

AUTOMATION TESTING

EPAM SYSTEMS

A Training Report

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

Bachelor of Technology

Computer Science and Engineering

LOVELY PROFESSIONAL UNIVERSITY

PHAGWARA, PUNJAB



From 11/01/2023 to Present

SUBMITTED BY

Name of the student: K. Bharath

Registration Number: 11906717

Signature of the student: bharath

SUBMITTED TO

Name of the supervisor: Sakshi

Designation:

Signature of the supervisor:

Student Declaration

To whom so ever it may concern

I, Bharath simha Reddy, Regd.NO: 11906717, hereby declare that the work done by me on "**Automation Testing**" from 11th January-2023 to Present, under the supervision of **Sakshi mam, EPAM Systems**, and Name of Internal supervisor - Sakshi Lovely Professional University, Phagwara, Punjab, is a record of original work for the partial fulfilment of the requirements for the award of the degree Computer Science and Engineering.

Name of the Student: K.Bharath simha Reddy

Registration Number: 11906717

Bharath

Signature of the student:

Dated: 05/05/2023

Declaration by the supervisors

To whom so ever it may concern

This is to certify that K. Bharath simha Reddy., Regd.NO: 11906717, from Lovely Professional University, Phagwara, Punjab, has worked as a trainee in “EPAM SYSTEMS” on “Automation Testing” under my supervision from 11th January, 2023 to present. It is further stated that the work carried out by the student is a record of original work to the best of my knowledge for the partial-fulfillment of the requirements for the award of the B-TECH, Computer Science and Engineering.

Name of External Supervisor

Name of Internal Supervisor:
Sakshi

Designation of the External Supervisor

Assistant Professor
Designation of the Internal Supervisor

Signature of the external Supervisor

Signature of the Internal Supervisor

Dated:

Dated:

Table of Contents

S.No.	Title	Page
1	Declaration by Student	2
2	Declaration by Supervisors	3
3	Undertaking form	5
4	List of Content	4
5	Chapter-I INTRODUCTION OF THE COMPANY	6
6	Chapter-2 INTRODUCTION OF THE PROJECT UNDERTAKEN	14
7	Chapter-3 TECHNOLOGIES LEARNT DURING INTERNSHIP	26
8	Chapter-4 CONCLUSION	53
9	Chapter-5 References	54

Undertaking by the student for submitting Final Certificate of six months Internship/OJT

Reg No: 11906717

Student Name: K. Bharath simha Reddy

Program Name: B Tech, CSE

Batch Year: 2019-2023

Course Code: CSE441

Mobile No: 9390165536

I understand that I have been provisionally allowed to appear for the ETP viva and I hereby declare that since I am on 6 months Internship/OJT, thus I shall submit my final certificate of 12 months Internship/OJT to university after completion of my Internship/OJT but not later than Feb 2023.

I am aware that in case, I am unable to submit the same **till** the above-mentioned date, my final evaluation of internship/OJT shall be discarded by the university, and I grade shall be awarded in the result.

Signature of Student

Signature of TPC-School

Signature of HOS

CHAPTER1

INTRODUCTION TO COMPANY

1.1 About EPAM

Since 1993, EPAM Systems, Inc. (NYSE: EPAM) has leveraged its advanced software engineering heritage to become the foremost global digital transformation services provider – leading the industry in digital and physical product development and digital platform engineering services. Through its innovative strategy; integrated advisory, consulting, and design capabilities; and unique ‘Engineering DNA,’ EPAM’s globally deployed hybrid teams help make the future real for clients and communities around the world by powering better enterprise, education, and health platforms that connect people, optimize experiences, and improve people’s lives. In 2021, EPAM was added to the S&P 500 and included in the list of Forbes Global 2000 companies.

In 2002, EPAM was ranked publicly for the first time as a fast-growing company by Deloitte & Touche. In the same year, the company became the first Russian player on the London Stock Exchange.

In 2012, EPAM was listed on New York Stock Exchange under the moniker EPAM, becoming the first company from Belarus on NYSE.

In 2004, EPAM acquired Fathom Technology, a software development services company based in Budapest, Hungary, expanding its offshore services beyond North America. In 2006, EPAM secured an equity investment from Siguler Guff to fund its competitive growth plans.

In 2006, EPAM acquired VDI, a software development services company with delivery centers in Russia, which expanded the company’s presence in the CIS region. That year, EPAM CEO Arkadiy Dobkin was named one of the Top 25 Most Influential Consultants of the Year by *Consulting Magazine*.

In late 2012, EPAM made two acquisitions – Thoughtcorp, which expanded its service offerings in Agile, business intelligence and mobile, and Empathy Lab, which established a digital engagement practice focusing on customer experience, design and eCommerce.

EPAM made two acquisitions in 2018 to expand its service offerings: Continuum (now EPAM Continuum) and TH_NK to add consulting capabilities and develop its digital and service design practices. Also that year, EPAM launched InfoNgen®, a text analytics and sentiment analysis enterprise software product that uses artificial intelligence.

The company also productized TelescopeAI®, an artificial intelligence-based platform for IT operations and workforce management, which won a 2019 Big Innovation Award presented by the Business Intelligence Group.

In 2019, EPAM joined the Blockchain in Transport Alliance (BiTA). That year, the company launched EPAM Continuum, its service for business, experience and technology consulting.

The company also launched EPAM SolutionsHub, a catalogue of its software products, accelerators and open source platforms. As part of its SolutionsHub launch, EPAM also released the Open Source Contributor Index (OSCI), a tool that ranks the top open-source contributors by a commercial organization.

In August 2021, EPAM expanded its presence in Latin America through the acquisition of Colombia-based S4N, a software development services firm specializing in the design and development of modern software products and enterprise platforms.

In July 2021, EPAM acquired CORE SE, a professional service provider specializing in IT strategy and technology-driven transformations, to further expand its Western European footprint in the DACH region.

In May 2021, EPAM acquired Just-BI, a Netherlands-based consultancy specializing in SAP/S4HANA and enterprise data and analytics program management.

EPAM acquired Israel-based cybersecurity services provider White-Hat Ltd. in May 2021.

In April 2021, EPAM acquired PolSource, a Salesforce Consulting Partner with more than 350 Salesforce specialists across the Americas and Europe.

In December 2021, EPAM joined the S & P 500 American market index.

In May 2021, EPAM Systems ranked 1,804 on the Forbes Global 2000 list.

In 2022, EPAM announced several strategic acquisitions and investments, including a majority stake in U.S. software firm Contino and a minority stake in U.K. AI-driven data analytics firm Noodle.io

Selected by Newsweek as a 2021 Most Loved Workplace, EPAM's global multi-disciplinary teams serve customers in more than 45 countries across five continents. As a recognized leader, EPAM is listed among the top 15 companies in Information Technology Services on the Fortune 1000 and ranked as the top IT services company on Fortune's 100 Fastest-Growing Companies list for the last three consecutive years. EPAM is also listed among Ad Age's Top 25 World's Largest Agency Companies for three consecutive years, and Consulting Magazine named EPAM Continuum a top 20 Fastest-Growing Firm.

61,000 TOTAL EPAMERS (Data for Q1 2022) | 45+ COUNTRIES (Data as of Q1 2022) | \$1.172B Q1 2022 REVENUE (Data as of Q1 2022) | 27% Revenue Growth 5 Year Revenue CAGR (2016-2021)

EPAM India is one of the best Product Development Services companies in India with a 'startup-like culture' and leverages our global Engineering Excellence practices and borderless delivery model to provide unique career opportunities to our employees and best-in-class digital solutions to our customers. Our teams help our customers be more competitive and disruptive in the marketplace. We adopt a global growth strategy, thinking and acting like start-ups, working in multidisciplinary global teams, and delivering results relentlessly.

5,500+ EPAMERS (100% growth in 2021) | 150+ ACTIVE CUSTOMERS | 05 OFFICE LOCATIONS (Hyderabad, Pune, Bangalore, Gurgaon, Chennai).

MARKET POSITION:- EPAM is a leading global provider of software product development and digital platform engineering services to hundreds of Fortune 500 and 1000 clients located around the world, primarily in North America, Europe, Asia and Australia. We focus on building long-term partnerships with customers, enabling them to reimagine their businesses through a digital lens. Our industry expertise includes financial services, travel and consumer, software and hi-tech, business information and media, life sciences and healthcare, as well as other emerging industries.

VALUE PROPOSITION:- EPAM helps its customers thrive in a market constantly challenged by the pressures of digitization, delivering true end-to-end value through our innovative and scalable software solutions, best-in-class business consulting and experience design, and a continually evolving mix of advanced capabilities. Our historical core competency in software development and product engineering, as well as our work with global enterprise leaders and emerging technology companies, created the foundation for our evolution of other offerings, which include advanced technology and intelligent enterprise solutions and innovative digital engagement. This combination, along with our deep industry-specific domain expertise, helps us speed our customers' digital transformation journeys.

EPAM'S UNIQUE APPROACH:- Over the past 25 years, we have helped our customers through each wave of technology change, building solutions that help them level the playing field and stay competitive through constant market disruption. Using our software engineering heritage as a foundation, we continue to disrupt ourselves by deepening our technology experience and adding strategic business and innovation consulting, design thinking and physical-digital capabilities to provide business value through human-centric innovation.

Company's Vision and Mission

EPAM Systems is a digital platform engineering and software development company that has a clear and concise vision and mission statement. Their vision is to be recognized as a leading technology company that delivers innovative solutions to its clients and helps them to transform their businesses. This shows that they aspire to be at the forefront of the industry, constantly pushing the boundaries of what is possible to help their clients succeed.

Their mission statement is focused on creating meaningful software solutions that enable their clients to achieve their business objectives and deliver value to their customers. They emphasize the importance of being a trusted partner to their clients, providing exceptional service and expertise in the latest technologies. This highlights their commitment to delivering quality solutions and building long-term relationships with their clients.

EPAM Systems also values diversity and inclusion, as evident in their mission statement. They are committed to building a diverse and inclusive culture where their employees can thrive and grow professionally. This shows that they recognize the importance of fostering an environment that values differences and encourages collaboration and innovation.

Overall, EPAM Systems' vision and mission statement reflect their dedication to providing innovative solutions, delivering exceptional service, and fostering a diverse and inclusive culture.

In addition to their focus on providing innovative solutions, EPAM Systems values diversity and inclusion. They are committed to building a diverse and inclusive culture where their employees can thrive and grow professionally. They recognize the importance of fostering an environment that values differences and encourages collaboration and innovation.

EPAM Systems also places a strong emphasis on continuous improvement. They are dedicated to staying up-to-date with the latest technologies and industry trends, and they strive to incorporate this knowledge into their work to deliver the best possible solutions to their clients.

In summary, EPAM Systems' vision and mission statement reflect their dedication to providing innovative solutions, delivering exceptional service, fostering a diverse and inclusive culture, and continuously improving their work. They are committed to helping their clients succeed and building long-term partnerships based on trust and collaboration.

EPAM Systems Mission Statement:

To become the transformation platform for adaptive enterprises.

EPAM Systems Vision Statement:

No one has added EPAM Systems' vision statement yet.

EPAM Systems Values:

- VALUE THE INDIVIDUAL
- FOCUS ON THE CUSTOMER
- ACT AS A TEAM
- STRIVE FOR EXCELLENCE
- ACT WITH INTEGRITY

Origin and growth of company

EPAM Systems is a global technology and software engineering company that provides a wide range of IT solutions and services to clients worldwide. The company's success is built on a foundation of strong core values that guide its operations and shape its culture. Here is the EPAM Systems Company Core Values statement:

- 1. Strive for excellence:** We are committed to delivering exceptional quality and value to our clients through continuous improvement, innovation, and an unwavering focus on excellence.
- 2. Respect and care:** We treat our clients, employees, and partners with respect, fairness, and empathy, and we strive to create a work environment that is safe, diverse, and inclusive.
- 3. Collaborate and communicate:** We believe that teamwork and communication are essential for success, and we encourage open, honest, and constructive dialogue to foster trust, respect, and collaboration among our team members.

4. Take ownership: We take ownership of our work and hold ourselves accountable for delivering results that meet or exceed our clients' expectations.

5. Be proactive: We anticipate and respond to our clients' needs quickly and efficiently, and we take a proactive approach to solving problems and delivering solutions that drive business success.

6. Embrace change: We embrace change as an opportunity for growth and innovation, and we are constantly adapting to new technologies, markets, and business models to stay ahead of the curve.

7. Act with integrity: We act with honesty, integrity, and transparency in all our business dealings, and we strive to build strong, lasting relationships based on trust and mutual respect.

EPAM Systems' Company Core Values are not just words on a page; they are the guiding principles that shape our culture, drive our performance, and inspire us to be our best selves every day.

Origin and growth of company

EPAM Systems is a global provider of digital platform engineering and software development services. The company was founded in 1993 by Arkadiy Dobkin in Princeton, New Jersey. Dobkin, who remains the CEO to this day, started the company with just a few employees and focused on providing software development services to businesses in the United States.

In the early years, EPAM focused on developing custom software solutions for clients in industries such as healthcare, finance, and retail. However, as the company grew, it expanded its offerings to include digital platform engineering, product development, design thinking, and other services to help clients navigate the digital landscape.

In 2002, EPAM established its first development center in Minsk, Belarus, which allowed the company to tap into a highly skilled talent pool and offer cost-effective solutions to clients. This move was a game-changer for EPAM, as it allowed the company to scale its services quickly and efficiently.

Over the years, EPAM continued to expand its global presence through both organic growth and strategic acquisitions. In 2012, the company went public on the New York Stock Exchange (NYSE) under the ticker symbol EPAM. Today, EPAM has a team of over 50,000 professionals working across the globe and serves clients in more than 35 countries.

EPAM's success can be attributed to its focus on innovation, customer-centric approach, and commitment to delivering exceptional quality and value to its clients. The company has received numerous industry recognitions and awards, including being named a Leader in the Gartner Magic Quadrant for Application Services for six consecutive years.

In addition to its core software development and digital engineering services, EPAM has also invested heavily in emerging technologies such as artificial intelligence (AI), machine learning (ML), blockchain, and the Internet of Things (IoT). This has allowed the company to stay at the forefront of technological innovation and provide cutting-edge solutions to clients across a wide range of industries.

EPAM Systems is a leading software engineering services provider, specializing in product development, digital platform engineering, and digital and product design. The company was founded in 1993 in Princeton, New Jersey, by Arkadiy Dobkin, and has since grown to become a global organization with over 43,000 employees across 35 countries.

Arkadiy Dobkin, a software engineer, founded EPAM Systems in 1993 in Princeton, New Jersey. Dobkin had immigrated to the United States from Minsk, Belarus, in the late 1970s, and he saw an opportunity to provide high-quality software engineering services to American companies.

Initially, EPAM focused on providing software development services to companies in the financial services industry. The company quickly gained a reputation for its expertise in the field, and in 1998, it opened a development center in Minsk, Belarus.

Growth of the Company

EPAM's growth has been fueled by its focus on providing high-quality software engineering services to its clients. The company has expanded its service offerings to include digital platform engineering, product development, and digital and product design.

EPAM has also expanded its geographic footprint, opening development centers in countries around the world, including Hungary, Russia, Ukraine, and India. Today, EPAM has over 43,000 employees across 35 countries.

In addition to its organic growth, EPAM has also made several strategic acquisitions to expand its capabilities. In 2018, the company acquired Continuum, a global innovation design firm, and TH_NK, a digital agency based in the UK.

EPAM Systems has grown from a small software development company to a global organization with over 43,000 employees across 35 countries. The company's focus on providing high-quality software engineering services to its clients, as well as its strategic acquisitions, have been key drivers of its growth. With its continued focus on innovation and expansion, EPAM is poised for continued success in the future.

EPAM Systems is a multinational corporation that provides digital platform engineering and software development services to clients worldwide. The company was founded in 1993 in Minsk, Belarus, and has since grown into a leading provider of digital solutions, with operations in over 30 countries.

EPAM's growth has been driven by a number of factors, including its focus on delivering high-quality, innovative solutions to its clients. The company has built a reputation for excellence in the industry, and has consistently been recognized for its work in digital transformation and software development.

In addition to its focus on delivering value to clients, EPAM has also expanded its global footprint through strategic acquisitions and partnerships. In recent years, the company has made several key acquisitions, including Dextrys, a software development firm based in China, and Ricston, a UK-based integration services provider.

