An

INTERNSHIP REPORT

On

SOWTWARE TESTING AUTOMATION

Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
CIVIL ENGINEERING

By

P. LAHIRI SAI TEJA

(Roll No. 21AK5A0125)

Under the Mentorship of
Sri. T. Sai Krishna, M. Tech (PhD)
Assistant Professor, Civil Engineering Department

DEPARTMENT OF CIVIL ENGINEERING

ANNAMACHARYA INSTITUTE OF TECHNOLOGY AND SCIENCES, TIRUPATI
(AUTONOMOUS)
Approved by AICTE, New Delhi & Permanent Affiliation to JNTUA, Anantapuramu.
Three B. Tech Programmes (CIVIL, ECE & CSE) are accredited by NBA, New Delhi.
Accredited by NAAC with 'A' Grade, Bangalore. Accredited by Institution of Engineers (India), KOLKATA.
A-grade awarded by AP Knowledge Mission. Recognized under sections 2(f) & 12(B) of UGC Act 1956.
Venkatapuram (V), Renigunta (M), Tirupati, Tirupati District, Andhra Pradesh – 517520.

2023-2024

ANNAMACHARYA INSTITUTE OF TECHNOLOGY AND SCIENCES, TIRUPATI
(AUTONOMOUS)
DEPARTMENT OF CIVIL ENGINEERING

CERTIFICATE

This is to certify that Mr. **P. LAHIRI SAI TEJA (21AK5A0125)** has carried out Virtual Internship on "**SOFWARE TESTING AUTOMATION**" and submitted to the Department of Civil Engineering of Annamacharya Institute of Technology and sciences, in partial fulfilment of the requirements for the award of the Degree of **Bachelor of Technology** in Civil Engineering is meeting the Academic Regulations.

T. Sai Krishna, M. Tech (PhD)
Assistant Professor
Department of Civil Engineering

Date:………………..

A. Anil Reddy, M. Tech (PhD)
Assistant Professor
Department of Civil Engineering

Date:……………….

DECLARATION

I am, P. LAHIRI SAI TEJA **(21AK5A0125),** Studying final year B. Tech in Civil Engineering in 2023-24 at Annamacharya Institute of Technology and Sciences, here by declare that this Internship titled "**SOFTWATE TESTING AUTOMATION**" has been done by me. The Internship work carried out is original and has not been submitted to any other University or Institution for the award of any credits. I promise to meet all the mandatory requirements as specified by the Academic regulations.

Date:

Signature of the Student
P. LAHIRI SAI TEJA
(21AK5A0125)

ACKNOWLEDGEMENT

It is our insightful duty and pleasure to express my gratitude to all those who helped me in competition of this work successfully.

I would like to express my deepest appreciation to **Andhra Pradesh State Council of Higher Education (APSCHE)** for their commitment to the betterment of technical education and the opportunities they have made available to our students.

I look forward to the continued collaboration between **Smart internz and APSCHE** to provide more student Internships to gain hands-on experience and become better-prepared professionals.

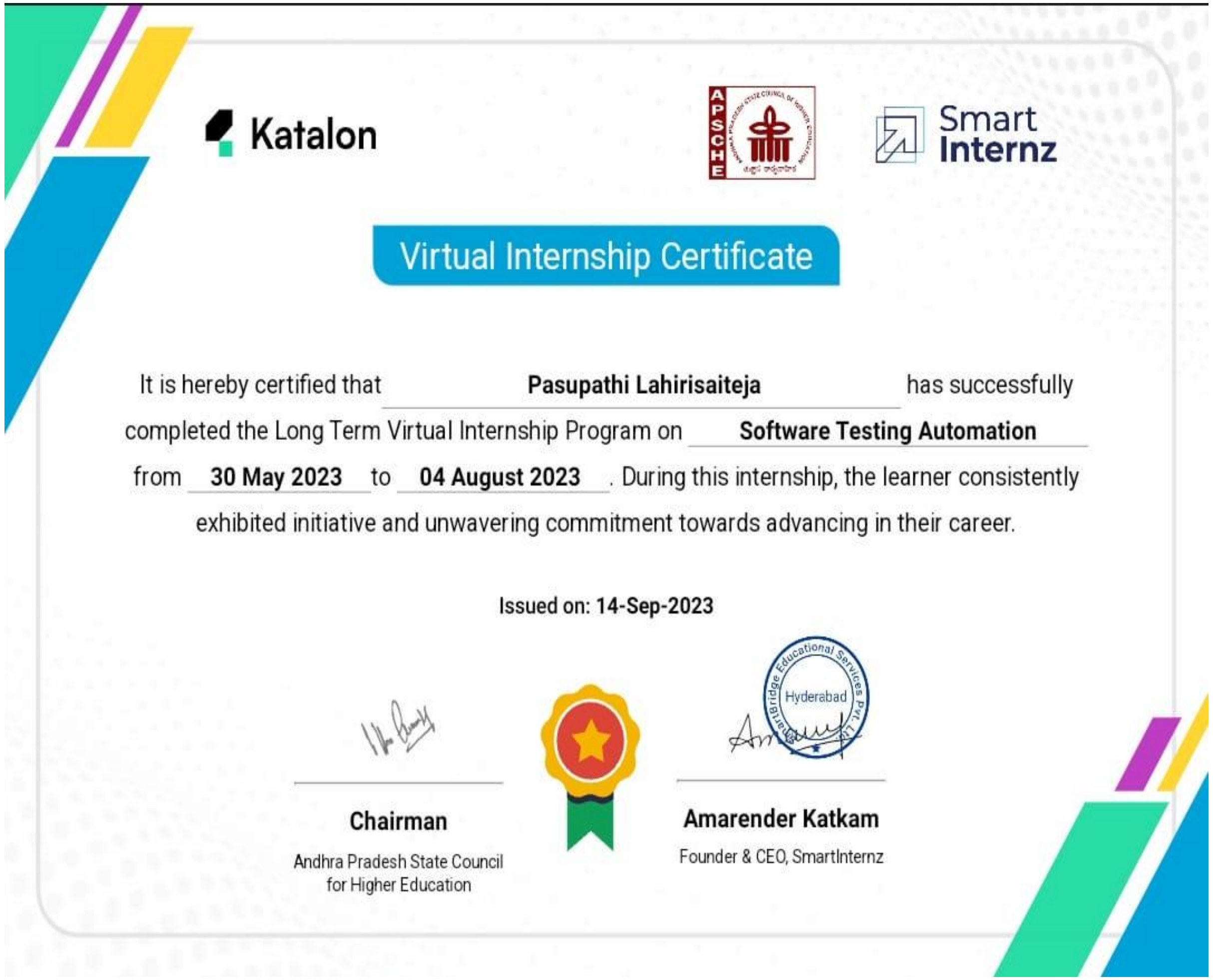
I would like to extend my heartfelt thanks to Principal **Dr. C. Nadhamuni Reddy** for his constant encouragement and support during the Internship period.

