



**Marwadi**  
University  
Marwadi Chandarana Group



FACULTY OF  
**COMPUTER**  
**APPLICATIONS**

**A Python Project for Academic Year 2024 - 2025**

# **Railway Management System Project Report**

Subject Code: 05BC3404

Subject Name: Python Programing

Submitted By: Neel Gardhariya  
[92300527211]

Submitted To:  
Prof. Jitesh Solanki

## **Introduction**

**The Railway Management System is a Python application aimed at simplifying railway booking management. It offers a streamlined way to create, view, update, delete, and filter bookings through a menu-driven console interface. Using CSV file handling, this system enables lightweight, database-independent functionality. By emphasizing robust data validation, structured workflows, and a user-friendly approach, the system ensures reliability and efficiency.**

## **Technical Information**

### **Technologies Used:**

#### **Technical Information**

**Programming Language: Python 3.x**

**Data Storage: CSV File**

**Libraries: csv for reading and writing data re for input validation**

**(e.g., mobile numbers and email addresses) datetime for date validation and comparison**

**Pretty Table for displaying tabular data in a readable format**

**shutil for backup and restore operations Development**

#### **Tools:**

**Code Editors: Visual Studio Code, PyCharm**

**Execution Environment: Command Line Interface (CLI)**

#### **System Requirements:**

**Software: Python 3.x**

**Hardware: Any standard system with at least 512MB RAM and minimal storage**

## **Railway Management System Requirements and Project Structure**

### **System Requirements Software:**

- Python 3.x (recommended)
- Any text editor or Python IDE, such as Visual Studio Code or PyCharm, for code modifications
- Command Line Interface (CLI) or Terminal for executing the program

### **Hardware:**

- Any system capable of running Python efficiently
- At least 512MB RAM (minimal requirement)
- Minimal disk storage for CSV files

### **Project Structure Main Components:**

#### **1. Core Script:**

- railway\_management.py – Contains all functionalities, including booking management, data handling, validation, and backup operations.

