Assignment-1

Employee management system

**Problem Statement:**

Develop a Single Page Web Application using Angular with backend as NodeJS and DB like MySQL DB or Mongo DB and perform CRUD operations

1. Create an employee record

2. Read all employee records

3. Update an employee record

4. Delete an employee record

**Solution:**

**Description:**

The **Employee Management System** is a full-stack web application built using Angular and Node.js. It provides functionalities for managing employees within an organization. The application supports **Admin** and **Employee** roles.

* **Admin** users can view all employees, add new ones, edit any employee’s details, and delete employees.
* **Employee** users can only **view and edit their own profiles**.

It includes authentication, role-based access, CRUD operations, and MongoDB integration.

**Tech Stack:**

|  |  |
| --- | --- |
| Layer | Technology |
| Frontend | Angular 19, Bootstrap 5 |
| Backend | Node.js, Express.js |
| Database | MongoDB |
| Authentication | Basic login via email/password |
| API Communication | HTTPClient (Angular) |
| Dev Tools | VS Code, MongoDB Compass |

**Directory Structure:**

employee-management/

├── employee-management-backend/

│ ├── controllers/

│ │ └── employeeController.js # Handles CRUD operations and login

│ ├── models/

│ │ └── employeeModel.js # Mongoose schema for Employee

│ ├── routes/

│ │ └── employeeRoutes.js # Defines API routes

│ ├── server.js # Express app entry point

│ └── .env # DB URI, port config (optional)

│

├── employee-management-frontend/

│ ├── src/

│ │ ├── app/

│ │ │ ├── components/

│ │ │ │ ├── navbar/

│ │ │ │ │ └── navbar.component.ts/html/css

│ │ │ │ ├── employee-list/

│ │ │ │ │ └── employee-list.component.ts/html/css

│ │ │ │ ├── employee-form/

│ │ │ │ │ └── employee-form.component.ts/html/css

│ │ │ │ ├── nonemployee-list/

│ │ │ │ │ └── nonemployee-list.component.ts/html/css #List of employees for non-admins

