



C. V. RAMAN GLOBAL UNIVERSITY

Bhubaneswar, Odisha, India

Programming For C++

TEAM-6

TOPIC: RAILWAY RESERVATION SYSTEM

Submitted By:

Sl.No.	NAME	Regd. No.
1	VINAY PRABHAKAR	2201020148
2	SUNNY KUMAR	2201020423
3	PRIYANSHU KUMAR BHADANI	2201020565
4	KUNAL CHIRANIA	2201020698
5	VIVEK KUMAR	2201020767

ACKNOWLEDGEMENT

We would like to express my special thanks of gratitude to Sikander Sir who gave me an opportunity to work on this project. It helped us a lot to learn about research and we came to know about a lot of new things. we are really thankful to all of them. Secondly, we would also thanks our parents and friends who helped us in finalizing this project within the limited time frame.

This project has been a source to learn and bring our theoretical knowledge to the real-life world. So, we would really acknowledge his help and guidance for this project. Once again, thank you everyone who are involved in making this project a success.

CERTIFICATE

This is to certify that **TEAM -6** of PROGRAMMING FOR C++ CLASS of B.Tech 6th semester has successfully completed the project entitled **RAILWAY RESERVATION SYSYEM C++ LANGUAGE** to our satisfaction and submitted the same during the academic year **2024-25**. The project is the result of his efforts & endeavors.

Under the guidance of – Sikander Sir

CONTENT

1. Introduction
2. Proposed System
3. Requirements
4. Source Code
5. Output
6. Conclusion
7. Reference/Bibliography

Introduction

In Railway Reservation System we are perform different tasks. we are going to collect all data of passenger and about the train .Firstly user have to choose which class and where he/she want to travel. After that user have to provide the date when will passenger want the Railway ticket and when will passenger return also after getting the data from passenger we check the Train is available or not. If Train is not available then we are going to print Train is unavailable and calculate the rent as per the data chart.

Here we are using user Defined functions.

We are also using Looping statements as well as switch case.

There are three types of loops :-

1. For loop
2. While loop
3. Do while loop

Benefits of switch case :-

1. we can check multiple statement
2. Easy to use and easy to control the conditional statement
3. Allows a Variable to be tested for equality against a list of values.

There are three type of conditional statements :-

1. If
2. Else if
3. Else

Proposed System

We are using different library that are given below :

1. **#include<iostream>**: For standard input and output operations.
2. **#include<fstream>**: Provides file stream classes (e.g., ifstream, ofstream) for file input/output.
3. **#include<string.h>**: Enables access to a set of character functions for handling strings.
4. **#include<random>**: Facilitates the use of random number generation and related functionalities.
5. **#include<vector>**: Allows the use of the vector container for dynamic arrays in C++.
6. **#include<sstream>**: Supports string stream operations, useful for parsing and formatting strings.
7. **#include<cstdlib>**: Provides general-purpose C library functions, including memory allocation and random number generation.
8. **#include<ctime>**: Includes functions related to date and time, often used for time-based operations in C++.

We are using File Handling.

There are Different type of File :-

1. Text File
2. Binary File
3. Csv File

Here, we are using File handling to Store data and Fetch the data from the File.
By using Write and Read mode we are going to add and fetch the data.

Requirements

There are following requirements we are using :-

- 1.Input from User
- 2.Different types of libraries
- 3.Different type of Data type
- 4.A new File that can store data.
- 5.Functions
 - 1.In-built
 - 2.User Define
- 6.Switch cases
- 7.Structure
- 8.Arrays(String Type)
- 9.Conditinal Statements
- 10.Looping Statements
- 11.Exit & Break statements

