

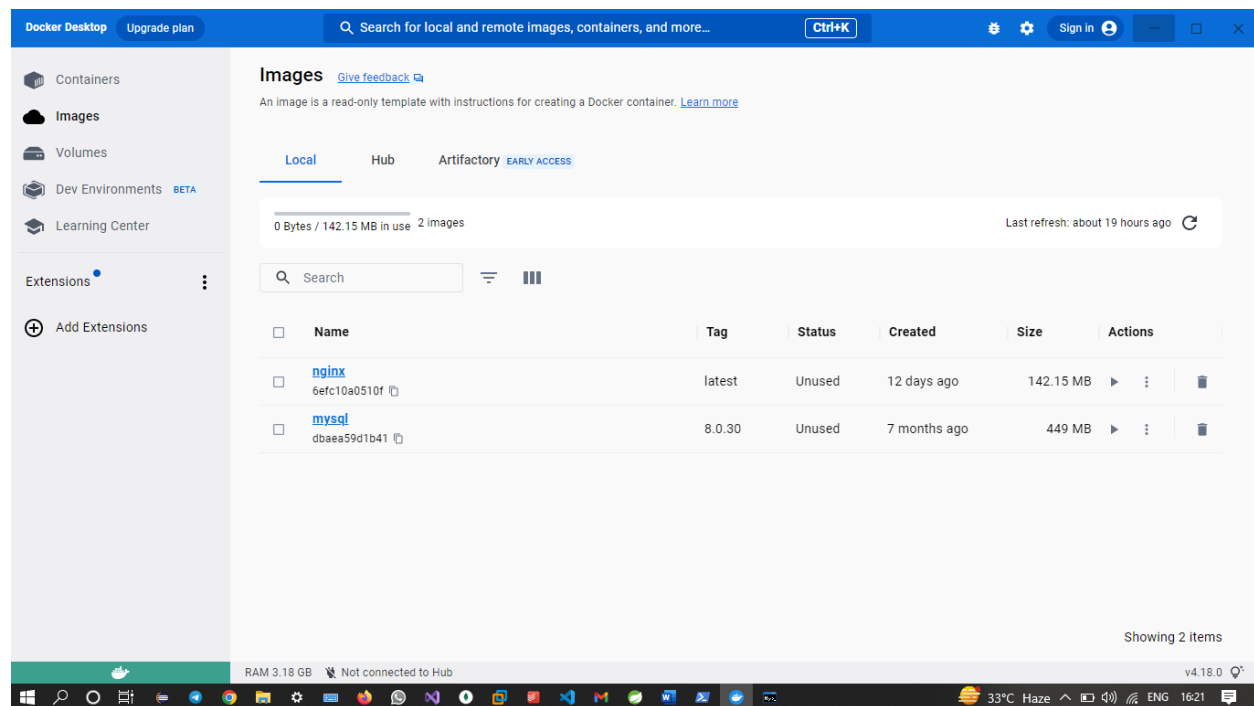
Dockerizing a project using VS-code, Spring Tool Suit and Command prompt.

Download MySQL image

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

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\hits> docker pull mysql:8.0.30
8.0.30: Pulling from library/mysql
295ca2342728: Pull complete
79af4312a7e0: Pull complete
48d3d73d1704: Pull complete
521b8724b397: Pull complete
b08cf36084a14: Pull complete
0115482cc006: Pull complete
a360b08917ea: Pull complete
12deeb3c1323: Pull complete
ee1dc10db1e9: Pull complete
640e40ad29c: Pull complete
1921fe8879a2: Pull complete
Digest: sha256:3c1aab708f6e57fc4dcca36bacc7219acfb4ec450f3fb6d370ea89409e906
Status: Downloaded newer image for mysql:8.0.30
docker.io/library/mysql:8.0.30
PS C:\Users\hits> docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
nginx         latest    6efc10a0510f   12 days ago    142MB
mysql         8.0.30    dbaea59d1b41   6 months ago   449MB
PS C:\Users\hits>
```

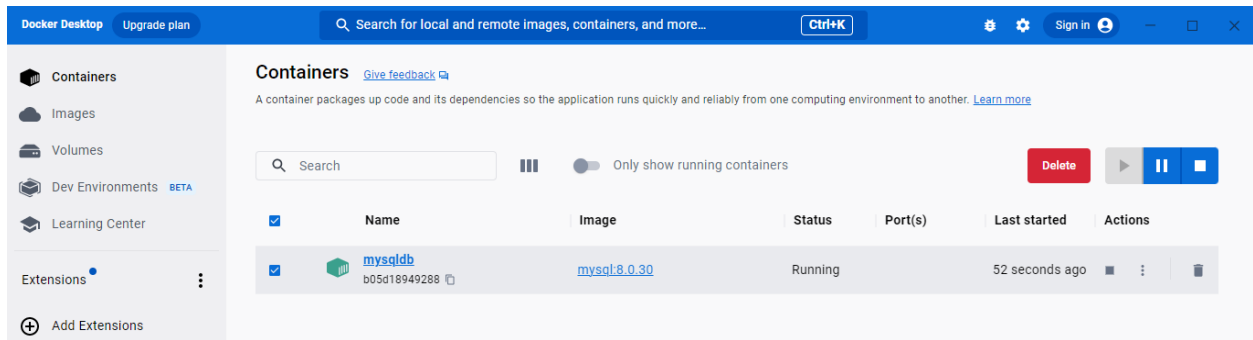


Create docker network

```
PS C:\Users\hits> docker network create springboot-mysql-net
rd7991d6f9a15ee219da531008b7195963b4c12b49c4eef073118ca523cb6333
PS C:\Users\hits> docker network ls
NETWORK ID          NAME                DRIVER             SCOPE
a73967a8686b        bridge              bridge              local
392ce11fa07f        host                host                local
dac6dec9bcc4        none                null                local
rd7991d6f9a1        springboot-mysql-net bridge              local
PS C:\Users\hits>
```

Run mysql image by providing a username and password after this container will be created.:

```
Windows PowerShell
PS C:\Users\hits> docker run --name mysqldb --network springboot-mysql-net -e MYSQL_ROOT_PASSWORD=1234 -e MYSQL_DATABASE=etour -e MYSQL_PASSWORD=1234 -d mysql:8.0.30
b05d18949288363bf53ee20a151da6eeab9f0c6b0bb0b4c4d46476d9bbdba00d
PS C:\Users\hits>
```



Login to ur mysql and switch to database created:

```

Windows PowerShell
PS C:\Users\Hites> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                    NAMES
b05d18949288   mysql:8.0.30  "docker-entrypoint.s..."  2 minutes ago Up 2 minutes    3306/tcp, 33060/tcp    mysqladb

PS C:\Users\Hites> docker exec -it b05 bash

bash-4.4# mysql -uroot -p1234
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.30 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