│ │ │ │ ├── profile/

│ │ │ │ │ └── employee-profile.component.ts/html/css #Profile of employee

│ │ │ │ ├── login/

│ │ │ │ │ └── login.component.ts/html/css

│ │ │ ├── services/

│ │ │ │ └── employee.service.ts # HTTP calls to backend

│ │ │ ├── app.component.ts

│ │ │ ├── app.module.ts

│ │ │ └── app.routes.ts

│ │ └── main.ts

│ └── angular.json

│

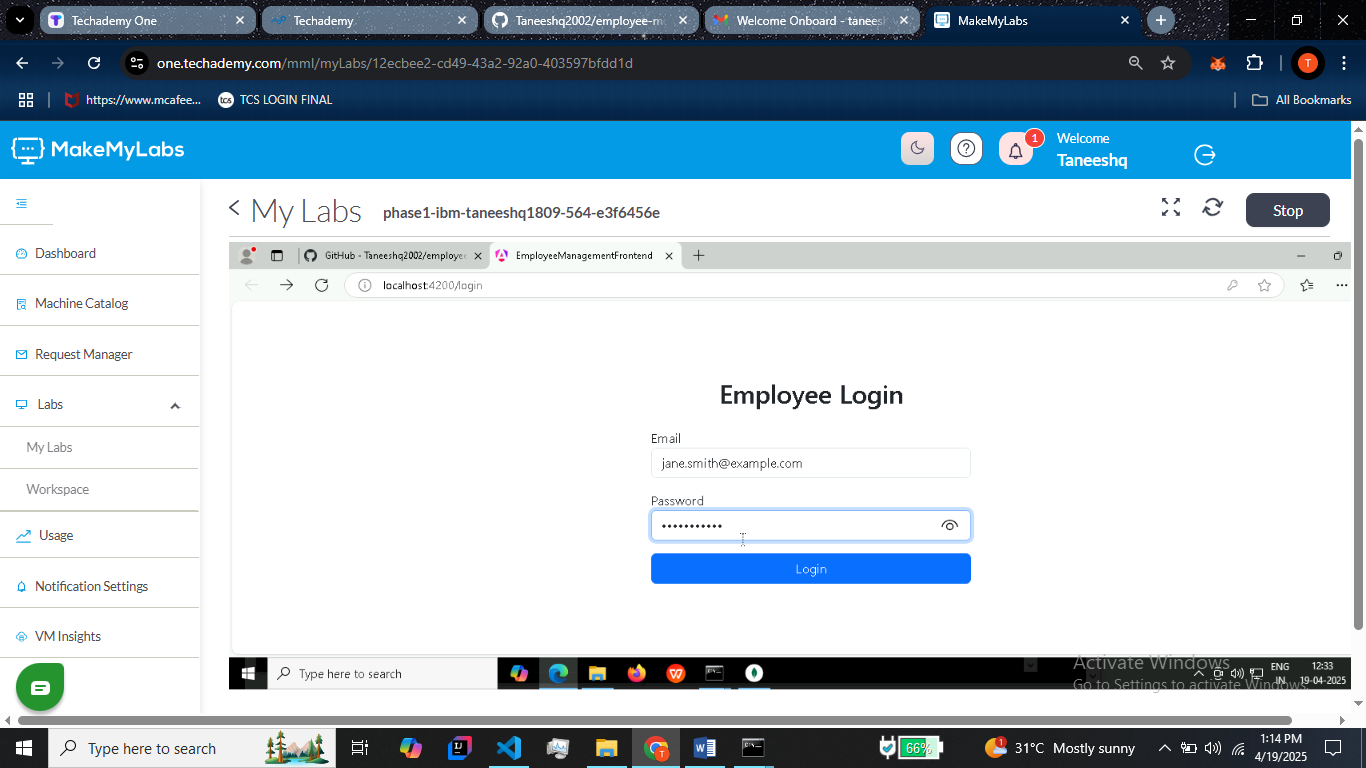
