

# **VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

Jnana Sangama, Belgavi-590018, Karnataka, INDIA



## **MINI PROJECT REPORT**

**On**

### **“Airline Reservation System”**

Submitted in partial fulfillment of the requirements for the V Semester

**Bachelor of Engineering**

**In**

**INFORMATION SCIENCE AND ENGINEERING**

**For the Academic year**

**2018-2019**

**BY**

**Manish Kumar**

**1PE16IS053**

**Karan Kumbhani**

**1PE16IS044**

**Under The Guidance Of**

**Rashma B. M.**

**Assistant Professor**

**Dept of Information Science & Engineering, PESIT-BSC.**

**Suchitra Charles**

**Assistant Professor**

**Dept of Information Science & Engineering, PESIT-BSC.**



**Department of Information Science and Engineering**

**PESIT Bangalore South Campus**

**Hosur Road, Bengaluru-560100**

**PESIT Bangalore South Campus**  
Hosur Road, Bengaluru-560100  
Department of Information Science and Engineering



**CERTIFICATE**

*This is to certify that the project work entitled "**Airline Reservation System**" is a bonafide work carried out by **Manish Kumar, Karan Kumbhani** and bearing USNs **1PE16IS053, 1PE16IS044** respectively in partial fulfilment of DBMS mini project (15CSL58) in 5<sup>th</sup> semester of Degree of Bachelors (**Bachelors of Engineering**) in **Information Science and Engineering** of **Visvesvaraya Technological University, Belagavi** during the year 2018-2019.*

*It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the Report. The project report has been approved as it satisfies the academic requirements in respect of mini project work prescribed for said*

Signatures:

\_\_\_\_\_  
Project Guide  
Asst./Associate Professor,  
Dept. of ISE  
PESIT-BSC, Bengaluru

\_\_\_\_\_  
Head Dept of ISE  
**Dr. Annapurna D**  
PESIT-BSC, Bengaluru

External Viva

Name of the Examiners

Signature with date

1.

2.

\_\_\_\_\_  
\_\_\_\_\_

## **ACKNOWLEDGEMENT**

We would like to thank our Lab Teachers, Mrs. Rashma B. M. and Mrs. Suchitra Charles for their aid and evaluation over the course of completion of this project. We would also like to thank our University, Visvesvaraya Technological University for the opportunity to make this project which has given us the chance to learn about these topics. Additionally, we would like to thank our college, PES Institute of Technology for allowing us to use the college resources. Finally, I would like to thank our friends and family for their ceaseless and tireless support throughout this project.

# CONTENTS

ABSTRACT.....	III
LIST OF FIGURES.....	IV
LIST OF TABLES.....	V
1. Chapter 1: Introduction.....	1
1.1. Introduction.....	1
1.1.1. Purpose of the project .....	1
1.1.2. Scope.....	1
1.1.3. Definitions, acronyms and abbreviations.....	1
1.2. Literature survey.....	2
1.3. Existing System.....	3
1.4. Proposed System.....	3
1.5.Statement of the Problem.....	4
2. Chapter 2: Software Requirement Specifications.....	5
2.1. Software Requirements Specifications.....	5
2.2. Operating Environment.....	5
2.2.1. Hardware Requirements.....	5
2.2.2. Software Requirements.....	6
2.3.Functional Requirements.....	6
2.4. Non functional requirements.....	6
2.5. User characteristics.....	7
2.6. Advantages of Airline Reservation System.....	7
2.7. Summary.....	8
3. Chapter 3: Design.....	9
3.1.ER diagram.....	9
3.2.Schema diagram.....	10
3.3.UML diagram .....	10
3.4.Data Flow Diagrams .....	11

4. Chapter 4: Implementation.....	13
4.1. Implementation.....	13
4.2. Programming Language Selection.....	13
4.3. Query for Creation of Tables.....	13
4.4. Description of tables.....	15
4.5. Triggers used.....	16
4.6. Procedure used.....	17
4.7. Important Codes.....	17
5 Chapter 5: Results snaps.....	20
6 Chapter 6: Conclusion. ....	23
6.1. Conclusion.....	23
6.2. Limitations of the Project.....	23
6.3. Future Enhancement (if any).....	23
6.4. References.....	24

## ABSTRACT

Our Project ‘Airline reservation System’ is a computerized system used to store and retrieve information and conduct transactions related to air travel. The aim the project is to expose the relevance and importance of Airline Reservation Systems.

The system allows the airline passenger to search for flights that are available between the two travel cities, namely the “Departure city” and “Arrival city” for a particular departure date. The system is designed such that flights are available on all days. The system displays all the flight’s details such as flight no, name, price etc.

Then the system checks for the availability of seats on the flight. If the seats are available then the system allows the passenger to book a seat. Otherwise it asks the user to choose another flight.

The system asks the customer to enter his details such as name, age, email and contact number to book a flight. The system also allows the customer to cancel his/her reservation, if any problem occurs.

The main purpose of this software is to reduce the manual errors involved in the airline reservation process and make it convenient for the customers to book the flights as and when they require. The software allows customer to make reservations, modify reservations or cancel a particular reservation.

## LIST OF FIGURES

Fig 3.4.1 E R Diagram

Fig 3.4.2 Schema Diagram

Fig 3.4.3 Use Case Diagram

Fig 3.4.4 Admin Data Flow Diagram

Fig 3.4.5 Customer Data Flow Diagram

Fig 4.1 Enter

Fig 4.2 Login

Fig 4.3 Page 1

Fig 4.4 Page 2

Fig 5.1 Admin Login Page

Fig 5.2 Admin Add Flights Page

Fig 5.3 Admin View Records Page

Fig 5.4 Customer Sign up Page

Fig 5.5 Customer Login Page

Fig 5.6 Customer Choose Source and Destination Page

Fig 5.7 Customer Choose Flight Page

Fig 5.8 Customer View Booking Page

## LIST OF TABLES

4.4.1. Admin

4.4.2. Payment

4.4.3. Planes

4.4.4. Aircrafts

4.4.5. Customer

4.4.6. Records



# Chapter 1

## Introduction

### 1.1. Introduction

Considering the volumes of data that needs to be tracked and accessed, it would be very difficult to manage the accuracy and quality of data manually and deliver them accordingly. It would be almost impossible to get the details required in case of manual maintenance of data. The TWC (Travel With Comfort) is an innovative solution that helps in managing huge loads of flight orders. The Airline Reservation System simplifies the manual work and allows smooth administration of the operations of transportation.

#### 1.1.1. Purpose

This project is aimed to reduce the manual work involved in data maintenance in the Flight Booking and automates the Airline Reservation System. This project is developed mainly to simplify the manual work and allows smooth administration of the operations of airlines. The purpose of the project is to computerize the administrative operations of a Flight Booking and to develop software which is user friendly, simple, fast, and cost – effective. It deals with the collection of Users, Employees, Flights and Booking information, Fare details, etc. Traditionally, it was done manually. The main function of the system is to enter and book Flights and retrieve these details as and when required, and also to manipulate these details meaningfully.

