**Name-Lovepreet kaur**

**Student id-2036239**

**Module-(7CC005)Web Technologies**

**Module Guide-Alix Bergeret**

**SELLING OF PRODUCT FROM THE WAREHOUSE**

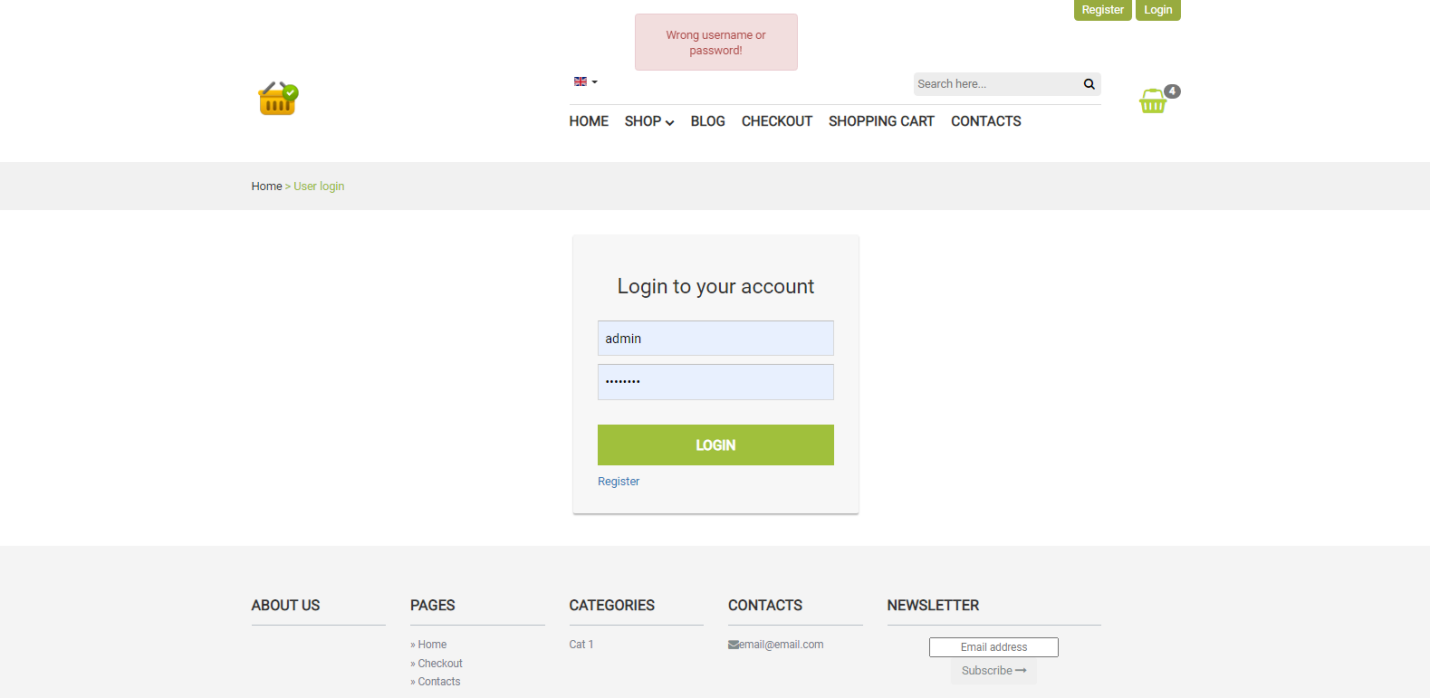
**Table of Contents**

[Testing 3](#_Toc74068251)

[Reference List 6](#_Toc74068252)

