**A Python Project for Academic Year 2025 - 2026**

Airline Booking SYSTEM

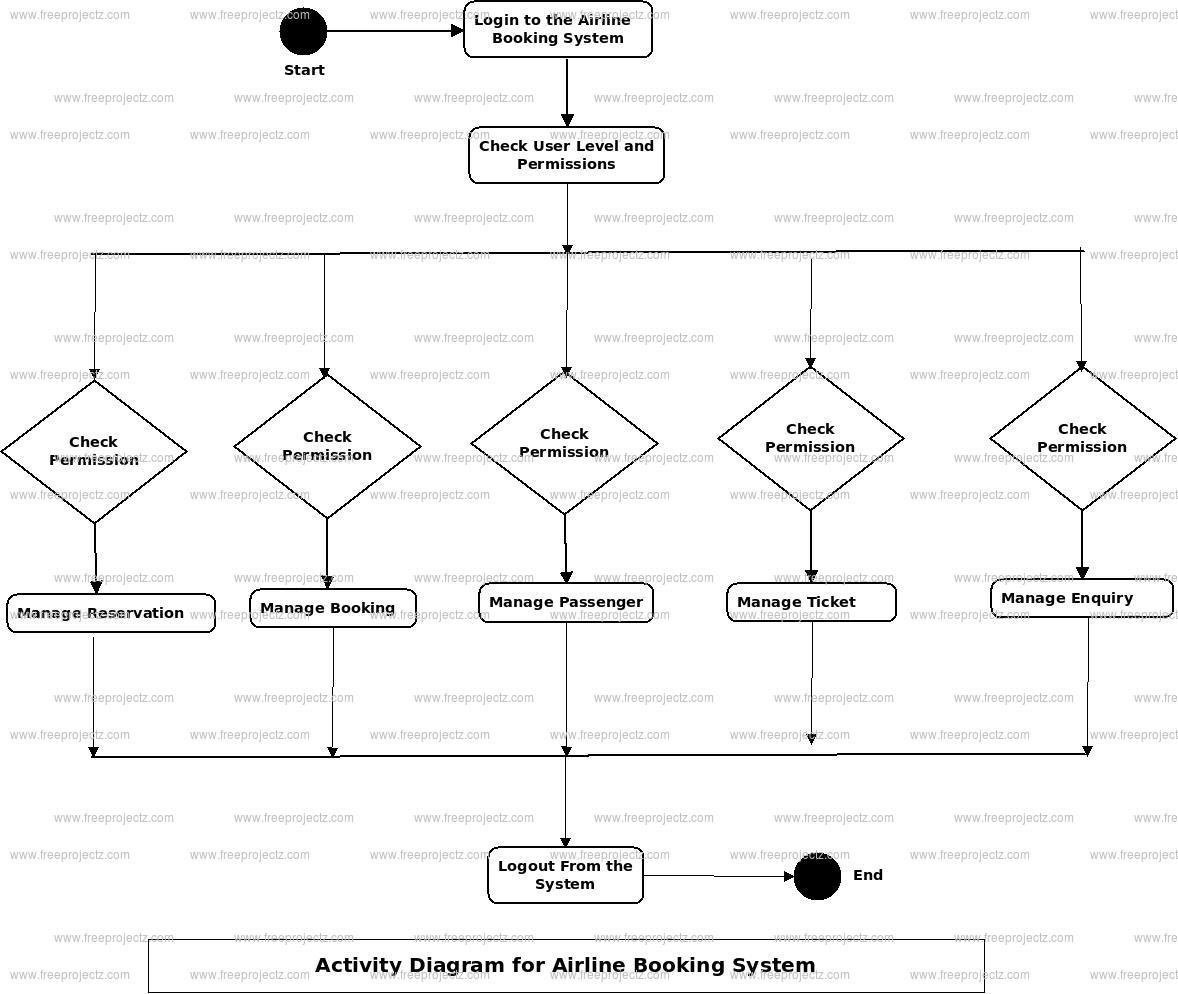
Subject Code : 05BC3404

Subject Name : Python Programing

SubmittedBy :Vishal parekh [92300527233]

SubmittedTo: Prof. Jitesh Solanki

Coding Diagram



Code Explaination

**1. CSV File Setup**

* A CSV file (booking\_data.csv) is used as the **database** to store all booking records.
* If the file doesn't exist, it initializes the file with headers like Name, Booking ID, etc.

