

**WEB AUTOMATION USING SELENIUM WITH
CUCUMBER FRAMEWORK**

A PROJECT REPORT

Submitted by

ARYA RAJAN (TKM19MCA006)

to

The APJ Abdul Kalam Technological University

In partial fulfillment of the requirements for the award of the Degree of

MASTER OF COMPUTER APPLICATIONS



**Thangal Kunju Musaliar College of Engineering
Kerala**

DEPARTMENT OF COMPUTER APPLICATIONS

MAY 2022

DECLARATION

I undersigned hereby declare that the project report WEB AUTOMATION USING SELENIUM WITH CUCUMBER FRAMEWORK, submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of Prof. Natheera Beevi M. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: Kollam

Date: 12-05-22

A handwritten signature in black ink, appearing to read 'ARYA RAJAN', is written over a light blue rectangular stamp. The stamp contains some faint, illegible text.

ARYA RAJAN

DEPARTMENT OF COMPUTER APPLICATIONS
TKM COLLEGE OF ENGINEERING



C E R T I F I C A T E

This is to certify that, the report entitled **“WEB AUTOMATION USING SELENIUM WITH CUCUMBER FRAMEWORK”** project is submitted by **ARYA RAJAN (TKM19MCA006)** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the degree of Master of Computer Applications, is a bonafide record of the project work carried out by her under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Internal Supervisor

Head of the Department

External Examiner

ACKNOWLEDGEMENT

First and foremost I thank GOD almighty and my parents for the success of this project. I owe sincere gratitude and heart full thanks to everyone who shared their precious time and knowledge for the successful completion of our project.

I am extremely grateful to **Dr.Fousia M Shamsudeen**, Head of the Department, for providing us with best facilities.

I would like to express my sincere gratitude to my project guide **Prof. Natheera Beevi M**, Department of Computer Applications, who motivated me throughout the project.

I profusely thank all other faculty members in the department and all other members of TKM College of Engineering, for their guidance and inspirations throughout our course of study.

I owe my thanks to our friends and all others who have directly or indirectly helped us in the successful completion of this project.

ARYA RAJAN

ABSTRACT

Automation testing is the application of tools and technology to testing software with the goal of reducing testing efforts, delivering capability faster and more affordably. Automation testing is a need of fast and efficient software development life cycle (SDLC) for high productivity and good quality of product. It helps in building better quality software with less effort. Manual testing is performed by carefully executing predefined test cases, comparing the results to the expected behavior and recording the results. Manual tests are repeated each time the source code changes and is prone to errors.

The objective of this project is to design and develop Web Automation using selenium and cucumber framework. The framework uses automation testing to reduce effort and speed up the process of automated testing on Web. Cucumber and Selenium Framework is fast becoming a popular combination for web application test automation. Many organizations use Selenium for functional and regression testing. Selenium and Cucumber are a great combination when it comes to web application automation, as Cucumber allows you to write your tests quickly in English-like language and Selenium allows you to run on various combinations of browsers. Cucumber framework supports many languages, such as Java, Scala, Groovy, etc. beyond Ruby. Selenium also supports many languages, such as Java, .Net, etc. The experimental results show that the proposed method of automation testing based on cucumber and selenium can verify the web application software effectively, improve the testing efficiency and help enterprises save costs.

To ease the work of a tester the automated tools can execute, report and compare the result with previous test runs. The project "Web Automation Using Selenium With Cucumber Framework" focuses on Automation Testing of Mpowered Health Application as a regression suite. Mpowered Health is a consumer-driven healthcare technology company committed to creating a better healthcare experience for consumers and enterprises to take charge of their healthcare. Cucumber and Selenium framework is used to run the all functionality of application. Serenity is used to generate the report for Passed, Failed Test Case.

Contents

1	INTRODUCTION	4
1.1	Existing System	5
1.2	Objective	5
1.3	Company Profile	5
2	LITERATURE SURVEY	7
2.1	Purpose of the Literature Review	7
2.2	Related Works	8
3	METHODOLOGY	12
3.1	System Architecture	13
3.2	Architecture Explanation	14
3.3	Software Requirement and Specification	15
3.3.1	Eclipse	15
3.3.2	Java	16
3.4	Tools and Frameworks	17
3.4.1	Maven	17
3.4.2	Cucumber	18
3.4.3	Selenium	20
3.4.4	Serenity	20
4	RESULTS AND DISCUSSIONS	22
4.1	Test Plan	22

4.2	Testing Methods	23
4.3	Result Analysis	25
5	CONCLUSION	27
5.1	Advantages	28
5.2	Future Enhancement	28
	REFERENCES	29
	APPENDIX	30

List of Figures

3.1	Automated Lifecycle Testing	13
3.2	Selenium-Cucumber Tests Execution Flow	14
3.3	Serenity Report	21
4.1	Console output of the integration test execution by Maven.	23
4.2	Console output of the execution by Maven run	26
4.3	Serenity Report of Execution	26
5.1	Landing Welcome Page.	31
5.2	Login Scenario	32
5.3	Login Positive Scenario	33
5.4	Login Negative Scenario - User Enter Invalid Credentials	34

Chapter 1

INTRODUCTION

Web Automation is an excellent asset for web developers as it helps them automate a lot of the repetitive testing they may have had to do manually. This allows them to locate bugs and glitches in the software faster, which helps increase efficiency and streamlines their workflow. Testing is an important phase in the software development life cycle to improve the quality, reliability and performance of the system. Automated testing helps to develop a high-quality, strong and reliable software. It provides several benefits to a software development company such as improvement in testing efficiency, effectiveness and delivery.

