EmpName :Rishi Kesava Vigneshver Manisekaran

EmpId :12575

Project Title : Campus Event Aggregator

Q7) Campus Event Aggregator

A console application for discovering, managing, and promoting university events. It connects students with campus happenings—academic, cultural, and social—while enabling organizers to manage event listings and attendance efficiently.

User Registration and Login
As a student, I want to register and log in to the system, so that I can access and sign up for campus events.

Browse Events
As a student, I want to view a list of upcoming campus events, so that I can stay informed about academic, cultural, and social activities.

Search and Filter Events
As a student, I want to search and filter events by category, date, or location, so that I can find events that interest me.

Register for Events
As a student, I want to register for events, so that I can attend and participate.

View My Registered Events
As a student, I want to view the events I've registered for, so that I can manage my schedule.

User, Event

Tasks:

1. **Perform W3H analysis on the given Case Study**

PROJECT: Campus Event Aggregator	
What?	**How?**
What are the Modules used in the Campus Event Aggregator ? 1. User 2. Event --- 1.What are the functionalities in the User module? a. User can Login b. User can add Register Event c. User can delete Register Event d. User can Edit Register Event e. User can view Register Event --- 2.What are the functionalities in the Event module? a. Students can add the Title , Description, Date, Location, Category b. Students can delete the Title , Description, Date, Location, Category	1.User a. User can Login **Method 1**: Login through Name and password **Method 2**: Login through Email id and password **Method 3**: Login through Social profiles --- b. User can add Appointment Scheduling, Online Booking, Billing, tracking customers, staff Members, Inventory Tracking **Method 1**: Add Register Event directly in the MySQL table using MVC through API **Method 2**: Add Register Event through the Excel **Method 3**: Add Register Event through the UI --- c. User can delete Register Event **Method 1**: Delete Register Event by Name **Method 2**: Delete Register Event by category **Method 3**: Delete Register Event by Id ---

c. Students can Edit the Title , Description, Date, Location, Category

d. Students can view the Title , Description, Date, Location, Category

----

3.What are the database tables required?

a. tbl_User

b. tbl_Appointments

----

d. User can Edit Register Event

Method 1: Edit Register Event by Name

Method 2: Edit Register Event by category

Method 3: Edit Register Event by Id

----

e. User can view Register Event

Method 1: view Register Event by Name

Method 2: view Register Event by category

Method 3: view Register Event by Id

2.User

a. Students can add Title , Description, Date, Location, Category

Method 1: Add the Title , Description, Date, Location, Category directly in the MySQL table using MVC through API

Method 2: Add the Title , Description, Date, Location, Category through the Excel

Method 3: Add the Title , Description, Date, Location, Category through the UI

----

b. Students can delete the Title , Description, Date, Location, Category

Method 1: Delete the Title , Description, Date, Location, Category by Name

Method 2: Delete the Title , Description, Date, Location, Category by category

Method 3: Delete the Title , Description, Date, Location, Category by Id

----

c. Students can Edit the Title , Description, Date, Location, Category

Method 1: Edit the Title , Description, Date, Location, Category by Name

Method 2: Edit the Title , Description, Date, Location, Category by category

Method 3: Edit the Title , Description, Date, Location, Category by Id

----

d. Students can view the Title , Description, Date, Location, Category

Method 1: view the Title , Description, Date, Location, Category by Name

Method 2: view the Title , Description, Date, Location, Category by category

Method 3: view the Title , Description, Date, Location, Category by Id

----

Why?	Why not?
1.User a. User can Login Method 2: Login through Email id and password Login using email id and password is more secure because the email is more unique so no one can duplicate it. For this platform the email id and password factor authentication is more sufficient. This is more secure, and you can login with only the owner of the email. -- b. User can add Register Event Method 1: Add Register Event directly in the MySQL table using MVC through API MYSQL is a good practice and easy way to add the more no of Appointment Scheduling, Online Booking, Billing, tracking customers, staff Members, Inventory Tracking in one single table . This is also easy and more comfortable one as well as Appointment Scheduling, Online Booking, Billing, tracking customers, staff Members, Inventory Tracking friendly and easy understanding. This is more sufficient for this platform. -- c. User can delete Register Event Method 3: Delete Register Event by Id Id is the unique one and act as a primary key in the database, so it is more useful one for the deleting the Datas whose having this ID this the right way of deleting Register Event	1.User a. User can Login The Email id and password is more sufficient for this platform and others do have some duplications of names and social profiles so that's not good appropriate one for the secure platform. -- b. User can add Register Event MYSQL is More sufficient when compared to others and the others may leads to change in main table of the database and somewhat difficult to add Register Event s in a Single manner. These are the major reasons why I'm not using the other methods. -- c. User can delete Register Event Id is the unique one and act as a primary key in the database, so it is more useful one for the deleting the Datas and others having the duplications of the data those are not the right method for delete Register Event . -- d. User can Edit Register Event Id is the unique one and act as a primary key in the database, so it is more useful one for the editing the Datas and others having the duplications of the data those are not the right method for edit Register Event .

by their id.

