**Requirement Traceability Matrix**

**Project:- FlightSure QA Sprint (48 Hours)**

**Team Name:- The Debugging Knights**

**Approved By:- Pooja Yadav**

**Date:- 18 May 2025**

**Context:**

You are part of a QA team at **FlightSure**, a new-age startup launching a **flight + hotel booking platform**. The MVP is live in staging, but riddled with **critical bugs, performance bottlenecks, flaky UIs, and insecure APIs**.

The product is going live in 3 days. You’ve been brought in to **test, report, and automate across UI + API layers**, ensuring maximum test coverage and quality assurance in **48 hours**.

**Team Size:**

2 Members (must include: Manual Tester, Automation Tester, API Tester, Report Owner, Performance Lead)

**Application Under Test (A mock system):**

Use any of the following (or organizer-provided system):

* <https://phptravels.com/demo/>
* (Optionally: A custom-hosted travel booking system)

**Problem Statement – What You Must Do:**

**Part 1: End-to-End Functional and Exploratory Testing**

* Explore the entire booking flow:
  + Search → Filter → Select Flight/Hotel → Add Details → Payment (dummy flow).
* Identify **critical bugs** across user journeys (functional + UI + UX).
* Cover **edge cases** like:
  + Invalid dates, passengers = 0, city names with special characters, etc.
* **Deliverables**:
  + Test Scenarios + Cases (Spreadsheet or TestRail)
  + Bug Report with evidence, impact level, reproducibility steps
  + Exploratory Testing Mind Map (Optional but bonus)

**Part 2: Automation with Selenium/Cypress + REST Assured/Postman**

**UI Automation:**

* Automate:
  + Search Flights/Hotels
  + Selection and checkout
  + Validations for filters, prices, passenger input, etc.
* Handle:
  + Dynamic waits
  + Pagination (if present)
  + Screenshots on failure

**API Automation:**

* Identify API calls using DevTools or Swagger
* Test APIs for:
  + Search
  + Booking
  + Cancellation
* Include:
  + Positive + Negative tests
  + Schema validations
  + Response time assertions

**Part 3: Performance and Load Testing (Bonus but Weighted)**

* Use **JMeter** or **k6** to simulate:
  + 200 users searching for flights simultaneously
  + 50 users performing booking concurrently
* Identify performance bottlenecks:
  + Response time
  + Server errors
  + Recommendations for optimization

**Part 4: Collaborative Twist (Live Product Change)**

* Midway through the hackathon (e.g., Hour 20), the product team will announce a **UI or API change**:
  + New validation
  + Field added/removed
  + Endpoint structure changed
* Your team must:
  + Re-test impacted flows
  + Update automation scripts
  + Provide **Impact Report + Fix Plan** within 3 hours

**Part 5: Business + UX Review (Critical)**

* From the perspective of a **business stakeholder**, identify:
  + Broken flows hurting conversion
  + Bad UX (confusing search, CTAs, layout issues)
* Suggest:
  + At least **5 improvements** with business reasoning

**✅ RTM Summary**

* **Total Requirements**: 8
* **Fully Covered**: 6
* **Partially Covered**: 2 (UI responsiveness, Edge cases)
* **Total Bugs Logged**: 5
* **Traceability**: 100% of functional requirements are mapped to test scenarios and test cases

**🧾 Remarks**

* All **critical business flows** are traceable from requirement to execution to defect.
* Defects logged are traceable to respective test cases and test scenarios.
* **Edge case handling** and **responsive UI** need further attention.
* Ensure regression after bug fixes to maintain traceability continuity.

**Final Deliverables:**

|  |  |
| --- | --- |
| Category | Expected Artifacts |
| Functional Testing | Test Cases, Bug Reports, Exploratory Notes |
| UI Automation | GitHub Repo + README |
| API Automation | Postman Collection or REST Assured Project |
| Performance | JMeter/k6 Report |
| Collaboration | Slack/Discord screenshot, Change Impact Report |
| Final Submission | 5–7 min Demo Video, Executive Summary PDF |