🎨 **2. Styling and Layout**

* Global styling uses constants for background color, button style, and font.
* Styling is applied uniformly across widgets using setStyleSheet().

👤 **3. Login Page (LoginPage)**

* A simple login form with username and password fields.
* If correct credentials (vishal and 2318) are entered, the user is allowed into the admin dashboard.
* Otherwise, an error message is shown.

🛠️ **4. Admin Dashboard (AdminPage)**

* Displays three buttons:
  + **Create Booking**: Goes to the booking form.
  + **Manage Bookings**: Goes to the booking manager.
  + **Logout**: Returns to login page.

📄 **5. Create Booking Page (CreateBooking)**

* Form with input fields for all required booking details (Name, ID, Class, etc.).
* On submission:
  + Data is validated (no empty fields).
  + Data is saved into the booking\_data.csv.
  + Confirmation message is shown.

🔍 **6. Manage Bookings Page (ManageBookings)** Includes the following features:

* **Search by Booking ID**: Filters and displays matching row.
* **View All**: Loads and displays all data from CSV.
* **Update**:
  + User edits a selected row directly in the table.
  + Saves the updated data back to the CSV.
* **Delete**:
  + Deletes selected booking after confirmation.
* **Sort by Cost**:
  + Sorts all records by the "Cost" column (as float).
* **Export**:
  + Backs up all booking records into a new file backup\_bookings.csv.

□ **7. Stack-Based Navigation**

* QStackedWidget is used to switch between pages like:
  + Login → Admin → Create/Manage
  + This avoids opening multiple windows and gives a **single-window app experience**.

🚀 **8. Main Execution**

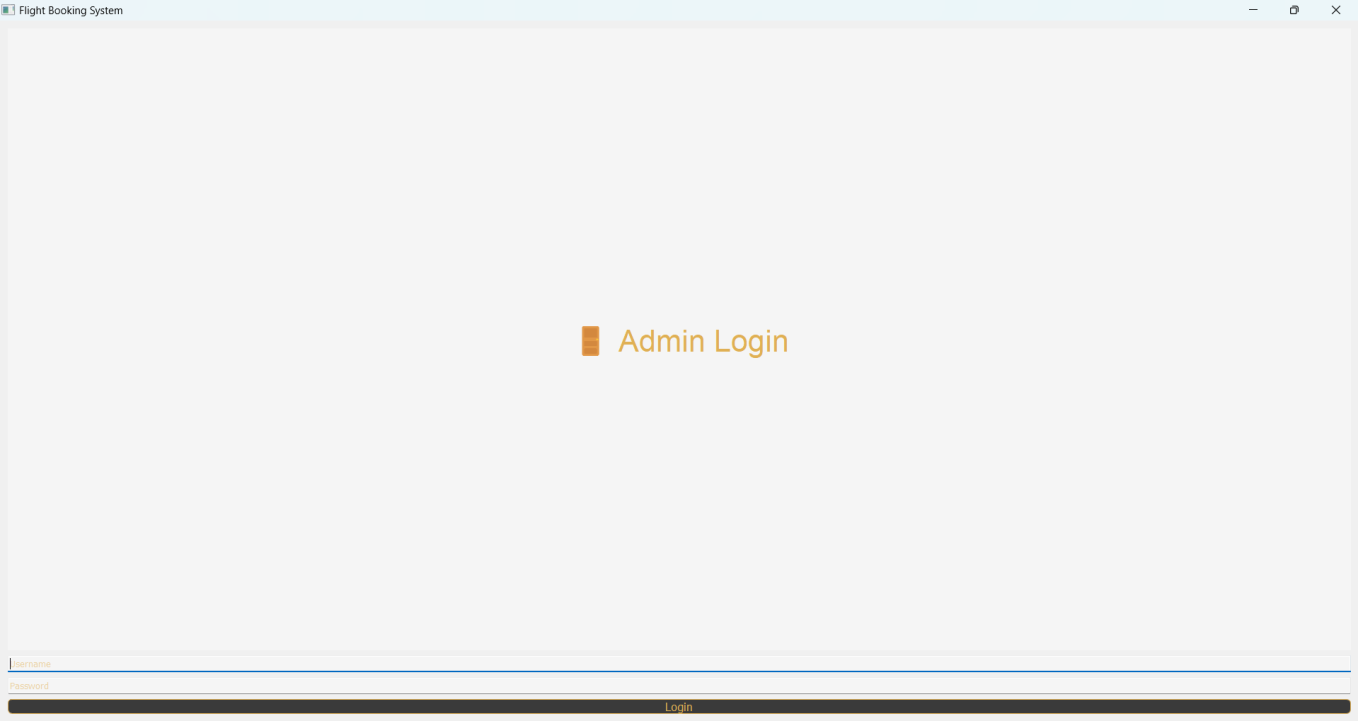
* Initializes the app, adds all widgets to the stack, shows the

main window, and starts the app loop.

