="state">
<option>--select--</option>
<option>Delhi</option>
<option>Mumbai</option>
<option>kolkata</option>
<option>Chennai</option>
<option>Lucknow</option>
<option>Goa</option>
<option>Simla</option>

```

```

    </select>
</p>
<p align="right" class="style1">City
  <input type="text" name="city" />
</p>
<p align="right" class="style1">Pin
  <input type="text" name="pin" />
</p>
<p align="right" class="style1">E-mail
  <input type="text" name="email" />
</p>
<p align="right" class="style1">Phone
  <input type="text" name="phone" />
</p>
<p align="right" class="style1">Hotel Name
  <input name="t1" type="text" id="t1" />
</p>
<p align="right" class="style1">Hotel Address
  <input name="t2" type="text" id="t2" />
</p>
<p align="right" class="style1">Type of Rooms
  <input name="t3" type="text" id="t3" />
</p>
<p align="right" class="style1">No. of Adults
  <select name="t4" id="t4">
    <option value="1">1</option>
    <option value="2">2</option>
    <option value="3">3</option>
    <option value="4">4</option>
    <option value="5">5</option>
    <option value="6">6</option>
  </select>
</p>
<p align="right" class="style1">No. of Child<span class="style2">
  <select name="t5" id="t5">
    <option value="1">1</option>
    <option value="2">2</option>
    <option value="3">3</option>
    <option value="4">4</option>
    <option value="5">5</option>
    <option value="6">6</option>
  </select>
</span></p>
<p align="right" class="style1">Date of Booking
  <input name="t6" type="text" id="t6" /> (dd/mm/yyyy,e.g. 08/09/2012)
</p>
<p align="right" class="style1">Date of CheckIn
  <input name="t7" type="text" id="t7" />
</p>
<p align="right" class="style1">Date of CheckOut
  <input name="t8" type="text" id="t8" />
</p>
<p align="right" class="style1">Other Details
  <input name="t9" type="text" id="t9" />
</p>
<p align="right" class="style1">

```

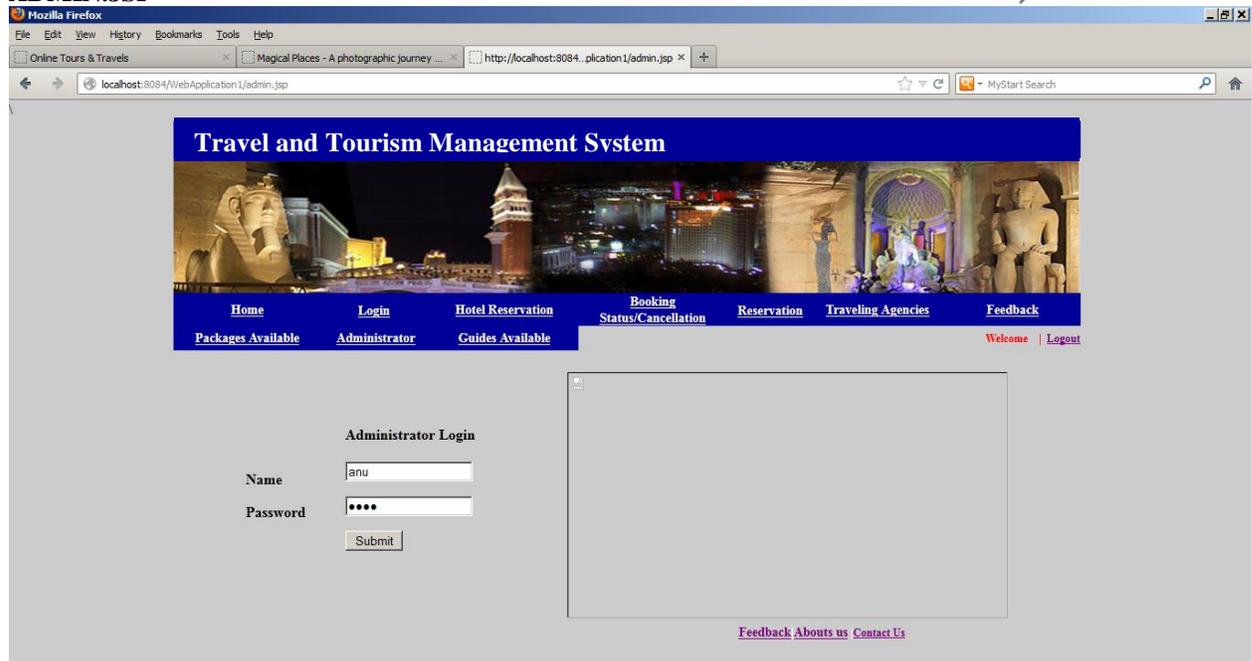
```



```

in

ADMIN.JSP



```

<% @ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title></title>
<style type="text/css">
<!--
body {
    background-color: #CCCCCC;
    margin-left: 0px;
    margin-top: 0px;
}
.style2 {
    font-size: 24px;

```

```

        font-weight: bold;
    }
-->
</style>
<script language="javascript" type="text/javascript">
    window.history.forward(1);
</script>
</head>

<body><%
if(session.getAttribute("ins1")!=null)
{

    response.sendRedirect("logout1.jsp");
}%>
<script>

function validate(form)
{
if (form.name.value=="")
    {
        alert("Name should not be blank. Please enter it.");
        form.name.focus(); return false;
    }

if (form.password.value=="")
    {
        alert("password should not be blank. Please enter it.");
        form.password.focus(); return false;
    }

    }
</script>

<table width="200" border="0" align="center">
<tr>
<td colspan="3" align="center"><% @include file="header.jsp"%></td>
</tr>
<tr>
<td colspan="3" align="center"><table width="837" height="189" border="0" cellpadding="0"
cellspacing="0" bgcolor="#FFFFFF">
<tr bgcolor="#CCCCCC">
<td height="19" colspan="3">&nbsp;</td>
</tr>
<tr>
<td width="151" bgcolor="#CCCCCC"><p>&nbsp;</p>
<p>&nbsp;</p>
<p><strong>Name</strong></p>
<p><strong>Password</strong></p>

<p>&nbsp;</p>
<p>&nbsp;</p></td>
<td width="343" bgcolor="#CCCCCC"><form id="form1" name="form1" method="post"
action="adminlog" onsubmit="return validate (this)">
<label>
<strong>Administrator Login</strong><br />
<br />

```

```

        <input name="name" type="text" id="name" />
        </label>
    <p>
        <label>
            <input name="password" type="password" id="password" />
        </label>
    </p>

    <p>
        <label>
            <input type="submit" name="Submit" value="Submit" />
        </label>
    </p>
</form></td>
<td width="343" bgcolor="#CCCCCC"><div align="right"><span class="style2"><a
href="home.jsp"></a></span></div></td>
</tr>
</table></td>
</tr>
</table>
<% @include file="foot.jsp"%>
</body>
</html>

```

Afminlog.jsp

```
package src;
```

```

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.sql.*;
import javax.servlet.http.HttpSession;

public class adminlog extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {
            String str=request.getParameter("name");
            String str1=request.getParameter("password");

            Connection con=Connect.getConnection();
            Statement statement = con.createStatement();
            String query="select * from admin where name='"+str+"' and password='"+str1+"'";
            ResultSet x = statement.executeQuery(query);
            HttpSession session = request.getSession(true);
            int i=0;
            while(x.next()==true)
            {
                i++;}

```

```

        if(i>0)
        {
response.sendRedirect("admin menu.jsp");
session.setAttribute("ins1",str);
        }
else
        {
response.sendRedirect("invalida.jsp");

        }
    }
    catch(Exception obj)
    {

        // out.print(obj.getMessage());

    }
    out.println("<html>");
    out.println("<head>");
    out.println("<title>Servlet adminlog</title>");
    out.println("</head>");
    out.println("<body>");
    out.println("<h1>Servlet adminlog at " + request.getContextPath() + "</h1>");
    out.println("</body>");
    out.println("</html>");

