CHAPTER 1

INTRODUCTION

Railway management system is a Database application which is helpful for maintaining the railway system. This database becomes useful for the admins to schedules their details. If any modifications occurs in schedules means the passenger may came to know it.

This project is about creating the database for railway system. The aim is to design and develop a database maintaining the records of different trains, passengers, tracks, stations, schedule and routes. The project also consists of station names, the tracks that connect those stations, schedules of the train and the information of the station.

This mini-project is implemented using HTML and PHP. Operations supported by the application are insert, delete, update and retrieve. The database that we created is one that holds the data about a railway system

1.1 HTML

HTML, which stands for Hyper Text Mark-Up Language, is the language for describing structured documents as well as the language used to create web pages in the Internet. The language is based on an existing, international formatting standard SGML, Standard Generalized Mark-Up Language, which is used for text processing.

HTML documents are nothing but web pages which contains HTML tags and plain text. The purpose of a web browser is to read HTML documents and display them as web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page.

History

HTML, which stands for Hyper Text Mark-Up Language, is the language for describing structured documents as well as the language used to create web pages in the Internet. The language is based on an existing, international formatting standard SGML, Standard Generalized Mark-Up Language, which is used for text processing. HTML is a simplified version of SGML.

Tools in order to use HTML

Tools help us in process of creating HTML document. Some are as follows

- TEXT EDITOR: To create the HTML code we require a text editor or a word processor. Such as, Notepad, WordPad. We are using notepad++ in developing this project.

- WEB BROWSER: The code created by an editor should be executed. This operation can be performed with help of a web browser. Such as Internet Explorer, Netscape navigator, Mozilla Firefox etc.
- GRAPHICS SOFTWARE: To include picture we require a graphic software like Adobe Photoshop.
- Web server: To make the document is to be available on the internet then, we will have to host it on a web server.

1.1.1 Significant Language Features

HTML files are written in ACSII text, so the user can use any text editor to create his/her web page, though a browser of one sort or another is necessary to view the web page. HTML is case insensitive with its language commands. The characters within the document, however, are case sensitive. The language consists of various "tags" which are known as elements. These allow the browser to understand (and put into the desired/specified format) the layout, background, headings, titles, lists, text and/or graphics on the page. The elements are classified according to their function in the HTML document. There are head elements and body elements. The head elements identify properties of the entire document, while body elements actually mark text as content and show a change in the appearance in one way or another. Most elements have a beginning and an ending which encompass the text the user wishes to mark with the tag. All HTML documents must begin with the element and end with the element. Some of the other elements which may be used are tags to create lists--both ordered lists as well as unordered lists. The user may also create larger or smaller, bolder, italicized, or underlined text. Attributes may be used along with the elements. These perform functions such as placement of text, indication of the source files of images, and identification of links to the document or part of the document.

1.1.2 HTML Code

Copy and paste the following HTML code into your newly open text file. Which just displays hello world..

```
<html>
<header><title>This is title</title></header>
```

```
<body>
  This is sample text...
  <!-- We use this syntax to write comments -->
  <!-- Page content and rest of the tags here.... -->
  <!-- This is the actual area that gets shown in the browser →
Hello world
</body>
</html>
```

1.1.3 HTML TAGS

HTML tags are keywords surrounded by angle brackets like <html>. These are in pair format such that every first tag in pair is start tag where as second tag is end tag. These start and end tags are also called as opening tags and closing tags respectively.

Tags Used In Project

The HTML tags are the basis, in order to do this Project. By using some of the important and basically taught tags are used in this Project. Here are some of the tags used in making the Project called RAILWAY MANAGEMENT SYSTEM.

HTML Attributes

Attributes provide additional information about HTML elements.

- HTML elements can have **attributes**
- Attributes provide **additional information** about an element
- Attributes are always specified in **the start tag**
- Attributes come in name/value pairs like: **name="value"** Some basic text formatting HTML tags are listed:

Tag	**Description**
<html>	Defines an HTML document
<body>	Defines the document's body

<h1> to <h6>	Defines header 1 to header 6
 	Inserts a single line break
<b>	Defines bold text

Some of the HTML tags used to create a table are listed:

In an HTML file we can create tables with the Table tags, which in turn will render the browser to display the table in the web page.

Tag	**Description**
<table>	Defines a table
<th>	Defines a table header
<tr>	Defines a table row
<td>	Defines a table cell

A Simple Form

A form in a web page allows the users to input various data online. In an HTML document; forms can be created with the Form tags. In the following table, some basic Form tags are listed:

Tag	**Description**
<form>	Defines a form for user input
<input>	Defines an input field
<label>	Defines a label to a control
<select>	Defines a selectable list
<option>	Defines an option in the drop box
<button>	Defines a push button

Image Tags

In an HTML document we can insert and display images by using the image tags.
In the following table, some basic Image tags are listed:

Tag	Description
<IMG SRC attributes>	Defines an image

The "src" attribute is used to display an image on a web page. "src" stands for "source", and its value is the url of the image to be displayed on the page. The url indicates the location where the image is stored. Attributes may be height, width, align so on.

