

# \* Project

## File of C

# Programming

- Name - Vedans Rawat
- Course - Btech CSE 1 Year
- Subject - C programming
- Submitted To - Rahul Prasad
  - Batch - 20
- Topic :- Online Train Ticket Booking Management.

## **Abstract**

- The *Online Train Booking Management System* is a C-based application designed to simplify and automate the process of reserving train tickets. This project provides users with essential functionalities such as viewing available trains, checking schedules, booking tickets, generating PNR details, and managing passenger records. Implemented using file handling in C, the system stores train and booking information securely in text or binary files, enabling retrieval and updates without requiring a database.
  
- The project demonstrates key programming concepts including structures, functions, arrays, pointers, and file I/O. It aims to provide a user-friendly interface while ensuring efficient data processing and minimal errors during booking operations. By simulating a real-world reservation workflow, the system highlights how C programming can be applied to develop practical, menu-driven management software. This project also emphasizes modular design, data validation, and basic security measures, making it a valuable learning tool for beginners exploring system development in C.



## Objectives of an Online Train Ticket Booking System

### **1. Provide a Fast and Convenient Ticket Booking Experience**

- Allow users to search trains, check availability, and book tickets anytime, anywhere.
- Reduce the need for physical counters and long queues.
- Ensure a simple, intuitive, and user-friendly interface.

### **2. Automate the Entire Reservation Process**

- Handle seat availability, PNR generation, fare calculation, and booking confirmations automatically.
- Reduce manual errors and operational delays.
- Streamline backend operations for administrators.

### **3. Offer Real-Time Train and Seat Information**

- Display live seat availability.
- Show updated train schedules, routes, and stoppages.
- Notify users instantly about delays, cancellations, or changes.

## **4. Enable Secure and Multiple Payment Options**

- Provide secure payment gateways (UPI, cards, net banking, wallets).
- Ensure encrypted transactions and user data protection.
- Issue e-tickets instantly after successful payments.

## **5. Maintain User Accounts and Booking History**

- Allow users to create profiles.
- Save passenger details for quick future bookings.
- Provide downloadable past tickets, invoices, and cancellation receipts.

## **6. Facilitate Easy Ticket Cancellation and Refunds**

- Provide hassle-free cancellation options.
- Calculate refunds automatically according to rules.
- Notify users about refund status.

## **7. Improve Accessibility and Inclusivity**

- Support mobile devices, low-bandwidth mode, and regional languages.
- Provide accessibility features for elderly and differently-abled users.

## **8. Ensure Reliability and High System Performance**

- Handle heavy traffic during peak times (festivals, weekends).

- Ensure minimal downtime and fast response time.
- Use scalable architecture for future expansion.

## 9. Integrate Additional Travel-Related Services (Optional)

- Seat selection, meal booking, hotel packages, etc.
- Provide travel insurance options.
- Offer train tracking features.

## 10. Provide Administrative Tools

- Let admins add/manage trains, schedules, routes, and fares.
- Give analytics reports on bookings, revenue, and traffic.
- Support customer service with ticket management tools.

### Problem Definition

→ In traditional train ticket reservation systems, users often experience difficulties such as long queues, manual errors, limited access to train schedules, and slow processing of booking requests. Manual record-keeping is prone to data loss, duplication, and inefficient retrieval of passenger or train information. These challenges highlight the need for a simple, automated system that can manage bookings reliably and quickly.

## ❖ **Advantages of an Online Train Ticket Booking System**

### **1. Convenience & 24/7 Availability**

- Users can book tickets anytime, anywhere.
- There is no need to visit railway stations or ticket counters.
- Ideal for people with busy schedules.

### **2. Time-Saving**

- Instant train search, seat availability check, and booking.
- Avoids long queues and waiting times at counters.

### **3. Real-Time Information**

- Live updates on train timings, seats, fares, delays, and cancellations.
- Helps passengers plan their journey better.

## **4. Secure and Multiple Payment Options**

- Digital payments (UPI, cards, wallets, net banking).
- Secure, encrypted transactions reduce risks of fraud.

## **5. Easy Cancellations & Refunds**

- One-click cancellation process.
- Automatic refund calculations and status tracking.

## **6. Accessibility for Everyone**

- Works on mobile, laptop, and low bandwidth.
- Accessible features for elderly or differently-abled users.
- Support for regional languages (optional features).