#### 1.1.2. Scope

The project provides a very simple application which simplifies the manual work done by the operation team of Airline Reservation System. This application saves the data of employees and users in the database. Allows users to search for flights, book or cancel the existing flights. Our project allows users to view the booked flights data stored in the database and to see the statistics.

#### 1.1.3. Definitions, acronyms and abbreviations

**Personal Details:** Details of passengers such as user id, phone number, address, passport no, e-mail address etc.

**Contact Details:** Details of contact associated with the passenger.

SRS: System Requirement Specification

WWW: World Wide Web

MySQL: is a RDBMS based on SQL which is used for adding, removing, and modifying information in the database.

RDBMS: Relational Database Management System

HTML: Hypertext Markup Language

PHP: Hypertext Preprocessor

CSS: Cascading Style Sheet

HTTP: Hypertext Transfer Protocol

## **1.2. Literature Survey**

Airline computerized reservation systems are in the primary form of travel computerizations in the world. These systems manage the millions of business transactions. The Computer Reservation System function as extremely powerful and valuable distribution and marketing tools for their airline owners. Today's travel agency competitive environment is largely defined and controlled by airline Computer Reservation System (CRS).

The electronic age reservation were managed manually using records book and manifest and other printed media, many reservation system in today's world are still met in this way. Airline Reservation System (ARS) that Airline Reservation System used to be a standalone system whereby every airline had it's own system disconnected from other Airline Reservation system or ticketing agents and makes uses of only a designated numbers of airline employees.

During the 1970's Travel Agency pushed for the access to the Airline systems and also that today's air travel information is linked, stored and retrieved by a network of computer Reservation systems (CRS), Which are accessible by multiple airlines and travel agents.

An American Airline was the first to establish an automated booking system in the year 1946 using a system to track information and improve efficiency. This automated booking system endured years of development and alterations.

The Next booking systems were developed in the year 1953 by a Trans-Canada airline, this Trans-Canada Airlines developed a computer based system with remote terminals that eventually took over operations in 1953. In the same year that the Trans Canada company developed their computer based system, the American Airlines worked closely with IBM to develop an improved system and the Airline Reservation System (ARS) and the semi- Automated Business Research System ( SABRE), which was launched thereafter in 1960. The development ARS was completed in 1964 and has been recognized as the largest data processing system in existence.

### **1.3. Existing System**

In few countries if a person wants to book a flight ticket, he use to follow one of these things: Manually goes to the Airport and book his ticket, Downloading the ticket form as paper document, filling it manually and submitting it at Airport. Fill the Ticket form on system and get the print out as paper documents to submit it at Airport. Booking the Ticket at some particular registered ticket counters in online.

Even above approaches make a ticket booking online, it was not completely done on online. Passenger may not have much freedom over this approach. Hence the Passenger may or may not be satisfied with this approach as it includes manual intervention like travelling to Airport for booking his ticket, cannot upload and download the latest updates. No use of Web services and remoting. Risk of mismanagement and of data when the project is under development, less Security. No proper coordination between different applications and Users and fewer users.

### **1.4. Proposed System**

The Proposed system ensures the complete freedom for users, where user at his own system can logon to this website and can book his ticket. Our proposed system allows only registered users to book the tickets, view timings and cancel their tickets. In this Proposal the entire work is done on online and ticket with id is also provided for passengers as a print document. Here passengers can send their queries and suggestions through a feedback form. To debug the existing system, remove procedures those cause data redundancy, make navigational sequence proper To build strong password mechanism. Advantages are User friendliness provided in the application with various

controls, the system makes the overall project management much easier and flexible and it provides high level of security with different level of authentication.

### **1.5. Statement of the Problem**

The users can easily purchase an e-ticket by going to the ticket sale website, searching and selecting the destination, entering the details such as name, way of travel, luggage information and dates and finally making the payment via bank cards, bank transfer or through online payment companies. The e-ticket is then emailed or texted on the telephone of the customer. While previously travel agents and airlines assisted customers in making ticket purchases, today with the improved internet system, it is getting easier and easier to book the flights on your own. After the customer makes the purchase, the electronic record and the details of the ticket are saved into airline's database. The database is integrated with the passenger service system, which is then connected to the airports, airlines, travel agencies for sharing real time information.

## Chapter 2

### Software Requirements Specification

#### 2.1. Software Requirements Specification

The Airline Reservation System operates with a client-server architecture, and as such, must have minimum hardware and software to run the server/browser along with all its dependencies.

The system is used by Admins who can add or remove an aircraft or manipulates data using a computer with a HTML compatible browser. The system is also used by the customers of the Airline who can book tickets based on their preference using a computer at his/her home, with a HTML/php compatible browser.

The server software runs in a dedicated centralised server hosting center for the Airline database. The scripts and http server run on the server, and require a Php interpreter, along with the dependencies for the scripts, as well as the MySQL server.

#### 2.2. Operating Environment

The production ready software is meant to run on a variety of verified hardware and software. As such, many of the required dependencies are available cross platform, both for the front end as well as the backend. Some of the verified software and hardware are specified below, along with software and hardware that are supposed to be compatible.

##### 2.2.1. Hardware Requirements

The Hardware requirements are very minimal and the program can be run on most of the machines.

- Processor - Intel 486/Pentium processor or better
- Processor Speed - 500 MHz or above
- Hard Disk - 20GB(approx)
- RAM - 64MB or above

- Storage Space - Approx. 2MB

### 2.2.2 Software Requirements

- Technology Implemented : Apache Server, MySQL Server
- Language Used : PHP
- Database : My SQL
- User Interface Design : HTML, CSS
- Web Browser : Google Chrome

### 2.3. Functional Requirements

The functional requirements of this project are:

- The project allows login of two entities, Admin and Customer:
- The Admin has a set of functionalities described as:
  - > He can login into the system.
  - > He can log out from his current account.
  - > He can add/delete aircrafts in the database.
  - > He can view all the records of all the bookings that have been made.
- The Customer has a set of functionalities described as:
  - > He can sign up in the system using his credentials.
  - > He can login in to the system using his username and password.
  - > He can check the flights based on the source and destination.
  - > He can book a flight.
  - > He can log out from his current account.
  - > He can view all his bookings.

### 2.4. Non-functional requirements

The non-functional requirements of this projects are:

- Security :

The website does not allow access to any functionality by directly jumping to any particular link to that function's page. Additionally, anything that is needed to be done can only be done by first logging in. There are multiple accounts, with login by Admin only allowing Admin functionalities and Customer allowing only Customer functionalities.

- Data Integrity :

The project does not allow entry of data in case data is invalid. This is very important as if invalid data is added, then it can cause large problems, such as getting tickets for flights which are invalid, creating aircrafts for invalid or incorrect destinations, or inserting invalid source/destination, each of which can have cascading effects.

