

Universitatea Tehnica din Cluj-Napoca
Facultatea de Automatica si Calculatoare
Departamentul Calculatoare

Documentatie Proiect

Aplicatie Web: YoutubeToText

Autori:

- Hruban Darius-Cristian (Backend & Data Base)
- Bogdan Campean Alexandru (Frontend & Design)

Cluj-Napoca
Ianuarie 2025

1 Descriere generala a proiectului

Pentru acest proiect, ne-am propus sa realizam o aplicatie web utila studentilor si nu numai, care sa rezolve o problema simpla: extragerea textului dintr-un videoclip de pe YouTube. Proiectul este realizat in Python folosind framework-ul Django si mediul de dezvoltare PyCharm.

Initial, am plecat de la un template basic de Django, dar am dezvoltat functionalitatile specifice nevoilor noastre. Ideea principala a fost sa avem o platforma unde un utilizator sa poata intra, sa isi faca un cont pentru a-si pastra istoricul, si sa converteasca link-uri video in text.

Ce am realizat in final:

- Un sistem complet de autentificare (Log In / Sign Up) pentru ca fiecare utilizator sa aiba datele protejate.
- O pagina principala unde se introduce link-ul de YouTube. In spate, aplicatia descarca sunetul si foloseste inteligenta artificiala (biblioteca `whisper`) pentru a transcrie tot ce se vorbeste in videoclip.
- Un sistem de salvare a datelor: dupa conversie, textul apare pe ecran, iar utilizatorul poate apasa un buton sa il salveze. Acesta se duce in baza de date `db.sqlite3`.
- Un "Dashboard" personal unde utilizatorul vede toate convertirile facute de el in trecut, ordonate cronologic.

Practic, site-ul automatizeaza procesul de a lua notite dintr-un video.

2 Impartire pe task-uri

Deoarece suntem o echipa de doi studenti, am incercat sa ne impartim munca in asa fel incat sa acoperim atat partea de logica din spate (Backend), cat si partea vizuala (Frontend).

2.1 Backend si Baze de Date

S-a realizat in principal functionalitatea site-ului, adica de tot ce inseamna cod Python care face aplicatia sa "miste". Concret, task-urile mele au fost:

- **Configurarea Bazei de Date:** S-au creat modelele in `models.py`, clasa `ConvertedVideo` care retine titlul, continutul textului, data crearii si leaga fiecare conversie de utilizatorul care a facut-o (folosind `ForeignKey` catre `User`).
- **Logica de Conversie (Engine-ul):** Functia `convert_youtube` in `views.py`, integrarea bibliotecilor externe:
 - `yt_dlp` pentru a descarca doar partea audio din link-ul de YouTube.
 - `whisper` (modelul base) pentru a procesa fisierul audio `.mp3` si a-l transforma in text.
- **Managementul fisierelor:** Sistemul care scrie temporar rezultatul intr-un fisier `de_citit.txt` si apoi il citeste pentru a-l salva in baza de date cand utilizatorul confirma.

- **Logica paginilor:** Functiile din `views.py` care fac legatura intre pagini (`index`, `view_dashboard`, `save_from_file`).

2.2 Frontend si Interfata Utilizator

Realizeaza estetica site-ului si experienta utilizatorului. Task-urile au inclus:

- **Design-ul paginilor:** S-au creat si stilizat template-urile HTML pentru paginile principale: pagina de pornire, dashboard-ul si pagina de vizualizare a textului.
- **Sistemul de Login/Register:** Interfata pentru autentificare (`login.html` si `signup.html`) si s-a asigurat ca formularele trimit datele corect catre backend.
- **Afisarea datelor:** Modul in care textul convertit este afisat utilizatorului si cum apare lista de videoclipuri in Dashboard, facand site-ul sa arate ca o aplicatie moderna.

3. 3. Prezentarea Interfetei Grafice

In aceasta sectiune prezentam modul in care arata aplicatia finala, evidentiind paginiile principale create.

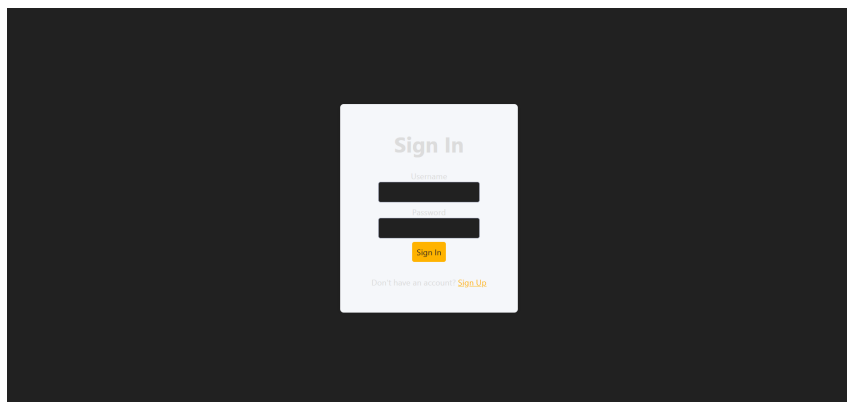


Figure 1: Pagina de Autentificare (Log In)

