ADDA LABFAT
Q) DOCKER

📄 Step 1: Create the Java File

HelloWorld.java:

```
public class HelloWorld {
  public static void main(String[] args) {
    System.out.println("Hello from Dockerized Java!");
    System.out.println("Containerization successful 🚀 ");
  }
}
```

📁 Folder Structure

```
java-docker-app/
├── Dockerfile
└── HelloWorld.java
```

Dockerfile :

```
# Use OpenJDK image
FROM openjdk:17

# Set working directory inside container
WORKDIR /app

# Copy source file to container
COPY HelloWorld.java .

# Compile the Java file
RUN javac HelloWorld.java
```

Command to run the Java program

CMD ["java", "HelloWorld"]

Step 3: Build and Run the Docker Image

Run these commands from your terminal inside the java-docker-app folder:

bash

CopyEdit

Build the image

docker build -t java-hello:v1 .

Run the container

docker run --name myjavaapp java-hello:v1

Expected output:

csharp

CopyEdit

Hello from Dockerized Java!

Containerization successful 🚀

1)GITHUB ACTIONS:
YAML WORKFLOW FILE:
EVENTS, JOBS,RUNNERS,STEPS AND ACTION
We are checking out the code and running superlinter against it

name: Super-Linter

on: push

jobs:

```
 super-lint:
  name: Lint code base
  runs-on: ubuntu-latest
  steps:
   - name: Checkout code
     uses: actions/checkout@v2

   - name: Run Super-Linter
     uses: github/super-linter@v3
     env:
      DEFAULT_BRANCH: main
      GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
```

Wrong code :

```
import os,sys


def my_function():
  print("This is a test function")
  x=1
  y =  2
  print(x+y )
my_function( )
```

Correct code :

```
import os
import sys


def my_function():
  print("This is a test function")
  x = 1
  y = 2
```

```
    print(x + y)


my_function()
```

2nd wrong code :

```
def hello():
    print("hi")


def bye():
    print("bye")


print(hello())
```

correct code :