

Event Management App;

install and Json-server :

db.json file---->

```
{
  "employees": [
    {
      "id": 2,
      "firstname": "Steve",
      "lastname": "Palmer",
      "email": "steve@codingthesmartway.com"
    },
    {
      "id": 3,
      "firstname": "Ann",
      "lastname": "Smith",
      "email": "ann@codingthesmartway.com"
    },
    {
      "firstname": "Harry",
      "lastname": "Kumar",
      "email": "hkumar01@example.com",
      "id": 4
    },
    {
      "firstname": "Max",
      "lastname": "Jhonson",
      "email": "max11@example.com",
      "id": 6
    },
    {
      "firstname": "modi",
      "lastname": "n",
      "email": "modi@n.pm",
      "id": 7
    }
  ]
}
```

App.Module.ts

```

import { NgModule } from '@angular/core';
import { FormsModule, ReactiveFormsModule } from '@angular/forms';
import { BrowserModule } from '@angular/platform-browser';
import { HttpClientModule } from '@angular/common/http';

import { AppComponent } from './app.component';
import { HomeComponent } from './home/home.component';
import { AdminComponent } from './admin/admin.component';
import { UpdatepasswordComponent } from
'./updatepassword/updatepassword.component';
import { EmployeeComponent } from './employee/employee.component';
import { RouterModule, Routes } from '@angular/router';

const appRoutes: Routes = [
  { path: '', component: HomeComponent },
  { path: 'admin', component: AdminComponent },
  { path: 'employee', component: EmployeeComponent },
  { path: 'updatepassword', component: UpdatepasswordComponent },
];

@NgModule({
  declarations: [
    AppComponent,
    HomeComponent,
    AdminComponent,
    UpdatepasswordComponent,
    EmployeeComponent
  ],
  imports: [
    BrowserModule,FormsModule,HttpClientModule,ReactiveFormsModule,RouterModule.forRoot
    (appRoutes)
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }

```

app.component.ts

```

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

```

```

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

```

app.component.html

```

<div class="container">
  <div class="row">
    <div class="col1">
      <ul class="nav nav-tabs">
        <li
          role="presentation"
          class="nav-item"
          routerLinkActive="active"
          [routerLinkActiveOptions]="{ exact: true }"
        >
          <a class="nav-link" routerLink="/" style="text-decoration: none"
            ><button>Home</button></a>
          </li>
        <br />
        <li role="presentation" class="nav-item" routerLinkActive="active">
          <a class="nav-link" routerLink="admin" style="text-decoration: none"
            ><button>Admin Login</button></a>
          </li>
        <br />
        <li role="presentation" class="nav-item" routerLinkActive="active">
          <a
            class="nav-link"
            routerLink="updatepassword"
            style="text-decoration: none"
            ><button>Change/update Password</button></a>
          </li>
      </ul>
    </div>
  </div>
  <div class="row">
    <div class="col2">
      <router-outlet></router-outlet>
    </div>
  </div>
</div>

```

</div>

app.component.css

```
ul
{
list-style-type: none;
}
.container{
    background-color: rgb(213, 225, 232);
    width: 100%;
    margin: 5px;
    padding: 5px;

}
.coll{
    background-color: rgb(167, 234, 239);
    padding: 5px;
margin: 5px;
}
```

admin.component.ts

```
import { Component, OnInit } from '@angular/core';
import { FormControl, FormGroup, Validators } from '@angular/forms';
import { Router } from '@angular/router';
```

```
@Component({
    selector: 'app-admin',
    templateUrl: './admin.component.html',
    styleUrls: ['./admin.component.css'],
})
export class AdminComponent implements OnInit {
    constructor(private router: Router) {}
```

```
    signupForm: FormGroup;
```

```
    ngOnInit(): void {
        this.signupForm = new FormGroup({
```

```

        username: new FormControl(null, Validators.required),
        password: new FormControl(null, [Validators.required]),
    });
}

onSubmit() {
    console.log(this.signupForm.value);
    this.router.navigate(['/employee']);
}
}