----

d. User can Edit Register Event

Method 3: Edit Register Event by Id

Id is the unique one and act as a primary key in the database, so it is more useful one for the editing the Datas whose having this IDthis the right way of editing Register Event by their id.

----

e. User can view Register Event

Method 3: view Appointment Scheduling, Online Booking, Billing, tracking customers, staff Members, Inventory Tracking by Id

Id is the unique one and act as a primary key in the database, so it is more useful one for the viewing the Datas whose having this ID this the right way of viewing Register Event by their id.

2.Event

a. Students can Login

Method 2: Login through Email id and password

Login using email id and password is more secure because the email is more unique so no one can duplicate it. For this platform the email id and password factor authentication is more sufficient. This is more secure, and you can login with only the owner of the email.

----

b. Students can add the Title , Description, Date, Location, Category

----

e. User can view Appointment Scheduling, Online Booking, Billing, tracking customers, staff Members, Inventory Tracking

Id is the unique one and act as a primary key in the database, so it is more useful one for the viewing the Datas and others having the duplications of the data those are not the right method for view Register Event .

2.Event

a. Students can add the Title , Description, Date, Location, Category

MYSQL is More sufficient when compared to others and the others may leads to change in main table of the database and somewhat difficult to add the Title , Description, Date, Location, Categorys in a Single manner. These are the major reasons why I'm not using the other methods.

----

b. Students can delete the Title , Description, Date, Location, Category

Id is the unique one and act as a primary key in the database, so it is more useful one for the deleting the Datas and others having the duplications of the data those are not the right method for delete the Title , Description, Date, Location, Category.

----

c. Students can Edit the Title , Description, Date, Location, Category

Id is the unique one and act as a primary key in

Method 1: Add the Title , Description, Date, Location, Category directly in the MySQL table using MVC through API

MYSQL is a good practice and easy way to add the more no of Title , Description, Date, Location, Category in one single table . This is also easy and more comfortable one as well as Title , Description, Date, Location, Category friendly and easy understanding. This is more sufficient for this platform.

----

c. Students can delete the Title , Description, Date, Location, Category

Method 3: Delete the Title , Description, Date, Location, Category by Id

Id is the unique one and act as a primary key in the database, so it is more useful one for the deleting the Datas whose having this ID this the right way of deleting the Title , Description, Date, Location, Category by their id.

----

d. Students can Edit the Title , Description, Date, Location, Category

Method 3: Edit the Title , Description, Date, Location, Category by Id

Id is the unique one and act as a primary key in the database, so it is more useful one for the editing the Datas whose having this IDthis the right way of editing the Title , Description, Date, Location, Category by their id.

----

e. Students can view the Title , Description, Date, Location, Category

Method 3: view Title , Description, Date, Location,

the database, so it is more useful one for the editing the Datas and others having the duplications of the data those are not the right method for edit the Title , Description, Date, Location, Category.

----

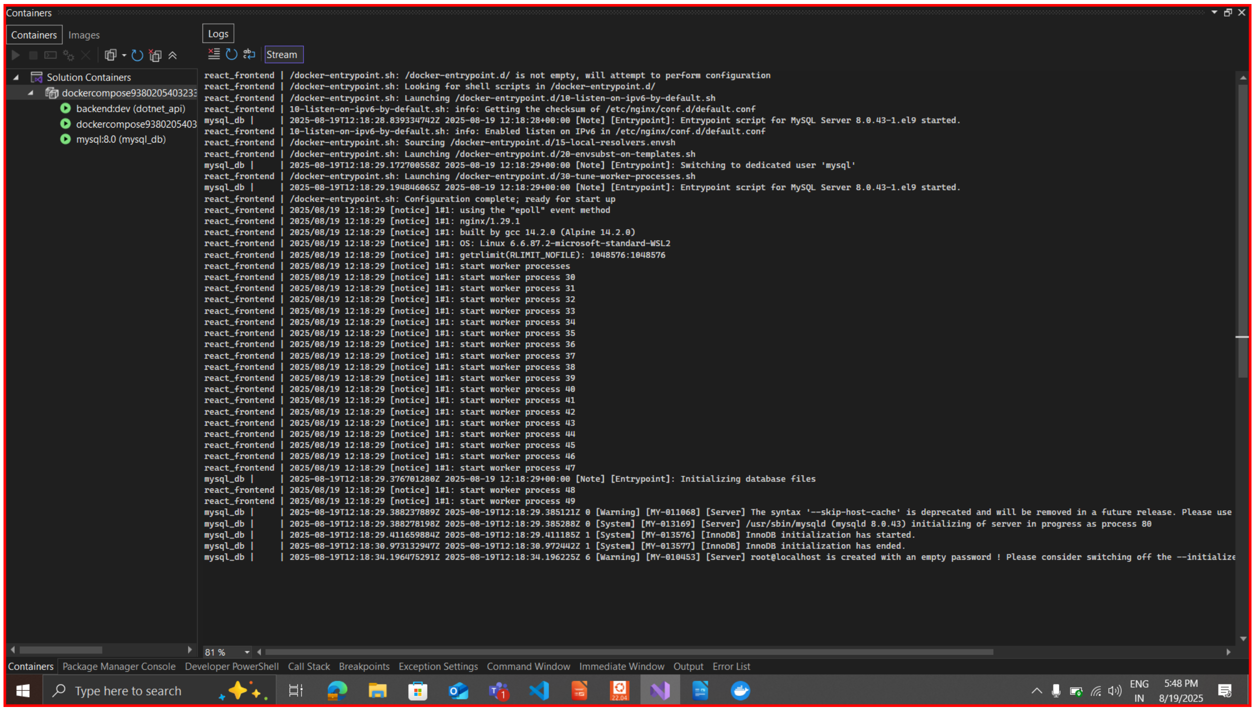
d. Students can view Title , Description, Date, Location, Category

