Internship at Riskcovry

"Automated Website Testing and Reporting System: Enhancing QA Efficiency with Java, Cucumber, Playwright, SQL, and Jenkins"

Submitted by:

Siddarth M.P **PES1UG20CS423**

Under the guidance of

Ravikumar DS
Tech Lead,
Riskcovry

9 Weeks Internship

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

TABLE OF CONTENTS

COMPLETION LETTER

DECLARATION

We hereby declare that the project entitled **"Automated Website Testing and Reporting System: Enhancing QA Efficiency with Java, Cucumber, Playwright, SQL, and Jenkins"** has been carried out at **Riskcovry** by me under the guidance of **Ravikumar DS, Tech Lead** and submitted in partial fulfilment of the credits for the degree of **Bachelor of Technology** in **Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester for **9 weeks**. The matter embodied in this report has not been submitted to any other university or institution for the award of any degree.

PES1UG20CS423 **Siddarth M.P** 

ACKNOWLEDGEMENT

I would like to express my gratitude to our guide Ravikumar DS, Tech Lead at Riskcovry for their continuous guidance, assistance and encouragement throughout the development of this project.

I am grateful to the internship coordinator Prof. Mahitha G, Dept. of Computer Science and Engineering, PES University for organising, managing and helping out with the entire process.

I take this opportunity to thank Dr Mamatha H R, Chairperson and Dr Shylaja S S, (former Chairperson, Department of Computer Science and Engineering), PES University, for all the knowledge and support I have received from the department.

I would like to thank Dr B.K. Keshavan, Dean of Faculty, PES University for his help.

I am deeply grateful to Dr M. R. Doreswamy, Chancellor, PES University, Prof. Jawahar Doreswamy, Pro-Chancellor – PES University, Dr Suryaprasad J, Vice-Chancellor, PES University for providing me with various opportunities and enlightenment every step of the way.

Finally, this internship could not have been completed without the continual support and encouragement from my parents and friends.

Company Introduction

Riskcovry is an innovative and dynamic insurance technology company that revolutionizes the insurance landscape with its comprehensive platform. Offering a centralized hub of over 130 insurance products from more than 40 esteemed insurers across Life, Health, and General domains, Riskcovry provides a seamless plug-and-play experience across all fundamental insurance distribution channels. This unified platform harnesses the transformative potential of a single, simplified API, granting users unprecedented access to the entire insurance universe at their fingertips.

Through Riskcovry's pioneering technology, regulatory compliance, and diverse insurance offerings converge harmoniously. The platform's modular structure ensures a robust and efficient solution, enabling partners to deliver 100% digital insurance purchasing journeys, streamlined claim registration processes, and insightful performance dashboards that span the entirety of the buy-manage-claim lifecycle. Moreover, Riskcovry's cutting-edge approach empowers businesses to expedite their market entry by up to 50%, significantly enhancing speed to market.

With a user-centric approach, Riskcovry's centralized website acts as a nexus between customers and an array of esteemed insurance providers such as SBI, ICICI, and RELIANCE. Users can seamlessly navigate through filters, access information about their previous policies, and receive tailored policy suggestions curated from the offerings of various insurance companies. Once a user selects a preferred policy, the platform seamlessly redirects to the chosen insurer's payment gateway, culminating in a smooth and intuitive end-to-end process.

Project Title

Automated Website Testing and Reporting System: Enhancing QA Efficiency with Java, Cucumber, Playwright, SQL, and Jenkins

Is it part of another bigger project?

Yes, our project "Automated Website Testing and Reporting System: Enhancing QA Efficiency with Java, Cucumber, Playwright, SQL, and Jenkins" is indeed part of a bigger project aimed at enhancing software quality assurance and test automation within the company. The focus of this larger initiative is to streamline the testing process, improve testing efficiency, and ensure the quality of software products being developed. By automating the website traversal and running test cases developed by the QA team, our project directly contributes to achieving these goals.

Furthermore, our project's scalability is a notable aspect. The integration of payment gateways for additional ICs (presumably referring to Integration Components) demonstrates that the project is designed to accommodate future expansion and inclusion of new features.

One of the significant advantages of our project is its potential to significantly enhance the efficiency of the QA team. Previously, the manual testing of test cases was time-consuming and potentially error-prone. Through automation, our project reduces the need for repetitive manual tasks, allowing QA team members to focus on more critical aspects of testing and quality assurance. As a result, the company's reliance on a large number of QA positions may be reduced, leading to potential cost savings and resource optimization.

In summary, our project is an integral part of a broader company initiative to improve software quality assurance, enhance test automation, and streamline the testing process. The increased efficiency brought about by our project's automation contributes to optimized resource allocation and potential cost savings.

