Devops

Assignment No 1

20F-0369

Muhammad Mustajab

[School]

[Course title]

# **Task: GitHub Action:**

GitHub Actions is a powerful and flexible automation platform provided by GitHub for automating various tasks in your software development workflow. It allows you to build, test, and deploy your code directly from your GitHub repository. Here's an exploration of some key concepts and features of GitHub Actions:

# **Workflow:**

A workflow is a set of automated actions that you define in a YAML file. It describes the entire automation process, including when and how each action should run. Workflows are typically triggered by events such as code pushes, pull requests, or scheduled jobs.

**Workflow YAML File:**

The workflow configuration is defined in a YAML file (.github/workflows/workflow-name.yml) within your repository. This file specifies the name of the workflow, the event trigger, and the sequence of jobs and actions to execute.

**Events:**

GitHub Actions can be triggered by various events. Common triggers include `push` (when code is pushed to the repository), `pull\_request` (when a pull request is opened or updated), and `schedule` (for scheduled tasks).

**Jobs:**

A workflow can consist of one or more jobs, which are executed in parallel by default. Each job can run on a specific virtual environment, such as Ubuntu, macOS, or Windows. You can define the steps to be executed in each job.

# **Actions:**

Actions are the individual tasks or steps within a job. GitHub provides a marketplace of pre-built actions that you can use, or you can create your own custom actions. Actions can perform tasks like building your code, running tests, deploying to a server, or sending notifications.

# **Environment Variables:**

You can set environment variables for your workflow to store secrets or configuration values. These variables can be used within your actions and jobs.

# **Secrets:**

GitHub Actions allows you to store sensitive data such as API keys or access tokens as encrypted secrets in your repository settings. These secrets can be accessed by your workflow without revealing their values in plain text.

# **Caching:**

You can cache dependencies and build artifacts to speed up workflow execution. This is particularly useful for large projects where downloading dependencies or building code takes a significant amount of time.

# **Matrix Builds:**

You can configure matrix builds to test your code on multiple versions of an operating system, programming language, or any other parameter. This helps ensure compatibility across different environments.

# **Workflow Status:**

GitHub Actions provides status checks and badges that can be displayed in your repository to indicate the current status of your workflows, making it easy to see if builds are passing or failing.

# **Scheduled Jobs:**

You can schedule workflows to run at specific times or intervals, which is useful for tasks like daily backups or automated reports.

# **Workflow Visualizations:**

GitHub provides a visual representation of your workflows and their execution in the Actions tab of your repository, making it easy to track the progress of your automation.

# **Notifications:**

You can set up notifications and alerts to notify team members or stakeholders when a workflow succeeds or fails.