**Employee Management Application**

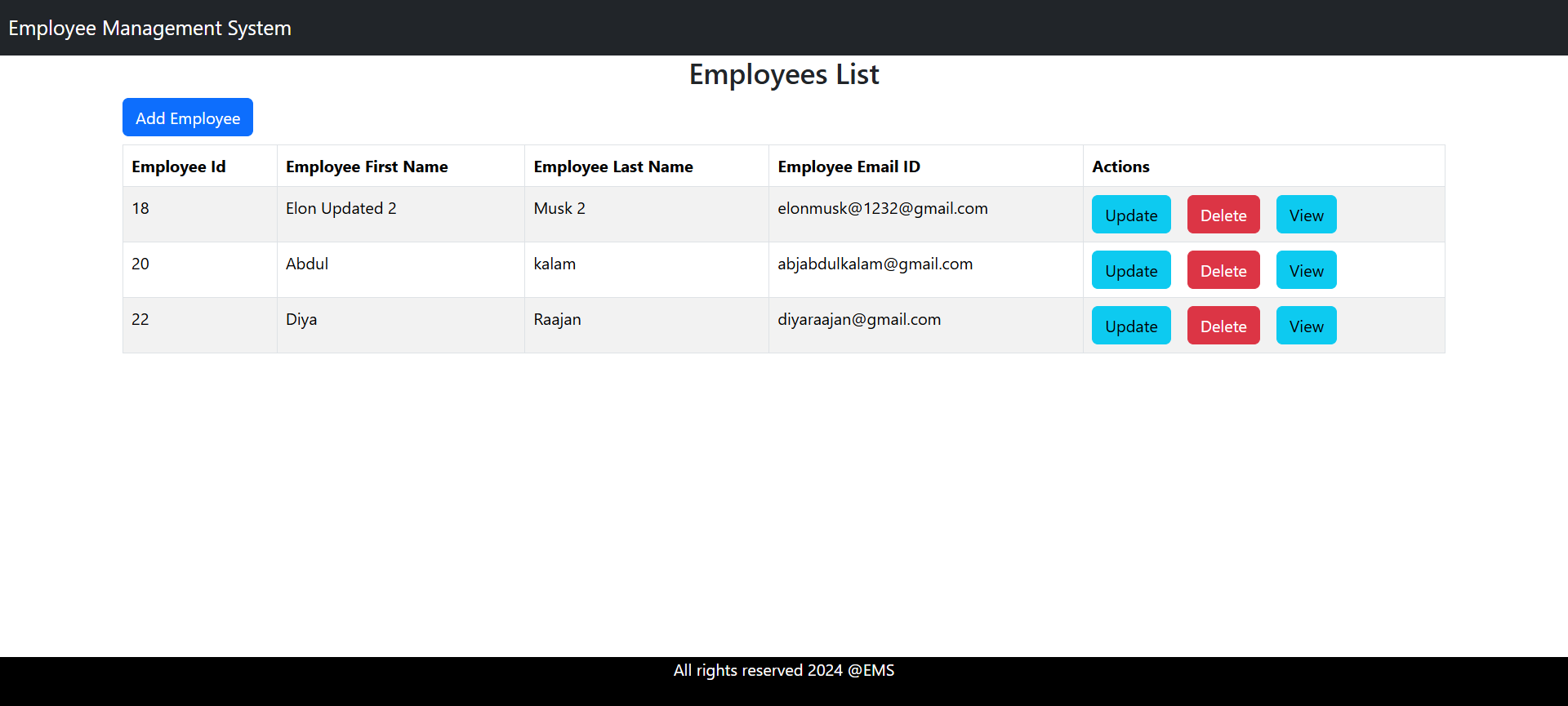
**Overview: -**

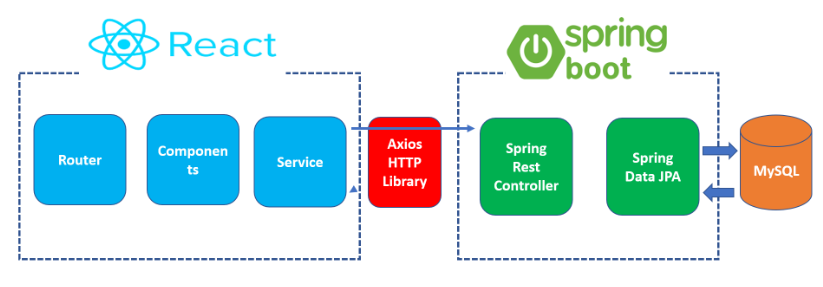
I have built the full-stack web application that is a basic **Employee Management Application with**belowCRUD operations.

* Create Employee
* List Employee
* Update Employee
* Delete Employee
* View Employee

Here, I developed the website using **Spring boot and React Js.** Attached below is final version of **Employee Management Application** when you search this below URL in your browser after running my project in your local machine.

