Event Management App

DESCRIPTION:

Build an Angular app where the admin can view and manage employees.

Scenario:

I am 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"
    },
    {
     "id": 2,
     "first_name": "Steve",
     "last_name": "Palmer",
     "email": "steve@codingthesmartway.com"
    },
    {
     "id": 3,
     "first_name": "Ann",
     "last_name": "Smith",
     "email": "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

Angular:

Angular is a development platform, built on TypeScript. As a platform, Angular includes:

- Angular is a platform and framework for building single-page client applications using HTML and TypeScript.
- A component-based framework for building scalable web applications.
- A collection of well-integrated libraries that cover a wide variety of features, including routing, forms management, client-server communication, and more.
- A suite of developer tools to help you develop, build, test, and update your code

Algorithm:

Step -1: First I created a components called dashboard component,signup, and login component.

Using the commands to create the components

"ng g c dashboard "→dashboard

"ng g c login "→login

"ng g c signup "→signup

Step- 2: Create a api service for performing the curd operation:

"ng g s api" by using this command.

Step -3: Add the appropriate module in that components.

Step- 4: we can create db json called db.json for storing the values into the db:

For installing and run use the following command:

Installing the command is **”npm install -g json-server”.**

To run server the command is **”json-server --watch db.json”.**

Create a database db.json:

```
{
 "posts": [
  { "id": 1, "title": "json-server", "author": "typicode" }
 ],
 "comments": [
  { "id": 1, "body": "some comment", "postId": 1 }
 ],
 "profile": { "name": "typicode" }
}
```

Step -5: After adding the appropriate modules run the server for webpage

“**ng serve**” Command is used .

It will get the localhost port number is **http://localhost:4200**.

Code:

eventdash.component.html:

```
<nav class="navbar navbar-light bg-dark">
    <div class="container-fluid">
        <h1 style="color:aliceblue;">Event Manager</h1>
        <div class="d-flex">
            <button (click)="clickAddEmployee()" type="button"  data-bs-toggle="modal"
data-bs-target="#exampleModal" class="btn btn-success">Add Employee</button>
            <button routerLink="/login" class="btn btn-danger mx-2">Logout</button>
        </div>
    </div>
</nav>

<table class="table mt-3">
    <thead>
        <tr>
            <th scope="col"> Employee Id</th>
            <th scope="col"> first Name</th>
            <th scope="col"> last Name</th>
            <th scope="col"> Email Id</th>
            <th scope="col">Action</th>
        </tr>
    </thead>
    <tbody>
        <tr *ngFor="let row of employeeData">
            <td>{{row.id}}</td>
            <td>{{row.firstName}}</td>
            <td>{{row.lastName}}</td>
            <td>{{row.email}}</td>

            <td>
                <button  (click)="onEdit(row)" type="button"  data-bs-toggle="modal"
data-bs-target="#exampleModal" class="btn btn-info">Edit</button>
                <button (click)="deleteEmployee(row)" class="btn btn-danger mx-3">
Delete</button>
            </td>
        </tr>
    </tbody>
</table>

  <!-- Modal -->
  <div class="modal fade" id="exampleModal" tabindex="-1" aria-
labelledby="exampleModalLabel" aria-hidden="true">
    <div class="modal-dialog">
      <div class="modal-content">
        <div class="modal-header">
          <h5 class="modal-title" id="exampleModalLabel">Employee Details</h5>
```

```
          <button type="button" class="btn-close" data-bs-dismiss="modal" aria-
label="Close"></button>
        </div>
        <div class="modal-body">
            <form [formGroup]="formValue">
                <div class="mb-3">
                  <label for="exampleInputEmail1" class="form-label">First Name</label>
                  <input type="text"  formControlName="firstName" class="form-control"
id="exampleInputEmail1" aria-describedby="emailHelp">

                </div>
                <div class="mb-3">
                  <label for="exampleInputPassword1" class="form-label">last Name</label>
                  <input type="text" formControlName="lastName" class="form-control"
id="exampleInputPassword1">
                </div>
                <div class="mb-3">
                    <label for="exampleInputPassword1" class="form-label">Emai Id</label>
                    <input type="text" formControlName="email" class="form-control"
id="exampleInputPassword1">
                  </div>
              </form>
        </div>
        <div class="modal-footer">
          <button type="button" id="cancel"class="btn btn-secondary" data-bs-
dismiss="modal">Close</button>
          <button (click)="postEmployeeDetails()"*ngIf="showAdd"   type="button"
class="btn btn-primary">Add</button>
          <button (click)="updateEmployeeDetails()"*ngIf="showUpdate" type="button"
class="btn btn-primary">Update</button>
        </div>
      </div>
    </div>
  </div>
```

