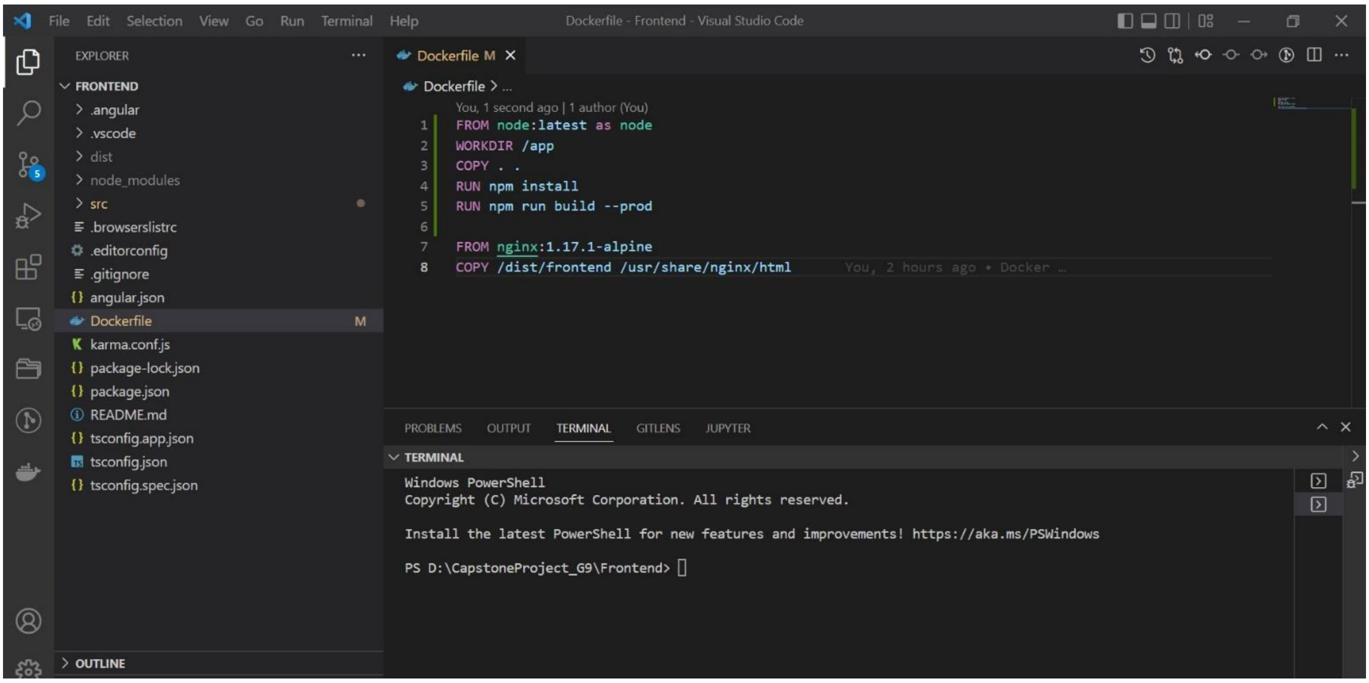
Google Cloud

Frontend (Angular) Docker Image

1. Open your Frontend folder in Visual Studio Code, make a docker file as “Dockerfile” out of your src directory, and edit it as follows.



```
FROM node:latest as node
WORKDIR /app
COPY . .
RUN npm install
RUN npm run build --prod

FROM nginx:1.17.1-alpine
COPY /dist/frontend /usr/share/nginx/html
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\CapstoneProject_G9\Frontend>
```

2. Now push the changes done in your frontend folder to your git repository (or Refer below git command).

git init

git remote add origin “Repository URL”