Automation of software testing is the new trend to ensure a high performance application in a short period. Many organizations use Selenium for functional and regression testing. Selenium and Cucumber are a great combination when it comes to web application automation, as Cucumber allows you to write your tests quickly in English-like language and Selenium allows you to run on various combinations of browsers.

This project proposes a automated testing framework for Web Applications. This framework will enable the tester to perform automated testing of different browsers and to reduce effort and speed up the process of automated testing on Web.

1.1 Existing System

Testing is an important phase in the software development life cycle to improve the quality, reliability and performance of the system. According to the existing software testing system, testing of Web applications used visual verification to verify the results.

The first problem is that it is very time-consuming. The second problem of the previous system is that the accuracy and efficiency of the result ,since it is performed manually by the testers. There might be lots of errors can happen. Supposing the institution establish an enforcement, it might need to waste a lot of human resource and time which in turn will not be practical at all. Thus, all the application testing in the previous system is not reliable for quality analysis. Therefore, evolution is needed to be done to the previous system to improve efficiency, accuracy and performance.

1.2 Objective

The main goal of project is:

- To generate report of the test with all the informations regarding passed , failed test cases.
- To make code independent.Instead of hardcoding test data ,storing and reading all the data from Excel sheet
- Only need to change in Config.properties file if any configuration change needed.

1.3 Company Profile

Mpowered Health is a consumer-driven healthcare technology company committed to creating abetter healthcare experience for consumers and enterprises. The California-based company empowers consumers to take charge of their healthcare. The company provides enterprise solutions

in compliance, consumer acquisition engagement to enable healthcare organizations to serve their consumers more effectively and achieve their organizational goals. Mpowered Health is building the future of healthcare by connecting consumers and healthcare enterprises. We empower consumers to take charge of their healthcare by providing solutions that improve transparency, choice, access and convenience. In recognizing that consumer empowerment cannot be a zero-sum game, we enable healthcare organizations to serve their consumers more effectively and achieve their organizational goals. Mpowered Health empowers consumers to take charge of their healthcare by providing solutions that improve transparency, choice, access and convenience. Mpowered Health, in recognizing that consumer empowerment cannot be a zero-sum game, enables healthcare organizations to serve their consumers more effectively and achieve their organizational goals.

Chapter 2

LITERATURE SURVEY

Literature review is the comprehensive study and interpretation of literature that relates to a particular topic. When one uses literature review research questions are identified, then one seek to answer this research questions by searching for and analyzing relevant literature. Some importance of literature reviews is that new insights can be developed by the re-analyzing the results of the study. A literature review is both a summary and explanation of the complete and current state of knowledge on a topic as found in academic books and journal articles. There are two kinds of literature reviews you might write at university: one that students are asked to write as a stand-alone assignment in a course, and the other that is written as part of an introduction to, or preparation for, a longer work, usually a thesis or research report. The focus and perspective of your review and the kind of hypothesis or thesis argument you make will be determined by what kind of review you are writing. One way to understand the differences between these two types is to read published literature reviews or the first chapters of theses and dissertations in your own subject area. Analyses the structure of their arguments and note the way they address the issues.

2.1 Purpose of the Literature Review

1. It gives readers easy access to research on a particular topic by selecting high quality articles or studies that are relevant, meaningful, important and valid and summarizing them into one complete report.

2. It provides an excellent starting point for researchers beginning to do research in a new area by forcing them to summarize, evaluate, and compare original research in that specific area.
3. It ensures that researchers do not duplicate work that has already been done.
4. It can provide clues as to where future research is heading or recommend areas on which to focus.
5. It highlights the key findings.
6. It identifies inconsistencies, gaps and contradictions in the literature.
7. It provides a constructive analysis of the methodologies and approaches of other researchers.

2.2 Related Works

Here, we take some of the papers related to WEB AUTOMATION USING SELENIUM WITH CUCUMBER FRAMEWORK;

Rafi et al. discuss the benefits and restrictions of automated software testing. The survey tries to close gap between academia and industry. It finds the benefits comes from stronger sources of evidence while restrictions often comes from experience reports, it also shows the benefits are related to test reusability and save effort in execution phase while the restrictions are found in the automated setup process, choose the tool and lack of experience [1]. A study shows that automated scripts for testing are more efficient, accurate and cost effective over the manual testing . It shows manual testing is error prone compared to automated testing.

Rao et al.analyze software automation testing, definition, characteristics and functions. The study proposed method to improve the overall testing process[2].Hanna et al. make comparisons between various scripting techniques that used in automated testing. The study presents

an overview of different scripting techniques. It shows using these techniques will reduce the costs, speed up the process, and the products will take a short time to be delivered. In this paper we analyze software automation testing, definition, characteristics and functions are discussed. We propose a method to improve the overall process of software automation testing.[3]

Singh et al. describes a methodology that provides immediate test feedback to the software developer which allows achieving continuous testing during the development process. The Test Orchestration System useful for a pipeline of test phases. Continuous delivery of software in an industry requires periodic testing of various types. The future plan is making a Test Orchestration system cloud based, so the resources can be used in the best way. In this paper, a comprehensive review of the automated testing approach is presented to be used by other researchers in this field of study. The review result shows that the automated testing approach is suitable to enhance the regression testing with some plausible options of tools e.g. Selenium.[4]

A. Jain, M. Jain and S. Dhankar, describe A Comparison of RANOREX and QTP Automated Testing Tools and their impact on Software Testing. This paper highlights the comparison between QTP and Ranorex automation tools available in market and their use in the software project scenario. And also discuss about the purpose of testing can be quality assurance, verification, and validation or reliability estimation.[5]

