AIRLINES MANAGEMENT SYSTEM

VISHNU INSTITUTE OF TECHNOLOGY- VITB , Bhimavaram

TEAM ID - LTVIP2025TMID30597

TEAM MEMBERS DETAILS:

S.NO	NAME OF THE STUDENT
1.	M Pavani
2.	Davadula Chandini Jaya Priya
3.	Chitturi Jyothika
4.	Donthamsetti Sri Mahalakshmi Charishma

Airlines Management System:

A Salesforce-Based Operational Solution for Modern Aviation

1. INTRODUCTION

1.1 Project Overview

The aviation industry is one of the most dynamic and operationally complex sectors in the modern world. With growing air traffic, diverse routes, increased passenger expectations, and stringent safety regulations, efficient airline management has become a critical priority. Digital transformation is now key in ensuring that airlines can manage operations seamlessly while maintaining customer satisfaction.

This project—Airlines Management System—is a Salesforce-based cloud application developed to centralize and automate the core processes of an airline. From flight scheduling and passenger reservations to crew assignment and baggage handling, the system covers it all. By leveraging Salesforce's declarative tools such as custom objects, automation flows, dashboards, and reports, the solution provides a scalable, secure, and user-friendly platform for airline operations.

The system not only supports backend data management but also empowers administrative users with role-based access, report-driven insights, and efficient user interfaces—all contributing to streamlined workflows and better operational decision-making.

Scenario 1:

Real-Time Booking and Flight Management in a Mid-Sized Airline

A regional airline faces challenges in managing increasing passenger volumes, last-minute booking changes, and maintaining timely communications. By adopting the Airlines Management System, the airline configures custom objects for flights, bookings, and passengers. A Lightning App provides centralized access, while Flows automate the reservation process. Reports and dashboards allow management to view daily flight occupancy, check-in statuses, and booking trends. This results in smoother check-ins, faster service, and improved customer feedback.

Scenario 2:

Crew Scheduling and Access Management in a Growing Aviation Startup

An emerging private airline struggles with assigning flight crew and ensuring secure access control. Using Salesforce profiles and roles, the airline defines structured access to flight and crew records. Each crew member has visibility only into relevant flight data, while admins manage assignments via the Crew object. Apex triggers validate key data entries (e.g., mandatory contact fields), and dashboards highlight staff allocations per flight. This setup enhances operational security and improves overall crew coordination.

1.2 Purpose

The primary purpose of the Airlines Management System is to streamline and automate the core operational processes of an airline—specifically focusing on flight scheduling, passenger management, crew assignment, and booking workflows—by leveraging the capabilities of the Salesforce Lightning Platform. In the modern aviation industry, operational efficiency, data accuracy, and customer experience are key pillars for success. This system aims to replace outdated manual methods and fragmented tools with a centralized, cloud-based solution that can manage end-to-end airline operations in a secure and scalable environment.

Traditional airline operations often involve excessive paperwork, repetitive data entry, and coordination across disconnected systems. This leads to common issues such as booking errors, flight mismanagement, scheduling conflicts, and data redundancy. The purpose of this project is to resolve these challenges by designing a modular application that integrates data, logic, automation, and reporting into one cohesive platform.

Key goals of the project include:

- Digitizing flight and passenger records to eliminate manual entries and reduce operational delays.
- Creating booking workflows using Salesforce Flows to guide users step-by-step and enforce business rules.
- Automating data validation using Apex triggers (e.g., phone number verification), ensuring only clean, usable data enters the system.
- Assigning crew members efficiently through a centralized interface with proper role permissions and object relationships.
- Providing real-time analytics via Salesforce Reports and Dashboards for better decision-making by airline management.

Beyond operational improvements, this project also aims to:

- Serve as a learning model for low-code and cloud-based application development.
- Demonstrate the use of Salesforce’s declarative tools (like Object Manager, Schema Builder, Flow Builder) along with programmatic logic (like Apex Classes and Triggers).
- Deliver a scalable foundation that could eventually evolve into a commercial-grade airline ERP system with real-time integrations and external service APIs.

2. IDEATION PHASE

2.1 Problem Statement

In the aviation industry, managing flight operations, passenger data, and crew assignments efficiently is critical to ensuring timely service, customer satisfaction, and operational compliance. Many airlines—especially regional and low-cost carriers—still rely on fragmented systems or manual processes to handle bookings, crew schedules, and performance tracking.

These outdated methods result in:

- Duplicate data entries
- Delayed or mismanaged bookings
- Lack of real-time visibility into operational KPIs
- Poor coordination between departments

There is a clear need for a centralized, cloud-based system that provides:

- Seamless passenger and flight management
- Automated workflows
- Data-driven dashboards
- Role-based access for airline staff

The problem:
Airline operations lack a unified digital platform that supports efficient booking management, crew scheduling, and performance monitoring—leading to manual delays, poor data integrity, and limited operational insight.

2.2 Empathy Map Canvas

The Empathy Map Canvas helps visualize what the target users (airline staff and passengers) feel, think, and experience while interacting with current airline systems. It guides the ideation process to develop a user-centered solution.

User: Airline Booking Staff / Operational Admin

Category	Insights
Says	“I need to check availability and assign passengers quickly.”
Thinks	“I wish I had one place to see all flight and passenger data.”
Does	Manually searches for flights, enters booking data, coordinates crew
Feels	Frustrated due to time-consuming and repetitive tasks

User: Airline Passenger

Category	Insights
Says	“I want to know my booking status instantly.”
Thinks	“Is my flight on time? Did my seat get confirmed?”
Does	Calls customer support, checks for booking confirmation emails
Feels	Anxious about confirmation delays or errors

2.3 Brainstorming

The Airlines Management System, built on the Salesforce platform, offers a comprehensive suite of features designed to streamline and automate airline operations. The system is modular, user-friendly, and adaptable to different scales of airline businesses. Below are the key features and their functional

1. Flight Management

- Custom object to manage flight details including flight number, routes, schedules, departure/arrival times, and PNR.
- Picklist-controlled origin and destination cities with field dependencies to maintain route accuracy.
- Validation of unique flight names and departure constraints.

2. Passenger Management

- A dedicated object for passenger information such as name, contact details, passport data, nationality, and date of birth.
- Includes field validation (e.g., phone number mandatory via Apex Trigger).
- Allows real-time data viewing, editing, and deletion.

3. Booking System

- Automated booking ID generation using auto-number format.
- Lookup relationships with Flight and Passenger objects to ensure integrity.
- Custom picklists for travel class (Economy, Business, First Class) and traveler type (Adult, Child, Infant).
- Date-based reservation validation.

4. Crew Assignment Module

- Stores crew member data including roles, qualifications, and assigned flights.
- Picklist-driven roles such as Pilot, Co-Pilot, Flight Attendant, On-board Chefs, etc.

5. Role-Based Access Control

- User profiles (Senior Admin, Management Admin, General Admin, Crew Member) with object-level permissions.
- Roles assigned hierarchically to reflect organizational structure (e.g., Crew reports to Management Admin).
- Ensures secure data access and separation of concerns.

6. Reports and Dashboards

- Multiple Salesforce reports created (e.g., Bookings with Flight, Crew by Flight).
- Dashboards configured with components like donut charts and summaries for executive overviews.
- Real-time visibility into flight status, passenger load, and crew deployment.

7. Automation & Validation

- Apex trigger ensures phone number entry before saving passenger data.
- Test class ensures >75% code coverage for Apex trigger logic.

8. Visual Schema and Field Relationships

- Schema Builder used to map all object relationships (Passenger, Booking, Flight, Crew).
- Lookup and master-detail relationships maintained for integrity and reporting efficiency.

9. Flight Operations Module

This module allows the airline to manage its fleet operations efficiently.

- Flight Object: Custom object that captures flight-specific details like flight name, flight number (PNR), departure and arrival times, and route.
- Field Dependencies: Used for “Departs From” and “Departs To” fields to ensure city combinations are contextually valid.
- Picklist Values: Ensure data standardization across origin and destination airports.

10. Booking & Reservation System

This module enables seamless booking and seat assignment.

- Booking Object: Auto-numbering of bookings using a formatted booking ID (Bk-{0000}).
- Relationships: Linked to Passenger and Flight objects via Lookup Relationships to ensure data consistency.

11. Reporting and Dashboards

Visual analytics support decision-making and operational insight.

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

The Customer Journey Map outlines the step-by-step interaction between the user (typically an airline staff member or admin) and the Airlines Management System during routine operations.

Persona: Airline Admin / Booking Staff

Stage	Action
Awareness	User logs into Salesforce and accesses the Airlines Management App
Consideration	Navigates tabs like Flights, Passengers, and Bookings to check current data
Booking Process	Uses the Booking Flow to reserve seats and generate booking IDs
Crew Assignment	Opens Crew tab to assign staff to specific flights
Monitoring	Uses dashboards to view booking trends, flight occupancy, and crew status
Reporting	Generates reports for bookings, passengers, and flights for analysis

This journey is streamlined by Salesforce's user-friendly UI and role-based accessibility.

