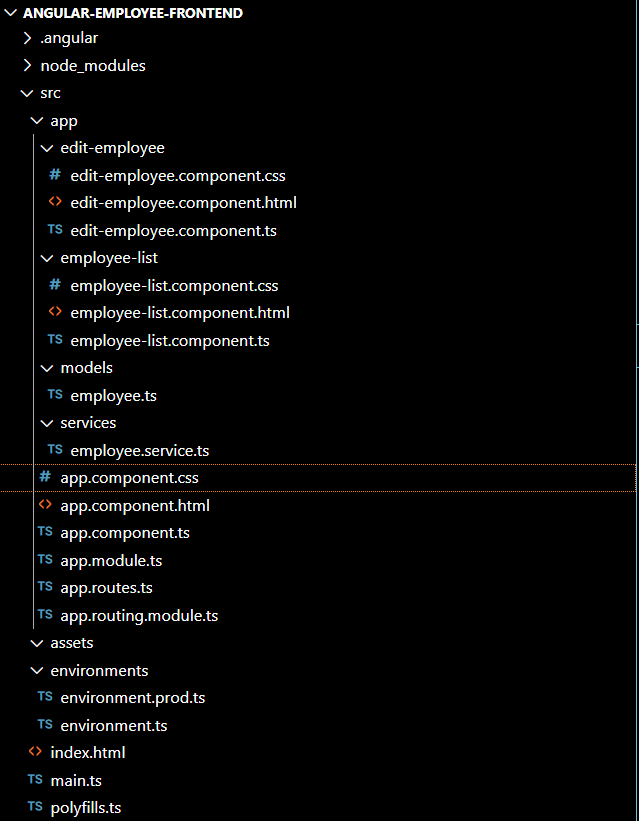
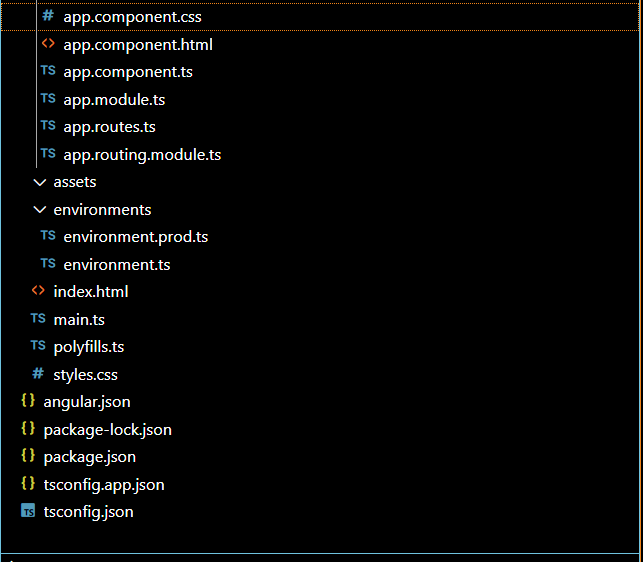
**Module 7 - Spring REST using Spring Boot 3**

**Display Employee List and Edit Employee form using RESTful Web Service**

**FrontEnd Structure:**





form {

  margin-top: 20px;

}

input {

  margin-bottom: 10px;

}

<h2>Edit Employee</h2>

<form (ngSubmit)="onSubmit()">

  <label>Name:

    <input [(ngModel)]="employee.name" name="name">

  </label><br>

  <label>Department:

    <input [(ngModel)]="employee.department" name="department">

  </label><br>

  <label>Salary:

    <input [(ngModel)]="employee.salary" name="salary" type="number">

  </label><br>

  <button type="submit">Update</button>

</form>

import { Component, OnInit } from '@angular/core';

import { ActivatedRoute } from '@angular/router';

import { HttpClient, HttpClientModule } from '@angular/common/http';

import { CommonModule } from '@angular/common';

import { FormsModule } from '@angular/forms';

