## RAILWAY RESERVATION MANAGEMENT SYSTEM

*A Project Report submitted in the partial fulfillment of the Requirements for the award of the degree*

## BACHELOR OF TECHNOLOGY

**in**

## COMPUTER SCIENCE & ENGINEERING

Submitted by

## A.Bala Keerthimai (18471A05C4)

## K.Sruthi (18471A05E5)

## G.Sri Gowri Poojitha (18471A05D8)

## T.Vishnu Priya (18471A05H8)

Under the esteemed guidance of

**P. Lakshmi Narayana M.Tech..,**

**Asst.Prof**



## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**NARASARAOPETA ENGINEERING COLLEGE (AUTONOMOUS)**

**(Affiliated to JNTUK, Kakinada, Approved by AICTE &Thrice Accredited by NBA)**

**2020 - 2021**

**NARASARAOPETA ENGINEERING COLLEGE**

**(AUTONOMOUS)**

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



**CERTIFICATE**

**This is to certify that the project that is entitled with the name “**RAILWAY RESERVATION MANAGEMENT SYSTEM**” is a bonafide work done by the team A.Bala Keerthimai (18471A05C4), G.Sri Gowri Poojitha(18471A05D8), T.Vishnupriya (18471A05H8), K.Sruthi(18471A05E5) in partial fulfillment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in the Department of COMPUTER SCIENCE AND ENGINEERING during 2019-2020.**

PROJECT GUIDE PROJECT CO-ORDINATOR

**P.Lakshmi Narayana, M.Tech Roopa Tirumalasetty M.Tech..,**

**Asst.Prof Asst.Prof**

HEAD OF THE DEPARTMENT EXTERNAL EXAMINER

**Dr. S. N. TirumalaRao, M.Tech., Ph.D**

**Professor**

# ACKNOWLEDGEMENT

We wish to express my thanks to carious personalities who are responsible for the completion of the project. We are extremely thankful to our beloved chairman sri **M.V.Koteswara Rao,B.Sc.,** who took keen interest in us in every effort throughout this course. We owe out sincere gratitude to our beloved principal **Dr.M.Sreenivasa Kumar, M.Tech., Ph.D., MISTE., FIE(I).,** for showing his kind attention and valuable guidance throughout the course.

We express our deep felt gratitude towards **Dr.S.N.Tirumala Rao, M.Tech.,Ph.D.,** HOD of CSE department and also to our guide **P.Lakshmi Narayana M.Tech., Assistant Professor** of CSE department whose valuable guidance and unstinting encouragement enable us to accomplish our project successfully in time.

We extend our sincere thanks towards **Roopa Tirumalasetty M.tech.., Assistant** **professor** & Project coordinator of the project for extending her encouragement. Their profound knowledge and willingness have been a constant source of inspiration for us throughout this project work.

We extend our sincere thanks to all other teaching and non-teaching staff to department for their cooperation and encouragement during our B.Tech degree.

We have no words to acknowledge the warm affection, constant inspiration and encouragement that we received from our parents.

We affectionately acknowledge the encouragement received from our friends and those who involved in giving valuable suggestions had clarifying out doubts which had really helped us in successfully completing our project.

By

A.Bala Keerthimai(18471A05C4)

K.Sruthi(18471A05E5)

G.Sri GowriPoojitha(18471A05D8)

T.Vishnu Priya(18471A05H8)

**ABSTRACT**

In India train is the preferred mode of transport most widely used by all class of people. The present system needs much advancement in railway communication system in the area of journey reservation, safety and rescue operations. It is pertinent to bring out here the present expectation of passenger should be of interactive mode and user friendly.

The Railway Reservation System facilitates the passengers to enquire about the trains available on the basis of source and destination, Booking and Cancellation of tickets, enquire about the status of the booked ticket, etc. The aim of case study is to design and develop a database maintaining the records of different trains, train status, and passengers.

This project contains Introduction to the Railways reservation system.It enables users to login for reserving tickets,if they are not registered users they must register to book the tickets.It is the computerized system of reserving the seats of train seats in advanced. It is mainly used for long route. On-line reservation has made the process for the reservation of seats very much easier than ever before.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | **INDEX** |  |
| **S. No** | **Contents** |  | **Page No** |
| I | List of Figures |  | IV |
| II | List of Tables |  | V |
| 1 | Introduction |  | 1 |
| 2 | System Analysis |  | 2 |

|  |  |
| --- | --- |
| 2.1 | Literature Survey |
| 2.2 | Existing System |
| 2.3 | Proposed System |
| 2.4 | System Requirements |

|  |
| --- |
| 2 |
| 2 |
| 3 |
| 4 |

1. System Design

|  |
| --- |
| 4 |
| 5 |
| 6 |
| 10  17 |

|  |  |
| --- | --- |
| 3.1 | Module Description |
| 3.2 | Database design |
| 3.3  3.4  3.5 | E-R Diagram  Database Tables  Uml diagrams |

1. Implementation 21
2. Testing and Test Cases

|  |
| --- |
| 22 |
| 23 |
| 23 |
| 23 |
| 24 |
| 26 |

|  |  |
| --- | --- |
| 5.1 | Introduction |
| 5.2 | Testing Strategies |
| 5.3 | Unit Testing |
| 5.4  5.5 | Integration Testing  System Testing |
| 5.6 | Test Cases |

1. Output Screens 30
2. Conclusion 41
3. Future Enhancements 42
4. Bibliography 43

# List of Figures

|  |  |  |
| --- | --- | --- |
| **S. no** | **Figure** | **Page No** |
| 1 | Fig : 3.2 : Database diagram | 5 |
| 2 | Fig : 3.3 : E-R Diagram | 6 |
| 3 | Fig : 3.5.1 : Use case diagram for user | 11 |
| 4 | Fig : 3.5.2 : Use case diagram for admin | 12 |
| 5 | Fig : 3.5.3: Class Diagram | 13 |
| 6 | Fig : 3.5.3 : Sequence diagram for user | 14 |
| 7 | Fig : 3.5.4 : Sequence diagram for administrator | 15 |
| 8 | Fig :3.5.4 :Activity diagram for user | 16 |
| 9 | Fig :3.5.4 : Activity diagram for administrator | 17 |
| 10 | Fig : 5.5.1 : Admin login test case 1 | 27 |
| 11 | Fig : 5.5.2 : Admin login test case 2 | 28 |
| 12 | Fig : 5.5.3 : User login test case 3 | 29 |
| 13 | Fig : 5.5.4 : User registration test case 4 | 30 |
| 14 | Fig : 5.5.5 : User registration test case 5 | 31 |
| 15 | Fig : 6.1 : Home Page | 32 |
| 16 | Fig : 6.2 : Registration Page | 33 |
| 17 | Fig : 6.3 : User login page | 34 |
| 18 | Fig : 6.4 :Admin login page | 35 |
| 19 | Fig : 6.5 : Add train page | 36 |
| 20 | Fig : 6.6 : Add station page | 37 |
| 21 | Fig : 6.7 :Add route page | 38 |
| 22  23 | Fig : 6.8 :Booking Tickets page  Fig : 6.9:Train details page | 39  40 |