## **7. Booking History & User Profiles**

- Saves passenger details for faster future bookings.
- Users can view/download past tickets anytime.

## **8. Reduces Operational Costs**

- Less staff are needed at physical counters.

- Automated systems reduce manual work and errors.

## 9. Wider Reach

- People from all regions can access the booking service.
- Encourages more people to use trains due to easier booking.

## 10. Enhances Customer Satisfaction

- Faster service, instant confirmation, and fewer errors.
- Users can track train status and get timely updates.

## Implementation

→ The implementation of the *Online Train Booking Management System* in C involves designing a menu-driven application that uses structures, functions, and file handling to manage train and passenger data. The project is developed step-by-step as follows:



## CODE

```
#include <stdio.h>
#include <string.h>
```

```
int main() {
    char from[30], to[30], date[20], travelClass[30], trainName[50] ,
booking[8] ;

    int passengers;
    int trainChoice, classChoice,choice;
    float total,amount=0;

    printf("=====\\n");
    printf("    TRACK & RIDE EXPRESS - BOOK TICKET\\n");
    printf("=====\\n\\n");

    printf("Select Journey Details:\\n");

    printf("From : ");
    scanf("%29s", from);

    printf("To : ");
    scanf("%29s", to);

    printf("Date of Journey (DD/MM/YYYY): ");
    scanf("%19s", date);

    printf("Number of Passengers: ");
    scanf("%d", &passengers);

    printf("\\nSelect Class:\\n");
    printf("1. First Class AC\\n");
    printf("2. Second Class AC\\n");
    printf("3. Third Class AC\\n");
    printf("4. Sleeper\\n");
```

```
printf("Enter choice: ");
scanf("%d", &classChoice);

switch (classChoice){
    case 1:
        strcpy(travelClass, "First Class AC");
        choice=1;
        break;
    case 2:
        strcpy(travelClass, "Second Class AC");
        choice=2;
        break;
    case 3:
        strcpy(travelClass, "Third Class AC");
        choice=3;
        break;
    case 4:
        strcpy(travelClass, "Sleeper");
        choice=4;
        break;
    default:
        printf("Invalid choice!\n");
        return 0;
}

printf("\nAvailable Trains:\n");
printf("1. Rajdhani Express (%s to %s)\n",from , to );
printf("2. Shatabdi Express (%s to %s)\n",from , to);
printf("3. Bangalore Rajdhani Express (%s to %s)\n",from , to);
printf("4. Amarnath Express (%s to %s)\n",from , to );
printf("5. Vande Bharat Express (%s to %s)\n",from , to );
printf("6. Tejas Express (%s to %s)\n",from , to );
```

```
printf("Enter train choice: ");
scanf("%d", &trainChoice);

if (trainChoice == 1)
    strcpy(trainName, "Rajdhani Express");
else if (trainChoice == 2)
    strcpy(trainName, "Shatabdi Express");
else if (trainChoice == 3)
    strcpy(trainName, "Banglore Rajdhani Express");
else if (trainChoice == 4)
    strcpy(trainName, "Amarnath Express");
else if (trainChoice == 5)
    strcpy(trainName, "Vande Bharat Express");
else if (trainChoice == 6)
    strcpy(trainName, "Tegas Express");
else {
    printf("Invalid train choice!\n");
    return 0;
}
printf("\nAmount of ticket/n");
if (trainChoice == 1)
{
    if(choice == 1)
    {
        printf("\nthe price of one ticket in first class AC is 5390.00\n");
        amount=amount+5390.00;
    }
    else if(choice == 2)
    {
        printf("\nthe price of one ticket in the second class AC is
4340.00\n");
        amount=amount+4340.00;
    }
}
```

```
    }
    else if(choice == 3)
    {
        printf("\nthe price of one ticket in the second class AC is
3000.00\n");
        amount=amount+3000.00;
    }
    else if(choice == 4)
    {
        printf("\nthe price of one ticket in the second class AC is
1200.00\n");
        amount=amount+1200.00;
    }
}
else if (trainChoice == 2)
{
    if(choice == 1)
    {
        printf("\nthe price of one ticket in the second class AC is
3330.00\n");
        amount=amount+3330.00;
    }
    else if(choice == 2)