Background colour

Using bgcolour attribute this can be done. This is body tag attribute. Six digit hexadecimal code represent the colours.

Syntax: <body text="text_color" bgcolor = "background_color">

Anchor tag

Anchor tag is used to link two or more different web pages.

Ex: click here where href stands for hyper link reference.

Areas of Application

HTML only has one area of application at this time and that is the development of web pages. However, not all browsers support all the tags in all versions of HTML. Because of this, it is wise not to design your web page for a specific browser, because what may look fantastic on your browser has no guarantee of looking great on someone else's browser.

1.2 PHP

PHP is a general-purpose scripting language that is especially suited to server-side web development, in which case PHP generally runs on a web server. Any PHP code in a requested file is executed by the PHP runtime, usually to create dynamic web page content or dynamic images used on websites or elsewhere.

PHP originally stood for Personal Home Page, but it now stands for the recursive backronym PHP. Hypertext Pre-processor. PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management system and web frameworks.

PHP developer

PHP developers develop programs, applications, and web sites using the dynamic scripting language PHP. PHP is known for web development and business applications. Depending on job function, PHP developers may be classified as software developers or web developers.

Tags Description

<?php	to open PHP section
?>	to close PHP sections
ECHO	prints the lines

1.3 DATABASE

A database is a collection of information that is organized so that it can easily be accessed, managed, and updated. In one view, databases can be classified according to types of content: bibliographic, full-text, numeric, and images. **Database** software systems are programmed in SQL, and examples include Microsoft SQL Server, MySQL, Oracle SAP HANA and FoxPro.

A DBMS system is also required to protect the integrity of data and provide its security. A database management system (**DBMS**) is system software for creating and managing databases.

The **DBMS** provides users and programmers with a systematic way to create, retrieve, update and manage data.

1.4 MYSQL

MySql is a powerful database. It's very good and free of charge. Many developers in the world selected mysql and php for developing their website.

The MySQL database has become the world's most popular open source database because of its consistent fast performance, high reliability and ease of use. It's used in more than 6 million installations ranging from large corporations to specialized embedded applications on every continent in the world. (Yes, even Antarctica!)

Not only is MySQL the world's most popular open source database, it's also become the database of choice for a new generation of applications built on the LAMP stack (Linux, Apache, MySQL, PHP / Perl / Python.) MySQL runs on more than 20 platforms including Linux, Windows, OS/X, HP-UX, AIX, Netware, giving you the kind of flexibility that puts you in control.

Whether you're new to database technology or an experienced developer or DBA, MySQL offers a comprehensive range of certified software, support, training and consulting to make you successful.

1.5 WAMP

The acronym WAMP refers to a set of free (open source) applications, combined with Microsoft Windows, which are commonly used in Web server environments. The WAMP stack provides developers with the four key elements of a Web server: an operating system, database, Web server and Web scripting software. The combined usage of these programs is called a server stack. In this stack, Microsoft Windows is the operating system (OS), Apache is the Web server, MySQL handles the database components, while PHP, Python, or PERL represents the dynamic scripting languages.

1.6 Notepad

Notepad is a generic text editor included with all versions of Microsoft Windows that allows you to create, open and read plaintext files. If the file contains special formatting or is not a plaintext file, it cannot be read in Microsoft Notepad. The image is a small example of what the Microsoft Notepad may look like while running.

1.7 Web Browser

Google Chrome is a free web browser from Google which we are using here. With its clean design and advanced features, Chrome has quickly become one of the most popular web browsers worldwide. In this lesson, we'll talk about the features of Google Chrome, how to download and install Chrome to your computer, and how to sign in to Chrome using a Google account.

CHAPTER 2

System Analysis and Design

In this chapter, a complete description of the project development is discussed. The requirements of the project identified are showcased. The database design is done Using HighLevel Conceptual Data Models

2.1 Requirement Analysis

Following requirements were identified during the requirement collection and analysis.

1. Admin can login into the railway sytem by using the username and password.
2. Admin can do all the operation like insertion,deletion and updation.
3. The database include passengers(P-id) each passenger has a unique details like p_name,age,gender,coach,coach_type,seat_no and train-id.
4. The Admin can modify the details of train includes(t_id,t_name,t_type).
5. Admin can reschedule the train movement by considering moves_on(t_id,route_no).
6. Admin can schedule the train moving timings by using schedules(schedule_id, source, destination, start_time, end_time, train_id, route_no).
7. Passengers can search the train schedule in particular time by using source and destination place.

Feasibility Study

The feasibility study carried out showed that the requirements that were to be included could be provided by the use of RDBMS software such as MySQL which is available as an open source and for the front end HTML pages with processing capability provided by the Scripting language such as PHP and Javascript.

2.2 ER-Diagram

Following is the conceptual representation of the requirements identified as an ER Diagram.

