VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JNANA SANGAMA”, BELAGAVI-590018

A MINI PROJECT REPORT ON

BUS TICKET RESERVATION SYSTEM

Submitted in partial fulfillment of the requirements

For the award of degree of

Bachelor of Engineering

In

Computer Science and Engineering

By

ASHWINI J

[1KS18CS007]

Under the guidance of

Mr. Sougandhika Narayan
Asst. Prof, Dept. Of CSE

Mr. Prashanth H S
Asst. Prof, Dept. Of CSE

Department of Computer Science & Engineering

K.S.INSTITUTE OF TECHNOLOGY

#14, Raghuvanahalli, Kanakapura Main Road, Bengaluru-560109

2020-21

K.S.INSTITUTE OF TECHNOLOGY

#14, Raghuvanahalli, Kanakapura Main Road, Bengaluru-560109

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that mini project work entitled **"BUS TICKET RESERVATION SYSTEM"** carried out by **Ms. ASHWINI J** bearing USN **1KS18CS007** bonafide student of **K.S. Institute of Technology** in the partial fulfilment for the award of the **Bachelor of Engineering** in **Computer Science & Engineering** of the **Visvesvaraya Technological University, Belagavi,** during the year 2020. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of mini Project work prescribed for the said degree for the 5th semester.

Dr . Rekha. B. Venkatapur
Prof & HOD, CS & E Department

Dr . K.V.A.Balaji
Principal/CEO, KSIT

Mrs. Sougandhika Narayana
Asst . Prof, Dept. Of CSE

Mr Prashanth H.S
Asst. Prof, Dept. Of CSE

Name of the Examiners **Signature with date**

1.

2.

ACKNOWLEDGEMENT

I take this opportunity to thank one and all involved in building this project. Firstly I would like to thank the college for providing us an opportunity to work on the project.

I would also like to thank the management of **K. S. Institute of Technology** for providing all the resources required for the project.

I wish to acknowledge my sincere gratitude to our beloved Principal, **Dr . K.V.A.Balaji** for his encouragement and providing all facilities for the accomplishment of this project.

This project would not have been possible without the support of our beloved **Prof & HOD, Dr. Rekha.B.Venkatapur, Dept. of CSE.**

I am also highly grateful to my project guides, **Mrs. Sougandhika Narayana**, Asst . Prof, Dept. Of CSE and **Mr Prashanth H.S**, Asst . Prof, Dept. Of CSE ,who have been very generous in assisting and supporting, to do this Project **"Bus Ticket Reservation System".**

I also would like to thank all other teaching and non-teaching staff members who have extended their support and co-operation while bringing up this project.

ASHWINI J

(1KS18CS007)

ABSTRACT

Traveling is a large growing business across all countries. Bus reservation system deals with maintenance of records of details of each passenger who had reserved a seat for a journey. It also includes maintenance of information like schedule and details of each bus.

I observed the working of the Bus reservation system and after going through it, I get to know that there are many operations, which they have to do manually. It takes a lot of time and causes many errors. Due to this, sometimes a lot of problems occur and they were facing many disputes with customers. To solve the above problem, and further maintaining records of items, seat availability for customers, price of per seat, bill generation and other things, I have developed this reservation system.

Customer can check availability of bus and reserve selective seats. The project provides and checks all sorts of constraints so that user does give only useful data and thus validation is done in an effective way.

CONTENTS

Chapter 1

INTRODUCTION

1.1 OVERVIEW

Online Bus Reservation System provides bus transportation system, a facility to reserved seats, cancellation of seats and different types of enquiry which need an instant and quick reservation. This system can be used by the users in performing online reservation via internet for their all business purposes. Users can use this program directly on their websites and no need to install it.

1.2 PROBLEM STATEMENT

The focus of the project is to computerize traveling company to manage data, so that all the transactions become fast and there should not be any error in transaction like calculation mistake, bill generation and other things. It replaces all the paper work. It keeps records of all bills also, giving to ensure 100% successful implementation of the computerized Bus reservation system.

1.3 DATABASE MANAGEMENT SYSTEM

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. The DBMS essentially serves as an interface between the database and end users application programs, ensuring that data is consistently organized and remains easily accessible.

The DBMS manages three important things: the data, the database engine that allows data to be accessed, locked and modified, and the database schema, which defines the database’s logical structure. These three foundational elements help to provide concurrency, security, data integrity and uniform administration procedures.

Typical database administration tasks supported by the DBMS include change management, performance monitoring/tuning and backup and recovery. Many database management systems are also responsible for automated rollbacks, restarts and recovery as well as the logging and auditing of activity.

1.4 SQL

SQL is a standard language for storing, manipulating and retrieving data in databases. Originally based upon relational algebra and tuple relational calculus, SQL consists of a data definition language, data manipulation language, and data control language. The scope of SQL includes data insert, query, update and delete, schema creation and modification, and data access control.

SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987.[13]Since then, the standard has been revised to include a larger set of features. Despite the existence of such standards, most SQL code is not completely portable among different database systems without adjustments.

1.5 HTML / JavaScript

HTML is a markup language used for structuring and presenting content on the web and the fifth and current major version of the HTML standard.

HTML5 includes detailed processing models to encourage more interoperable implementations; it extends, improves and rationalizes the markup available for documents, and introduces markup and application programming interfaces (APIs) for complex web applications.

JavaScript often abbreviated as JS, is a high-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm.

Alongside HTML and CSS, JavaScript is one of the three core technologies of the World Wide Web. JavaScript enables interactive web pages and thus is an essential part of web applications.

1.6 PHP

PHP is a widely used open source general purpose scripting language that is especially suites for web development and can be embedded into html basically, a server-side scripting language designed primarily for web development but also used as a general-purpose programming language.

PHP code may be embedded into HTML or HTML5 markup, 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 software 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.

XAMPP

XAMPP is a small and light Apache distribution containing the most common web development technologies in a single package. Its contents, small size, and portability make it the ideal tool for students developing and testing applications in PHP and MySQL. XAMPP is available as a free download in two specific packages: full and lite. While the full package download provides a wide array of development tools, XAMPP Lite contains the necessary technologies that meet the Ontario Skills Competition standard.

APACHE

The Apache HTTP Server is web server software notable for playing a key role in the initial growth of the World Wide Web. In 2009 it became the first web server software to surpass the 100 million web site milestone. Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation. Since April 1996 Apache has been the most popular HTTP server software in use.

Chapter 2

REQUIREMENTS SPECIFICATION

A computerized way of handling information about property and users details is efficient, organized and time saving, compared to a manual way of doing so. This is done through a database driven web application whose requirements are mentioned in this section.

2.1 OVERALL DESCRIPTION

A reliable and scalable database driven web application with security features that is easy to use and maintain is the requisite.

2.2 SPECIFIC REQUIREMENTS

2.2.1 SOFTWARE REQUIREMENTS

➢ Technology Implemented	: Xampp Server
➢ Language Used	: PHP
➢ Database	: My SQL
➢ User Interface Design	: HTML, CSS

2.2.2 HARDWARE REQUIREMENTS

- Processor – Pentium IV or above
- RAM – 2 GB or more
- Hard disk – 3 GB or more
- Monitor – VGA of 1024x768 screen resolution
- Keyboard and Mouse

2.3. FUNCTIONAL REQUIREMENTS

The functional requirements of this project are:

The project allows login of an entity, User:

The User has a set of functionalities described as:

> He/ She has to register to the system.

> He/ She can login into the system.

> He/She can enquire for buses in the desired route.

> He/ She can book and cancel the bookings.

> He/ She can give feedback on the services provided

2.4. NON-FUNCTIONAL REQUIREMENTS

The non-functional requirements of these projects are:

• Security:

The website does not allow access to any functionality by directly jumping to any particular link to that function’s page. Additionally, anything that is needed to be done can only be done by first logging in.

• Data Integrity:

The project does not allow entry of data in case data is invalid. This is very important as, if invalid data is added, then it can cause large problems, such as illegal activities, loss of money etc.

• Serviceability:

The system shall be designed so that technical support personnel are able to monitor and manage in operation.

