VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi - 590018

Mini Project Report
on

"AIRLINE MANAGEMENT SYSTEM"

Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF ENGINEERING
in

COMPUTER SCIENCE & ENGINEERING
by

Thanmayee N Shetty	**4MT20CS171**
Tejaswini Peeru Gouda	**4MT20CS170**

Under the Guidance of

Mr. Pradeep Nazareth	**Dr. Dinesha L**
Senior Assistant Professor	**Associate Professor**

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

(Accredited by NBA)

MANGALORE INSTITUTE OF TECHNOLOGY &ENGINEERING

Accredited by NAAC with A+ Grade, An ISO 9001: 2015 Certified Institution
(*A Unit of Rajalaxmi Education Trust®, Mangalore - 575001*)
Affiliated to VTU, Belagavi, Approved by AICTE, New Delhi
Badaga Mijar, Moodabidri-574225, Karnataka

2022-23

MANGALORE INSTITUTE OF TECHNOLOGY &ENGINEERING

Accredited by NAAC with A+ Grade, An ISO 9001: 2015 Certified Institution
(*A Unit of Rajalaxmi Education Trust®, Mangalore - 575001*)
Affiliated to VTU, Belagavi, Approved by AICTE, New Delhi
Badaga Mijar, Moodabidri-574225, Karnataka

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

(Accredited by NBA)

CERTIFICATE

This is to certify that **Ms. THANMAYEE N SHETTY (4MT20CS171), Ms. TEJASWINI PEERU GOUDA (4MT20CS170)** has satisfactorily completed the mini project entitled **"AIRLINE MANAGEMENT SYSTEM"** for the **DBMS Laboratory with Mini Project (18CSL58)** lab as prescribed by the VTU for 5th semester B.E. Computer Science and Engineering branch for the academic year 2022 – 2023.

…………………………	**..………………………..**	**...………………………**
Mr. Pradeep Nazareth	**Dr. Dinesha L**	**Dr. Ravinarayan B**
SeniorAssistant Professor	**Associate Professor**	**Associate Professor & HOD**
Dept. of CSE	**Dept. of CSE**	**Dept. of CSE**
MITE, Moodabidri	**MITE, Moodabidri**	**MITE, Moodabidri**

Name of Examiners	**Signature of the Examiners**
1. …………………………..	**…………………………...**
2. …………………………..	**…………………………...**

i. ACKNOWLEDGEMENT

The successful completion of any significant task is the outcome of invaluable aggregate combination of different people in radial direction explicitly and implicitly. We would therefore take opportunity to thank and express our gratitude to all those without whom the completion of project would not be possible.

We express our thanks to **Mr.Pradeep Nazareth, Senior Assistant Professor and Dr.Dinesha L, Associate Professor**, Department of Computer Science and Engineering for having provided all the facilities that helped us in timely completion of this report.

We express our sincere gratitude to **Dr. Ravinarayan B, Associate Professor & Head of the Department, Computer Science and Engineering** for his support and guidance.We would like to thank **Dr. M S Ganesha Prasad**, **Principal, Mangalore Institute of Technology and Engineering, Moodabidri** for his support and encouragement.

I express my sincere gratitude to our institution and management for providing us with good infrastructure, laboratory facilities, qualified and inspiring staffs, and whose guidance was of immense help in completion of this seminar successfully.

THANMAYEE N SHETTY (4MT20CS171)

TEJASWINI PEERU GOUDA (4MT20CS170)

ii. ABSTRACT

An airline management system is a database management system (DBMS) mini-project that aims to automate and streamline various processes related to airline operations. This system would store and manage data such as flight schedules, passenger information, ticket prices, and booking information, as well as provide a user-friendly interface for customers to book and manage their flights. Additionally, the system would also include features such as flight scheduling and tracking, inventory management, and financial reporting.

The system can connect airport management systems to provide more accurate and relevant information to the passengers and the management.

Overall, the goal of this mini-project is to improve efficiency, reduce costs, and enhance the customer experience for an airline.

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

Chapter 1

INTRODUCTION

DBMS stands for database management system refers to the technology for creating and managing databases. DBMS is a software tool to organize (create, retrieve, update and manage) data in a database. The main aim of the DBMS is to supply a way to store up and retrieve database information that is both convenient and efficient. By data, we mean known facts that can be recorded and have embedded meaning. Database system are meant to handle a large collection of information and providing mechanisms that can do the manipulation that stored information. Moreover, the database system must ensure the safety of the information stored, despite system crashes or attempts at unauthorized access.

The aim of case study is to design and develop a database maintaining the records of different flights, flight stchedules and passenger information. The record of user status in airline databsase system includes its flight id ,name, source, destination, departure and arrival dates and days on which it is available, whereas record of flight status includes dates for which tickets can be booked, total number of seats available. The database has been developed. Airline management system is a website has various kinds of information that help regarding booking of tickets via airways. Users will be able to search the flight availability, the exact fare, the arrival and departure date of the flight. They can also book the ticket .

Users frequently need to know about their ticket reservation status, ticket availability on a particular date or for a place, flight arrival or departure details etc. Customer information centers at the airports are unable to serve such queries at peak periods. On most of the reservation systems, there are long queues, so it takes a long time for any individual to book the ticket. As now there are no call center facilities available to solve the queries of the passengers. Flight booking will help the passengers to book their tickets for their journey either for economic purpose or for business purpose, search flight status and get details of particular flight schedules including their fare details. To use this system its users should be signed up and should have a valid login and password to make their reservation. The main aim of this login module is to keep each passenger's safe which will not allowed to make fraud and make their transactions secure. After making registration passengers have to enter the source and destination airport name and the system will able to provide the list of all available flights for particular destination and their arrival and departure date in available flight status. Upon selecting particular flight, system will able to display the economic or business fair of the available flights. To book their tickets passengers have to select class type and total number of adult passengers and children passengers. When all the selection work will be done, passengers will be provided with a user id.

1.1 Problem Statement

The current problem with offline flight booking systems is that they are prone to errors and are not very efficient. These systems usually involve manual processes such as filling out paper forms and calling travel agents, which can lead to mistakes and delays. Additionally, offline systems do not offer real-time updates, so customers may not be able to see the latest flight information or availability.

Online flight booking systems are useful because they provide a convenient and efficient way for customers to book flights. These systems typically offer real-time updates on flight schedules, availability, and prices, making it easy for customers to find and book the best flight options. Additionally, online systems are usually user-friendly and easy to navigate, allowing customers to quickly find and book flights without the need for assistance from travel agents.

1.2 Objectives

- The main objective of the Airline Database System is to manage the details of flight, flight status, flight reservation and booking . It also manages all the information about the user who booked the flight , available flight details.
- The project is totally at administrative end and thus only the administrator is guaranteed the access. .
- The purpose of the project is to build a website to reduce the manual work by providing the a convenient and efficient way for customers to book flights.

1.3 HTML5

HTML is the language for describing the structure of Web pages. HTML allows users to create and structure sections, headings, links, paragraphs, and more, on a website using various tags and elements. Almost everything you want to create on a web page can be done using a specific HTML code.