[**http://localhost:3000**](http://localhost:3000)



**ReactJS + Spring Boot CRUD Full Stack Architecture:** 

**Full Stack App Development:**

I build two separate projects:

1. **sprint boot-backend (server)** – To develop REST API’s and It will be exposed from backend by Spring REST Controller
2. **react-frontend (client)** – Consumes REST API’s through AXIOS HTTP Client Library

**Tools and Technologies**

**Server-side tools and technologies used**

* IntelliJ Idea IDE
* Spring Boot v3.3.1
* MySQL Database v8.3
* Maven v4.0
* JDK v17
* Open API (Swagger) for Documentation
* Postman

**Client-side tools and technologies used**

* Visual Studio Code IDE
* React JS v18
* Vite JS 🡪 Build Tool 🡪 Provides dev server :5173
* Modern JavaScript (ES6)
* NodeJS
* NPM
* Bootstrap 5
* Axios HTTP Client Library

**Full Stack Project Structure:-**

**Backend Project Structure:- [Project Name - EMS-Backend]**

* src/main/java: Contains the Java source files.
* src/main/resources: Contains application properties files.
* pom.xml: Maven configuration file.

**Frontend Project Structure:-[Project Name -EMS-Frontend]**

* src/components: Contains the React components and related files.
* src/service : Contains the configuration of Axios Http Client Library
* src/App.jsx : Contains the Routes file
* package.json: npm configuration file.
* vite.config.js : Vite configuration file.

**Project setup methods:**

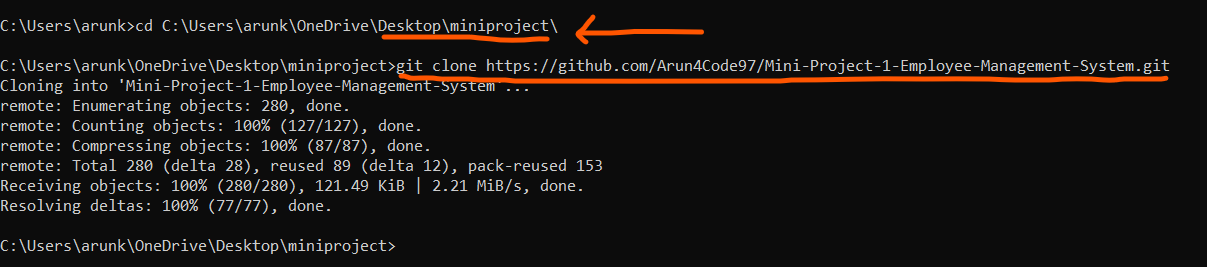
**Step 1: Clone the project repository which I have mentioned in Zen project submission portal:**

1.Goto command prompt and ensure you have created a dedicated directory. In my case, Folder name is **miniproject.**

cd C:\Users\arunk\OneDrive\Desktop\miniproject

2.Ensure you are located inside the minipoject folder and issue the below command for cloning the repository

git clone https://github.com/Arun4Code97/Mini-Project-1-Employee-Management-System.git

****

**Way 1: Working in single IDE (Not recommended)**

**Quick overview:**

We can open both back end and front project in same IDE as a Separate modules, In this case, You can open any IDE and clone the repository using below command using git bash and ensure you have required tools and technologies for this project.

Goto any terminal, **locate to specific frontend /backend using #cd command and run the project** and ensure you use separate command prompt window for both after changing few configuration details in both projects which is listed below.

**For running backend project,**

# PS C:\Your cloned repository folder path \Mini-Project-1-Emploee-Management-System\backend> mvn spring-boot:run

**For running frontend project,**

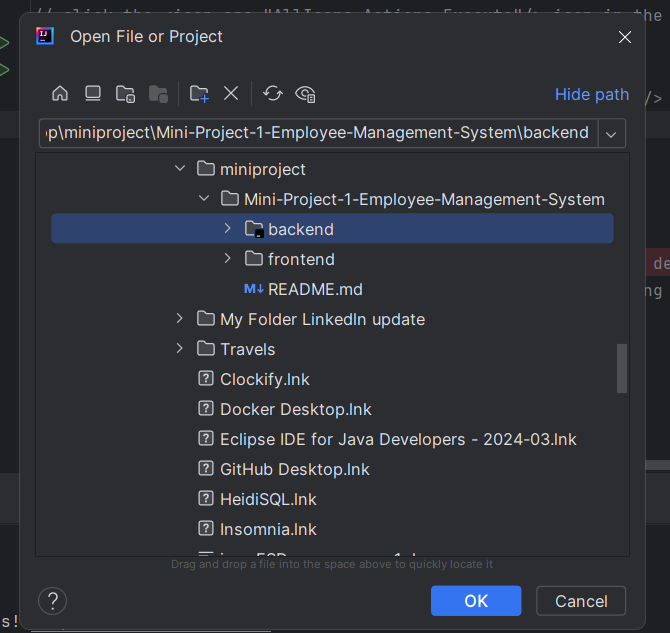
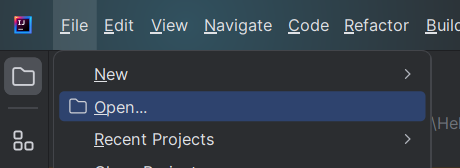
# C:\ Your cloned repository folder path \Mini-Project-1-Emploee-Management-System\frontend> npm install && npm run dev

