Software Testing

EPAM

A Training Report

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

Bachelor of Technology

Computer Science and Engineering

(Software Testing)

LOVELY PROFESSIONAL UNIVERSITY

PHAGWARA, PUNJAB

From 13/01/2023 to 28/04/2023

SUBMITTED BY

Name of the student: Mohiuddin

Registration Number: 11914561

Signature of the student: .Mohiuddin

SUBMITTED TO

Name of the supervisor: **Sakshi**

Designation:

Signature of the supervisor:

Student Declaration

To whom so ever it **may concern**

I, Mohiuddin, 11914561, hereby declare that the work done by me on

"SOFTWARE TESTING TRAINING" from 13th january-2023 to 28th April, under the supervision of **Sakshi**, **EPAM. Designation**, **EPAM**, Designation, Lovely professional University, Phagwara, Punjab, is a record of original work for the partial fulfillment of the requirements for the award of the degree, Computer Science and Engineering.

Name of the Student (Registration Number)

Mohiuddin(l1914561)

Mohiuddin

Signature of the student Dated: 01/05/2023

Declaration by the supervisors

To whom so ever it may concern

This is to certify that Mohuddin 11914561 from Lovely Professional University, Phagwara, Punjab, has worked as a trainee in **Epam** on “**Software Testing**” under my Supervision from **13th January 2023 to 28th April.** 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
Human Resource Manager	Assistant Professor
Designation of the External Supervisor	Designation of the Internal Supervisor
Signature of the external Supervisor	Signature of the Internal Supervisor
Dated:	Dated:

Table of Contents

CHAPTER!

INTRODUCTION TO COMPANY

1.1 About EPAM

EPAM Systems is a global technology services company that provides software engineering, digital platform engineering, and product development services to businesses in various industries. The company was founded in 1993 in Princeton, New Jersey, and has since grown to over 47,000 employees across 35 countries.

EPAM's services include application development, testing and maintenance, digital platform engineering, UX design, data and analytics, and consulting. The company serves clients in various industries, including financial services, healthcare, retail and distribution, travel and hospitality, and media and entertainment.

EPAM is recognized as a leader in the software development industry and has won numerous awards for its work. The company has been listed on the Forbes list of America's Best Employers, the Fortune 100 Fastest-Growing Companies list, and the Forbes Global 2000 list of the world's largest public companies.

EPAM's mission is to help clients leverage digital technologies to transform their businesses and improve their customer experiences. The company is committed to innovation, collaboration, and delivering high-quality solutions to its clients.

Company's Vision and Mission

EPAM's vision is to be the digital platform engineering and software engineering partner of choice for the world's leading organizations. The company aims to achieve this by delivering innovative and high-quality solutions that help clients transform their businesses and stay ahead of the competition.

EPAM's mission is to help clients leverage digital technologies to create engaging and personalized experiences for their customers. The company is committed to delivering measurable value to its clients by providing expert guidance, innovative solutions, and exceptional service.

EPAM's values are centered around innovation, collaboration, excellence, customer focus, and social responsibility. The company believes in fostering a culture of continuous learning and development, where employees are encouraged to innovate and contribute to the success of their clients.

Overall, EPAM's vision and mission are focused on helping its clients succeed in the digital age by providing expert guidance and innovative solutions that transform their businesses and enhance their customer experiences.

Company Core Values:

Innovation: EPAM encourages its employees to be innovative and to explore new ideas and approaches in order to develop cutting-edge solutions for its clients**.**

Collaboration: EPAM values teamwork and collaboration, recognizing that the best solutions are often developed through the combined efforts of a diverse group of individuals**.**

Excellence: EPAM strives for excellence in everything it does, from the quality of its solutions to the level of service it provides to its clients.

Customer focus: EPAM is committed to understanding its clients' needs and preferences in order to deliver solutions that meet or exceed their expectations**.**

Social responsibility: EPAM is committed to being a responsible corporate citizen and to making a positive impact in the communities where it operates.