```

admin.component.html

```

<div
  class="container m-2 border justify-center"
  style="width: 400px; background-color: rgb(205, 196, 196)"
>
  <h4 class="text-center">Login Page</h4>
  <form [formGroup]="signupForm" (ngSubmit)="onSubmit()" class="form-center">
    <div class="row">
      <div class="form-group">
        <label for="username">Username</label>
        <input
          type="text"
          id="username"
          class="form-control"
          formControlName="username"
        />
        <span
          class="small"
          *ngIf="
            !signupForm.get('username').valid &&
            signupForm.get('username').touched
          ">
          >Please enter a valid username</span>
        </div>
      </div>
      <div class="row">
        <div class="form-group">
          <label for="password">Password</label>
          <input
            type="text"
            id="password"
            class="form-control"
            formControlName="password"
          >
        </div>
      </div>
    </div>
  </form>

```

```

    />
    <span
      class="small"
      *ngIf="
        !signupForm.get('password').valid &&
        signupForm.get('password').touched
      "
    >Incorrest Password!</span>
  >
</div>
</div>
<button class="btn btn-primary m-2" type="submit">Submit</button>
</form>
</div>

```

admin.componenty.css

```

input.ng-invalid.ng-touched {
  border: 1px solid red;
}

```

home.component.ts

```

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

```

```

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

  constructor(private router: Router) {}

  ngOnInit(): void {}

  gotoAdmin() {
    this.router.navigate(['/admin']);
  }
}

```

```
}
```

home.component.html

```
<h4 class="text-center">Welcome to Event Management App ....!!</h4>
<hr class="new6" />
<p class="text-center"><b>Please Login to Continue!!</b></p>
<div class="container mt-50 text-center">
  <button class="btn btn-info" (click)="gotoAdmin()">LogIn</button>
</div>
```

home.component.css

```
hr.new5 {
  border: 8px solid #6D214F;
  border-radius: 5px;
}
```

employee.component.ts

```
import { HttpClient } from '@angular/common/http';
import { Component, OnInit } from '@angular/core';
import { NgForm } from '@angular/forms';
import { map } from 'rxjs/operators';
import { EmpData } from './empdata.model';
```

```
@Component({
  selector: 'app-employee',
  templateUrl: './employee.component.html',
  styleUrls: ['./employee.component.css'],
})
export class EmployeeComponent implements OnInit {
  fetchedEmp: EmpData[] =[];
  backendurl = 'http://localhost:3000/employees';
  constructor(private http: HttpClient) {}

  ngOnInit(): void {
```

```

    this.fetchEmp();
  }

  onCreateEmp(empData: { firstname: string; lastname: string;email: string; },
form: NgForm) {
    this.http.post(this.backendurl, empData).subscribe((responseData) => {
      console.log(responseData);
      this.fetchEmp();
      form.reset();
    });
  }
  onFetchEmps() {
    this.fetchEmp();
  }

  onDeletePost(id: number) {
    this.http.delete(this.backendurl + '/' + id).subscribe((response) => {
      console.log('Employee deleted: ' + response);
      this.fetchEmp();
    });
  }

  updateEmployee(id:number,empData: { firstname: string; lastname: string;email:
string;}, form: NgForm ) {
    this.http.put(this.backendurl + '/' + id,empData).subscribe((responseData) => {
      console.log(responseData);
      this.fetchEmp();
      form.reset();
    });
  }

  fetchEmp() {
    this.http
      .get(this.backendurl)
      .pipe(
        map((responseData) => {
          const empsArray: EmpData[] = [];
          for (const key in responseData) {
            var x = { ...responseData[key] };
            empsArray.push(x);
          }
          return empsArray;
        })
      )
      .subscribe((employee) => {
        this.fetchedEmp = employee;
      });
  }

```



```
}
```

employee.component.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Event Management App</title>
    <link
      rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css"
    />
  </head>