I would like to express my heartfelt thanks to **Mr. A. Anil Reddy,** Assistant Professor & HOD-CE, during the progress of Internship for his timely suggestions and help in spite of his busy schedule.

My heartfelt thanks to my Internship Mentor **Mr. T. Sai Krishna,** Assistant Professor Department of CE, for his valuable guidance and suggestions in analysis and testing throughout the period, till the end of Internship work completion.

Finally, I would like to express my sincere thanks to faculty members of Civil Engineering Department, Lab Technicians, Internship company trainers and friends, one and all that has helped me to complete the Internship successfully.

P. LAHIRI SAI TEJA
(Roll Number: 21AK5A0125)



Katalon

APSCHE

Smart Internz

Virtual Internship Certificate

It is hereby certified that **Pasupathi Lahirisaiteja** has successfully completed the Long Term Virtual Internship Program on **Software Testing Automation** from **30 May 2023** to **04 August 2023** . During this internship, the learner consistently exhibited initiative and unwavering commitment towards advancing in their career.

Issued on: **14-Sep-2023**

Chairman
Andhra Pradesh State Council
for Higher Education

SmartBridge Educational Services Pvt. Ltd.
Hyderabad

Amarender Katkam
Founder & CEO, SmartInternz

ABSTRACT

Automation testing is an emerging field that draws maximum benefits with minimum efforts. It is to increase quality and reliability of software. Selenium is a set of software tools that supports test automation. Selenium is a well-established testing framework and it is used with a large number of browsers. As Selenium supports a variety of programming languages, it becomes easier for the tester to write scripts in his preferred programming language. But the professional must have adequate expertise in the specific language to write test script and for each web page its own test case script needs to be written.

Hence, we are designing a generic framework which can be used for automation testing of web elements on the web pages. In our framework we have created a utility file that will contain most of the generic functions needed by the user to operate on the webpage. These functions are reusable and can be used multiple times for to check for the web page functionality.

Here any software user can import this generic framework and perform their own tests. This framework is designed using selenium WebDriver Automation Tool, Java platform on Eclipse editor and Firefox browser. This makes the framework user friendly interface for creating and executing test suites. It is portable and platform independent. It also enables to perform different types of testing. JUnit is used for report generation which gives the output of the test i.e. it tells whether the test has passed or failed.

CONTENTS

CHAPTER-1

INTRODUCTION

Introduction

Automation testing has become a recognized domain in the world of software testing. As the name implies, automation testing involves the use of automated tools to carry out test cases with minimal human intervention, then comparing various outcomes and generating test reports.

Automated testing is a crucial part of every Agile team to keep up with the demands for fast but high-quality software projects. By leveraging automation, teams can enhance result efficiency, improve bug detection, and much more.

Thanks to advances in this field, testers and QA engineers save tremendous time and effort on initial testing and projects requiring repeated executions of the same test.

CHAPTER-2

WEEKLY OVERVIEW

Week 1	Orientation
Week 2	First Technical Session and Self-Paced learning
Week 3	Second Technical Session and Project Introduction
Week 4	Third Technical session and Project Development
Week 5	Project Submission
Week 6	Course Completion

CHAPTER-3

MODULES

Automated Testing is a technique where the Tester writes scripts on their own and uses suitable Software or Automation Tool to test the software. It is an Automation Process of a Manual Process. It allows for executing repetitive tasks without the intervention of a Manual Tester.

It is used to automate the testing tasks that are difficult to perform manually.

Automation tests can be run at any time of the day as they use scripted sequences to examine the software.

Automation tests can also enter test data can compare the expected result with the actual result and generate detailed test reports.

The goal of automation tests is to reduce the number of test cases to be executed manually but not to eliminate manual testing.

It is possible to record the test suit and replay it when required.

Why Transform From Manual to Automated Testing?

In the year 1994, An aircraft completing its Routine flight crashed just before landing. This was due to some bug or defect in the Software. The Testers didn’t even care about the final testing and hence this accident happened. So in order to replace for few of the Manual Tests (mandatory), there is a need for Automation Testing. Below are some of the reasons for using automation testing:

- **Quality Assurance:** Manual testing is a tedious task that can be boring and at the same time error-prone. Thus, using automation testing improves the quality of the software under test as more test coverage can be achieved.
- **Error or Bug-free Software:** Automation testing is more efficient for detecting bugs in comparison to manual testing.
- **No Human Intervention:** Manual testing requires huge manpower in comparison to automation testing which requires no human intervention and the test cases can be executed unattended.
- **Increased test coverage:** Automation testing ensures more test coverage in comparison to manual testing where it is not possible to achieve 100% test coverage.
- **Testing can be done frequently:** Automation testing means that the testing can be done frequently thus improving the overall quality of the software under test.

Manual Testing vs. Automated Testing

Below are some of the differences between manual testing and automated testing:

Parameters	Manual Testing	Automated Testing
Reliability	Manual testing is not accurate at all times due to human error, thus it is less reliable.	Since it is performed by third-party tools and/or scripts, therefore it is more reliable.
Investment	Heavy investment in human resources.	Investment in tools rather than human resources.
Time efficiency	Manual testing is time-consuming due to human intervention where test cases are generated manually.	Automation testing is time-saving as due to the use of the tools the execution is faster in comparison to manual testing.
Programming knowledge	There is no need to have programming knowledge to write the test cases.	It is important to have programming knowledge to write test cases.
Regression testing	There is a possibility that the test cases executed the first time will not be able to catch the regression bugs due to the frequently changing requirements.	When there are changes in the code, regression testing is done to catch the bugs due to changes in the code.