- Automatic data processing :

a lot of information is processed by the project instead of relying on the user to add perfect information and perform numerous functions each time. Examples include deleting old dates and flights, retrieving prices, validating inserted information in aircrafts etc. This is an important task as it can be performed much more efficiently and quickly by the system than by a human.

## **2.5. User characteristics**

The user characteristics are:

- The first user is the admin, who holds administrative powers in the context of the Project, which include:

- > adding/deleting aircrafts in the database.
- > viewing all the records of all the bookings that have been made.

- The second user is the customer, who can book tickets to a flight sitting at home or from any place, his functions include:

- > checking the flights based on the source and destination.
- > booking a flight.
- > logging out from his current account.
- > viewing all his bookings

## **2.6. Advantages of Airline Reservation System**

The advantages of the Airline Reservation System are:

- It is based on a Client-Server System, meaning multiple admins and customers can be supported.

- It operates on a minimalistic User Interface so that any user of the system can do what they need to do with almost no training and extreme ease.
- Erroneous data is not entered into the system and rendered invalid
- Old aircrafts are removed automatically without user intervention, meaning old data does not clog up the system.
- All IDs are attached to items automatically after being randomly generated, so as to prevent mismatches and they operate solely on the back-end, meaning that users only need to be concerned with human-useable data.

## **2.7. Summary**

We have developed a highly comprehensive and easy to use system for any small Airline. It is easy to implement and requires no training to use. It provides options for admins and customers. It is error-proof and does large amount of work in the background. Thus, the system aids to simplify the processes used by cashier and manager as well as reduce operational costs, the primary concerns of all businesses.



