

TEST PLAN DOCUMENTATION FOR MYSHUTTLE PROJCT

1. Introduction

MyShuttle is a sample Java application that provides booking system, admin portal and a control system for the drivers. The application uses entirely open source software including Linux, Java, Apache, and MySQL which creates a web front end, an order service, and an integration service.

2. Test Plan Objective

The objective of this test plan is to define the strategy, scope, and resources required for testing the MyShuttle application. The plan outlines the overall approach to ensure that the application meets the functional, non-functional, and quality standards set by the stakeholders.

3. Scope of Testing

- **Functional Testing (Login, Dashboard, User management):**
The testing will focus on login functionality, dashboard navigation, and user management, including user creation, updates, deletions, and role-based access control.
- **UI/UX Testing:**
UI/UX testing will focus on verifying the user interface for consistency, responsiveness, and ease of navigation. It will ensure that the design elements align with the intended user experience, providing a smooth and intuitive interaction across different devices and screen sizes.
- **Integration Testing (with Apache Tomcat, Docker, Database):**
Integration testing will focus on ensuring seamless interaction between Apache Tomcat, Docker, and the database. It will verify that the application is correctly deployed in the Docker environment, that Apache Tomcat serves the application as expected, and that the database is properly integrated for data storage and retrieval.

4. Test Objectives

The testing for MyShuttle will focus on:

- **Functional Testing:** Validating the core features of the application, including login, dashboard, and user management workflows to ensure proper functionality.
- **Usability Testing:** Ensuring the user interface is intuitive, easy to navigate, and provides a seamless user experience across different devices.
- **Performance Testing:** Evaluating the application's performance, including response time and stability, under normal and peak load conditions.
- **Security Testing:** Ensuring that user data, including personal and payment information, is securely handled and protected from vulnerabilities.
- **Compatibility Testing:** Verifying the platform's compatibility across various browsers (Chrome, Firefox, Safari, Edge) and devices (desktop, tablet, mobile).

5. Test Approach

Test Types:

- a. **Manual Testing:** Exploratory, functional, UI, and integration testing to cover a wide range of test cases manually.
- b. **Automated Testing:** Automation for repetitive, regression, and smoke test cases to ensure fast and consistent execution.
- c. **Performance Testing:** Load testing using tools like JMeter to verify the application's behavior under heavy load conditions.

Testing Levels:

- d. **Unit Testing:** Performed by developers to ensure individual components (e.g., login, user management) are working as expected.
- e. **Integration Testing:** Ensuring smooth integration of different modules like user authentication, dashboard, and database connectivity.
- f. **System Testing:** Verifying end-to-end functionality of MyShuttle, ensuring all modules work together.
- g. **User Acceptance Testing (UAT):** End-users validate the system to ensure it meets business requirements and expectations.

Testing Tools:

- h. **Test Management:** Azure DevOps Test Plans to organize and manage test cases, test runs, and reporting.
- i. **Automation Tools:** Selenium for functional UI testing, Postman for API testing, and JMeter for performance/load testing.
- j. **Bug Tracking:** Azure DevOps to log, track, and manage defects, ensuring they are resolved before release.

6. Test Deliverables

Test Cases: Comprehensive test cases covering all features of MyShuttle, including login, dashboard functionalities, user management, payment integration, and order management.

Test Execution Reports: Detailed reports showing the pass/fail status of each test case, including execution time, environment details, and any deviations from expected results.

Defect Logs: A detailed record of any issues identified during testing, including defect severity, steps to reproduce, environment details, and the status of each defect (open, resolved, closed).

Test Summary Report: A final report summarizing the testing effort, including the number of test cases executed, defect statistics, overall quality of the application, and the final approval for release.

7. Test Schedule

Phase	Start Date	End Date
Test Planning	2024-12-13	2024-12-16
Interface creation	2024-12-16	2024-12-17
Testcase Creation	2024-12-17	2024-12-17
Test Implementation	2024-12-18	2024-12-18
Verify Testcase	2024-12-19	2024-12-19
Test Completion	2024-12-19	2024-12-19

8. Resource Planning

- **QA Engineers (4):** Responsible for functional, usability, exploratory, and manual testing to ensure all features are thoroughly tested, including user workflows and platform behavior under various scenarios.
- **Automation Engineers (4):** Tasked with designing and executing automated tests, focusing on regression, smoke, and performance tests to streamline the testing process and ensure consistent results across multiple test cycles.
- **Project Manager (1):** Oversees the entire testing process, ensuring that testing activities are executed as per the plan, coordinating between team members, and guaranteeing timely completion of all testing tasks.

9. Test Environment

Hardware:

- a. Test workstations with **Windows** (Windows Server used for backend services).
- b. **Mobile devices** for compatibility testing (iOS and Android devices for ensuring responsiveness across various screen sizes and platforms).

Software:

- c. **Browsers:** Chrome, Firefox, Safari, Edge for cross-browser compatibility testing.
- d. **Automation Tools:** Selenium for functional and regression testing, JMeter for performance testing, Postman for API testing, LoadRunner for additional load and performance testing.
- e. **Test Management:** Azure DevOps for managing test cases, defects, and overall test planning.

Database:

- f. **Azure SQL Database** populated with test data for products, users, and orders, providing a realistic environment for testing features such as product catalog, user management, and order processing.

10. Risk and Mitigation

Risk	Impact	Mitigation Strategy
------	--------	---------------------

Incomplete Test Data	Delays and inaccurate results.	Prepare data in advance, create mock data.
Deployment Failures	Delays and instability.	Test deployment in dev, prepare rollback.
Compatibility Issues	Functionality may break on platforms.	Test early across browsers and devices.

11. Test Plan

Day	Task	Time Allocated	Responsibility
Day 1	Setup test environment, test data preparation	7 hours	QA Engineers, Automation Engineers
Day 2	Functional testing (Login, Dashboard, User Management)	7 hours	QA Engineers
Day 3	Integration testing (Tomcat, Docker, Database)	7 hours	QA and Automation Engineers
Day 4	Performance and Compatibility testing	7 hours	QA Engineers

12. Approval

Project Manager : Suresh nanjan

Testers: Linesh Mishra, Rajkiran Kadambalu, Sameera Mohammed, Samson Karre.