A PROJECT REPORT
ON
Online Air Ticket booking

Prepared By:-
Shakti Prasadiya
Prashant Vasoya

Submitted to:-

SHREE M. & N. VIRANI SCIENCE COLLEGE.
RAJKOT
The practical training is almost important in understanding the theoretical aspect. Viewing to this important we have prepared this project to enrich our knowledge regarding software management system.

By preparing this report I have understood the need of practical training in the education field. It is easier to work with computerized system. It saves time, effort and space and also increases the efficiency by creating interest.

The entire faculty is highly educated and they are very co-operated with each other and they help all the students so friendly in nature. While studying in this college we get knowledge in all fields.
ACKNOWLEDGEMENT

With immense pleasure we would like to prevent this report on the project assignment on “M.&N. VIRANI SCIENCE COLLEGE”.

We express our sincere thanks to MR. Pradip Vanpariya & MR. Haresh Khachariya, our Project Guide and other faculties of our college who provide us sufficient guidance and allowed us to develop Software & System study and implement it on trial base. Last but not least we are extremely thankful to our parents for giving us their constant and precious guidance, support and motivation during the completion of our project.
# INDEX

<table>
<thead>
<tr>
<th>SR.NO.</th>
<th>HEADING</th>
<th>PAGE NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>01.</td>
<td>PROJECT PROFILE</td>
<td>01</td>
</tr>
<tr>
<td>02.</td>
<td>INTRODUCTION OF PROJECT</td>
<td>02</td>
</tr>
<tr>
<td></td>
<td>• HARDWARE REQUIREMENT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• SOFTWARE REQUIREMENT</td>
<td></td>
</tr>
<tr>
<td>03.</td>
<td>PROJECT PLAN</td>
<td>03</td>
</tr>
<tr>
<td></td>
<td>• NETWORK CHART</td>
<td></td>
</tr>
<tr>
<td>04.</td>
<td>INTRODUCTION OF LANGUAGE</td>
<td>04</td>
</tr>
<tr>
<td>05.</td>
<td>ANALYSIS</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>• FESIBILTY STUDY</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• REQUIREMENT SPECIFACATION &amp; ANALYSIS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DESIGN</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• CODING</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• TESTING</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• MAINTENANCE</td>
<td></td>
</tr>
<tr>
<td>06.</td>
<td>MODULE DESCRIPTION</td>
<td>16</td>
</tr>
<tr>
<td>07.</td>
<td>SYSTEM DIAGRAM</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>• DATA FLOW DIAGRAM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• FLOW CHART</td>
<td></td>
</tr>
<tr>
<td>08.</td>
<td>SOFTWARE MODEL</td>
<td>22</td>
</tr>
<tr>
<td>09.</td>
<td>SDLC</td>
<td>24</td>
</tr>
<tr>
<td>10.</td>
<td>TESTING &amp; DEBUGGING</td>
<td>26</td>
</tr>
<tr>
<td>11.</td>
<td>SNAPSHOTS</td>
<td>29</td>
</tr>
<tr>
<td>12.</td>
<td>DATA DICTIONARY</td>
<td>46</td>
</tr>
<tr>
<td>13.</td>
<td>BIBLOGRAPHY</td>
<td>51</td>
</tr>
</tbody>
</table>
PROJECT PROFILE

Project Title: “onlineairticketreservation”

Front-End: PHP

Back-End: MySql

Project-Guide: Mr. Pradip Vanpariya
Mr. Haresh Khachariya

Submitted By: Shakti Prasadiya
Prashant Vasoya

Submitted To College: Shree M. & N. Virani Science
Rajkot.
INTRODUCTION OF OUR PROJECT

Our project is developed for OnLine Airticket Reservation. Our project is developed using PHP that is server-side scripting language for Windows Xp or Windows 7 user.

Our Project of online Airticket Reservation handles various activities related to Resvering Seats, Searching Flights, Adding new flights, Delete Flights.

