**PURBANCHAL UNIVERSITY**

**Biratnagar, Nepal**

****

A Project report on

**“RAILWAY RESERVATION SYSTEM”**

In the partial fulfillment for the requirement of the 1st Semester Project-I (subject code- BIT 106CO) in the completion of **Bachelor of Information Technology (BIT)** degree at **KIST college** **of Information Technology**, under **Purbanchal University.**

**Submitted By: - Submitted To: -**

1. **Hawana Tamang**

1. **Kushal Pathak**

**Purbanchal University**

1. **Sama Nemkul Shrestha**
2. **Srijana Limbu**

**Under The Guidance of**

**Mr. Prawesh Dhungana**

**Lecturer, BIT**

**KIST COLLEGE OF INFORMATION AND TECHNOLOGY KAMALPOKHARI, KATHMANDU NEPAL**

**KIST COLLEGE OF INFORMATION AND TECHNOLOGY KAMALPOKHARI, KATHMANDU NEPAL**

****

**CERTIFICATE**

This is to certify that the project work entitled **“RAILWAY RESERVATION SYSTEM”** is carried out by **HAWANA TAMANG (5413), KUSHAL PATHAK (5398), SAMA NEMKUL SHRESTHA (5399),** **SRIJANA LIMBU (5428),** bona fide students of **KIST COLLEGE OF INFORMATION AND TECHNOLOGY** in partial fulfillment for the award of **BACHELOR IN INFORMATION AND TECHNOLOGY** of the **PURBANCHAL UNIVERSITY, BIRATNAGAR NEPAL**, during the year **2020-2021**. It is certified that all corrections indicatedfor internal assessment have been incorporated in the report submitted in the department library. The project report has been approved, as it satisfied the academic requirements in respect of the project work prescribed for the said degree.

The details of the students are as follows: -

|  |  |  |
| --- | --- | --- |
| **NAME** | **REGISTRATION NO.** | **SYMBOL NO.** |
| Hawana Tamang | 058-3-2-04179-2020 | 313420 |
| Kushal Pathak | 058-3-2-04722-2020 | 313423 |
| Sama Nemkul Shrestha | 058-3-2-04736-2020 | 313437 |
| Srijana Limbu | 058-3-2-04745-2020 | 313446 |

Course Semester: - 1st Semester

Subject: - Project-I

Subject Code: - BIT (106CO)



Mr. Deepak Khadka

Program Coordinator, BIT



**KIST COLLEGE OF INFORMATION AND TECHNOLOGY KAMALPOKHARI, KATHMANDU**

Lab Sheet

Of

**“Data Structure and Algorithm”**

**Developed by**

**Romiya Dangol**

Is approved and is acceptable in qualify form.

**Mr. Prawesh Dhungana**

**Subject Teacher**

**ACKNOWLEDGEMENT**

It is with greatest satisfaction and euphoria that we are submitting our project report entitled **“RAILWAY RESERVATION SYSTEM”.** We have completed it as a part of the curriculum of **PURBANCHAL UNIVERSITY.**

We also take this opportunity to express a deep sense of gratefulness to our **BIT Coordinator Mr.** **Deepak Khadka** and **Lecturer Mr. Prawesh Dhungana** for their amiable support, valuableinformation and guidance which helped us in completing this task throughout its various stages. We are indebted to all members of **KIST College,** for the valuable support and suggestion provided by them using their specific fields’ knowledge. We are grateful for their cooperation during the period of our project.

Finally, we would also like to express our gratefulness towards **Purbanchal University** for designing such a wonderful course structure. It will help us to get more knowledge in the field of Information Technology & help us to have a bright future in the field of technology.

We hope our university will accept this attempt as a successful project.

Last but not the least, our sincere thanks to our parents, teaching and non-teaching staffs of our college and also my friends.

**HAWANA TAMANG (313420)**

**KUSHAL PATHAK (313423)**

**SAMA NEMKUL SHRESTHA (313437)**

**SRIJANA LIMBU (313446)**

**STUDENT’S DECLARATION**

We hereby declare that the project report entitled “**RAILWAY RESERVATION SYSTEM**” is a result of our own work. If we are found guilty of copying any other report or published information and showing as our original work, we understand that we shall be liable and punishable by **Purbanchal University**.

We further certify that this Project submitted in partial fulfillment of the requirement for the award of Bachelor in Information Technology (**BIT**) of the **Purbanchal University** is our original work and has not been submitted for award of any other degree or other similar title or prize.

|  |  |  |  |
| --- | --- | --- | --- |
| S.N. | Name | Registration No. | Symbol No. |
| 1 | Hawana Tamang | 058-3-2-04719-2020 | 313420 |
| 2 | Kushal Pathak | 058-3-2-04722-2020 | 313423 |
| 3 | Sama Nemkul Shrestha | 058-3-2-04736-2020 | 325146 |
| 4 | Srijana Limbu | 058-3-2-04745-2020 | 325155 |

**TO WHOM IT MAY CONCERN**

This is to certify that **Miss. Hawana Tamang, Mr. Kushal Pathak, Miss. Sama Nemkul Shrestha and Miss. Srijana Limbu of Bachelor in Information Technology (BIT)** has studied as per the curriculum of **BIT 1st Semester** and completed the project entitled “**RAILWAY RESERVATION** **SYSTEM”**.This project is the original work of **Miss. Hawana Tamang, Mr.** **Kushal Pathak, Miss. Sama Nemkul Shrestha and Miss. Srijana Limbu** and was carried out under the supervision of **Mr.** **Prawesh Dhungana** as per the guidelines provided by **Purbanchal University** and certified as per the student’s declaration that project “**Railway Reservation System**” has not been presented anywhere as a part of any other academic work.

The detail of the student is as follows:

Name of Students : Hawana Tamang

Kushal Pathak

Sama Nemkul Shrestha

Srijana Limbu

Semester : 1st

Subject Code : BIT 106C0

Project Title : **Railway Reservation System**

…………………………….

Mr. Deepak Khadka  
Program Coordinator, BIT  
KIST College of Information Technology

# ABSTRACT

The purpose of **“Railway Reservation System”** is to automate the existing manual system by the help of computerized equipment’s and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. The required software and hardware are easily available and easy to work with.

**Railway Reservation System,** as declared above, can lead to error free, secure, reliable and fastmanagement system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus, it will help organization in better utilization of resources. The organization can maintain computerized records without redundant entries. That means that one need not be distracted by information that is not relevant, while being able to reach the information.

The aim is to automate its existing manual system by the help of computerized equipment and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. Basically, the project describes how to manage for good performance and better services for the clients.