Origin and growth of company

EPAM Systems was founded in 1993 in Princeton, New Jersey by Arkadiy Dobkin and Leo Lozner, who were both software engineers. The company started as a small software development startup with a handful of employees, primarily serving clients in the financial services industry.

In the early years, EPAM focused on providing custom software development services to its clients. However, as the company grew, it expanded its offerings to include digital platform engineering, product development, and consulting services.

EPAM experienced significant growth throughout the 2000s and 2010s, expanding its global footprint and increasing its employee base. The company went public in 2012, trading on the New York Stock Exchange under the ticker symbol "EPAM."

Today, EPAM has over 47,000 employees across 35 countries, serving clients in various industries, including financial services, healthcare, retail and distribution, travel and hospitality, and media and entertainment. The company has received numerous awards for its work, including being recognized as a leader in the software development industry.

EPAM's growth has been driven by its focus on delivering innovative solutions to its clients and providing exceptional service. The company has also invested heavily in its employees, fostering a culture of continuous learning and development to ensure that its workforce is equipped to handle the challenges of the digital age.

Various departments and their functions

EPAM Systems has various departments that work together to deliver high-quality solutions to its clients. Here are some of the key departments and their functions:

Software Engineering: This department is responsible for developing custom software solutions for clients across various industries. The team works closely with clients to understand their business needs and develop software that meets their specific requirements.

Digital Platform Engineering: This department focuses on designing and building digital platforms that enable clients to deliver engaging and personalized experiences to their customers. The team leverages cutting-edge technologies to create scalable and secure platforms that can handle large volumes of data and traffic.

Product Development: This department is responsible for developing and launching new products for clients in various industries. The team works closely with clients to identify market opportunities and develop products that meet customer needs and preferences.

Quality Assurance and Testing: This department is responsible for ensuring that all software and digital platforms developed by EPAM meet the highest quality standards. The team uses a variety of testing methodologies to identify and resolve issues before software is released to clients.

UX/UI Design: This department is responsible for designing user interfaces and experiences that are intuitive, engaging, and visually appealing. The team works closely with clients to understand their target audience and design interfaces that meet their needs and preferences.

Data and Analytics: This department focuses on leveraging data and analytics to drive business insights and inform decision-making. The team works with clients to collect, analyze, and visualize data, providing actionable insights that can improve business performance.

Consulting: This department provides strategic guidance to clients, helping them navigate the complex landscape of digital transformation. The team works closely with clients to understand their business objectives and develop solutions that drive growth and profitability.

CHAPTER2

INTRODUCTION TO INDIVIDUAL PROJECT

Objectives of the work undertaken:

- To automate the script for calculating the pricing of a Google Cloud Platform instance using Selenium WebDriver, framework unit test, and Page Object concepts.

- To ensure that the monthly rent matches the amount received when the test is done manually.

- To use the Page Object concept to abstract the pages of the application and to interact with the web elements.

- To use the framework unit test to create test scripts for the given steps.

- To use Property files to store test data for at least two different environments.

- To use XML suites for smoke tests and other tests.

- To include an option for running the test with Jenkins and browser parameterization, test suite, and environment.

- To display the test results on the job chart and archive the screenshots as artifacts in case of test failure.

Scope of the Work:

- The scope includes automating the given script for calculating the pricing of a Google Cloud Platform instance using Selenium WebDriver, framework unit test, and Page Object concepts.

- The scope also includes using Property files to store test data for at least two different environments.

- The scope includes using XML suites for smoke tests and other tests.

- The scope includes incorporating an option for running the test with Jenkins and browser parameterization, test suite, and environment.

