**CHAPTER 1**

**INTRODUCTION**

**1.1 General Overview**

Our website has various kinds of information that helps regarding booking of tickets via railways .

Users will be able to search the train availability ,the exact fare ,the arrival and departure time of the train and they can also book the ticket by using the debit ,credit or master card and after booking the ticket if the user want to cancel it then they can easily do it also.

**1.2 Survey**

Railway passengers frequently need to know about their ticket reservation status, ticket availability on a particular train or for a place, train arrival or departure details, special trains etc.. Customer information centers at the railway stations are unable to serve such queries at peak periods. The number of the reservation counters available to the passengers and customers are very less. 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 centers facilities available to solve the queries of the passengers.

The online railway ticket reservation system aims to develop a web application which aims at providing trains details, trains availability, as well as the facility to book ticket in online for customers.

So, we thought of developing a web based application which would provide the users all these facilities from his terminal only as well as help them in booking their tickets. The Application was to be divided into two parts namely the user part , and the administrator part. And each of these has their corresponding features.

We decided to give the name of the website “ONLINE TRAIN TICKET RESEVATION SYSTEM”.

.

**1.3 Objectives: ONLINE TRAIN TICKET RESEVATION SYSTEM PROJECT**

The objective of the online train ticket reservation system

Project is to design software to fully automate the process of issuing a railway ticket. That is:- 1. To create a database of the trains

2. To search the trains it’s arrival and departure time,distance between source and destination.

3.To check the availability of the ticket.

4. To calculate fare.

5.To book the ticket.

6.To cancel the ticket if necessary.

**1.3 Analysis**

Online railway ticket reservation is a online ticket booking website, which is capable of booking ticket and serach the train availavility . This website is mainly created to fulfil the following requirements, it comprises of the following properties:-

♣ A central database that will store all information.

♣ An online website that will provide real- time information about the availability of tickets their prices .

♣ Every registered user is able to view his booking id that has been made in his/her name.

♣ Every registered user can change his password any time he wants to change.

♣ Every guest user can search train availability ,price of the ticket,arrival and departure time ,distance between source and destination etc.

♣ Every registered user has the facilities to print his ticket any time he wishes.

Administration login

• In admin mode the administrator can make changes in train details.

• He can also view all booking that has been made by different users.

♣ The booking window contains all the facilities at one place, the user can simply login to his account and can book his ticket.

The report has been organized into seven sections, each section dealing with a different aspect of project and its development. A brief overview of each of the section is in order.

The Software Requirements Specification section will deal with the technical details of the proposed system. It will contain information regarding the developmental and operational environments, user interfaces, functional and requirements specifications and the exception and error handling features of the system.

The Design section will provide a detailed insight into the working of the system, how the system to be framed to make the implementation error free and to avoid a possible rework. All the features mentioned in the SRS section will be dealt with in detail through the means of architectural design, block diagram, detailed data flow diagrams, structure charts, flow chart and any other relevant method.

Through the use of pseudo-codes, the Implementation Details section will provide the final design step before actual implementation of the system. It will list all the functions that will be used in the system, parameters used by them, what results they will produce and how they will interact with the rest of the system.

All queries regarding the actual performance of the developed system will be cleared in the Results and Discussion section. It will list all the test cases, results of those tests and a discussion on whether these tests yielded the desired results or not. If negative, this section will also provide information regarding the failure.

Concluding all this will be the Summary and Conclusion section which will round up the entire discussion.

This will be followed by the Reference/Bibliography section that will tell about the books and site that have been referred by us while developing the website.

**CHAPTER.2**

**SOFTWARE REQUIREMENTS SPECIFICATION-**

**“ONLINE TRAIN TICKET RESEVATION SYSTEM PROJECT”**

**2.1 Development Environments**

**Hardware**

Intel core 2 duo T6400 2.00 GHz with 2GB RAM, 250 GB hard disk space and other Standard accessories.

Environment and Applications:

* Microsoft Windows 7 or higher.
* XAMPP.
* Internet Explorer or Google Chrome.

**2.2 Operating environment**:

**Hardware configuration:**

The minimum configuration for hardware is given below:

* Intel® Pentium® or higher processor.
* 65 MB RAM or higher

**Software configuration:**

* Microsoft® Windows® XP or later versions
* Any standard web browser

**CHAPTER.3**

**SDLC (SOFTWARE DEVELOPMENT LIFE CYCLE) –**

**ONLINE TRAIN TICKET RESEVATION SYSTEM PROJECT**

Every activity has a life cycle and software development process is not an exception for the same. Even if you are not aware of SDLC you still must be following it unknowingly. But if a software professional is aware about SDLC he can execute the project in a much controlled fashion. One of the big benefits of this awareness is that hot blooded developers will not start directly execution (coding) which can really lead to project running in an uncontrolled fashion. Second it helps customer and software professional to avoid Confusion by anticipating the problems and issues before hand. In short SDLC defines the various stages in a software life cycle. But before we try to understand what SDLC is all about. We need to get a broader view of the start and end of SDLC. Any project started if it does not have a start and end then its already in trouble. It’s like if you go out for a drive you should know where to start and where to end or else you are moving around endlessly.

Below is the figure that shows typical flow in SDLC which has five main models .

· Waterfall - Big Bang and Phased model .

· Iterative - Spiral and Incremental model.

**Iterative model**

Iterative model was introduced because of problems faced in Waterfall model.

The iterative waterfall model is used in the development of the system. The system is developed in increments, each increments adding some functional capability to the system until the full system is fully implemented.

The advantage of this approach is that it will result in better testing, as testing of each increment is easier than testing the entire system in totality. Furthermore, this approach provided us with important feedback that was very useful in the implementation of the system.

Feasibility Study

Integration and System testing

Requirement analysis and specification

Coding and unit testing

Design

Maintenance

Development Schedule

The work on the proposed ONLINE RAILWAY TICKET RESERVATION was started on 17th June, 2011 and it was estimated to be over by 17th of Augusest, 2011.

The following Gantt chart has explained the estimated duration of the different phases of the software development work diagrammatically:

Activities

.

**3.1 Implementation Details**

**3.1.1 System Overview**

Online railway ticket reservation system is a web based application. It has been developed using ASP.NET and C# as the code end programming language. A MS SQL Server database consisting of multiple tables is used for data storage.

This Website is a online travel booking website, which is capable of booking tour of national and international destination with easy steps at the cheapest rate . It allows the Administrator to perform all operations and view bookings. The general user is however given only restricted access.

Travelmasti helps in booking tour packages. The user can’t book any tour until he is a registered user.

Feasibility Study

Requirement gathering and specification

Database Design

GUI Design

Code database part

Code GUI part

Unit & System testing

17th-30th june

1st -15th july

25th -5th auguest

5th-17th auguest

16th-25th july

**3.2MySQL**

**MySQL server: -** **ONLINE TRAIN TICKET RESEVATION SYSTEM PROJECT**

MySQL server is basically a database server which is mostly used for storing user data into the required database in a specific table for easy access of these data in the future. The database server works when a local application invokes it. Before storing data into a database with the help of database server, PHP sends a SQL query to MySQL database server for establishing a connection to the server by using the loopback address, i.e., 127.0.0.1 along with the user name and password for getting authentication from the database server. Since connection is established locally with the database server by PHP, so there is no chance of getting access to database server for modifying database by the client. Also , when PHP request for information retrieval from the database server by sending SQL query then the database server sends back the result of query after executing it by the database server. Another method of accessing database server is only possible by locally, i.e., only administrator of that particular computer can get all the access facilities of the database server as well as all the database which are not available to restricted users. PHP can update information in the database server if it has the administrator username and password to access thespecific database, otherwise connection will be rejected by MySQL server and the database cannot be updated.

Since before the dawn of the computer age , people have been using databases. Before computers, a database may have been a Rolodex containing phone numbers of the important people you knew, or it was a filing cabinet that contained all the personnel records for the company. Today, databases are computer-based and are found virtually everywhere. From desktop databases of your record collection to Web-enabled databases that run large corporations.

**CHAPTER.4**

**IMPLEMENTATION**

Following are the related files

• account.php: the main page of the website after login

• book.php: for booking seats

• goforcancellation.php, cancel.php: for cancelling seats

• captcha.php: for captcha generation

• changepassword.php, changepasswordfinal.php: for chaning passwords

• dblink.php: for establishing link with the database

• findTrains.php: finding trains between station

• footer.php: general footer code for all php files

• generate.php: generating test data and some tables

• getAvailability: getting availability of seats

• guard.php: destroys timed out sessions and redirects to index.php

• header.js: Javascript functions to assist the website on client side

• header.php: general header code for most php files

• history.php: getting the booking history of current user account

• index.php: main page of the website when not logged in

• logout.php: Logs the user out

• pnrStatus.php: echoes back with PNR status related data

• Railways.sql: contains SQL code for generating tables and sequences

• register.php: registers new users into the website

• tdr.php, tdrfiling.php, tdrfinal.php: TDR related files

• triggers.sql: triggers for checking wherther validity constraints are satisfied

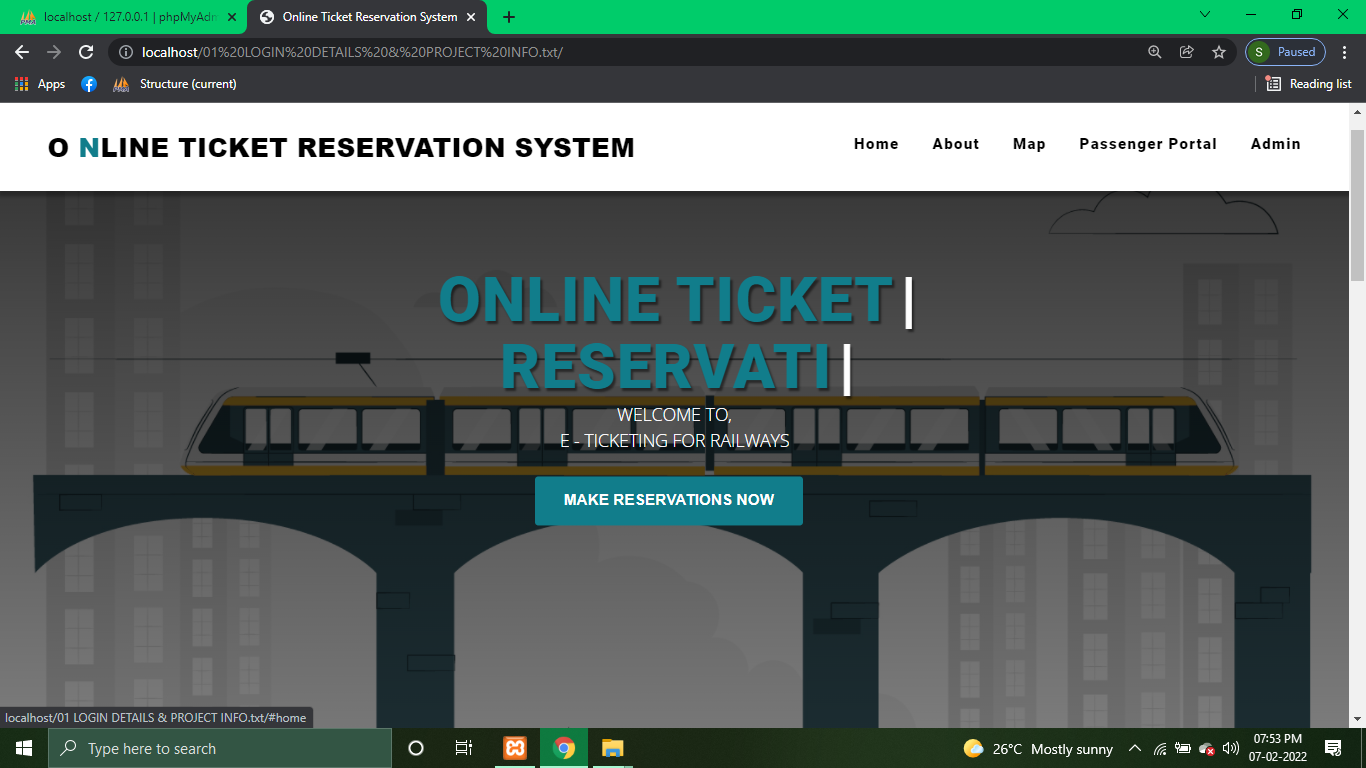
• updateprofile.php, updateprofilefinal.php: User profile updation

