**MINI PROJECT -1**

Backend Development:

**Entity:**

package com.example.Employee.management.Entity;

import io.swagger.v3.oas.annotations.media.Schema;

import jakarta.persistence.\*;

import jakarta.validation.constraints.\*;

import lombok.\*;

@Entity

@Data

@NoArgsConstructor

@AllArgsConstructor

@Table(name = "Employees")

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.*IDENTITY*)

@Schema(description = "Employee ID", example = "1")

private Long id;

@NotBlank(message = "Name is required")

@Size(min = 3, message = "Name should have at least 3 characters")

private String name;

@Email(message = "Invalid Email")

@NotBlank(message = "Email is Required")

private String email;

@NotBlank(message = "Department is Required")

private String department;

}

**Controller:**

package com.example.Employee.management.Controller;

import com.example.Employee.management.Entity.Employee;

import com.example.Employee.management.Repository.EmployeeRepository;

import com.example.Employee.management.Service.EmployeeService;

import io.swagger.v3.oas.annotations.Operation;

import io.swagger.v3.oas.annotations.tags.Tag;

import jakarta.validation.Valid;

import lombok.RequiredArgsConstructor;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.ResponseEntity;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.validation.BindingResult;

import org.springframework.web.bind.annotation.\*;

import java.util.List;

import java.util.Optional;

@Controller

@RequiredArgsConstructor

@Tag(name = "Employee Controller", description = "CRUD operations and view pages for employees")

public class EmployeeController {

@Autowired

private EmployeeService employeeService;

@Autowired

private EmployeeRepository employeeRepository;

@GetMapping("/")

public String viewHome(Model model) {

model.addAttribute("employees", employeeService.getAllEmployees());

return "index";

}

@GetMapping("/employee/add")

public String addEmployeeForm(Model model) {

model.addAttribute("employee", new Employee());

return "add";

}

@PostMapping("/employee/save")

public String saveEmployee(@Valid @ModelAttribute("employee") Employee employee, BindingResult result, Model model) {

Optional<Employee> existing = employeeRepository.findByEmail(employee.getEmail());

if (existing.isPresent()) {

result.rejectValue("email", "error.employee", "Email already exists");

}

if (result.hasErrors()) {

return "add";

}

employeeRepository.save(employee);

return "redirect:/";

}

@GetMapping("/employee/edit/{id}")

public String editForm(@PathVariable Long id, Model model) {

model.addAttribute("employee", employeeService.getEmployeeById(id));

return "edit";

}

@PostMapping("/employee/update")

public String updateEmployee(@Valid @ModelAttribute("employee") Employee employee, BindingResult result, Model model) {

Optional<Employee> existing = employeeRepository.findByEmail(employee.getEmail());

if (existing.isPresent() && !existing.get().getId().equals(employee.getId())) {

result.rejectValue("email", "error.employee", "Email already exists");

}

if (result.hasErrors()) {

return "edit";

}

employeeService.saveEmployee(employee);

return "redirect:/";

}

@GetMapping("/employee/delete/{id}")

public String deleteEmployee(@PathVariable Long id) {

employeeService.deleteEmployee(id);

return "redirect:/";

}

@GetMapping("/employee/view/{id}")

public String viewEmployee(@PathVariable Long id, Model model) {

model.addAttribute("employee", employeeService.getEmployeeById(id));

return "view";

}

// REST endpoints

@Operation(summary = "Get all employees")

@ResponseBody

@GetMapping("/api/employees")

public ResponseEntity<?> getAll() {

return ResponseEntity.*ok*(employeeService.getAllEmployees());

}

@Operation(summary = "Get single employee")

@ResponseBody

@GetMapping("/api/employees/{id}")

public ResponseEntity<?> getById(@PathVariable Long id) {

return ResponseEntity.*ok*(employeeService.getEmployeeById(id));

