Web Automation of Capital Management Softwareusing Selenium and Java

Shantanu Jha
Department of ISE
RV College of Engineering
Bangalore, Karnataka
jha.shantanu378@gmail.com

Sushmitha N.
Department of ISE
RV College of Engineering
Bangalore, Karnataka
sushmithan@rvce.edu.in

Akshay Suryavanshi
Department of ETE
RV College of Engineering
Bangalore, Karnataka
akshayy.9191@gmail.com

Dr Ranjani G
Department of ETE
RV College of Engineering
Bangalore, Karnataka
ranjanig@rvce.edu.in

***Abstract*—— Modern software development approaches have come to rely heavily on automated testing, which provides many advantages over manual testing in terms of efficiency, dependability, and scalability. This paper provides a thorough analysis of the developments in automated testing methods, emphasizing the most important approaches, instruments, and industry best practices. The paper examines several automated testing methodologies, including functional testing, regression testing, and performance testing, after providing an outline of the basics of automated testing and outlining its advantages and disadvantages. The paper also explores popular automation technologies, including Selenium, and talks about their features, capabilities, and suitability for various testing scenarios. Furthermore, the paper examines emerging trends and future directions in automated testing, including the integration of artificial intelligence and machine learning techniques. Through this review, the paper aims to provide researchers and practitioners with valuable insights into the evolving landscape of automated testing, enabling them to make informed decisions and advancements in software quality assurance.**

***Index Terms*—Automated Testing, Software Quality Assurance, Test Automation Tools, Selenium, Functional Testing, Regression Testing, Performance Testing**

I. Introduction

The use of automated testing has revolutionized quality assurance and is now a fundamental component of contemporary software development. Software tools use this technique to test software applications before they are released, comparing the results to what is predicted and what is really achieved. By (technet.microsoft.com) using this approach, you can make sure that the program always fulfills requirements and works as intended, even after multiple updates and iterations. Several tools and methods are used in (doi.org) automated testing to mimic user activities with the software, including typing text, pressing buttons, and navigating between displays. Selenium is a popular web automation tool that lets testers check expected results and automate browser actions. In order to interact with the website or application being tested, test scripts are usually written in programming languages like Java, which Selenium supports. Web browser automation in telecom mimics the actions a human tester would perform during manual testing by inserting JavaScript code into the browser driver. This automation can validate various aspects of telecom software, such as user authentication, service activation, and real-time communication features. The automation programs then compare the results to the expected outcomes, highlighting any inconsistencies or errors that require further investigation. Web browser automation replicates the actions a human tester would take during manual testing by inserting JavaScript code into the browser driver. After that, the automation programs compare the results to what was anticipated, highlighting any inconsistencies or mistakes that need more research. Essentially, when compared to manual testing methods, automated testing improves efficiency, dependability, and accuracy by streamlining the testing process. Software engineers may guarantee the same level of quality across updates and releases by using technologies like Selenium and programming languages like Java

II. Literature Review

[1], For load test execution, the asset utilization of Selenium-based load tests in different designs was read. The two most important findings are (doi.org) that headless browsers use far fewer resources than other types of browser implementations. Similarly, a load driver's limit can be increased. by minimum 20% by sharing instances among client cases.

In[2], It is discovered that in a software development life cycle (SDLC), software testing is thought to be the most important progress. The main goal of testing methods is to compare the results that are obtained with those that the end-user is aware of. The task of a tester is made easier by employing specialized software to automate the component that involves execution. This focuses on using Selenium's webdriver to test an application and demonstrate how to use it in conjunction with other tools like TestNG, Maven, and so on, enabling progressively easier methods of enhancing the quality of testing.

In[3], Consideration is given to the benefits and drawbacks of utilizing Git. A vast array of commands in Git make it easier for developers to deal with and utilize the repository. The developer can work more productively as a result. Git enables developers to branch more frequently without affecting the master code. The developer is free to act whichever they choose on the private branch. Git has the drawback that, if the merging process is not carefully supervised, it may destroy or contaminate knowledge regarding modifications made to the code history, which is crucial for ongoing data development.

In [4], It is a given that software testing is the process of running a program with the intention of identifying flaws in order to produce software that has no defects. As a result, several testing techniques have been applied over time. This area of research has seen significant advancements. In the future, this field will become increasingly significant. There are several approaches to test case creation. The author discussed the various viewpoints on software testing, including the significance of testing standards and programming techniques. Furthermore, pertinent information regarding the many kinds of testing standards is also given.

