

# DOCUMENTAȚIA PROIECTULUI

Proiect WeatherApp

STUDENȚI

Iosif Mihai

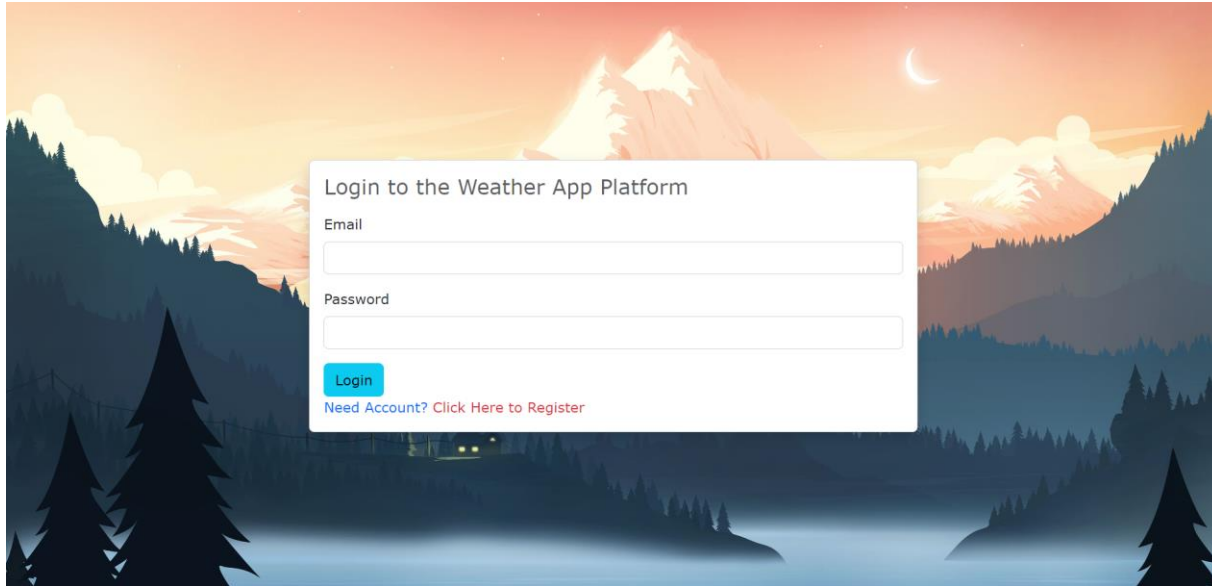
Modorcea

Roberto

# Cuprins

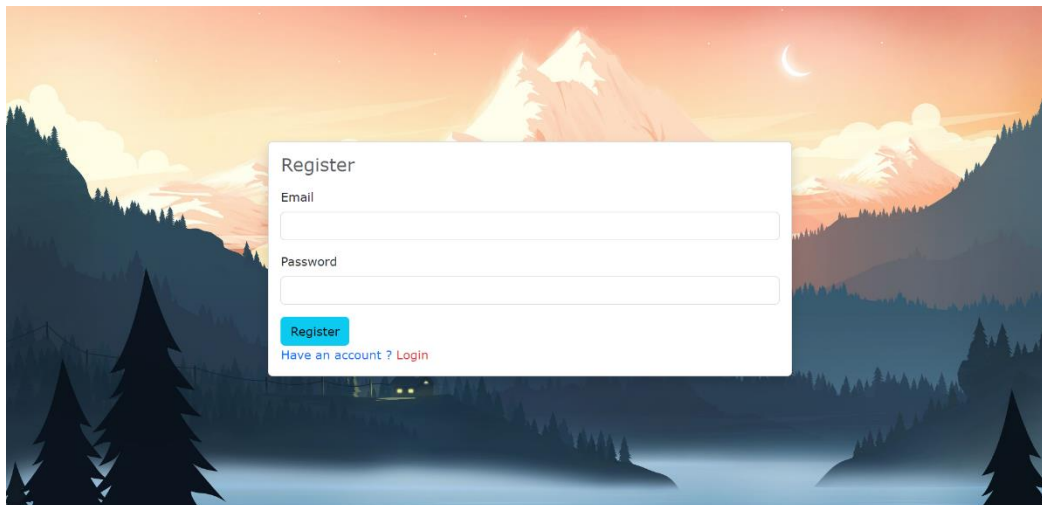
Prezentarea proiectului .....	3
Tehnologiile folosite si partea de “backend” .....	4
Arhitectura.....	6
Cum poate fi utilizata aplicatia.....	7
Concluzii.....	8

## Prezentarea Proiectului



Proiectul Weather consta in dezvoltarea unei pagini web pentru a cauta vremea in diferite orase din lume.

Proiectul ofera posibilitatea utilizatorului de a-si crea un cont sau de a se autentifica, in cazul in care acesta nu are un cont creat.