}

}

**Repository:**

**package com.example.Employee.management.Repository;**

**import com.example.Employee.management.Entity.Employee;**

**import org.springframework.data.jpa.repository.JpaRepository;**

**import java.util.Optional;**

**public interface EmployeeRepository extends JpaRepository<Employee,Long> {**

**Optional<Employee> findByEmail(String email);**

**}**

**Service:**

**package com.example.Employee.management.Service;**

**import com.example.Employee.management.Entity.Employee;**

**import com.example.Employee.management.Repository.EmployeeRepository;**

**import lombok.RequiredArgsConstructor;**

**import org.springframework.beans.factory.annotation.Autowired;**

**import org.springframework.stereotype.Service;**

**import java.util.List;**

**import java.util.NoSuchElementException;**

**@Service**

**@RequiredArgsConstructor**

**public class EmployeeService {**

**@Autowired**

**private EmployeeRepository employeeRepository;**

**public List<Employee>getAllEmployees(){**

**return employeeRepository.findAll();**

**}**

**public Employee getEmployeeById(Long id) {**

**return employeeRepository.findById(id).orElseThrow(() -> new NoSuchElementException("Employee not found"));**

**}**

**public Employee saveEmployee(Employee employee) {**

**return employeeRepository.save(employee);**

**}**

**public void deleteEmployee(Long id) {**

**employeeRepository.deleteById(id);**

**}**

**}**

**Exceptions:**

**package com.example.Employee.management.Exception;**

**import org.springframework.ui.Model;**

**import org.springframework.web.bind.annotation.ControllerAdvice;**

**import org.springframework.web.bind.annotation.ExceptionHandler;**

**import java.util.NoSuchElementException;**

**@ControllerAdvice**

**public class GlobalExceptionHandler {**

**@ExceptionHandler(NoSuchElementException.class)**

**public String handleNotFound(NoSuchElementException ex, Model model) {**

**model.addAttribute("errorMessage", ex.getMessage());**

**return "error";**

**}**

**@ExceptionHandler(Exception.class)**

**public String handleAllExceptions(Exception ex, Model model) {**

**model.addAttribute("message", ex.getMessage());**

**return "error";**

**}**

**}**

**FrontEnd Development:**

**addEmployee:**

**<!DOCTYPE html>**

**<html xmlns:th="http://www.thymeleaf.org">**

**<head>**

**<title>Add Employee</title>**

**<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet"/>**

**</head>**

**<body>**

**<div class="container mt-4">**

**<h2>Add Employee</h2>**

**<form th:action="@{/employee/save}" th:object="${employee}" method="post">**

**<div class="mb-3">**

**<label>Name</label>**

**<input th:field="\*{name}" class="form-control" />**

**<p class="text-danger" th:errors="\*{name}"></p>**

**</div>**

**<div class="mb-3">**

**<label>Email</label>**

**<input th:field="\*{email}" class="form-control" />**

**<p class="text-danger" th:errors="\*{email}"></p>**

**</div>**

**<div class="mb-3">**

**<label>Department</label>**

**<input th:field="\*{department}" class="form-control" />**

**<p