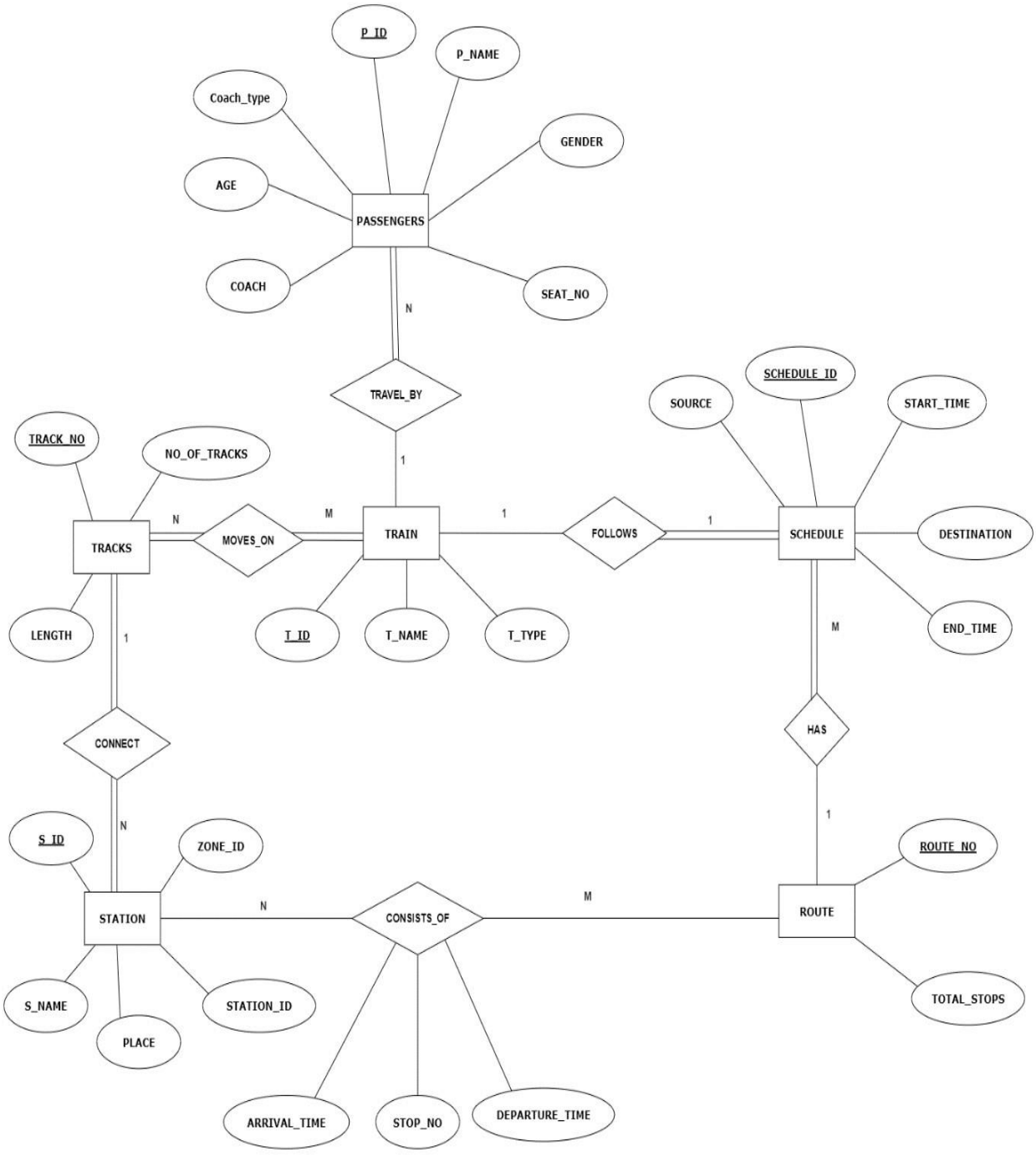
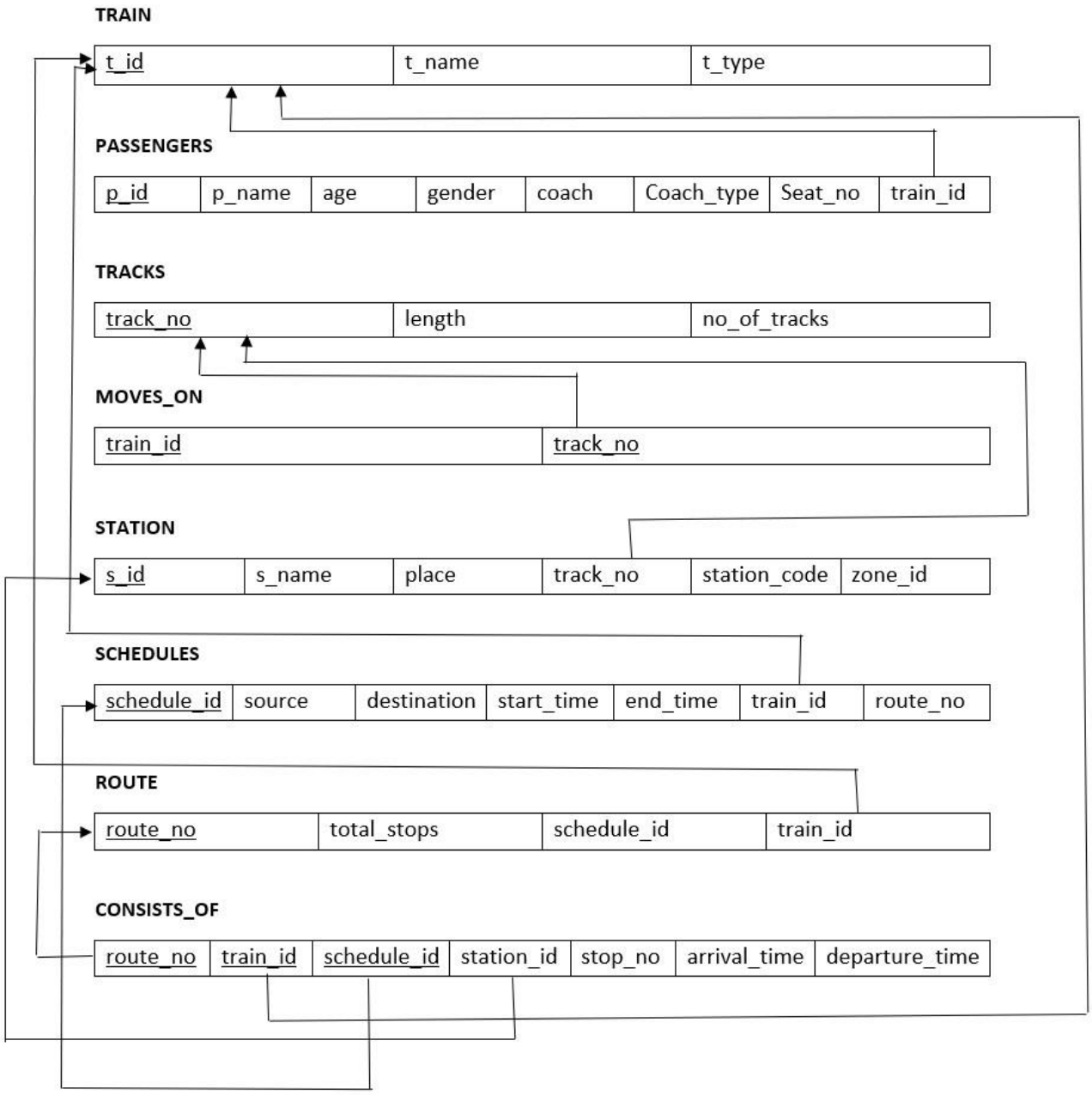