├── package.json

├── README.md

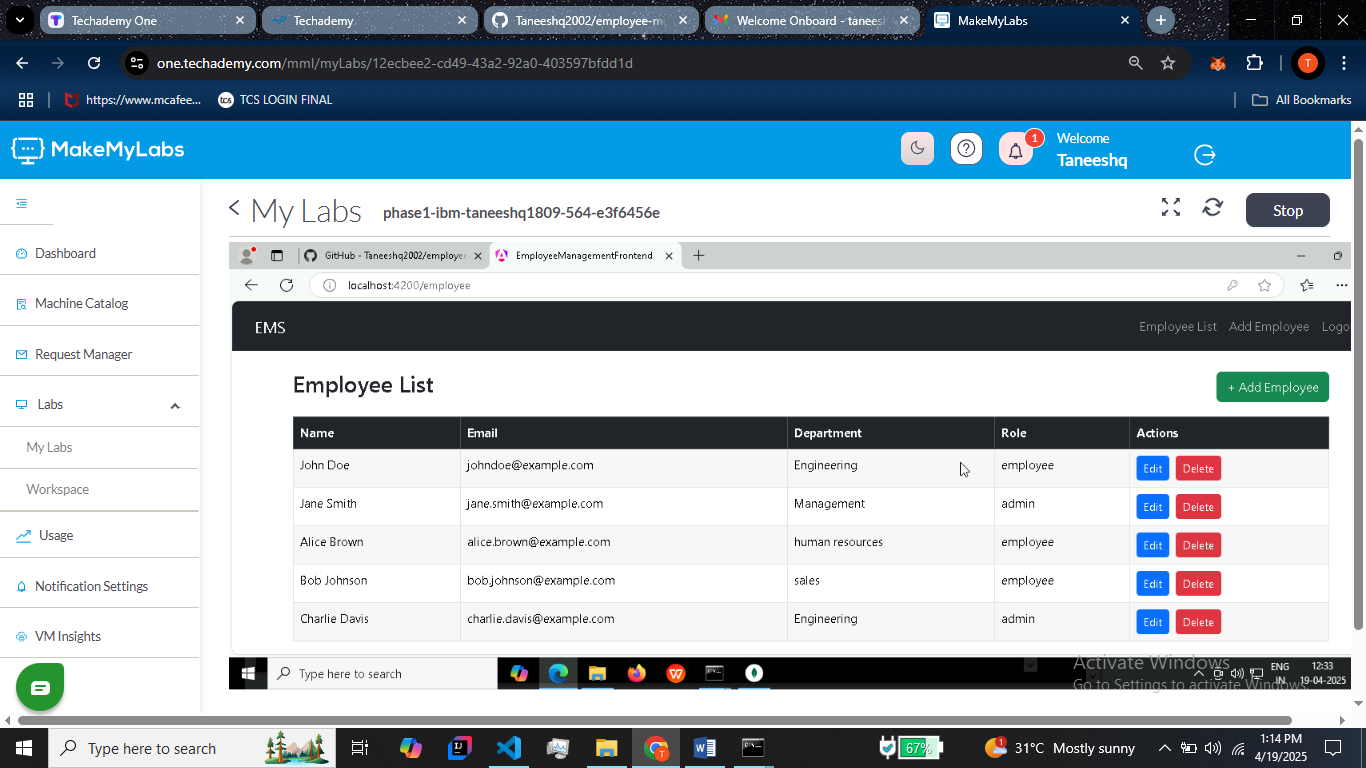
└── node\_modules/

**Output:**

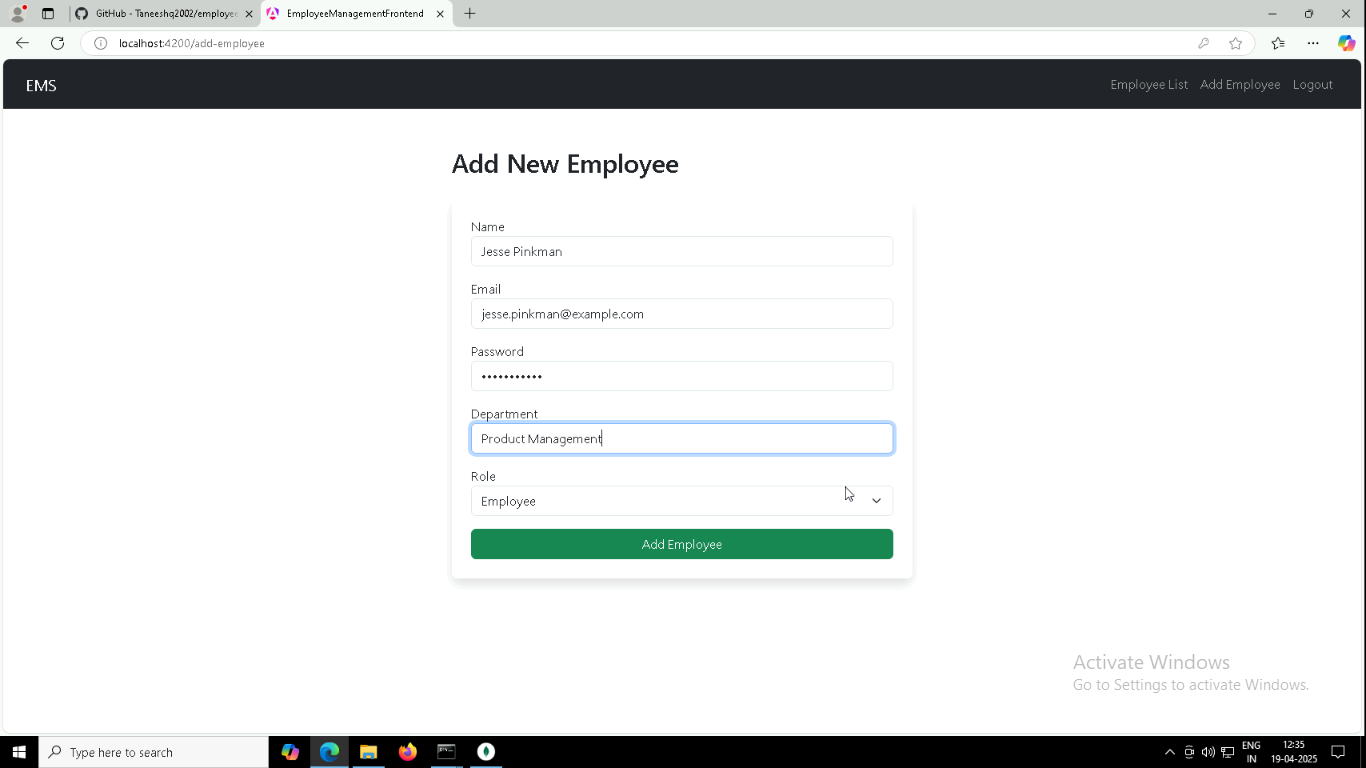
**1)Admin login:**

****