class="text-danger" th:errors="\*{department}"></p>**

**</div>**

**<button type="submit" class="btn btn-success">Save</button>**

**<a href="/" class="btn btn-secondary">Cancel</a>**

**</form>**

**</div>**

**</body>**

**</html>**

**Edit:**

**<!DOCTYPE html>**

**<html xmlns:th="http://www.thymeleaf.org">**

**<head>**

**<title>Edit Employee</title>**

**<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet"/>**

**</head>**

**<body>**

**<div class="container mt-4">**

**<h2>Edit Employee</h2>**

**<form th:action="@{/employee/update}" th:object="${employee}" method="post">**

**<input type="hidden" th:field="\*{id}" />**

**<div class="mb-3">**

**<label>Name</label>**

**<input th:field="\*{name}" class="form-control" />**

**<p class="text-danger" th:errors="\*{name}"></p>**

**</div>**

**<div class="mb-3">**

**<label>Email</label>**

**<input th:field="\*{email}" class="form-control" />**

**<p class="text-danger" th:errors="\*{email}"></p>**

**</div>**

**<div class="mb-3">**

**<label>Department</label>**

**<input th:field="\*{department}" class="form-control" />**

**<p class="text-danger" th:errors="\*{department}"></p>**

**</div>**

**<button type="submit" class="btn btn-success">Update</button>**

**<a href="/" class="btn btn-secondary">Cancel</a>**

**</form>**

**</div>**

**</body>**

**</html>**

**View:**

**<!DOCTYPE html>**

**<html xmlns:th="http://www.thymeleaf.org">**

**<head>**

**<title>View Employee</title>**

**<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet"/>**

**</head>**

**<body>**

**<div class="container mt-4">**

**<h2>Employee Details</h2>**

**<table class="table table-striped">**

**<tr>**

**<th>ID</th>**

**<td th:text="${employee.id}"></td>**

**</tr>**

**<tr>**

**<th>Name</th>**

**<td th:text="${employee.name}"></td>**

**</tr>**

**<tr>**

**<th>Email</th>**

**<td th:text="${employee.email}"></td>**

**</tr>**

**<tr>**

**<th>Department</th>**

**<td th:text="${employee.department}"></td>**

**</tr>**

**</table>**

**<a href="/" class="btn btn-primary">Back to List</a>**

**</div>**

**</body>**

**</html>**

**Index:**

**<!DOCTYPE html>**

**<html xmlns:th="http://www.thymeleaf.org">**

**<head>**

**<title>Employee List</title>**

**<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet"/>**

**</head>**

**<body>**

**<div class="container mt-4">**

**<h2 class="text-center">Employees</h2>**

**<a href="/employee/add" class="btn btn-primary mb-3">Add New</a>**

**<table class="table table-bordered">**

**<tr><th>Name</th><th>Email</th><th>Department</th><th>Actions</th></tr>**

**<tr th:each="emp : ${employees}">**

**<td th:text="${emp.name}"></td>**

**<td th:text="${emp.email}"></td>**

**<td th:text="${emp.department}"></td>**

**<td>**

**<a th:href="@{'/employee/view/' + ${emp.id}}" class="btn btn-info btn-sm">View</a>**

**<a th:href="@{'/employee/edit/' + ${emp.id}}" class="btn btn-warning btn-sm">Edit</a>**

**<a th:href="@{'/employee/delete/' + ${emp.id}}" class="btn btn-danger btn-sm" onclick="return confirm('Are you sure?')">Delete</a>**

