https://66405eedbcb7ce8c1a800ea6--timelyfroyo5c39a1.netlify.app/index.html

Project Code: B41_SDET-010_TestMavericks

Project Name: Big Basket

Test Plan Big Basket

Submitted By:

Vinita Kumawat(fs40_393032)

Date:17/12/2024 to 21/12/2024

Index Table

Project Information

Project Code	B41_SDET-010_TestMavericks
Website URL	https://66405eedbcb7ce8c1a800ea6--timelyfroyo5c39a1.netlify.app/index.html
Tester	Vinita Kumawat
Instructor	Anjali Shaw

➢ Introduction

The test plan is intended to outline the approach for testing the application hosted at Timely Froyo. This web application likely serves as a front-end platform for an online service or product (possibly related to frozen yogurt). The plan details how the software testing will be structured to ensure it meets the requirements and works as expected. This test plan outlines the software testing approach for the product page of the demo website. The primary goal is to ensure the page's functionality, performance, security, and usability meet the required standards. This test plan will help identify potential issues before the product page is deployed to production.

➢ Objective

The primary objective of this test plan is to verify that the web application functions as expected in various environments, is free from defects, and provides a seamless user experience. The testing will ensure that all features perform as intended, both in terms of functionality and usability.

➢ Scope

This testing effort will focus on the following key areas:

- **Functionality Testing:** Verifying that all features, including navigation, user interactions, forms, and buttons, work correctly.
- **Usability Testing:** Ensuring the user interface is intuitive, and the application is user-friendly.
- **Performance Testing:** Checking how the application performs under various load conditions.
- **Compatibility Testing:** Verifying the application works across different browsers and devices.

➢ Out of scope:

- Payment gateway testing (if applicable).
- Backend or database testing (if the application is static or doesn't involve a server-side component).

➢ Testable Features

- **Homepage functionality:** Links, buttons, images, and form actions.
- **Login Functionality**:
  - Valid/invalid credentials handling.
  - Error messages for empty or incorrect fields.
- **Sign-Up Functionality:**
  - Validation of mandatory fields, password strength, and email format.
  - The proper registration process and redirect to the login page.
- **Logout Functionality:**
- Session invalidation after logout.
- Session restoration prevention using the browser's back button.
- **Navigation:** Menu and page transitions.
- **User interactions:** Form submissions, dynamic content updates, and data validations.
- **Compatibility:** Testing on different browsers (Chrome, Firefox, Safari, Edge) and mobile responsiveness.
- **Error handling:** Ensuring proper error messages and redirection for invalid input or system failures.

➢ Testing Approach

The testing will be carried out using a combination of manual and automated techniques:

- **Manual Testing:** For functionality, usability, and compatibility checks.
- **Performance Testing:** Load testing using tools like Apache JMeter.
- **Security Testing:** Manual penetration testing to check for vulnerabilities.

➢ Roles & Responsibilities

Roles	**Responsibilities**

QA Lead	Oversee the testing process, ensure all phases are on track, and communicate with stakeholders.
Test Engineers	Execute test cases, log bugs, and report progress
Developers	Assist in resolving bugs, clarifying requirements, and supporting testing.
Project Manager	Ensure the testing schedule aligns with project timelines.
UI/UX Designers	Ensure usability tests align with design standards

➢ Test Schedule

The testing will be carried out as per the following schedule:

Days	**Schedule**
17/12/2024	Test Environment Setup, Test Case Design, and Initial Functionality Testing.
18/12/2024	Execution of functionality, usability, and compatibility testing.
19/12/2024 &20/12/2024	usability, and compatibility testing.
21/12/2024	Bug reporting, final testing, and closure.

s

➢ Test Deliverables

- **Test Plan Document:** Defines the scope, objectives, and approach.
- **Test Cases:** Detailed test cases for each feature.
- **Test Logs:** Records of tests executed, including results and issues encountered.
- **Bug Reports:** Detailed reports of identified defects with severity levels.
- **Test Summary Report:** A final report summarizing testing activities, results, and recommendations.