Chapter 3

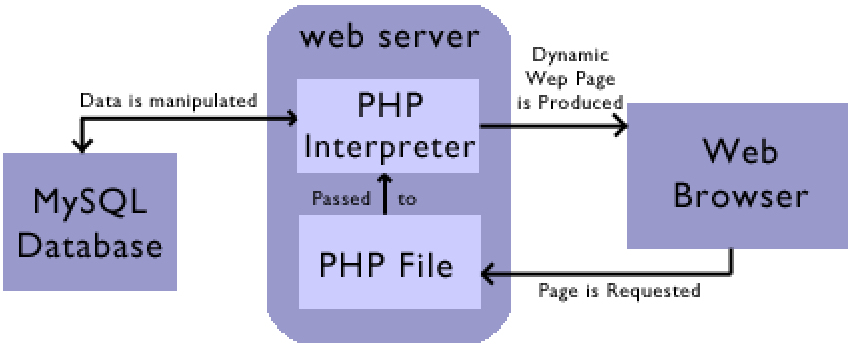
DETAILED DESIGN

3.1 SYSTEM DESIGN

First user accessed the website through browser. That means user types the URL of the website in browser and hit go. The page request on browser will reach to the Web Server (Apache). Web server will collect that requested page (HTML or PHP or Image file etc) from its document root.

Now if it is a static element like HTML , CSS , image file or Java Script file then Apache will send it directly to browser. And Browser will render it to user on screen If it is a PHP file then Apache sends the content of the file to PHP Interpreter. PHP interpreter interprets the PHP code and executes it. if DB operation is required it performs the same (E) PHP Interpreter generates output (if the PHP code is to generate any output) and sends to Apache Apache sends that content to browser Browser renders it to users' screen All static components like HTML, CSS files ,Image Files , Java Scripts etc doesn't need interpreter.

Our web browsers are built to render them and display on screen properly. That is why if user requests for these kind of components Apache collects them from Document root and sends back to Browser directly. Only if requested page is a PHP page Apache will send it to PHP interpreter to get it translated and executed. That is why though those listed static components reside on Server we will consider them as part of User Interface and as they get rendered at user's browser we may refer them as Client side components. In web technology Browsers are Client terminals.



3.2 ENTITY RELATIONSHIP DIAGRAM

An entity–relationship model is usually the result of systematic analysis to define and describe what is important to processes in an area of a business.

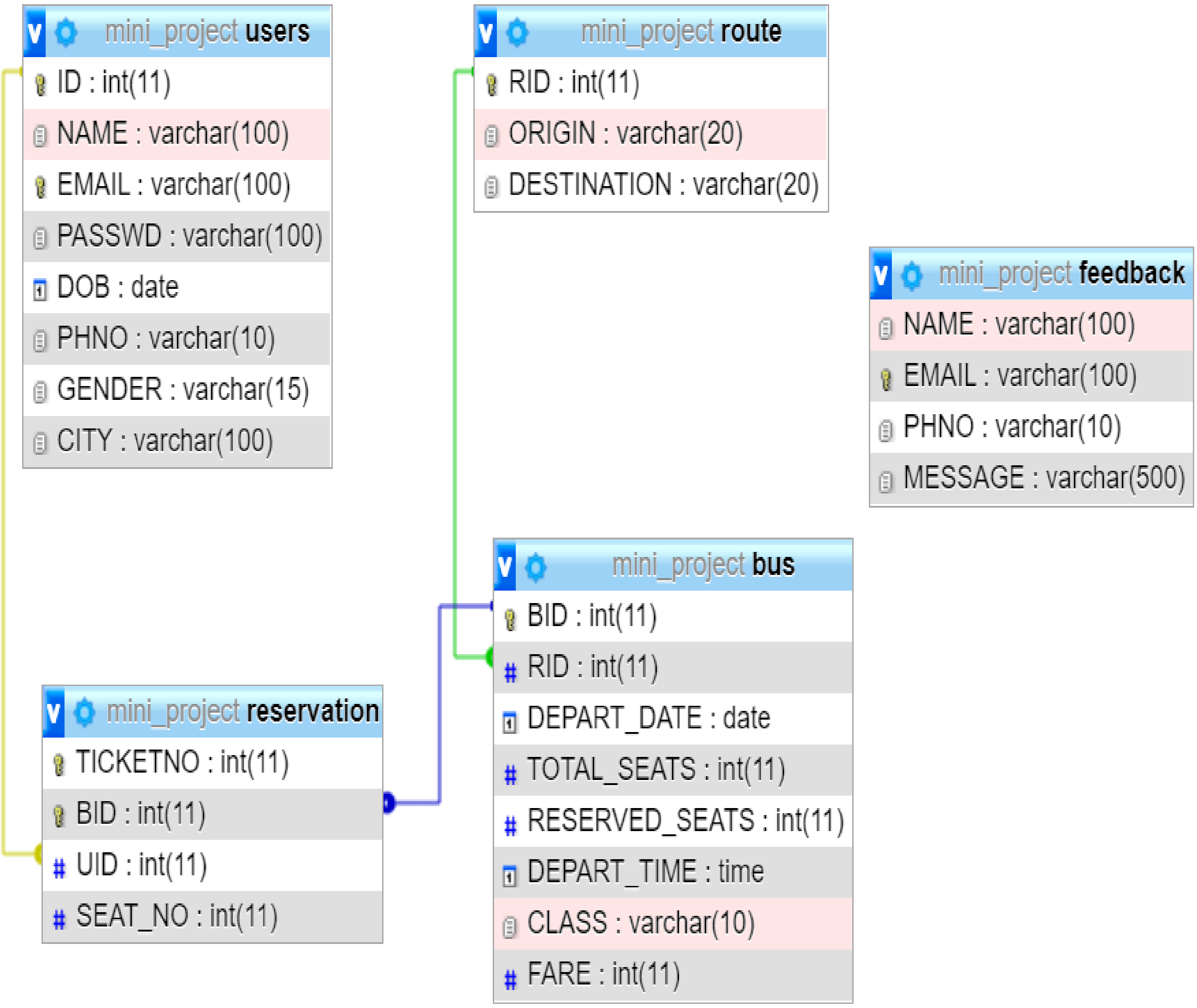
An E-R model does not define the business processes; it only presents a business data schema in graphical form. It is usually drawn in a graphical form as boxes (entities) that are connected by lines (relationships) which express the associations and dependencies between entities.

Entities may be characterized not only by relationships, but also by additional properties (attributes), which include identifiers called "primary keys". Diagrams created to represent attributes as well as entities and relationships may be called entity-attribute-relationship diagrams, rather than entity-relationship models.

An ER model is typically implemented as a database. In a simple relational database implementation, each row of a table represents one instance of an entity type, and each field in a table represents an attribute type. In a relational database a relationship between entities is implemented by storing the primary key of one entity as a pointer or "foreign key" in the table of another entity.

There is a tradition for ER/data models to be built at two or three levels of abstraction. Note that the conceptual-logical-physical hierarchy below is used in other kinds of specification, and is different from the three schema approach to software engineering. While useful for organizing data that can be represented by a relational structure, an entity-relationship diagram can't sufficiently represent semi-structured or unstructured data, and an ER Diagram is unlikely to be helpful on its own in integrating data into a pre-existing information system.

Cardinality notations define the attributes of the relationship between the entities. Cardinalities can denote that an entity is optional.



mini_project users
- ID : int(11)
- NAME : varchar(100)
- EMAIL : varchar(100)
- PASSWD : varchar(100)
- DOB : date
- PHNO : varchar(10)
- GENDER : varchar(15)
- CITY : varchar(100)

mini_project route
- RID : int(11)
- ORIGIN : varchar(20)
- DESTINATION : varchar(20)

mini_project feedback
- NAME : varchar(100)
- EMAIL : varchar(100)
- PHNO : varchar(10)
- MESSAGE : varchar(500)

mini_project bus
- BID : int(11)
- RID : int(11)
- DEPART_DATE : date
- TOTAL_SEATS : int(11)
- RESERVED_SEATS : int(11)
- DEPART_TIME : time
- CLASS : varchar(10)
- FARE : int(11)

mini_project reservation
- TICKETNO : int(11)
- BID : int(11)
- UID : int(11)
- SEAT_NO : int(11)