EPAM has also been successful in attracting and retaining top talent in the industry. The company has a strong culture of innovation and collaboration, and has invested heavily in its workforce, providing employees with training and development opportunities to ensure they stay at the forefront of the industry.

EPAM's financial performance has been strong, with revenues increasing from \$869 million in 2016 to \$2.66 billion in 2020. The company's net income has also steadily increased, from \$98 million in 2016 to \$390 million in 2020.

Overall, EPAM Systems' growth can be attributed to its commitment to delivering value to clients, investing in technology and talent, and expanding its global footprint through strategic acquisitions and partnerships. The company's focus on innovation and collaboration has allowed it to stay ahead of the curve in an ever-evolving industry, and its strong financial performance is a testament to its success.

Various departments and their functions

EPAM Systems is a global digital platform engineering and software development company that provides a wide range of services, including consulting, product design, engineering, and testing. The company has a diverse range of departments, each with its own functions, which play a critical role in ensuring that EPAM delivers high-quality services to its clients. In this report, we will provide an overview of EPAM's various departments and their functions.

1. Delivery Department:

The delivery department is responsible for ensuring that EPAM's services are delivered on time, within budget, and to the highest quality standards. This department is responsible for managing project teams and coordinating with clients to ensure that their needs are met. They also work to identify risks and develop plans to mitigate them. The delivery department is composed of project managers, technical leads, business analysts, and quality assurance analysts.

2. Technology Department:

The technology department is responsible for researching and developing new technologies, tools, and methodologies that can be used to improve EPAM's services. This department is composed of software engineers, architects, and researchers who work on cutting-edge technologies such as artificial intelligence, machine learning, and blockchain. They also work closely with the delivery department to ensure that the new technologies and tools are properly integrated into EPAM's service offerings.

3. Consulting Department:

The consulting department is responsible for providing strategic advice and guidance to EPAM's clients. This

department is composed of consultants who work with clients to identify their business needs and develop strategies to address them. They also work to identify opportunities for growth and improvement in clients' operations.

4. Sales Department:

The sales department is responsible for identifying and securing new business for EPAM. This department is composed of sales representatives who work to build relationships with potential clients, identify their needs, and propose solutions that meet their requirements. They also work to negotiate contracts and ensure that clients are satisfied with EPAM's services.

5. Human Resources Department:

The human resources department is responsible for recruiting, training, and managing EPAM's workforce. This department is composed of recruiters, trainers, and human resources managers who work to identify talented individuals, provide them with the training and support they need to succeed, and ensure that they are properly compensated and rewarded for their work.

6. Marketing Department:

The marketing department is responsible for promoting EPAM's services and building brand awareness. This department is composed of marketers who work to develop marketing strategies, create promotional materials, and build relationships with industry influencers and media outlets.

7. Finance Department:

The finance department is responsible for managing EPAM's financial resources. This department is composed of financial analysts, accountants, and financial managers who work to ensure that EPAM's financial operations are efficient, effective, and compliant with relevant regulations.

CHAPTER2

INTRODUCTION TO PROJECT

Throughout the period of the internship, we learned a lot of skills. Every skill that we learned or gained have different scopes and each skill plays a vital role in our lives. The scope of some skills that we learned are mentioned below:

- Teamwork
- Problem solving
- Co-ordination
- Interpersonal Skills
- Time Management

Training Undertaken:

AUTOMATION TESTING

Automation testing is a software testing technique that involves the use of automated tools and scripts to test a software application. The primary goal of automation testing is to ensure that the software application meets the expected requirements and works as expected in different scenarios.

In automation testing, testers use specialized software tools to write scripts that perform the same steps repeatedly, without the need for human intervention. These scripts are used to simulate user interactions and to verify that the software behaves correctly under different conditions.

Automation testing is usually faster and more accurate than manual testing, as it allows for the execution of test cases at a higher speed and with greater precision. It is particularly useful for testing large and complex software applications that require extensive testing across multiple platforms and configurations.

Objectives of the work undertaken:

The objectives of automation testing can vary depending on the specific project and the goals of the organization, but some common objectives of automation testing are:

- 1. Increase efficiency:** Automation testing can help to speed up the testing process, reduce the time and effort required for manual testing, and increase the overall efficiency of the testing process.
- 2. Improve accuracy:** Automation testing can help to eliminate human error and increase the accuracy of testing results, which can improve the quality of the software being tested.
- 3. Increase test coverage:** Automation testing can help to increase the test coverage by allowing for the testing

of a larger number of scenarios, which can help to identify more defects and issues in the software.

4. Reusability of test scripts: Automation testing allows for the creation of reusable test scripts that can be used repeatedly, saving time and effort in future testing cycles.

5. Continuous testing: Automation testing can be integrated with continuous integration and continuous delivery (CI/CD) pipelines, allowing for continuous testing and rapid feedback on software changes.

Overall, the objective of automation testing is to improve the efficiency, accuracy, and quality of the testing process, while also reducing the overall cost of testing.

Scope of the Work:

The scope of automation testing work can vary depending on the specific project and the needs of the organization. However, some common areas of scope for automation testing work are:

1. Test planning and design: Automation testing work can include the planning and design of test cases, test suites, and test scenarios that will be automated.

2. Test automation tool selection: The automation testing work can involve selecting the appropriate automation tool(s) for the specific project, considering factors such as cost, features, ease of use, and compatibility with the application under test.

3. Test script development: The automation testing work can involve developing and scripting automated test cases using the selected automation tool(s).

4. Test script maintenance: The automation testing work can involve maintaining the automated test scripts, including updating them as needed to accommodate changes in the application under test.

5. Test execution and reporting: The automation testing work can involve executing the automated test scripts, monitoring the results, and generating test reports that provide insight into the quality of the application under test.

6. Integration with other testing activities: The automation testing work can be integrated with other testing activities, such as manual testing, performance testing, and security testing, to provide a comprehensive testing approach.

Role and Profile:

I have been working in this project as a Automation Tester Intern

Roles and Responsibilities of a Automation Tester

An automation tester is responsible for creating and executing automated test scripts to verify the functionality, performance, and reliability of software applications. The following are some of the key roles and responsibilities of an automation tester:

1. Test Planning: The automation tester should be involved in the planning and design phase of the software

development life cycle (SDLC). This includes defining the scope of testing, identifying the requirements, and creating test plans.

2. Test Automation: The automation tester is responsible for developing automated test scripts that can be used to test various aspects of the software application. This includes developing test cases, identifying test data, and developing test scripts.

3. Test Execution: The automation tester should execute automated test scripts on different environments such as different operating systems, browsers, and mobile devices. They should also ensure that the test results are accurate, reliable and that any defects are properly documented and communicated.

4. Defect Management: The automation tester should manage the defects identified during testing. This includes identifying the root cause of the defect, reporting it to the development team, and ensuring that it is fixed in a timely manner.

5. Test Reporting: The automation tester should generate reports on test results and provide feedback to the development team. This includes communicating the test results to stakeholders, highlighting any defects or issues that need to be addressed, and making recommendations for improvements.

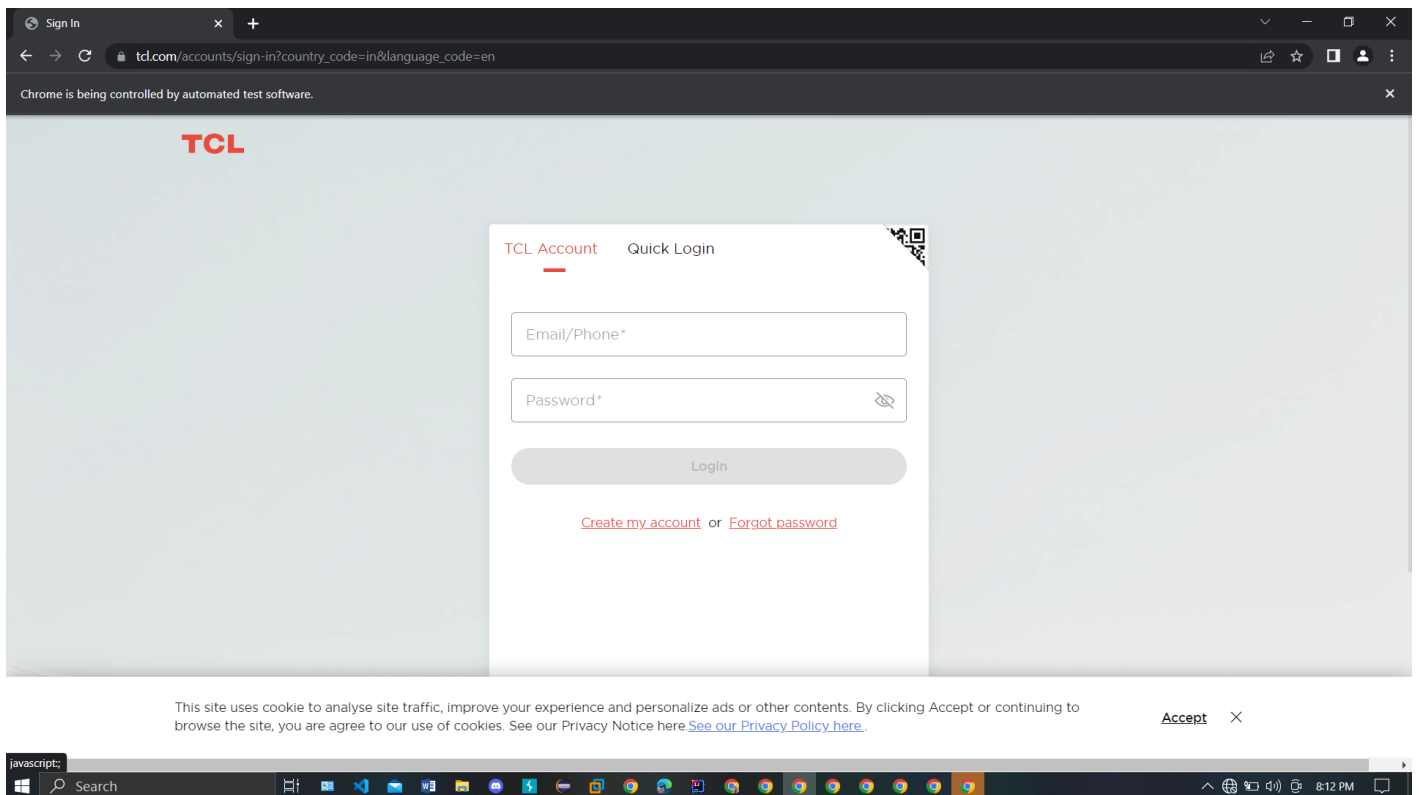
6. Continuous Integration/Continuous Deployment: The automation tester should ensure that the automated tests are integrated with the continuous integration and continuous deployment (CI/CD) process. This includes setting up and maintaining the automated test suite, monitoring the test results, and ensuring that the tests are executed as part of the CI/CD pipeline.

7. Test Automation Framework: The automation tester should have experience in developing and maintaining a test automation framework that can be used to automate various aspects of the testing process.

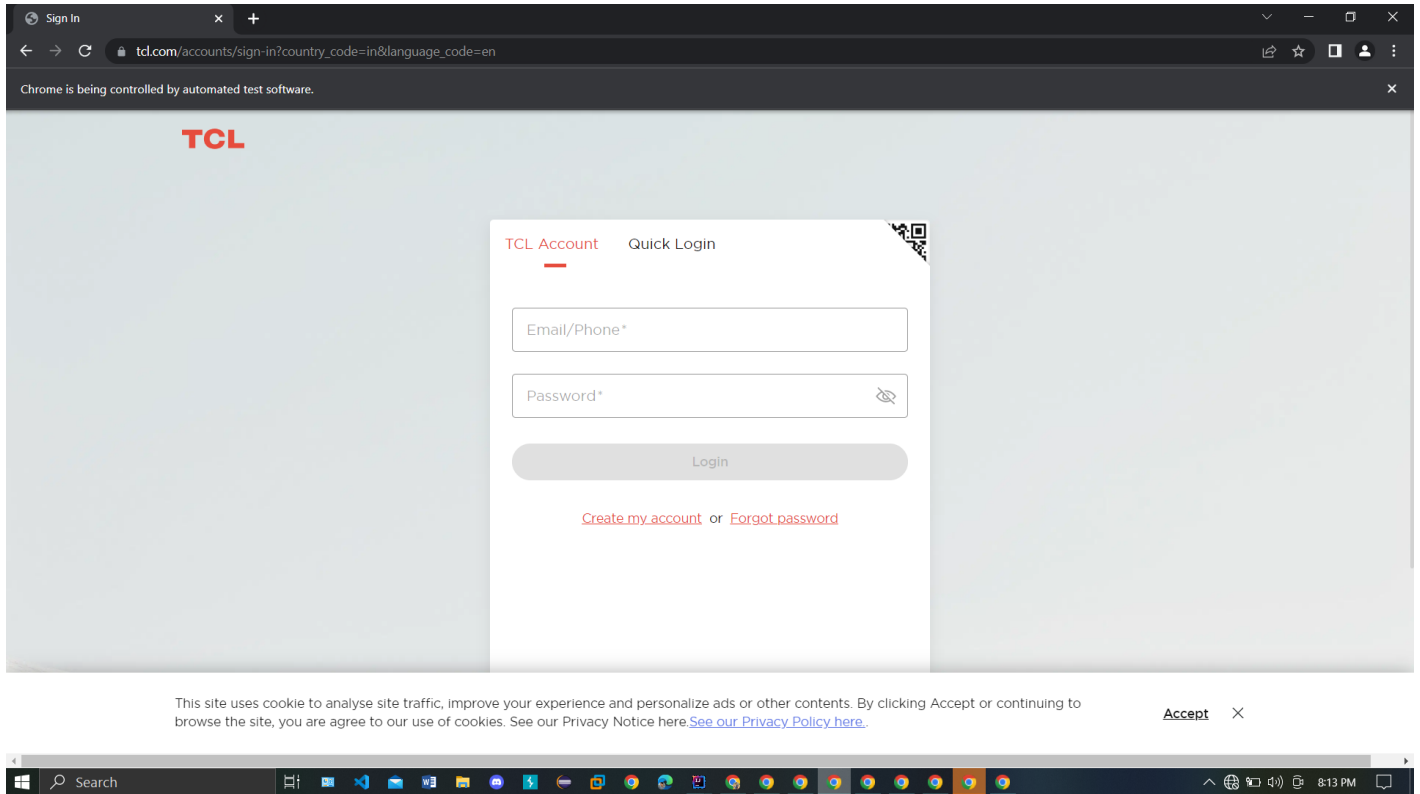
Overall, the automation tester plays a crucial role in ensuring that software applications are delivered with high quality and reliability. They are responsible for automating the testing process and ensuring that defects are identified and fixed in a timely manner.

GROUP PROJECT:-

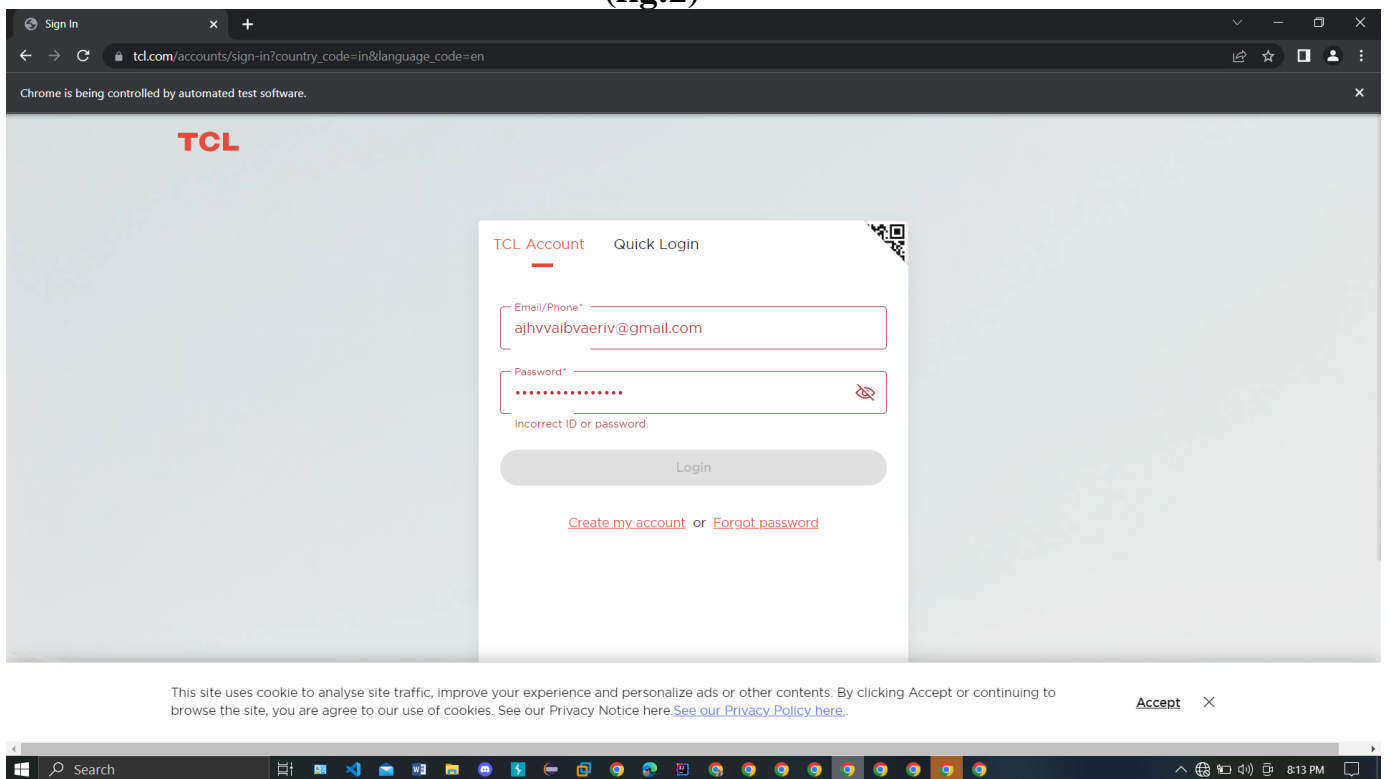
In this project we have tested(<https://www.tcl.com/in/store>) TCL website, product is to add to cart successfully.



