Spring Boot MVC Application Documentation

1. Project overview :

This documentation provides comprehensive guide to setting up, configuration, and running a spring boot MVC application. This guide includes setting up the development environment, configuring the application, and running the project.

2. Prerequisites:

- Java Development kit(JDK)
- Maven
- IDE(eclipse)
- Git
-

3. Setting up project:

1. Add spring initializer to add dependencies - either we can add spring initializer externally or install spring inside eclipse IDE
2. Create New project in eclipse and choose starter project
3. Create artifact id and group id
4. Choose dependencies which is needed for project
   a. Spring web – for developing web or rest API
   b. Spring boot dev tools – for live reload server
   c. MySQL driver - database
   d. Validation – for user input restriction
   e. Lombok – to avoid boiler plate code
   f. Spring data JPA – to connect mysql
   g. Thymeleaf – template engine

4. Configuring the application:

1. After creating a project, to connect spring boot to database add some properties in application properties like database uri,username,password and hibernate for creating a database
2. In main folder create a packages for control, model, service and repository
3. Create a model entity and add some validation to those fields, create repository interface, create service interface and class, create controller class
4. Add exception handling and validation
5. Create frontend templates in static → template folder and connect frontend to backend by using thymeleaf template engine

5. Running the Application:

1. Run the main class as java application or we can also run it by right click on the project and run as spring boot app
2. By default the tomcat server is loaded and type a link in website like “localhost:8080/endpoints” to check our application running.

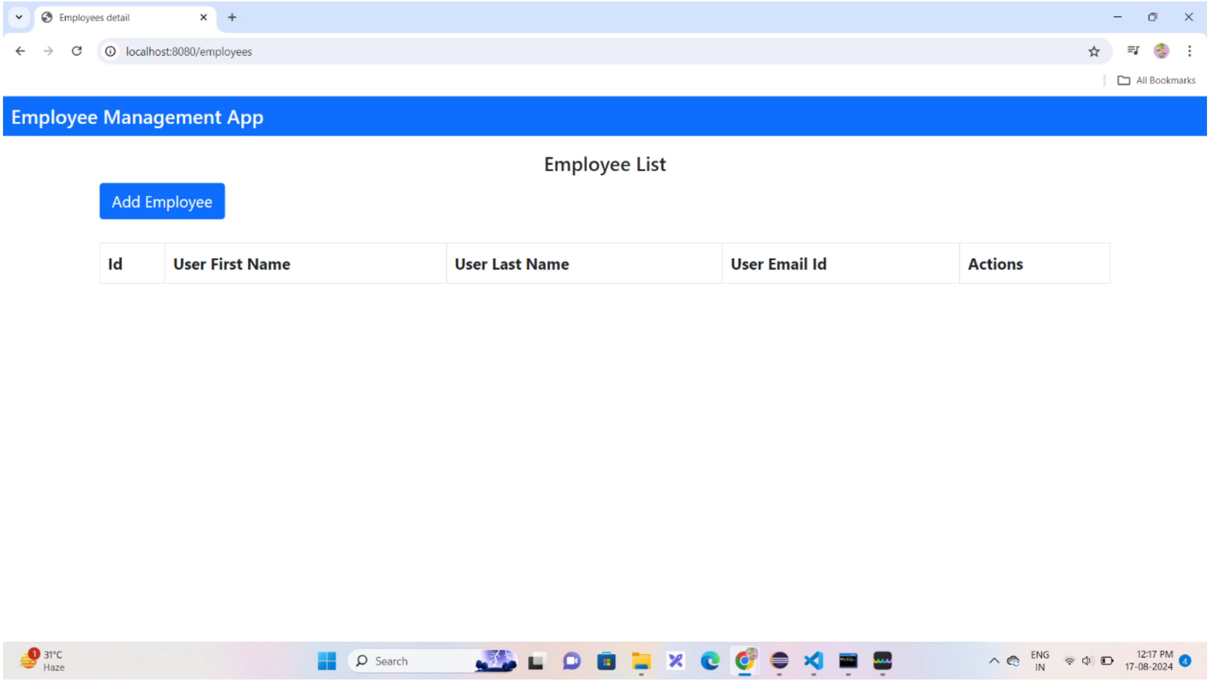
6. Deploying the project using railway:

1. After completing the project, create a deployable link which can be accessed by any user on any server

2. Create an empty project in railway app and connect mysql database, after create another empty project and add project to deploy from github
3. After deploying, generate a link for user access →Here is my project deployed URL: "https://calm-manifestation-production.up.railway.app/employees" ←

