

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

**AIRLINES**

**RESERVATION**

Subject Code: 05BC3404

Subject Name: Python Programing

Submitted By: Hasti Trambadiya

[92300527029]

Nandni Korat

[92300527028]

Jay Tank

[9230052126]

Submitted To:

Prof. Kinjal Raval

**Introduction**

* This project is a **console-based Airline Reservation System** developed in Python, designed to handle the core functionalities of passenger booking and record management. It uses Python’s built-in csv module for file operations, storing data persistently in a CSV file named Airlines\_Reservation.csv.

The system supports essential features for managing airline reservations, including:

* **Adding New Passengers**: Records passenger details such as name, email, age, travel route, date, gender, nationality, selected airline, and class.
* **Displaying Records**: Lists all stored passenger records in a readable format.
* **Searching**: Supports search functionality by both Receipt\_No and passenger name.
* **Updating Records**: Allows updating an existing passenger’s details.
* **Deleting Records**: Deletes a passenger's record based on their receipt number.
* **Counting**: Displays total passengers and aggregates passenger count per airline.
* **Clearing Records**: Provides an option to wipe all data, resetting the CSV with only headers.
* **Persistent Storage**: All passenger data is saved and managed through a CSV file, allowing the data to remain even after the program is closed.

**Technical Information**

**Technologies Used**

* **Programming Language: Python 3.x**
* **Data Storage:CSV (Comma-Separated Values) and TXT FILE**
* **Libraries Used:**
  + **csv – For reading and writing data to CSV files.**
  + **os – For handling file operations.**

**Development Tools**

* **Code Editor:**
* **Visual Studio Code (VS Code)**
* **PyCharm**
* **Any other Python-supported IDE**
* **Version Control System:**
* **Git (optional, for tracking changes)**

**System Requirements**

* **Software:**
* **Python 3.x (recommended)**
* **Any text editor or IDE for code modifications**
* **CLI for executing the program**
* **Hardware:**
* **Any system capable of running Python**
* **At least 512MB RAM (minimal requirement)**
* **Minimal disk storage for CSV files**

**Project Structure**

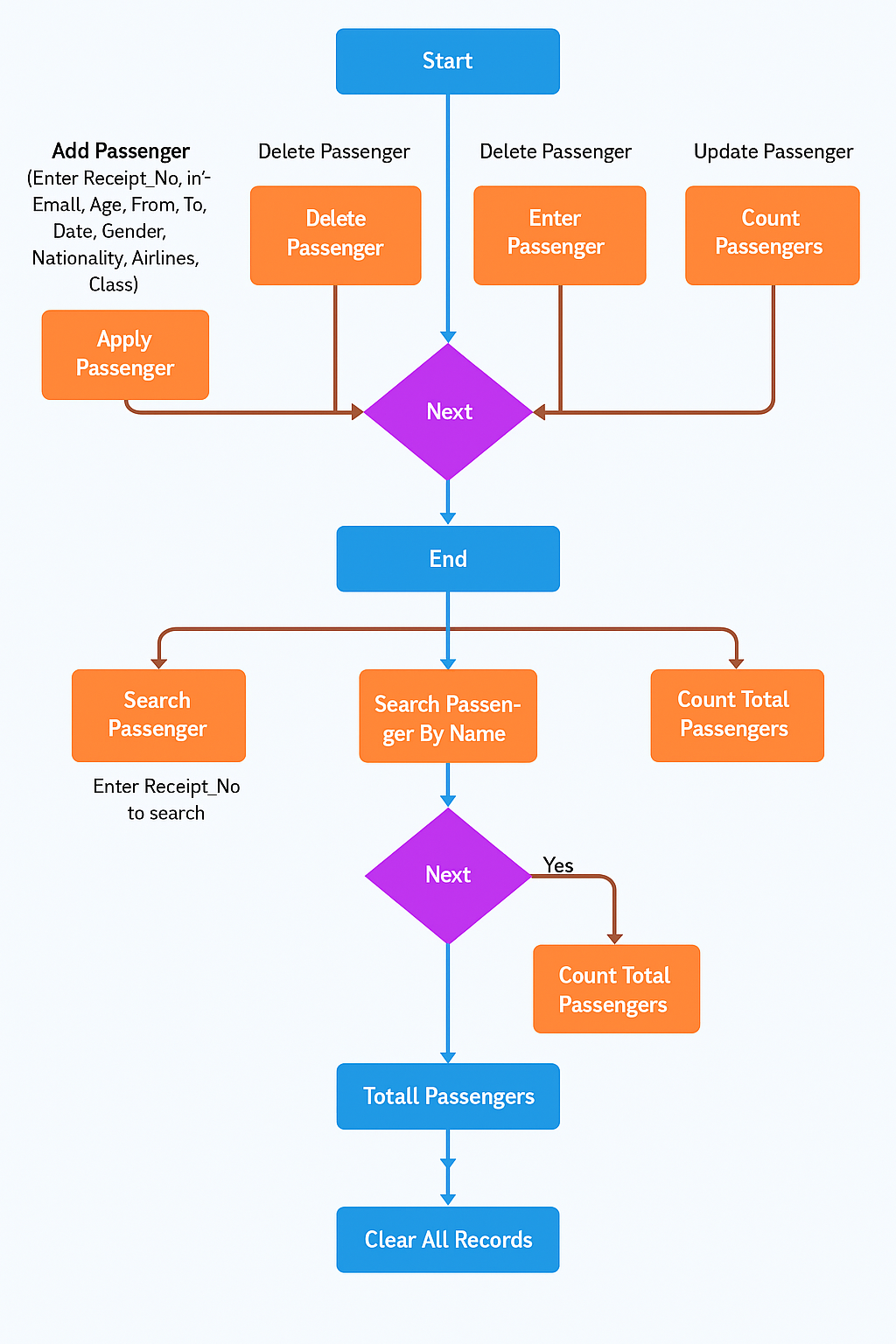
* The **airlines reservation.py** file contains all the core functionalities, including passenger management, reservation tracking, and data handling. The **Airlines\_Reservation.csv** file acts as the primary database, storing all customer and order details persistently.