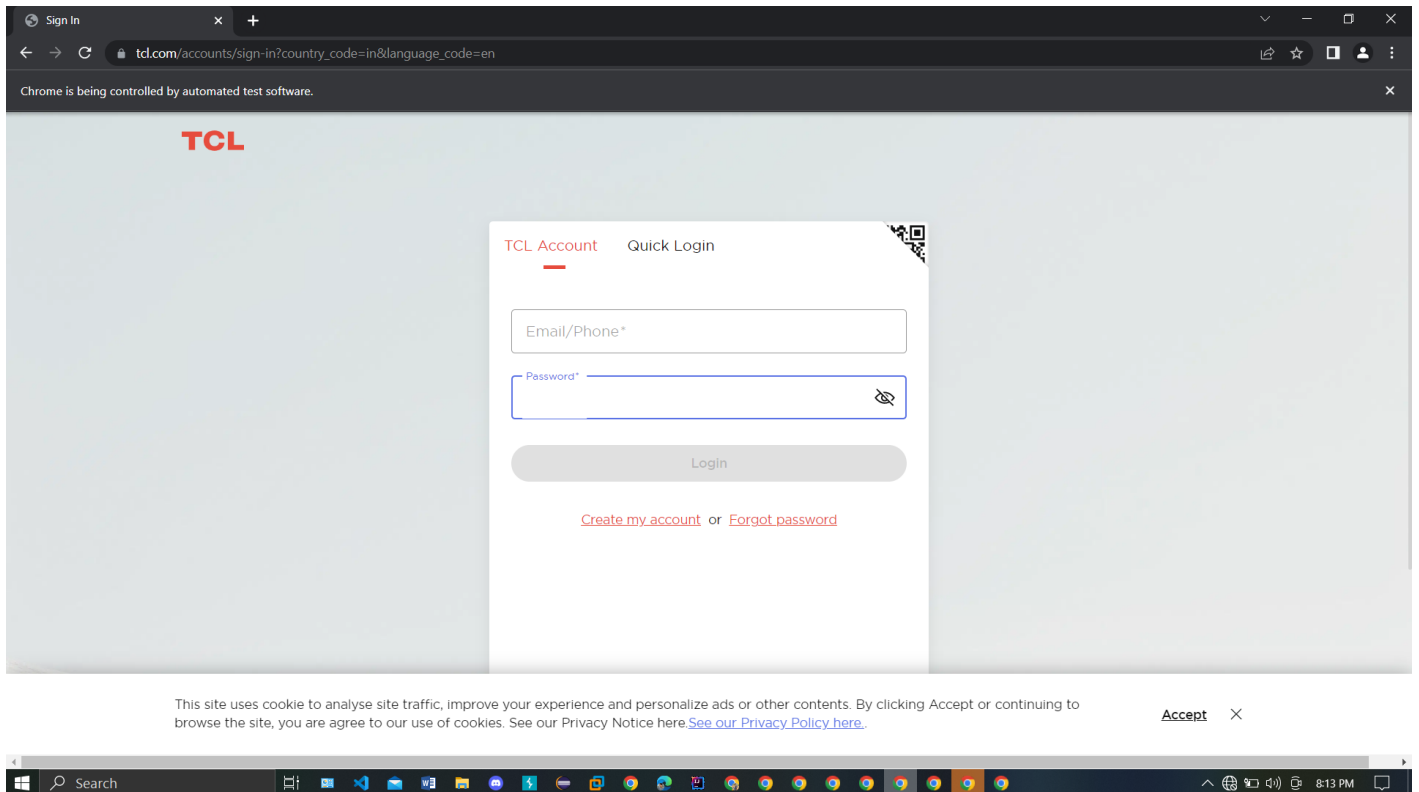
(Fig:1)



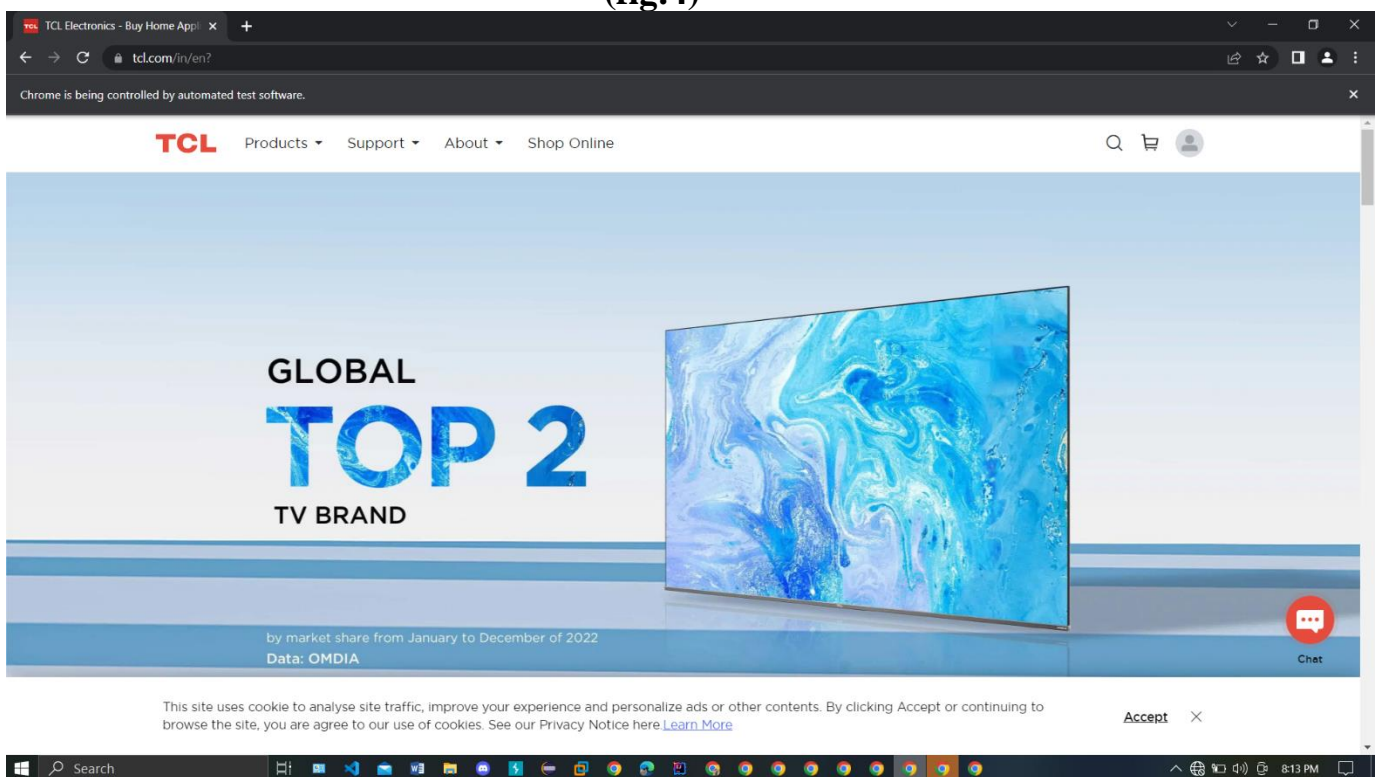
(fig:2)



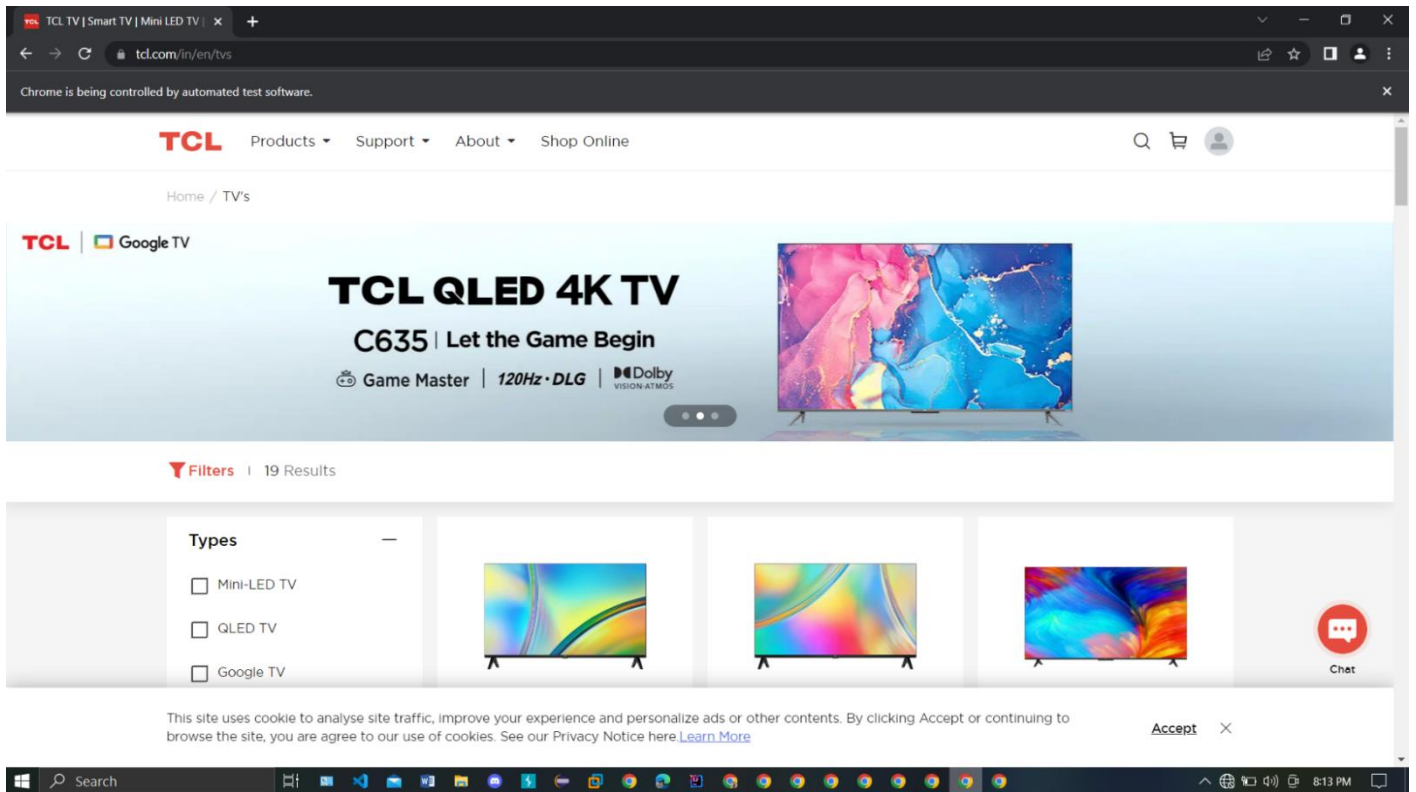
(fig:3)



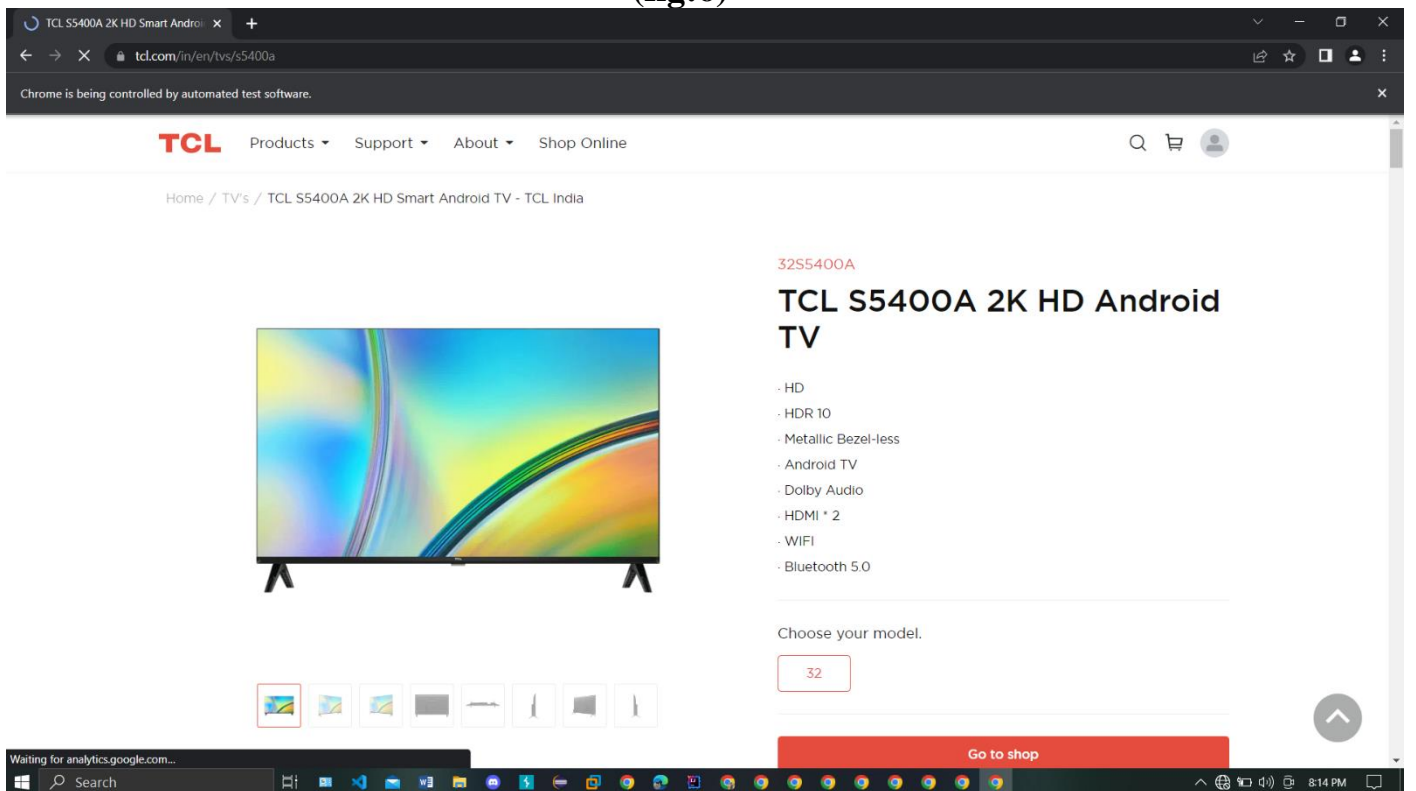
(fig:4)