In [5], It is evident that software testing can be done in two ways: manually or by using automation technologies to assess the product's quality, find flaws, and build user confidence. Automation tools assist in creating and running test scripts, saving money and effort compared to manual testing. The primary focus now accessible to support design and execution activity is automation tools.

III. Background and Objectives

A. Problem Statement

To Automate the web elements within a Capital Management Software involving several test cases related to the functioning of the product.

B. Purpose

Web automation is essential for enhancing the efficiency, accuracy, and consistency of interactions with web applications. Selenium, a robust and widely-used web automation framework, coupled with Java, provides a powerful combination for automating web-based tasks. The purpose of this project is to leverage Selenium and Java to create automated solutions that address various challenges in web application testing and repetitive task execution.

C. Motivation

Web applications play a crucial role in various domains such as e-commerce, finance, healthcare, and education. Ensuring their functionality, performance, and reliability is vital for business success and user satisfaction. The motivation for utilizing Selenium for web automation in Java stems from the need to address the inherent challenges of manual testing and task execution, while leveraging the power of automation to enhance efficiency, accuracy, and scalability.

D. Formulation of Objectives

The Search project is based on the following objectives:

1) To Automate the test cases in the different sprints based on their priority.
2) To Verify and Validate the API Requests.
3) To implement summarized versions of answers to capture key points from videos.

E. Scope and Relevance

In the realm of Information Science, which intersects with Computer Science, the project delves into crucial areas such as Web Automations, Cloud Computing and Netwok Security.These topics form the foundation of modern-day critical systems, shaping how data is validated and kept reliably. [5]
At its core, the project relies on fundamental principles of Computer Science, particularly those related to Authentication and Browser Automation across all browser platforms.Hence By leveraging these basic principles, the project aims to enhance reliability through web automations within the context of Information Science. It employs concepts from Computer Science to secure your data on cloud and ensure network security at all times. [4][6]

IV. Experiment Setup

In this section, a detailed account of the experimental setup employed for the implementation of Web Automation of Capital Management Software has been explained. The setup involved a combination of hardware and software components, including the development machine, storage, network infrastructure, and the necessary tools and frameworks. The hardware and software requirements for successfully replicating the experimental environment has been outlined accordingly as well.
The experimental setup for the project involves several essential steps to evaluate the effectiveness and performance of the developed software. The first step is to plan which includes Capital Planning where requirements are Identified as an optimistic spend curve to be defined, planned, socialized and approved by operations, stakeholders and finance.It involves capital planning and construction project management system to be applied to the software.
After upon, the second step is to build the Capital Software by Utilizing methods of providing cost information for a project, product, or service. The Capital project also needs to Control's the quality of a project's scope, time, and cost to maximize the project owner's satisfaction.
The final step is to operate the Software and identify bugs and critical features using Automation and Provide daily oversight of residential, commercial, or industrial real estate by a third-party contractor To evaluate the search engine's performance, a set of representative queries are to be tested on the automation software to identify smoke and regression testing.These test cases are handed to the testers by the Quality Engineers itself.

A. Hardware Requirements

For the development machine, typically include specifications related to processing power, memory, connectivity, input/output interfaces and other hardware-related functionalities.The specific requirements are an AOS-CX Switches, Typically required for connectivity. A 16GB RAM (M2 pro Chip) required to withstand the processing power of the Capital Software.Virtual Machines (up to 10GB RAM) is necessary to withstand huge data's on server side efficiently and a Ubuntu VM to provide operating system compatibility as well.

B. Software Requirements

The development environment requires an operating system that supports Linux, macOS, and Windows to accommodate various user preferences. Java is employed as the primary programming language due to its versatility and comprehensive libraries. Frameworks such as TestNG, Open source libraries like Selenium and Playwright are utilized to streamline development and enhance application functionality.
Visual Studio Code serves as the integrated development environment (IDE) for its user-friendly interface and robust features. Selenium Webdriver is used for web automation purposes by utilizing TestNG framework which lays down the various suites of test cases and implement them according to the severity of the test cases itself.
AutoIT is another support tool used by automation engineers to automate Windows commands. The automatable commands are mouse movements, single clicks and double clicks, Keyboard strokes and mouse hovers.