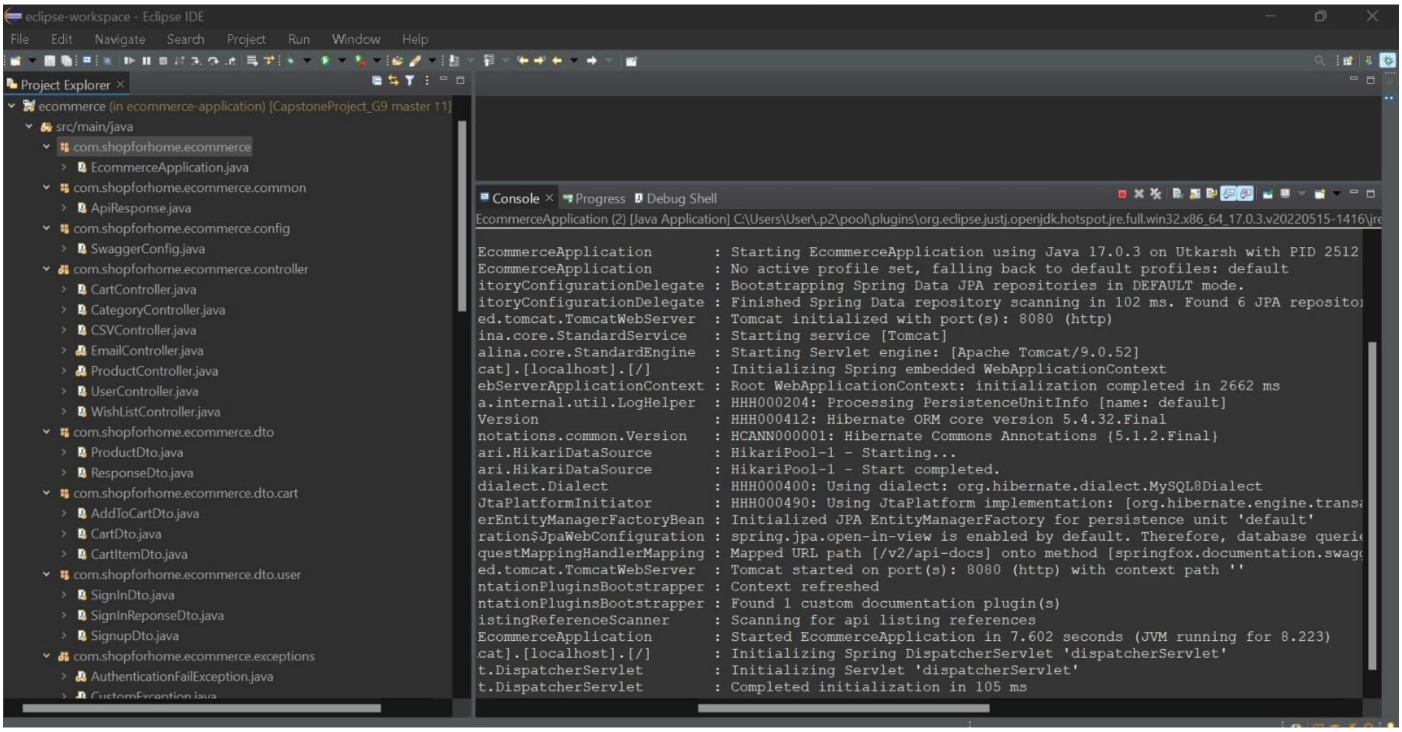
git add .

git commit -m “Docker”

git push -u origin master.

Backend (Spring Boot)

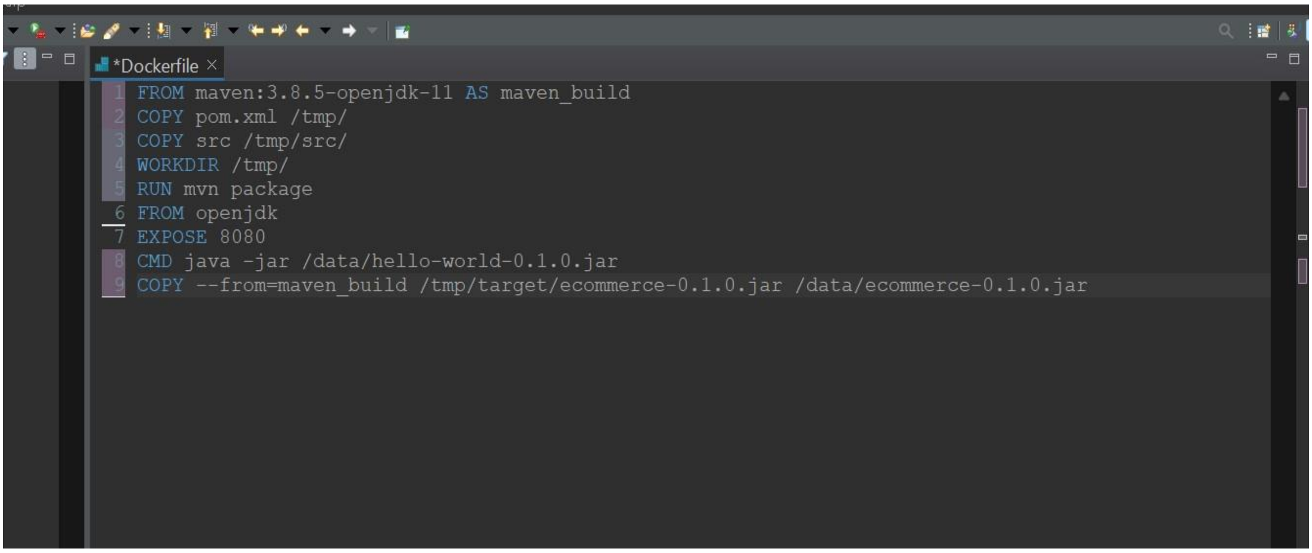
1. Now, open the ecommerce-application on your Eclipse IDE.


