**Hands-On Assignment: Git, GitHub, And Open-Source Exploration**

**Task 1: Cloning and Forking**

**Cloning a Repository**

* I first scouted the Github website and chose a public repository called FLUTTER-SIMPLE-PORTFOLIO
* Then I created a folder in the local machine called GitPractice
* I navigated to the path of GitPractice
* Then I performed a git init followed by git clone <repository url>
* I managed to have a look at the different folders and files within the cloned repository

**Forking a Repository**

* Here, I started with forking the previously cloned repository that is, FLUTTER-SIMPLE-PORTFOLIO
* I created another folder in the local machine called GitForkPractice and navigated to it in the terminal.
* As usual, I did git init and then cloned the forked repository.

**Task 2: Managing Branches**

**Creating and Switching Branches**

* Here, I just performed git checkout -b feature-update

**Making Changes and Committing**

* Here, I managed to create a new file (echo “There is fun in constant practice in that it makes you discover even more than what you expected. And that is coding for you.” >code.txt)
* Then I performed git add .
* I then did: git commit -m “Interesting commit”

**Merging Changes**

* I first did: git remote add upstream <original-repo url>
* Then I did: git fetch upstream
* I then did: git checkout -b main upstream/main
* Lastly, I merged the changes: git merge feature-update

**Task 3: Handling Conflicts**

**Creating Conflicts**

* When merging I encountered this error, “fatal: refusing to merge unrelated histories”.

Resolving Conflicts

* I managed to resolve this using: git merge feature-update --allow-unrelated-histories
* Unfortunately, I forgot to commit the changes

**Task 4: Github Pages**

* I created an a simple index.html file in the terminal under the main branch. Used nano to write content of the file.
* I did git add index.html and git commit -m “The message”
* I then pushed the changes using: git push upstream main
* Paid a visit to the repository and under the settings, I navigated to the Pages section where I was able to enable Github Pages for the repository as well as set the source branch to ‘main’ with ‘\root’ as the folder.

**Task 5: Open-Source Exploration**

**Exploring Open-Source Projects**

* I searched for an open-source project on Github with the keyword ‘health’.
* I stumbled upon a project that checks if the application resources are running as they should and creates a service status panel. It was called Health Monitor (Laravel Server & App Health Monitor and Notifier)
* I read through the readme file that elaborated clearly the project together. I also observed the pull requests, forks, issues and contribution guidelines

**Opening an Issue**

* Under the issues section in the open-source project, I managed to initiate a new issue entitled “Flutter Integration”
* I asked if the given solution could be used to monitor the performance and health of flutter apps.