V. METHODOLOGY

Following is the methodology for the project in the form of list of tasks to be fulfilled:

1) **Assignment of Functional Test Cases:**
The approach to Automation of test cases starts with the planning meeting. Each milestone on which the Automation team works, is scattered into number of Sprints, which spans for over 3 weeks. Hence the sprint planning is done for the upcoming 3 weeks, which involves the capacity planning and the workload planning of the entire team.
During the planning meeting each one of the team members are assigned set of Product Backlog Items (PBI) related to the work which will be assigned. Based on the efforts required by each PBI (in term of hours), each PBIs' are allotted number of tasks. The number of tasks in the PBI help in uniform distribution of team's effort towards the completion of the PBIs throughout the planned Sprint. Each task which is created linking to the related PBI, will have a original estimate, which is the number of hours required to complete the part or whole PBI.

2) **Assignment of Modules and Manual Testing for the Capital Project:**
The functioning of the Capital Project software has been divided into various modules, each of them are assigned different modules, in which certain number of test cases are to be automated. The test cases for the different modules are written by the manual testers, who manually test the functionality of the software. After checking for the manual functionality of the modules, the test cases are written and given for the automation team to check the functionality through automation.
Manual Testing plays a major role here as before writing the test methods,it is important to understand the flow of work undertaken.
Once the set of test cases are assigned, the functioning of the feature is understood as mentioned in the test case and the required methods or statements are written in the framework for automation, after importing the dependencies and setting up the project.

3) **Creation of Project Object Model Page and Test Classes:**
The POM page and test page are to be setup, which involves the required steps for the flow of the automation of the test case. Then the test cases are executed on the required website by setting up the Azure logging and the report generation. Later the analysis of the report is done to identify and issues or bugs. The methodology given below along with the flowchart is a well-organized flow of automation testing, along with a clear picture of the steps followed by a Quality Engineers to automate the test case.

4) **Execution of Test Cases:**
TestNG Framework is utilized by identifying test cases in different groups such as Regression, Smoke and Sanity tests based upon how severe bugs have been encountered and later on dividing each test case using their test info for local runs.Using this Framework highlighting the test cases description becomes fairly easy as well.
Furthermore, the test scripts can be ran easily by inheriting the base classes, importing the required dependencies and running scripts for web automation.

5) **Setting up Azure Login and S3 Report Generation:**
In order to contribute the automated script towards the master branch, Azure Devops is used in abundance which helps you setup your own branch for local running of script and as and when required pushing changes as approved by senior developers into the master branch.
Another major thing is the Report Generation as internal runs can be performed for your automated test cases on the cloud and based on the test cases passed, the whole organization may/can receive auto generated reports with pass/fail ratios easily.

6) **Analysing Reports for Automated Test Cases:**
To improve the efficiency of test scripts automated, you can also analyse test reports as Azure generates screenshots of locations where test cases seems to have failed and generate terminal errors as well which can be analyzed by automation teams to improve their code quality easily.

By following this methodology, the project aims to provide users with a powerful and efficient search engine that

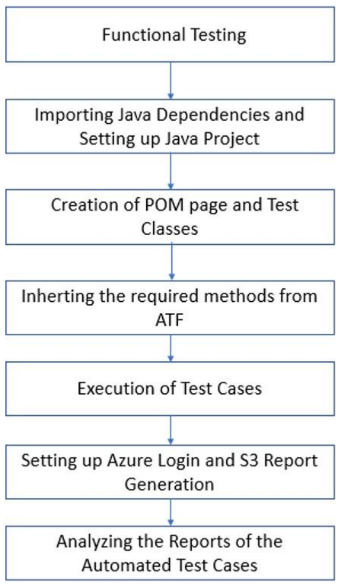

Functional Testing
Importing Java Dependencies and Setting up Java Project
Creation of POM page and Test Classes
Inherting the required methods from ATF
Execution of Test Cases
Setting up Azure Login and S3 Report Generation
Analyzing the Reports of the Automated Test Cases

Fig. 1. Architecture Diagram

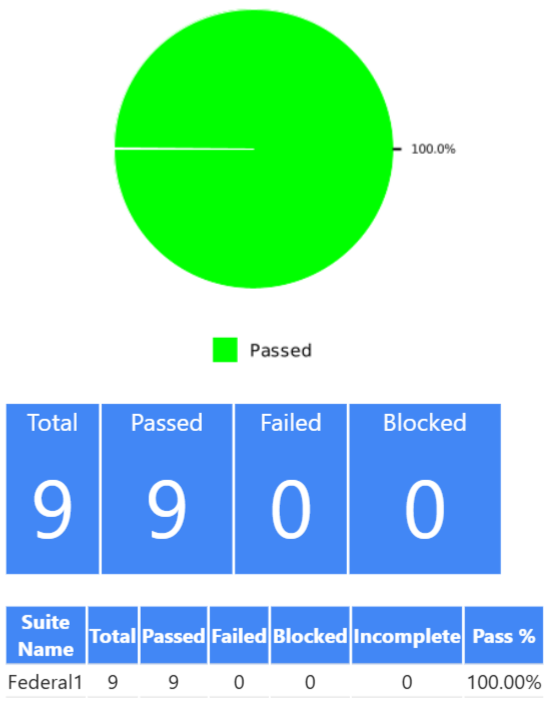
enhances the accessibility and usability of video content.