Resource Requirements

Software Requirements

- Windows-Xp & O.S 7
- Microsoft Office
- DreamWeaver
- PHP-Mysql

Hardware Requirements

- Intel Processor
- 256mb RAM
- 1GB Hard Disk
- VGA or higher resolution screen supported by MS Window
- CD-ROM Drive
NETWORK CHART
INTRODUCTION OF LANGUAGE

ABOUT HTML

➢ Today we have access a variety of information through web site on internet.

➢ We can access a web site if we have a connection to internet and browser computer.

➢ popular browsers are Microsoft Internet Explorer, Netscape Navigator, opera, etc.

➢ When you connect to web site your browsers display a file in special format by a web server.

➢ A content of file are stored in a special format using “Hyper Text Markup Language” often called HTML.

➢ HTML files are generally stored on webserver or remote computer.

➢ An HTML file or document is interpreted by a web browser and then you can see the web page on your computer.

➢ HTML is invented as simple way of creating a web page.

➢ HTML is a special case of SGML-standard Generalized Markup Language.

➢ HTML provides way of displaying webpage with text and images.

➢ HTML is not a programming language but a markup language.

➢ HTML file is simple text file containing a small markup tags so you can say that HTML consists of data and tags.

➢ HTML file must have .htm or .html extension.
An HTML file can be created using a simple text editor e.g Notepad, Frontpage, Dream weaver.

Every HTML document should start with the tag `<HTML>` and end with the tag `</HTML>`, this tells the browser that this is an HTML document.

Each HTML document includes a head and a body.
**About css**

- Css stands for cascading style sheet.
- Ccss is used to control the layout of multiple web pages all at once.
- Style defines how to display html elements. Style are normally stored in stylesheets.
- Styles were added to HTML 4.0 to solve a problem. External stylesheets are stored in CSS files.

**Advantage of css:**
- Styles solve a common problem.
- Style sheet can save a lot of work.
- Multiple styles will cascade into one

**Types of style sheet**
- 1) external style sheet
- 2) internal style sheet (inside the `<head>` tag)
- 3) inline style (inside an HTML element)

**General syntax**

A css syntax is made up of three parts:
- i) A tag
- ii) An Attribute
- iii) A value

- The tag is normally the HTML element/tag you wish to define, and the attribute you wish to change and each attribute can take a value.
- The property and value are separated by a colon, and surrounded by curly braces:
About javascript

- Javascript is used in millions of webpages to improve the design, validate forms, detect browsers, create cookies, and much more.

- Javascript is the most popular scripting language on the internet, and works in all major browsers, such as internet explore, Mozilla, Firefox, NetScape, and Opera.

- Javascript was designed to add interactivity to HTML pages. Javascript is a scripting language.

- A scripting language is a lightweight programming language. A javascript consists of lines of executable computer code.

- A javascripting is usually embeded directly into HTML pages. Javascript is an interpreted language. Everyone can use javascript without purchasing a license.

Features of javascripts:

1. Javascript gives HTML designers a programming tool
2. Javascript can put dynamic text into an HTML page
3. Javascript can react to events.
4. Javascript can read and write HTML elements.
5. Javascript can be used to validate data
6. Javascript can be used to detect the visitor’s browser.
7. Javascript can be used to create cookies.
➢ By entering the document.write command between the <script> and </script> tags, the browser will recognize it as a javascript command and execute the code line.
About php

- The full form of PHP is “Hypertext Preprocessor”. Its original name was “Personal Home Page”

- PHP is server-side scripting language, which can be embedded in HTML or used as a stand-alone.

- PHP doesn’t do anything about what page looks and sounds like. In fact most of what PHP does is invisible to the end user.

- Someone looking at a PHP page will not necessarily be able to tell that it was not written purely in HTML, because usually the result of PHP is HTML.

- PHP is an official module of Apache Http Server.

- PHP is fully cross-platform, meaning it runs native on several flavours of Unix as well as on Windows and now on Mac OS X.

Advantage of PHP

- Cost
- Ease of use
- HTML-support
- Cross-platform compatibility
- PHP is compatible with the three leading web server.
- Stability
- Speed
PHP Syntax

- A PHP file normally contains HTML tags, just like an HTML file, and some PHP scripting code.