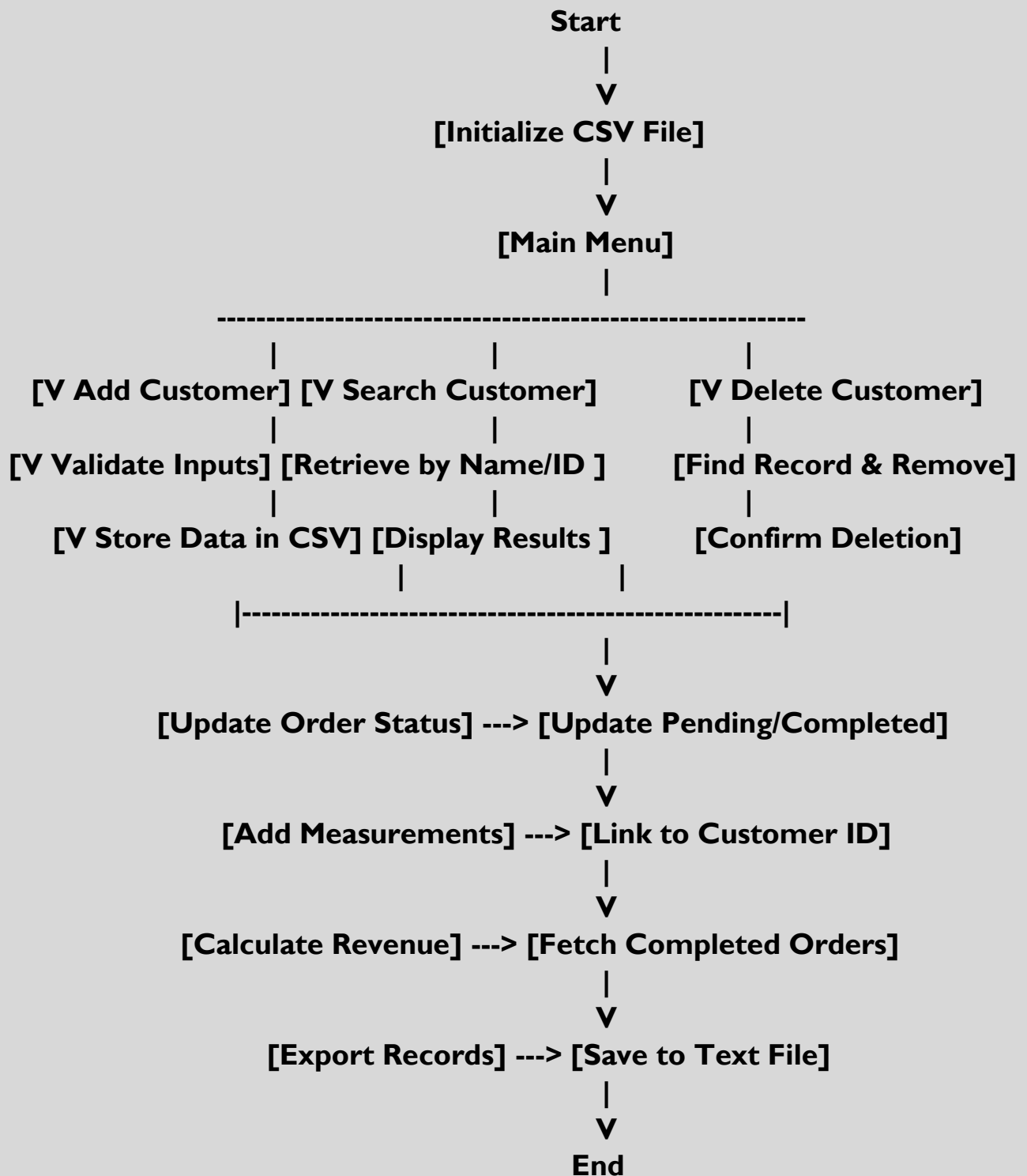
#### **2. Data Storage:**

- railway\_bookings.csv – Acts as the primary database to store booking records persistently.

#### **3. Log File:**

- activity\_log.txt – Tracks all system activities, such as bookings, edits, and deletions.

## **Diagram(s)**



[Flow chart of Railway Management System]

