**Sourcecode:**

**User.component.ts**

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

import{ CrudHttpService } from'../crud-http.service';

import{ User } from'../user';

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

@Component({

  selector:'app-user',

  templateUrl:'./user.component.html',

  styleUrls: ['./user.component.css']

})

exportclassUserComponentimplementsOnInit {

  title = 'EventManagementApp';

  todo={id:"",firstname:"",lastname:"",email:""}

  todoList:any = [];

  constructor(privatecrudHttpService: CrudHttpService,privaterouter:Router) { }

  ngOnInit(): void {

  }

  //user:User= new User();

  isBookAdded = false;

  user = {

    id:'',

    firstname:'',

    lastname:'',

    email:'',

  };

  createTodo(){

    letdata = {

      //id: new Date().getTime(),

      firstname:this.user.firstname ,

      lastname:this.user.lastname,

      email:this.user.email,

    }

     this.crudHttpService.create(data).subscribe(result=>this.todoList=

    this.router.navigateByUrl('/admin'));

  }

}

<divclass="container">

    <tableclass="table">

      <thead>

        <tr>

          <th>ID</th>

          <th>FirstName</th>

          <th>LastName</th>

          <th>Email</th>

          <th>Actions</th>

        </tr>

      </thead>

      <tbody>

        <tr\*ngFor="let user of todoList">

          <td>{{user.id}}</td>

          <td>{{user.firstname}}</td>

          <td>{{user.lastname}}</td>

          <td>{{user.email}}</td>

          <td>

          <!--<button type="button" class="btnbtn-link" (click)="editTodo(todo)">Edit</button>

            -->

            <buttonclass="btnbtn-primary"(click)="show()">Update Info</button>

              <buttontype="button"class="btnbtn-link"(click)="deleteTodo(user.id)">Delete</button>

          </td>

        </tr>

      </tbody>

    </table>

    <!--<button type="button" class="btnbtn-success" (click)="createTodo()">Create</button>-->

  </div>

  <div\*ngIf="flag"style="background-color: yellowgreen;">

    <divclass="jumbotron">

      <divclass="container">

          <divstyle="width: 500px; margin: auto;">

              <divclass="submit-form">

                  <divclass="form-group">

                    <labelfor="id">ID: {{user.id}}</label>

                    <inputclass="form-control"id="id"   [(ngModel)]="user.id"  name="id"/>

                  </div>

                  <divclass="form-group">

                    <labelfor="name">FirstName: {{user.firstname}}</label>

                    <inputclass="form-control"id="firstname"   [(ngModel)]="user.firstname"  name="firstname"/>

                  </div>

                  <divclass="form-group">

                      <labelfor="country">LastName: {{user.lastname}}</label>

                      <inputclass="form-control"id="lastname"[(ngModel)]="user.lastname"  name="lastname"/>

                    </div>

                  <divclass="form-group">

                    <labelfor="email">Email: {{user.email}}</label>

                    <inputclass="form-control"id="email"[(ngModel)]="user.email"  name="email"/>

                  </div>

                  <button(click)="update(user)"  class="btnbtn-success">Update</button>

              </div>

            </div>

      </div>

  </div>

</div>

**Admin.component.ts**

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

import{ CrudHttpService } from'../crud-http.service';

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

@Component({

  selector:'app-admin',

  templateUrl:'./admin.component.html',

  styleUrls: ['./admin.component.css']

})

exportclassAdminComponentimplementsOnInit {

  title = 'EventManagementApp';

  todoList:any = [];

  constructor(privatecrudHttpService: CrudHttpService,privaterouter:Router){}

  ngOnInit(): void {

    this.listTodos();

  }

  listTodos(){

    this.crudHttpService.list().subscribe((response)=>{

      this.todoList = response;

    },(error=>{

    }));

  }

  user = {

    id:'',

    firstname:'',

    lastname:'',

    email:'',

  };

  isBookAdded = true;

  flag=false;

  show(){

    this.flag=!this.flag;

  }

  update(user:any){

    letdata = {

      id:this.user.id,

      firstname:this.user.firstname ,

      lastname:this.user.lastname,

      email:this.user.email,

    }

    this.crudHttpService.update(user.id,data).subscribe(

     (response)=>{

      this.listTodos();

