DEVOPS- ASSIGNMENT- 2

Name: MAYANK
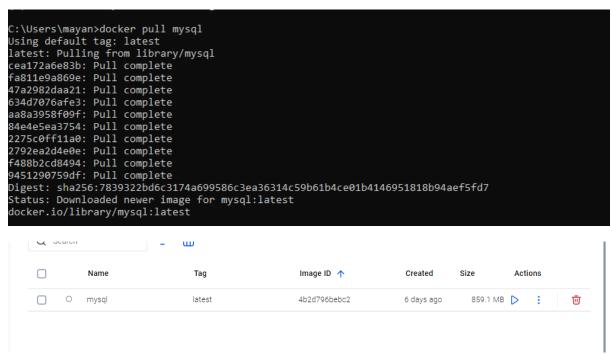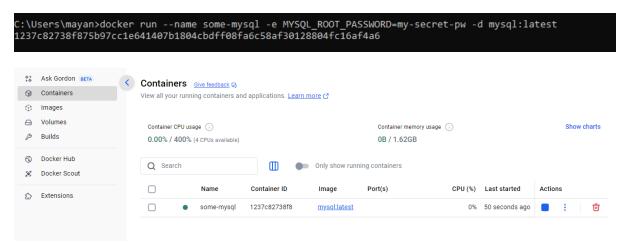
ROLL NO: 205224010

BRANCH: M.TECH DATA ANALYTICS

Q1. A) Create a Container with PostgresDB or mySQL database installed

Step 1: We need to pull the docker image using the below command.

```
C:\Users\mayan>docker pull mysql
Using default tag: latest
latest: Pulling from library/mysql
cea172a6e83b: Pull complete
fa811e9a869e: Pull complete
47a2982daa21: Pull complete
634d7076afe3: Pull complete
aa8a3958f09f: Pull complete
84e4e5ea3754: Pull complete
2275c0ff11a0: Pull complete
2792ea2d4e0e: Pull complete
f488b2cd8494: Pull complete
9451290759df: Pull complete
Digest: sha256:7839322bd6c3174a699586c3ea36314c59b61b4ce01b4146951818b94aef5fd7
Status: Downloaded newer image for mysql:latest
docker.io/library/mysql:latest
```

		Name	Tag	Image ID ↑	Created	Size	Actions		
☐	○	mysql	latest	4b2d796bebc2	6 days ago	859.1 MB	▷	⋮	🗑

Step 2: Run the docker image using the MySql

```
C:\Users\mayan>docker run --name some-mysql -e MYSQL_ROOT_PASSWORD=my-secret-pw -d mysql:latest
1237c82738f875b97cc1e641407b1804cbdff08fa6c58af30128804fc16af4a6
```

- Ask Gordon BETA
- Containers
- Images
- Volumes
- Builds
- Docker Hub
- Docker Scout
- Extensions

Containers Give feedback

View all your running containers and applications. Learn more

Container CPU usage ⓘ
0.00% / 400% (4 CPUs available)

Container memory usage ⓘ
0B / 1.62GB

Show charts

Only show running containers

		Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions		
☐	●	some-mysql	1237c82738f8	mysql:latest		0%	50 seconds ago	■	⋮	🗑

Step 3: Check whether the docker image is running or not

```
C:\Users\mayan>docker ps
CONTAINER ID   IMAGE          COMMAND                CREATED         STATUS              PORTS                     NAMES
1237c82738f8   mysql:latest   "docker-entrypoint.s…"  2 minutes ago   Up About a minute   3306/tcp, 33060/tcp       some-mysql
```

