**COURSE: AI ENGENEERING**

**ASSIGNMENT: ONE**

**QUESTION: FILE MANEGEMENT WITH GITHUB**

**NAME: ZAINAB WYTE CHIKWAMBA**

**FILE MANEGEMENT WITH GITHUB**

File management in GitHub involves creating, editing, moving, and deleting files within repositories.

**Creating New Files**:

Directly on GitHub:

Go to your repository.

Click the "Add file" button.

Enter a file name and content.

Commit the file with a descriptive message.

Using the Command Line:

Create the file locally.

Add it to your Git staging area: git add <filename>

Commit the file: git commit -m "Commit message"

Push the changes to GitHub: git push origin <branch\_name>

**Adding an Existing File:**

On GitHub:

Go to your repository.

Click "Upload files".

Select the file to upload.

Commit the file.

Using the Command Line:

Same as creating a new file.

**Moving a File**:

On GitHub:

Click the file you want to move.

Click the "Move file" button.

Choose the new

locationand commit.

Using the Command Line:

Rename the file locally: git mv <old\_filename> <new\_filename>

Add and commit the change.

**Editing Files**:

On GitHub:

Click the file to edit.

Click the "Edit" button.

Make your changes and commit.

Using the Command Line:

Edit the file locally.

Add and commit the changes.

**Renaming a File:**

On GitHub:

Click the file you want to move.

Click the "Move file" button.

Choose the new

Location and commit.

Using the Command Line:

Rename the file locally: git mv <old\_filename> <new\_filename>

Add and commit the change

**Deleting Files:**

On GitHub:

Click the file to delete.

Click the "Delete file" button.

Confirm the deletion.

Using the Command Line:

Delete the file locally: git rm <filename>

Commit the change.

Using Files

Navigating Code:

Use the file tree on the left to explore files and directories.

Search for files using the search bar.

**Viewing Files**:

Click on a file to view its content.

Use the "Blame" view to see who last modified each line.

Use the "History" tab to see the file's version history.

**Tracking File Changes:**

Use the "Compare" view to see changes between commits.

Use the "Pull Requests" tab to see proposed changes to files.

**Permanent Links:**

Right-click on a file and select "Copy permalink" to get a permanent link to the file.

Source Code Archives:

Download a ZIP file of the repository

Working with Non-Code Files:

GitHub can handle various file types, including images, PDFs, etc.

Managing Large Files

Git Large File Storage (LFS):

Use LFS to store large files outside of the main Git repository.

Install the Git LFS command-line tool.

Configure LFS to track specific file patterns.

Move large files to LFS.

Collaboration

Use branches to work on different features or bug fixes without affecting the main codebase.