# **Aims/Description**

GitHub is a web-based user interface that uses Git, it is an open-source version control software that lets several people make distinct changes to the source code at the same point in time. It is the largest software development platform and an open-source repository in the world. It offers cloud storage for source code, supports all general programming languages, and streamlines the iteration process for developers. It shortens the process of working with other people and makes it easy to collaborate on projects. Team members can work on files and easily merge their edits in the main branch of the project. This practical involved an investigation of this product. The aims were:

1. The GitHub tool was investigated about tracking changes in any set of files, usually used for managing work among developers collaboratively developing source code during software development life cycle.

2. An exploration of various features like pull request, managing the source code, actions, project was to be carried out

3. An evaluation of the features of the tool was to be carried out with reference to the importance of these features to the DevOps process.

4. Further investigation into the selection of tools for source code management was to be carried out.

5. Research into the use of the tool for SDLC purpose regarding create pull request for a repository, merge request, automating releases, collaborating with your user, and automatically run your software testing scripts was to be carried out.

6. The author was inexperienced in this area and needed to do additional research to align the tool to its role in the DevOps process cycle.

# **Method**

1. Documentation relating to GitHub was reviewed
2. An account was created. The account aided lot of features to setup and working in the directory.
3. Created a new repository ‘CalculatorApp’ in github and added desired files in it (readme.md, git. in it) and copied new repository URL to connect the remote and local repository.
4. Installed the gihub plugin for eclipse via marketplace
5. A maven java project called ‘CalculatorApp’ was created using eclipse and a java file called “CalculatorApp.java” was added with methods – add, subtract, multiply and divide. Updated required dependencies in root maven pom.xml in the project for junit and maven compiler version
6. Corresponding unit test case was written using junit to test the 4 methods exposed via the java file
7. Test results were verified using the maven clean and maven test actions via eclipse
8. Add remote orgin for the github repo and committed / pushed the code to the above created repository
9. Github actions was used to create the automated workflow. Github actions was part of the repository itself, which required basic details about the JDK and maven actions.
10. Java plugin for maven was enable for the github actions to configure the maven process
11. Maven action clean and test was added to the workflow and it was configured to trigger on push of the code to the main branch
12. Github actions added ‘maven.yml’ to the .github/workflows folder on the root level, on commit of the file, it automatically triggered the build that execute the test cases

Results:

1. Reviewing the github documents lead to a greater understanding of the options available to automate the build and test execution flow
2. A github repo was successfully created to push the already existing local java maven project
3. A CalculatorApp was created to demonstrate the basic functionality of writing test driven development and same test was used to do automated testing on the code developed
4. Maven configurations was performed successfully to run the program as well as the test case on the local eclipse environment
5. Remote origin was added for the local repo and code was committed to the gihub cloud
6. Github actions plugin “java for maven” was enabled before creating the workflow
7. Workflow file maven.yml was updated with required steps to trigger the build and run the maven goals
8. On successful commit to the main branch, the build triggered automatically and the automated test cases were run successfully as expected
9. An end to end flow of updating the code to gihub and triggering the automated test cases were verified

Github

Version control

Bitbucket

Githubactions

Devops pipeline

Trigger build on commit

Is same feature available in Bitbucket

Compare and contrast gihub vs bitbucket features

Why did u choose github

What is the advantage of it over other

Guthub action – plugin features

Quickly installed java for maven plugin

By default workglow gave sample code, which can be easily modified for basic scenariios

I learned java, especially unit test testin, pros advantange development help, test driven development

U developed test cases, so every time when there is a change in source code all cases were tested automatcailly

How it will be useful in a wider team, multiple team members advtanges

 Github actually makes it really easy to get this workflow Github actually makes it really easy to get this workflow

Once there, we'll immediately see some starter workflows that Github provides for us to dive in with. Since we're using a node project, we can go ahead and click **Set up this workflow** under the **Node.js** workflow.

### Creating a new action

### Creating a new action

We'll start by navigating to the **Actions** tab on our repository page.

After the page loads, Github will land you on a new file editor that already has a bunch of configuration options added.

We're actually going to leave this "as is" for our first step. Optionally, you can change the name of the file to tests.yml or something you'll remember.

You can go ahead and click **Start commit** then either commit it directory to the master branch or add the change to a new branch. For this walkthrough, I'll be committing straight to master.

To see our new action run, we can again click on the **Actions** tab which will navigate us back to our new Actions dashboard.

Add a test which reveals a problem.

Open an issue on GitHub/GitLab.Fix the broken test

GitHub is a website and cloud-based service that helps developers store and manage their code, as well as track and control changes to their code. To understand exactly what GitHub is, you need to know two connected principles:

* Version control,git

## What Is Version Control?

Version control helps developers track and manage changes to a software project’s code. As a software project grows, version contro (kinnsmann, n.d.)l becomes essential. Take WordPress…

At this point, WordPress is a pretty big project. If a core developer wanted to work on one specific part of the WordPress codebase, it wouldn’t be safe or efficient to have them directly edit the “official” source code.

Instead, version control lets developers safely work through **branching** and **merging**.

With **branching**, a developer duplicates part of the source code (called the **repository**). The developer can then safely make changes to that part of the code without affecting the rest of the project.

Then, once the developer gets his or her part of the code working properly, he or she can **merge** that code back into the main source code to make it official.

All of these changes are then tracked and can be reverted if need be.

## What Is Git?

## What Is Git?

Git is a **specific open-source version control system** created by Linus Torvalds in 2005.

Specifically, Git is a **distributed version control system**, which means that the entire codebase and history is available on every developer’s computer, which allows for easy branching and merging.

GitHub is a for-profit company that offers a cloud-based Git repository hosting service. Essentially, it makes it a lot easier for individuals and teams to use Git for version control and collaboration.

GitHub’s interface is user-friendly enough so even novice coders can take advantage of Git. Without GitHub, using Git generally requires a bit more technical savvy and use of the command line.

GitHub is so user-friendly, though, that some people even use GitHub to manage other types of projects – [like writing books](http://braythwayt.com/2015/01/29/how-i-write-books-with-github-and-leanpub.html).

Additionally, anyone can sign up and host a public code repository for free, which makes GitHub especially popular with open-source projects.

As a company, GitHub makes money by selling hosted private code repositories, as well as other business-focused plans that make it easier for organizations to manage team members and security. We utilize Github extensively at Kinsta to manage and develop internal projects.

(Kinsman *et al.* 2021)