1.4 CSS3

CSS stands for Cascading Style Sheet**.** It gives an additional style to the HTMLdocument. A cascading style sheet is a language that is designed to define the document formatting and look written in a markup language. Generally, CSS is applied with HTML documents to change various styles of user interfaces and web pages. CSS is the language for describing the presentation of Web pages, including colors, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML based markup language.

1.5 PHP

PHP is a general-purpose scripting language geared towards web development. PHP code is usually processed on a web server by a PHP interpreter implemented as a module, a daemon or as a Common Gateway Interface (CGI) executable. On a web server, the result of the interpreted and executed PHP code – which may be any type of data, such as generated HTML or binary image data – would form the whole or part of an HTTP response. Various web template systems, web content management systems, and web frameworks exist which can be employed to orchestrate or facilitate the generation of that response. Additionally, PHP can be used for many programming tasks outside the web context, such as standalone graphical applications and robotic drone control. PHP code can also be directly executed from the command line.

1.6 SQL

SQL (Structured Query Language) is a standardized programming language that's used to manage relational databases and perform various operations on the data in them. The uses of SQL include modifying database table and index structures; adding, updating and deleting rows of data; and retrieving subsets of information from within a database for transaction processing and analytics applications. Queries and other SQL operations take the form of commands written as statements commonly used SQL statements include select, add, insert, update, delete, create, alter.

Chapter 2

REQUIREMENT ANALYSIS AND SPECIFICATION

2.1 Functional Requirements

These are statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations. In some cases, the functional requirements may also explicitly state what the system should not do. The functional requirements for a system describe what the system should do. These requirements depend on the type of software being developed, the expected users of the software and the general approach taken by the organization when writing requirements. When expressed as user requirements, the requirements are usually described in an abstract way. However, functional system requirements describe the system function in detail, its inputs and outputs, exceptions, and so on. Functional requirements for a software system may be expressed in several ways.

The function Requirements of Airline Management System are as follows:

Register Module :

- User SignUp : The user needs to provide their username, password, and he has to confirm his password once more while registering.
- Admin SignUp : The admin needs to provide his username, password, and he has to confirm his password while registering. The signed in Admins can only Add the flights, Update and delete the flights. These details will be stored in the database.

Admin Login Module:

- For login admin will input their user name and password and password which will be compared with a database content.

User Login Module:

- For login user will input their user name and password and password which will be compared with a database content.

Booking Module:

- Users can book the flight by filling the proper details such as source and destination, departure and arrival dates and other passenger information etc.

Hardware Specifications:

Processor	: Intel(R) Core(TM) i3-1005G1 CPU @ 1.20GHz 1.19 GHz
Main Memory	: 8.00 GB RAM
Hard disk	:1TB

Software Specifications:

Operating system	: Windows 11
Front end	: HTML, CSS
Back end	: SQL, PHP
Software	: Visual Studio Code, XAMPP

2.2 Non-Functional Requirements

Non-functional requirements are requirements that are not directly concerned with the specific functions delivered by the system. They may relate to emergent system properties such as reliability, response time and store occupancy. Alternatively, they may define constraints on the system such as the capabilities of I/O devices and the data representations used in system interfaces. The plan for implementing functional requirements is detailed in the system design. The plan for implementing non-functional requirements is detailed in the system architecture. Non-functional requirements are often called qualities of a system. Other terms for non- functional requirements are "constraints", "quality attributes", "quality goals", "quality of service requirements" and "non-behaviroul requirements". Qualities, that are non-functional requirements, can be divided into two main categories: Execution qualities, such as security and usability, which are observable at run time.

Reliability:

- Porter reservation system shall be available 24 hours a day, 7 days a week.
- Porter reservation shall show always provide real time information about the reservation information.
- Porter reservation system shall robust enough to have a high degree of fault tolerance. For example, if the user enters a negative number of passengers or a value too large, the system should not crash and shall identify the invalid input and produce a suitable error message.

Usability:

- Porter reservation system shall provide a easy-to-use graphical interface similar to other existing reservation system so that the users do not have to learn new style or interaction.
- The web interface should be intuitive and easily navigable users should be able to understand the menu and options provided by Porter reservation system.
-

Performance:

- The system provides user-friendly interface, any common people with little technical knowledge can use the system.
- System is robust, reliable and fast, provides more efficiency.

Security:

- System provides security for the admin by only allowing them to enter into the account with their respective username and password.
- Only admin have the privileges to delete/add database contents which are used by the user.

Chapter 3

SYSTEM DESIGN

System Design process partitions the system into subsystems based on the requirements. It establishes overall system architecture and is concerned with identifying various components, specifying relationships among components, specifying software structure, maintaining a record of design decisions and providing a blueprint for the implementation phase. Design consists of architecture design and detailed design is concerned with the details of how to package processing modules and how to implement the processing algorithms, data structures and interconnections among modules and data structures.

3.1 ER Diagram

ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships.

In Figure 3.1 there are total six entities namely Admin, Feedback, Userlogin, Users, Flights and Cities. If we look into the relation between the Admin and Feedback where in admin Id is the Primary Key. we took relation “handles”as a relationship with the cardinality ratio 1:N because N number of feedbacks can be handled by a single user.We took a relation “gives” as a relationship between Userlogin and Feedback where User Id is the primary key in Userlogin entity with cardinality ratio 1:N because 1 user can give N number of Feedbacks. We took a relation “manage” as a relationship between Userlogin and Users entity where User id is the primary key with cardinality ratio 1:1 because 1 Userlogin manages only one Users . We took a relation “books” as relationship between Users and Flights entity where Flight_id is the primary key with cardinality ratio 1:N because 1 Users can book N number of flights. We took a relation “adds” as a relationship between Admin and cities entity with cardinality ratio 1:N because 1 admin can add N number of cities. We took a relation “handle” as a relationship between admin and flights with cardinality ratio 1:N because 1 admin can handle N number of Flights. We took a relation “adds” as a relationship between Flights and cities entity with cardinality ratio M:N because M Flights can add N number of cities.

