

Project: Event Management Application.

Source Code-

Db.json-

```
{
  "posts": [
    {
      "id": 1,
      "title": "json-server",
      "author": "typicode"
    }
  ],
  "comments": [
    {
      "id": 1,
      "body": "some comment",
      "postId": 1
    }
  ],
  "users": [
    {
      "fullname": "Gaurav Mishra",
      "email": "gaurav0143@gmail.com",
      "password": "Gaurav@123",
      "mobile": "8525635645",
      "id": 6
    },
    {
      "fullname": "Nikhil Bhardwaj",
      "email": "nikhil@gmail.com",
      "password": "123456",
      "mobile": "1234567890",
      "id": 7
    },
    {
      "id": 8,
      "fullname": "Rajendra Kumar Mishra",
      "mobile": "7454",
      "email": "gaurav0143@gmail.com",
      "password": "Gaurav@123"
    },
    {
      "fullname": "Ram Kumar",
      "email": "mnhh@gmail.com",
      "password": "1234",
      "mobile": "7856221877",
      "id": 9
    }
  ]
}
```

```

    },
    {
      "fullname": "Angoor",
      "email": "angur0143@gmail.com",
      "password": "0000",
      "mobile": "Tiwari",
      "id": 10
    },
    {
      "fullname": "Rajendra Kumar Mishra",
      "email": "gaurav0143@gmail.com",
      "password": "Gaurav@123",
      "mobile": "7454",
      "id": 11
    },
    {
      "fullname": "Rajendra Kumar Mishra",
      "email": "gaurav0143@gmail.com",
      "password": "Gaurav@123",
      "mobile": "74544",
      "id": 12
    },
    {
      "id": 13,
      "fullname": "Alok Pandey",
      "mobile": "7856489885",
      "email": "alokji0143@gmail.com",
      "password": "1234"
    },
    {
      "fullname": "Alok Pandey",
      "email": "alokji0143@gmail.com",
      "password": "1234",
      "mobile": "7856489885",
      "id": 14
    }
  ],
  "profile": {
    "name": "typicode"
  }
}

```

Package.json-

```

{
  "name": "event-management",
  "version": "0.0.0",
  "scripts": {

```

```

    "ng": "ng",
    "start": "ng serve",
    "build": "ng build",
    "json:server": "json-server --watch db.json",
    "watch": "ng build --watch --configuration development",
    "test": "ng test"
  },
  "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": "^14.2.3",
    "@angular/cli": "~14.2.3",
    "@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"
  }
}

```

App.module.ts-

```

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

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { LoginComponent } from './login/login.component';

import { CrudOperationComponent } from './crud-operation/crud-
operation.component';

```

```

import { FormsModule } from '@angular/forms';
import { ReactiveFormsModule } from '@angular/forms';
import { HttpClientModule } from '@angular/common/http';

@NgModule({
  declarations: [
    AppComponent,
    LoginComponent,

    CrudOperationComponent,

  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    FormsModule,
    HttpClientModule,
    ReactiveFormsModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }

```

app.components.ts-

```

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

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

```

App-routing.module.ts-

```

import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';

import { CrudOperationComponent } from '../crud-operation/crud-operation.component';
import { LoginComponent } from '../login/login.component';

```

```

const routes: Routes = [
  {path: '',redirectTo:'login',pathMatch:'full'},
  {path:'login',component:LoginComponent},

{path:'ManageUser',component:CrudOperationComponent},
];

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