Source Code

```
#include<fstream>
#include<string.h>
#include<random>
#include<vector>
#include<sstream>
#include<cstdlib>
#include<ctime>

using namespace std;
#include<iostream>
int pnr_generator();
vector<string> simple_token(string str);

class Admin{
    // Admin purpose
private:
    char admin_user[100];
    char admin_pass[100];

public:
    void get_admin_details() {
        cout<<"Enter Username (Without Spaces): ";
        cin>>admin_user;

        cout<<"Enter Password: ";
        cin>>admin_pass;
    }

    void display_admin_details() {
        cout<<"Username: "<<admin_user<<endl;
        cout<<"Password: "<<admin_pass<<endl;
    }

    string get_user_name() {
        return admin_user;
    }

    string get_user_pass() {
        return admin_pass;
    }
};
```



```

class Train{
    // Train Details To Enter By the Admin Mode
public:
    string t_name;
    string t_code;
    int sl, gen, ac;
    float sl_fare, gen_fare, ac_fare;

    string start_point;
    string end_point;

    void get_train_details() {
        cin.ignore();
        cout<<"Enter name of Train: ";
        getline(cin, t_name);

        // cin.ignore();
        cout<<"Enter Train code: ";
        getline(cin, t_code);

        cout<<"Enter Starting Point: ";
        cin>>start_point;

        cout<<"Enter End Point: ";
        cin>>end_point;

        cout<<"Enter total number of seats in Sleeper category: ";
        cin>>sl;

        cout<<"Enter price per Sleeper seat: ";
        cin>>sl_fare;

        cout<<"Enter total number of seats in General category: ";
        cin>>gen;

        cout<<"Enter price per General seat: ";
        cin>>gen_fare;

        cout<<"Enter total number of seats in AC category: ";
        cin>>ac;

        cout<<"Enter price per AC seat: ";
        cin>>ac_fare;
    }

    void display_train_details() {
        cout << "\n|Train Name" << "\t" << "|Train Code" << "\t" <<
"|Boarding pt." << "\t" << "|Destination pt." << "\t";
    }
}

```

```

        cout << "|SL-Class" << "\t"<< "|SL-Class Fare" << "\t" << "|GEN-
Class" << "\t" << "|GEN-Class Fare" << "\t";
        cout << "|AC-Class" << "\t"<< "|AC-Class Fare" << "\t";

        cout << "|" << t_name << "\t\t" << "|" << t_code << "\t\t" <<
        "|" << start_point << "\t\t" << "|" << end_point << "\t\t\t";
        cout << "|" << sl << "\t\t" << "|" << sl_fare << "\t\t" << "|" <<
gen<< "\t\t" << "|" << gen_fare << "\t\t";
        cout << "|" << ac << "\t\t" << "|" << ac_fare << "\t\t";
    }
};

class Passenger{
    // Passenger Info
private:
    char passenger_user[100];
    char passenger_pass[100];

public:
    void get_p_details() {
        cout<<"Enter Username: ";
        cin>>passenger_user;

        cout<<"Enter Password: ";
        cin>>passenger_pass;
    }

    void display_p_details() {
        cout<<"Username: "<<passenger_user<<endl;
        cout<<"Password: "<<passenger_pass<<endl;
    }

    string get_user_name() {
        return passenger_user;
    }

    string get_user_pass() {
        return passenger_pass;
    }
};

class Reservation{
public:
    char t_code[20];
    int nosr, pnr;
    char pname[10][100];
    int age[10];
    int d,m,y;

```

```

char category[10];

void pdetails() {
    cout<<"Enter Train Code: ";
    cin.ignore();
    cin.getline(t_code, 20);

    cout<<"Enter number seats you want to reserve: ";
    cin>>nosr;

    for(int i = 0; i<nosr; i++) {
        cout<<"Enter Passenger Name: ";
        cin.ignore();
        cin.getline(pname[i], 100);

        cout<<"Enter Passenger Age: ";
        cin>>age[i];
    }

    cout<<"Enter Date of Travel in the Format DD MM YYYY(Separation
by spaces): ";
    cin>>d>>m>>y;

    cout<<"Enter Category(SL, GEN, AC): ";
    cin>>category;

    pnr = pnr_generator();
}

void display_pdetails() {
    cout<<"Train Code: "<<t_code<<" PNR: "<<pnr<<endl;
    cout<<"Boarding Date: "<<d<<"-"<<m<<"-"<<y<<endl;
    for(int i = 0; i<nosr; i++) {
        cout<<"Name: "<<pname[i]<<" Age: "<<age[i]<<" Category:
"<<category<<endl;
    }
}

string gen_pdetails() {
    string name = "";
    for(int i = 0; i<nosr; i++) {
        string temp = (string)pname[i]+" "+to_string(age[i])+" ";
        name = name + temp;
    }

    string str = (string)t_code+" "+to_string(pnr)+"
Boarding_Date:"<to_string(d)<to_string(m)<to_string(y)

