Final Project

Object Oriented Programming

Teacher: Sir Muhammad Abbas

Railway Booking System using object oriented programming in C++

The program is a Railway Booking System that allows users to select a train, choose their seats, and make a reservation.

It provides different classes of seats: First Class, Business Class, and Economy Class.

The program generates a unique token for each reservation

Steps:

1. The program displays a menu where users can select a train or exit.
2. After selecting a train, users enter their details such as name, departure, and destination stations.
3. The program presents a seating plan for the selected train and class.
4. Users choose a seat by specifying the row and seat number.
5. The program checks if the seat is available and reserves it if it's not occupied.
6. Users can reserve multiple seats if desired.
7. Once the seat selection is complete, the program displays the passenger's details, including the train name, name, from, to, token, seat type, row numbers, and seat numbers.

Benefits:

* Provides a user-friendly interface for booking train seats.
* Allows users to choose their preferred train, seat type, and seat location.
* Generates unique tokens for each reservation, ensuring data security.
* Supports multiple seat reservations in a single booking.
* Provides clear information about the passenger's booking details.

Challenges:

* Validating user input to ensure correct seat and train selections.
* Handling seat availability and preventing double booking.
* Managing the seating plan and updating seat availability after each reservation.
* Implementing a random token generation algorithm.
* Handling user interactions and displaying relevant information effectively.

Things used in this program:

* C++ programming language.
* Libraries: iostream, random, sstream, string, vector, ctime, fstream.
* Struct: BookingDetails to store passenger details.
* Class: Train to manage train-related operations.
* Random number generation for token generation.
* File handling using fstream to write program details to a text file.
* Console output for displaying menus and information to the user.
* User input handling and validation.

Source code of project

### Railway Booking System using c++

### 

**The output of this program is as follows:**

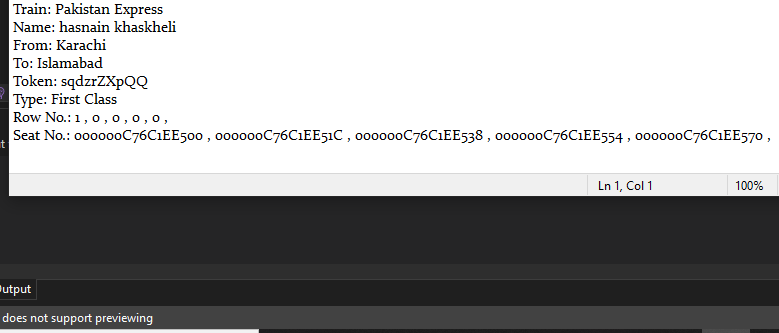
1. The Program Introduction:
   * The program is a Railway Booking System.
   * It allows users to select a train, choose their seats, and make a reservation.
   * Different classes of seats are available: First Class, Business Class, and Economy Class.
   * Each reservation is assigned a unique token for identification.
2. User Interaction:
   * Users are greeted with a welcome message and presented with a menu.
   * They can choose between selecting a train or exiting the program.
   * After selecting a train, users are prompted to enter their personal details, such as their name, departure station, and destination.
3. Seat Selection:
   * Users are shown a seating plan for the selected train and class.
   * The program prompts users to choose a seat by specifying the row and seat number.
   * It checks if the selected seat is available and reserves it if it's not already occupied.
   * Users can continue to reserve additional seats if desired.
4. Reservation Details:
   * Once seat selection is complete, the program displays the passenger's reservation details.
   * This includes the train name, passenger's name, departure and destination stations, and the assigned token.
   * The program also indicates the selected seat type (First Class, Business Class, or Economy Class), as well as the row and seat numbers.
5. Program Termination:
   * After displaying the reservation details, the program thanks the user for using the service.
   * It prompts the user to press any key to continue, clearing the screen for the next user or menu display.
6. File Output:
   * The program writes the program details, including the train name, passenger's name, departure and destination stations, token, seat type, row numbers, and seat numbers, to a text file named "program\_details.txt".

Overall, the program provides a user-friendly interface, guiding users through train selection, seat reservation, and providing clear information about their booking details.

# View

# 

1. File Output in data base

**As for flow chart**

****

# Proven and function of project

|  |
| --- |
| The provided code is an example of a Railway Booking System written in C++ programming language. However, it is not a proven and functional system as it is a simplified version for demonstration purposes. To make it a functional program, it would require further development and thorough testing.  Here are some considerations to keep in mind:  1. Input Validation: The code should include robust input validation to handle edge cases and prevent unexpected behavior or crashes. For example, it should validate user inputs for seat selection, train choices, and range checks.  2. Error Handling: The program should have proper error handling mechanisms in place to gracefully handle exceptional situations, such as invalid inputs, file I/O errors, or seat availability issues.  3. Data Persistence: In a real-world application, the program would need to store and retrieve data from a database or file system. The current code writes the program details to a text file, but additional functionality for data persistence and retrieval would be required.  4. Authentication and Security: A practical railway booking system would incorporate user authentication mechanisms and security measures to protect sensitive passenger information and prevent unauthorized access.  5. \Performance and Scalability: As the system scales with a larger number of trains, passengers, and reservations, the code should be optimized for performance and efficiency to handle the increased load effectively.  User Interface: While the current code provides a basic command-line interface, a production-ready system would likely include a more user-friendly and intuitive graphical user interface (GUI) to enhance the user experience.  It's important to note that the provided code serves as a starting point and should be further developed, tested, and validated to create a fully functional and secure Railway Booking System. It is crucial to follow industry best practices, conduct comprehensive testing, and adhere to programming and security standards before deploying the system in a production environment. |

## Function of the project

|  |
| --- |
| Railway Booking System includes the following functions:  Train::Train(const string& trainName, int trainCapacity): Constructor for initializing the train object.  Train::getName() const: Getter function to retrieve the train's name.  Train::getCap() const: Getter function to retrieve the train's capacity.  Train::enterDetails(): Function for entering passenger details and returning a BookingDetails object.  Train::generateToken(): Function to generate a unique token for each reservation.  Train::displayPassengerDetails(): Function to display the passenger's reservation details.  Train::displayMenu(): Function to display the main menu.  Train::selectTrain(): Function to handle train selection and invoke other functions accordingly.  Train::settingPlan(): Function to manage seat selection, reservation, and seating plan display.  main(): The main function that initializes a Train object and starts the program.  Additional functionality such as input validation, error handling, data persistence, security measures, and user interface improvements would be required to make it a fully functional and secure railway booking system. |