eclipse-workspace - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Project Explorer
ecommerce [in ecommerce-application] [CapstoneProject_G9 master ↑1]
src/main/java
com.shopforhome.ecommerce
EcommerceApplication.java
com.shopforhome.ecommerce.common
ApiResponse.java
com.shopforhome.ecommerce.config
SwaggerConfig.java
com.shopforhome.ecommerce.controller
CartController.java
CategoryController.java
CSVController.java
EmailController.java
ProductController.java
UserController.java
WishListController.java
com.shopforhome.ecommerce.dto
ProductDto.java
ResponseDto.java
com.shopforhome.ecommerce.dto.cart
AddToCartDto.java
CartDto.java
CartItemDto.java
com.shopforhome.ecommerce.dto.user
SignInDto.java
SignInReponseDto.java
SignupDto.java
com.shopforhome.ecommerce.exceptions
AuthenticationFailException.java
CustomException.java
Console
Progress
Debug Shell

EcommerceApplication (2) [Java Application] C:\Users\User\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.3.v20220515-1416\jre

```
EcommerceApplication       : Starting EcommerceApplication using Java 17.0.3 on Utkarsh with PID 2512
EcommerceApplication       : No active profile set, falling back to default profiles: default
itoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
itoryConfigurationDelegate : Finished Spring Data repository scanning in 102 ms. Found 6 JPA reposito
ed.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 8080 (http)
ina.core.StandardService   : Starting service [Tomcat]
alina.core.StandardEngine  : Starting Servlet engine: [Apache Tomcat/9.0.52]
cat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
ebServerApplicationContext : Root WebApplicationContext: initialization completed in 2662 ms
a.internal.util.LogHelper  : HHH000204: Processing PersistenceUnitInfo [name: default]
Version                    : HHH000412: Hibernate ORM core version 5.4.32.Final
notations.common.Version   : HCANN000001: Hibernate Commons Annotations {5.1.2.Final}
ari.HikariDataSource       : HikariPool-1 - Starting...
ari.HikariDataSource       : HikariPool-1 - Start completed.
dialect.Dialect            : HHH000400: Using dialect: org.hibernate.dialect.MySQL8Dialect
JtaPlatformInitiator       : HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transa
erEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
ration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database querie
questMappingHandlerMapping : Mapped URL path [/v2/api-docs] onto method [springfox.documentation.swagg
ed.tomcat.TomcatWebServer  : Tomcat started on port(s): 8080 (http) with context path ''
ntationPluginsBootstrapper : Context refreshed
ntationPluginsBootstrapper : Found 1 custom documentation plugin(s)
istingReferenceScanner     : Scanning for api listing references
EcommerceApplication       : Started EcommerceApplication in 7.602 seconds (JVM running for 8.223)
cat].[localhost].[/]       : Initializing Spring DispatcherServlet 'dispatcherServlet'
t.DispatcherServlet        : Initializing Servlet 'dispatcherServlet'
t.DispatcherServlet        : Completed initialization in 105 ms
```