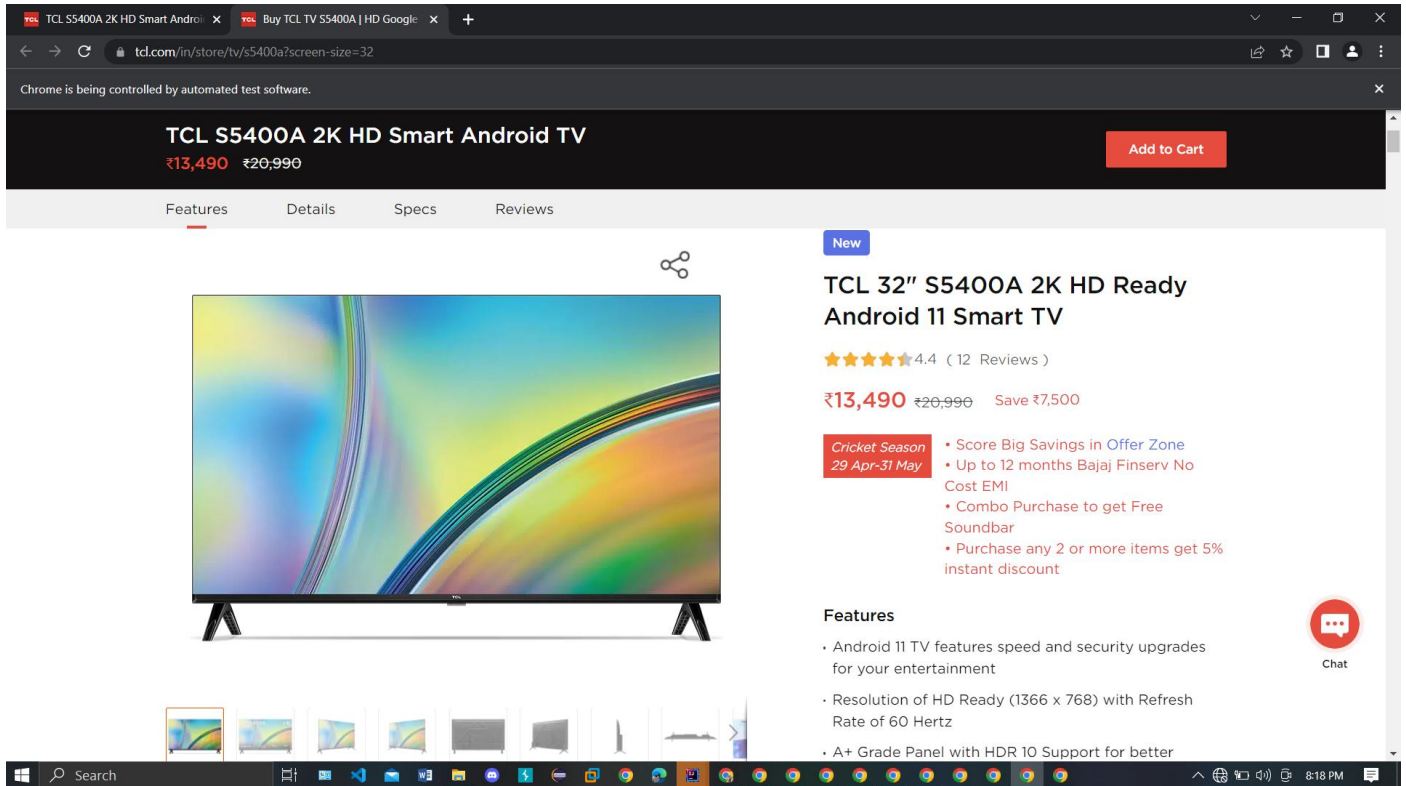
(fig:5)



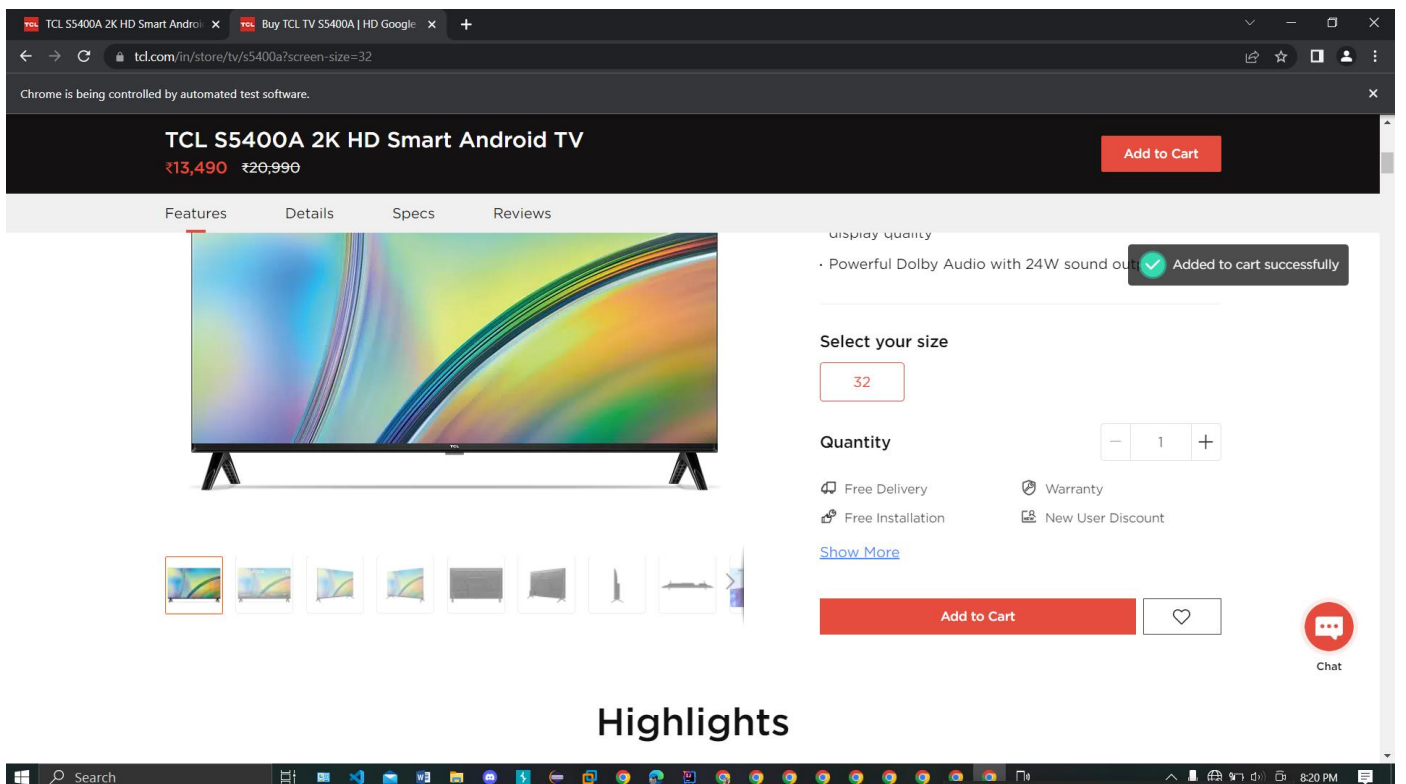
(fig:6)



(fig:7)



(fig:8)



Highlights

(fig:9)

fig1 & fig2 we can see The login page of TCL website.

fig3 we can see when we enter wrong details it displays Incorrect ID or Password.

fig4 we can see the wrong details is cleared.

fig5 we can see that we entered correct details and it will takes to the home page.

in fig5 it clicks the products and it displays in fig6,

in fig6 it clicks on the 1st product and it displays as in fig7.

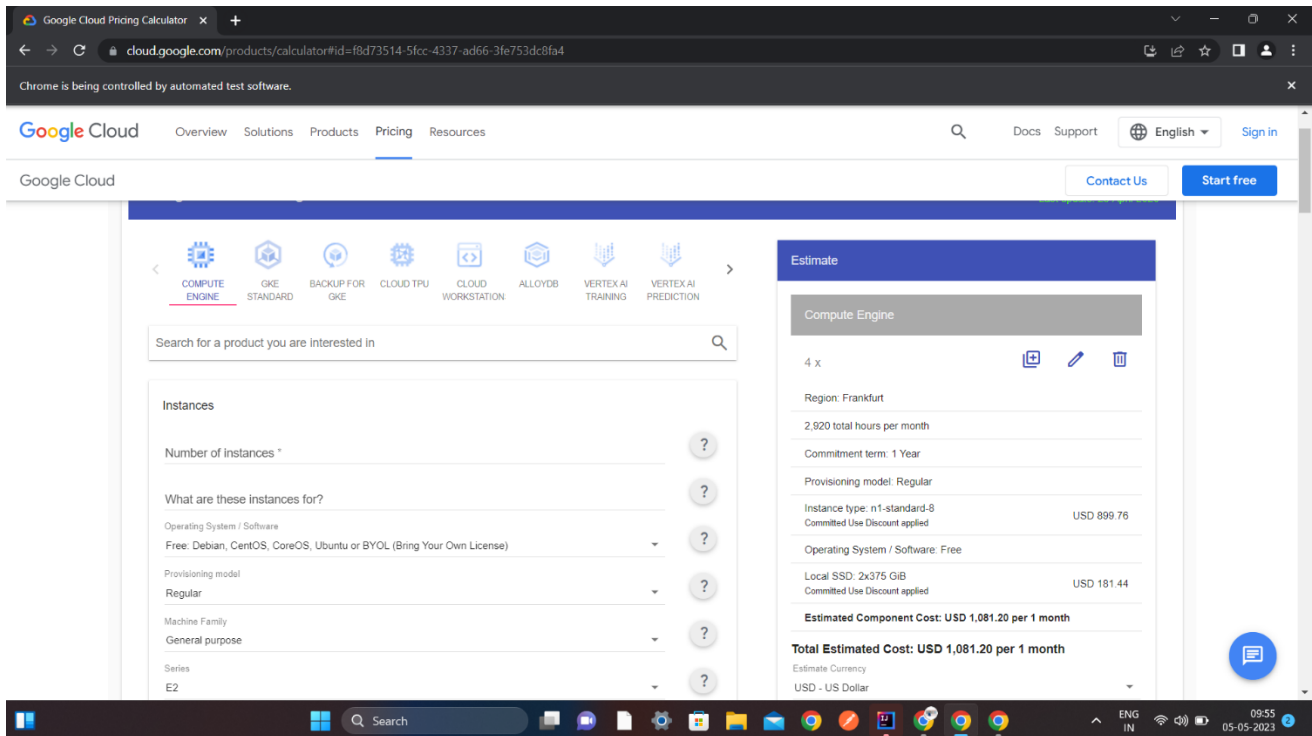
in fig7 we can see go to shop, it clicks and it goes to fig8.

in fig8 we can see the add to cart button it clicks on that.

In fig9 we can see the message that added to cart successfully.

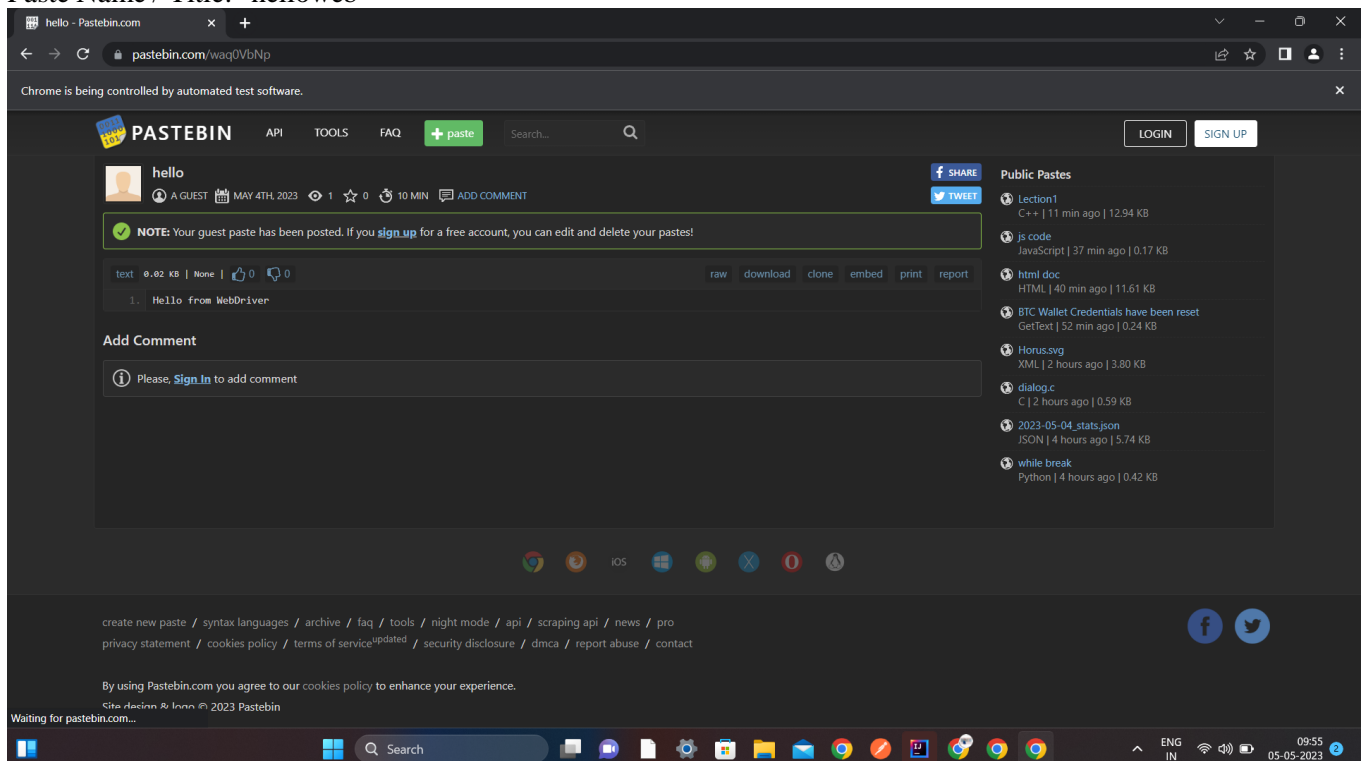
INDIVIDUAL PROJECT

1. Open <https://cloud.google.com/>.
2. Click the search button at the top of the portal page and enter "Google Cloud Platform Pricing
3. Calculator" into the search field.
4. Click on the search button to start searching.
5. Click "Google Cloud Platform Pricing Calculator" in the search results and go to the calculator page.
6. Click COMPUTE ENGINE at the top of the page.
6. Fill out the form with the following data:
 - Number of instances: 4
 - What are these instances for?: leave blank
 - Operating System / Software: Free: Debian, CentOS, CoreOS, Ubuntu, or another User-Provided OS
 - VM Class: Regular
 - Instance type: n1-standard-8 (vCPUs: 8, RAM: 30 GB)
 - Select "Add GPUs"
 - Number of GPUs: 1
 - GPU type: NVIDIA Tesla V100
 - Local SSD: 2x375 Gb
 - Datacenter location: Frankfurt (europe-west3)
 - Committed usage: 1 Year
7. Click Add to Estimate.
8. Check that the data is correct in the following fields: VM Class, Instance type, Region, local SSD, commitment term.
9. Check that the monthly rent matches the amount received when the test is done manually.