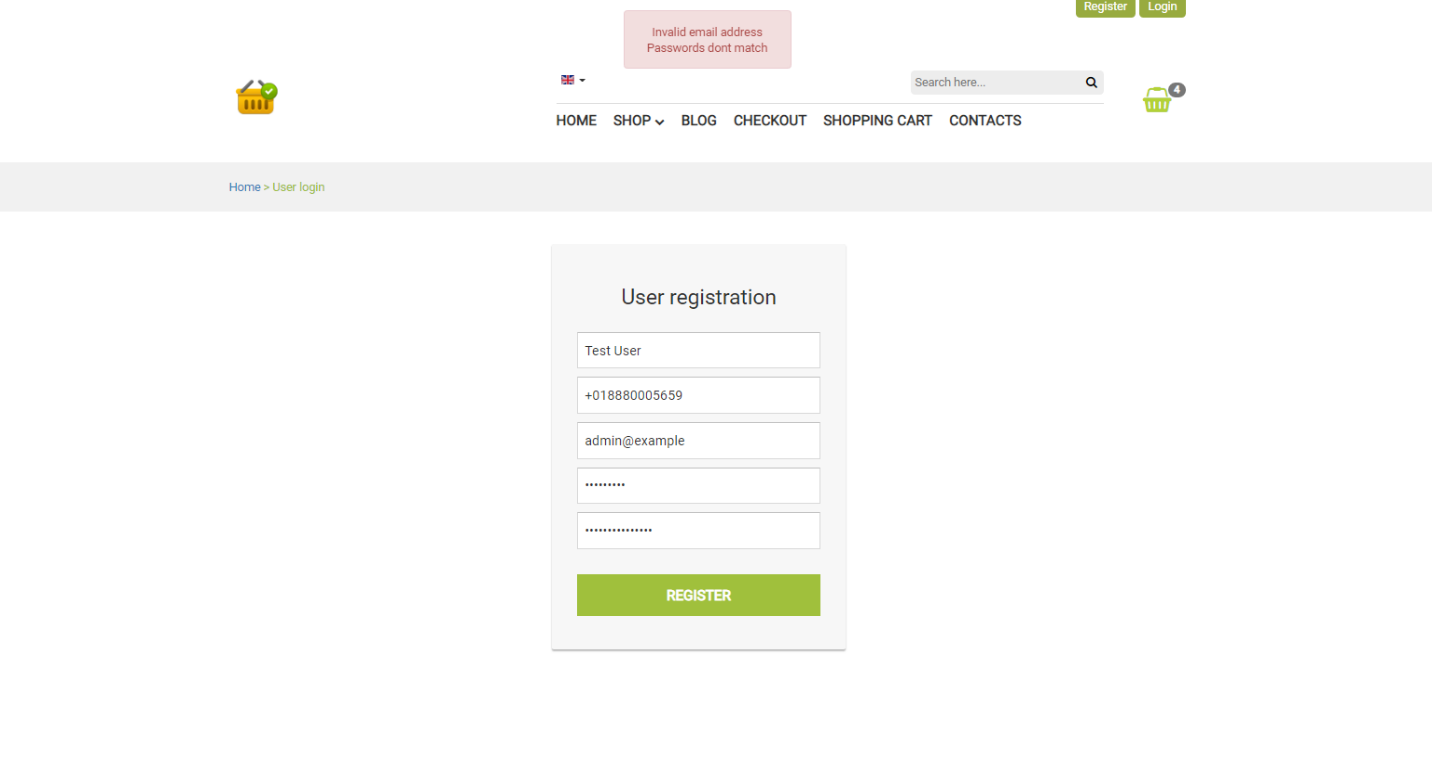
# Testing

Software testing is a process of analysing or checking if the overall product of the software fulfils the requirements which are expected for the purpose of ensuring the product of software is defect-free software (Alghamdi *et al*. 2019). It includes the process of executing the components of the software or system with the help of tools that are automated or manual tools of any system or software components for the purpose of evaluating the interest of properties. The objective of software testing is to point out the errors, requirement gap, or missing of any requirement contrasting to the main requirements of the software. Whether there are any errors or any bugs that are occurring in the following software, they can be detected at an early stage and can be cleared up before software product delivery.



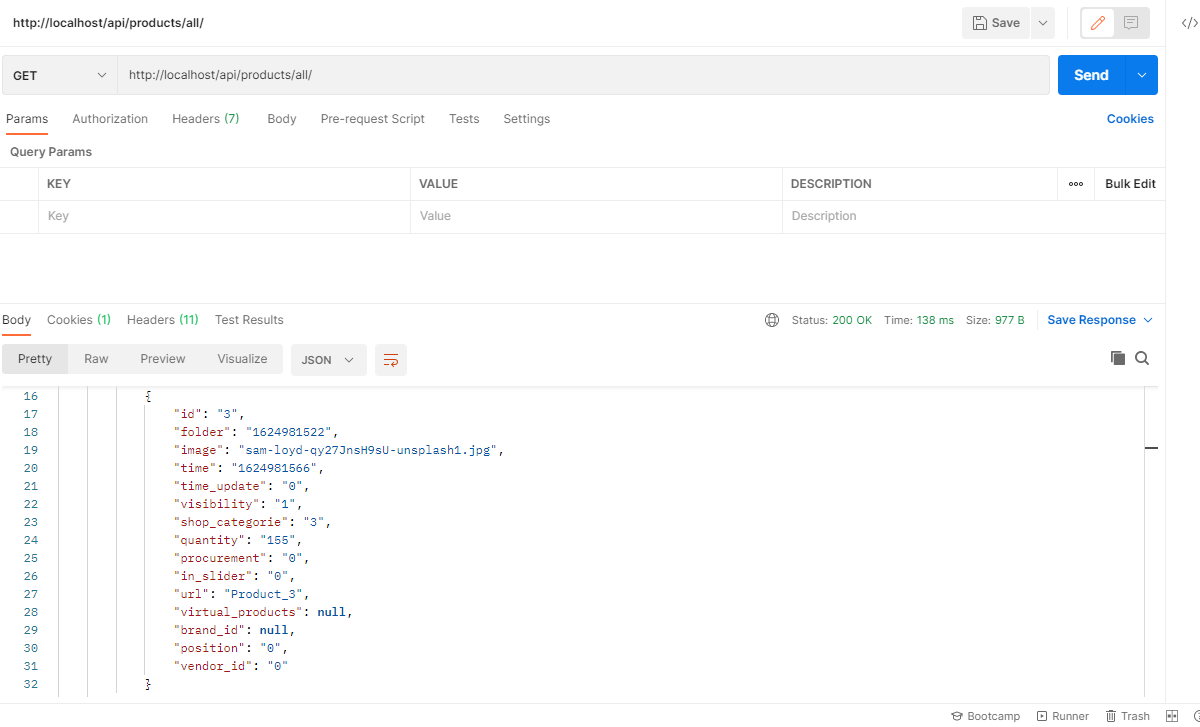
**Figure 1: Login testing with wrong credentials**  
Source: Google Chrome