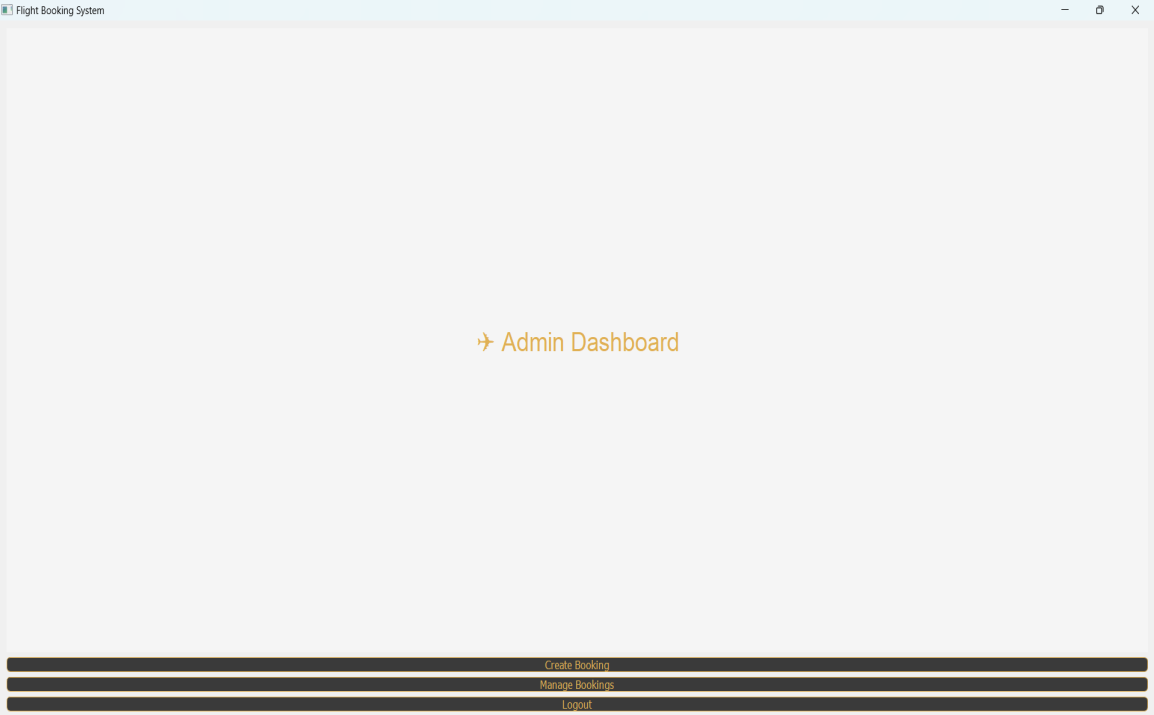
✅ **Summary of Functionalities:**

* **Admin Login**
* **Booking Creation**
* **Data Validation**
* **Search/View/Update/Delete Bookings**
* **Sort by Cost**
* **Export to CSV**
* **Stacked page navigation in a single GUI window**