**Note:** But for better understanding I am suggesting to open my front end and back end in separate IDE’s.

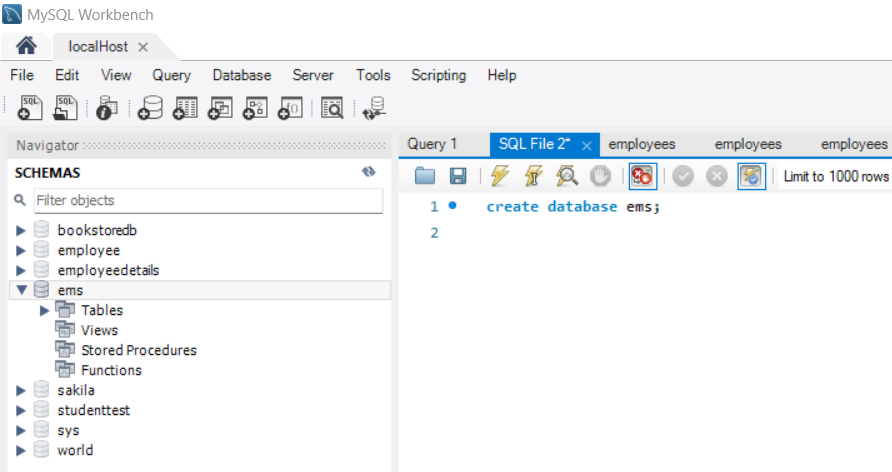
**Way 2: Working in different IDE.**

**Step 2: Backend Setup:-**

1. **Opening project** - Goto intellij, In the left top corner, click on open project, select the backend folder and click on **“Ok”**



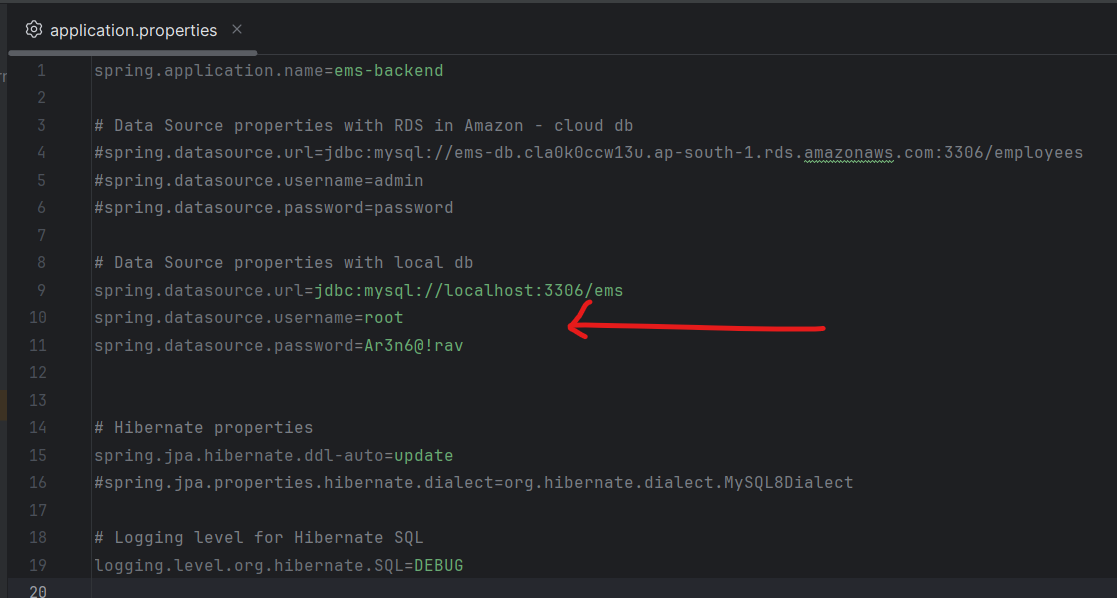
1. **Creating database** -Before doing configuration in the project, need to create a database in MySQL server (MySQL Workbench) with the following command:



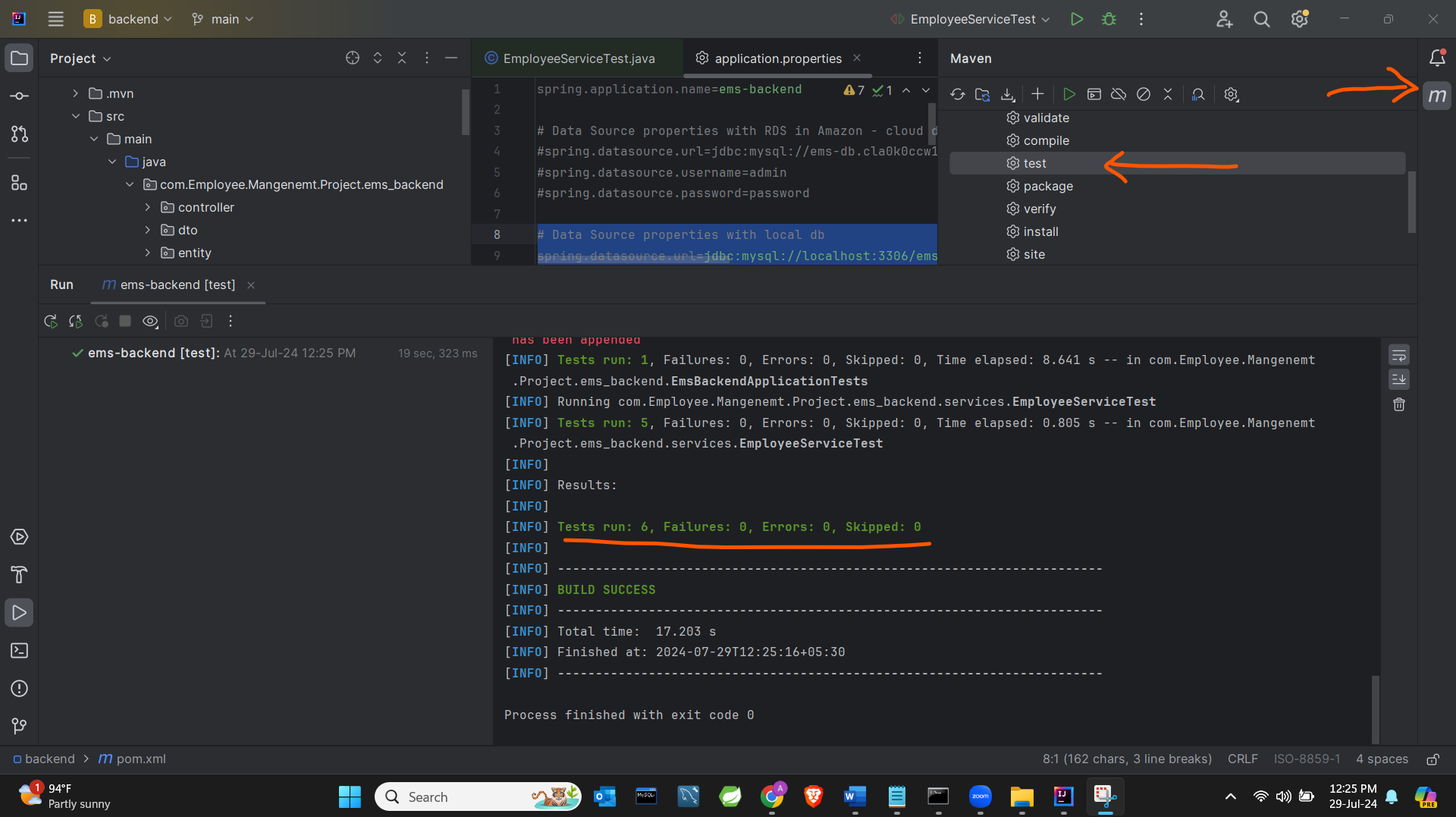
1. **Configuring MySQL Database -** configure *application.properties* or application.yml to connect to your MySQL database.

In my case, The database name is **ems** and MySQL username and password properties set as below.

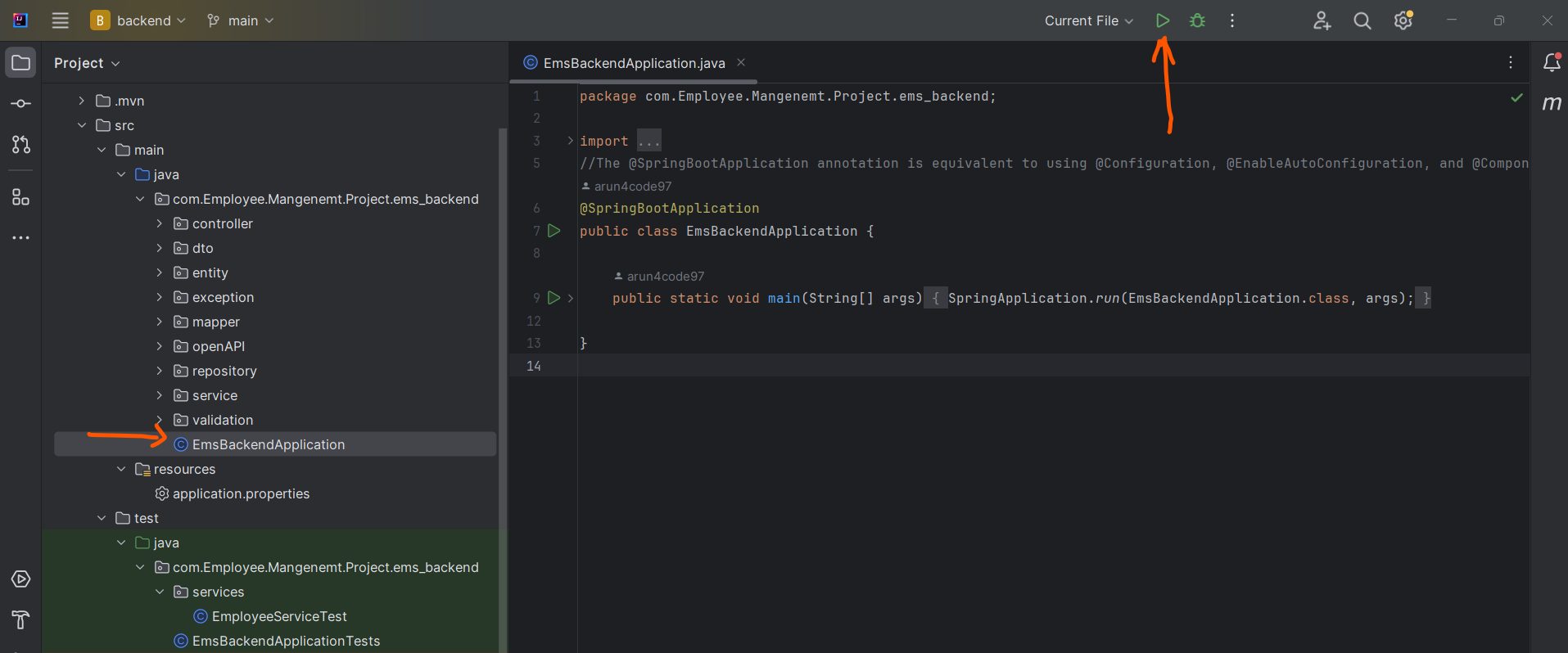
spring.datasource.url=jdbc:mysql://localhost:3306/Your database name  
spring.datasource.username=your user name  
spring.datasource.password=your password