Fig. 3.2.1 Enhanced ER diagram of Bus Reservation System

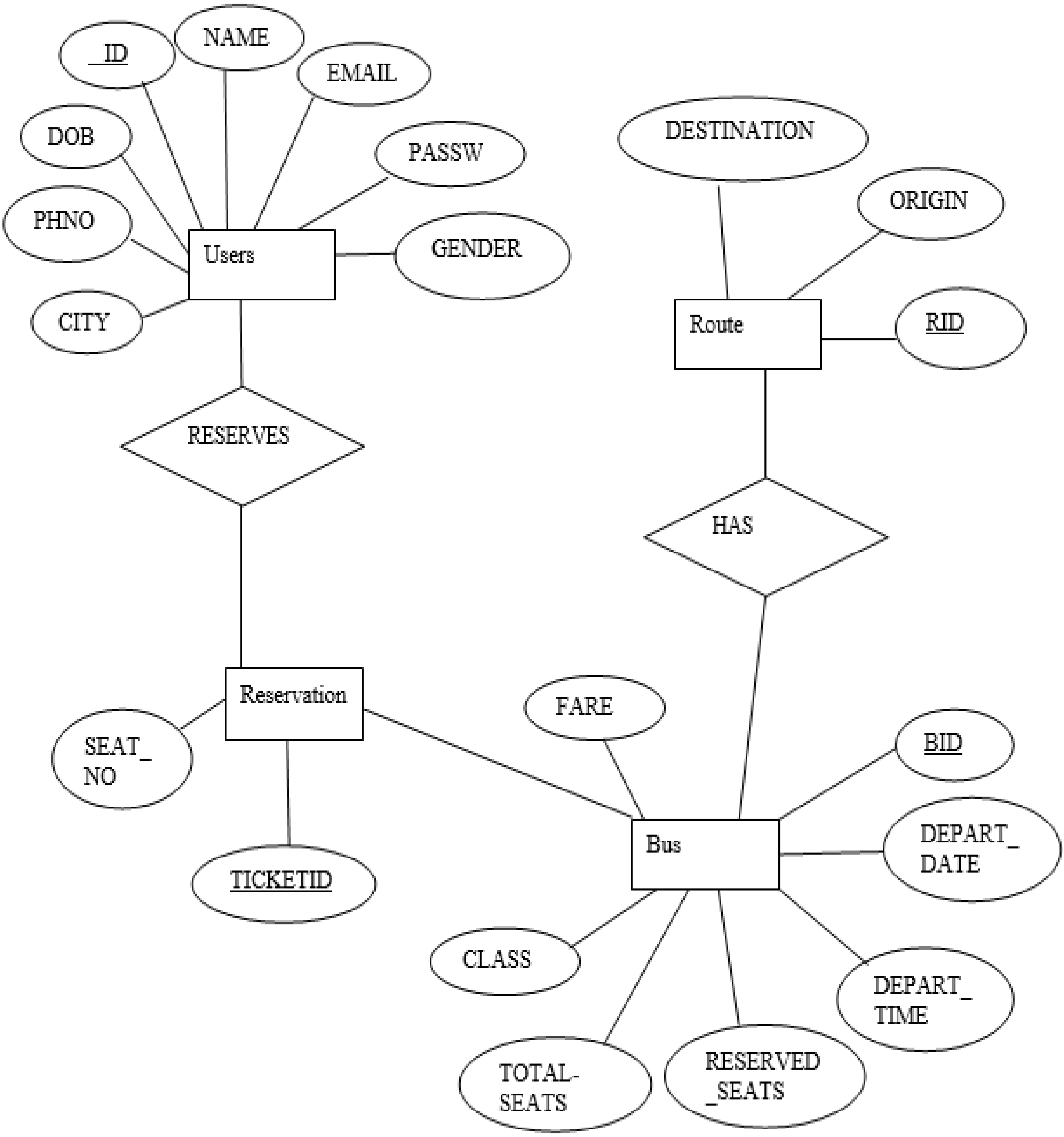
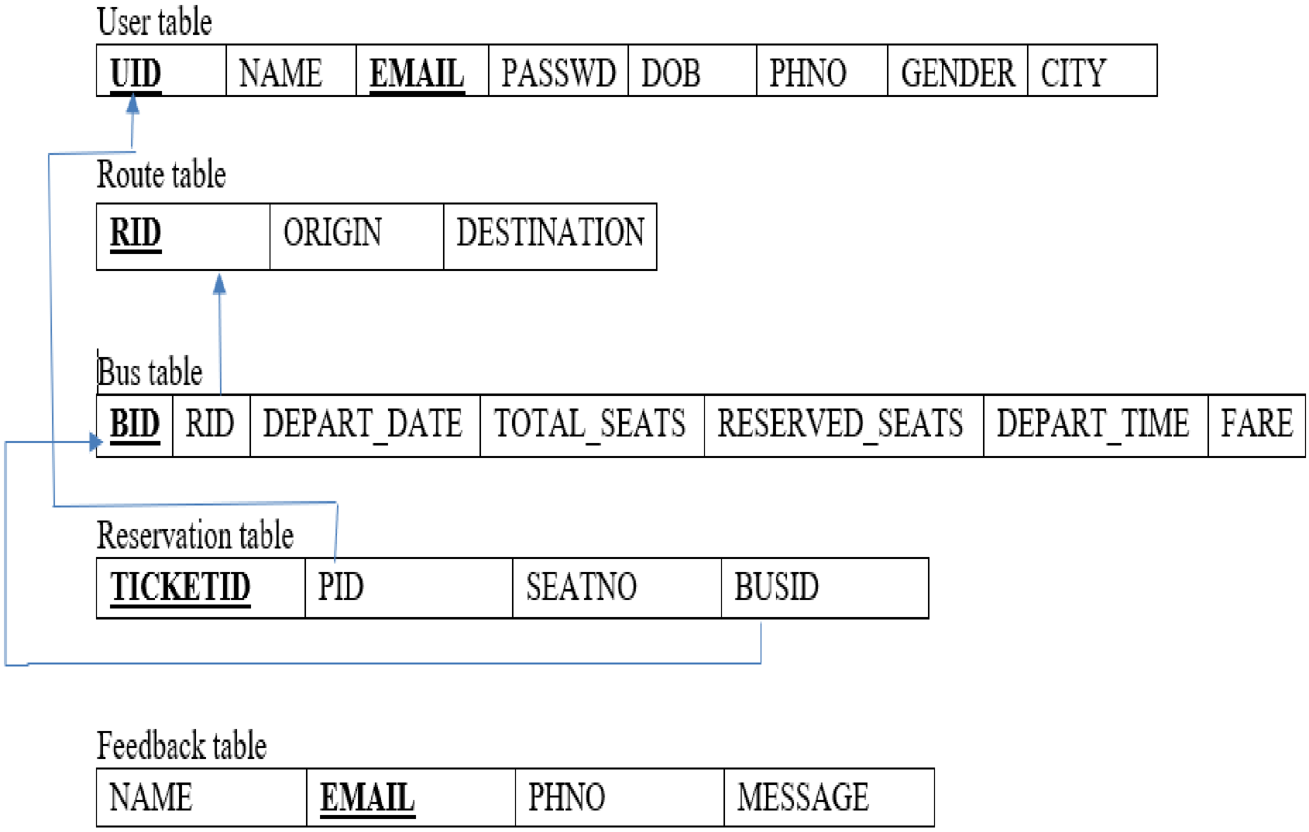


Fig. 3.2.2, ER diagram of Stock Market System

3.3 RELATIONAL SCHEMA

The term "schema" refers to the organization of data as a blueprint of how the database is constructed. The formal definition of a database schema is a set of formulas called integrity constraints imposed on a database. A relational schema shows references among fields in the database. When a primary key is referenced in another table in the database, it is called a foreign key. This is denoted by an arrow with the head pointing at the referenced key attribute. A schema diagram helps organize values in the database. The following diagram shows the schema diagram for the database.



User table

UID	NAME	**EMAIL**	PASSWD	DOB	PHNO	GENDER	CITY

Route table

RID	ORIGIN	DESTINATION

Bus table

BID	RID	DEPART_DATE	TOTAL_SEATS	RESERVED_SEATS	DEPART_TIME	FARE

Reservation table

TICKETID	PID	SEATNO	BUSID

Feedback table

NAME	**EMAIL**	PHNO	MESSAGE

3.4 DESCRIPTION OF TABLES

1. Users: Stores the users registered in the website
   - ID : Unique ID for each user
   - NAME : Name of the user.
   - EMAIL: Email id of the user .
   - PASSWD: Password of the user.
   - DOB: Date of birth of the user.
   - PHNO: Phone number of the user.
   - GENDER: Gender of the user.
   - CITY: City in which the user stays.

2. Route: It stores the details of routes.
   - RID: Unique id for a particular route.
   - ORIGIN: Starting point of the route.
   - DESTINATION: The final destination of the route

3. Bus: It stores the details of the bus
   - BID: Unique for a bus.
   - RID: Company of which shares were bought
   - DEPART_DATE: Date of departure.
   - DEPART_TIME: Time of departure
   - TOTAL_SEATS: Total seats available in the bus.
   - RESERVED_SEATS: Number of seats reserved
   - FARE: Fare for one ticket.