2. Create a new file from the src/main/java and name it as “Dockerfile” and edit the file as given below.

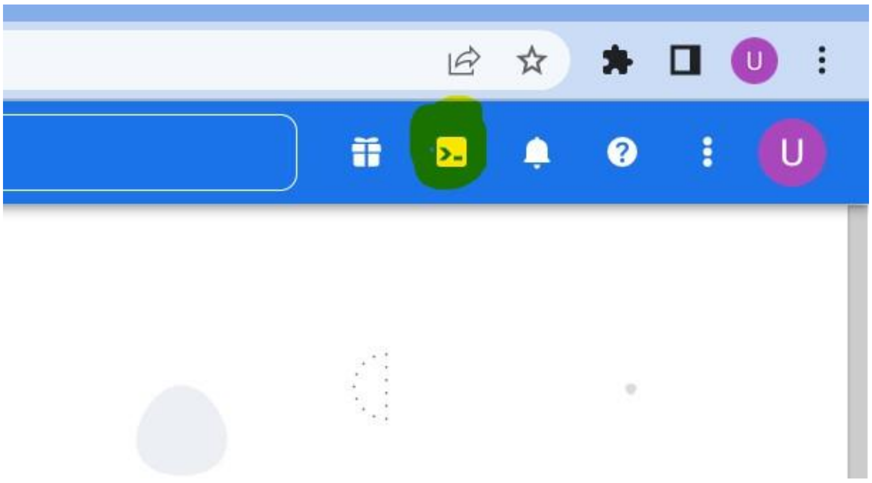


```
FROM maven:3.8.5-openjdk-11 AS maven_build
COPY pom.xml /tmp/
COPY src /tmp/src/
WORKDIR /tmp/
RUN mvn package
FROM openjdk
EXPOSE 8080
CMD java -jar /data/hello-world-0.1.0.jar
COPY --from=maven_build /tmp/target/ecommerce-0.1.0.jar /data/ecommerce-0.1.0.jar
```

3. Again save the changes, and push the code to your git hub repository.

Google Cloud console

1. Login to GCP console and open Cloud Shell.



2. Create a directory for the your application

```
mkdir ecommerce
```

3. Switch to the directory

```
cd ecommerce
```

4. Clone the git repository

```
git clone “your repository”
```

5. Switch to the Frontend folder

```
cd G9_Capstone_project/Frontend
```

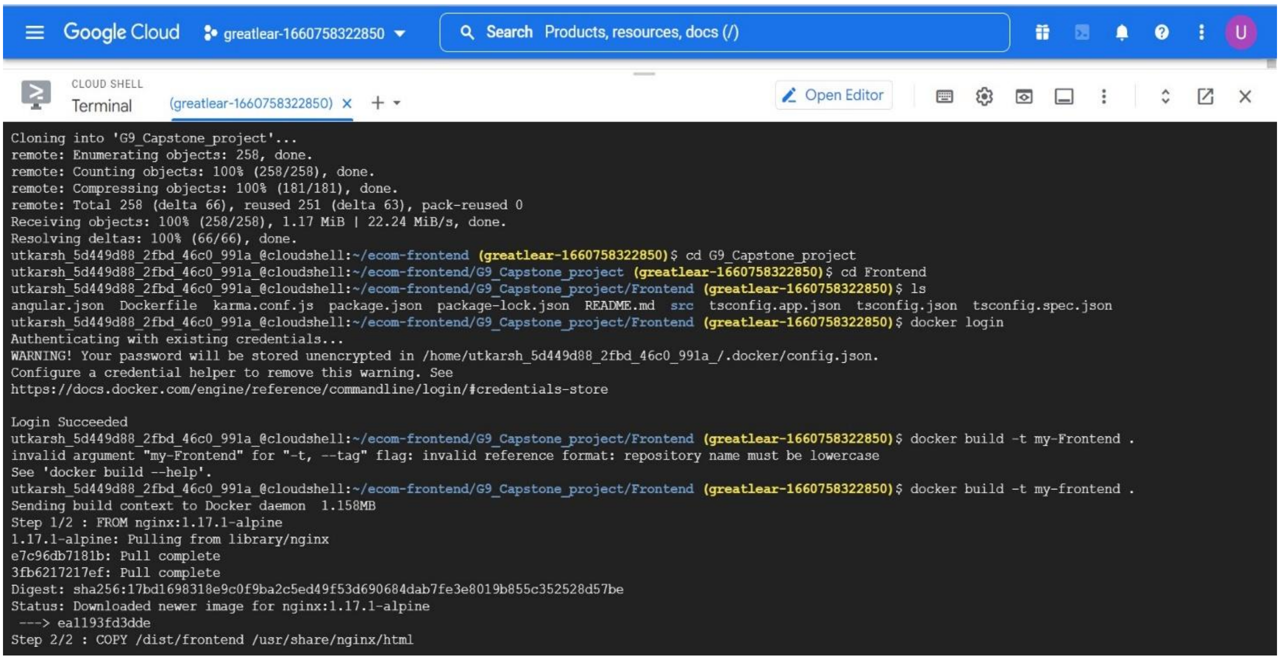
6. Login to the docker account

```
docker login
```

7. Build the docker image

docker build –pull –rm -f “Dockerfile” -t ecommerce/frontend:v1 .