Dubey and Shiwani tried to solve and add some new ideas to support testing by Ranorex and Test Complete. They provided a comparative study between Ranorex and Test Complete automated tools depend on concepts and features like creating test scripts, capacity to reuse the scripts and result reports. The main objective of this paper is to examine the features supported by these two testing tools that help in decreasing the used resources in script maintenance and efficiently reuse script. The elementary objective is to investigate the features and concepts supported by these two functional testing tools in order to access unconventionally what pros and cons of the tools and what could be the guidelines for its additional expansion.[6]

Shaukat et al. make taxonomy and a comparative study of different testing tools that available in the market to ensure the quality of software products. The results are generated from the data received from different companies. Interviews and test to verify the performance of tools not done and also experiments results are not presented in this research. For future work interviews, test performance and experiments results should conduct. In this paper, we are proposing and doing survey of the taxonomy of different testing tools and further more doing the comparison of these tools. Since there exists manual testing tools also but we are focusing on the just automated testing tools that are web based as well as application based. The comparison of various tools that are management, loading and functional have been done on the basis of their attributes such as Operating system support, Programming language support, Browser support, License and etc.[7]

A.Mahalakshmi discuss Web Application Automation using Selenium. In this project we test the web application using selenium c. Testing is the precedence modules in the purchasing a website. Initially test planning created based on the testing a website. Test design and test method made by test diagram information. Before developing take a look at scripts the scope of testing need to be documented clearly. Atlast writing scripts based totally on the precedence then textual content execution and reporting carried out step by step.[8]

Mr Tarik Sheth discuss Analysis Of Code Coverage Through Gui Test Automation And Back End Test Automation. The objective of the paper is to conduct a comparative study of automated tools such as available in market in Selenium and cucumber test tool. The aim of this research paper is to evaluate and compare automated software testing tools to determine their usability and effectiveness. There is wide variety of software testing tools available in market. Software testing tools has major features likes: web testing, window application etc.[9]

Dilara Ateşoğulları , Alok Mishra discuss about Automation testing tool: A comparative view. Effective software testing leads to assurance towards high quality in software development. Automation testing tool facilitates in faster testing process in testing stage thus completion and implementation of

software on time. One of the most significant issues for automation is to select the automation-testing tool and the appropriate framework. The objective of this paper is to assess and compare twenty-one available automation-testing tools on twenty attributes in comprehensive manner. This study will assist software testing professionals and researchers towards further insight in this area.[10]

Chapter 3

METHODOLOGY

Test Automation is applied with the structural sequence of testing process, that can have all the stages of testing life cycle with additional phases for including the test preparation and automation implementation processes. This entire process is known as Automation Testing Life Cycle. what is the testing process in agile development? First, in each iteration, testers need to review the requirements of the iteration to fully understand the requirements and find errors or omissions in the requirements during the review process, then design and write test cases according to the requirements. After the completion of test case writing, other members of the project team need to be invited to participate in the test case review to ensure the correctness of test cases and requirements coverage. After the test case is completed, if the coding of the tested module has been completed, the test environment can be built and the test can be executed. During the execution of the test, if defects are found, the defect report needs to be submitted. After the defect is modified, the defect needs to be retested. When all the test cases of the iteration are executed and passed, the testing task for this iteration is completed, and then the requirement review for the next iteration can be started. When the development and testing of all iterations are completed, all functions of the software version are developed, system testing and acceptance testing of the software can be carried out.

At the beginning of the test design, testers first need to consider the test environment. In the reality of mobile devices and platform fragmentation, it is impossible for testers to exhaust all the versions of devices and operating systems to achieve full testing coverage.

With limited time and effort, we can achieve as much testing coverage as possible from the perspective of input/output ratio. Because most mobile applications support iOS and Android operating systems, it is necessary to test mobile applications in both iOS and Android systems. The version of the operating system to be tested should cover the version of the operating system used by the vast majority of current users. The mobile devices tested should choose the tablets and mobile phones with high market share. The overall concept behind Automated Testing Lifecycle is illustrated in the following image.

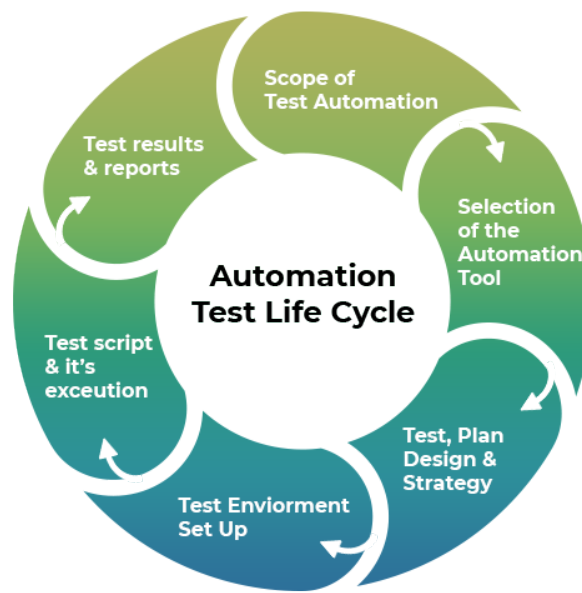


Figure 3.1: Automated Lifecycle Testing

3.1 System Architecture

Cucumber is a tool that supports behavior driven development (BDD) automation and can be easily integrated with Selenium. Selenium and Cucumber are a great combination when it comes to web application automation, as Cucumber allows you to write your tests quickly in English-like language and Selenium allows you to run on various combinations of browsers. When using Selenium with Cucumber for test automation, tests are written in feature files that can be understood by various stakeholders in an agile environment such as Business Analysts.