Types of Automation Testing

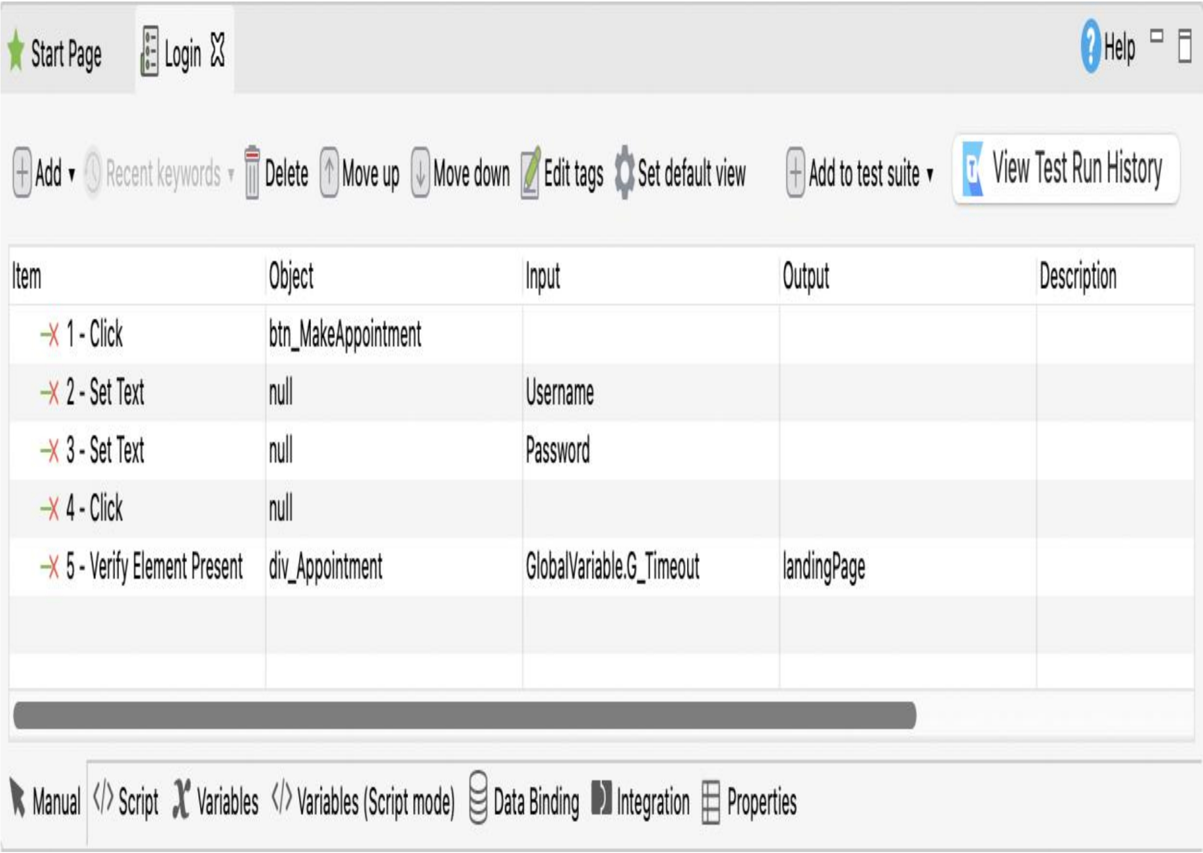
Below are the different types of automation testing:

- **Unit testing:** Unit testing is a phase in software testing to test the smallest piece of code known as a unit that can be logically isolated from the code. It is carried out during the development of the application.
- **Integration testing: component:** Integration testing is a phase in software testing in which individual software components are combined together and tested as a group. It is carried out to check the compatibility of the component with the specified functional requirements.
- **Smoke testing:** Smoke testing is a type of software testing that determines whether the built software is stable or not. It is the preliminary check of the software before its release in the market.
- **Performance testing:** Performance testing is a type of software testing that is carried out to determine how the system performs in terms of stability and responsiveness under a particular load.
- **Regression testing:** Regression testing is a type of software testing that confirms that previously developed software still works fine after the change and that the change has not adversely affected existing features.
- **Security testing:** Security testing is a type of software testing that uncovers the risks, and vulnerabilities in the security mechanism of the software application. It helps an organization to identify the loopholes in the security mechanism and take corrective measures to rectify the gaps in security.
- **Acceptance testing:** Acceptance testing is the last phase of software testing that is performed after the system testing. It helps to determine to what degree the application meets end users' approval.
- **API testing:** API testing is a type of software testing that validates the Application Programming Interface(API) and checks the functionality, security, and reliability of the programming interface.
  UI Testing: UI testing is a type of software testing that helps testers ensure that all the fields, buttons, and other items on the screen function as desired.