@Component({

  selector: 'app-edit-employee',

  standalone: true,

  imports: [CommonModule, FormsModule, HttpClientModule],

  templateUrl: './edit-employee.component.html'

})

export class EditEmployeeComponent implements OnInit {

  employee: any = {};

  id: number = 0;

  constructor(private route: ActivatedRoute, private http: HttpClient) {}

  ngOnInit(): void {

    this.id = +this.route.snapshot.paramMap.get('id')!;

    this.http.get(`http://localhost:8086/employees/${this.id}`).subscribe(data => {

      this.employee = data;

    });

  }

  onSubmit() {

    this.http.put(`http://localhost:8086/employees/${this.id}`, this.employee).subscribe(() => {

      alert('Employee updated successfully!');

    });

  }

}

/\* Remove default margins/padding \*/

body, html {

  margin: 0;

  padding: 0;

  font-family: Arial, sans-serif;

  background-color: #f5f5f5;

}

/\* Container if you want optional padding \*/

.container {

  width: 100%;

  padding: 20px;

  box-sizing: border-box;

}

/\* Table should cover full horizontal width \*/

table {

  width: 100%;

  border-collapse: collapse;

  background-color: white;

}

/\* Header and cells \*/

th, td {

  padding: 12px;

  border: 1px solid #ccc;

  text-align: left;

}

/\* Table header styling \*/

thead {

  background-color: #e0e0e0;

}

/\* Button styling \*/

button {

  padding: 6px 12px;

  background-color: #4CAF50;

  color: white;

  border: none;

  border-radius: 4px;

  cursor: pointer;

}

button:hover {

  background-color: #45a049;