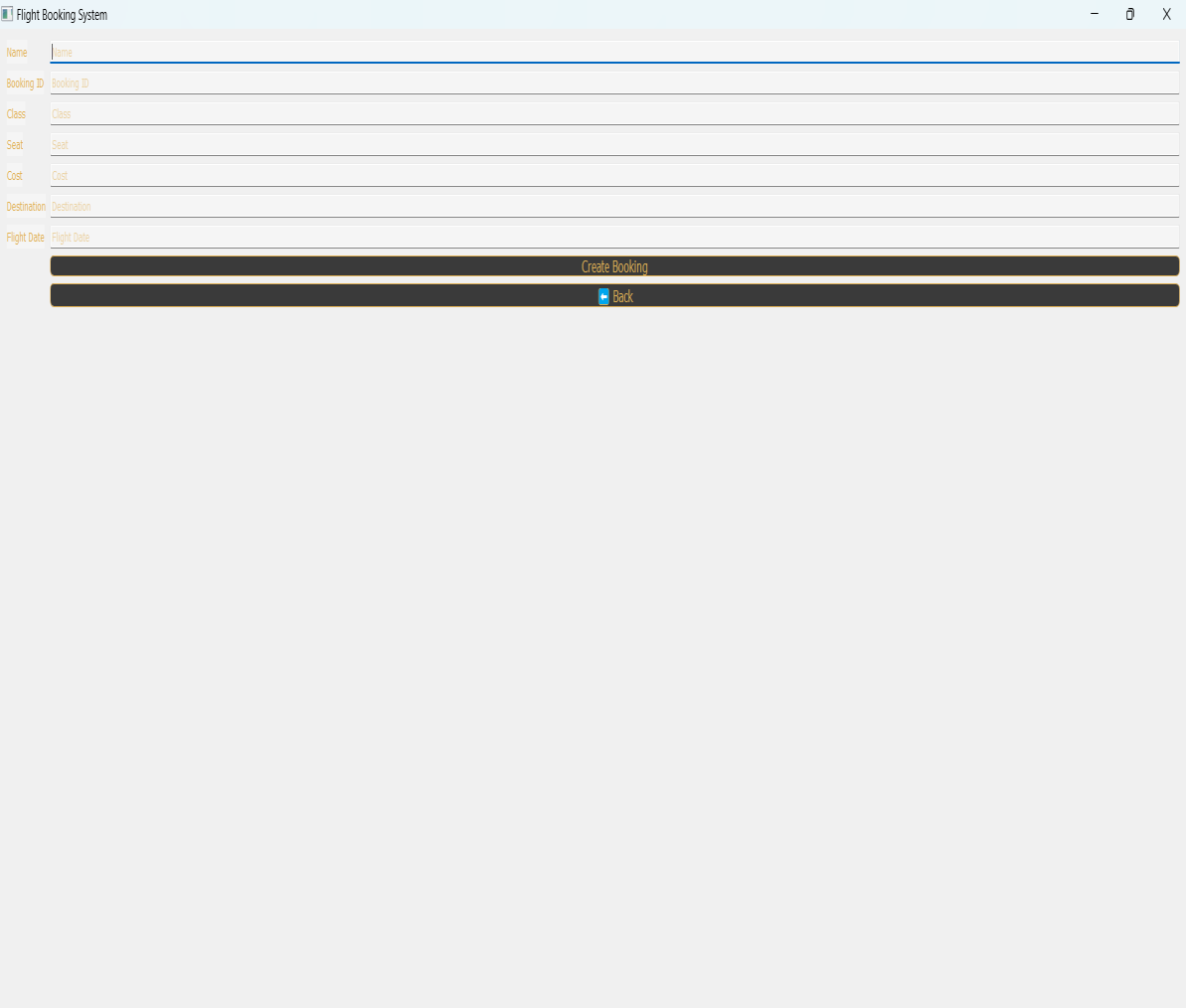
Output Photos :