```

login.component.html-

```

<div class="container">
  <div class="row">
    <div class="col-md-6">
      <div class="card">
        <div class="text-center">
          <h1>Login</h1>
          <h6>Please enter Email & password !!</h6>
        </div>
        <form [formGroup]="loginForm" (ngSubmit)="login()">
          <div class="form-group">
            <label for="exampleInputEmail1">Email address</label>
            <input formControlName="email" type="email" class="form-
control" id="exampleInputEmail1" aria-describedby="emailHelp"
placeholder="Enter email">
            <small id="emailHelp" class="form-text text-muted
">We'll never share your email with anyone else.</small>
          </div>
          <div class="form-group">
            <label for="exampleInputPassword1">Password</label>
            <input formControlName="password" type="password"
class="form-control" id="exampleInputPassword1" placeholder="Password">
          </div>

          <button type="submit" class="btn btn-primary
">Login</button>
        </form>
      </div>
    </div>
  </div>
</div>

```

login.component.css-

```
.card{
  margin-top: 10px;
  border: none;
  width: 500px;
  padding: 40px;
  position: absolute;
  top: 50%;
  left: 55%;
  background: linear-gradient(90deg, rgb(189, 171, 16) 0%,rgb(36, 36, 97)
100%);
  color:white;
  font-size: 16px;
}
a{
  margin-top: 20px;
  color: rgb(237, 69, 237);
  font-size: 20px;
  font-weight: 500;
}
h1{
  font-size: 32px;
  font-weight: 500;
  color: white;
}
.btn{
  align-items: center;
  text-align: center;
  margin-left: 25% ;
  width: 200px;
  font-weight: 500;
  background-color: rgb(7, 43, 249);
  border: none;
  color: white;
}
```

login.component.ts-

```
import { HttpClient } from '@angular/common/http';
import { Component, OnInit } from '@angular/core';
import { FormGroup, FormBuilder } from '@angular/forms'
import { Route, 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 formBuilder: FormBuilder, private
HttpClient: HttpClient, private route: Router) { }

  ngOnInit(): void {
    this.loginForm=this.formBuilder.group({
      email:[''],
      password:['' ]
    })
  }

  login(){

    this.HttpClient.get<any>("http://localhost:3000/users")
    .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 successful");
        this.loginForm.reset();
        this.route.navigate(["ManageUser"])
      }
      else{
        alert("Incorrect Credentials")
      }
    })
  }
}

```

Crud-operation.component.html-

```

<div class="container">
  <div class="row">
    <h2 style="color: white;">Event Management</h2>
    <button type="button" class="btn btn-primary" data-toggle="modal" data-
target="#exampleModal"> Add User</button>
  </div>
  <table class="table table-bordered table-striped">

```

```

<thead>

<tr>

<th>Id</th>

<th>Name</th>

<th>UserName</th>

<th>Email</th>
<th>Action</th>

</tr>

</thead>

<tbody>

<tr *ngFor="let u of users">

<td>{{u.id}}</td>

<td>{{u.fullname}}</td>

<td>{{u.mobile}}</td>

<td>{{u.email}}</td>
<td>
        <button class="btn btn-success me-2" type="button"
(click)="UpdateUser(u)" data-toggle="modal" data-
target="#exampleModal">Edit</button>
        <button class="btn btn-danger"
(click)="DeleteUser(u)">Delete</button>
</td>
</tr>

</tbody>

</table>

<!-- Modal -->
<div class="modal fade" id="exampleModal" tabindex="-1" role="dialog"
aria-labelledby="exampleModalLabel" aria-hidden="true">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">

```



```

        <h5 class="modal-title" id="exampleModalLabel">User Details</h5>
        <button type="button" class="close" data-dismiss="modal" aria-
label="Close">
            <span aria-hidden="true">&times;</span>
        </button>
    </div>
    <div class="modal-body">
        <form [formGroup]="signupForm" (ngSubmit)="AddUser()">
            <div class="form-group">
                <label for="exampleInputEmail1">Full Name</label>
                <input formControlName="fullname" type="text"
class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp"
placeholder="Enter your name">
            </div>
            <div class="form-group">
                <label for="exampleInputPassword1">Mobile Number</label>
                <input formControlName="mobile" type="text" class="form-
control" id="exampleInputPassword1" placeholder="Enter phone number">
            </div>
            <div class="form-group">
                <label for="exampleInputEmail1">Email</label>
                <input formControlName="email" type="email"
class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp"
placeholder="Enter email">
            </div>
            <div class="form-group">
                <label for="exampleInputEmail1">Password</label>
                <input formControlName="password" type="password"
class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp"
placeholder="Enter password">
            </div>

            <div class="modal-footer">
                <button type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>
                <button *ngIf="btnSaveShow" type="submit" id="add" class="btn
btn-primary" >Add User</button>
                <button *ngIf="btnUpdateShow" (click)="UpdateUserDetails()"
class="btn btn-primary" >Update</button>
            </div>
        </form>
    </div>
</div>
</div>