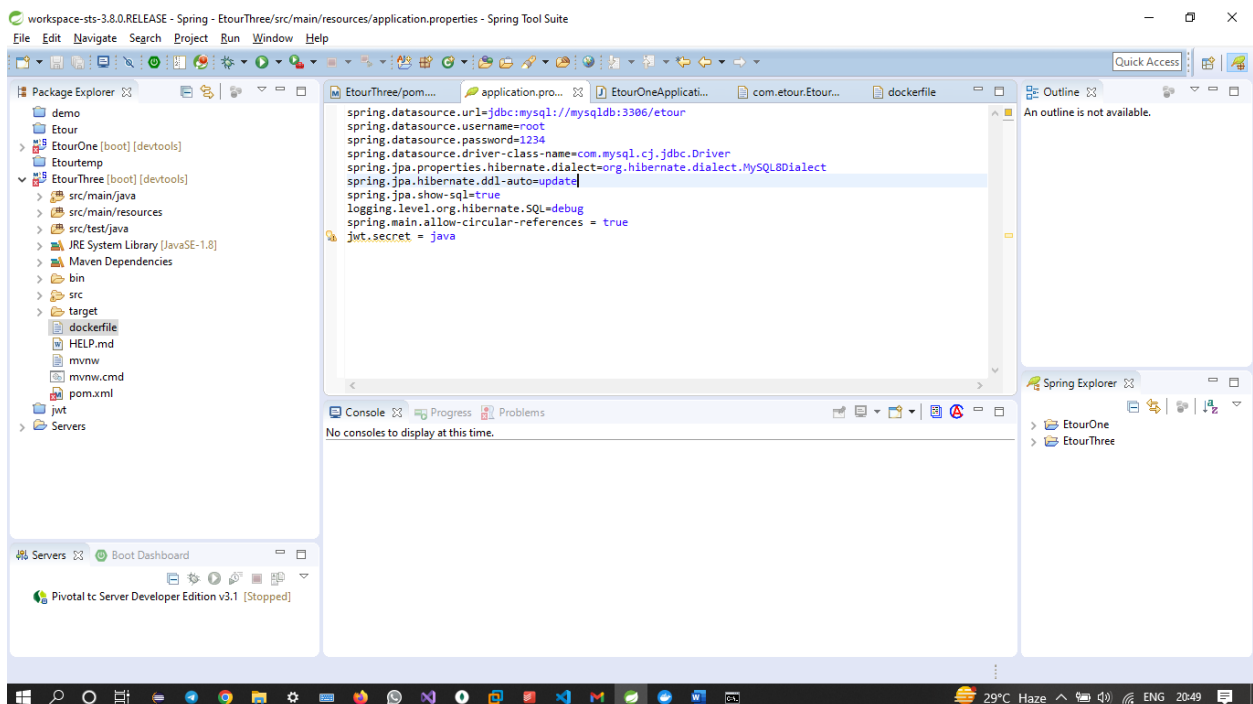
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

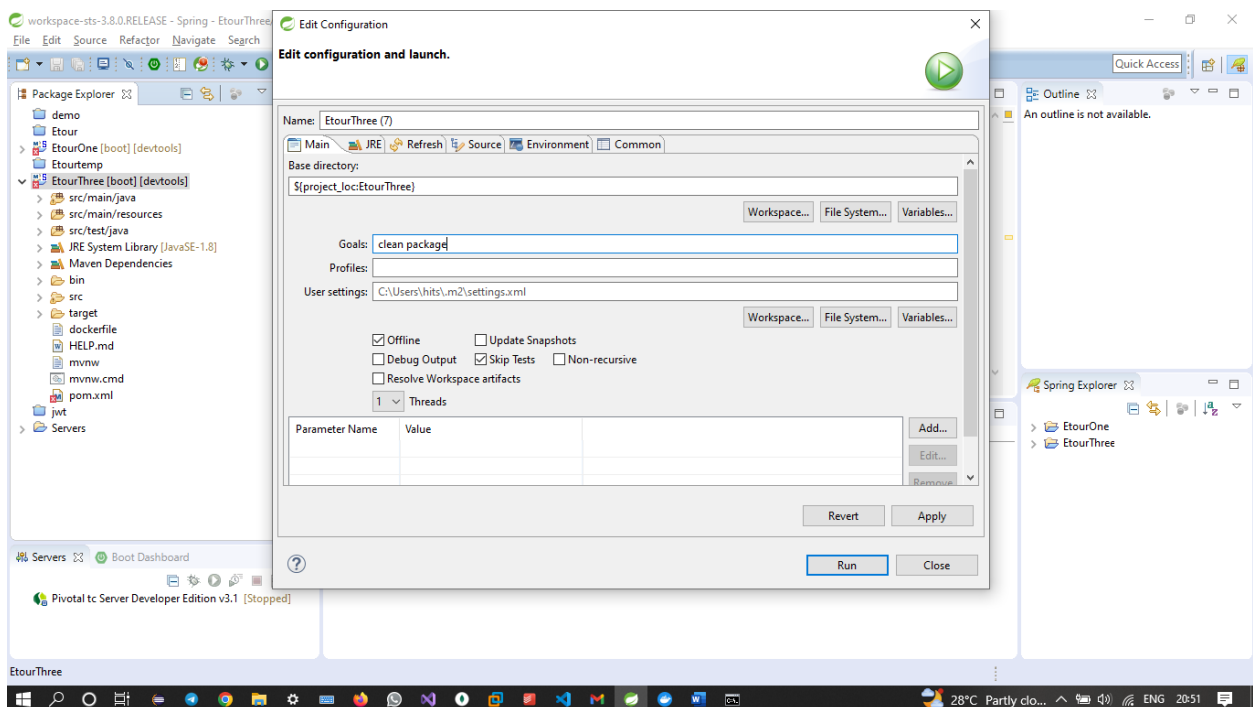
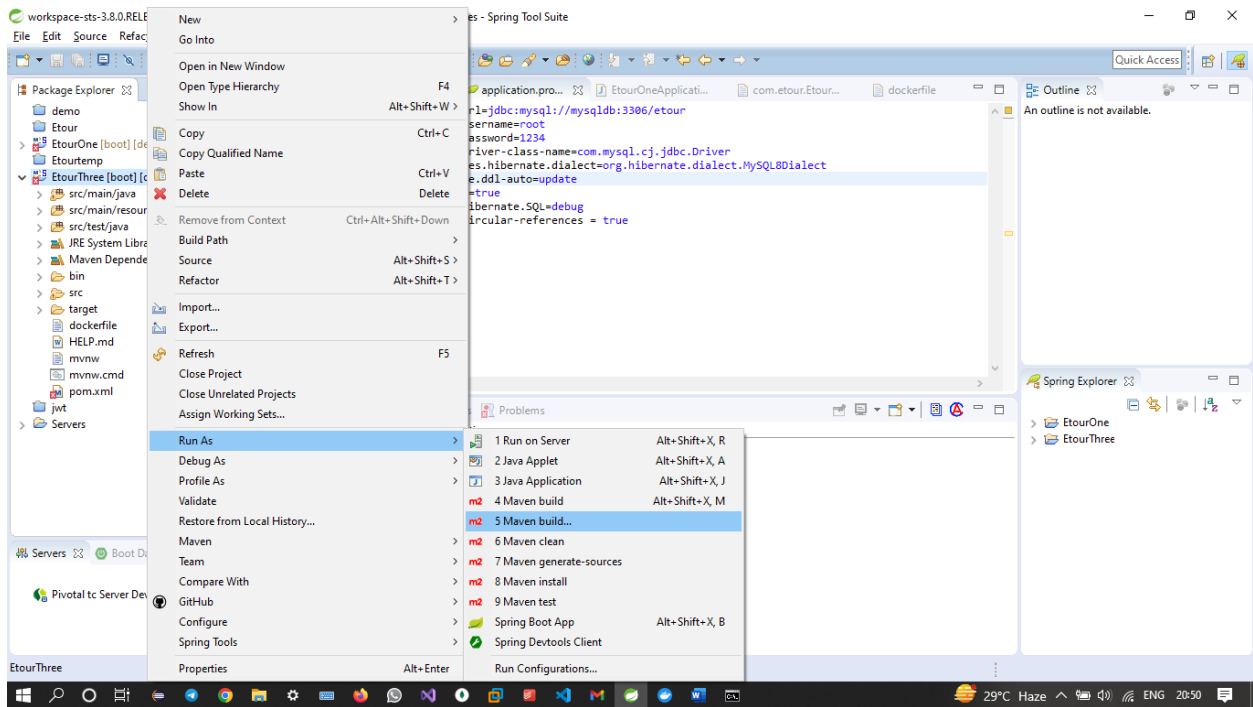
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