**</td>**

**</tr>**

**</table>**

**</div>**

**</body>**

**</html>**

**Errors:**

**<!DOCTYPE html>**

**<html xmlns:th="http://www.thymeleaf.org">**

**<head>**

**<title>Error</title>**

**</head>**

**<body>**

**<div class="container mt-5">**

**<h2 class="text-danger">Something went wrong</h2>**

**<p th:text="${errorMessage}"></p>**

**<a href="/" class="btn btn-primary">Go Home</a>**

**</div>**

**</body>**

**</html>**

**<!DOCTYPE html>**

**<html lang="en" xmlns:th="http://www.thymeleaf.org">**

**<head>**

**<meta charset="UTF-8">**

**<title>Title</title>**

**<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet"/>**

**</head>**

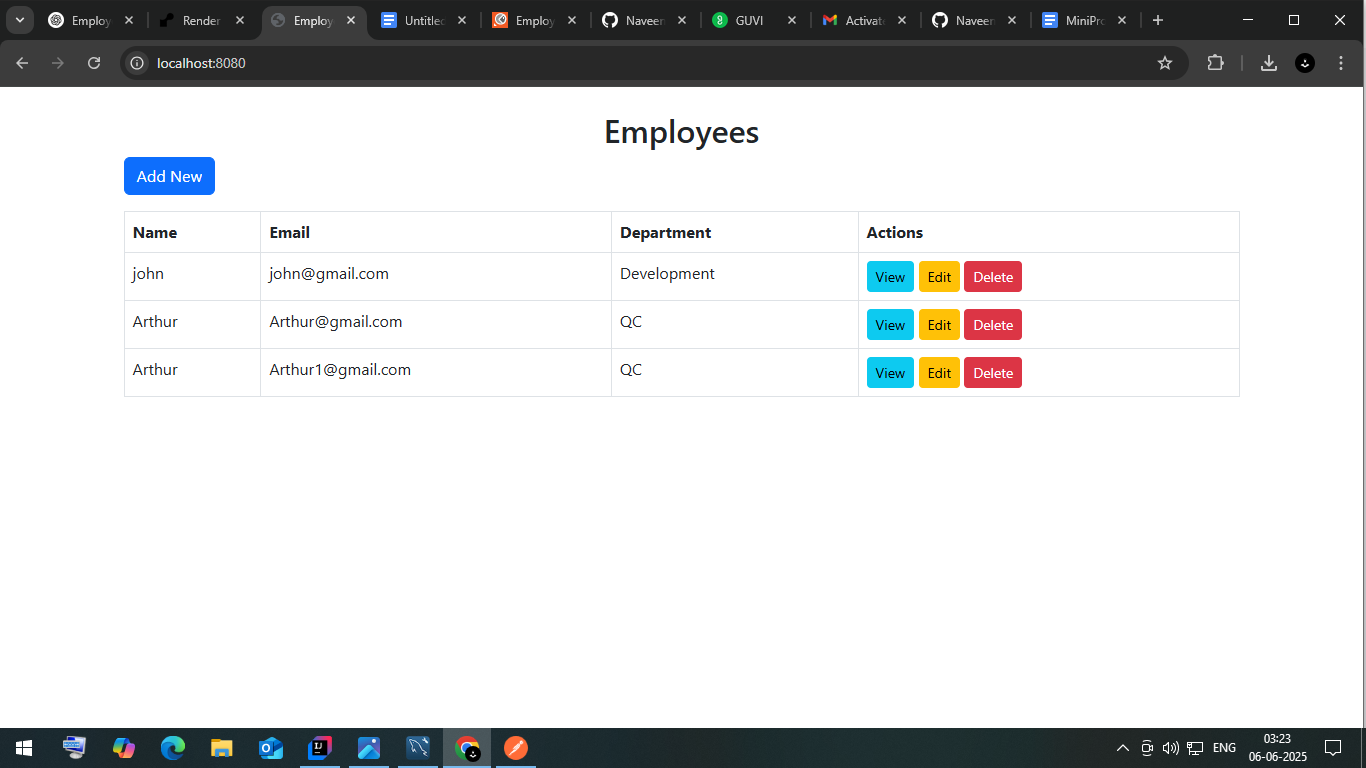
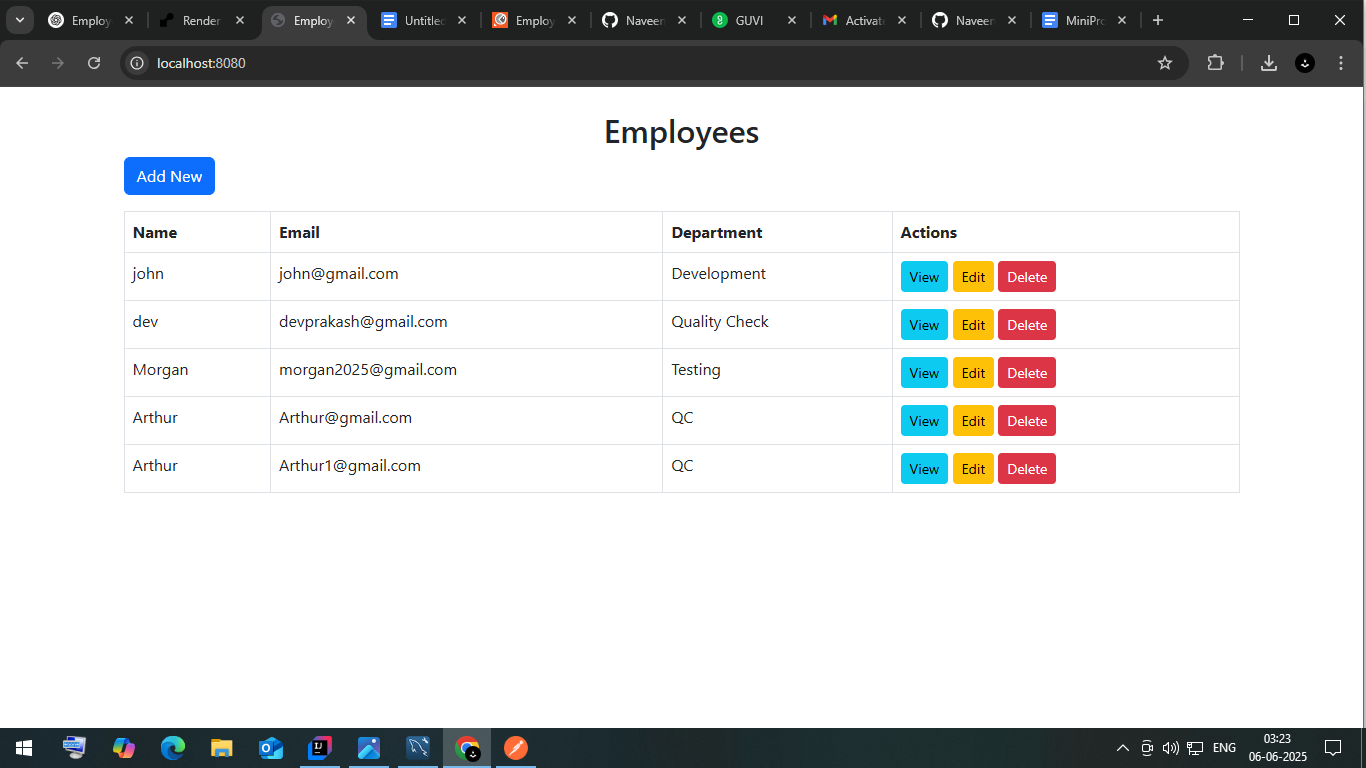
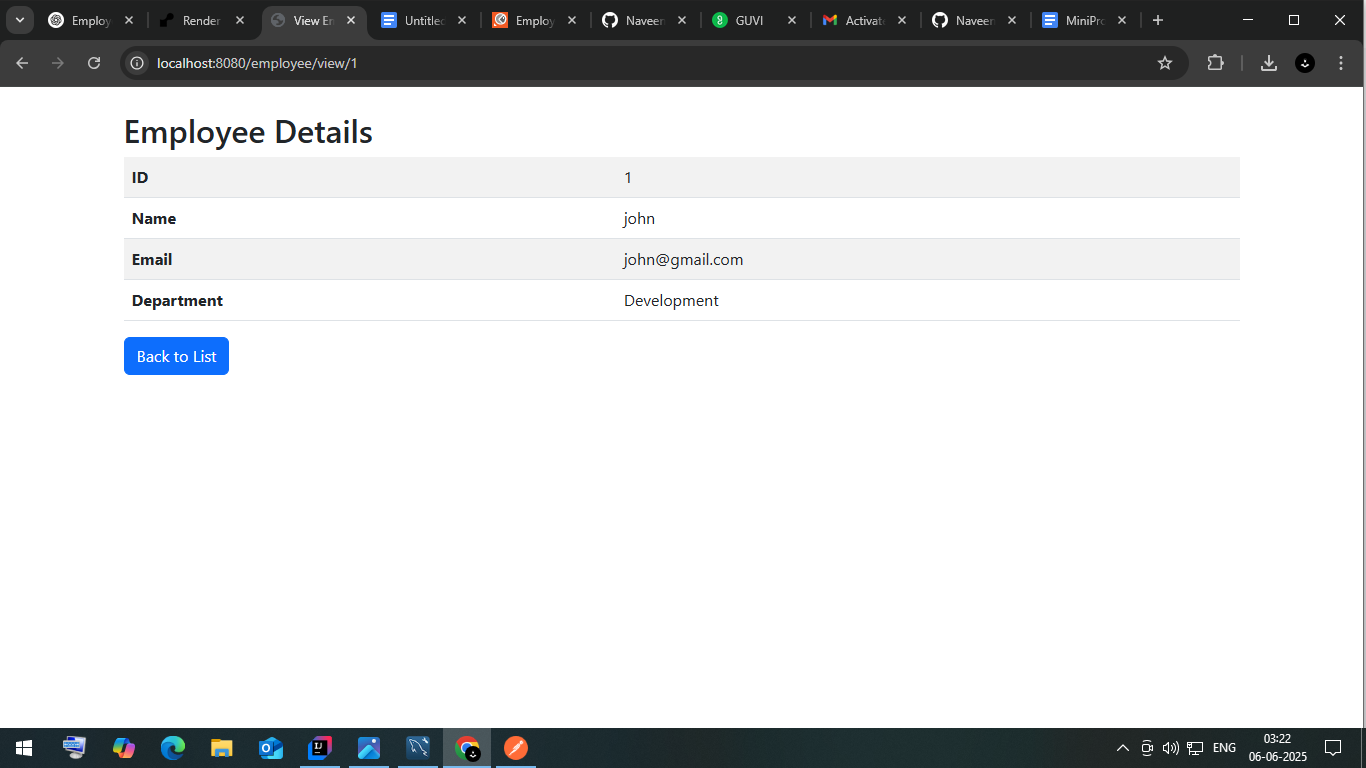
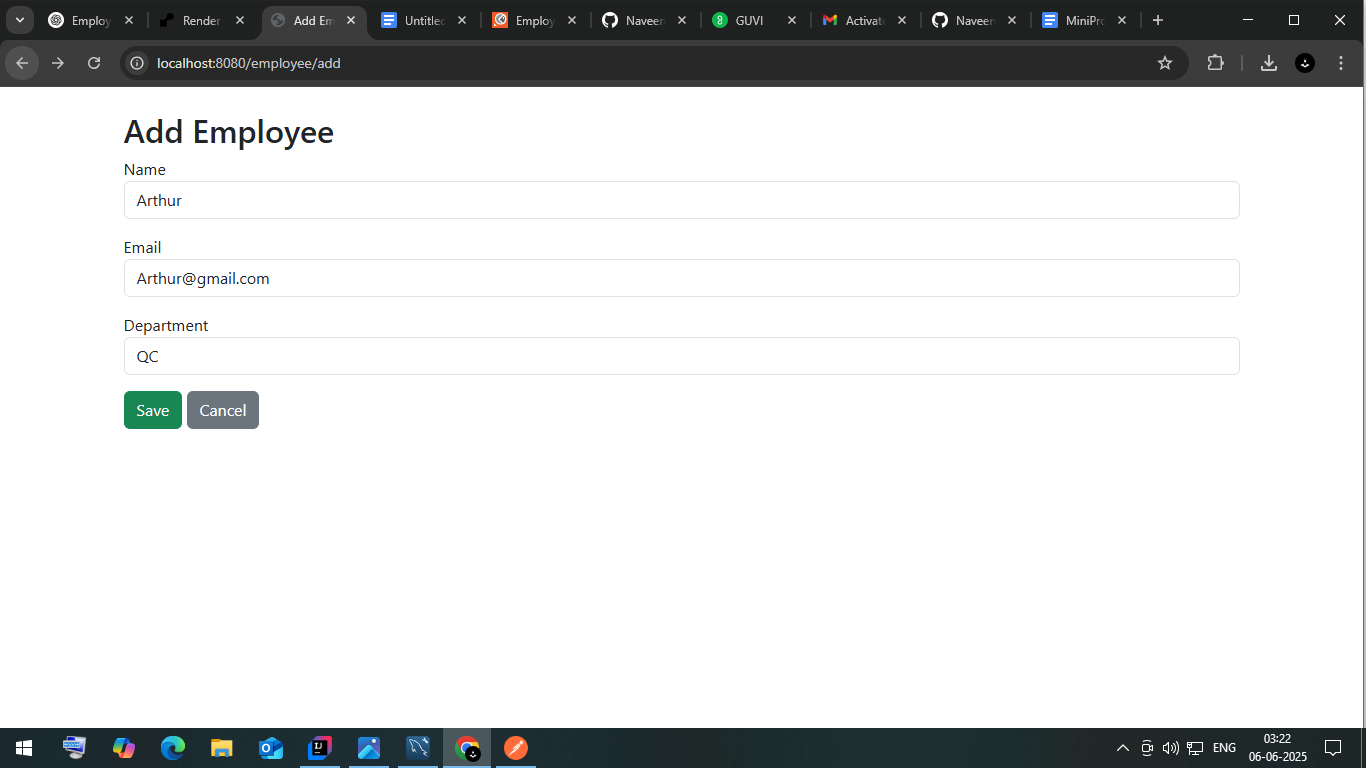
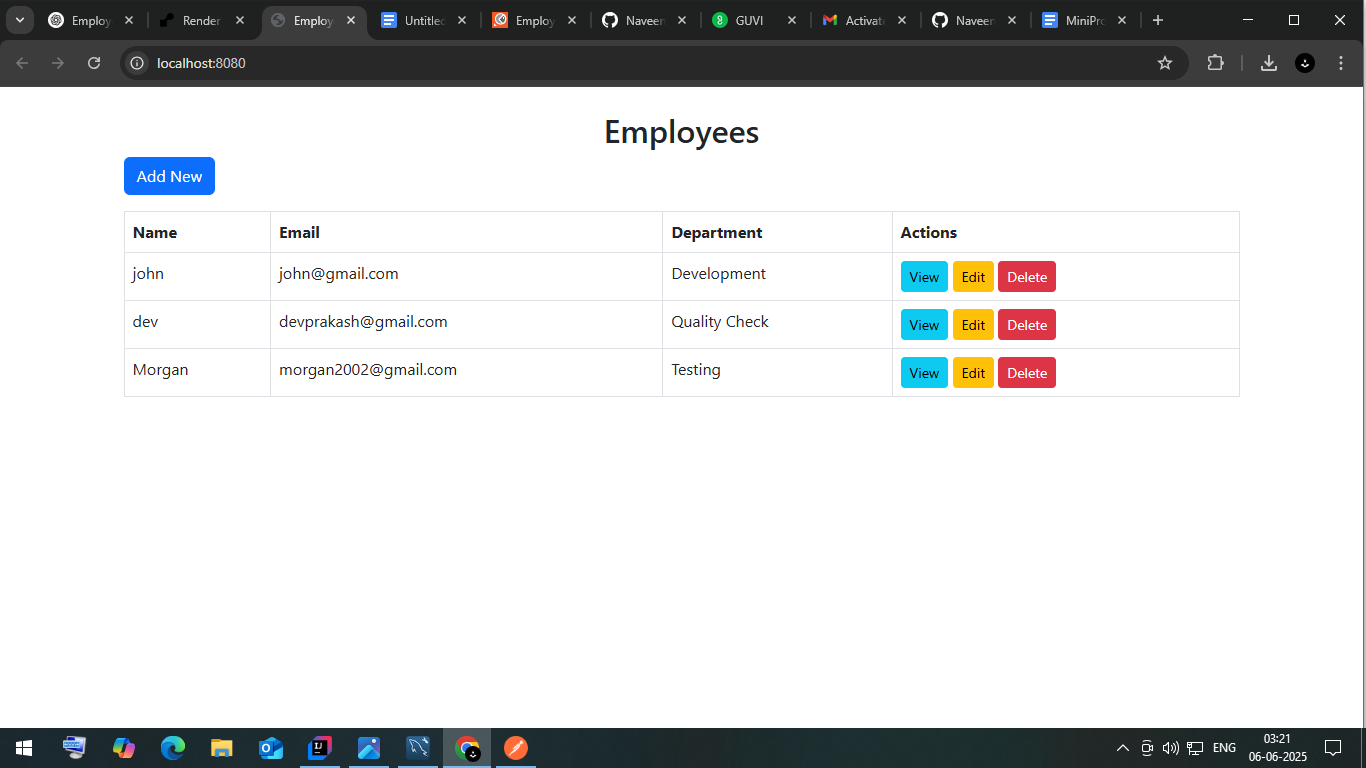
**<body>**

**<h2>Error Occurred</h2>**

**<p th:text="${message}" class="btn btn-secondary"></p>**

**</body>**

**</html>**

**Output Screenshots: **

**API DOcumentation link:**

[**https://documenter.getpostman.com/view/45666348/2sB2x2Ka1H#get-started-here**](https://documenter.getpostman.com/view/45666348/2sB2x2Ka1H#get-started-here)