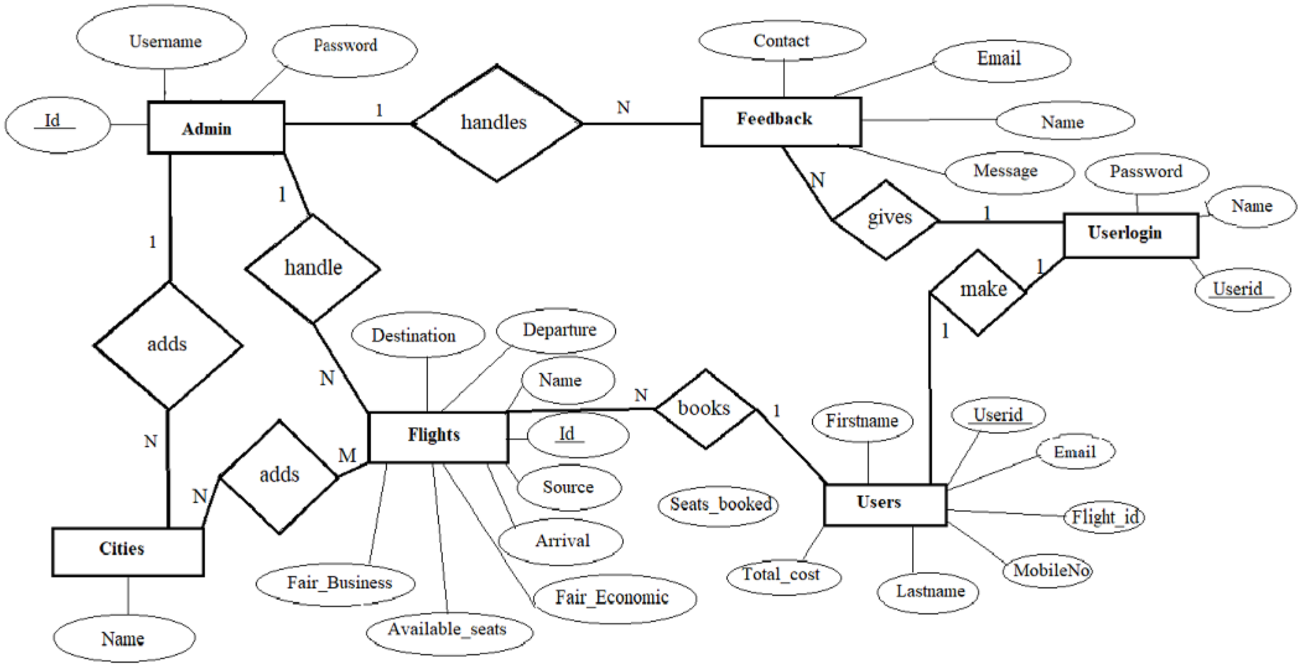


Figure 3.1 ER Diagram of Airline Management System

3.2 Schema Diagram

The design of the database is called a schema. This tells us about the structural view of the database. It gives us an overall description of the database. A database schema defines how the data is organized using the schema diagram. A schema diagram is a diagram which contains entities and the attributes that will define that schema. A schema diagram only shows us the database design. It does not show the actual data of the database.

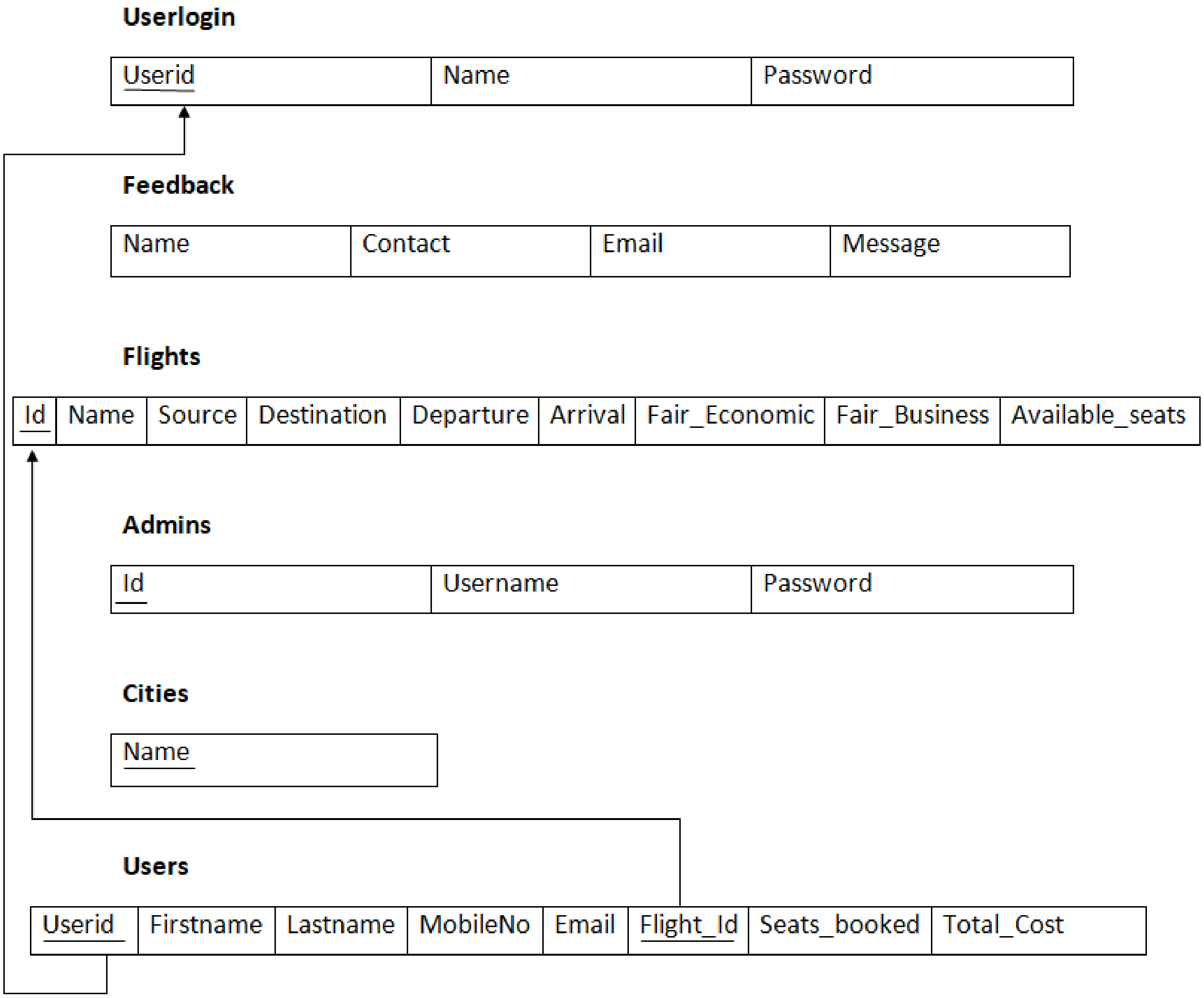


Figure 3.2 Schema Diagram of Airline Management System

In Fig 3.2 Userlogin relation has three entities named as User Id, Name, Password where User id is the primary key.

In feedback relation , it has four entities i.e. Name, Contact, Email and Message.

In relation Flights, the entities are Id, Name, Source, Destination, Departure, Arrival, Fair_Economic, Fair_Business and Available_Seats where Id is the primary Key.

In relation Admin, the entities are Id, Username and Password where Id is the primary Key.

In relation cities, the entity is Name.

In Relation Users, the entities are Userid, Firstname, Lastname, MobileNo, Email, Flight_Id, Seats_booked and Total_cost. Where Userid is the primary key and Flight_Id is foreign Key.

Chapter 4

IMPLEMENTATION

PHP: Hypertext Pre-processor (or simply PHP) is a server-side scripting language designed for web development, and also used as a general-purpose programming language. PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management systems, and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.