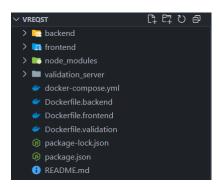**Step 4:** Connect to the database and test by creating a TABLE

```
C:\Users\mayan>docker exec -it some-mysql mysql -u root -p

Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 9.3.0 MySQL Community Server - GPL

Copyright (c) 2000, 2025, 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> CREATE DATABASE name;
Query OK, 1 row affected (0.097 sec)

mysql> USE name;
Database changed
mysql> CREATE TABLE test_table (
    ->      id INT AUTO_INCREMENT PRIMARY KEY,
    ->      name VARCHAR(100)
    -> );
Query OK, 0 rows affected (0.965 sec)

mysql> INSERT INTO test_table (name) VALUES ('MAYANK_205224010');
Query OK, 1 row affected (0.992 sec)

mysql> SELECT * FROM test_table;
+----+------------------+
| id | name             |
+----+------------------+
|  1 | MAYANK_205224010 |
+----+------------------+
1 row in set (0.064 sec)
```

Q1 B) Deploy VReqST – A requirement specification tool in a container.
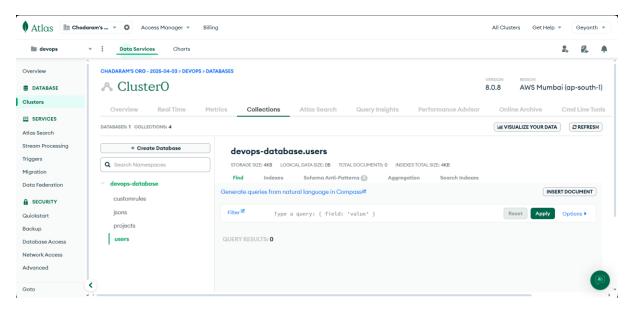
Step 1 : Clone the Repository using the commands

```
1  git clone https://github.com/<your-repository>/VReqST-2
2  cd VReqST-2/VReqST-main/VReqST
```

Step 2: Verify the folder structure

Step 4: Create Collections in MongoDB Atlas and use MONGO_URI to link to the application (the link is given in docker-composer file)

Step 3: Setup Docker

a. **Install Docker & Docker Compose -** Ensure both Docker and Docker Compose are installed and working.
b. **Create docker file `docker-compose.yml` -** Used to run the docker containers

c. **Create subfolders for different services** - Configure `Dockerfile.frontend`, `Dockerfile.backend`, `Dockerfile.validation`

```yaml
1   version: '3.8'
2
3   services:
4     backend:
5       build:
6         context: .
7         dockerfile: Dockerfile.backend
8       ports:
9         - "5002:5002"
10      environment:
11        - <MONGODB_URI>
12
13    validation:
14      build:
15        context: .
16        dockerfile: Dockerfile.validation
17      ports:
18        - "5001:5001"
19      environment:
20        - <MONGODB_URI>
21
22    frontend:
23      build:
24        context: .
25        dockerfile: Dockerfile.frontend
26      ports:
27        - "3000:3000"
28      depends_on:
29        - backend
30        - validation
```

d. **Build and Start Container**

```
1   docker-compose build
2   docker-compose up -d
```

Outputs:

```
D:\M.Tech\2nd Sem\Devops\Assignments\devops_tool\VReqST-2\VReqST-main\VReqST>docker-compose up -d
time="2025-04-19T21:23:15+05:30" level=warning msg="D:\\M.Tech\\2nd Sem\\Devops\\Assignments\\devops_tool\\VReqST
VReqST-main\\VReqST\\docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove i
 avoid potential confusion"
[+] Running 4/4
 ✓ Network vreqst_default       Created                                                                       0
 ✓ Container vreqst-backend-1   Started                                                                       3
 ✓ Container vreqst-validation-1 Started                                                                      3
 ✓ Container vreqst-frontend-1  Started                                                                       2
```