Testing software in a proper way guarantees the reliability of the following software, safety, and the possibility of high performance that further saves them time, saves money, and satisfies the user of the software. This is the importance of testing software. There are several advantages of software testing. Such as, it decreases the factors of risk and provides security to the software. Software testing reduces the cost of fixing the software as the "bugs" are caught in this process. It also ensures a better quality of the software (Mishra, *et al*. 2017). "Automated software testing" is a technique of testing software and comparing the actual result with the expected result. This can be done by writing the scripts of testing or using any other tools for automation testing.

****

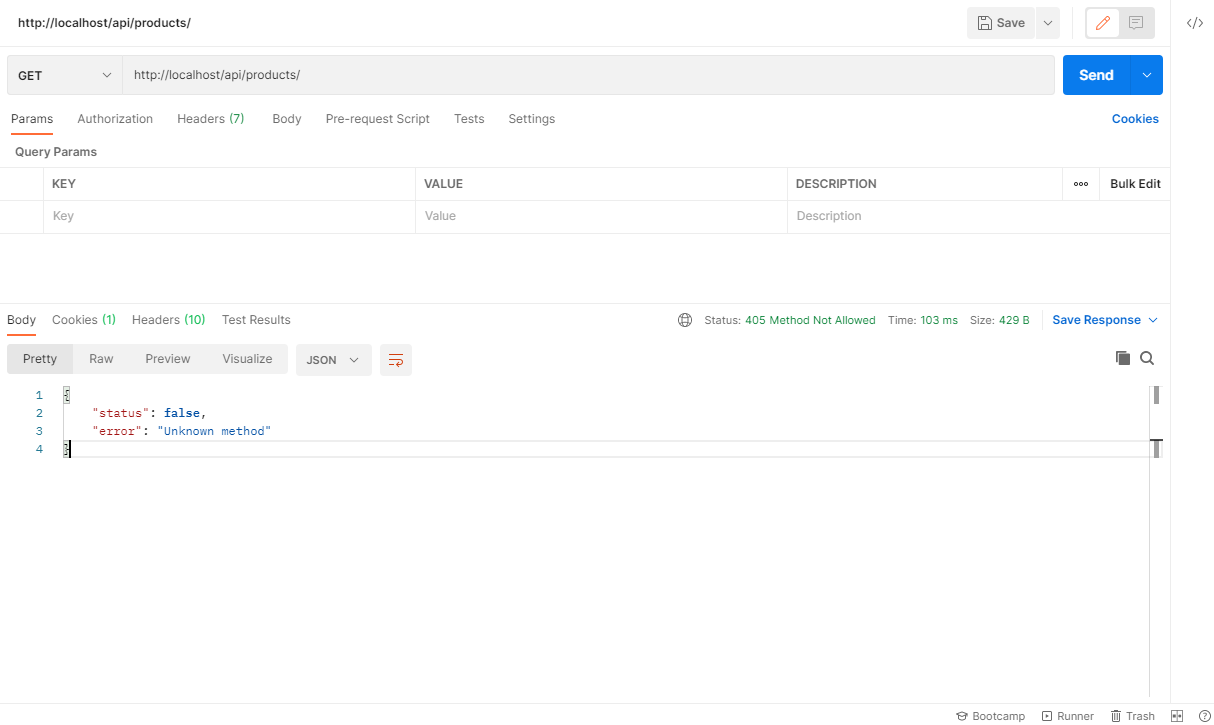
**Figure 2: Registration testing with wrong email address and invalid password  
Source: Google Chrome**

"Test automation" is used for the automation of repetitive work and other manually difficult tasks of testing. In the process of software testing, "test automation" is used for the purpose of controlling the implementation of the tests and the comparison of actual results with expected results separately from the software to be tested. Software testing for continuous delivery and ongoing testing, test automation is critical. "Automated tester" works instead of spending the code for the tests and the tester will make improvements to these following tests simultaneously as the required adjustments. However, after the completion of the test, "automated testing" permits the recycled utilization of tests to repeat this entire process.