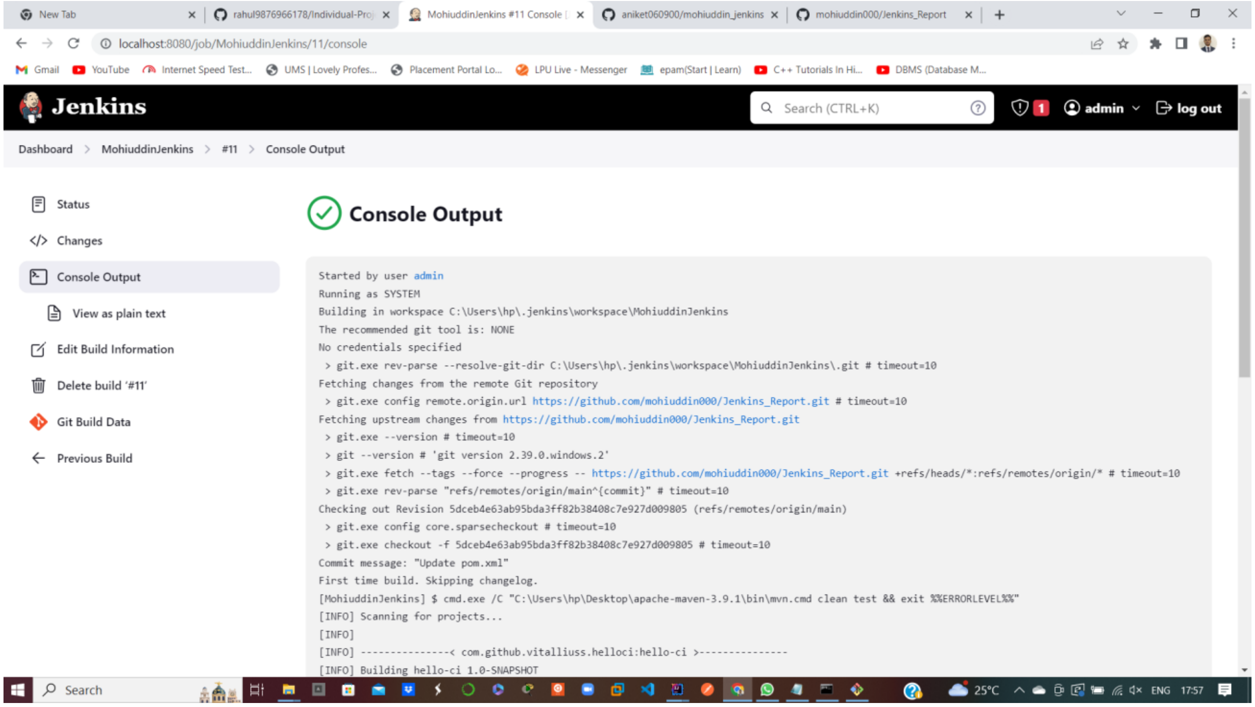
- The scope includes displaying the test results on the job chart and archiving the screenshots as artifacts in case of test failure.

- The scope does not include testing the functionality of the Google Cloud Platform Pricing Calculator or any other component of the Google Cloud Platform.

- The scope does not include testing the performance or security of the Google Cloud Platform Pricing Calculator or any other component of the Google Cloud Platform.