* 1. **List of Tables**

|  |  |  |
| --- | --- | --- |
| **S. no** | **Table** | **Page No** |
| 1 | Table: 3.4.1: Train Details Table | 7 |
| 2 | Table: 3.4.2 :Station Details Table | 7 |
| 3 | Table : 3.4.3 :Route Details Table | 8 |
| 4 | Table : 3.4.4 : Booking Details Table | 8 |
| 5 | Table : 3.4.5 : User Details Table | 9 |
| 6 | Table : 3.4.6: Admin Details Table | 9 |
| 7  8 | Table : 3.4.7 :Passenger Details Table  Table : 5.5 : Test cases data Table | 10  26 |
|  |  |  |
|  |  |  |

# 1. INTRODUCTION

Online Railway Reservation System explains how booking is done in train . Generally in a train booking can be of two types:-

1. Online Booking.
2. Counter Booking

The Railway Reservation maintains the booking details in a Database .It contains the details of different trains with A.C, non-A.C with facilities which might be further classified. The Database also consists of a record of passenger bookings that are already booked. This project not only maintains the record of passenger bookings but also reduces the paper work in the present system.

There is a reception information counter attended by highly qualified and trained persons .

## Online Booking:

With the help of this people can book their tickets online through internet, sitting in their home by a single click of mouse. Using their credit cards people can easily get their tickets done within minutes.

## Counter Booking:

This is the oldest method of booking the tickets. The reservation counters are there at railway department from where people can get the tickets to their respective destinations.

# 2. SYSTEM ANALYSIS

## LITERATURE SURVEY

The scope of “ONLINE RAILWAY RESERVATIONSYSTEM” is to display the train details and the classes provided in that train .The administrator can see the user profile who are registered for booking a trains. Here admin is going to maintain the traindetails, passenger details and registration details of the user who are registered for booking the train.

## EXISTING SYSTEM

Before computerization, it was manual register entry and long queues.

The booking clerk would see the availability, if not available the customer asks for next best date.

There were different queues for different destinations, people had to stand in queue where the register being used

To overcome these circumstances we developed online railway reservation system

## DISADVANTAGES

Time consuming process.

There is no surety of availability of booking train.

The existing system may not provide security data base. Chances of human errors.

## PROPOSED SYSTEM

The project “Online Railway Reservation System” is developed in ASP.net.

The main aim of this project is to book tickets through online and cancellation of tickets without going to railway stations.

And through this we also know about the train status like at which time it will leave source station and at which time it will reach the destination.

Using this application a user can get every information about train.

# SYSTEM REQUIREMENTS

## HARDWARE REQUIREMENTS

Processor : Intel® Core™ i3 CPU

Hard Disk : 20 GB or more

Monitor : 15 VGA colour

Mouse : Optical Mouse

Keyboard : 110 keys enhanced

RAM : 1GB or more

## SOFTWARE REQUIREMENTS

Operating System : Windows 8 or Windows 10 Language : ASP.Net, C#

Front End : Microsoft Visual Studio 2013

Back End : SQL Server Management Studio 2012

# 3.DESIGN

## MODULE DESCRIPTION

The system consists of 2 modules. They are:- 1.Administrator

1. User

### ADMINISTRATOR MODULE

Administrator can access all information with some given username and password. The admin has special rights, like he can insert, update and delete train information, he can view the registration details of the user.

### USER MODULE

User can fill registration form.

Already Registered user can login with his/her own user id and password. Get information about reserved train(if train is already reserved).

Check train availability and user can view their profile. He can enter their details for booking the train.

## DATABASE DESIGN

Design creates a representation or model, provides details about software data structure, architecture, interfaces and components that are necessary to implement a system. Design is the perfect way to accurately translate a customers requirement in to a finished software product.

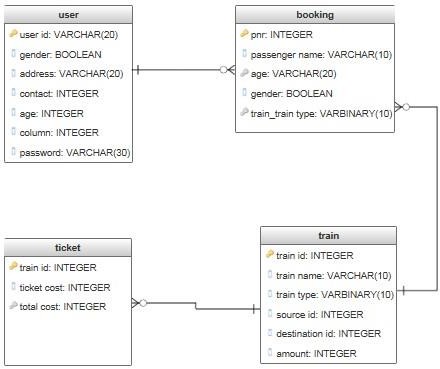


Fig:3.2. Database Diagram

## ER-DIAGRAM

An Entity-Relationship diagram is a specialized graphic that illustrates the interrelationships between entities in a database. These diagrams use symbols to represent different types of information.

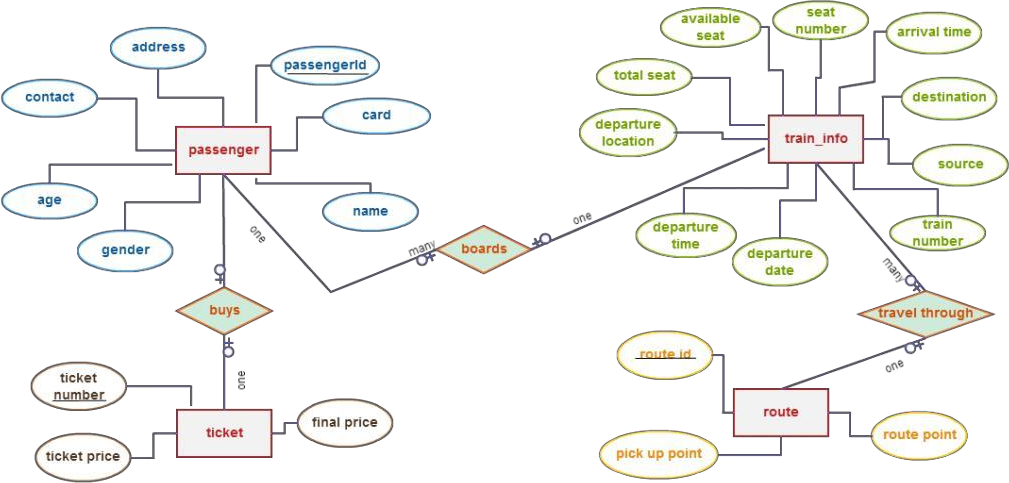


Fig:3.3. ER-Diagram

## DATABASE TABLES

### TRAIN TABLE

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Constraint** |
| Train id | Int | Primarykey |
| Train name | Varchar(30) | - |
| Train type | Varchar(20) | - |
| Source stn | Int | Foreign key |
| Destination stn | Varchar(20) | Foreign key |
| Class type | Varchar(20) | - |