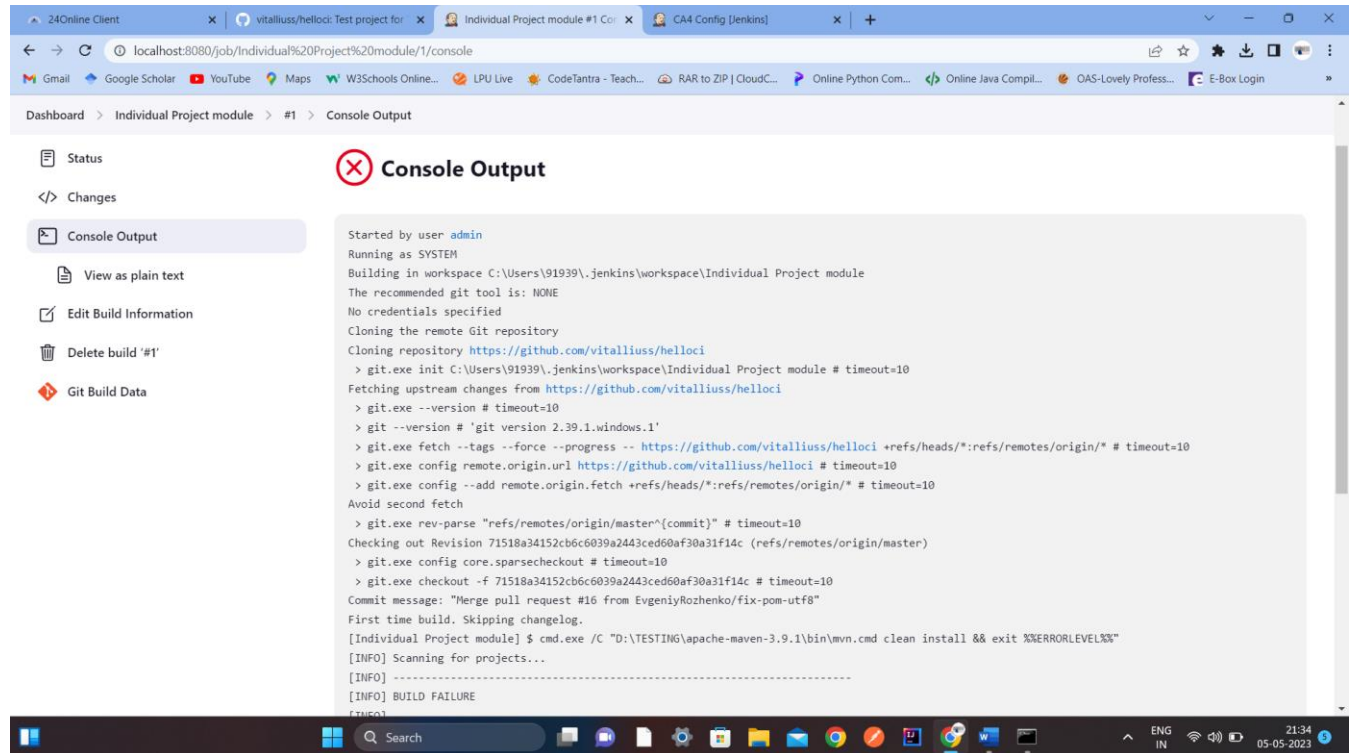


(Fig:10)

1. Open <https://pastebin.com/> or a similar service in any browser.
2. Create 'New Paste' with the following attributes:
 - Code: "Hello from WebDriver"
 - Paste Expiration: "10 Minutes"
 - Paste Name / Title: "helloworld"



1. Install Jenkins
2. Create a task which will perform the following:
 - Clone the project (<https://github.com/vitalliuss/hellocli>)
 - Launch tests from the project in Java directory with the help of mvn test goal
3. Set up build triggers so that the task is performed every 5 minutes



The screenshot shows the Jenkins web interface in a browser. The address bar indicates the URL is `localhost:8080/job/Individual%20Project%20module/1/console`. The page title is "Console Output". On the left sidebar, there are links for "Status", "Changes", "Console Output" (selected), "View as plain text", "Edit Build Information", "Delete build '#1'", and "Git Build Data". The main content area displays the console output for build #1. The output shows the build was started by user 'admin' and is running as 'SYSTEM'. It details the cloning of the repository from `https://github.com/vitalliuss/hellocli` and the execution of `git.exe` commands to initialize, fetch, and checkout the repository. The commit message is "Merge pull request #16 from EvgeniyRozhenko/fix-pom-utf8". The build then runs the command `cmd.exe /C "D:\TESTING\apache-maven-3.9.1\bin\mvn.cmd clean install && exit %ERRORLEVEL%"`. The output shows "Scanning for projects..." and then "BUILD FAILURE".

```
Started by user admin
Running as SYSTEM
Building in workspace C:\Users\91939\jenkins\workspace\Individual Project module
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/vitalliuss/hellocli
> git.exe init C:\Users\91939\jenkins\workspace\Individual Project module # timeout=10
Fetching upstream changes from https://github.com/vitalliuss/hellocli
> git.exe --version # timeout=10
> git --version # 'git version 2.39.1.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/vitalliuss/hellocli +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe config remote.origin.url https://github.com/vitalliuss/hellocli # timeout=10
> git.exe config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision 71518a34152cb6c6039a2443ced60af30a31f14c (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f 71518a34152cb6c6039a2443ced60af30a31f14c # timeout=10
Commit message: "Merge pull request #16 from EvgeniyRozhenko/fix-pom-utf8"
First time build. Skipping changelog.
[Individual Project module] $ cmd.exe /C "D:\TESTING\apache-maven-3.9.1\bin\mvn.cmd clean install && exit %ERRORLEVEL%"
[INFO] Scanning for projects...
[INFO] -----
[INFO] BUILD FAILURE
```


Back to Epam-Final-Project [Index](#) [Zip](#)

1 suite [Switch Retro Theme](#) **Test results**

All suites

All Test Suite

Info

- C:\Users\Tuf\jenkins\workspace\Epam-Final-Project\testng.xml
- 3 tests
- 0 groups
- Times
- Reporter output
- Ignored methods
- Chronological view

Results

- 5 methods, 5 passed
- Passed methods (hide)
 - checkInformationInVmClassString
 - checkInformationInVmClassString
 - check_Data_is_correct
 - check_Data_is_correct
 - loginVerification

org.example.Google_Price_Cal_Test

- checkInformationInVmClassString
- checkInformationInVmClassString
- check_Data_is_correct
- check_Data_is_correct

org.example.Pagebin_Project_Test

- loginVerification

TestNG Results Trend [Jenkins]

localhost:8080/job/Individual%20Project/testngreports/

Jenkins Search (CTRL+K) admin log out

Dashboard > Individual Project > TestNG Results

TestNG Results Trends

Latest Test Results (build #3)

- Total Tests: 5 (±0)
- Failed Configurations: 0 (±0)
- Failed Tests: 0 (±0)
- Skipped Tests: 0 (±0)
- Skipped Configurations: 0 (±0)

Build History trend

Filter builds...

#3
04-May-2023, 3:52 pm

#1

CHAPTER3

TECHNOLOGIES LEARNT

Selenium:-

As an automation tester using Selenium, there are several technologies that you can use to generate reports on your test results. Some of these technologies include:

Selenium is a popular open-source tool for automating web browsers, which is commonly used for testing web applications. It provides a framework for writing automated tests in a variety of programming languages, such as Java, Python, Ruby, and C#, among others.

Software Development Methodologies:

Selenium is a popular open-source software testing framework used to automate web browsers. When it comes to software development methodologies, there are a few different approaches that can be used with Selenium.

- 1. Agile methodology:** Agile methodology is a popular approach in software development that emphasizes collaboration and flexibility. With Selenium, an Agile approach can be used to ensure that testing is done continuously throughout the development process, allowing for quick feedback and adjustments.
- 2. Waterfall methodology:** Waterfall is a more traditional approach to software development that follows a linear process. With Selenium, the Waterfall approach can be used to ensure that testing is done at specific stages of development, such as after the requirements gathering or design phase.
- 3. Test-driven development (TDD):** TDD is an approach that emphasizes writing automated tests before writing any code. With Selenium, TDD can be used to ensure that the code being developed meets the requirements and specifications of the software.
- 4. Behavior-driven development (BDD):** BDD is an approach that focuses on defining the behavior of the software in terms of specific scenarios or use cases. With Selenium, BDD can be used to ensure that the software meets the requirements of the business and end-users.
- 5. Continuous Integration/Continuous Delivery (CI/CD):** CI/CD is an approach that emphasizes frequent and automated testing and delivery of software. With Selenium, CI/CD can be used to ensure that the testing is automated and integrated into the development process, allowing for quick feedback and adjustments.

Selenium can be used with a variety of software development methodologies, including Agile, Waterfall, TDD, BDD, and CI/CD. The choice of methodology will depend on the specific needs and requirements of the project.

SQL:-

Introduction to SQL:

Structured Query Language (SQL) is a standard programming language used to manage relational databases. SQL allows users to store, manipulate, and retrieve data from a database. SQL is used by businesses and organizations to manage large amounts of data, and it is also used by individuals for personal projects. SQL can be used to create databases, tables, and views, as well as to insert, update, and delete data.

Database Basics:

A database is a collection of data that is organized in a specific way so that it can be easily managed, accessed, and updated. Databases are used to store data for various purposes, such as for business applications, e-commerce websites, and social media platforms. A database is typically made up of tables, which are organized into rows and columns. Each table contains data about a specific type of object or entity, and the columns represent the attributes or characteristics of that entity.

SQL Foundation:

SQL has three main parts: Data Definition Language (DDL), Data Manipulation Language (DML), and Data Control Language (DCL). DDL is used to create and modify the structure of a database, such as creating tables and indexes. DML is used to insert, update, and delete data from a database. DCL is used to control access to the database, such as granting or revoking user privileges.

SQL for Analysis:

SQL can be used for data analysis, which involves querying a database to extract specific information or insights. SQL can be used to join tables, filter data, and perform calculations. SQL is often used in business intelligence and data warehousing applications, where large amounts of data are stored and analyzed to make informed business decisions. SQL is also used in data science and machine learning applications, where data is analyzed to develop predictive models and make data-driven decisions.

SQL is based on the relational model of data, which organizes data into tables or relations. A table consists of rows and columns, with each row representing a unique record and each column representing a specific piece of information. SQL allows users to manipulate data in these tables using a variety of commands.

SQL commands are divided into several categories, including data definition language (DDL), data manipulation language (DML), data control language (DCL), and transaction control language (TCL). DDL commands are used to create and modify database objects such as tables, indexes, and views. DML commands are used to manipulate data in these objects, such as inserting, updating, and deleting records. DCL commands are used to control access to the database by creating users and setting permissions. TCL commands are used to manage transactions, such as committing or rolling back changes.

1. Data Definition Language (DDL): SQL allows you to define the structure of your database using DDL statements. These statements include CREATE, ALTER, and DROP. CREATE is used to create a new database or table, ALTER is used to modify the structure of an existing database or table, and DROP is used to delete a database or table.

2. Data Manipulation Language (DML): SQL also allows you to manipulate data in your database using DML statements. These statements include SELECT, INSERT, UPDATE, and DELETE. SELECT is used to retrieve data from a database, INSERT is used to add data to a database, UPDATE is used to modify existing data in a database, and DELETE is used to delete data from a database.

3. Joins: SQL supports several types of joins, including INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL OUTER JOIN. Joins allow you to combine data from two or more tables into a single result set.

4. Aggregation: SQL supports several aggregation functions, including SUM, AVG, MAX, MIN, and COUNT. These functions allow you to perform calculations on data in your database.

5. Constraints: SQL allows you to specify constraints on your database tables. These constraints include PRIMARY KEY, FOREIGN KEY, UNIQUE, CHECK, and NOT NULL. Constraints ensure that your data is consistent and accurate.

6. Indexes: SQL supports indexes, which are used to improve the performance of queries on your database tables. Indexes are created on one or more columns of a table, and they allow the database to quickly find the data you are looking for.

7. Transactions: SQL supports transactions, which allow you to group multiple SQL statements together into a single unit of work. Transactions ensure that your database remains consistent even if an error occurs during a group of SQL statements.

8. Stored Procedures: SQL allows you to create stored procedures, which are precompiled SQL statements that can be executed on demand. Stored procedures are used to improve performance and simplify database maintenance.

9. Views: SQL allows you to create views, which are virtual tables that are based on the result of a SELECT statement. Views are used to simplify complex queries and to control access to sensitive data.

10. Triggers: SQL supports triggers, which are special stored procedures that are automatically executed when certain events occur in your database. Triggers are used to enforce business rules and to ensure data consistency.

Practical Applications of SQL

SQL is widely used in data-related fields such as data analysis, data warehousing, and business intelligence. Some practical applications of SQL include:

1. Data Analysis: SQL can be used to analyze large datasets and extract insights from them. It allows users to group, filter, and aggregate data, making it easier to analyze trends and patterns.

2. Data Warehousing: SQL can be used to manage large data warehouses, which are databases designed for storing and analyzing large amounts of data. SQL allows users to manipulate data in these warehouses and create reports and dashboards to visualize the data.

3. Business Intelligence: SQL can be used to support business intelligence (BI) applications, which are tools that help organizations analyze and visualize data. SQL allows users to manipulate data and create reports and dashboards that can be used to make data-driven decisions.

Commonly Used SQL Commands

Some of the most commonly used SQL commands include:

1. SELECT: The SELECT command is used to retrieve data from one or more tables. It allows users to specify which columns they want to retrieve and apply filters to the data using the WHERE clause.

2. INSERT: The INSERT command is used to add new records to a table. It allows users to specify the values for each column in the new record.

3. UPDATE: The UPDATE command is used to modify existing records in a table. It allows users to specify which records to update and which columns to modify.

4. DELETE: The DELETE command is used to delete records from a table. It allows users to specify which records to delete based on a specified condition.

5. JOIN: The JOIN command is used to combine data from two or more tables into a single result set. It allows users to specify how the tables are related using a join condition.

Cloud Computing:

Cloud computing is a model for delivering computing resources such as servers, storage, databases, networking, software, analytics, and more, over the Internet ("the cloud"). Instead of businesses or individuals having to purchase and maintain their own physical infrastructure and hardware, they can leverage cloud computing services provided by third-party providers such as Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform, and many others.

Cloud computing provides a flexible and scalable way to consume and pay for computing resources. Users can rapidly provision and de-provision resources on-demand, and pay only for what they use, without having to make large upfront investments. Cloud computing also offers the advantage of allowing users to access their applications and data from anywhere with an internet connection.

The three main types of cloud computing services are Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). IaaS provides users with access to raw computing resources such as virtual machines, storage, and networking. PaaS provides a platform for developers to build and deploy applications. SaaS provides users with access to software applications that are hosted in the cloud and accessed through a web browser or app.

There are three main types of cloud computing: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).

1. Infrastructure as a Service (IaaS):

IaaS is a type of cloud computing service that provides users with access to virtualized computing resources such as servers, storage, networking, and other infrastructure components. Users can deploy and manage their own software and applications on these resources, giving them full control and flexibility over their computing environment.

Examples of IaaS providers include Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform, and DigitalOcean.

2. Platform as a Service (PaaS):

PaaS is a type of cloud computing service that provides users with a platform for developing, deploying, and managing applications. PaaS providers offer a preconfigured environment with a set of tools and services that developers can use to build and run their applications.

Examples of PaaS providers include Heroku, Google App Engine, Microsoft Azure App Service, and AWS Elastic Beanstalk.

3. Software as a Service (SaaS):

SaaS is a type of cloud computing service that provides users with access to software applications that are hosted in the cloud and accessed through a web browser or app. With SaaS, users do not have to manage the underlying infrastructure, as the software provider handles all maintenance and upgrades.

Examples of SaaS providers include Salesforce, Microsoft Office 365, Google Workspace, and Dropbox.

IAM Introduction: Users, Groups, Policies

IAM stands for Identity and Access Management, which is a service provided by cloud computing platforms such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform. IAM enables users to securely manage access to their resources by creating and managing users, groups, and policies.

Users are individual entities that are identified by a unique name and a set of credentials, such as a username and password. Each user can have specific permissions and access to resources within an organization. In IAM, users can be created, modified, and deleted as needed, and their access can be controlled using policies.

Groups are collections of users that share similar permissions and access levels to resources. By creating groups, an organization can manage access to resources more efficiently and ensure that users have access to only the resources they need. Groups can be created based on department, project, or any other criteria that is relevant to an organization's structure.

Policies are sets of rules that determine the level of access a user or group has to a resource. Policies can be attached to users, groups, or resources to define permissions for actions such as read, write, and delete. IAM policies are written in JSON format and define the resources that are allowed or denied access, as well as the actions that can be performed on those resources.

With IAM, users can be granted access to specific resources based on their role, responsibilities, and needs. This allows organizations to control access to their resources and ensure that sensitive data is protected from unauthorized access.

IAM Policies

IAM policies are used to define permissions for accessing resources within an organization. Policies are written in JSON format and can be attached to users, groups, or resources. They define what actions can be performed on which resources, and by whom.

IAM policies consist of the following elements:

1. Resource: The AWS resource to which the policy applies, such as an Amazon S3 bucket or an EC2 instance.
2. Effect: The effect of the policy, which can be either "allow" or "deny". This determines whether the specified action is allowed or denied.
3. Action: The AWS service action that the policy is granting or denying permission for, such as "s3:GetObject" or "ec2:StartInstances".
4. Principal: The AWS account, user, or role that the policy is granting or denying permission to.
5. Condition: Optional conditions that must be met for the policy to take effect, such as requiring the request to come from a specific IP address or be made over SSL.

IAM policies can be used to grant users and groups access to specific resources or actions within an AWS account. For example, an IAM policy can be created to allow a group of users to access an S3 bucket to read and write objects, while denying access to delete objects. Policies can also be used to grant temporary access to resources for specific tasks, and then revoked once the task is completed.