## Chapter 3

### Design

#### 3.1. E R Diagram

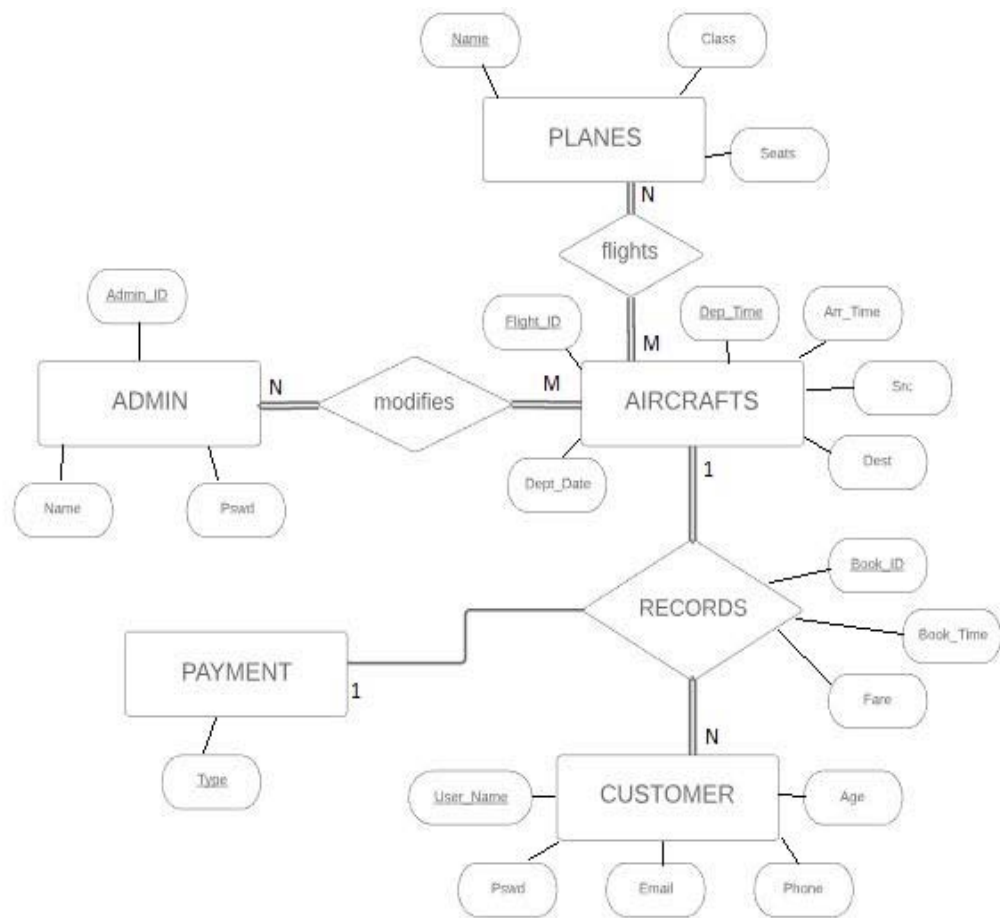


Fig 3.1 E R Diagram 1

### 3.2. Schema Diagram

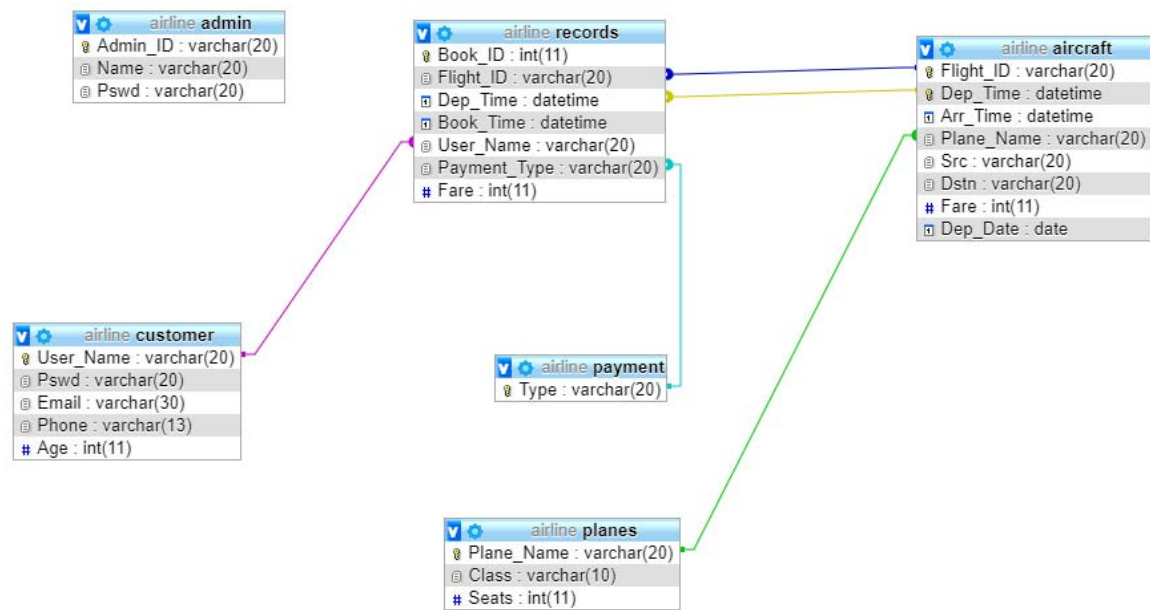


Fig 3.2 Schema Diagram 1

### 3.3. UML Diagram

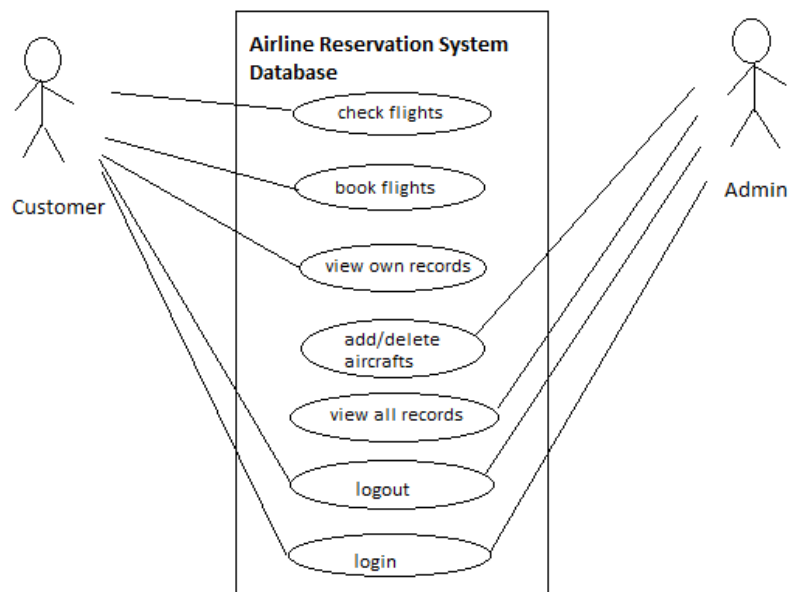


Fig 3.3 Use Case Diagram 1

### 3.4. Data Flow Diagrams

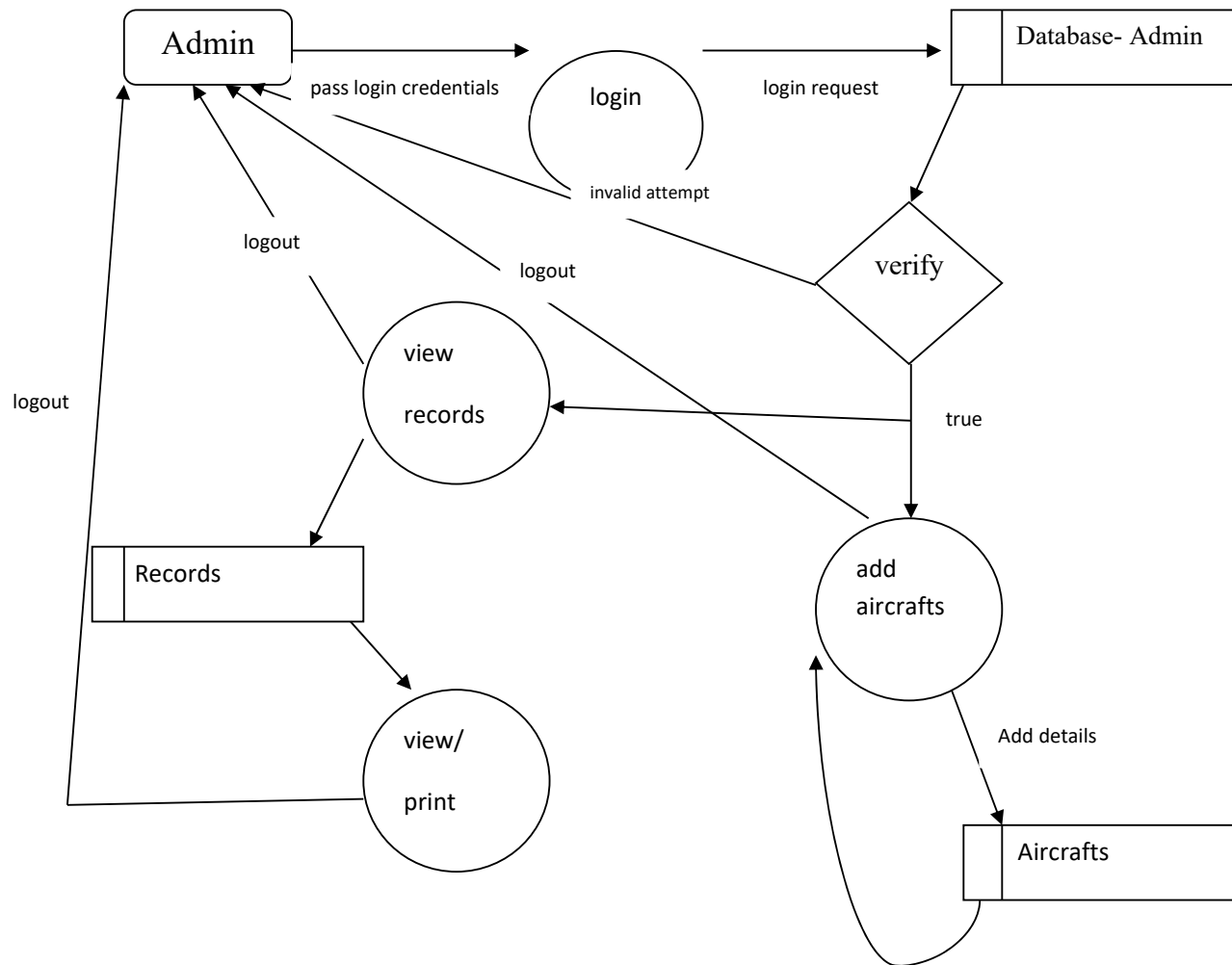


Fig 3.4 Admin data flow Diagram

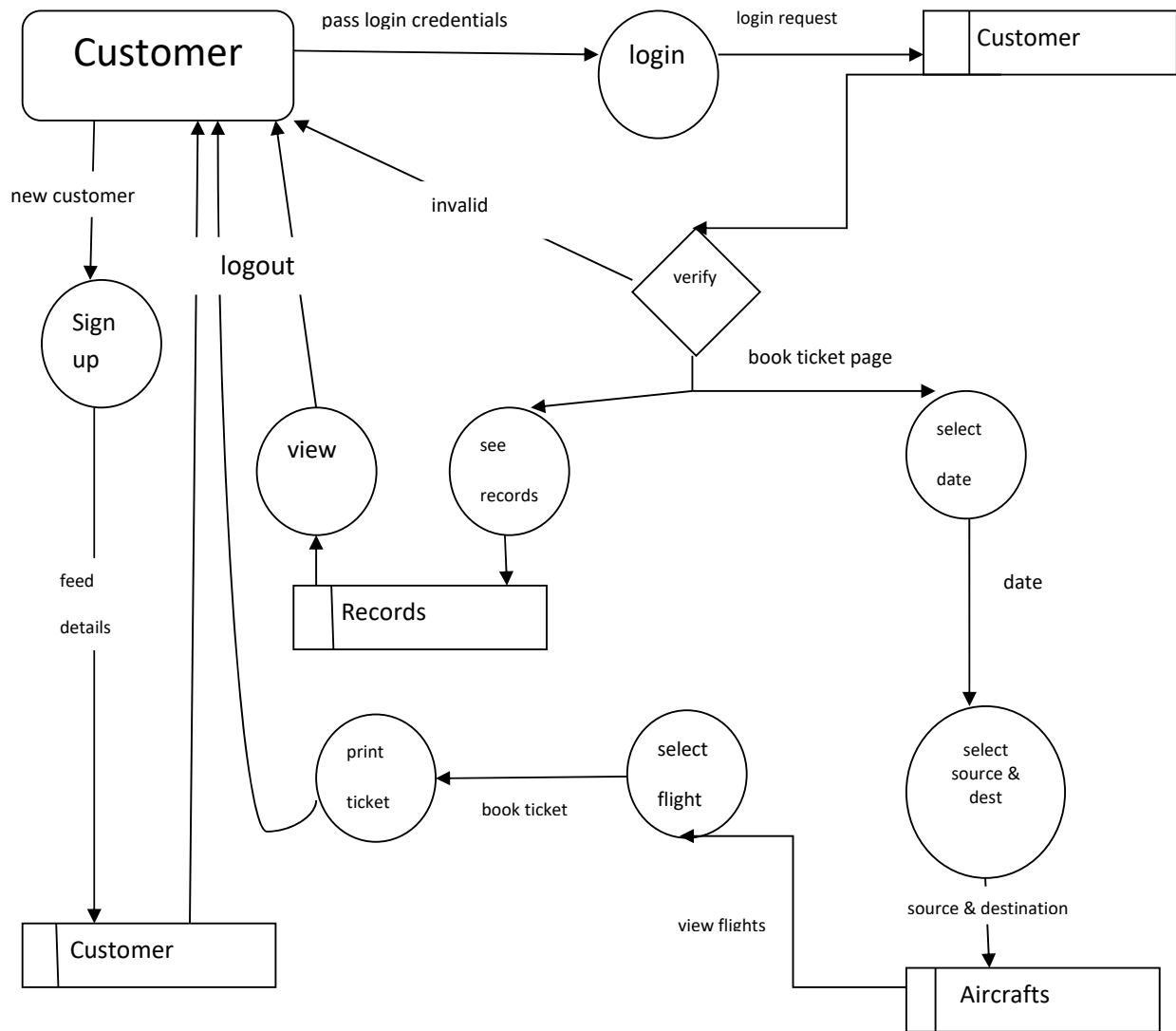


Fig 3.5 Customer data flow diagram

## Chapter 4

### Implementation

#### 4.1. Implementation

The project is implemented in HTML, Php and CSS for Front-End (website) and MySQL for Back-End (database). It has a three-tier architecture with Front-End forming Application Layer and Back-End forming Middle Layer and Database. Input validation is done in Php. Php communicates with the local server. The server responds with HTML code that is then displayed on the Front-End. The server communicates with MySQL using mysqlconnector.

#### 4.2. Programming Language Selection

Php was used to communicate with MySQL because PHP is the most popular scripting language for web development. It is free, open source and server-side (the code is executed on the server). MySQL is a Relational Database Management System (RDBMS) that uses Structured Query Language (SQL). It is also free and open source. The combination of PHP and MySQL gives unmet options to create just about any kind of website - from small contact form to large corporate portal.

HTML and CSS is used for the front-end as it versatile and easy to use.

#### 4.3. Queries For Creation of Tables

- create database Airline;
- create table Admin(  
Admin\_ID varchar(20) primary key,  
Name varchar(20),  
Pswd varchar(20));

- create table Customer(  
User\_Name varchar(20) primary key,  
Pswd varchar(20),  
Email varchar(30),  
Phone varchar(13),  
Age int);
- create table Planes(  
Plane\_Name varchar(20) primary key,  
Class varchar(10),  
Seats int);
- create table Aircraft(  
Flight\_ID varchar(20),  
Dep\_Time DateTime,  
Arr\_Time DateTime,  
Plane\_Name varchar(20),  
foreign key(Plane\_Name) references Planes(Plane\_Name) on update cascade on delete cascade,  