}

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit
the code.">
/**
 * Handles the HTTP
 * <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Handles the HTTP
 * <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)

```

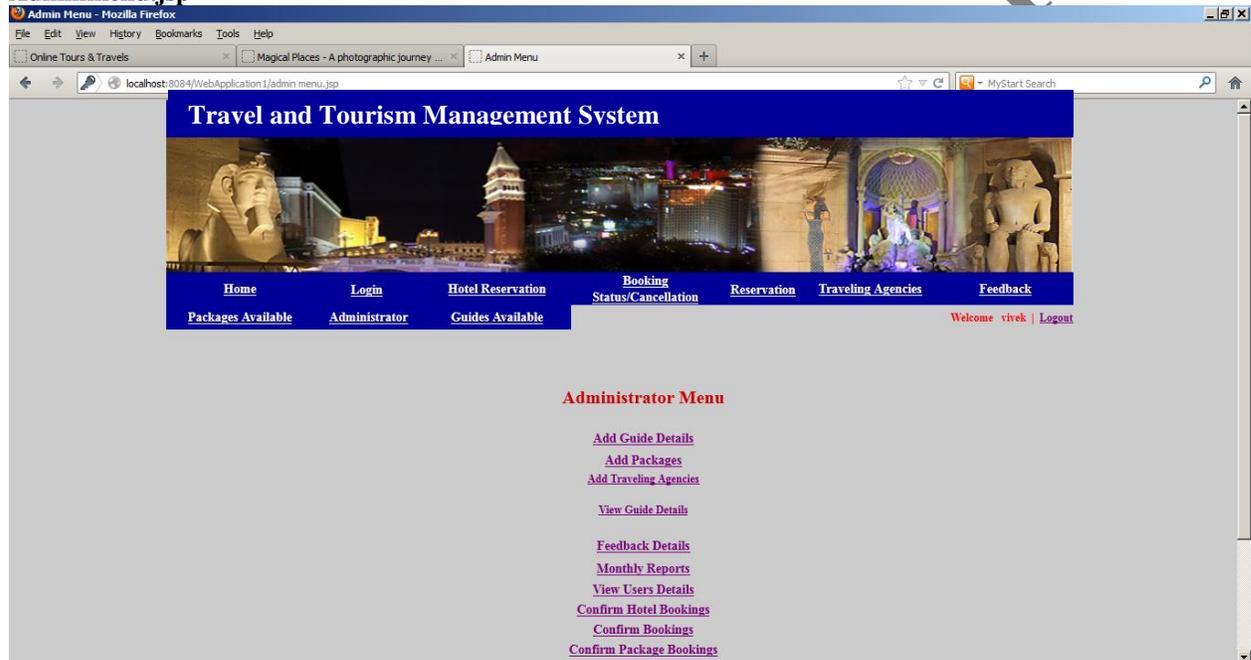
```

        throws ServletException, IOException {
    processRequest(request, response);
    }

/**
 * Returns a short description of the servlet.
 *
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {
    return "Short description";
} // </editor-fold>

```

Adminmenu.jsp



```

<% @ page language="java" %>
<% @ page session="true" %>
<html>
<head>
<title>Admin Menu</title>
<style type="text/css">
<!--
.style10 {font-size: 24px; color: #0000FF;}
.style12 {
    font-size: 16px;
    font-weight: bold;
}
.style13 {
    color: #0000FF;
    font-weight: bold;
}
.style14 {color: #0000FF}
.style17 {color: #990000; font-weight: bold; }
.style19 {color: #000000; font-weight: bold; }

```

```

.style20 {font-size: 14px}
.style21 {color: #990000; }
.style22 {font-size: 20px; font-weight: bold; }
body {
    background-color: #CCCCCC;
}
-->
</style>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"></head>
<body>
<form name="form1" method="post" action="">
<table width="720" border="0" align="center" cellspacing="0">
<tr>
    <td colspan="2" height="120"><% @ include file="header.jsp" %>
    <%
    if(session.getAttribute("ins1")!=null)
    { %>
        </td>
    </tr>

<tr>
<td width="36" align="left" valign="top">
<td width="680" align="left" valign="top" bgcolor="#CCCCCC">
    <table width="420" height="304" border="0" align="center">
        <tr>
            <td height="35" align="left" valign="top">&nbsp;</td>
        </tr>
        <tr>
            <td height="35" align="left" valign="top"><div align="center" class="style10">
                <p class="style22">Administrator Menu </p>
            </div></td>
        </tr>
        <tr>
            <td width="414"><div align="center" class="style20"><a href="guides.jsp"><strong>Add Guide
Details </strong></a></div></td>
        </tr>
        <tr>
            <td width="414"><div align="center" class="style20"><a href="packages.jsp"><strong>Add Packages
</strong></a></div></td>
        </tr>
        <tr>
            <td><div align="center"><span class="style17 style14 style20"><span class="style21"><a
href="travelinag.jsp">Add Traveling Agencies </a></span></span></div></td>
        </tr>
        <tr>
            <td height="47"><div align="center"><span class="style17 style14 style20"><span class="style21"><a
href="showguide.jsp">View Guide Details </a> </span></span></div></td>
        </tr>
        <tr>
            <td height="27"><div align="center"><strong><a href="showfeedback.jsp" class="style20">Feedback
Details</a> </strong></div></td>

```

```

        </tr>
        <tr>
        <td height="21" align="center"><a href="daterepo.jsp" class="style20"><strong>Monthly Reports
</strong></a></td>
        </tr>

        <tr>
        <td><div align="center" class="style19"><a href="showusers.jsp" class="style20">View Users
Details</a> </div></td>
        </tr>
        <tr>
        <td><div align="center" class="style19"><a href="confirmhotel.jsp" class="style20">Confirm Hotel
Bookings</a> </div></td>
        </tr>
        <tr>
        <td><div align="center" class="style19"><a href="confirmbook.jsp" class="style20">Confirm
Bookings </a></div></td>
        </tr>
        <tr>
        <td><div align="center" class="style19"><a href="showpackbook.jsp" class="style20">Confirm
Package Bookings </a></div></td>
        </tr>
        <tr>
        <td><div align="center" class="style19"><a href="showair.jsp" class="style20">View Bookings By
Air </a></div></td>
        </tr>

        <tr>
        <td><div align="center" class="style12"><a href="showtrain.jsp" class="style20">View Bookings By
Train </a></div></td>
        </tr>
        <tr>
        <td height="21" align="center"><a href="showbus.jsp" class="style20"><strong>Bookings By Bus
</strong></a></td>
        </tr>

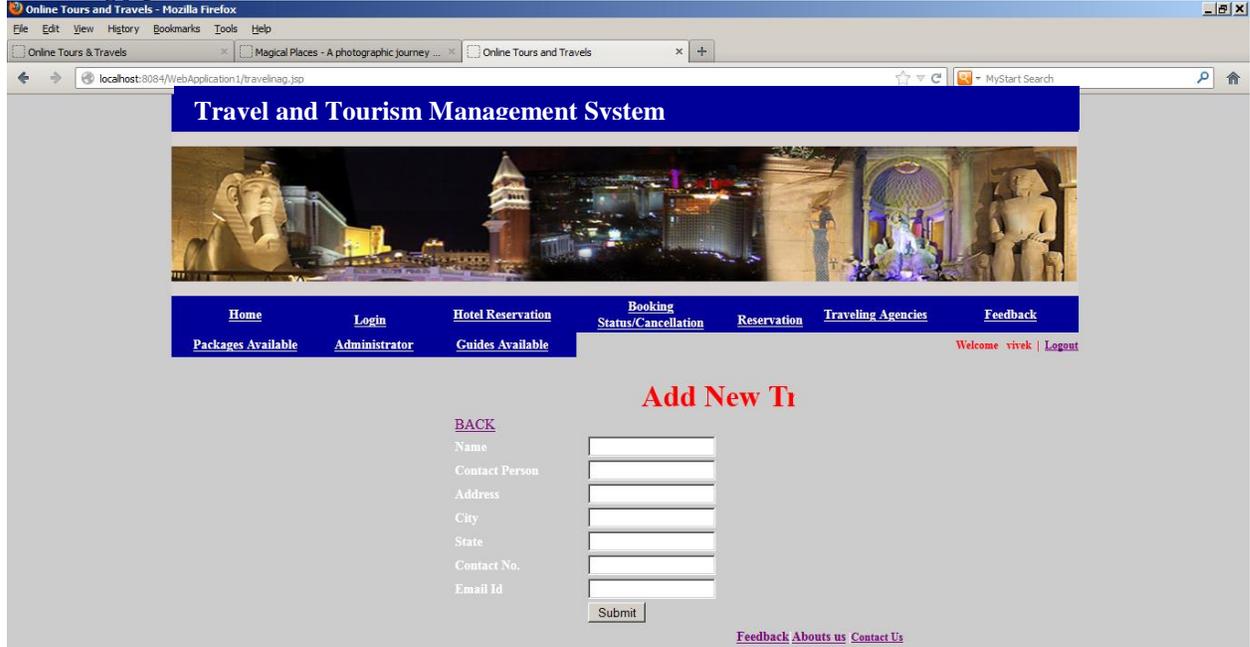
        <tr>
        <td height="57"><p align="center" class="style13"><a href="showhotel.jsp" class="style20">View
Hotel Detail </a></p>
        <p align="center">&nbsp;</p> </td>
        </tr>
</table>

