Test Plan for Katalon Demo Application to Make Appointment

Created by: *Poonam Prasad*

---

1. Objective

This document outlines the test plan for the **Katalon-demo-cura.herokuapp** application.
The objective is to ensure that all features and functionalities work as expected for the target audience, which includes **patients and healthcare staff making online appointments** via the web interface.

---

2. Scope

- **Features to be tested:**
  - Login and logout functionality
  - Appointment scheduling
  - Facility selection
  - Healthcare program selection
  - Date and comment input
  - Appointment confirmation
  - Navigation between pages
  - UI layout responsiveness
- **Types of Testing:**
  - Manual testing
  - Automated testing (using Selenium/Katalon Studio)

  - Performance testing
  - Accessibility testing
- **Environments:**
  - Browsers: Chrome, Firefox, Edge
  - Operating Systems: Windows 10, macOS, Linux
  - Devices: Desktop, laptop, tablet, mobile
- **Evaluation Criteria:**
  - Number of defects found
  - Severity and priority of bugs
  - Time taken for execution and resolution
  - Successful completion of all test cases
- **Team Roles and Responsibilities:**
  - **QA Lead (1):** Test planning, coordination, and reporting
  - **Test Engineers (4):** Test case creation, execution, bug logging

---

3. Inclusions

- **Introduction:** Test plan to ensure that the core appointment system works smoothly across platforms
- **Test Objectives:**
  - Ensure users can make appointments without errors
  - Validate login system with positive/negative flows
  - Confirm UI responsiveness and cross-browser compatibility

    - Identify and report issues affecting user experience

---

4. Exclusions

- Third-party integrations (e.g., email confirmation or backend database operations)
- Performance testing beyond agreed threshold
- Browser versions older than 2 years

---

5. Test Environments

Component	Details
Operating Systems	Windows 10, macOS Monterey, Ubuntu 22.04
Browsers	Chrome (latest), Firefox (latest), Edge
Devices	Desktop, Laptop, Tablet, Mobile
Network Connectivity	Wi-Fi, Ethernet, 4G
Tools Required	Chrome DevTools, Katalon Studio, Postman
Security Protocols	Username/password, HTTPS
Access Permissions	All testers have login credentials

---

6. Defect Reporting Procedure

- **Defect Criteria:** Any deviation from functional or UI expectations
- **Steps for Reporting:**
    - Document issue in JIRA using standard bug template
    - Provide clear reproduction steps and screenshots

    - Assign severity (Blocker, Critical, Major, Minor)
- **Triage:** QA lead reviews, prioritizes, and assigns bugs
- **Tools:** JIRA for tracking, Slack/email for updates
- **Metrics:** Open vs closed defects, turnaround time, re-open rate

---

7. Test Strategy

Step 1: Test Scenarios and Test Cases

- **Techniques Used:**
    - Equivalence Partitioning
    - Boundary Value Analysis
    - Use Case Testing
    - Exploratory Testing
- **Test Scenarios:**
    - Successful login
    - Invalid login attempt
    - Appointment form filled correctly
    - Validation when leaving fields empty
    - Appointment date picker functionality
    - Drop-down selections for facility
    - Selecting different healthcare programs

Step 2: Testing Procedure

- **Smoke Testing:** Initial check for login and appointment page loading
- **In-depth Testing:** Detailed test case execution post build
- **Cross-browser & Cross-device Testing:** Parallel testing
- **Defect Logging:** JIRA, with daily bug triage meetings
- **Types of Testing:**
  - Smoke, Sanity, Regression, Functional, Usability

Step 3: Best Practices

- Shift Left Testing – Early involvement in development cycle
- End-to-End Scenario Testing
- Retesting and Regression testing
- Exploratory testing in parallel with scripted cases

---

8. Test Schedule

Task	Duration	Start Date	End Date
Test Plan Preparation	1 day	1 May	1 May
Test Case Design	2 days	2 May	3 May
Test Execution	7 days	4 May	10 May
Defect Reporting/Retests	3 days	11 May	13 May
Final Summary Reporting	1 day	14 May	14 May

---

9. Test Deliverables

- Test Plan Document

- Test Scenarios and Test Cases (Excel)
- Defect Reports (JIRA)
- Daily Execution Logs
- Final Test Summary Report

---

10. Entry and Exit Criteria

Requirement Analysis

- **Entry:** Access to application URL, understanding of workflow
- **Exit:** Finalized test scenarios and test plan approval

Test Execution

- **Entry:** Approved test cases, stable build deployed
- **Exit:** Execution complete, all high/critical defects closed

Test Closure

- **Entry:** Final test case execution complete
- **Exit:** Summary reports submitted and approved

---

11. Tools

- JIRA – Bug tracking
- Excel – Test case documentation
- Katalon Studio – Automation

- Postman – Optional API testing
- Snipping Tool / Screenshot apps – Evidence gathering

12. Risks and Mitigations

Risk	Mitigation Strategy
Application goes down	Inform Devs, test offline flows, monitor uptime
Incomplete requirements	Schedule quick clarifications
Tight deadlines	Prioritize critical flows, parallel execution
Browser/device compatibility	Use BrowserStack or test labs

13. Approvals

Documents for Client Approval:

- Test Plan
- Test Scenarios
- Test Cases
- Test Execution and Summary Reports