Table:3.4.1 Train Table

### STATION TABLE

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Constraint** |
| Station id | Int | Primary key,not null |
| Station name | Varchar(20) | - |

Table:3.4.2Station Table

### ROUTE TABLE

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Constraint** |
| Train id | Int | Primary key |
| Station id | Int | - |
| Arraival time | Varchar(15) | - |
| Departure time | Varchar(15) | - |

Table:3.4.3 Route Table

### BOOKING TABLE

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Constraint** |
| PNR | Int | Primary key |
| DOJ | Date | - |
| Train id | Int | Foreign key |
| Booked by | Varchar(30) | Foreign key |
| Quota | Varchar(10) | - |
| Amt | Decimal(18,2) | - |

Table:3.4.4Booking Table

### USER TABLE

**`**

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Constraint** |
| Email id | Varchar(30) | Primary key |
| Password | Varchar(20) | - |
| Full name | Varchar(30) | - |
| Gender | Varchar(8) | - |
| Age | Int | - |
| Phone no Va | rchar(15) |  |
| City | Varchar(10) |  |
| State | Varchar(15) |  |

Table:3.4.5 User Table

### ADMIN TABLE

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Constraint** |
| User id | Varchar(20) | Primary Key |
| Password | Varchar(20) | - |

Table:3.4.6 Admin Table

## PASSENGER TABLE

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Constraint** |
| PNR | Varchar(10) | Primary key |
| Passenger name | Varchar(20) | - |
| Source name | Varchar(20) | Foreign key |
| Destination name | Varchar(20) | Foreign key |
| DOJ | Date | - |
| Train id | Int | Foreign key |
| Age | Int | - |
| Gender | Varchar(8) | - |
| Booked by | Varchar(20) | Foreign key |

Table:3.4.7 Passenger table

# UML DIAGRAMS

### USECASE DIAGRAM

A **UseCase Diagram** shows a set of use cases, actors and their relationships. Use case diagrams illustrate the static use case view of a system. Use case diagrams are especially important in organizing and modelling the behaviour of a system.

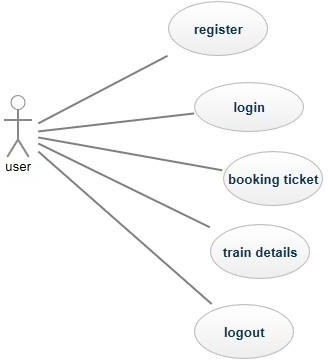


Fig:3.5.1 Usecase diagram for user

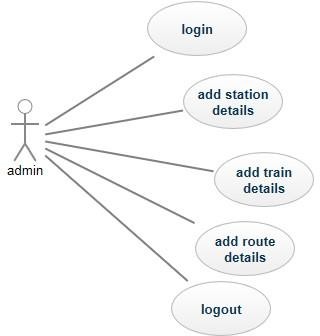


Fig:3.5.1Usecase diagram for admin

### CLASS DIAGRAM

**Class diagram** shows a set of classes, interfaces, collaborations and their relationships. Class diagrams are the most common diagram found in modelling object- oriented systems. Class diagrams illustrate the static design view of a system. Class diagrams that include active classes are used to address the static process view.

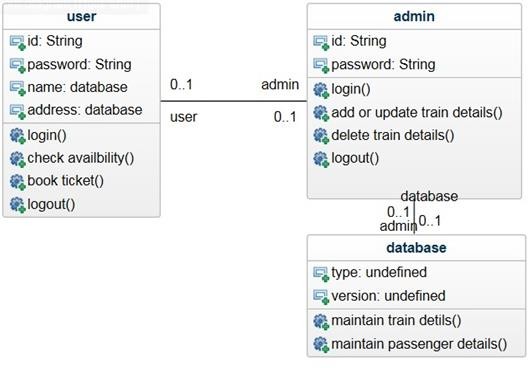


Fig:3.5.2 Class diagram

### SEQUENCE DIAGRAM

A **Sequence diagram** is an interaction diagram that emphasizes the time ordering of messages. A sequence diagram shows a set of objects and the messages sent and received by those objects. Sequence diagrams illustrate the dynamic view of a system.

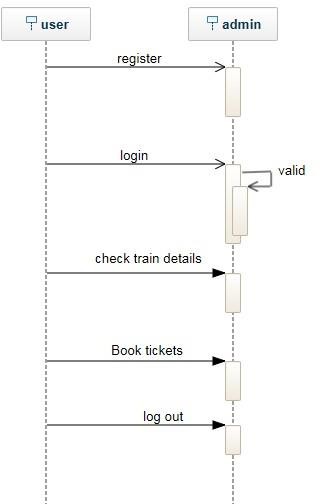


Fig:3.5.3 Sequence diagram for user

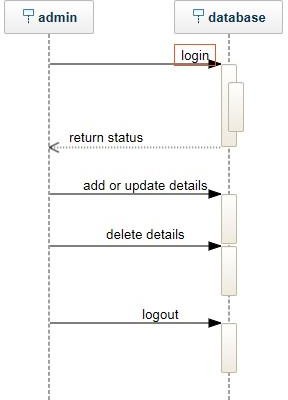


Fig:3.5.3 Sequence diagram for administrator

### ACTIVITY DIAGRAM

An Activity diagram shows the flow from one activity to other activity within a system. Activity diagrams illustrate the dynamic view of a system. Activity diagrams are especially important in modelling the function of a system.