4. Reservation: It stores the details of the reservation.
   - UID : Unique identification number given to the user.
   - BUSID: Id of the bus.
   - TICKETID: Ticket number .
   - SEAT_NO: Seat identification number.

5. Feedback: It stores the details of the reservation.
   - NAME: Name of the user.
   - EMAIL: Email id of the user giving feedback.
   - PHNO: Phone number of the user giving feedback .
   - MESSAGE: Feedback of the user.

Chapter 4

IMPLEMENTATION

4 .1. IMPLEMENTATION

The project is implemented in HTML, JavaScript and CSS for Front-End (website) and MySQL for Back-End (database). It has three-tier architecture with Front-End forming Application Layer and Back-End forming Middle Layer and Database. PHP is used to communicate with the local server.

4.2 MODULES AND THEIR ROLES

4.2.1 Login: Login for the new user.

```
<?php
     session_start();
     $con = mysqli_connect("localhost","root","","mini_project");
     if(!$con)
         {
     die('Could not connect: ' . mysql_error());
     }

     if($_SERVER['REQUEST_METHOD'] === 'POST')
    {
        $query="SELECT * FROM users WHERE EMAIL='$_POST[Emailid]' AND
PASSWD='$_POST[Password]'";

     $result=mysqli_query ($con , $query);
     $value=mysqli_fetch_assoc($result);
     $numrows=mysqli_num_rows($result);
    if($numrows!=0)
   {
           $_SESSION['userid']=$value['ID'];
          header("location:dashboard.php");
    }
    else
   {
           echo "<link rel='stylesheet' href='css/style.css'>";
           echo "<script language='javascript'>";
          echo "alert('WRONG INFORMATION')";
                 }
       }
 ?>
```

4.2.2 Booking tickets:

```
 <?php
  $con = mysqli_connect("localhost","root","","mini_project");
  if (mysqli_connect_errno()) {
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
    exit();
  }
  if($_SERVER['REQUEST_METHOD'] === 'POST'){
  if(isset($_POST['Submit'])){
      $result=mysqli_query($con, "SELECT *
                         FROM route
                         WHERE origin='$_POST[Source]' and
                         destination='$_POST[Destination]'");
      $row=mysqli_fetch_assoc($result);
      $route_id=$row['RID'];
      $result1=mysqli_query($con,"SELECT  *
                                      FROM bus
                                      WHERE RID=$route_id and
                                      (TOTAL_SEATS-RESERVED_SEATS)>='1'
                                       AND DEPART_DATE='$_POST[date]'
                                        AND CLASS='$_POST[Class]'");
      $row1=mysqli_fetch_assoc($result1);
 if(!$row1){
     echo "<p style='font-size:19px;color:black;background-color:white;padding:10px;
     margin-top:20px;border-radius:5px;width:75%'>
      No buses found for the given details. </p>";
  }

  else{
    $bus_id=$row1['BID'];
    $reserved=$row1['RESERVED_SEATS'];
     $result3=mysqli_query($con,"UPDATE bus
                                     SET RESERVED_SEATS=RESERVED_SEATS+1
                                     WHERE BID=$bus_id");
     $uid=$_SESSION['userid'];

     $result4=mysqli_query($con,"INSERT INTO reservation(UID,BID,SEAT_NO)
                                      VALUES ('$uid','$bus_id','$reserved'+1)");

     $result5=mysqli_query($con,"SELECT * FROM reservation where
                                       TICKETNO=(SELECT LAST_INSERT_ID())");
     $row5=mysqli_fetch_assoc($result5);
     $ticketnumber=$row5['TICKETNO'];
     echo "<p style='background-color:white;font-size:19px;padding-left:15px;padding-
top:20px;width:70%;margin-top:30px;border-radius:5px;padding-bottom:20px;'>Your
booking is successfull!!";

    echo "</p> ";
     echo "</p>";
```

```
      $result6=mysqli_query($con,"SELECT                                              c
r.TICKETNO,r.SEAT_NO,b.BID,rt.ORIGIN,rt.DESTINATION,b.DEPART_DATE
,b.DEPART_TIME,b.CLASS,b.FARE FROM bus b, reservation r,route rt
WHERE rt.RID=b.RID AND r.BID=b.BID AND r.TICKETNO='$ticketnumber'");
      $row6=mysqli_fetch_assoc($result6);
      echo "<table style='width:90%';>";
      echo "<tr style='border:3px solid black'>
      <th>Ticket number</th>
      <th>Seat Number</th>
      <th>Bus Id</th>
      <th>Origin</th>
      <th>Destination</th>
      <th>Date of Departure</th>
      <th>Departure Time</th>
      <th>Class</th>
      <th>Fare</th>
          </tr>";

      echo "<tr><td>";
      echo $row6['TICKETNO'];
      echo "</td><td>";
      echo $row6['SEAT_NO'];
      echo "</td><td>";
      echo $row6['BID'];
      echo "</td><td>";
      echo $row6['ORIGIN'];
      echo "</td><td>";
      echo $row6['DESTINATION'];
      echo "</td><td>";
      echo $row6['DEPART_DATE'];
      echo "</td><td>";
     echo $row6['DEPART_TIME'];
      echo "</td><td>";
      echo $row6['CLASS'];
       echo "</td><td>";
       echo $row6['FARE'];
      echo "</td></tr></table>";
     echo "<p class='goback'><a href='bookinguser.php'>Go Back</a></p>";
    }
    }
    }
    mysqli_close($con);
```

4.2.3 Enquiry

```
  <?php
   $con = mysqli_connect("localhost","root","","mini_project");
   if (mysqli_connect_errno()) {
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
    exit();
  }
  if($_SERVER['REQUEST_METHOD'] === 'POST'){
  if(isset($_POST['Submit'])){
    $result=mysqli_query($con, "SELECT * FROM route
                                                 WHERE origin='$_POST[Source]'
                                                  and destination='$_POST[Destination]'");
    $row=mysqli_fetch_assoc($result);
    $route_id=$row['RID'];
    $result6=mysqli_query($con,"SELECT
                              b.BID,rt.ORIGIN,rt.DESTINATION,b.DEPART_DATE,
                               b.DEPART_TIME,b.CLASS
                               FROM bus b,route rt
                               WHERE rt.RID=b.RID AND rt.RID='$route_id'
                               AND b.DEPART_DATE='$_POST[date]'");
     $num_rows = mysqli_num_rows($result6);

     if($num_rows==0){
         echo    "<span    style='font-size:22px;color:black;background-color:white;padding:5px;
margin-top:20px;border-radius:5px;width:75%'>
          No buses found for the given details.
          </span>";
          echo "<p class='goback'><a href='enquiry.php'>Go Back</a></p>";
      }

     else{
            echo "  <div class='available'>Available Buses are:</div>";


            echo "<table style='width:90%';>";
            echo "<tr style='border:3px solid black'>
           <th>Bus Id</th>
           <th>Origin</th>
           <th>Destination</th>
           <th>Date of Departure</th>
           <th>Departure Time</th>
          <th>Class</th>
          </tr>";
          while(  $row6=mysqli_fetch_assoc($result6)){
          echo "<tr><td style='border-left:2px solid black'>";
          echo $row6['BID'];
           echo "</td><td>";
           echo $row6['ORIGIN'];
           echo "</td><td>";
           echo $row6['DESTINATION'];
         echo "</td><td>";
```

```
            echo $row6['DEPART_DATE'];
            echo "</td><td>";
            echo $row6['DEPART_TIME'];
            echo "</td><td>";
            echo $row6['CLASS'];
            echo "</td></tr>";

    }
    echo "</table>";
      echo "<p class='goback'><a href='enquiry.php'>Go Back</a></p>";
    }
   }
   }?>
```