Sample picture of the project:

1. End point url →"http://localhost:8080/employees"



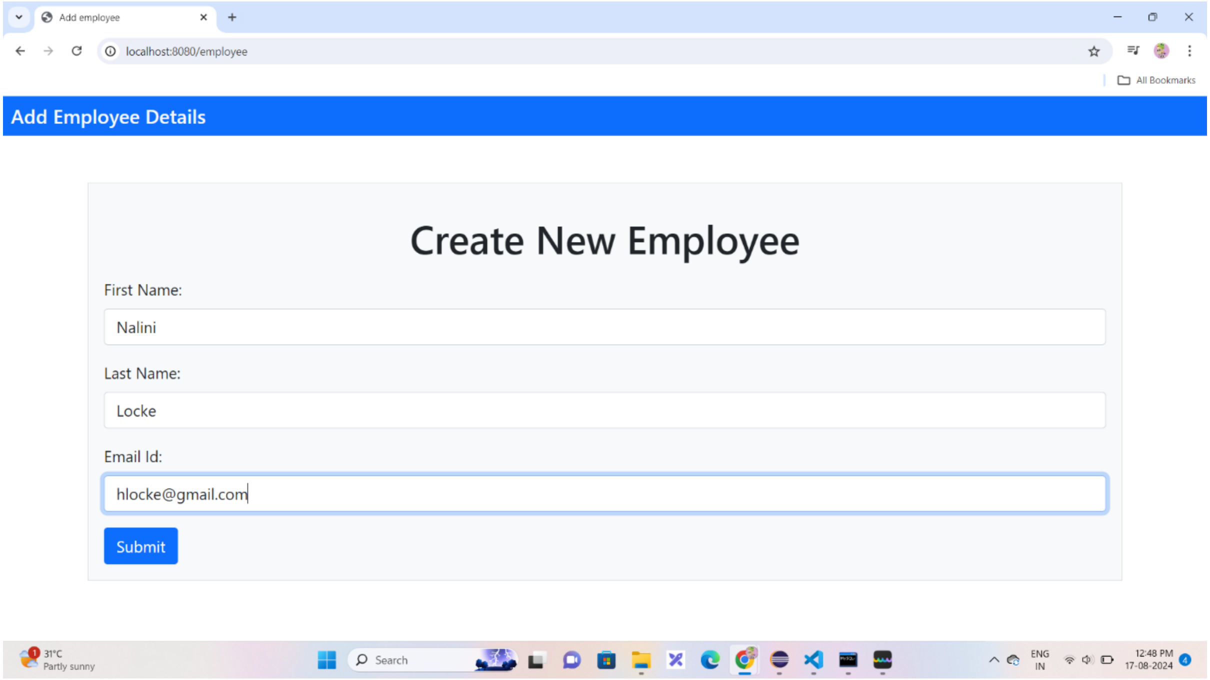
Employee Management App

Employee List

Add Employee

Id	User First Name	User Last Name	User Email Id	Actions

2. End point url →"http://localhost:8080/employee"



Add Employee Details

Create New Employee

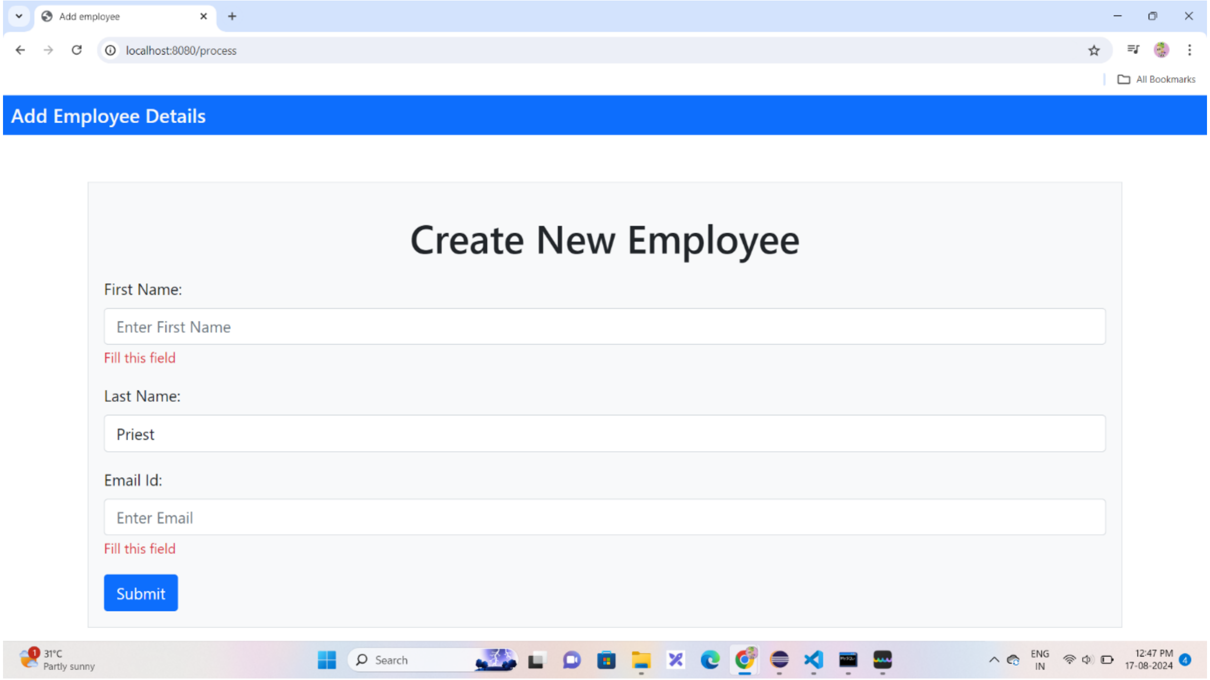
First Name: Nalini

Last Name: Locke