eventdash.component.ts:

```
import { Component, OnInit } from '@angular/core';
import{FormBuilder,FormGroup} from '@angular/forms';
import { ApiService } from '../shared/api.service';
import { EmployeeModel } from './eventdash.model';


@Component({
  selector: 'app-eventdash',
  templateUrl: './eventdash.component.html',
  styleUrls: ['./eventdash.component.css']
})
export class EventdashComponent implements OnInit {
  formValue !:FormGroup;
  employeeModelObj :EmployeeModel=new EmployeeModel();
  employeeData !:any;
```

```
showAdd !:boolean;
showUpdate !:boolean;

constructor( private formbuilder:FormBuilder,
  private api: ApiService) { }

ngOnInit(): void {
  this.formValue=this.formbuilder.group({
    firstName :[''],
    lastName :[''],
    email :['']
  })
  this.getAllEmployee();
}
clickAddEmployee(){
  this.formValue.reset();
  this.showAdd=true;
  this.showUpdate=false;
}
postEmployeeDetails(){
  this.employeeModelObj.firstName =this.formValue.value.firstName;
  this.employeeModelObj.lastName =this.formValue.value.lastName;
  this.employeeModelObj.email =this.formValue.value.email;

  this.api.postEmployee(this.employeeModelObj)
  .subscribe(res=>{
    console.log(res);
    alert("Employee Added Successfully");
    let ref=document.getElementById('cancel');
    ref?.click();
    this.formValue.reset();
    this.getAllEmployee();
  },
  err=>{
    alert("Something went worng");

  })
}

getAllEmployee(){
  this.api.getEmployee()
  .subscribe(res=>{
    this.employeeData=res;

  })
}

deleteEmployee(row : any){
  this.api.deleteEmployee(row.id)
  .subscribe(res=>{
    alert("Employee deleted")
```

```
      this.getAllEmployee();
    })
  }
  onEdit(row :any){
    this.showAdd=false;
    this.showUpdate=true;

    this.employeeModelObj.id=row.id;
    this.formValue.controls['firstName'].setValue(row.firstName);
    this.formValue.controls['lastName'].setValue(row.lastName);
    this.formValue.controls['email'].setValue(row.email);

  }
  updateEmployeeDetails(){
    this.employeeModelObj.firstName =this.formValue.value.firstName;
    this.employeeModelObj.lastName =this.formValue.value.lastName;
    this.employeeModelObj.email =this.formValue.value.email;

    this.api.updateEmployee(this.employeeModelObj,this.employeeModelObj.id)
    .subscribe(res=>{
      alert("Updated Successfully");
      let ref=document.getElementById('cancel');
      ref?.click();
      this.formValue.reset();
      this.getAllEmployee();

    })

  }
}
```

eventdash.model.ts:

```
export class EmployeeModel{
    id : number =0;
    firstName :string ='';
    lastName : string='';
    email : string ='';
}
```

login.component.css:

```
.card{
    border: none;
    width: 500px;
    padding: 40px;
    position: absolute;
    top:50%;
    left: 50%;
    transform:translate(-50%, -50%);
    background: rgb(13, 137, 232);
    background: linear-gradient(90deg,rgb(230, 81, 160) 0%,rgb(114, 98, 236)100%);


}
```

login.component.html:

```
<div class="container">
    <div class="row">
        <div class="col-md-6">
            <div class="card">
                <div class="text-center">
                    <h1>Admin Login</h1>
                    <h6>Please Enter email & password !!</h6>
                </div>

                    <form [formGroup]="loginForm" (ngSubmit)="login()" >
                        <div class="mb-3">
                          <label for="exampleInputEmail1" class="form-label">Email
address</label>
                          <input formControlName="email" type="email" class="form-
control" id="exampleInputEmail1" aria-describedby="emailHelp">
                          <div id="emailHelp" class="form-text">We'll never share your
email with anyone else.</div>
                        </div>
                        <div class="mb-3">
                          <label for="exampleInputPassword1" class="form-
label">Password</label>
                          <input formControlName="password" type="password" class="form-
control" id="exampleInputPassword1">
                        </div>
                        <button type="submit" class="btn btn-primary">Login</button>
                      </form>

                <a style="color:white; margin-top:10px"routerLink="/signup"> New user?
Click to signup!!</a>
            </div>
        </div>
    </div>
</div>
```

login.component.ts:

```
import { HttpClient } from '@angular/common/http';
import { Component, OnInit } from '@angular/core';
import{ FormGroup,FormBuilder} from '@angular/forms'
import { Router } from '@angular/router';

@Component({
  selector: 'app-login',
  templateUrl: './login.component.html',
  styleUrls: ['./login.component.css']
})
export class LoginComponent implements OnInit {
  public loginForm!:FormGroup;
```

```
  constructor( private fromBuilder : FormBuilder,private http:HttpClient,private
router:Router) { }

  ngOnInit(): void {
    this.loginForm=this.fromBuilder.group({
      email:[''],
      password:['']
    })
  }
  login(){
    this.http.get<any>("http://localhost:3000/signupUsers/")
    .subscribe(res=>{
      const user =res.find((a:any)=>{
        return a.email ===this.loginForm.value.email && a.password
===this.loginForm.value.password
      });
      if(user){
        alert("Login Success!!");
        this.loginForm.reset();
        this.router.navigate(['dashboard'])
      }else{
        alert("user not Found");
      }
    },err=>{
      alert("something Went worng!!")
    })
  }
}
```

api.service.ts:

```
import { Injectable } from '@angular/core';
import{ HttpClient} from '@angular/common/http';
import{map} from 'rxjs/operators'

@Injectable({
  providedIn: 'root'
})
export class ApiService {

  constructor(private http:HttpClient) { }

  postEmployee( data :any){
    return this.http.post<any>("http://localhost:3000/posts",data)
    .pipe(map((res:any)=>{
      return res;
    }))
  }
  getEmployee(){
    return this.http.get<any>("http://localhost:3000/posts")
```

```
    .pipe(map((res:any)=>{
      return res;
    }))
  }
  updateEmployee(data :any,id:number){
    return this.http.put<any>("http://localhost:3000/posts/"+id,data)
    .pipe(map((res:any)=>{
      return res;
    }))
  }
  deleteEmployee(id : number){
    return this.http.delete<any>("http://localhost:3000/posts/"+id)
    .pipe(map((res:any)=>{
      return res;
    }))
  }
}
```

signup.component.css:

```
.card{
    border: none;
    width: 500px;
    padding: 40px;
    position: absolute;
    top:50%;
    left: 50%;
    transform:translate(-50%, -50%);
    background: rgb(13, 137, 232);
    background: linear-gradient(90deg,rgb(230, 81, 160) 0%,rgb(144, 98, 236)100%);


}
```