IAM policies can be created, modified, and deleted through the AWS Management Console, AWS Command Line Interface (CLI), or AWS SDKs. Organizations should regularly review and audit their IAM policies to ensure that users have only the access they need to perform their job duties, and that there are no unnecessary permissions or potential security risks.

IAM Security Tools

IAM (Identity and Access Management) provides several security tools that can help organizations to better manage and secure their resources in the cloud. Some of these tools are:

- 1. Multi-factor authentication (MFA):** MFA is a security feature that adds an extra layer of protection to user accounts. With MFA, users are required to provide a second factor of authentication, such as a code from a mobile app, in addition to their password, to access their account. IAM supports MFA for users and root accounts.
- 2. IAM Roles:** IAM roles are a way to delegate access to AWS resources to entities that are not AWS accounts or users, such as applications or services. By using IAM roles, organizations can grant temporary access to resources and reduce the risk of exposing long-term credentials.
- 3. IAM Policies:** IAM policies are used to define permissions for accessing resources within an organization. They can be used to grant or deny access to specific resources or actions within an AWS account. IAM policies can also be used to enforce security best practices, such as requiring the use of MFA or SSL encryption.
- 4. Access Analyzer:** Access Analyzer is a tool that helps organizations to identify resources that are publicly accessible and could be potential security risks. Access Analyzer provides recommendations for remediation and can help organizations to ensure that their resources are properly secured.
- 5. IAM Access Advisor:** IAM Access Advisor is a tool that provides visibility into the permissions granted to IAM roles and users. It helps organizations to identify and remove unnecessary permissions, reducing the risk of insider threats and unauthorized access.
- 6. AWS Organizations:** AWS Organizations is a tool that helps organizations to manage multiple AWS accounts. It allows organizations to centralize management of policies and permissions across multiple accounts, providing better visibility and control over resources.

these IAM security tools can help organizations to improve their security posture and reduce the risk of data breaches and other security incidents in the cloud.

Shared Responsibility Model for IAM

The shared responsibility model for IAM (Identity and Access Management) defines the division of responsibilities between the cloud service provider (CSP) and the customer regarding the management and security of IAM resources. This model helps to ensure that both parties are accountable for their respective roles in securing cloud resources.

Under the shared responsibility model for IAM, the CSP is responsible for the security of the cloud infrastructure, including the physical servers, network, and data centers. The CSP provides IAM services, including the IAM service itself, as well as tools and features that customers can use to secure their IAM resources.

On the other hand, the customer is responsible for managing access to their resources and implementing security controls to protect their data. This includes creating and managing IAM users, groups, and policies, as well as implementing security best practices such as Multi-Factor Authentication (MFA), monitoring and auditing IAM resources, and enforcing password policies.

In general, the CSP is responsible for providing a secure platform and IAM service, while the customer is responsible for securing their own resources and data within that platform. The specific division of responsibilities may vary depending on the cloud service provider and the services being used.

It is important for customers to understand their responsibilities under the shared responsibility model for IAM and to implement best practices to secure their IAM resources. This includes regularly reviewing and auditing IAM policies and permissions, monitoring for suspicious activity and potential security risks, and staying up-to-date on the latest security features and updates provided by the CSP.

IAM Summary

In summary, IAM (Identity and Access Management) is a cloud computing service that provides a centralized way to manage access to resources within an organization. With IAM, organizations can create and manage users and groups, assign specific permissions and access levels to resources, and enforce security policies.

IAM enables organizations to control access to their resources by creating and managing users, groups, and policies. Users are individual entities that are identified by a unique name and a set of credentials, such as a username and password. Groups are collections of users that share similar permissions and access levels to resources. Policies are sets of rules that determine the level of access a user or group has to a resource.

IAM provides several security tools to help organizations better manage and secure their resources in the cloud, such as Multi-Factor Authentication (MFA), IAM Roles, IAM Policies, Access Analyzer, IAM Access Advisor, and AWS Organizations.

IAM is an essential service for organizations that want to control access to their resources and ensure that sensitive data is protected from unauthorized access in the cloud.

EC2 - Elastic Compute Cloud

Amazon Elastic Compute Cloud (EC2) is a web service that provides scalable computing resources in the cloud. EC2 enables customers to quickly launch virtual machines, also known as instances, and run a variety of workloads, such as web applications, batch processing, and data analytics. EC2 instances can be launched in multiple regions around the world and are available in a variety of instance types, which are optimized for different workloads and performance requirements.

Some of the key features of EC2 include:

- 1. Scalability:** EC2 enables customers to quickly scale up or down their computing resources based on changing demand. Instances can be launched and terminated in minutes, and EC2 Auto Scaling enables customers to automatically adjust the number of instances based on predefined scaling policies.
- 2. Customizability:** EC2 instances can be customized with a wide range of options, including different operating systems, application software, storage options, and network configurations. Customers can also choose from a variety of instance types with different CPU, memory, and networking capabilities.

3. Security: EC2 provides a variety of security features, such as Virtual Private Cloud (VPC), Security Groups, and Network Access Control Lists (ACLs), which enable customers to isolate their instances and control access to their resources.

4. Reliability: EC2 is designed to provide high availability and reliability. Instances are launched in multiple Availability Zones, which are physically separate data centers within a region, providing greater resilience to failures and disasters.

5. Pay-per-use pricing: EC2 pricing is based on a pay-per-use model, where customers only pay for the resources they consume. This enables customers to optimize their costs and avoid upfront capital expenditures.

EC2 is a flexible and scalable computing service that enables customers to run a wide range of workloads in the cloud. It provides a variety of features and options to meet the needs of different use cases and workloads, and is widely used by businesses of all sizes around the world.

EC2 Instance Types Basics

Amazon EC2 (Elastic Compute Cloud) provides a variety of instance types that are optimized for different workloads and performance requirements. Each instance type has a unique combination of CPU, memory, storage, and network capacity, providing customers with a wide range of options to choose from.

Here are some of the key factors to consider when selecting an EC2 instance type:

1. Compute: Each instance type provides a different level of CPU performance, which is measured in vCPUs (virtual CPUs). Some instance types are optimized for compute-intensive workloads, while others are optimized for memory-intensive workloads.

2. Memory: The amount of memory available on an instance is important for memory-intensive workloads, such as data analytics and machine learning. Some instance types have a high memory-to-CPU ratio, providing more memory for a given amount of vCPUs.

3. Storage: EC2 instances can be configured with different types of storage, including local storage and network-attached storage (EBS). Some instance types have high-performance storage optimized for I/O-intensive workloads, while others provide larger amounts of storage for data-intensive workloads.

4. Network: Each instance type has a different level of network performance, which is important for network-intensive workloads, such as web applications and streaming. Some instance types provide high-bandwidth networking, while others have lower networking performance.

Some of the common EC2 instance types include:

1. General Purpose: These instance types provide a balance of compute, memory, and networking resources and are suitable for a wide range of workloads, including web servers, small databases, and development environments. Examples include the t3 and m5 instance families.

2. Compute Optimized: These instance types provide high levels of compute performance and are suitable for compute-intensive workloads, such as batch processing, gaming, and scientific modeling. Examples include the c5 and c6g instance families.

3. Memory Optimized: These instance types provide high levels of memory capacity and are suitable for memory-intensive workloads, such as data analytics, in-memory databases, and machine learning. Examples include the r5 and x1 instance families.

4. Storage Optimized: These instance types provide high levels of storage capacity and are suitable for storage-intensive workloads, such as data warehousing, log processing, and media storage. Examples include the i3 and d3 instance families.

selecting the right EC2 instance type is important for optimizing performance and cost for different workloads. Customers should carefully evaluate their requirements and choose an instance type that provides the right balance of compute, memory, storage, and network capacity.

Automated Testing Basics:

Object-Oriented (OO) Design Patterns are reusable solutions to common problems that arise when designing and developing software applications. They provide a standard way of solving problems, making code more flexible, modular, and easier to maintain.

There are three main categories of design patterns:

- 1. Structural Patterns:** concerned with the composition of classes or objects to form larger structures.
- 2. Behavioral Patterns:** concerned with the interaction between objects and how they operate together to form larger systems.
- 3. Creational Patterns:** concerned with object creation mechanisms, trying to create objects in a manner suitable to the situation.

Here are some examples of commonly used patterns:

- 1. Factory Pattern:** The Factory Pattern is a creational pattern that provides an interface for creating objects in a super class, but allows subclasses to alter the type of objects that will be created. It provides an interface to create objects, but lets subclasses decide which class to instantiate. This pattern is useful when we need to create objects without exposing the instantiation logic to the client.
- 2. Strategy Pattern:** The Strategy Pattern is a behavioral pattern that allows us to define a family of algorithms, encapsulate each one, and make them interchangeable. It provides a way to select an algorithm at runtime, based on the client's needs. This pattern is useful when we have a group of algorithms that can be used interchangeably.

3. Builder Pattern: The Builder Pattern is a creational pattern that allows us to create complex objects step by step, rather than all at once. It separates the construction of a complex object from its representation, so that the same construction process can create different representations. This pattern is useful when we need to create objects with many different options or configurations.

4. Singleton Pattern: The Singleton Pattern is a creational pattern that restricts the instantiation of a class to a single instance and provides a global point of access to that instance. This pattern is useful when we need to ensure that only one instance of a class is created and when we need to provide a global point of access to that instance.

Test Automation and xUnit Test Framework:

Test automation is the practice of using software tools to automatically execute tests on a software application or system. This is done to save time and reduce errors that can occur when tests are performed manually. One of the most popular test automation frameworks is xUnit.

xUnit is a family of unit testing frameworks that originated with JUnit, a testing framework for Java. The xUnit framework has since been implemented in many programming languages, including C#, Python, and Ruby.

The xUnit framework is based on the idea of writing automated tests that can be executed repeatedly to verify that the code being tested behaves as expected. The framework provides a structure for organizing tests, including test fixtures (which set up the environment for a group of tests) and individual test cases. Tests are typically written using an assertion library, which allows developers to specify the expected behavior of the code being tested.

xUnit also provides features for test discovery, test execution, and reporting. Test discovery refers to the process of automatically finding and loading all the tests that are part of a project. Test execution involves running all the tests and reporting the results. xUnit provides various mechanisms for executing tests, including command-line tools, GUI test runners, and build integration tools.

In summary, test automation using the xUnit framework is a powerful way to increase the speed and accuracy of software testing. By automating tests, developers can catch bugs early in the development cycle and ensure that their code behaves as expected across a wide range of scenarios.

Test Automation (UI/API/Performance/Security/Mobile):

Test automation refers to the use of software tools and frameworks to automate the execution of tests. This can include different types of tests such as UI testing, API testing, performance testing, security testing, and mobile testing.

UI testing involves testing the user interface of an application to ensure that it is functioning correctly and meets the requirements. This can include testing the layout, navigation, and functionality of buttons and other UI elements.

API testing involves testing the application programming interface (API) of an application to ensure that it is working as expected and can communicate with other software components. This can include testing the functionality of APIs and checking for error handling and response codes.

Performance testing involves testing the performance of an application to ensure that it can handle a large number of users and requests without slowing down or crashing. This can include load testing, stress testing, and capacity testing.

Security testing involves testing the security of an application to ensure that it is protected against various types of attacks, including SQL injection, cross-site scripting (XSS), and other vulnerabilities.

Mobile testing involves testing mobile applications to ensure that they are functioning correctly on different devices and operating systems.

Build tools (Maven):

Build tools like Maven are used to automate the build process of an application, including compiling the code, packaging it into a distributable format, and running automated tests.

TestNG:

TestNG is a testing framework for Java that allows developers to write automated tests in a structured and organized manner. It provides features such as test configuration, test suites, and test reports. TestNG is widely used in the industry for automated testing.

API Automation:

API automation is the process of automating the testing of application programming interfaces (APIs) that are used in client-server architectures. In a client-server architecture, the client (usually a web browser or mobile application) communicates with the server (usually a web server or application server) using APIs.

APIs are a set of protocols, routines, and tools for building software applications. They define the way that different software components should interact with each other. In client-server architectures, APIs are used to enable communication between the client and the server.

API automation involves automating the testing of these APIs to ensure that they are working correctly and providing the expected results. This can be done using a variety of tools and techniques, including automated testing frameworks, scripting languages, and test automation software.

The goal of API automation is to improve the quality of the software application by identifying and fixing issues early in the development process. It can also help to reduce the time and cost associated with manual testing by automating repetitive tasks.

Client-Server Architecture:

Client-server architecture is a distributed computing model in which the client (a user interface) sends requests to the server (a software program) over a network, and the server responds with the requested data or service. This architecture allows for better scalability and reliability as multiple clients can connect to the same server and share its resources.

HTTP:

HTTP (Hypertext Transfer Protocol) is a protocol used for transmitting data over the internet. It is the foundation of data communication in the World Wide Web. HTTP requests are sent by clients to servers, and HTTP responses are sent by servers to clients. HTTP is a stateless protocol, meaning that each request/response transaction is independent of any previous requests/responses.

HTTP Request Methods:

HTTP request methods are used by clients to request data or services from servers. The most common HTTP request methods are:

- GET: retrieves data from the server
- POST: sends data to the server to create or update a resource
- DELETE: deletes a resource on the server
- PATCH: updates a resource on the server

GET: The GET method is used to retrieve data from the server. The request includes a URL and an optional set of query parameters. The server responds with the requested data in the response body.

POST: The POST method is used to send data to the server to create or update a resource. The request includes a URL and a request body that contains the data to be sent to the server. The server responds with a confirmation message in the response body.

DELETE: The DELETE method is used to delete a resource on the server. The request includes a URL that identifies the resource to be deleted. The server responds with a confirmation message in the response body.

PATCH: The PATCH method is used to update a resource on the server. The request includes a URL that identifies the resource to be updated, and a request body that contains the changes to be made. The server responds with a confirmation message in the response body.

JSON: JSON (JavaScript Object Notation) is a lightweight data format used for exchanging data between web applications and servers. It is easy to parse and has become a popular alternative to XML. JSON data is represented as key-value pairs, and arrays of key-value pairs.

XML: XML (Extensible Markup Language) is a data format used for exchanging data between web applications and servers. It is more verbose than JSON, but it has better support for complex data structures. XML data is represented using markup tags, similar to HTML.

Postman: Postman is a popular tool used for testing HTTP APIs. It allows developers to send HTTP requests to an API and view the response. With Postman, developers can create and save HTTP requests, organize them in collections, and share them with team members. Postman also provides features such as environment variables, tests, and scripts, which can be used to automate testing and streamline API development.

Rest Assured: Rest Assured is a Java-based library used for testing RESTful APIs. It provides a fluent API for writing HTTP requests and assertions in Java code. Rest Assured supports all HTTP request methods and provides features such as authentication, cookies, and proxies. Rest Assured can also be integrated with popular testing frameworks such as JUnit and TestNG.

Selenium WebDriver

Selenium WebDriver is a popular open-source testing tool that is widely used for automating web application testing. It is a browser automation tool that allows you to automate interactions with a web browser and simulate user actions such as clicking buttons, filling out forms, and navigating between pages.

Selenium WebDriver supports a variety of programming languages such as Java, Python, C#, Ruby, JavaScript, and PHP, making it a versatile tool for testing web applications in different environments.

Using Selenium WebDriver, you can write test scripts that can run on multiple browsers such as Chrome, Firefox, Safari, and Internet Explorer, and on different operating systems like Windows, Linux, and Mac OS.

Selenium WebDriver provides a simple and intuitive API that allows you to interact with the web page elements and execute different types of actions such as clicking, typing, selecting, and scrolling. You can also use it to perform advanced actions like mouse movements, keyboard shortcuts, and JavaScript execution.

Selenium WebDriver is a powerful tool for automating web application testing, and it is widely used by developers, testers, and quality assurance professionals to improve the quality and reliability of their software.

HTML (Hypertext Markup Language): HTML is a standard markup language used to create web pages. It provides the structure of the page, including text, images, and other media. HTML tags define the content and the layout of the page, and allow you to add links, tables, and forms to the page. HTML is the backbone of any website and is essential for creating dynamic and interactive web pages.

CSS (Cascading Style Sheets): CSS is a style sheet language used to describe the presentation of a document written in HTML. It is used to define the layout, colors, fonts, and other visual aspects of a web page. CSS allows developers to separate the content and the presentation of a web page, making it easier to maintain and update the page. CSS can also be used to create responsive designs that adapt to different screen sizes and devices.