4.2.4 Cancellation

```
<?php
$con = mysqli_connect("localhost","root","");
if(!$con)
    {
die('Could not connect: ' . mysqli_error());
}
mysqli_select_db($con,'mini_project');
if($_SERVER['REQUEST_METHOD'] === 'POST'){
if(isset($_POST['Submit'])){
 $ticketnum=$_POST['tno'];
$res=mysqli_query($con,"SELECT *
                                          FROM reservation
                                         WHERE TICKETNO = '$ticket num'") ;
$nrows=mysqli_num_rows($res);
if($nrows!=0){
 $query="DELETE FROM reservation WHERE TICKETNO='$ticketnum'";



 $query1="UPDATE bus SET RESERVED_SEATS=RESERVED_SEATS-1
                WHERE    RESERVED_SEATS>0
                AND
             BID in (SELECT BID FROM reservation WHERE TICKETNO='$ticketnum')";
 $result=mysqli_query($con,$query1);
 if(mysqli_query($con,$query))
       {
         echo "<p style='background-color:white;font-size:19px;padding-left:15px;
padding-top:20px;width:70%;margin-top:30px;border-radius:5px;
padding-bottom:20px;'>Booking for ticket number ".$ticketnum . " cancelled successfully.</p>";
            echo "<p class='goback'><a href='cancellation.php'>Go Back</a></p>";
        }
}
else{
  echo "<p style='background-color:white;font-size:19px;padding-left:15px;
padding-top:20px;width:70%;margin-top:30px;border-radius:5px;padding-bottom:20px;'>The entered ticket number ".$ticketnum . " does not exist.</p>";
    echo "<p class='goback'><a href='cancellation.php'>Go Back</a></p>";


}
     }
   }
 ?>
```

4.1.4 Register

```
<?php
        $validate=true;
        $error = "";
        $email = "";
        $date = "mm/dd/yyyy";

        if($_SERVER['REQUEST_METHOD'] === 'POST'){

       if(isset($_POST['submitpost'])){
        $con = mysqli_connect("localhost","root","");
        if(!$con)
            {
        die('Could not connect: ' . mysqli_error());
        }

        mysqli_select_db($con,'mini_project');
        $email = trim($_POST["Email"]);
        $date = trim($_POST["date"]);
        $q1 = "SELECT * FROM users WHERE EMAIL = '$email'";
        $r1 = $con->query($q1);
        // if the email address is already taken.
        if($r1->num_rows > 0)
        {
          $validate = false;
        }
        if($validate == true)
      {
          $dateFormat = date("Y-m-d", strtotime($date));
        $query = "INSERT INTO
                        users(PASSWD,NAME,PHNO,GENDER,EMAIL,DOB,CITY)
VALUES('$_POST[Password]','$_POST[Name]','$_POST[phno]','$_POST[gender]',
'$_POST[Email]','$_POST[date]','$_POST[city]')";
       $r2 = mysqli_query($con,$query);

         if ($r2 === true)
         {
            // header("Location: Login.php");
            echo "<script>alert('Thank you for Submitting the details. Click OK to
Continue');</script>";
               mysqli_close($con);
             exit();
         }
     }
     else
     {
         $error = "email address is not available. Signup failed.";
          echo "<script>alert('An account already exists for the given email id.Signup
failed.');</script>";
         mysqli_close($con);
     }
    }
      }
      ?>
```

4.3 RESULT

The resulting system is able to:

- Authenticate user credentials during login.
- Salted encryption for security of user passwords.
- Register new users.
- Allow users to enquire buses in the desired route and date.
- Allow user to book tickets by specifiying the route, date and class.
- Ability to cancel tickets.

Chapter 5

TESTING

5.1 SOFTWARE TESTING

Testing is the process used to help identify correctness, completeness, security and quality of developed software. This includes executing a program with the intent of finding errors. It is important to distinguish between faults and failures. Software testing can provide objective, independent information about the quality of software and risk of its failure to users or sponsors. It can be conducted as soon as executable software (even if partially complete) exists. Most testing occurs after system requirements have been defined and then implemented in testable programs.

5.2 MODULE TESTING AND INTEGRATION

Module testing is a process of testing the individual subprograms, subroutines, classes, or procedures in a program. Instead of testing whole software program at once, module testing recommend testing the smaller building blocks of the program. It is largelywhite box oriented. The objective of doing Module testing is not to demonstrate proper functioning of the module but to demonstrate the presence of an error in the module. Module testing allows implementing of parallelism into the testing process by giving the opportunity to test multiple modules simultaneously.

The final integrated system too has been tested for various test cases such as duplicate entries and type mismatch.

5.3 LIMITATIONS

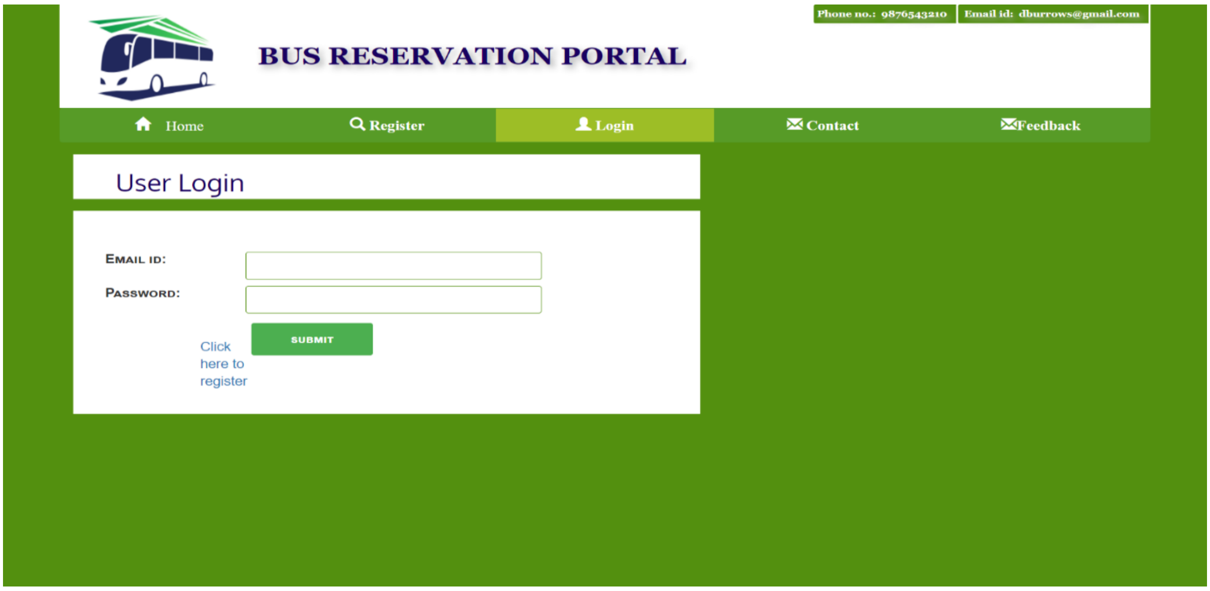
- A user once registered cannot make any changes to his details like address, phone number, email etc.
- Payment verification steps needs to be added as the users make their transactions
- A user once logged in can book only one ticket at a time.
- The number of routes and buses are limited.
- This project doesn’t have admin login, any changes to be made has to be done in the back end.

Chapter 6

SNAPSHOTS

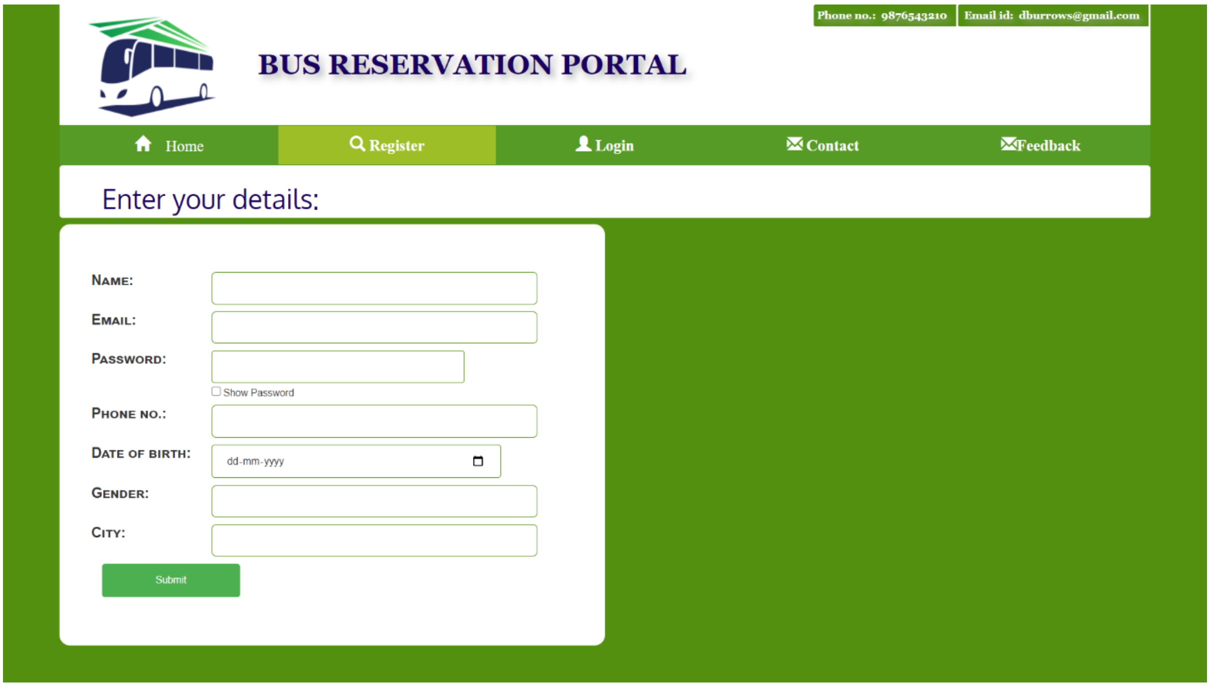
This chapter consists of working screenshots of the project.