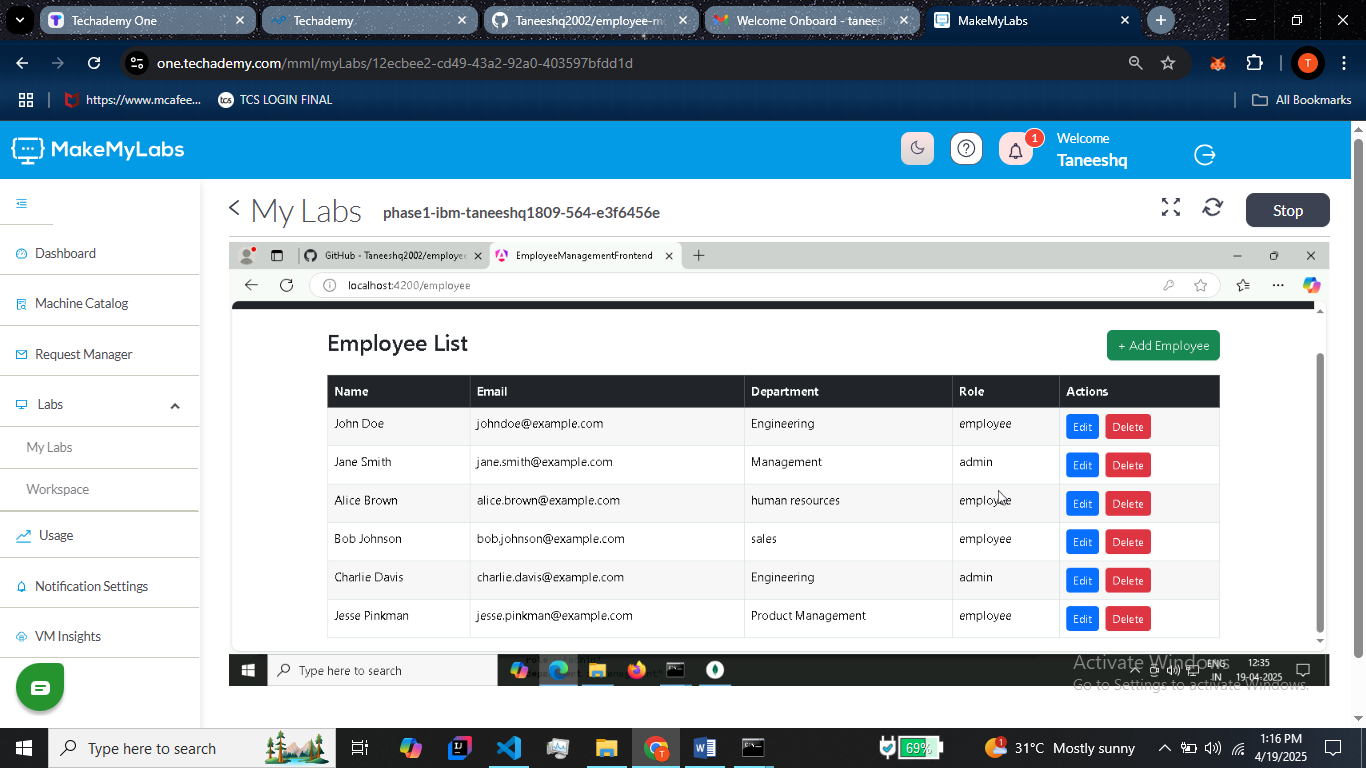
**2) List of employees for admin POV**

****

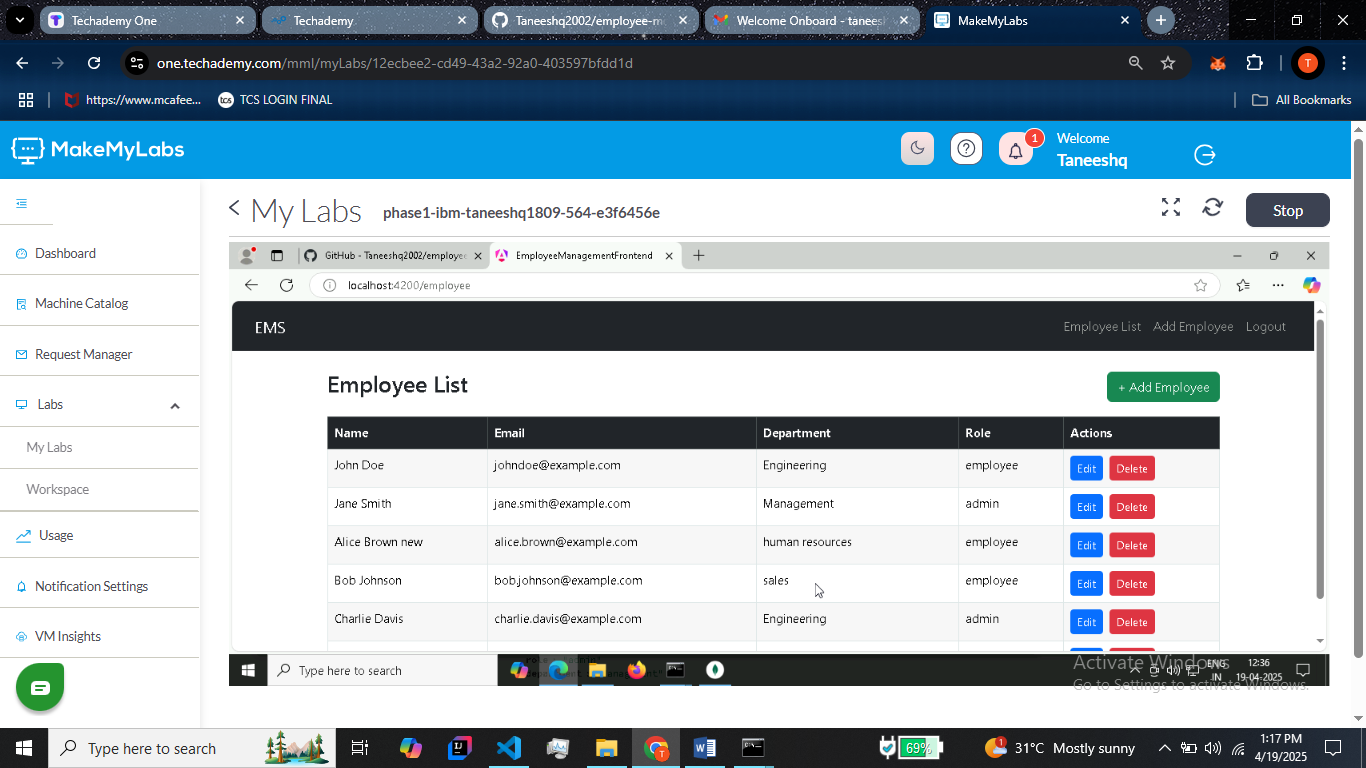
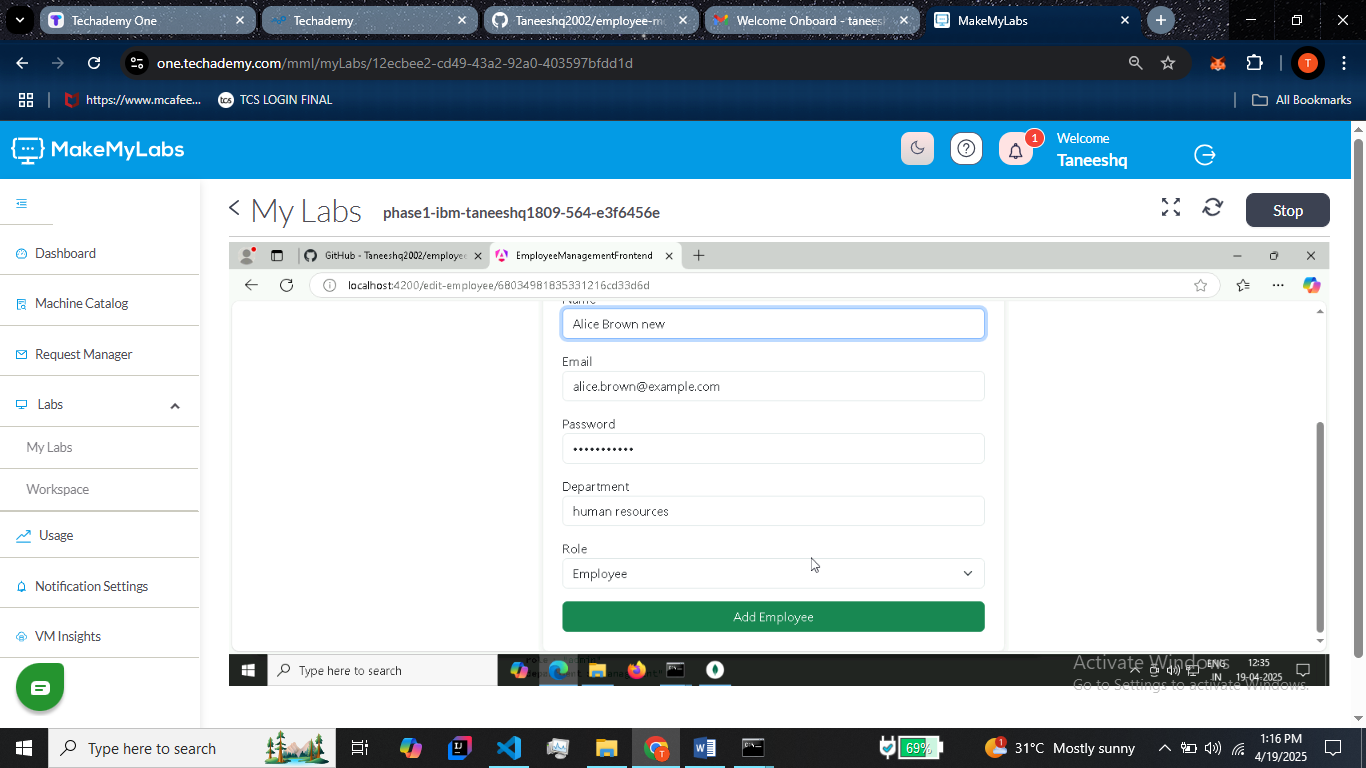
**3)Adding new employee:**