3.2 Solution Requirement

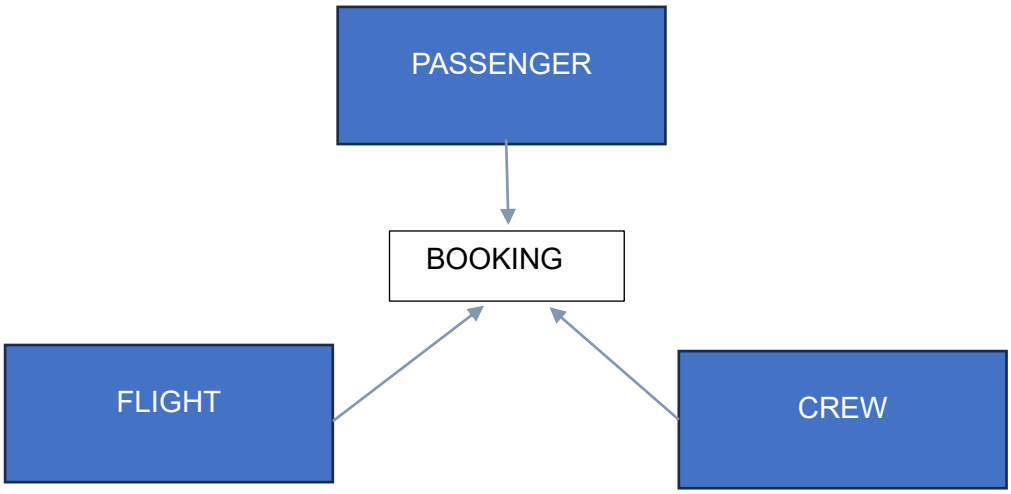
Functional Requirements:

- Add, update, and delete flight records
- Manage passenger information (Name, Passport, Phone, Nationality)
- Create and manage bookings through screen flows
- Assign crew to specific flights
- Generate real-time reports and dashboards
- Validate phone number entry via Apex logic

Non-Functional Requirements:

- Cloud-hosted and accessible from any browser
- Secure access based on roles and profiles
- Scalable to accommodate increasing bookings and users
- Low-code/no-code environment for ease of development
- Fast response times using Salesforce's cloud infrastructure

3.3 Data Flow Diagram



- Passengers book flights via the Booking module
- Each booking is linked to a flight
- Crew is assigned to flights for operational tracking

3.4 Technology Stack

Layer	Technology / Tool
Platform	Salesforce Lightning Platform
Database Layer	Salesforce Custom Objects (Flight, Booking, etc.)
Backend Logic	Apex Classes and Triggers
Automation	Salesforce Flows (Screen Flow for Booking)
UI Layer	Lightning App with custom Tabs and Page Layouts
Reporting	Salesforce Reports and Dashboards
Relationships	Schema Builder (for visual object relationships)

4. PROJECT DESIGN

4.1 Problem–Solution Fit

The problem of manually managing airline bookings, passenger data, and crew assignments results in time-consuming processes, increased human error, and lack of real-time insights. Airline staff often juggle multiple disconnected systems, leading to inefficiencies and operational delays.

Our solution addresses these pain points by offering a centralized, cloud-based application using Salesforce. The system ensures:

- Streamlined booking workflows
- Automated data validation
- Role-based access for secure data handling
- Real-time dashboards for operational visibility

By aligning real user needs (speed, visibility, control) with the solution's capabilities, the system achieves a strong problem–solution fit.

4.2 Proposed Solution

The Airlines Management System is developed using Salesforce's declarative tools and low-code automation features. It centralizes all major airline functions including flight management, passenger bookings, and crew scheduling within a secure and scalable Lightning App.

Key Modules:

- Flight Management: Create, edit, and schedule flights with route and timing data.
- Passenger Handling: Manage personal details, passport info, and nationality.
- Booking System: Auto-generate booking IDs and link passengers to flights.
- Crew Assignment: Allocate staff roles like Pilot, Co-Pilot, and Attendant to flights.
- Reports & Dashboards: Visualize real-time metrics like booking trends and occupancy.
- Validation & Automation: Apex and Flow logic ensure clean, efficient data handling.

4.3 Solution Architecture

The solution follows Salesforce's multi-layered architecture:

1. User Interface Layer

- Lightning App with custom tabs: Flights, Passengers, Bookings, Crew
- Screen Flow: For guided booking creation
- Reports & Dashboards: Used for real-time operational insights

2. Application Logic Layer

- Apex Trigger: Validates phone number before passenger record insertion
- Apex Class: Handles modular backend logic
- Test Class: Ensures 75%+ code coverage
- Screen Flow Logic: Used to automate booking creation

3. Data Layer

- Objects:
    - Flight__c: Flight details
    - Passenger__c: Personal data
    - Booking__c: Flight-passenger link
    - Crew__c: Staff and role assignments
- Relationships:
    - Booking → Flight (Lookup)
    - Booking → Passenger (Lookup)
    - Crew → Flight (Lookup)

4. Access Control Layer

- Profiles: Define object-level permissions (e.g., Crew Member = read-only)
- Roles: Determine record visibility (e.g., Management Admin sees crew data)
- Users: Assigned both profile and role

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

The Airlines Management System was executed using a phase-wise planning model, aligned with agile and iterative development principles. Each stage was mapped with specific goals, outcomes, and deliverables as per SmartBridge's structured learning journey. The aim of the planning process was to ensure clarity, avoid overlapping responsibilities, and track progress efficiently from ideation to execution.

Objective of Planning

The purpose of this project planning phase was to:

- Define milestones across the project life cycle
- Allocate weekly goals for each module (e.g., object creation, flow design, reports)
- Ensure time for testing, refinement, and presentation
- Use SmartInternz LMS and Salesforce Dev Org for continuous delivery

Weekly Development Timeline:

Week	Phase	Activities / Deliverables
Week 1	Ideation Phase	- Finalized problem statement - Created empathy map canvas - Performed team brainstorming
Week 2	Requirement Analysis	- Created customer journey map - Drafted solution requirements - Outlined data flow diagram - Chose technology stack
Week 3	Object Modeling & Setup	- Created Salesforce objects: Flight, Passenger, Booking, Crew - Added custom fields and relationships
Week 4	UI & Automation Design	- Created tabs and Lightning App - Designed screen flows for booking - Developed Apex trigger for phone validation
Week 5	Reporting and Access Control	- Built reports and dashboards - Configured profiles and roles - Created users and role hierarchy
Week 6	Testing & Finalization	- Conducted Apex test class and flow validation - Compiled screenshots - Recorded demo and final documentation

Tools & Platforms Used

Tool/Platform	Purpose
Salesforce Developer Org	Main platform for building and testing the application
Flow Builder	Automation via screen flows for booking module
Developer Console	Apex trigger, class, and test class development
SmartInternz LMS	Weekly module tracking and deliverable submission
Schema Builder	Visual relationship mapping
Report/Dashboard Builder	Real-time analytics and user data insights
Google Docs / MS Word	Documentation and team collaboration

Change Management

During development, feedback was incorporated at each checkpoint. For example:

- After Week 3, the booking object was enhanced with auto-numbering and field dependencies
- During Week 5, role hierarchy was added to secure access control

Outcome of Planning

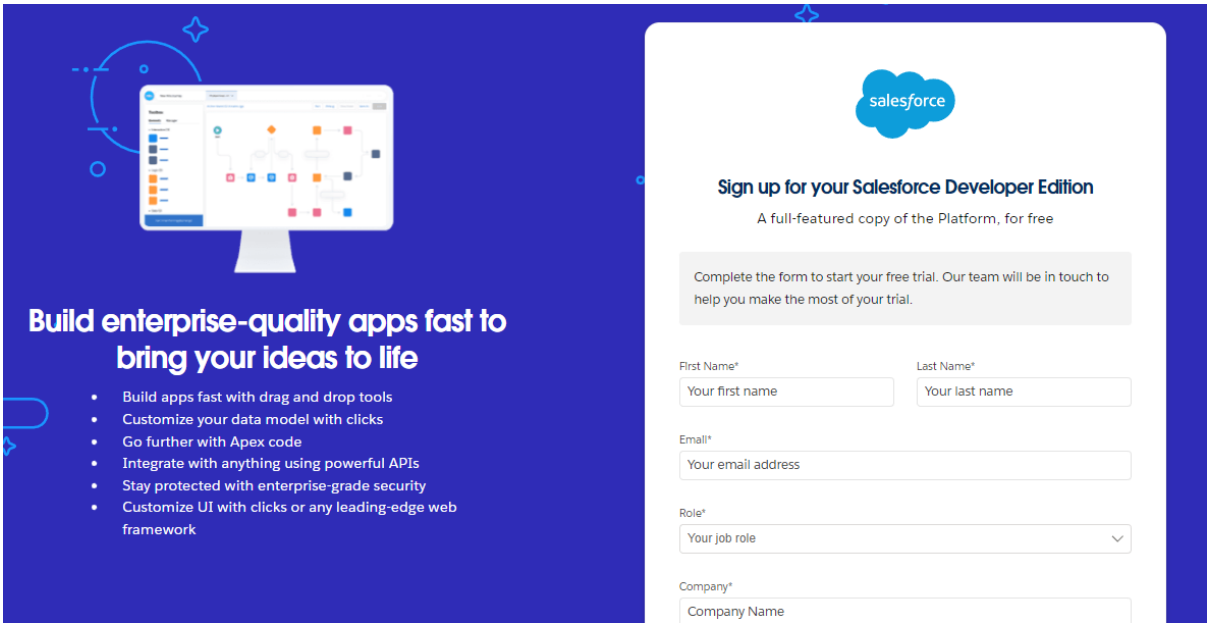
This phase-wise planning ensured that:

- No module was missed or delayed
- The project remained within scope and timeline
- Weekly tasks aligned with Smart Bridge’s structured checkpoints
- The system was tested thoroughly before final submission

Creating Developer Account

Creating a developer org in salesforce.

1. Go to https://developer.salesforce.com/signup
2. On the sign up form, enter the following details :



1. First name & Last name
2. Email
3. Role : Developer
4. Company : College Name
5. County : India
6. Postal Code : pin code
7. Username : should be a combination of your name and company

Click on sign me up after filling these.

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

The Airlines Management System was thoroughly tested to validate its logic, automation flow, field dependencies, and overall performance. Testing focused on both functional accuracy and code efficiency, ensuring the system meets Salesforce's best practices

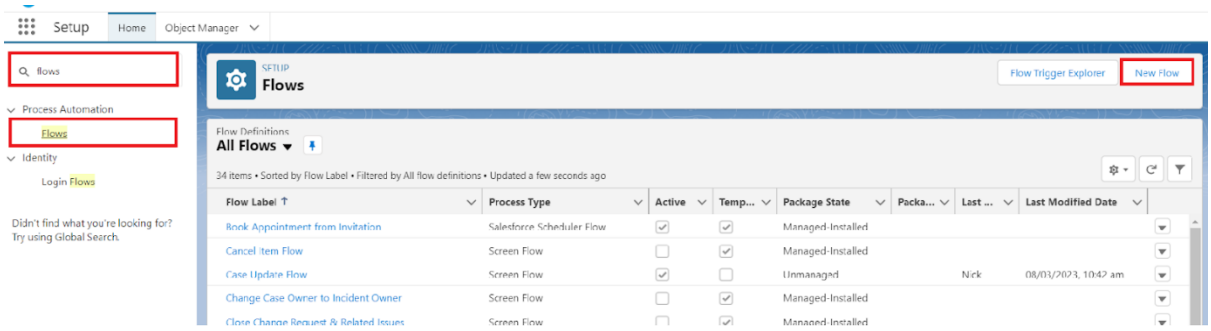
Flow Testing

The Booking process was implemented using a Screen Flow, designed to collect booking data and create records automatically.