6.1 LOGIN PAGE



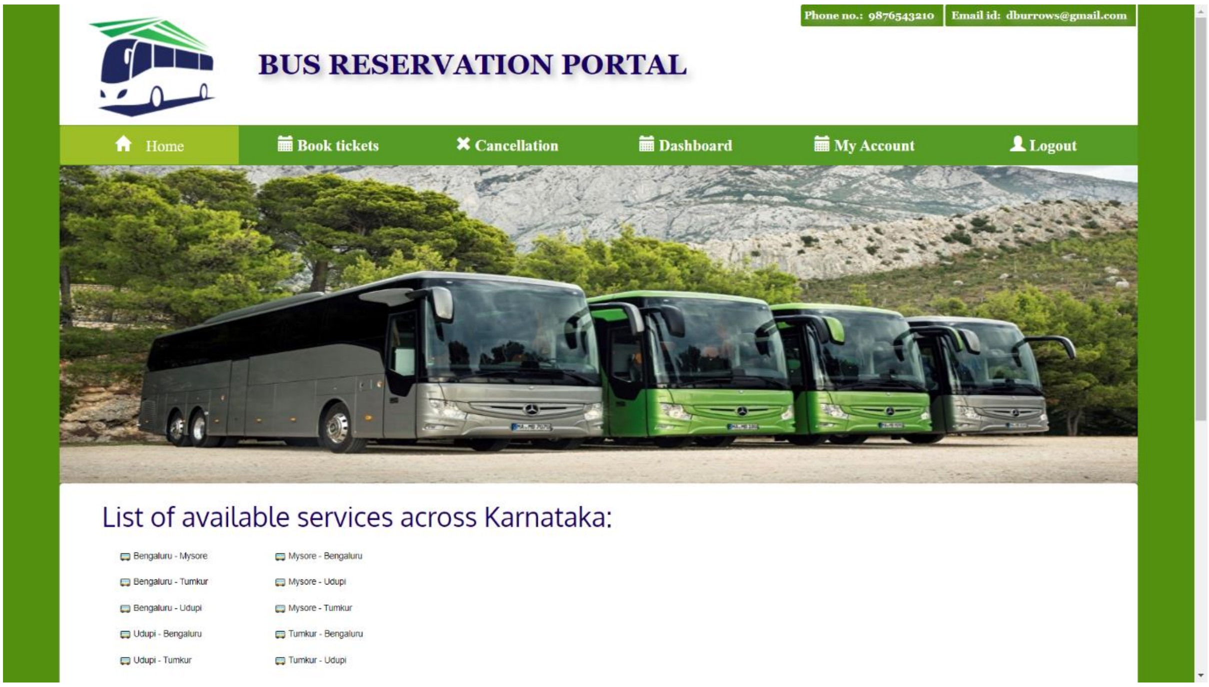
This is the login page for existing users and is the first page shown to any customer.

6.2 REGISTRATION PAGE



This is the registration page for any new users.

6.3 HOME PAGE



Phone no.: 9876543210 Email id: dburrows@gmail.com

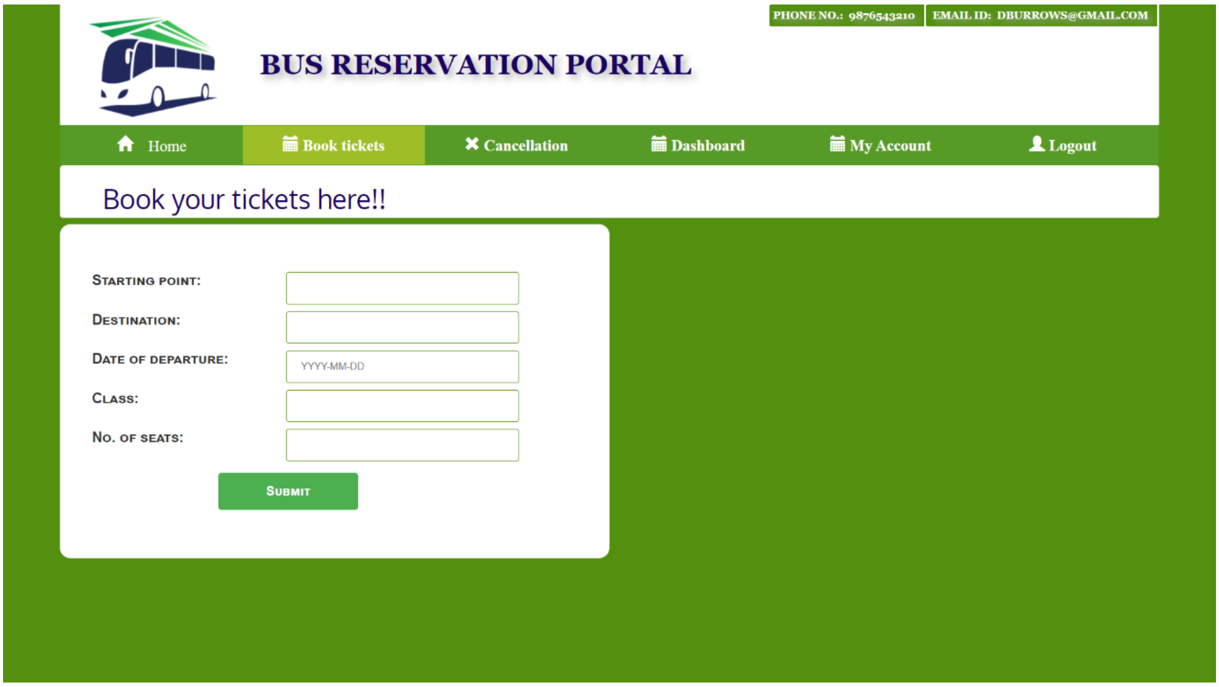
BUS RESERVATION PORTAL

List of available services across Karnataka:

- Bengaluru - Mysore
- Bengaluru - Tumkur
- Bengaluru - Udupi
- Udupi - Bengaluru
- Udupi - Tumkur
- Mysore - Bengaluru
- Mysore - Udupi
- Mysore - Tumkur
- Tumkur - Bengaluru
- Tumkur - Udupi

First home page shown to customers after login.

6.4 BOOKING



PHONE NO.: 9876543210 EMAIL ID: DBURROWS@GMAIL.COM

BUS RESERVATION PORTAL

Book your tickets here!!