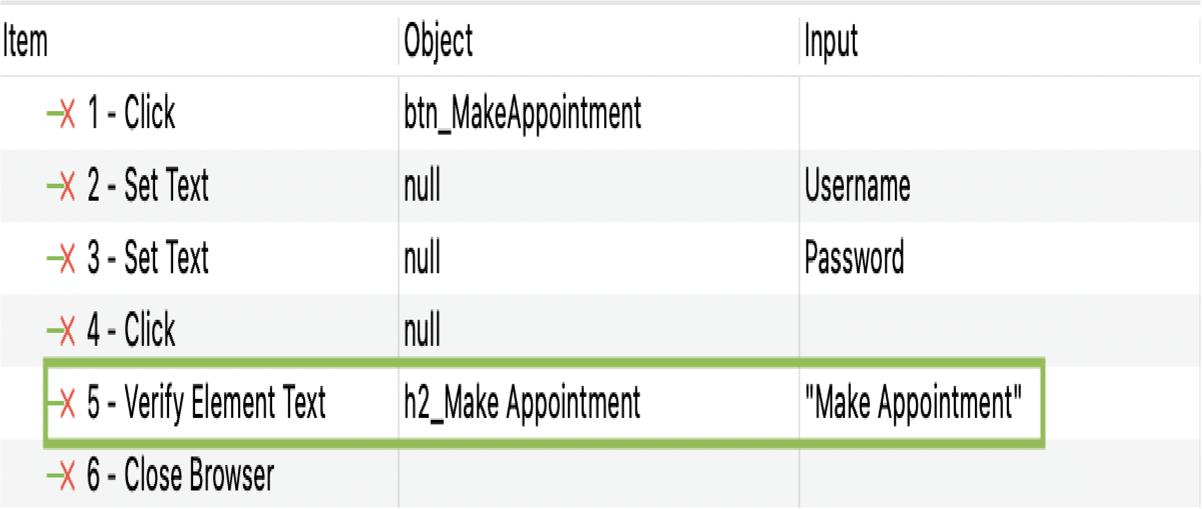
Test Automation Frameworks

Some of the most common types of automation frameworks are:

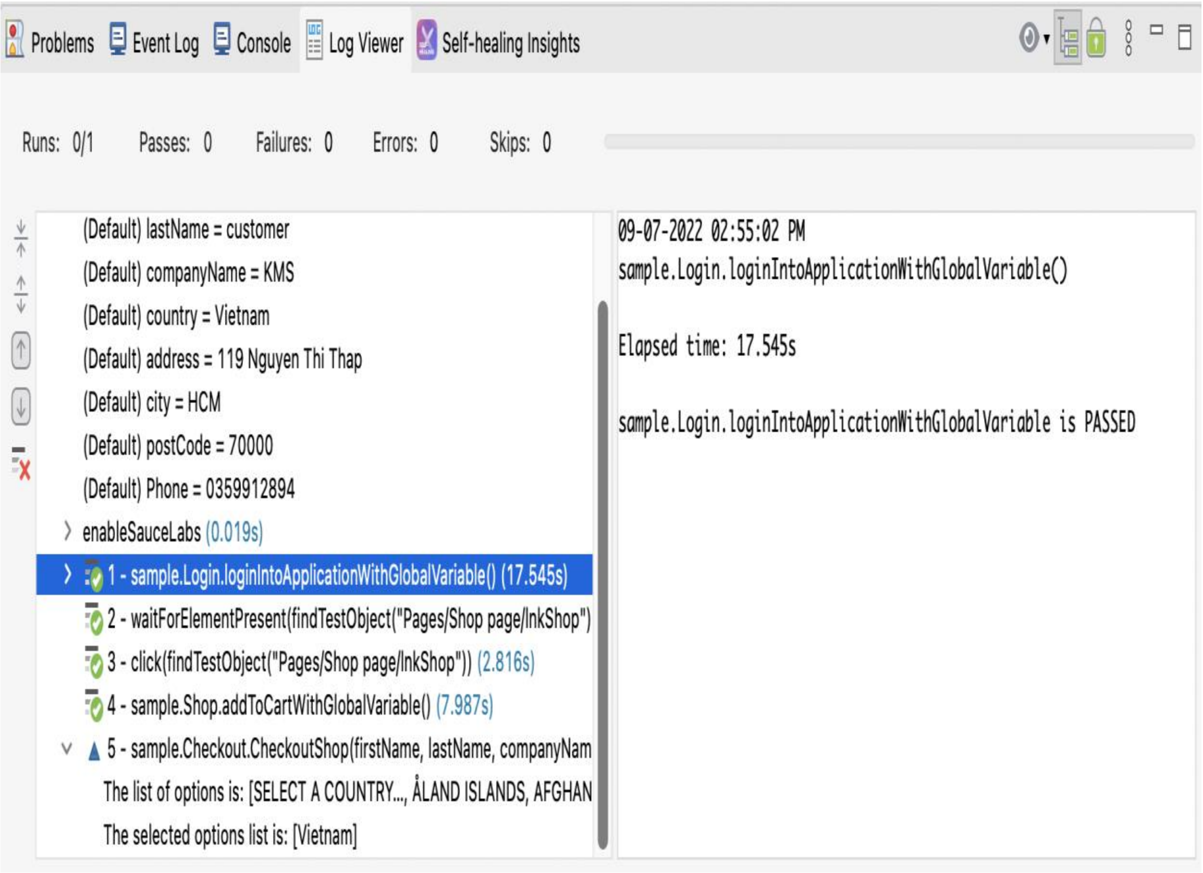
- **Linear framework:** This is the most basic form of framework and is also known as the record and playback framework. In this testers create and execute the test scripts for each test case. It is mostly suitable for small teams that don't have a lot of test automation experience.
- **Modular-Based Framework:** This framework organizes each test case into small individual units known's as modules each module is independent of the other, having different scenarios but all modules are handled by a single master script. This approach requires a lot of pre-planning and is best suited for testers who have experience with test automation.
- **Library Architecture Framework:** This framework is the expansion of a modular-based framework with few differences. Here, the task is grouped within the test script into functions according to a common objective. These functions are stored in the library so that they can be accessed quickly when needed. This framework allows for greater flexibility and reusability but creating scripts takes a lot of time so testers with experience in automation testing can benefit from this framework.



Item	Object	Input	Output	Description
1 - Click	btn_MakeAppointment			
2 - Set Text	null	Username		
3 - Set Text	null	Password		
4 - Click	null			
5 - Verify Element Present	div_Appointment	GlobalVariable.G_Timeout	landingPage	



Item	Object	Input
1 - Click	btn_MakeAppointment	
2 - Set Text	null	Username
3 - Set Text	null	Password
4 - Click	null	
5 - Verify Element Text	h2_Make Appointment	"Make Appointment"
6 - Close Browser		



Runs: 0/1   Passes: 0   Failures: 0   Errors: 0   Skips: 0

```
(Default) lastName = customer
(Default) companyName = KMS
(Default) country = Vietnam
(Default) address = 119 Nguyen Thi Thap
(Default) city = HCM
(Default) postCode = 70000
(Default) Phone = 0359912894
> enableSauceLabs (0.019s)
> 1 - sample.Login.loginIntoApplicationWithGlobalVariable() (17.545s)
  2 - waitForElementPresent(findTestObject("Pages/Shop page/lnkShop")
  3 - click(findTestObject("Pages/Shop page/lnkShop")) (2.816s)
  4 - sample.Shop.addToCartWithGlobalVariable() (7.987s)
v 5 - sample.Checkout.CheckoutShop(firstName, lastName, companyNam
    The list of options is: [SELECT A COUNTRY..., ÅLAND ISLANDS, AFGHAN
    The selected options list is: [Vietnam]
```

```
09-07-2022 02:55:02 PM
sample.Login.loginIntoApplicationWithGlobalVariable()

Elapsed time: 17.545s

sample.Login.loginIntoApplicationWithGlobalVariable is PASSED
```

Which Tests to Automate?