signup.component.html:

```
<div class="container">
    <div class="row">
        <div class="col-md-6">
            <div class="card">
                <div class="text-center">
                    <h1>Sign Up</h1>
                    <h6>Register Your Self..!</h6>
                </div>
                <form >
                    <form [formGroup]="signupForm" (ngSubmit)="signUp()">
                        <div class="mb-3">
                            <label for="exampleInputPassword1" class="form-label">First
Name</label>
```

```
                          <input formControlName="FirstName" type="text" class="form-
control" id="exampleInputPassword1">
                        </div>
                        <div class="mb-3">
                          <label for="exampleInputPassword1" class="form-label">Last
Name</label>
                          <input formControlName="LastName" type="text" class="form-
control" id="exampleInputPassword1">
                        </div>

                      <div class="mb-3">
                        <label for="exampleInputEmail1" class="form-label">Email
address</label>
                        <input formControlName="email" type="email" class="form-
control" id="exampleInputEmail1" aria-describedby="emailHelp">
                        <div id="emailHelp" class="form-text">We'll never share your
email with anyone else.</div>
                      </div>
                      <div class="mb-3">
                        <label for="exampleInputPassword1" class="form-
label">Password</label>
                        <input  formControlName="password"type="password" class="form-
control" id="exampleInputPassword1">
                      </div>
                      <button type="submit" class="btn btn-primary">Signup</button>
                    </form>
                    <a style="color:white; margin-top:10px" routerLink="/login">
Already register? Click to Login!!</a>
                </form>
            </div>
        </div>
    </div>
</div>
```

signup.component.ts:

```
import { HttpClient } from '@angular/common/http';
import { Component, OnInit } from '@angular/core';
import{FormGroup,FormBuilder}from '@angular/forms'
import { Router } from '@angular/router';

@Component({
  selector: 'app-signup',
  templateUrl: './signup.component.html',
  styleUrls: ['./signup.component.css']
})
export class SignupComponent implements OnInit {

  public signupForm !:FormGroup;
```

```
  constructor(private formBuilder : FormBuilder,private http:HttpClient,private
router:Router) { }

  ngOnInit(): void {
    this.signupForm=this.formBuilder.group({
      FirstName:[''],
      LastName:[''],
      email:[''],
      password:[''],
    })
  }
  signUp(){
    this.http.post<any>("http://localhost:3000/signupUsers/",this.signupForm.value)
    .subscribe(res=>{
      alert("Signup Successfull !!");
      this.signupForm.reset();
      this.router.navigate(['login']);
    },err=>{
      alert("Somethimng went wrong !!")
    })

  }
}
```

db.json:

```
{
  "posts": [
    {
      "id": 1,
      "firstName": "Sebastian",
      "lastName": "Eschweiler",
      "email": "sebastian@codingthesmartway.com"
    },
    {
      "id": 2,
      "firstName": "Steve",
      "lastName": "Palmer",
      "email": "steve@codingthesmartway.com"
    },
    {
      "id": 3,
      "firstName": "Ann",
      "lastName": "Smith",
      "email": "ann@codingthesmartway.com"
    }
  ],
  "signupUsers": [
    {
      "FirstName": "Uma",
      "LastName": "Lakshmi",
      "email": "umalakshmichamanthula@gmail.com",
```

```
      "password": "Uma@1472",
      "id": 3
    },
    {
      "FirstName": "UMA",
      "LastName": "LAKSHMI",
      "email": "umalakshmi@gmail.com",
      "password": "Uma@123",
      "id": 4
    },
    {
      "FirstName": "ani",
      "LastName": "",
      "email": "ani@gmail.com",
      "password": "Ani@1123",
      "id": 5
    },
    {
      "FirstName": "UMA",
      "LastName": "Lakshmi",
      "email": "umalakshmi@gmail.com",
      "password": "Uma@123",
      "id": 6
    }
  ],
  "profile": {
    "name": "typicode"
  }
}
```


app-routing.module.ts:

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { EventdashComponent } from './eventdash/eventdash.component';
import { LoginComponent } from './login/login.component';
import { SignupComponent } from './signup/signup.component';

const routes: Routes = [
  {
    path:'',redirectTo:'login',pathMatch:'full'
  },
  {path:'login',component:LoginComponent},
  {
    path:'signup',component:SignupComponent
  },
  {
    path:'dashboard',component:EventdashComponent
  }
];
```

```
@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

app.component.spec.ts:

```
import { TestBed } from '@angular/core/testing';
import { RouterTestingModule } from '@angular/router/testing';
import { AppComponent } from './app.component';

describe('AppComponent', () => {
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      imports: [
        RouterTestingModule
      ],
      declarations: [
        AppComponent
      ],
    }).compileComponents();
  });

  it('should create the app', () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app).toBeTruthy();
  });

  it(`should have as title 'eventmanager'`, () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app.title).toEqual('eventmanager');
  });

  it('should render title', () => {
    const fixture = TestBed.createComponent(AppComponent);
    fixture.detectChanges();
    const compiled = fixture.nativeElement as HTMLElement;
    expect(compiled.querySelector('.content span')?.textContent).toContain('eventmanager
app is running!');
  });
});
```

app.component.html:

```
<router-outlet>

</router-outlet>
```

app.component.ts:

```
import { Component } from '@angular/core';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'eventmanager';
}
```

app.module.ts:

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import{ReactiveFormsModule} from '@angular/forms'

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { EventdashComponent } from './eventdash/eventdash.component';
import { HttpClientModule } from '@angular/common/http';
import { LoginComponent } from './login/login.component';
import { SignupComponent } from './signup/signup.component';