**According’s:** RRS refers to Railway Reservation System.

**TABLE OF CONTENTS**

[ABSTRACT](#_Toc96164445)

[CHAPTER 1 1](#_Toc96164446)

[INTRODUCTION 1](#_Toc96164447)

[1.1. INTRODUCTION 1](#_Toc96164448)

[1.2. PROBLEM STATEMENT 2](#_Toc96164449)

[1.3. OBJECTIVES 2](#_Toc96164450)

[1.4. SCOPE 3](#_Toc96164451)

[1.5. ADVANTAGES 4](#_Toc96164452)

[CHAPTER 2 5](#_Toc96164453)

[SYSTEM DESIGN 5](#_Toc96164454)

[2.1. ALGORITHM 5](#_Toc96164455)

[2.2. FLOWCHART 7](#_Toc96164456)

[CHAPTER 3 17](#_Toc96164465)

[REQUIREMENT ANALYSIS AND IMPLEMENTATION 17](#_Toc96164466)

[3.1. SYSTEM REQUIREMENTS 17](#_Toc96164467)

[3.2. SYSTEM METHODOLOGY 18](#_Toc96164468)

[3.3. REQUIREMENT ANALYSIS 19](#_Toc96164469)

[3.4. SYSTEM DESIGN 20](#_Toc96164472)

[3.5. FUNCTIONAL ANALYSIS 21](#_Toc96164473)

[3.6. IMPLEMENTATION 22](#_Toc96164474)

[3.7. INTEGRATION AND TESTING 22](#_Toc96164475)

[3.8. DEPLOYMENT AND MAINTENANCE 23](#_Toc96164476)

[CHAPTER 4 24](#_Toc96164477)

[CONCLUSION AND FUTURE SCOPE 24](#_Toc96164478)

[4.1. CONCLUSION 24](#_Toc96164479)

[4.2. FUTURE SCOPE 25](#_Toc96164480)

[REFERENCES 26](#_Toc96164481)

[5. APPENDICES 27](#_Toc96164482)

[5.1 SCREENSHOTS 27](#_Toc96164483)

[5.2. SOURCE CODE 37](#_Toc96164484)

# CHAPTER 1

## INTRODUCTION

### 1.1. INTRODUCTION

The RRS has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and, in some case, reduce the hardships faced by this existing system. Moreover, this system is designed for the particular need of the company to carry out operations in a smooth and effective manner.

Railway Reservation System is a desktop-based application ticketing. This application is developed by using C programming language. It is an easy and time-efficient way of storing the data. These data can be easily accessed by the admin.

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 railway reservation system facilitates the passenger to enquire about the trains available on the basis of source and destination, booking and cancellation of tickets, enquiry about the status of the booked ticketed, etc. The aim of the case study is to design and develop a file maintaining records of different trains, trains status and passengers. It is computerized system of reserving the seats of train seats in advance. It is mainly used for the long route. In the given project we will be developing a website which will help users to find train details, book and cancel tickets and the exact rates of their tickets to the desired destination.

### 1.2. PROBLEM STATEMENT

The major challenges that passengers had faced while manual booking system was it was time consuming and less efficient. The number of ticket counters were less than the passengers booking ticket so, there was a long queue while booking a ticket. In order to cancel a ticket an individual should travel to railway station for the cancellation process. Railway passengers frequently need to know about their ticket reservation status, train arrival or departure details etc. Customer information centers at the railway station are unable to solve such queries at peak periods.

### 1.3. OBJECTIVES

The objectives of the system are-

* To manage the details of Train, Booking, Payment, Seat, Ticket.
* It manages all the information about Train Customer, Ticket etc.
* To calculate the fare.
* To reserve and cancel the ticket if necessary.
* It contains information about the trains.
* Railway time table.
* Train between stations

### 1.4. SCOPE

This software package can be readily used by non-programming personal avoiding human handled chance of error. This project is only used by administrative users.

The aim of this proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing systems. The system provides proper security and reduces the manual work.

This project has a large scope as it has the following features which help in making it easy to use, understand and modify it:

* Management of Passenger Data
* Security of data
* Ensure data accuracy
* Minimize manual data entry
* Minimum time needed for various processing
* Automation of the Placement Procedure
* Greater Efficiency
* Better service
* Minimum time required
* No need to do paper work
* To save the environment by using paper free work.

### 1.5. ADVANTAGES

* Ticket will be available 24/7.
* Manage the information of Train.
* To increase efficiency of managing the Train, Booking etc.
* It generates the report in Train, Booking, and Customer etc.
* Cut the overloads of the staff’s member.
* The transactions are executed in off-line mode, hence on-line data for Train, booking capture and modification is not possible.
* It tracks all the information of Booking, Passenger, seat etc.
* Shows the information and description of the Train, Receipt, and Passenger details.
* Editing, adding and updating of Records is improved which results in proper resource management of Train data. [1]

# CHAPTER 2

## SYSTEM DESIGN

### 2.1. ALGORITHM

Step 1: start

Step 2: press any key to continue

Step 3: enter password

Step 4: Compare password with password saved in a file.

Step 5: if matches go to step 3

Else go to step 8

Step 6: if step 3 is repeated for four times, then exit the program, go to step 17

Step 7: Show menu

Step 8: Enter choice from menu

Step 9: if choice is a digit from 1 to 7 go to step

Else

Step 9.1: Display "Invalid choice"

Step 9.2: go to step 7

Step 10: if choice is 1

Step 10.1: enter the number of passengers

Step 10.2: enter the name of passenger

Step 10.3: enter the gender of passenger

Step 10.4: enter the age of passenger

Step 10.5: enter the phone number of passenger

Step 10.6: enter the date of reservation

Step 10.7: please select a unique id

Step 10.8: enter number of seats

Step 10.9: Press Enter To View Available Trains

Step 10.10: Enter train number

Step 10.10.1: select the seat class

If choice is 1 go to step 10.10.1.1

Else go to step 10.10.1.2

Step 10.10.1.1: enter seat number

Step 10.10.1.2: enter seat number

Step 10.10.2: confirm the ticket booking

If input is y go to step 10.10.2.1

Else if input is n go to step 10.10.2.2

Else go to step 10.10.2.3

Step 10.10.2.1: print reservation is done

Step 10.10.2.2: print reservation is cancelled

Step 10.10.2.3: print invalid choice go to step 10.10.2

Step 10.11: enter any key to go to main menu

Step 11: if choice 2

Step 11.1: enter the number of passengers

Step 11.2: confirm the ticket booking

If input is y go to step 11.2.1

Else if input is n go to step 11.2.2

Else go to step 11.2.3

Step 11.2.1: print reservation is done

Step 11.2.2: print reservation is cancelled

Step 11.2.3: print invalid choice go to step 10.10.2

Step 11.3: enter any key to go to main menu

Step 12: if choice 3

Step 12.1: enter the specific train number

If choice is from 1001 to 1010 go to step 12.1.1

Else go to step 12.1.2

Step 12.1.1: enter any key to go to main menu

Step 12.1.2: print invalid input enter again, go to step 12.1

Step 13: if choice is 4

Step 13.1: enter any key to go to main menu

Step 14: if choice is 5

Step 14.1: enter the receipt id to cancel the ticket

If receipt id matches then go to step 14.2

Else go to step 14.3

Step 14.2: print your reservation is successfully cancelled

Step 14.3: please enter a valid receipt id, go to step 14.1

Step 14.4: enter any key to go to main menu

Step 15: if choice is 6

Step 15.1: Enter the receipt ID of the passenger to search for the detail

Step 15.2: Enter the number of passengers

Step 15.3: print the details of the passenger you searched

Step 15.4: enter any key to go to main menu

Step 16: if choice is 7, go to step 17

Step 17: end.

### 2.2. FLOWCHART

F

T

Press any key to continue

Enter username and password

If username and password is correct?

If username and password is incorrect for 4 time?

**START**

SORRY!!!! LOGIN UNSUCESSFULL

**STOP**

F

A

Show WELCOME TO OUR SYSTEM!!! YOUR LOGIN IS SUCCESSFUL

Press any key to continue

SHOW MENU

Figure 1: Flowchart 1

# 

A

If choice is 1?

1

F

T

F

If choice is 2?

2

T

If choice is 3?

3

F

T

T

If choice is 5?

If choice is 4?

4

F

T

F

T

5

If choice is 7?

INVALID CHOICE

A

F

T

F

6

If choice is 6?

Figure 2: Flowchart 2

1

Enter Details:

Please Enter to view available trains:

4

Enter train number::

B

Figure 3: Flowchart 3

**T**

**F**

**T**

**F**

Enter seat number:

Confirm reservation(y/n):

If the choice is y?

