**Test Plan For Webapp With Multiple Step Registration Flow And Payment System**

**1.0. Description:**

Webapp with multiple step registration flow where the user must verify the email address and make an initial payment.

**1.1. Features:**

Signup – “Create an account” page, that requests standard user information like First name, Last name, username, email, password, repeated password, checkbox for terms and conditions, optional checkbox for a subscription to project newsletter, silent captcha, and “Create account” button. User is forced to read terms and conditions before being allowed to create an account.

Login and password recovery – Login page that requests user credentials for an already created account, and a “Forgot your password?” button that opens a modal which prompts the user for their email address, in order to provide the user with the password reset link.

Account verification – Modal screen that informs the user that the verification link has been sent to their email address, verification code field where the user can input the verification code sent in their welcome email, “Verify email” button that verifies the email, and “Change email” button that allows the user to enter a new email address for verification.

Purchase App Credits – “Purchase App Credits” page where the user is required to make an initial payment to complete the registration process. Page contains a field describing the amount of app credits the user wants to receive and the conversion rate between the app credits and the selected currency (wireframe displayed USD-United States Dollars ($)). Payment method choice between Credit Card or PayPal is provided and selecting the Credit Card option prompts the user for their Credit Card details with a standard pre-populated with examples field of card number, expiration date, security code, and card holder’s name. “Purchase” button is available to complete the transaction, after which the user is redirected to the dashboard.

**2.0. Test Objectives:**

List of all the major test objectives that are targeted by the test plan and test cases.

* Verify that the signup and registration process is working as expected and without any errors or bugs.
* Verify that the user cannot proceed with the registration process without scrolling through the terms and conditions.
* Verify that the homepage buttons lead to correct modals/pages.
* Verify that the login process is working as expected and without any errors or bugs.
* Verify that the logout buttons redirect to homepage, and that logging back in returns the user to the same page.
* Verify that the account verification process is working as expected and without any errors or bugs.
* Verify that the code and email sending system works as expected.
* Verify that the purchasing app credits process is working as expected without any errors or bugs.
* Verify system performance and responsiveness (wait/load time) is satisfactory.
* Verify authorizing requests with postman for security purposes.

**3.0. Deliverables**

Deliverables are documents and specific data that is provided before, during, and after testing.

Deliverables before testing

* Test plan documents
* Test case documents

Deliverables during testing

* Test data
* Error logs
* Bug reports

Deliverables after testing

* Test reports
* Test efficiency
* Test effectiveness

**4.0. Bug Reporting**

In the event of finding a bug or an issue following points will be provided:

* Reproduction steps in as much detail as possible
* Error logs (if applicable)
* Pictures detailing the issue
* Video (if picture does not create enough understanding)
* Expected result
* Actual Result
* Bug Severity
* Environment on which the bug was found
* Simple and easily understandable bug title

**5.0. Tools**

Potential list of tools for test case management, bug tracking, and testing

* Azure DevOps/Qase/TestRail/Jira
* ShareX
* Postman
* JMeter

**6.0. Notes:**

* Verify with the client/product owner/project manager if the “Log in” button on the home page should lead to the terms and conditions modal (this is redundant since the users who created an account have already agreed to terms and conditions, this is not redundant in case the project will have regular updates to the terms and conditions)
* Verify with the client/product owner/project manager if the account verification uses both, the verification link, and a verification code. Both are mentioned in the wireframe, but this is also redundant
* Verify with the client/product owner/project manager that the user flow for “Already have an account? Log in.” should skip the terms and conditions modal (in the user story and the wireframe it is stated that login and signup action both require the user to read through the terms and conditions, but the wireframe skips terms and conditions from the “Already have an account?” popup) (test case will be written under the assumption that terms and conditions must be read at all points of contact for login and signup actions)