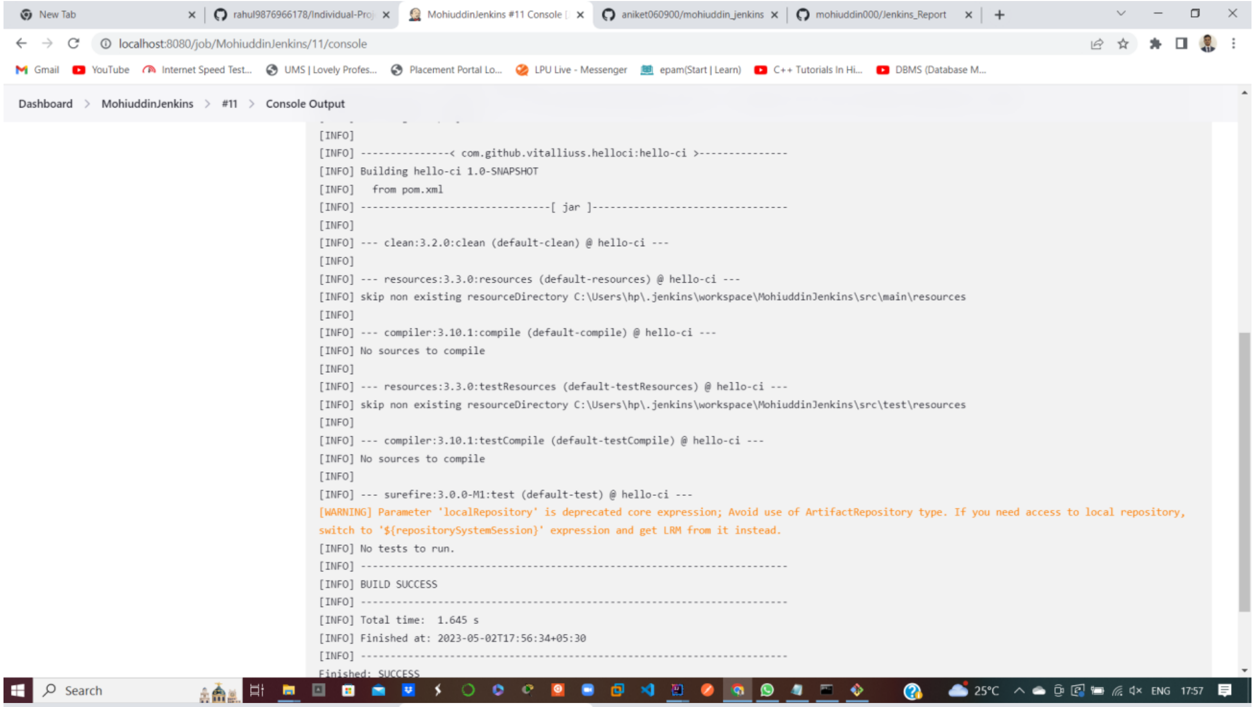
- The scope does not include testing the compatibility of the Google Cloud Platform Pricing Calculator or any other component of the Google Cloud Platform with different browsers, operating systems, or devices.

Jenkins Report (module 4th)