Cucumber also comes with its ability to support multiple scripts and programming languages and JUnit is used to execute these scripts and generate the output. Cucumber doesn't actually interact with the browser and performs actions on the website under test. Interacting with the website is the action that is performed by Selenium WebDriver.

Selenium-Cucumber Tests Execution Flow



Figure 3.2: Selenium-Cucumber Tests Execution Flow

3.2 Architecture Explanation

- Runner class is the main class of project. it used for running test classes
- Each of these features will have scenarios that must be tested using Selenium integrated with Cucumber. A file that stores data about features, their descriptions, and the scenarios to be tested, is called a Feature File.
- A Steps Definitions file stores the mapping data between each step of a scenario defined in the feature file and the code to be executed.

- Config.properties file contains all the configuration informations like Url,Browser path,Device version etc.
- Some of the functions are common to all the functionalities such as launching the application,reading data from Excel sheet etc.

Since these functions are reusable it is specific under Reusable functions package.

- After executing the functionalities we get Extent report. Extent report helps to know how much functionalities have passed or failed.

3.3 Software Requirement and Specification

The software used for the project:

- Eclipse
- Java

3.3.1 Eclipse

Eclipse is an integrated development environment (IDE) used in computer programming. It contains a base workspace and an extensible plug-in system for customizing the environment. Eclipse is written mostly in Java and its primary use is for developing Java applications, but it may also be used to develop applications in other programming languages via plug-ins, including Ada, ABAP,C, C++, Clojure, COBOL, D, Erlang, Fortran, Groovy, Haskell, JavaScript, Julia, Lasso, Lua, NATURAL, Perl, PHP, Prolog, Python, R, Ruby (including Ruby on Rails framework), Rust, Scala, and Scheme. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++, and Eclipse PDT for PHP, among others.

Plugins include:

- Cucumber Eclipse Plugin 1.0.0.202005150629
- Eclipse Web Developer Tools 3.21

Steps to configure Eclipse for the framework to work: To configure Eclipse environment for the testing framework , the following steps have to be performed.

- Go to Help tab on Eclipse
- Click on Eclipse Marketplace.
- In the window that opened, enter the plugins names that are above mentioned on the search bar. Hit enter
- Click on Install option. Once installation is done Eclipse will be restarted.

3.3.2 Java

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let application developers write once, run anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but has fewer low-level facilities than either of them. The Java runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in traditional compiled languages.

As of 2019, Java was one of the most popular programming languages in use according to GitHub, particularly for client-server web applications, with a reported 9 million developers.

3.4 Tools and Frameworks

The tools used for the project:

- Maven
- Cucumber
- Serenity

3.4.1 Maven

Maven is a build automation tool used primarily for Java projects. Maven can also be used to build and manage projects written in C, Ruby, Scala, and other languages. The Maven project is hosted by the Apache Software Foundation, where it was formerly part of the Jakarta Project.

Maven addresses two aspects of building software: how software is built, and its dependencies. Unlike earlier tools like Apache Ant, it uses conventions for the build procedure, and only exceptions need to be written down. An XML file describes the software project being built, its dependencies on other external modules and components, the build order, directories, and required plug-ins. It comes with pre-defined targets for performing certain well-defined tasks such as compilation of code and its packaging. Maven dynamically downloads Java libraries and Maven plug-ins from one or more repositories such as the Maven 2 Central Repository, and stores them in a local cache. This local cache of downloaded artifacts can also be updated with artifacts created by local projects. Public repositories can also be updated.

Maven is built using a plugin-based architecture that allows it to make use of any application controllable through standard input. A C/C++ native plugin is maintained for Maven 2. Alternative technologies like Gradle and sbt as build tools do not rely on XML, but keep the key concepts Maven introduced.

With Apache Ivy, a dedicated dependency manager was developed as well that also supports Maven repositories.

Steps to create a Maven project in Eclipse :

In order to create a Maven project on Eclipse , the following steps are to be done:

- Open Eclipse and Go to File - New - Others.
- Select Maven Project and click on Next.
- Un-check the ‘Use default Workspace location‘ and with the help of the Browse button choose your workspace where you would like to set up your Maven project.
- Select the archetype, for now just select the ‘maven-archetype-quickstart‘ and click on Next.
- Specify the Group Id Artifact Id and click on Finish.

3.4.2 Cucumber

Cucumber is a software tool that supports behavior-driven development (BDD).Central to the Cucumber BDD approach is its ordinary language parser called Gherkin. It allows expected software behaviors to be specified in a logical language that customers can understand. As such, Cucumber allows the execution of feature documentation written in business-facing text. It is often used for testing other software. It runs automated acceptance tests written in a behavior-driven development (BDD) style.

Behavior-driven development’s approach involves the usage of shared language that enhances communication between various tech and non-tech teams. Tests are more user-focused and based on the system’s behavior.

In BDD, “Given-When-Then” is the proposed approach for writing test cases. Consider the below example for better understanding:

- Given the user has entered invalid credentials
- When the user clicks submit button
- Then display the proper validation message

Benefits of using Cucumber Testing Tools:

Involving stakeholders becomes easier regardless of their programming knowledge

- Testers can write Test scripts without having in-depth knowledge of programming.
- Plugins are faster as compared to Selenium
- Supports various programming languages
- Code can be reused
- Simple and quick setup
- Flexible with different software platforms like Selenium, Ruby on Rails, Watir, Spring framework, and so forth.