Src varchar(20),  
Dstn varchar(20),  
Fare int,  
Primary key (Flight\_ID, Dep\_Time) );
- create table Payment(  
Type varchar(20) primary key);
- create table Records(  
Book\_ID varchar(20) primary key,  
Flight\_ID varchar(20),  
foreign key(Flight\_ID) references Aircraft(Flight\_ID) on update cascade,  
Book\_Time DateTime,  
User\_Name varchar(20),

foreign key(User\_Name) references Customer(User\_Name) on delete cascade on update cascade,

Payment\_Type varchar(20),

foreign key(Payment\_Type) references Payment(Type) on update cascade);

#### 4.4. Description of tables

The tables used in this project are:

- 4.4.1. Admin** : which stores information about the Airline Administrators who can make changes to the schedule and so on.

Field	Type	Null	Key	Default	Extra
Admin_ID	varchar(20)	NO	PRI	NULL	
Name	varchar(20)	YES		NULL	
Pswd	varchar(20)	YES		NULL	

- 4.4.2. Payment** : which store information about the type of payment method used.

Field	Type	Null	Key	Default	Extra
Type	varchar(20)	NO	PRI	NULL	

- 4.4.3. Planes** : which stores information about all the planes that the airline owns which ply between different cities under different aircraft\_id.

Field	Type	Null	Key	Default	Extra
Plane_Name	varchar(20)	NO	PRI	NULL	
Class	varchar(10)	YES		NULL	
Seats	int(11)	YES		NULL	

- 4.4.4. Aircrafts** : The planes when fly between one set of cities it's aircraft\_id varies on it's to and fro journey.

Field	Type	Null	Key	Default	Extra
Flight_ID	varchar(20)	NO	PRI	NULL	
Dep_Time	datetime	NO	PRI	NULL	
Arr_Time	datetime	YES		NULL	
Plane_Name	varchar(20)	YES	MUL	NULL	
Src	varchar(20)	YES		NULL	
Dstn	varchar(20)	YES		NULL	
Fare	int(11)	YES		NULL	
Dep_Date	date	YES		NULL	

**4.4.5. Customer :** which stores all the information about the customer of the airline company.

Field	Type	Null	Key	Default	Extra
User_Name	varchar(20)	NO	PRI	NULL	
Pswd	varchar(20)	YES		NULL	
Email	varchar(30)	YES		NULL	
Phone	varchar(13)	YES		NULL	
Age	int(11)	YES		NULL	

**4.4.6. Records :** this table contains all the information about all the bookings made by the customers for individual aircrafts.