```

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

Crud-operation.component.css-

```
.row{
  background: linear-gradient(90deg,rgb(194, 183, 31) 0%,rgb(33, 33, 87)
100%);
  display: flex;
  justify-content: space-between;
  margin: 3px;
}
.row button{
  width: 200px;
  height: 3rem;
  margin: 30px;
  background-color: green;
  color: white;
  padding:2px;
  border: none;
  transition: 1s;
}
.row button:hover{
  background-color: rgb(54, 90, 54);
}
.btn{
  margin: 10px;
}
.container{
  margin-top: 30px;
}
.row h2{
  margin: 20px;
}
```

Crud-operation.component.ts-

```
import { Component, OnInit } from '@angular/core';
import { ListService } from '../list.service';
import { UserClass } from '../UserClass';
import { FormGroup, FormBuilder } from '@angular/forms'
import { HttpClient } from '@angular/common/http';
import { Router } from '@angular/router';
@Component({
```

```

    selector: 'app-crud-operation',
    templateUrl: './crud-operation.component.html',
    styleUrls: ['./crud-operation.component.css']
  })
}
export class CrudOperationComponent implements OnInit {

  constructor(private service:ListService,private formBuilder:FormBuilder
,private http:HttpClient,private router:Router) { }
  public signupForm!:FormGroup
  users: UserClass[] = [];
  usersObj:UserClass=new UserClass()

  btnUpdateShow:boolean = false;

  btnSaveShow:boolean = true;

  ngOnInit(): void {
    this.service.getAllUsers().subscribe(result=>this.users=result);
    this.signupForm=this.formBuilder.group({
      fullname:[''],
      email:[''],
      password:[''],
      mobile:['']
    })
  }
  AddUser(){
    this.SaveShowBtn();
    this.http.post<any>("http://localhost:3000/users",this.signupForm.value)
    .subscribe(res=>{
      alert("User added successfully");
      this.signupForm.reset();
      this.router.navigate(['ManageUser'])

      window.location.reload()
    },err=>{
      alert("Something went wrong")
    })
  }
  DeleteUser(row :any){
    this.service.DeleteUserDetails(row.id).subscribe(res=>{
      alert("User Deleted")
      window.location.reload();
    })
  }
}

```

```
UpdateUser(row :any){

this.signupForm.controls['fullname'].setValue(row.value);
this.signupForm.controls['email'].setValue(row.value);
this.signupForm.controls['password'].setValue(row.value);

this.signupForm.controls['mobile'].setValue(row.value);
this.usersObj.id=row.id
this.UpdateShowBtn();
}
UpdateUserDetails(){
  this.usersObj.fullname=this.signupForm.value.fullname
  this.usersObj.email=this.signupForm.value.email
  this.usersObj.password=this.signupForm.value.password

  this.usersObj.mobile=this.signupForm.value.mobile
  this.service.UpdateUserDetailing(this.usersObj,this.usersObj.id).subscribe(res=>{
    alert("Data updated successfully")

    window.location.reload();
  })
}
UpdateShowBtn()
{
  this.btnUpdateShow = true;
  this.btnSaveShow = false;
}
SaveShowBtn()
{
  this.btnUpdateShow = false;
  this.btnSaveShow = true;
}
}
```