Sesiunea: mai 2024

Liceul Teoretic "Lucian Blaga" Cluj-Napoca

Clasa: 12A

Youbify

Lucrare pentru atestarea compentențelor profesionale la
INFORMATICĂ

Profesor îndrumător,

Morari Brîndușa

Realizator,

Crișan Alexandru-Dan

CUPRINS

1. Misiunea

Aplicația web intitulată “Youbify” își propune să rezolve una dintre problemele cele mai întâlnite în rândul utlizatorilor serviciilor de streaming al muzicii. Atunci când, inevitabil, utilizatorul migrează spre o platformă nouă, toate playlisturile sale ce conțin piesele preferate rămân “blocate” în cadrul serviciului de streaming folosit anterior.

1.1. Cum îi ajută YOUBIFY pe utilizatorii săi?

Website-ul automatizeaza procesul de transfer al playlisturilor de pe Spotify pe Youtube Music, muncă extrem de anevoioasă pentru pasionații de muzică care sunt nevoiți să parcurgă manual toate piesele lor favorite.

2. Tehnologii și aspecte teoretice

Aplicația web este programată folosind în principal Python. Aceasta este împărțită în 2 părți componente, *frontend* și *backend*.

2.1. Frontend

Partea de frontend a unei aplicații este responsabilă cu a interacționa direct cu utilizatorul și a-i asigura o experiență plăcută acestuia. Pentru structura website-ului s-a folosit HTML, pentru design CSS (cu framework-ul „Tailwind”), iar pentru funcționalitate JavaScript.

2.2. Backend

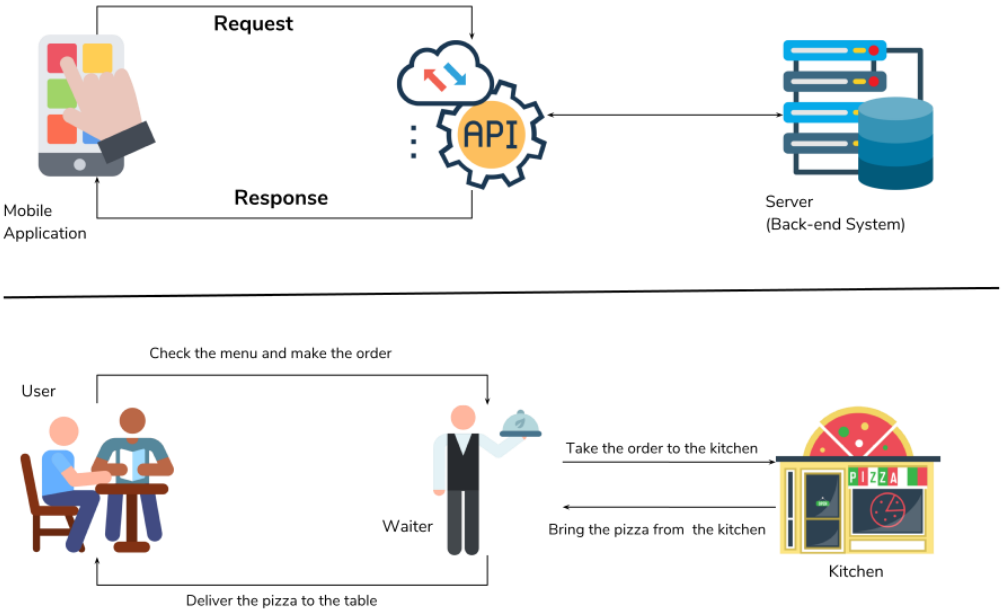
Backendul unei aplicații este responsabil cu procesarea/stocarea datelor primite de la frontend și cu asigurarea unui canal de comunicare între cele două părți.

2.3. Framework

Un framework reprezintă o un ansamblu standardizat de concepte și software, ce facilitează procesul de dezvoltare și integrare a diferitelor componente ale unui proiect în domeniul IT. Framework-urile folosite în cadrul “Youbify” sunt Tailwind (pentru CSS) și Django (pentru Python).

2.4. API

„API” vine de la „Application Programming Interface” și reprezintă un mod prin care 2 sau mai multe programe/componente ale calculatorului pot comunica una cu cealaltă. API-urile facilitează transferul de date și reprezintă fundația tuturor programelor și a tehnologiei de care o persoană beneficiază în viața de zi cu zi. În cadrul proiectului, API-ul a fost dezvoltat cu ajutorur framework-ului Django pentru limbajul de programare Python.