Below are some of the parameters to decide which tests to automate:

- **Monotonous test:** Repeatable and monotonous tests can be automated for further use in the future.
- **A test requiring multiple data sets:** Extensive tests that require multiple data sets can be automated.
- **Business critical tests:** High-risk business critical test cases can be automated and can be scheduled to run regularly.
- **Determinant test:** Determinant test cases where it is easy for the computer to decide whether the test is failed or not can be automated.

Tedious test: Test cases that involve repeatedly doing the same action can be automated so that the computer can do the repetitive task as humans are very poor at performing the repetitive task with efficiency, there increase the chances of error.

Automation Testing Process

1. **Test Tool Selection:** There will be some criteria for the Selection of the tool. The majority of the criteria include: Do we have skilled resources to allocate for automation tasks, Budget constraints, and Do the tool satisfies our needs?
2. **Define Scope of Automation:** This includes a few basic points such as the Framework should support Automation Scripts, Less Maintenance must be there, High Return on Investment, Not many complex Test Cases
3. **Planning, Design, and Development:** For this, we need to Install particular frameworks or libraries, and start designing and developing the test cases such as NUnit, JUnit, QUnit, or required Software Automation Tools
4. **Test Execution:** Final Execution of test cases will take place in this phase and it depends on Language to Language for .NET, we'll be using NUnit, for Java, we'll be using JUnit, for JavaScript, we'll be using QUnit or Jasmine, etc.
5. **Maintenance:** Creation of Reports generated after Tests and that should be documented so as to refer to that in the future for the next iterations.

Criteria to Select Automation Tool

Following are some of the criteria for selecting the automation tool:

- **Ease of use:** Some tools have a steep learning curve, they may require users to learn a completely new scripting language to create test cases and some may require users to maintain a costly and large test infrastructure to run the test cases.
- **Support for multiple browsers:** Cross-browser testing is vital for acceptance testing. Users must check how easy it is to run the tests on different browsers that the application supports.
- **Flexibility:** No single tool framework can support all types of testing, so it is advisable to carefully observe what all tool offers and then decide.

- **Ease of analysis:** Not all tools provide the same sort of analysis. Some tools have a nice dashboard feature that shows all the statistics of the test like which test failed and which test passed. On the other hand, there can be some tools that will first request users to generate and download the test analyses report thus, not much user-friendly. It depends entirely on the tester, project requirement, and budget to decide which tool to use.
- **Cost of tool:** There are some tools that are free and some are commercial tools but there are many other factors that need to be considered before making a decision whether to use free or paid tools. If a tool takes a lot of time to develop test cases and it is a business-critical process that is at stake then it is better to use paid tool that can generate test cases easily and at a faster rate.
- **Availability of support:** Free tools mostly provide community support on the other hand commercial tools provides customer support, and training material like tutorials, videos, etc. Thus, it is very important to keep in mind the complexity of the tests before selecting the appropriate tool

Best Practices for Test Automation

Below are some of the best practices for test automation that can be followed:

- **Plan self-contained test cases:** It is important to ensure that the test is clearly defined and well-written. The test cases should be self-contained and easy to understand.
- **Plan the order to execute tests:** Planning the test in the manner that the one test creates the state for the second test can be beneficial as it can help to run test cases in order one after another.
- **Use tools with automatic scheduling:** If possible use tools that can schedule testing automatically according to a schedule.
- **Set up an alarm for test failure:** If possible select a tool that can raise an alarm when a test failure occurs. Then a decision needs to be made whether to continue with the test or abort it.
- **Reassess test plans as the app develops and changes:** It is important to continuously reassess the test plan as there is no point in wasting resources in testing the legacy features in the application under test.

Popular Automation Tools

- **Selenium:** Selenium is an automated testing tool that is used for Regression testing and provides a playback and recording facility. It can be used with frameworks like JUnit and Test NG. It provides a single interface and lets users write test cases in languages like Ruby, Java, Python, etc.
- **QTP:** Quick Test Professional (QTP) is an automated functional testing tool to test both web and desktop applications. It is based on the VB scripting language and it provides functional and regression test automation for software applications.
- **Sikuli:** It is a GUI-based test automation tool that is used for interacting with elements of web pages. It is used to search and automate graphical user interfaces using screenshots.

- **Appium:** Apium is an open-source test automation framework that allows QAs to conduct automated app testing on different platforms like iOS, Android, and Windows SDK.
- **Jmeter:** Apache JMeter is an open-source Java application that is used to load test the functional behavior of the application and measure the performance.

