**Full Stack Java Developer**

Phase-End Project Problem Statement



**Phase-End Project 4**

**Event Management** **App**

**Project Agenda:** Build an Angular app where the admin can view and manage employees.

**Scenario:**

You are working as a web developer and your company has decided to launch a new app called Event Management. The administration team of the organization can view and add their resources such as employees and keep their data saved for future references.

The backend development has been outsourced as APIs and the frontend will be developed in-house by using Angular as a client-side framework. During the sprint planning, you agree to lead this project and develop an app that will let the admin find employees using APIs outsourced by backend engineers.

The tasks that need to be performed by you are:

* Build the application using Angular
* Create a temporary database server for CRUD operation using a JSON local server
* Validate all employee management forms using Form Validation
* Render the application as a Single Page Application

**Tools Required**:

* Angular
* Bootstrap

**The Following Requirements Should Be Met:**

* Admin login page where admin can change the password after logging in if he wants to.
* Admin can view a master list of employee details.
* Admin can create, remove, update, or delete employee details.

**Refer the following steps for JSON server creation:**

1) Execute the command given below:  
npm install -g json-server  
  
2) Create a file with the name db.json and add the code given below:  
{  
    "employees": [  
      {  
        "id": 1,  
        "first\_name": "Sebastian",  
        "last\_name": "Eschweiler",  
        "email": "[sebastian@codingthesmartway.com](mailto:sebastian@codingthesmartway.com)"  
      },  
      {  
        "id": 2,  
        "first\_name": "Steve",  
        "last\_name": "Palmer",  
        "email": "[steve@codingthesmartway.com](mailto:steve@codingthesmartway.com)"  
      },  
      {  
        "id": 3,  
        "first\_name": "Ann",  
        "last\_name": "Smith",  
        "email": "[ann@codingthesmartway.com](mailto:ann@codingthesmartway.com)"  
      }  
    ]  
  }  
   
3) Add the code given below in the script section of your package.json file in the root folder of your Angular application  
  
    "json:server": "json-server --watch db.json"  
  
Note: Before executing your Angular Application, run the command given below to start your json server:  
npm run json-server

**Package.json file**

{

  "name": "event-management-app",

  "version": "0.0.0",

  "scripts": {

    "ng": "ng",

    "start": "ng serve",

    "build": "ng build",

    "watch": "ng build --watch --configuration development",

    "test": "ng test",

    "json:server": "json-server --watch db.json"

  },

  "private": true,

  "dependencies": {

    "@angular/animations": "^14.2.0",

    "@angular/common": "^14.2.0",

    "@angular/compiler": "^14.2.0",

    "@angular/core": "^14.2.0",

    "@angular/forms": "^14.2.0",

    "@angular/platform-browser": "^14.2.0",

    "@angular/platform-browser-dynamic": "^14.2.0",

    "@angular/router": "^14.2.0",

    "rxjs": "~7.5.0",

    "tslib": "^2.3.0",

    "zone.js": "~0.11.4"

  },

  "devDependencies": {

    "@angular-devkit/build-angular": "^14.2.2",

    "@angular/cli": "~14.2.2",

    "@angular/compiler-cli": "^14.2.0",

    "@types/jasmine": "~4.0.0",

    "jasmine-core": "~4.3.0",

    "karma": "~6.4.0",

    "karma-chrome-launcher": "~3.1.0",

    "karma-coverage": "~2.2.0",

    "karma-jasmine": "~5.1.0",

    "karma-jasmine-html-reporter": "~2.0.0",

    "typescript": "~4.7.2"

  }

}

**To run the server:**

json-server --watch ./src/app/data.json

**Sample URI to perform CRUD operations after running the json-server:**

|  |  |  |
| --- | --- | --- |
| **HTTP Methods** | **URI** | **Description** |
| GET | /employees | Get all employees |
| GET | /employees/1 | Get employee by ID |
| POST | /employees | Add employee |
| PUT | /employees | Update employee |
| DELETE | /employees/1 | Delete employee by ID |

**Snapshots of login page:**Graphical user interface, Word

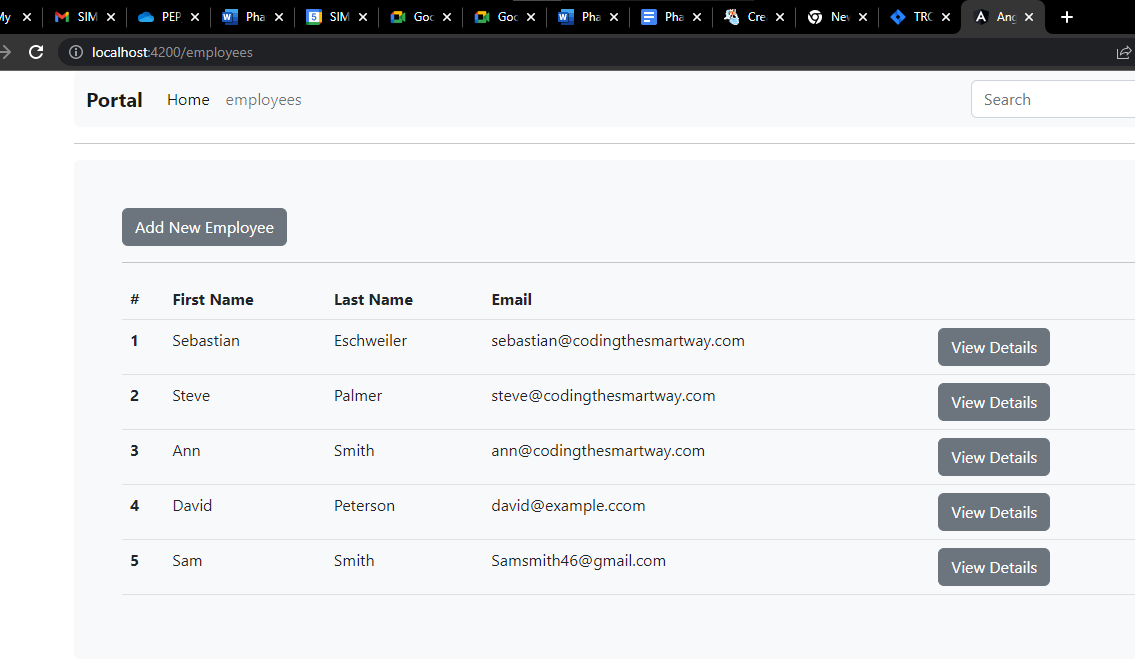
Description automatically generated

**Employee details:**

Graphical user interface, application

Description automatically generated

**View Employee details:**



Graphical user interface, text, application, email

Description automatically generated