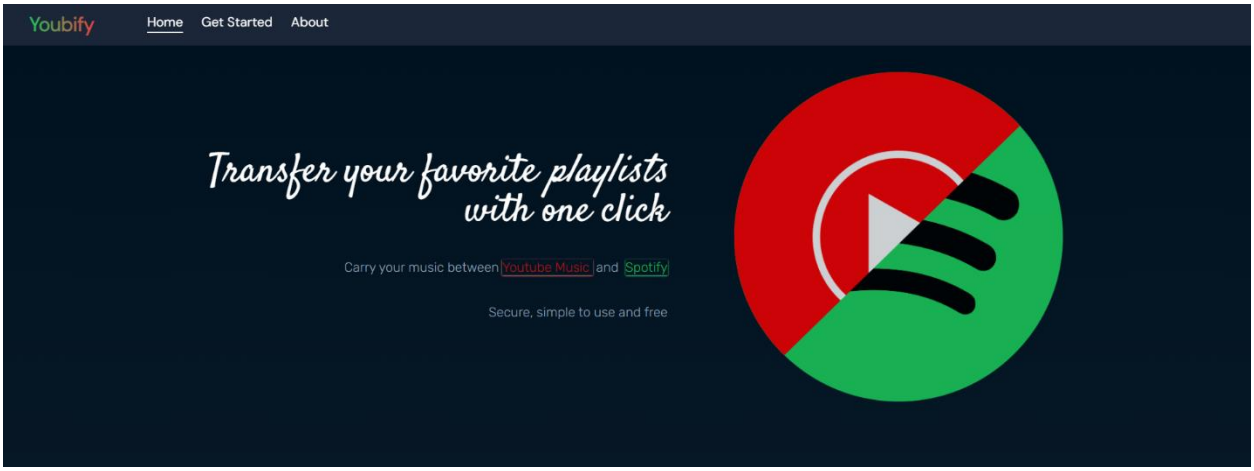


Imaginea de mai sus reprezintă o analogie între un API și modul în care funcționeaza un restaurant.

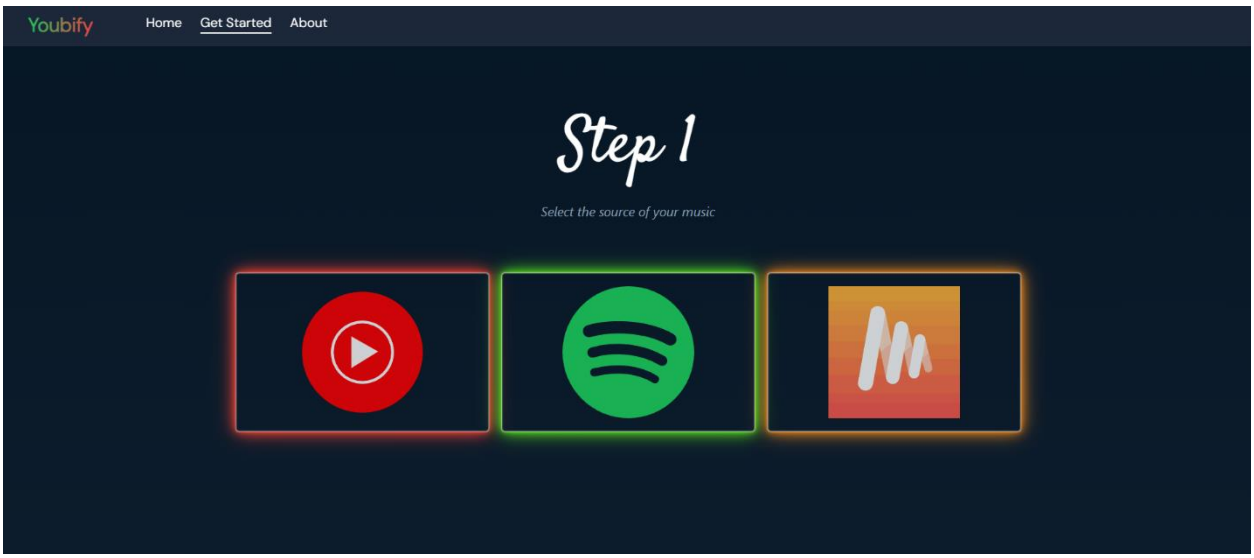
3. Descrierea aplicației web

Proiectul este alcătuit dintr-o singură pagină, acest tip de aplicații purtând numele de SPA-uri (Single Page Application). În bara de navigare se pot remarca cele 3 secțiuni ale website-ului: “Home”, “Get Started” și “About”.