Email Id: hlocke@gmail.com

Submit

3. End point url →" http://localhost:8080/process"



Add Employee Details

Create New Employee

First Name: Enter First Name

Fill this field

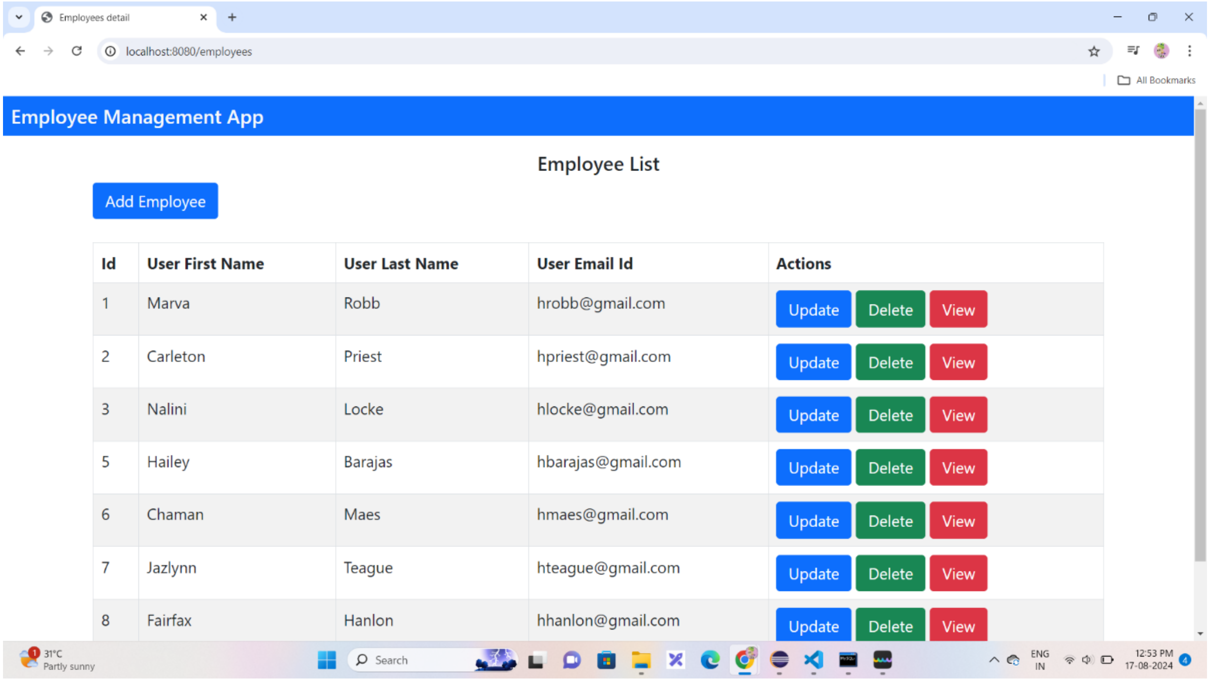
Last Name: Priest

Email Id: Enter Email

Fill this field

Submit

4. End point url →" http://localhost:8080/delete/id"



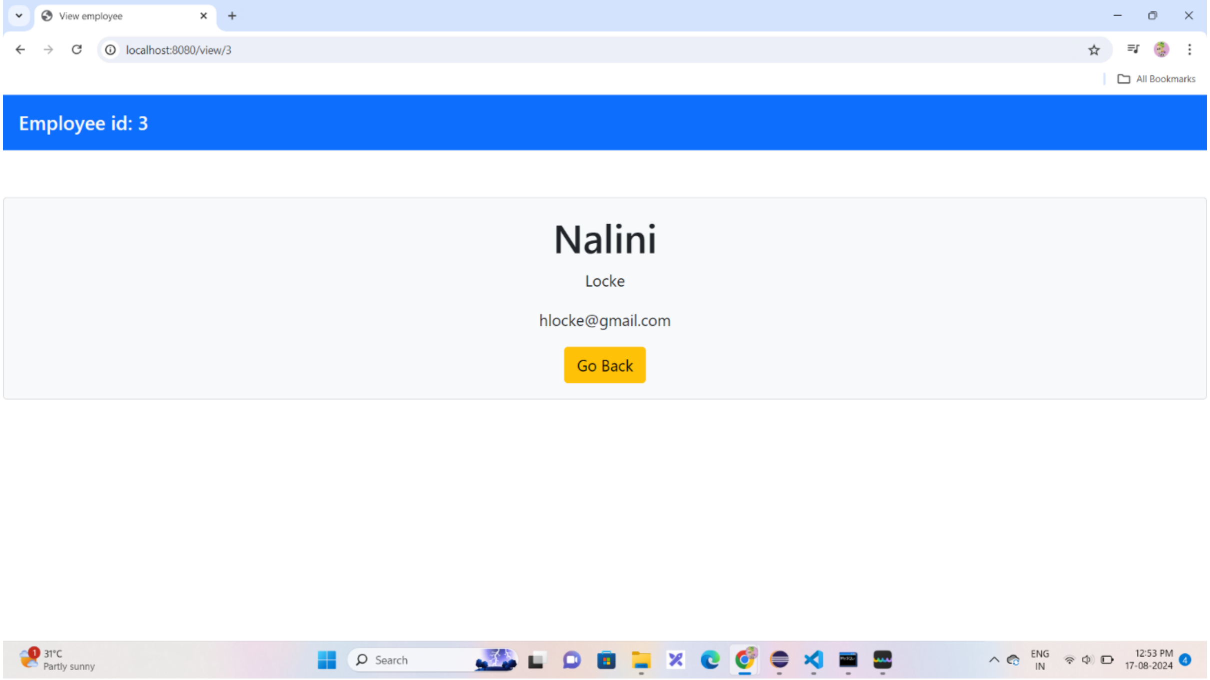
Employee Management App

Employee List

Add Employee

Id	User First Name	User Last Name	User Email Id	Actions
1	Marva	Robb	hrobb@gmail.com	Update Delete View
2	Carleton	Priest	hpriest@gmail.com	Update Delete View
3	Nalini	Locke	hlocke@gmail.com	Update Delete View
5	Hailey	Barajas	hbarajas@gmail.com	Update Delete View
6	Chaman	Maes	hmaes@gmail.com	Update Delete View
7	Jazlynn	Teague	hteague@gmail.com	Update Delete View
8	Fairfax	Hanlon	hhanlon@gmail.com	Update Delete View