@NgModule({
  declarations: [
    AppComponent,
    EventdashComponent,
    LoginComponent,
    SignupComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    ReactiveFormsModule,
    HttpClientModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

index.html:

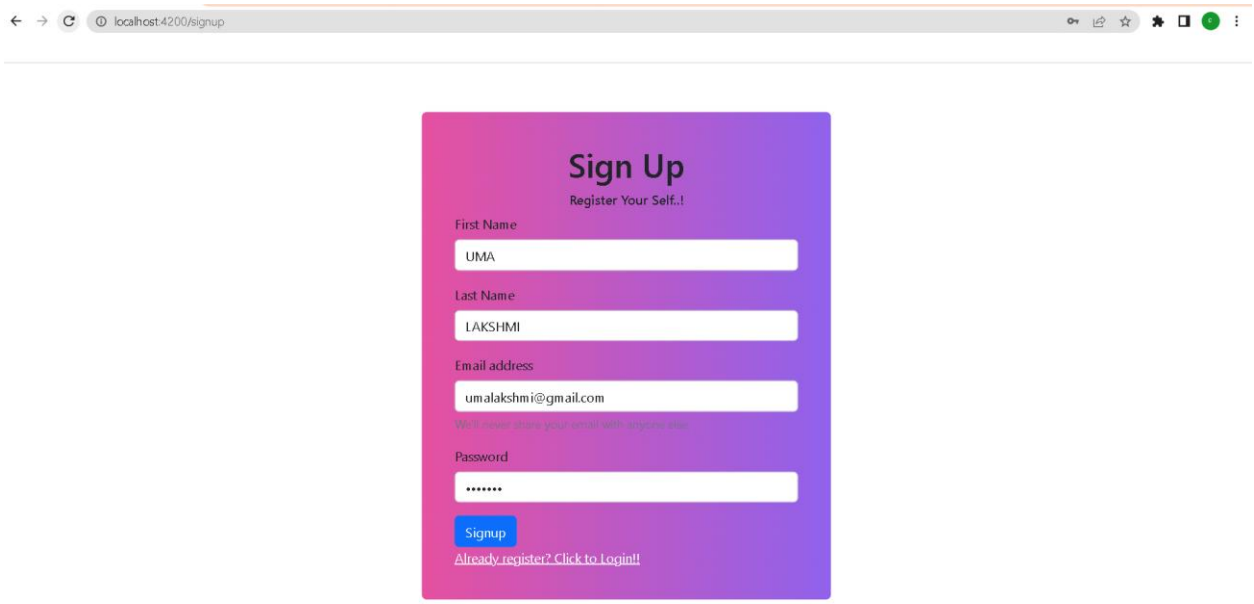
```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Eventmanager</title>
```

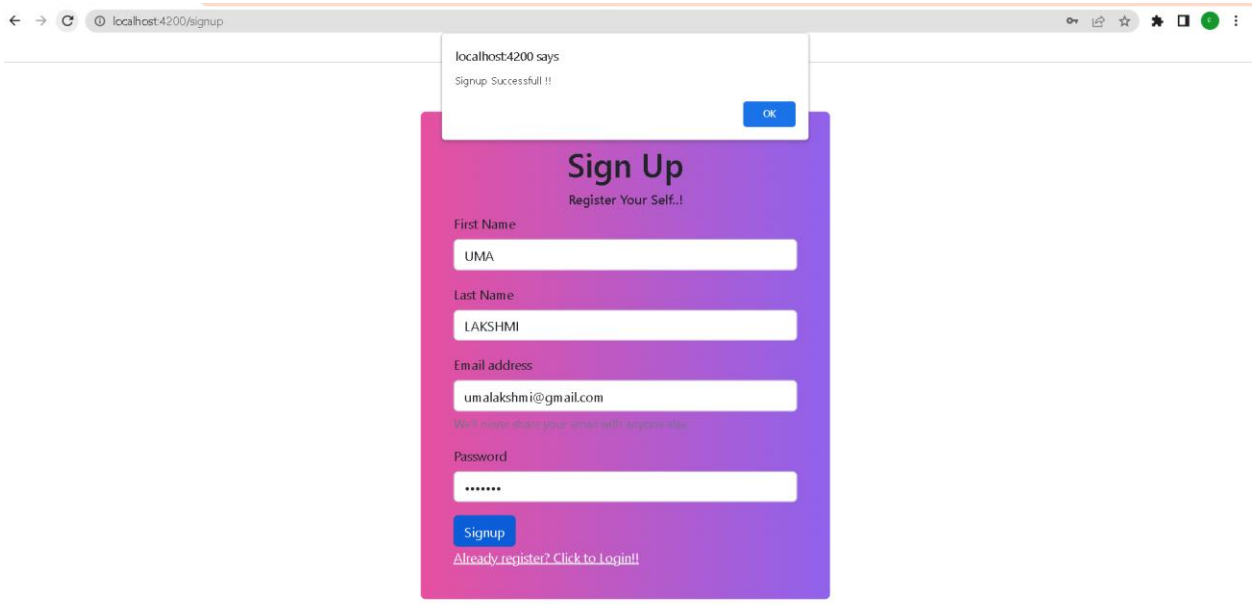
```
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
  <!-- CSS only -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT"
crossorigin="anonymous">
<!-- JavaScript Bundle with Popper -->
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"
integrity="sha384-u1OknCvxWvY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTPOOmMi466C8"
crossorigin="anonymous"></script>

</head>
<body>
  <app-root></app-root>
</body>
</html>
```