Id is the unique one and act as a primary key in the database, so it is more useful one for the viewing the Datas and others having the duplications of the data those are not the right method for view the Title , Description, Date, Location, Category.

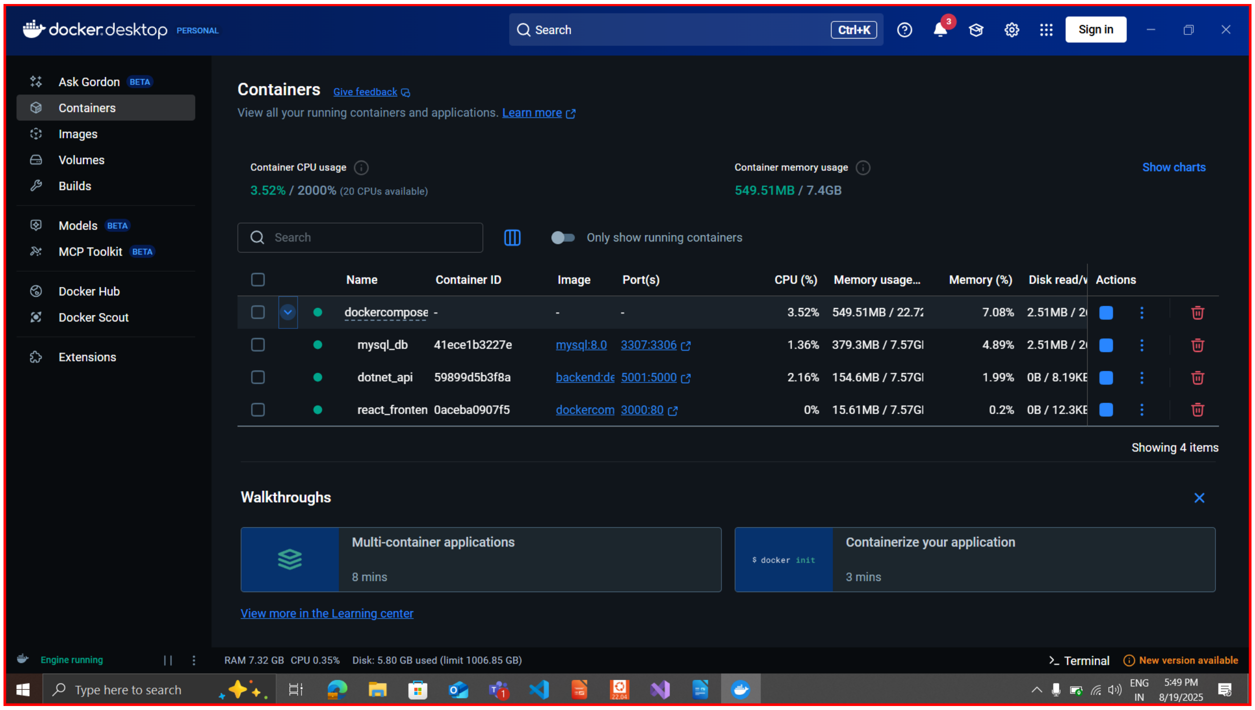
Category by Id

Id is the unique one and act as a primary key in the database, so it is more useful one for the viewing the Datas whose having this ID this the right way of viewing the Title , Description, Date, Location, Category by their id.