Reservation Done

If the choice is n?

B

Reservation Not Done!

Invalid choice entered! Enter again:

**T**

B

Display

1. Sleeper class

2. A.C class

If the choice is 1?

Display seat matrix

If the choice is 2?

C

**F**

**F**

Invalid choice entered! Enter again:

Figure 4: Flowchart 4

**F**

If choice is y?

Reservation done

A

If choice is n?

Reservation not done!

A

C

1)3A class

2)2A class  
3)1st class AC

Enter your choice of subclass:

If choice is 1 or 2 or 3?

Display total bill amount

Display seat matrix

Enter seat numbers

Confirm reservation(y/n):

Invalid choice entered! Enter again:

**F**

**T**

**T**

Figure 5: Flowchart 5

**T**

**T**

2

Enter number of passengers

Display receipt

Confirm ticket(y/n):

If choice is y?

Reservation done

A

If choice is n?

Reservation not done!

A

Invalid choice entered! Enter again:

**F**

**F**

Figure 6: Flowchart 6

# 

3

Enter specific train number (1001-1010):

If choice is 1001?

Display details of 1001

If choice is 1002?

Display details of 1002

If choice is 1003?

Display details of 1003

If choice is 1004?

Display details of 1004

If choice is 1005?

Display details of 1005

If choice is 1006?

Display details of 1006

If choice is incorrect?

# 

If choice is 1007?

Display details of 1007

If choice is 1008?

Display details of 1008

If choice is 1009?

Display details of 1009

If choice is 1010?

Display details of 1010

A

# 

Figure 7: Flowchart 7

# 

# 

4

Display all available trains

A

Figure 8: Flowchart 8

T

5

Display Cancel a Ticket

Enter the receipt id to cancel the ticket:

If receipt id is correct?

The amount you get after the cancellation charge is:

Your reservation is cancelled successfully.

Please enter a valid receipt id

F

Figure 9: Flowchart 9

# 

6

Enter the receipt ID of the passenger to search for the details:

Enter the number of passengers:

Display search details

A

# 

Figure 10: Flowchart 10

# CHAPTER 3

## REQUIREMENT ANALYSIS AND IMPLEMENTATION

### 3.1. SYSTEM REQUIREMENTS

Following hardware and software requirement should be met for flawless running of this system:

**MINIMUM REQUIREMENTS:**

**PROCESSOR**: Intel core i3 or i5

**SPEED**: 1.5Hz

**RAM**: 4 GB

**HARDDISK**: 20MB (At least 80MB of free space)

**MONITOR**: LCD MONITOR

**OPERATING SYSTEM**: WINDOWS XP, 2000 Professional

**COMPILER**: DEV C++

### 3.2. SYSTEM METHODOLOGY

**WATERFALL MODEL**

The waterfall model is a classical model used in system development life cycle to create a system with linear and sequential approach. It is termed as waterfall because the model develops systematically from one phase to another in a downward fashion. In waterfall model the requirements are very well documented, clear and fix. The project done under waterfall model is short and the product definition is stable. [2]

**The sequential phases described in the Waterfall model are:**

Deployment and Maintenance

Integration and Testing

Implementation

System Design

Requirement Analysis

Figure 11:Waterfall model

### 3.3. REQUIREMENT ANALYSIS

**FUNCTIONAL REQUIREMENT**

In software and system engineering, a functional requirement defines a function of a system or its component, where a function is described as a specification of behavior between input and outputs. [3]

### 

RAILWAY RESERVATION SYSTEM

ADMIN

### 

Figure 12:Use Case Diagram

### 3.4. SYSTEM DESIGN

System design is the process of designing the architecture, components, and interfaces for a system so that it meets the end-user requirements. A good system design is to organize the program modules in such a way that are easy to develop and change. There are many strategies or techniques for performing system design.

* **Importance**:
* If any pre-existing code need to be understood, organized, and pieced together.
* It is common for the project team to have to write some code and produce original programs that support the application logic of the system.

There are many strategies or techniques for performing system design.

* **Top-down approach:** Top-down integration testing is an integration testing technique used in order to simulate the behavior of the lower-level modules that are not yet integrated. Each system is divided into several subsystems and components. Each of the subsystems is further divided into a set of subsystems and components.
* **Advantages of top-down approach:**
* The main advantage of the top-down approach is that its strong focus on requirements helps to make a design responsive according to its requirements.

|  |  |  |
| --- | --- | --- |
| **No.** | **Function module** | **Function Description** |
| **1.** | **login()** | This function is for security purpose so that person other than admin cannot manipulate the system or program. |
| **2.** | **details()** | This function is for adding the passenger details. |
| **3.** | **seat()** | This function shows the details of seat in seat matrix for reservation. |
| **4.** | **cal()** | This function is used for booking seat class according to the choice of passenger. |
| **5.** | **bill()** | This function is for the printing of receipt of the passenger. |
| **6.** | **viewdetails()** | This function is to show the details of available train with source and destination. |
| **7.** | **specifictrain()** | This function is to show the details of specific train. |
| **8.** | **cancel()** | This function allows admin to cancel the reservation. |
| **9.** | **search()** | This function helps admin to search the details of passenger. |

### 3.5. FUNCTIONAL ANALYSIS

### 3.6. IMPLEMENTATION

This phase is initiated after the system has been testes and accepted by the user. System performance is compared to performance objectives established during the planning phase. System implementation is a process of ensuring that the information system is operational. Implementation allows the users to take over its operation for use and evaluation.

Implementation includes user notification, user training, installation of hardware, installation of software onto production computers, and integration of system into daily work processes. This phase continues until the system is operating in production in accordance with the defined user requirements.

* We used C programming to implement our project.
* File Handling was used for the data and records.
* Functions for sub modules.
* The system is first developed in small programs called units, which are integrated in the next phase. The testing of each developed unit individually is referred as unit testing.

### 3.7. INTEGRATION AND TESTING

The systems integration test function is to ensure that the developed systems meet all the technical requirements with the components and subsystems integrated. All the modules/functions are tested. Individual functions are provided and output is generated. The code is tested through the unit testing.

* **Unit Testing:** A testing technique using which individual modules are tested to determine if there are any issues to be fixed. It is concerned with functional correctness of the standalone modules. The main aim is to isolate each unit of the system to identify, analyze and fix the defects.
* **Advantages of unit testing**
  + Reduces defects in the newly developed features or reduces bugs when changing the existing functionality.
  + Reduces Cost of testing defects are captured in very early phase
  + Improves design and allows better refactoring of code.
  + Unit tests, when integrated with build gives the quality of the build as well.

### 3.8. DEPLOYMENT AND MAINTENANCE

* The deployment phase is the final phase of the software development life cycle (SDLC) and puts the product into production.
* After the project team tests the product and the product passes each testing phase, the product is ready to go live. This means the product is ready to be used in a real environment by all end users of the product.
* Once the functional and non-functional testing is done, the product us deployed in the customer environment or released into the market.
* After the product is deployed to the user’s market from there the maintenance phase starts
* Once the product or the system is in use there will be many patches to be fixed.
* The user might ask for new features and enhancements. It is the responsibility of the maintenance team to attend to these requests and to fix the bugs that are found.
* The maintenance effort revisits all the other stages of the software life cycle.
* Each modification requires planning, specification, design, coding, testing, installation.

# CHAPTER 4

## CONCLUSION AND FUTURE SCOPE

### 4.1. CONCLUSION

Our project is only a humble venture to satisfy the needs to manage the project work. Several user-friendly coding has also been adopted. This package shall prove to be a powerful package in satisfying all the requirements of the school. The objective of software planning is to provide a frame work that enables the manager to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses.