VI. Results and Discussions

The project's results provide remarkable insights that demonstrate its ability to perform capital planning efficiently and at the same time perform web automations to search for possible bugs that can be raised amongst the developers. Developing a strong Capital Software is essential to utilize it's capability to it's fullest extend.Therefore Azure provides with Consolidated reports which not only divide test cases based on suite names but also helps to identify the blockers for the software itself.

Hence without the user pin pointing issues with the software bugs, non-automatable test cases and redundant test cases can be highlighted by the team so that any blocker present can be addressed quickly and test cases which have a higher prioity number should be automated by the framework quicker as compared to other test cases. Finally, Any feature deemed necessary can be pointed to the developers so that customer satisfaction is kept at utmost priority and the software itself is not compromised.

To sum up, the initiative was effective in achieving its goals of encouraging capital project management and improving user experience. The project's outcomes highlight how capital management can be performed through steps like capital planning, identify reimbursements, fund transactions to all the way to project delivery and validating all these data using web automation. Hence it facilitates an industry-ready solution that automates every phase of the plan-build-maintain-operate lifecycle and ensure any critical bugs are tested by dividing them in suites like smoke, sanity and regression tests and finally testing them using web automation.



Suite Name	Total	Passed	Failed	Blocked	Incomplete	Pass %
Federal1	9	9	0	0	0	100.00%

Fig. 2. Consolidated Report

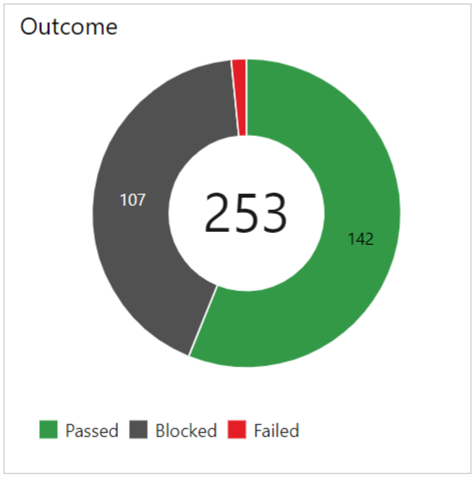


Fig. 3. Internal Runs Report

VII. Conclusion

Automation testing becomes a must because the urgent development deadlines of current software markets need to be met by the developers. Previously, dates were set and builds occurred at intervals that were measured in days or even months. Novel processes for development such as those associated with object-oriented programming and languages like Java require continuous software builds. As with successive cycles, new functionalities are added to the system under test, automation scripts need to be added, reviewed and maintained for each release thus acting as a cycle. Improving the effectiveness of automation scripts comes through maintenance. Hence it is a challenging task for Automation process to carry out the automation process along with catering to the new needs or updates by the developer team. Also maintaining good pass rate of the overall test automation scripts.

It can be now concluded that, the automation process for testing different scenarios in each software is better than the manual testing of the same scenarios, as it is based on code and script. Also compared to the manual testing, automation is faster, as it is assisted with a programming language. Though the automation process is considered faster and efficient compared to manual testing, adhoc testing is not allowed. Whereas adhoc testing can be carried out in manual testing. Hence, With the compatibility of the automation tools with almost all the available browsers, the set of scenarios in a given software can be tested across various browsers at a time, which is not easy in manual testing as more resources (Manual testers + Device) will be required to manually test the software on the different browsers at a time.Since automation is not yet completely foolproof, going forward, the milestone to be achieved for the automation process is to increase the scope of automation and become the only best solution to test the software.

VIII. Future Scope

Web automation using Selenium in Java has significant potential for advancements, particularly within the telecommunication domain, where efficient and reliable web applications are crucial. The integration of Artificial Intelligence (AI) and Machine Learning (ML) can revolutionize testing and automation processes. For instance, AI can analyze user interactions and network behaviors to generate and prioritize test cases that mirror real-world usage patterns, while self-healing mechanisms can dynamically update test scripts as applications evolve. This approach ensures continuous, accurate testing despite frequent updates, enhancing the robustness of telecommunication applications.