Docker



```
react_frontend | /docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
react_frontend | /docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
react_frontend | /docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
react_frontend | 10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
mysql_db |     | 2025-08-19T12:18:28.839334742Z 2025-08-19 12:18:28+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.43-1.el9 started.
react_frontend | 10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
react_frontend | /docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
react_frontend | /docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
mysql_db |     | 2025-08-19T12:18:29.172700558Z 2025-08-19 12:18:29+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
react_frontend | /docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
mysql_db |     | 2025-08-19T12:18:29.194846065Z 2025-08-19 12:18:29+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.43-1.el9 started.
react_frontend | /docker-entrypoint.sh: Configuration complete; ready for start up
react_frontend | 2025/08/19 12:18:29 [notice] 1#1: using the "epoll" event method
react_frontend | 2025/08/19 12:18:29 [notice] 1#1: nginx/1.29.1
react_frontend | 2025/08/19 12:18:29 [notice] 1#1: built by gcc 14.2.0 (Alpine 14.2.0)
react_frontend | 2025/08/19 12:18:29 [notice] 1#1: OS: Linux 6.6.87.2-microsoft-standard-WSL2
react_frontend | 2025/08/19 12:18:29 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
react_frontend | 2025/08/19 12:18:29 [notice] 1#1: start worker processes
react_frontend | 2025/08/19 12:18:29 [notice] 1#1: start worker process 30
react_frontend | 2025/08/19 12:18:29 [notice] 1#1: start worker process 31
react_frontend | 2025/08/19 12:18:29 [notice] 1#1: start worker process 32
react_frontend | 2025/08/19 12:18:29 [notice] 1#1: start worker process 33
react_frontend | 2025/08/19 12:18:29 [notice] 1#1: start worker process 34
react_frontend | 2025/08/19 12:18:29 [notice] 1#1: start worker process 35
react_frontend | 2025/08/19 12:18:29 [notice] 1#1: start worker process 36
react_frontend | 2025/08/19 12:18:29 [notice] 1#1: start worker process 37
react_frontend | 2025/08/19 12:18:29 [notice] 1#1: start worker process 38
react_frontend | 2025/08/19 12:18:29 [notice] 1#1: start worker process 39
react_frontend | 2025/08/19 12:18:29 [notice] 1#1: start worker process 40
react_frontend | 2025/08/19 12:18:29 [notice] 1#1: start worker process 41
react_frontend | 2025/08/19 12:18:29 [notice] 1#1: start worker process 42
react_frontend | 2025/08/19 12:18:29 [notice] 1#1: start worker process 43
react_frontend | 2025/08/19 12:18:29 [notice] 1#1: start worker process 44
react_frontend | 2025/08/19 12:18:29 [notice] 1#1: start worker process 45
react_frontend | 2025/08/19 12:18:29 [notice] 1#1: start worker process 46
react_frontend | 2025/08/19 12:18:29 [notice] 1#1: start worker process 47
mysql_db |     | 2025-08-19T12:18:29.376701280Z 2025-08-19 12:18:29+00:00 [Note] [Entrypoint]: Initializing database files
react_frontend | 2025/08/19 12:18:29 [notice] 1#1: start worker process 48
react_frontend | 2025/08/19 12:18:29 [notice] 1#1: start worker process 49
mysql_db |     | 2025-08-19T12:18:29.388237889Z 2025-08-19T12:18:29.385121Z 0 [Warning] [MY-011068] [Server] The syntax '--skip-host-cache' is deprecated and will be removed in a future release. Please use
mysql_db |     | 2025-08-19T12:18:29.388278198Z 2025-08-19T12:18:29.385288Z 0 [System] [MY-013169] [Server] /usr/sbin/mysqld (mysqld 8.0.43) initializing of server in progress as process 80
mysql_db |     | 2025-08-19T12:18:29.411659884Z 2025-08-19T12:18:29.411185Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
mysql_db |     | 2025-08-19T12:18:30.973132947Z 2025-08-19T12:18:30.972442Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
mysql_db |     | 2025-08-19T12:18:34.196475291Z 2025-08-19T12:18:34.196225Z 6 [Warning] [MY-010453] [Server] root@localhost is created with an empty password ! Please consider switching off the --initialize
```



Containers

View all your running containers and applications. Learn more

Container CPU usage
3.52% / 2000% (20 CPUs available)

Container memory usage
549.51MB / 7.4GB

	Name	Container ID	Image	Port(s)	CPU (%)	Memory usage...	Memory (%)	Disk read/v	Actions
□	dockercompose	-	-	-	3.52%	549.51MB / 22.7:	7.08%	2.51MB / 2(	
□	mysql_db	41ece1b3227e	mysql:8.0	3307:3306	1.36%	379.3MB / 7.57GI	4.89%	2.51MB / 2(	
□	dotnet_api	59899d5b3f8a	backend:de	5001:5000	2.16%	154.6MB / 7.57GI	1.99%	0B / 8.19KE	
□	react_fronten	0aceba0907f5	dockercom	3000:80	0%	15.61MB / 7.57GI	0.2%	0B / 12.3KE	