****

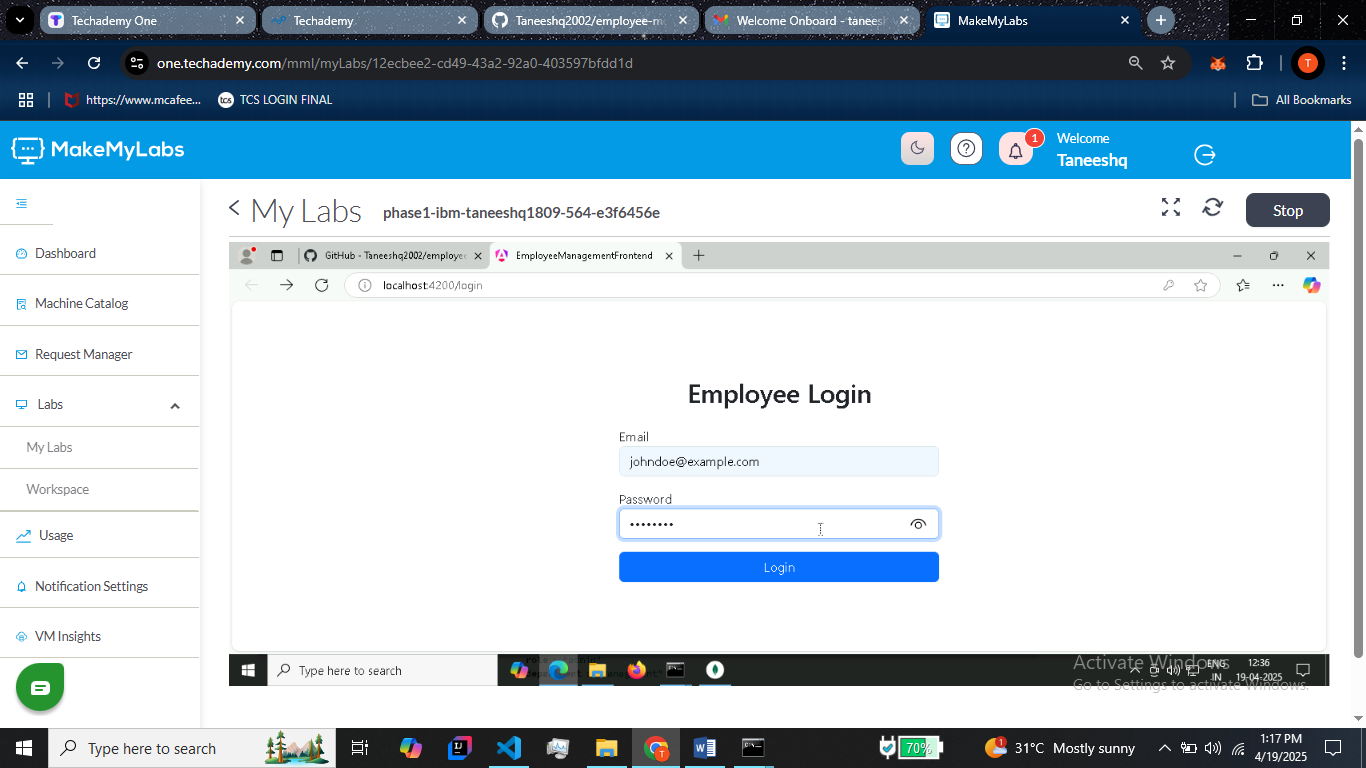
**4) Updated list**

****

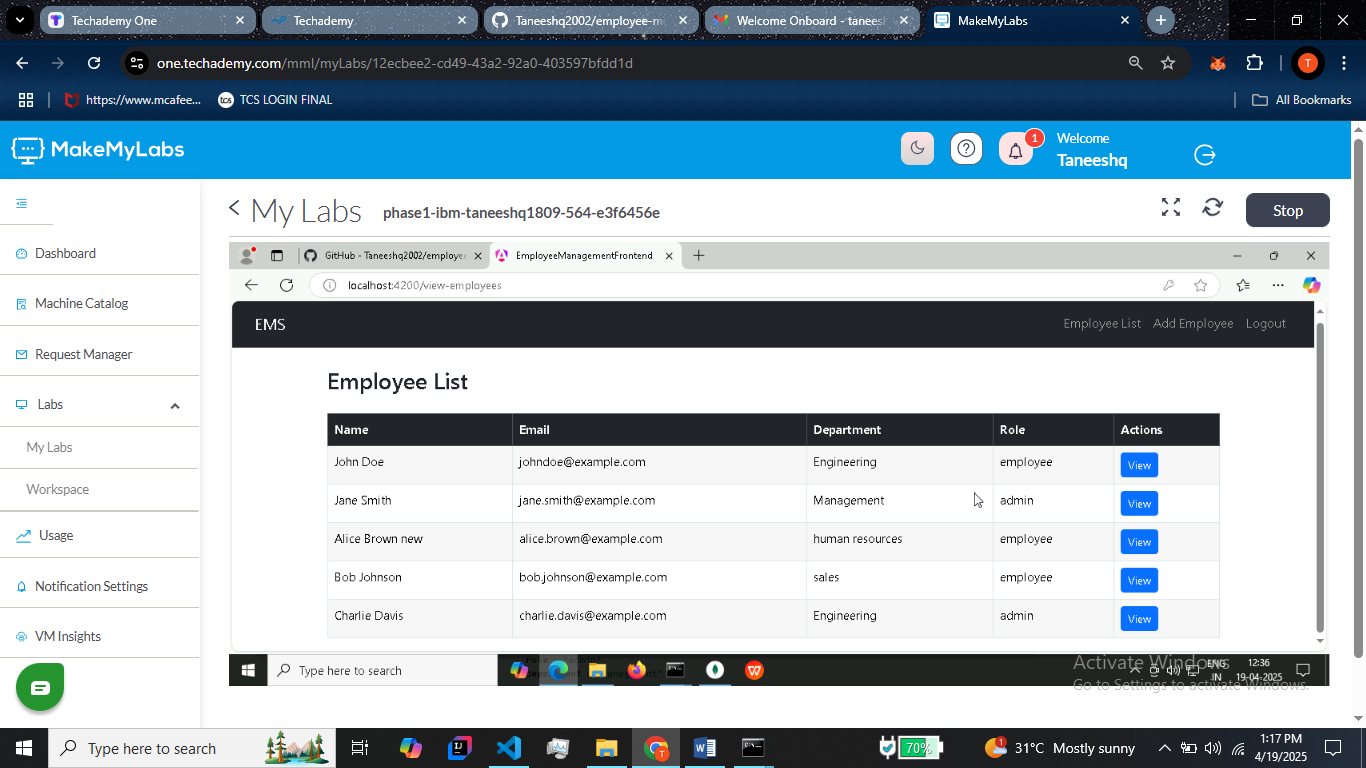
**5) Editing employee details(/edit-employee)**

