Abstract

**Aim:** To select an application and perform testing techniques over it using a testing tool of our choice.

**Application Selected:** Shuup E-Commerce Platform

**Software type:** Open Source Web Application.

**Features of Shuup** include Newsletter, Email, Website Design, Inventory, Payment Gateways, and Shipping etc.

**Built on:** Django (Python web based framework)

**Software Testing Tools selected:**

* Functional Testing Tool: Selenium
* Non-Functional Testing Tool: JMeter

**Types of testing we intend to do:**

* Functional Tests:
  + Unit Testing
  + Black Box Testing
* Non-Functional Tests:
  + Load Testing
  + Stress Testing

**Items not tested**

The following modules were not tested:

* Addons - Shuup contains facilities for installing, detecting, loading and configuring additional functionality with little or no system administration knowledge needed.
* Packages that can be loaded in this way are called Addons.
* Admin - creation of new admin was not tested.
* Settings - System settings, Notifications, Data Import and GDPR were not tested.

**Testing Metrics**

* No. of test cases planned vs executed : Planned (100 ) Executed (100)
* No. of test cases passed/failed: Test Cases passed (100) Test Cases failed (0)
* No of defects identified and their Status & Severity: 1, low severity.
* Defects distribution

- If asterisk field is left unfilled in Order status, Django key error is shown instead of alert box.

**Types of testing performed**

**1 Functional Testing**

**1.1 Unit Testing**

This is a level of software testing where individual units/ components of a software are tested.

The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output.

Number of test cases: 25

Modules tested:

- Login

- Orders

- Campaigns

- Coupons

- Discounts

**1.2 Blackbox Testing**

This is also known as Behavioral Testing, is a software testing method in which the internal structure/design/implementation of the item being tested is not known to the tester.

Number of test cases: 25

Modules tested:

- Reports

- Taxes

- Users

- API

- Customer Tax Groups

- Telemetry

**2 Non-Functional Testing**

**2.1 Usability Testing**

Usability testing is a technique used in user-centered interaction design to evaluate a product by testing it on users. This can be seen as an irreplaceable usability practice, since it gives direct input on how real users use the system.

Number of test cases: 18

Modules tested:

- Contacts

- Shops

**2.2 Compatibility Testing**

Compatibility Testing is a type of Software testing to check whether your software is capable of running on different hardware, operating systems, applications, network environments or

Mobile devices.

Number of test cases: 32

Modules tested:

- Forms

**Test Environment**

Test Environment consists of elements that support test execution with software, hardware and network configured. Test environment configuration must mimic the production environment in order to uncover any environment/configuration related issues.

We used Python environment for automated testing. Automated testing is the execution of the test plan by a script instead of a human. Python comes with a set of tools and libraries to enable creation of automated tests for our application.

Environment requirements:

* Python3 must be installed on the system
* Selenium must be installed on the system.

**Test Case Execution**

1. **Functional Testing**
   1. **LOGIN MODULE**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | 1 | | **Test Priority** | High | |
| **Test Title** | Verify Successful login | | **Description** | To check if the admin/user is  able to successfully log in to the  application | |
| **Test designed by** |  | | **Test Design Date** |  | |
| **Test executed by** |  | | **Test Execution Date** |  | |
| **Precondition** | | Valid username and password.  Properly configured python test environment.  Pass/Fail criteria should be well known. | | |
| **Dependencies** | | Valid url | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status** |
| 1 | Launch application | http://127.0.0.1:8000/sa | The page is correctly  displayed and the link is not broken. | The page is correctly  displayed and the link  is not broken | Pass |
| 2 | Enter correct username and  Password and hit the login button. | Username: admin  Password:\* | The user successfully logs in to the system. | The user successfully logs in to the system. | Pass |

|  |  |
| --- | --- |
| **Post Conditions** | Home page of the application appears. |

* 1. **ORDER MODULE**