Steps to implement BDD testing using Cucumber:

- To proceed with Cucumber implementation, we need to create packages to store the feature files, step definition code and testrunner code.
- To create a new package in src/test/java, right click the folder → New → Package.
- Now create the feature file in the package. Right click → New → File → Enter name test.feature. If you don't find 'File', then click on 'Others' and then select the 'File' option.
- Create a class test.java to write the gluecode for the features written. Right click Package → New → Class → enter name as test and save.

- To run the feature files and their respective code, we need to write a TestNG runner class. Right click Package → New → Class → enter name as testrunner.

3.4.3 Selenium

Selenium is a portable framework for testing web applications. Selenium provides a playback tool for authoring functional tests without the need to learn a test scripting language (Selenium IDE). It also provides a test domain-specific language (Selenese) to write tests in a number of popular programming languages, including C, Groovy, Java, Perl, PHP, Python, Ruby and Scala. The tests can then run against most modern web browsers. Selenium runs on Windows, Linux, and macOS.

Advantages of Selenium :

- Since Selenium is open-source, there is no licensing cost involved.
- Test scripts can be written in any of these programming languages: Java, Python, C, PHP, Ruby, Perl .Net
- Tests can be carried out in any of these OS: Windows, Mac or Linux
- Tests can be carried out using any browser: Mozilla Firefox, Internet Explorer, Google Chrome, Safari or Opera
- It can be integrated with tools such as TestNG JUnit for managing test cases and generating reports
- It can be integrated with Maven, Jenkins Docker to achieve Continuous Testing

3.4.4 Serenity

Serenity BDD is a library that makes it easier to write high quality automated acceptance tests, with powerful reporting and living documentation features. To run our Serenity tests with JUnit, we need to @RunWith the SerenityRunner, test runner. Serenity Runner instruments the step libraries and ensures that the test results will be recorded and reported on by the Serenity reporters.

Serenity is an open-source reporting library that enables developers to write easily-understandable and better-structured acceptance criteria for test automation projects. In addition to generating meaningful reports for each test, the tool also shows the list of features tested in each test case.

Advantages :

- It helps to how many test cases are Failed,Passing,Pending,Ignored,Skipped.
- It generate as a graphical structure
- They show the time needed for test execution.

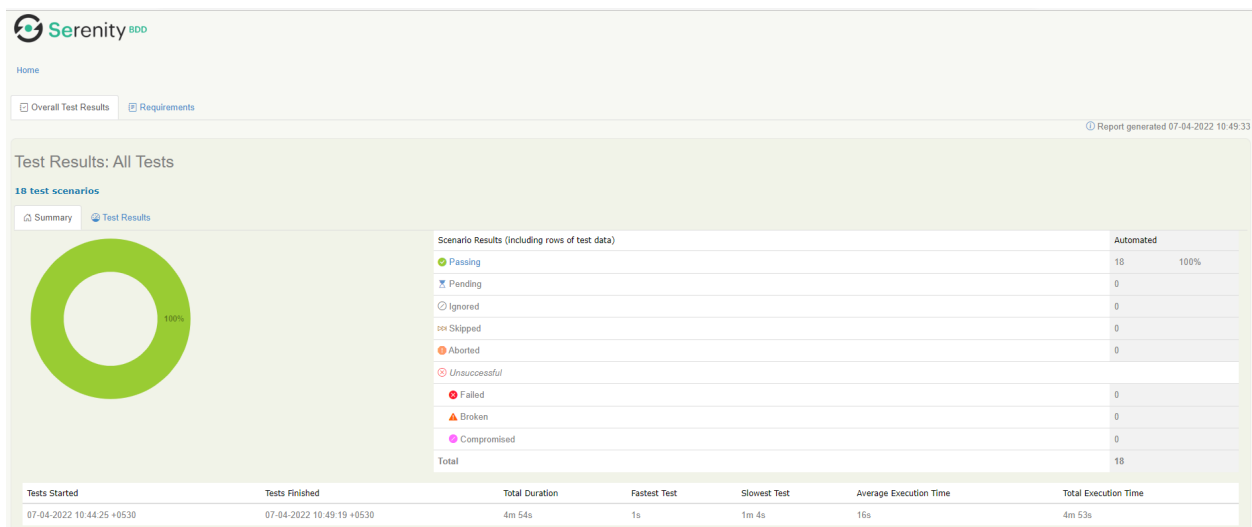


Figure 3.3: Serenity Report

Chapter 4

RESULTS AND DISCUSSIONS

For good performance of the system, application testing is needed. The test helps to identify errors, gaps or missing requirements in contrary to the actual requirements. It can be either done manually or using automated tools. Manual testing is the process of testing the software manually to find the defects. Automation testing is the process of testing the software using an automation tool to find the defects. Each test gave the appropriate results such as accuracy, better performance of the system etc. The following test methods are used for this framework.

4.1 Test Plan

A test plan is a systematic approach to test a system. The plan typically contains a detailed understanding of what the eventual workflow will be. Normally testing of any large system will be in two parts.

- The functional verification and validation against the requirement specification
- Performance evaluation against the indicated requirements

Testing activity is involved right from the beginning of the project. At the very first stage of testing, the goals and objectives are set. This simplifies the limits or borders of testing process. Before testing, the tester should plan what kind of data he/she is giving for test. Give data inputs as functional, boundary, stress, performance, usability values etc.

4.2 Testing Methods

- **Unit Testing**

Unit testing is a level of software testing where individual units/ components of application are tested. The purpose is to validate that each unit of the application performs as designed. A unit is the smallest testable part of any software. It usually has one or a few input and usually a single output.