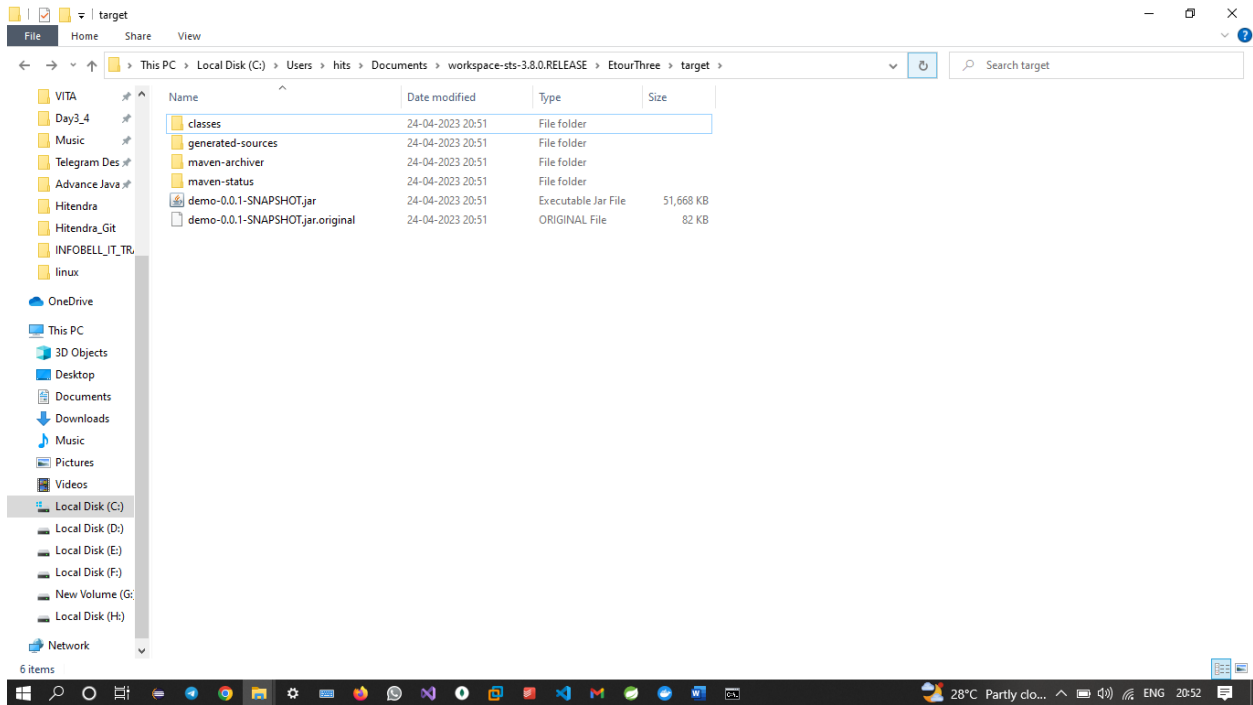
mysql> show databases
+-----+
| Database |
+-----+
| etour    |
| information_schema |
| mysql    |
| performance_schema |
| sys      |
+-----+
5 rows in set (0.01 sec)

mysql> use etour
Database changed
mysql>
  
```

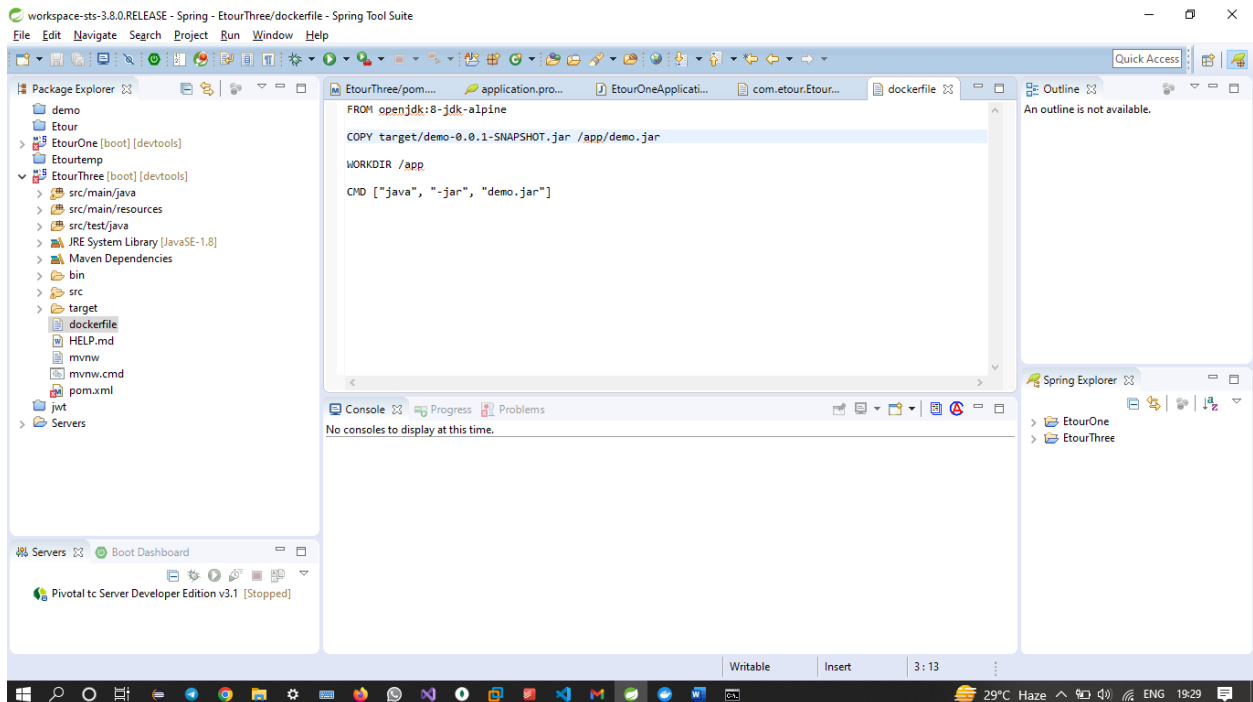
In Application.properties file of Spring-Boot Application add details of MySQL container.







FROM openjdk:8-jdk-alpine - Use the official OpenJDK 8 image as the base image. Copy the demo-0.0.1-SNAPSHOT.jar file from your local directory to the /app directory inside the container, and rename it to demo.jar.



Build the above dockerfile and will get Spring-boot container.

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.1288]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hites\Documents\workspace-sts-3.8.0.RELEASE\EtourThree>docker build -t springbootmysql .
[+] Building 5.5s (8/8) FINISHED
-> [internal] load build definition from Dockerfile
-> => transferring dockerfile: 32B
-> [internal] load .dockerignore
-> => transferring context: 2B
-> [internal] load metadata for docker.io/library/openjdk:8-jdk-alpine
-> [internal] load build context
-> => transferring context: 52.92MB
-> CACHED [1/3] FROM docker.io/library/openjdk:8-jdk-alpine@sha256:94792824df2df33402f201713f932b58cb9de94a0cd524164a0f2283343547b3
-> [2/3] COPY target/demo-0.0.1-SNAPSHOT.jar /app/demo.jar
-> [3/3] WORKDIR /app
-> exporting to image
-> => exporting layers
-> => writing image sha256:fa55ef1adbb085794a19ec691e81b7c33d388f9daa2482b8975ca33511fa409e
-> => naming to docker.io/library/springbootmysql

C:\Users\hites\Documents\workspace-sts-3.8.0.RELEASE\EtourThree>docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
springbootmysql      latest          fa55ef1adbb9    16 seconds ago  158MB
nginx                latest         6efc10a0510f    12 days ago     142MB
mysql                8.0.30         dbae59d1b41     6 months ago    449MB

C:\Users\hites\Documents\workspace-sts-3.8.0.RELEASE\EtourThree>
```