</tr>
</table>
<%
}
else
{
response.sendRedirect("admin.jsp");
}%>
<% @include file="foot.jsp"%>
</form>
</body>

```

</html>

Traveling.jsp



```
<% @ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
```

```
<title>Online Tours and Travels </title>
```

```
<style type="text/css">
```

```
<!--
```

```
body {
```

```
background-color: #CCCCCC;
```

```
margin-left: 0px;
```

```
margin-top: 0px;
```

```
}
```

```
.style2 {
```

```
color: #CC0033;
```

```
font-weight: bold;
```

```
}
```

```
.style4 {color: #CC0033; font-weight: bold; font-size: 14px; }
```

```
.style5 {font-size: 33px }
```

```
.style6 { color: #FF0000;
```

```
font-weight: bold;
```

```
}
```

```
-->
```

```
</style>
```

```
</head>
```

```
<script>
```

```

function validate(form)
{
if (form.name.value=="")
    {
        alert("Agency Name should not be blank. Please enter it.");
        form.name.focus(); return false;
    }

    if (form.cper.value=="")
        {
            alert("Person name not be blank. Please enter it.");
            form.cper.focus(); return false;
        }

if (form.address.value=="")
    {
        alert("Address Should not be blank. Please enter it.");
        form.address.focus(); return false;
    }

    if (form.city.value=="")
        {
            alert("City should not be blank. Please enter it.");
            form.city.focus(); return false;
        }

    if (form.state.value=="")
        {
            alert("State should not be blank. Please enter it.");
            form.state.focus(); return false;
        }

if (form.phone.value=="")
    {
        alert("Phone No. should not be blank. Please enter it.");
        form.phone.focus(); return false;
    }
    if(isNaN(form.phone.value))
    {
        alert("Phone No. Should be a number ");
        form.phone.focus();
        return false;
    }
if((form.phone.value).length < 10)
    {
        alert(" Your mobile no. must be atleast 10 digits ");
        form.phone.focus();
        return false;
    }
if (form.email.value=="")
    {
        alert("Email should not be blank. Please enter it.");
        form.email.focus(); return false;
    }

    var x=form.email.value
    var atpos=x.indexOf("@");
    var dotpos=x.lastIndexOf(".");
    if (atpos<1 || dotpos<atpos+2 || dotpos+2>=x.length)
        {
            alert("Not a valid e-mail address");
            form.email.focus();
        }
}

```

```

return false;
}

}
</script>
<body>
<form id="form1" name="form1" method="post" action="agency" onsubmit="return validate
(this)">
  <% @include file="header.jsp"%>
<table width="379" align="center">
  <tr>
  <td colspan="2">
    <span class="style1">
      <marquee>
        </marquee>
      </span>
      <marquee>
        <div align="center" class="style5"><span class="style1"><strong> Add New Traveling
Agency </strong> </span></div>
        </marquee>
        <a href="admin menu.jsp">BACK</a></div>
      </td>
    </tr>
    <tr>
    <td width="140"><span class="style4">Name </span></td>
    <td width="223"><input name="name" type="text" id="name" /></td>
    </tr>
    <tr>
    <td><span class="style4">Contact Person </span></td>
    <td><input name="cper" type="text" /></td>
    </tr>
    <tr>
    <td><span class="style4">Address</span></td>
    <td><input name="address" type="text" id="address" /></td>
    </tr>
    <tr>
    <td><span class="style4">City</span></td>
    <td><input name="city" type="text" id="city" /></td>
    </tr>
    <tr>
    <td><span class="style4">State</span></td>
    <td><input name="state" type="text" id="state" /></td>
    </tr>
    <tr>
    <td><span class="style4">Contact No.</span></td>
    <td><input name="phone" type="text" id="phone" /></td>
    </tr>
    <tr>
    <td><span class="style4">Email Id </span></td>
    <td><input name="email" type="text" id="email" /></td>
    </tr>
    <tr>
    <td>&nbsp;</td>
    <td><input type="submit" name="Submit" value="Submit" />
      <span class="style6">
        <%if(session.getAttribute("ins")!=null){String str=session.getAttribute("ins").toString();}%>
      </span></td>
    </tr>
  </table>

```

```

        </table>
<% @include file="foot.jsp"%>
</form>
</body>
</html>

```

Agency.jsp

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package src;

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.sql.*;
import javax.servlet.http.HttpSession;
public class agency extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        HttpSession session = request.getSession(true);

        if(session.getAttribute("ins1")!=null)
        {
            try {
                String name=request.getParameter("name");
                String email=request.getParameter("email");
                String add=request.getParameter("address");
                String c=request.getParameter("city");
                String d=request.getParameter("state");
                String e=request.getParameter("phone");
                String f=request.getParameter("email");

                out.println("<html>");
                out.println("<head>");
                out.println("<title>Servlet agency</title>");
                out.println("</head>");
                out.println("<body>");

                Connection con=Connect.getConnect();
                Statement smt=con.createStatement();
                String query="insert into traveling_agency
                values('"+name+"','"+email+"','"+add+"','"+c+"','"+d+"','"+e+"','"+f+"')";
                smt.executeUpdate(query);
                session.setAttribute("ins","Record is Inserted");

                response.sendRedirect("travelinag.jsp");
            }

```

```

        }catch(Exception obj)
        {
            out.print(obj.getMessage());
        }
        finally {
            out.close();
        }
    }
    else
    {
        response.sendRedirect("admin.jsp");
    }

    out.println("<h1>Servlet agency at " + request.getContextPath() + "</h1>");
    out.println("</body>");
    out.println("</html>");

}

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit
the code.">
/**
 * Handles the HTTP
 * <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Handles the HTTP
 * <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Returns a short description of the servlet.
 *

```

```

* @return a String containing servlet description
*/
@Override
public String getServletInfo() {
    return "Short description";
} // </editor-fold>
}

```

Showguide.jsp



```

<%@ page language="java" %>
<%@ page session="true" %>
<html>
<head>
<title>ONLINE TOURS AND TRAVELS</title>
<style type="text/css">
<!--
.style7 {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 16px;
    font-weight: bold;
    color: #0000FF;
}
.style8 {
    font-size: 18px;
    color: #CC3300;
}
.style19 {
    color: #990000;
    font-weight: bold;
}
.style20 {color: #330066}
.style22 {color: #330066; font-weight: bold; }
body {
    background-color: #CCCCCC;
}