```

```

        +" "+string(category)+" "+name;

        return str;
    }
};

void login_signin(Admin &A, const string &filename, Train &T);
void login_signin_p(Passenger &P, const string &filename, Reservation &R);
void displayFileContent(const string &filename);
// int searchAndDisplay(const string &filename, const string &searchStr);
string search(const string &filename, const string &searchStr);
void admin_management(Train &T);
void admin_mode_management(Admin &A, Train &T);
void user_mode_management(Passenger &P, Reservation &R);
void user_management(Reservation &R);

int main() {
    cout<<"***** Welcome to Main Menu
*****"<<endl;
    int option;
    Admin A;
    Passenger P;
    Train T;
    Reservation R;
    string username, password, str, filename;
    fstream file;

    while(1){
        cout<<"1. Admin"<<endl;
        cout<<"2. User"<<endl;
        cout<<"3. Exit Program"<<endl;

        cout<<"Enter choice: ";
        cin>>option;

        switch(option) {
            case 1:
                cout<<"*****ADMIN MODE*****"<<endl;
                admin_mode_management(A, T);
                break;

            case 2:
                cout<<"*****USER MODE*****"<<endl;
                user_mode_management(P, R);
                break;

            case 3:

```

```

        exit(0);

    default:
        cout<<"Enter a valid option."<<endl;
        break;
    }
}

return 0;
}

int pnr_generator() {

    srand(time(0));
    int lb = 20, ub = 100;
    int random = (rand() % (ub - lb + 1)) + lb;
    return random;
}

void displayFileContent(const string &filename) {
    fstream file(filename);
    string line;
    vector<string> v;

    if(file.is_open()) {
        // cout<<"File Content: "<<endl;
        while(getline(file, line)) {
            v = simple_token(line);
            for(auto value: v) {
                cout<<value<<"\t";
            }
            cout<<endl;
        }
        cout<<endl;

        file.close();
    }else{
        cout<<"Failed to open file."<<endl;
    }
}

void admin_management(Train &T){
    cout<<"***** Welcome to Admin Management
*****"<<endl;

    int option1;
    string str, line, x, t_code, filename = "Train_details.txt";
    fstream file, temp;
    // const char *p;

```

```

do {
    cout<<"1. Add Train Details"<<endl;
    cout<<"2. Delete Train Details"<<endl;
    cout<<"3. Display Train Details"<<endl;
    cout<<"4. Go Back to Main Menu"<<endl;

    cout<<"Enter your option: ";
    cin>>option1;

    switch(option1) {
        case 1:
            T.get_train_details();

            str = (string)T.t_name+" "+(string)T.t_code+"
" +(string)T.start_point+" "+(string)T.end_point+" "+
            to_string(T.sl)+" "+to_string(T.sl_fare)+"
" +to_string(T.gen)+" "+to_string(T.gen_fare)+" "+to_string(T.ac)+"
" +to_string(T.ac_fare);

            file.open(filename, ios::in|ios::out|ios::app);
            file<<str<<endl;
            file.close();
            break;

        case 2:
            cout<<"Enter Train Code: ";
            cin>>t_code;

            str = search(filename, t_code);
            // cout<<str<<endl;

            file.open(filename, ios::in | ios::out | ios::app);
            temp.open("temp1.txt", ios::in | ios::out | ios::app);
            while(getline(file, line)) {
                if(line == str) {
                    // cout<<"True"<<endl;
                }
                if(line != str) {
                    temp<<line<<endl;
                }
            }

            temp.close();
            file.close();

            remove("Train_details.txt");
            rename("temp1.txt", "Train_details.txt");