****

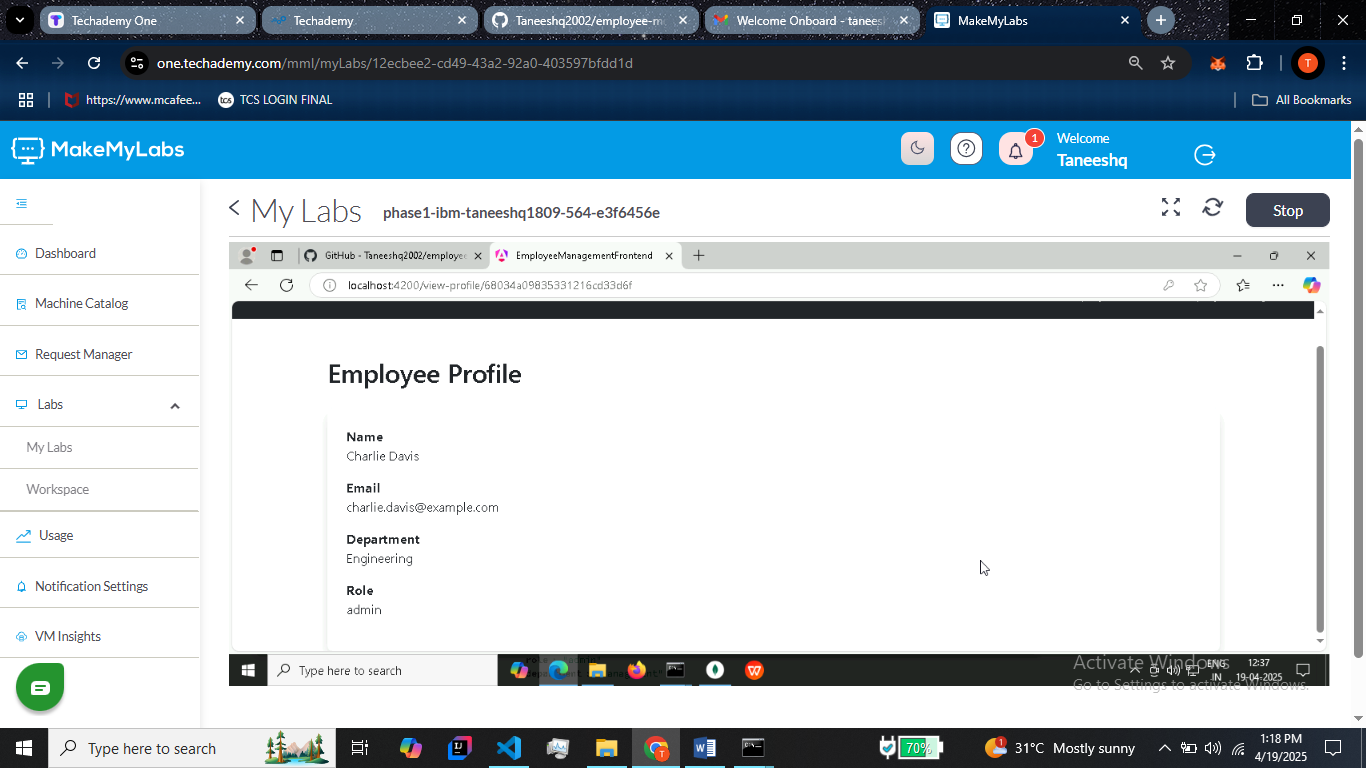
**6)Login for non-admin employees:**

****

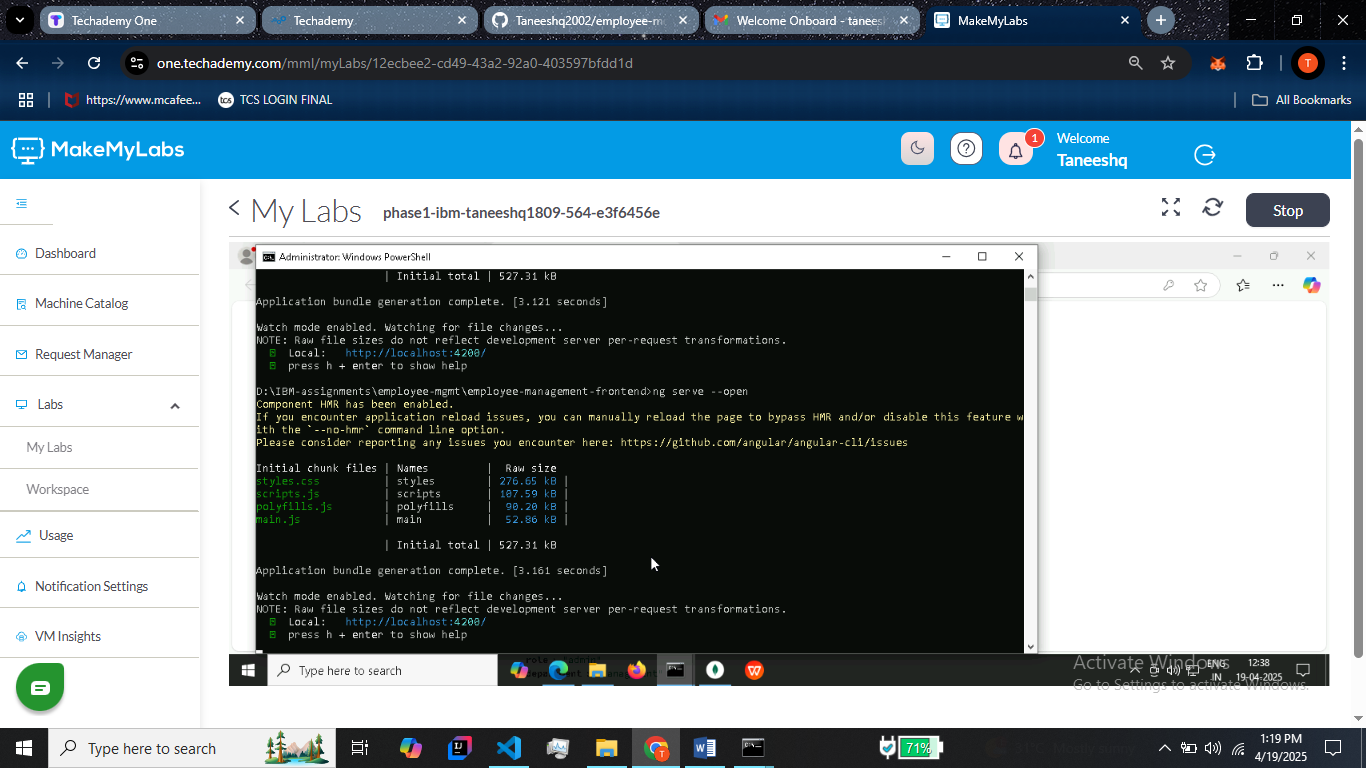
**7) List of employees for non-admin POV**