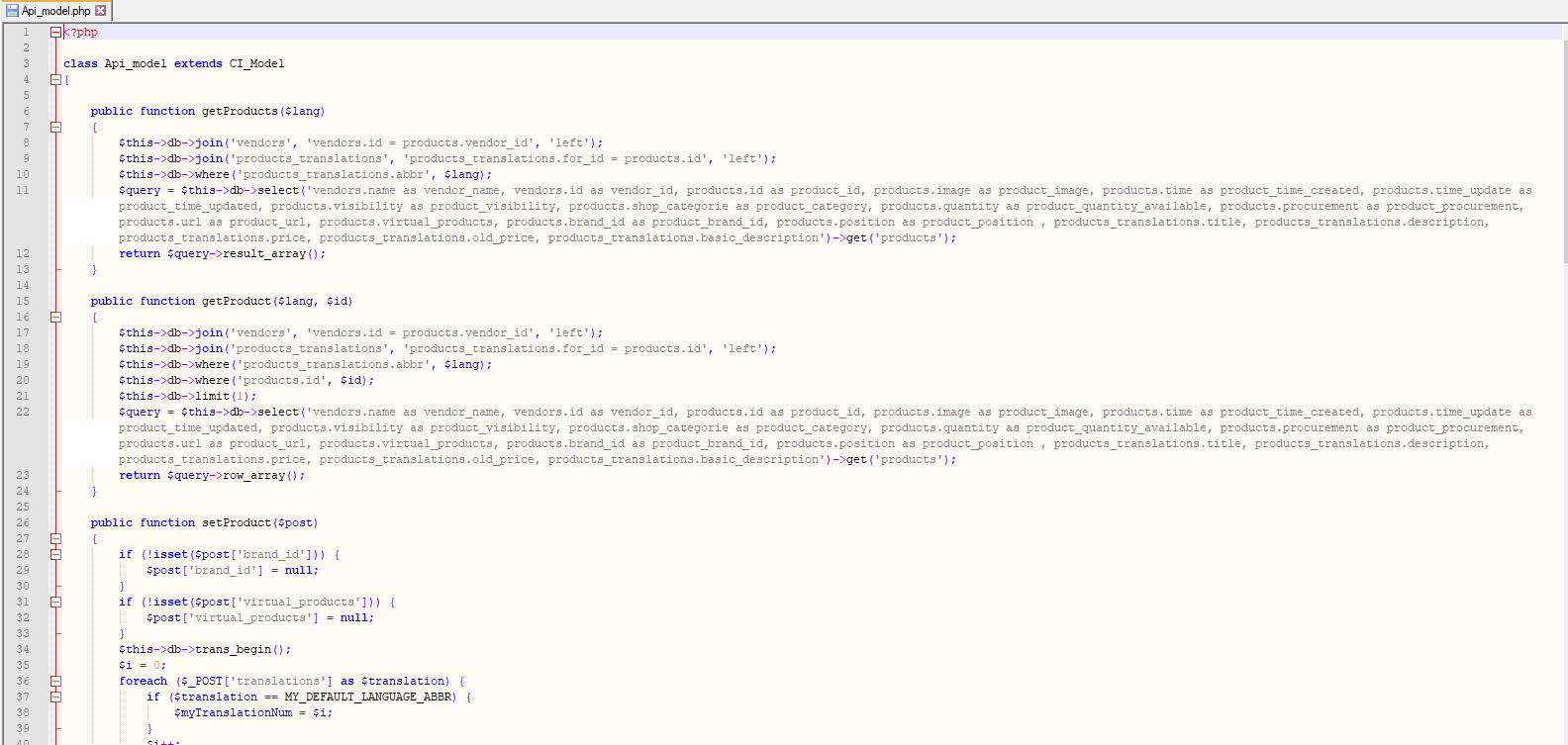
**Figure 3: API Endpoint Testing  
Source: Postman (API endpoint testing tool)**

Selenium HQ is in charge of evaluating the website that is being constructed. Selenium is a set of tools that are widely used during the testing community for cross-browser testing. Selenium can't help developers with desktop programs, but it can help one with browsers. It is one of the most popular tool suites for web app automated testing since it supports a wide range of web browsers, making it quite powerful. Selenium is a tool that simplifies web browser communication (Butt *et al*. 2017). It is mostly used to create processes that allow the user experience to be evaluated on the website much more quickly than it could be done manually. A simple Selenium test can verify that the computer can retrieve a file, but a more complex test may automate the entire journey. Selenium works with a range of browsers, including Chrome, Edge, and many others. Selenium's design is essentially a "client-server" design.