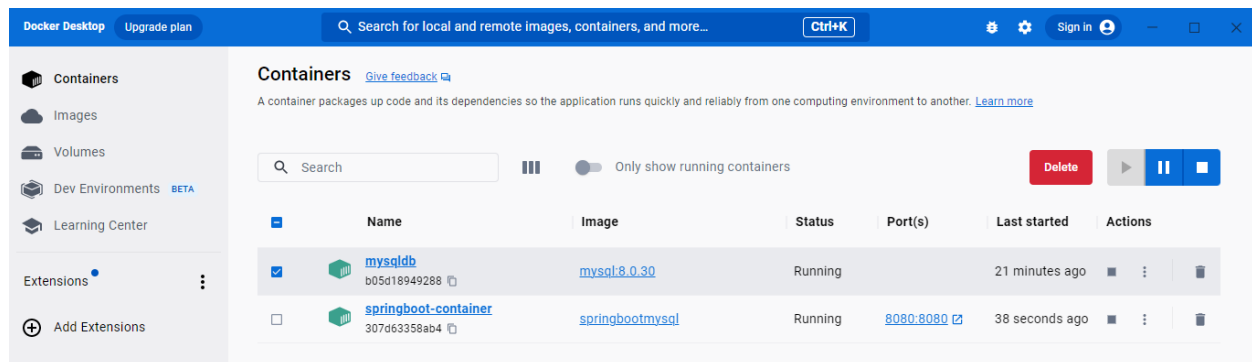
```
C:\Windows\System32\cmd.exe

C:\Users\hites\Documents\workspace-sts-3.8.0.RELEASE\EtourThree>docker run --network springboot-mysql-net --name springboot-container -p 8080:8080 -d springbootmysql
307d63358ab458d20df7e365f17203a05a1a1cbc64c6ceb026600026d7ba684

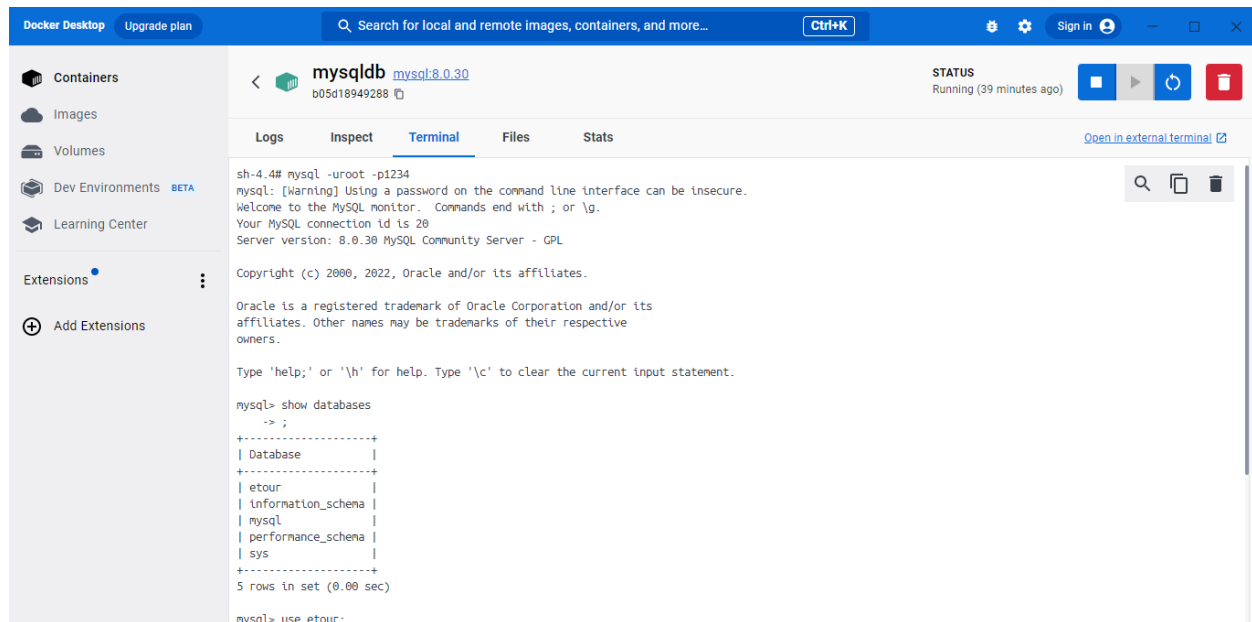
C:\Users\hites\Documents\workspace-sts-3.8.0.RELEASE\EtourThree>docker ps
CONTAINER ID   IMAGE             COMMAND                  CREATED        STATUS        PORTS                               NAMES
307d63358ab4   springbootmysql   "java -jar demo.jar"    15 seconds ago Up 13 seconds  0.0.0.0:8080->8080/tcp             springboot-container
b05d18949288   mysql:8.0.30     "docker-entrypoint.s..." 5 hours ago    Up 20 minutes  3306/tcp, 33060/tcp              mysqladb

C:\Users\hites\Documents\workspace-sts-3.8.0.RELEASE\EtourThree>
```

As you can see now MySQL and Spring-boot container are running.



Check if the database is created and If the tables are present in given database.



The screenshot shows the Docker Desktop interface. On the left, the 'Containers' tab is selected. The main panel displays the 'mysql' container (ID: b05d18949288) which is running. The 'Terminal' tab is active, showing the following output:

```
sh-4.4# mysql -uroot -p1234
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 20
Server version: 8.0.30 MySQL Community Server - GPL

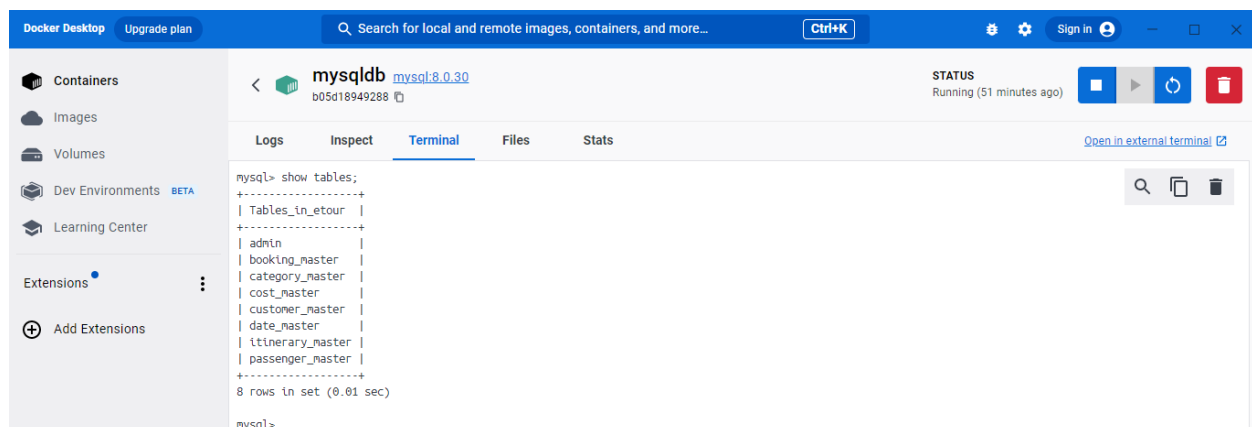
Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases
+-----+
| Database |
+-----+
| etour    |
| information_schema |
| mysql    |
| performance_schema |
| sys      |
+-----+
5 rows in set (0.00 sec)

mysql> use etour;
```