Project Overview and Role Responsibilities: Unveiling My Contribution

Playwright Automation Specialist:

- Design and implement automated interactions using Playwright to mimic user behavior.
- Develop Playwright scripts to navigate through web pages, interact with elements, and perform actions.
- Collaborate with the frontend team to identify HTML tags whose actions can be automated using Playwright.
- Ensure synchronization mechanisms are in place to handle dynamic web content.
- Integrate Playwright scripts into the automated test suite and Jenkins pipeline.
- Debug and resolve Playwright-specific issues in test scripts.
- Stay informed about Playwright updates and improvements to leverage new features.

Automation Documentation:

Maintain documentation for the test automation framework, including guidelines for creating new test cases and best practices for script maintenance.
Document any modifications or updates made to the automated test suite.

Abstract and Scope

The domain of the internship is "Software Quality Assurance (QA) and Test Automation." During the internship, the focus was on automating website traversal and running test cases developed by the QA team automatically. The technologies used, such as Java, Cucumber Playwright, SQL, and Jenkins, indicate a strong emphasis on test automation, software testing, and continuous integration in the field of software development and quality assurance.

The project's scope extended to designing and developing the system using Java programming language within the Eclipse IDE. The utilization of the Cucumber framework enabled the creation of human-readable test scenarios, enhancing

collaboration between technical and non-technical stakeholders. The automation tool Playwright was chosen for its cross-browser capabilities and modern web technology support.

The project's modular architecture allowed seamless integration with various insurance partners through dedicated feature files, promoting scalability and adaptability. Additionally, dynamic website changes were accommodated, ensuring that the automation system continued to function effectively even in the face of evolving interfaces.

API testing was an integral part of the project, encompassing interactions with backend systems and various payment gateways. User journey scenarios were covered, offering QA personnel the flexibility to validate application states during the testing process. The reporting mechanism provided comprehensive insights through video and HTML tracing reports, fostering transparency and efficient debugging.

Project design details with technologies used

Design Details :
Automation Framework: Modular architecture for efficient, scalable automation. Website pages divided into distinct modules with specific interactions.

Test Scenarios: Gherkin language for clear, collaborative scenarios. Cucumber framework creates feature files with user interactions and expected outcomes.

Automation Integration: Playwright chosen for cross-browser capabilities. Integrated Playwright APIs for seamless web interaction using XPath expressions.

Dynamic Adaptability: Structured for dynamic website changes. Specific methods and dynamic XPath expressions ensure adaptability.

API Testing: Integrated API testing with automated calls for backend interactions and validation of outcomes.

Technologies Used:
- Eclipse IDE for development environment.
- Java programming language for versatility and community support.

- Cucumber framework for defining human-readable test scenarios.
- Playwright as the automation tool for cross-browser testing and modern web technology support.
- SQL for storing and managing test data examples.
- Jenkins for continuous integration and test execution.

Implementation details

Automated Website Testing and Reporting System Setup:

- The system was built from scratch using Eclipse IDE as the development environment.
- Java programming language was chosen for its versatility and strong community support.
- The Cucumber framework was employed to create feature files in Gherkin language to define test scenarios in a human-readable format.
- Playwright was selected as the automation tool for its capabilities in cross-browser testing and support for modern web technologies.
- SQL was utilized to store and manage test data examples for different test cases.

Extensibility for Multiple Partners:

- The automation framework was designed with modularity in mind, allowing easy extension to accommodate multiple insurance partners.
- Each insurance company's automation was integrated separately, with dedicated feature files for each partner.
- Tags were used in the feature files to identify and run tests specific to each insurance company.
- The QA personnel could effortlessly choose the insurance company they wanted to test by selecting the corresponding tag, leading to the execution of all feature files containing that specific tag.

Dynamic Handling of Website Changes:

- The implementation was structured into separate modules based on well-established design principles, with each website page having its own module.
- In the feature files, simple English language test scenarios were written, connected to respective methods (step definitions), and linked with screen objects to facilitate interactions with the website.

- The framework was architected to adapt to dynamic changes on the website, such as the addition of new fields or user interface elements, ensuring uninterrupted test execution.

API Testing and Integration:

- API testing was automated as an integral part of the complete flow, including handling different partner APIs and various payment gateways.
- The automation framework utilized API calls to retrieve responses and validate the expected outcomes.
- Integration with payment gateways for each insurance company was automated, ensuring seamless payment processing tailored to the specific needs of each partner.

User Journey and Assisted Journey:

- The automation coverage extended to both user journey and assisted journey scenarios.
- QA personnel had the flexibility to pause the automation at any point during the journey for inspection and validation of the current application state.