    {
        printf("\nthe price of one ticket in the second class AC is
2020.00\n");
        amount=amount+2020.00;
    }
    else if(choice == 3)
    {
        printf("\nthe price of one ticket in the Third class AC is
1520.00\n");
    }
}
```

```
        amount=amount+1520.00;
    }
    else if(choice == 4)
    {
        printf("\nthe price of one ticket in the second class AC is
990.00\n");
        amount=amount+990.00;
    }
}
else if (trainChoice == 3)
{
    if(choice == 1)
    {
        printf("\nthe price of one ticket in the first class AC is 3300.00\n");
        amount=amount+3300.00;
    }
    else if(choice == 2)
    {
        printf("\nthe price of one ticket in the second class AC is
2810.00\n");
        amount=amount+2810.00;
    }
    else if(choice == 3)
    {
        printf("\nthe price of one ticket in the third class AC is 1960.00\n");
        amount=amount+1960.00;
    }
    else if(choice == 4)
    {
        printf("\nthe price of one ticket in the sleeper Class is 980.00\n");
        amount=amount+980.00;
    }
}
```

```
}

else if (trainChoice == 4)
{
    if(choice == 1)
    {
        printf("\nthe price of one ticket in the First class AC is 4900.00\n");
        amount=amount+4900.00;
    }
    else if(choice == 2)
    {
        printf("\nthe price of one ticket in the second class AC is
3330.00\n");
        amount=amount+3330.00;
    }
    else if(choice == 3)
    {
        printf("\nthe price of one ticket in the Third class AC is
2150.00\n");
        amount=amount+2150.00;
    }
    else if(choice == 4)
    {
        printf("\nthe price of one ticket in the sleeper class is 1100.00\n");
        amount=amount+1100.00;
    }
}

else if (trainChoice == 5)
{
    if(choice == 1)
    {
```

```
    printf("\nthe price of one ticket in the First class AC is 4000.00\n");
    amount=amount+4000.00;
}
else if(choice == 2)
{
    printf("\nthe price of one ticket in the second class AC is
3200.00\n");
    amount=amount+3200.00;
}
else if(choice == 3)
{
    printf("\nthe price of one ticket in the Third class AC is
1500.00\n");
    amount=amount+1500.00;
}
else if(choice == 4)
{
    printf("\nthe price of one ticket in the sleeper class is 1020.00\n");
    amount=amount+1020.00;
}
}
else if (trainChoice == 6)
{
    if(choice == 1)
    {
        printf("\nthe price of one ticket in the First class AC is 4750.00\n");
        amount=amount+4750.00;
    }
    else if(choice == 2)
    {
        printf("\nthe price of one ticket in the second class AC is
3652.00\n");
    }
}
```

```

        amount=amount+3652.00;
    }
    else if(choice == 3)
    {
        printf("\nthe price of one ticket in the Third class AC is
1965.00\n");
        amount=amount+1965.00;
    }
    else if(choice == 4)
    {
        printf("\nthe price of one ticket in the sleeper class is 1600.00\n");
        amount=amount+1600.00;
    }
}
else
{
    printf("Invalid train choice!\n");
    return 0;
}
printf("\nDo you want to continue booking ticket\n");
scanf("%7s",booking);
if(strcmp(booking, "yes") ==0 || strcmp(booking, "YES") ==0)
{
    total = passengers * amount;

    printf("\n-----\n");
    printf("      BOOKING SUMMARY\n");
    printf("\n-----\n");
    printf("Train Name   : %s\n", trainName);
    printf("From       : %s\n", from);
    printf("To        : %s\n", to);
    printf("Date      : %s\n", date);
}

```

```

printf("Passengers : %d\n", passengers);
printf("Class      : %s\n", travelClass);
printf("Ticket Price : ₹%.2f\n", amount);
printf("\n-----\n");
printf("Total Amount : ₹%.2f\n", total);
printf("\n-----\n");
printf("\nPayment Successful! \n");
printf("Your e-ticket has been confirmed.\n");
printf("\n-----\n");
printf("Thank you for choosing Track & Ride Express!");
printf("\n-----\n");
amount=0;
choice=0;
}
else if(strcmp(booking,"no") == 0 || strcmp(booking,"NO") == 0)
{
    printf("\nTHANK YOU FOR VISITING\n");
}
else
{
    printf("\ninvalid choice\n");
}
return 0;
}