- Home: Secțiune ce prezintă logoul Youbify și o scurtă descriere a proiectului



- Get Started: Secțiunea ce conține funcționalitatea principală a proiectului, locul în care utilizatorul își va transfera playlisturile.



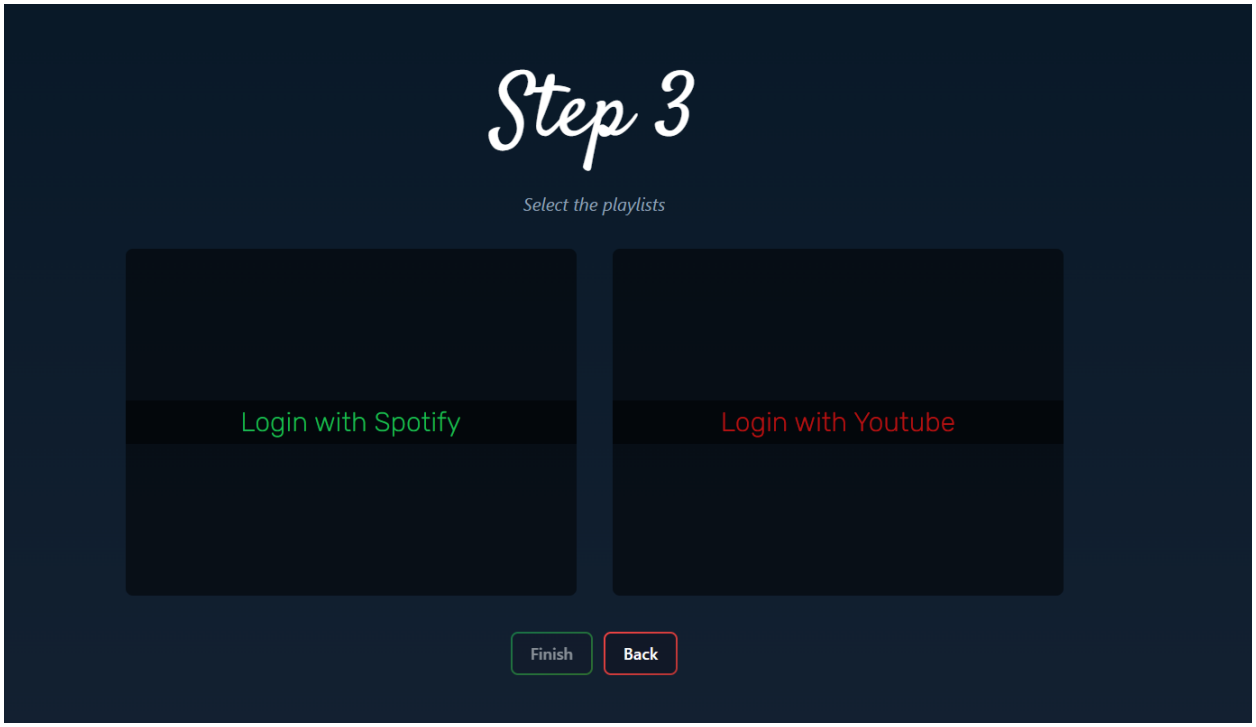
- About: Utilizatorii pot afla mai multe informații despre aplicația web