}

<h2>Employee List</h2>

<table \*ngIf="employees.length > 0" border="1">

  <thead>

    <tr>

      <th>ID</th>

      <th>Name</th>

      <th>Department</th>

      <th>Salary</th>

      <th>Action</th>

    </tr>

  </thead>

  <tbody>

    <tr \*ngFor="let emp of employees">

      <td>{{ emp.id }}</td>

      <td>{{ emp.name }}</td>

      <td>{{ emp.department || 'N/A' }}</td>

      <td>{{ emp.salary }}</td>

      <td><button (click)="editEmployee(emp.id)">Edit</button></td>

    </tr>

  </tbody>

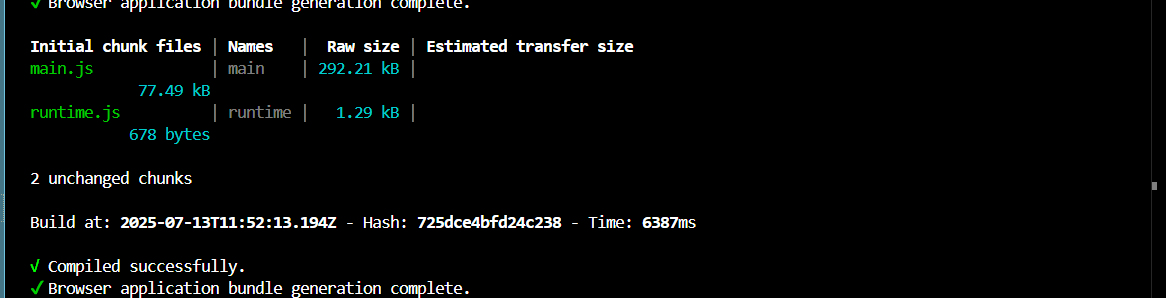
</table>

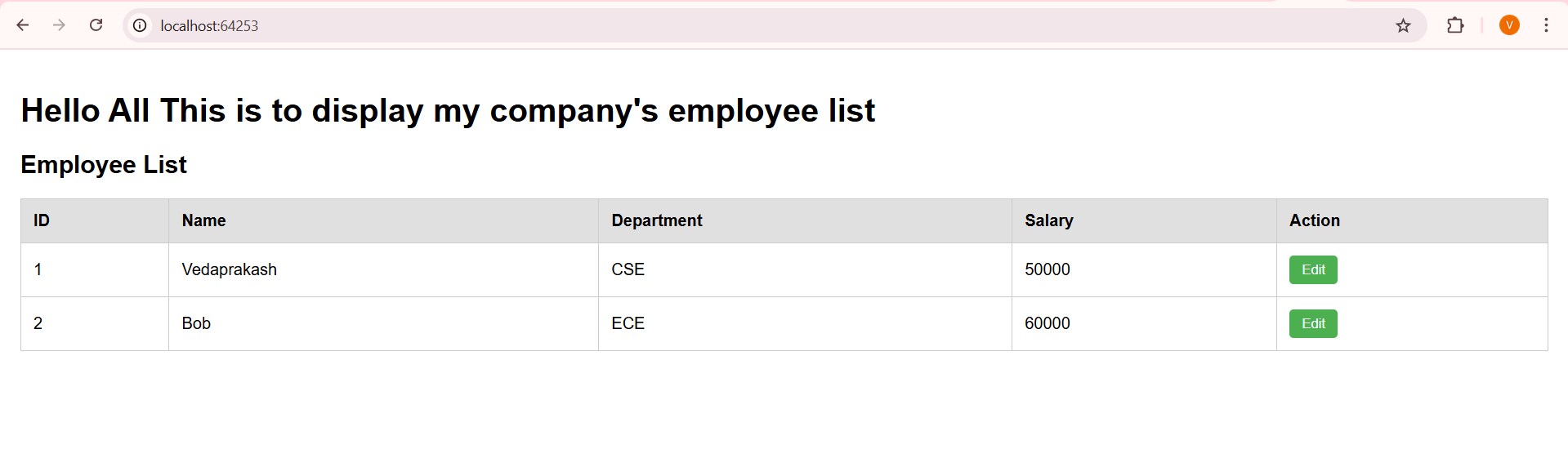
import { Component, OnInit } from '@angular/core';

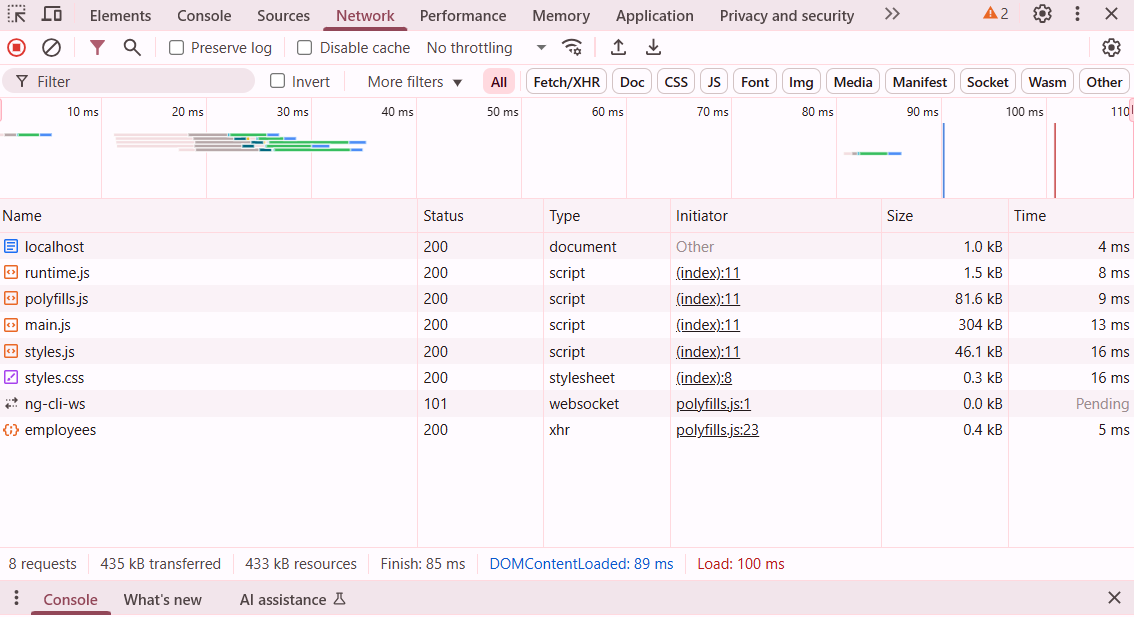
import { CommonModule, JsonPipe } from '@angular/common';

import { EmployeeService, Employee } from '../services/employee.service';