- **Integration testing**

Integration testing tests the interface between modules of the software application. The different modules are first testing individually and then combined to make a system. Testing the interface between the small units or modules is integration testing. It is usually conducted by software integration tester and in continuation to the development.

```
[WARNING] io.cucumber:messages/maven-metadata.xmlfailed to transfer from https://ew
[WARNING] io.cucumber:messages/maven-metadata.xmlfailed to transfer from https://ew
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ MPIWeb ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 13 resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ MPIWeb ---
[INFO] No sources to compile
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 7.278 s
[INFO] Finished at: 2022-05-16T14:43:59+05:30
[INFO] -----

Process finished with exit code 0
```

Figure 4.1: Console output of the integration test execution by Maven.

Test Cases

A specific set of steps and data along with expected results of a particular test objective. A test case should only test one limited subset of a feature or functionality. Test case documents for each functionality/testing area will be written, reviewed and maintained separately in excel sheets. In system testing, test data should cover the possible values of each parameter based on the requirements. Since testing every value is impractical, a few values should be chosen from each equivalence class. An equivalence class is a set of values that should all be treated the same. Ideally, test cases that check our error conditions are written separately from the functional test cases and should have steps to verify the error messages and logs. Realistically, if error test cases are not yet written, it is OK for testers to check for error conditions when performing normal functional test cases.

Implementation

Implementation is the process of having the system personnel check out and put new equipment to use, train the users to use the new system and construct any file that are needed to see it. The final and impartment phases in the system life cycle are the implementation of the new system. System implementation refers to the steps necessary to install a new system to put into operation. The implementation has different meaning, ranging from the conversion of a basic application to complete replacement of computer system. Implementation includes all these activities that take place to convert from old system to new one. The new system may be totally new replacing an existing manual or automated system or it may be major modification to an existing system. The methods of implementation and time scale adopted are found out initially. The system is tested properly and at the same time the users are trained in the new procedure. Proper implementation is essential to provide a reliable system to meet organizational requirements. Successful implementations may not guarantee improvement in the organization involves the following things:

- Careful planning
- Investigation of the system and constraint

- Design the methods to achieve the change over
- Train the staff in the changed phase
- Evaluation of change over method Implementation methods

There are several methods for handling the implementation and consequent conversion from the old to new automated system. The most secure for this conversion is to run the old and new system in parallel. This method offers high security but the cost for maintaining the two systems in parallel is very high. Another method is direct cut over the existing system to automated system. The chance may take place within a week or within a day.

Implementation phase

It includes a description of all activities that most occur to implement the new system and put into operation.

It consists of the following steps:

- List all files required for the implementation.
- Identify all data required to build new files during the implementation.
- List all new document and procedure that go to the new system.

4.3 Result Analysis

Console Output;

Results obtained by executing the framework with 4 functionalities/ features , 18 scenarios / test cases and 18 steps. From this output we will be able to collect information such as how many test cases are passed, failed or skipped ,the total time taken to execute as an integration suite, whether the build is success or failure etc.

If any test cases fails,from output we will get in which feature which all test cases are failed.

```
Keep reports forever: https://reports.cucumber.io/profile
View your Cucumber Report at:
https://reports.cucumber.io/reports/f9f1da3e-8815-48a0-a887-15b8b04e527b
This report will self-destruct in 24h.
Keep reports forever: https://reports.cucumber.io/profile

results :

Tests run: 18 Failures: 0, Errors: 0, Skipped: 0

INFO] -----
INFO] BUILD SUCCESS
INFO] -----
INFO] Total time: 02:50 min
INFO] Finished at: 2021-06-20T10:01:55:05:00
```

Figure 4.2: Console output of the execution by Maven run

Serenity Report :

Extent Reports are the other source of information about the testing framework. The reports will be generated and stored under the "Reports" folder. Inside reports folder each reports will be saved within a folder named with the date and time of execution. So it will enable the testers to keep track of execution history. And from the report we can analyze the rate of failures, passes and skipped cases. Also date , started time,ended time are available in the reports.