- A PHP Scripting block starts with <?php and ends with ?>

- A PHP scripting block can be placed anywhere in the document.

- Each line code in PHP must end with a semicolon. The semicolon is a separator and is used to distinguish one set of instructions from another.

Data types:

- PHP stores whole numbers in a platform-dependent range. This range is typically that of 32-bit signed integers. Unsigned integers are converted to signed values in certain situations; this behavior is different from other programming languages.

- Integer variables can be assigned using decimal (positive and negative), octal, and hexadecimal notations.

- Floating point numbers are also stored in a platform-specific range. They can be specified using floating point notation, or two forms of scientific notation.

- PHP has a native Boolean type that is similar to the native Boolean types in Java and C++. Using the Boolean type conversion rules, non-zero values are interpreted as true and zero as false, as in Perl and C++.

- The null data type represents a variable that has no value. The only value in the null data type is NULL.
✓ Variables of the "resource" type represent references to resources from external sources. These are typically created by functions from a particular extension, and can only be processed by functions from the same extension; examples include file, image, and database resources.

✓ Arrays can contain elements of any type that PHP can handle, including resources, objects, and even other arrays. Order is preserved in lists of values and in hashes with both keys and values, and the two can be intermingled.

✓ PHP also supports strings, which can be used with single quotes, double quotes.

✓ The Standard PHP Library (SPL) attempts to solve standard problems and implements efficient data access interfaces and classes.
PHP Functions:

- PHP has hundreds of base functions and thousands more from extensions.
- These functions are well documented on the PHP site, but unfortunately, the built-in library has a wide variety of naming conventions and inconsistencies.
- PHP currently has no functions for thread programming.

About DreamWeaver

- Adobe Dreamweaver is a web development application originally created by Macromedia, and is now developed by Adobe Systems, who acquired Macromedia in 2005.

- Dreamweaver is available for both Mac and Windows operating systems. Recent versions have incorporated support for web technologies such as CSS, JavaScript, and various server-side scripting languages and frameworks including ASP, ColdFusion, and PHP.
Features:

✓ Although a hybrid WYSIWYG and code-based web design and development application, Dreamweaver's WYSIWYG mode can hide the HTML code details of pages from the user, making it possible for non-coders to create web pages and sites.

✓ One criticism of this approach is that it has the potential to produce HTML pages whose file size and amount of HTML code is larger than an optimally hand-coded page would be, which can cause web browsers to perform poorly.

✓ This can be particularly true because the application makes it very easy to create table-based layouts.

✓ In addition, some website developers have criticized Dreamweaver in the past for producing code that often does not comply with W3C standards, though recent versions have been more compliant.

✓ Dreamweaver 8.0 performed poorly on the Acid2 Test, developed by the Web Standards Project. However, Adobe has focused on support for standards-based layout in recent and current versions of the application, including the ability to convert tables to layers.

✓ Dreamweaver allows users to preview websites in locally-installed web browsers.

✓ Dreamweaver can utilize third-party "Extensions" to enable and extend core functionality of the application, which any web developer can write (largely in HTML and JavaScript).

✓ Dreamweaver is supported by a large community of extension developers who make extensions available (both commercial and free) for most web development tasks from simple rollover effects to full-featured shopping carts.
About Mysql

MySQL is an open source, SQL Relational Database Management System (RDBMS) that is free for many uses (more detail on that later).

Early in its history, MySQL occasionally faced opposition due to its lack of support for some core SQL constructs such as subselects and foreign keys.

Ultimately, however, MySQL found a broad, enthusiastic user base for its liberal licensing terms, perky performance, and ease of use.

Its acceptance was aided in part by the wide variety of other technologies such as PHP, Java, Perl, Python, and the like that have encouraged its use through stable, well-documented modules and extensions.

MySQL has not failed to reward the loyalty of these users with the addition of both subselects and foreign keys as of the 4.1 series.