- Running the Application

```
D:\M.Tech\2nd Sem\Devops\Assignments\devops_tool\VReqST-2\VReqST-main\VReqST>docker-compose up --build -d
time="2025-04-19T21:30:42+05:30" level=warning msg="D:\\M.Tech\\2nd Sem\\Devops\\Assignments\\devops_tool\\VReqST-2\\
VReqST-main\\VReqST\\docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to
 avoid potential confusion"
Compose can now delegate builds to bake for better performance.
 To do so, set COMPOSE_BAKE=true.
[+] Building 226.9s (26/26) FINISHED                                                          docker:desktop-linux
 => [validation internal] load build definition from Dockerfile.validation                                     0.0s
 => => transferring dockerfile: 154B                                                                           0.0s
 => [backend internal] load build definition from Dockerfile.backend                                           0.1s
 => => transferring dockerfile: 280B                                                                           0.0s
 => [frontend internal] load metadata for docker.io/library/node:14                                            2.2s
 => [backend internal] load .dockerignore                                                                      0.0s
 => => transferring context: 2B                                                                                0.0s
 => [validation internal] load .dockerignore                                                                   0.1s
 => => transferring context: 2B                                                                                0.0s
 => [backend internal] load build context                                                                      17.3s
 => => transferring context: 32.59MB                                                                           17.1s
 => [frontend 1/5] FROM docker.io/library/node:14@sha256:a158d3b9b4e3fa813fa6c8c590b8f0a860e015ad4e59bbce5744d  0.2s
 => => resolve docker.io/library/node:14@sha256:a158d3b9b4e3fa813fa6c8c590b8f0a860e015ad4e59bbce5744d2f6fd8461  0.0s
 => [validation internal] load build context                                                                   1.1s
 => => transferring context: 67.22kB                                                                           0.9s
 => CACHED [frontend 2/5] WORKDIR /app                                                                          0.0s
 => CACHED [validation 3/4] COPY validation_server/ .                                                          0.0s
 => CACHED [validation 4/4] RUN npm install                                                                    0.0s
 => [validation] exporting to image                                                                            0.6s
 => => exporting layers                                                                                        0.0s
 => => exporting manifest sha256:054f0155f3a3378988bbb8a442d1b686e32e8f2052deebecee896cec402ce587              0.0s
```

```
 => [backend] resolving provenance for metadata file                                                           0.1s
 => [frontend internal] load build definition from Dockerfile.frontend                                         0.0s
 => => transferring dockerfile: 177B                                                                           0.0s
 => [frontend internal] load .dockerignore                                                                     0.1s
 => => transferring context: 2B                                                                                0.0s
 => [frontend internal] load build context                                                                     201.8s
 => => transferring context: 664.95MB                                                                          200.6s
 => CACHED [frontend 3/5] COPY frontend/ .                                                                      0.0s
 => CACHED [frontend 4/5] RUN npm install                                                                      0.0s
 => CACHED [frontend 5/5] RUN ls -la /app/client  # Add this line                                              0.0s
 => [frontend] exporting to image                                                                              0.4s
 => => exporting layers                                                                                        0.0s
 => => exporting manifest sha256:25d3061c140fb874e71d76009238eeeec972e7437de7cce4db936ad66f562193             0.0s
 => => exporting config sha256:00413a90cc6393cfac6ab26a6665f5d79e1368a46754e1a20d739e9019c950a3               0.0s
 => => exporting attestation manifest sha256:ba87c8f2edb3299ed013c51304019001bc46aa6302e5e2d3f20e1acc866eecf5  0.1s
 => => exporting manifest list sha256:3100cf308e74f287d2b670743cfa1993cd21145778fe7368091658cda829af30        0.1s
 => => naming to docker.io/library/vreqst-frontend:latest                                                      0.0s
 => => unpacking to docker.io/library/vreqst-frontend:latest                                                   0.0s
 => [frontend] resolving provenance for metadata file                                                          0.0s
[+] Running 7/7
 ✓ backend                       Built                                                                         0.0s
 ✓ frontend                      Built                                                                         0.0s
 ✓ validation                    Built                                                                         0.0s
 ✓ Network vreqst_default        Created                                                                       0.1s
 ✓ Container vreqst-backend-1    Started                                                                       4.2s
 ✓ Container vreqst-validation-1 Started                                                                       4.2s
 ✓ Container vreqst-frontend-1   Started                                                                       2.9s
```

Step 4: Access the Application

Open your browser and go to: **http://localhost:5000**

Open your browser and go to: **http://localhost:5000**