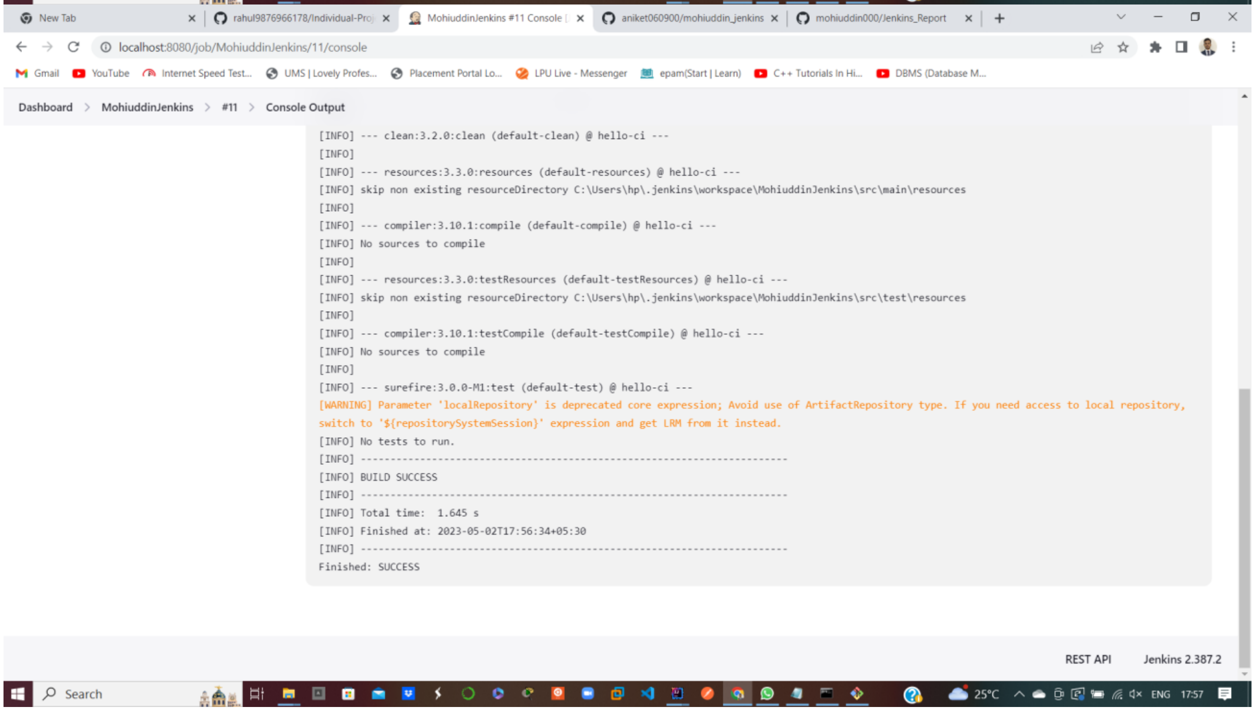


Console Output

```
Started by user admin
Running as SYSTEM
Building in workspace C:\Users\hp\.jenkins\workspace\MohiuddinJenkins
The recommended git tool is: NONE
No credentials specified
 > git.exe rev-parse --resolve-git-dir C:\Users\hp\.jenkins\workspace\MohiuddinJenkins\.git # timeout=10
Fetching changes from the remote Git repository
 > git.exe config remote.origin.url https://github.com/mohiuddin000/Jenkins_Report.git # timeout=10
Fetching upstream changes from https://github.com/mohiuddin000/Jenkins_Report.git
 > git.exe --version # timeout=10
 > git --version # 'git version 2.39.0.windows.2'
 > git.exe fetch --tags --force --progress -- https://github.com/mohiuddin000/Jenkins_Report.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git.exe rev-parse "refs/remotes/origin/main^{commit}" # timeout=10
Checking out Revision 5dceb4e63ab95bda3ff82b38408c7e927d009805 (refs/remotes/origin/main)
 > git.exe config core.sparsecheckout # timeout=10
 > git.exe checkout -f 5dceb4e63ab95bda3ff82b38408c7e927d009805 # timeout=10
Commit message: "Update pom.xml"
First time build. Skipping changelog.
[MohiuddinJenkins] $ cmd.exe /C "C:\Users\hp\Desktop\apache-maven-3.9.1\bin\mvn.cmd clean test && exit %%ERRORLEVEL%%"
[INFO] Scanning for projects...
[INFO]
[INFO] --------------< com.github.vitalliuss.helloci:hello-ci >---------------
[INFO] Building hello-ci 1.0-SNAPSHOT
```



```
[INFO]
[INFO] ---------------< com.github.vitalliuss.helloci:hello-ci >---------------
[INFO] Building hello-ci 1.0-SNAPSHOT
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- clean:3.2.0:clean (default-clean) @ hello-ci ---
[INFO]
[INFO] --- resources:3.3.0:resources (default-resources) @ hello-ci ---
[INFO] skip non existing resourceDirectory C:\Users\hp\.jenkins\workspace\MohiuddinJenkins\src\main\resources
[INFO]
[INFO] --- compiler:3.10.1:compile (default-compile) @ hello-ci ---
[INFO] No sources to compile
[INFO]
[INFO] --- resources:3.3.0:testResources (default-testResources) @ hello-ci ---
[INFO] skip non existing resourceDirectory C:\Users\hp\.jenkins\workspace\MohiuddinJenkins\src\test\resources
[INFO]
[INFO] --- compiler:3.10.1:testCompile (default-testCompile) @ hello-ci ---
[INFO] No sources to compile
[INFO]
[INFO] --- surefire:3.0.0-M1:test (default-test) @ hello-ci ---
[WARNING] Parameter 'localRepository' is deprecated core expression; Avoid use of ArtifactRepository type. If you need access to local repository,
switch to '${repositorySystemSession}' expression and get LRM from it instead.
[INFO] No tests to run.
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  1.645 s
[INFO] Finished at: 2023-05-02T17:56:34+05:30
[INFO] ------------------------------------------------------------------------
Finished: SUCCESS
```



```
[INFO] --- clean:3.2.0:clean (default-clean) @ hello-ci ---
[INFO]
[INFO] --- resources:3.3.0:resources (default-resources) @ hello-ci ---
[INFO] skip non existing resourceDirectory C:\Users\hp\.jenkins\workspace\MohiuddinJenkins\src\main\resources
[INFO]
[INFO] --- compiler:3.10.1:compile (default-compile) @ hello-ci ---
[INFO] No sources to compile
[INFO]
[INFO] --- resources:3.3.0:testResources (default-testResources) @ hello-ci ---
[INFO] skip non existing resourceDirectory C:\Users\hp\.jenkins\workspace\MohiuddinJenkins\src\test\resources
[INFO]
[INFO] --- compiler:3.10.1:testCompile (default-testCompile) @ hello-ci ---
[INFO] No sources to compile
[INFO]
[INFO] --- surefire:3.0.0-M1:test (default-test) @ hello-ci ---
[WARNING] Parameter 'localRepository' is deprecated core expression; Avoid use of ArtifactRepository type. If you need access to local repository,
switch to '${repositorySystemSession}' expression and get LRM from it instead.
[INFO] No tests to run.
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  1.645 s
[INFO] Finished at: 2023-05-02T17:56:34+05:30
[INFO] ------------------------------------------------------------------------
Finished: SUCCESS
```

