AUTOMATION FRAMEWORK FOR

DISPLAY BOOKSHELVES

[**https://www.urbanladder.com**](https://www.urbanladder.com)

**Submitted By**

KINESTHETICS

SRAVYA KATUKURI

SAILAJA KANDLURI

NACHIKET KISAN SHEJWAL

MOHAMMAD KHALEED DUDEKULA

912790

912797

912649

912786

CONTENTS

1.Introduction

1.1 Purpose

1.2 Description of module

2.Artifacts

2.1 FRD

2.2 Test Case Documents

2.2.1 Test Scenario

2.2.2 Test Plan

2.2.3 Test case

2.3 Automation Script

3.Specifications

Browsers Supporting

4.Developing Framework

5. Conclusion

1. **Introduction**

TheFramework document is expected to provide the complete information about the framework used for project. It is expected to contain the types of framework, Artifacts having FRD, Testcase Document, and Automation Script, Specifications with browser support. Automation framework is expected to be done when the scope of test cases is 80% automatable. Framework is expected to cover user processes.

**1.1 Purpose**

The main purpose of this framework document is to intend the support or guidance about the automation process and script.

* 1. **Description of module**

Display Bookshelves is a module present in Urbanladder.com it is an online furniture shop. It has different types of furniture set like Bookshelves, Study Chairs, Study Tables, Beds, Dining etc. It also provides detailed description of the products like price, features and material type.

**2.Artifacts**

* 1. **FRD**
* The functional requirements document (**FRD**) is a formal statement of an application's functional requirements. It serves the same purpose as a contract. The developers agree to provide the capabilities specified. The client agrees to find the product satisfactory if it provides the capabilities specified in the **FRD.**
* In the Control System Toolbox software, you can use **FRD** models to store, manipulate, and analyse frequency response data. An **FRD** model stores a vector of frequency points with the corresponding complex frequency response data you obtain either through simulations or experimentally.

2.2 Test case document

### 2.2.1 Test Scenario:

### Test Scenario in software testing is a method in which actual scenarios are used for testing the software application instead of test cases. The purpose of scenario testing is to test end to end scenarios for a specific complex problem of the software. Scenarios help in an easier way to test and evaluate end to end complicated problems.

**2.2.2 Test Plan:**

“To plan is to act.” A test plan is a detailed document that outlines the test strategy, testing objectives, resources required for testing, test schedule and test deliverables. Test planning describes the scope, scheduled in the test activities. It also allocates the task for each test environment. It measures the amount of risk involved in the process. Test plan contains the test ID, the features to be tested, the entry and exit criteria and then the type of testing.

### 2.2.3 Test Case:

When you need to execute a particular function or a feature in the software application, we require test cases. We give a set of conditions to each test cases, and the tester determines if the input satisfies the defined objective in software testing. In general, the QA team is responsible to write the Test cases.

The Test cases will examine the expected and actual result and gives a pass or fail status at the time of execution. These test cases usually contain the title, description, test steps, expected result, and actual result. It is essential to write the test cases as it has a significant impact on the testing phase.

**2.3. Automation Script:**

* In FRD (Functional Requirements Document) Artifact the functional requirements from user perspective is done in MS Word document, PDF, PLAN Docx.
* In Manual Test Case Artifacts, the manual test case preparation for manual testing is done in MS Excel.
* Auto Script Artifacts contains the script format in java and selenium.

APPROACHES USED:

* Keyword driven using “config.properties” file.
* Data Driven using Apache POI.
* Custom HTML reports using Extent reports dependencies.

Extent Report Version:3.1.5

* Explicit waits.

**3.Specifications**

Browsers Supporting

* Chrome is the best browser which can perform browser compatibility testing.
* Microsoft Edge is also the best browser which can perform browser compatibility testing more efficiently.

**4. Developing Framework:**

* Eclipse-IDE

Eclipse-IDE for java development will be used for running and holding the complete project. It will work as an engine for the testing environment. Eclipse Oxygen.3 is expected to be used.

* Selenium

Selenium is an open-source tool that automates web browsers. It provides a single interface that lets you write test scripts in programming languages.

Selenium dependencies are added to Maven POM.xml file.

Version: 3.141.59

* Maven

Maven is a Java tool. Apache Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information.

* TestNG

TestNG is a testing framework designed to simplify a broad range of testing needs, from unit testing (testing a class in isolation of the others) to integration testing (testing entire systems made of several classes, several packages and even several external frameworks, such as application servers).

Version: 7.3.0 TestNG dependencies are added.

* Apache POI

Apache POI is a popular API that allows programmers to create, modify, and display MS Office files using Java programs.

Version: 4.1.0 Apache POI dependencies are added

* POM

POM stands for "Project Object Model". It is an XML representation of a Maven project held in a file named pom.xml.

**5.Conclusion**

Framework document gives an overview of the project. In this document we are discussing about the Artifacts, Specifications, developing framework which are used for automation testing in Urban Ladder.