STARTING POINT:

DESTINATION:

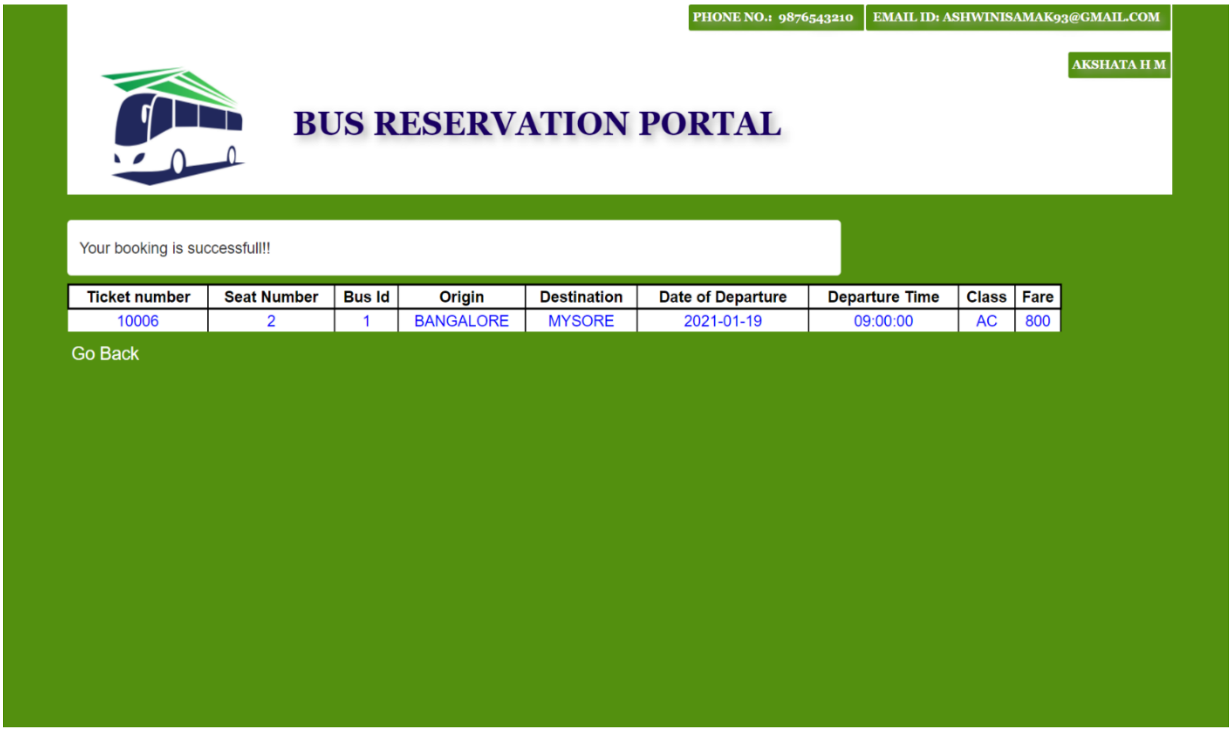
DATE OF DEPARTURE: YYYY-MM-DD

CLASS:

NO. OF SEATS:

This is the page for booking tickets.

6.5 BOOKING CONFIRMATION PAGE

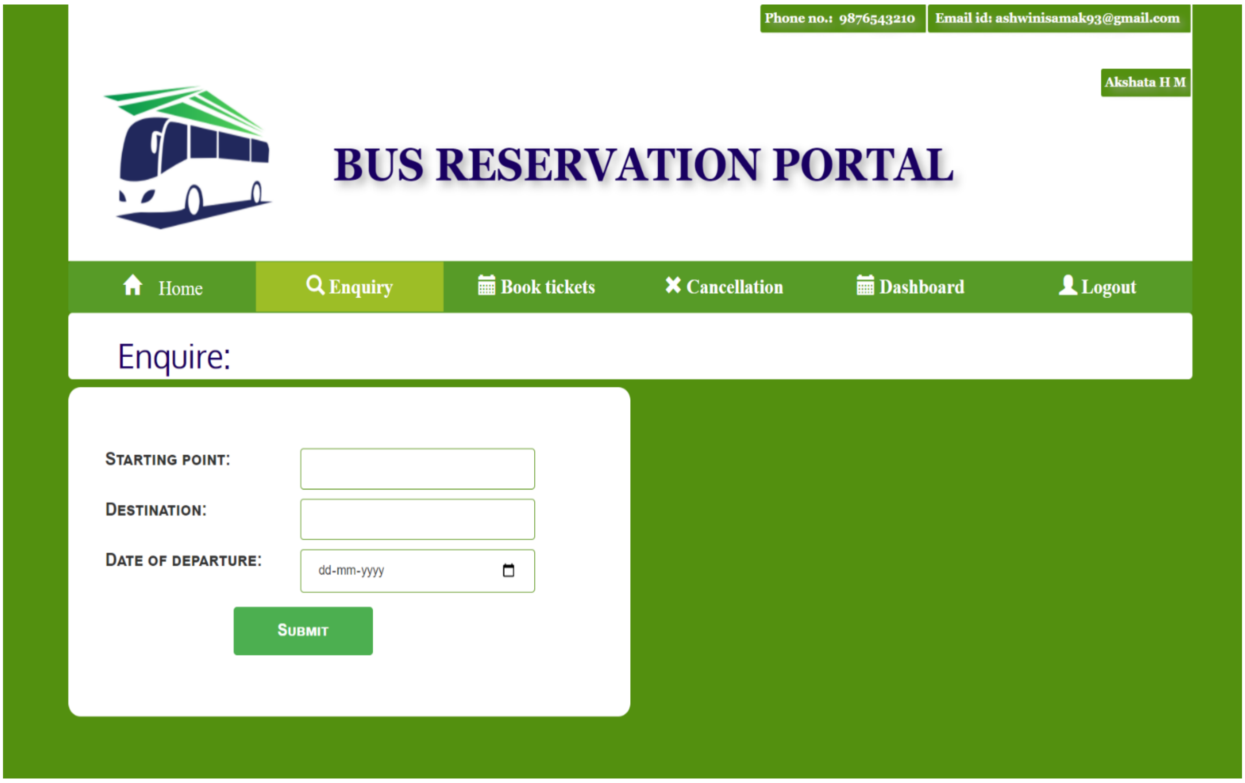


Your booking is successfull!!

Ticket number	Seat Number	Bus Id	Origin	Destination	Date of Departure	Departure Time	Class	Fare
10006	2	1	BANGALORE	MYSORE	2021-01-19	09:00:00	AC	800

This is the confirmation page after booking.

6.6 ENQUIRY PAGE



Enquire:

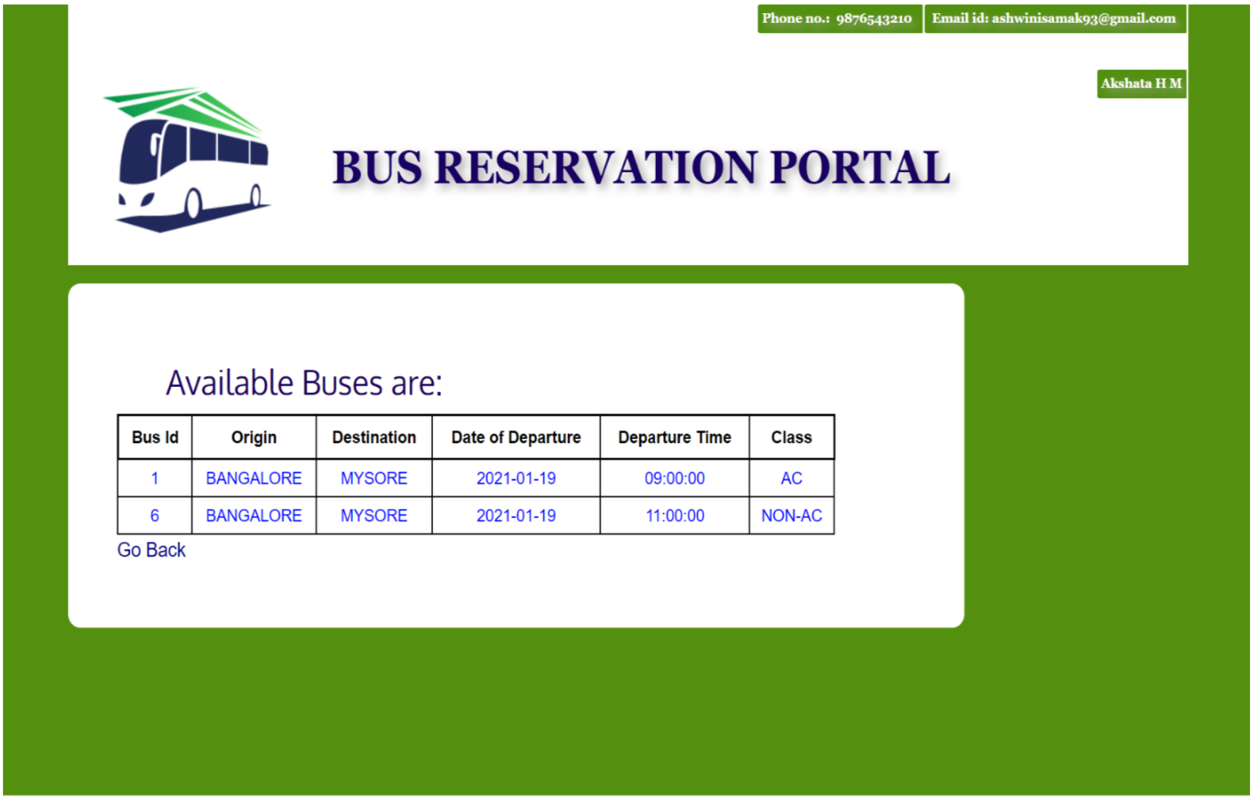
STARTING POINT:

DESTINATION:

DATE OF DEPARTURE: dd-mm-yyyy

This is the page for enquiry of available buses.