**At the end it is concluded that we have made effort on following points**

* A description of the background and context of the project and its relation to work already done in the area.
* The description of Purpose, Scope, and applicability.
* We define the problem on which we are working in the project.
* We describe the requirement Specifications of the system and the actions that can be done on these things.
* We included features and operations in detail, including screen layouts.
* Finally, the system is implemented and tested according to test cases.

### 4.2. FUTURE SCOPE

In a nutshell, it can be summarized that the future scope of the project circles around maintaining information regarding:

* We can give more advance software for **“Railway Reservation System”** including more facilities.
* We can add the monthly pass system, freight revenue enhancement, etc.
* Online booking can be integrated.
* Reservation details, train schedule, etc. can be updated and reports can be generated.
* Integrate multiple load balancers to distribute the loads of the system.
* User interface can be made more effective.

The above-mentioned points are the enhancements which can be done to increase the applicability and usage of this project. Here we can maintain the records of Contact and credential. Enhancements can be done to maintain all the Contact, Credential, Telephone, Profile, Emails.

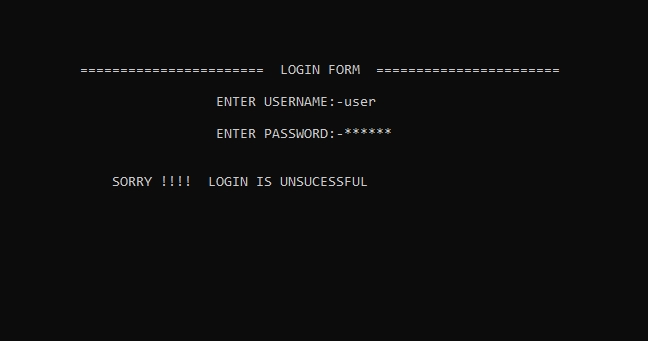
# REFERENCES

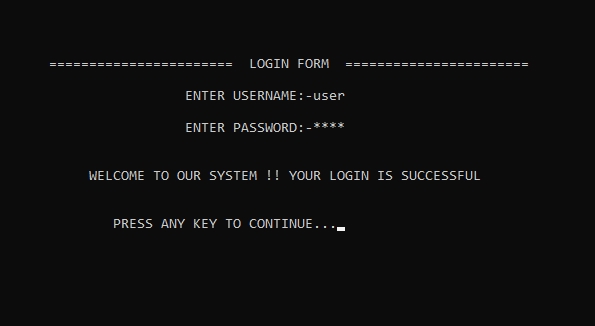
|  |  |
| --- | --- |
| [1] | F. Projectz, "Free Projectz," [Online]. Available: https://www.freeprojectz.com/premium-synopsis/synopsis-railway-reservation-system. |
| [2] | R. S.PRESSMAN, SOFTWARE ENGINEERING , A PRACTITIONER'S APPROACH, vol. Fourth Edition, E. M. munson, Ed., McGraw-Hill Companies,inc, 1997. |
| [3] | J. A.Hoffer, Modern System Analysis and Design, Benjamin-Cummings Pub Co(May 1 1996). |
| [4] | p. t. a. m. Source code and project, "ode-projects," Source code and project, [Online]. Available: https://code-projects.org/?fbclid=IwAR1JGfaQnPUnNBqi4UPANOLqwepE2BnC5AYtmR6jctZsaNafmt39Z8yFD5g. |
| [5] | Vishwassathish, "github," [Online]. Available: https://github.com/vishwassathish/Basic-Railway-Reservation-System/blob/master/Railways.c. |