Field	Type	Null	Key	Default	Extra
Book_ID	int(11)	NO	PRI	NULL	auto_increment
Flight_ID	varchar(20)	YES	MUL	NULL	
Dep_Time	datetime	YES		NULL	
Book_Time	datetime	YES		NULL	
User_Name	varchar(20)	YES	MUL	NULL	
Payment_Type	varchar(20)	YES	MUL	NULL	
Fare	int(11)	YES		NULL	

#### 4.5. Trigger used

```

DELIMITER $$
CREATE TRIGGER Before_Insert_User
BEFORE INSERT ON customer
FOR EACH ROW
BEGIN
    IF (EXISTS(SELECT 1 FROM customer WHERE User_Name =
NEW.User_Name)) THEN
        SIGNAL SQLSTATE VALUE '45000'; SET
MESSAGE_TEXT = 'INSERT failed due to duplicate
mobile number';
    END IF;
END$$
DELIMITER ;

```

The purpose of the trigger is to prevent a existing customer from creating new duplicate account by checking their mobile numbers entered while sign up.



## 4.6. Procedures used

```

DELIMITER //
CREATE PROCEDURE insert_cust (IN name varchar(20),IN pass
varchar(20),IN email varchar(30) ,IN phone varchar(13), ,IN age int)
BEGIN
    insert into Customer(User_Name,Pswd,Email,Phone,Age)
    values (name,pass,email,phone,age);
END//
DELIMITER;

```

If the Customer is not already registered a new account for that Customer is created with the help of this procedure.

## 4.7. Important Codes

### 4.7.1. Enter page(php)

```

81 <?php
82 if(isset($_POST["insert"])){
83 if(!empty($_POST['fid']) && !empty($_POST['planename']) && !empty($_POST['from']) && !empty($_POST['to'])
84 && !empty($_POST['deptime']) && !empty($_POST['fare']) && !empty($_POST['arrtime'])&& !empty($_POST['deptime'])) {
85     $fid=$_POST['fid'];
86     //seats=$_POST['seats'];
87     $planename=$_POST['planename'];
88     $from=$_POST['from'];
89     $to=$_POST['to'];
90     $deptime=$_POST['deptime'];
91     $arrtime=$_POST['arrtime'];
92     $fare=$_POST['fare'];
93     $deptime=$_POST['deptime'];
94     $con=mysqli_connect('localhost','root','','airline') or die(mysql_error());
95
96     $sql="SELECT * FROM aircraft WHERE Dep_Time='$deptime' AND Flight_ID='$planename'";
97     $query=mysqli_query($con,$sql);
98     $numrows=mysqli_num_rows($query);
99     if($numrows==0)
100     {
101         // $sql="INSERT INTO custlogin(username,idproof,age,email,contactnum,password)
102         //VALUES('$user','$id','$age','$email','$cnum','$pass')";
103     $sql="INSERT INTO aircraft(Flight_ID,Dep_Time,Arr_Time,Plane_Name,Src,Dstn,Fare,Dep_Date)
104     VALUES('$fid','$deptime','$arrtime','$planename','$from','$to','$fare','$deptime')";
105
106     $result=mysqli_query($con,$sql);
107     // $lastInsertID = mysqli_insert_id($con);
108     if($result){
109         echo " Successfully Inserted.. ";
110     } else {
111         echo "Failure!";
112     }
113
114     } else {
115         echo "That entry(or Pid) already exists! Please try again with another.";
116     }
117

```

Figure 4.1 Enter page

### 4.7.2. User login(php)

```
22 </html>
23 <?php
24 $con=mysqli_connect("localhost","root","","airline");
25 //mysql_select_db("Practice")
26 //if (isset($_POST['email']))
27
28 if(isset($_POST['submit']))
29 {
30     $email = $_POST['email'];
31     $pass = $_POST['pass'];
32
33     $sql="SELECT * FROM customer WHERE User_Name='".$email."' AND Pswd='".$pass."'";
34     $query=mysqli_query($con,$sql);
35     $numrows=mysqli_num_rows($query);
36     if($numrows!=0)
37     {
38         while($row=mysqli_fetch_assoc($query))
39         {
40             $dbename=$row['User_Name'];
41             $dbpassword=$row['Pswd'];
42         }
43
44         if($email == $dbename && $pass == $dbpassword)
45         {
46             session_start();
47             $_SESSION['sess_user']=$email;
48
49             /* Redirect browser */
50             header("Location: page1.php");
51         }
52     }
53     else
54     {
55         echo "Invalid username or password!";
56     }
57 }
58 //<https://www.youtube.com/watch?v=yLFLVBbB9AM>
59 ?>
60
```

Figure 4.2 User login

### 4.7.3. Page 1 Choose date, time, source and destination(php)

```

70 <?php
71
72 if(isset($_POST["proceed"])){
73 if(!empty($_POST['from1']) && !empty($_POST['to1']) && !empty($_POST['depdate'])) {
74     $from=$_POST['from1'];
75     $to=$_POST['to1'];
76     $depdate=$_POST['depdate'];
77
78     //$var = '20/04/2012';
79     $date = str_replace('/', '-', $depdate);
80     $depdate= date('Y-m-d', strtotime($date));
81
82     $con=@mysqli_connect('localhost','root','','airline') or die(mysql_error());
83     $user=$_SESSION["sess_user"];
84     $today = strtotime('today');
85     $date_timestamp = strtotime($depdate);
86
87     if ($date_timestamp < $today) {
88         ?>
89         <script>
90             window.alert('Enter Valid Date!!..');
91             window.history.back();
92         </script>
93         <?php
94     } else{
95
96         if ($from==$to){
97             ?>
98             <script>
99                 window.alert('Pickup and Destination cannot be same');
100                 window.history.back();
101             </script>
102             <?php
103         }
104         else{
105             //$sql="INSERT INTO airport(pick,dest,depdate,airportid) VALUES('$from','$to','$depdate','')";
106
107             if (mysqli_connect('localhost' 'root' '' 'airline')) {

```

Figure 4.3 Page 1

### 4.7.4. Page 2 view available flights(php)

```

1 <?php
2 session_start();
3 if(!isset($_SESSION['sess_from']) && !isset($_SESSION['sess_user']) && !isset($_SESSION['sess_to']) && !isset($_SESSION['sess_depdate']))){
4     header("location:page1.php");
5     exit();
6
7 } else {
8     $pattern = '/[ ]/';
9     $string = "1998-11-09 15:00:00";
10    //echo "<pre>", print_r( preg_split($pattern, $string), 1 ), "</pre>";
11    //echo $string
12
13    ?>
14    <!doctype html>
15
16    <html>
17    <head>
18    <title>Welcome2</title>
19    <style>
20        body{
21            background-image: url("hello.jpg");
22            margin-top: 100px;
23            margin-bottom: 100px;
24            margin-right: 150px;
25            margin-left: 80px;
26            background-size: 100%;
27            background-attachment: fixed;
28            color: #261A15;
29            font-family: 'Vantramanav', sans-serif;;
30            font-size: 100%;
31
32        }
33        h1 {
34            color: rgb(44,62,80);
35            font-family: verdana;
36            font-size: 100%;
37        }
38        h2 {
39            color: rgb(1,50,67);

