

JOYSON SAMUEL P

JAYAPRAKASH P

JAYASIVAA KRISHNAA B

SHAIKH FIRAAS A

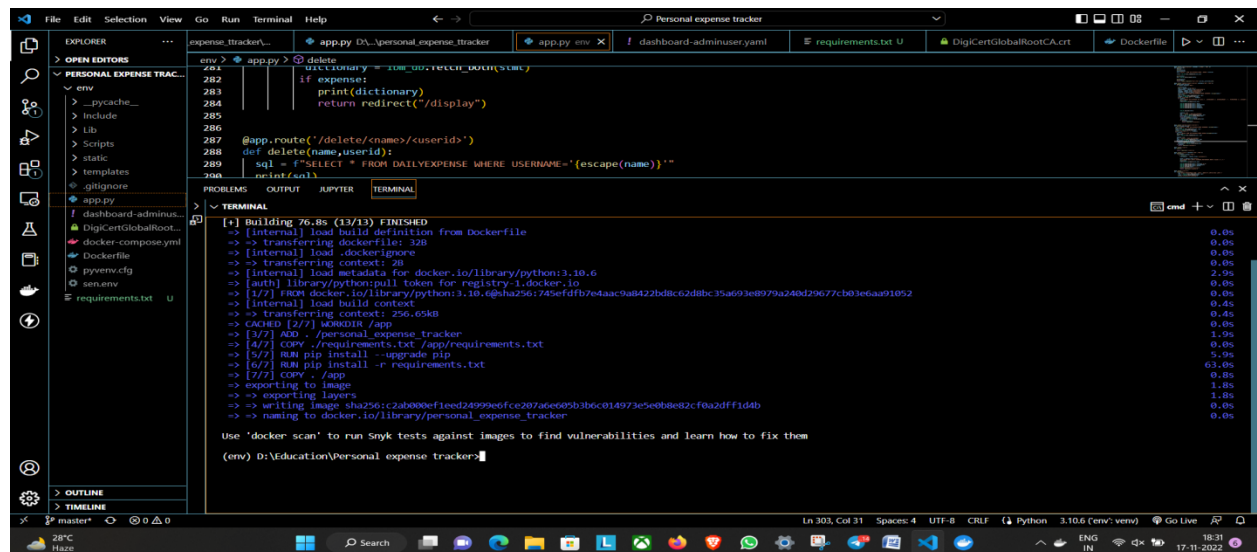
SPRINT-4

SCREEN SHOTS

1) DEPLOYING FLASK APPLICATION TO DOCKER CONTAINER

Step-1 -BUILD A DOCKER IMAGE

```
(env) D:\Education\Personal expense tracker>docker build -f env/Dockerfile -t personal_expense_tracker .
```



The screenshot shows a VS Code editor with a project named 'Personal expense tracker'. The Explorer pane on the left shows the project structure, including 'env' (Dockerfile, requirements.txt, .env), 'static', 'templates', and 'app.py'. The main editor shows the 'app.py' file with a Flask application. The terminal pane at the bottom shows the output of the 'docker build' command, which successfully builds the 'personal_expense_tracker' image. The output includes details about the build context, the Dockerfile, and the resulting image SHA256 hash.

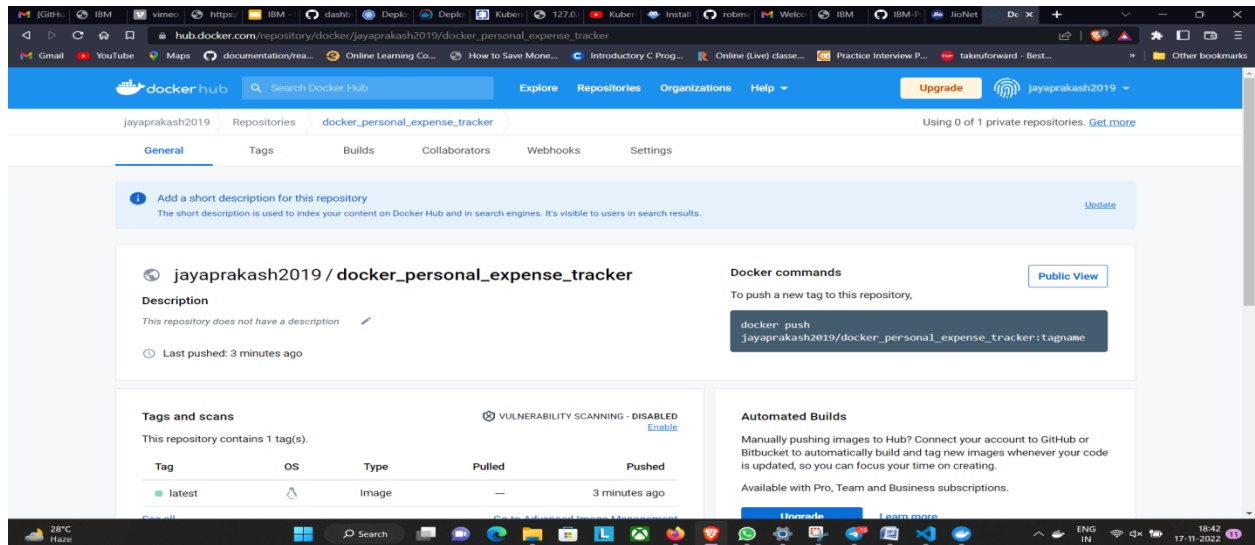
```
[+] Building 76.8s (13/13) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 32B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/python:3.10.6
=> [auth] library/python:pull token for registry-1.docker.io
=> [1/7] FROM docker.io/library/python:3.10.6@sha256:745efdfb7e4aac9a8422b8bc62d8bc35a693e8979a240d29677cbb3e6aa91052
=> [internal] load build context
=> => transferring context: 256.65kB
=> CACHED [2/7] WORKDIR /app
=> [3/7] ADD . /personal_expense_tracker
=> [4/7] COPY ./requirements.txt /app/requirements.txt
=> [5/7] RUN pip install --upgrade pip
=> [6/7] RUN pip install -r requirements.txt
=> [7/7] COPY . /app
=> exporting to image
=> => writing image sha256:c2ab000ef1eed24999e0fce207a6e05b3b6c014973e5e0b8e82cfaa2dff1d4b
=> => naming to docker.io/library/personal_expense_tracker

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
(env) D:\Education\Personal expense tracker>
```

