**Project Report: Radio Channels Registration System**

**Introduction**

Radio Channels Registration System: Is the Online System that based on the registration or managing of Radio frequencies under the regulatory authorities or Organizations by taking the credentials and other authorizations and authentications and review them then allocate the radio frequency spectrum efficiently to ensure the different radio services operate without interference and also to provide Allocation,Licensing,Monitoring and Coordination to those that meet the technical and legal requirements where then the Organization Utilize database systems to store information about allocated frequencies, license holders .

**Methodology**

* Software development methodology was adopted.
* Technologies like Html/Css, JScript, Angular and Springboot was used.
* User requirements gathered through online registration form.

**Results**

* Successfully developed a angular pages that components and services for the radio channels registration service and Channel details added in the MySQL database by connection with Spring boot.

**Discussion**

* Challenges like the frequently failure during the running or starting of the programs like that of Angular and Springboot and severely in the starting or opening of Xamp(MySQL).
* The higher needs of internet for programs running processes.
* And also the integrating and interaction with new technologies during system formation
* Then the user feedback or interactions with the system improved usability in the system.

**Conclusion**

* The project achieved its goal of forming registration web page using Angular but also difficultly data storage and managing achieved.
* Recommendations the continuing provision of training to the system users and monitoring of system performance.

**Appendices**

* Code snippets for key functionalities.

Add-channel.component.html

* <div class="container">
* <h1> Register Channel</h1>
* <br>
* <form (ngSubmit) = "onSubmit()">
* <div class="container">
* <div class="form-group">
* <label> Channel Name</label>
* <input type="text" class ="form-control" id = "channelName"
* [(ngModel)] = "channels.channelName" name = "channelName">
* </div>
* </div>
* <div class="container">
* <div class="form-group">
* <label> Address:</label>
* <input type="text" class ="form-control" id = "address"
* [(ngModel)] = "channels.address" name = "address">
* </div>
* </div>
* <div class="container">
* <div class="form-group">
* <label> Frequency Range:</label>
* <input type="text" class ="form-control" id = "frequencyRange"
* [(ngModel)] = "channels.frequencyRange" name = "frequencyRange">
* </div>
* </div>
* <div class="container">
* <div class="form-group">
* <label> Des cription:</label>
* <input type="text" class ="form-control" id = "channelDescription"
* [(ngModel)] = "channels.channelDescription" name = "channelDescription">
* </div>
* </div>
* <p>
* <div>
* <button onclick="alert('Channel Successful Registered!')">Submit</button>
* </div>
* </form>
* </div>

add-channel.component.ts

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

import { Channels } from '../channels';

import { ChannelsService } from '../channels.service';

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

@Component({

  selector: 'app-add-channel',

  templateUrl: './add-channel.component.html',

  styleUrls: ['./add-channel.component.css']

})

export class AddChannelComponent implements OnInit {

  channels: Channels = new Channels();

  constructor(private channelService: ChannelsService,

    private router: Router) { }

  ngOnInit(): void {

  }

  saveChannel(){

    this.channelService.addChannel(this.channels).subscribe( data =>{

      console.log(data);

      this.goToChannelList();

      alert("Channel Successful Added!")

    },

      error => {

        alert("Fail to Add Channel!")

      });

  }

  goToChannelList(){

    this.router.navigate(['/channels']);

  }

  onSubmit(){

    console.log(this.channels);

    this.saveChannel();

  }

}

Channel-details.component.html

<h2> View Channels Details</h2>

<hr>

<br>

<div>

    <div>

        <label> <b> Channel Name: </b></label> {{channels.channelName}}

    </div>

    <div>

        <label> <b> Address: </b></label> {{channels.address}}

    </div>

    <div>

        <label> <b> Frequency Range: </b></label> {{channels.frequencyRange}}

    </div>

    <div>

        <label> <b> Description: </b></label> {{channels.channelDescription}}

    </div>

</div>

Channel-details.component.ts

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

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

import { Channels } from '../channels';

import { ChannelsService } from '../channels.service';