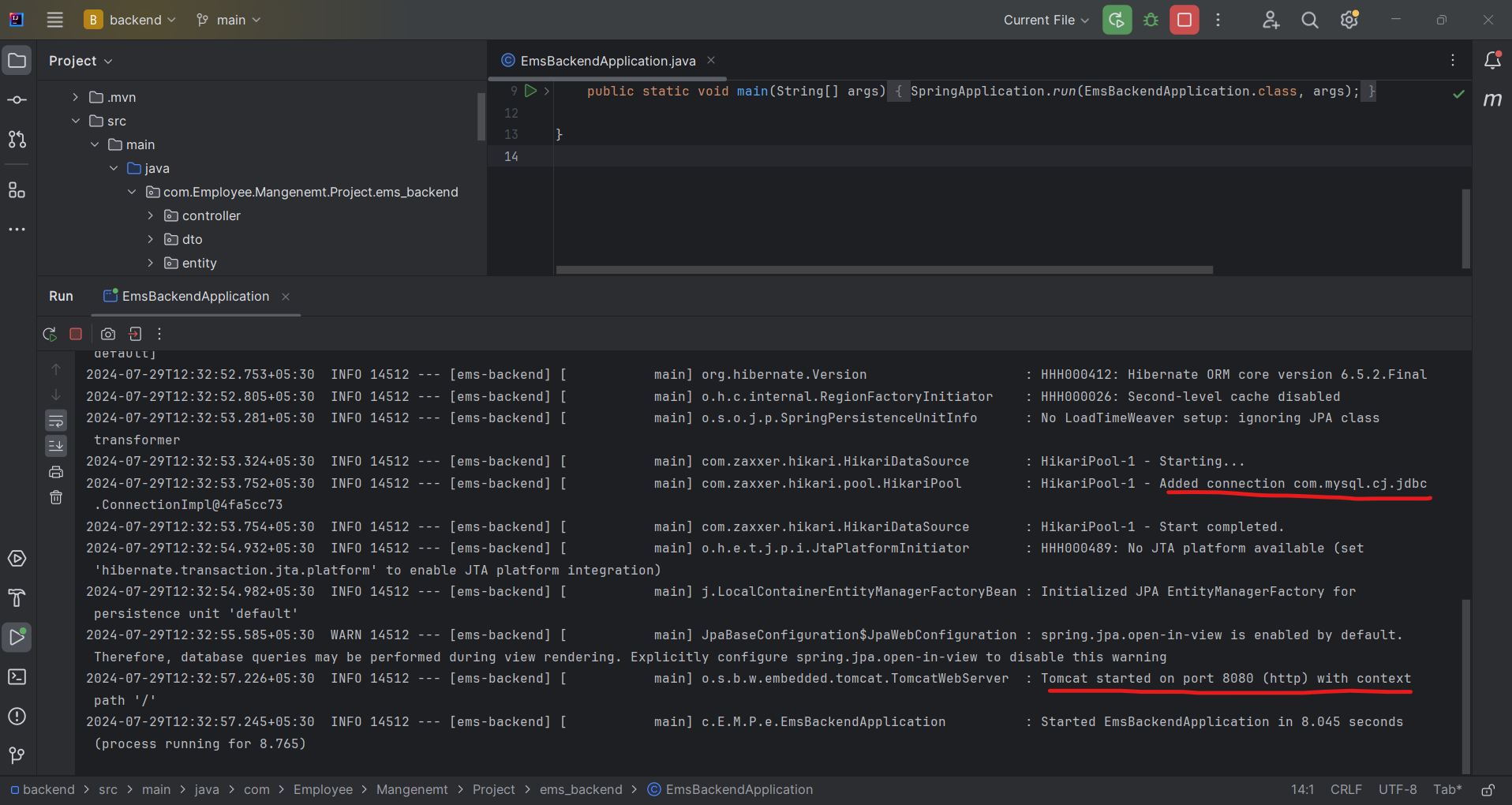


1. **Running the unit testing** of Service layer using Maven test and ensure all 6 test files are running successfully.