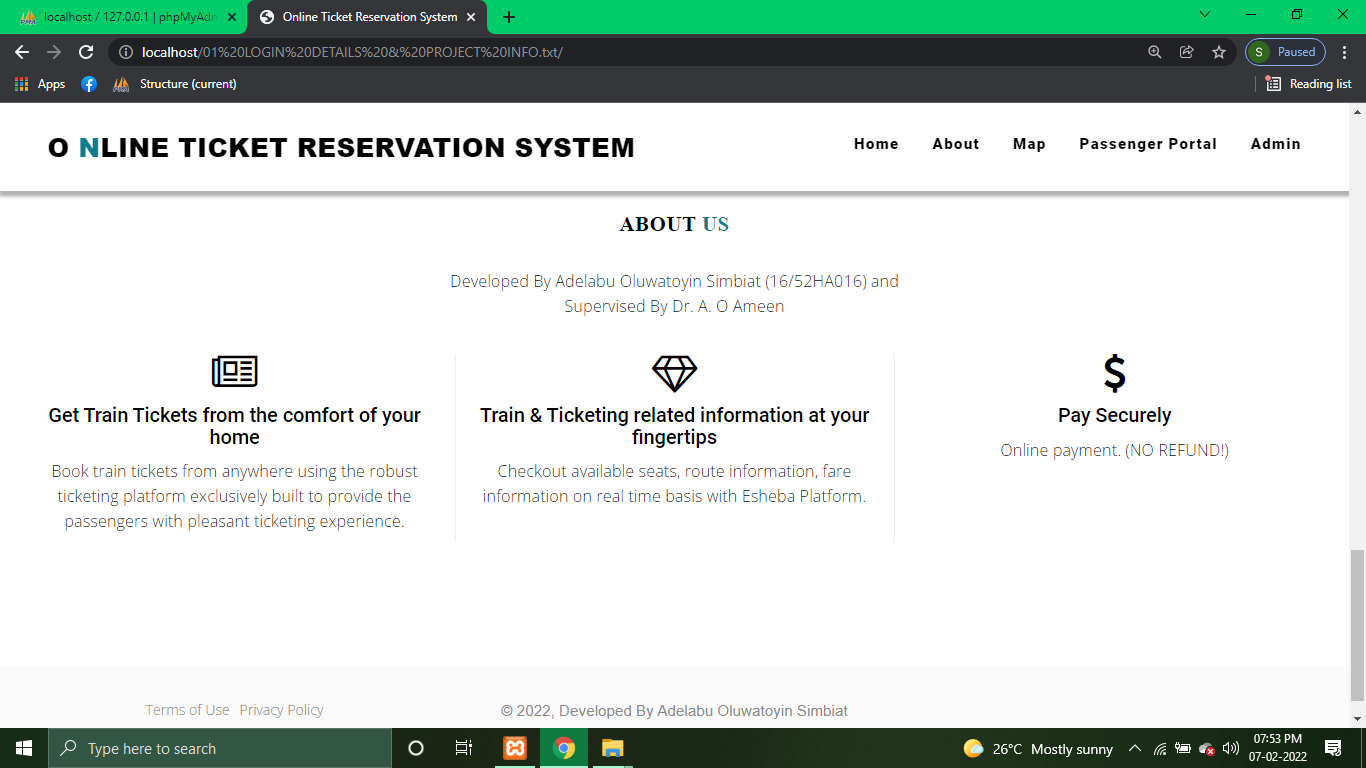
• datagencodes: Python and Bash codes for generating and importing reaslitic data.

**CHAPTER.5**

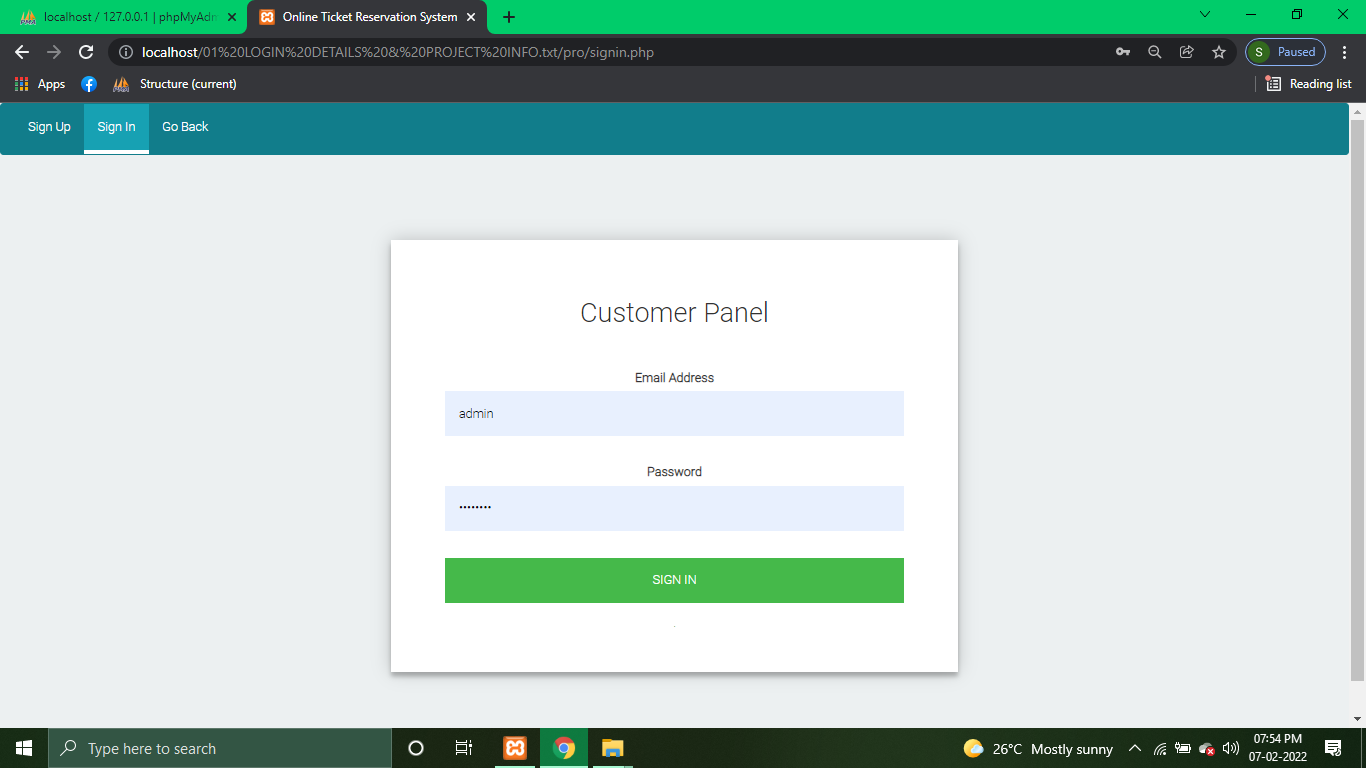
**SCREENSHOTS**

5.1HOMEPAGE:

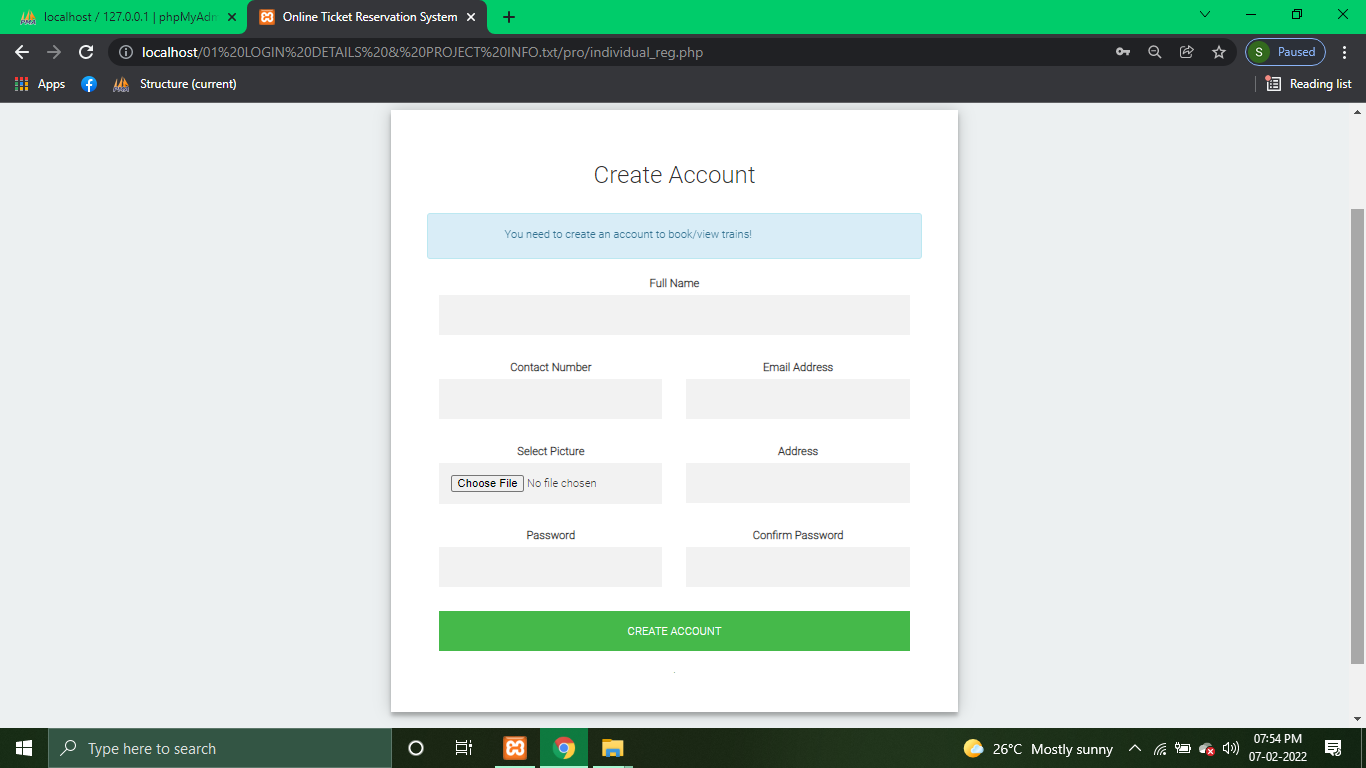


5.2ABOUTUS: 