```



```

{
%>
<tr>
<td><div align="center"><span class="style3"><%=rs.getString(1)%></span></div></td>
<td><div align="center"><span class="style3"><%=rs.getString(2)%></span></div></td>
<td><div align="center"><span class="style3"><%=rs.getString(3)%></span></div></td>
<td><div align="center"><span class="style3"><%=rs.getString(4)%></span></div></td>
<td><div align="center"><span class="style3"><%=rs.getString(5)%></span></div></td>
<td><div align="center"><span class="style3"><%=rs.getString(6)%></span></div></td>
<td><div align="center"><span class="style3"><%=rs.getString(7)%></span></div></td>

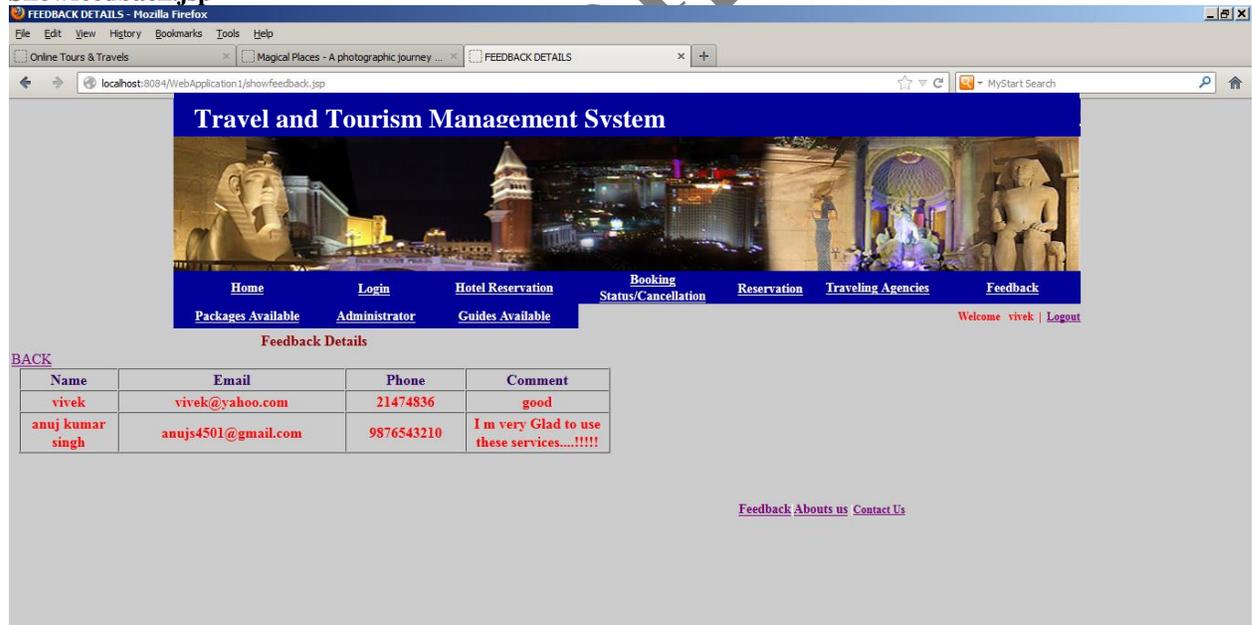
</tr>
<% }%>
</table></td>
</tr>
</table>

<h1 class="style8">&nbsp;</h1>
</td>
</tr>
</table>
<% @include file="foot.jsp"%>
</form>
</body>
</html>

```

ipline.in

Showfeedback.jsp



```

<% @ page language="java" %>
<% @ page session="true" %>
<html>
<head>
<title>FEEDBACK DETAILS</title>
<style type="text/css">
<!--
.style7 {

```

```

        font-family: Arial, Helvetica, sans-serif;
        font-size: 16px;
        font-weight: bold;
        color: #0000FF;
    }
    .style8 {
        font-size: 18px;
        color: #CC3300;
    }
    .style19 {
        color: #990000;
        font-weight: bold;
    }
    .style20 {color: #330066}
    .style22 {color: #330066; font-weight: bold; }
    body {
        background-color: #CCCCCC;
    }
-->
</style>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"></head>
<body>
<form name="form1" method="post" action="">
<% @ include file="header.jsp" %>

<table width="671" height="160" border="0">
<tr>
<td width="244" height="156" colspan="2" align="left" valign="top"><div align="center">
<% @ include file="connect.jsp"%>
<%
Statement smt=con.createStatement();
String query="select * from feedbacks";
ResultSet rs=smt.executeQuery(query);
%>
<span class="style19">Feedback Details</span>
</div> <a href="admin menu.jsp">BACK</a></div>
<table width="650" border="1" align="center" cellspacing="0" textcolor="red">
<tr>
<td width="65"><div align="center"><span class="style22">Name</span></div></td>
<td width="74"><div align="center"><span class="style20 style1"><strong>Email</strong></span>
</div></td>
<td width="74"><div align="center"><span class="style20 style1"><strong>Phone</strong></span>
</div></td>
<td width="74"><div align="center"><span class="style20 style1"><strong>Comment</strong></span>
</div></td>

</tr>
<%
while(rs.next())
{
%>
<tr>
<td><div align="center"><span class="style3"><%=rs.getString(1)%></span></div></td>
<td><div align="center"><span class="style3"><%=rs.getString(2)%></span></div></td>
<td><div align="center"><span class="style3"><%=rs.getString(3)%></span></div></td>
<td><div align="center"><span class="style3"><%=rs.getString(4)%></span></div></td>

```

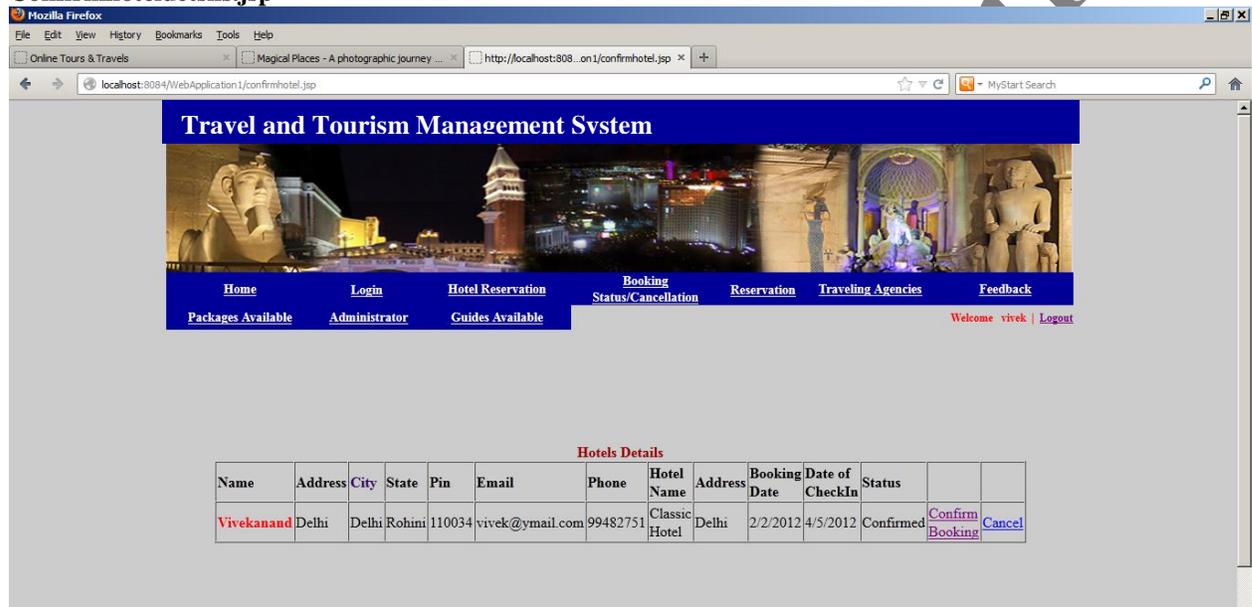
```

</tr>
<% }%>
</table>

<h1 class="style8">&nbsp;</h1>
</td>
</tr>
</table>
<% @include file="foot.jsp"%>
</form>
</body>
</html>

```

Confirmhoteldetsils.jsp



```

<% @ page language="java" %>
<% @ page session="true" %>
<html>
<head>
<title></title>
<style type="text/css">
<!--
.style7 {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 16px;
    font-weight: bold;
    color: #0000FF;
}
.style8 {
    font-size: 18px;
    color: #CC3300;
}
.style19 {
    color: #990000;
    font-weight: bold;
}
.style20 {color: #330066}

```

```

body {
    background-color: #CCCCCC;
}
-->
</style>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"></head>
<body>
<form name="form1" method="post" action="">
<% @ include file="header.jsp" %>
<table width="731" border="0" align="center" cellspacing="0">
<tr>
    <td colspan="2" height="120"></td>
</tr>

<tr>
<td width="150" align="left" valign="top">

<td width="750" align="left" valign="top" bgcolor="#CCCCCC">
    <table width="671" height="282" border="0">
        <tr>
            <td width="244" height="278" colspan="2" align="left" valign="top" bgcolor="#CCCCCC"><div
align="center">
                <% @ include file="connect.jsp"%>
                <%
                    if(session.getAttribute("ins1")!=null)
                { %>
                    <%
                        Statement smt=con.createStatement();
                        String query="select * from hotl";
                        ResultSet rs=smt.executeQuery(query);
                        %>
                            <span class="style19">Hotels Details</span>
                            </div>
                            <table width="637" border="1" align="center" cellspacing="0" textcolor="red">
                                <tr>
                                    <td width="65"><strong>Name</strong></td>
                                    <td width="74"><strong>Address</strong></td>
                                    <td width="74"><span class="style20 style1"><strong>City </strong></span> </td>
                                    <td width="74"><strong>State</strong></td>
                                    <td width="65"><strong>Pin</strong></td>
                                    <td width="74"><strong>Email </strong></td>
                                    <td width="74"><strong>Phone </strong></td>
                                    <td width="74"><strong>Hotel Name</strong></td>
                                    <td width="65"><strong>Address</strong></td>
                                    <td width="74"><strong>Booking Date</strong></td>
                                    <td width="74"><strong>Date of CheckIn </strong></td>
                                    <td width="74"><strong>Status </strong></td>
                                    <td width="74"><strong></strong></td>

                                <td width="74"><strong></strong></td>
                                </tr>
                                <%
                                    while(rs.next())
                                {
                                %>