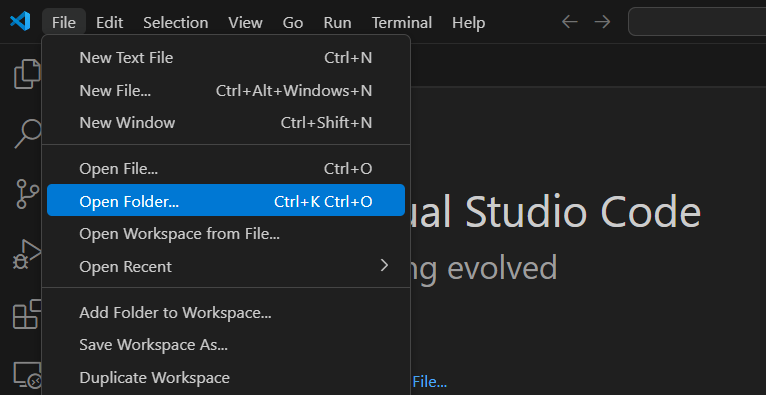
1. **Running the EmsBackendApplication.java** and ensure application is running on default port number 8080.

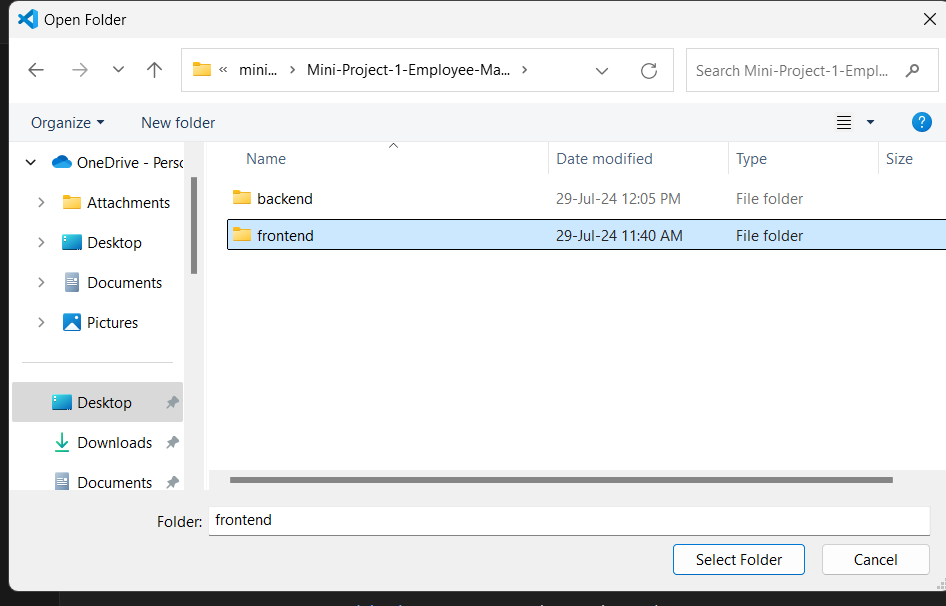




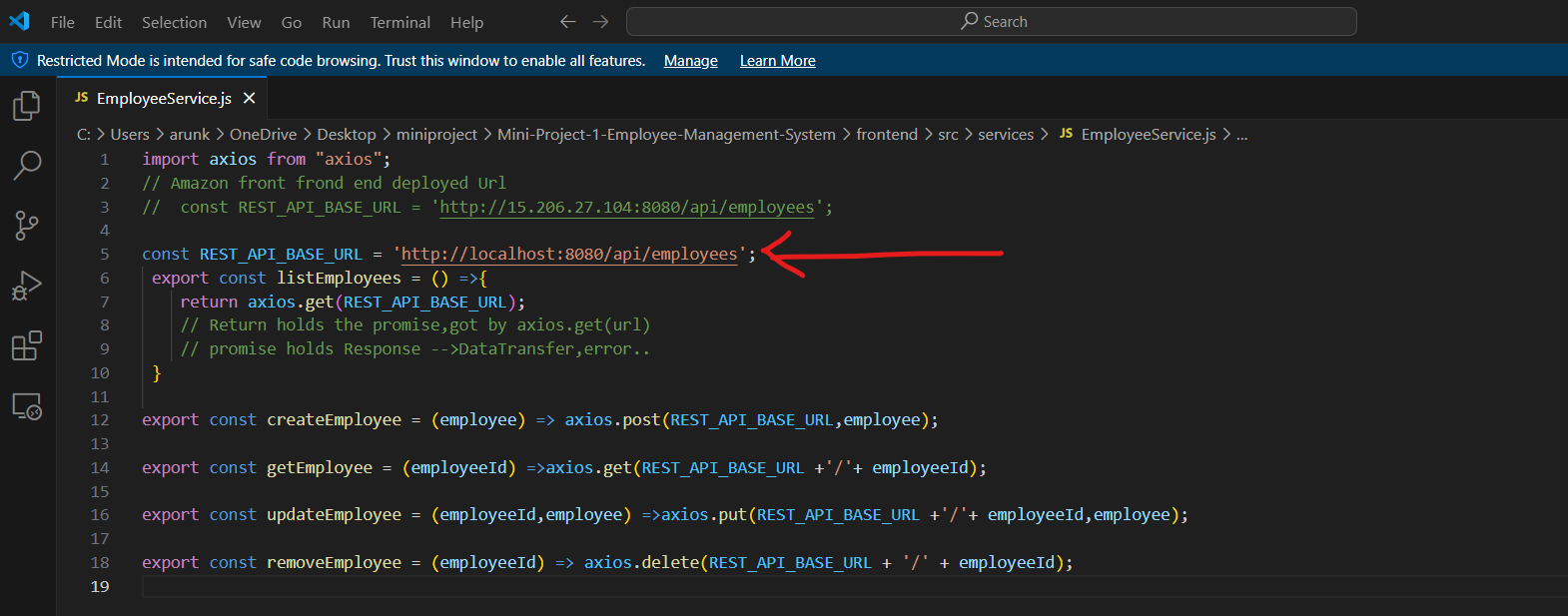
**Step 3: Frontend Setup:-**

1. **Opening the frontend project in VScode IDE**: Go to Vs studio code and click on Open Folder . Then select frontend folder in your cloned directory.