Airlines\_Resevation/

│── airlines reservation.py # Main script containing all functions

│── Airlines\_Reservation.csv # Data storage file (created automatically if missing)

**Diagram(s)**

****

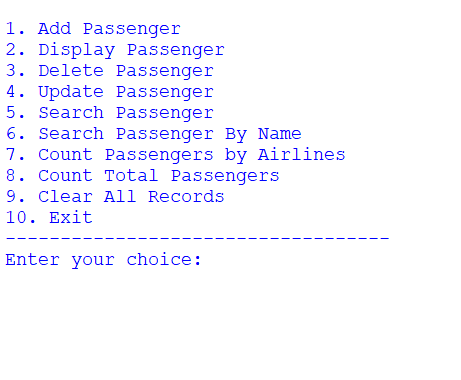
[Flow chart of Airlines Reservation]

**Features**

* **Add, view, update, delete passenger details.**
* **Search passenger by name or receipt no.**
* **Count passenger by Airlines.**
* **Count total number of passengers.**
* **Clear all records.**

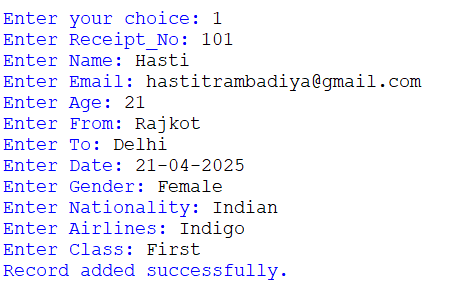
**Output Screenshots**

**Below are the screenshots of the system's functionality:**

****

**MAIN MENU**

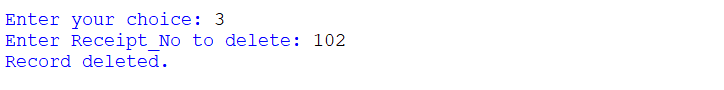
* **Adding a Passengers:**

****

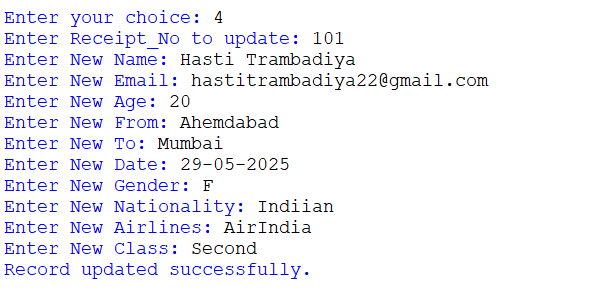
* **Display Passengers:**



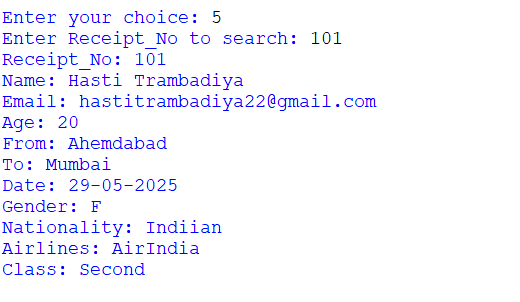
* **Delete Passenger**

****

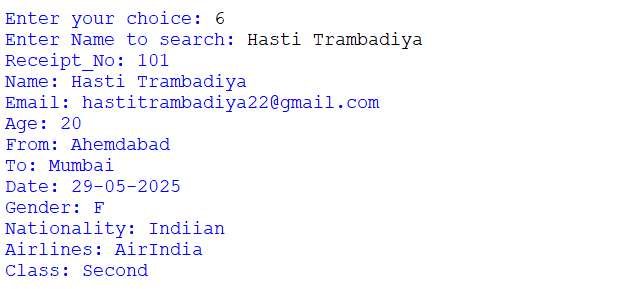
* **Update Passenger:**

****

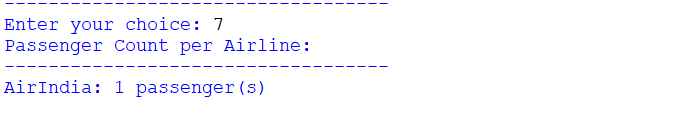
* **Search Passengers**



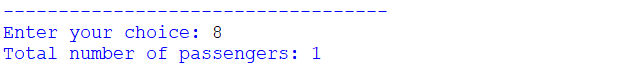
* **Search Passengers by Name:**

****

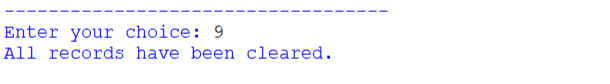
* **Count Passengers by Airlines:**

****

* **Count Total Passengers:**



* **Clear all records:**

****

**Learning Objects**

This project helps in understanding and developing essential programming skills:

1. File Handling in Python:  
   Manage structured passenger data with CSV files (no external database required).
2. Input Validation Techniques:  
   Ensure data integrity by validating email, age, and other user inputs.
3. CRUD Operations:  
   Perform Create, Read, Update, and Delete operations on reservation records.
4. Menu-Driven Program Design:  
   Create an interactive console-based user experience with menu navigation.