3.1. Software folosit în dezvoltarea aplicației

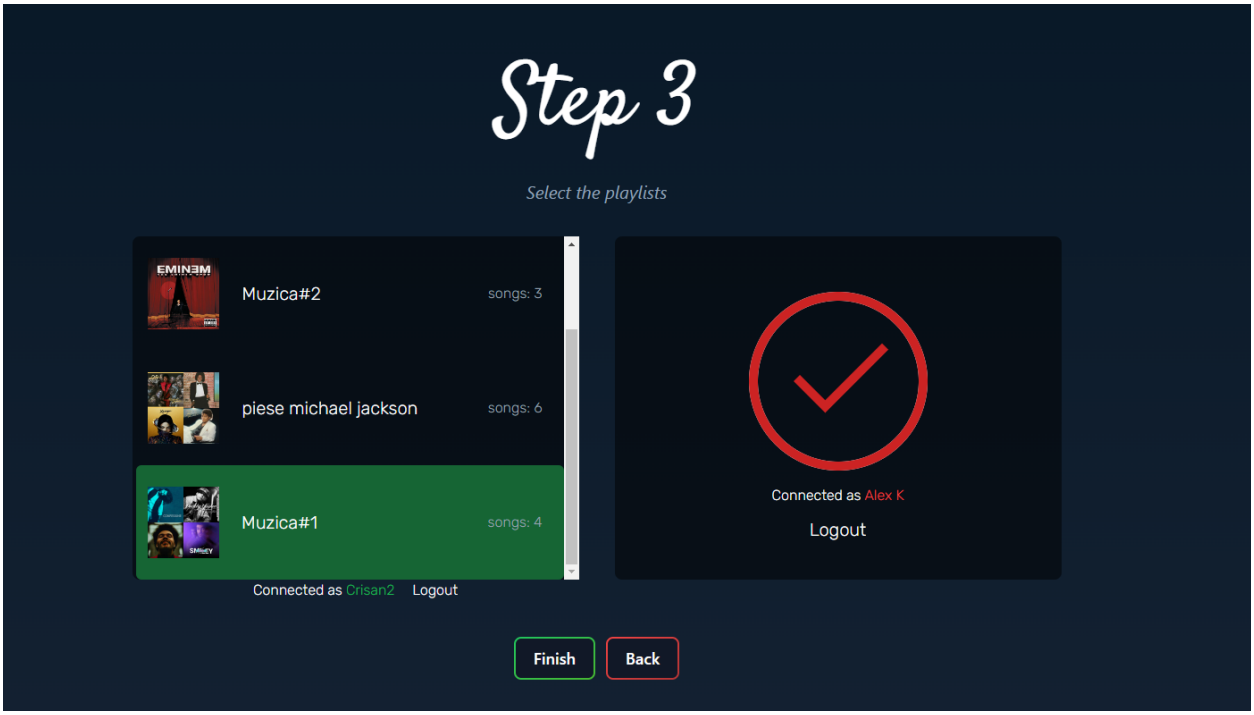
Pentru scrierea codului a fost folosit editorul de cod “Visual Studio Code” împreună cu o mulțime de extensii puse la dispoziție de acesta. Anumite elemente grafice au fost realizate în Photoshop, iar API-ul s-a testat cu ajutorul platformei Postman.

3.2. Cum folosești aplicația?

Procesul de transfer al playlisturilor începe din cadrul secțiunii “Get Started”. Utlizatorul trebuie să selecteze serviciile de streaming între care dorește să realizeze transferul, iar apoi să se conecteze cu propriile sale conturi.



Odată conectat, utilizatorul își va putea vizualiza propriile sale playlisturi și le va putea selecta pe cele pe care dorește să le transfere.