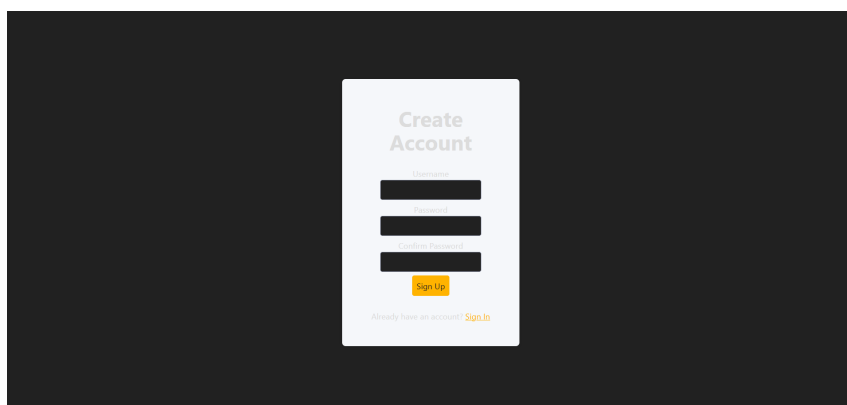


Figure 2: Pagina de Inregistrare (Sign Up)

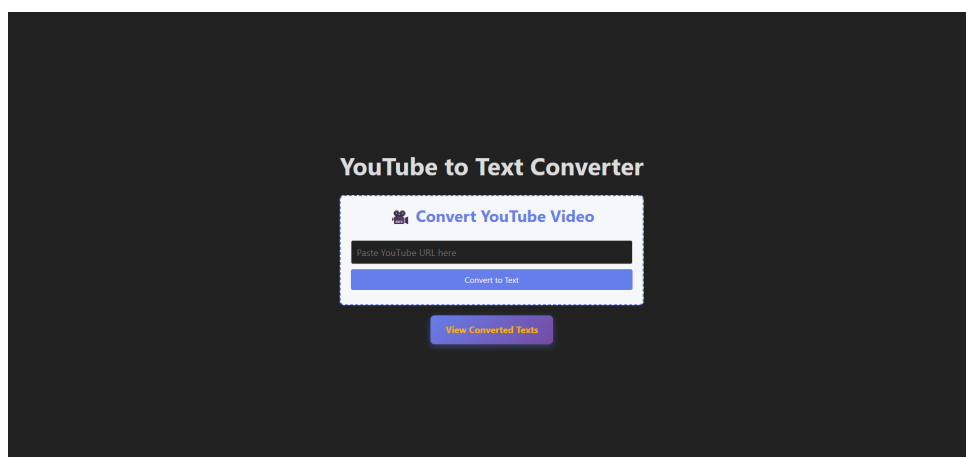


Figure 3: Pagina Principala - Conversie Link YouTube

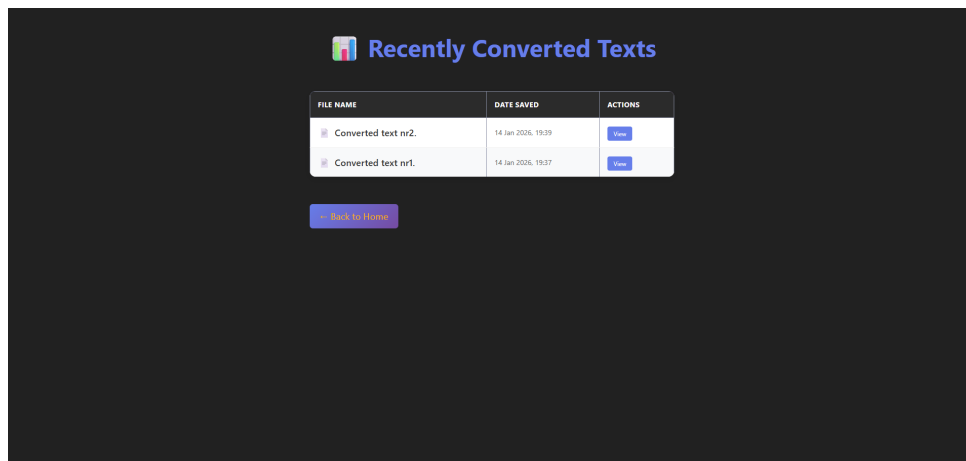


Figure 4: Dashboard - Istoricul Conversiilor Salvate

4 Anexe - Codul Sursa

Mai jos este atasat codul principal pentru partea de Backend si logica aplicatiei.