Figure 2.2 ER Diagram for Railway management system

2.3 Relational Schema

The relational schema diagram has been derived from the ER-Diagram in Figure 2.2 using the ER Relational mapping algorithm



2.3 Relational Schema Diagram Showing the Primary key and Foreign key relationships

2.4 Functional Requirements

Functional requirements of a software project interpret the function of a part. It defines its functions, input and output. The typical functional requirements include:

Application contains 2 modules:

- Admin module
- Customer module

Admin module

- Admin can insert passengers details, train details, tracks details, route details, schedules details, moves on details, station details, consists of details.
- Admin can update the train details.
- Admin can able to reschedules the train details.
- Admin can able to delete any train schedule in particular time.
- Admin can able to update the passenger seat number.
- Admin can able to alter the train routes.

Passenger module

- Passenger can search the train schedule details by using source and destination place.

2.5 Non- Functional Requirements

A non-functional requirement specifies the canon of the articular process not the particular judgment of the system and particular behavior of the process. Non-functional requirements define how the system works.

For a railway management system, the following non-functional requirements would need to be considered:

- Performance: The system must be designed for high performance and scalability to accommodate large amounts of data and transactions.
- Reliability: The system must be highly reliable, with built-in redundancy and backup mechanisms to ensure the integrity of data and minimize downtime.
- Usability: The system must be easy to use and understand, with intuitive interfaces and clear documentation.

- Data consistency: The system must enforce data consistency, with built-in validation rules to ensure data accuracy and completeness.
- Data security: The system must be secure, with mechanisms in place to prevent unauthorized access and protect sensitive data
- Maintainability: The system must be designed for maintainability, with clear documentation and a modular architecture for ease of updates and modifications.

2.6 Use Case Diagram

The use case diagrams usually refer to behavioral diagrams helps people to understand the interaction between user and system. Use case diagram identify different users of the system.It is used to define some set of actions, which is called as use cases.Actors are the result of some valuable use cases.Use case figures are also called as unified modeling language.