Databases in general are useful, arguably the most consistently useful family of software products—the “killer product” of modern computing.

Like many competing products, both free and commercial, MySQL isn’t a database until you give it some structure and form.

You might think of this as the difference between a database and an RDBMS (that is, RDBMS plus user requirements equals a database).
ANALYSIS

✓ For any system, it is necessary to gather and interpret facts; diagnosing the problems and using the information recommend improvement or enhancements in the system.

✓ Analysis is concerned with devising a precise, concise, understandable, and correct model of the real world.

✓ Before building anything complex, such as a house, computer program, or a hardware software system, the builder must understand the requirements and the real world environment in which it will exist.

✓ The purpose of object oriented analysis is to model in real world system so that it can be understood.

✓ To do this, you must examine requirements, analyze their implications.

✓ You must abstract importance real world features first and defers small details until later.

✓ The successful analysis model states what must be done, with out restricting how it is done and avoid implementation decision.

✓ The result of analysis should understand the problems as a preparation for design.

✓ Analysis begins with a problem statement generated by clients and possibly the developers the statement may be incomplete or informal,
analysis makes it more precise and exposes ambiguities and inconsistencies.

✓ The problem statement must be understood the analysis model is a precise, concise representation of problem that permits answer the question and building a solution.

✓ The analysis model addresses the three aspects of object:
  • static structure (object model),
  • sequencing of the interactions (dynamic model) and
  • data transformation (functional model).
Feasibility Study:

✔ The Main aim of feasibility study is to determine whether developing the product is financially and technically feasible
✔ The feasibility study involves analysis of the problem and collections of data which would be input to the system, the processing required to be carried out on these data, the output data required to be produced by the system as well as study of various constraints on the behavior of the system, the collected data are analyzed to arrive at following:

- An abstract definition of the problem
- Formulation of the different solution strategies
- Examination of alternative solution strategies and their benefits indicating and time in respect of each of the alternative solution
- A cost/Benefit analysis is performed to determine which solution is the best at this stage, it may also be determine whether any of the solutions is not feasible due to high cost, resource constraints or extraordinary technical reasons.
Requirements Analysis and specification

✓ The aim of the requirement analysis and specification phase is to understand the exact requirements of the customer and to document them properly.

✓ This phase consists of two distinct activities:
  • Requirement Analysis
  • Requirement Specification

✓ The goal of the requirement analysis is to collect and analyze all related data and information with a view to understanding the customer requirements clearly and weeding out inconsistencies and incompleteness in these requirements.

✓ During requirement specification, the user requirements are properly organized and documented in a SOFTWARE REQUIREMENT SPECIFICATION (SRS) document.

✓ The SRS document addresses the functional requirements, the nonfunctional requirements, and the special requirements on the maintaince and development of the software products, if any.

✓ During this phase, the user’s manuals and the system test plan are also produced.

✓ The SRS document must specify all functional and performance requirements, the formals of inputs, outputs, and required standards, to be followed, and all design constraints.

✓ The requirements at this stage are written using the end-user terminology.
Design
✓ The goal of the design phase is to transform the requirements specifications into a structure that is suitable for implementation in some programming language.
✓ There are two distinct design approach being used by industries
  • Traditional design approach
  • Object oriented design

Coding
✓ The purpose of this phase of software development is to translate the software design into source code
✓ During the implementation phase, each component of the design is implemented as a program module.

Testing
• Unit testing:
  o The purpose of unit testing is to determine the correct working of the individual modules.
• Integration testing
  o During this testing the different modules are integrated in a planned manner
  o In integration testing, the system is partially tested.
• System testing
  o The goal of system testing is to ensure that the developed system functions according to its requirements as specified in the SRS document.
**Maintenance**

- Maintenance involves performing any one or more of the following three kinds of activities.
  1. Correcting errors that were not discovered during the product development phase. This is called corrective Maintenance.
  2. Improving the implementation of the system and enhancing the functionalities of the system according to the customer’s requirements. This is called perfective Maintenance.
  3. Porting the software to a new environment, e.g. to a new computer or to a new operating system, this called adaptive Maintenance.
Module Description

Two Modules:
1) Client Module
2) Admin Module

Client Module

Login & Registration process
- User has to login himself with this system, and they can only use this system.
- If user has no login then first user have to register. After this registration user have to login and after that user can use this system.
- The third person is the Administrator who knows all the activities done by users and can control the database.
- This registration is free for all the users.
- Our registration form has following features:
  - If same username already exists or invalid username/email-id is entered then error message is displayed.
• User have to fill all the fields properly which is in the form.

