**Step 1: Define Requirements and Features**

* **Identify Core Features:**
  + Customer registration and login
  + Flight search and booking
  + Manage flight reservations
  + Seat selection
  + Purchase in-flight services (food, drinks, etc.)
  + View booking history and details
  + Admin features (if required): Add flights, manage bookings, etc.

**Step 2: Choose Your Technology Stack**

* **Frontend:** HTML, CSS, JavaScript (for web app)
* **Backend:**
  + **Option 1:** Node.js, Python (Flask/Django), Java (Spring Boot)
  + **Option 2:** C++ for file handling
* **Data Storage:**
  + Files (since a database is not mandatory)
  + Consider using JSON, CSV, or plain text files.

**Step 3: Design the Flow and UI**

* **Create Wireframes:**
  + Sketch or use tools like Figma to create a basic design of each page (e.g., home page, login, booking page, etc.).
* **Define User Flow:**
  + Example flow: User registers → Logs in → Searches for flights → Selects flight and seats → Makes payment → Receives booking confirmation.

**Step 4: Set Up the Project**

* **Create a New Project:**
  + Set up your project directory structure (e.g., separate folders for frontend, backend, and data files).
  + Initialize version control (e.g., Git).

**Step 5: Develop the Frontend**

* **Create Basic Pages:**
  + HTML for structure
  + CSS for styling
  + JavaScript for interactivity (e.g., form validation, dynamic content loading)
* **Pages to Create:**
  + Home
  + Login/Register
  + Search Flights
  + Booking Confirmation
  + Seat Selection
  + Purchase In-Flight Services

**Step 6: Develop the Backend**

* **Set Up the Server:**
  + Choose a framework (e.g., Express for Node.js, Flask for Python).
  + Implement basic routing (e.g., handle requests for booking, viewing reservations).
* **Implement Core Functionalities:**
  + **Customer Management:** Handle registration, login, and profile management.
  + **Flight Management:** Search, book, and manage reservations.
  + **Data Handling:** Read/write data to files for flight details, bookings, and user data.

**Step 7: Integrate Frontend and Backend**

* **Connect Frontend to Backend:**
  + Use AJAX/fetch API for sending data from the frontend to the backend.
  + Ensure the frontend can retrieve data (e.g., available flights, booking details) from the backend.

**Step 8: Implement Additional Features**

* **In-Flight Services:**
  + Create an interface for users to select and purchase services.
  + Store the selection in files and update the booking details.
* **Seat Selection:**
  + Implement seat maps and selection functionality.
  + Update the booking data with the selected seat.

**Step 9: Testing and Debugging**

* **Test Each Component:**
  + Test user flows (e.g., registration, booking) and fix any issues.
  + Validate data handling (e.g., ensure data is correctly read from and written to files).
* **Cross-Browser Testing:** Ensure compatibility across different web browsers (if applicable).

**Step 10: Finalize and Submit**

* **Code Cleanup:**
  + Remove unused code, add comments, and ensure readability.
* **Documentation:**
  + Prepare a README file explaining how to set up and run the project.
* **Submission:**
  + Package your project and submit it according to the assignment guidelines.