Showing 4 items

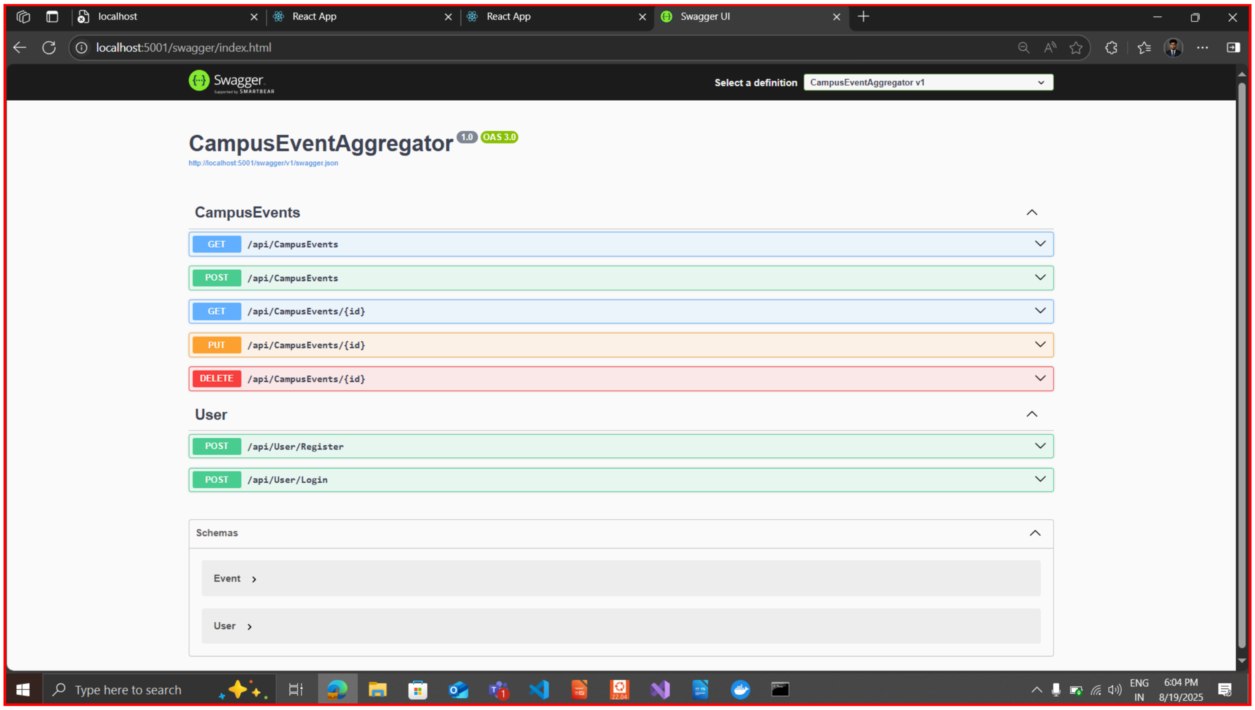
Walkthroughs

Multi-container applications
8 mins

$ docker init
Containerize your application
3 mins

View more in the Learning center

RAM 7.32 GB CPU 0.35% Disk: 5.80 GB used (limit 1006.85 GB)