Advantages of Automation Testing

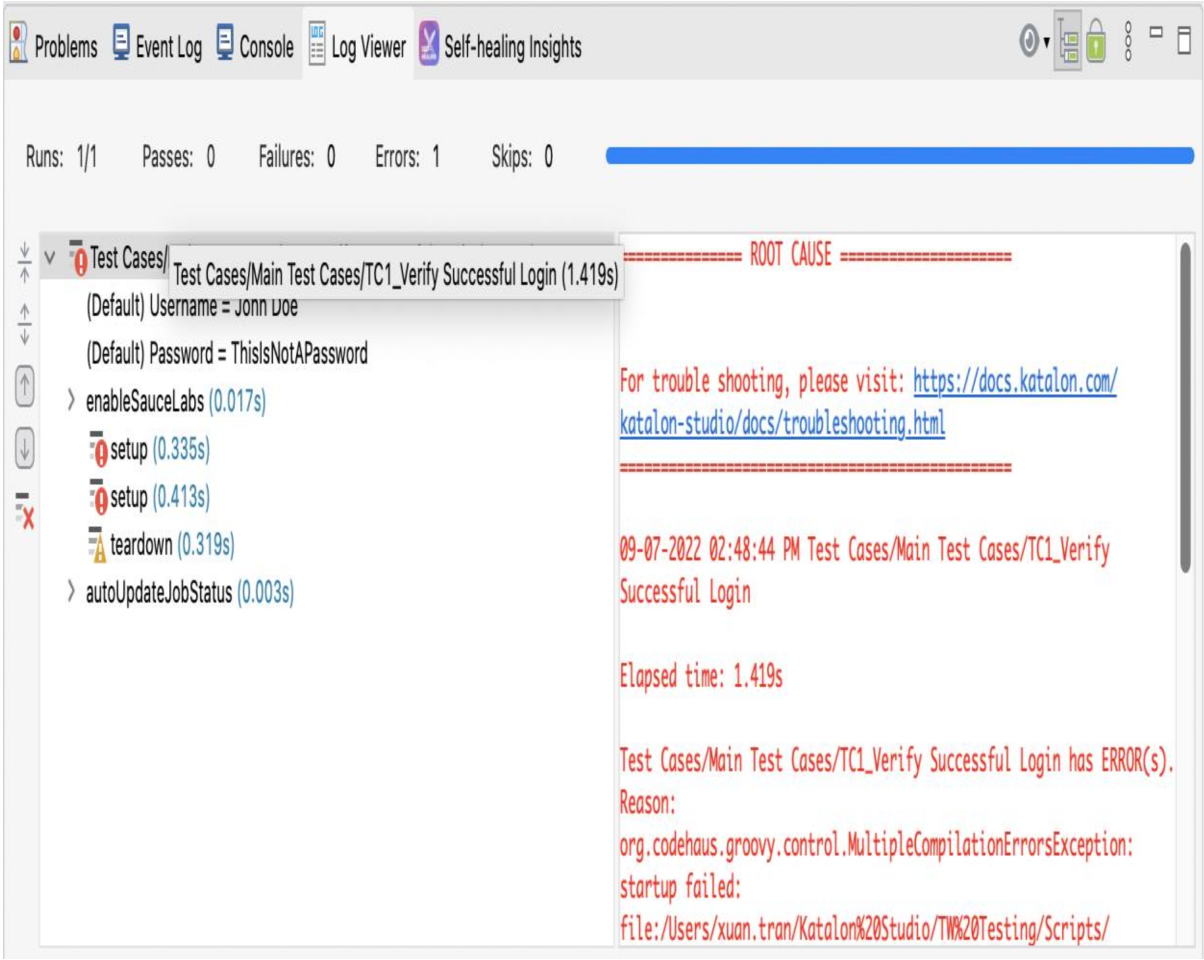
- **Simplifies Test Case Execution:** Automation testing can be left virtually unattended and thus it gives an opportunity to monitor the results at the end of the process. . Thus, simplifying the overall test execution and increasing the efficiency of the application.
- **Improves Reliability of Tests:** Automation testing ensures that there is equal focus on all the areas of the testing, thus ensuring the best quality end product.
- **Increases amount of test coverage:** Using automation testing, more test cases can be created and executed for the application under test. Thus, resulting in higher test coverage and the detection of more bugs. This allows for the testing of more complex applications and more features can be tested.
- **Minimizing Human Interaction:** In automation testing, everything is automated from test case creation to execution thus there are no changes for human error due to neglect. This reduces the necessity for fixing glitches in the post-release phase.
- **Saves Time and Money:** The initial investment for automation testing is on the higher side but it is cost-efficient and time-efficient in long run. This is due to the reduction in the amount of time required for test case creation and execution which contributes to the high quality of work.
- **Earlier detection of defects:** Automation testing documents the defects, thus making it easier for the development team to fix the defect and give a faster output. The earlier defect is identified, the more easier and cost-efficient it is to fix the defects.

Disadvantages of Automation Testing

- **High initial cost:** Automation testing in the initial phases requires a lot of time and money investment. It requires a lot of effort for selecting the tool and designing customized software.
- **100% test automation is not possible:** Generally, the effort is to automate all the test cases but in practical real situations not all test cases can be automated there are some test cases that require human intervention for careful observation. There is always a human factor, i.e., it can’t test everything like humans(design, usability, etc.).
- **Not possible to automate all testing types:** It is not possible to automate tests that verify the user-friendliness of the system. Similarly, if we talk about the graphics or sound files, even their testing cannot be automated as automated tests typically use textual descriptions to verify the output.
- **Programming knowledge is required:** Every automation testing tool uses any one of the programming languages to write test scripts. Thus, it is mandatory to have programming knowledge for automation testing.
- **False positives and negatives:** Automation tests may sometimes fail and reflect that there is some issue in the system but actually there is no issue present and in some

cases, it may generate false negatives if tests are designed to verify that some functionality exists and not to verify that it works as expected