This project uses **HTML** as front-end tool. Hypertext Mark-up Language (HTML) is the standard mark-up language for creating web pages and web applications. With Cascading Style Sheets (**CSS**), it forms a triad of cornerstone technologies for the world wide web. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Browsers do not display the HTML tags but use them to interrupt the content of the page.

4.1 CODE SNIPPETS :

```
<?php

$servername = "localhost";

 $username = "root";

 $password = "";

 $database = "trial";

// Create connection

 $conn = mysqli_connect($servername, $username, $password,$database);

// Check connection

 if (!$conn) {

     die("Connection failed: " . mysqli_connect_error());

 }

 echo "Connected successfully<br>";

?>
```

Figure 4.1 Code Snippet for connection module

Figure 4.1 defining the database server, database user, database password and database project for establishing connection with the database.

```
<?php
// initializing variables
$username = "";
$errors = array();
 include 'config.php';
```

```
  $username = $_POST['username'];
  $password_1 = $_POST['password1'];
  $password_2 = $_POST['password2'];
// form validation: ensure that the form is correctly filled ...
  // by adding (array_push()) corresponding error unto $errors array
  if ($password_1 != $password_2) {
            array_push($errors, "The two passwords do not match");
  }
  // first check the database to make sure
  // a user does not already exist with the same username and/or email
  $user_check_query = "SELECT * FROM admins WHERE Username='$username' LIMIT 1";
  $result = mysqli_query($conn, $user_check_query);
  $user = mysqli_fetch_assoc($result);
  if ($user) { // if user exists
    if ($user['username'] === $username) {
      array_push($errors, "Username already exists");
    }
  }
  // Finally, register user if there are no errors in the form
  if (count($errors) == 0) {
            $password = $password_1;//encrypt the password before saving in the database

            $query = "INSERT INTO admins (Username, Password)
                                    VALUES('$username','$password')";
            mysqli_query($conn, $query);
    echo "Account Created. PLease login again.";
    header("Refresh:2; url= admin.html");
  }
  else {
    $arrlength = count($errors);
    for($x = 0; $x < $arrlength; $x++) {
        echo $errors[$x];
        echo "<br>";
      header("Refresh:2; url= signup.html");

     }

   }
```

Figure 4.2 Code Snippet for Admin Login module

Figure 4.2 the username and password must be matched will the registered username and password.
If the password is incorrect ‘The two passwords do not match’ message will be displayed.
If the username is incorrect ‘Your login name is invalid’ message will be displayed.
If all the details are right then it takes the user for the next page.

```
<?php
// initializing variables
$name = "";
$errors = array();
 include 'config.php';
  $name = $_POST['us_name'];
  $password_1 = $_POST['pass_no1'];
  $password_2 = $_POST['pass_no2'];
  // form validation: ensure that the form is correctly filled ...
  // by adding (array_push()) corresponding error unto $errors array
  if ($password_1 != $password_2) {
	array_push($errors, "The two passwords do not match");
  }
  // first check the database to make sure
  // a user does not already exist with the same username and/or email
  $user_check_query = "SELECT * FROM userlogin WHERE name='$name' LIMIT 1";
  $result = mysqli_query($conn, $user_check_query);
  $user = mysqli_fetch_assoc($result);
  if ($user) { // if user exists
    if ($user['us_name'] === $name) {
      array_push($errors, "Username already exists");
    }
  }
  // Finally, register user if there are no errors in the form
  if (count($errors) == 0) {
	$password = $password_1;//encrypt the password before saving in the database
	$query = "INSERT INTO userlogin (name, Password)
					VALUES('$name','$password')";
	mysqli_query($conn, $query);
    echo "Account Created. PLease login again.";
    header("Refresh:2; url= userlogin.html");
  }
  else {
    $arrlength = count($errors);
    for($x = 0; $x < $arrlength; $x++) {
      echo $errors[$x];
      echo "<br>";
      header("Refresh:2; url= usersignup.html");
     }
    }
```

Figure 4.3 Code Snippet for User Sign Up module

Figure 4.3 the input fields are stored in the database using post operation, the password and the confirm password should be matching, if it is not matching then the alert query will be displayed. Insert query is used for inserting the fields.

```
<<?php
include 'config.php';
$id = $_POST['id'];
$name = $_POST['name'];
$source = $_POST['source'];
$destination = $_POST['destination'];
$departure = $_POST['departure'];
$arrival = $_POST['arrival'];
$fair_economic = $_POST['Fair_Economic'];
$fair_business = $_POST['Fair_Business'];
$Available_seats = $_POST['Available_seats'];
header("Refresh:2; url=addflight.html");
$sql = "INSERT INTO flights(Id, Name, Source, Destination, Departure, Arrival, Fair_Economic,
Fair_Business, Available_seats) VALUES ('$id', '$name', '$source', '$destination', '$departure', '$arrival',
'$fair_economic', '$fair_business', '$Available_seats')";
if((mysqli_query($conn,$sql))){
  echo "Flight Added!!!";
$sql1 = "INSERT IGNORE INTO cities (Name) VALUES('$source')";
$sql2 = "INSERT IGNORE INTO cities (Name) VALUES('$destination')";
mysqli_query($conn,$sql1);
mysqli_query($conn,$sql2);
header("Refresh:2; url=welcome.php");
mysqli_close($conn);
}
else{
  echo "Flight Not  Added!!!";
}
?>
```

Figure 4.4 Code Snippet for Add flight module

Figure 4.4 Admin can add the flight by filling the proper details such as Flight id, flight name, source and destination, departure and arrival dates and economic and business fairs etc in Add Flight Page.

Chapter 5

TESTING

Software testing is the process used to identify the correctness, security, completeness and quality of developed computer software. This includes the process of executing the program or applications with the intent of finding errors. An individual unit, functions or procedures of a developed project is verified and validated and these units are fit for use.

5.1 Testing process

Best testing process is to test each subsystem separately, as we have done in the project. Best done during implementation. Best done after small sub-steps of the implementation rather than large chunks. Once each lowest level unit has been tested, units are combined with related units and retested in combination. This proceeds hierarchically bottom-up until the entire system is tested as a whole.

Typical levels of testing:

- Module- package, abstract data type, class
- Sub-system- collection of related modules, cluster of classes, method-message paths
- Acceptance testing- whole system with real data

Alpha testing is acceptance testing with a single client (common for bespoke systems).

Beta testing involves distributing systems to potential customers to use and provide feedback. In this project, beta testing has been followed. This exposes the system to situations and errors that might not be anticipated by us.

5.1.1 Unit testing

Unit testing is the process of testing individual software components unit or modules. Since it needs the detailed knowledge of the internal program design and code this task is done by the programmer and not by testers.

5.1.2 Integration Testing

Integration testing is another aspect of testing that is generally done in order to uncover errors associated with the flow of data across interfaces. The unit-tested modules are grouped together and tested in small segments, which makes it easier to isolate and correct errors. This approach is continued until we have integrated all modules to form the system as a whole. After the completion of each module it has been combined with the remaining module to ensure that the project is working properly as expected.