XPath: XPath is a language used to query XML documents, including HTML files. It is used to navigate and select elements and attributes within an XML document, which is useful for automating web scraping and testing. XPath expressions can be used to locate elements on a web page based on their attributes, such as class,

ID, or name. XPath can also be used to traverse the document tree and select elements based on their relationships with other elements.

WEBDRIVER:

WebDriver is a crucial component in Selenium that enables the automation of web browsers for testing web applications. When generating reports in Selenium, it's important to include information about the WebDriver that was used during the test.

The WebDriver API provides a set of methods for controlling the browser, such as navigating to a page, interacting with elements on the page, and retrieving information about the page. To use WebDriver in Selenium, you typically instantiate a driver object for the browser you want to automate (such as Chrome, Firefox, or Edge), and then use the methods provided by the WebDriver API to control the browser.

When generating reports in Selenium, you can include information about the WebDriver that was used during the test, such as the version of the browser, the operating system it was run on, and any custom configurations or settings that were applied. This information can be useful for debugging issues that may be specific to certain browsers or configurations, and for identifying areas where additional testing may be necessary.

In addition, you can also include information about the WebDriver commands that were executed during the test, such as the URL of the page that was loaded, the element that was interacted with, and the result of the action (e.g. whether the element was successfully clicked or not). This information can help you identify issues or defects in your web application and prioritize areas for improvement in your test automation.

JS Executer: JS Executer is a feature in Selenium that allows JavaScript code to be executed within the context of a web page. It is useful for interacting with complex web applications and executing JavaScript-based tests. JavaScript is a powerful scripting language that can be used to manipulate the contents of a web page, add new elements, and modify existing ones. JS Executer allows you to execute JavaScript code as part of your Selenium tests, giving you greater control over the behavior of the web page and allowing you to test more complex scenarios.

Automation Framework:

An automation framework is a set of guidelines, rules, and best practices that are used to create a test automation suite. It provides a structured approach to designing and building automated tests, making it easier to maintain and update test cases over time.

There are many types of automation frameworks, including:

1. **Modular Framework:** This framework breaks down the test automation process into individual, reusable components. These components can be combined in different ways to create specific test cases.

2. Data-driven Framework: In this framework, test cases are designed to run on multiple sets of data. The framework separates the test logic from the test data, allowing for easier maintenance and scalability.

3. Keyword-driven Framework: In this framework, test cases are designed using a set of keywords that represent specific actions or events. The framework provides a library of keywords that can be used to create test cases quickly and easily.

4. Hybrid Framework: This framework combines two or more of the above frameworks to create a more comprehensive solution. For example, it may use a modular framework for component testing and a framework for integration testing.

Regardless of the type of framework used, there are some key components that are typically included in an automation framework. These include:

Test Case Management: This includes defining test cases, creating test scripts, executing tests, and reporting results.

Test Data Management: This includes managing test data and parameters, which may be stored in a separate file or database.

Test Environment Management: This includes setting up the test environment, including test hardware, software, and network configurations.

Error Handling and Reporting: This includes handling errors and exceptions, logging test results, and reporting defects.

Test Automation Tools: This includes selecting the right tools for the job, such as test management tools, test execution tools, and test automation frameworks.

When building an automation framework, it's important to follow best practices and standards to ensure the framework is scalable, maintainable, and reusable. Some of these best practices include:

1. Keep the framework modular and reusable.

2. Use a consistent coding style and naming convention.

3. Use version control for all code and scripts.

4. Use a continuous integration and continuous delivery (CI/CD) pipeline.

5. Document the framework and its components thoroughly. an automation framework is a structured approach to designing and building automated tests. It provides a set of guidelines and best practices that help ensure the test suite is scalable, maintainable, and reusable.

Production AT Framework:

Production AT Framework refers to a framework used for automation testing of software applications. There are several different frameworks that can be used for automation testing, each with their own strengths and weaknesses. Here are some of the most common frameworks:

Test-Driven Development (TDD): In this framework, tests are written before the code is written. The developer writes a failing test and then writes code to make the test pass. TDD helps ensure that the code is testable and that it meets the requirements.

Keyword-Driven Testing (KDT): KDT is a framework that separates test design and execution. Test cases are designed using keywords that represent actions, and these keywords are then used to generate test scripts. KDT is useful for creating test cases quickly and efficiently.

Data-Driven Testing (DDT): DDT is a framework that separates test data from test scripts. Test scripts are designed to work with different sets of test data, allowing for a wide range of test scenarios to be executed with minimal script modifications.

Domain-Driven Design (DDD): DDD is a framework that focuses on modeling the domain of the software application. It uses a common language to describe the problem domain and the solution domain, helping to ensure that the software application meets the needs of the users.

Behavior-Driven Development (BDD): BDD is a framework that focuses on the behavior of the software application. It uses natural language to describe the behavior of the application in terms of user stories and scenarios. BDD helps ensure that the software application meets the needs of the users and that it behaves as expected.

BDD with Cucumber: Cucumber is a tool that supports BDD. It allows test scenarios to be written in natural language and provides a way to execute those scenarios against the software application.

ATF Architecture: ATF Architecture refers to the architecture of the automation testing framework itself. It defines how the various components of the framework are organized and how they interact with each other. Good ATF architecture is important for ensuring that the framework is scalable, maintainable, and easy to use.

Continuous Integration with Jenkins:

Continuous Integration (CI) is a software development practice that involves integrating code changes from multiple developers into a shared repository several times a day. The aim of CI is to catch errors early in the development process, reduce the time required for testing, and ensure that the code is always in a deployable state.

Jenkins is an open-source automation server that can be used to set up CI pipelines. Jenkins provides a range of plugins and integrations that can be used to automate tasks such as building, testing, and deploying code changes. In addition, Jenkins can be used to generate reports on the CI process, providing developers and stakeholders with visibility into the status of code changes and the health of the overall system.

Here are the steps to set up a CI pipeline with Jenkins and reporting:

Install Jenkins: Jenkins can be installed on a server or a virtual machine. Once installed, Jenkins can be accessed via a web browser.

Create a New Job: In Jenkins, a job is a set of instructions that tells Jenkins how to build, test, and deploy code changes. To create a new job, click on the “New Item” button and select “Freestyle project”.

Configure the Job: In the job configuration, specify the source code repository, build tools, and any required build parameters. This can be done via the job configuration page in Jenkins.

Add Build Steps: Add build steps to the job to specify the tasks that should be performed during the build process. This can include compiling the code, running tests, and packaging the application.

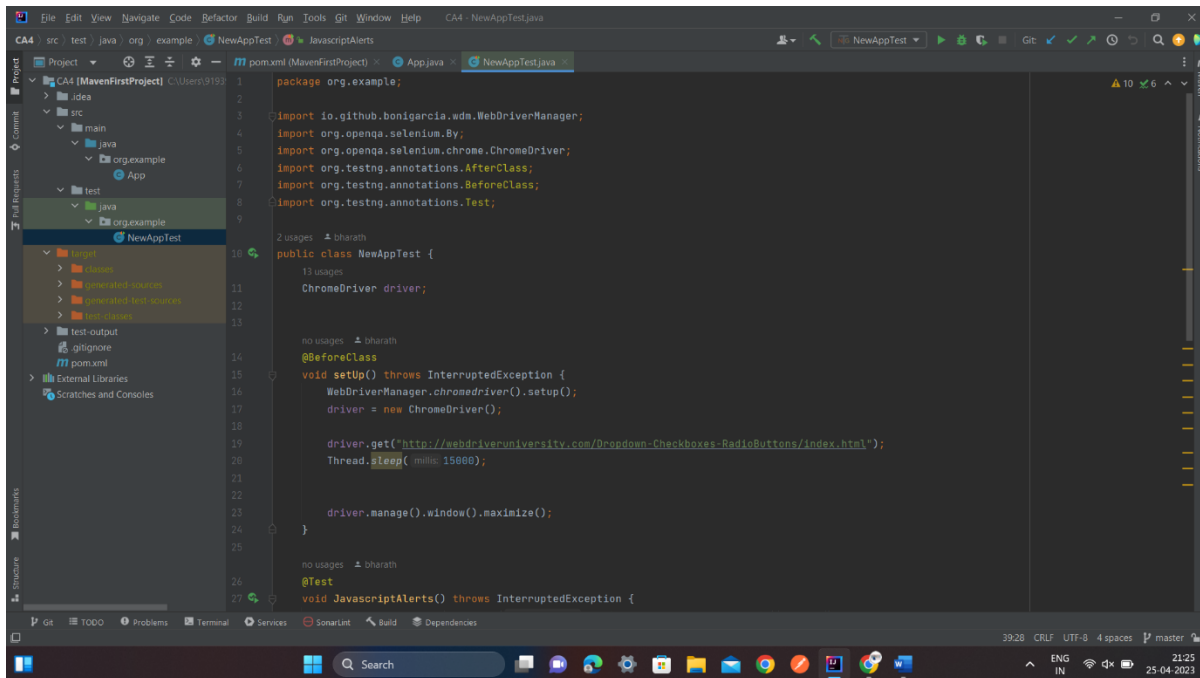
Add Reporting Plugins: Jenkins has a large library of plugins that can be used to generate reports on the CI process. Some popular reporting plugins include the Cobertura plugin for code coverage reports, the JUnit plugin for test reports, and the Checkstyle plugin for code quality reports. These plugins can be added to the job configuration to generate reports automatically.

Run the Job: Once the job configuration is complete, save the job and run it manually. Jenkins will automatically perform the tasks specified in the job configuration and generate reports on the CI process.

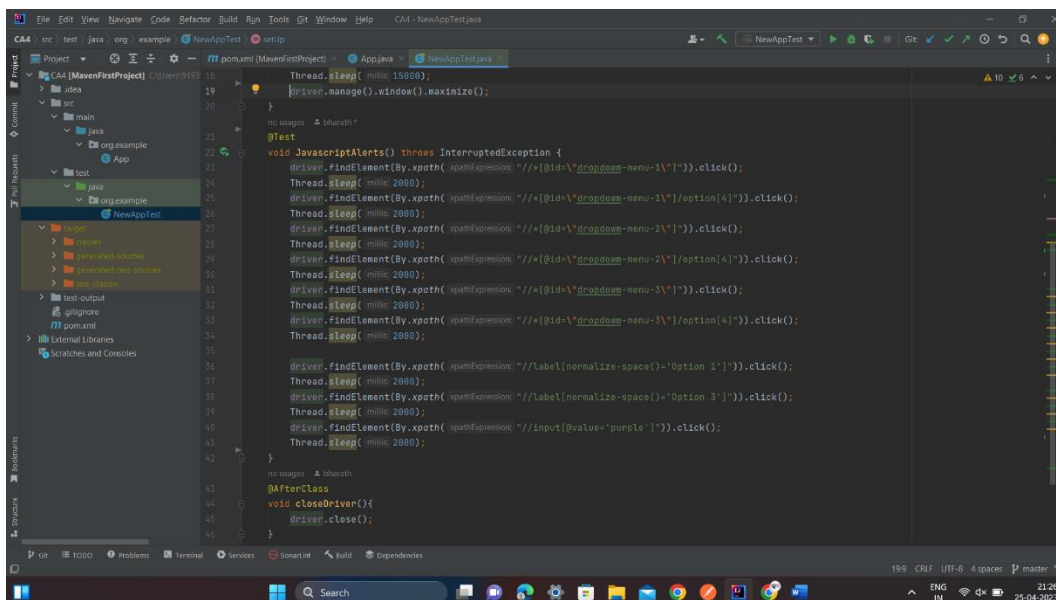
Review the Reports: Jenkins provides a range of reports on the CI process, including test results, code coverage, and code quality. Review these reports to understand the status of the code changes and identify any issues that need to be addressed.

By setting up a CI pipeline with Jenkins and reporting, developers and stakeholders can get a better understanding of the status of code changes and the health of the overall system. This can help to catch errors early in the development process, reduce the time required for testing, and ensure that the code is always in a deployable state.

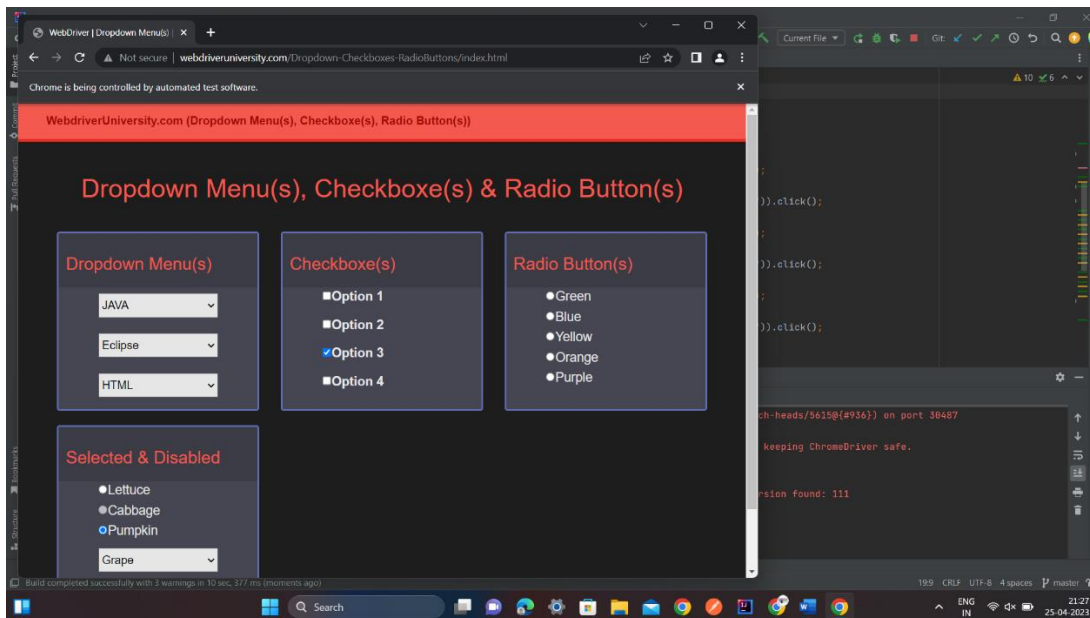
Assignment Done During Training:



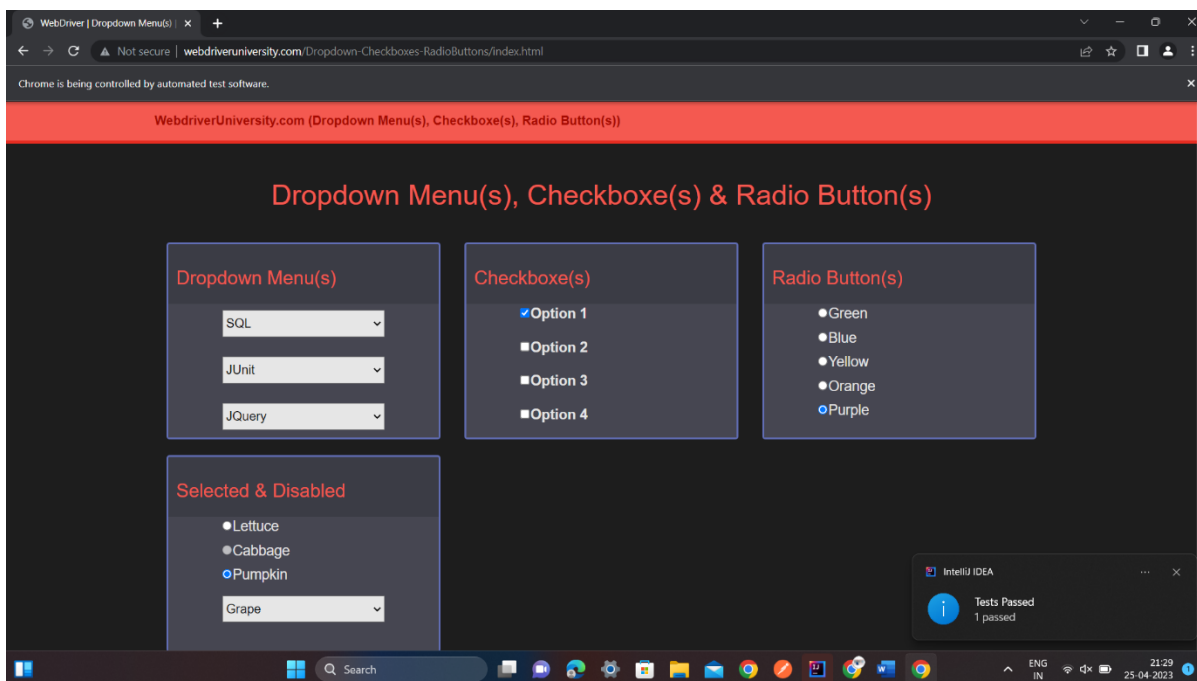
(fig:1)