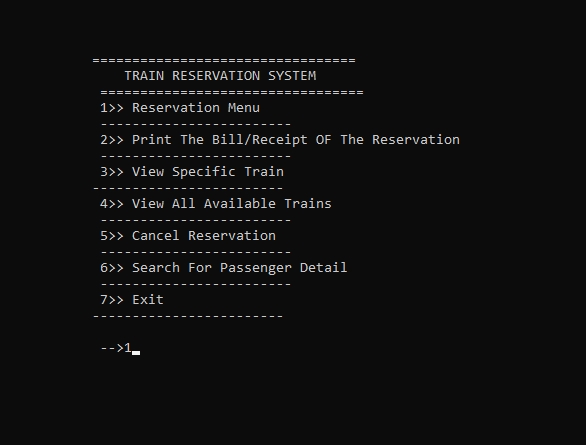
# 5. APPENDICES

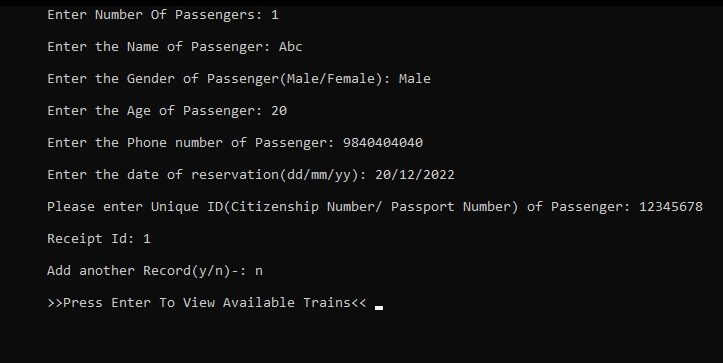
## 5.1 SCREENSHOTS

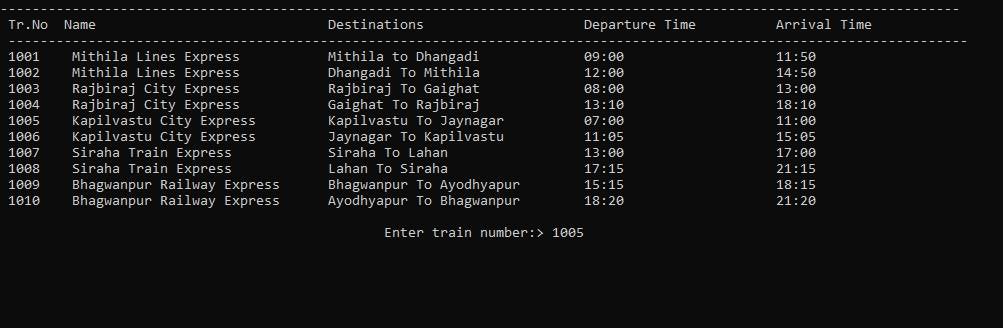


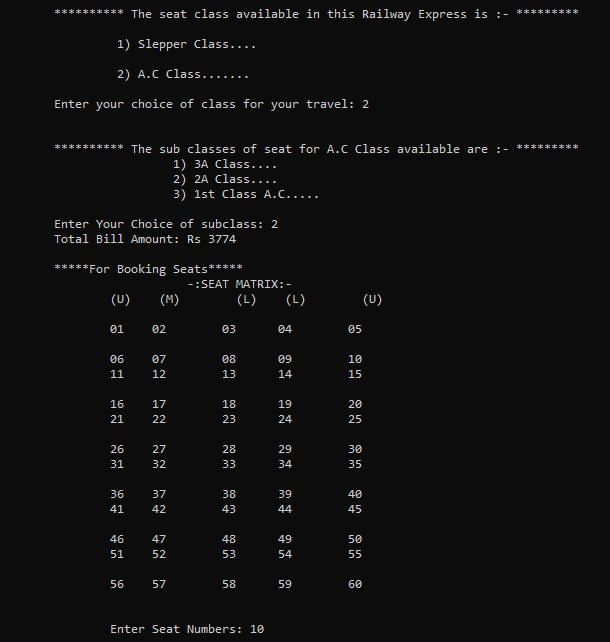


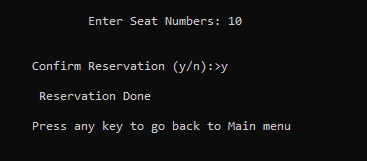


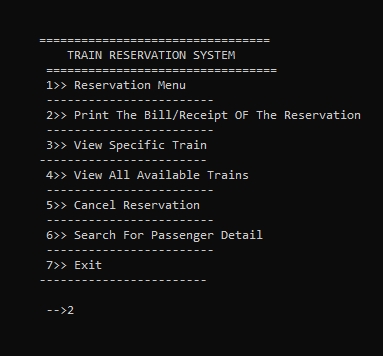


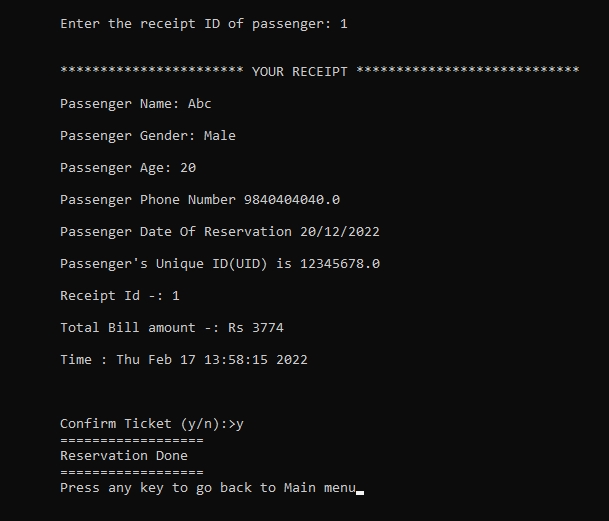


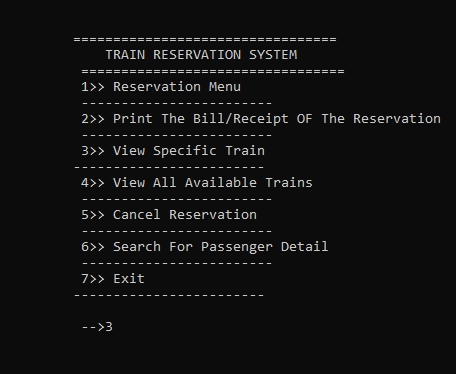


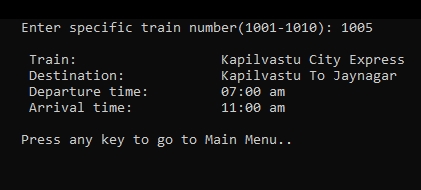


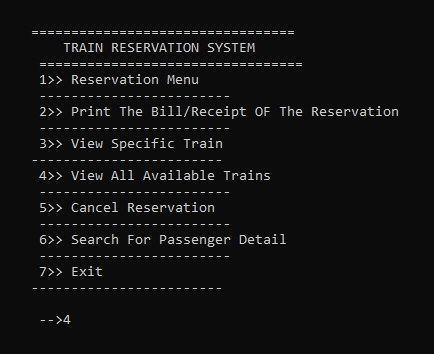


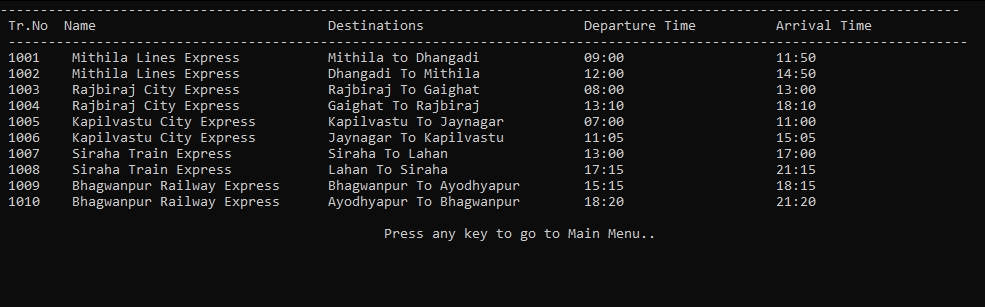


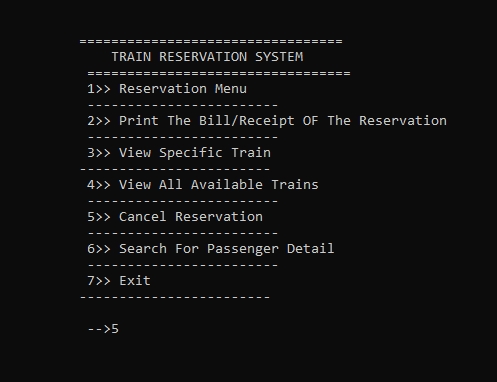


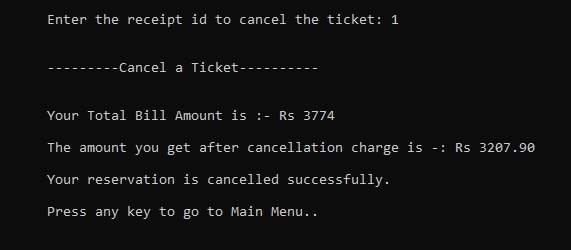


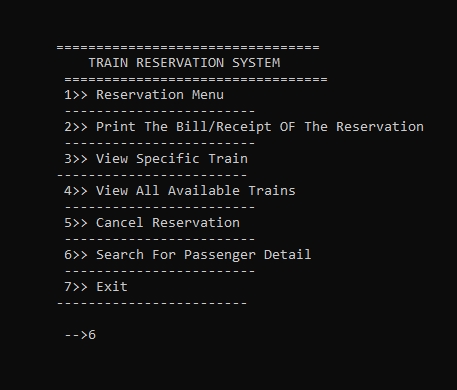


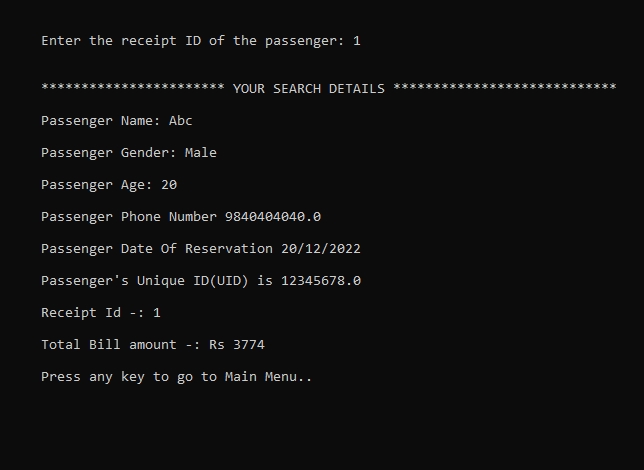


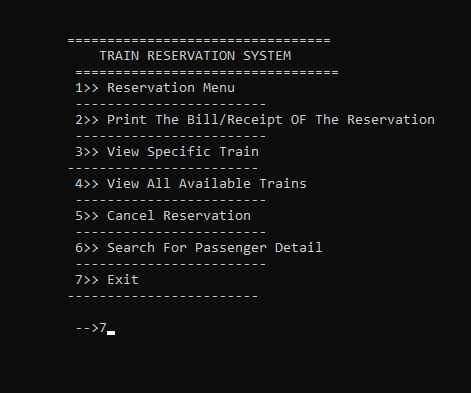


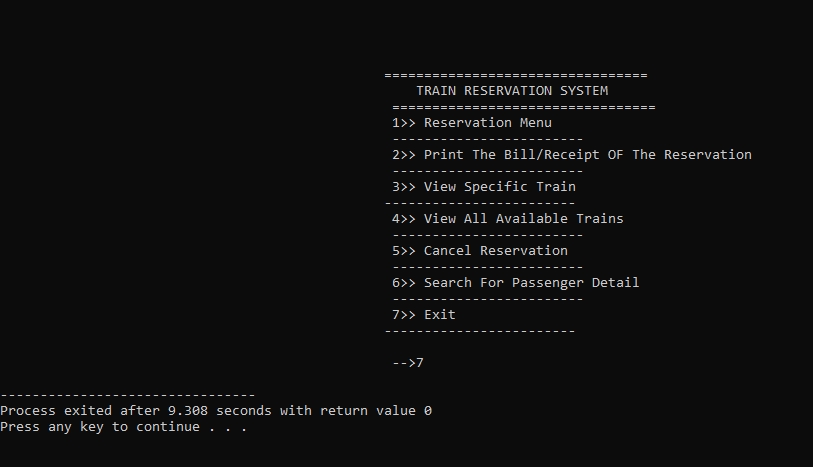












## 5.2. SOURCE CODE

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

#include<string.h>

#include<time.h>

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*GLOBAL VARIABLES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//ALl the globle variables and the composite data types will be declared here

// Global variables

char source[20], des[20], train[40];

char station[40], cla[40];

int time1, time2, a[55];

int j;

struct pd

{

char val[20],name[20],gen[6],gender[6];

int ag,age,d,m,year;

double Ph\_no,Phno,UID;

int train\_num;

int num\_of\_seats,RID;

}passdetails,pd1;

struct billpay

{

int bill\_amt;

int RID1;

}bp;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*FUNCTION PROTOTYPE\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//function prototypes to be used

void details(void); //main reservation function //main reservation function

int seat(int);

int cal(int, int, int);

void bill();

void viewdetails(void); //view details of all the trains

void specifictrain(void); //print data related to specific train

void login();

void cancel(void);

void search(void);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*FUNCTION DECLARATION\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*MAIN()\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main()

{

system("cls");

printf("\t\t=================================================\n");

printf("\t\t| |\n");

printf("\t\t| ----------------------------- |\n");

printf("\t\t| TRAIN TICKET RESV. SYSTEM |\n");

printf("\t\t| ----------------------------- |\n");

printf("\t\t| |\n");

printf("\t\t| |\n");

printf("\t\t| |\n");

printf("\t\t| BROUGHT TO YOU BY |\n");

printf("\t\t| | SOUL CYNICS | |\n");

printf("\t\t| |\n");

printf("\t\t=================================================\n\n\n");

printf(" \n Press any key to continue:");

getch();

system("cls");

login();

int menu\_choice,choice\_return;

start:

system("cls");

printf("\n=================================\n");

printf(" TRAIN RESERVATION SYSTEM");

printf("\n=================================");

printf("\n1>> Reservation Menu");

printf("\n------------------------");

printf("\n2>> Print The Bill/Receipt OF The Reservation");

printf("\n------------------------");

printf("\n3>> View Specific Train");

printf("\n------------------------");

printf("\n4>> View All Available Trains");

printf("\n------------------------");

printf("\n5>> Cancel Reservation");

printf("\n------------------------");

printf("\n6>> Search For Passenger Detail");

printf("\n------------------------");

printf("\n7>> Exit");

printf("\n------------------------");

printf("\n\n-->");

scanf("%d",&menu\_choice);

switch(menu\_choice)

{

case 1:

details();

break;

case 2:

bill();

break;

case 3:

specifictrain();

printf("\n\nPress any key to go to Main Menu..");

getch();

break;

case 4:

viewdetails();

printf("\n\nPress any key to go to Main Menu..");

getch();

break;

case 5:

cancel();

printf("\n\nPress any key to go to Main Menu..");

getch();

break;

case 6:

search();

printf("\n\nPress any key to go to Main Menu..");

getch();

break;

case 7:

return(0);

default:

printf("\nInvalid choice");

}

goto start;

return(0);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*RESERVATION()\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Adding Details Of The Passengers()\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void details()

{

FILE \*fp,\*fq;

fp=fopen("Railway\_Reservation.txt","a+");

if(fp==NULL)

{

puts("File cannot be opened ");

exit(0);

}

fq=fopen("Railway\_Reservation1.txt","a+");

if (fq==NULL)

{

puts("File cannot be opened ");

exit(0);

}

int i,ag,m,year,d;

double Phno,UID;

int train\_num;

int a1, a2, b, c,RID;

int x = 0, e, r;

char o, gen[6], confirm,another;

char val[20], gender[6];

system("cls");

printf("Enter Number Of Passengers: ");

fflush(stdin);

scanf("%d", &j);

another='y';

while(another=='y'|| another=='Y')

{

printf("\nEnter the Name of Passenger: ");

fflush(stdin);

gets(passdetails.val);

printf("\nEnter the Gender of Passenger(Male/Female): ");

fflush(stdin);

gets(passdetails.gender);

printf("\nEnter the Age of Passenger: ");

fflush(stdin);

scanf("%d",&passdetails.ag);

printf("\nEnter the Phone number of Passenger: ");

fflush(stdin);

scanf("%lf",&passdetails.Phno);

printf("\nEnter the date of reservation(dd/mm/yy): ");

fflush(stdin);

scanf("%d/%d/%d",&passdetails.d,&passdetails.m,&passdetails.year);

printf("\nPlease enter Unique ID(Citizenship Number/ Passport Number) of Passenger: ");

fflush(stdin);

scanf("%lf", &passdetails.UID);

printf("\nReceipt Id: ");

fflush(stdin);

scanf("%d",&passdetails.RID);

fwrite(&passdetails,sizeof(passdetails),1,fp);

printf("\nAdd another Record(y/n)-: ");

another=getche();

}

bp.RID1 = passdetails.RID;

printf("\n\n>>Press Enter To View Available Trains<< ");

getch();

system("cls");

viewdetails();

label:

printf("\n\nEnter train number:> ");

scanf("%d",&train\_num);

do {

switch(train\_num)

{

case 1001:

{

a1 = 2099;

a2 = 1560;

// Calling cal() function

// with the three argument

// and return value

d = cal(a1, a2, j);

printf("Total Bill Amount: Rs %d \n",d);

}break;

case 1002:

{

a1 = 2099;

a2 = 1560;

// Calling cal() function with

// three argument & return value

d = cal(a1, a2, j);

printf("Total Bill Amount: Rs %d \n",d);

} break;

case 1003:

{

a1 = 1801;

a2 = 981;

// Calling cal() function with

// three argument & return value

d = cal(a1, a2, j);

printf("Total Bill Amount: Rs %d\n", d);

} break;

case 1004:

{

a1 = 1801;

a2 = 981;

// Calling cal() function with

// three argument & return value

d = cal(a1, a2, j);

printf("Total Bill Amount: Rs %d\n", d);

} break;

case 1005:

{

a1 = 2199;

a2 = 1780;

// Calling cal() function with

// three argument & return value

d = cal(a1, a2, j);

printf("Total Bill Amount: Rs %d\n", d);

} break;

case 1006:

{

a1 = 2199;

a2 = 1780;

// Calling cal() function with

// three argument & return value

d = cal(a1, a2, j);

printf("Total Bill Amount: Rs %d\n", d);

} break;

case 1007:

{

a1 = 1759;

a2 = 1200;

// Calling cal() function with

// three argument & return value

d = cal(a1, a2, j);

printf("Total Bill Amount: Rs %d\n", d);

} break;

case 1008:

{

a1 = 1759;

a2 = 1200;