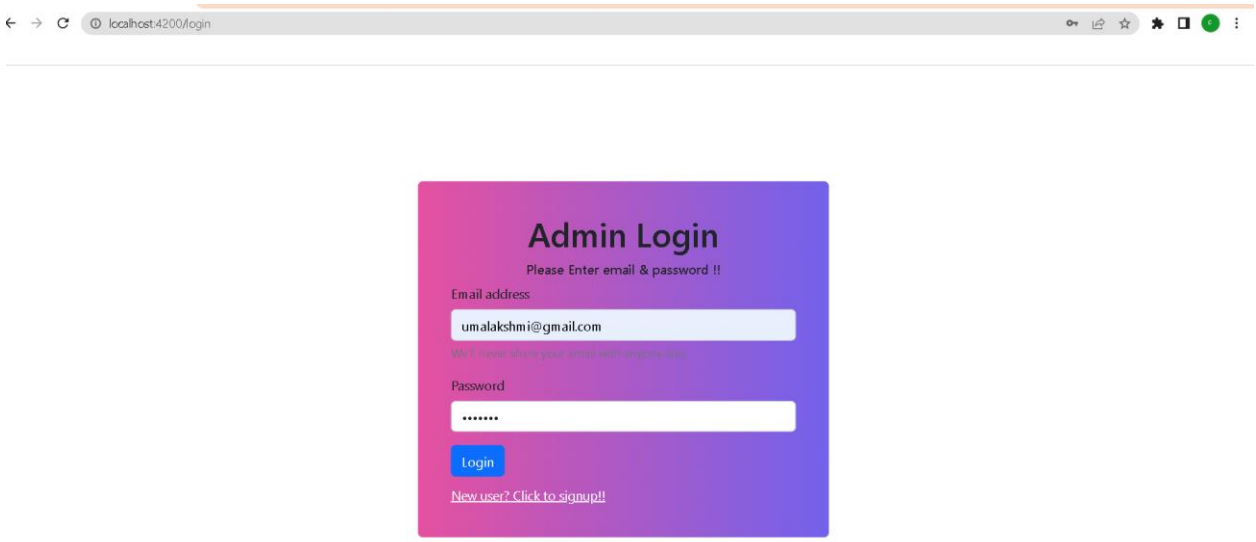
main.ts:

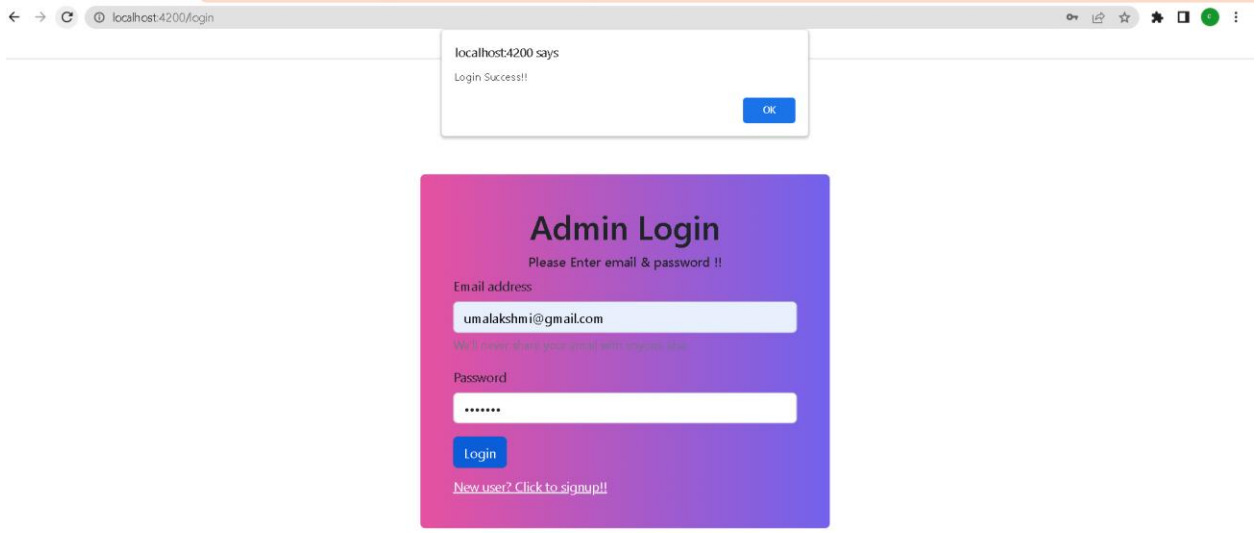
```
import { enableProdMode } from '@angular/core';
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';

import { AppModule } from './app/app.module';
import { environment } from './environments/environment';

if (environment.production) {
  enableProdMode();
}

platformBrowserDynamic().bootstrapModule(AppModule)
  .catch(err => console.error(err));
```

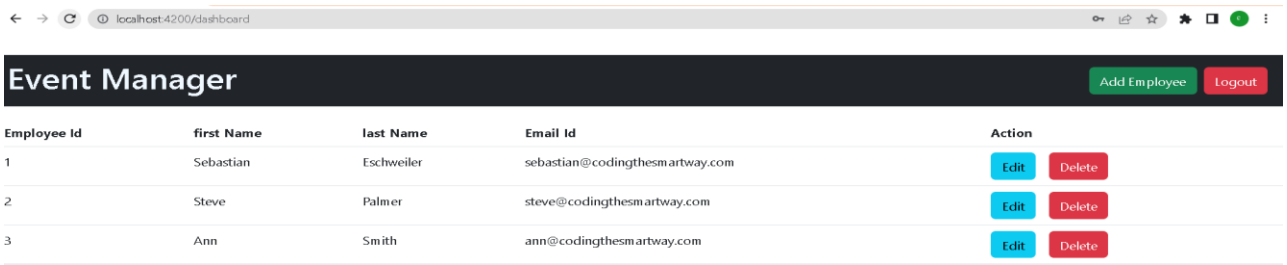
OUTPUT:

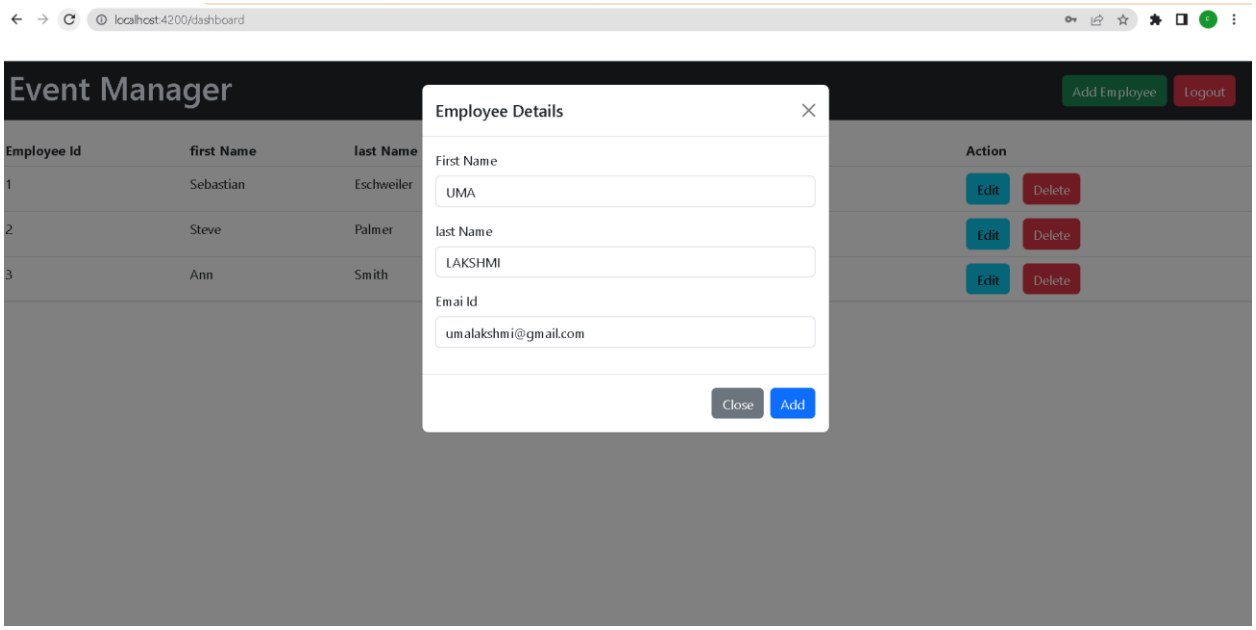

localhost:4200/signup
Sign Up
Register Your Self..!
First Name
UMA
Last Name
LAKSHMI
Email address
umalakshmi@gmail.com
Password
Signup
Already register? Click to Login!!

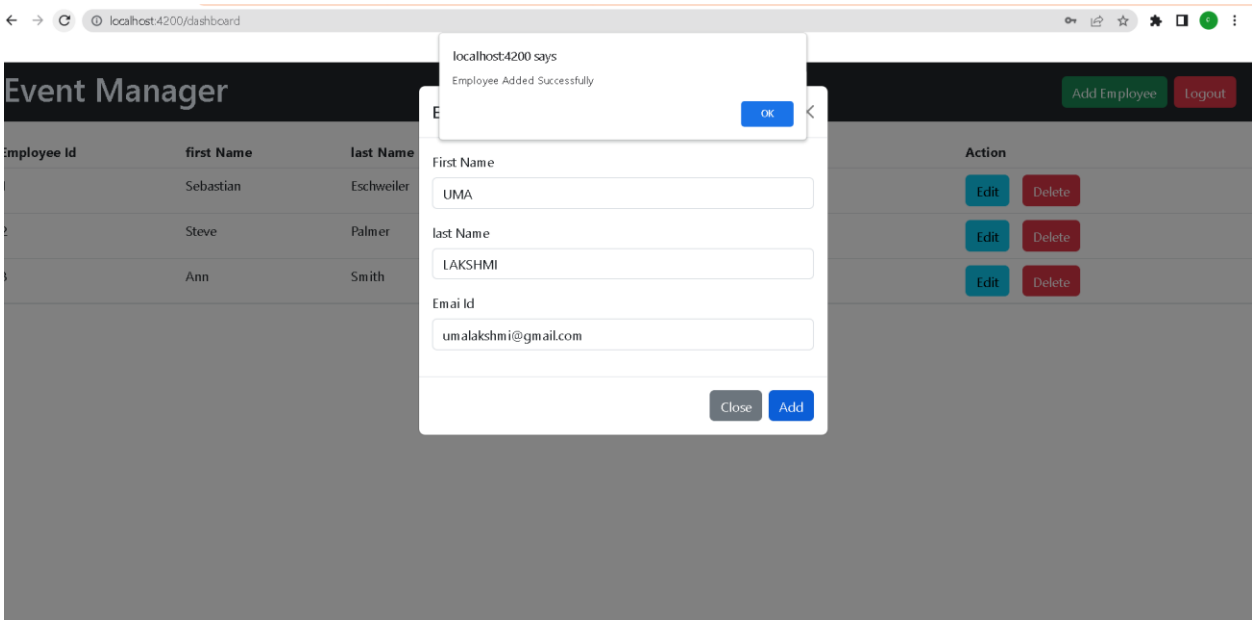

localhost:4200/signup
localhost:4200 says
Signup Successfull !!
OK
Sign Up
Register Your Self..!
First Name
UMA
Last Name
LAKSHMI
Email address
umalakshmi@gmail.com
Password
Signup
Already register? Click to Login!!