Select a definition CampusEventAggregator v1

CampusEventAggregator 1.0 OAS 3.0

http://localhost:5001/swagger/v1/swagger.json

CampusEvents

GET /api/CampusEvents

POST /api/CampusEvents

GET /api/CampusEvents/{id}

PUT /api/CampusEvents/{id}

DELETE /api/CampusEvents/{id}

User

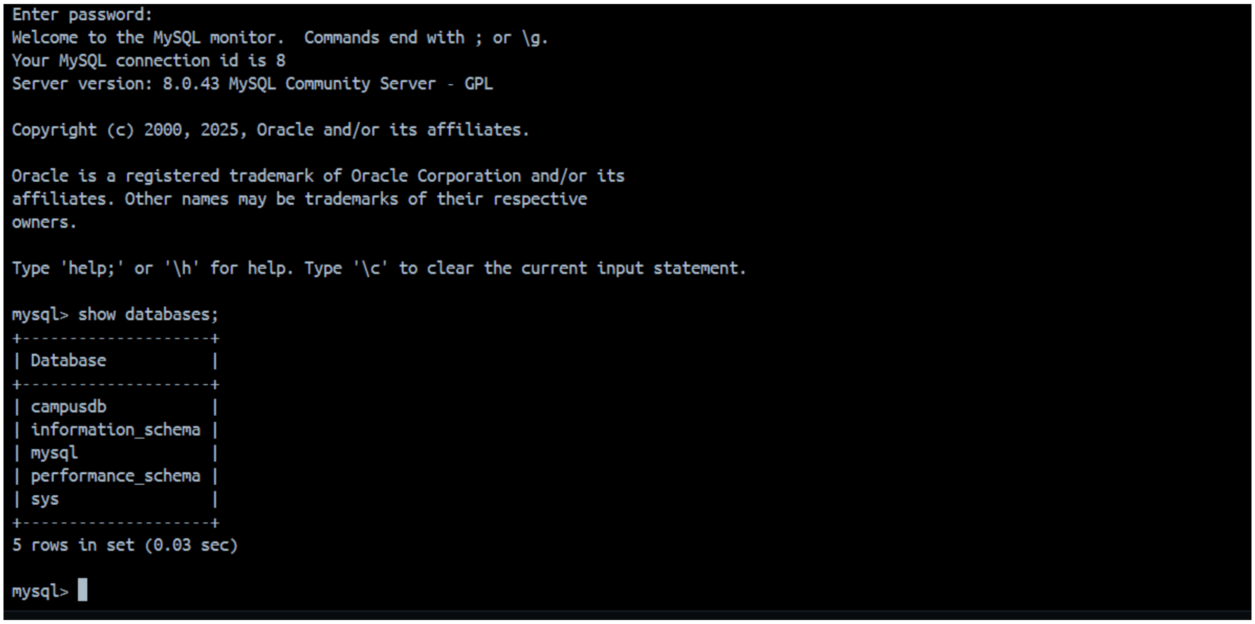
POST /api/User/Register

POST /api/User/Login

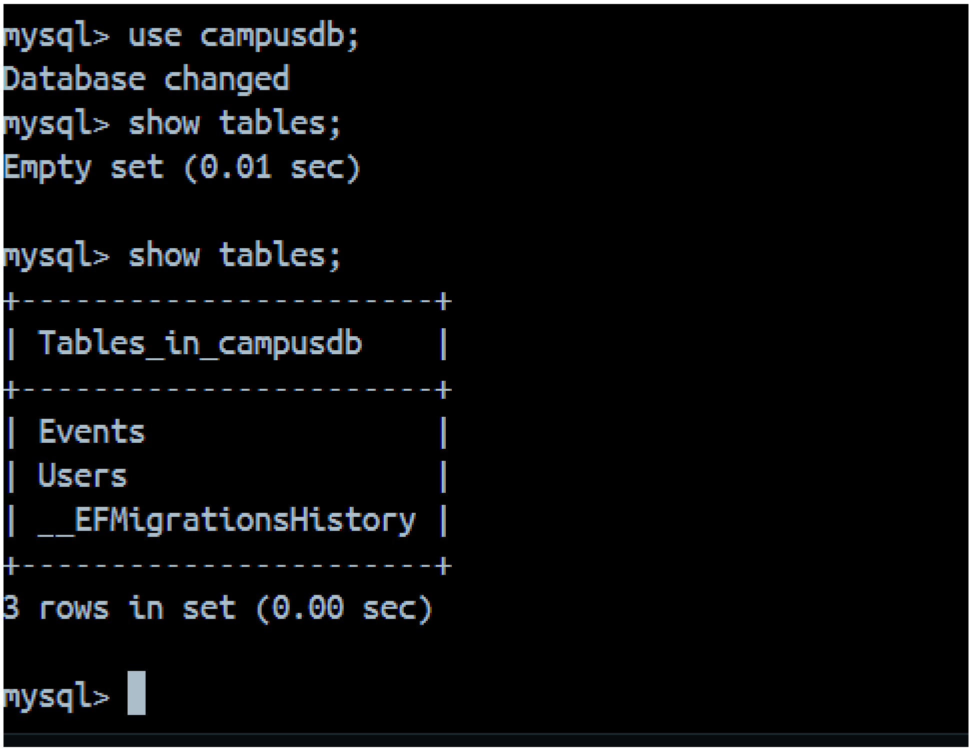
Schemas

Event ›

User ›

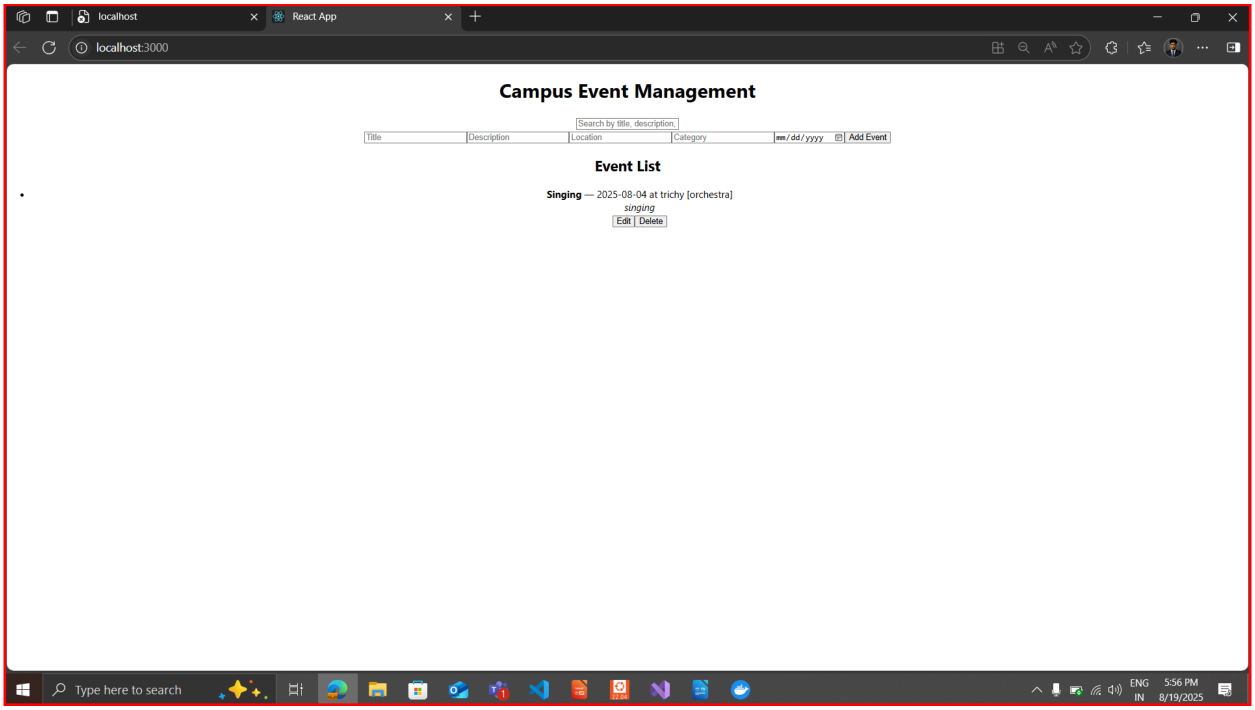


```
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.43 MySQL Community Server - GPL

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> show databases;