```

Figure 4.4 Page 2

## Chapter 5

### Result Snaps

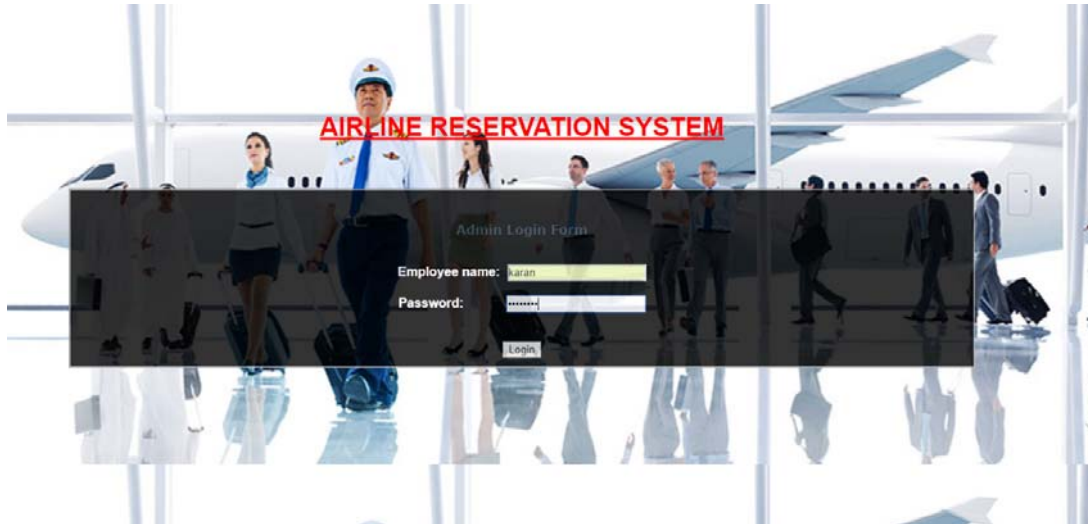


Fig 5.1 Admin login Page

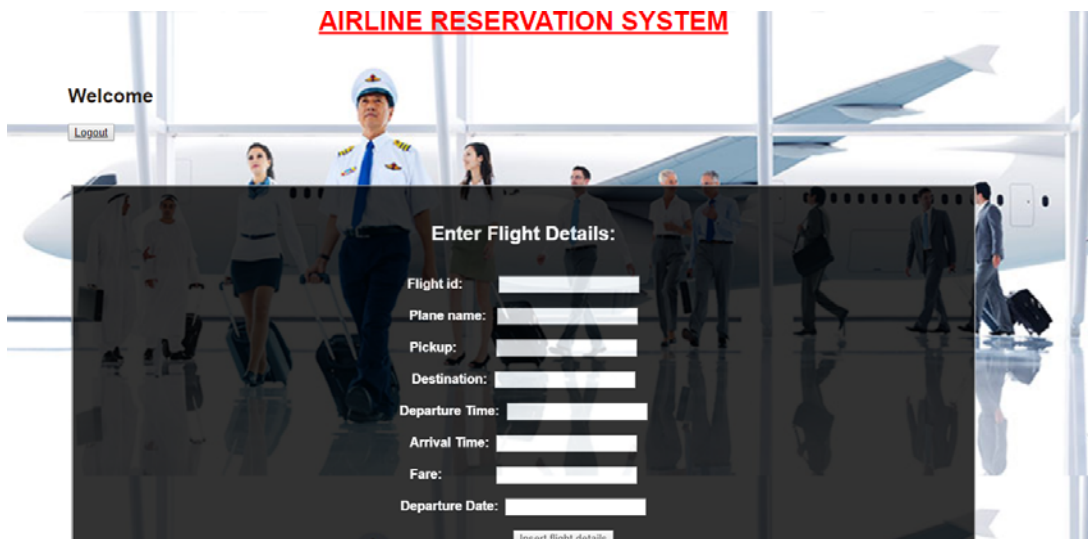


Fig 5.2 Admin Add Flights Page



Fig 5.3 Admin View Records Page

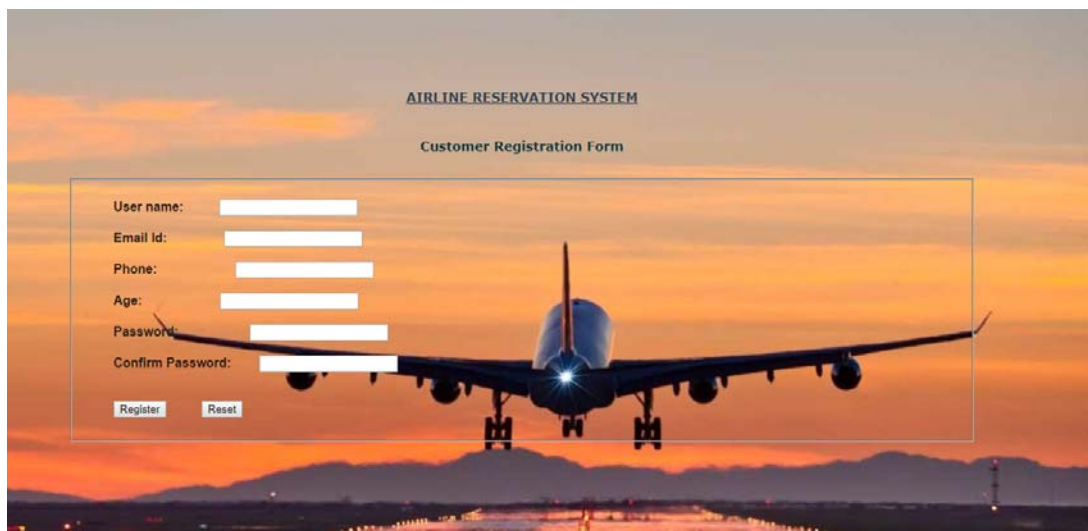


Fig 5.4 Customer Sign up Page

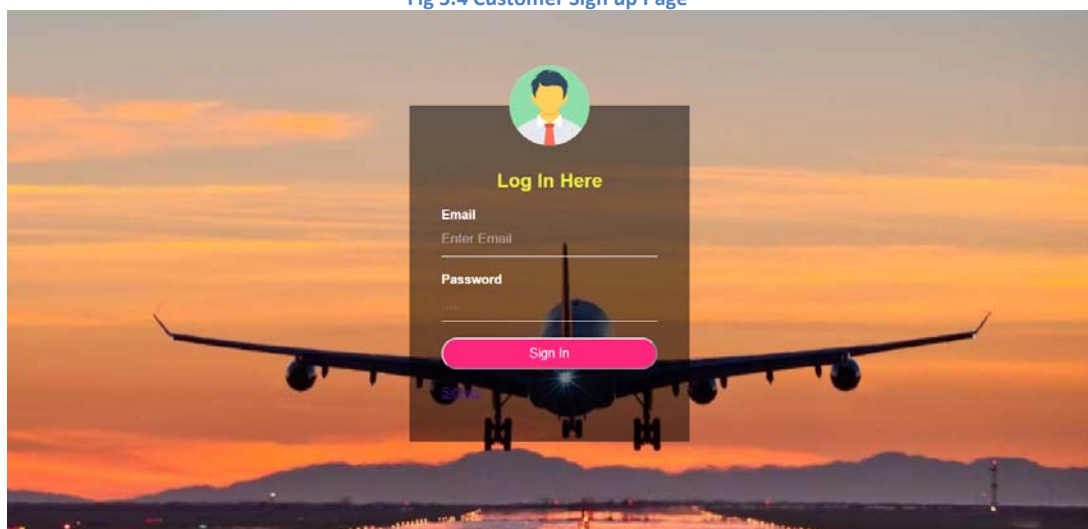


Fig 5.5 Customer Login Page



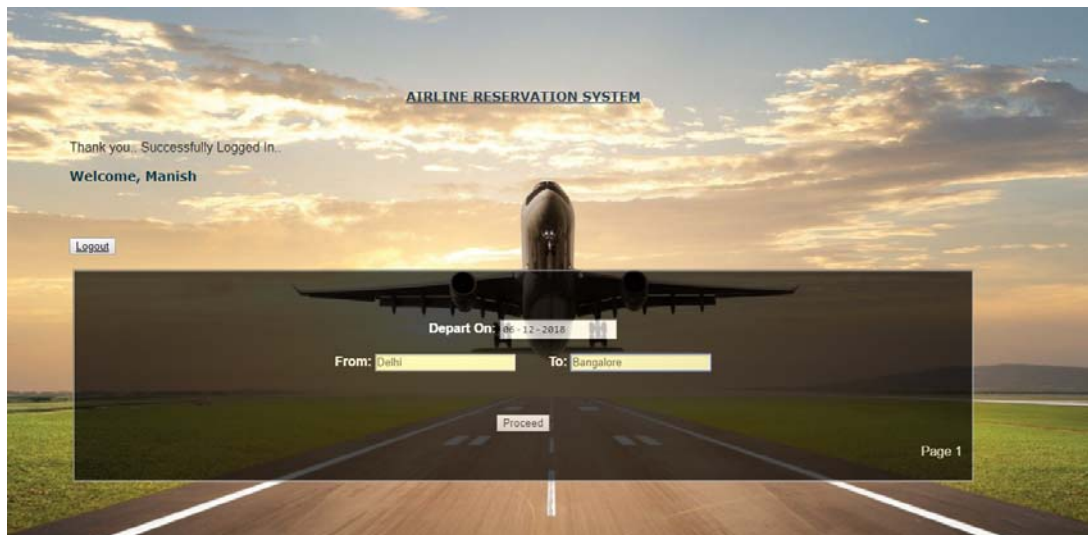


Fig 5.6 Customer Choose S &amp; D



Fig 5.7 Customer Choose Flight Page

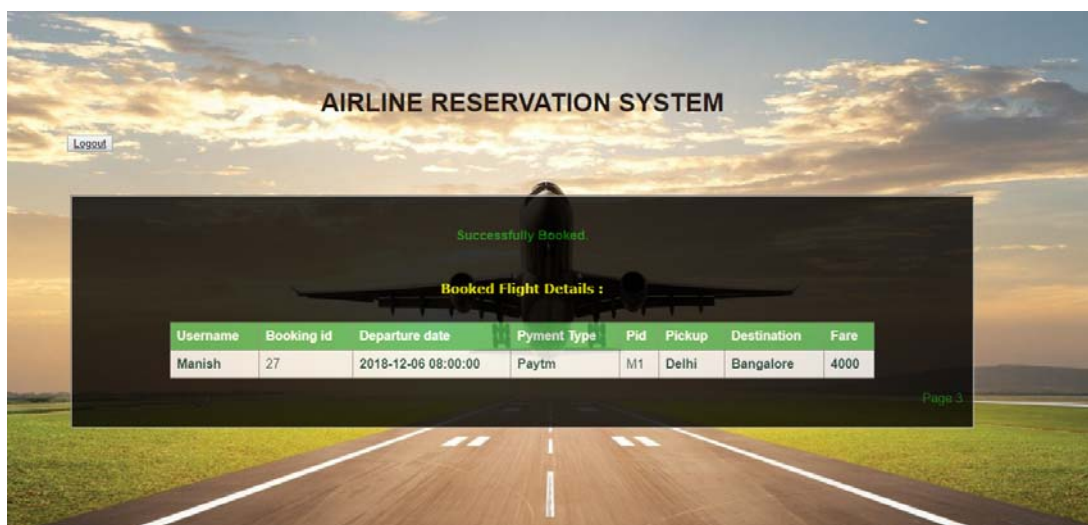


Fig 5.8 Customer Booking Details Page

## Chapter 6

### Conclusion

#### 6.1. Conclusion

The Airline Reservation System is a great improvement over the manual system which uses lots of manual work and paper. The computerization of the system speeds up the process. This system was thoroughly checked and tested with dummy data and found to be very reliable.

Thus, we have implemented a fully comprehensive and minimalistic efficient system for use by admins and customers without any additional training.

#### 6.2. Limitations of the Project

If there is a delay in some flight or it is cancelled due to some reason it is not removed from the bookings and aircrafts and it still appears there. New admins cannot be added to the system from the UI, for adding a new admin changes has to be made in the backend.