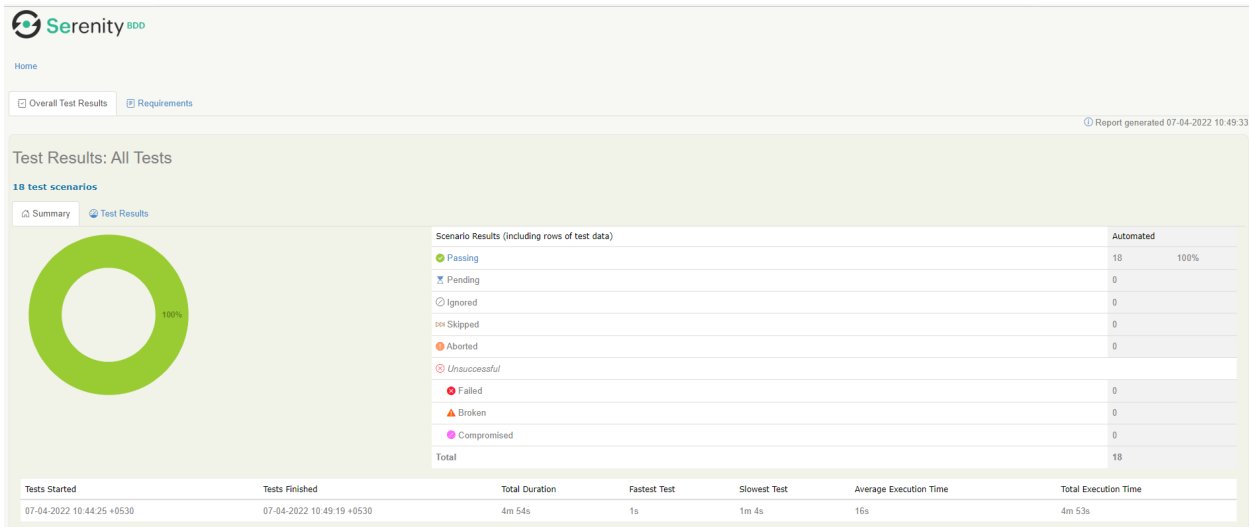


Figure 4.3: Serenity Report of Execution

Chapter 5

CONCLUSION

Testing and quality assurance is always important for the development of successful web/mobile application. Automation of software testing is a new trend that is taken up by the developers to deliver high performance applications in short periods of time. It is important to ensure that our web application will be compatible with mobile browsers and applications. mobile web apps testing tool helps to generate generalized test case in short time, without having knowledge of any scripting language.

The Purpose of this project is to design and develop Web Automation using selenium and cucumber framework. The framework uses automation testing to reduce effort and speed up the process of automated testing on Web. The main aim of Cucumber and Selenium Framework is fast becoming a popular combination for web application test automation. Many organizations use Selenium for functional and regression testing. Selenium and Cucumber are a great combination when it comes to web application automation, as Cucumber allows you to write your tests quickly in English-like language and Selenium allows you to run on various combinations of browsers. The experimental results show that the proposed method of automation testing based on cucumber and selenium can verify the web application software effectively, improve the testing efficiency and help enterprises save costs.

5.1 Advantages

The key Features are:

- To generate report of the test with all the informations regarding passed , failed and skipped test cases.
- To make code independent.Instead of hardcoding test data ,storing and reading all the data from Excel sheet.
- Only need to change in Config.properties file if any configuration change needed.

5.2 Future Enhancement

This project is based on Behavior Driven Development Framework.But in future that will change to Cypress framework.because Cypress is a developer-centric test automation framework that makes test-driven development (TDD) a reality for developers. Its design principle was to be able to package and bundle everything together to make the entire end-to-end testing experience pleasant and simple. Cypress has a different architecture than Selenium; while Selenium WebDriver runs remotely outside the browser, Cypress runs inside of it. This approach helps in understanding everything that happens inside and outside the browser to deliver more consistent results.It will used for automating the Web application and also in the Mpowered Health Website.

REFERENCES

- [1] D. Rafi, K. Moses and K. Petersen, "Benefits and Limitations of Automated Software Testing: Systematic Literature Review and Practitioner Survey", IEEE, pp. 36- 42, 2012.
- [2] Rao, A. Kumar, S. Prasad and E. Rao, "Quality Benefit Analysis of Software Automation Test Protocol", International Journal of Modern Engineering Research, vol. 2, no. 5, pp. 3930-3933, 2012
- [3] M. Hanna, N. El-Haggag and M. Sami, "A Review of Scripting Techniques Used in Automated Software Testing", International Journal of Advanced Computer Science and Applications, vol. 5, no. 1, pp. 194-202, 2014.
- [4] R. Singh, P. Sonavane and S. Inamdar, "Automated Software Testing using Test Orchestration System Based on Engineering Pipeline", International Journal of Advanced Research in Computer Science and Software Engineering, vol. 5, no. 5, pp. 1619-1624, 2015.
- [5] A. Jain, M. Jain and S. Dhankar, "A Comparison of RANOREX and QTP Automated Testing Tools and their impact on Software Testing", International Journal of Engineering, Management Sciences, vol. 1, no. 1, pp. 8- 12, 2014.
- [6] N. Dubey and S. Shiwani, "Studying and Comparing Automated Testing Tools; Ranorex and TestComplete", International Journal Of Engineering And Computer Science, vol. 3, no. 5, pp. 5916-5923, 2014.
- [7] K. Shaukat, U. Shaukat, F. Feroz, S. Kayani and A. Akbar, "Taxonomy of Automated Software Testing Tools", International Journal of Computer Science and Innovation, vol. 1, pp. 7-18, 2015.

- [8] A.Mahalakshmi, N.Sowmya, J.Suvethaa, S.M.Swaroop, "Web Application Automation using Selenium," International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075 (Online), Volume-9 Issue-7, May 2020
- [9] Mr Tarik Sheth¹, Ms. Priyanka Bugade, Ms. Sneha, Pokharkar, "Analysis Of Code Coverage Through Gui Test Automation And Back End Test Automation" International Journal of Innovative Science, Engineering Technology, Vol. 3 Issue 3, March 2016
- [10] Dilara Ateşoğulları , Alok Mishra " AUTOMATION TESTING TOOLS: A COMPARATIVE VIEW" International Journal on Information Technologies Security, № 4 (vol. 12), 2020 63