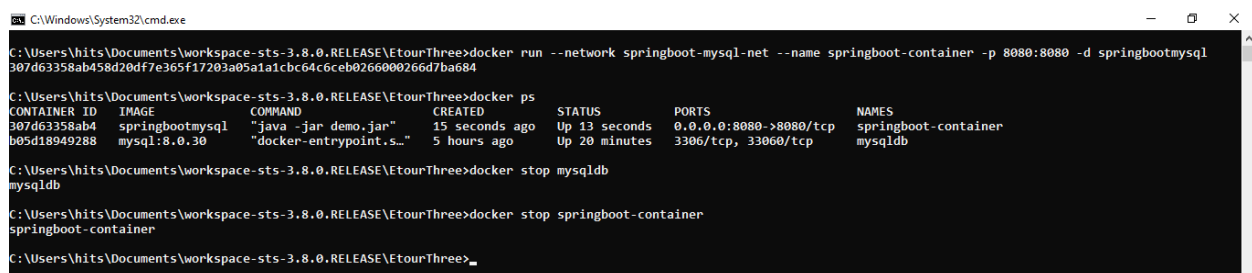


The screenshot shows the Docker Desktop interface. On the left, the 'Containers' tab is selected. The main panel displays the 'mysql' container (ID: b05d18949288) which is running. The 'Terminal' tab is active, showing the following output:

```
mysql> show tables;
+-----+
| Tables_in_etour |
+-----+
| admin            |
| booking_master   |
| category_master  |
| cost_master      |
| customer_master  |
| date_master      |
| itinerary_master |
| passenger_master |
+-----+
8 rows in set (0.01 sec)

mysql>
```

To stop running of containers:



The screenshot shows a Windows command prompt window with the following commands and output:

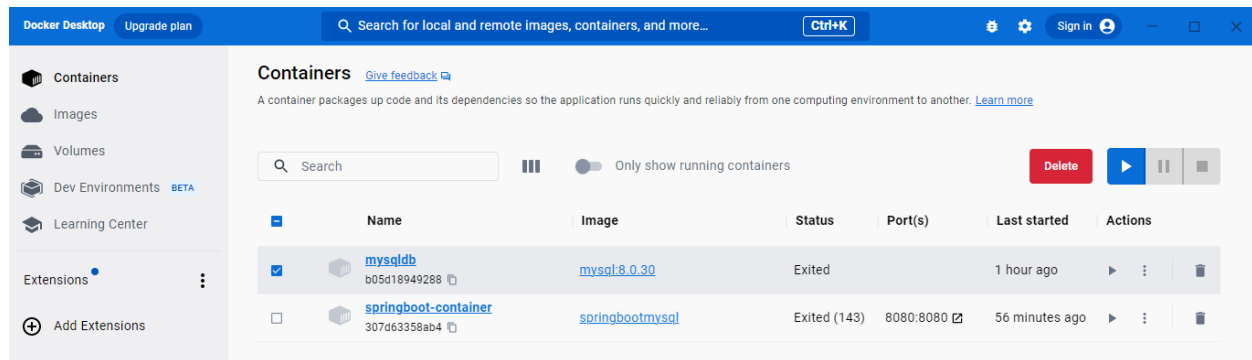
```
C:\Users\hites\Documents\workspace-sts-3.8.0.RELEASE\EtourThree>docker run --network springboot-mysql-net --name springboot-container -p 8080:8080 -d springbootmysql
307d63358ab458d20df7e365f17203a05a1a1cbc64c6ceb0266000266d7ba684

C:\Users\hites\Documents\workspace-sts-3.8.0.RELEASE\EtourThree>docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
307d63358ab4   springbootmysql  "java -jar demo.jar"    15 seconds ago Up 13 seconds  0.0.0.0:8080->8080/tcp             springboot-container
b05d18949288   mysql:8.0.30   "docker-entrypoint.s..." 5 hours ago    Up 20 minutes  3306/tcp, 33060/tcp             mysql

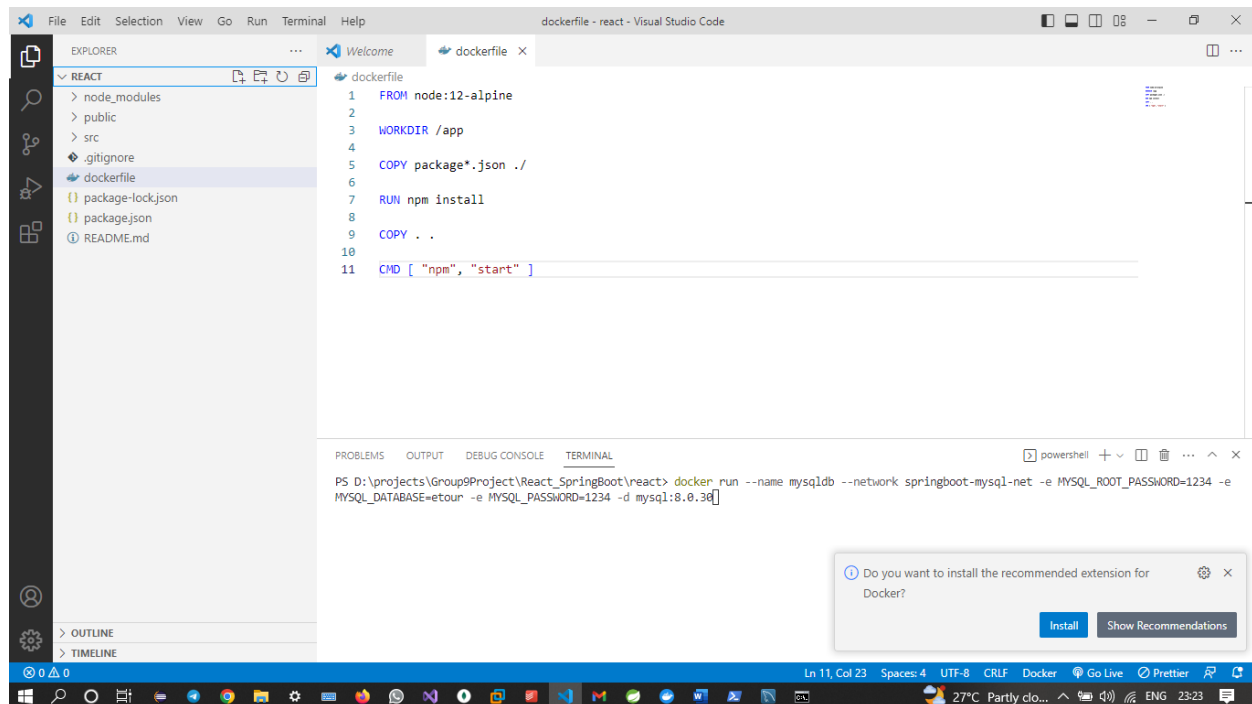
C:\Users\hites\Documents\workspace-sts-3.8.0.RELEASE\EtourThree>docker stop mysql
mysql

C:\Users\hites\Documents\workspace-sts-3.8.0.RELEASE\EtourThree>docker stop springboot-container
springboot-container

C:\Users\hites\Documents\workspace-sts-3.8.0.RELEASE\EtourThree>
```



FROM node:12-alpine - Use the official react image as the base image. And save as dockerfile. Copy package.json file from your local directory to the /app directory inside the container.