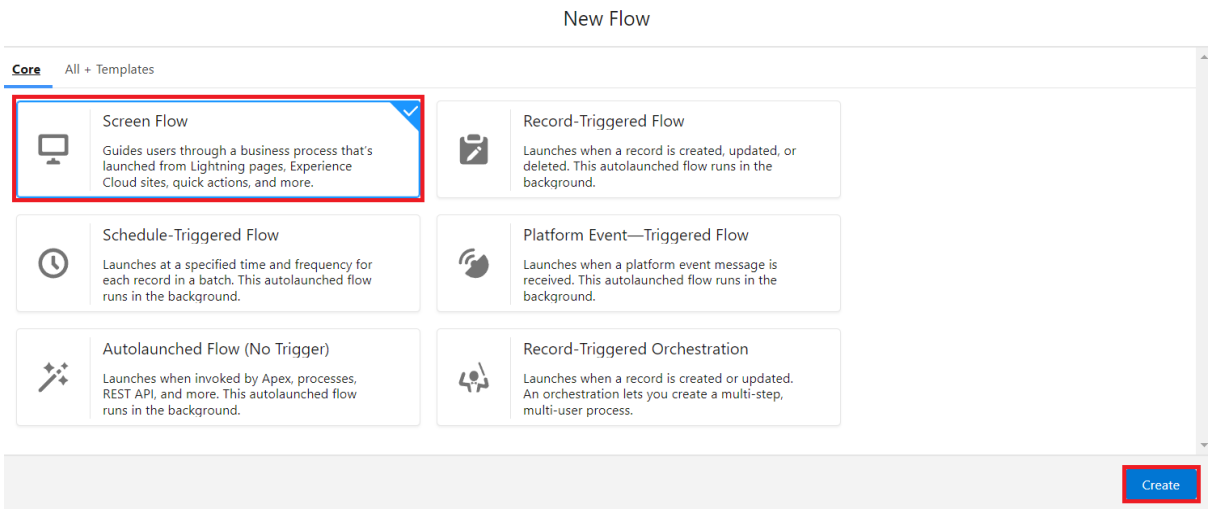
Flow Testing Activities:

- Used the Flow Debugger to simulate user input
- Validated correct mapping of fields (e.g., class, departure city)
- Ensured a confirmation screen appeared on successful record creation
- No crashes or unexpected behavior were encountered

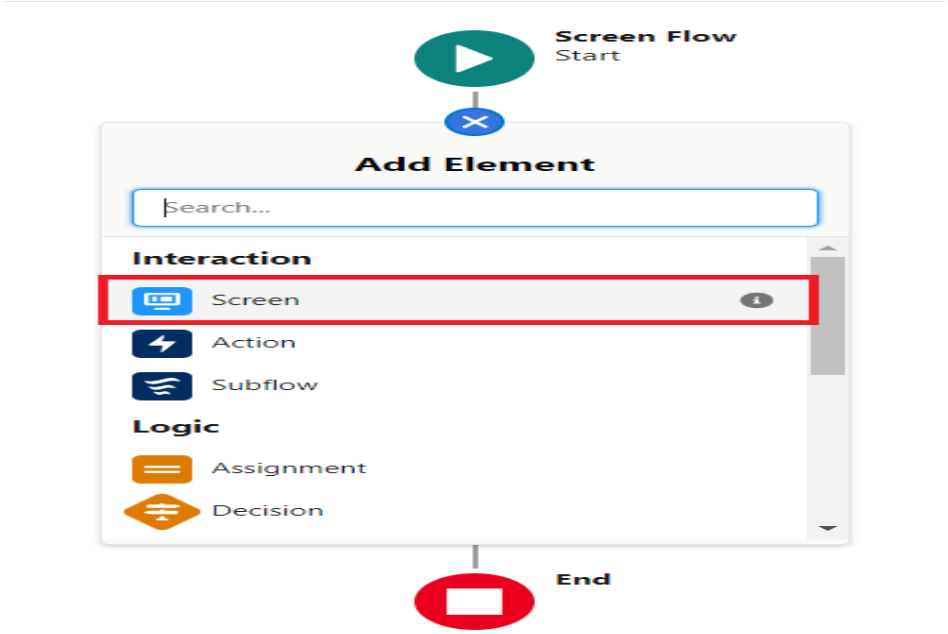
1. From Setup search for the "Flows" in quick find and select "New Flow".



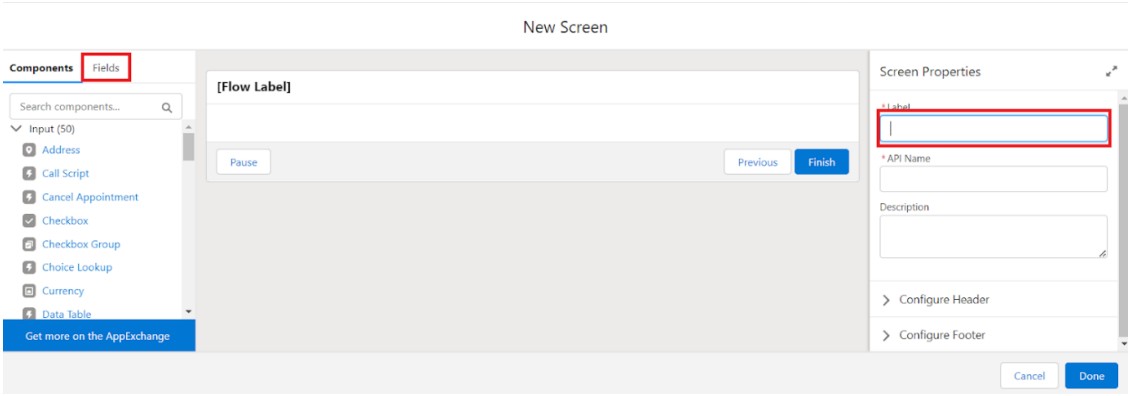
2. Select "Screen Flow" and then click on "Create".



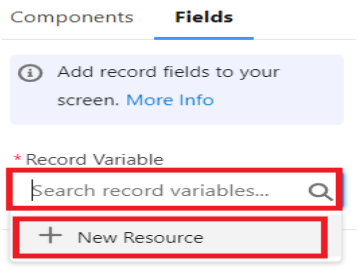
3. Place the cursor in between the Start and End element, a “+” icon appears, click on that and select “Screen”.



4. Give the label as “Booking Screen”, and select “Fields”.

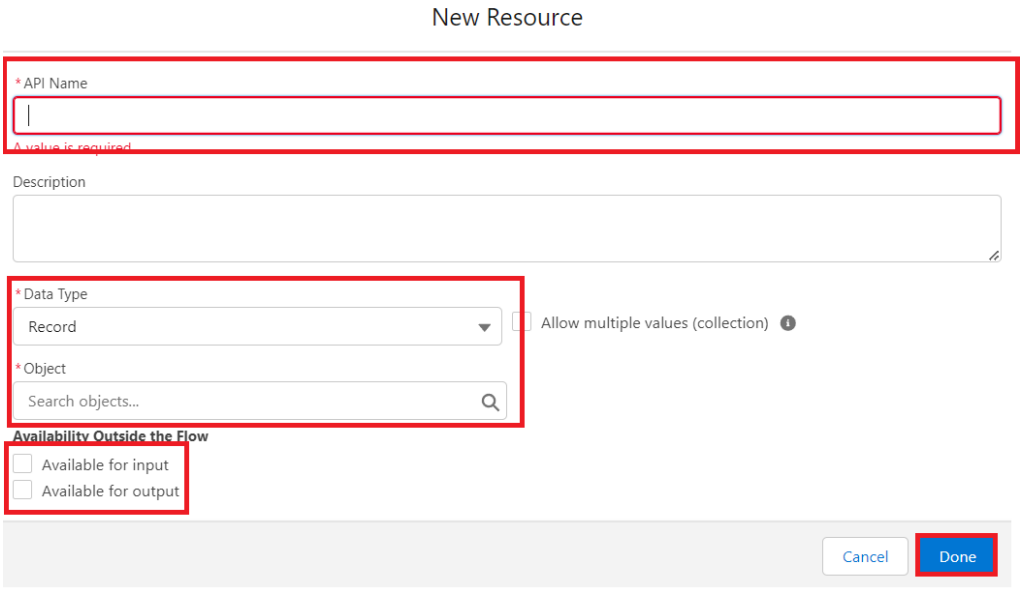


5. Click on the lookup icon in the input field and select “New Resource”.



6. Filling the fields as given below:

    1. API name : booking Object
    2. Datatype : Record
    3. Object : Booking
    4. Available for input : Checked
    5. Available for output: Checked

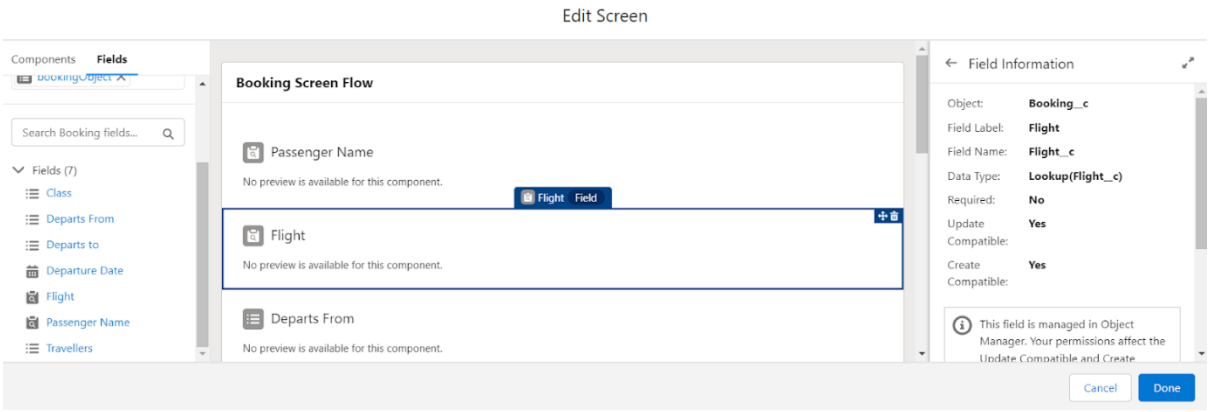


7. Click on done

8. Once you are done with the creation of a resource with the name “bookingObject” go back to step 5 and click on the lookup icon and select “bookingObject” from the list.