8. Authorize the Cloud shell and your frontend will start on port 4200



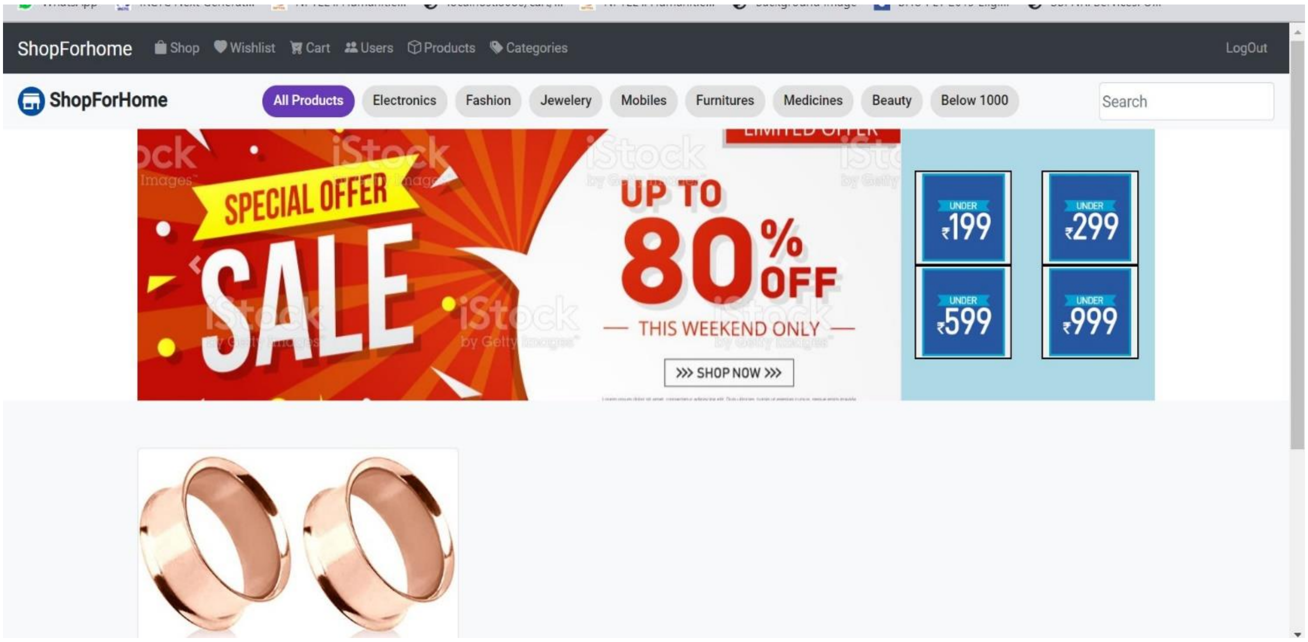
```
Cloning into 'G9_Capstone_project'...
remote: Enumerating objects: 258, done.
remote: Counting objects: 100% (258/258), done.
remote: Compressing objects: 100% (181/181), done.
remote: Total 258 (delta 66), reused 251 (delta 63), pack-reused 0
Receiving objects: 100% (258/258), 1.17 MiB | 22.24 MiB/s, done.
Resolving deltas: 100% (66/66), done.
utkarsh_5d449d88_2fbd_46c0_991a_@cloudshell:~/ecom-frontend (greatlear-1660758322850)$ cd G9_Capstone_project
utkarsh_5d449d88_2fbd_46c0_991a_@cloudshell:~/ecom-frontend/G9_Capstone_project (greatlear-1660758322850)$ cd Frontend
utkarsh_5d449d88_2fbd_46c0_991a_@cloudshell:~/ecom-frontend/G9_Capstone_project/Frontend (greatlear-1660758322850)$ ls
angular.json  Dockerfile  karma.conf.js  package.json  package-lock.json  README.md  src  tsconfig.app.json  tsconfig.json  tsconfig.spec.json
utkarsh_5d449d88_2fbd_46c0_991a_@cloudshell:~/ecom-frontend/G9_Capstone_project/Frontend (greatlear-1660758322850)$ docker login
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /home/utkarsh_5d449d88_2fbd_46c0_991a_/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
utkarsh_5d449d88_2fbd_46c0_991a_@cloudshell:~/ecom-frontend/G9_Capstone_project/Frontend (greatlear-1660758322850)$ docker build -t my-Frontend .
invalid argument "my-Frontend" for "-t, --tag" flag: invalid reference format: repository name must be lowercase
See 'docker build --help'.
utkarsh_5d449d88_2fbd_46c0_991a_@cloudshell:~/ecom-frontend/G9_Capstone_project/Frontend (greatlear-1660758322850)$ docker build -t my-frontend .
Sending build context to Docker daemon  1.158MB
Step 1/2 : FROM nginx:1.17.1-alpine
1.17.1-alpine: Pulling from library/nginx
e7c96db7181b: Pull complete
3fb6217217ef: Pull complete
Digest: sha256:17bd1698318e9c0f9ba2c5ed49f53d690684dab7fe3e8019b855c352528d57be
Status: Downloaded newer image for nginx:1.17.1-alpine
 ---> ea1193fd3dde
Step 2/2 : COPY /dist/frontend /usr/share/nginx/html
```