5.1.3System Testing

System testing tests a completely integrated system to verify that it meets its requirements. After the completion of the entire module they are combined together to test whether the entire project is working properly.

5.2 Test Cases

A Test Case is a software testing document, which consists of events, action, input, output, expected result and actual result. Technically a test case includes test description, procedure, expected result and remarks. Test cases should be based primarily on the software requirements and developed to verify correct functionality and to establish conditions that reveal potential errors.

Test cases no	Test Case	Expected results	Status
1	Logging into Admin login	Username and password provided correct	Connected Successfully
2	Logging into Admin login	Username incorrect	Invalid Username
3	Logging into Admin login	Password Incorrect	Invalid Password
4	Logging into Admin login	Any field left empty	Please fill out all the fields

Table 5.1 Test Case for Admin Login

Table 5.1 represents the test case for the Admin login module. It shows both successful and unsuccessful results for the test cases.

Test cases no	Test Case	Expected results	Status
1	Logging into User login	All details provided correctly	Connected Successfully
2	Logging into User login	Any one field is incorrect	Invalid Username or password
3	Logging into User login	Any field left empty	Please fill out all the fields

Table 5.2 Test Case for User Login

Table 5.2 represents the test case for the User Login module. It shows both successful and unsuccessful results for the test cases.

Test cases no	Test Case	Expected results	Status
1	Entering passenger information	All details provided correctly	Saved successfully
2	Entering passenger information	Any one filed is incorrect	Please provide valid inputs

Table 5.3 Test case for passenger information

Table 5.3 represents the test case for the entering passenger information module. It shows both successful and unsuccessful results for the test cases .

Test cases no	Test Case	Expected results	Status
1	Entering flight booking details	All details provided correctly	Booked Successfully
2	Entering flight booking details	Any one field is incorrect	Booking Unsuccessful
3	Entering flight booking details	Any field left empty	Booking Unsuccessful

Table 5.4 Test Case for flight booking

Table 5.4 represents the test case for the flight booking module. It shows both successful and unsuccessful results for the test cases whether the ticket is booked or not.

Chapter 6

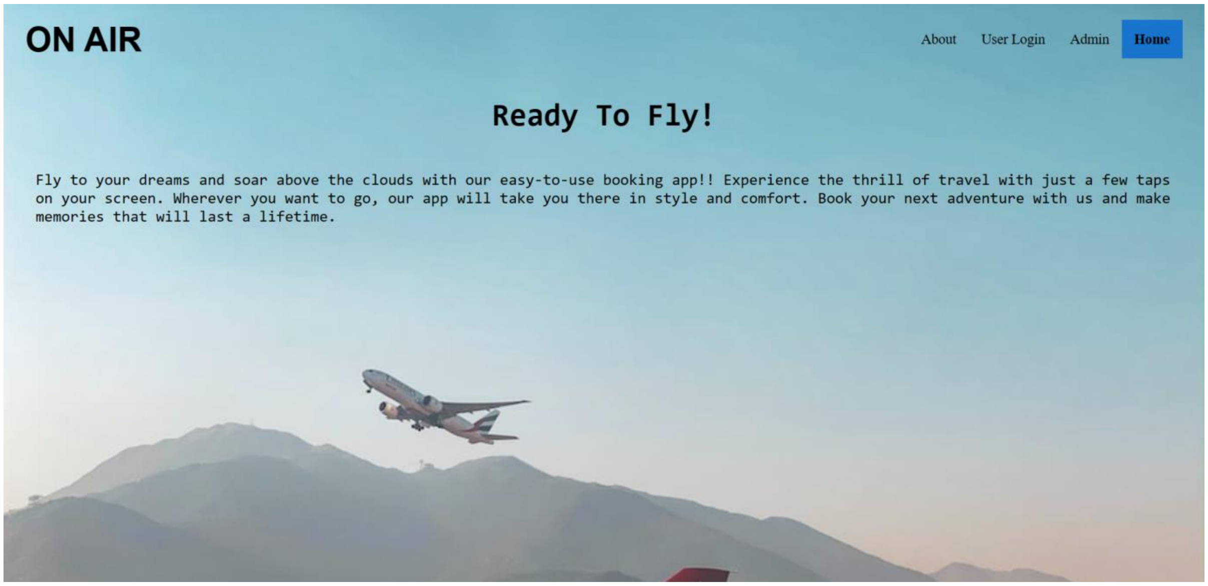
RESULT AND DISCUSSION

The Airline Management System is useful because it provides a convenient and efficient way for customers to book flights. These systems typically offer real-time updates on flight schedules, availability, and prices, making it easy for customers to find and book the best flight options. The System do offer real-time updates, so customers are able to see the latest flight information or availability.

The system can connect airport management systems to provide more accurate and relevant information to the passengers and the management. This is user-friendly and easy to navigate, allowing customers to quickly find and book flights without the need for assistance from travel agents.

Chapter 7

SCREENSHOTS



Ready To Fly!

Fly to your dreams and soar above the clouds with our easy-to-use booking app!! Experience the thrill of travel with just a few taps on your screen. Wherever you want to go, our app will take you there in style and comfort. Book your next adventure with us and make memories that will last a lifetime.

Fig 7.1 Screenshot of Homepage

Figure 7.1 Indicates the Home page. This contains the connections of About page, User login and Admin login pages.

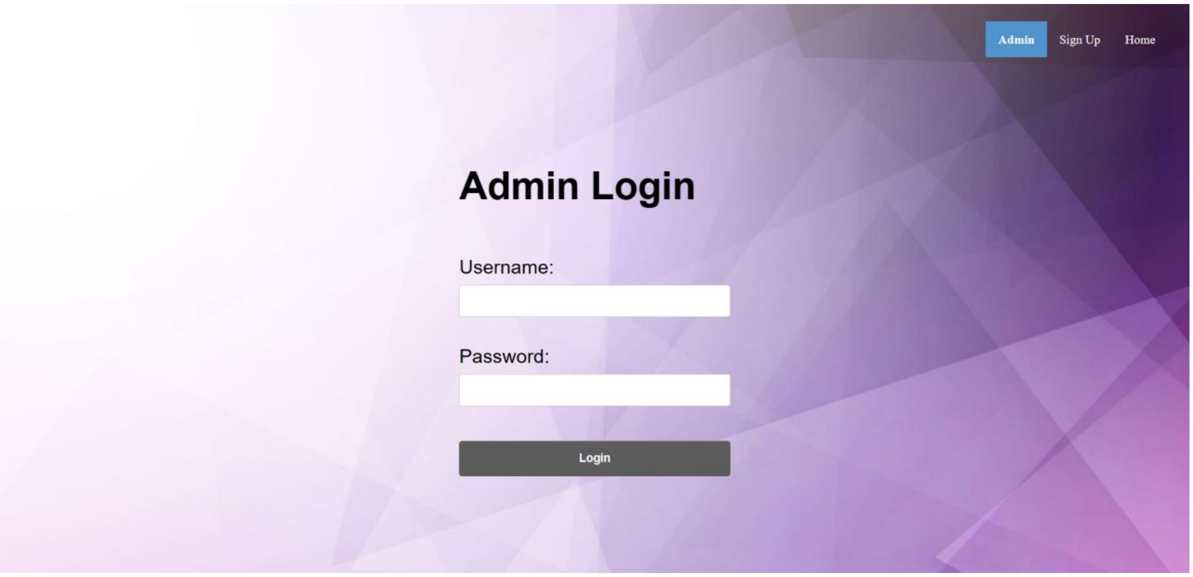


Fig 7.2 Screenshot of Admin Login

Figure 7.2 Indicates the Admin login page. If the admin has alredy registered to the website then he or she can manage the flight details.