```

```

        break;

        case 3:
            cout << "\n|Train Name" << "\t\t\t" << "|Train Code" << "\t"
<< "|Boarding pt." << "\t" << "|Destination pt." << "\t";
            cout << "|SL-Class" << "\t" << "|SL-Class Fare" << "\t" <<
"|GEN-Class" << "\t" << "|GEN-Class Fare" << "\t";
            cout << "|AC-Class" << "\t" << "|AC-Class Fare" << "\t";
            cout<<endl;
            displayFileContent(filename);
            break;

        case 4:
            break;

        default:
            cout<<"Enter a valid option"<<endl;
            break;
    }
}while(option1 != 4);
}

void admin_mode_management(Admin &A, Train&T){
    int option;
    string username, password, str, filename;
    fstream file;
    do{
        filename = "admin_login.txt";
        cout<<"1. Login"<<endl;
        cout<<"2. Add new user"<<endl;
        cout<<"3. Display Users"<<endl;
        cout<<"4. Go Back to Main Menu"<<endl;

        cout<<"Enter your option: ";
        cin>>option;

        switch(option) {
            case 1:
                login_signin(A, filename, T);
                break;

            case 2:
                A.get_admin_details();
                username = A.get_user_name();
                password = A.get_user_pass();
                str = username + " " + password;

```

```

        file.open(filename, ios::in|ios::out|ios::app);
        file<<str<<endl;
        file.close();
        cout<<"Successfully Added New Admin User"<<endl;
        break;

    case 3:
        cout<<"Username: "<<"\t"<<"Password: "<<endl;
        displayFileContent(filename);
        break;

    case 4:
        break;

    default:
        cout<<"Enter a valid option."<<endl;
        break;
    }

}while(option != 4);
}

void user_mode_management(Passenger &P, Reservation &R) {
    int option;
    string username, password, str, filename = "user_login.txt";
    fstream file;

    do{
        cout<<"1. Login"<<endl;
        cout<<"2. Sign In"<<endl;
        cout<<"3. Go back to Main Menu"<<endl;

        cout<<"Enter your option: ";
        cin>>option;

        switch(option) {
            case 1:
                login_signin_p(P, filename, R);
                break;

            case 2:
                P.get_p_details();
                username = P.get_user_name();
                password = P.get_user_pass();
                str = username + " " + password;
                file.open(filename, ios::in|ios::out|ios::app);
                file<<str<<endl;
                file.close();

```



```

        cout<<"Successfully Added New User"<<endl;
        break;

    case 3:
        break;

    default:
        cout<<"Enter a valid option"<<endl;
        break;
    }
}while(option != 3);
}

void user_management(Reservation &R) {
    int option;
    string str,pnr,x,line, head, sub_head, filename = "reservation.txt";
    fstream file, temp;
    vector<string> v;

    do{
        cout<<"1. Display Availabe Train Bookings"<<endl;
        cout<<"2. Reserve Seats"<<endl;
        cout<<"3. Cancel Reservation"<<endl;
        cout<<"4. Display Reservation"<<endl;
        cout<<"5. Generate E-Ticket"<<endl;
        cout<<"6. Go Back to Main Menu"<<endl;

        cout<<"Enter your option: ";
        cin>>option;

        switch(option) {
            case 1:
                cout << "\n|Train Name" << "\t\t\t" << "|Train Code" << "\t"
<< "|Boarding pt." << "\t" << "|Destination pt." << "\t";
                cout << "|SL-Class" << "\t" << "|SL-Class Fare" << "\t" <<
"|GEN-Class" << "\t" << "|GEN-Class Fare" << "\t";
                cout << "|AC-Class" << "\t" << "|AC-Class Fare" << "\t";
                cout<<endl;
                displayFileContent("Train_details.txt");
                break;

            case 2:
                R.pdetails();
                str = R.gen_pdetails();
                file.open(filename, ios::in|ios::out|ios::app);
                file<<str<<endl;
                file.close();
                break;