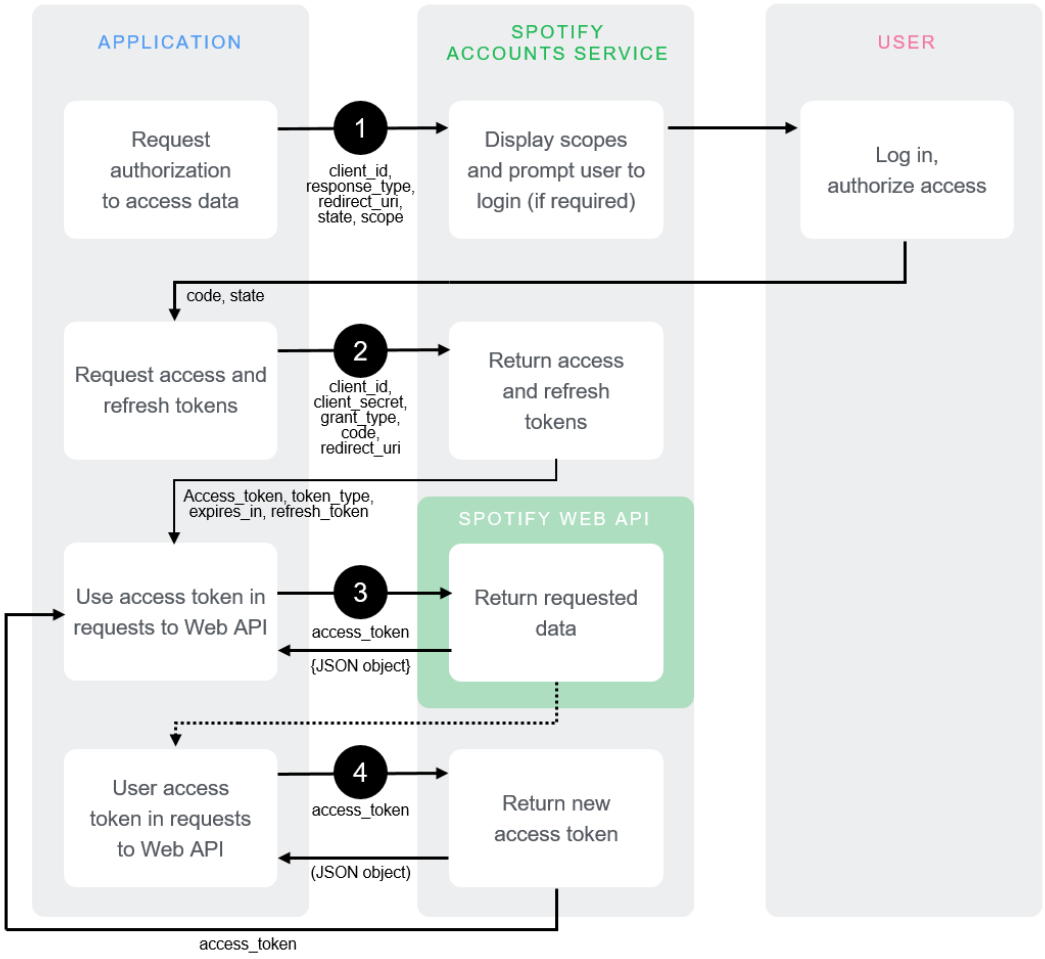
4. Aspecte tehnice

Acest capitol este dedicat anumitor aspecte de natură tehnică din aplicația web și vor fir prezentate fragmente de cod.

4.1. Autentificare

Serviciile menționate mai sus au un proces de identificare și autentificare bazat pe „token”-uri. Un token provine, de regulă, din elemente distince ale unui cont (email, ID, username etc) care sunt criptate, rezultând o serie unică de caractere (cifre, litere, caractere speciale). Acest lucru permite programatorilor sa acceseze anumite informații “protejate” într-un mediu sigur și controlat.

Atât Spotify, cât și Youtube, au un proces meticulos de obținere a tokenurilor, pe care dezvoltatorii de aplicații le folosesc în permanență pentru a comunica cu aceste servicii.



```python
def spotify_callback(request):
    code = request.GET.get('code')
    error = request.GET.get('error')

    response = post('https://accounts.spotify.com/api/token', data={
        'grant_type': 'authorization_code',
        'code': code,
        'redirect_uri': os.getenv("SPOTIFY_REDIRECT_URI"),
        'client_id': os.getenv("CLIENT_ID"),
        'client_secret': os.getenv("CLIENT_SECRET")
    }).json()

    access_token = response.get('access_token')
    token_type = response.get('token_type')
    refresh_token = response.get('refresh_token')
    expires_in = response.get('expires_in')
    error = response.get('error')

    acc_name = get_spotify_account_name(access_token)
    acc_playlists = spotify_get_playlists(access_token)
```

În interiorul acestei funcții are loc obținerea tokenului de la Spotify, în urma unui request către *https://accounts.spotify.com/api/token*. Apoi, acel token este folosit pentru a face rost de numele contului căruia îi aparține, împreună cu playlisturile acestuia.