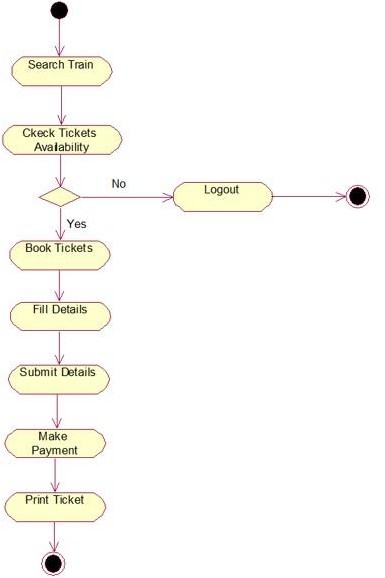


Fig:3.5.4Activity diagram for user

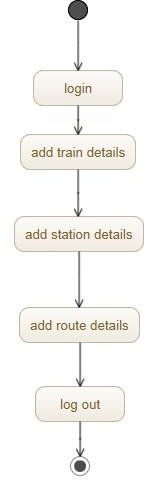


Fig:3.5.4Activity diagram for administrator

# 4.IMPLEMENTATION

## SAMPLE CODE

**User.aspx:**

using System; usingSystem.Collections.Generic; usingSystem.Linq; usingSystem.Web; usingSystem.Web.UI; usingSystem.Web.UI.WebControls;

publicpartialclassuserlogin :System.Web.UI.Page

{

protectedvoidPage\_Load(object sender, EventArgs e)

{

}

protectedvoidbtnLogin\_Click(object sender, EventArgs e)

{

lblMsg.Text = String.Empty;

Stringstmt = "Select count(\*) from [user] ";

stmt += " where email\_id='" + txtUserID.Text + "' and "; stmt += " password='" + txtPwd.Text + "'";

sqlhelperobjSql = newsqlhelper(); intcnt = objSql.MyRecordCount(stmt);

//IF USER EXISTS

if (cnt == 1)

{

Session["emailid"] = txtUserID.Text; Response.Redirect("default3.aspx");

}

else

lblMsg.Text = "Invalid User name or password";

}

protectedvoidlbRegister\_Click(object sender, EventArgs e)

{

Response.Redirect("user.aspx");

}

protectedvoidlblForgerPwd\_Click(object sender, EventArgs e)

{

Response.Redirect("userchangepwd.aspx");

}

}

## Admin.aspx:

using System; usingSystem.Collections.Generic; usingSystem.Linq; usingSystem.Web; usingSystem.Web.UI; usingSystem.Web.UI.WebControls;

publicpartialclassadmin : System.Web.UI.Page

{

protectedvoidPage\_Load(object sender, EventArgs e)

{

}

protectedvoidbtnLogin\_Click(object sender, EventArgs e)

{

if (txtUserID.Text == "administrator"&&txtPwd.Text == "admin")

{

txtUserID.Text = String.Empty; txtPwd.Text = String.Empty; Response.Redirect("default2.aspx");

}

else

lblMsg.Text = "Invalid User name or password";

}

protectedvoidbtnCancel\_Click(object sender, EventArgs e)

{

txtUserID.Text = String.Empty; txtPwd.Text = String.Empty;

}

}

Booking.aspx:

using System; usingSystem.Collections.Generic; usingSystem.Data; usingSystem.Linq; usingSystem.Web; usingSystem.Web.UI; usingSystem.Web.UI.WebControls;

publicpartialclassbooking : System.Web.UI.Page

{

protectedvoidPage\_Load(object sender, EventArgs e)

{

if (!Page.IsPostBack)

{

myCreatePassengersTable();

}

}

privatevoidmyCreatePassengersTable()

{

DataTabledtTemp = newDataTable(); dtTemp.Columns.Add("PassengerName", typeof(String)); dtTemp.Columns.Add("Age", typeof(int)); dtTemp.Columns.Add("Gender", typeof(String)); Session["passengerInfo"] = dtTemp;

}

protectedvoidbtnAddPassenger\_Click(object sender, EventArgs e)

{

divPassengers.Visible = true;

}

protectedvoid btnAdd\_Click1(object sender, EventArgs e)

{

DataTable \_dt = (DataTable)Session["passengerInfo"];

\_dt.Rows.Add(txtPname.Text, int.Parse(txtAge.Text), ddlGender.SelectedItem.Text);

Session["passengerInfo"] = \_dt;

gvPassengers.DataSource = \_dt.DefaultView; gvPassengers.DataBind();

}

protectedvoidbtnBooking\_Click(object sender, EventArgs e)

{

DataTable \_dt = (DataTable)Session["passengerInfo"]; sqlhelperobjSql = newsqlhelper(); objSql.myBulkPassengerDetails(Convert.ToDateTime(txtDoj.Text), int.Parse(ddlTrain.SelectedItem.Value), Session["emailid"].ToString(), ddlQuota.SelectedItem.Value.ToString(), \_dt);

}

protectedvoidbtnPay\_Click(object sender, EventArgs e)

{

lblMsg.Text = "Ticket booked sucessfully";

}

# 5.TESTING AND TEST CASES

## INTRODUCTION

One of the purposes of the testing is to validate and verify the system. Verification means checking the system to ensure that it is doing what the function is supposed to do and Validation means checking to ensure that system is doing what the user wants it to do.

No program or system design is perfect; communication between the user and the designer is not always complete or clear, and time is usually short. The result is errors and more errors. Theoretically, a newly designed system should have all the pieces in working order, but in reality, each piece works independently. Now is the time to put all the pieces into one system and test it to determine whether it meets the user's requirements. This is the best chance to detect and correct errors before the system is implemented. The purpose of system testing is to consider all the likely variations to which it will be subjected and then push the system to its limits. If we implement the system without proper testing then it might cause the problems.

1. Communication between the user and the designer.
2. The programmer's ability to generate a code that reflects exactly the system specification.
3. The time frame for the design.

Theoretically, a new designed system should have all the pieces in working order, but in reality, each piece works independently. Now is the time to put all the pieces into one system and test it to determine whether it meets the requirements of the user. The process of system testing and the steps taken to validate and prepare a system for final implementation are:

## TESTING STRATEGIES

In order to make sure that the system does not have errors, the different levels of testing strategies that are applied at differing phases of software development are:

## UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .It is done after the completion of an individual unit before integration.

