Pra 2nd Exploring git commands

A. Create a file demo-repo on git hub

```
Git clone <link>
Cd demo-repo
Echo “Hello.Git” > hello.txt
Git status
Git add hello.txt
Git commit -m “Added hello file”
Git push origin main
```

B. Branching & Merging

```
Git checkout -b feature1
Echo “Feature branch work” >> hello.txt
Git add .
Git commit -m “Updated hello file from feature1”
Git push origin feature1
```

C. Merge to Main Branch

```
Git checkout main
Git merge feature1
Git add .
Git commit -m “Resolved merge conflict”
Git push origin main
```

Pra 3rd Implement Github Operation using Git

```
Git clone url

Cd repo-name

Git checkout -b new-fearture

Echo “This is a new feature “ >feature.txt

Git status

Git add .

Git commit -m  “Added new feature file”

Git push origin new-feature

Git checkout main

Git pull origin main

Git checkout new-feature

Git merge main

Git add .

Git commit -m “Resolved merge conflicts”
```

Pra 5th Version control tools demonstration

```
Git config –global user.name “ your name”
Git config –global user.email youremail@gmail.com
Mkdir version_control_demo
Cd version_control_demo
Git init
Echo “Hello version control” > demo.txt
Git status
Git add demo.txt
Git commit -m “Initial commit with demo.txt”
Git checkout -b new-feature
Echo “Added new feature” >> demo.txt
Git add demo.txt
Git commit -m “updated demo.txt with feature”
Git checkout main
Git merge new-feature
Git add .
Git commit -m “Resolved merge conflict”
Git log
Git diff
Git branch
```

Pra 6th Gitlab merge request

```
Git clone url
Cd demo-gitlab
Git checkout -b feature-branch
Echo “This is a feature” > feature.txt
Git add .
Git commit -m “Added feature.txt”
Git push origin feature-branch
Git checkout feature-branch
Git fetch
Git merge main
Git push origin –delete feature-branch
```

Pra 7th Build an image for a sample web application using docker file.

- myapp/

```
├── app.py
└── requirements.txt – flask
```

- App.py

```
from flask import Flask
app = Flask(_name_)


@app.route("/")
def home():
    return "Hello from Docker!"


if _name_ == "_main_":
    app.run(host="0.0.0.0", port=5000)
```

- Dockerfile

```
FROM python:3.9-slim
WORKDIR /app
COPY requirements.txt .

RUN pip install -r requirements.txt
COPY . .
```

```
EXPOSE 5000
CMD ["python", "app.py"]
```

- cd myapp

- docker build -t myflaskapp .

- docker run -d -p 5000:5000 myflaskapp

- http://localhost:5000