**Reserve the Ticket:**

✔ After the login the process the reservation form will be open.
✔ User have to fill that form, whatever city is selected related flight page will open.
✔ If user selects the “rond trip” then return flight also will appear.
✔ If user selects the ”one trip” then return flight will not appear.
✔ From that page user have to select the flight and book the ticket.
✔ After booking the ticket related information will displayed.
✔ This information is seat no, total price etc..

**Admin Module**

Administrator is the one person who knows all the things about the system. He also monitors the activities that user’s are doing. There are key activities that administrator can do are as below

♦ Database control
♦ View all the flights
♦ Add new flight
♦ Update flight
♦ Delete the flight
♦ Search the flight
♦ Know how many flights are reserved
**SYSTEM DIAGRAM**

**DFD (Data Flow Diagram):**

The DFD (Data Flow diagram) is also known as Bubble chart. It is a simple graphical notation that can be used to represent a system in terms of the input data to the system. Various processing carried out on this data, and the output data generated by the system. The main reason the DFD technique is so popular is on the account of the fact that it is very simple formulism—it is simple to understand and use. A DFD model uses a very limited number of primitive symbols.

1. **An External entity**

2. **Process**

3. **Data Store**

4. **Output**

5. **Data flow**
DATA FLOW DIAGRAM

0 Level Diagram :-

Login form → Registration → Booking Process → Registration Complet
1 Level Diagram :-

- **Login form**
  - Login successfully
  - No Flights
  - Login form
  - gives related flight

- **Registration**
  - not login
  - then
  - Store in database

- **Reservation form**
  - Select the Flight
  - Booking process
  - Allocate the seatno
  - Information about booked tickets.
FLOWCHART

Enter Login

check userid & password

Yes

Reservation process

Check flight available or not

Check seat available

Select the flight

Book the flight
SOFTWARE MODEL

Which Model We Used

Waterfall Model

- This model is the simplest model for software model.
- This model defines activities in terms of phases, in linear fashion.
- Waterfall model divides the life cycle of a software development process into the following figure.

![Waterfall Model Diagram]

- Feasibility Study
- Requirement Analysis
- Design
- Coding & Unit testing
- Integration & System
- Testing
- Maintainance
This model is named “waterfall model” because its diagrammatic representation resembles a cascade of waterfalls.

The different phases starting from the feasibility study to the integration and system testing are known as the development phases.

Each phase typically requires relatively different amounts of effort

Among all the lifecycle phases, the maintenance phase typically requires the maximum effort.

In development phases, the integration and system testing phase usually needs the maximum effort.

Each phase of the lifecycle has a well-defined starting and ending point. So, the development engineers know precisely when to stop a phase and start the next phase.

Limitation of Waterfall Model:-

Waterfall model assumes that the requirements of a system can be frozen before the system design. This is possible for system designed to automate an existing system. But for new system, determining the requirement is difficult.

Freezing the requirements usually requires choosing hardware. As large project might take few years to complete.

Waterfall model stipulates that the requirements be completely satisfied before the rest of the development proceeds.

It is document driven process that require formal document at the end of each phase. This approach makes process documentation heavy and it is not suitable for
many applications. For interactive applications elaborate documentation of user interface, is not feasible.
SDLC

✓ The six stages of the SDLC are designed to build on one another, taking the output from the previous stage, adding additional effort, and producing results that leverage the previous effort and are directly traceable to the previous stages.

✓ This topdown approach is intended to result in a quality product that satisfies the original intentions of the user.