Build the above dockerfile and will get react app container.

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.1288]
(c) Microsoft Corporation. All rights reserved.

D:\projects\Group9Project\React_SpringBoot\react>docker build -t my-react-app .
[+] Building 1139.1s (10/10) FINISHED
-> [internal] load build definition from Dockerfile
-> => transferring dockerfile: 154B
-> [internal] load .dockerignore
-> => transferring context: 2B
-> [internal] load metadata for docker.io/library/node:12-alpine
-> [internal] load build context
-> => transferring context: 617.62MB
-> [1/5] FROM docker.io/library/node:12-alpine@sha256:d4b15b3d48f42059a15bd659be60afe21762aae9d6cbeaf124440895c27db68
-> => resolve docker.io/library/node:12-alpine@sha256:d4b15b3d48f42059a15bd659be60afe21762aae9d6cbeaf124440895c27db68
-> => sha256:d4b15b3d48f42059a15bd659be60afe21762aae9d6cbeaf124440895c27db68 1.43kB / 1.43kB
-> => sha256:4517380049fc3c9aacceae7764fcf3500354b0ac8a47e4afb35b5bb6b75b9498 1.16kB / 1.16kB
-> => sha256:b66d28039b8c9a8d9032f9aa64a792a60c2c1644591627bf946a3b27c8b 6.58kB / 6.58kB
-> => sha256:df9b9388f04d6279a7410b85cedfcb2208c0a003da7ab5613af71079148139 2.81MB / 2.81MB
-> => sha256:3bf6d738020517f4622814e8c21db4b4aaa78ae7cab4e4f872c11696886c6285 24.91MB / 24.91MB
-> => sha256:7939e01ee5e4737cf7fdb6d1dfe31ca4c2697109290462f694710761450aef0 2.36MB / 2.36MB
-> => extracting sha256:df9b9388f04d6279a7410b85cedfcb2208c0a003da7ab5613af71079148139
-> => extracting sha256:31f0fb9de071269230cb0f786012ae4e81d26e489b1fe922e57b5201e6bc9ab0 451B / 451B
-> => extracting sha256:3bf6d738020517f4622814e8c21db4b4aaa78ae7cab4e4f872c11696886c6285
-> => extracting sha256:7939e01ee5e4737cf7fdb6d1dfe31ca4c2697109290462f694710761450aef0
-> => extracting sha256:31f0fb9de071269230cb0f786012ae4e81d26e489b1fe922e57b5201e6bc9ab0
-> [2/5] WORKDIR /app
-> [3/5] COPY package*.json ./
-> [4/5] RUN npm install
-> [5/5] COPY . .
-> exporting to image
-> exporting layers
-> writing image sha256:f45b20d714a75265101ff0915b9ecd4faaeb9cf0feeb21f225282784cd2425
-> naming to docker.io/library/my-react-app

D:\projects\Group9Project\React_SpringBoot\react>docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
my-react-app latest f45b20d714a7 57 seconds ago 1.07GB
springbootmysql latest fa55ef1adb9 2 hours ago 158MB
nginx latest 6efc10a0510f 12 days ago 142MB
mysql 8.0.30 dbae59d1b41 6 months ago 449MB

D:\projects\Group9Project\React_SpringBoot\react>
```

```
C:\Windows\System32\cmd.exe - docker run -p 3000:3000 --network=springboot-mysql-net my-react-app
Line 3:10: 'Card' is defined but never used no-unused-vars
Line 3:24: 'FormControl' is defined but never used no-unused-vars
Line 82:9: 'token' is assigned a value but never used no-unused-vars

src/FunctionBased/Passenger.js
Line 1:10: 'useState' is defined but never used no-unused-vars
Line 5:10: 'useNavigate' is defined but never used no-unused-vars

src/FunctionBased/PassengerForm.js
Line 10:17: 'setpaxid' is assigned a value but never used no-unused-vars
Line 11:24: 'setpassbookingid' is assigned a value but never used no-unused-vars
Line 13:26: 'setPasscustomerid' is assigned a value but never used no-unused-vars

src/FunctionBased/PaymentPage.js
Line 32:17: img elements must have an alt prop, either with meaningful text, or an empty string for decorative images jsx-a11y/alt-text
Line 50:17: img elements must have an alt prop, either with meaningful text, or an empty string for decorative images jsx-a11y/alt-text

src/FunctionBased/RegistrationForm.js
Line 29:11: 'token' is assigned a value but never used no-unused-vars

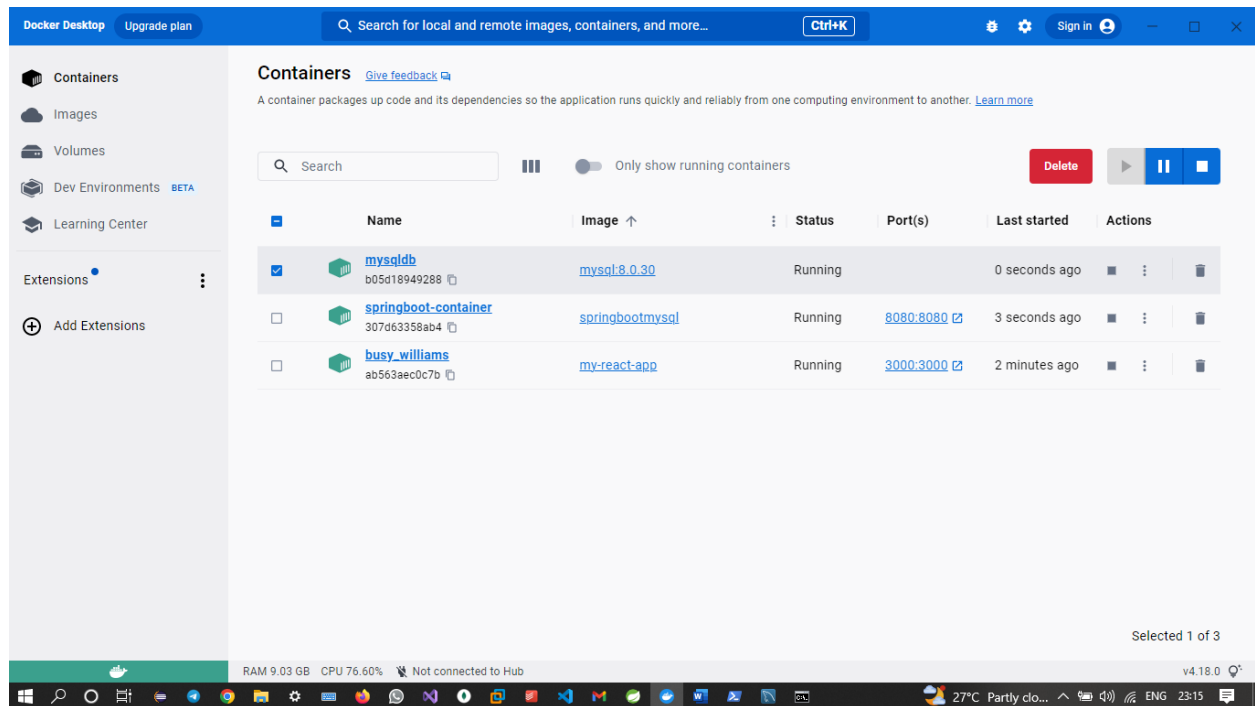
src/FunctionBased/Routetable.js
Line 2:40: 'Outlet' is defined but never used no-unused-vars
Line 14:8: 'ContentSearch' is defined but never used no-unused-vars
Line 20:10: 'NavigationBar' is defined but never used no-unused-vars
Line 25:8: 'BookingPageTrial' is defined but never used no-unused-vars

src/FunctionBased/Search.js
Line 3:10: 'Outlet' is defined but never used no-unused-vars
Line 13:9: 'token' is assigned a value but never used no-unused-vars
Line 22:6: React Hook useEffect has a missing dependency: 'code'. Either include it or remove the dependency array react-hooks/exhaustive-deps
Line 41:26: Expected '===' and instead saw '==' eeqeq