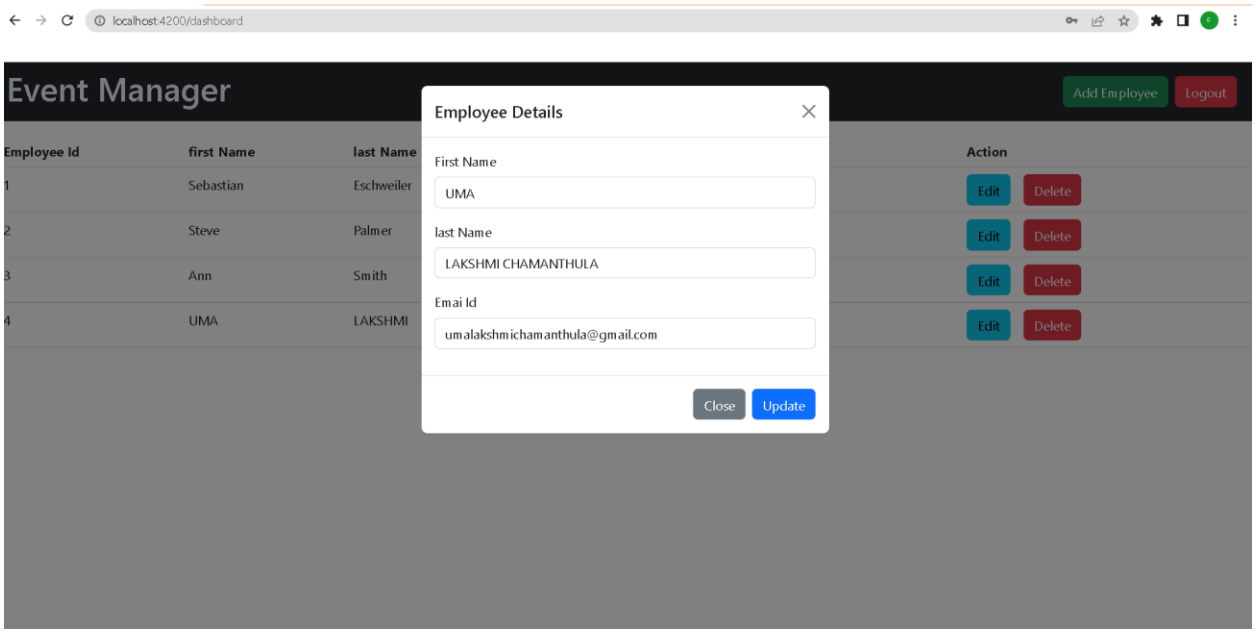

localhost:4200/login
Admin Login
Please Enter email & password !!
Email address
umalakshmi@gmail.com
Password
Login
New user? Click to signup!!

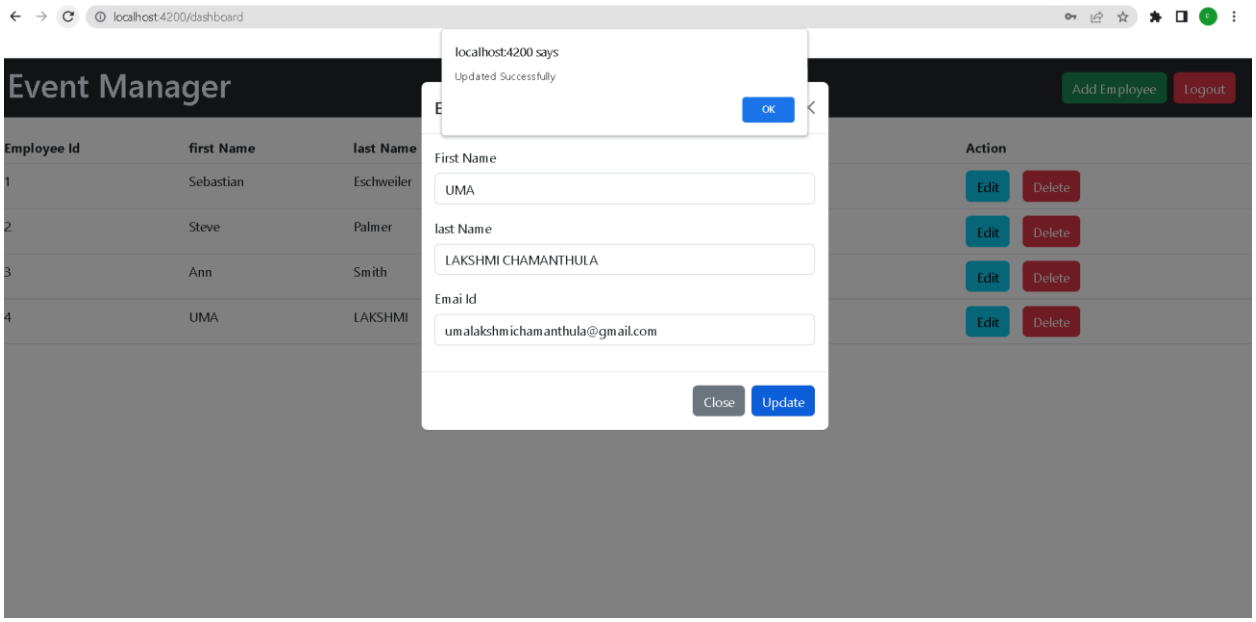

localhost:4200/login
localhost:4200 says
Login Success!!
OK
Admin Login
Please Enter email & password !!
Email address
umalakshmi@gmail.com
Password
Login
New user? Click to signup!!