```

```

case 3:
    cout<<"Enter your PNR number: ";
    cin>>pnr;

    x = search(filename, pnr);
    if(x != "NULL"){
        file.open(filename, ios::in | ios::out | ios::app);
        temp.open("temp.txt", ios::in | ios::out | ios::app);
        while(getline(file, line)) {
            if(line != x){
                temp<<line<<endl;
            }
        }
        temp.close();
        file.close();

        remove("reservation.txt");
        rename("temp.txt", "reservation.txt");
    }

    break;

case 4:
    cout<<"Enter your PNR number: ";
    cin>>pnr;

    cout<<"Your Reservation Data: "<<endl;
    cout<<endl;
    x = search(filename, pnr);
    if(x!="NULL"){
        v = simple_token(x);
        cout<<"Train Code: "<<v[0]<<" "<<"PNR number: "<<v[1]<<"
"<<v[2]<<" "<<"Category: "<<v[3]<<endl;
        cout<<"Name: "<<"\t"<<"Age: "<<endl;
        for(auto it = v.begin()+4; it<v.end(); it++) {
            cout<<*it<<" ";
        }
    }
    cout<<endl;
    break;

case 5:
    file.open("e_ticket.txt", ios::in | ios::out);
    head = "***** E-Ticket
*****";
    sub_head = "\t***** Case 7 Railway Reservation System
*****";

```

```

        cout<<"Enter your PNR number: ";
        cin>>pnr;

        x = search(filename, pnr);
        if(x!="NULL"){
            file<<head<<endl;
            file<<sub_head<<endl;
            v = simple_token(x);
            file<<"\n\nTrain code: "<<v[0]<<endl;
            file<<"PNR number: "<<v[1]<<endl;
            file<<v[2]<<endl;
            file<<"Category: "<<v[3]<<endl;
            file<<"Name: "<<"\t\t"<<"Age: "<<endl;
            // cout<<v.size();
            for(int i = 4; i<v.size(); i++) {
                file<<v[i]<<" ";
            }
            cout<<endl;
        }

        cout<<"Your E-Ticket is Generated"<<endl<<endl;

        break;

    case 6:
        break;

    default:
        cout<<"Enter a valid option"<<endl;
        break;
}

}while(option!=6);
}

void login_signin(Admin &A, const string &filename, Train &T) {

    fstream file(filename);
    A.get_admin_details();

    string searchStr = A.get_user_name() + " " + A.get_user_pass();
    string x = search(filename, searchStr);

    if(x != "NULL") {
        cout<<"Successfully Logged"<<endl;
        admin_management(T);
    }else{
        cout<<"Don't have any Account? Sign In"<<endl;
    }
}

```

```

    }
}

void login_signin_p(Passenger &P, const string &filename, Reservation &R) {

    fstream file(filename);
    P.get_p_details();

    string searchStr = P.get_user_name() + " " + P.get_user_pass();
    string x = search(filename, searchStr);

    if(x != "NULL") {
        cout<<"Successfully Logined"<<endl;
        user_management(R);
    }else{
        cout<<"Don't have any Account? Sign In"<<endl;
    }
}

string search(const string & filename, const string & searchStr){
    fstream file(filename);
    string line;

    int lineNumber = 1;
    while(getline(file, line)) {
        if(line.find(searchStr) != string::npos) {
            return line;
        }
    }
    file.close();

    return "NULL";
}

vector<string> simple_token(string str) {
    stringstream ss(str);
    string word;
    vector<string> words;

    while(ss>>word){
        words.push_back(word);
    }

    return words;
}

```

Output

USER HAVE TO PRESS 1,2,3 FOR

```
PS C:\Users\nayak\OneDrive\Desktop\railway_reservation_system> cd "c:\
if ($?) { g++ railway_reservation_system4.cpp -o railway_reservation_sy
***** Welcome to Main Menu *****
1. Admin
2. User
3. Exit Program
Enter choice:
```

LOGIN SIGN SUCESSFULL

```
Enter choice: 2
*****USER MODE*****
1. Login
2. Sign In
3. Go back to Main Menu
Enter your option: 1
Enter Username: debasish
Enter Password: 1234
Successfully Logged
```