  <body>
    <div style="background-color: #438cc4">
      <nav>
        <div
          class="navbar navbar-dark navbar-expand-sm"
          style="background-color: #a5a7cb"
        >
          <div class="container">
            <div class="container2">
              
              
            </div>
          </div>
          <div>
            <a class="navbar-brand" href="#" style="text-decoration: none"
              ><i>Employee Management App</i></a>
          </div>
        </div>
      </nav>
    </div>
  </body>
</html>
```

```

</div>
<button
  class="navbar-toggler"
  data-toggle="collapse"
  data-target="#navbarNav2"
  type="button"
>
  <span class="navbar-toggler-icon"></span>
</button>
<div
  class="collapse navbar-collapse justify-content-end"
  id="navbarNav2"
>
  <ul class="navbar-nav" style="color: #ffffff">
    <li class="nav-item">
      <a
        class="nav-link"
        href="/employee#all"
        style="text-decoration: none"
      >All Employee</a
    >
    </li>
    <li class="nav-item">
      <a
        class="nav-link"
        href="/employee#new"
        style="text-decoration: none"
      >New Employee</a
    >
    </li>
    <li class="nav-item">
      <a
        class="nav-link"
        href="/employee#update"
        style="text-decoration: none"
      >Update Employee</a
    >
    </li>
    <li class="nav-item">
      <a
        class="nav-link"
        href="/employee#delete"
        style="text-decoration: none"
      >Delete Employee</a
    >
    </li>
  </ul>
</div>
</div>
</div>

```

```

</nav>

<div class="row m-3" id="all">
  <div class="col">
    <h1 style="color: #ffffff">All Employee Details</h1>
    <p *ngIf="fetchedEmp.length < 1">No Employees Available !!</p>
    <ul class="list-group" *ngIf="fetchedEmp.length >= 1">
      <li
        class="list-group-item border border-circular"
        *ngFor="let emp of fetchedEmp"
        style="width: 20%; background-color: #d2dae2"
      >
        <p>{{ emp.firstname }}</p>
        <p>{{ emp.lastname }}</p>
        <p>{{ emp.email }}</p>
      </li>
    </ul>
  </div>
</div>
<hr class="new1" />
<div
  class="container m-2 justify-content-center"
  style="width: 100%"
  id="new"
>
  <h1 style="color: #ffffff">Add New Employee</h1>
  <form #f="ngForm">
    <div class="row m-3">
      <div class="form-group align-center" style="color: #ffffff">
        <label for="firstname">FirstName</label>
        <input
          style="background-color: #d2dae2"
          type="text"
          id="firstname"
          class="form-control"
          name="firstname"
          ngModel
          required
        />
      </div>
    </div>
    <div class="row m-3">
      <div class="form-group" style="color: #ffffff">
        <label for="lastname">LastName</label>
        <input
          style="background-color: #d2dae2"
          type="text"
          id="lastname"
          class="form-control"
          name="lastname"
        >
      </div>
    </div>
  </form>
</div>

```

```

        ngModel
        required
    />
</div>
</div>

<div class="row m-3">
    <div class="form-group" style="color: #ffffff">
        <label for="email">Email</label>
        <input
            style="background-color: #d2dae2"
            id="email"
            name="email"
            class="form-control"
            ngModel
            required
        />
    </div>
</div>
<div class="row m-3">
    <div class="col">
        <button
            class="btn btn-info btn-sm"
            [disabled]="!f.valid"
            (click)="onCreateEmp(f.value, f)"
        >
            ADD
        </button>
    </div>
</div>
</form>

<hr class="new1" />

<div
    class="container m-2 justify-center"
    id="update"
    style="width: 100%"
>
    <h1 style="color: #ffffff">Update Employee</h1>
    <form #f="ngForm">
        <div class="row m-3">
            <div class="form-group" style="color: #ffffff">
                <label for="id">ID</label>
                <input
                    style="background-color: #d2dae2"
                    type="numeric"
                    id="id"
                    class="form-control"
                    name="id"

```