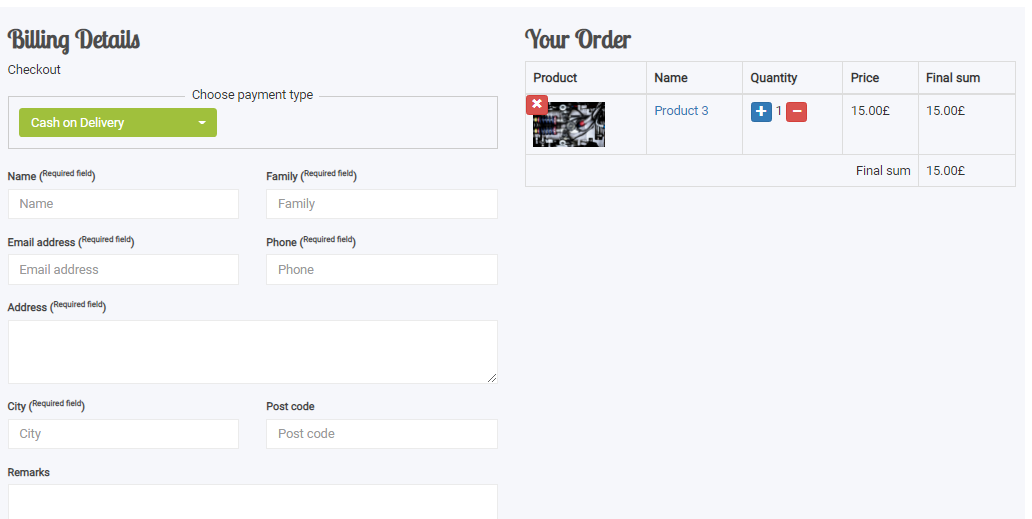
**Figure 4: API Endpoint Testing with wrong request  
Source: Postman (API endpoint testing tool)**

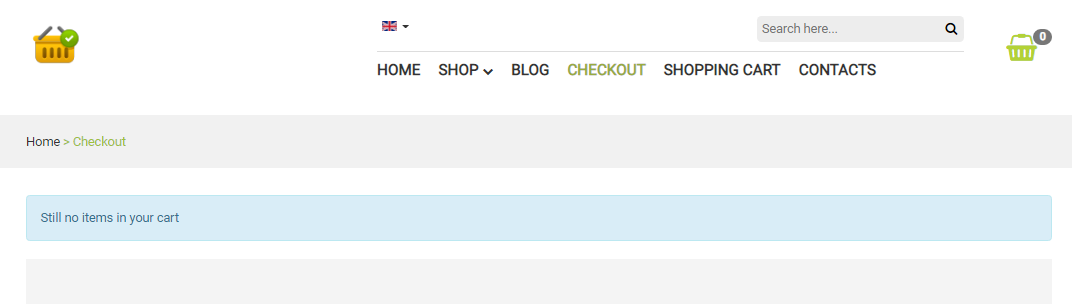
A "Web driver API" is included in the client architecture, which may be used to create various program codes that interact with application parts and pages. It also offers a class named "Remote Web driver," which allows Selenium to communicate with a web server. The architecture of the server comprises a component that receives requests from the Selenium Client's "Remote Web Driver" class. It also offers the "Web Driver API," which allows web browsers to execute tests on host PCs. It also features "Selenium Grid," which was established by Selenium Server and provides grid programmable buttons, and also a central hub, nodes in various settings, and browser abilities. Selenium works with a variety of programming languages, including Java, Python, C#, PHP, and others (Hanna *et al*. 2018). Selenium is incredibly versatile since testers may choose whatever language they use to create test cases.



**Figure 5: API modelling test  
Source: Code**

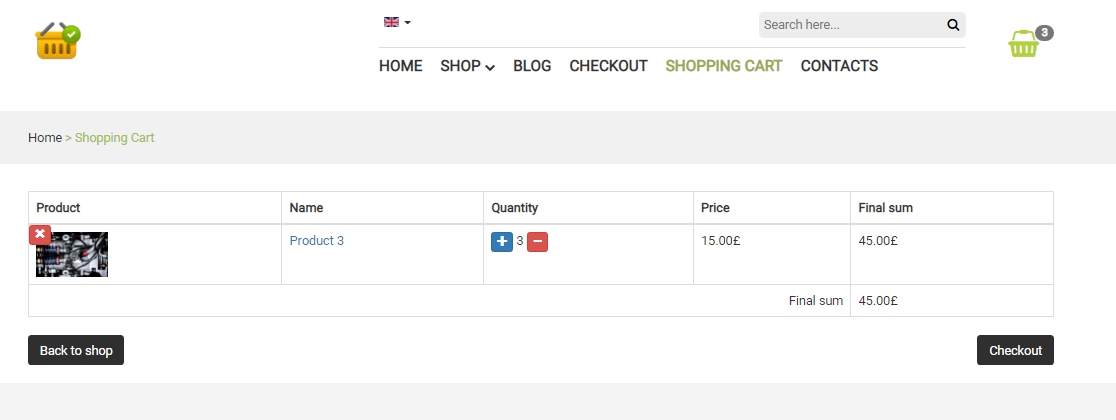
The components of Selenium are Selenium IDE, Selenium RC, Selenium Grid, and Selenium WebDriver. From all the mentioned components of Selenium, the component "Selenium IDE" is mainly a document tool. It's basically a browser add-on or plugin that works with Firefox and Chrome and rapidly generates tests using its record and playback functions. Users do not need to know any test software to ensure proper induction testing. When dealing with "Selenium RC", users must be able to communicate coherently in one language.