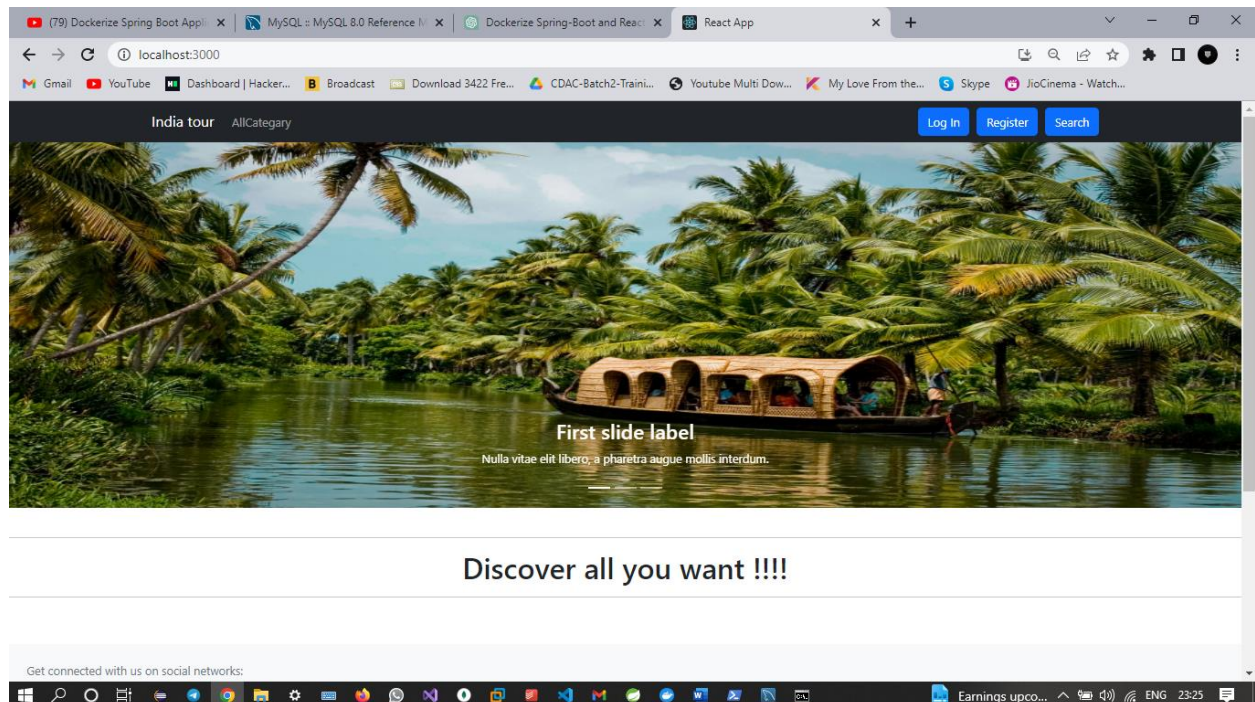
src/FunctionBased/UpdateCategory.js
Line 2:10: 'Button' is defined but never used no-unused-vars
Line 5:10: 'Form' is defined but never used no-unused-vars
Line 16:7: React Hook useEffect was passed a dependency list that is not an array literal. This means we can't statically verify whether you've passed the correct dependencies react-hooks/exhaustive-deps
Line 16:7: React Hook useEffect has a missing dependency: 'code'. Either include it or remove the dependency array react-hooks/exhaustive-deps