## INTEGRATION TESTING

In Integration Testing, the different units of the system are integrated together to form the complete system and this type of testing checks the system as whole to ensure that it is doing what is supposed to do. The testing of an integrated system can be carried out top-down, bottom-up, or big-bang. In this type of testing, some parts will be tested with white box testing and some with black box testing techniques. This type of testing plays very important role in increasing the systems productivity. We have checked our system by using the integration testing techniques.

### VERIFICATION TESTING

In Verification Testing, the different units of system are verified together whether the exact output to the system is existing or not.

### VALIDATION TESTNG

In Validation Testing,software validation is achieved through a series of black- box tests that demonstrate conformity with the requirements.After each validation test case has been conducted,one or two possible conditions exists:

1. The function or performance characteristics conform to specification and are accepted.
2. A deviation from specification is uncovered and a deficiency list is created.

## SYSTEM TESTING

A part from testing the system to validate the functionality of software against the requirements, it is also necessary to test the non-functional aspect of the system. Some examples of non-functional tools include tests to check performance, data security, usability/user friendliness, volume, load/stress that we have used in our project to test the various modules.

System testing consists of the following steps:

1. Program(s) testing.
2. String testing.
3. System testing.
4. System documentation.
5. User acceptance testing.

### ACCEPTANCE TESTING

After the developer has completed all rounds of testing and he is satisfied with the system, then the user takes over and re-tests the system from his point of view to judge whether it is acceptable according to some previously identified criteria. This is almost always a tricky situation in the project because of the inherent conflict between the developer and the user. In this project, it is the job of the bookstores to check the system that whether the made system fulfills the goals or not.

Why System Testing?

Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved. Inadequate testing results in two types of problems:

1. The time lag between the cause and the appearance of the problem.
2. The effect of system errors on the files and records within the system.

Another reason for system testing is its utility as a user-oriented vehicle before implementation.

### WHITE BOX TESTING

White box testing, sometimes called glass-box testing is a test case design method that uses the control structure of the procedural design to derive test cases. Using white box testing methods, the software engineer can derive test cases that

* Guarantee that allindependent paths within amodulehave been exercised at least once.
* Exercise all logical decisions on their true and false sides.
* Execute all loops at their boundaries and within their operational bounds and
* Exercise internal data structure to assure their validity. “First add record then save it” else it should be saved. **BLACK BOX TESTING**

This method treats the coded module as a black box. The module runs with inputs that are likely to cause errors. Then the output is checked to see if any error occurred. This method can be used to test all errors, because some errors may depend on the code or algorithm.

## TEST CASES

### 

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sno** | **Testcase** | **Input** | **Expected**  **Behaviour** | **Actual**  **Behaviour** | **Output** |
| 1 | Admin login | Invalid | Admin Login | Home Page | Failure |
| 2 | Admin Login | Credinals Valid | Page Admin | Admin Login | Success |
| 3 | User Login | Credinals  In Valid data | Login page Home Page | Page  User Home | Success |
| 4 | User | Invalid data | User | Page  Home Page | Failure |
| 5 | Registration  User Page | Valid data | Registration page  User | User | Success |
|  |  |  | Registration  Page | Registration Page |  |

Table:5.5 Test case Table

### TEST CASE 1

Input: **Admin Login**

Expected Behavior: **Admin Login Page** Actual Behavior:**Home Page** Result:**Failure**

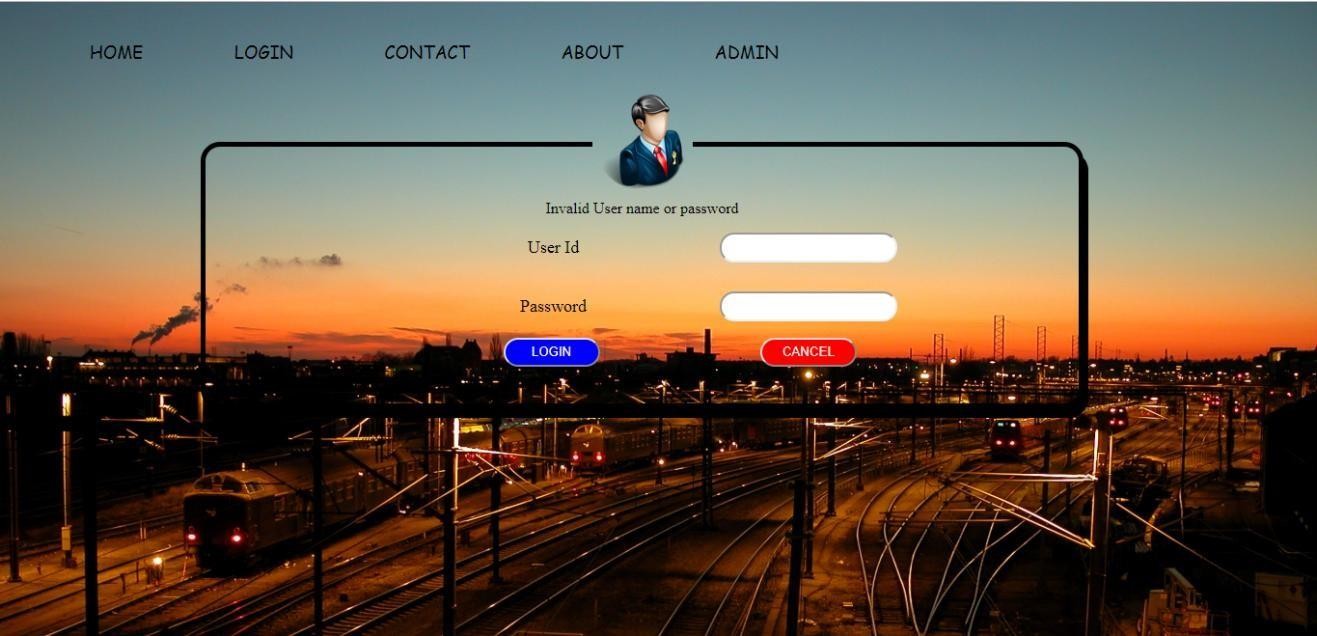


Fig:5.5.1 Test case 1

### TEST CASE 2

Input: **Admin Login**

Expected Behavior:**Admin Login page** Actual Behavior: **Admin Login Page** Result:**Success**



Fig:5.5.2 Test case 2

### TEST CASE 3

Input: **User Login**

Expected Behavior: **Home Page** Actual Behavior: **User Home page** Result: **Failure**