9. Now run the docker

docker run –rm -d -p 4200:4200/tcp ecommerce/frontend:v1

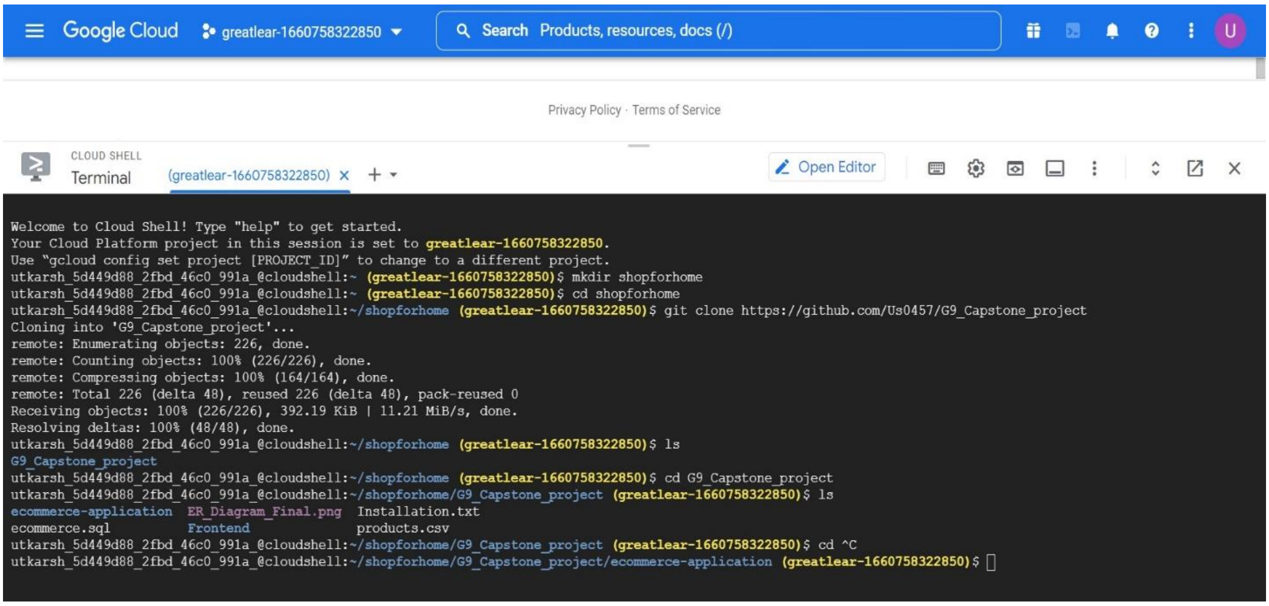
10. Find the web preview button and click on Preview on 4200



11. Now move to the **ecommerce-application** folder

cd . .