5. Error Handling and Debugging:  
   Apply try-except blocks to catch and handle runtime errors effectively.
6. Use of External Libraries:  
   Implement third-party libraries like tabulate for improved output formatting (optional).
7. Date and Time Management:  
   Use Python’s datetime module to manage and validate dates (planned enhancement).
8. Business Logic Implementation:  
   Track bookings, filter data, and count passengers per airline for insights.
9. Data Exporting and Formatting:  
   Support for data backups and human-readable text file outputs (planned enhancement).
10. Code Modularity and Maintainability:  
     Write reusable, organized functions to support scalability and easier debugging.

**Conclusion**

* The **Airline Reservation** is a simple yet effective solution designed to manage passenger records without relying on complex databases. Utilizing Python’s file handling capabilities and a structured CSV format, the system enables essential functionalities such as adding, updating, deleting, searching, and summarizing passenger data efficiently.
* Throughout the development of this project, we gained practical experience in file operations, menu-driven programming, and data validation. The system emphasizes the importance of structured data management and highlights how lightweight tools can still fulfill critical business needs.
* Looking ahead, the system can be further enhanced by incorporating a graphical user interface (GUI) to improve usability and accessibility. Additionally, migrating from CSV-based storage to a relational database system like SQLite or MySQL would significantly improve scalability, data integrity, and performance—making the system suitable for larger-scale deployment.