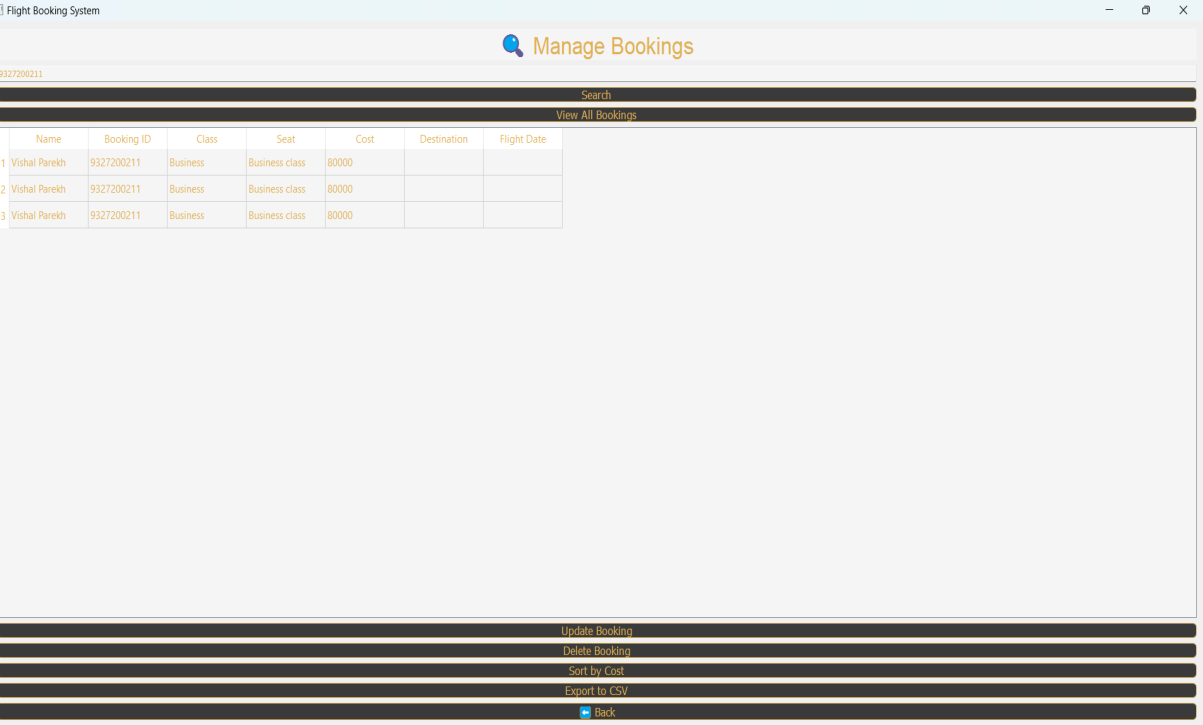
Admin login Page:



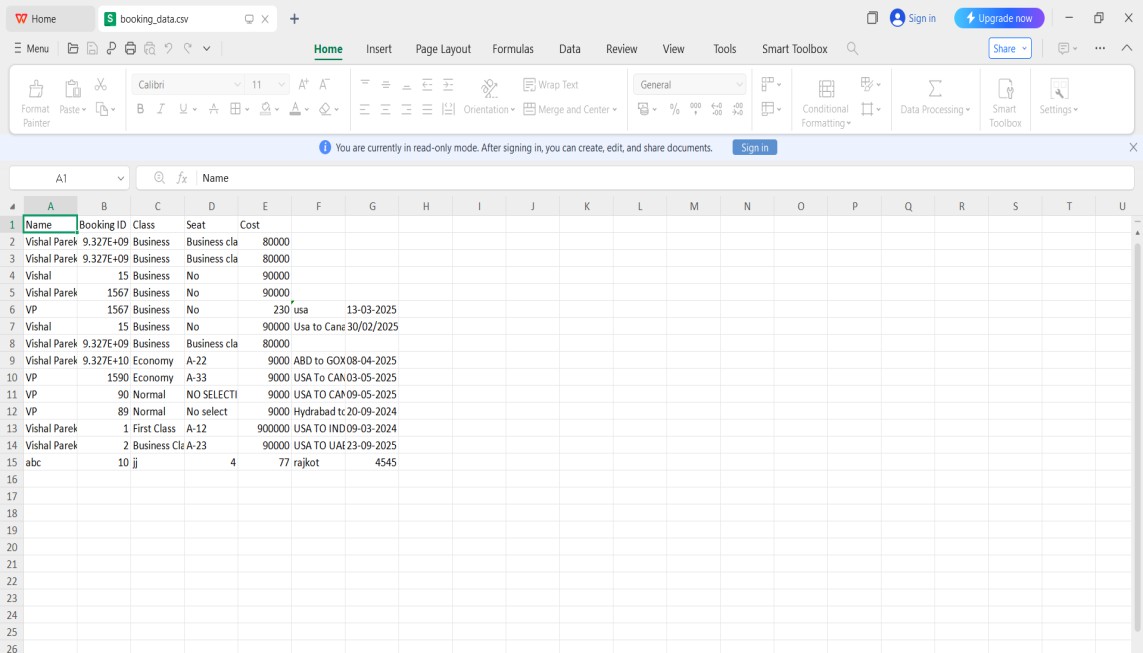
After Login Page:



Create Booking Page:



Manage Booking Page:



File Handling Data Store To Csv:

Conclusion :

This project, Flight Booking System, provides an admin interface to create, manage, and track flight bookings using a simple GUI built in PyQt5 and data storage in CSV.

It demonstrates the use of Python's GUI programming, file handling, and modular class-based design. The project enhances understanding of GUI event-driven programming and file I/O operations in Python.