cd ecommerce-application

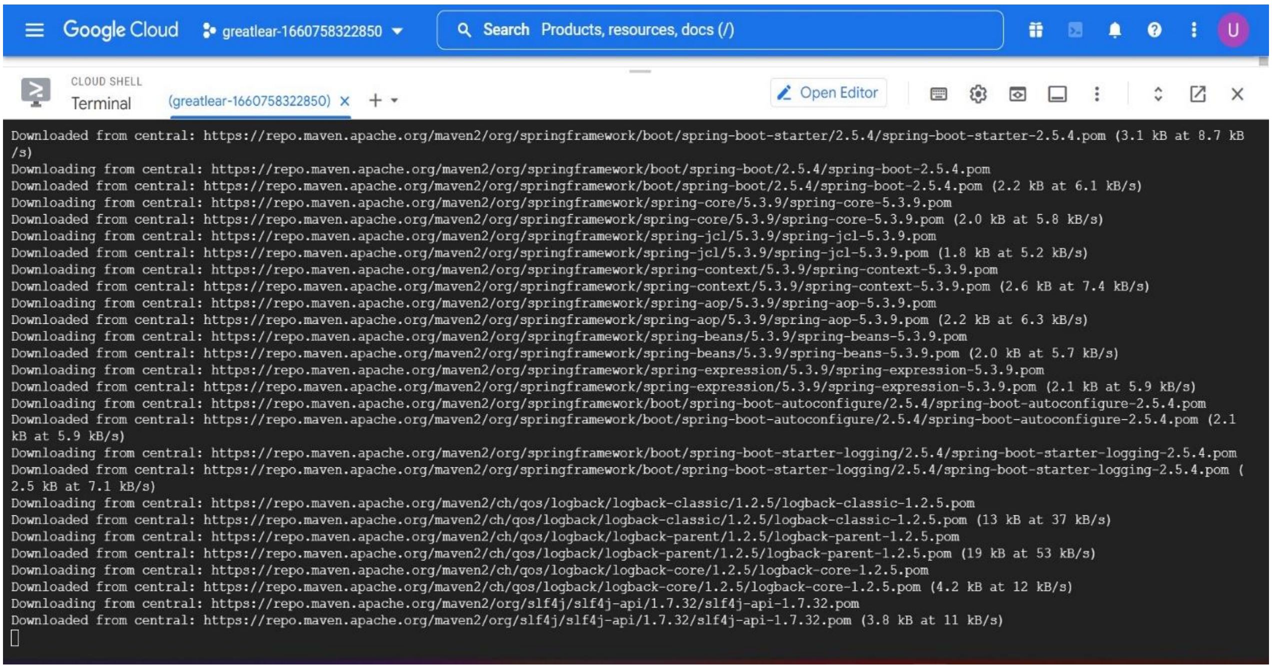


Google Cloud
greatlear-1660758322850
Search Products, resources, docs (/)
Privacy Policy · Terms of Service
CLOUD SHELL
Terminal
(greatlear-1660758322850)
Open Editor

```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to greatlear-1660758322850.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
utkarsh_5d449d88_2fbd_46c0_991a_@cloudshell:~ (greatlear-1660758322850)$ mkdir shopforhome
utkarsh_5d449d88_2fbd_46c0_991a_@cloudshell:~ (greatlear-1660758322850)$ cd shopforhome
utkarsh_5d449d88_2fbd_46c0_991a_@cloudshell:~/shopforhome (greatlear-1660758322850)$ git clone https://github.com/Us0457/G9_Capstone_project
Cloning into 'G9_Capstone_project'...
remote: Enumerating objects: 226, done.
remote: Counting objects: 100% (226/226), done.
remote: Compressing objects: 100% (164/164), done.
remote: Total 226 (delta 48), reused 226 (delta 48), pack-reused 0
Receiving objects: 100% (226/226), 392.19 KiB | 11.21 MiB/s, done.
Resolving deltas: 100% (48/48), done.
utkarsh_5d449d88_2fbd_46c0_991a_@cloudshell:~/shopforhome (greatlear-1660758322850)$ ls
G9_Capstone_project
utkarsh_5d449d88_2fbd_46c0_991a_@cloudshell:~/shopforhome (greatlear-1660758322850)$ cd G9_Capstone_project
utkarsh_5d449d88_2fbd_46c0_991a_@cloudshell:~/shopforhome/G9_Capstone_project (greatlear-1660758322850)$ ls
ecommerce-application  ER_Diagram_Final.png  Installation.txt
ecommerce.sql          Frontend              products.csv
utkarsh_5d449d88_2fbd_46c0_991a_@cloudshell:~/shopforhome/G9_Capstone_project (greatlear-1660758322850)$ cd ^C
utkarsh_5d449d88_2fbd_46c0_991a_@cloudshell:~/shopforhome/G9_Capstone_project/ecommerce-application (greatlear-1660758322850)$ 
```

12. Now build the docker image for the spring boot app.

docker build -t ecommerce/backend:v2 .