✓ Too many software development efforts go awry when the development team and user personnel get caught up in the possibilities of automation.

✓ Instead of focusing on high priority features, the team can become mired in a sea of “nice to have” features that are not essential to solve the problem, but in themselves are highly attractive.

✓ This is the root cause of a large percentage of failed and/or abandoned development efforts, and is the primary reason the development team utilizes the SDLC.
Phases of SDLC

Project planning

Requirement definition

Design

Development

Integration & test

Installation & Acceptance
TESTING & DEBUGGING

System testing is a critical process that takes as much 50% of the system development time. The common view of testing held by users is that it is performed to prove that there are no errors in the program.

Testing is too important for success of the system. If all parts of the system are correct, the goal will be achieved successfully. System testing is also called the “put it to all together” phase where the elements of the system are put together to examine its validity and reliability.

Hardware, software, manpower and live data are combined in an effort to produce the necessary result.

The variety of tests are conducted as:

- Output Testing
- Hardware and Software testing
- Recovery and security

The test strategy will include five different types of testing as described below

**Logical Testing**

This is used to test every aspect of each form and query as soon as it is implemented. In this test results are compared with expected results.

**Functional Testing**

Each menu item should be tested in the form to ensure that no functionalities have been missed out.

**System Testing**
When the system is complete, the whole range of test should be carried out again to ensure that no errors have been introduced.

**Recovering Testing**

The computer will be re-booted while the database is open to ensure that data is not lost or corrupted in the event of power-failure.

**Acceptance Testing**

The user will be involved and asks to test all the capabilities of the system to ensure that all required functions are present and working in the manner expected. The user involvement gives the final look to System.

**Test Plan**
Analysis and design department makes all analysis for the system and forward the test cases, flow of the system and scope of the system and scope of the system to development department.

Development department implements all the forms and sends a to Quality Assurance department for testing.

Quality department checks the form for test cases and also performs integrated testing. If any error or bug found it returns to development department otherwise sends to analysis department.

Development department receives bug, reports, and after completing currently running modules solves those bug reports.

Analysis department receives error free forms and stores it permanent.
✓ After completing form level testing, system integrity testing starts. Thus, the system is tested per cycle and then it is developed further.
SNAPSHOTS

Client-side Snapshots

Welcome Page

- This is our welcome page.
- In this page menus are displayed.
- Using this menu you can refer this site
This is our login page.

Through this page user will login, if he has no login then first he have to register himself by filling the register form.

Register form will open by clicking on register link.
Registration form

- This is our registration form.
- To register user have to fill this form properly.
Reservation form

- This is our reservation form for trip.
- If the user wants to reserve a flight, they must fill out this reservation form properly.
This is our detail page
In this page all flights related information is available
Information like flightno, origin, destination, departure time, arrival time, tickets rate is shown.
Select Trip

- Display all flight in your choice
- This page shows related flights which we will select in Trip form.
Booking page

- This is booking page.
- In this page selected trip will display and my book now button is shown
Through this page user will get all the information related flight which he will book and Download your ticket
Contact –Us

- this is our contact menu.
- In this menu developers related information is appeared.
AdminSide Snapshots

Admin Login Page

- This is our admin login page.
- Through this page user can login.
- After login the page admin can do anything.

Admin Homepage
Add New Trip

- In this page admin can insert a new Trip
- This trip display in client side
Update Trip

Through this page admin can do editing means can update the Trip.