CHAPTER3

TECHNOLOGIES LEARNT

Software Development Methodologies:

- **High Level Overview**: An overview of software development methodologies, their benefits, and their various types.

- **Waterfall:** An introduction to the traditional Waterfall methodology, which is a linear and sequential approach to software development.

- **Agile:** An overview of the Agile methodology, which emphasizes flexibility, customer collaboration, and incremental and iterative development.

- **Scrum:** An in-depth look at the Scrum framework, which is a popular Agile methodology that uses sprints, backlogs, and daily stand-ups to manage projects.

- **Kanban:** An introduction to the Kanban methodology, which emphasizes visual management, continuous flow, and limiting work in progress.

- **Extreme Programming:** An overview of Extreme Programming (XP), which is an Agile methodology that emphasizes customer involvement, continuous testing, and frequent releases.

- **Test-Driven Development:** An introduction to Test-Driven Development (TDD), which is a software development approach that focuses on creating automated tests before writing code.

- **Behavior-Driven Development:** An overview of Behavior-Driven Development (BDD), which is a software development approach that focuses on describing the behavior of a system in natural language.

- **Summary:** A summary of the main points of each methodology, their strengths and weaknesses, and how to choose the right methodology for your project.

- Extras: Advanced topics such as DevOps, Continuous Integration/Continuous Deployment (CI/CD), and how to combine different methodologies for hybrid approaches.

Version Control with GIT:

- **Version control concept**: Understanding the basics of version control, its benefits, and its various types.

- **Download, install and configure GIT:** Installing GIT on your local machine and configuring it with your user details and preferences.

- **GitHub:** Introduction to GitHub, a web-based hosting service for version control, and its features such as repositories, issues, and pull requests.

- **Git graphical tools:** Overview of graphical user interfaces (GUI) and Integrated Development Environments (IDEs) that can be used to work with Git.

- **Git internals:** Understanding the inner workings of Git, including how Git stores and manages versions of files, branches, commits, and merges.

- **Undoing changes:** How to undo changes made to files or the repository using Git commands such as revert, reset, and checkout.

- **Branching and merge:** Creating and managing branches in Git, and merging changes from one branch to another.

- **Tags:** Creating and managing tags in Git to mark specific points in the repository's history, such as release versions.

- **Stash:** How to use Git stash to temporarily save changes that are not yet ready to be committed.

- **Remotes:** Working with remote repositories in Git, such as cloning, pushing, and pulling changes from remote repositories.

- **Branching strategies**: Overview of different branching strategies and workflows such as Gitflow and Github Flow.