```

```

<tr>
<td><span class="style3"><%=rs.getString(1)%></span></td>
<td><%=rs.getString(2)%></td>
<td><%=rs.getString(3)%></td>
<td><%=rs.getString(4)%></td>
<td><%=rs.getString(5)%></td>
<td><%=rs.getString(6)%></td>
<td><%=rs.getString(7)%></td>
<td><%=rs.getString(8)%></td>
<td><%=rs.getString(9)%></td>
<td><%=rs.getString(13)%></td>
<td><%=rs.getString(14)%></td>
<td><%=rs.getString(17)%></td>
<td width="74"><a href="confirm.jsp?n=<%=rs.getString(18)%>">Confirm Booking</a></td>
<td width="74"><a href="cancel.jsp?n=<%=rs.getString(18)%>">Cancel</a></td>

```

```

</tr>
<% }%>
</table></td>
</tr>
</table>

```

```

<h1 class="style8">&nbsp;</h1>
</td>
</tr>
</table>
<%
}
else
{
response.sendRedirect("admin.jsp");
}%>
<%@include file="foot.jsp"%>
</form>
</body>
</html>

```

Projecthelpline.in

5.2 CODING STANDARDS

Reviewing of Code efficiency for a module is carried out after the module is successfully compiled and all the syntax errors eliminated. Code efficiency review is extremely cost-effective strategies for reduction in coding errors in order to produce high quality code. Normally, two types of efficiency are carried out on the code of a module - code optimization and code inspection. The procedure and final objective of these two efficiency techniques are very different as discussed below.

5.3 OPTIMIZATION OF CODE

Code optimization is an informal code analysis technique. In this technique, after a module has been coded, it is successfully compiled and all syntax errors are eliminated. Some members of the development team are given the code a few days before the optimization meeting to read and understand the code. Each member selects some test cases and simulates execution of the code by hand (i.e. trace execution through each statement and function execution). The main objectives of the optimization are to discover the algorithmic and logical errors in the code. The members note down their findings to discuss these in a optimization meeting where the coder of the module is also present.

Even though a code optimization is an informal analysis technique, several guidelines have evolved over the years for making this naïve technique more effective and useful. Of course, these guidelines are based on personal experience, common sense, and several subjective factors. Therefore are based on personal experience, common sense, and several subjective factors. Therefore, guidelines should be considered as examples rather than as rules to be applied dogmatically. Some of these guidelines are the following:

The team performing the code optimization should not be either too big or too small. Ideally, it should consist of three to seven members.

CHAPTER 6

TESTING

6.1 TESTING PHASE

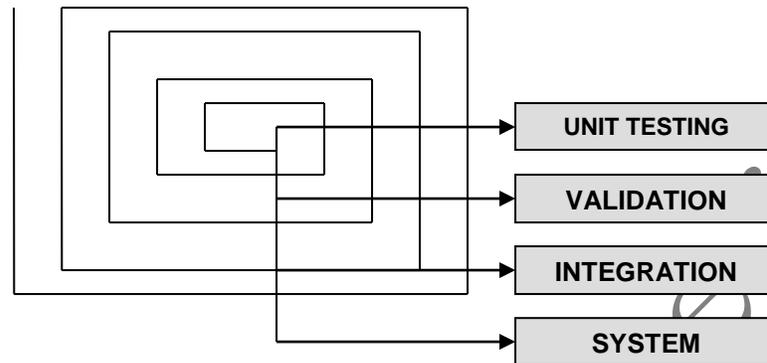
In a software development project, errors can be injected at any stage during development. There are different techniques for detecting and eliminating errors that originate in that phase. However, no technique is perfect, and it is expected that some of the errors of the earlier phases will finally manifest themselves in the code. This is particularly true because in the earlier phases and most of the verification techniques are manual because no executable code exists. Ultimately, these remaining errors will be reflected in the code. Hence, the code developed during the coding activity is likely to have some requirement errors and design errors, in addition to errors introduced during the coding activity. Behavior can be observed, testing is the phase where the errors remaining from all the previous phases must be detected. Hence, testing performs a very critical role for quality assurance and for ensuring the reliability of software.

During testing, the program to be tested is executed with a set of test cases, and the output of the program for the test cases is evaluated to determine if the program is performing as expected. Due to its approach, dynamic testing can only ascertain the presence of errors in the program; the exact nature of the errors is not usually decided by testing. Testing forms the first step in determining the errors in a program. Clearly, the success of testing in revealing errors in programs depends critically on the test cases.

Testing a large system is a very complex activity, and like any complex activity it has to be broken into smaller activities. Due to this, for a project, incremental testing is generally performed, in which components and subsystems of the system are tested separately before integrating them to form the system for system testing. This form of testing, though necessary to ensure quality for a large system, introduces new issues of how to select components for testing and how to combine them to form subsystems and systems.

6.2 LEVELS OF TESTING

The different types of testing are as follows:



1. UNIT TESTING:

This is the smallest testable unit of a computer system and is normally tested using the white box testing. The author of the programs usually carries out unit tests.

2. INTEGRATION TESTING:

In integration testing, the different units of the system are integrated together to form the complete system and this type of testing checks the system as whole to ensure that it is doing what is supposed to do. The testing of an integrated system can be carried out top-down, bottom-up, or big-bang. In this type of testing, some parts will be tested with white box testing and some with black box testing techniques. This type of testing plays very important role in increasing the systems productivity. We have checked our system by using the integration testing techniques.

3. SYSTEM TESTING:

A part from testing the system to validate the functionality of software against the requirements, it is also necessary to test the non-functional aspect of

the system. Some examples of non-functional tools include tests to check performance, data security, usability/user friendliness, volume, load/stress that we have used in our project to test the various modules.

System testing consists of the following steps:

1. Program(s) testing.
2. String testing.
3. System testing.
4. System documentation.
5. User acceptance testing.

4. FIELD TESTING:

This is a special type of testing that may be very important in some projects. Here the system is tested in the actual operational surroundings. The interfaces with other systems and the real world are checked. This type of testing is very rarely used. So far our project is concerned; we haven't tested our project using the field testing.

5. ACCEPTANCE TESTING:

After the developer has completed all rounds of testing and he is satisfied with the system, then the user takes over and re-tests the system from his point of view to judge whether it is acceptable according to some previously identified criteria. This is almost always a tricky situation in the project because of the inherent conflict between the developer and the user. In this project, it is the job of the bookstores to check the system that whether the made system fulfills the goals or not.

WHY SYSTEM TESTING?

Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved. Inadequate testing results in two types of problems:

1. The time lag between the cause and the appearance of the problem.

2. The effect of system errors on the files and records within the system.

ACTIVITY NETWORK FOR SYSTEM TESTING

The test plan entails the following activities:

1. Prepare test plan.
2. Specify conditions for user acceptance testing.
3. Prepare test data for program testing.
4. Prepare test data for transaction path testing.
5. Plan user training.
6. Compile/assemble programs.
7. Prepare job performance aids.
8. Prepare operational documents.

PREPARE TEST: A workable test plan must be prepared in accordance with established design specifications. It includes the following items:

- Outputs expected from the system.
- Criteria for evaluating outputs.
- A volume of test data.
- Procedure for using test data.
- Personnel and training requirements.

SPECIFY CONDITIONS FOR USER ACCEPTANCE TESTING

Planning for user acceptance testing calls for the analyst and the user to agree on conditions for the test.

PREPARE TEST DATA FOR PROGRAM TESTING

As each program is coded, test data are prepared and documented to ensure that all aspects of the program are properly tested.

PREPARE TEST DATA FOR TRANSACTION PATH TESTING

This activity develops the data required for testing every condition and transactions to be introduced into the system. The path of each transaction from origin to destination is carefully tested reliable results.