@Component({

  selector: 'app-channel-details',

  templateUrl: './channel-details.component.html',

  styleUrls: ['./channel-details.component.css']

})

export class ChannelDetailsComponent implements OnInit {

  id!: number;

  channels!: Channels;

  constructor(private route: ActivatedRoute, private channelService: ChannelsService) { }

  ngOnInit(): void {

    this.id = this.route.snapshot.params['id'];

    this.channels = new Channels();

    this.channelService.getChannelById(this.id).subscribe( data => {

      this.channels = data;

    });

  }

}

Channel-list.component.html

<h3> Channels List</h3>

<hr>

<br>

<table class = "table table-striped">

    <thead>

        <tr>

            <th> Id</th>

            <th> Channel Name </th>

            <th> Address </th>

            <th> Frequency Range</th>

            <th> Description</th>

            <th> Actions </th>

        </tr>

    </thead>

    <tbody>

        <tr \*ngFor = "let channel of channel" >

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

            <td> {{ channel.channelName }} </td>

            <td> {{ channel.address }} </td>

            <td> {{ channel.frequencyRange }} </td>

            <td> {{ channel.channelDescription }} </td>

            <td>

                <button (click) = "channelUpdate(channel.id)" class = "button"> Update</button>

                <button (click) = "deleteChannel(channel.id)" class = "button" style="margin-left: 10px"> Delete</button>

                <button (click) = "channelDetails(channel.id)" class = "button" style="margin-left: 10px"> View</button>

            </td>

        </tr>

    </tbody>

</table>

Channel-list.component.ts

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

import { ChannelsService } from '../channels.service';

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

import { Channels } from '../channels';

@Component({

  selector: 'app-channel-list',

  templateUrl: './channel-list.component.html',

  styleUrls: ['./channel-list.component.css']

})

export class ChannelListComponent implements OnInit {

  channel!: Channels[];

  constructor(private channelService: ChannelsService,

    private router: Router) { }

  ngOnInit(): void {

    this.getChannel();

  }

  private getChannel(){

    this.channelService.getChannelList().subscribe(data => {

      this.channel = data;

    });

  }

  channelDetails(id: number){

    this.router.navigate(['channeldetails', id]);

  }

  channelUpdate(id: number){

    this.router.navigate(['channelupdate', id]);

  }

  deleteChannel(id: number){

    this.channelService.deleteChannel(id).subscribe( data => {

      console.log(data);

      this.getChannel();

      alert("Channel Deleted Successful!")

    },

    error => {

      alert("Fail to Delete Channel!")

    });

  }

}

Channel-update.component.html

    <h1> Update Channel</h1>

    <br>

    <form (ngSubmit) = "onSubmit()">

        <div class="container">

                <div class="form-group">

                    <label> Channel Name</label>

                    <input type="text" class ="form-control" id = "channelName"

                        [(ngModel)] = "channels.channelName" name = "channelName">

                </div>

            </div>

        <div class="container">

                <div class="form-group">

                    <label> Address:</label>

                    <input type="text" class ="form-control" id = "address"

                        [(ngModel)] = "channels.address" name = "address">

                </div>

            </div>

        <div class="container">

                <div class="form-group">

                    <label> Frequency Range:</label>

                    <input type="text" class ="form-control" id = "frequencyRange"

                        [(ngModel)] = "channels.frequencyRange" name = "frequencyRange">

                </div>

            </div>

        <div class="container">

                <div class="form-group">

                    <label> Description:</label>

                    <input type="text" class ="form-control" id = "channelDescription"

                        [(ngModel)] = "channels.channelDescription" name = "channelDescription">

                </div>

            </div>

        <p>

        <div>

          <button onclick="alert('Channel Update Successful')"></button>

        </div>

    </form>

Channel-update.component.ts

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

import { Channels } from '../channels';