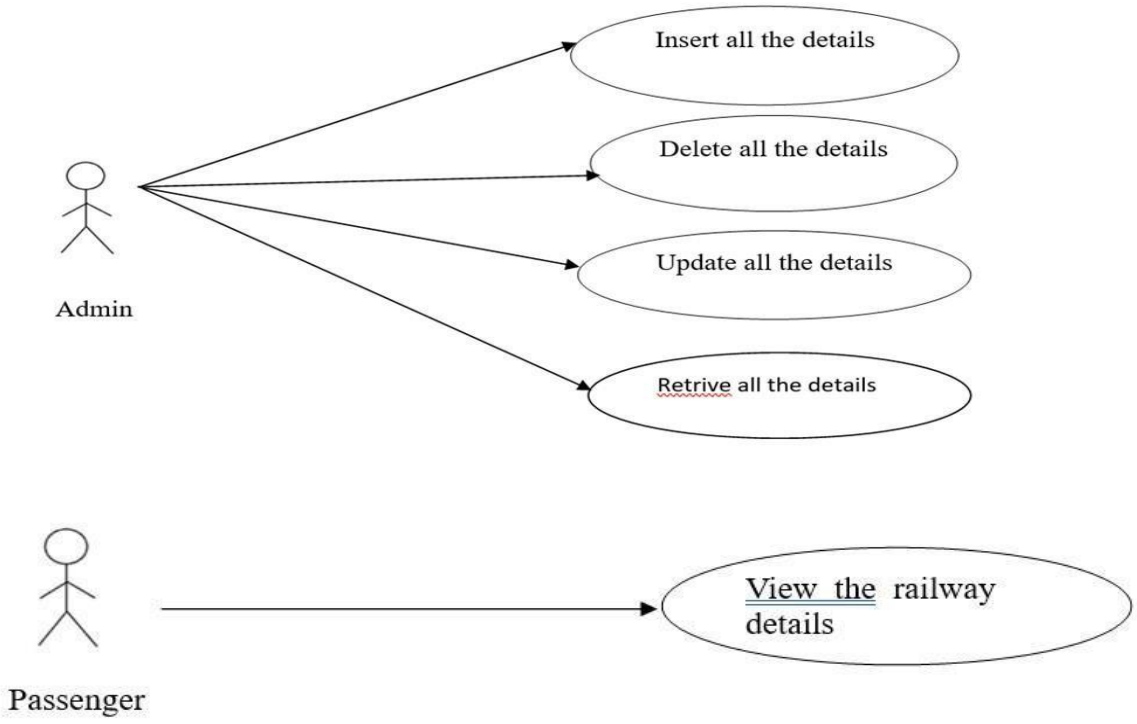


Figure 2.6 Use Case Diagram

CHAPTER 3

SYSTEM IMPLEMENTATION

3.1 Database Design

```
create table train
(
t_id int primary key, t_name
varchar(20),
t_type varchar(20)
);

create table passengers
(
p_id int primary key,
p_name varchar(20),
age int, gender
varchar(20), coach int,
coach_type varchar(20),
seat_no varchar(20),
train_id int,
foreign key(train_id)references train(t_id)
);

create table tracks
(
track_no int primary key ,  length
varchar(20),
```

```
no_of_tracks int
);

create table station
(
s_id int primary key,
s_name varchar(20),
place varchar(20),
track_no int,
station_code
varchar(20),
zone_idvarchar(20),
foreign key(track_no)
referencestracks(track_n
o)
);

create table moves_on  ( train_id int,  track_no
int, primary key(train_id,track_no), foreign
key(train_id) references train(t_id),  foreign
key(track_no) references tracks(track_no)
);

create table schedules
(
schedule_id int,  source varchar(20),  destination
varchar(20),  start_time  varchar(20),   end_time
varchar(20),  train_id int,  route_no int, primary
key(schedule_id,  train_id,  route_id),   foreign
```

```
key(train_id)  references  train(t_id),       foreign
key(route_no) references route(route_no)
);

create table route
(
route_no int,  total_stops
int,  schedule_id int,
train_id int,   primary key(route_no, schedule_id,
train_id),  foreign key(schedule_id) references
schedules(schedule_id),
foreign key(train_id) references train(t_id)
);

create table consists_of
(
route_no int,  train_id int,  schedule_id int,  station_id int,  stop_no
int, arrival_time varchar(20),  departure_time varchar(20),
primary key(roue_no, train_id, schedule_id, station_id),  foreign
key(route_no) references
route(route_no),  foreign key(train_id) references train(t_id),  foreign
key((schedule_id) references
schedules(schedule_id), foreign key(station_id) references station(s_id)
);
```

3.2 Database Connectivity

```
<?php
$servername='localhost';
$username='root';
```

```
$password=' ';
$dbname='railway';



// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
//       Check  connection       if ($conn-
>connect_error) {    die("Connection failed: " .
$conn->connect_error);  }
?>
```

3.3 Implementation of Database Operations

PHP Code for Insertion

```
$sql1="INSERT INTO `train`(`t_id`, `t_name`, `t_type`) VALUES ('$t_id','$t_name','$t_type')";
echo "$sql1";
if ($conn->query($sql1) === TRUE)
{
   echo "<script>alert('Data  Inserted Successfully'); window.location.href='train1.php';
</script>";
}  else
{          echo "error";
echo "<script>alert('Data could not be inserted'); window.location.href='train1.php';  </script>";
}
```