PLAN USER TRAINING

User training is designed to prepare the user for testing and converting the system. User involvement and training take place parallel with programming for three reasons:

- The system group has time available to spend on training while the programs are being written.
- Initiating a user-training program gives the systems group a clearer image of the user's interest in the new system.
- A trained user participates more effectively in system testing.

The training plan is followed by preparation of the user training manual and other text materials.

COMPILE / ASSEMBLE PROGRAMS

All programs have to be compiled / assembled for testing.

PREPARE JOB PERFORMANCE AIDS

In this activity the materials to be used by personnel to run the system are specified and scheduled. This includes a display of materials.

PREPARE OPERATIONAL DOCUMENTS

During the test plan stage, all operational documents are finalized including copies of the operational formats required by the candidate system.

SYSTEMS TESTING

The computer department to ensure that the system functions as specified does this testing. This testing is important to ensure that a working system is handed over to the user for acceptance testing.

ACCEPTANCE TESTING

The user to ensure that the system functions, as the user actually wanted performs this testing. With prototyping techniques, this stage becomes very much a formality to check the accuracy and completeness of processing. The screen layouts and output should already have been tested during the prototyping phase.

An error in the program code can remain undetected indefinitely. To prevent this from happening the code was tested at various levels. To successfully test a system, each condition, and combinations of conditions had to be tested. Each program was tested and linked to other programs. This unit of program is tested and linked to other units and so on until the complete system has been tested.

The purpose of testing is to ensure that each program is fully tested. To do so a test plan had to be created. The test plan consists of a number of test runs such as the valid paths through the code, and the exception and error handling paths. For each test run there is a list of conditions tested, the test data used and the result expected. The test plan was then reviewed to check that each path through the code is tested correctly. It is the responsibility of the programmer to collect the data that will produce the required test condition.

6.3 TEST CASES

		TEST DATA Specifications for Travel and Tourism Management System user form1			
	Test Date	20/8/2012	Programmer name:		
	Tested By:		Project NAME:	Travel and Tourism Management System	
	Name	<input style="width: 150px; height: 20px;" type="text"/>		the fields are required.Can enter only letters, spaces, No numeric & special characters are allowed(Length upto 32 characters)	
	Address	<input style="width: 150px; height: 20px;" type="text"/>		the fields are required.Can enter only letters, numeric values,spaces, hyphens, and apostrophes, special characters are allowed(Length upto 32 characters)	
	Agency Id	<input style="width: 200px; height: 20px;" type="text"/>		the fields are required can enter only letters along with numeric values(length upto 8 characters)	
	PhoneNo	<input style="width: 240px; height: 20px;" type="text"/>		only 8 or 10 digit You may use numbers	
	Email Id	<input style="width: 120px; height: 20px;" type="text"/> @ site extension		the fields are required.Can enter only letters,numeric values,special characters. no spaces are allowed (Length upto 32 characters)	
	City	<input style="width: 220px; height: 20px;" type="text"/>		the fields are required.Can enter only letters and spaces are allowed (Length upto 10 characters)	
	State	<input style="width: 150px; height: 20px;" type="text"/>		the fields are required.Can enter only letters and spaces are allowed(Length upto 10 characters)	

POSITIVE TEST CASES FOR ADD AGENCY

T.C ID	PRE-CONDITION	T.C DESCRIPTION	T.C DATA	EXPECTED	ACTUAL	RESULT
1	User should be on https:// TRAVEL AND TOURISM MANAGEMENT SYSTEM MAnd is on Name Field	Check the functionality of Name field	Ravi	the fields are required.Can enter only letters, spaces, No numeric & special characters are allowed(Length upto 32 characters)	Ok	Pass
2	User should be on https://https:// TRAVEL AND TOURISM MANAGEMENT SYSTEM M And is on Name Field	Check the functionality of Name field	A 2	the fields are required.Can enter only letters, spaces, No numeric & special characters are allowed(Length upto 32 characters)	Ok	Pass
3	User should be on https://https:// TRAVEL AND TOURISM MANAGEMENT SYSTEM M And is on Name Field	Check the functionality of Name field	A Ravi	the fields are required.Can enter only letters, spaces, No numeric & special characters are allowed(Length upto 32 characters)	Ok	Pass
4	User should be on https://https:// TRAVEL AND TOURISM MANAGEMENT SYSTEM MAnd is on Name Field	Check the functionality of Name field	anuj	the fields are required.Can enter only letters, spaces, No numeric & special characters are allowed(Length upto 32 characters)	Ok	Pass
5	User should be on https:/ TRAVEL AND TOURISM MANAGEMENT SYSTEM MAnd is on Name Field	Check the functionality of Branch Name field	Ravi Kumar	the fields are required.Can enter only letters, spaces, No numeric & special characters are allowed(Length upto 32 characters)	Ok	Pass
6	User should be on https://EXPRESS DELIVERY,COURIER AND SHIPPING SERVICES SITE And is on Branch Address Field	Check the functionality of Branch Address field	Nangloi	the fields are required.Can enter only letters, numeric values,spaces, hyphens, and apostrophes, special characters are allowed(Length upto 32 characters)	Ok	Pass
7	User should be on https://MANAGEMENT And is on address Field	Check the functionality of Address field	477/7 Arya Mohalla	the fields are required.Can enter only letters, numeric values,spaces, hyphens, and apostrophes, special characters are allowed(Length upto 32 characters)	Ok	Pass

8	User should be on https://EXPRESS DELIVERY,COURIER AND SHIPPING SERVICES SITE And is on Branch Address Field	Check the functionality of Branch Address field	448/7	the fields are required.Can enter only letters, numeric values,spaces, hyphens, and apostrophes, special characters are allowed(Length upto 32 characters)	Ok	Pass
9	User should be on https:// TOUR & TRAVEL USING MANAGEMENT And is on Agency id Field	Check the functionality of Agency Id field	c-15	the fields are required can enter only letters along with numeric values(length upto 8 characters)	Ok	Pass
10	User should be on https://https:// TRAVEL AND TOURISM MANAGEMENT SYSTEM MAnd is on phoneno Field	Check the functionality of PhoneNO field	9899184526	Will accept only numeric upto 8 to 10 .	Ok	Pass
11	User should be on https://https:// TRAVEL AND TOURISM MANAGEMENT SYSTEM M And is on phoneno Field	Check the functionality of Phoneno field	9899456358	Will accept only numeric upto 8 to 10 .	Ok	Pass
12	User should be on https://https:// TRAVEL AND TOURISM MANAGEMENT SYSTEM M And is on Email Id Field	Check the functionality of Email Id field	Shaila.sharma@gmail.com	the fields are required.Can enter only letters,numeric values,special characters. no spaces are allowed (Length upto 32 characters)	Ok	Pass
13	User should be on https://https:// TRAVEL AND TOURISM MANAGEMENT SYSTEM MAnd is on Email Id Field	Check the functionality of Email Id field	neha_verma@gmail.com	the fields are required.Can enter only letters,numeric values,special characters. no spaces are allowed (Length upto 32 characters)	Ok	Pass
14	User should be on https://TRAVEL AND TOURISM MANAGEMENT SYSTEM M And is on City Field	Check the functionality of City field	New Delhi	the fields are required.Can enter only letters and spaces are allowed (Length upto 32 characters)	Ok	Pass
15	User should be on https://TRAVEL AND TOURISM MANAGEMENT SYSTEM MAnd is on state Field	Check the functionality of state field	NewDelhi	the fields are required.Can enter only letters and spaces are allowed (Length upto 32 characters)	Ok	Pass

16	User should be on https://TRAVEL AND TOURISM MANAGEMENT SYSTEM And is on Contact Person Field	Check the functionality of Contact person field	Deepak	the fields are required.Can enter only letters, spaces, No numeric & special characters are allowed(Length upto 32 characters)	Ok	Pass
17	User should be on https://TRAVEL AND TOURISM MANAGEMENT SYSTEM And is on Contact Person Field	Check the functionality of Contact person field	Amit Kumar	the fields are required.Can enter only letters, spaces, No numeric & special characters are allowed(Length upto 32 characters)	Ok	Pass