DISPLAY ALL THE AVAILABLE OPERATION

```
Successfully Logged
1. Display Availabe Train Bookings
2. Reserve Seats
3. Cancel Reservation
4. Display Reservation
5. Generate E-Ticket
6. Go Back to Main Menu
Enter your option: █
```

ENTERING THE PASSENGER DETAILS

```
Successfully Logged
1. Display Availabe Train Bookings
2. Reserve Seats
3. Cancel Reservation
4. Display Reservation
5. Generate E-Ticket
6. Go Back to Main Menu
Enter your option: 2
Enter Train Code: 1235
Enter number seats you want to reserve: 5
Enter Passenger Name: DEBASISH
Enter Passenger Age: 18
Enter Passenger Name: VINAY
Enter Passenger Age: 19
Enter Passenger Name: PRIYANSHU
Enter Passenger Age: 19
Enter Passenger Name: SURAJ
Enter Passenger Age: 20
Enter Passenger Name: BHABANI
Enter Passenger Age: 21
Enter Date of Travel in the Format DD MM YYYY(Separation by spaces): 21 11 2023
```

PRESS 6 For Exiting the program AND GO TO MAIN MENU

```
1. Display Availabe Train Bookings
2. Reserve Seats
3. Cancel Reservation
4. Display Reservation
5. Generate E-Ticket
6. Go Back to Main Menu
Enter your option: 6
1. Login
2. Sign In
3. Go back to Main Menu
Enter your option: █
```

DATA STORED IN FILES

```
14598 36 Boarding_Date:12:5:2023 s1 debasish 19 vinay 25 suraj 20 priyanshu 21
1456 97 Boarding_Date:21:0:1857290240 DEBASISH 18 VINAY 19 PRIYANSHU 19 BHABANI 19 SURAJ 20
97 Boarding_Date:21:0:1857290240 18 19 19 19 20
97 Boarding_Date:21:0:1857290240 18 19 19 19 20
97 Boarding_Date:21:0:1857290240 18 19 19 19 20
97 Boarding_Date:21:0:1857290240 18 19 19 19 20
97 Boarding_Date:21:0:1857290240 18 19 19 19 20
97 Boarding_Date:21:0:1857290240 18 19 19 19 20
97 Boarding_Date:21:0:1857290240 18 19 19 19 20
97 Boarding_Date:21:0:1857290240 18 19 19 19 20
97 Boarding_Date:21:0:1857290240 18 19 19 19 20
97 Boarding_Date:21:0:1857290240 18 19 19 19 20
97 Boarding_Date:21:0:1857290240 18 19 19 19 20
97 Boarding_Date:21:0:1857290240 18 19 19 19 20
97 Boarding_Date:21:0:1857290240 18 19 19 19 20
97 Boarding_Date:21:0:1857290240 18 19 19 19 20
97 Boarding_Date:21:0:1857290240 18 19 19 19 20
97 Boarding_Date:21:0:1857290240 18 19 19 19 20
```

LOGIN INFORMATION

```
debasish 1234
debasish 1234
DEBASISH 1234
```

Conclusion

In conclusion, a railway reservation system in C++ is a valuable tool for managing and facilitating the process of renting vehicles to customers. It offers several advantages, such as efficient vehicle tracking, streamlined reservations, automated billing, and improved customer service. By utilizing the C++ programming language, the system can be implemented with robustness and reliability.

With the railway reservation system in C++, customers can conveniently make reservations, check vehicle availability, and receive accurate pricing information. The system can also handle online bookings, enabling customers to reserve train tickets from anywhere at any time. By automating these processes, the system minimizes human error and reduces the time and effort required to manage reservations and rental transactions.

Overall, a well-designed train management system in C optimizes the management of a train reservation company's operations, enhances customer experience, and increases overall efficiency. It allows for smoother and more streamlined processes, ensuring that tickets are efficiently utilized, customers are served promptly, and accurate records are maintained. Implementing such a system can result in improved profitability and customer satisfaction for the train management system.

Reference

1. W3school → <https://www.w3schools.com/>
2. Geeksforgeek → <https://www.geeksforgeeks.org/>
3. Java point → <https://www.javatpoint.com/>