Reporting and Feedback Mechanism:

- The automation framework generated comprehensive video reports and HTML tracing reports for each test run, providing valuable insights for debugging and future reference.
- Two frameworks were seamlessly integrated to facilitate automation:
- Cucumber, which allowed for providing parameters and tags to specify which tests to run.
- Jenkins, where QA personnel could customize test runs based on partner, user, and flow criteria.
- QA personnel could input relevant data into SQL, which was then utilized for test runs tailored to different partners.
- Test examples that were not present in the database were automatically skipped during execution.
- Jenkins reports offered detailed information on test progress, encountered errors, and the point of termination.
- A dedicated feedback table was implemented to log the output of each test case, encompassing success status, error messages, and the URL where the automation stopped.

Scalability and Server Storage:

- The automation framework was designed to accommodate large-scale testing, enabling efficient testing of multiple partners and insurance companies.
- Video reports and HTML tracing reports were securely stored on a server for future reference and analysis, ensuring easy access and retrieval of historical testing data.
- By meticulously implementing the aforementioned features and best practices, the Automated Website Testing and Reporting System significantly enhanced the efficiency of Quality Assurance (QA) processes. The system provided a robust and extensible solution for testing multiple partners' websites, each integrated with its specific payment gateway, streamlining the testing process for a diverse array of insurance companies.

Learning outcomes

By embarking on the journey of crafting the Automated Website Testing and Reporting System employing Java, Cucumber, Playwright, SQL, and Jenkins, I've reaped invaluable learning outcomes. My proficiency with automation tools and frameworks, notably Playwright for web automation, Cucumber for scenario definition, and Jenkins for continuous integration, has greatly expanded. Implementing this system in Java has bolstered my programming skills, nurturing the ability to compose efficient and maintainable code.

Through practical exposure to SQL for data management, I've gained insights into effective database handling and querying. Expertise in automating web applications and conducting UI tests via Playwright and XPath has enriched my skill set. The integration of distinct payment gateways has illuminated the complexities of integrations and financial transactions.

The modular design's tutelage in extensibility and maintenance has been instrumental, allowing effortless integration of new partners and features. The ethos of Test Driven Development and Behavior-Driven Development fostered by Cucumber has crystallized clearer requirements and better collaboration.

Automating API testing and validation has deepened my capacity to manage backend interactions and data validation. Version control through Git has nurtured collaborative skills within teams, and Jenkins' configuration for continuous integration underscores the importance of a seamless testing process.

The project's demands have honed my problem-solving and critical thinking abilities, catalyzing adeptness in troubleshooting. Crafting video reports, HTML tracing reports, and feedback mechanisms underscores my knack for informative documentation.

Dealing with the intricacies of a complex project has sharpened my project management acumen, ingraining the significance of meticulous planning and resource allocation. My adherence to testing best practices and industry standards has underscored the foundations of effective testing methodologies.

In sum, developing the Automated Website Testing and Reporting System has nurtured a comprehensive skill set, transforming me into a proficient automation engineer adept at testing web applications and navigating intricate integrations.

Conclusion

In conclusion, the development of the Automated Website Testing and Reporting System using Java, Cucumber, Playwright, SQL, and Jenkins has been a highly rewarding and enriching experience. Throughout this project, I have acquired valuable technical skills and practical knowledge that have significantly enhanced my proficiency as an automation engineer.

The project enabled me to deepen my understanding of automation tools and frameworks, such as Playwright for web automation and Cucumber for behavior-driven testing. My expertise in Java programming has also been sharpened, allowing me to write efficient and maintainable code for the automation system.

Working with SQL to manage test data examples and integrating multiple payment gateways for each insurance company has exposed me to real-world challenges and solutions in database management and complex API integrations. Furthermore, the modular design of the system has taught me valuable lessons in structuring projects for extensibility and easy maintenance.

Embracing a test-driven and behavior-driven approach with Cucumber has improved my ability to define clear requirements and collaborate effectively with other stakeholders. Additionally, configuring Jenkins for continuous integration has empowered me to streamline the testing process and ensure a seamless flow of automated tests.

The project has strengthened my problem-solving and troubleshooting skills, as I faced and resolved various challenges throughout the development process. Moreover, creating comprehensive reports and documentation has emphasized the importance of clear communication and meticulous documentation practices.

Overall, this project has not only expanded my technical capabilities but also instilled essential project management skills, emphasizing proper planning, time management, and resource organization.

Developing the Automated Website Testing and Reporting System has been a rewarding journey that has provided me with a well-rounded skill set as an automation engineer. I am confident that the knowledge and experience gained from this project will prove invaluable in my future endeavors, as I continue to contribute to the field of software testing and quality assurance with greater efficiency and confidence.