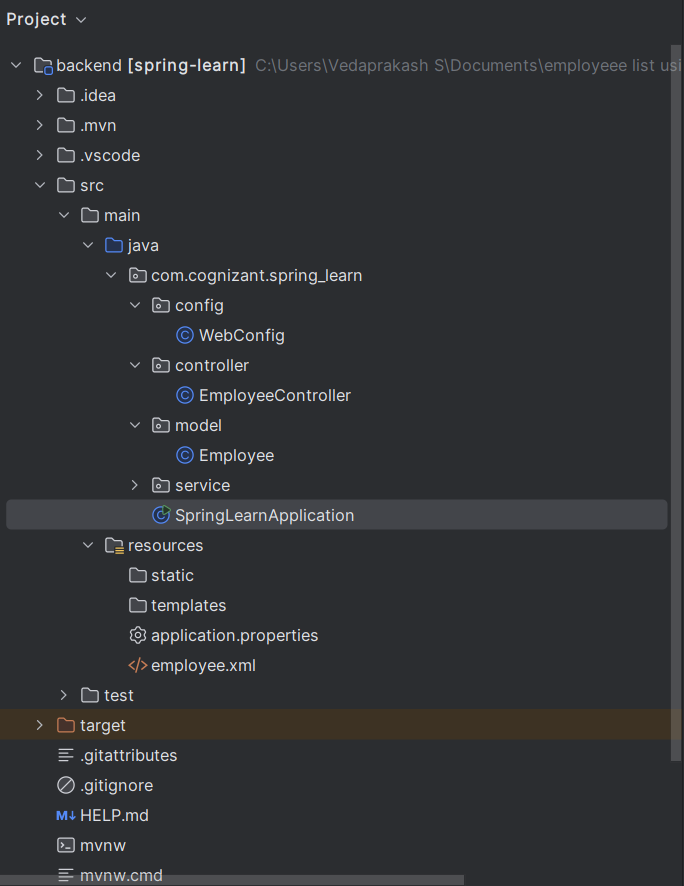
**Output:**

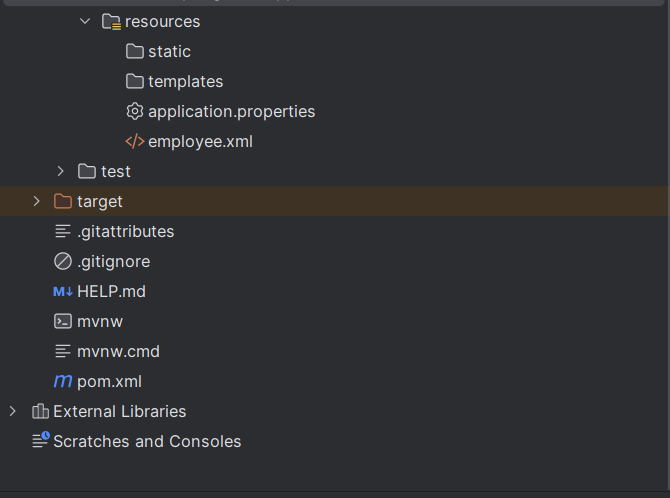






**BackEnd Structure:**





**WebConfig.java:**

package com.cognizant.spring\_learn.config;  
  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.Configuration;  
import org.springframework.web.servlet.config.annotation.CorsRegistry;  
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;  
  
