**Name: Abhisek Rout**

# **Test Plan Document for Airline Reservation Application**

## **1. Introduction**

### **Purpose:**

The purpose of this document is to outline the strategy, objectives, resources, and schedule for testing the airline reservation application to ensure it meets quality and performance standards before release.

### **Scope:**

This test plan covers the functional and non-functional testing of the airline reservation application including:

* Flight search and booking
* Payment processing
* User authentication
* Check-in
* Flight cancellation/refund
* Loyalty program
* Mobile and web platform compatibility

**Out of Scope:**

* Backend airline operations (e.g., crew scheduling)
* External partner systems (unless directly integrated)

## **2. Test Objectives**

* Validate that all business-critical workflows perform as expected.
* Ensure system stability, performance, and security.
* Identify and eliminate bugs or performance bottlenecks before deployment.

## **3. Test Items**

* User Registration/Login Module
* Flight Search Module
* Booking and Seat Selection Module
* Payment Gateway Integration
* E-ticket Generation and Email Notification
* Online Check-in
* Flight Cancellation and Refund Processing
* Loyalty Points Management
* Admin Dashboard (for managing flights, pricing, users, etc.)

## **4. Features to Be Tested**

* Search flights with filters (date, destination, number of passengers)
* Book selected flights (including seat selection)
* Process payment through different methods (credit card, wallet, UPI)
* User registration and secure login (including multi-factor authentication)
* Mobile responsiveness (iOS and Android)
* Generate and email e-tickets
* Check-in functionality with seat changes
* Cancel booking and process refund
* Earn/redeem loyalty points
* Admin updates to flights/schedules/pricing

## **5. Features Not to Be Tested**

* Third-party flight APIs (assumed functional from provider)
* Hardware integration (e.g., airport kiosk printers)
* Localization beyond English (not in current scope)

## **6. Test Approach**

### **Functional Testing:**

* Manual testing of workflows (booking, check-in, cancellation)
* Positive and negative test cases

### **Regression Testing:**

* Automated regression suite for core functionalities after every build

### **Performance Testing:**

* Load testing for concurrent users during peak booking hours

### **Security Testing:**

* Penetration testing
* Session timeout and data encryption validation

### **Compatibility Testing:**

* Across devices (smartphones, tablets, desktop)
* Across browsers (Chrome, Firefox, Safari, Edge)

## **7. Test Environment**

* Staging environment simulating production
* Realistic user and booking data
* BrowserStack for cross-browser/device testing
* Database backup/restore enabled

## **8. Test Deliverables**

* Test Plan Document (this document)
* Test Cases and Test Scripts
* Test Summary Report
* Defect Logs
* Traceability Matrix

## **9. Schedule**

|  |  |  |
| --- | --- | --- |
| **Activity** | **Start Date** | **End Date** |
| Test Planning | May 6 | May 8 |
| Test Case Design | May 9 | May 12 |
| Environment Setup | May 10 | May 11 |
| Functional Testing | May 13 | May 20 |
| Regression & Performance | May 21 | May 25 |
| Bug Fix Verification | May 26 | May 28 |
| Final Test Report | May 29 | May 30 |

## **10. Resources**

* 1 Test Manager
* 3 QA Engineers
* 1 Automation Engineer
* 1 Performance Tester

## **11. Risks & Mitigation**

|  |  |
| --- | --- |
| **Risk** | **Mitigation Strategy** |
| Unstable API responses | Use mock servers for early testing |
| Delay in development handover | Parallel test case design using requirements |
| Browser/device compatibility issues | Use cloud-based device testing tools |