****

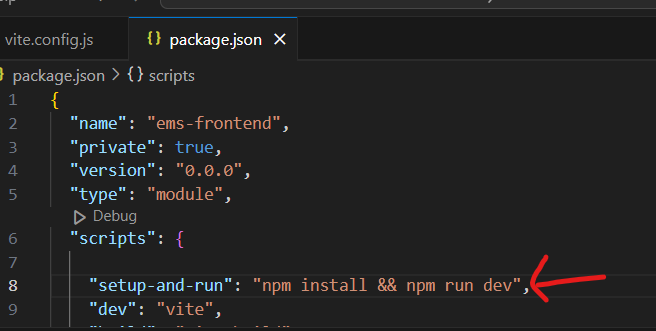
****

2. Go to **EmployeeService.js file** and **ensure the API Base URL is configured correctly** so that HTTP client library will consume our backend APIs properly.



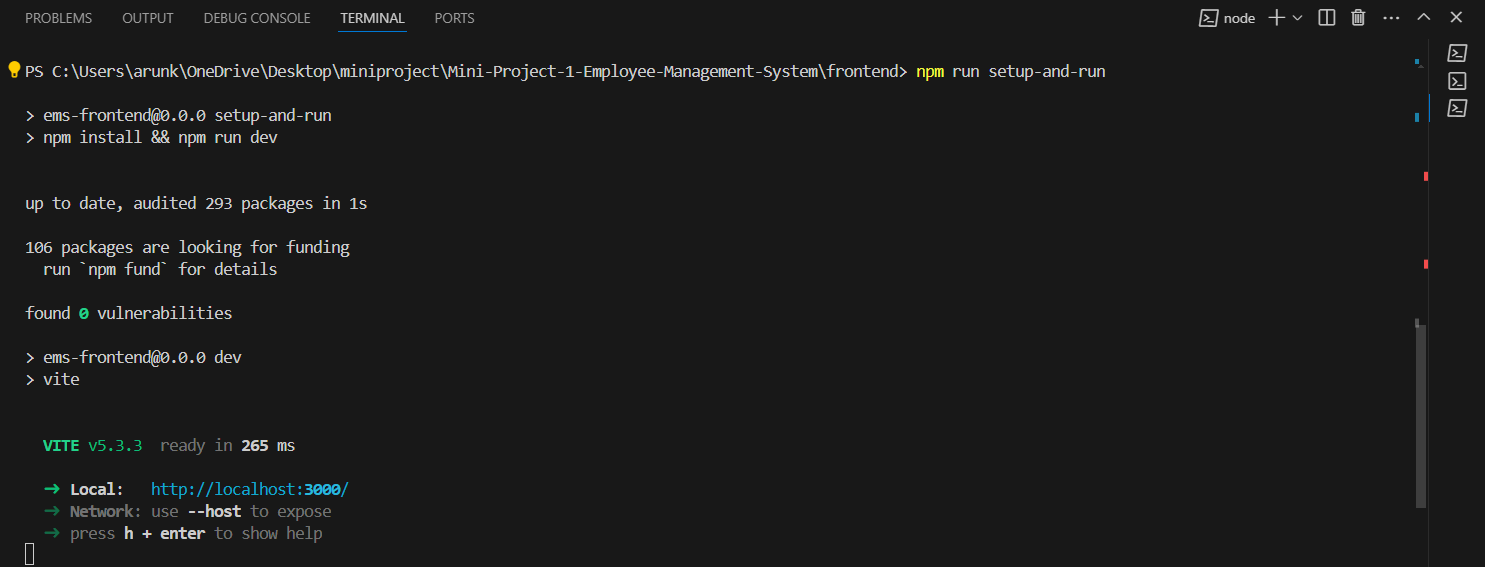
3. **Running the project** - Go to Terminal and issue below command.