6.7 ENQUIRY INFORMATION

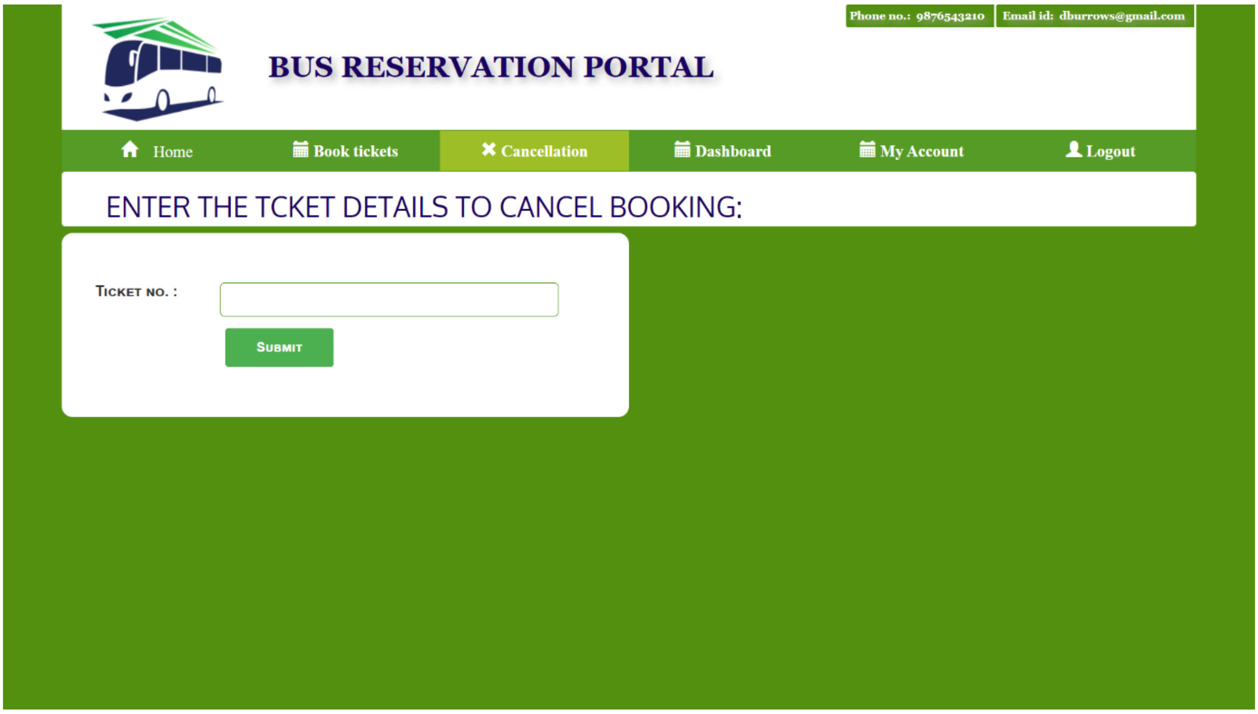


Available Buses are:

Bus Id	Origin	Destination	Date of Departure	Departure Time	Class
1	BANGALORE	MYSORE	2021-01-19	09:00:00	AC
6	BANGALORE	MYSORE	2021-01-19	11:00:00	NON-AC

Go Back

6.8 CANCEL BOOKINGS

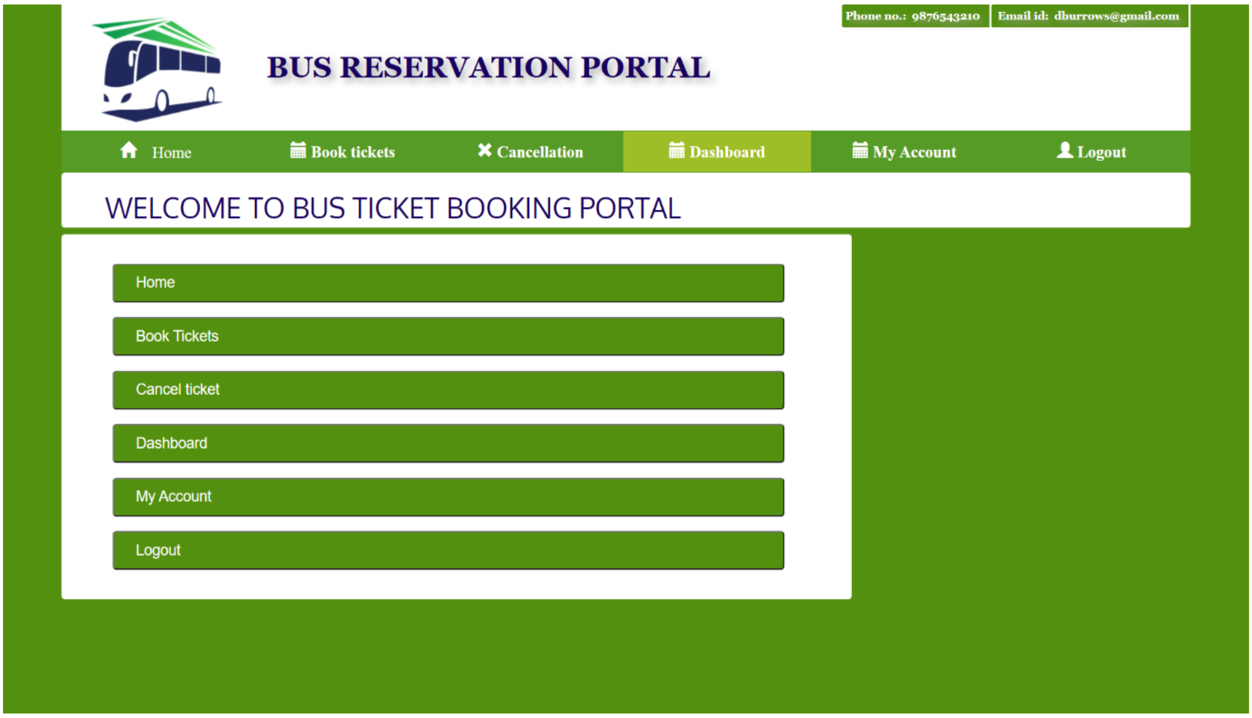


ENTER THE TCKET DETAILS TO CANCEL BOOKING:

Ticket no. :

This page is for cancelling tickets.

6.9 DASHBOARD PAGE



Phone no.: 9876543210

Email id: dburrows@gmail.com

BUS RESERVATION PORTAL

WELCOME TO BUS TICKET BOOKING PORTAL

CONCLUSION

Conclusion

This project is developed successfully and the performance is found to be satisfactory. This project is designed to meet the requirements of assigning jobs. It has been developed in PHP and the database has been builtin My SQL server keeping in mind the specifications of the system. The user will be able to book the ticket using this website.. I have designed the project to provide the user with easy retrieval of data, details of theatre and necessary feedback as much as possible.

To implement this as a web application we used php as the technology. Php has advantages such as enhanced performance, scalability, built- in security and simplicity. To build any web application using PHP we need a programming language such as PHP and so on. MySQL was used as back-end database since it is one of the most popular open source databases, and it provides fast data access, easy installation and simplicity. For front end we used HTML and CSS.

FUTURE ENHANCEMENTS

Future upgrades to this project will implement:

- Facility where the user can book multiple seats at a time.
- Admin login and control part to be added from where the admin can have a control of the database.
- Additional feature where the user can get discounts on booking tickets.
- Addition of more routes and buses.

REFERENCES

1. Ramakrishnan, R., & Gehrke, J. (2011). Database management systems. Boston: McGraw-Hill.

2. Monson-Haefel, R. (2007). J2EE Web services. Boston, Mass: Addison-Wesley. Silberschatz A., Korth H. F., & Sudarshan S. (2011).

3. Database systems concepts. Estados Unidos: McGraw-Hill Companies, Inc.

4. Hanna P. (2002): JSP 2.0 The Complete Reference, Second Edition McGraw Hill Education.

5. David F. (2011). JavaScript: The Definitive Guide Sixth edition.

6. https://www.w3schools.com

7. https://www.canvasjs.com

8. https://getbootstrap.com/

9. https://fontawesome.com

10. https://www.javatpoint.com/