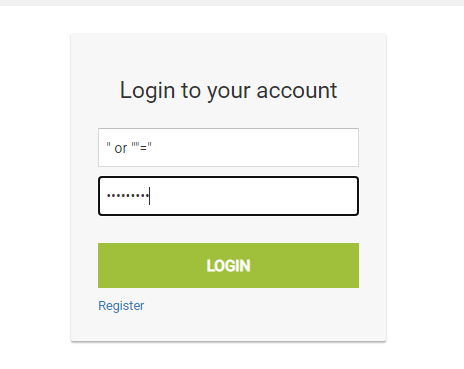
**Figure 6: Cart quantity add/remove testing  
Source: Google Chrome**

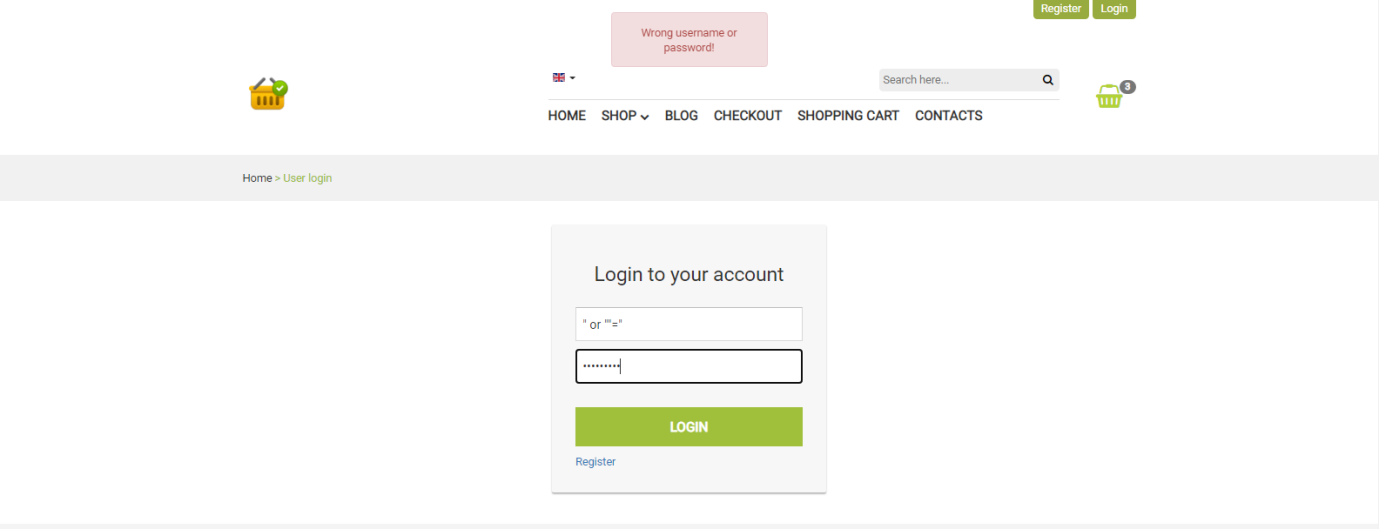
Users may use this platform to construct responsive idea experiments in a number of programming languages (Kim *et al*. 2018). Client server modules are the two main components of **"Selenium RC".** It has a variety of architectural features as well as restrictions. So, in other words, it can be said that "Selenium WebDriver" is a more advanced version of Selenium RC. This was released to the industry to address the flaws in Selenium RC. Because this is a more complex form of RC, it has a very different composition than RC. "Selenium WebDriver", like Selenium RC, supports a broad range of networks and only requires an understanding of one language. "Selenium Grid" is a framework for running test cases across several browsers, PCs, and software applications at the same time.

****

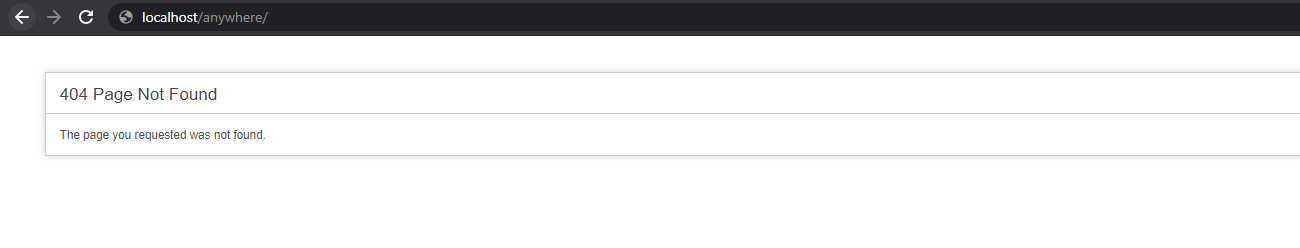
**Figure 7: Pricing calculation testing  
Source: Google Chrome**

This approach makes evaluating cross-browser accessibility much easier. Grid 1 and Grid 2 are the two variations of this grid. Since the Selenium programmers constructed language interfaces to make multiple languages acceptable, Selenium provides a wide range of packages and others. If one wishes to utilize the browser engine, for instance, one may utilize the Python Keys. Users may get all of the language interfaces from the official website of Selenium (Garousi*et al*. 2017).