PHP Code for Deletion

```
 $up = "delete from `train`
  where `t_id`= '$t_id'";
  $sel="SELECT t_id,t_name,t_type FROM `train` WHERE 1";
  $result=mysqli_query($connect,$up);
$result1=mysqli_query($connect,$sel);   if($result)
 {
      echo 'Data Updated';
   }else{
      echo 'Data Not Updated';
   }
```

PHP Code for Updation

```
$query="UPDATE train SET t_id='$t_id',t_name='$t_name',t_type='$t_type'WHERE t_id=
'$t_id'";
$data=mysqli_query($conn,$query);
if($data){
echo "<script>alert('Record Updated')</script>";
?>
<META HTTP-EQUIV="Refresh" CONTENT="0; URL= http://localhost/dbms/train2.php">
<?php
} else {  echo "failed to
update";
}
}
?>
```

PHP Code for Retrive

```
$sql = "SELECT * FROM train  ";
 $result=$conn->query($sql); if
 ($result->num_rows > 0) {   //
 output data of each row   echo
 "<html>";         echo   "<table
 border=1>";   echo
 "<tr><th>t_id</th><th>t_name</th><th>t_type</th><th>options</th><th>options</th></tr>";
 while($row = $result->fetch_assoc()) { echo "<tr><td>";   echo $row['t_id'];echo "</td><td>";   echo
 $row['t_name'];echo "</td><td>";   echo $row['t_type'];echo "</td><td> <a href='train
 delete.php?rn=$row[t_id]'
 onclick='returncheckdelete()'><b>Delete</b></a></td><td><a
 href='trainupdate.php?rn=$row[t_id]&tn=$row[t_name]&tp=$row[t_type]'><b>Update</b></td
 >";        echo
 "</tr>";
 }    echo
 "</table>";   echo
 "</center>";   echo
 "</html>"; }  else
 {
  echo "NOTHING TO RETRIEVE";
 }
```

CHAPTER 4

Results and Discussion

In this chapter the results of the project are discussed. The snapshot of the project showing various functionalities like insert, delete, update and retrieval are showcased.

Fig. 4.1 Welcome page of project

Figure 4.1 shows the welcome page of the project. In this page Admin may access the Login page or Passenger may view the details.

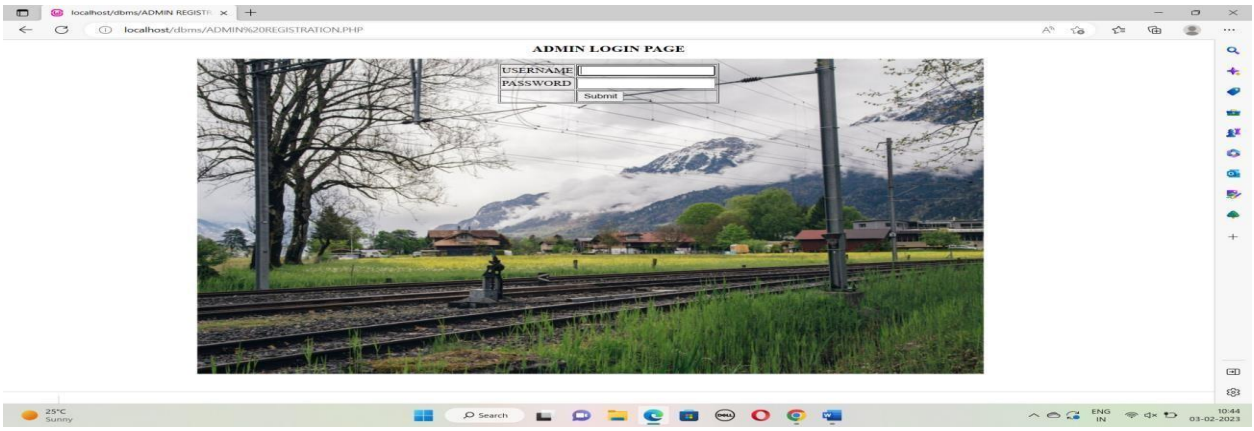


Fig. 4.2 Admin login page

Figure 4.2 shows the Admin login page. In this page Admin can log in by using username and password.