Admin can also delete the Trip.
This page is display booking user in seat, date time and amount
Trip Date And Time

- Display admin in insert trip on date, time and arrival time
Plane Detail

- Display plane company, capacity and plane image
- Insert, update and delete plane company, capacity and image
# Data Dictionary

## Database Table

<table>
<thead>
<tr>
<th>Table</th>
<th>Action</th>
<th>Records</th>
<th>Type</th>
<th>Collation</th>
<th>Size</th>
<th>Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>account_detail</td>
<td></td>
<td>10</td>
<td>MyISAM</td>
<td>latin1_swedish_ci</td>
<td>2.2 KIB</td>
<td></td>
</tr>
<tr>
<td>airoplane</td>
<td></td>
<td>3</td>
<td>MyISAM</td>
<td>latin1_swedish_ci</td>
<td>2.1 KIB</td>
<td></td>
</tr>
<tr>
<td>booking_detail</td>
<td></td>
<td>11</td>
<td>MyISAM</td>
<td>latin1_swedish_ci</td>
<td>2.5 KIB</td>
<td></td>
</tr>
<tr>
<td>city</td>
<td></td>
<td>34</td>
<td>MyISAM</td>
<td>latin1_swedish_ci</td>
<td>2.7 KIB</td>
<td></td>
</tr>
<tr>
<td>contact</td>
<td></td>
<td>0</td>
<td>MyISAM</td>
<td>latin1_swedish_ci</td>
<td>1.0 KIB</td>
<td></td>
</tr>
<tr>
<td>country</td>
<td></td>
<td>5</td>
<td>MyISAM</td>
<td>latin1_swedish_ci</td>
<td>2.1 KIB</td>
<td></td>
</tr>
<tr>
<td>login</td>
<td></td>
<td>2</td>
<td>MyISAM</td>
<td>latin1_swedish_ci</td>
<td>2.0 KIB</td>
<td></td>
</tr>
<tr>
<td>news</td>
<td></td>
<td>0</td>
<td>MyISAM</td>
<td>latin1_swedish_ci</td>
<td>1.0 KIB</td>
<td></td>
</tr>
<tr>
<td>state</td>
<td></td>
<td>14</td>
<td>MyISAM</td>
<td>latin1_swedish_ci</td>
<td>2.3 KIB</td>
<td></td>
</tr>
<tr>
<td>trip</td>
<td></td>
<td>10</td>
<td>MyISAM</td>
<td>latin1_swedish_ci</td>
<td>2.4 KIB</td>
<td></td>
</tr>
<tr>
<td>tripdetail</td>
<td></td>
<td>10</td>
<td>MyISAM</td>
<td>latin1_swedish_ci</td>
<td>2.4 KIB</td>
<td></td>
</tr>
<tr>
<td>user</td>
<td></td>
<td>10</td>
<td>MyISAM</td>
<td>latin1_swedish_ci</td>
<td>3.3 KIB</td>
<td></td>
</tr>
<tr>
<td><strong>12 table(s)</strong></td>
<td></td>
<td><strong>105</strong></td>
<td></td>
<td></td>
<td><strong>26.1 KIB</strong></td>
<td><strong>0 B</strong></td>
</tr>
</tbody>
</table>

---

**Note:** The table shows the structure of the database with various tables, their records, types, collations, sizes, and overheads. The database is named `airlines`.
Admin Login Table

Registration Table
### Trip Details Table

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Collation</th>
<th>Attributes</th>
<th>Null</th>
<th>Default</th>
<th>Extra</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>int(11)</td>
<td></td>
<td></td>
<td>No</td>
<td>None</td>
<td>auto_increment</td>
<td></td>
</tr>
<tr>
<td>Trip_Name</td>
<td>varchar(200)</td>
<td>latin1_swedish_ci</td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>From_Country</td>
<td>varchar(200)</td>
<td>latin1_swedish_ci</td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>From_State</td>
<td>varchar(200)</td>
<td>latin1_swedish_ci</td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>From_City</td>
<td>varchar(200)</td>
<td>latin1_swedish_ci</td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>To_Country</td>
<td>varchar(200)</td>
<td>latin1_swedish_ci</td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>To_State</td>
<td>varchar(200)</td>
<td>latin1_swedish_ci</td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>To_City</td>
<td>varchar(200)</td>
<td>latin1_swedish_ci</td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Price</td>
<td>int(8)</td>
<td></td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Booking Detail Table