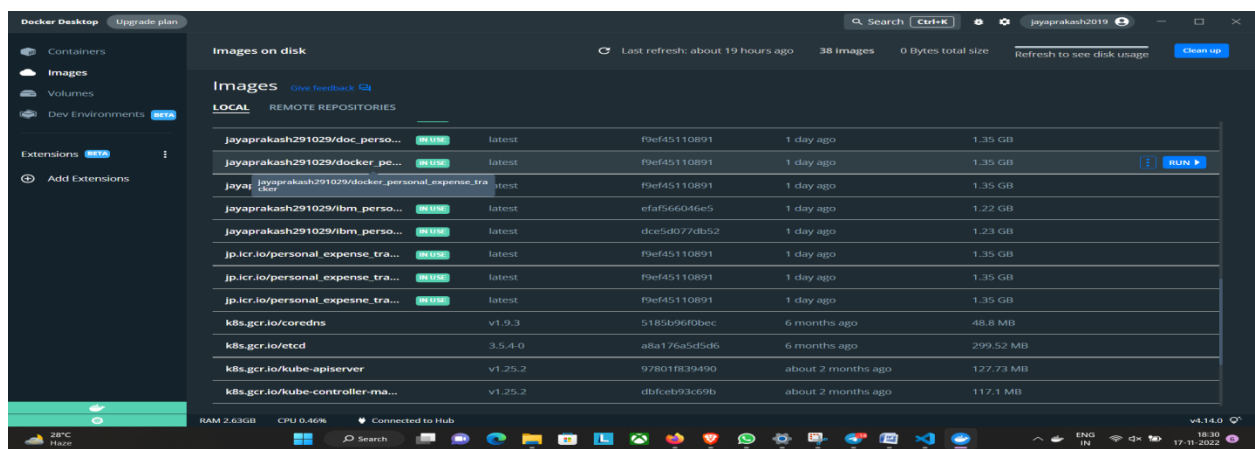
STEP-2 TAG THE DOCKER IMAGE

```
(env) D:\Education\Personal expense tracker>docker tag personal_expense_tracker:latest jayaprakash291029/docker_personal_expense_tracker:latest
```

STEP-3 PUSH TO DOCKER HUB



```
(env) D:\Education\Personal expense tracker>docker push jayaprakash2019/docker_personal_expense_tracker:latest
The push refers to repository [docker.io/jayaprakash2019/docker_personal_expense_tracker]
3655c00592af: Layer already exists
4e39d5163298: Layer already exists
4ce332575e93: Layer already exists
3ce39c0f2f61: Layer already exists
28dd888b77b5: Layer already exists
42cbd32d79aa: Layer already exists
bfc1deb8136e: Layer already exists
1f123186824c: Layer already exists
3d6eb1152931: Layer already exists
100796cdf3b1: Layer already exists
54acb5a6fa0b: Layer already exists
8d51c618126f: Layer already exists
9ff6e4d46744: Layer already exists
a89d1d47b5a1: Layer already exists
655ed1b7a428: Layer already exists
latest: digest: sha256:cfb52b015825dfdea48b8f0b87f7a7246f3ea98e72d4a2e877c048138113a8f1 size: 3479
```



STEP-4 RUN THE DOCKER IMAGE

```
(env) D:\Education\Personal expense tracker>docker run -p 3001:3000 jayaprakash291029/docker_personal_expense_tracker
```

```
<ibm_db.IBM_DBConnection object at 0x000001CAD86C6EF0>
connection successful...
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:3000
* Running on http://100.127.133.202:3000
Press CTRL+C to quit
* Restarting with stat
<ibm_db.IBM_DBConnection object at 0x0000025B7FC43E70>
connection successful...
* Debugger is active!
* Debugger PIN: 102-049-649
```