**Test Plan for Recipe sharing platform**

**Prepared by:** Gintaras Jezepčikas, Vladimir Michailov, Liucija Ivanauskienė, Alina Trečiokė

**Introduction**

The objective of this test plan is to ensure that the recipe sharing platform is fully functional, user-friendly and meets the specified requirements.

**Test Scope**

* User account management
* User profile management
* JWT token and authorization
* Home page functionalities
* Recipe creation and submission
* Recipe page management
* Follow and follower's functionalities
* Admin functionalities
* Error page
* User Interface testing

**Test Objectives**

* Verify that users can successfully create accounts, log in, log out.
* Verify that the user can successfully add and edit the profile picture, update username, email and password. The user should be able to see a list of added recipes and favorite recipes.
* Verify that the access token is created, and the user can be successfully authorized.
* Testing the main page. Verify that the user can successfully use the search input, choose recipe category, see all recipes in one page, add new recipe button is active, the carousel of featured recipes is functioning.
* Testing the New Recipe form. We will verify that the user can successfully fill all input fields and add the recipe.
* Testing the page of the recipe. Verify that the recipe contains the image, the name, category, author. Like, follow and add to favorites buttons are active. Users can see these sections: description, ingredients, steps to produce, preparation time, comments. Authorized users can leave the comment under the recipe.
* Testing the ability to follow, unfollow another user, status changing.
* Testing admin login, category creation, editing, deleting functionalities.
* Testing error page occurrence. ("Page Not Found") and button with backlink to Registration page should occur.
* Ensure that the user interface is intuitive, user-friendly, and consistent across all modules of the system. This includes testing the layout, design, and navigation of the application.

**Test Strategy**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Level** | **Test Types** | **Testing Technique** | **Tools used** | **Description** |
| System Testing | Functional Testing | Experience-based Technique | Manual Testing, Test Cases | Testers interacted with the system as end users trying out various functionalities and observing the system's behavior. |
|  |  |  |  |  |
| System Testing | Regression Testing | Automated Testing | Selenium | Involves re-running our test cases from the functional testing phase to ensure that previously developed and tested software still performs after a change. |
|  |  |  |  |  |
| Integration Testing | Functional Testing | Automated Testing | Postman | Testing the interaction between different components of the software. |
|  |  |  |  |  |
| Acceptance Testing | User Acceptance Testing (UAT) | Black Box Testing | Manual Testing, Test Cases | The final phase of testing, where actual software users test the software to make sure it can handle. |
|  |  |  |  |  |
| System Testing | End-to-End Testing | Automated Testing | Selenium | Testing the entire software from start to end. |

**Test Schedule**

|  |  |  |
| --- | --- | --- |
| **Activity** | **Start Date** | **End Date** |
| Test Plan Development | 2024-07-09 | 2024-07-10 |
| Test Case Development and Test Environment Setup | 2024-07-10 | 2024-07-17 |
| Test Execution and Bug Fixing | 2024-07-12 | 2024-07-29 |
|  |  |  |

**Resource Planning**

QA Engineers and Testersare responsible for executing and maintaining test cases, automated test scripts and logging defects.

**Risk Management**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk** | **Potential Impact** |  | **Likelihood** | **Mitigation Strategy** |
| Delays in development | Could push back the start of the testing phase, leaving less time for testing. |  | High | Early involvement in the development process to help identify and address potential delays. |
| Unavailability of test environments | Could prevent testing from taking place as scheduled. |  | Low | Having a backup test environment ready to use in case the primary environment is unavailable. |
| High defect leakage | Lots of bugs are fixed after release |  | High | Through test case review and execution to catch as many defects as possible during the testing phase |

**Test Criteria**

* **Entry Criteria**
  + The development team has delivered a stable build for testing.
  + The test environment is ready, and test data has been prepared.
* **Exit Criteria**
  + All planned test cases have been executed.
  + All critical and high priority bugs have been fixed and retested.

**Deliverables**

* **Test Case Report Document:** This document, maintained in an Excel file, will contain all the test cases that were executed during the testing phase.
* **Test Scripts:** If automated testing was used, the test scripts will be provided. These scripts can be used for regression testing in future development cycles.
* **Bug Reports:** A detailed report of all the bugs found during testing will be provided and tracked in Jira, and the link to the project's Jira board will be provided.
* **Test Summary Report:** This report will be provided in Jira where it will be linked with the project.
* **Test cases include:** 
  + Test Scenario ID
  + Test case ID
  + Test case Name
  + Test case Description
  + Test details (test steps, test data, expected result, actual result)