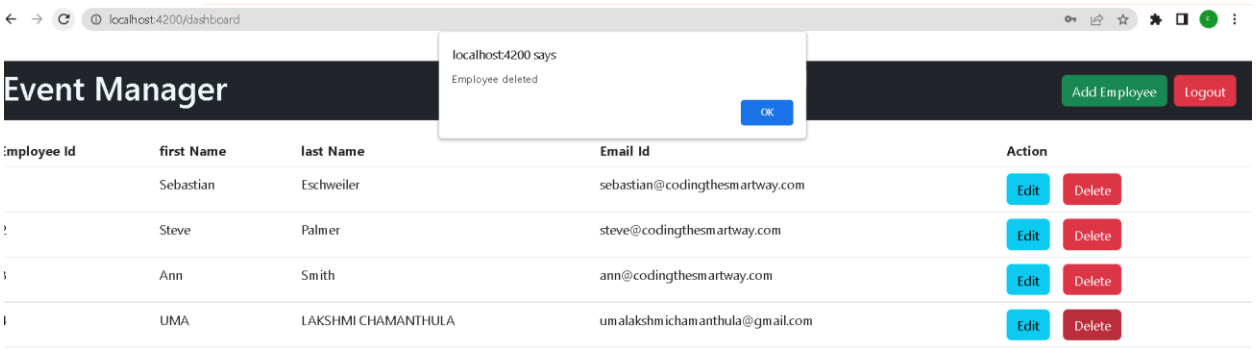

localhost:4200/dashboard
Event Manager
Add Employee
Logout
Employee Id
first Name
last Name
Email Id
Action
1
Sebastian
Eschweiler
sebastian@codingthesmartway.com
Edit
Delete
2
Steve
Palmer
steve@codingthesmartway.com
Edit
Delete
3
Ann
Smith
ann@codingthesmartway.com
Edit
Delete

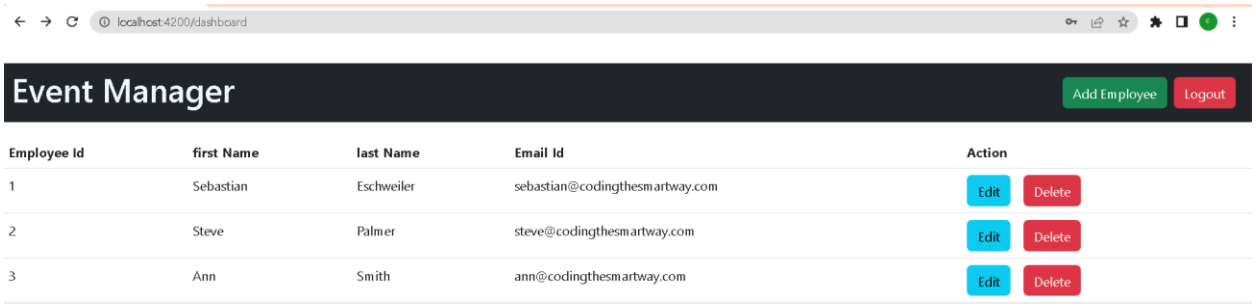

localhost:4200/dashboard
Event Manager
Add Employee
Logout
Employee Details
First Name
UMA
last Name
LAKSHMI
Emai Id
umalakshmi@gmail.com
Close
Add
Employee Id
first Name
last Name
Action
1
Sebastian
Eschweiler
Edit
Delete
2
Steve
Palmer
Edit
Delete
3
Ann
Smith
Edit
Delete


localhost:4200/dashboard
localhost:4200 says
Employee Added Successfully
OK
Event Manager
Add Employee
Logout
Employee Id
first Name
last Name
Action
Sebastian
Eschweiler
Steve
Palmer
Ann
Smith
Edit
Delete
First Name
UMA
last Name
LAKSHMI
Emai Id
umalakshmi@gmail.com
Close
Add


localhost:4200/dashboard
Event Manager
Add Employee
Logout
Employee Details
Employee Id
first Name
last Name
Action
1
Sebastian
Eschweiler
2
Steve
Palmer
3
Ann
Smith
4
UMA
LAKSHMI
Edit
Delete
First Name
UMA
last Name
LAKSHMI CHAMANTHULA
Emai Id
umalakshmichamanthula@gmail.com
Close
Update


localhost:4200/dashboard
localhost:4200 says
Updated Successfully
OK
Event Manager
Add Employee
Logout
Employee Id
first Name
last Name
Action
1
Sebastian
Eschweiler
Edit
Delete
2
Steve
Palmer
Edit
Delete
3
Ann
Smith
Edit
Delete
4
UMA
LAKSHMI
Edit
Delete
First Name
UMA
last Name
LAKSHMI CHAMANTHULA
Emai Id
umalakshmichamanthula@gmail.com
Close
Update


localhost:4200/dashboard
localhost:4200 says
Employee deleted
OK
Event Manager
Add Employee
Logout
Employee Id
first Name
last Name
Email Id
Action
Sebastian
Eschweiler
sebastian@codingthesmartway.com
Edit
Delete
Steve
Palmer
steve@codingthesmartway.com
Edit
Delete
Ann
Smith
ann@codingthesmartway.com
Edit
Delete
UMA
LAKSHMI CHAMANTHULA
umalakshmichamanthula@gmail.com
Edit
Delete


localhost:4200/dashboard
Event Manager
Add Employee
Logout
Employee Id
first Name
last Name
Email Id
Action
1
Sebastian
Eschweiler
sebastian@codingthesmartway.com
Edit
Delete
2
Steve
Palmer
steve@codingthesmartway.com
Edit
Delete
3
Ann
Smith
ann@codingthesmartway.com
Edit
Delete