**Project Proposal: Railway Reservation System**

Proposed By: Ashim Acharya, Nitesh Shahi and Abiral

**Course**: C Programming

Institution: Kings Collage

Date: 29 May, 2025

**Project Proposal: Railway Reservation System**

**1.Introduction**

The Railway Reservation System is a software application designed to automate the process of booking and managing train seats. It allows users to check seat availability, book and cancel tickets, and view a passenger list in real-time. This project aims to simulate a real-world railway ticketing environment using the C programming language.

**2. Objective**

**The primary objective of this project is to:**

Develop a menu-driven system for railway reservation

Enable users to book and cancel train tickets

Maintain a passenger list

Display real-time seat availability

Understand the practical use of arrays, structures, and file handling in C

**3. Scope of the Project**

**This project covers the following features:**

A single train with a fixed number of seats (e.g., 100 seats)

Booking a ticket by entering passenger name and seat number

Cancelling a ticket using seat number

Viewing available seats

Viewing the list of booked passengers

The system can be expanded in the future to support multiple trains, classes (e.g., AC, sleeper), and integration with databases.

**4. Tools and Technology**

Programming Language: C Programming

Compiler: GCC

Platform: Windows (Console Application)

Development Environment: Code Blocks

**5. Project Planning Based on Time & Resources**

| **Factor** | **Decision Made** |
| --- | --- |
| Time | Target completion in 1 week (2–3 hours per day) |
| Team Size | 3 people |
| Resources | C language, Code: Blocks IDE, no external libraries |
| UI | Console only |
| Storage | In-memory (no file system or DB to keep it simple) |
| Features Included | View trains, check availability, book ticket, cancel ticket |
|  |  |

**Important Code Element Used to Make Railway Reservation System:**



**Testing and Validation**

To make sure the Railway Reservation System works accurately and reliably, we will apply the following testing methods:

**Unit Testing:** Test individual functions such as seat availability check, booking, and cancellation separately to verify they perform as expected.

**Integration Testing:** Combine related modules after unit testing to ensure they work together properly. for example, booking a ticket should update seat availability correctly.

**System Testing:** Perform end-to-end testing of the entire system, including searching for trains, checking seats, booking tickets, and confirming bookings.

**Validation:** Validate all inputs to ensure users provide correct information (like valid train numbers and positive seat counts), and verify outputs for correctness, such as updated seat counts and booking confirmations.

**Performance Testing:** Test the system’s response and stability under heavy use by simulating multiple simultaneous bookings.

**5. System Design**

**Data Structures:** Arrays and structures will be used to store seat and passenger information.

**Input/Output:** Console-based user interface for menu navigation and input prompts.

**Modules**:

Seat initialization

Booking module

Cancellation module

Passenger list module

Main menu driver

**User Interface (UI) for Railway Reservation System in C**

Use a Command-Line Interface (CLI) because C doesn’t support GUI natively and CLI is simple to build with printf and scanf.

Display a text menu with options like View Trains, Check Seat Availability, Book Ticket, Cancel Ticket, and Exit.

Use printf to show menus and instructions; use scanf or fgets to get user input.

Validate inputs to handle invalid choices or wrong data.

After each action, show clear confirmation or error messages to guide the user.

Keep the program running in a loop until the user chooses to exit.

Format output (like train lists and booking details) clearly for easy reading.

**6. Challenges**

Keeping track of which seats are booked and which are available.

Handling all this without a database, which requires efficient use of C structures and memory.

Users entering invalid seat numbers (e.g., 0 or 105 when only 100 seats exist).

Input strings (like names) being too long or containing special characters.

Preventing the system from crashing due to incorrect input types (e.g., letters when numbers are expected).

Trying to cancel a ticket for a seat that hasn’t been booked.

Unexpected user actions or inputs.

**7. Budget**

Since this project is a small-scale academic or prototype system developed using open-source tools and existing hardware, the budget is minimal. Here's a basic estimate:

|  |  |
| --- | --- |
| Item | Cost in Nepalese Rupees |
| Computer (already available) | 0 |
| C Compiler (GCC / Turbo C) | 0 |
| |  | | --- | | Code Editor (VS Code / Code Blocks) |  |  | | --- | |  | | 0 |
| |  | | --- | | Internet (for research & testing) |  |  | | --- | |  | | 0 |
| Printing and Documentation | 50 |
| Time Cost | 3000 |
| Total Cost | RS 3050 |

**8. Conclusion**

This project offers practical experience in C programming, focusing on file operations, structured programming, and real-world system simulation. It will enhance problem-solving skills and lay the groundwork for more complex system development in the future.