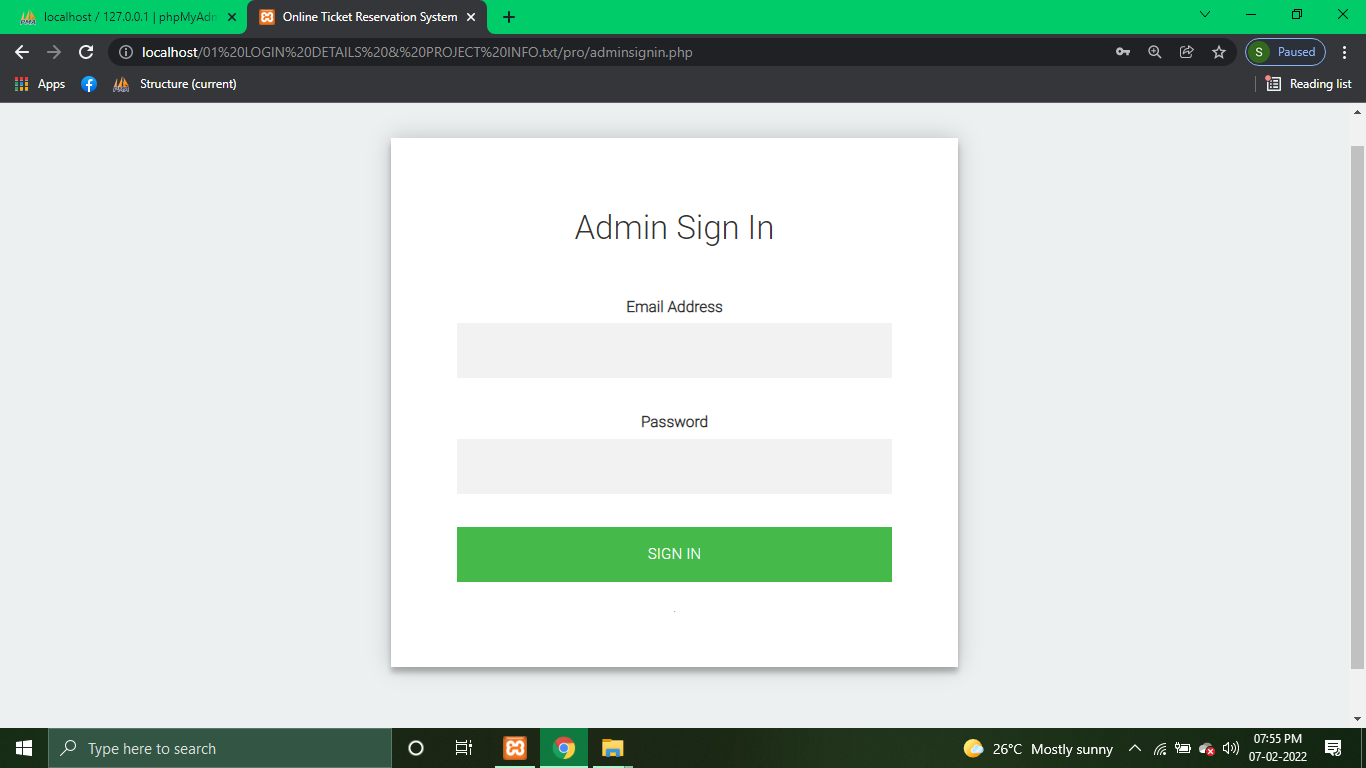
5.3CUSTOMER PANEL:



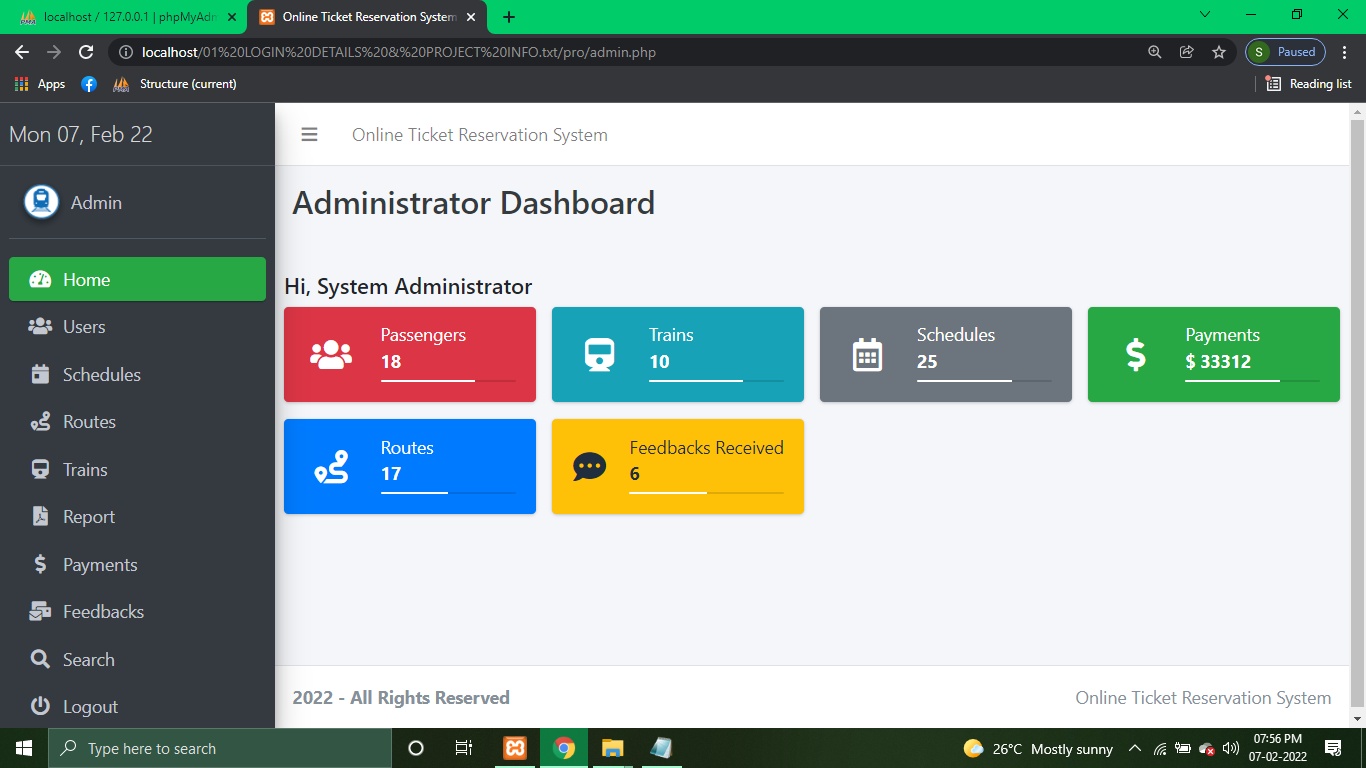
5.4CUSTOMER REGISTER PAGE:



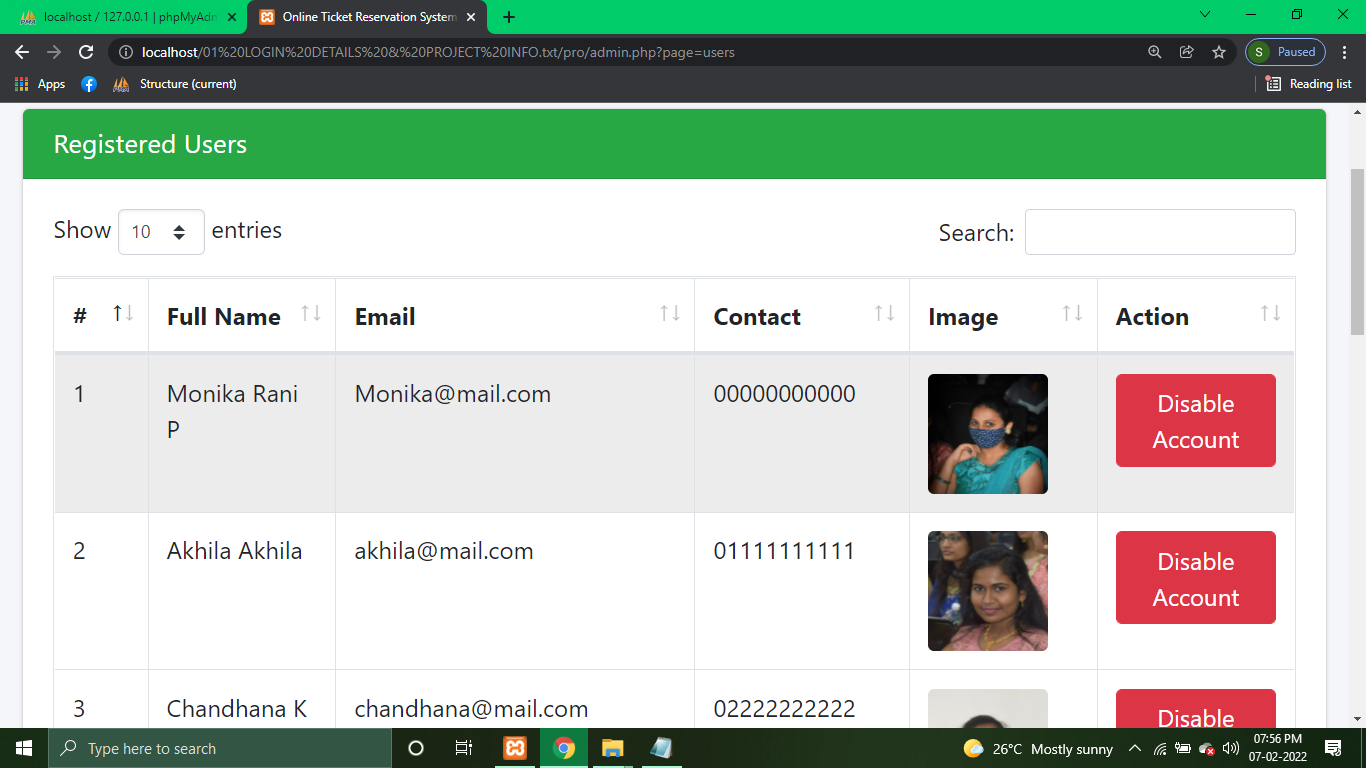
5.5ADMIN SIGN IN:

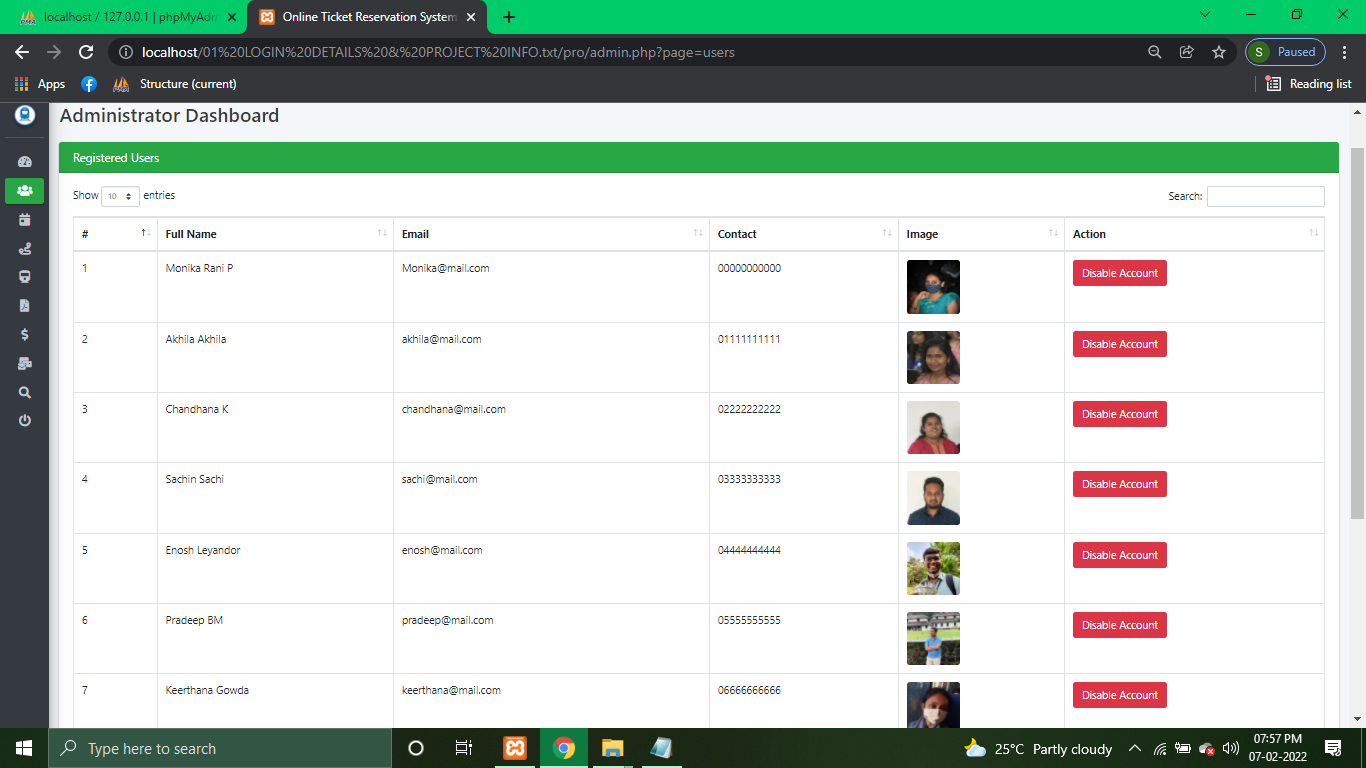


5.6ADMINISTRATOR DASHBOARD:

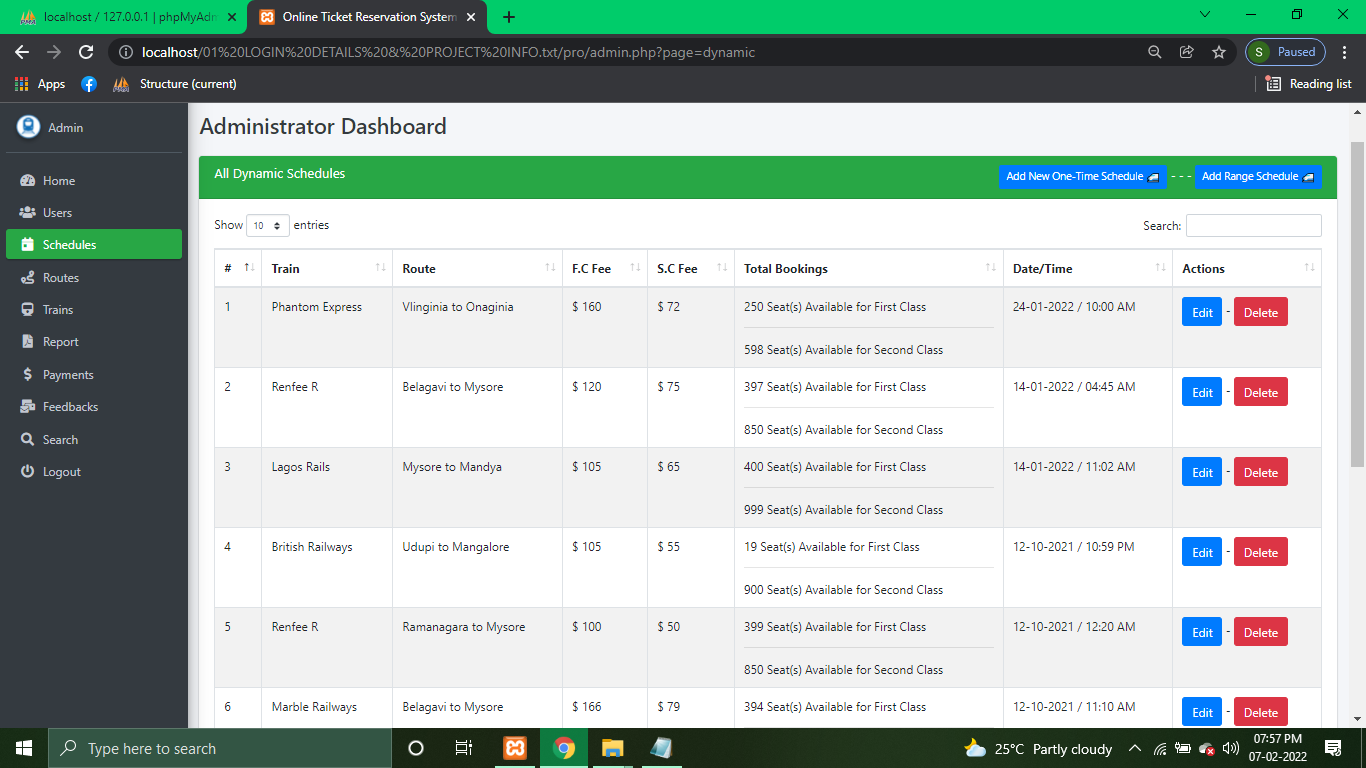


5.7REGISTERED USERS:

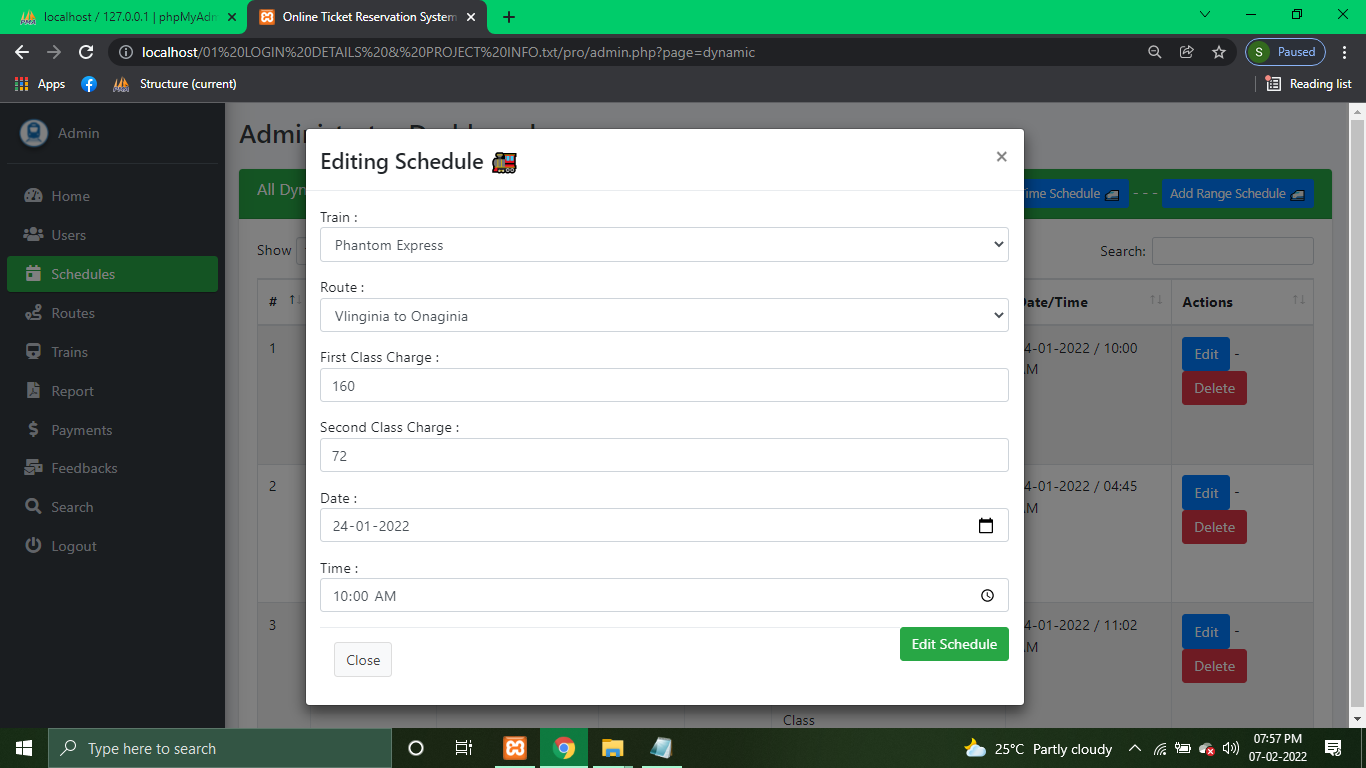




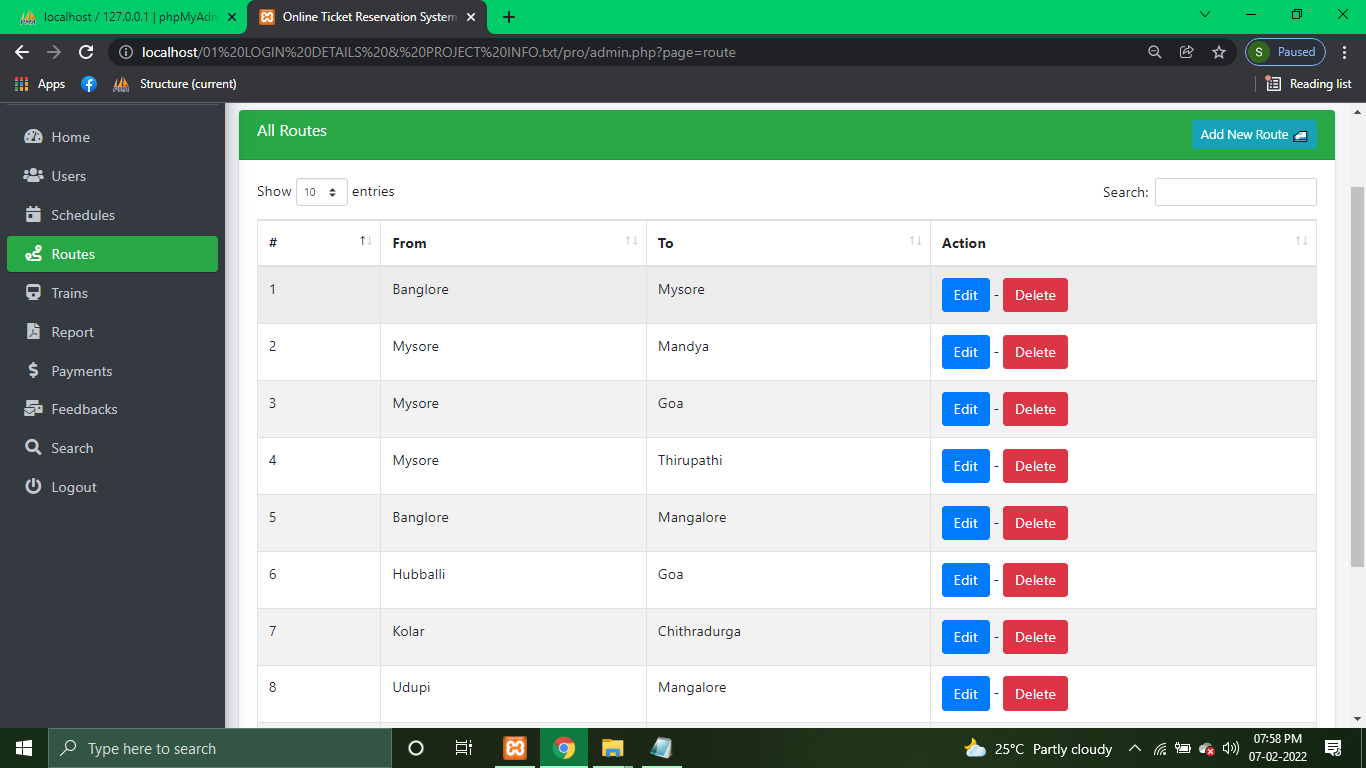
5.8SCHEDULES:



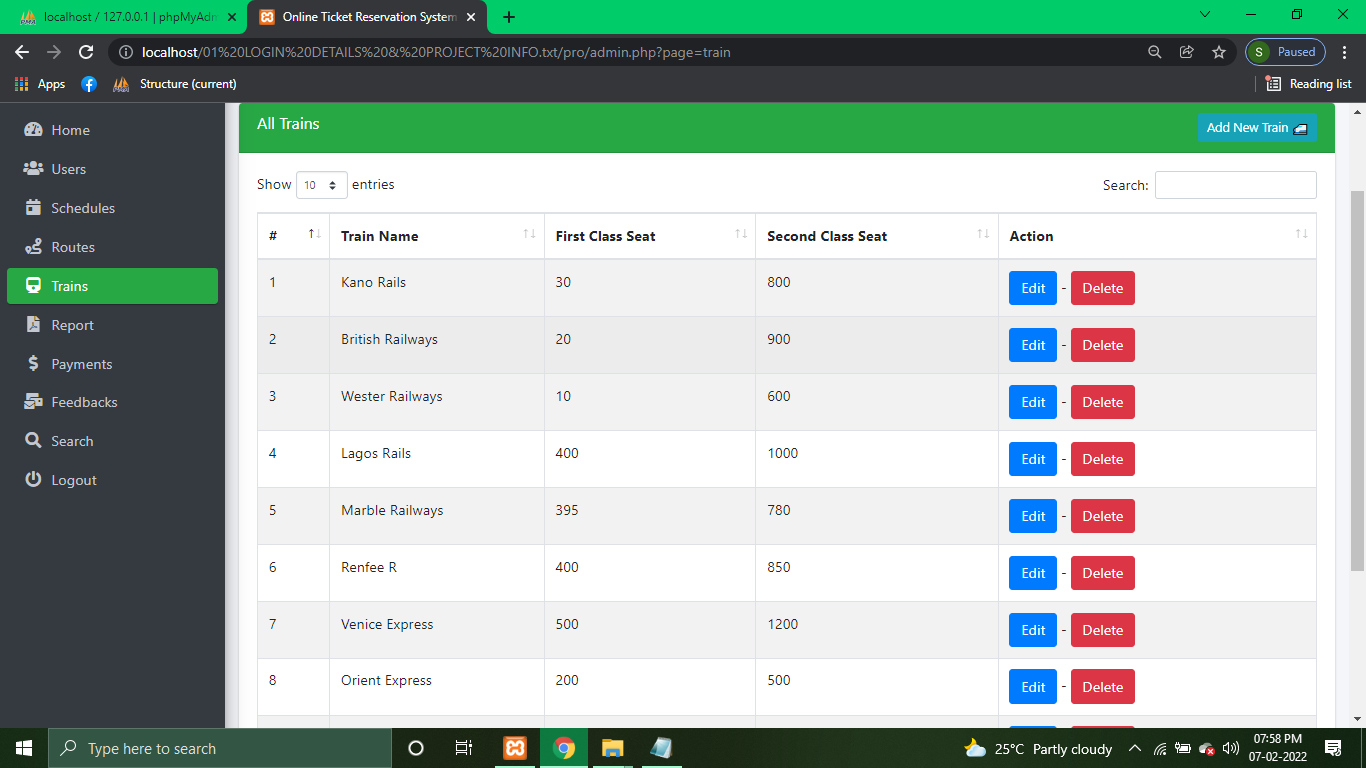
5.9EDITING SCHEDULES:



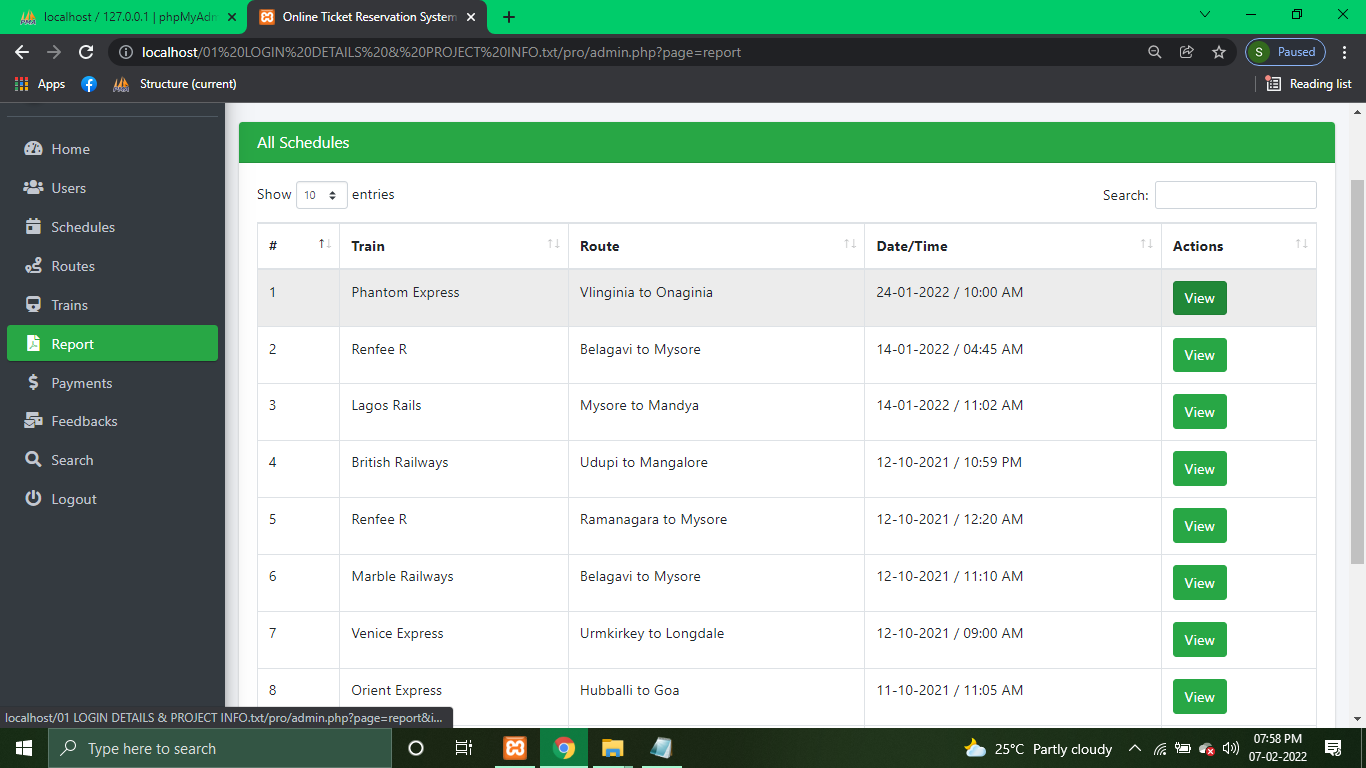
5.10 ROUTES:



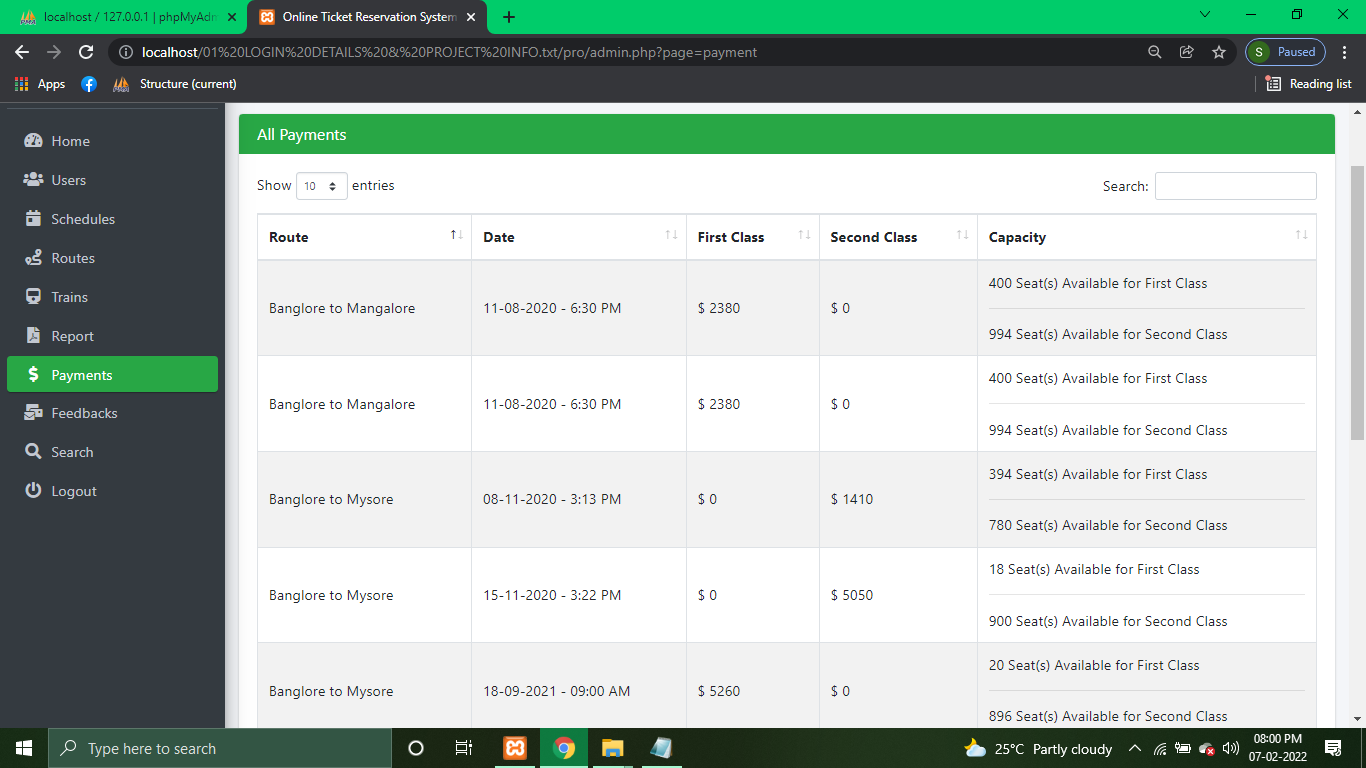
5.11 ALL TRAINS:



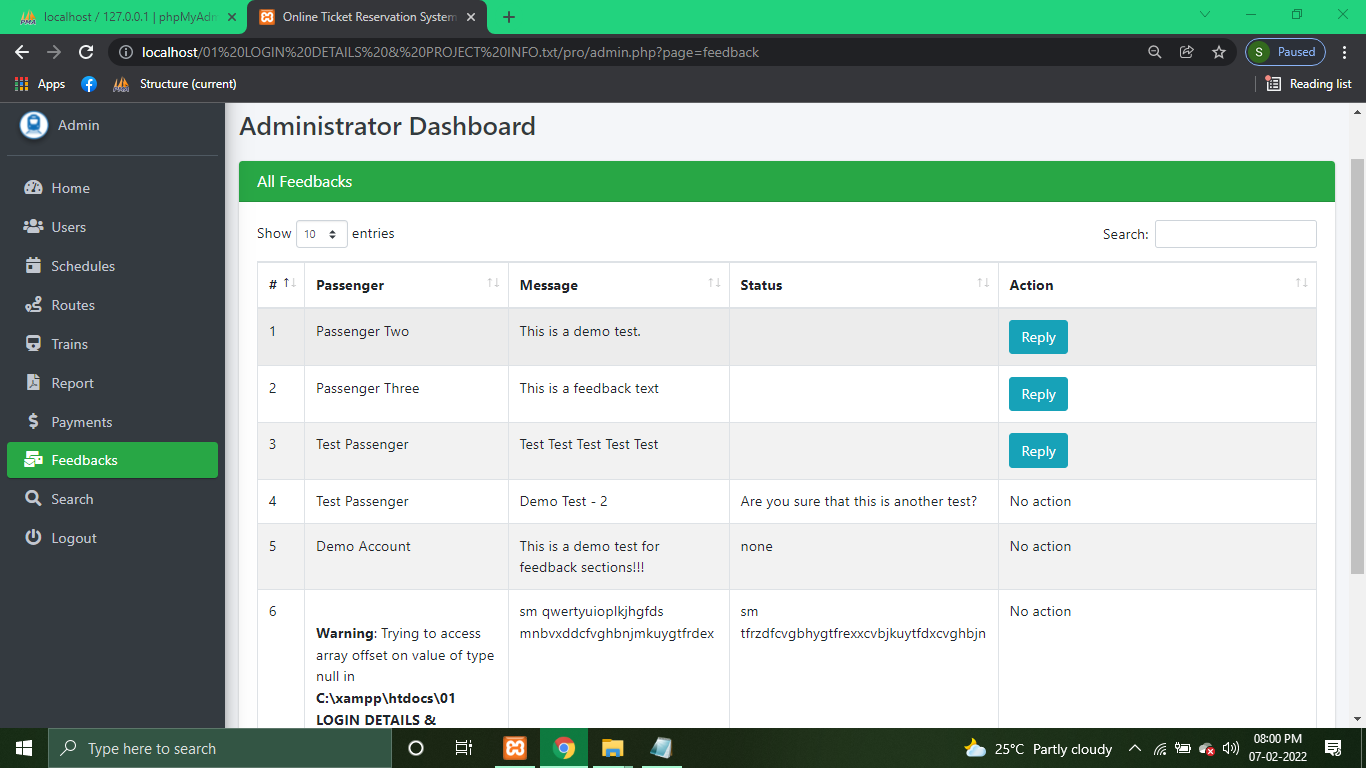
5.12 REPORTS:



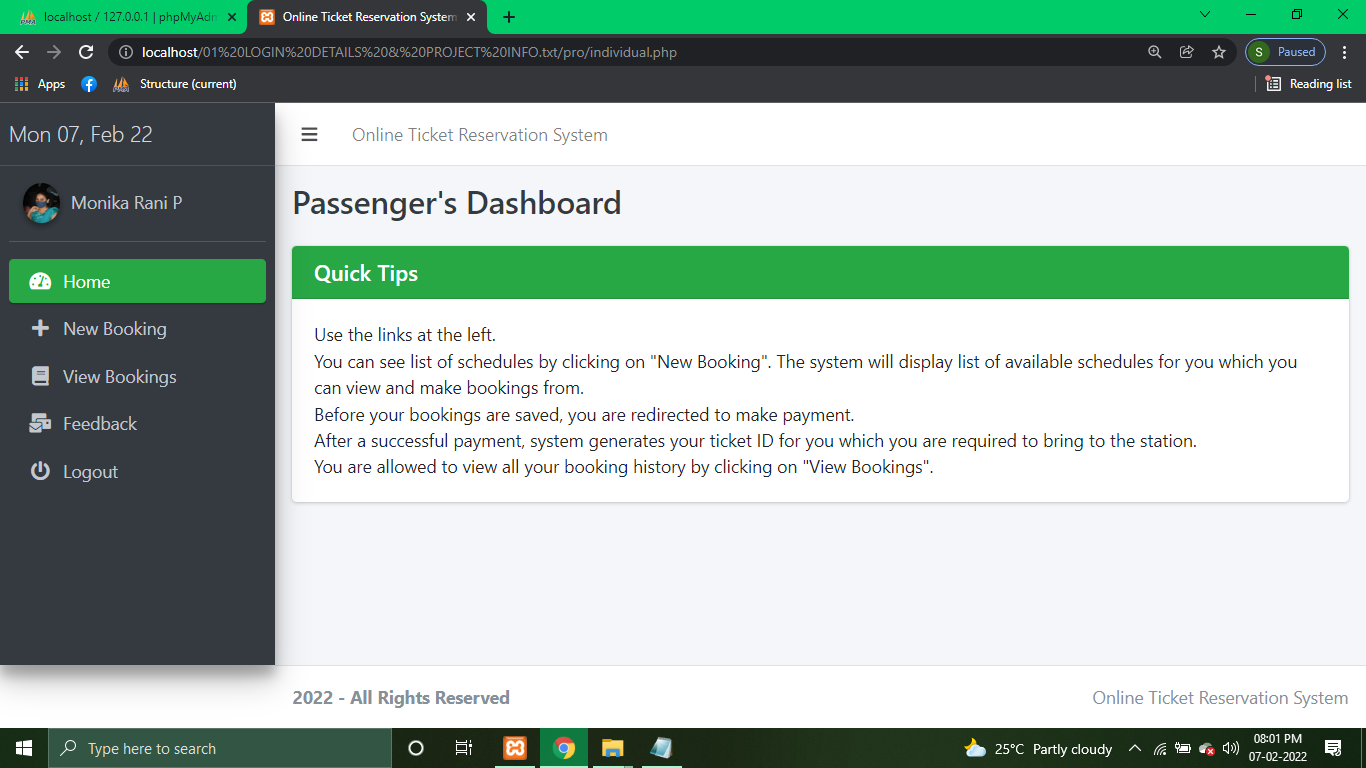
5.13 PAYMENTS:



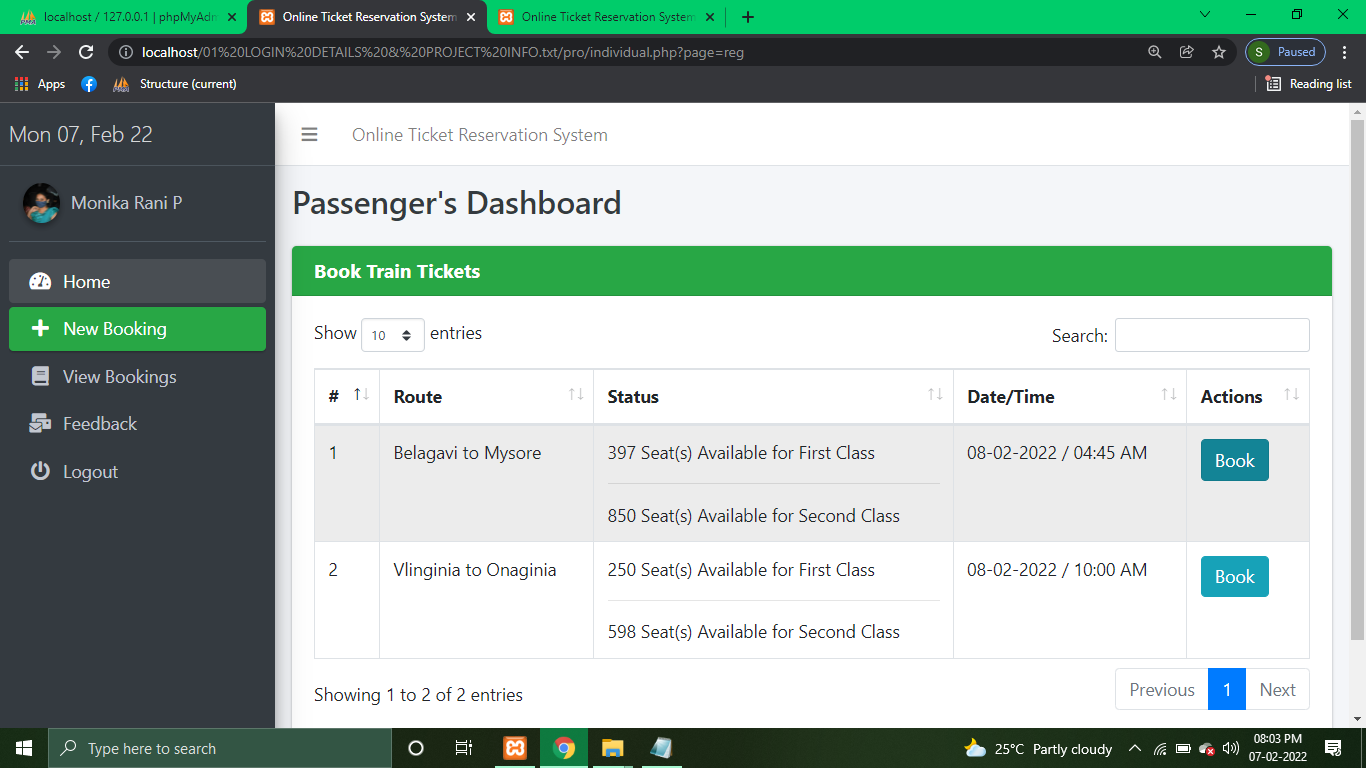
5.14 FEEDBACKS:

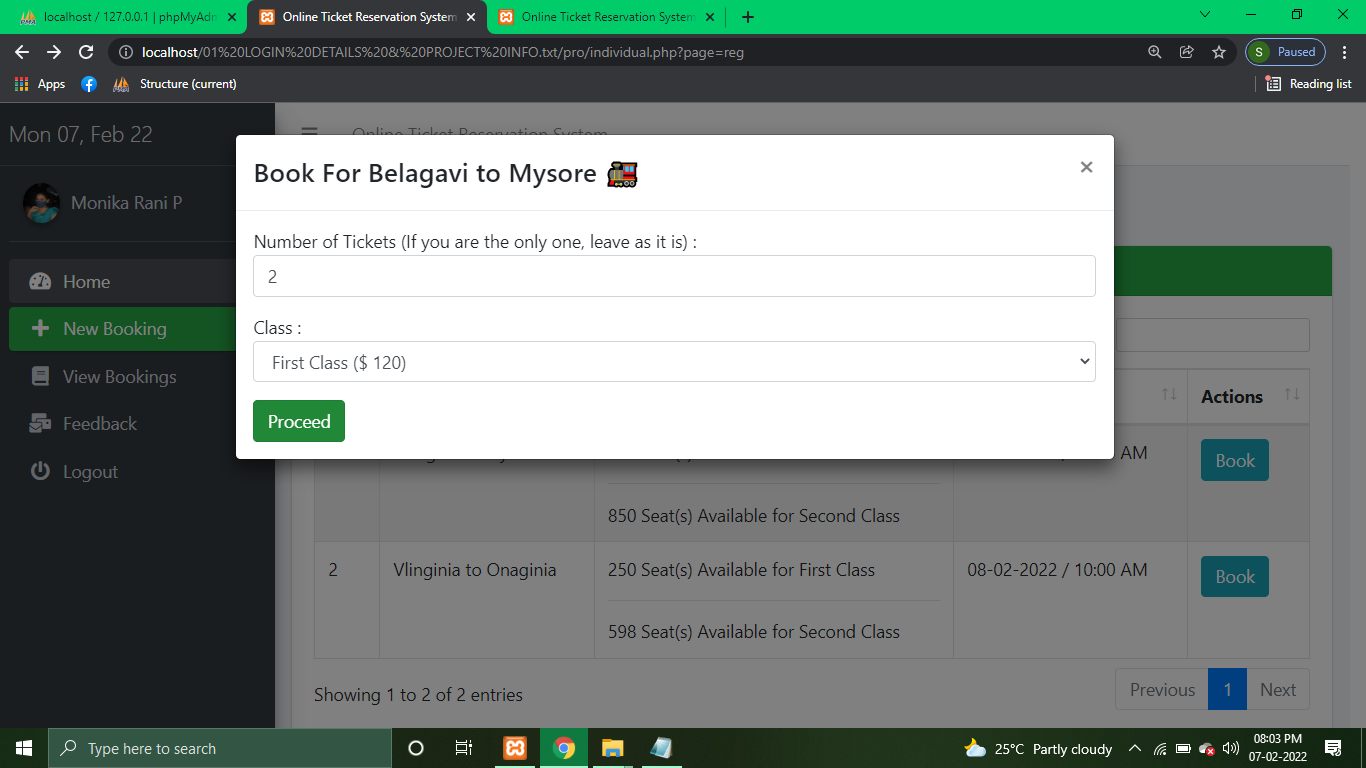


5.15PASSENGER’S DASHBOARD:

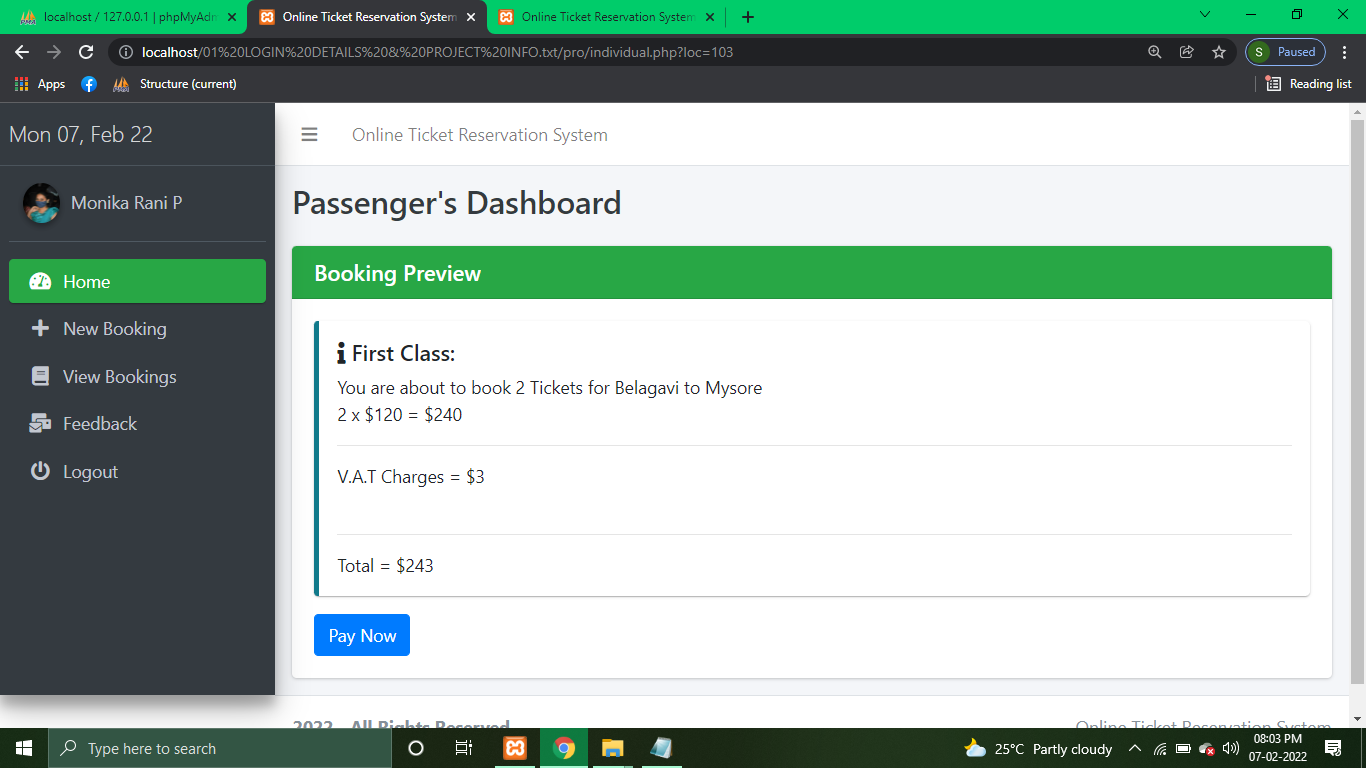


5.16NEW BOOKING:

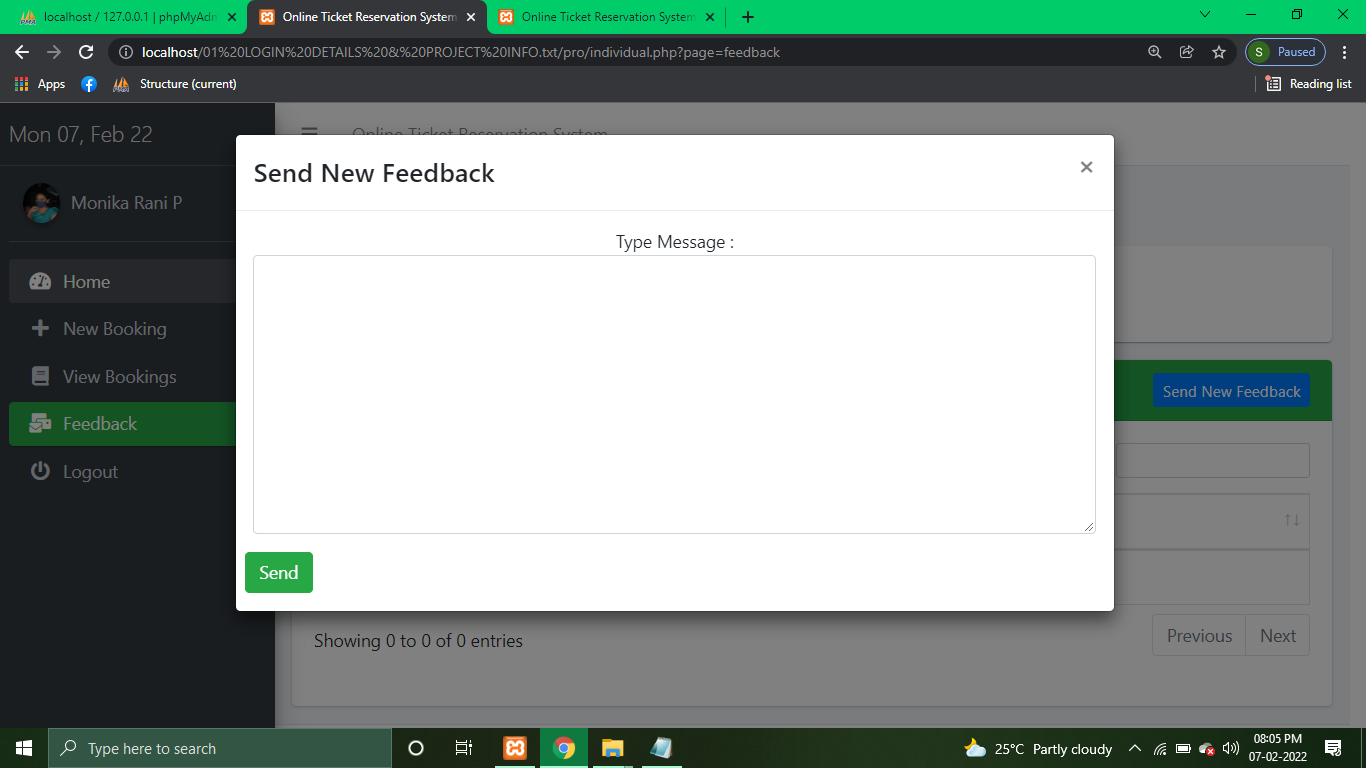




5.17BOOKING PREVIEW:



5.18SEND FEEDBACKS:



**CHAPTER.6**

**SOURCE CODE:**

**Admin:**

<?php

@session\_start();

$file\_access = true;

include '../conn.php';

include 'admin\_session.php';

include '../constants.php';

if (@$\_GET['page'] == 'print' && isset($\_GET['code'])) {

printClearance($\_GET['code']);

// echo "<script>window.location='admin.php'</script>";

}

if (@$\_GET['page'] == 'report' && isset($\_GET['id'])) {

printReport($\_GET['id']);

// echo "<script>window.location='admin.php'</script>";

}

$fullname = "System Administrator";

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<meta http-equiv="x-ua-compatible" content="ie=edge">

<link rel="stylesheet" href="plugins/datatables-bs4/css/dataTables.bootstrap4.css">

<title><?php echo SITE\_NAME, ' - ', ucwords($\_SESSION['category']); ?> </title>

<!-- Font Awesome Icons -->

<link rel="stylesheet" href="plugins/fontawesome-free/css/all.min.css">

<!-- Theme style -->

<link rel="stylesheet" href="dist/css/adminlte.min.css">

</head>

<body class="hold-transition sidebar-mini">

<div class="wrapper">

<!-- Navbar -->

<nav class="main-header navbar navbar-expand navbar-white navbar-light">

<!-- Left navbar links -->

<ul class="navbar-nav">

<li class="nav-item">

<a class="nav-link" data-widget="pushmenu" href="#"><i class="fas fa-bars"></i></a>

</li>

<li class="navbar-nav">

<a class="nav-link" href="#"><?php echo SITE\_NAME ?></a>

</li>

</ul>

<!-- Right navbar links -->

</nav>

<!-- /.navbar -->

<!-- Main Sidebar Container -->

<aside class="main-sidebar sidebar-dark-success elevation-4">

<!-- Brand Logo -->

<a href="admin.php" class="brand-link">

<span class="brand-text font-weight-light"><?php echo date("D d, M y"); ?></span>

</a>

<!-- Sidebar -->

<div class="sidebar">

<!-- Sidebar user panel (optional) -->

<div class="user-panel mt-3 pb-3 mb-3 d-flex">

<div class="image">

<img src="images/trainlg.png" class="img-circle elevation-2" alt="User Image">

</div>

<div class="info">

<a href="#" class="d-block">Admin</a>

</div>

</div>

<!-- Sidebar Menu -->

<nav class="mt-2">

<ul class="nav nav-pills nav-sidebar flex-column" data-widget="treeview" role="menu"

data-accordion="false">

<!-- Add icons to the links using the .nav-icon class

with font-awesome or any other icon font library -->

<li class="nav-item">

<a href="admin.php" class="nav-link <?php if (!isset($\_GET['page'])) echo 'active'; ?>">