    },(error=>{

    })

    );

  }

  deleteTodo( id: any ){

    this.crudHttpService.delete(id).subscribe((response)=>{

      this.listTodos();

    },(error=>{

    }));

  }

}

Use.component.html

<divclass="jumbotron">

    <divclass="container">

        <divstyle="width: 500px; margin: auto;">

            <divclass="submit-form">

              <div\*ngIf="!isBookAdded">

                <divclass="form-group">

                  <labelfor="name">FirstName</label>

                  <inputclass="form-control"id="firstname"required[(ngModel)]="user.firstname"name="firstname"/>

                </div>

                <divclass="form-group">

                    <labelfor="country">LastName</label>

                    <inputclass="form-control"id="lastname"required[(ngModel)]="user.lastname"name="lastname"/>

                  </div>

                <divclass="form-group">

                  <labelfor="email">Email</label>

                  <inputclass="form-control"id="email"required[(ngModel)]="user.email"name="email"/>

                </div>

                 <button(click)="createTodo()"class="btnbtn-success">Create</button>

              </div>

              <div\*ngIf="isBookAdded">

                <h3>Data added SUccessFully</h3>

              </div>

            </div>

          </div>

    </div>

</div>

**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{ HttpClientModule } from'@angular/common/http';

import{ AdminComponent } from'./admin/admin.component';

import{ UserComponent } from'./user/user.component';

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

@NgModule({

  declarations: [

    AppComponent,

    AdminComponent,

    UserComponent,

  ],

  imports: [

    BrowserModule,

    AppRoutingModule,

    HttpClientModule,

    FormsModule,

    ReactiveFormsModule

  ],

  providers: [],

  bootstrap: [AppComponent]

})

exportclassAppModule{ }

data.json

{

  "todos": [

    {

      "id": 1,

      "firstname": "Nilesh",

      "lastname": "Jain",

      "email": "nilesh@gmail.com"

    },

    {

      "firstname": "Manish",

      "lastname": "Chaudhry",

      "email": "mkch@gmail.com",

      "id": 2

    },

    {

      "firstname": "Aditya",

      "lastname": "kumar",

      "email": "adk@gmail.com",

      "id": 3

    },

    {

      "id": 6,

      "firstname": "Anish",

      "lastname": "Srivastava",

      "email": "anis@gmail.com"

    },

    {

      "id": 7,

      "firstname": "Himanshu",

      "lastname": "ranjan",

      "email": "hinam12@gmil.com"

    },

    {

      "id": 8,

      "firstname": "Vinit",

      "lastname": "Mishra",

      "email": "vm@gmail.com"

    }

  ]

}

Crud-http.service.ts

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

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

import{ catchError, Observable, throwError } from'rxjs';

@Injectable({

  providedIn:'root'

})

exportclassCrudHttpService {

  apiUrl: string = 'http://localhost:3000/todos';

  headers = newHttpHeaders().set('Content-Type', 'application/json');

  constructor(privatehttp: HttpClient) { }

  // Create

  create(data: any): Observable<any> {

    letAPI\_URL = `${this.apiUrl}`;

    returnthis.http.post(API\_URL, data)

      .pipe(

        catchError(this.handleError)

      )

  }

  // Read

  list() {

    returnthis.http.get(`${this.apiUrl}`);

  }

  // Update

  update(id: any, data: any): Observable<any> {

    letAPI\_URL = `${this.apiUrl}/${id}`;

    returnthis.http.put(API\_URL, data, { headers:this.headers }).pipe(

      catchError(this.handleError)

    )

  }

  // Delete

  delete(id: any): Observable<any> {

    varAPI\_URL = `${this.apiUrl}/${id}`;

    returnthis.http.delete(API\_URL).pipe(

      catchError(this.handleError)

    )

  }

  // Handle API errors

  handleError(error: HttpErrorResponse) {

    if (error.errorinstanceofErrorEvent) {

      console.error('An error occurred:', error.error.message);

    } else {

      console.error(

        `Backend returned code ${error.status}, ` +

        `body was: ${error.error}`);

    }

    returnthrowError(

      'Something bad happened; please try again later.');

  };

}