S. B. JAIN INSTITUTE OF TECHNOLOGY, MANAGEMENT & RESEARCH, NAGPUR.

Project Report

Name	:	**Laxmi rani rana**
USN	:	**CS23097**
Branch	:	**CSE**
Session	:	**2023:24**
Year / Semester	:	**I/II**
Course	:	**Advance C Programming**
Course Code	:	**N-PCCCS101P**
Project Topic	:	**Railway Ticket Booking System**
Project Guide	:	**Prof. Roshani Talmale**

Department of First Year Engineering, S.B.J.I.TM.R, Nagpur

S. B. JAIN INSTITUTE OF TECHNOLOGY, MANAGEMENT AND RESEARCH, NAGPUR

(An Autonomous Institute, Affiliated to RTMNU, Nagpur)

FIRST YEAR ENGINEERING DEPARTMENT

"Emerge as a leading Institute for developing competent and creative Professionals"

Index

Serial No.	Title
1	**Introduction (ProjectDetails)**
2.	**Major Library andFunctions**
3.	**Source Code**
4.	**Result (output)**
5.	**Conclusion**
6.	**References**

Aim: Develop a simple railway ticket booking system that allows users to view available trains, book tickets for specific journeys, display ticket details, and cancel booked tickets. The system should support multiple trains, each with its schedule and intermediate stations, and handle passenger details for booking tickets. Additionally, the system should calculate fares based on the chosen coach type and distance travelled between stations. The project aims to provide a user-friendly interface for managing railway bookings efficiently.

Objectives:

1.Implement a user-friendly interface for booking railway tickets, displaying availabletrains, and managing bookings.

2.Develop functionality to calculate fares based on coach type and distance traveledbetween stations.

3.Create a robust system for storing passenger and ticket information, allowing users toview ticket details and cancel bookings**.**

Major C aspects used in the project:

Primitive Data Types:

int: Used for integer values, such as train numbers, ages, distances, etc.

char: Used for single characters, such as gender, coach types, etc.

float: Although not explicitly used in the provided code, it could be used for floating-point values if required.

Array Types:

Arrays of primitive types (char, int) are used extensively throughout the code to store strings and collections of data. For example, arrays of char are used to store names, phone numbers, station names, etc.

Structures:

struct: Used to define custom data types such as Passenger, Ticket, and Train. These structures encapsulate related data fields into a single unit.

Pointer Types:

Pointers are not explicitly used in the provided code, but arrays implicitly decay into pointers in certain contexts, such as when passing them to functions.

User-Defined Types:

Structures (struct) are user-defined types used to encapsulate related data fields.

Enumerated Types:

Enumerations are not used in the provided code, but they could be useful for defining constants representing coach types, gender options, etc., in a more readable way.

Theory:

Introduction:

Data types in programming languages play a crucial role in defining the kind of data a variable can hold, how it is stored in memory, and what operations can be performed on it. In the C programming language, various data types are used to represent different kinds of values, from simple integers to complex structures. This paper provides an overview of the data types used in C programming, their characteristics, and their applications in real-world scenarios.

Primitive Data Types:

Integer Types (int): Integers are whole numbers without any fractional part. In C, integer types are used to represent numerical values such as counts, indices, and identifiers. The int type is the most commonly used integer type in C, providing a range of values determined by the underlying hardware architecture.

Character Type (char): The char type is used to represent single characters, such as letters, digits, and special symbols. Characters are stored as ASCII values in memory, allowing operations such as comparison and manipulation.

Floating-Point Types (float, double): Floating-point types are used to represent real numbers with a fractional part. The float type provides single-precision floating-point numbers, while the double type offers double-precision numbers with higher precision and a larger range.

Derived Data Types:

Arrays: Arrays are collections of elements of the same data type arranged in contiguous memory locations. They allow efficient storage and retrieval of multiple values using a single identifier.

Structures (struct): Structures are user-defined data types that allow bundling together different data types under a single name. They enable the creation of complex data structures representing real-world entities with multiple attributes.

Pointers: Pointers are variables that store memory addresses as their values. They enable dynamic memory allocation, efficient manipulation of arrays and strings, and support for complex data structures such as linked lists and trees.

Enumerated Types (enum):

Enumerated types allow the programmer to define custom data types with a finite set of symbolic constants. They provide a more readable and expressive way of representing integer values with meaningful labels.