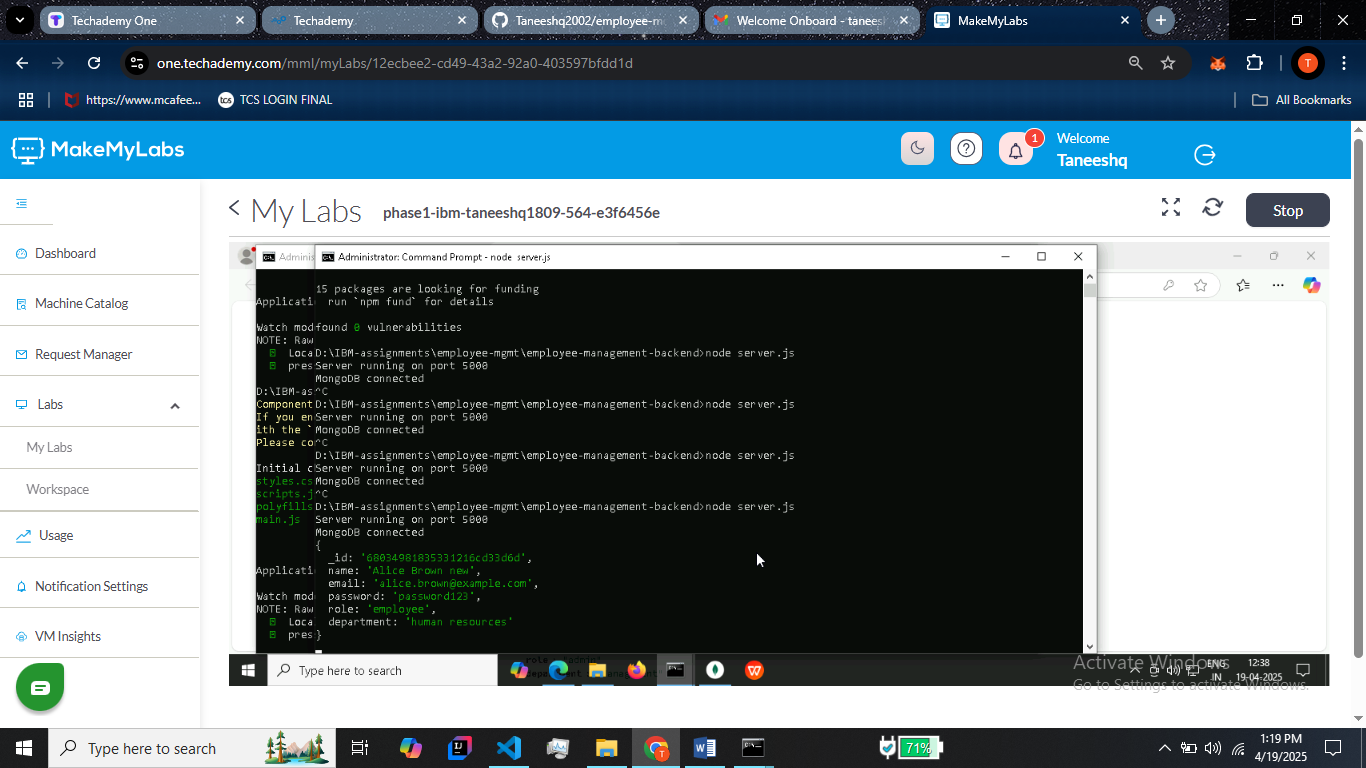
****

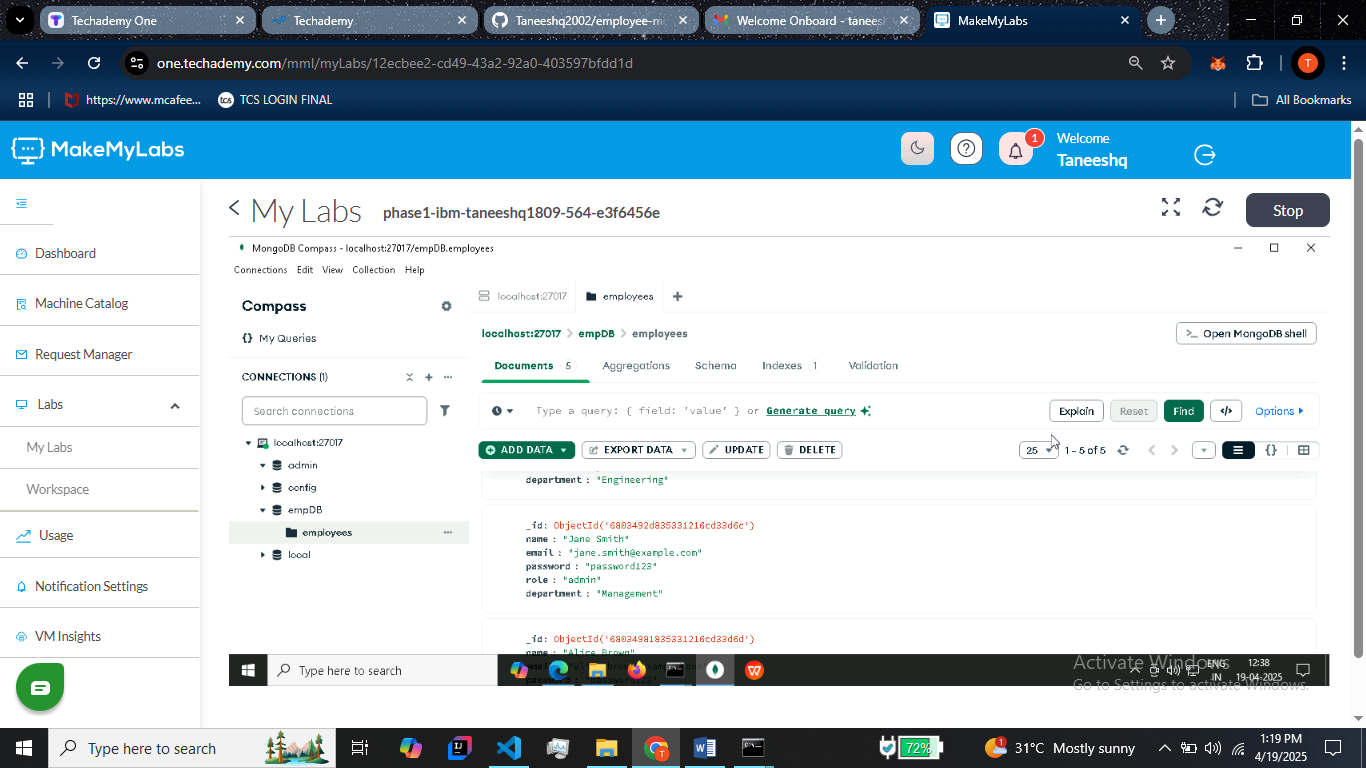
**8) Profile view of employees:**

****

**9) Backend:**

****

****

****