// Calling cal() function with

// three argument & return value

d = cal(a1, a2, j);

printf("Total Bill Amount: Rs %d\n", d);

} break;

case 1009:

{

a1 = 2205;

a2 = 1905;

// Calling cal() function with

// three argument & return value

d = cal(a1, a2, j);

printf("Total Bill Amount: Rs %d\n", d);

} break;

case 1010:

{

a1 = 2205;

a2 = 1905;

// Calling cal() function with

// three argument & return value

d = cal(a1, a2, j);

printf("Total Bill Amount: Rs %d\n", d);

} break;

default:

printf("Please Enter The Correct Choice.....\n");

goto label;

}

} while (x);

bp.bill\_amt=d;

//fseek(fq,0, SEEK\_END);

fwrite(&bp,sizeof(bp),1,fq);

printf("\n \*\*\*\*\*For Booking Seats\*\*\*\*\*\n");

// Calling seat() function with number of passenger

seat(j);

printf("\n\nConfirm Reservation (y/n):>");

start:

scanf(" %c",&confirm);

if(confirm == 'y')

{

printf("==================");

printf("\n Reservation Done\n");

printf("==================");

printf("\nPress any key to go back to Main menu");

}

else

{

if(confirm=='n'){

printf("\nReservation Not Done!\nPress any key to go back to Main menu!");

}

else

{

printf("\nInvalid choice entered! Enter again-----> ");

goto start;

}

}

fclose(fp);

fclose(fq);

getch();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*PRINTTICKET()\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// Code for printing receipt

void bill()

{

char confirm;

long int recsize,recsize1;

time\_t currentTime;

time(&currentTime);

int i,c,reID,x;

FILE \*fp,\*fq;

fp=fopen("Railway\_Reservation.txt","r+");

if(fp==NULL)

{

puts("File cannot be opened ");

exit(0);

}

fq=fopen("Railway\_Reservation1.txt","r+");

if (fq==NULL)

{

puts("File cannot be opened ");

exit(0);

}

system("cls");

recsize=sizeof(passdetails);

recsize1=sizeof(bp);

//fseek(fp,0, SEEK\_END);

//fseek(fq,0, SEEK\_END);

top5:

printf("\n\nEnter the receipt ID of passenger: ");

fflush(stdin);

scanf("%d",&reID);

rewind(fp);

rewind(fq);

printf("\n\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* YOUR RECEIPT \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

while(fread(&passdetails,recsize,1,fp)>0)

{

if(reID == passdetails.RID)

{

x=1;

printf("\t\t\n\nPassenger Name: ");

puts(passdetails.val);

printf("\t\t\nPassenger Gender: ");

puts(passdetails.gender);

printf("\t\t\nPassenger Age: %d",passdetails.ag);

printf("\t\t\n\nPassenger Phone Number %.1lf",passdetails.Phno);

printf("\t\t\n\nPassenger Date Of Reservation %d/%d/%d",passdetails.d,passdetails.m,passdetails.year);

printf("\t\t\n\nPassenger's Unique ID(UID) is %.1lf",passdetails.UID);

}

}

while(fread(&bp,recsize1,1,fq)>0)

{

if(reID == bp.RID1)

{

printf("\t\t\n\nReceipt Id -: %d",bp.RID1);

printf("\t\t\n\nTotal Bill amount -: Rs %d",bp.bill\_amt);

}

}

if(x!=1)

{

printf("\nSorry there is no record of the entered Receipt ID.\n\nPlease enter a Valid Receipt ID");

goto top5;

}

printf("\t\t\n\nTime : %s",ctime(&currentTime));

printf("\n");

printf("\n\nConfirm Ticket (y/n):>");

start:

scanf(" %c",&confirm);

if(confirm == 'y')

{

printf("==================");

printf("\n Reservation Done\n");

printf("==================");

printf("\nPress any key to go back to Main menu");

}

else

{

if(confirm=='n'){

printf("\nReservation Not Done!\nPress any key to go back to Main menu!");

}

else

{

printf("\nInvalid choice entered! Enter again-----> ");

goto start;

}

}

getch();

fclose(fp);

fclose(fq);

}

int cal(int y1, int y2, int h)

{

int b, c, i, t, r, n;

printf("\n\n\*\*\*\*\*\*\*\*\*\* The seat class available in this Railway Express is :- \*\*\*\*\*\*\*\*\*\n");

printf("\n\t 1) Slepper Class....\n");

printf("\n\t 2) A.C Class.......\n");

printf("\t\nEnter your choice of class for your travel: ");

scanf("%d", &i);

switch (i) {

case 1: {

strcpy(cla,"Slepper Class");

b = y2 \* h;

c = b + (b \* 0.18);

} break;

case 2: {

printf("\n\n\*\*\*\*\*\*\*\*\*\* The sub classes of seat for A.C Class available are :- \*\*\*\*\*\*\*\*\*\n");

printf("\t\t 1) 3A Class....\n");

printf("\t\t 2) 2A Class....\n");

printf("\t\t 3) 1st Class A.C.....\n");

printf("\t\nEnter Your Choice of subclass: ");

scanf("%d", &n);

switch (n) {

case 1: {

strcpy(cla,"3A Class");

b = y1 \* h;

c = b + (b \* 0.18);

} break;

case 2: {

strcpy(cla,"2A Class");

b = (y1 + 1000) \* h;

c = b + (b \* 0.18);

} break;

case 3: {

strcpy(cla,"1st Class A.C.");

b = (y1 + 5000) \* h;

c = b + (b \* 0.18);

} break;

default: {

printf("\t\tEnter Right Choice......\n");

}

}

} break;

default: {

printf("\t\tEnter Right Choice......\n");

}

}

return c;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*SPECIFICTRAIN()\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void specifictrain()

{

system("cls");

int train\_num;

printf("Enter specific train number(1001-1010): ");

scanf("%d",&train\_num);

if (train\_num==1001)

{

printf("\nTrain:\t\t\tRed Lines Express");

printf("\nDestination:\t\tBoston to Manhattan");

printf("\nDeparture time:\t\t09:00 am ");

printf("\nArrival time:\t\t11:50 am");

}

if (train\_num==1002)

{

printf("\nTrain:\t\t\tRed Lines Express");

printf("\nDestination:\t\tManhattan to Boston");

printf("\nDeparture time:\t\t12:00 pm ");

printf("\nArrival time:\t\t02:50 pm");

}

if (train\_num==1003)

{

printf("\nTrain:\t\t\tLA City Express");

printf("\nDestination:\t\tBoston to L.A");

printf("\nDeparture time:\t\t08:00 am ");

printf("\nArrival time:\t\t01:00 pm");

}

if (train\_num==1004)

{

printf("\nTrain:\t\t\tLA City Express");

printf("\nDestination:\t\tL.A to Boston");

printf("\nDeparture time:\t\t01:10 pm ");

printf("\nArrival time:\t\t06:10 pm");

}

if (train\_num==1005)

{

printf("\nTrain:\t\t\tIron City Express");

printf("\nDestination:\t\tBoston to Chicago");

printf("\nDeparture time:\t\t07:00 am ");

printf("\nArrival time:\t\t11:00 am");

}

if (train\_num==1006)

{

printf("\ntrain:\t\t\tIron City Express");

printf("\nDestination:\t\tChicago to Boston");

printf("\nDeparture time:\t\t11:05 am ");

printf("\nArrival time:\t\t03:05 pm");

}

if (train\_num==1007)

{

printf("\ntrain:\t\t\tKeystone Express");

printf("\nDestination:\t\tBoston to Washington");

printf("\nDeparture time:\t\t01:00 pm ");

printf("\nArrival time:\t\t05:00 pm");