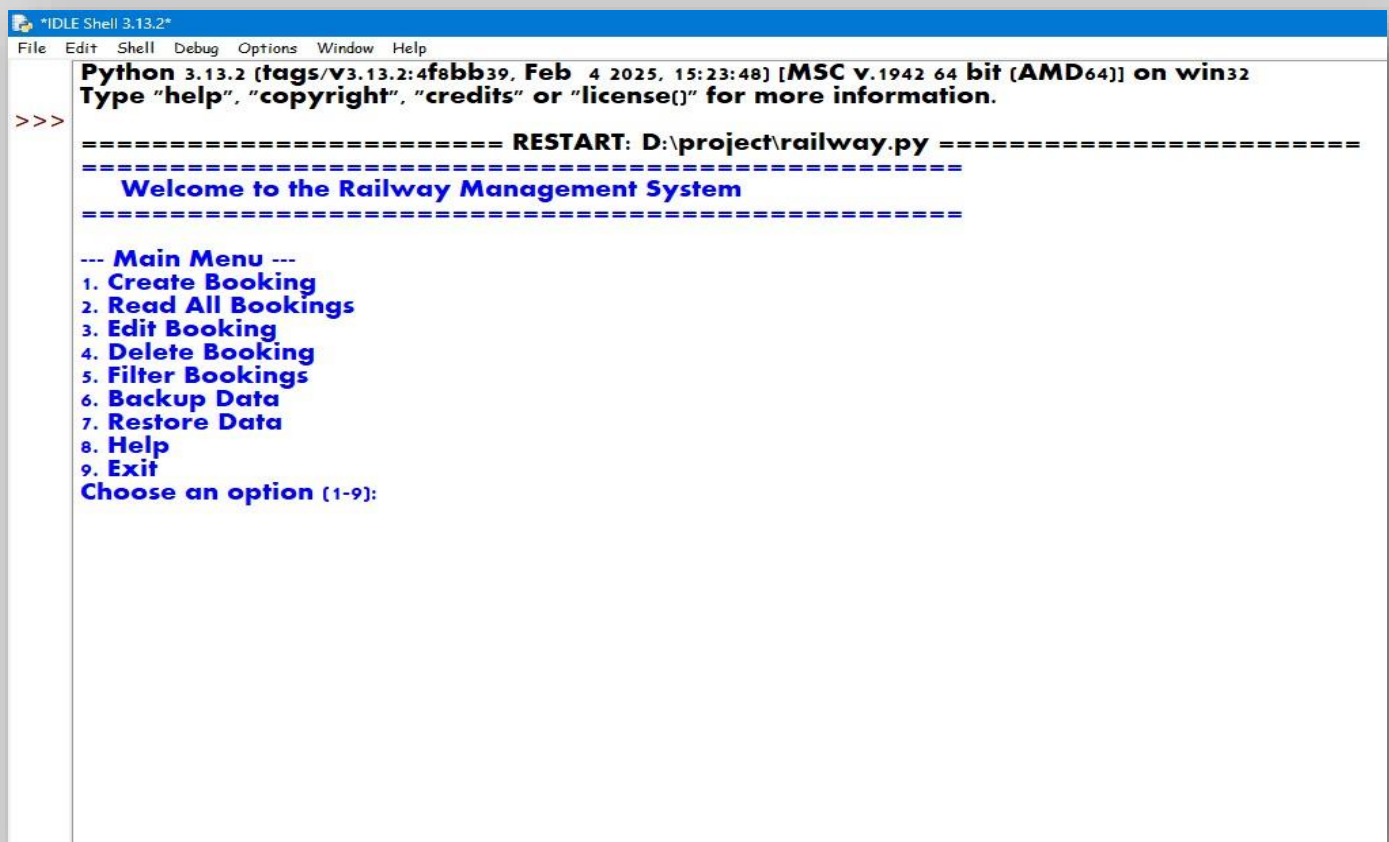
## Features

1. Add, view, update, and delete bookings.
2. Validate key inputs such as mobile numbers, email addresses, and travel dates.

3. Filter bookings by travel date, station, or class.
4. Display records in tabular format for better readability.
5. Log user activities (e.g., booking creation and editing).
6. Perform backup and restore operations for data security.

## Output Screenshots

Below are the screenshots of the system's functionality:



```
*IDLE Shell 3.13.2*
File Edit Shell Debug Options Window Help
Python 3.13.2 (tags/v3.13.2:4f8bb39, Feb  4 2025, 15:23:48) [MSC v.1942 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\project\railway.py =====
=====
Welcome to the Railway Management System
=====

--- Main Menu ---
1. Create Booking
2. Read All Bookings
3. Edit Booking
4. Delete Booking
5. Filter Bookings
6. Backup Data
7. Restore Data
8. Help
9. Exit
Choose an option [1-9]:
```

## MAIN MENU

### Adding a booking

```
Enter Mobile Number (10 digits): 9714900044
Enter Name: divyesh
Enter Departure Station: rajkot
Enter Arrival Station: surat
Enter Train Type (Express/Passenger) [Default: Express]: express
Enter Class (Sleeper/AC/General) [Default: AC]: AC
Enter Email: divyesh18@gamil.com
Enter Meal Preference (Veg/Non-Veg/Vegan) [Default: Veg]: Veg
Enter Travel Date (dd/mm/yyyy): 20/04/2025
Booking created successfully!
```

### Read all Bookings

#### ===== All Bookings =====

--- Booking 1 ---

```
Mobile Number : 6352133705
Name          : neel patel
Departure Station : rajkot
Arrival Station  : surat
Train Type      : Express
Class           : AC
Email           : nikpatel2162@gamil.com
Meal Preference  : Veg
Travel Date     : 21/04/2025
=====
```

## Edit booking

```
--- Main Menu ---
1. Create Booking
2. Read All Bookings
3. Edit Booking
4. Delete Booking
5. Filter Bookings
6. Backup Data
7. Restore Data
8. Help
9. Exit
Choose an option (1-9): 3
Enter Mobile Number of the booking to edit: 9714900044
Current Booking Details:
['9714900044', 'divyesh', 'rajkot', 'surat', 'express', 'AC', 'divyesh18@gamil.com', 'Veg', '20/04/2025']
Enter new details (press Enter to keep current value):
Name [divyesh]:
Departure Station [rajkot]:
Arrival Station [surat]:
Train Type [express]:
Class [AC]:
Email [divyesh18@gamil.com]:
Meal Preference [Veg]:
Travel Date [20/04/2025]: 21/04/2025
Booking updated.
```