Algorithm:

Initialization:

   a. Declare and initialize data structures for trains, tickets, and passengers.
   b. Populate the list of available trains with details such as train numbers, names,source, destination, and intermediate stations.

User Authentication:

   a. Prompt the user to enter their username and password.
   b. Validate the entered credentials against predefined values for authentication.
   c.

Main Menu:

Display a menu with options for different functionalities.

Option 1: Book Ticket:

   1. Prompt the user to enter the train number for booking.
   2. Validate the entered train number against available trains.
   3. Display stations for the selected train, including source and destination.
   4. Prompt the user to enter the source and destination stations.
   5. Validate the entered stations against intermediate stations for the selected train.

Department of First Year Engineering, S.B.J.I.TM.R, Nagpur

Option 2:Train number:

1. 0 (10XXX or 00XXX): Konkan Railway.
2. 0 (20XXX): It is a new nomenclature used by railways to designate superfast or premium category trains.
3. 1 (Y1XXX): Central Railway trains, some West Central and North Central Railway trains.

Option 3:Train name:

1. Enter the source and destination cities.
2. Click on "Search Trains".

3. You will land on the page where all the trains running between the source and destination station are listed, along with the seat availability.

Option 3:Number of tickets:

1.Define a single number

2.Iterate through each number of tickets and convert them into words using the defined function.

3.Print the ticket count and its equivalent in words.

Option 4: Exit:

a. Terminate the program and log out the user.
b. Validation Functions:
c. Implement functions to validate user inputs, such as train numbers, station names, dates, and passenger details.

CODE:

```c
#include <stdio.h>
#include <stdbool.h>
#include <stdlib.h>
#include <string.h>

#define MAX_SEATS_FIRST_CLASS 50
#define MAX_SEATS_SECOND_CLASS 100

int available_seats_first_class = MAX_SEATS_FIRST_CLASS;
int available_seats_second_class = MAX_SEATS_SECOND_CLASS;

void book_ticket(int train_number, char *train_name, int num_tickets, char class) {

    if(class != 'F' && class != 'S') {
        printf("Invalid class selected.\n");
        return;
    }

    if (class == 'F' && num_tickets > available_seats_first_class) {
        printf("Sorry, there are only %d seats available in First Class.\n",
available_seats_first_class);
        return;
    } else if (class == 'S' && num_tickets > available_seats_second_class) {
        printf("Sorry, there are only %d seats available in Second Class.\n",
available_seats_second_class);
        return;
```

```
    }

    if (class == 'F') {
        available_seats_first_class -= num_tickets;
    } else if (class == 'S') {
        available_seats_second_class -= num_tickets;
    }

    printf("Tickets booked successfully for %d passengers in %c class for %s (Train
Number: %d).\n", num_tickets, class, train_name, train_number);
}

int main() {
    int train_number;
    char train_name[100];
    int num_tickets;
    char class;
    char option;

    printf("Welcome to the Train Ticket Booking System\n");

    printf("Enter the train number: ");
    scanf("%d", &train_number);

    printf("Enter the train name: ");
    scanf("%s", train_name);

    do
        printf("Enter the number of tickets: ");
```

```
        scanf("%d", &num_tickets);


        printf("Enter the class (F for First Class, S for Second Class): ");
        scanf(" %c", &class);


        book_ticket(train_number, train_name, num_tickets, class);


        printf("Do you want to book more tickets? (Y/N): ");
        scanf(" %c", &option);


    } while(option == 'Y' || option == 'y');


    printf("Thank you for using the Train Ticket Booking System.\n");


    return 0;
}
```
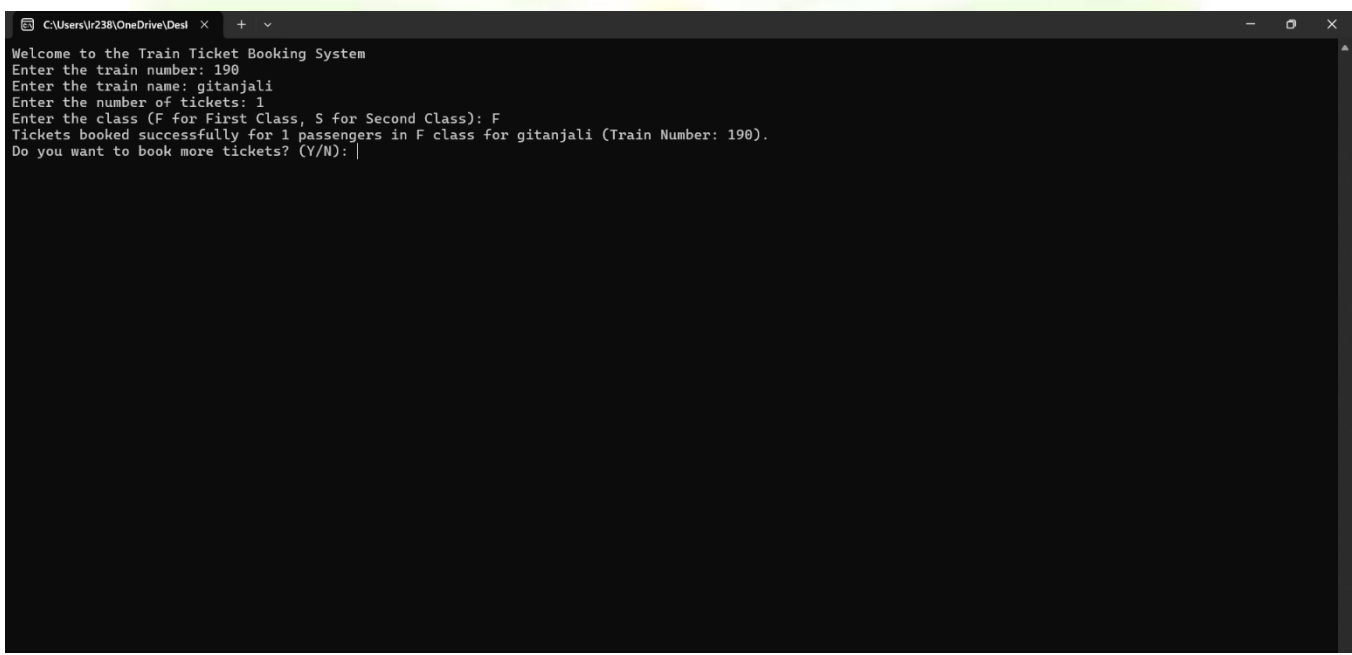
Output:

Department of First Year Engineering, S.B.J.I.TM.R, Nagpur

```
C:\Users\Ir238\OneDrive\Desl

Welcome to the Train Ticket Booking System
Enter the train number: 190
Enter the train name: gitanjali
Enter the number of tickets: 1
Enter the class (F for First Class, S for Second Class): F
Tickets booked successfully for 1 passengers in F class for gitanjali (Train Number: 190).
Do you want to book more tickets? (Y/N): Y
Enter the number of tickets: 2
Enter the class (F for First Class, S for Second Class): S
Tickets booked successfully for 2 passengers in S class for gitanjali (Train Number: 190).
Do you want to book more tickets? (Y/N):
```



```
C:\Users\Ir238\OneDrive\Desl

Welcome to the Train Ticket Booking System
Enter the train number: 190
Enter the train name: gitanjali
Enter the number of tickets: 1
Enter the class (F for First Class, S for Second Class): F
Tickets booked successfully for 1 passengers in F class for gitanjali (Train Number: 190).
Do you want to book more tickets? (Y/N): Y
Enter the number of tickets: 2
Enter the class (F for First Class, S for Second Class): S
Tickets booked successfully for 2 passengers in S class for gitanjali (Train Number: 190).
Do you want to book more tickets? (Y/N): N
Thank you for using the Train Ticket Booking System.

--------------------------------
Process exited after 41.11 seconds with return value 0
Press any key to continue . . . |
```

Conclusion:

This railway ticket booking system enables users to securely log in, browse available trains, book tickets with fare calculation, view ticket details, and cancel reservations. With a clear interface and essential features like authentication and booking management, it offers a streamlined user experience. However, further enhancements such as payment integration and seat availability checks could improve its functionality and convenience.

Overall, it provides a reliable solution for managing train travel arrangements efficiently.

References:

https://www.geeksforgeeks.org/online-railway-ticket-reservation-system/

C Programming Language" by Brian W. Kernighan and Dennis M. Ritchie