```

## Testing And Result

```
===== TRACK & RIDE EXPRESS - BOOK TICKET =====
Select Journey Details:
From : dehradun
To : delhi
Date of Journey (DD/MM/YYYY) : 24/11/2025
Number of Passengers: 2

Select Class:
1. First Class AC
2. Second Class AC
3. Third Class AC
4. Sleeper
Enter choice: 1

Available Trains:
1. Rajdhani Express (dehradun to delhi)
2. Shatabdi Express (dehradun to delhi)
3. Bangalore Rajdhani Express (dehradun to delhi)
4. Amarnath Express (dehradun to delhi)
5. Vande Bharat Express (dehradun to delhi)
6. Tejas Express (dehradun to delhi)
Enter train choice: 5

Amount of ticket/
the price of one ticket in the First class AC is 4000.00

Do you want to continue booking ticket
yes

----- BOOKING SUMMARY -----
Train Name      : Vande Bharat Express
From           : dehradun
To             : delhi
Date           : 24/11/2025
Passengers     : 2
Class          : First Class AC
Ticket Price   : ₹4000.00

Total Amount    : ₹8000.00

----- Payment Successful!
Your e-ticket has been confirmed.

----- Thank you for choosing Track & Ride Express!
```

## ❖ Future Scope of an Online Train Ticket Booking System

### 1. AI-Based Ticket Booking Assistance

- Smart chatbot for booking, cancellations, and inquiries.
- Personalized travel suggestions based on past journeys.
- Predictive seat availability and fare forecasting.

### 2. Advanced Train Tracking & Live Updates

- Real-time GPS-based train tracking.
- Push notifications for delays, platform changes, or cancellations.
- Integration with live railway operation data.

### **3. Dynamic Pricing & Smart Recommendation Engine**

- AI-powered fare optimization during peak/off-peak times.
- Smart recommendations for alternate trains, routes, or dates.
- Automatic alerts when seats become available (tatkal/waitlist).

### **4. Integration With Other Transportation Services**

- Combined booking for taxis, buses, metros, and flights.
- One-stop travel solution within the same platform.
- Multi-modal transport planning and fare comparison.

### **5. Enhanced Security Features**

- Biometric login (fingerprint, face ID).
- Fraud detection using machine learning.
- End-to-end encrypted transactions with blockchain technology.

### **6. Offline Booking via Mobile App**

- “Lite Mode” for low internet connectivity areas.
- Booking via SMS/WhatsApp for remote regions or elderly users.

## **7. Smart Coach & Seat Selection**

- 3D coach view for selecting seats visually.
- Real-time seat occupancy maps.
- Smart suggestions for best seats based on preference.

## **8. Voice-Based Booking**

- Voice search and voice booking for trains (in multiple languages).
- Useful for visually impaired or elderly users.

## **9. Integration With Tourism & Hospitality**

- Add-ons: hotels, local travel passes, meal pre-orders, tour packages.
- Personalized itineraries based on destination.

## **10. Data Analytics & Insights**

- Predictive analytics for railway traffic, revenue, and crowd management.
- Admin dashboards for decision-making and operational efficiency.

## **11. Cloud-Based Scalability**

- Ability to handle millions of users simultaneously.

- Auto-scaling during peak hours (festivals, holidays).
- Improves performance and reduces system downtime.



## Conclusion

- The Online Train Ticket Booking System successfully transforms the traditional, manual reservation process into a fast, efficient, and user-friendly digital solution. By enabling users to search trains, check seat availability, and book tickets anytime and anywhere, the system greatly enhances convenience and reduces dependency on physical ticket counters. Its automated features—such as secure payments, instant confirmations, and easy cancellations—improve accuracy, reduce human errors, and ensure a smoother travel experience.
- Furthermore, the system benefits administrators through streamlined operations, real-time data management, and reduced workload. With its scalability, strong security, and potential for future enhancements like AI-based assistance, real-time train tracking, and multi-modal integration, the platform is well-positioned to meet the evolving needs of passengers and the railway industry.
- Overall, the Online Train Ticket Booking System provides a reliable, efficient, and modern approach to train reservations, improving customer satisfaction and contributing to digital transformation in the transportation sector.

## References

- 1. Google**
- 2. “The C Programming Language”**

