Test Plan Document for Airline Application

1.Test Plan Identifier

TestPlan_Airline

2. Introduction

This test plan outlines the testing strategy for the Airline Application. The objective is to validate the functionalities including flight search, booking, payment, cancellations, check-in, and user account management.

3. Objectives and Tasks

- Verify all core modules work as expected.

- Ensure smooth user experience across devices and platforms.

- Validate business rules for booking and pricing.

- Test integration with payment gateways, airline inventory systems, and third-party APIs.

- Test edge cases and exception handling.

4. Scope

In Scope:

- Flight search (one-way, round-trip, multi-city)

- Booking and seat selection

- Payment processing

- Cancellation and refund

- Online check-in

5. Test Strategy

- **Unit Testing:** Done by developers.

- **Integration Testing:** Test interfaces with payment gateways, airline APIs.

- **System Testing:** End-to-end scenarios including all modules.

- **Regression Testing:** After each deployment.

- **UAT:** By business Owners.

- **Performance Testing:** Stress and load testing using tools.

6. Test Criteria

- **Entry Criteria:** All development is completed, test environment is ready, test cases reviewed.

- **Exit Criteria:** All critical test cases pass, major bugs resolved, sign-off obtained.

7. Test Environment

- Web and Mobile platforms

- Browsers: Chrome, Firefox, Safari, Edge

- Mobile OS: iOS, Android

8. Roles and Responsibilities

- **Test Manager:** Oversee testing, manage resources

- **Test Engineers:** Write and execute test cases

- **Developers:** Fix bugs, unit test

- **Business Analyst:** Review test coverage

9. Schedule

Phase	Start Date	End Date
Test Planning	May 8	May 10
Test Design	May 11	May 15
Test Execution	May 16	May 28
Bug Fix & Retest	May 29	June 5
UAT	June 6	June 12
Final Sign-off	June 13	June 15

10. Risks and Mitigations

- **Risk:** Payment gateway downtime
  Mitigation: Use mock/stub environment for testing

- **Risk:** Delay in test environment setup
  Mitigation: Set up parallel test environments early