<i class="nav-icon fas fa-tachometer-alt"></i>

<p>

Home

</p>

</a>

</li>

<li class="nav-item">

<a href="admin.php?page=users" class="nav-link

<?php

echo (@$\_GET['page'] == 'users') ? 'active' : '';

?>

">

<i class="nav-icon fas fa-users"></i>

<p>

Users

</p>

</a>

</li>

<li class="nav-item">

<a href="admin.php?page=dynamic" class="nav-link

<?php

echo (@$\_GET['page'] == 'dynamic') ? 'active' : '';

?>

">

<i class="nav-icon fas fa-calendar-day"></i>

<p>

Schedules

</p>

</a>

</li>

<li class="nav-item">

<a href="admin.php?page=route" class="nav-link <?php

echo (@$\_GET['page'] == 'route') ? 'active' : '';

?>">

<i class="nav-icon fas fa-route"></i>

<p>

Routes

</p>

</a>

</li>

</li>

<li class="nav-item">

<a href="admin.php?page=train" class="nav-link <?php

echo (@$\_GET['page'] == 'train') ? 'active' : '';

?>">

<i class="nav-icon fas fa-train"></i>

<p>

Trains

</p>

</a>

</li>

<li class="nav-item">

<a href="admin.php?page=report" class="nav-link <?php

echo (@$\_GET['page'] == 'report') ? 'active' : '';

?>">

<i class="nav-icon fas fa-file-pdf"></i>

<p>

Report

</p>

</a>

</li>

<li class="nav-item">

<a href="admin.php?page=payment" class="nav-link <?php

echo (@$\_GET['page'] == 'payment') ? 'active' : '';

?>">

<i class="nav-icon fas fa-dollar-sign"></i>

<p>

Payments

</p>

</a>

</li>

<li class="nav-item">

<a href="admin.php?page=feedback" class="nav-link <?php

echo (@$\_GET['page'] == 'feedback') ? 'active' : '';

?>">

<i class="nav-icon fas fa-mail-bulk"></i>

<p>

Feedbacks

</p>

</a>

</li>

<li class="nav-item">

<a href="admin.php?page=search" class="nav-link <?php

echo (@$\_GET['page'] == 'search') ? 'active' : '';

?>">

<i class="nav-icon fas fa-search"></i>

<p>

Search

</p>

</a>

</li>

<li class="nav-item">

<a href="admin.php?page=logout" class="nav-link">

<i class="nav-icon fas fa-power-off"></i>

<p>

Logout

</p>

</a>

</ul>

</nav>

<!-- /.sidebar-menu -->

</div>

<!-- /.sidebar -->

**Admin sign in:**

<?php

session\_start();

require\_once '../conn.php';

$file = "admin";

?>

<?php

$cur\_page = 'signup';

include 'includes/inc-header.php';

if (isset($\_POST['email'])) {

$email = $\_POST['email'];

$password = $\_POST['password'];

if (!isset($email, $password)) {

?>

<script>

alert("Ensure you fill the form properly.");

</script>

<?php

} else {

//Check for login

$password = md5($password);

$check = $conn->prepare("SELECT \* FROM users WHERE email = ? AND password = ?");

$check->bind\_param("ss", $email, $password);

if (!$check->execute()) die("Form Filled With Error");

$res = $check->get\_result();

$no\_rows = $res->num\_rows;

if ($no\_rows == 1) {

$row = $res->fetch\_assoc();

$id = $row['id'];

session\_regenerate\_id(true);

$\_SESSION['category'] = "super";

$\_SESSION['admin'] = $id;

?>

<script>

alert("Access Granted!");

window.location = "admin.php";

</script>

<?php

} else { ?>

<script>

alert("Access Denied.");

</script>

<?php

}

}

}

?>

<div class="signup-page">

<div class="form">

<h2>Admin Sign In</h2>

<br>

<form class="login-form" method="post" role="form" id="signup-form" autocomplete="off">

<!-- json response will be here -->

<div id="errorDiv"></div>

<!-- json response will be here -->

<div class="col-md-12">

<div class="form-group">

<label>Email Address</label>

<input type="text" required name="email">

</div>

</div>

<div class="col-md-12">

<div class="form-group">

<label>Password</label>

<input type="password" name="password" id="password">

<span class="help-block" id="error"></span>

</div>

</div>

<div class="col-md-12">

<div class="form-group">

<button type="submit" id="btn-signup">

SIGN IN

</button>

</div>

</div>

<p class="message">

<a href="#">.</a><br>

</p>

</form>

</div>

</div>

</div>

<script src="assets/js/jquery-1.12.4-jquery.min.js"></script>

</body>

</html>

**Train:**

<?php

if (!isset($file\_access)) die("Direct File Access Denied");

$source = 'train';

$me = "?page=$source";

?>

<div class="content">

<!-- Main content -->

<section class="content">

<div class="container-fluid">

<div class="row">

<div class="col-sm-12">

<div class="card card-success">

<div class="card-header">

<h3 class="card-title">

All Trains</h3>

<div class='float-right'>

<button type="button" class="btn btn-info btn-sm" data-toggle="modal"

data-target="#add">

Add New Train &#128645;

</button></div>

</div>

<div class="card-body">

<table id="example1" style="align-items: stretch;"

class="table table-hover w-100 table-bordered table-striped<?php //

?>">

<thead>

<tr>

<th>#</th>

<th>Train Name</th>

<th>First Class Seat</th>

<th>Second Class Seat</th>

<th style="width: 30%;">Action</th>

</tr>

</thead>

<tbody>

<?php

$row = $conn->query("SELECT \* FROM train");

if ($row->num\_rows < 1) echo "No Records Yet";

$sn = 0;

while ($fetch = $row->fetch\_assoc()) {

$id = $fetch['id'];

?>

<tr>

<td><?php echo ++$sn; ?></td>

<td><?php echo $fullname = $fetch['name']; ?></td>

<td><?php echo $fetch['first\_seat']; ?></td>

<td><?php echo $fetch['second\_seat']; ?></td>

<td>

<form method="POST">

<button type="button" class="btn btn-primary" data-toggle="modal"

data-target="#edit<?php echo $id ?>">

Edit

</button> -

<input type="hidden" class="form-control" name="del\_train"

value="<?php echo $id ?>" required id="">

<button type="submit"

onclick="return confirm('Are you sure about this?')"

class="btn btn-danger">

Delete

</button>

</form>

</td>

</tr>

<div class="modal fade" id="edit<?php echo $id ?>">

<div class="modal-dialog modal-lg">

<div class="modal-content">

<div class="modal-header">

<h4 class="modal-title">Editing <?php echo $fullname;

?></h4>

<button type="button" class="close" data-dismiss="modal"

aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

<div class="modal-body">

<form action="" method="post">

<input type="hidden" class="form-control" name="id"

value="<?php echo $id ?>" required id="">

<p>Train Name : <input type="text" class="form-control"

name="name" value="<?php echo $fetch['name'] ?>"

required minlength="3" id=""></p>

<p>First Class Capacity : <input type="number" min='0'

class="form-control"

value="<?php echo $fetch['first\_seat'] ?>"

name="first\_seat" required id="">

</p>

<p> Class Capacity : <input type="number" min='0'

class="form-control"

value="<?php echo $fetch['second\_seat'] ?>"

name="second\_seat" required id="">

</p>

<p>

<input class="btn btn-info" type="submit" value="Edit Train"

name='edit'>

</p>

</form>

<div class="modal-footer justify-content-between">

<button type="button" class="btn btn-default"

data-dismiss="modal">Close</button>

</div>

</div>

<!-- /.modal-content -->

</div>

<!-- /.modal-dialog -->

</div>

<!-- /.modal -->

<?php

}

?>

</tbody>

</table>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</section>

</div>

<div class="modal fade" id="add">

<div class="modal-dialog modal-lg">

<div class="modal-content" align="center">

<div class="modal-header">

<h4 class="modal-title">Add New Train &#128646;

</h4>

<button type="button" class="close" data-dismiss="modal" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

<div class="modal-body">

<form action="" method="post">

<table class="table table-bordered">

<tr>

<th>Train Name</th>

<td><input type="text" class="form-control" name="name" required minlength="3" id=""></td>

</tr>

<tr>

<th>First Class Capacity</th>

<td><input type="number" min='0' class="form-control" name="first\_seat" required id=""></td>

</tr>

<tr>

<th>Second Class Capacity</th>

<td><input type="number" min='0' class="form-control" name="second\_seat" required id="">

</td>

</tr>

<tr>

<td colspan="2">

<input class="btn btn-info" type="submit" value="Add Train" name='submit'>

</td>

</tr>

</table>

</form>

</div>

</div>

<!-- /.modal-content -->

</div>

<!-- /.modal-dialog -->

</div>

<?php

if (isset($\_POST['submit'])) {

$name = $\_POST['name'];

$first\_seat = $\_POST['first\_seat'];

$second\_seat = $\_POST['second\_seat'];

if (!isset($name, $first\_seat, $second\_seat)) {

alert("Fill Form Properly!");

} else {

$conn = connect();

//Check if train exists

$check = $conn->query("SELECT \* FROM train WHERE name = '$name' ")->num\_rows;

if ($check) {

alert("Train exists");

} else {

$ins = $conn->prepare("INSERT INTO train (name, first\_seat, second\_seat) VALUES (?,?,?)");

$ins->bind\_param("sss", $name, $first\_seat, $second\_seat);

$ins->execute();

alert("Train Added!");

load($\_SERVER['PHP\_SELF'] . "$me");

}

}

}

if (isset($\_POST['edit'])) {

$name = $\_POST['name'];

$first\_seat = $\_POST['first\_seat'];

$second\_seat = $\_POST['second\_seat'];

$id = $\_POST['id'];

if (!isset($name, $first\_seat, $second\_seat)) {

alert("Fill Form Properly!");

} else {

$conn = connect();

//Check if train exists

$check = $conn->query("SELECT \* FROM train WHERE name = '$name' ")->num\_rows;

if ($check == 2) {

alert("Train name exists");

} else {

$ins = $conn->prepare("UPDATE train SET name = ?, first\_seat = ?, second\_seat = ? WHERE id = ?");

$ins->bind\_param("sssi", $name, $first\_seat, $second\_seat, $id);

$ins->execute();

alert("Train Modified!");

load($\_SERVER['PHP\_SELF'] . "$me");

}

}

}

if (isset($\_POST['del\_train'])) {

$con = connect();

$conn = $con->query("DELETE FROM train WHERE id = '" . $\_POST['del\_train'] . "'");

if ($con->affected\_rows < 1) {

alert("Train Could Not Be Deleted. This Train Has Been Tied To Another Data!");

load($\_SERVER['PHP\_SELF'] . "$me");

} else {

alert("Train Deleted!");

load($\_SERVER['PHP\_SELF'] . "$me");

}

}

?>

**REFFERENCES**

Website

* Dynamicdrive.com
* Freecsstemplates.com

Books

* Microsoft Official Custom Workshop
* ASP.NET The Complete Reference (Herbert Schild)
* Database Management System,Rameez Elmasri & Shamkant Navathe.

**RESULTS**

Online railway ticket reservation system was successfully designed and developed as per

the specifications. It was extensively tested using a database which contains data similar to

what can be expected in an actual database. The system was found to work satisfactorily

without any errors under all conditions.