**# npm run setup-and-run.**

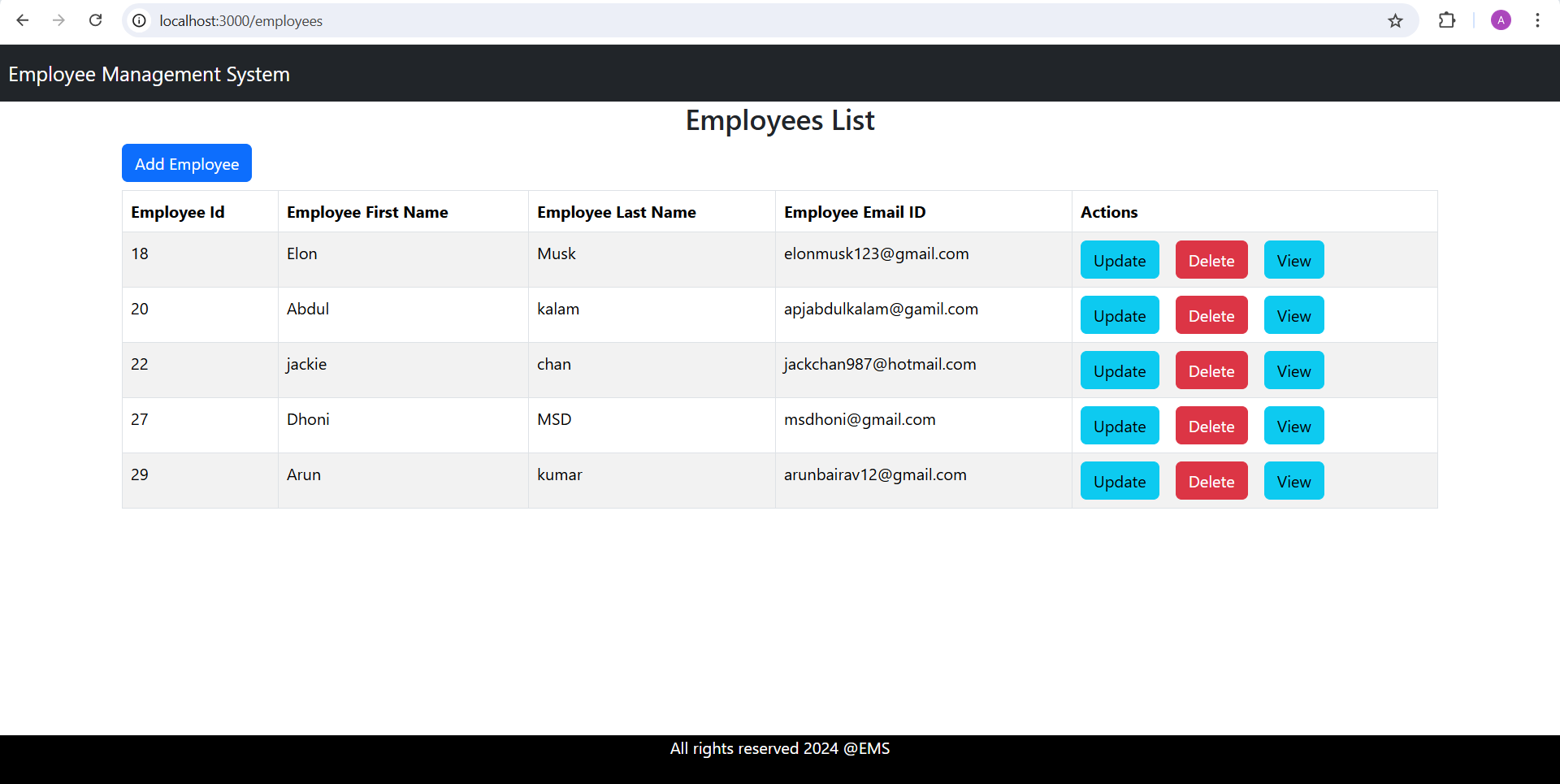
****

Here, npm runs setup-and-run script in single go to install all project dependencies and run the project in development server.

1. **Ensure application is successfully running** in port number :3000



1. **Access the application -** Open your browser and navigate to http://localhost:3000 to use the application.



**Troubleshooting: -**

Common Errors in Backend:

* **Could not connect to database:** Ensure your database is running and the credentials are correct.
* **Port already in use:** Change the port in application.properties using server.port.

Common Errors in Frontend

* **API request failed:** Ensure the backend is running and the API URL is correct.
* **Port already in use:** Change the port in vite.config.js