➢ Entry and Exit Criteria

➢ Entry Criteria

The entry criteria define the conditions that must be met before starting the testing process:

1. **Requirement Finalization:**

- All functional and non-functional requirements of the product page must be documented and finalized.
- This ensures that there is no ambiguity about the features being tested, such as product information display, buttons, and navigation functionalities.

2. **Test Plan:**
    - The test plan document must be prepared and approved.
    - It outlines the scope, approach, resources, roles, tools, and schedule to ensure organized testing.
3. **Availability of Test Artifacts:**
    - All necessary documents and inputs, such as requirements documents, wireframes, test data, and mock-ups, must be made available for reference during testing.
    - This helps testers validate the product page against the expected behaviour.
4. **Environment Readiness:**
    - The test environment (test server or staging environment) must be stable and configured to support testing.
    - The product page URL should be accessible, and the environment should mimic the production environment.
5. **Code Deployment:**
    - The product page code must be successfully deployed on the testing environment.
    - Any incomplete or unstable builds should not be provided for testing.
6. **Tool Accessibility:**
    - Testing tools, such as bug tracking tools (e.g., JIRA) and test management tools, must be accessible.
    - Tools for cross-browser testing and performance monitoring should also be available.
7. **Availability of Test Team:**
    - The test team members must be available and assigned roles to execute the planned test cases.
    - Any skill gaps should be addressed with proper guidance or training.
8. **Unit Testing Completion:**

- Developers must complete unit testing to ensure that individual components of the product page are working correctly.
- This minimizes the risk of discovering trivial bugs during system testing.

➢ Exit Criteria

The exit criteria define the conditions that must be satisfied to consider the testing phase complete:

1. **Test Execution:**
    - All planned test cases should be executed.
    - This includes functional, compatibility, exploratory, and basic performance testing for the product page.
2. **Defect Resolution:**
    - All critical and major defects identified during testing must be fixed and verified.
    - Minor defects can be deferred with proper documentation and stakeholder approval.
3. **Test Coverage:**
    - Ensure that the testing covers all product page functionalities, including UI, responsiveness, and interactive elements.
    - A high percentage of test case coverage (e.g., 95%+) should be achieved.
4. **UAT Completion:**
    - User Acceptance Testing (UAT) must be completed successfully.
    - This ensures the product page meets end-user expectations and is ready for release.
5. **Performance Metrics Met:**
    - The product page should meet performance criteria, such as page load speed, responsiveness, and smooth functionality.
    - Tools like Page Speed Insights can help validate this.
6. **Documentation Delivery:**
    - All testing-related documents, including test cases, defect reports, execution logs, and a test summary report, must be delivered to stakeholders.
7. **Regression Testing:**

    - Regression testing must be performed to ensure that any new changes or bug fixes do not impact existing functionalities.

8. **Approval from Stakeholder:**
    - Final approval must be obtained from stakeholders, such as the Product Manager or client, to confirm that testing is complete, and the product page is ready for deployment.

➢ Tools

- **Test Management Tool:** Jira or TestRail to track test cases and defects.
- **Bug Tracking Tool:** Jira for defect management.
- **Cross-browser Testing Tools:** Browser Stack or Sauce Labs.
    - **Google Sheet**
    - **Google doc**
    - **X Mind Map**

➢ Risks & Mitigation

- **Risk 1:** Limited browser compatibility.
    - **Mitigation:** Ensure compatibility testing is thorough across major browsers and devices.
- **Risk 2:** Delays in feature delivery.
    - **Mitigation:** Establish clear communication with the development team to track progress and prioritize features for testing.
- **Risk 3:** Incomplete test coverage.
    - **Mitigation:** Review test cases regularly to ensure all critical features are covered, including edge cases.

➢ Approvals

- **Test Plan Approval:** The test plan needs to be reviewed and approved by the QA lead, project manager, and relevant stakeholders before execution.
- **Test Case Approval:** The development team should review and approve test cases to ensure alignment with requirements.