Test Plan for Airline Application

1. Test Plan ID

TP-ARS-001

2. Introduction

This test plan outlines the strategy, objectives, schedule, scope, resources, and deliverables for testing the Airline Reservation System. It ensures that booking, payment, cancellation, managing passenger information, handling payments and check-in features work as intended**.**

3. Objectives

- Verify core functionalities: flight search, booking, payment, and ticket generation.
- Validate user roles: admin, customer, agent.
- Ensure system handles errors and edge cases gracefully.
- Confirm system performance and security.

4. Test Strategy:-

- To verify that all core functionalities flight search, booking, payment, ticket generation, cancellation, and check-in—work as intended.
- To ensure the system handles valid and invalid user inputs gracefully without crashes or data corruption.
- To validate the accuracy of fare calculations, discounts, and tax components.
- To ensure a seamless user experience across devices (mobile, tablet, desktop).
- To confirm integration with external services such as payment gateways, email/SMS notifications, and airline APIs.
- To test system performance under normal and peak loads.
- To identify and fix security vulnerabilities that could expose user or financial data.
- To deliver a bug-free, secure, and user-friendly product before UAT and production deployment.

5. Scope

In Scope:

- Search and book flights
- Payment processing
- Ticket cancellation and refund
- Seat selection and check-in
- User registration and login
- Admin flight management

Out of Scope:

- Third-party airline integrations
- In-flight entertainment system

6. Test Items

- Flight Search Module
- Booking Module
- Payment Gateway Integration
- Check-in Module
- Cancellation and Refund
- User Login/Signup
- Admin Dashboard

7. Test Types

- Functional Testing
- Integration Testing
- UI/UX Testing
- Performance Testing
- Security Testing
- Regression Testing

8. Test Approach

- Manual testing for UI and UX
- Automated tests for regression and API
- Smoke tests on every deployment
- Exploratory testing for edge cases

9. Entry and Exit Criteria

Entry Criteria:

- Code is deployed to test environment
- Test data is ready
- All dependencies are resolved

Exit Criteria:

- 95%+ test cases passed
- No critical/high severity defects
- Regression is successful

10. Test Deliverables

- Test Plan Document
- Test Cases (Excel/Test Management Tool)
- Bug Reports
- Test Summary Report
- Automation Scripts (if any)

11. Resources

- Test Manager
- QA Engineers
- Developer (for support)

12. Schedule

Phase	Start Date	End Date
Planning	May 20	May 21
Test Case Writing	May 22	May 24
Test Execution	May 25	May 31
Retesting & Closure	June 1	June 3
Summary Report	June 4	June 5

13. Risks and Mitigation

Risk	Mitigation Plan
Delay in development delivery	Escalate early, adjust testing window
Payment gateway issues	Use sandbox/test environments
Lack of test data	Create mock/test data in dev environment