NEGATIVE TEST CASES FOR REGISTRATION FORM

T.C ID	PRE-CONDITION	T.C DESCRIPTION	T.C DATA	EXPECTED	ACTUAL	RESULT
1	User should be on https://TRAVEL AND TOURISM MANAGEMENT SYSTEM M And is on Agency Name Field	Check the functionality of Agency Name field	Ravinder Tour & Travel	the fields are required.Can enter only letters, spaces, No numeric & special characters are allowed(Length upto 32 characters)	Invalid	Fail
2	User should be on https://TRAVEL AND TOURISM MANAGEMENT SYSTEM M And is on Agency Name Field	Check the functionality of Agency Name field	Delhi Tour & travels123	the fields are required.Can enter only letters, spaces, No numeric & special characters are allowed(Length upto 32 characters)	Invalid	Fail
3						
4	User should be on https://TRAVEL AND TOURISM MANAGEMENT SYSTEM M And is on Agency Address Field	Check the functionality of Agency Address field	B-12/2 Pahar	the fields are required.Can enter only letters, numeric values,spaces, hyphens, and apostrophes, special characters are allowed(Length upto 32 characters)	Invalid	Fail
5						
	User should be on https://TRAVEL AND TOURISM MANAGEMENT SYSTEM M E And is on Agency Id Field	Check the functionality Agency Id field	5	the fields are required can enter only letters along with numeric values special characters are not allowed(length upto 8 characters)	Invalid	Fail
6						
7	User should be on https://TRAVEL AND TOURISM MANAGEMENT SYSTEM M And is on Phoneno Field	Check the functionality of Phoneno field	011-25487454	Will accept only numeric values upto 8 to 10 .	Invalid	Fail
8	User should be on https://TRAVEL AND TOURISM MANAGEMENT SYSTEM M And is on Phone Field	Check the functionality of PhoneNO field	9899171458	Will accept only numeric values upto 8 to 10 .	Invalid	Fail

9	User should be on https://TRAVEL AND TOURISM MANAGEMENT SYSTEM M And is on Email Id Field	Check the functionality of Email Idfield	<u>kuar48@gmil.c</u> <u>m</u>	the fields are required.Can enter only letters,numeric values,special characters. no spaces are allowed (Length upto 32 characters)	Invalid	Fail
	User should be on https://TRAVEL AND TOURISM MANAGEMENT SYSTEM M And is on Email Id Field	Check the functionality of Email Id field	singh @gail.com	the fields are required.Can enter only letters,numeric values,special characters. no spaces are allowed (Length upto 32 characters)	Invalid	Fail
10						
11	User should be on https:// TRAVEL AND TOURISM MANAGEMENT SYSTEM MAnd is on city Field	Check the functionality of city field	Kamla nagar	the fields are required.Can enter only letters ,no numeric values,hyphens no spaces and any other special charactersare allowed (Length upto 32 characters)	Invalid	Fail
12	User should be on https://TRAVEL AND TOURISM MANAGEMENT SYSTEM M And is on state Field	Check the functionality of state field	Prem Nagar	the fields are required.Can enter only letters ,no numeric values,hyphens no spaces and any other special charactersare allowed (Length upto 32 characters)	Invalid	Fail

6.4 VERIFICATION & VALIDATION

The objectives of verification, validity activities are to assess and improve the quality of the work products generated during development and modification of the software. Quality depends upon the various attributes like correctness, completeness, consistency, reliability, usefulness, usability, efficiency and conformance to standards.

The terms verification and validation are used synonymously. These are defined as under: -

Verification: “Are we building the product right?”

Validation: “Are we building the right product?”

Verification activities include proving, testing, and reviews. Validation is the process of evaluating software at the end of the software development to ensure compliance with the software requirements. Testing is a common method of validation. Clearly, for high reliability we need to perform both activities. Together, they are often called V&V activities.

The major V&V activities for software development are inspection, reviews, and testing (both static and dynamic). The V&V plan identifies the different V&V tasks for the different phases and specifies how these tasks contribute to the project V&V goals. The methods to be used for performing these V&V activities, the responsibilities and milestones for each of these activities, inputs and outputs for each V&V task, and criteria for evaluating the outputs are also specified.

CHAPTER 7

SYSTEM IMPLEMENTATION

The platform is the hardware and software combination that the Client/Server runs on. While hardware systems vary widely in features and capabilities, certain common features are needed for the operating system software.

7.1 HARDWARE SPECIFICATIONS

Hardware is a set of physical components, which performs the functions of applying appropriate, predefined instructions. In other words, one can say that electronic and mechanical parts of computer constitute hardware.

□ Processors and memory

The best system to start with is one based on Pentium II with a minimum 32 MB of RAM. Adequate performance requires at least 64 MB of RAM. But for a database server at least 64 to 128 MB of RAM is required.

□ Video displays

Earlier, the IBM-compatible computers had a simple text-only monochrome for the video display. Now, they use the advanced high-resolution color displays. For Client/Server systems one should have VGA or better video display.

In the following table TLA stands for the various types of adapters that can be used with IBM compatible PCs and the standard resolution for each one of them.

ADAPTER TYPE	TLA	STANDARD RESOLUTION
Monochrome Display Adapter	MDA	Text only (80 characters by 25 lines)
Color Graphics Adapter	CGA	640 × 200
Enhanced Graphics Adapter	EGA	640 × 350
Video Graphics Array	VGA	640 × 480

Super VGA	SVGA	800× 600 or 1024× 768
-----------	------	-----------------------

□ **Mouse**

A mouse is a must for the client software running under Windows OS or any other graphical environment.

□ **Keyboard**

Each client must have a 104 keys extended keyboard.

HARDWARE

Processor :Pentium 2.4 GHz or above

Memory : 256 MB RAM or above

Printer : Laser printer

Pen Drive : 5 GB

7.2 SOFTWARE REQUIREMENTS

The software is a set of procedures of coded information or a program which when fed into the computer hardware enables the computer to perform the various tasks. Software is like a current inside the wire, which cannot be seen but its effect can be felt.

Operating system : windows XP(professional)

Front-End Tool : JSP ,Servelets,Net Beans

Back-End : ORACLE

7.3 CONVERSION AND TRAINING

An important aspect of is to make sure that the new design is implemented to establish standards. The term implementation has different meanings, ranging form the conversion of a basic application to a complete replacement of a computer system. Implementation is

used here to mean the process of converting a new or revise system into an operational one. Conversion is one aspect of implementation. Conversion means changing form one system to another. The objective is to put the tested system into operation while holding costs, risks, and personnel irritation to a minimum. It involves creating computer-compatible files, training the operation staff, and installing terminal and hardware. A critical aspect of conversion is not disrupting the functioning of the organization.

7.4 TRAINING NEEDS

Training needs refer to the gaining of knowledge required for running the system.

First of all the system is a computer based system therefore the person should have good knowledge about computer and its working.

He should know how to use software's on the computer.

For a better usage and working of the software the organization should appoint a person who has good knowledge of all the required software. The organization gets a person trained through different institutes present in the market. The training should be as per the above requirements.

CHAPTER 8

COST ESTIMATION OF THE PROJECT

Cost in a project is due to the requirements for software, hardware, and human resources. Hardware resources are computer time, terminal time and memory required for the project. Software resources include the tools and compilers needed during development. The bulk of cost of software development is due to human resources needed. Cost estimates are determined in terms of person-months (PM).

Total No. Of Persons Involved In This Project:

1. Administrator
2. Senior Programmer
3. Junior Programmers
4. On line Users.

Since this Project will complete in 4 months

COST ESTIMATE: (Salary of Project Manager + Salary of Senior Programmer + 2 * Salary of Junior Programmer) * 2

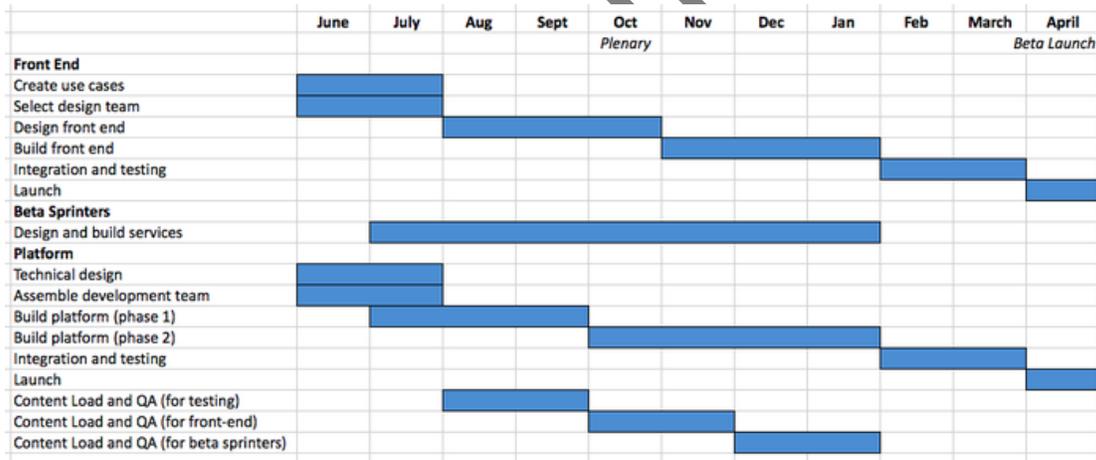
CHAPTER 9

GANTT & PERT CHART

GANTT CHART

Gantt charts mainly used to allocate resources to activities. The resources allocated to activities include staff, hardware, and software. Gantt charts (named after its developer Henry Gantt) are useful for resource planning. A Gantt chart is special type of bar chart where each bar represents an activity. The bars are drawn along a timeline. The length of each bar is proportional to the duration of the time planned for the corresponding activity.