+--------------------+
| Database           |
+--------------------+
| campusdb           |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.03 sec)

mysql> 
```



```
mysql> use campusdb;
Database changed
mysql> show tables;
Empty set (0.01 sec)

mysql> show tables;
+-----------------------+
| Tables_in_campusdb    |
+-----------------------+
| Events                |
| Users                 |
| __EFMigrationsHistory |
+-----------------------+
3 rows in set (0.00 sec)

mysql> 
```

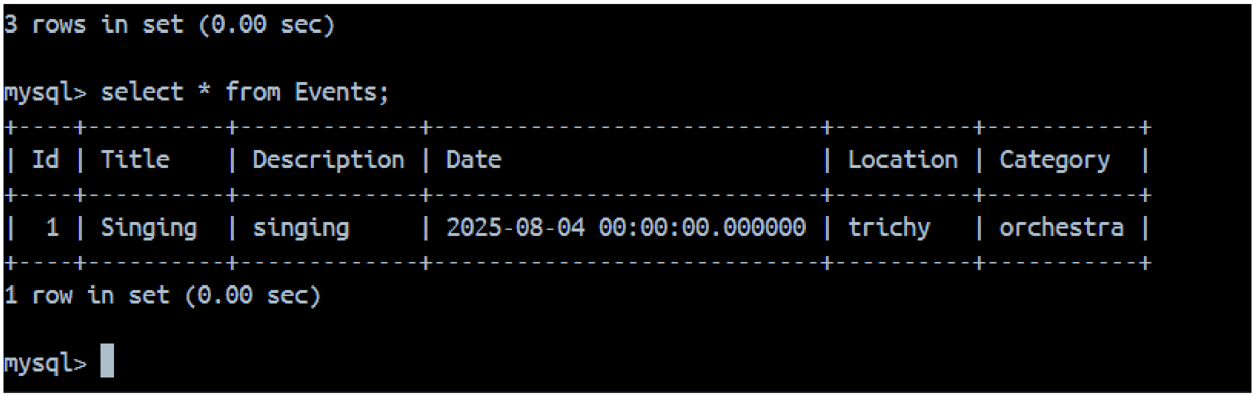
simple Frontend with react through Docker Container