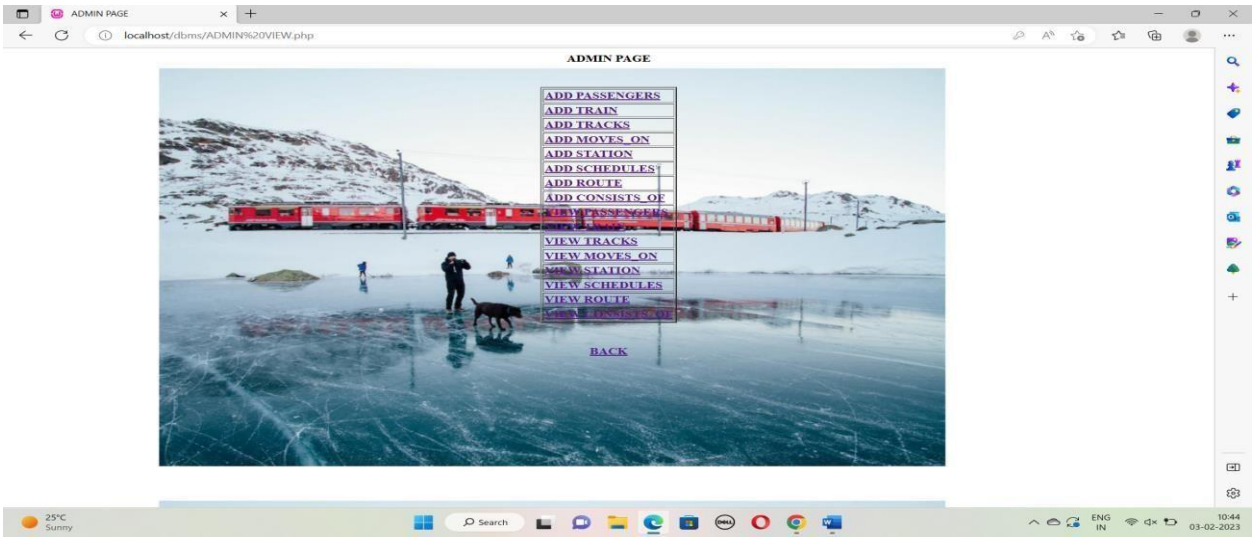


Fig. 4.3 Admin view page

Figure 4.3 shows the Admin view page. In this page Admin can insert the details and view the details.

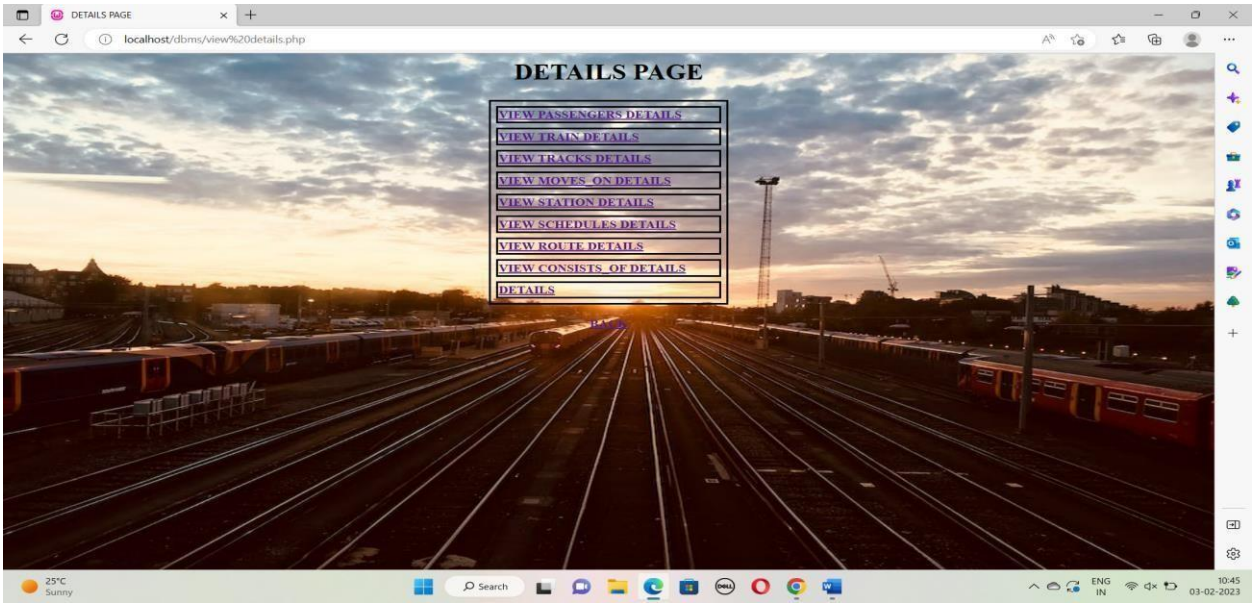


Fig. 4.4 Passengers view details page

Figure 4.4 shows the Passengers view details page. In this page Passengers view the railway details.