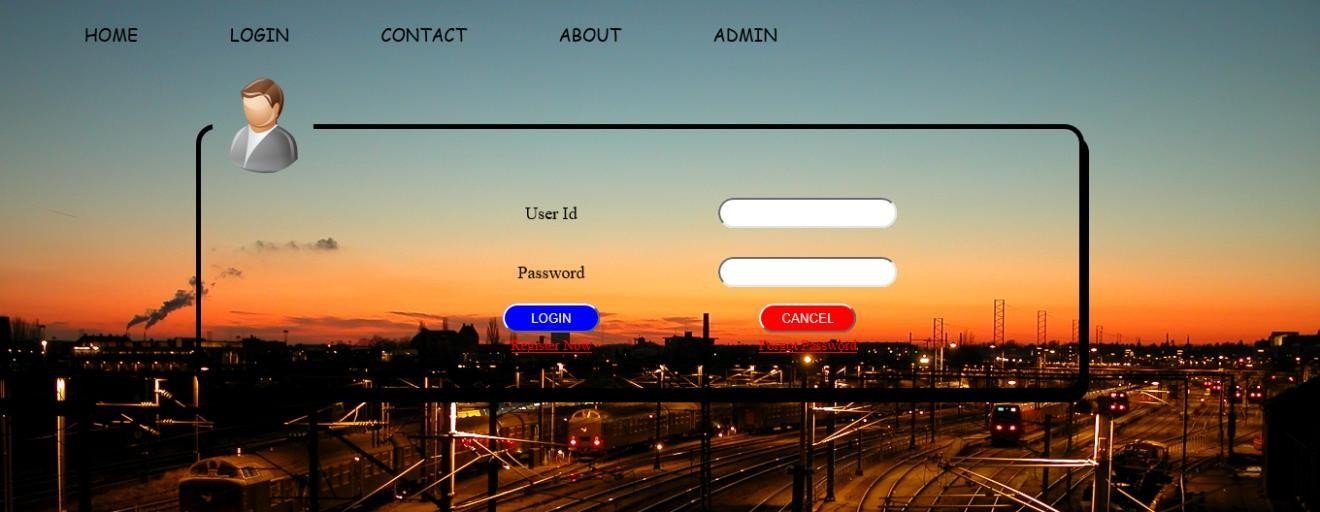


Fig:5.5.3Test case 3

### TEST CASE 4

Input: **User Registration**

Expected Behavior: **User Registration Page** Actual Behavior: **Home Page** Result:**Failure**



Fig:5.5.4 Test case 4

### TEST CASE 5

Input: **User Registration** Expected Behavior: **User Page** Actual Behavior: **User Page** Result:**Success**