APPENDIX

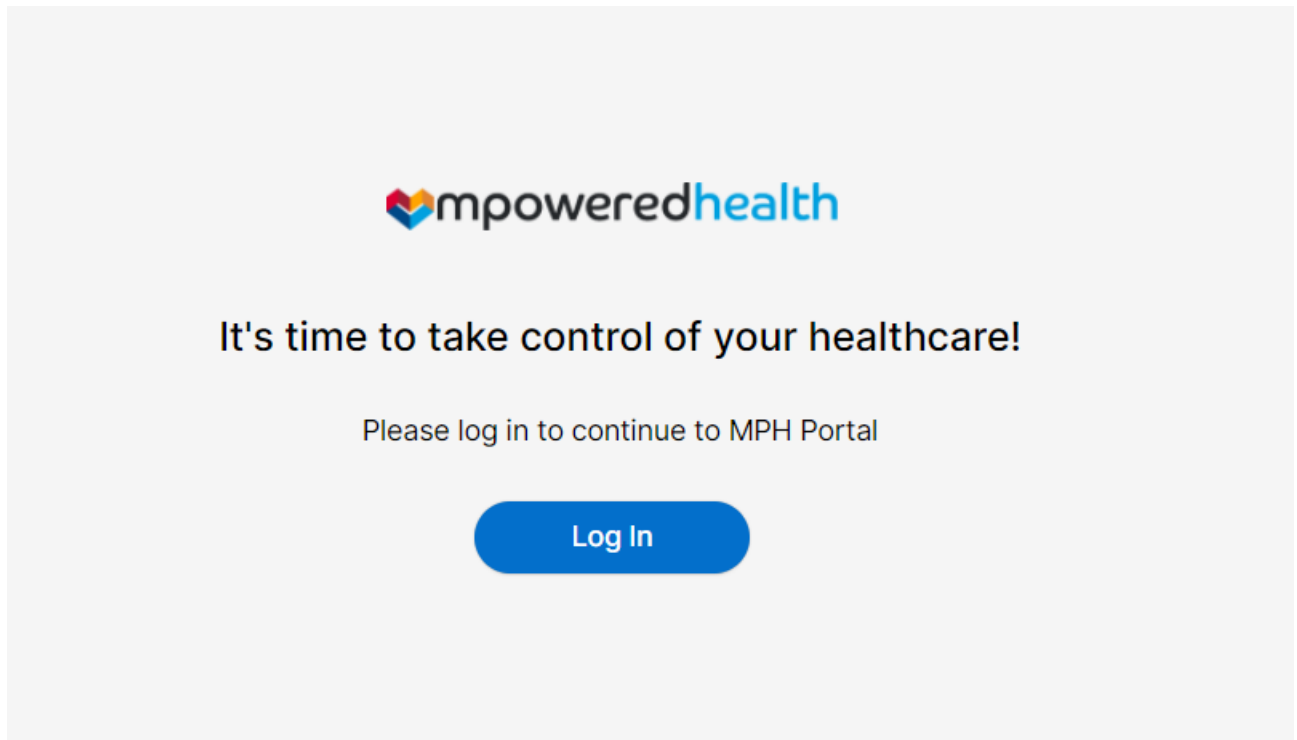
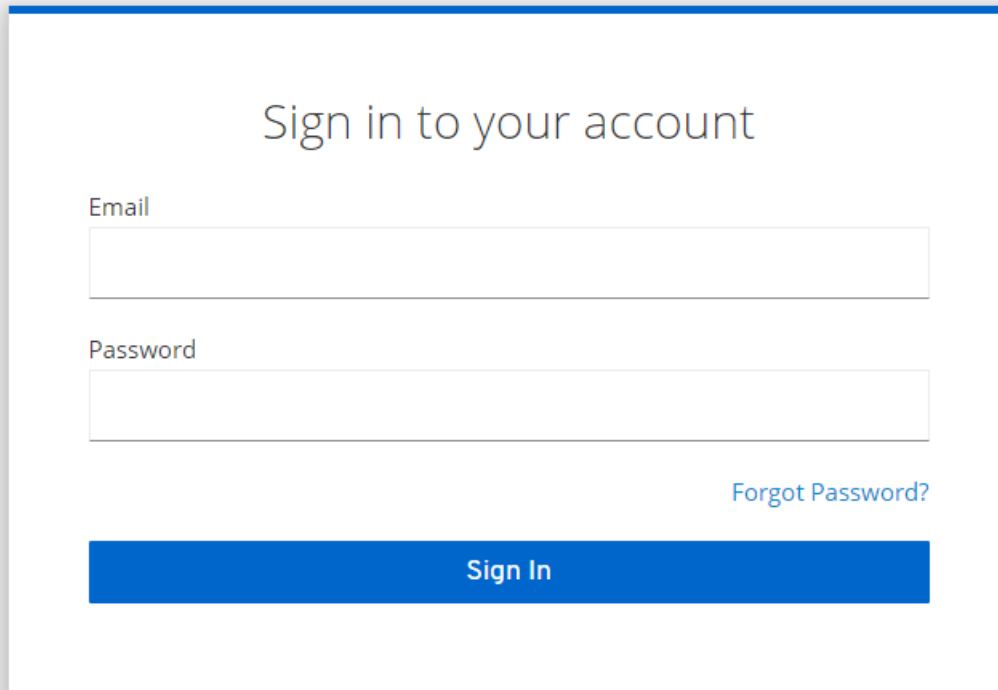


Figure 5.1: Landing Welcome Page.

MPH Partners



Sign in to your account

Email

Password

[Forgot Password?](#)

Sign In

Figure 5.2: Login Scenario

MPH Partners

Sign in to your account

Email
admin

Password
.....

[Forgot Password?](#)

Sign In

Figure 5.3: Login Positive Scenario

MPH Partners

The screenshot displays a login form titled "Sign in to your account". It features two input fields: "Email" and "Password". The "Email" field contains the text "internal" and has a red exclamation mark icon on its right side. Below the "Email" field, the text "Invalid username or password." is displayed. The "Password" field is empty and also has a red exclamation mark icon on its right side. A blue link labeled "Forgot Password?" is positioned to the right of the "Password" field. At the bottom of the form is a blue button labeled "Sign In".

Figure 5.4: Login Negative Scenario - User Enter Invalid Credentials