Lab 4 Source Code

Advanced Software Engineering

**Team 1\_2** – Srichakradhar Reddy Nagireddy (32) & Sathwik Banginimatam (5)

# Ionic App

## Login and SignUp:

onLogin(form: NgForm) {

    this.submitted = true;

    console.log(form.form.value);

    localStorage.setItem('user', JSON.stringify(form.form.value));

    if (form.valid) {

      this.userData.login(this.login.username);

      this.router.navigateByUrl('/app/tabs/movies');

    }

  }

onSignup(form: NgForm) {

    this.submitted = true;

    if (form.valid) {

      this.userData.signup(this.signup.username);

      localStorage.setItem('user', JSON.stringify(form.form.value))

      this.router.navigateByUrl('/app/tabs/movies');

    }

  }

We have stored the user credentials in localStorage and Authenticate and used route guard to validate login credentials before navigating to the API pages.

## OMDB API

this.movieService.getDetails(id).subscribe(result => {

    this.information = result;

});

The movie Services calls OMDB API to fetch the movie list and details.

export class MovieService {

  url = 'http://www.omdbapi.com/';

  apiKey = 'b0aeeae0'; // <-- Enter your own key here!

  /\*\*

   \* Constructor of the Service with Dependency Injection

   \* @param http The standard Angular HttpClient to make requests

   \*/

  constructor(private http: HttpClient) { }

  /\*\*

  \* Get data from the OmdbApi

  \* map the result to return only the results that we need

  \*

  \* @param {string} title Search Term

  \* @param {SearchType} type movie, series, episode or empty

  \* @returns Observable with the search results

  \*/

  searchData(title: string, type: SearchType): Observable<any> {

    return this.http.get(`${this.url}?s=${encodeURI(title)}&type=${type}&apikey=${this.apiKey}`).pipe(

      map(results => results['Search'])

    );

  }

  /\*\*

  \* Get the detailed information for an ID using the "i" parameter

  \*

  \* @param {string} id imdbID to retrieve information

  \* @returns Observable with detailed information

  \*/

  getDetails(id) {

    return this.http.get(`${this.url}?i=${id}&plot=full&apikey=${this.apiKey}`);

  }

}

## Image Captioning API

async uploadImageData(formData: FormData) {

    const loading = await this.loadingController.create({

      message: 'Analyzing...',

    });

    await loading.present();

    var headers = new HttpHeaders();

    headers.append("Accept", 'application/json');

    headers.append('Content-Type', 'application/x-www-form-urlencoded');

    headers.append('Access-Control-Allow-Origin', '\*');

    headers.append('Access-Control-Allow-Methods', 'POST, GET, OPTIONS, PUT');

    headers.append('Apikey', this.apiKey);

    const uploadOpts: FileUploadOptions = {

      fileKey: 'file',

      fileName: this.path,

      headers: { 'Content-Type': 'application/multipart-formdata', 'Apikey': '6d626dcd-d35e-4d72-b459-48c70851413a' }

    };

    const fileTransfer: FileTransferObject = this.transfer.create();

    fileTransfer.upload(this.imageData, this.url, uploadOpts)

      .then((data) => {

        loading.dismiss();

        console.log(data.response);

        this.textToSpeech('Looks like ' + JSON.parse(data.response)['BestOutcome']['Description']);

        this.presentToast('Looks like ' + JSON.parse(data.response)['BestOutcome']['Description']);

      }, (err) => {

        console.log(err);

      });

}

After fetching the image from camera or gallery, it is cropped and stored locally.

Then it is sent to the Cloudmersive Image recognition and description API to fetch the results

Text to Speech module is used to speak the results.