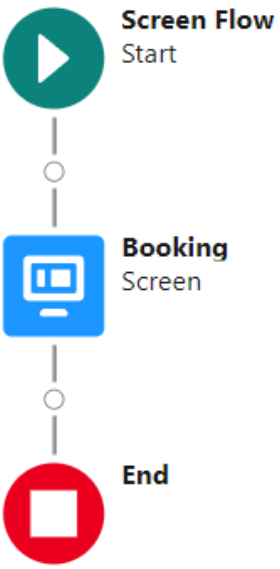
9. After selecting “bookingObject” you’ll have all the fields which are in the object, drag and drop each field on the screen.\

10. Your screen will look like as shown below:



11. Click on done and Save.

12. The flow will look like as shown below.



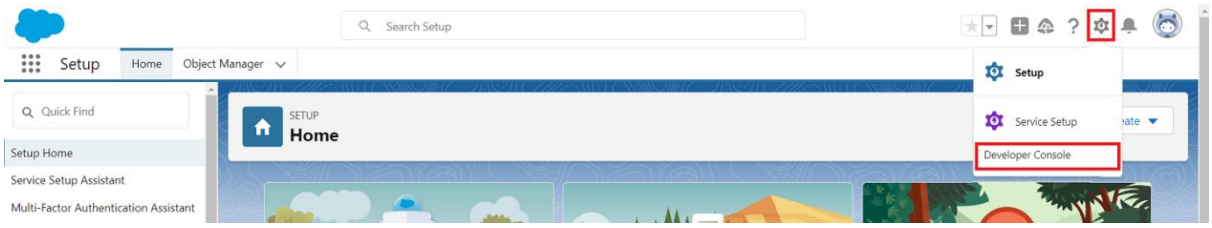
Why Apex Testing Matters

Salesforce enforces a minimum 75% code coverage requirement for Apex code before it can be deployed to production. However, in real-world and academic projects, testing isn't just about code coverage — it's also about ensuring that:

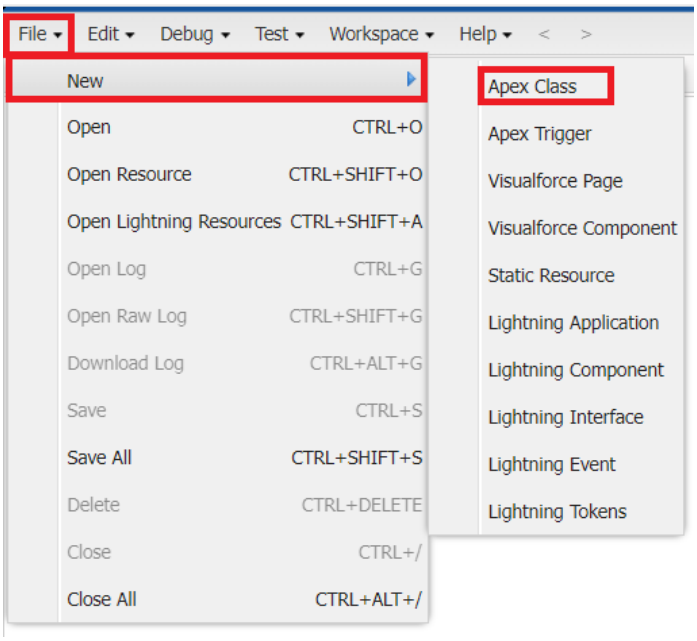
- Business rules are enforced
- Invalid data is prevented
- Automation behaves as expected
- The system remains stable with growing data
- Use Case: Passenger Phone Number Validation
- A key business rule in the Airlines Management System is that every Passenger record must include a valid phone number. This rule is enforced using an Apex Trigger that runs before insertion of a Passenger__c record.

Create an Apex Class

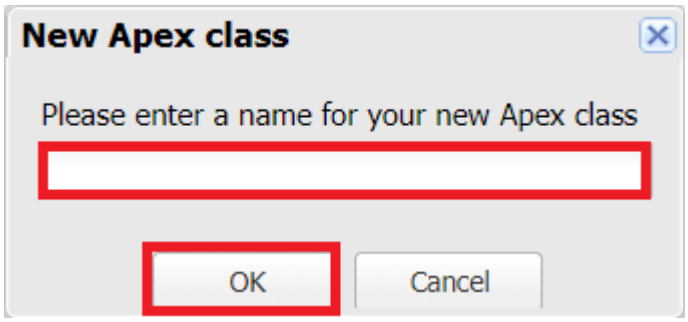
1. Go to Setup --> Click on the gear icon --> Select Developer Console.



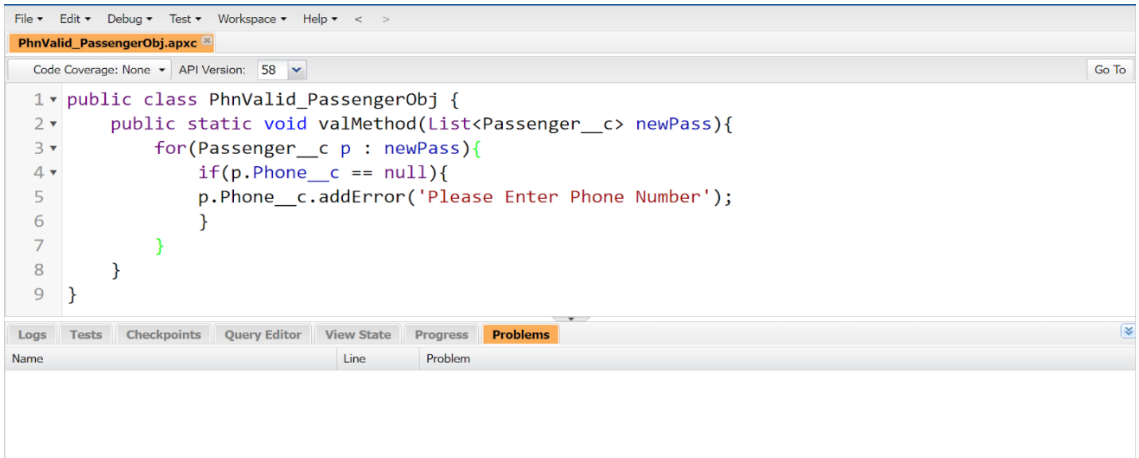
2. Then we can see the Developer console. Click on the developer console and you will navigate to a new console window.
3. To create a new Apex Class follow the below steps:
   Click on the file --> New --> Apex Class.



4. Give the Apex Class name as “PhnValid_PassengerObj”.



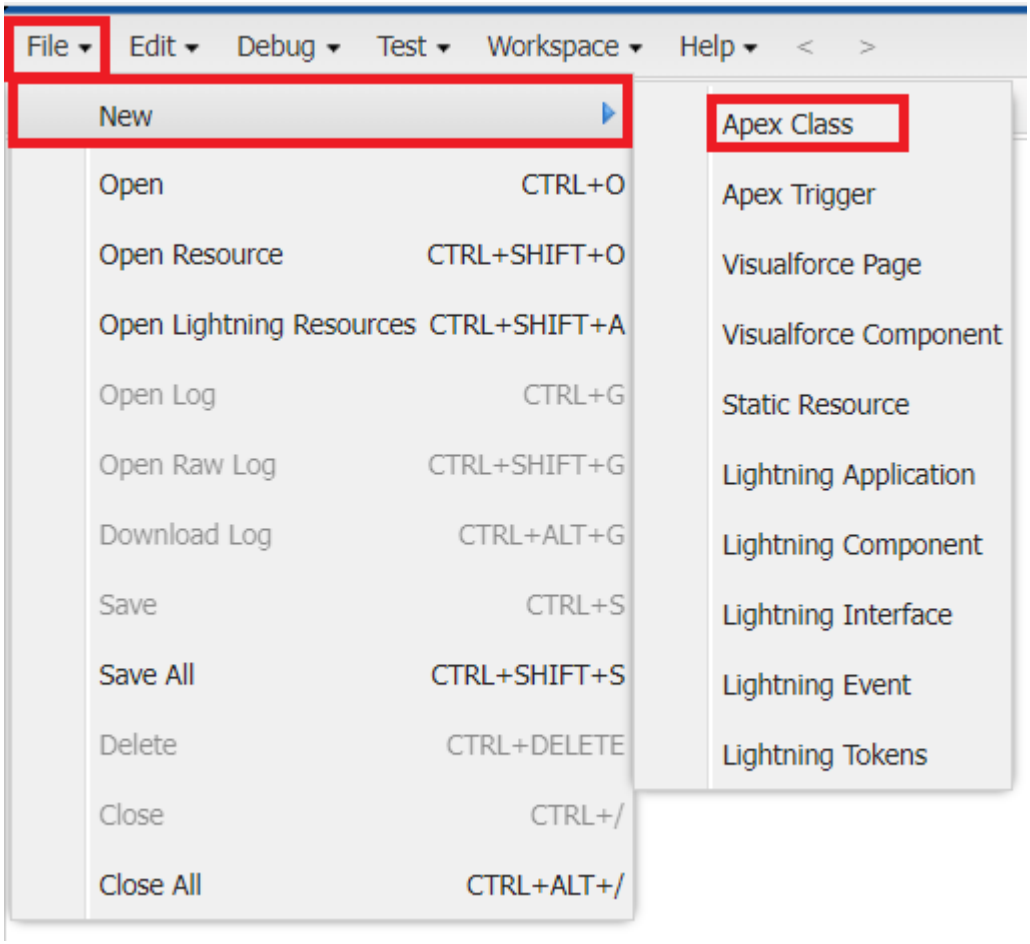
5. Click ok.
6. Now write the code logic here



