Airline Reservation Application Test Plan

1. Test Plan Overview

- **Application Name**: Airline Reservation System
- **Version**: 1.0
- **Test Lead**: Sourabh Jain
- **Test Team**: Ravi, Ajay, Rahul, Vijay.
- **Test Duration**: 05-05-2025 to 25-05-2025
- **Test Environment**: UAT, PREPROD, INTEGERATED PREPROD

2. Test Objectives

- Validate core functionalities: flight search, booking, payment, and confirmation.
- Ensure system performance under varying loads.
- Verify security measures to protect user data.
- Assess usability and user interface responsiveness.
- Confirm compatibility across different devices and browsers.

3. Scope of Testing

In-Scope

- Flight search and filtering by date, destination, and airline.
- Booking process including passenger details and seat selection.
- Payment gateway integration and transaction processing.
- Email and SMS confirmation notifications.
- User account creation and management.
- System performance and load handling.
- Security protocols for data encryption and user authentication.

Out-of-Scope

- Backend database performance tuning.
- Third-party API performance beyond basic integration.
- Non-functional aspects unrelated to user interaction (e.g., server-side logging).

4. Test Methodology

To ensure the aircraft complies with all safety, regulatory, functional, and performance standards.

5. Approach

Agile and Spiral model

6. Test Types

Functional Testing

- **Flight Search**: Verify that users can search for flights based on various parameters such as origin, destination, date, and number of passengers. Ensure that the system returns accurate and relevant results.
- **Booking Process**: Test the booking process to ensure users can select flights, enter passenger details, choose seats, and complete payments successfully.
- **Payment Processing**: Validate that the system accurately calculates fares, applies taxes, and processes payments through integrated gateways.
- **Confirmation Notifications**: Ensure that users receive email and SMS confirmations with correct flight details and booking references.

Non-Functional Testing

- **Performance Testing**: Assess the system's response time and stability under different loads. Test the system's ability to handle a large number of concurrent users and transactions.
- **Security Testing**: Verify that sensitive data, such as personal information and payment details, are encrypted. Test for vulnerabilities like SQL injection and cross-site scripting (XSS).
- **Usability Testing**: Evaluate the user interface for ease of navigation, clarity of information, and overall user experience.
- **Compatibility Testing**: Ensure the application functions correctly across various browsers (e.g., Chrome, Firefox, Safari) and devices (e.g., desktops, tablets, smartphones).

Regression Testing

- Re-test previously validated functionalities after updates or bug fixes to ensure that new changes have not adversely affected existing features.

Integration Testing

- Test interactions between the reservation system and external services such as payment gateways, email/SMS providers, and Global Distribution Systems (GDS).

7. Test Deliverables

- **Test Plan Document**: Outlines the testing strategy, objectives, scope, and resources.
- **Test Cases**: Detailed scenarios covering all aspects of the application.
- **Test Scripts**: Automated scripts for repetitive or complex test scenarios.
- **Test Data**: Sample data sets for testing various functionalities.
- **Test Execution Reports**: Logs and results from test executions.
- **Defect Reports**: Documentation of identified issues and their resolution status.
- **Final Test Summary Report**: A comprehensive overview of testing activities, results, and recommendations.

8. Test Schedule

Phase	Start Date	End Date
Test Planning	5-05-2025	15-05-2025
Test Design	16-05-25	20-05-2025
Test Execution	21-05-2025	24-05-2025
Defect Fixes & Retesting	24-05-2025	27-05-2025
Final Reporting	28-05-2025	31-05-2025

9. Resources Required

- **Hardware**: Servers for hosting the application, testing machines with various configurations.
- **Software**: Testing tools (e.g., Selenium for automation, JMeter for performance testing), browsers for compatibility testing.

- **Human Resources**: Testers with expertise in functional, performance, and security testing.

10. Risk Management

- **Potential Risks**:
  - Delays in third-party service integration (e.g., payment gateways).
  - Limited access to real-time flight data for testing purposes.
  - Resource constraints leading to testing delays.
- **Mitigation Strategies**:
  - Establish early communication with third-party vendors.
  - Use mock data where real-time data is unavailable.
  - Allocate additional resources if necessary to meet deadlines