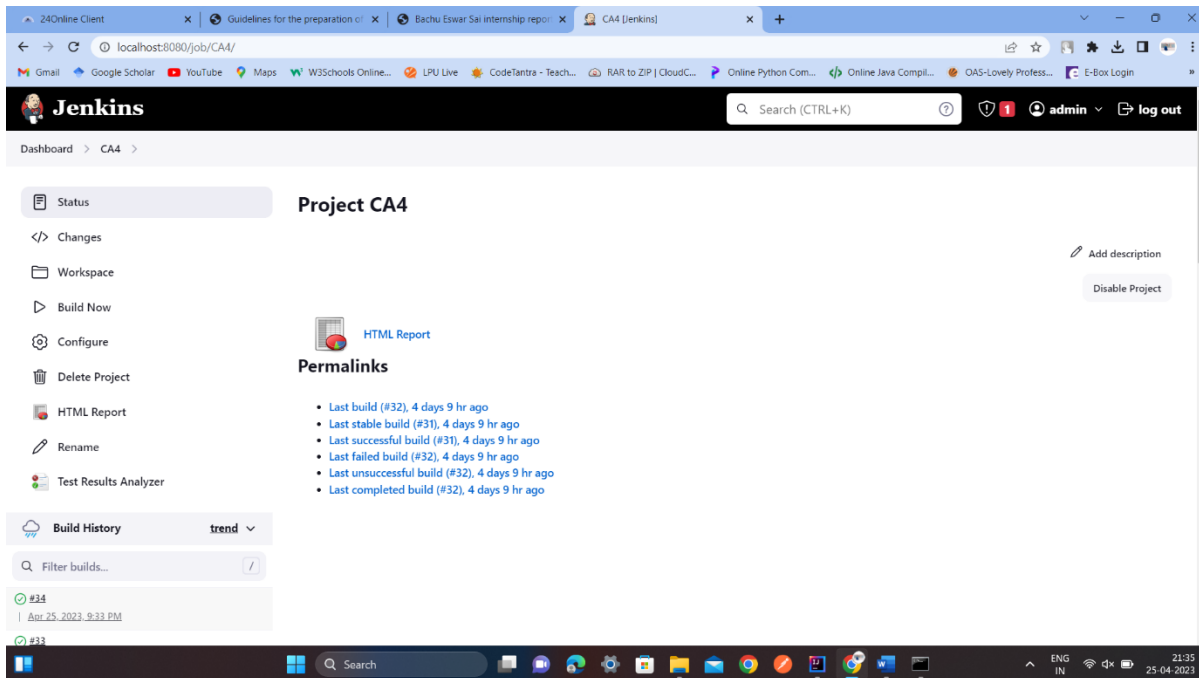
(fig:2)



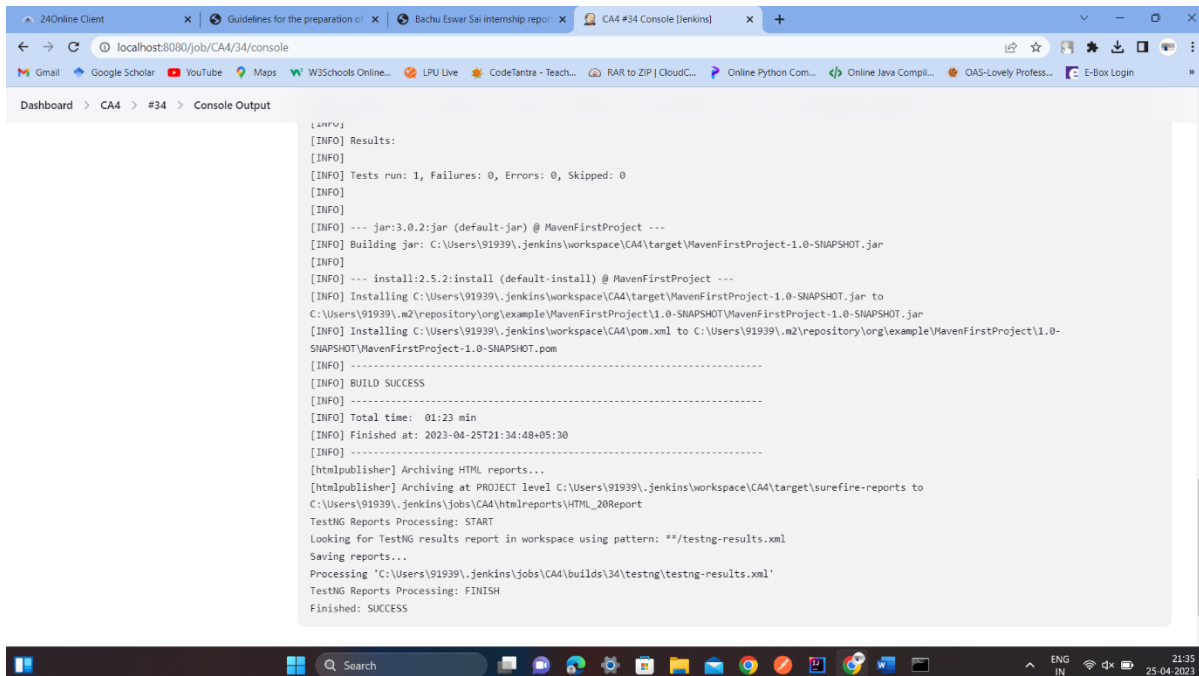
(fig:3)



(fig:4)



(fig:5)



(fig:6)

In fig 1&2 we can see the code for automation Testing, in fig3 we can see the output before Testing & in fig4 we can see the output after Automation Testing, In fig5 we can see the code is Running in Jenkins, In fig6 we can see the output of the code in Jenkins.

API testing:

What is API testing?

API testing is the process of validating that an API is working as expected. API testing can be performed manually on an ad-hoc basis, or it can be automated with a testing tool that executes test scripts at predetermined intervals or in response to events. Traditionally, API testing has occurred at the end of the development phase, right before changes are deployed to production, but an increasing number of teams are running tests earlier in the API lifecycle. This approach to API testing, which is known as “shifting left,” supports rapid iteration by enabling teams to catch and remediate issues as soon as they are introduced.

Here, we'll discuss the role that API testing plays in an API-first world—and clarify the relationship between API testing and API monitoring. We'll also review some of the most common approaches to API testing, as well as some best practices. Finally, we'll discuss how the Postman API Platform enables teams to implement an effective API testing strategy that meets their unique needs.

What is the relationship between API testing and API monitoring?

API testing and API monitoring share the goal of ensuring that APIs remain reliable and performant, but these processes are typically performed at different stages of the API lifecycle. API testing occurs during development, and its primary purpose is to help teams catch issues before they reach production and impact users. API monitoring may utilize this same testing logic, but it occurs after the API has been deployed to production. API monitoring also involves gathering and visualizing API telemetry data, which teams can use to perform historical analysis and surface long-term performance trends.

What are the benefits of API testing?

API testing plays a crucial role in modern software development workflows, and its benefits cannot be overstated. These benefits include:

- **Quality assurance:** API testing helps preserve consumer trust and protect the business's reputation by enabling teams to continuously ensure their API performs as expected.
- **Early issue detection and resolution:** A shift-left approach to API testing allows teams to identify defects as soon as they are introduced. This makes the development process more predictable and helps teams stay on schedule.
- **Resource conservation:** More and more teams are automating their approach to API testing, which saves time and allows team members to focus their bandwidth on innovation.
- **Rapid iteration:** Many teams execute their API tests within CI/CD pipelines, which enables them to automatically validate every code change before it reaches production. This approach supports more frequent releases while reducing the risk of bugs and regressions.

API testing refers to the process of testing APIs (Application Programming Interfaces) to ensure that they are functioning correctly and meeting the requirements of the application or system they are integrated with. An API

is a set of protocols, routines, and tools for building software applications, and it defines how different software components should interact with each other.

API testing involves verifying that an API can handle different types of requests and responses, and that it can perform various functions and operations as expected. The purpose of API testing is to ensure that the API meets the functional requirements, performs as expected, and returns the correct responses for each input or request.

Types of API Testing:

1. Functional testing: This type of testing focuses on the functional requirements of the API, and it verifies that the API is performing its intended function. It involves testing the API's input and output, error handling, and the response time.

2. Performance testing: This type of testing evaluates the performance of the API under various conditions, including high load, stress, and endurance testing. It helps identify the bottlenecks and performance issues that may impact the performance of the application or system.

3. Security testing: This type of testing evaluates the security of the API and verifies that it is protected against common security threats such as SQL injection, cross-site scripting, and other attacks.

4. Usability testing: This type of testing focuses on the usability of the API and verifies that it is easy to use and understand by developers and end-users.

API Testing Techniques:

1. Unit Testing: This type of testing involves testing the individual units of code that make up the API to ensure that they are working correctly.

2. Integration Testing: This type of testing involves testing the integration between different components of the API to ensure that they are working correctly together.

3. End-to-End Testing: This type of testing involves testing the entire system or application, including the API, to ensure that it is working as expected.

4. Mock Testing: This type of testing involves creating a mock or simulated version of the API to test its behavior and performance under different conditions.

API Testing Tools:

There are several tools available for API testing, including both open-source and commercial options. Some of the most popular tools include:

1. Postman: A popular tool for testing and documenting APIs, Postman provides a user-friendly interface for creating and executing API tests.

2. SoapUI: A free and open-source tool for testing SOAP and REST APIs, SoapUI allows users to create and run API tests and provides detailed reporting and analysis.

3. JMeter: An open-source load testing tool, JMeter allows users to test the performance and scalability of APIs under different load conditions.

4. Swagger: A tool for creating and documenting APIs, Swagger provides a range of features for testing, debugging, and monitoring APIs.

Why use Postman for API testing?

The Postman API Platform includes several API testing features that help teams streamline their testing workflows and catch issues throughout the API lifecycle. With Postman, you can:

- **Test any API with pre-configured code snippets:** Postman includes a JavaScript-based library of code snippets that enable teams to easily author tests that validate their API's performance, reliability, and behavior. These snippets can be executed against a variety of API architectures, including REST, GraphQL, SOAP, and gRPC.
- **Validate complex, end-to-end workflows:** Teams can use Postman's Collection Runner to chain requests together, execute them in specific sequences, and log test results. This enables teams to create test suites that validate the most sophisticated, business-critical user journeys.
- **Automate test executions on Postman Cloud:** With Postman, users can schedule test executions to occur at specified times and frequencies—and get notified when tests fail. These executions will run on Postman's infrastructure, so teams don't have to worry about failures that result from unavailable local resources.
- **Execute API tests within CI/CD pipelines:** Teams can use Newman or the Postman CLI to run collections and tests within their CI/CD pipeline. This helps ensure backwards compatibility by safeguarding the production environment from breaking changes
- **Monitor test results and troubleshoot bugs:** Postman users can debug test failures with the Postman Console, which enables them to inspect every network call—including headers, certificates, requests, and responses. They can also monitor results from manual and automated runs—and access reports to identify testing trends.
- **Create a dedicated testing environment:** Postman lets users store values in variables at the environment level, which can be used in any request's URL, headers, and body. This allows users to run automated API tests in a dedicated testing environment before deploying code to production.
- **Incorporate Postman API tests into incident response workflows:** Postman integrates with several third-party monitoring and incident response tools, such as Datadog, New Relic, and OpsGenie, which allows teams to augment their incident response workflows with Postman test data.

Postman

Postman is an API testing tool that can generate reports in various formats such as HTML, JSON, and CSV. These reports can be customized to include specific metrics and can be shared with other team members.

four operations that we can perform:-

GET

POST

PUT

DELETE

Manual Testing

Manual testing is a software testing process in which test cases are executed manually without using any automated tool. All test cases executed by the tester manually according to the end user's perspective. It ensures whether the application is working, as mentioned in the requirement document or not. Test cases are planned and implemented to complete almost 100 percent of the software application. Test case reports are also generated manually.

Manual Testing is one of the most fundamental testing processes as it can find both visible and hidden defects of the software. The difference between expected output and output, given by the software, is defined as a defect. The developer fixed the defects and handed it to the tester for retesting.

Manual testing is a process of verifying the functionality, usability, performance, and other aspects of software applications or systems manually, without the use of automation tools or scripts. It involves testing software applications or systems using various techniques and methods to ensure that they meet the required standards and specifications.

Manual testing is performed by trained testers or quality assurance professionals who execute test cases and scenarios to identify defects or errors in the software. The testing process involves executing the software with various inputs and evaluating the outputs to determine if the software behaves as expected.

The manual testing process can be broken down into several phases, including:

- 1. Test Planning:** This phase involves defining the testing objectives, test scope, test environment, and resources required for testing.
- 2. Test Case Development:** This phase involves creating test cases, test scripts, and test scenarios based on the functional requirements, design specifications, and other documents.
- 3. Test Execution:** This phase involves executing the test cases and scenarios and recording the results. Testers should document any defects, issues, or errors they encounter.
- 4. Defect Reporting:** This phase involves reporting defects or errors to the development team or project manager for resolution.

5. Test Closure: This phase involves reviewing the test results, creating test reports, and evaluating the overall effectiveness of the testing process.

Manual testing techniques include:

1. Functional Testing: This technique focuses on testing the functionality of the software application or system to ensure that it meets the required specifications and works as intended.

2. Usability Testing: This technique focuses on testing the user interface and user experience of the software application or system to ensure that it is user-friendly and easy to use.

3. Performance Testing: This technique focuses on testing the performance of the software application or system under various load conditions to ensure that it can handle the expected user load.

4. Compatibility Testing: This technique focuses on testing the compatibility of the software application or system with different devices, browsers, and operating systems to ensure that it works correctly on all supported platforms.

Manual testing tools include:

1. Test Management Tools: These tools help testers manage test cases, test scripts, and test scenarios, and track testing progress.

2. Defect Tracking Tools: These tools help testers track defects and issues and manage the resolution process.

3. Test Execution Tools: These tools help testers execute test cases and scenarios and record the results.

4. Documentation Tools: These tools help testers create and manage test reports and other documentation related to the testing process.

Manual testing is mandatory for every newly developed software before automated testing. This testing requires great efforts and time, but it gives the surety of bug-free software. Manual Testing requires knowledge of manual testing techniques but not of any automated testing tool.

Why we need manual testing

Whenever an application comes into the market, and it is unstable or having a bug or issues or creating a problem while end-users are using it.

If we don't want to face these kinds of problems, we need to perform one round of testing to make the application bug free and stable and deliver a quality product to the client, because if the application is bug free, the end-user will use the application more conveniently.

If the test engineer does manual testing, he/she can test the application as an end-user perspective and get more familiar with the product, which helps them to write the correct test cases of the application and give the quick feedback of the application.

Types of Manual Testing

There are various methods used for manual testing. Each technique is used according to its testing criteria. Types of manual testing are given below:

- White Box Testing
- Black Box Testing
- Gray Box Testing

CHAPTER4

CONCLUSION

All these learnings with my prior knowledge of Java/C++ made me suitable to work on a live project and contribute to the organization.

Though not much can be disclosed about the project, but the learnings sure can, so here are the things that I can conclude from my about 4 months of internship experience:

- All the learnings till date were finally used while working on the live project.
- No number of personal projects can replicate the experience of working on a live project.
- As a fresher, every day was a new learning.
- With each task assigned to me, it felt like a responsibility on our shoulders that had to be carried out diligently.
- Our wonderful team never led stress to develop, and the work was being carried out in a smooth, orderly manner.
- Completing every assigned task before deadline gave me confidence to continue moving forward.
- Any hurdles faced were looked upon and the optimum way was paved for us.
- Work-Life balance was always taken in consideration.
- I learnt so much about interpersonal skills, making connections, Independence, working hard no matter what you're doing, Taking constructive criticism well.
- Also, during the time of my Internship I learnt that what you learnt in college are helpful but not sufficient, so you must acquire new and improved skills and how to apply them to solve the problems in work and in life as well.

CHAPTER5

References

- <https://learn.epam.com/start>
- <https://www.guru99.com/software-testing.html>
- <https://www.javatpoint.com/selenium-tutorial>
- <https://artoftesting.com/selenium-tutorial>
- <https://stackoverflow.com/questions/tagged/selenium-webdriver>