A user once books his flight cannot make any changes to his details like address, phone number etc. moreover there is no feature to edit the customer details. Payment verification steps needs to be added.

Only one ticket can be booked per person, in order to book multiple ticket the user has to signup/sign in so that his/her details are fed automatically.

#### 6.3. Future Enhancements

The Airline Reservation System can be enhanced by including more functionality like entering Seat details, Track the number of trips, maintain Customers Feedback, Reports, Billing etc.

We can further add an improvised booking system which far more efficient and reliable.

## References

### Book references

1. Learn to Code HTML and CSS: Develop and Style Websites (Web Design Courses) 1st, Kindle Edition by Shay Howe
2. *PHP 6 and MySQL 5* - Larry Ullman

### Website references

1. <https://www.codecademy.com/>
2. <https://dash.generalassemb.ly/>
3. <https://www.w3schools.com/>
4. <http://www.tutorialspoint.com/php/>
5. <https://killerphp.com>
6. <https://www.w3schools.com/>
7. <https://www.youtube.com/watch?v=ueWpNe0PG34>
8. <https://blog.udemy.com/xampp-tutorial/>
9. [https://www.tutorialspoint.com/plsql/plsql\\_triggers.html](https://www.tutorialspoint.com/plsql/plsql_triggers.html)
10. <https://www.geeksforgeeks.org/sql-procedures/>