Gantt chart is a project scheduling technique. Progress can be represented easily in a Gantt chart, by coloring each milestone when completed. The project will start in the month of December and end after 4 months at the end of March.



P1

Pert Chart

PERT (Project Evaluation and Review Technique) charts consist of a network of boxes and arrows. The boxes represent activities and the arrows represent task dependencies.

PERT chart represents the statistical variations in the project estimates assuming a normal distribution. Thus in a PERT chart instead of making a single estimate for each task, *pessimistic*, *likely*, and *optimistic* estimates are also made. The boxes of PERT charts are usually annotated with the pessimistic, likely, and optimistic estimates for every task. Since all possible completion times between the minimum and maximum durations for every task have to be considered, there are many critical paths, depending on the permutations of the estimates for each task. This makes critical path analysis in PERT charts very complex. A critical path in a PERT chart is shown by using thicker arrows. The PERT chart representation of the **TRAVEL AND TOURISM MANAGEMENT SYSTEM**.

problem of Figure A. is shown in Figure B.

Task	ES	EF	LS	LF	ST
Specification Part	0	15	0	15	0
Design Database Part	15	60	15	60	0
Design GUI Part	15	45	90	120	75
Code Database Part	60	165	60	165	0
Code GUI Part	45	90	120	165	75
Integrate and Test	165	285	165	285	0
Write User Manual	15	75	225	285	210

Figure A : Different Tasks for the **TRAVEL AND TOURISM MANAGEMENT SYSTEM** are shown in above table.

PERT (Program Evaluation and Review Technique) CHART

PERT CHART

Researching and developing a method to assist children in clearly understanding the steps and the methods of division.

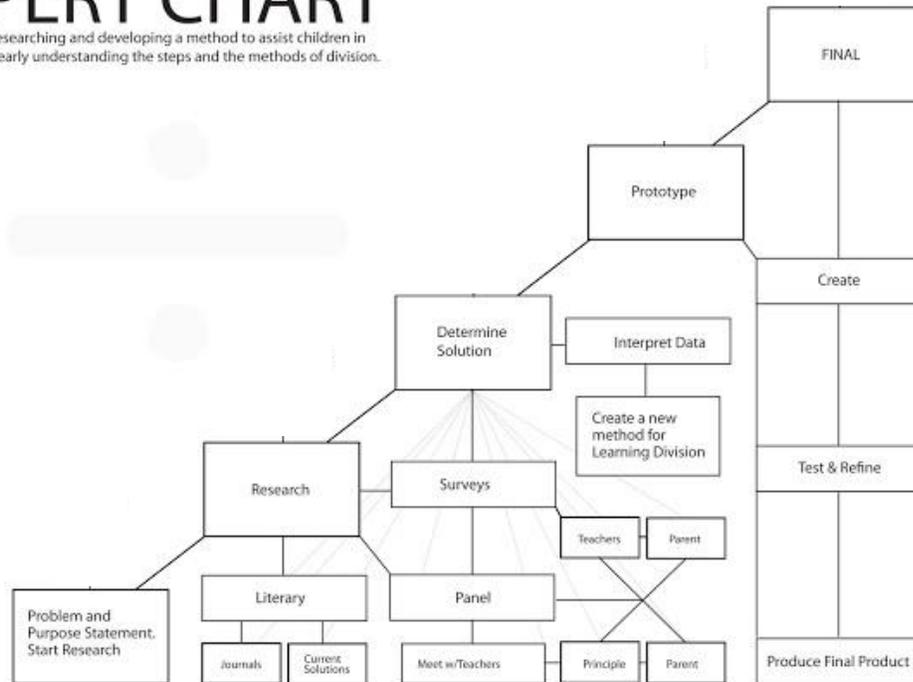


FIGURE B: PERT chart representation of the **TRAVEL AND TOURISM MANAGEMENT SYSTEM**

PERT charts are a more sophisticated form of activity chart. In activity diagrams only the estimated task durations are represented. Since the actual durations might vary from the estimated durations, the utility of the activity diagrams is limited.

CHAPTER 10

SCOPE OF FUTURE APPLICATION

Scope of Future Application

This project can be used in the travel agencies . This site is about various travel agencies,their packages and information about the routes for different places in India.

Utmost care and back-up procedures must be established to ensure 100% successful implementation of the online system. In case of system failure, the organization should be in a position to process the transaction with another organization or if the worst comes to the worst, it should be in a position to complete it manually.

Scope of Improvement

Utmost care and back-up procedures must be established to ensure 100% successful implementation of the computerized system. In case of system failure, the organization should be in a position to process the transaction with another organization or if the worst comes to the worst, it should be in a position to complete it manually. This project can be used in the Travel Mastii Pvt. Ltd. with adding some more useful modules in the project like account of the all regular tourists then it will be power full solution.

Utmost care and back-up procedures must be established to ensure 100% successful implementation of the computerized Travel Agency activities. In case of system failure, the organization should be in a position to process the transaction with another organization or if the worst comes to the worst, it should be in a position to complete it manually.

Research Methodology

This project is designed to meet the requirements of Traveling Agencies. It has been developed in JSP, keeping in mind the specifications of the system.

For designing the system we have used simple data flow diagrams.

Overall the project teaches us the essential skills like:

CHAPTER 11

CONCLUSION

This project is designed to meet the requirements of an **Travel and Tourism Management System** system. It has been developed in JSP and the database has been built in DREAMWEAVER and ORACLE, keeping in mind the specifications of the system.

For designing the system we have used simple data flow diagrams.

Overall the project teaches us the essential skills like:

- ❑ Using system analysis and design techniques like data flow diagram in designing the system.
- ❑ Understanding the database handling and query processing using Access.

CHAPTER 12

BIBLIOGRAPHY

- | | |
|------------------------------------|----------------------|
| ➤ Java Programming | E.Balagurusamy |
| ➤ Java Projects Selly | Tony Martin, Dominic |
| ➤ Java Professional project Bembey | Kuljeet kaur, Pooja |
| ➤ MY SQL The Complete Reference | George Koch |
| ➤ A Complete Reference For Oracle | Tata Mack grow Hill |

References

- www.travelagentsofindia.com
- www.traveliteindia.com
- www.transindiaholidays.com

CHAPTER 12

GLOSSARY

Algorithm: A step-by-step set of instructions that describe the instructions written in a programming language like C, BASIC, JSP etc.

Alignment: To a hardware engineer this means the disk drive head's ability to read and write information without error, but for a data operator this means the text's relationship to the left and right margins, as in "centered", "left-justified" and so forth.

Alphanumeric characters: Alpha stand for the alphabet and numeric stands for the number. Together they are characters that consist of letters and numbers.

API: Acronym for application program interface. API defines a standard way that programs work with pull-down menus, dialog boxes, and windows.

ASCII: A standard that stands for American standard code for information interchange. ASCII defines a standard way for representing characters on computers.

CASE: CASE is an acronym for computer-aided software engineering. It is a method of programming that reduces program correction, changes, etc. It is very useful when developing large projects where the number of people involved are many, and control is very important.

Client: A local area network term that describe a computer that can request information, such as an application, from a file server, and can also perform independently from the network server by using a client application.

Client application: A program that only works on the computer that actually has the program on its hard disk. Because client application only run on one computer and can not be accessed by other computers in the network, a failure in one part of the network doesn't affect a client application.

Client/server Network: A network where some programs and files are shared in one big computer, but each computer connected to the network can also run on its own.