Investigation of error logs

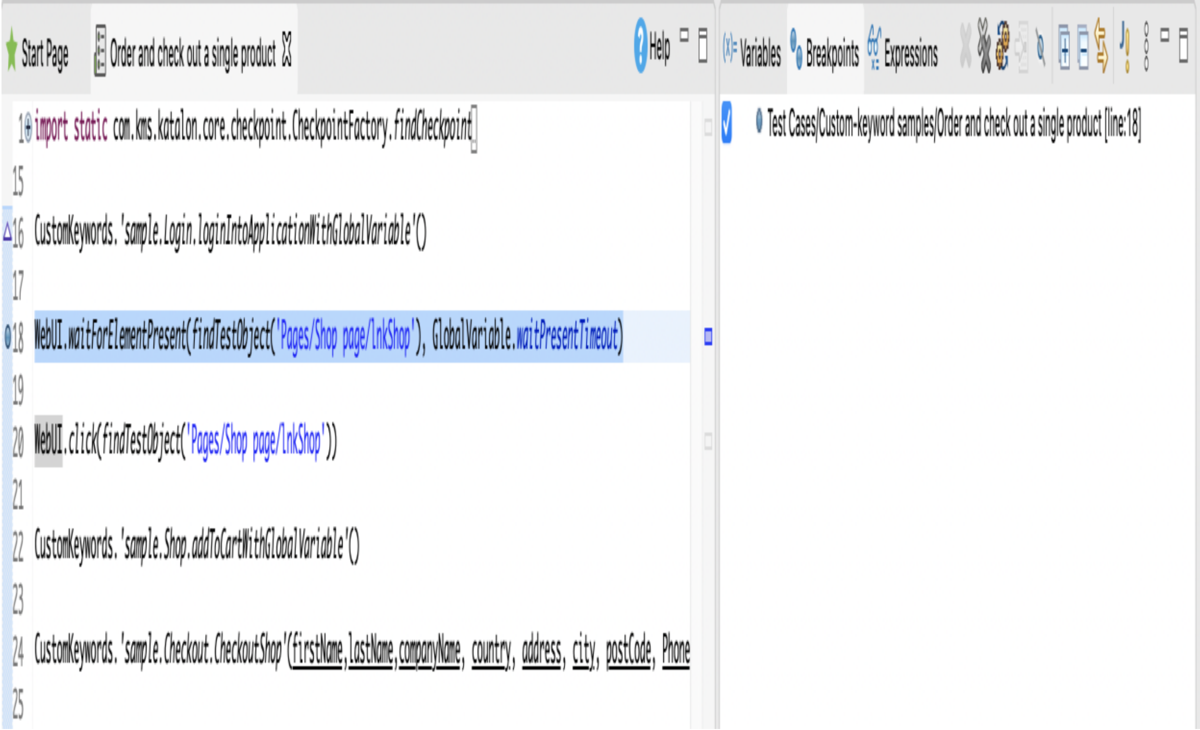


Runs: 1/1 Passes: 0 Failures: 0 Errors: 1 Skips: 0

```
v Test Cases/  Test Cases/Main Test Cases/TC1_Verify Successful Login (1.419s)
    (Default) Username = John Doe
    (Default) Password = ThisIsNotAPassword
  > enableSauceLabs (0.017s)
    setup (0.335s)
    setup (0.413s)
    teardown (0.319s)
  > autoUpdateJobStatus (0.003s)
```

```
============== ROOT CAUSE ===================

For trouble shooting, please visit: https://docs.katalon.com/
katalon-studio/docs/troubleshooting.html
============================================

09-07-2022 02:48:44 PM Test Cases/Main Test Cases/TC1_Verify
Successful Login

Elapsed time: 1.419s

Test Cases/Main Test Cases/TC1_Verify Successful Login has ERROR(s).
Reason:
org.codehaus.groovy.control.MultipleCompilationErrorsException:
startup failed:
file:/Users/xuan.tran/Katalon%20Studio/TW%20Testing/Scripts/
```

Debugging



```
import static com.kms.katalon.core.checkpoint.CheckpointFactory.findCheckpoint

CustomKeywords.'sample.Login.loginIntoApplicationWithGlobalVariable'()

WebUI.waitForElementPresent(findTestObject('Pages/Shop page/lnkShop'), GlobalVariable.waitPresentTimeout)

WebUI.click(findTestObject('Pages/Shop page/lnkShop'))

CustomKeywords.'sample.Shop.addToCartWithGlobalVariable'()

CustomKeywords.'sample.Checkout.CheckoutShop'(firstName,lastName,companyName, country, address, city, postCode, Phone

```

☑ Test Cases/Custom-keyword samples/Order and check out a single product [line: 18]

CHAPTER-4

PROJECT

Programming For Automation Testing

```
<html>

<head>

   <link href="Style.css" rel="stylesheet" />

   <title>Amazon Login Page Html Code With CSS (Free Download)!</title>

</head>

<body>

   <div id="header_wrapper">

      <div id="header">



         <form action="post">

            <li>Email or Phone<br><input type="text" name="email"></li>

            <li>Password<br><input type="password" name="password"><br><a
href="">Forgotten account?</a></li>

            <li><input type="submit" name="login" value="Log In"></li>

         </form>



      </div>

   </div>



   <div id="wrapper">



      <div id="div1">
```

```
    </div>


    <div id="div2">
      <h1>Create an account</h1>


      <p>It's free and always will be.</p>
      <li><input type="text" placeholder="First Name" id="Firstname"><input type="text"
placeholder="Surname" id="surname"></li>
      <li><input type="text" placeholder="Mobile number or email"></li>
      <li><input type="password" placeholder="New password"></li>


      <p>Birthday</p>


      <li>
        <select><option>Day</option></select>
        <select><option>Month</option></select>
        <select><option>Year</option></select>
        <a href="">Why do I need to provide my date of birth?</a>
      </li>


      <li><input type="radio">Female <input type="radio">Male</li>
      <li id="terms">By clicking Create an account, you agree to our <a
href="">Terms</a> and that <br>you have read our <a href="">Data Policy</a>, including
our <a href="">Cookie Use</a>.</li>
      <li><input type="submit" value="Create an account"></li>
      <li id="create_page"><a href="">Create a Page</a> for a celebrity, band or
business.</li>
    </div>
    </div>

    <div id="footer_wrapper">

        <div id="footer1">

            <a href="#">Sign Up</a><a href="#">Log In</a><a href="#">Messenger</a><a href="#">DotNetTec</a><a href="#">Mobile</a><a href="#">Find Friends</a>
            <a href="#">Badges</a><a href="#">People</a><a href="#">Pages</a><a href="#">Places</a><a href="#">Games</a><a href="#">Locations</a>
            <a href="">Celebrities</a><a href="">Groups</a><a href="">Moments</a><a href="">About</a>
            <a href="">Create Advert</a><a href="">Create Page</a><a href="">Developers</a>
            <a href="">Careers</a><a href="">Privacy</a><a href="">Cookies</a><a href="">Ads</a><a href="">Terms</a><a href="">Help</a>

        </div>
    </div>
</body>
</html>
```