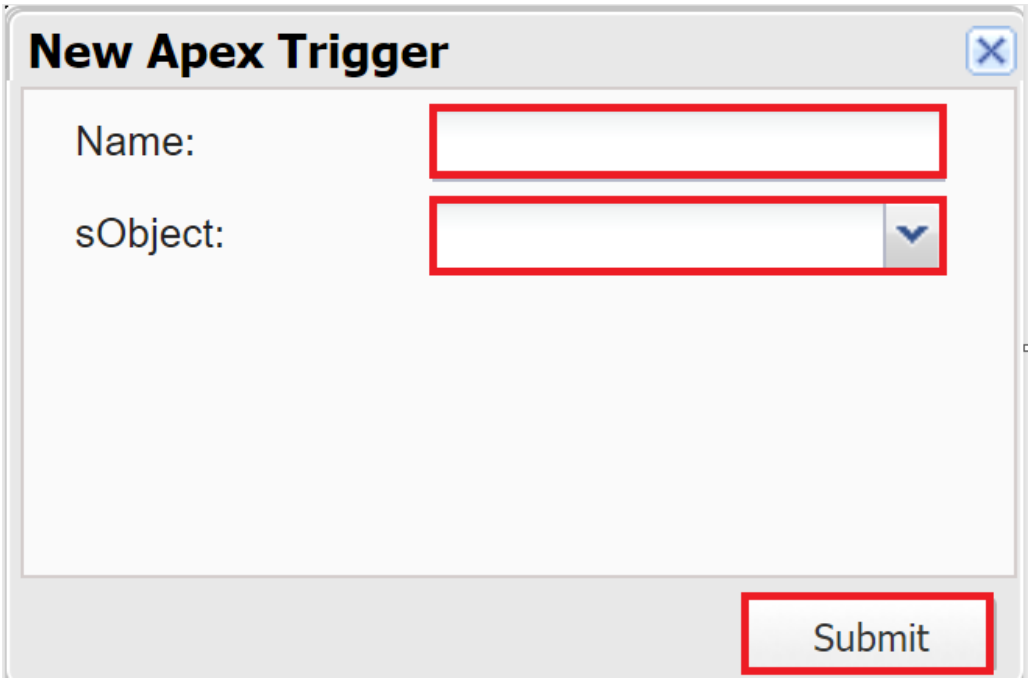
7. Save the code

Create an Apex Trigger

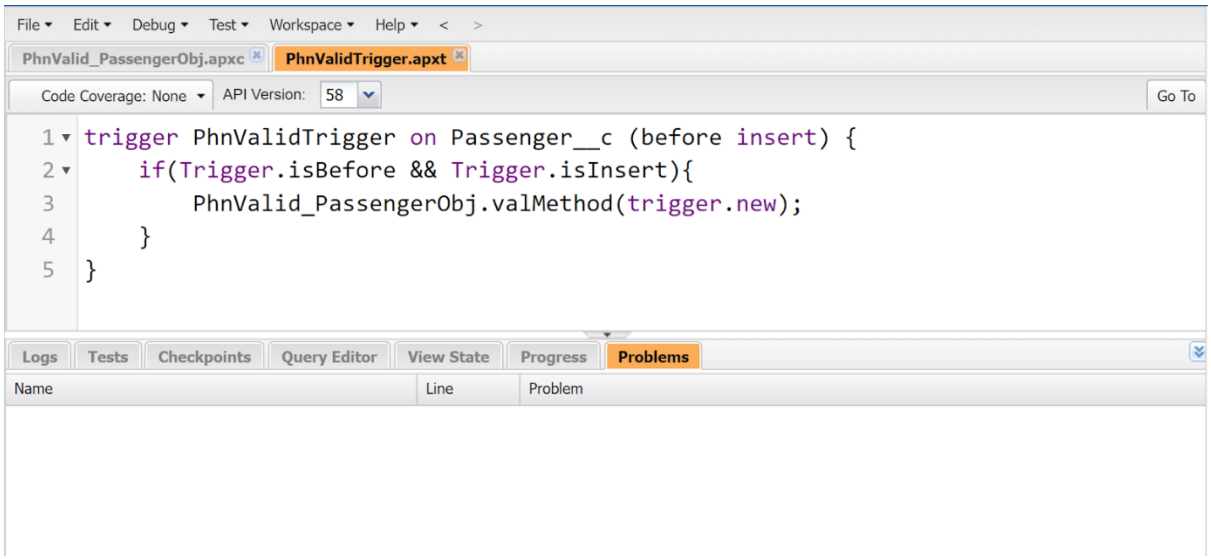
1. To create a new Apex Class follow the below steps:
   Click on the file --> New --> Apex Class.



2. Give the Apex Trigger name as “PhnValidTrigger”, and select “Passenger__c” from the dropdown for sObject.



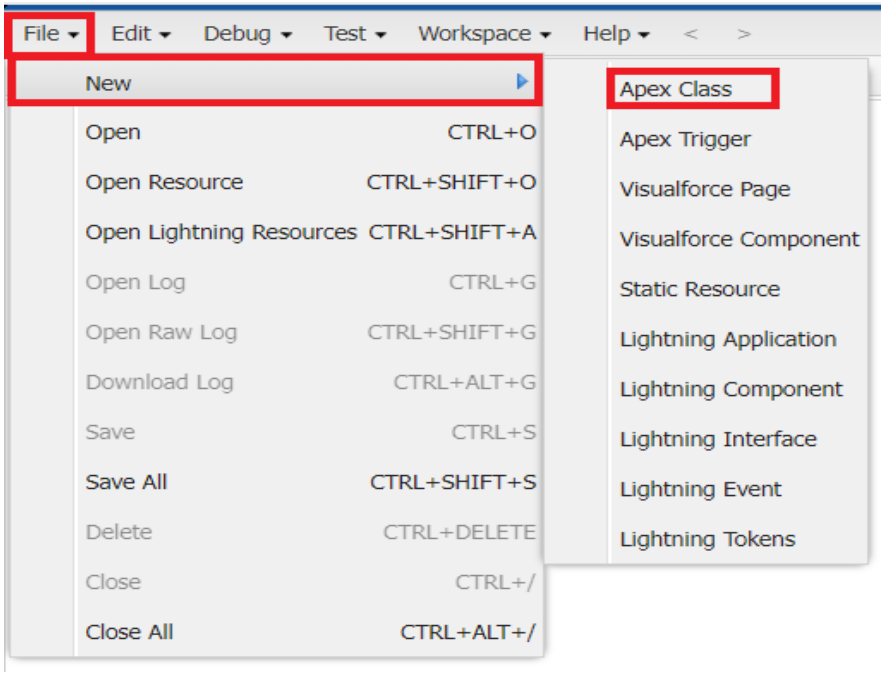
3. Click Submit.
4. Now write the code logic here



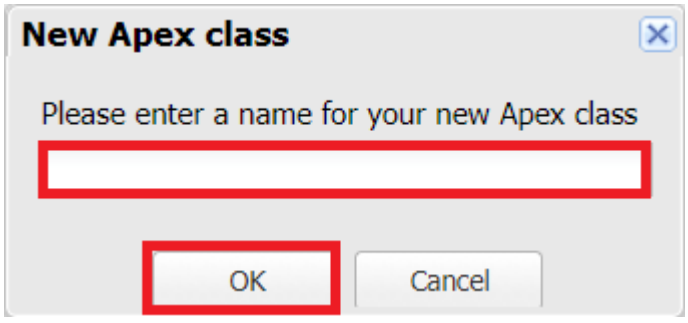
5.Save the code

Create an Apex Test Class

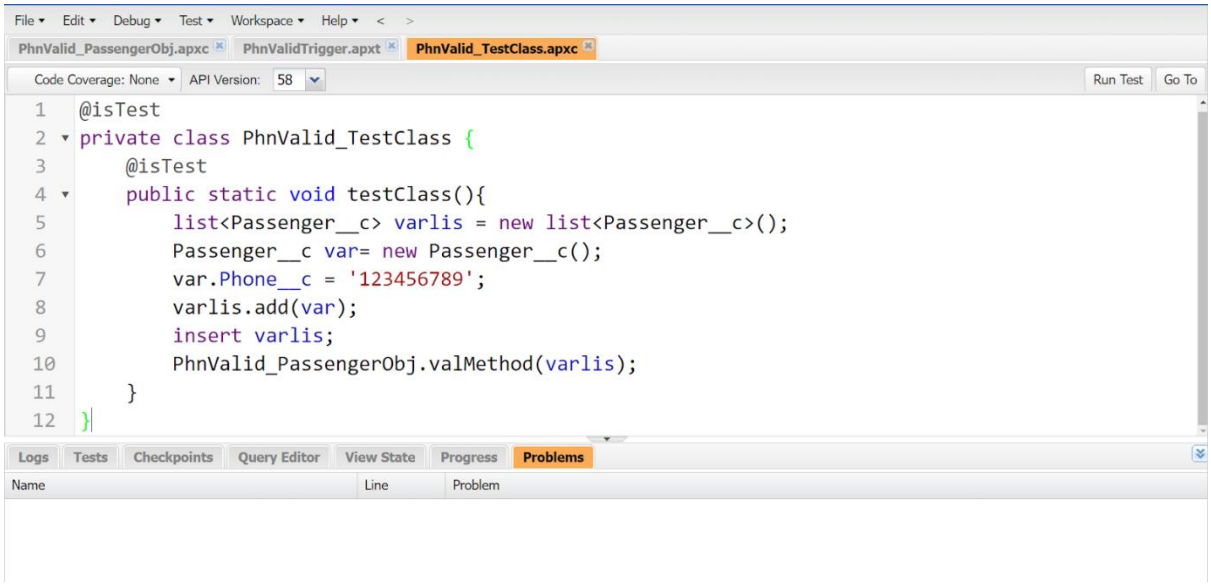
1. To create a new Apex Class follow the below steps:
   Click on the file --> New --> Apex Class.



2. Give the Apex Class name as “PhnValid_TestClass”.

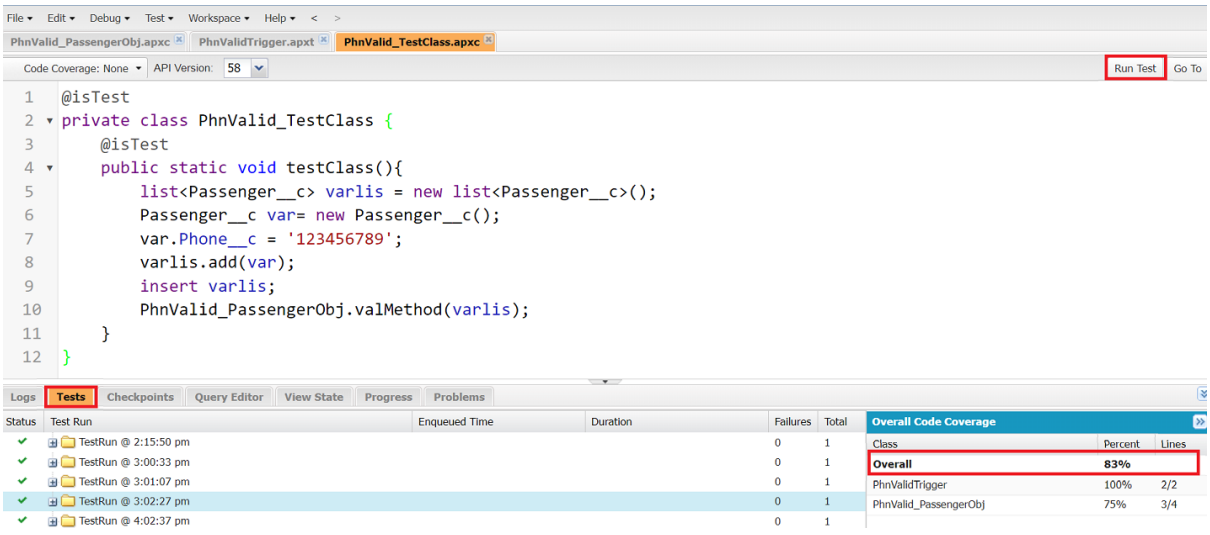


3. Click ok.
4. Now write the code logic here



5. Save the code.(click on file --> Save).

6. Click on “Run Test” and then click on Test under the terminal section and do check that your overall code coverage should be more than 75%.