**Fig:5.5.5 Test case 5**

# OUTPUT SCREENS

## HOME PAGE

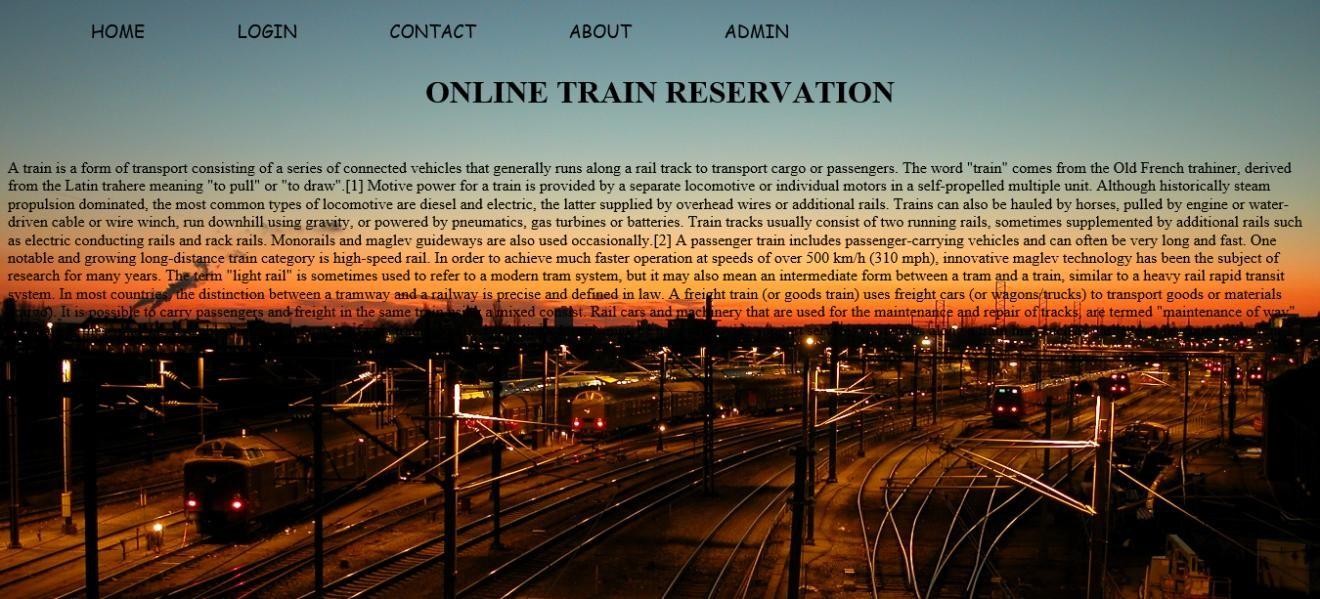


Fig:6.1 Output screen for home page

## USER REGISTRATION PAGE

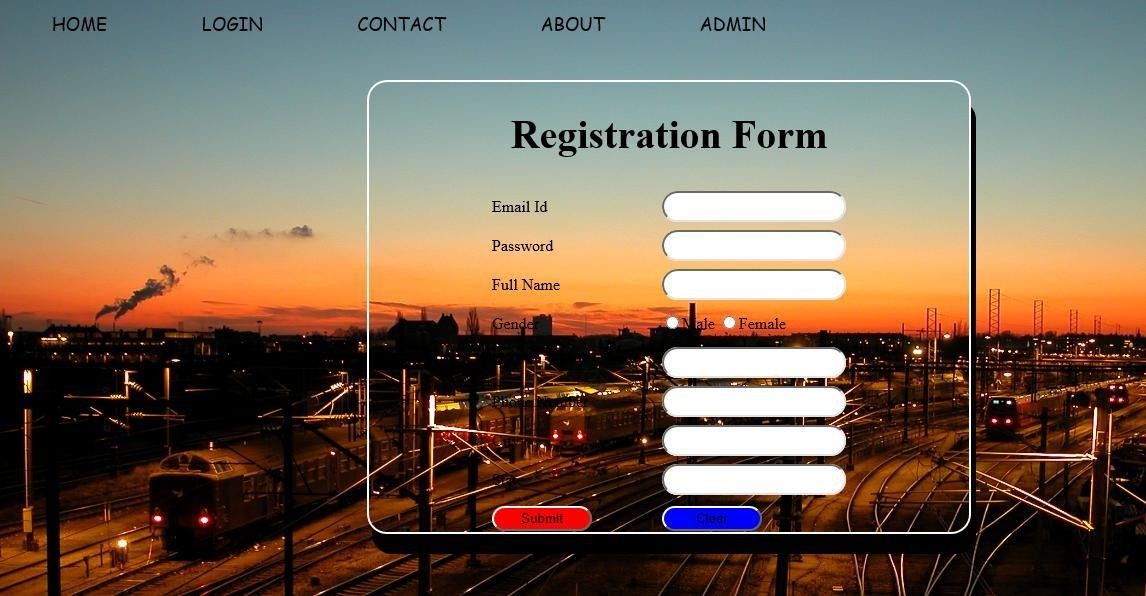


Fig:6.2 Output screen for user registration page

## USER LOGIN PAGE

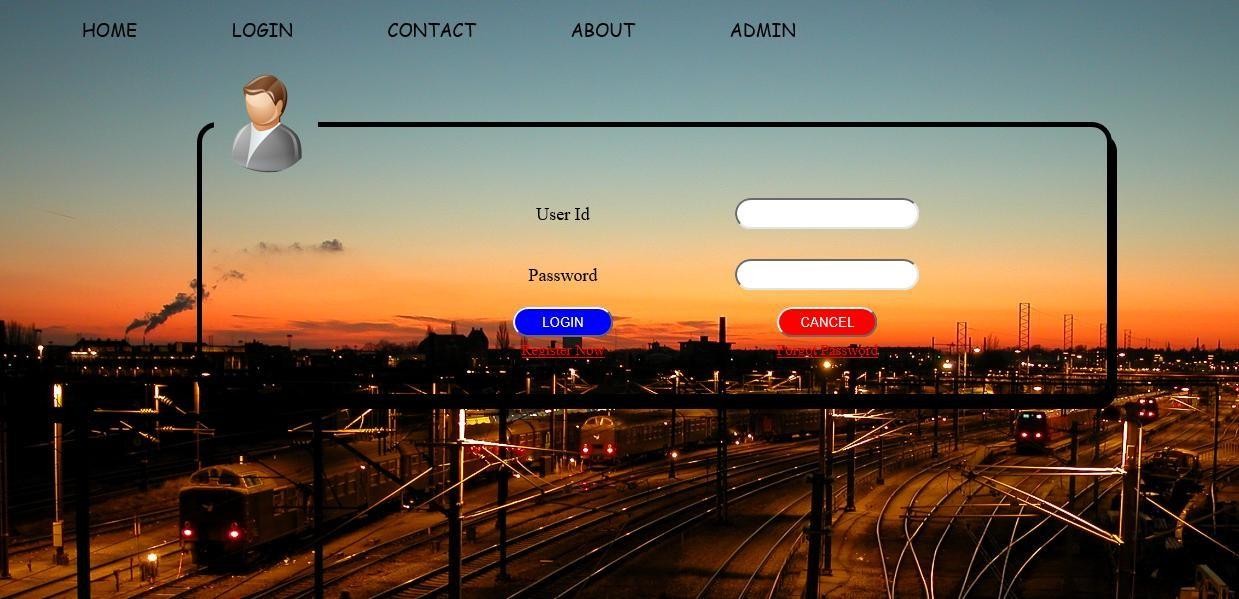


Fig:6.3Output screen for user login page

## ADMIN LOGIN PAGE

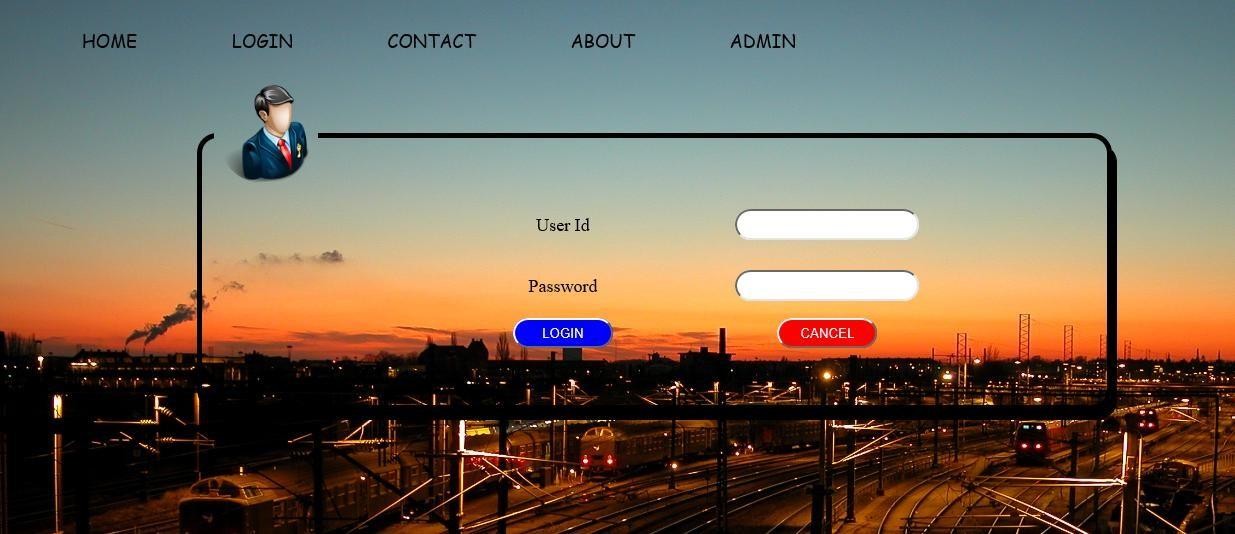


Fig:6.4Output screen for admin login pag

## ADD TRAINS

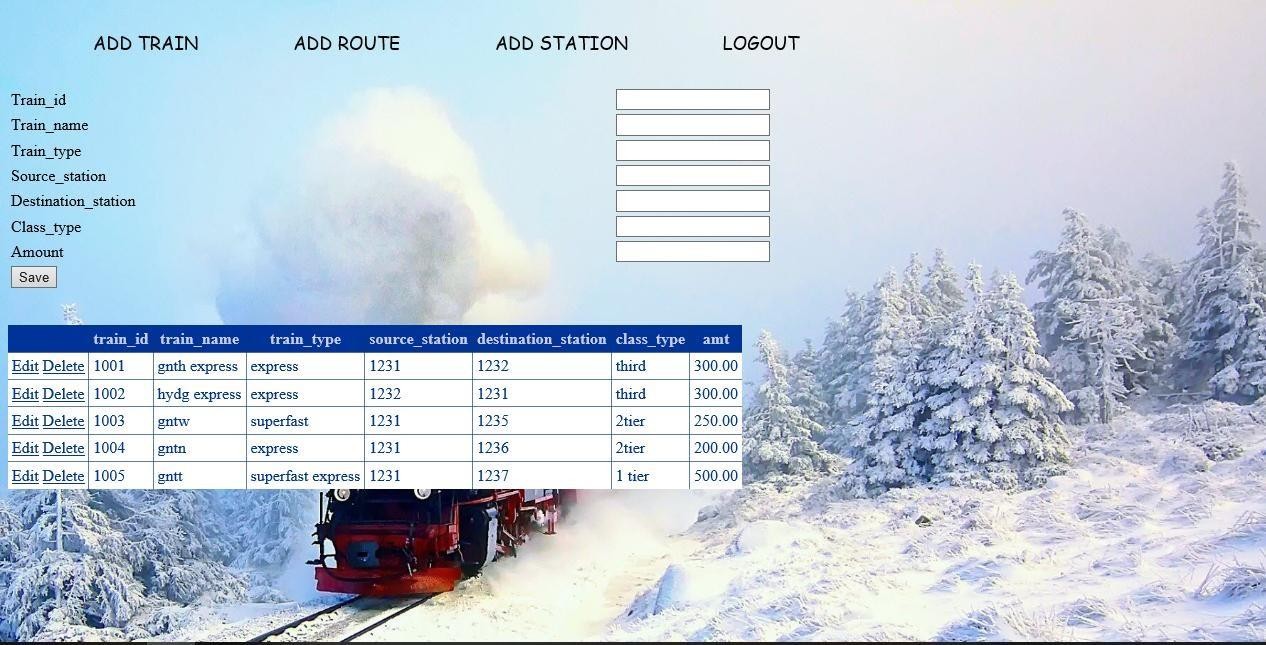


Fig:6.5 Output screen for add trains

## ADD STATION



Fig:6.6 Output screen for add station

## ADD ROUTE

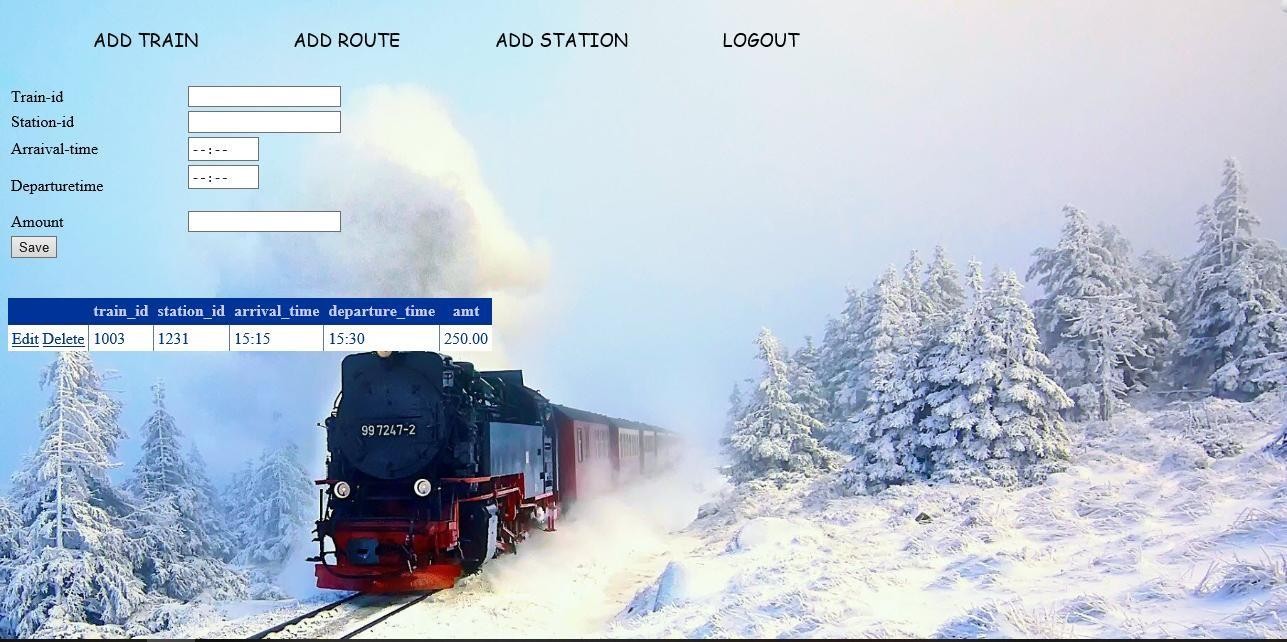


Fig:6.7Output screen for add route

## BOOKING TICKETS



Fig:6.8 Output screen for booking tickets

## TRAIN DETAILS



Fig:6.9 Output screen for train details

# CONCLUSION

Railway Reservation was successfully designed and is tested for accuracy and quality.By using this project data can be stored in digitalized format. Admin can easily maintain the records. Less chances for data corruption. By using this user can easily book the trains based on the availability of seats provided in the list. Security is provided.

# FUTURE ENHANCEMENT

Railway Reservation System was limited to India only but this can extend to World Wide . There is no facility to search for trains on this website will introduce the new facilities like cancelation. .

# BIBLIOGRAPHY

### Online Resource

<http://www.google.com/> <http://www.onlineshopping.com/> <http://www.w3schools.com/>

**REFERENCES**

### Asp.Net Books

Beginning ASP.NET 3.5 in C# 2008 By: George Shepherd

Sams Asp.net 3.5 Unleashed

By: Stephen Walther

Software Engineering by Roger S. Pressman

### Database Books

Teach Your Self SQL By: Ivan Baron