Software Testing Introduction:

- **Introduction to Software Functional Testing:** An overview of software functional testing, its importance, and the different types of functional testing.

- **Test Planning:** An overview of test planning, which involves identifying test objectives, test strategies, and test schedules.

- **Requirements Testing:** An introduction to requirements testing, which involves verifying that the software meets the specified requirements.

- **Test Cases and Test Scenarios:** An in-depth look at test cases and test scenarios, which are used to define the conditions under which software will be tested and the expected results.

- **Defect Reporting:** An overview of defect reporting, which involves identifying and documenting defects in the software.

- **Test Results Reporting:** An introduction to test results reporting, which involves analyzing and documenting the results of the software testing.

- **Test Automation Basics:** An overview of test automation, which involves using software tools to automate the testing process and reduce manual effort.

Java Basics:

- **Introduction to the Java Basics Course:** An overview of the course, its objectives, and the topics covered.

- **Data Types:** An introduction to data types in Java, including primitive and reference types, and how to declare and use them.

- **Conditions and Loops:** An introduction to conditional statements, such as if-else statements and switch statements, and loops such as for loops and while loops.

- **Arrays:** An introduction to arrays in Java, including how to declare and initialize arrays, and how to access and modify their elements.

- **Classes:** An introduction to classes in Java, including how to declare and instantiate classes, and how to use them to create objects.

- **Introduction to OOP:** An introduction to object-oriented programming (OOP) concepts, such as encapsulation, inheritance, and polymorphism.

- **Abstract Classes and Interfaces:** An introduction to abstract classes and interfaces, which are used to define abstract types that can be implemented by other classes.

- **Nested Classes:** An introduction to nested classes in Java, including static nested classes, inner classes, and anonymous classes.

- **Strings:** An introduction to strings in Java, including how to create and manipulate string objects.

- **Collections and Maps:** An introduction to collections and maps in Java, including how to use them to store and manipulate groups of objects.

- **Exceptions:** An introduction to exceptions in Java, including how to use try-catch blocks to handle exceptions and how to create custom exceptions.

- **Annotations:** An introduction to annotations in Java, which are used to provide metadata about code elements.

- **Generics:** An introduction to generics in Java, which allow you to define classes and methods that can work with different types of objects.

- **Enum:** An introduction to enumerations in Java, which are used to define a fixed set of values.

- **Wrapper Classes and Optional Classes:** An introduction to wrapper classes, which are used to represent primitive data types as objects, and optional classes, which are used to represent values that may be null.

- **Code Documentation:** An introduction to code documentation in Java, including how to use Javadoc to document your code.

Data & Analytics - Introduction to SQL:

- **Database Basics:** An introduction to databases, their types, components, and architecture.
- **SQL Foundation:** An introduction to Structured Query Language (SQL), including how to create and manipulate tables, perform queries, and use aggregate functions.
- **SQL for Analysis:** An introduction to using SQL for data analysis, including how to filter, sort, and group data, and how to join tables

Clean Code - Introduction to Clean Code:

- **Introduction to Clean Code:** An overview of clean code and why it is important in software development.
- **Writing Clean Functions:** Best practices for writing clean functions, including how to make them small, focused, and testable.
- **Naming:** Best practices for naming code elements, including classes, methods, and variables.
- **Comments:** Best practices for writing comments, including when to use them and what to include.

- **Error Handling:** Best practices for error handling, including how to handle exceptions and write code that is resilient to errors..

Cloud Computing - Introduction to Cloud Computing:

- **Introduction to Cloud Computing:** An overview of cloud computing and its benefits.
- **Cloud Deployment Models:** An introduction to different cloud deployment models, including public, private, and hybrid clouds.
- **Cloud Service Models:** An introduction to different cloud service models, including infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS).
- **Cloud Providers:** An introduction to different cloud providers, such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP).
- **Cloud Security:** An overview of cloud security, including best practices for securing cloud-based applications and data.