7.RESULTS

7.1 Output Screenshots

What Are Reports in Salesforce?

In Salesforce, a Report is a dynamic, filterable list of records displayed in table format. Reports allow users to gather specific sets of data — such as flight bookings, passenger details, or crew assignments — and analyze them in real time.

Each report is created from a report type (e.g., “Bookings with Flight Details”) and can be customized using:

- Filters (e.g., date ranges, class types)
- Groupings (e.g., bookings per flight)
- Summaries (e.g., total bookings)

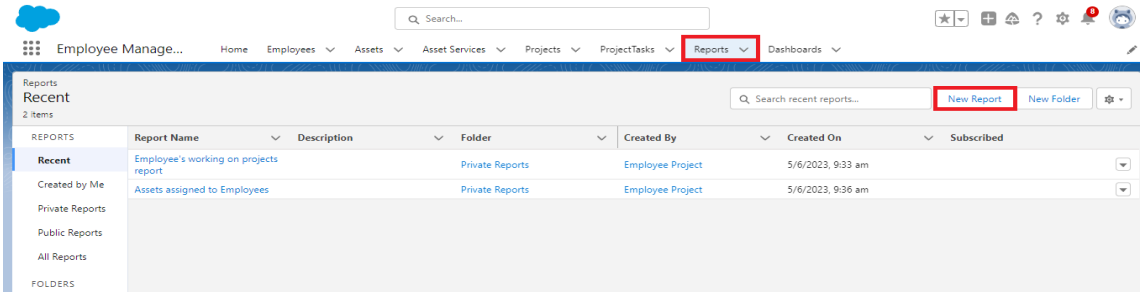
Reports Created in the Airlines Management System:

1. Bookings with Flight Details
    - Displays all bookings, associated flights, departure cities, and passenger class
2. Crew with Assigned Flights
    - Lists each crew member and the flight they’re assigned to
3. Passenger Summary Report
    - Shows total passengers per flight or per booking class

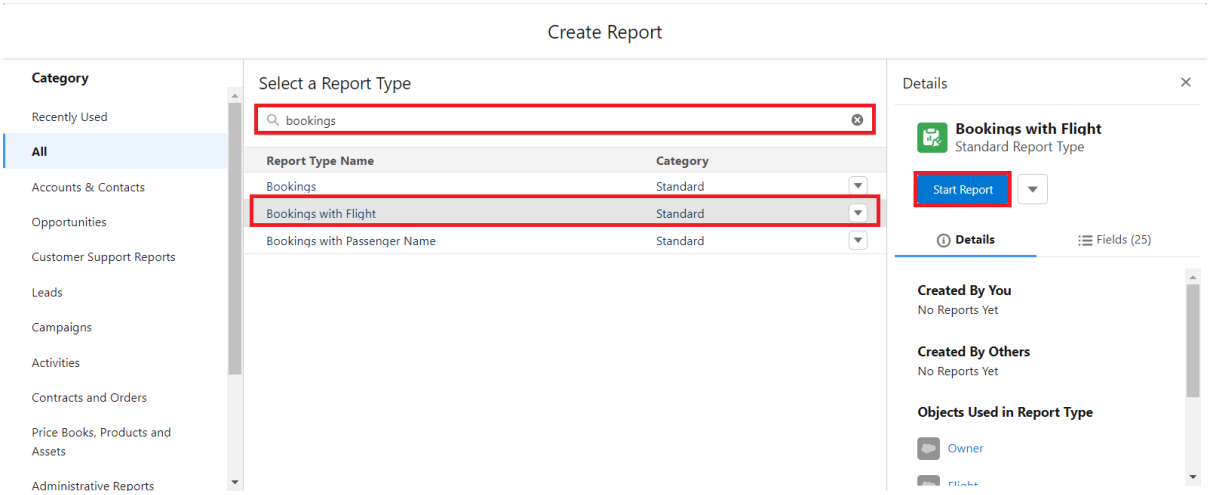
These reports are interactive, exportable, and refresh automatically when the data changes.

Create Report

1. Go to the app --> click on the reports tab
2. Click New Report.

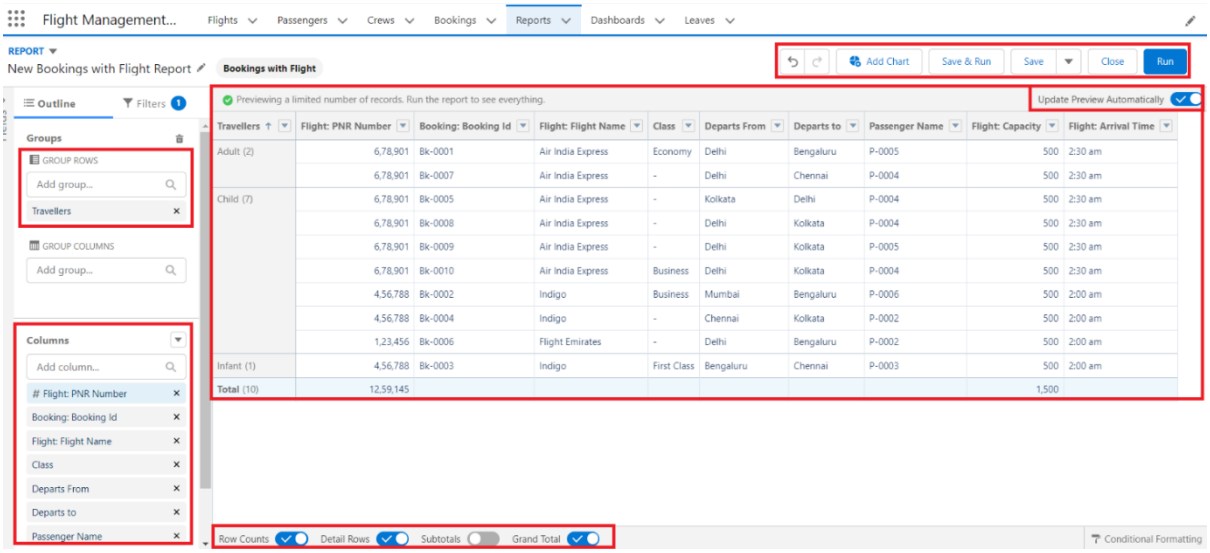


3. Select report type from category or from report type panel or from search panel --> click on start report.



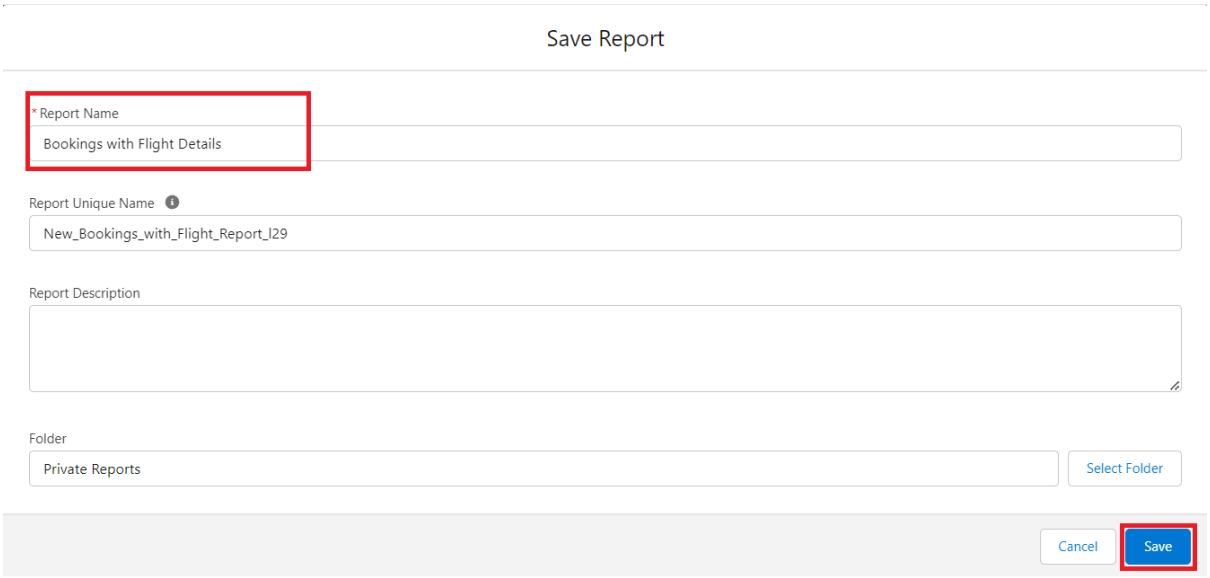
4. Customize your report

- Check the toggle on the update preview automatically button.
- Add fields from the left pane as shown below.
- Add the “Traveller” field in Group Rows.
- Turn the toggle off for the subtotal button.



5. Click on save.

6. Save the report as “Bookings with Flight Details”.



7. Click on Save.

What Are Dashboards in Salesforce?

A Dashboard in Salesforce is a visual representation of one or more reports. It helps stakeholders quickly understand patterns, insights, and performance metrics using graphs, charts, and gauges.