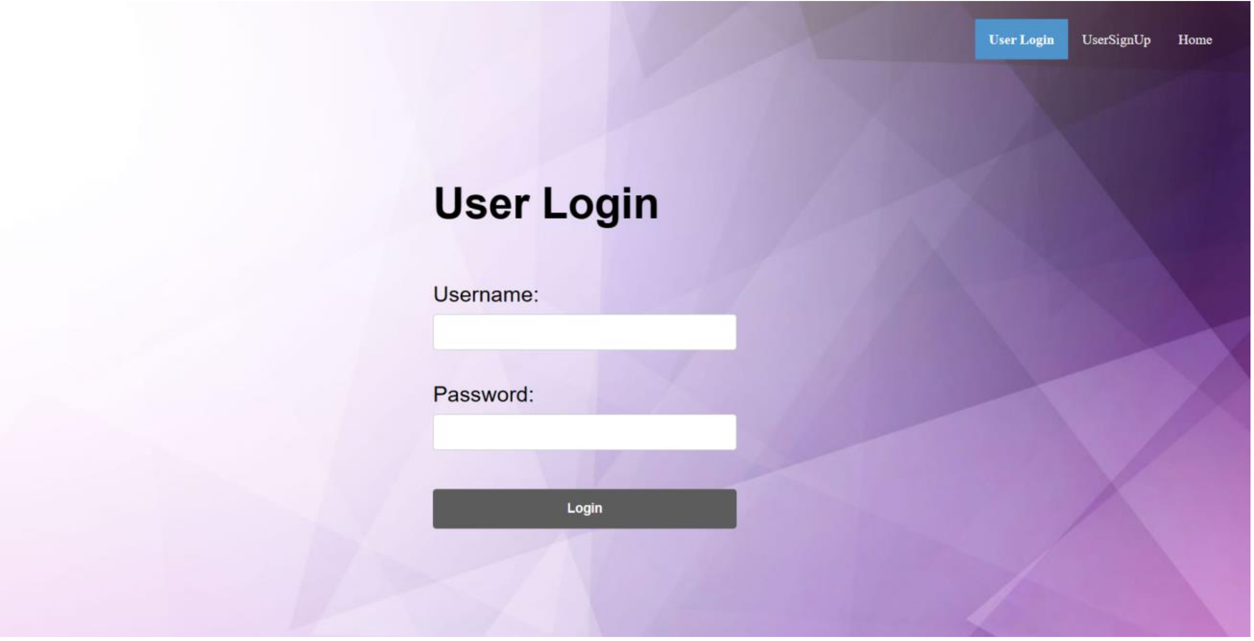
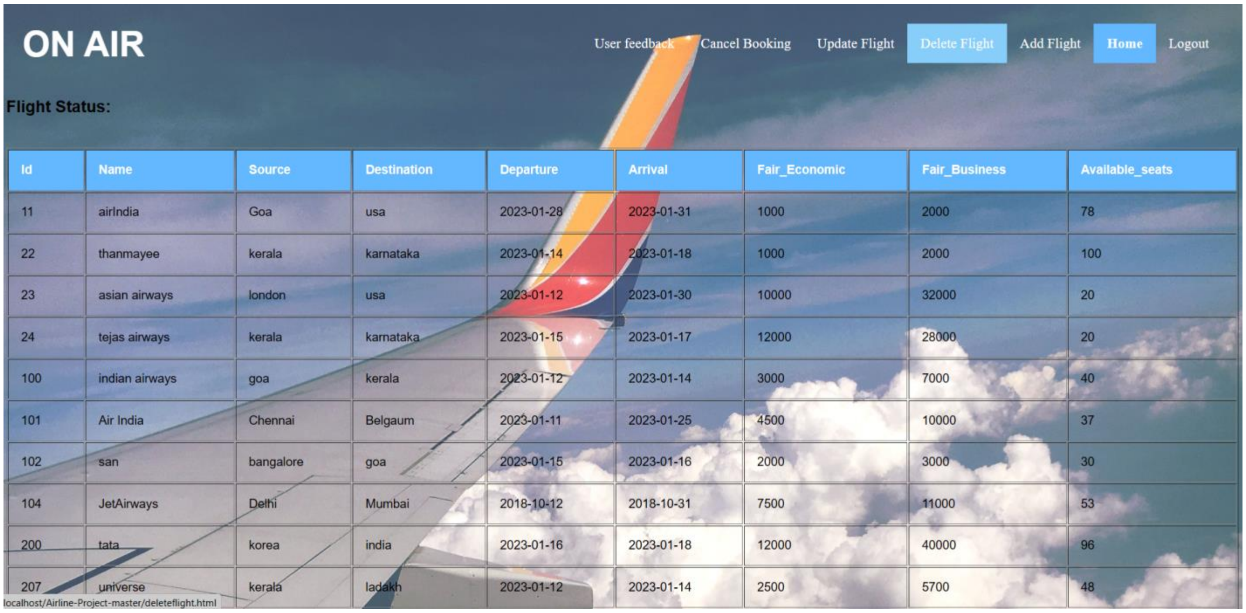


Fig 7.3 Screenshot of User Login

Figure 7.3 Indicates the User login page. If the user has alredy registered to the website then he or she can get access to the website by filling details.