4.1 Fisierul models.py

Aici este definita structura bazei de date.

```
1 from django.contrib.auth.models import User
2 from django.db import models
3
4 class ConvertedVideo(models.Model):
5     title = models.CharField(max_length=255)
6     content = models.TextField()
7     created_at = models.DateTimeField(auto_now_add=True)
8     user = models.ForeignKey(User, on_delete=models.CASCADE, null=True,
9                               blank=True)
10
11     class Meta:
12         ordering = ['-created_at'] # Cele mai recente primele
13
14     def __str__(self):
15         return self.title
```

4.2 Fisierul views.py

Aici se afla logica de conversie (Whisper) si managementul paginilor.

```
1 import datetime
2 import os
3 from django.contrib import messages
4 from django.contrib.auth import login, authenticate
5 from django.contrib.auth.models import User
6 from django.shortcuts import render, redirect
7 from IS2025 import settings
8 from YoutubeToText.models import ConvertedVideo
9
10 # Logica pentru conversie
11 def convert_youtube(request):
12     if request.method == 'POST':
13         youtube_url = request.POST.get('youtube_url')
14         try:
15             import yt_dlp
16             import whisper
17             import os
18
19             # 1. Download audio
20             ydl_opts = {
21                 'format': 'bestaudio/best',
22                 'outtmpl': 'downloads/audio.%(ext)s',
23                 'postprocessors': [{
24                     'key': 'FFmpegExtractAudio',
25                     'preferredcodec': 'mp3',
26                 }],
27             }
28             with yt_dlp.YoutubeDL(ydl_opts) as ydl:
29                 ydl.download([youtube_url])
```

```

30
31     # 2. Transcrie cu Whisper
32     model = whisper.load_model("base")
33     result = model.transcribe("downloads/audio.mp3")
34
35     # 3. Salveaza in fisier temporar
36     file_path = os.path.join(settings.BASE_DIR, 'de_citit.txt')
37     with open(file_path, 'w', encoding='utf-8') as f:
38         f.write(result['text'])
39
40     request.session['converted_text'] = result['text']
41     os.remove('downloads/audio.mp3')
42     messages.success(request, 'Video converted successfully!')
43
44     except Exception as e:
45         messages.error(request, f'Conversion failed: {str(e)}')
46
47     return redirect('YoutubeToText:index')
48
49 # Salvarea in baza de date
50 def save_from_file(request):
51     if request.method == 'POST':
52         file_path = os.path.join(settings.BASE_DIR, 'de_citit.txt')
53         try:
54             with open(file_path, 'r', encoding='utf-8') as f:
55                 content = f.read().strip()
56
57             if content:
58                 title = request.POST.get('title', 'YouTube_Conversion')
59                 ConvertedVideo.objects.create(
60                     title=title,
61                     content=content,
62                     user=request.user if request.user.is_authenticated
63                 )
64                 # Goleste fisierul
65                 with open(file_path, 'w', encoding='utf-8') as f:
66                     f.write('')
67             except Exception as e:
68                 messages.error(request, f'Error: {str(e)}')
69             return redirect('YoutubeToText:index')
70
71 # Dashboard View
72 def view_dashboard(request):
73     if request.user.is_authenticated:
74         converted_videos = ConvertedVideo.objects.filter(user=request.
75 user)
76     else:
77         converted_videos = ConvertedVideo.objects.all()
78     return render(request, 'dashboard.html', {'converted_videos':
79 converted_videos})

```

4.3 Fisierul urls.py (Aplicatie)

Rutele definite pentru navigarea in aplicatie.

```

1 from django.urls import path
2 from YoutubeToText import views

```

```

3
4 app_name = 'YoutubeToText'
5
6 urlpatterns = [
7     path('', views.index, name='index'),
8     path('convert/', views.convert_youtube, name='convert_youtube'),
9     path('dashboard/', views.view_dashboard, name='dashboard'),
10    path('save/', views.save_from_file, name='save_from_file'),
11    path('view/<int:video_id>/', views.view_video_detail, name='
view_video_detail'),
12 ]

```

4.4 Fisierul urls.py (Aplicatia principala (IS2025))

Rutele definite pentru navigarea in aplicatie.

```

1 from django.contrib import admin
2 from django.urls import path, include
3
4 from YoutubeToText import views
5
6 urlpatterns = [
7     path('admin/', admin.site.urls),
8     path('', views.view_login, name='login'),
9     path('signup/', views.view_signup, name='signup'),
10    path('app/', include('YoutubeToText.urls')),
11 ]

```