@Configuration  
public class WebConfig {  
  
 @Bean  
 public WebMvcConfigurer corsConfigurer() {  
 return new WebMvcConfigurer() {  
 @Override  
 public void addCorsMappings(CorsRegistry registry) {  
 registry.addMapping("/\*\*")  
 .allowedOrigins("http://localhost:4200", "http://localhost:59571","http://localhost:60685","http://localhost:63244","http://localhost:64253/")  
 .allowedMethods("GET", "POST", "PUT", "DELETE", "OPTIONS");  
 }  
 };  
 }  
}

**EmployeeController.java:**

package com.cognizant.spring\_learn.controller;  
  
import com.cognizant.spring\_learn.model.Employee;  
import com.cognizant.spring\_learn.service.EmployeeService;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.\*;  
  
import java.util.List;  
  
@RestController  
public class EmployeeController {  
  
 @Autowired  
 private EmployeeService employeeService;  
  
 @GetMapping("/employees")  
 public List<Employee> getAll() {  
 return employeeService.getAllEmployees();  
 }  
  
 @GetMapping("/employees/{id}")  
 public Employee getById(@PathVariable int id) {  
 return employeeService.getById(id);  
 }  
}

**Employee.java:**

package com.cognizant.spring\_learn.model;  
  
public class Employee {  
 private int id;  
 private String name;  
 private String department;  
 private double salary;  
 // Changed to String  
  
 public Employee() {  
 }  
  
   
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getDepartment() {  
 return department;  
 }  
  
 public void setDepartment(String department) {  
 this.department = department;  
 }  
  
 public double getSalary() {  
 return salary;  
 }  
  
 public void setSalary(double salary) {  
 this.salary = salary;  
 }  
}

**SpringLearnApplication.java:**

package com.cognizant.spring\_learn;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class SpringLearnApplication {  
 public static void main(String[] args) {  
 SpringApplication.*run*(SpringLearnApplication.class, args);  
 }  
}

**application.properties:**  
server.port=8086

**employee.xml:**

<beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://www.springframework.org/schema/beans  
 http://www.springframework.org/schema/beans/spring-beans.xsd">  
  
 <bean id="employee1" class="com.cognizant.spring\_learn.model.Employee">  
 <property name="id" value="1"/>  
 <property name="name" value="Vedaprakash"/>  
 <property name="department" value="CSE"/>  
 <property name="salary" value="50000"/>  
  
  
 </bean>  
  
  
 <bean id="employee2" class="com.cognizant.spring\_learn.model.Employee">  
 <property name="id" value="2"/>  
 <property name="name" value="Bob"/>  
 <property name="department" value="ECE"/>  
 <property name="salary" value="60000"/>  
  
  
 </bean>  
  
 <bean id="employeeList" class="java.util.ArrayList">  
 <constructor-arg>  
 <list>  
 <ref bean="employee1"/>  
 <ref bean="employee2"/>  
 </list>  
 </constructor-arg>  
 </bean>  
</beans>

**pom.xml:**

<project xmlns="http://maven.apache.org/POM/4.0.0"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0  
 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
  
 <groupId>com.cognizant</groupId>  
 <artifactId>spring-learn</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <packaging>jar</packaging>  
  
 <name>spring-learn</name>  
  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>2.7.18</version> <!-- Use a stable version -->  
 <relativePath/>  
 </parent>  
  
 <properties>  
 <java.version>11</java.version>  
 </properties>  
  
 <dependencies>  
  
 <!-- Spring Boot Starter Web -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
  
 <!-- Spring Boot Starter Test -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
 </dependency>  
  
 <!-- Lombok (Optional, for annotations like @Data) -->  
 <dependency>  
 <groupId>org.projectlombok</groupId>  
 <artifactId>lombok</artifactId>  
 <optional>true</optional>  
 </dependency>  
  
 </dependencies>  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 </plugins>  
 </build>  
  
</project>

**Output:**