}

if (train\_num==1008)

{

printf("\ntrain:\t\t\tKeystone Express");

printf("\n Destination:\t\tWashington to Boston");

printf("\nDeparture time:\t\t05:15 pm ");

printf("\nArrival time:\t\t09:15 pm");

}

if (train\_num==1009)

{

printf("\ntrain:\t\t\tMeteor Express");

printf("\nDestination:\t\tBoston to Miami");

printf("\nDeparture time:\t\t03:15 pm ");

printf("\nArrival time:\t\t06:15 pm");

}

if (train\_num==1010)

{

printf("\ntrain:\t\t\tMeteor Express");

printf("\nDestination:\t\tMiami to Boston");

printf("\nDeparture time:\t\t06:20 am ");

printf("\nArrival time:\t\t09:20 am");

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*VIEWDETAILS()\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void viewdetails(void)

{

system("cls");

printf("----------------------------------------------------------------------------------------------------------------");

printf("\nTr.No\tName\t\t\tDestinations\t\tDeparture Time\t\tArrival Time\n");

printf("----------------------------------------------------------------------------------------------------------------");

printf("\n1001\tRed Lines Express\tBoston to Manhattan\t\t09:00\t\t11:50");

printf("\n1002\tRed Lines Express\tManhattan To Boston\t\t12:00\t\t14:50");

printf("\n1003\tLA City Express\t\tBoston To L.A\t\t\t08:00\t\t13:00");

printf("\n1004\tLA City Express\t\tL.A To Boston\t\t\t13:10\t\t18:10");

printf("\n1005\tIron City Express\tBoston To Chicago\t\t07:00\t\t11:00");

printf("\n1006\tIron City Express\tChicago To Boston\t\t11:05\t\t15:05");

printf("\n1007\tKeystone Express\tBoston To Washington\t\t13:00\t\t17:00");

printf("\n1008\tKeystone Express\tWashington To Boston\t\t17:15\t\t21:15");

printf("\n1009\tMeteor Express\t\tBoston To Miami\t\t\t15:15\t\t\18:15");

printf("\n1010\tMeteor Express\t\tMiami To Boston\t\t\t18:20\t\t21:20");

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*SEATDETAILS()\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// Function for chosing seats

int seat(int p)

{

int i,a[50],k[50];

printf("\t -:SEAT MATRIX:- \n");

printf("\t(U) (M) (L) (L) "

" (U)\n\n");

printf("\t01 02 03\t04 "

"05\n\n");

printf("\t06 07 08\t09 "

"10\n");

printf("\t11 12 13\t14 "

"15\n\n");

printf("\t16 17 18\t19 "

"20\n");

printf("\t21 22 23\t24 "

"25\n\n");

printf("\t26 27 28\t29 "

"30\n");

printf("\t31 32 33\t34 "

"35\n\n");

printf("\t36 37 38\t39 "

"40\n");

printf("\t41 42 43\t44 "

"45\n\n");

printf("\t46 47 48\t49 "

"50\n");

printf("\t51 52 53\t54 "

"55\n\n");

printf("\t56 57 58\t59 "

"60\n");

top2:

printf("\n\n\tEnter Seat Numbers: ");

for (i = 0; i < p; i++)

{

scanf("%d", &a[i]);

if(k[i]==a[i])

{

printf("Sorry, The seat is already taken");

goto top2;

}

k[i]=a[i];

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Function for Cancel Menu()\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void cancel(void)

{

int recpID,status;

float cancelp;

long int recsize1;

FILE \*fq;

fq=fopen("Railway\_Reservation1.txt","r+");

if (fq==NULL)

{

fq=fopen("Railway\_Reservation1.txt","w+");

if (fq==NULL)

{

puts("File cannot be opened ");

exit(0);

}

}

recsize1= sizeof(bp);

system("cls");

top:

printf("\n\nEnter the receipt id to cancel the ticket: ");

fflush(stdin);

scanf("%d",&recpID);

rewind(fq);

printf("\n\n---------Cancel a Ticket----------\n");

while(fread(&bp,recsize1,1,fq)>0)

{

if(recpID == bp.RID1)

{

status=0;

printf("\n\nYour Total Bill Amount is :- Rs %d",bp.bill\_amt);

cancelp= bp.bill\_amt - bp.bill\_amt \* 0.15 ;

printf("\n\nThe amount you get after cancellation charge is -: Rs %.2f",cancelp);

}

}

if (status != 0)

{

printf("\n\nPlease enter a valid receipt ID");

goto top;

}

if(status == 0)

{

printf("\n\nYour reservation is cancelled successfully.");

}

getch();

fclose(fq);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Function for Search Menu()\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void search()

{

system("cls");

int i,j,recID,x;

long int resize,resize1;

FILE \*fp,\*fq;

fp=fopen("Railway\_Reservation.txt","r+");

if (fp==NULL)

{

puts("File cannot be opened ");

exit(0);

}

fq=fopen("Railway\_Reservation1.txt","r+");

if (fq==NULL)

{

puts("File cannot be opened ");

exit(0);

}

resize=sizeof(passdetails);

resize1=sizeof(bp);

flag5:

printf("\n\nEnter the receipt ID of the passenger: ");

fflush(stdin);

scanf("%d",&recID);

rewind(fp);

rewind(fq);

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* YOUR SEARCH DETAILS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

while(fread(&passdetails,resize,1,fp)>0)

{

if(recID == passdetails.RID)

{

x=1;

printf("\t\t\n\nPassenger Name: ");

puts(passdetails.val);

printf("\t\t\nPassenger Gender: ");

puts(passdetails.gender);

printf("\t\t\nPassenger Age: %d",passdetails.ag);

printf("\t\t\n\nPassenger Phone Number %.1lf",passdetails.Phno);

printf("\t\t\n\nPassenger Date Of Reservation %d/%d/%d",passdetails.d,passdetails.m,passdetails.year);

printf("\t\t\n\nPassenger's Unique ID(UID) is %.1lf",passdetails.UID);

}

}

if(x!=1)

{

printf("\nSorry there is no record of the entered Receipt ID.\n\nPlease enter a Valid Receipt ID");

goto flag5;

}

while(fread(&bp,resize1,1,fq)>0)

{

if(recID == bp.RID1)

{

printf("\t\t\n\nReceipt Id -: %d",bp.RID1);

printf("\t\t\n\nTotal Bill amount -: Rs %d",bp.bill\_amt);

}

}

getch();

fclose(fp);

fclose(fq);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Function for Login Menu()\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void login()

{

int a=0,i=0;

char uname[10],c=' ';

char pword[10],code[10];

char user[10]="user";

char pass[10]="pass";

do

{

printf("\n ======================= LOGIN FORM =======================\n ");

printf(" \n ENTER USERNAME:-");

scanf("%s", &uname);

printf(" \n ENTER PASSWORD:-");

while(i<10)

{

pword[i]=getch();

c=pword[i];

if(c==13) break;

else printf("\*");

i++;

}

pword[i]='\0';

//char code=pword;

i=0;

//scanf("%s",&pword);

if(strcmp(uname,"user")==0 && strcmp(pword,"pass")==0)

{

printf(" \n\n\n WELCOME TO OUR SYSTEM !! YOUR LOGIN IS SUCCESSFUL");

printf("\n\n\n\t\t\tPress any key to continue...");

getch();//holds the screen

break;

}

else

{

printf("\n SORRY !!!! LOGIN IS UNSUCESSFUL");

a++;

getch();//holds the screen

system("cls");

}

}

while(a<=2);

if (a>2)

{

printf("\nSorry you have entered the wrong username and password for four times!!!");

getch();

}

system("cls");

} [4] [5]