**Test Plan Document**

**Project Name : CURA Healthcare Service**

1. **INTRODUCTION:**

The CURA Healthcare Service is a web application that allows the user to login, schedule appointments with doctors and secure logout from the application. The test plan mainly focuses on validating the functionality, reliability and security of the core APIs namely:

* Login
* Make Appointment
* Logout

1. **TEST OBJECTIVES:**

The objective of this API testing is:

* Verify whether the user is able to successfully login with valid credentials.
* Ensure the user is able to make appointment with doctors through the Make Appointment API.
* Verify the users are able to logout properly and their sessions are securely terminated.
* Ensure the APIs return the correct HTTP status codes, response bodies and error messages for valid and invalid scenarios.
* Validate security aspects such as session handling and access control.

1. **SCOPE:**

**In Scope:**

Functional Testing for below APIs

* Login API
* Make Appointment API
* Logout API

Validation Features are

* HTTP methods and status codes
* Request / Response structure
* Authentication and Authorization (tokens)
* Error handling for invalid inputs and unauthorized access.

**Out of Scope:**

* UI Testing
* Database Validations.
* Test Strategy

1. **Test Strategy:**

**Manual Testing:**

* Use **POSTMAN** to test endpoints with various inputs using POST method.
* Validate request and response status codes, headers and response body.

**Automation Testing:**

* Use **Rest Assured** with **TestNG /POSTMAN.**

**Types of Testing**:

* **Functional Testing:** Validating the core functionalities of Login, Make Appointment and Logout APIs.
* **Positive Testing:** To verify that all APIs function correctly with valid and expected input data.
* **Negative Testing:** Verifying with invalid or incomplete data to test error handling.
* **Security Testing:** Verifying that unauthorized access is properly restricted.
* **Authentication Testing:** Validating session handling and token-based access controls.
* **Data Validation:** Validating the response body content, formats, and required fields.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| * **Boundary Testing:** Testing with edge case data. * **Error Handling Testing:** Verifying API returns appropriate HTTP status codes and error messages.  1. **Tools:**  * Manual: POSTMAN * Automation: Rest assured + TestNG * Test Execution: Allure Report * CI/CD: Jenkins * Defect Tracking: JIRA * Test Documentation: Google Docs / Excel.  1. **Test Scenarios:**   TC 01: Verify user is able to login with valid credentials.  TC02: Verify login fails by entering invalid credentials.  TC03: Verify appointment can be created with valid token.  TC04: Verify appointment can be created without authentication token.  TC05: Verify user is able to logout the session properly.  TC06: Verify user cannot access protected endpoint after logout.   1. **Test Environment:**  * URL:<https://katalon-demo-cura.herokuapp.com/> * Environment: Staging / QA * Authentication: Tokens  1. **Test Execution Plan:**  |  |  |  | | --- | --- | --- | | **Activity** | **Start Date** | **End Date** | | Test Case Design (Manual) | To Update | To Update | | Manual Test Case Execution | To Update | To Update | | Automation Script Development | To Update | To Update | | Automation Test Execution | To Update | To Update | | Defect Verification | To Update | To Update |  1. **Roles & Responsibilities:** 2. QA: write both test cases, execute manual and automation test cases, review results and raise defects if found any. 3. Developer: Fix bugs and support QA as needed. 4. **Entry & Exit Criteria:**   **Entry:**   * APIs are stable and available. * API documentation is ready.   **Exit:**   * All critical and high priority test cases are passed. * No open bugs. * All test cases are automated and validated.  1. **Defect Reporting:**  * Defects are logged using JIRA. * It includes Request and Response headers, Authentication tokens, Steps to be reproduced, Version etc. * Priority and Severity based on the defect from Low to Critical. * Defect Review call based on the process (daily to weekly).        1. **Risks and Mitigation:**   **Risk:**  No API docs available.  Session handling methods  **Mitigation:**  Use browser dev tools.  Add delay / retry logic in automation.   1. **Approvals:**  |  |  |  | | --- | --- | --- | | **Role** | **Name** | **Signature / Approval** | | QA Engineer |  |  | | QA Lead / Manager |  |  | | Product Owner |  |  |  1. **References:**   **CURA Healthcare Service** |