Q1) Pull any image from the docker hub, create its container, and execute it showing the output.

```
E:\VASEEM>docker pull alpine
Using default tag: latest
latest: Pulling from library/alpine
63b65145d645: Pull complete
Digest: sha256:69665d02cb32192e52e07644d76bc6f25abeb5410edc1c7a81a10ba3f0efb90a
Status: Downloaded newer image for alpine:latest
docker.io/library/alpine:latest

E:\VASEEM>
```

```
E:\VASEEM>docker pull alpine
Using default tag: latest
latest: Pulling from library/alpine
63b65145d645: Pull complete
Digest: sha256:69665d02cb32192e52e07644d76bc6f25abeb5410edc1c7a81a10ba3f0efb90a
Status: Downloaded newer image for alpine:latest
docker.io/library/alpine:latest

E:\VASEEM>docker run -it alpine
/ # ls
bin    dev    etc    home   lib    media  mnt    opt    proc   root   run    sbin   srv    sys    tmp    usr    var
/ # vi Hello-alpine
/ # cat Hello-alpine
Pulled by Docker Desktop
/ #
```

Q2) Create the basic java application, generate its image with necessary files, and execute it with docker.

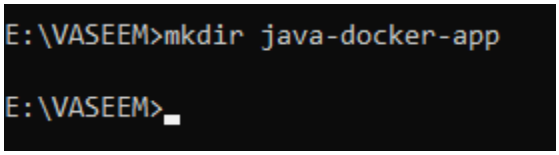
To create a java-app first I have created a folder named java-docker-app and I have added the files Hello.java and Dockerfile

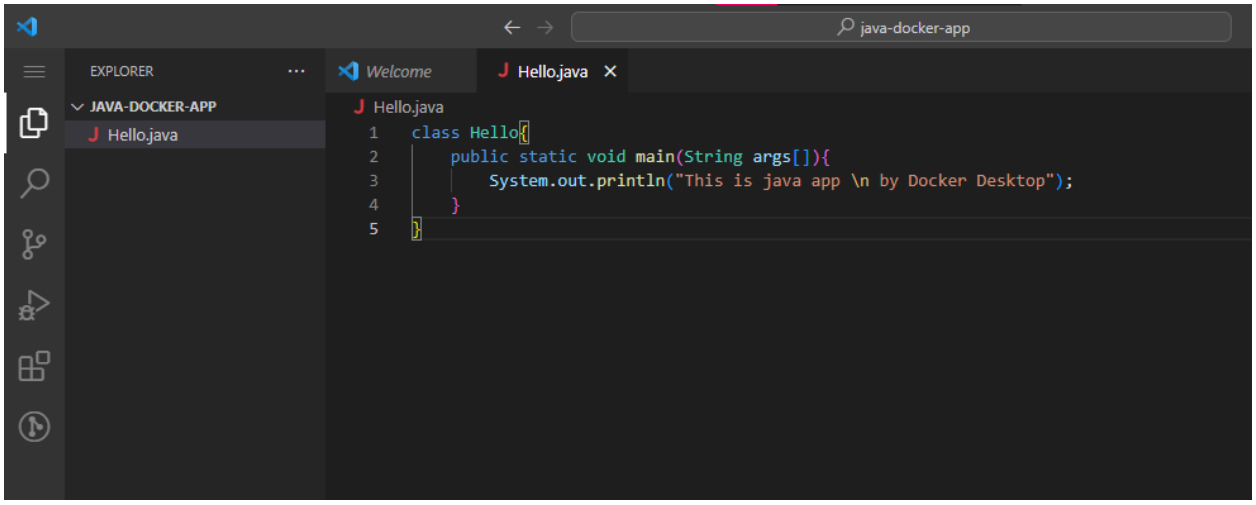
Hello.java contains the content that will display when we run the java-app application and the Dockerfile contains the commands that should be carried out while creating the java-app image. To create the java-app image the following command is used

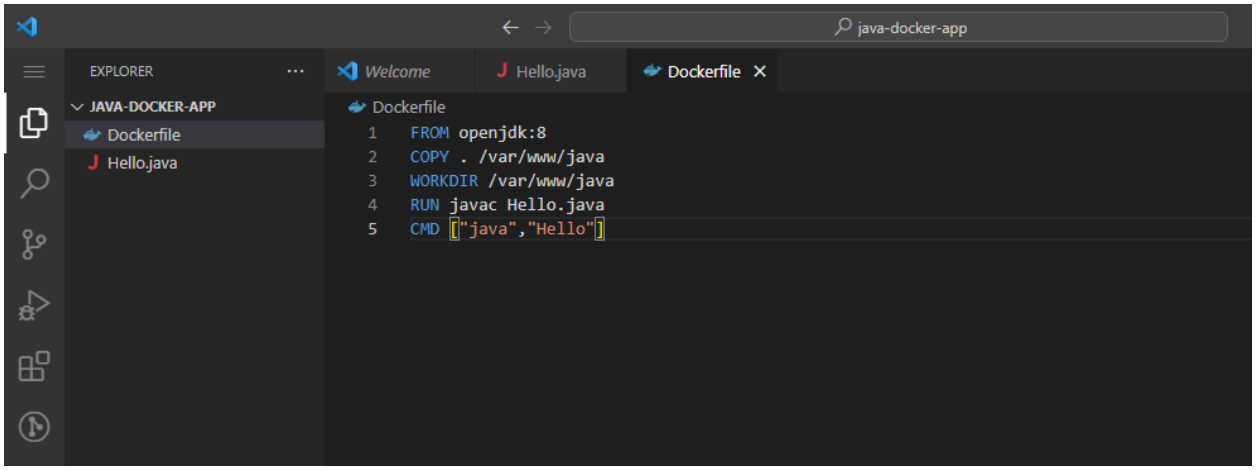
docker build –t java-app .

At last, to run the java-app I have executed the command

docker run java-app








```
E:\VASEEM>cd java-docker-app

E:\VASEEM\java-docker-app>
```

```
E:\VASEEM\java-docker-app>docker build -t java-app .
[+] Building 9.8s (10/10) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 31B
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load metadata for docker.io/library/openjdk:8
 => [auth] library/openjdk:pull token for registry-1.docker.io
 => [1/4] FROM docker.io/library/openjdk:8@sha256:86e863cc57215cfb181bd319736d0baf625fe8f150577f9eb58bd937f5452cb8
 => [internal] load build context
 => => transferring context: 61B
 => CACHED [2/4] COPY . /var/www/java
 => CACHED [3/4] WORKDIR /var/www/java
 => CACHED [4/4] RUN javac Hello.java
 => exporting to image
 => => exporting layers
 => => writing image sha256:b7edef4dad26399afcdd4649f4e638cf6a5c430b36116f2e801d3c1b3dfeffda
 => => naming to docker.io/library/java-app

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
```

```
E:\VASEEM\java-docker-app>docker run java-app
This is java app
 by Docker Desktop

E:\VASEEM\java-docker-app>
```