5. End point url →” http://localhost:8080/view/id”



Employee id: 3

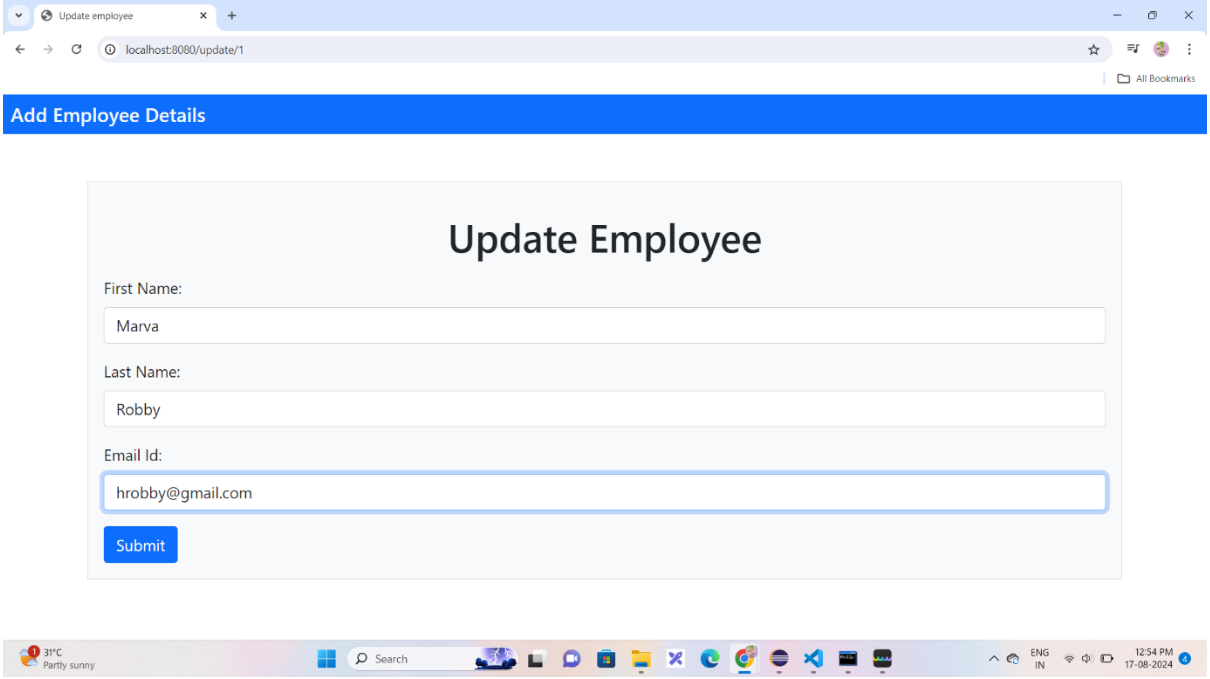
Nalini

Locke

hlocke@gmail.com

Go Back

6. End point url →"http://localhost:8080/update/id"



Add Employee Details

Update Employee

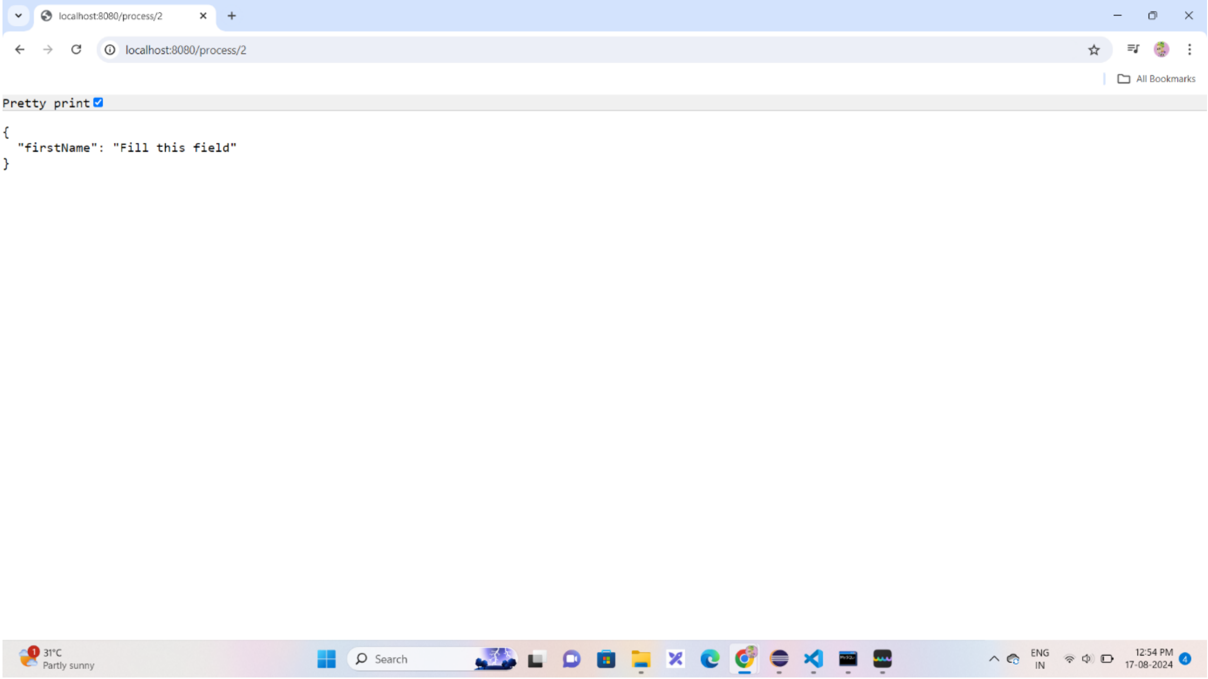
First Name: Marva

Last Name: Robby

Email Id: hrobby@gmail.com

Submit

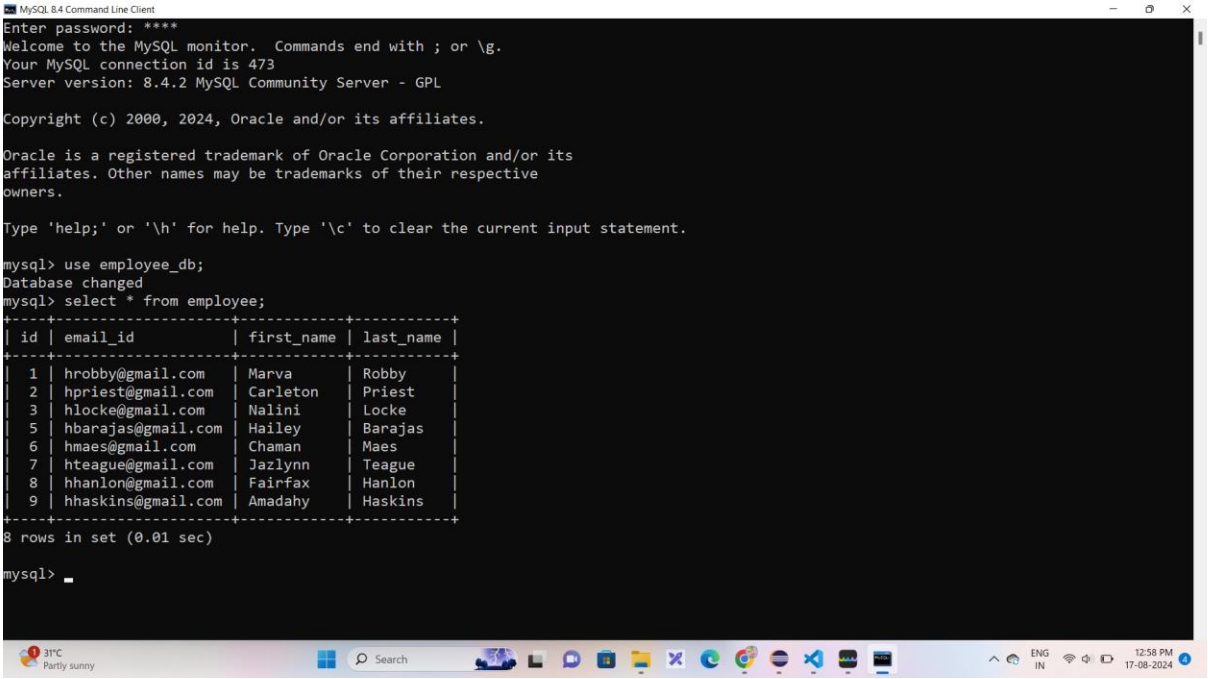
7. Handling errors for user understanding purposes



Pretty print ☑

```
{
  "firstName": "Fill this field"
}
```

8. Data stores in mysql database



```
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 473
Server version: 8.4.2 MySQL Community Server - GPL

Copyright (c) 2000, 2024, 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> use employee_db;
Database changed
mysql> select * from employee;
+----+--------------------+------------+-----------+
| id | email_id           | first_name | last_name |
+----+--------------------+------------+-----------+
|  1 | hrobby@gmail.com   | Marva      | Robby     |
|  2 | hpriest@gmail.com  | Carleton   | Priest    |
|  3 | hlocke@gmail.com   | Nalini     | Locke     |
|  5 | hbarajas@gmail.com | Hailey     | Barajas   |
|  6 | hmaes@gmail.com    | Chaman     | Maes      |
|  7 | hteague@gmail.com  | Jazlynn    | Teague    |
|  8 | hhanlon@gmail.com  | Fairfax    | Hanlon    |
|  9 | hhaskins@gmail.com | Amadahy    | Haskins   |
+----+--------------------+------------+-----------+
8 rows in set (0.01 sec)

mysql> _
```