Indiferent de preferinte, utilizatorul, de indata ce s-a conectat, poate vizualiza detaliile despre vremea din diferite orase. Odata cu cautarea orasului, utilizatorului ii apare detaliile din orasul respectiv ( temperatura instant, minima,maxima,umiditate,vant)

## **Tehnologiile folosite si partea de Back-end**

- Firebase

În proiectul nostru ne-am folosit de Firebase pentru partea de autentificare. Firebase oferă un sistem de autentificare ce necesită e-mailul și parola.

Pentru partea de register, utilizatorul trebuie să își introducă e-mail-ul și parola. Astfel datele sale sunt introduse în baza de date și reținute pentru ca utilizatorul să se poată înregistra mai târziu, astfel având acces la aplicație.

Pentru partea de login, utilizatorului i se cere să introducă e-mail-ul cu care s-a înregistrat și parola pe care a setat-o. După ce sunt verificate datele și se confirmă că acestea sunt în conformitate cu datele de introduse pentru crearea contului, înregistrarea este făcută cu succes, iar în acest moment utilizatorul are acces la dashboard.

```
export const environment = {  
  production: false,  
  firebase : {  
    apiKey: "AIzaSyAye5CMhP-odFehDZ51JIUNu8Yvh7ZfPck",  
    authDomain: "weather-login-40184.firebaseio.com",  
    projectId: "weather-login-40184",  
    storageBucket: "weather-login-40184.appspot.com",  
    messagingSenderId: "69170885310",  
    appId: "1:69170885310:web:8c4720d757de0e663be4ef"  
  },  
}
```

## **Tehnologiile folosite si partea de Back-end**

- RaidAPI

Pentru proiectul nostru am folosit de asemenea platforma RapidAPI. Aici am cautat un API pentru a gasi detalii despre vreme din diferite orase

```
weatherApiBaseUrl: 'https://community-open-weather-map.p.rapidapi.com/weather',  
XRapidAPIHostHeaderName: 'X-RapidAPI-Host',  
XRapidAPIHostHeaderValue: 'community-open-weather-map.p.rapidapi.com',  
XRapidAPIKeyHeaderName: 'X-RapidAPI-Key',  
XRapidAPIKeyHeaderValue: '2484201b4amshef719a419645b79p18c26ejsn799807e9c2d3'
```

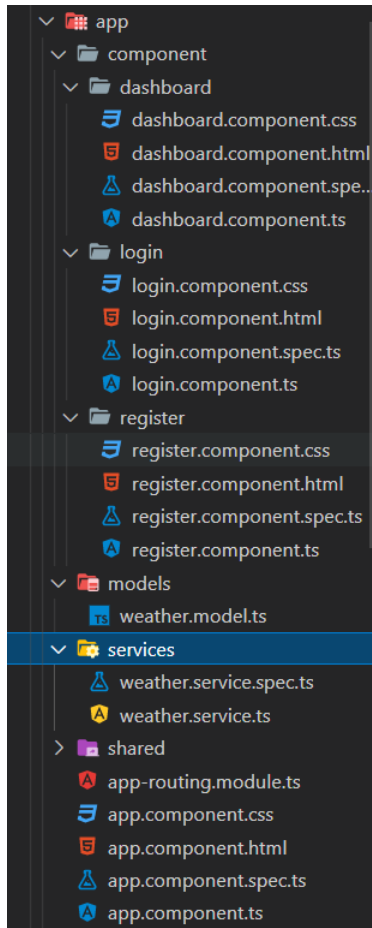
Am realizat conexiunea cu API-ul gasit, iar pe baza acestui API am facut request-uri pentru fiecare oras in parte, rezultand sa avem acces la detaliile necesare.

Request-urile realizate ne-au fost utile pentru a afisa detaliile despre vremea din orasul respectiv. Am facut cate un request in parte pentru accesarea tuturor particularitatilor pe care fiecare oras il are: numele orasului, temperatura instant, temperatura minima, temperatura maxima, vantul, umiditatea.

## Arhitectura

- Componente si servicii

Este impartita in trei componente : Dashboard, Login, Register:



- componenta 'dashboard.component' reprezinta pagina aplicatiei pentru vreme
- componenta 'login.component' care reprezinta fereastra de login, unde utilizatorul se logeaza
- componenta 'register.component' care reprezinta fereastra de register, unde utilizatorul se poate inregistra
- 

De asemenea avem si serviciul Weather care ne ajuta la afisarea informatiilor din API.

```
export class WeatherService {  
  
  constructor(private http: HttpClient) { }  
  
  getWeatherData(cityName: string): Observable<WeatherData> {  
    return this.http.get<WeatherData>(environment.weatherApiBaseUrl, {  
      headers: new HttpHeaders()  
        .set(environment.XRapidAPIHostHeaderName, environment.XRapidAPIHostHeaderValue)  
        .set(environment.XRapidAPIKeyHeaderName, environment.XRapidAPIKeyHeaderValue),  
      params: new HttpParams()  
        .set('q', cityName)  
        .set('units', 'metric')  
        .set('mode', 'json')  
    })  
  }  
}
```

## **Cum poate fi utilizata aplicatia ?**

Aplicatia porneste cu pagina de login, unde utilizatorul se conecteaza cu un cont deja existent, si optiunea de „register” unde utilizatorul isi creeaza un cont.