References

[1] 1.J. Gagnon et al., ”API Backward Compatibility Testing Using VSTS,” 2012 Ninth International Conference on Information Technology - New Generations, Las Vegas, NV, USA, 2012, pp. 237-241, doi: 10.1109/ITNG.2012.40.

[2] 1.B. E. Bennett, ”A Practical Method for API Testing in the Context of Continuous Delivery and Behavior Driven Development,” 2021 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), Porto de Galinhas, Brazil, 2021, pp. 44-47, doi: 10.1109/ICSTW52544.2021.00020.

[3] 1.Isha, A. Sharma and M. Revathi, ”Automated API Testing,” 2018 3rd International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 2018, pp. 788-791, doi: 10.1109/ICICT43934.2018.9034254.

[4] 1.K. Sawant, R. Tiwari, S. Vyas, P. Sharma, A. Anand and S. Soni, ”Implementation of Selenium Automation & Report Generation Using Selenium Web Driver & ATF,” 2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), Bhilai, India, 2021, pp. 1-6, doi: 10.1109/ICAECT49130.2021.9392455.

[5] A Simple But Tough-To-Beat Baseline For Sentence Embeddings by Sanjeev Arora, Yingyu Liang and Tengyu Ma1.S. Karus, ”XML development with plug-ins as a service,” 2012 Second International Workshop on Developing Tools as Plug-Ins (TOPI), Zurich, Switzerland, 2012, pp. 25-30, doi: 10.1109/TOPI.2012.6229806.

[6] 1.S. Raemaekers, A. van Deursen and J. Visser, ”The Maven repository dataset of metrics, changes, and dependencies,” 2013 10th Working Conference on Mining Software Repositories (MSR), San Francisco, CA, USA, 2013, pp. 221-224, doi: 10.1109/MSR.2013.6624031.

[7] C. Jain and R. Kaluri, “Design of automation scripts execution application for selenium webdriver and testng framework,” (docplayer.net) vol. 10, pp. 2440–2445, Jan. 2015.

[8] A. Begel, J. Bosch, and M. Storey, “Social networking meets software development: Perspectives from github, msdn, stack exchange, and topcoder,” IEEESoftware, vol. 30, no. 1, pp. 52–66, 2013

[9] D. Gaur and R. S. Chhillar, “Implementation of selenium with junit and test - ng,” 2012.

[10] S. Galler and B. Aichernig, “Survey on test data generation tools,” International Journal on Software Tools for Technology Transfer, vol. 16, Nov. 2013. DOI: 10. 1007/s10009-013-0272-3.

[11] L. Ma, C. Artho, C. Zhang, H. Sato, J. Gmeiner, and R. Ramler, “Grt: An automated test generator using orchestrated program analysis,” in 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE), 2015, pp. 842– 847.

[12] R.Ramler, D. Winkler, and M. Schmidt, “Random test case generation and manual unit testing: Substitute or complement in retrofitting tests for legacy code?” In 2012 38th Euromicro Conference on Software Engineering and Advanced Applications, 2012, pp. 286–293.

[13] R. A. Razak and F. R. Fahrurazi, “Agile testing with selenium,” in 2011 Malaysian Conference in Software Engineering, 2011, pp. 217–219.

[14] V. S. P. Ramya and P. V. Sagar, “Testing using selenium web driver,” 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT), vol. 14, no. 4, 2017.

[15] J. S.Just K.Herzig and B.Murphy, “Switching to git: The good, the bad, and the ugly,” in 2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE), 2016.

[16] A. J.Gaur, “A walk through of software testing techniques,” in 2016 InternationalConferenceSystemModelingAdvancement in Research Trends (SMART), 2016.

[17] D. Liu, “Use python api to automate script based on open stack platform,” in 2015 12th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), 2016.

[18] M. S. Alam, “A rest and http-based service architecture for industrial facilities,” in 2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS), 2020.

[19] H. K. Dhalla, “A performance analysis of native json parsers in java, python, ms.net core, javascript, and php,” in 2020 16th International Conference on Network and Service Management (CNSM), 2020.

[20] Isha, “Automated api testing,” in 2018 3rd International Conference on Inventive Computation Technologies (ICICT), 2020.

[21] A. Goyal, “Temporal json,” in 2019 IEEE 5th International Conference on Collaboration and Internet Computing (CIC), 2020.

[22] C.-P. B. A. T. S. M. Shariff H. Li and P.Flora, ““improving the testing efficiency of selenium-based load tests,” in Proceedings of the 14th International Workshop on Automation of Software Test, 2019.