## Delete Booking

```
2. Read All Bookings
3. Edit Booking
4. Delete Booking
5. Filter Bookings
6. Backup Data
7. Restore Data
8. Help
9. Exit
Choose an option (1-9): 4
Enter Mobile Number of booking to delete: 9714900044
Booking deleted.
```

## Filter Booking



```
--- Main Menu ---
1. Create Booking
2. Read All Bookings
3. Edit Booking
4. Delete Booking
5. Filter Bookings
6. Backup Data
7. Restore Data
8. Help
9. Exit
Choose an option (1-9): 5
Filter by:
1. Travel Date
2. Departure Station
3. Train Class
Choose filter option: 2
Enter filter value: rajkot
No bookings matched the filter.
```

## Backup Data

```
--- Main Menu ---
1. Create Booking
2. Read All Bookings
3. Edit Booking
4. Delete Booking
5. Filter Bookings
6. Backup Data
7. Restore Data
8. Help
9. Exit
Choose an option (1-9): 6
Backup successful!
```

## Restore Data

```
--- Main Menu ---
1. Create Booking
2. Read All Bookings
3. Edit Booking
4. Delete Booking
5. Filter Bookings
6. Backup Data
7. Restore Data
8. Help
9. Exit
Choose an option (1-9): 7
Data restored from backup!
```

## Price Calculation and Payment

```
--- Main Menu ---
1. Create Booking
2. Read All Bookings
3. Edit Booking
4. Delete Booking
5. Filter Bookings
6. Backup Data
7. Restore Data
8. View Statistics
9. Price Calculation and Payment
10. Exit
Choose an option (1-10): 9

--- Price Calculation and Payment ---
Booking 1: Sleeper Class - ₹500

Total Price: ₹500
Proceed to payment? (yes/no): yes
Payment successful!
```

## View Pending Orders

**--- Main Menu ---**

- 1. Create Booking**
- 2. Read All Bookings**
- 3. Edit Booking**
- 4. Delete Booking**
- 5. Filter Bookings**
- 6. Backup Data**
- 7. Restore Data**
- 8. Help**
- 9. Exit**

**Choose an option [1-9]: 9**

**Exiting system. Goodbye!**

## Learning Objectives

This project develops a practical understanding of key concepts and skills, including:

1. **File Handling with Python:** Learn to manage structured data using CSV files for persistent storage, eliminating the need for databases.
2. **Input Validation:** Implement robust techniques to validate user inputs, such as phone numbers, email addresses, and travel dates, ensuring data integrity.
3. **Data Operations (CRUD):** Gain hands-on experience with creating, reading, updating, and deleting (CRUD) booking records effectively.
4. **Error Handling and Debugging:** Strengthen your ability to identify and resolve errors through structured exception handling.
5. **External Libraries for Data Presentation:** Use tools like PrettyTable to enhance data visualization and improve system usability.
6. **Date and Time Management:** Apply the datetime module to handle travel dates, timestamps, and other time-related operations seamlessly.
7. **Business Logic Implementation:** Design and implement logic for calculating ticket prices, processing payments, and filtering bookings based on user-defined criteria.
8. **Data Backup and Restoration:** Develop mechanisms for creating backups of booking data and restoring it to ensure data safety.

9. **Code Modularity and Maintainability:** Write clean, modular code with reusable functions to facilitate long-term scalability and ease of maintenance.

## **Conclusion**

- The Railway Management System is a lightweight and efficient tool for managing booking records in a railway domain. It eliminates the need for complex database systems while ensuring seamless data retrieval, filtering, and payment processing. By utilizing Python's built-in capabilities and structured file handling, the system provides a practical and effective solution for railway management.
- Developing this project has provided valuable insights into Python's file handling capabilities, user input validation, and the importance of structured data management. The system demonstrates how modular programming and robust error handling can enhance functionality while maintaining simplicity. Future enhancements could include integrating a graphical user interface (GUI) for improved usability and transitioning from CSV file handling to a database system for better scalability and performance.