**Lab 03:**

* **Create an account for your GitHub if you do not have one.**
* **Application development:**

**Create a folder to dedicate your application development.**

**Create a python script:**

|  |
| --- |
| **nano app.py** |

**Sample of content:**

|  |
| --- |
| **def main():**  **print("Hello, World!")**  **if \_\_name\_\_ == "\_\_main\_\_":**  **main()** |

**Save and exit (Ctrl + X, then Y).**

**Test your application if it runs correctly**

* **Step 1: Install Git on Ubuntu**

**Update the Package List**:  
Ensure your system is up to date:

|  |
| --- |
| sudo apt update |

**Install Git**:  
Install Git using the following command:

|  |
| --- |
| sudo apt install git  git --version |

**Step 2:** Create a .gitignore file:  
This file tells Git which files to ignore (e.g., virtual environment files).

**nano .gitignore**

Add the following:

|  |
| --- |
| venv/  \_\_pycache\_\_/  \*.pyc |

**Step3: Initialize and Stage and Commit Files**:  
Add all files to the staging area and commit them:

|  |
| --- |
| git init  git add .  git commit -m "Initial commit" |

**Step4:** Create a GitHub Repository

1. Log in to GitHub:  
   Go to [GitHub](https://github.com/) and log in to your account.
2. Create a new repository:
   * Click the + icon in the top-right corner and select "New repository."
   * Name your repository (e.g., my-python-app).
   * Choose public or private.
   * Do not initialize the repository with a README, .gitignore, or license.
3. Copy the repository URL:  
   After creating the repository, copy the HTTPS or SSH URL (e.g., https://github.com/username/my-python-app.git).

**Step 5: Connect Your Local Repository to GitHub**

1. Add the remote repository:

|  |
| --- |
| git remote add origin <https://github.com/username/my-python-app.git>  Replace the URL with your repository's URL. |

1. Push your code to GitHub:

|  |
| --- |
| git branch -M main  git push -u origin main |

1. Verify On GitHub
   1. Refresh your GitHub repository page. You should see your Python files Uploaded.

**Step 6: (Optional)Add README File**

* **Create README file (nano README.md)**
* **Add description of your project**
* **Commit and push the changes**

**Step 7: (Optional)Set up a Requirements file**

1. Generate requirements.txt file

If your project uses external libraries, generate a requirements.txt file:

|  |
| --- |
| Pip freeze > requirements.txt  Information about pip freeze <https://dev.to/eskabore/pip-freeze-requirementstxt-a-beginners-guide-5e2m> |

1. Again, add and push the file

**Task 0:**

Take any Netflix movie dataset. Apply EDA and all visualization techniques, then deploy them to your GitHub account.

Keep a screenshot of commands while deploying it.

**Task 01:**

Create and host a personal website using GitHub pages.

|  |
| --- |
| Create an index.html file for the homepage:  Add a style.css file for styling (optional):  Add, commit, and push the files to the repository: |

**Deliverables**

* A GitHub repository named username.github.io with the website files.
* The live URL of the personal website (e.g., https://username.github.io).

**Task 02:**

Automate the deployment of your application using GitHub actions by creating .yml file

Follow the above steps to deploy your application.

Example: .yaml

|  |
| --- |
| name: Deploy Application  on:  push:  branches:  - main  jobs:  deploy:  runs-on: ubuntu-latest  steps:  - name: Checkout code  uses: actions/checkout@v2  - name: Set up Python  uses: actions/setup-python@v2  with:  python-version: '3.9'  - name: Install dependencies  run: |  python -m pip install --upgrade pip  pip install -r requirements.txt  - name: Run application  run: |  python app.py & |

**Deliverables**

* A GitHub repository with the Flask app and .yml workflow file.

Keys for Git

| **Command** | **Description** |
| --- | --- |
| git fetch origin | Fetches changes from the remote repository without merging. |
| git status | Shows the status of your local repository. |
| git log origin/main..main | Shows commits in your local branch but not in the remote branch. |
| git diff origin/main | Shows file differences between your local branch and the remote branch. |
| git pull origin main | Fetches and merges changes from the remote branch into your local branch. |
| git branch -r | Lists all remote branches. |
| git checkout -b new-branch origin/remote-branch | Creates a local branch to track a remote branch. |
| git show commit-hash | Shows changes made in a specific commit. |
| git remote -v | Shows the URL of the remote repository. |