webpack compiled with 1 warning
```


As shown in below images total three containers are running (MySQL-Backend, Spring Boot- middle ware React- Frontend) :




We can now use url: "localhost:3000". The website is fully functional.



React App


localhost:3000/HomePage

Gmail YouTube Dashboard | Hacker... Broadcast Download 3422 Fre... CDAC-Batch2-Traini... Youtube Multi Dow... My Love From the... Skype JioCinema - Watch...




International

Explore




Varsha_Sahal

View Details




Adventure_Venture

View Details



Speciality_Tours

View Details



Crricket_Dhamaka

Explore

Get connected with us on social networks:

34°C Haze ENG 12:01


React App

localhost:3000/Itineraries/25

Gmail YouTube Dashboard | Hacker... Broadcast Download 3422 Fre... CDAC-Batch2-Traini... Youtube Multi Dow... My Love From the... Skype JioCinema - Watch...

India tour AllCategory

Log In Register Search



Dubai

Tour Duration: 15

Single Person Cost	75000
Extra Person Cost	75000
Child with Bed	40000
Valid From	2023-01-01
Valid To	2024-01-01

Book

Dubai_and_MiddleEast

Day 1-3: Dubai, UAE Arrival in Dubai and check-in at the hotel

Day 1: Visit the Dubai Mall and the Burj Khalifa

Day 2: Take a desert safari tour and experience a traditional Bedouin camp

Day 3: Explore the old city of Dubai, including the Dubai Creek and the Al Fahidi Historical Neighborhood

Day 4-5: Abu Dhabi, UAE Travel to Abu Dhabi and check-in at the hotel

Day 4: Visit the Sheikh Zayed Grand Mosque and the Emirates Palace

Day 5: Explore the Saadiyat Island Cultural District, including the Louvre Abu Dhabi and the Guggenheim Abu Dhabi

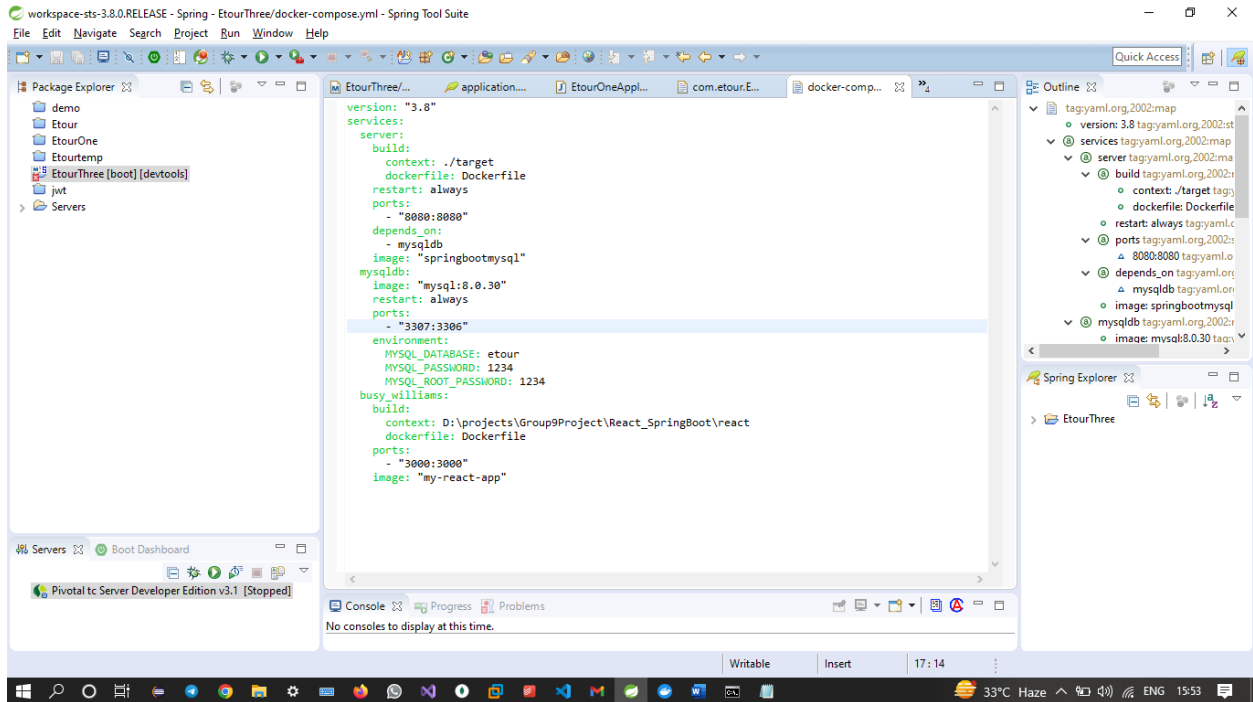
Day 6-7: Muscat, Oman Travel to Muscat and check-in at the hotel

Day 6: Visit the Sultan Qaboos Grand Mosque and the Mutrah Souq

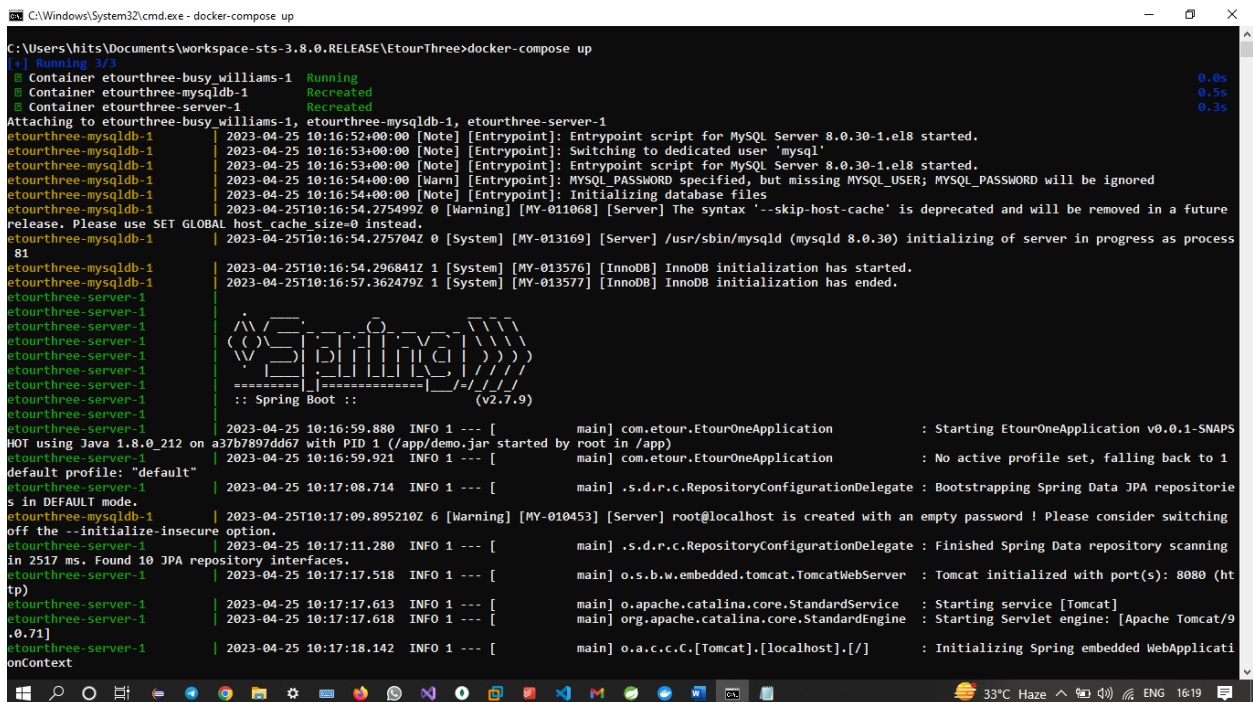
Day 7: Take a

34°C Haze ENG 12:02

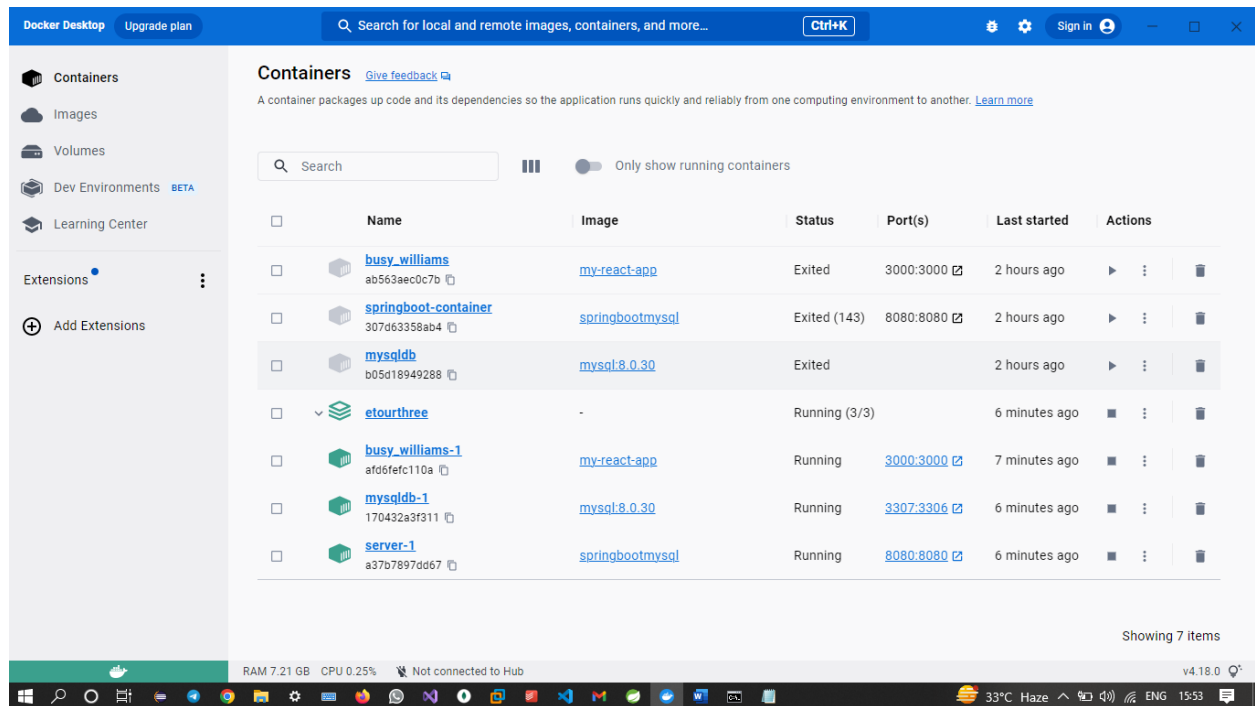
Now to dockerize a Spring Boot and React app as a composite file, we have to use docker compose to orchestrate this three images. For that we need to create 'docker-compose.yml' file in root of spring-Boot application



Now need to run the Docker containers using command "docker-Compose up"



A separate composed container created having all three containers as mentioned in yml file inside it. With “docker-compose up” we started running this container.



We can now use url: “localhost:3000”. The website is fully functional.