Dashboards are ideal for:

- Decision-making
- KPI monitoring
- Operational visibility

Each component in a dashboard is based on a specific report and can be displayed as:

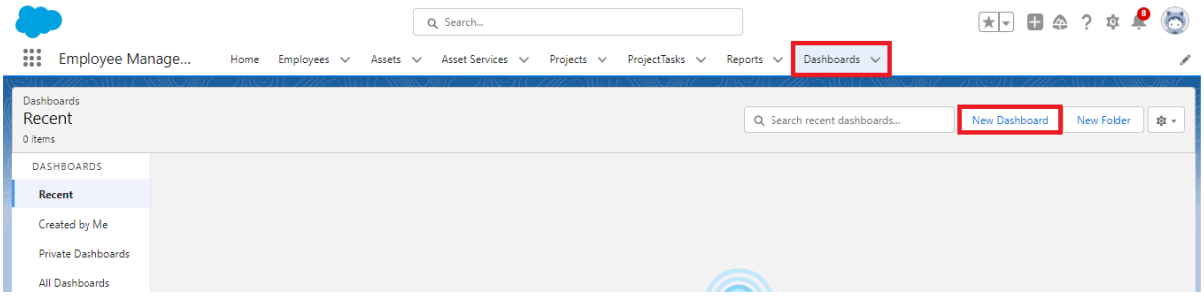
- Donut Chart
- Bar Graph
- Line Graph
- Metric Card
- Table

Dashboards Created in This Project:

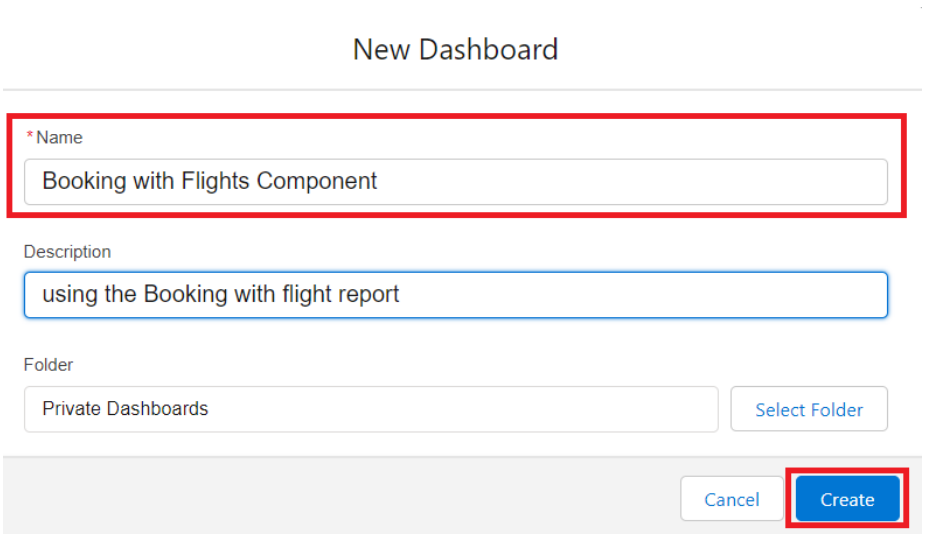
1. Booking Overview Dashboard
    - Components:
        - Donut chart for Booking Class Distribution (Economy, Business, First)
        - Bar graph for Number of Bookings per Flight
        - Record Count card for total bookings
2. Crew Assignment Dashboard
    - Components:
        - Table showing Crew members with assigned flights
        - Bar chart representing number of crew per flight
        - Highlight for under-assigned flights (optional)
3. Passenger Insights Dashboard
    - Components:
        - Pie chart showing Passenger Nationalities
        - Metric cards for Total Passengers, Unique Flight.

Create Dashboard

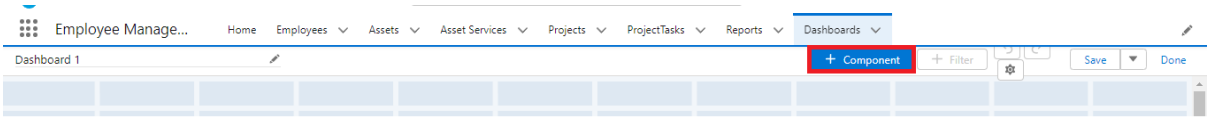
1. Go to the app --> click on the Dashboards tabs.



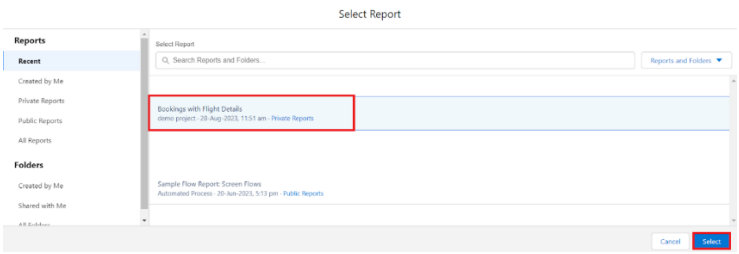
2. Give a Name and click on Create.



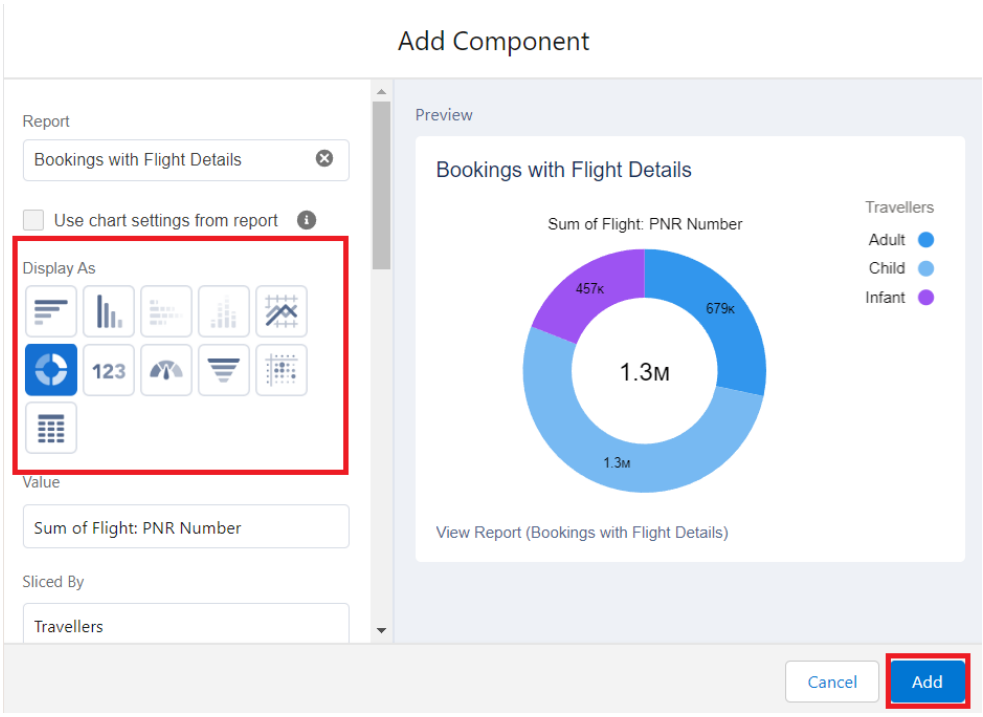
3. Select add component.



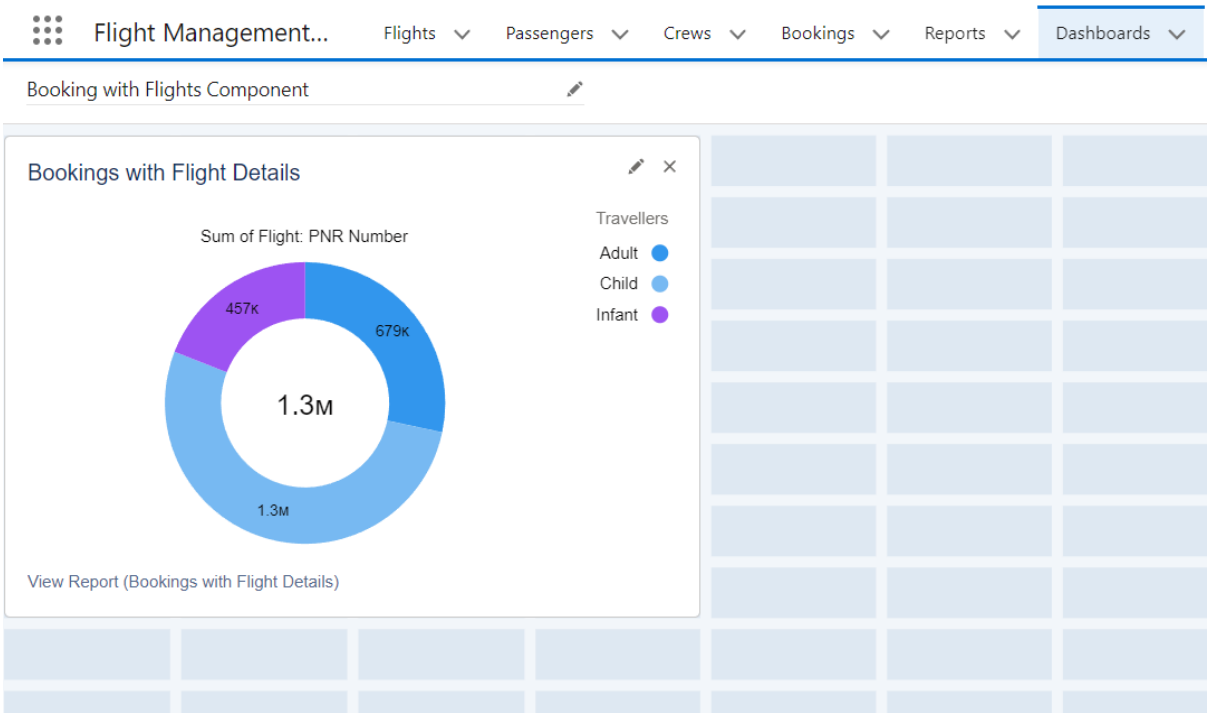
4. Select a Report and click on select.



5. Select the Donut option under display as and click on Add.



6. Your dashboard display as below:



8. ADVANTAGES & DISADVANTAGES

The Airlines Management System was built entirely on the Salesforce Lightning Platform, utilizing a combination of declarative (low-code) and programmatic (Apex) development methods. As with any technology solution, the implementation has both strengths and limitations. This section provides an in-depth analysis of the system's capabilities and its potential drawbacks based on the project scope, platform constraints, and real-world scenarios.

8.1 Advantages

1. Cloud-Based Accessibility

Being built on Salesforce's cloud infrastructure, the system is accessible from any internet-enabled device — desktop, laptop, or mobile. Users such as airline admins or booking agents can log into the system from anywhere, making it highly suitable for decentralized teams and remote operations. There is no need for on-premises server maintenance or installation, which reduces cost and overhead.

2. Low-Code Development Approach

Salesforce offers powerful tools like Flow Builder, Object Manager, and Schema Builder that enable rapid application development without the need for extensive coding. This allowed the project to be completed efficiently by focusing more on logic and structure than syntax. Even users with minimal programming knowledge can customize fields, create objects, and automate workflows.