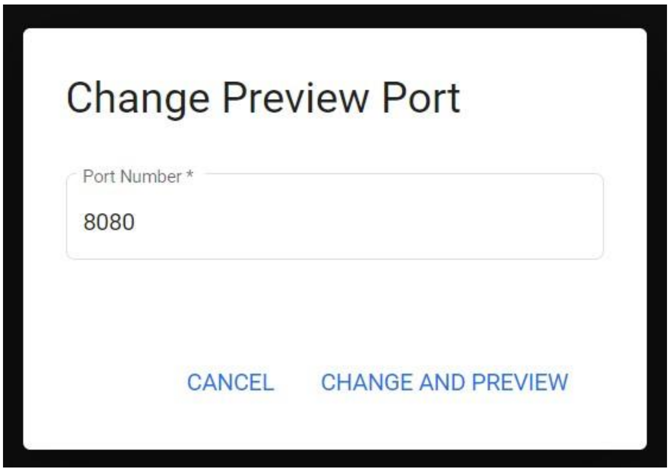
```
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-starter/2.5.4/spring-boot-starter-2.5.4.pom (3.1 kB at 8.7 kB
/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot/2.5.4/spring-boot-2.5.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot/2.5.4/spring-boot-2.5.4.pom (2.2 kB at 6.1 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/spring-core/5.3.9/spring-core-5.3.9.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/spring-core/5.3.9/spring-core-5.3.9.pom (2.0 kB at 5.8 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/spring-jcl/5.3.9/spring-jcl-5.3.9.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/spring-jcl/5.3.9/spring-jcl-5.3.9.pom (1.8 kB at 5.2 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/spring-context/5.3.9/spring-context-5.3.9.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/spring-context/5.3.9/spring-context-5.3.9.pom (2.6 kB at 7.4 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/spring-aop/5.3.9/spring-aop-5.3.9.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/spring-aop/5.3.9/spring-aop-5.3.9.pom (2.2 kB at 6.3 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/spring-beans/5.3.9/spring-beans-5.3.9.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/spring-beans/5.3.9/spring-beans-5.3.9.pom (2.0 kB at 5.7 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/spring-expression/5.3.9/spring-expression-5.3.9.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/spring-expression/5.3.9/spring-expression-5.3.9.pom (2.1 kB at 5.9 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-autoconfigure/2.5.4/spring-boot-autoconfigure-2.5.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-autoconfigure/2.5.4/spring-boot-autoconfigure-2.5.4.pom (2.1
kB at 5.9 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-starter-logging/2.5.4/spring-boot-starter-logging-2.5.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-starter-logging/2.5.4/spring-boot-starter-logging-2.5.4.pom (
2.5 kB at 7.1 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/ch/qos/logback/logback-classic/1.2.5/logback-classic-1.2.5.pom
Downloaded from central: https://repo.maven.apache.org/maven2/ch/qos/logback/logback-classic/1.2.5/logback-classic-1.2.5.pom (13 kB at 37 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/ch/qos/logback/logback-parent/1.2.5/logback-parent-1.2.5.pom
Downloaded from central: https://repo.maven.apache.org/maven2/ch/qos/logback/logback-parent/1.2.5/logback-parent-1.2.5.pom (19 kB at 53 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/ch/qos/logback/logback-core/1.2.5/logback-core-1.2.5.pom
Downloaded from central: https://repo.maven.apache.org/maven2/ch/qos/logback/logback-core/1.2.5/logback-core-1.2.5.pom (4.2 kB at 12 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/slf4j/slf4j-api/1.7.32/slf4j-api-1.7.32.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/slf4j/slf4j-api/1.7.32/slf4j-api-1.7.32.pom (3.8 kB at 11 kB/s)
```

13. Now, again authorize the cloud shell.

14.Run the container using the image.

```
docker run -d -p 4200:8080 ecommerce/backend
```

15. Again open the web preview button and select “Change Preview Port” and enter port 4200. Now, navigate to the frontend ang refresh the webpage.



Change Preview Port

Port Number *

8080