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Collation</th>
<th>Attributes</th>
<th>Null</th>
<th>Default</th>
<th>Extra</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>int(11)</td>
<td></td>
<td></td>
<td>No</td>
<td>None</td>
<td>auto_increment</td>
<td></td>
</tr>
<tr>
<td>Trip_Name</td>
<td>varchar(8)</td>
<td>latin1_swedish_ci</td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>seat_no</td>
<td>int(8)</td>
<td></td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>booking_date</td>
<td>varchar(20)</td>
<td>latin1_swedish_ci</td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ticket_nm</td>
<td>varchar(200)</td>
<td>latin1_swedish_ci</td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>amt</td>
<td>int(8)</td>
<td></td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>booking_time</td>
<td>varchar(200)</td>
<td>latin1_swedish_ci</td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_id</td>
<td>varchar(8)</td>
<td>latin1_swedish_ci</td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Trip Table

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Collation</th>
<th>Attributes</th>
<th>Null</th>
<th>Default</th>
<th>Extra</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>int(11)</td>
<td></td>
<td>No</td>
<td>None</td>
<td>None</td>
<td>auto_increment</td>
<td></td>
</tr>
<tr>
<td>trip_id</td>
<td>varchar(25)</td>
<td>latin1_swedish_ci</td>
<td>No</td>
<td>None</td>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>plane_id</td>
<td>varchar(25)</td>
<td>latin1_swedish_ci</td>
<td>No</td>
<td>None</td>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>on_date</td>
<td>varchar(50)</td>
<td>latin1_swedish_ci</td>
<td>No</td>
<td>None</td>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>time</td>
<td>varchar(50)</td>
<td>latin1_swedish_ci</td>
<td>No</td>
<td>None</td>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arrival_time</td>
<td>varchar(50)</td>
<td>latin1_swedish_ci</td>
<td>No</td>
<td>None</td>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Airoplane Table

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Collation</th>
<th>Attributes</th>
<th>Null</th>
<th>Default</th>
<th>Extra</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>int(11)</td>
<td></td>
<td>No</td>
<td>None</td>
<td>None</td>
<td>auto_increment</td>
<td></td>
</tr>
<tr>
<td>Plane_Company</td>
<td>text</td>
<td>latin1_swedish_ci</td>
<td>No</td>
<td>None</td>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Capacity</td>
<td>int(8)</td>
<td></td>
<td>No</td>
<td>None</td>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Image</td>
<td>varchar(100)</td>
<td>latin1_swedish_ci</td>
<td>No</td>
<td>None</td>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Account Detail Table

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Collation</th>
<th>Attributes</th>
<th>Null</th>
<th>Default</th>
<th>Extra</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>int(11)</td>
<td></td>
<td></td>
<td>No</td>
<td>None</td>
<td>auto_increment</td>
<td></td>
</tr>
<tr>
<td>booking_id</td>
<td>int(8)</td>
<td></td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>amt</td>
<td>int(8)</td>
<td></td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>total</td>
<td>int(8)</td>
<td></td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Country Table

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Collation</th>
<th>Attributes</th>
<th>Null</th>
<th>Default</th>
<th>Extra</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>tinyint(4)</td>
<td></td>
<td></td>
<td>No</td>
<td>None</td>
<td>auto_increment</td>
<td></td>
</tr>
<tr>
<td>country</td>
<td>varchar(20)</td>
<td>latin1_swedish_ci</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

State Table
### City Table

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Collation</th>
<th>Attributes</th>
<th>Null</th>
<th>Default</th>
<th>Extra</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>tinyint(4)</td>
<td></td>
<td>No</td>
<td>None</td>
<td>auto_increment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>city</td>
<td>varchar(50)</td>
<td>latin1_swedish_ci</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>stateid</td>
<td>tinyint(4)</td>
<td></td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>countryid</td>
<td>tinyint(4)</td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
BIBLIOGRAPHY

REFERENCE BOOK

1) PHP AND MYSQL :
   BIBLE
2) SOFTWARE ENGEERING :
   ROGER.S. PRESSMAN

Web Site

- www.google.com
- www.w3schools.com
- www.fly2kingfisher.com
- www.airindia.com