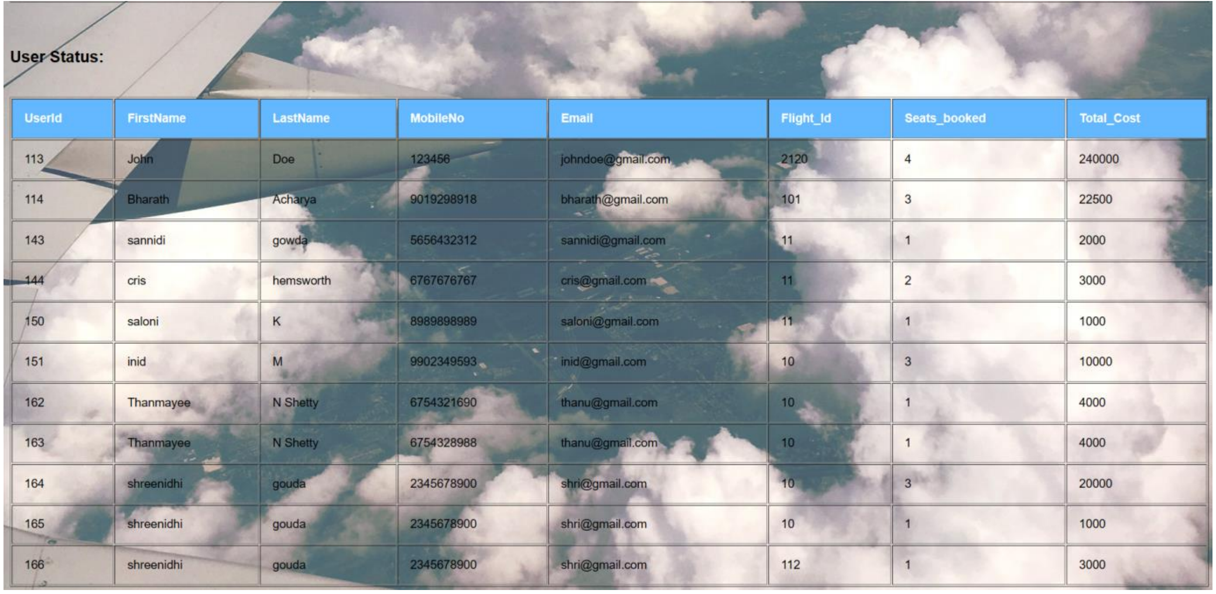
Flight Status:

Id	Name	Source	Destination	Departure	Arrival	Fair_Economic	Fair_Business	Available_seats
11	airIndia	Goa	usa	2023-01-28	2023-01-31	1000	2000	78
22	thanmayee	kerala	karnataka	2023-01-14	2023-01-18	1000	2000	100
23	asian airways	london	usa	2023-01-12	2023-01-30	10000	32000	20
24	tejas airways	kerala	karnataka	2023-01-15	2023-01-17	12000	28000	20
100	indian airways	goa	kerala	2023-01-12	2023-01-14	3000	7000	40
101	Air India	Chennai	Belgaum	2023-01-11	2023-01-25	4500	10000	37
102	san	bangalore	goa	2023-01-15	2023-01-16	2000	3000	30
104	JetAirways	Delhi	Mumbai	2018-10-12	2018-10-31	7500	11000	53
200	tata	korea	india	2023-01-16	2023-01-18	12000	40000	96
207	universe	kerala	ladakh	2023-01-12	2023-01-14	2500	5700	48

localhost/Airline-Project-master/deleteflight.html

Fig 7.4 Screenshot of Flight Status

Figure 7.4 Displays the Flight Status. Flight status contains the flight_id, flight name, source, destinations , departure and arrival day, price for economic and business classes and available seats.

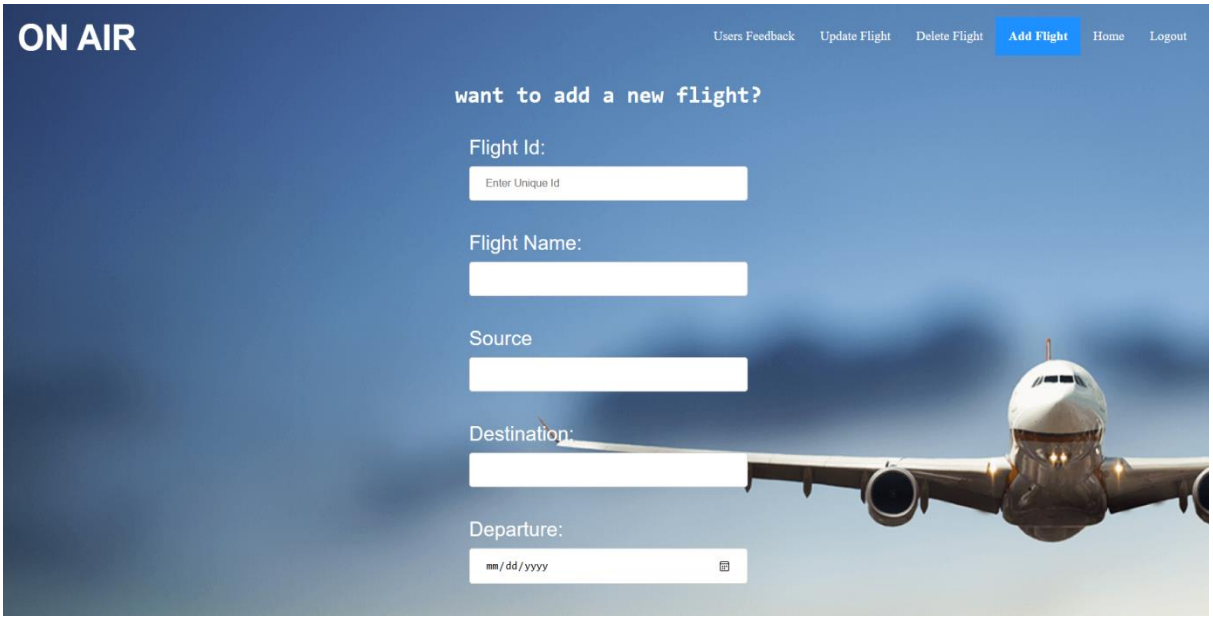


User Status:

UserId	FirstName	LastName	MobileNo	Email	Flight_Id	Seats_booked	Total_Cost
113	John	Doe	123456	johndoe@gmail.com	2120	4	240000
114	Bharath	Acharya	9019298918	bharath@gmail.com	101	3	22500
143	sannidi	gowda	5656432312	sannidi@gmail.com	11	1	2000
144	cris	hemsworth	6767676767	cris@gmail.com	11	2	3000
150	saloni	K	8989898989	saloni@gmail.com	11	1	1000
151	inid	M	9902349593	inid@gmail.com	10	3	10000
162	Thanmayee	N Shetty	6754321690	thanu@gmail.com	10	1	4000
163	Thanmayee	N Shetty	6754328988	thanu@gmail.com	10	1	4000
164	shreenidhi	gouda	2345678900	shri@gmail.com	10	3	20000
165	shreenidhi	gouda	2345678900	shri@gmail.com	10	1	1000
166	shreenidhi	gouda	2345678900	shri@gmail.com	112	1	3000

Fig. 7.5 Screenshot of User Status

Figure 7.5 Displays the User Status. User status contains the User id, first and last name, mobileno, Email, flight_id, seats_booked and Total cost .



want to add a new flight?