Automated Testing Basics + Java:

OO Design Principles & Patterns

:

- **Patterns in General:** An introduction to different types of design patterns, including structural, behavioral, and creational patterns.
- **Factory Pattern:** A creational pattern that provides a way to create objects without specifying their concrete classes.
- **Strategy Pattern:** A behavioral pattern that enables the selection of an algorithm at runtime.

- **Builder Pattern:** A creational pattern that separates the construction of a complex object from its representation.
- **Singleton Pattern:** A creational pattern that ensures a class has only one instance and provides a global point of access to it.

Introduction to Test Automation and xUnit Test Framework:

- **Introduction to Test Automation:** An overview of test automation and its benefits.
- **UI/API/Performance/Security/Mobile Testing:** An introduction to different types of test automation, including user interface (UI), application programming interface (API), performance, security, and mobile testing.
- **Build Tools (Maven):** An introduction to Maven, a build automation tool used primarily for Java projects.
- **TestNG:** An xUnit test framework for Java that supports parameterized, data-driven, and parallel testing.

API Automation:

- **Client-Server Architecture:** An overview of client-server architecture, which is the basis of most web applications.
- **HTTP and HTTP Request/Response:** An introduction to Hypertext Transfer Protocol (HTTP) and HTTP request/response messages.
- **JSON and XML:** An introduction to JavaScript Object Notation (JSON) and Extensible Markup Language (XML), two commonly used data formats for web applications.
- **Postman:** A popular tool for testing and debugging API requests.
- **Rest Assured:** A Java-based library for testing RESTful web services.

Selenium WebDriver (Basic+Advanced):

- **Introduction:** An overview of Selenium WebDriver, a popular web testing tool.
- **HTML and CSS:** An introduction to HTML and CSS, the building blocks of web pages.
- **XPath:** A language used for selecting nodes in an XML or HTML document.
- **Selenium WebDriver:** An introduction to the WebDriver API and its methods for interacting with web pages.
- **WebDriver Waiters:** An introduction to the wait methods in WebDriver, which can help synchronize tests with the page under test.
- **JS Executor:** An advanced feature of WebDriver that allows JavaScript code to be executed within the context of a web page.

Automation Framework:

- **Automation Framework:** Maven + xUnit + WebDriver: This involves the use of Maven as a build tool, xUnit as the testing framework, and WebDriver for automating the web application. This helps in creating a structured and scalable test automation framework.
- **Page Object:** This is a design pattern that separates the page elements of a web application from the test scripts, making the tests more maintainable and reducing code duplication.
- **Page Factory:** This is an extension of the Page Object pattern that uses annotations to initialize web elements, making the code more readable and maintainable.
- **Singleton:** This is a design pattern that ensures only one instance of a class is created, which can be useful in maintaining the state of the application during the test.
- **Production AT Framework:** This includes various approaches like TDD (Test Driven Development), KDT (Keyword Driven Testing), DDT (Data Driven Testing), DDD

(Domain Driven Design), BDD (Behavior Driven Development), and BDD + Cucumber. These approaches help in creating a robust and maintainable test automation framework that aligns with the development process.

- **ATF Architecture:** This is an architecture that helps in structuring the test automation framework and integrating it with the development process.
- **Continuous Integration with Jenkins:** This involves integrating the test automation framework with Jenkins, a popular continuous integration tool, to run the tests automatically and generate reports. This helps in achieving continuous testing and faster feedback.

CONCLUSION

In conclusion, this project on eBay website testing with Selenium and Java provides a detailed overview of several key technologies and methodologies used in software development and testing. The project covers software development methodologies such as Waterfall, Agile, Scrum, Kanban, extreme programming, test-driven development, and behavior-driven development. It also provides an overview of version control with GIT, software testing, Java basics, data and analytics, clean code, cloud computing, and automation frameworks.

By completing this project, learners will have a solid understanding of the concepts and tools used in software development and testing. They will also gain hands-on experience with technologies such as Selenium WebDriver, TestNG, Rest Assured, and Jenkins, which are widely used in the industry. This project is a valuable resource for anyone looking to learn about software development and testing and will provide a strong foundation for further learning and exploration in this field.