3. Built-in Security and Role-Based Access

The system takes full advantage of Salesforce's security model, which includes:

- Profiles (e.g., General Admin, Crew Member)
- Roles (e.g., CEO → Senior Admin → Crew)
- Permission Sets

These features ensure that users only see and edit the data relevant to their roles, reducing security risks and enforcing data confidentiality across departments.

4. Real-Time Reports and Dashboards

One of Salesforce's standout features is its ability to generate visual reports and dashboards in real time. In the Airlines Management System, this enables:

- Monitoring of booking trends
- Class-wise passenger distribution

- Crew assignments and workload

Decision-makers can use these dashboards to assess performance metrics and make operational adjustments immediately, without needing to export data or use external tools.

5. Scalability and Flexibility

The modular nature of the system — with separate objects for Flight, Passenger, Booking, and Crew — allows it to scale easily. More fields, automation flows, and user roles can be added as needed. This design flexibility ensures the application can evolve with growing operational needs.

6. Automation and Validation

Key processes are automated through Flows and Apex triggers, significantly reducing manual data entry errors. For example:

- A Flow collects user input during booking
- An Apex Trigger validates phone numbers on Passenger creation

This built-in logic enhances reliability and accuracy across all records.

7. Maintenance and Future Enhancements

System maintenance is simplified due to Salesforce’s declarative design. Any admin with the proper profile can:

- Add fields
- Modify layouts
- Update flows
- Generate new reports

This eliminates the need for backend redeployment, making the system future-ready and easy to update over time.

8.2 Disadvantages

1. Platform Lock-In

The system is heavily tied to the Salesforce platform. While this ensures stability and access to Salesforce's features, it also means that moving the system to another platform (e.g., AWS, custom MERN stack) would require a complete redevelopment from scratch.

2. Limited Offline Functionality

As a cloud-based system, the Airlines Management System is inaccessible without an internet connection. For airline offices located in low-connectivity regions or during network outages, this can interrupt operations. There is limited offline access via Salesforce mobile apps, but core functions like flow execution and report viewing may be hindered.

3. Licensing Costs in Real-World Use

While the Developer Org is free for academic and learning use, deploying the system in a real-world commercial environment would require Salesforce licenses — which can be expensive depending on the number of users and features needed (e.g., API usage, storage, advanced analytics).

4. Customization Limitations for Complex Logic

Salesforce's declarative tools are powerful, but highly complex scenarios (e.g., integrating flight tracking APIs, generating QR-coded boarding passes, or building multi-language support) may require advanced Apex development or third-party integrations, which increases project complexity and maintenance overhead.

5. Daily Limits on API and Storage

Salesforce imposes daily API request limits, record storage limits, and data usage quotas. These are sufficient for academic and small-scale projects but could become a bottleneck in enterprise-scale deployments where thousands of records or external integrations are involved.

9. CONCLUSION

The Airlines Management System was successfully designed and implemented using the Salesforce Lightning Platform, serving as a cloud-based solution for streamlining airline operations such as flight scheduling, passenger management, crew assignments, and booking workflows. This project demonstrates how declarative tools combined with programmatic logic can deliver a powerful, secure, and scalable enterprise-grade application.

From ideation to execution, the project followed a structured, phase-wise development model — beginning with problem identification and empathy mapping, progressing through requirement analysis, system design, and ending with testing and deployment. Each module, including Flight, Passenger, Booking, and Crew, was implemented as a custom object in Salesforce and enhanced through custom fields, automation flows, and Apex triggers.

Key deliverables such as Screen Flows, Validation Rules, Apex Test Classes, and Real-Time Dashboards ensured the solution was not only functional but also user-friendly and reliable. The use of Profiles, Roles, and Permission Sets provided robust access control, aligning the system with modern security standards.

Through detailed testing, including Apex-based performance validation and flow testing, the system was confirmed to be stable and ready for real-world usage. Reports and dashboards provided valuable insights into booking trends, passenger demographics, and operational efficiency — showcasing the system's potential as a decision-making tool for airline administrators.

The project not only solved the original problem of disconnected, manual airline processes but also proved the power and adaptability of low-code development using Salesforce. It emphasized the importance of data-driven management, user-centric design, and continuous improvement in enterprise applications.

Overall, this project represents a significant achievement in applying academic learning to solve real-world operational challenges using modern cloud technology.

The Airlines Management System project was conceptualized, designed, and successfully implemented using the Salesforce Lightning Platform. It addressed real-world challenges in airline operations by offering a centralized, cloud-based solution for handling flight records, passenger bookings, crew assignments, and performance analytics.

By combining Salesforce's declarative tools such as Flow Builder, Schema Builder, and Report Dashboards with programmatic logic through Apex Classes and Triggers, the project achieved a balance between low-code simplicity and backend automation.

Through multiple phases — from ideation and requirement analysis to testing and reporting — the system evolved into a secure, scalable, and role-driven platform that meets modern aviation administration needs. Thorough performance testing and data validation ensured reliability and compliance with platform standards.

This project demonstrates how cloud-native platforms like Salesforce can empower non-traditional developers and student teams to deliver enterprise-grade solutions using thoughtful planning, structured design, and user-focused implementation.

Key Takeaways

Aspect	Summary
Platform Used	Salesforce Lightning Experience
Main Modules	Flight, Passenger, Booking, Crew
Automation	Screen Flows (Bookings), Apex Trigger (Phone Validation)
Security Model	Roles, Profiles, and Permission Sets
Testing	Apex Test Class with >75% coverage, Flow Testing, Manual UI Validation
Visual Analytics	Dashboards and Reports for booking trends and crew assignments
Development Approach	Phase-wise with weekly planning and deliverables
Performance	Validated logic under simulated conditions using test methods
Outcome	Delivered a functional, scalable, and real-time airline management system

10. FUTURE SCOPE

As the Airlines Management System reaches its current operational maturity, there is significant potential to expand its functionality, intelligence, and connectivity. This section explores advanced features and real-world integrations that could evolve the project into a complete enterprise-level airline operations platform.

1. Real-Time Flight Tracking API Integration

Description:
Integrating external APIs such as FlightAware, AviationStack, or OpenSky Network can allow the system to fetch real-time data related to:

- Flight departure and arrival times
- Delay notifications
- Live aircraft location

Benefit:
Users (both staff and passengers) would no longer rely on third-party sources to check flight status — all updates would appear inside the Salesforce app. This improves operational visibility, reduces manual tracking, and enhances passenger experience.

Implementation Insight:
Use Salesforce's Named Credentials and External Services to make secure REST API calls and display the results on custom pages or flows.

2. Mobile App Extension with Salesforce Mobile SDK

Description:
Developing a mobile version of the system using the Salesforce Mobile SDK or Lightning Mobile App Builder will allow real-time access to:

- Booking records
- Flight assignments
- Dashboards
- Approvals and alerts

Benefit:
Mobile access increases efficiency, especially for airline staff on the move such as:

- Crew checking flight details
- Admins approving bookings
- Management tracking operations remotely

Implementation Insight:
Enable the existing Lightning App for Salesforce1 Mobile App, and customize mobile layout via compact layouts and quick actions.

3. Chatbot for Customer Service

Description:
Integrating a chatbot using Salesforce Einstein Bots or Service Cloud will help automate passenger support tasks like:

- Checking booking status
- Responding to FAQs
- Providing flight updates

Benefit:
Reduces manual workload for staff, ensures 24/7 availability, and improves customer satisfaction by providing instant responses.

Implementation Insight:
Use Salesforce Digital Engagement to deploy bots on web pages, portals, or social channels like WhatsApp or Messenger.

4. Predictive Analytics with Salesforce Einstein

Description:
Use Salesforce's AI engine — Einstein Analytics / CRM Analytics — to analyze historical booking and crew data for:

- Predicting peak travel times
- Suggesting optimal crew rotations
- Forecasting flight occupancy

Benefit:
Supports data-driven decisions, improves planning accuracy, and optimizes resource utilization.

Implementation Insight:
Upload historical datasets, build prediction models in Einstein Discovery, and visualize results in dashboards.

5. Payment Gateway Integration

Description:
Integrate a payment gateway such as Razorpay, Stripe, or PayPal to process online ticket payments within the booking flow.

Benefit:
Allows passengers to complete their booking journey in one system without switching platforms. It improves convenience and revenue tracking.

Implementation Insight:
Use Apex callouts and Lightning components to connect to payment APIs and display confirmation receipts.

6. Crew Performance & Attendance Tracking

Description:
Add functionality to:

- Log crew attendance
- Track flight hours per crew member
- Collect performance ratings from supervisors or passengers

Benefit:
Helps HR and Operations teams evaluate performance and ensure regulatory compliance for flight hours and rest periods.

Implementation Insight:
Use custom objects like Attendance__c, and link it to Crew__c using lookups. Reports can summarize attendance frequency and trends.

7. Multi-Language & Regional Support

Description:
Enable localization support to provide the user interface in different languages, especially for international airline staff or passengers.

Benefit:
Increases accessibility and improves usability for diverse language speakers — critical for international operations.

Implementation Insight:
Use Salesforce's Translation Workbench to manage UI translations for object names, picklist values, labels, and messages.

8. Advanced Role-Based Dashboards

Description:
Create dashboards tailored to each role:

- CEO: High-level KPIs
- Manager: Operational performance

- Crew Member: Assigned flight list

Benefit:
Delivers personalized and relevant data views, improves decision-making, and prevents information overload.

Implementation Insight:
Use dynamic dashboards with “View as” filters and assign dashboard folders with visibility controls based on roles.

9. API-Based Data Exchange with Airports

Description:
Enable bi-directional data exchange with airport systems via APIs for:

- Gate allocation
- Boarding status updates
- Luggage tracking

Benefit:
Improves operational coordination between airline and airport, reduces delays and errors in real-time logistics.

Implementation Insight:
Use Salesforce Platform Events or External Services to push and pull data between systems.

10. Deployment to Production

Description:
Move the solution from the Salesforce Developer Org to a Production Org, enabling:

- Use by actual airline employees
- Integration with real passengers and flights
- Security hardening and scalability

Benefit:
Prepares the project for real-world usage, including data compliance, monitoring, and enterprise-level configuration.

Implementation Insight:
Use Change Sets or Salesforce DX to migrate metadata. Ensure data encryption, backups, and audit logging are in place.

11. APPENDIX

GITHUB url - https://github.com/chandini-190404/airlines-management-system.git

DemoLink url - https://youtu.be/KDxCtWJsv8w?si=ZtcAad5pmcog9GAX