

# *Documentation*

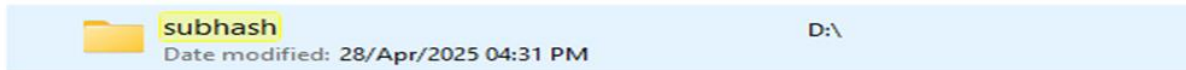
**Name : D-SUBHASH**

**Roll No: 234340**

## The Process

Git hub Django → Docker container → docker hub

### 1. Create a new folder:



- Go to file explorer
- Select the location to create a new file
- Click on new
- Give a name to the file

### 2. open git bash and check directory:

- Changing of directory to where the file is located

```
6LABSYSTEM48@6LABSYSTEM48 MINGW64 /d/subhash
$ cd d:
```

- Changing of directory to the file babi

```
6LABSYSTEM48@6LABSYSTEM48 MINGW64 /d
$ cd subhash
```

### 3. Cloning of Git:

- It is used to copy local copy to repository
- Git clone is the command in Git version
- SYNTAX:

```
git clone https://github.com/username/repository.git
```

#### EXAMPLE:

```
6LABSYSTEM48@6LABSYSTEM48 MINGW64 /d/subhash
$ git clone https://github.com/rajasoxit/batch02-django-app.git
fatal: destination path 'batch02-django-app' already exists and is not an empty directory.
```

### 4. Changing directory:

- Changing of directory to the file located



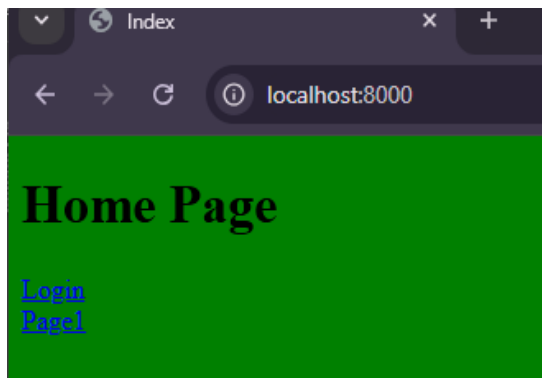
- Docker run is used in Docker file to execute and configure the docker image

```
6LABSYSTEM48@6LABSYSTEM48 MINGW64 /d/subhash/batch02-django-app (main)
$ docker run -p 8000:8000 batch02-django-app
watching for file changes with StatReloader
```

**EXAMPLE:** Docker run -p 8000:8000 django-app

### **9.Opening of browser:**

- Open the browser <https://localhost:8000/> to see the image the was developed



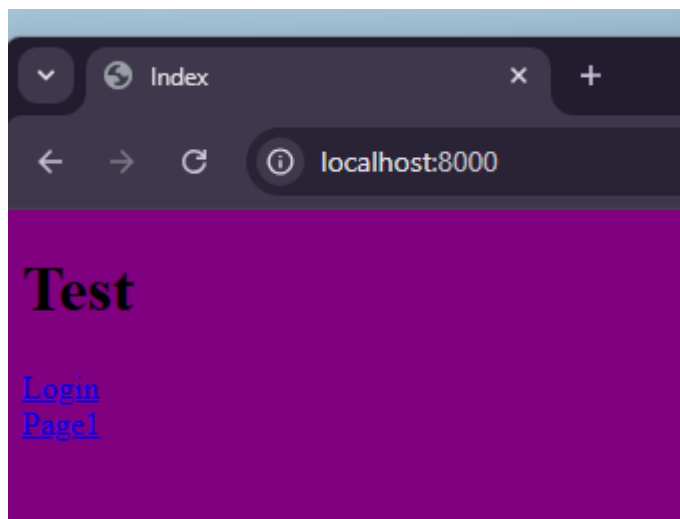
### **10.Visual Studio Code:**

- Open VS code
- Open the with latest created folder
- Go to templates
- Select index.html
- It shows the code that we previously coded

- Changing the “Home Page” to “TEST” and bgcolor “Green” to “purple”