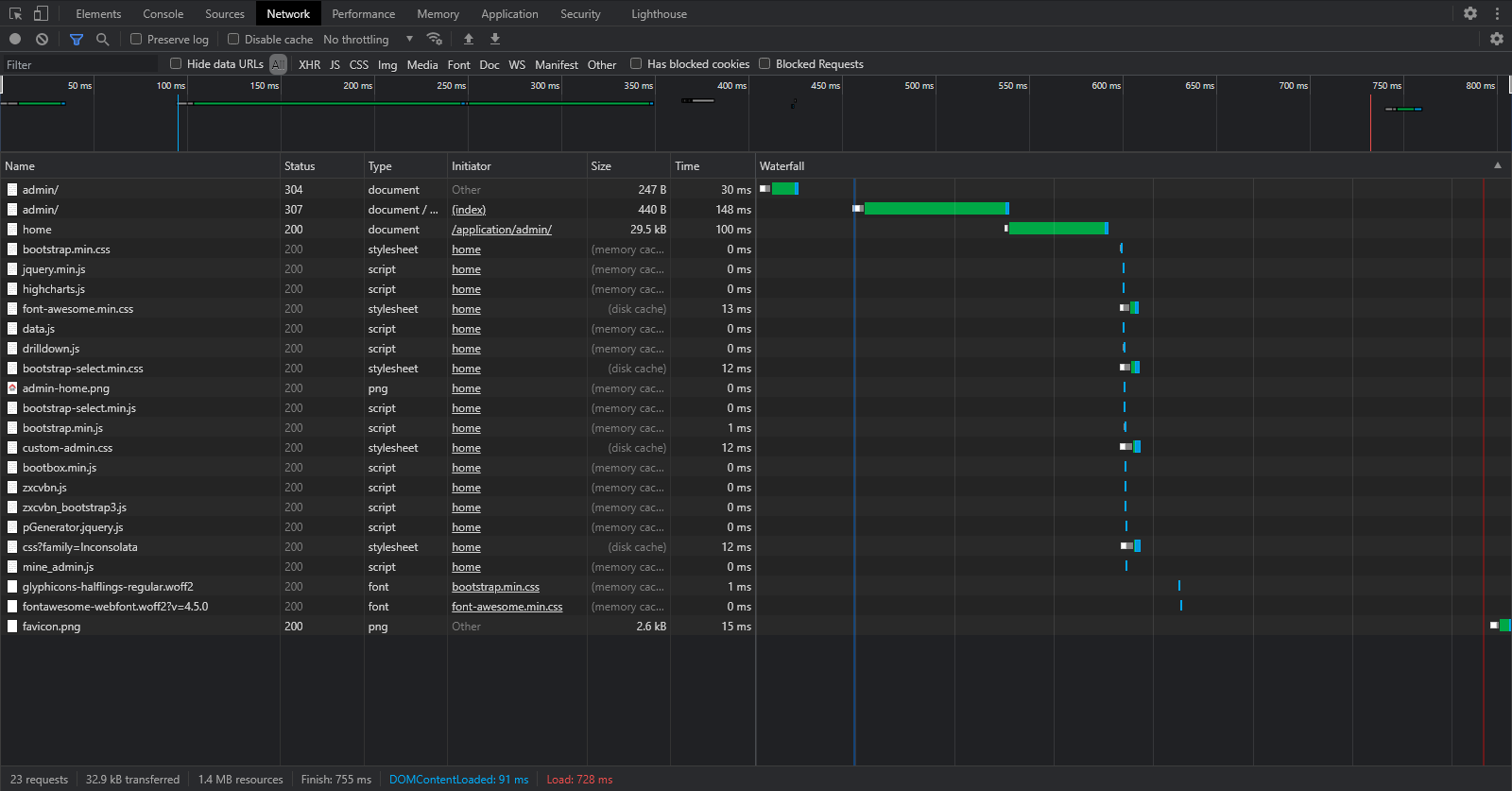




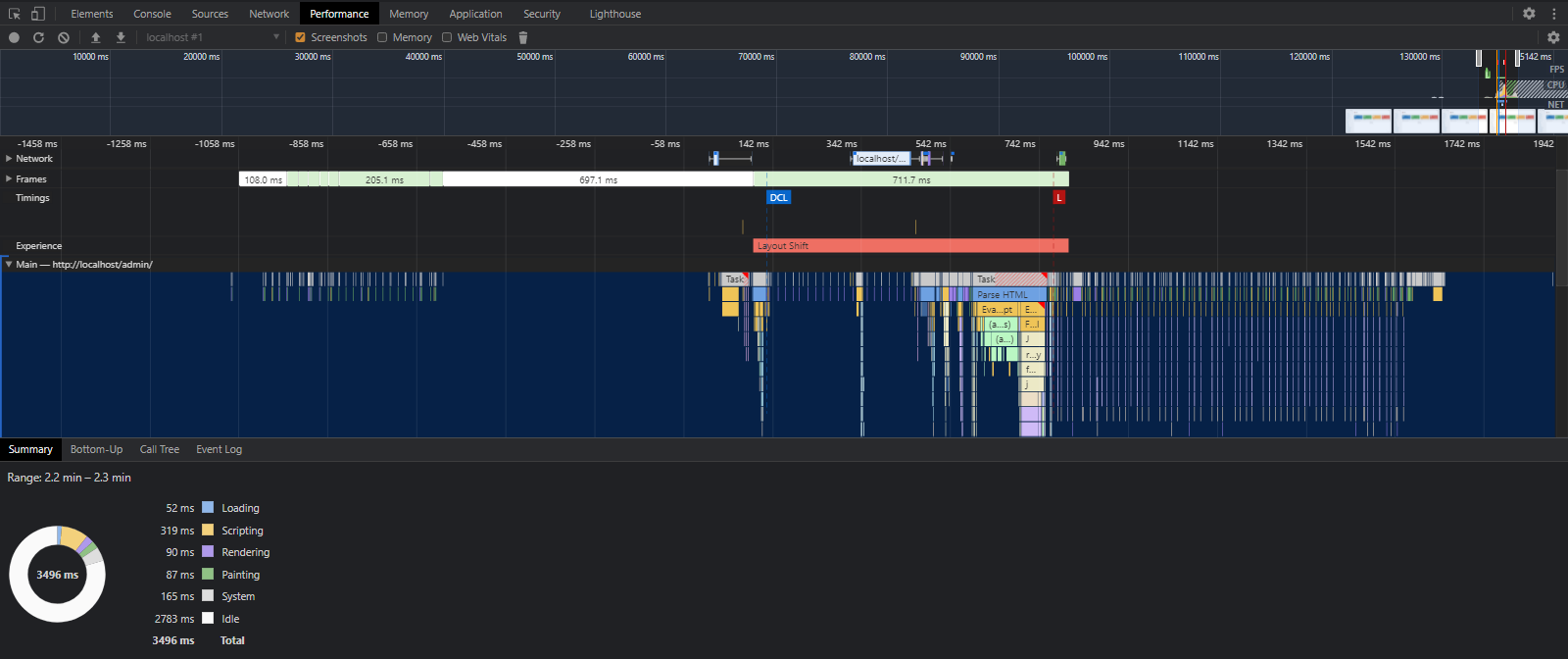
**Figure 8: SQL Injection Testing  
Source: Google Chrome**



**Figure 9: Testing of web app while triggering wrong web URL  
Source: Google Chrome**



**Figure 10: Resource loading testing  
Source: Chrome web-developer tools**

****

**Figure 11: Performance testing  
Source: Chrome web-developer tools**

The steps that are performed while testing the developed web application are as follows.

* First and foremost, a Web driver instance is generated.
* After that, the navigation to the web page is being done.
* On the website page, an HTML element may be found.
* The element of the website is the target of the action.
* Then, the browser is responsible for anticipating the answer.
* The tests and report's findings are carried out in accordance with a research system.
* Finally, the testing is completed once all of the processes have been completed.

# Reference List

**Journal**

Alghamdi, A.M. and Eassa, F.E., 2019. Software testing techniques for parallel systems: A survey. Int. J. Comput. Sci. Netw. Secur., 19(4), pp.176-186.

Butt, F.L., Bhatti, S.N., Sarwar, S., Jadi, A.M. and Saboor, A., 2017. Optimized order of software testing techniques in agile process–A systematic approach. International Journal of Advanced Computer Science and Applications (IJACSA), 8(1).

Garousi, V. and Felderer, M., 2017. Worlds apart: industrial and academic focus areas in software testing. IEEE Software, 34(5), pp.38-45.

Hanna, M., Aboutabl, A.E. and Mostafa, M.S.M., 2018. Automated software testing framework for web applications. International Journal of Applied Engineering Research, 13(11), pp.9758-9767.

Kim, Y., Hong, S., Ko, B., Phan, D.L. and Kim, M., 2018, April. Invasive software testing: Mutating target programs to diversify test exploration for high test coverage. In 2018 IEEE 11th International Conference on Software Testing, Verification and Validation (ICST) (pp. 239-249). IEEE.

Mishra, D., Ostrovska, S. and Hacaloglu, T., 2017. Exploring and expanding students’ success in software testing. Information Technology & People.