```

        ngModel
        required
    />
</div>
</div>
<div class="row m-3">
    <div class="form-group" style="color: #ffffff">
        <label for="data">New Email</label>
        <input
            style="background-color: #d2dae2"
            type="text"
            id="data"
            class="form-control"
            name="data"
            ngModel
            required
        />
    </div>
</div>

<div class="row m-3">
    <div class="col">
        <button
            (click)="updateEmployee(f.id, f.value, f)"
            type="button"
            data-bs-toggle="modal"
            data-bs-target="#exampleModal"
            class="btn btn-info"
        >
            Update
        </button>
    </div>
</div>
</form>
</div>

<hr class="new1" />

<div class="row m-3" id="delete" style="width: 100%">
    <div class="col">
        <h1 style="color: #ffffff">Delete Employee</h1>
        <p
            style="color: #ffffff"
            class="text-start"
            *ngIf="fetchedEmp.length < 1"
        >
            No Employees Available !!
        </p>
        <ul class="list-group" *ngIf="fetchedEmp.length >= 1">
            <li

```

```

        style="width: ; background-color: #d2dae2"
        class="list-group-item"
        *ngFor="let emp of fetchedEmp"
    >
        <p>{{ emp.firstname }}</p>
        <p>{{ emp.lastname }}</p>
        <p>{{ emp.email }}</p>
        <button
            class="btn btn-sm btn-danger float-right"
            (click)="onDeletePost(emp.id)"
        >
            Delete
        </button>
    </li>
</ul>
</div>
</div>
<hr class="new1" />
</div>
</div>
</body>
</html>

```

employee.component.css

```

hr.new1 {
    border-top: 1px dashed white;
}
.container2{
    align-items: center;

    height: 25%;
}

```

employee.model.ts

```
export interface EmpData {  
  id: number;  
  firstname: string;  
  lastname: string;  
  email: string;  
}
```

updatepassword.component.ts

```
import { Component, OnInit } from '@angular/core';  
import { FormControl, FormGroup, Validators } from '@angular/forms';  
import { Router } from '@angular/router';
```

```
@Component({  
  selector: 'app-updatepassword',  
  templateUrl: './updatepassword.component.html',  
  styleUrls: ['./updatepassword.component.css']  
})  
export class UpdatepasswordComponent implements OnInit {  
  constructor(private router: Router) {}
```

```
  signupForm: FormGroup;
```

```
  ngOnInit(): void {  
    this.signupForm = new FormGroup({  
      username: new FormControl(null, Validators.required),  
      password: new FormControl(null, [Validators.required]),  
    });  
  }
```

```
  onSubmit() {  
    console.log(this.signupForm.value);  
    this.router.navigate(['/admin']);  
  }
```

```
}
```

updatepassword.component.html

```
<div
  class="container m-2 border justify-center"
  style="width: 400px; background-color: darkgrey"
>
  <h4 class="text-center">Update Password</h4>
  <form [formGroup]="signupForm" (ngSubmit)="onSubmit()">
    <div class="row">
      <div class="form-group">
        <label for="username">Username</label>
        <input
          type="text"
          id="username"
          class="form-control"
          formControlName="username"
        />
        <span
          class="small"
          *ngIf="
            !signupForm.get('username').valid &&
            signupForm.get('username').touched
            "
          >Please enter a valid username</span>
      </div>
    </div>
    <div class="row">
      <div class="form-group">
        <label for="password">NewPassword</label>
        <input
          type="text"
          id="password"
          class="form-control"
          formControlName="password"
        />
        <span
          class="small"
          *ngIf="
            !signupForm.get('password').valid &&
            signupForm.get('password').touched
            "
          >Incorrest Password!</span>
      </div>
    </div>
  </form>
  <button class="btn btn-primary m-2" type="submit">Update</button>
```



```
</form>  
</div>
```

updatepassword.component.css

```
input.ng-invalid.ng-touched {  
  border: 1px solid red;  
}
```

the Whole code of all components are added here.

the Screen Shots of the code are Added in another pdf file.