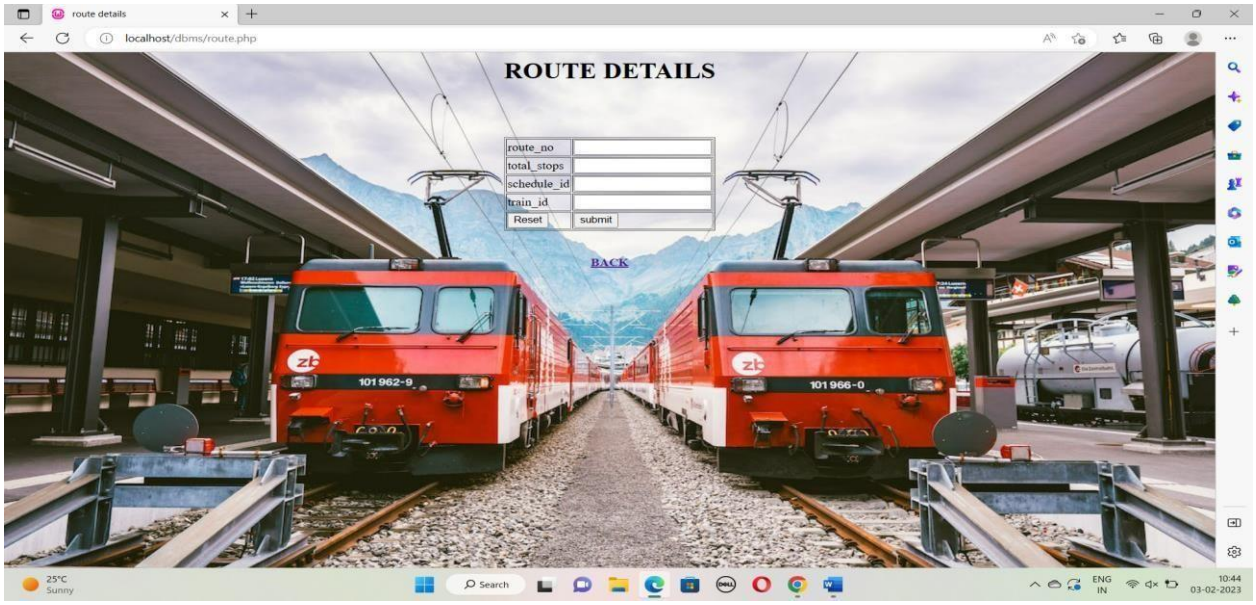


Fig. 4.5 Insert page of the Admin

Figure 4.5 shows the Insert page of the Admin. In this page Admin can insert the details.

Fig. 4.6 View page of the Admin

Figure 4.6 shows the View page of the Admin. In this page Admin can view all the details and perform other operations like delete and update.

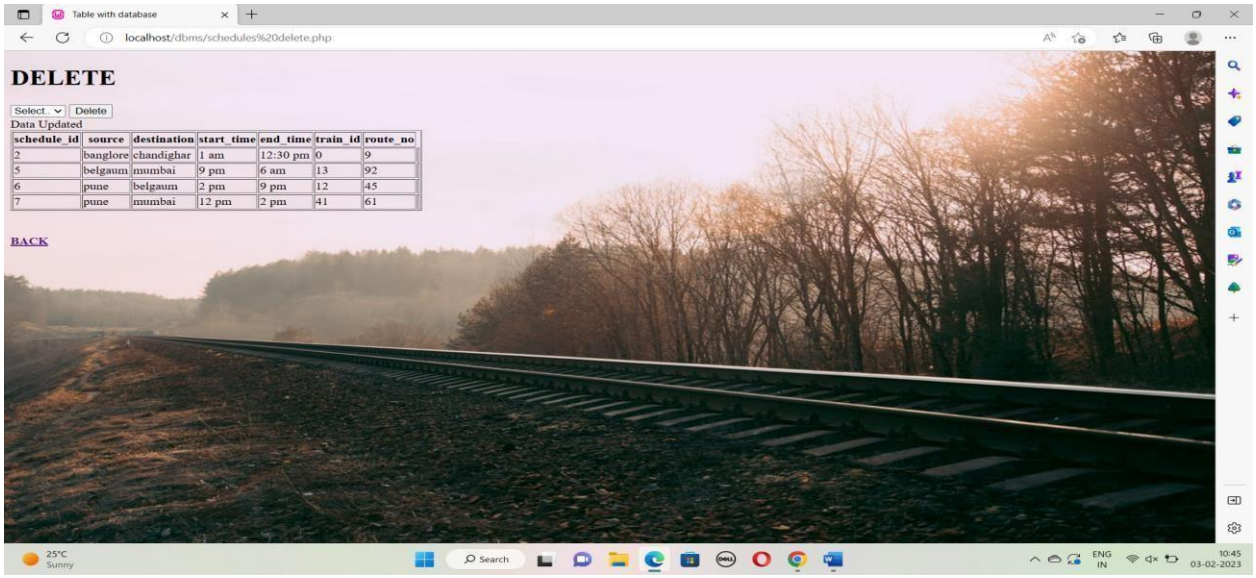


Fig. 4.7 Delete page of the Admin

Figure 4.7 shows the Delete page of Admin. In this page Admin can delete the stored information.

Fig. 4.8 Update page of the Admin

Figure 4.8 shows the Update page of the Admin. In this page Admin can update the stored information.

CHAPTER 5

CONCLUSION

Railway management system database helps us in centralizing the data used for managing the tasks performed in an railway system. The theoretical process involved in database design has been practically implemented. The project provides user friendly interface for the admins to interact with the database. All database operations including insertion, deletion, updation and Retrievals are supported along with support for trigger and stored procedure.

REFERENCES

[1] Database systems Models, Languages, Design and Application Programming, RamezElmasri and Shamkant B.Navathe, 7th Edition, 2017, Pearson.

[2] http:/www.w3schools.com/html/html_intro.asp

[3] http:/www.w3schools.com/css/css_background.asp

[4] http:/www.w3schools.com/sql/sql_update.asp