In momentul deschiderii aplicatiei se va deschide un formular necesar autentificarii.

In acest formular se gasesc campurile care trebuie obligatoriu completate:

- 'Email address', camp destinat introducerii email-ului,
- 'Password', unde utilizatorul isi va alege o parola,

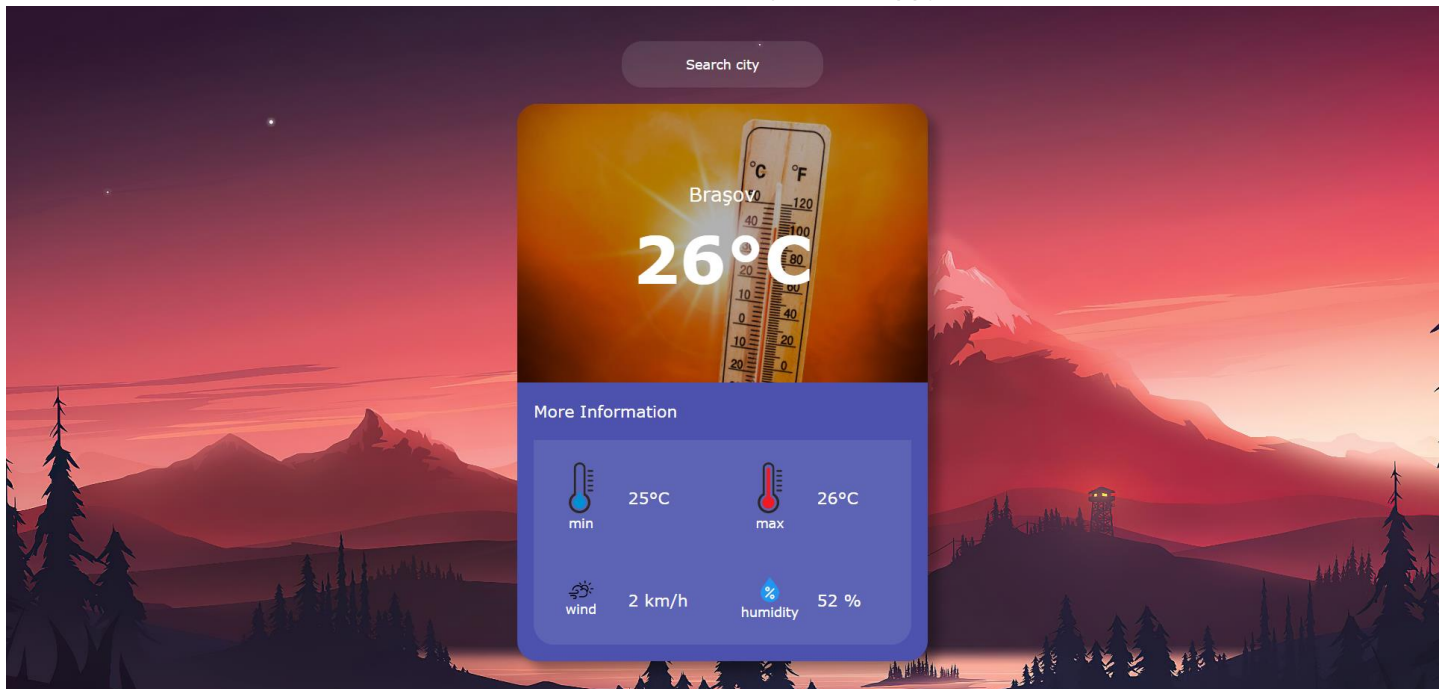
In momentul apasarii butonului de 'Register', va aparea formularul pentru a crea contul nou

Formularul contine doua campuri:

- 'Email address', acolo unde utilizatorul trebuie sa introduca email-ul inregistrat
- 'Password', unde trebuie introdusa parola asociata email-ului de mai sus.

Toate aceste date vor fi retinute in baza de date (FireBase) pentru a fi mai apoi folosite pentru conectarea utilizatorului.

Dupa ce utilizatorul si-a introdus corect datele contului, autentificarea este realizata cu succes, iar acum utilizatorul are acces la dashboard(weatherapp).



## **Concluzii**

Datorita acestui proiect am avut ocazia sa lucram cu FireBase. De asemenea am descoperit un mod foarte util de a conecta un API prin platforma RapidAPI, aceasta oferind o multitudine de API-uri potrivite pentru orice tip de aplicatie si limbaj de programare.

In concluzie cu ajutorul acestui proiect ne am dezvoltat cunostintele privind Angular si API-uri.