CSS Code:)

```
body {
    text-align: center;
    width: 100%;
    margin: 0 auto;
    padding: 0px;
    font-family: "lucida grande",tahoma,verdana,arial,sans-serif;
    background: linear-gradient(white, #D3D8E8);
}

#header_wrapper {
    width: 100%;
    min-width: 980px;
    background-color: #4c66a4;
}

#header {
    width: 980px;
    margin: 0px auto;
    padding: 0px;
    height: 85px;
}

    #header li {
        list-style-type: none;
        float: left;
        text-align: left;
        color: white;
    }

    #header #sitename {
        margin-top: 25px;
    }
        #header #sitename a {
            color: white;
            text-decoration: none;
            font-size: 30px;
            font-weight: 900;
        }

    #header form {
        margin-top: 15px;
        float: right;
    }

        #header form li {
            font-size: 13px;
            margin-left: 15px;
        }

            #header form li a {
                color: #A9BCF5;
                text-decoration: none;
            }

        #header form input[type="text"] {
            margin-top: 3px;
            margin-bottom: 3px;
            width: 150px;
            border: 1px solid #08298A;
```

```
            height: 25px;
            padding-left: 3px;
        }

        #header form input[type="password"] {
            margin-top: 3px;
            margin-bottom: 3px;
            width: 150px;
            border: 1px solid #08298A;
            height: 25px;
            padding-left: 3px;
        }

        #header form input[type="submit"] {
            height: 25px;
            margin-top: 20px;
            background-color: #084B8A;
            color: white;
            border: 1px solid #08298A;
        }

#wrapper {
    margin: 0 auto;
    padding: 0px;
    text-align: center;
    width: 980px;
}
```

```
#wrapper div {
   float: left;
   font-family: helvetica, arial, sans-serif;
}

#wrapper #div1 {
   margin-top: 30px;
   width: 590px;
   text-align: left;
}

   #wrapper #div1 p {
      font-size: 20px;
      font-family: arial;
      font-weight: bold;
      margin: 0px;
      color: #0e385f;
   }

#wrapper #div2 {
   margin-top: 10px;
   width: 390px;
   text-align: left;
}

   #wrapper #div2 h1 {
      margin: 0px;
      font-size: 37px;
      color: #2E2E2E;
   }

   #wrapper #div2 p {
      font-size: 18px;
      color: #2E2E2E;
   }

   #wrapper #div2 li {
      list-style-type: none;
      margin-top: 10px;
   }

      #wrapper #div2 li #firstname {
          width: 49%;
      }

      #wrapper #div2 li #surname {
          margin-left: 2%;
          width: 49%;
      }

      #wrapper #div2 li input[type="text"] {
          width: 100%;
          height: 40px;
          border-radius: 5px;
          padding-left: 10px;
          font-size: 18px;
          border: 1px solid #BDBDBD;
      }

      #wrapper #div2 li input[type="password"] {
          width: 100%;
          height: 40px;
          border-radius: 5px;
          padding-left: 10px;
          font-size: 18px;
          border: 1px solid #BDBDBD;
      }

      #wrapper #div2 li select {
          padding: 4px;
          float: left;
      }

      #wrapper #div2 li a {
          margin-left: 10px;
          width: 150px;
          color: #045FB4;
          text-decoration: none;
          font-size: 11px;
          display: inline-block;
```

```
        vertical-align: middle;
        line-height: 15px;
    }

        #wrapper #div2 li a:hover {
            text-decoration: underline;
        }

#wrapper #div2 li {
    color: #2E2E2E;
    font-size: 18px;
}

#wrapper #div2 #terms {
    color: #424242;
    font-size: 11px;
}

    #wrapper #div2 #terms a {
        display: inline;
        margin: 0px;
    }

#wrapper #div2 li input[type="submit"] {
    width: 205px;
    height: 45px;
    text-align: center;
    font-size: 19px;
    margin-top: 10px;
    margin-bottom: 10px;
    font-family: 'Freight Sans Bold', helvetica, arial, sans-serif !important;
    font-weight: bold !important;
    background: linear-gradient(#67ae55, #578843);
    background-color: #69a74e;
    box-shadow: inset 0 1px 1px #a4e388;
    border-color: #3b6e22 #3b6e22 #2c5115;
    border: 1px solid;
    border-radius: 5px;
    color: #fff;
    cursor: pointer;
    display: inline-block;
    position: relative;
    text-shadow: 0 1px 2px rgba(0,0,0,.5);
}

#wrapper #div2 #create_page {
    color: #424242;
    font-size: 13px;
    font-weight: bold;
}

    #wrapper #div2 #create_page a {
        display: inline;
        margin: 0px;
        font-size: 13px;
      }

#footer_wrapper {
  width: 100%;
  clear: both;
  float: left;
  margin-top: 30px;
  min-width: 995px;
  background-color: white;
  text-align: left;
}

#footer1 {
  width: 980px;
  margin: 0px auto;
  padding: 0px;
  border-bottom: 1px solid #E6E6E6;
  height: 30px;
  line-height: 30px;
  font-size: 12px;
  color: #848484;
}

  #footer1 a {
    color: #365899;
    display: inline;
```

```
        margin-left: 10px;
        text-decoration: none;
    }

        #footer1 a:hover {
            text-decoration: underline;
        }

#footer2 {
    width: 980px;
    margin: 0px auto;
    padding: 0px;
    font-size: 12px;
    color: #848484;
}

    #footer2 a {
        color: #365899;
        display: inline-block;
        margin: 5px;
        margin-left: 0px;
        min-width: 80px;
        text-decoration: none;
    }

        #footer2 a:hover {
            text-decoration: underline;
```

CHAPTER-5

RESULT&CONCLUSION

RESULT

Testing of Amazon by Automation testing is done successfully

CONCLUSION

In conclusion, automation testing is a critical component of any software development process. It ensures the quality and reliability of application while reducing time and effort required to perform a manual testing. With the right tools and approach, organizations can improve the speed and accuracy of their testing, catch bugs earlier in the development cycle, and ultimately deliver better products to their customers.

CHAPTER-6

VERIFIABLE CREDENTIALS

VERIFIABLE LINK OF CERTIFICATE:

https://apsche.smartinternz.com/certificate/student/f2c3b258e9cd8ba16e18f319b3c88c66