Campus Event Management

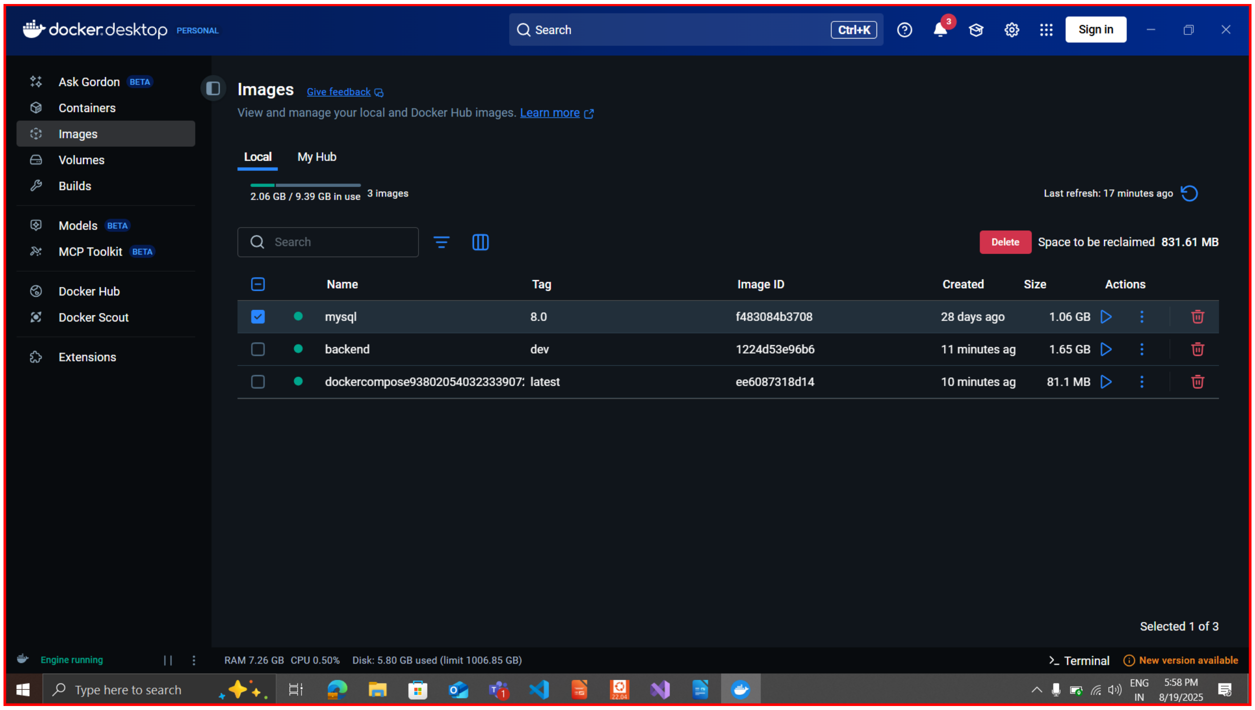
Event List

- **Singing** — 2025-08-04 at trichy [orchestra]
  singing



```
3 rows in set (0.00 sec)

mysql> select * from Events;
+----+---------+-------------+----------------------------+----------+-----------+
| Id | Title   | Description | Date                       | Location | Category  |
+----+---------+-------------+----------------------------+----------+-----------+
|  1 | Singing | singing     | 2025-08-04 00:00:00.000000 | trichy   | orchestra |
+----+---------+-------------+----------------------------+----------+-----------+
1 row in set (0.00 sec)

mysql> 
```

Reflected in Docker



Images Give feedback

View and manage your local and Docker Hub images. Learn more

2.06 GB / 9.39 GB in use 3 images

Last refresh: 17 minutes ago

Space to be reclaimed 831.61 MB

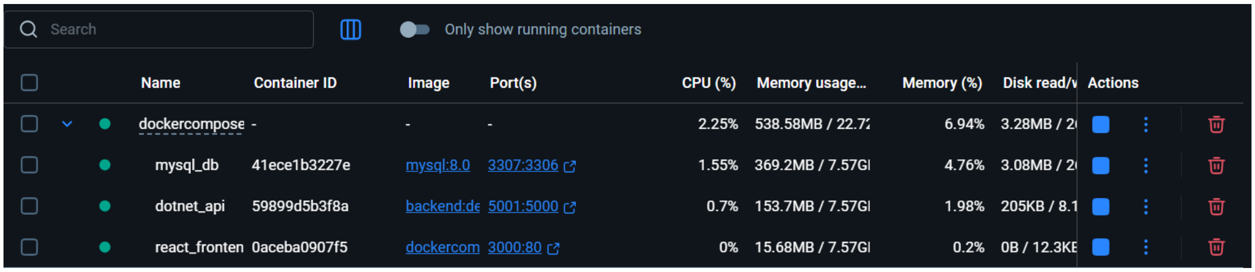
□		Name	Tag	Image ID	Created	Size	Actions
☑	●	mysql	8.0	f483084b3708	28 days ago	1.06 GB	
□	●	backend	dev	1224d53e96b6	11 minutes ag	1.65 GB	
□	●	dockercompose93802054032333907:	latest	ee6087318d14	10 minutes ag	81.1 MB	

Selected 1 of 3

Engine running

RAM 7.26 GB CPU 0.50% Disk: 5.80 GB used (limit 1006.85 GB)

Ip Adrress **192.168.62.173**



□			Name	Container ID	Image	Port(s)	CPU (%)	Memory usage...	Memory (%)	Disk read/w	Actions
□	⌄	●	dockercompose	-	-	-	2.25%	538.58MB / 22.7	6.94%	3.28MB / 2	■ ⋮ 🗑
□		●	mysql_db	41ece1b3227e	mysql:8.0	3307:3306	1.55%	369.2MB / 7.57G	4.76%	3.08MB / 2	■ ⋮ 🗑
□		●	dotnet_api	59899d5b3f8a	backend:de	5001:5000	0.7%	153.7MB / 7.57G	1.98%	205KB / 8.1	■ ⋮ 🗑
□		●	react_fronten	0aceba0907f5	dockercom	3000:80	0%	15.68MB / 7.57G	0.2%	0B / 12.3K	■ ⋮ 🗑

Frontend

http://192.168.62.173:3000/

Api

http://192.168.62.173:5001/

Sql

http://192.168.62.173:3307/