4.2. Stocarea tokenurilor

Pentru ca utilizatorul să nu fie nevoit sa își introduca datele personale de fiecare dată când dorește să folosească platforma, informațiile obținute în urma ultimei sale conectări sunt salvate într-un “session”. Acest lucru permite salvarea atât informațiilor de autentificare cât și cele ce vizează funcționalitatea aplicației (numele contului, detalii playlisturi etc).

```
request.session["spotify_creds"] = {
      "access_token": access_token,
      "refresh_token": refresh_token,
      "expires_in": expires_in,
      "token_type": token_type,
      "account_name": acc_name,
}
request.session["playlists"] = {
      "spotify": acc_playlists
}
```

4.3. Afișarea playlisturilor

În cadrul template-ului HTML, sunt accesate variabilele create în session (în exemplul anterior) și se creează iterativ câte un "panou" pentru fiecare playlist.

```
<div class="..." >
    {%for playlist in request.session.playlists.spotify%}
    <label for="{{playlist.name}}" id="for {{playlist.name}}"
     class="...">
        <input name="item_checkbox" type="checkbox"
           value="{{playlist.id}}" id="{{playlist.name}}"
           class="..."
        />

        <div class="...">
           <img class="..." src="{{playlist.images.0.url}}"/>
           <label class="..." for="{{playlist.name}}">
                {{playlist.name}}
           </label>
           <label class="..." for="{{playlist.name}}">
                songs: {{playlist.tracks.total}}
           </label>
        </div>
    </label>
    {%endfor%}
</div>
```

4.4. Transferul playlisturilor

Atunci când utilizatorul selectează playlisturile și apasă butonul “Finish”, frontend-ul realizează un request către backend, trimițând ID-urile playlisturilor în cauză. Backend-ul apoi procesează fiecare playlist și îl transferă.

```
def transfer(request):
    cred = Credentials(**request.session["youtube_creds"])
    data = json.loads(request.body)
    playlists_to_transfer = data.get('checked_items', [])

    for playlist in playlists_to_transfer:
        youtube_playlist_id= create_youtube_playlist(playlist["name"],
                                cred)

        tracks = get_playlist_tracks(playlist["id"],
                request.session["spotify_creds"]["access_token"])

        [ youtube_add_track_to_playlist(youtube_playlist_id,
           get_youtube_track_id(
            f'{t["track"]["name"]} {t["track"]["artists"][0]["name"]}',
            cred
           ),
          cred)
         for t in tracks]
    return redirect('base:home')
```

Totodată, se poate remarca folosirea unor funcții auxiliare, implementate într-un fișier separat. Următoarea pagină conține funcțiile apelate în programul principal.

```python
def create_youtube_playlist(playlist_name:str, creds):
    service = build(serviceName="youtube", version="v3",
                    credentials=creds)
    response = service.playlists().insert(
        part="snippet", body={"snippet": {"title":playlist_name}}
    ).execute()

    return response['id']
```

```python
def youtube_add_track_to_playlist(playlist_id: str, track_id: str,
                                  creds):
    service = build(serviceName="youtube", version="v3",
                    credentials=creds)

    retry_count = 0
    max_retries = 5
    backoff_time = 1

    # In caz ca serverele de la youtube returneaza o eroare
    # se reincearca procesul de pana la 5 ori
    while retry_count < max_retries:
        try:
            response = service.playlistItems().insert(
                part="snippet",
                body={
                    "snippet": {
                        "playlistId": playlist_id,
                        "resourceId": {"kind": "youtube#video",
                                       "videoId": track_id},
                    }
                },
            ).execute()
            return response
        except HttpError as err:
            if err.resp.status == 409:
                retry_count += 1
                time.sleep(backoff_time)
                backoff_time *= 2

    response = request.execute()
    videoid = response['items'][0]['id']['videoId']

    return videoid
```

5. Planuri de viitor

Câteva funcționalități care ar putea fi adăugate în proiect în viitor:

- Posibilitatea de a realiza transferul de playlisturi între mai multe servicii (de exemplu: Apple Music, Amazon Prime Music, Deezer)
- Opțiunea de a selecta doar anumite piese dintr-un playlist
- Secțiunea “Contact” pentru a fi raportate problemele platformei / a primi feedback de la utilizatori
- Îmbunătățire a UI-ului

6. Bibliografie

- https://console.cloud.google.com/
- https://developers.google.com/
- https://developer.spotify.com/
- https://stackoverflow.com/
- https://www.youtube.com/
- https://medium.com/
- https://www.djangoproject.com/
- https://tailwindcss.com/