Flight Id:

Enter Unique Id

Flight Name:

Source

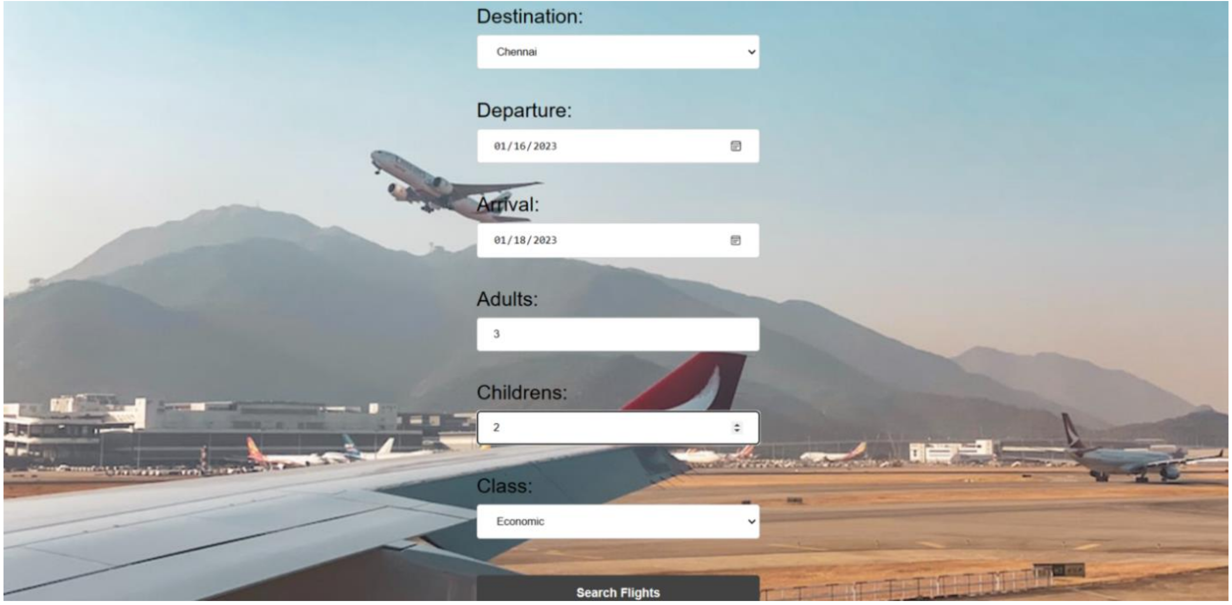
Destination:

Departure:

mm/dd/yyyy

Fig 7.6 Screenshot of Add Flight

Figure 7.6 Indecates the Add flight page. Admin can Add the new flight by specifying unique flight id and flight name and source, destinations and departure and arrival dates.



Destination: Chennai

Departure: 01/16/2023

Arrival: 01/18/2023

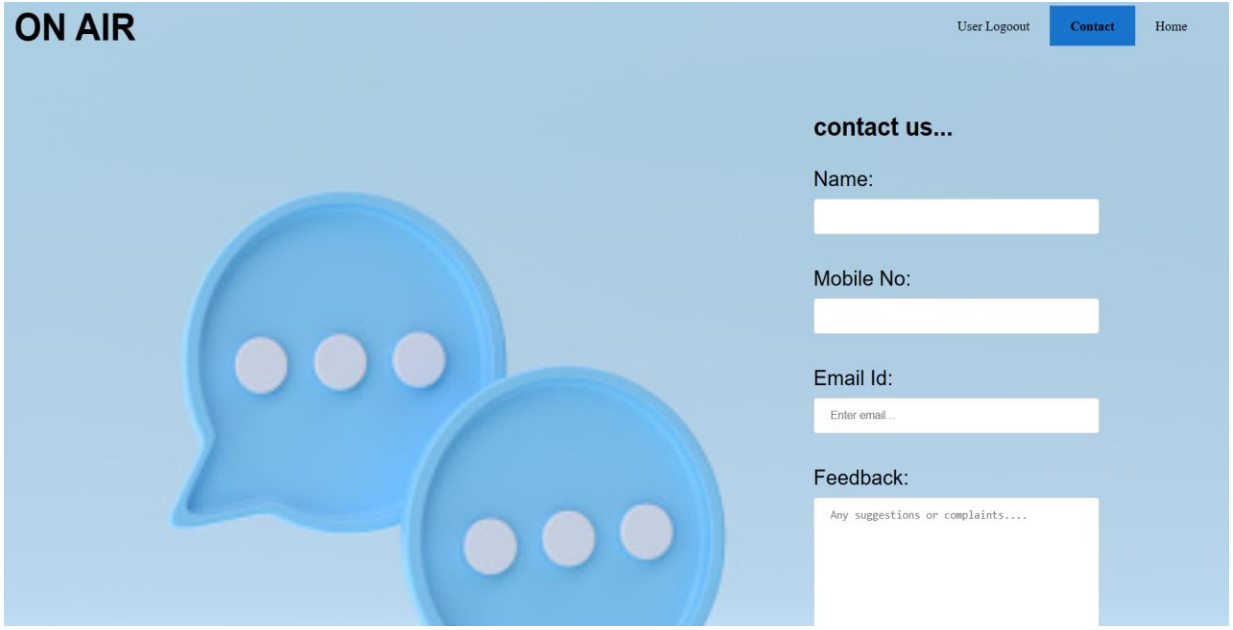
Adults: 3

Childrens: 2

Class: Economic

Fig 7.7 Screenshot of Flight booking

Figure 7.7 Indecates the Create flight booking page. Registered user can book the flight by giving inputs such as name, source, destinations and departure and arrival dates that he wants and he/she has to to specify the class (Economic/ Business).



contact us...

Name:

Mobile No:

Email Id: Enter email...

Feedback: Any suggestions or complaints....

Fig 7.8 Screenshot of User feedback

Figure 7.8 Indecates the User Feedback page. Registered user can contact by giving feedback. User should provide his/her Name, MobileNo, EmailId and can give a feedback.

Chapter 8

CONCLUSION & FUTURE ENHANCEMENTS

The Airline Management System facilitates the users to enquire about the flights available on the basis of source and destination, departure and arrival dates etc. The aim of case study is to design and develop a database maintaining the records of different flights, flight status and user status. With the help of online booking people can book their tickets online through internet, sitting in their home by a single click of mouse. Overall, the goal is to improve efficiency, reduce costs, and enhance the customer experience for an airline.

FUTURE ENHANCEMENTS

1. Virtual reality integration for previewing seat selection and cabin views.
2. AI-powered personalization and recommendations for flights and destinations.
3. Integration with other modes of transportation, such as trains and buses, for multi-modal trip planning.
4. Real-time flight tracking and updates for travelers.
5. In-app purchase options for upgrades, seat selection, and baggage fees.
6. Option to book and manage all aspects of travel, including hotels and rental cars.
7. Mobile boarding pass and e-ticket functionality.
8. Integration with popular travel and rewards programs.
9. Option to book and manage group and corporate travel.
10. Integration with social media platforms for easy sharing of travel plans with friends and family.

Chapter 9

REFERENCES

[1]Fundamentals of database system, Ramez Elmarsri and Shamkanth B. Navathe, 7th edition,2017, Pearson.

[2]W3Schools https://www.w3schools.com/html

[3] https://www.w3schools.com/php/php_forms.asp

[3] https://www.chat.openai.com/