### **11.open browser:**

- To see the changed file



### **12.Again build:**

- Build the changed image

- ```

6LABSYSTEM48@6LABSYSTEM48 MINGW64 /d/subhash/batch02-django-app (main)
$ docker build -t batch02-django-app .
[+] Building 31s (0/10) FINISHED
=> [internal] load build definition from Dockerfile
=> [internal] load metadata for docker.io/library/python:3.10-slim
=> [internal] load .dockerignore
=> [internal] transfer context: 28
=> [1/4] FROM docker.io/library/python:3.10-slim@sha256:3703888f4a219a17fcff5ace7ba3201085f4b3117d6b3b0c5826404e6d4f
=> [internal] load build context
=> [internal] load metadata for docker.io/library/python:3.10-slim@sha256:3703888f4a219a17fcff5ace7ba3201085f4b3117d6b3b0c5826404e6d4f
=> [internal] load build context
=> [internal] transfer context: 3.90kB
=> CACHED [1/4] WORKDIR /app
=> CACHED [2/4] RUN pip install --upgrade pip && pip install django djangorestframework gunicorn
=> CACHED [3/4] COPY . .
=> exporting to image
=> exporting layers
=> exporting manifest sha256:04c18b2d595329884a0728f668432b9944c9530cf4e6d8cc870132cd7f455
=> exporting config sha256:6a6f511839fbfbba9170c818ba42ba1d0fa1d196f6979dc11d8134f2
=> exporting attestation manifest sha256:487770ef64827b274c7f37dc9fa2f883468f62a071837d89e88af480c96
=> exporting manifest list sha256:aff76cd7702a4a070dc022ed27979888414a6d8d39370447400d9f9a9011
=> naming to docker.io/library/batch02-django-app:latest
=> unpacking to docker.io/library/batch02-django-app:latest

```
- EXAMPLE: docker build -t Django-app

### 13.Again run:

- Run the image that we build previously

```

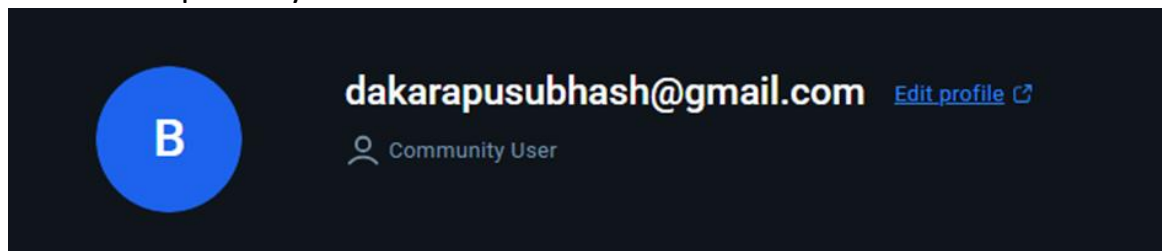
6LABSYSTEM48@6LABSYSTEM48 MINGW64 /d/subhash/batch02-django-app (main)
$ docker run -p 8000:8000 batch02-django-app
watching for file changes with StatReloader

```

EXAMPLE: docker run -p 8000:8000 django-app

### 14.Docker hub:

- Create a docker hub account
- Login to docker hub
- Create a Repository in the docker hub



EXAMPLE: batch02-django-app

### 15.Docker tag:

- It is used to create a tag to the docker image

SYNTAX:

```
docker tag SOURCE_IMAGE[:TAG] TARGET_IMAGE[:TAG]
```

- Source Image : the image that we have to create
- Target Image: the image for we have taken

```

6LABSYSTEM48@6LABSYSTEM48 MINGW64 /d/subhash/batch02-django-app (main)
$ docker tag batch02-django-app babibabi/babi

```

EXAMPLE: docker tag django-app rajasoxit/batch02-django-app

### 16.Docker Login:

- It is used to login into the docker
- It authenticating users to docker registers
- Pushing and pulling purpose

```
6LABSYSTEM48@6LABSYSTEM48 MINGW64 /d/subhash/batch02-django-app (main)
$ docker login
Authenticating with existing credentials... [Username: babibabi]

Info → To login with a different account, run 'docker logout' followed by 'docker login'

Login Succeeded
```

- Example: docker login

### **17.Docker Push:**

- It is used to push the image to a registry
- Deploy them to different environments or to the others

Syntax: docker push NAME[:TAG]

EXAMPLE: docker push subhash/batch02-django-app

```
6LABSYSTEM48@6LABSYSTEM48 MINGW64 /d/subhash/batch02-django-app (main)
$ docker push babibabi/babi
Using default tag: latest
The push refers to repository [docker.io/babibabi/babi]
765ef9c81879: Layer already exists
254e724d7786: Layer already exists
7a89cd18298a: Layer already exists
92e5f9e5de64: Layer already exists
d15668d6a1de: Layer already exists
a8043e92bcd8: Layer already exists
713e863ba87c: Pushed
0316df7fdf38: Pushed
latest: digest: sha256:2abf893ebbf2d82b2ed2e90a65b74bed463fc47b5c5ab5f137c26b7d3498401 size: 856
```