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

import { ChannelsService } from '../channels.service';

@Component({

  selector: 'app-channel-update',

  templateUrl: './channel-update.component.html',

  styleUrls: ['./channel-update.component.css']

})

export class ChannelUpdateComponent implements OnInit{

  id!: number;

  channels: Channels = new Channels();

  constructor(private channelService: ChannelsService,

    private route: ActivatedRoute,

    private router: Router) { }

  ngOnInit(): void {

    this.id = this.route.snapshot.params['id'];

    this.channelService.getChannelById(this.id).subscribe(data => {

      this.channels = data;

    },

      error => {

      });

  }

  onSubmit(){

    this.channelService.updateChannel(this.id, this.channels).subscribe( data =>{

      this.goToChannelList();

    }

    , error => console.log(error));

  }

  goToChannelList(){

    this.router.navigate(['/channels']);

  }

}

App-routing.module.ts

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

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

import { AddChannelComponent } from './add-channel/add-channel.component';

import { ChannelListComponent } from './channel-list/channel-list.component';

import { ChannelUpdateComponent } from './channel-update/channel-update.component';

const routes: Routes = [

  {

    path: '',

    component: AddChannelComponent

  },

  {

    path: 'channel-list',

    component: ChannelListComponent

  },

  {

    path: 'channel-update',

    component: ChannelUpdateComponent

  },

  {

    path: 'channel-details',

    component: ChannelUpdateComponent

  }

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule]

})

export class AppRoutingModule { }

app-component.html

<router-outlet></router-outlet>

app.module.ts

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

@Component({

  selector: 'app-root',

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

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  title = 'channels\_angular';

}

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

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

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

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

import { AppRoutingModule } from './app-routing.module';

import { AppComponent } from './app.component';

import { BrowserAnimationsModule } from '@angular/platform-browser/animations';

import { AddChannelComponent } from './add-channel/add-channel.component';

import { ChannelListComponent } from './channel-list/channel-list.component';

import { ChannelDetailsComponent } from './channel-details/channel-details.component';

import { ChannelUpdateComponent } from './channel-update/channel-update.component';

@NgModule({

  declarations: [

    AppComponent,

    AddChannelComponent,

    ChannelListComponent,

    ChannelDetailsComponent,

    ChannelUpdateComponent

  ],

  imports: [

    BrowserModule,

    AppRoutingModule,

    BrowserAnimationsModule,

    FormsModule,

    HttpClientModule

  ],

  providers: [],

  bootstrap: [AppComponent]

})

export class AppModule { }

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

import { Observable } from 'rxjs';

import { Channels } from './channels';

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

@Injectable({

  providedIn: 'root'

})

export class ChannelsService implements OnInit {

  private baseURL = "http://localhost:8080/api/v1/channels";

  constructor(private httpClient: HttpClient) { }

  ngOnInit(): void { }

  getChannelList(): Observable<Channels[]>{

    return this.httpClient.get<Channels[]>(`${this.baseURL}`);

  }

  addChannel(Channels: Channels): Observable<Object>{

    return this.httpClient.post(`${this.baseURL}`, Channels);

  }

  getChannelById(id: number): Observable<Channels>{

    return this.httpClient.get<Channels>(`${this.baseURL}/${id}`);

  }

  updateChannel(id: number, channels: Channels): Observable<Object>{

    return this.httpClient.put(`${this.baseURL}/${id}`, channels);

  }

  deleteChannel(id: number): Observable<Object>{

    return this.httpClient.delete(`${this.baseURL}/${id}`);

  }

}

app.component.ts

export class Channels {

    id!: number;

    channelName!: String

    address!: String;

    frequencyRange!: String;

    channelDescription!: String;

}

**References**

* Online documentation and videos through <https://www.youtube.com/watch?v=sv76HcgjLZM>.
* W3School notes.