Configurarea Mediului de Lucru  
  
Sistem de Operare

* Windows 10: Platforma principală de lucru pentru dezvoltarea și testarea aplicației.

IDE Utilizat

* Visual Studio: Utilizat pentru scrierea și gestionarea codului Python datorită suportului extins pentru limbajul Python, precum și pentru debugging-ul eficient.

1. **Instalarea Python**
   * Am descărcat și instalat Python versiunea 3.9.13 de pe site-ul oficial: [python.org](https://www.python.org/downloads/release/python-3130/).
   * Pentru verificarea instalării, executați comanda:  
      python --version
2. **Browser Utilizat**
   * Pentru rularea aplicației, am ales browser-ul Chrome. Înainte de configurare, am verificat versiunea instalată accesând secțiunea de setări a browser-ului.
3. **Descărcarea ChromeDriver**
   * După identificarea versiunii Chrome, am accesat site-ul oficial al ChromeDriver-ului ([chromium.org](https://sites.google.com/chromium.org/driver/)) pentru a descărca driver-ul corespunzător versiunii detectate.
4. **Adăugarea ChromeDriver în Variabilele de Mediu**
   * Am configurat variabilele de mediu (Path) pentru a permite accesarea ChromeDriver-ului din orice locație. Am adăugat calea completă către locația unde am salvat driver-ul: C:\Users\2021 august\Desktop\ChromeDriver>

**Environment**: Am lucrat in Visual Studio ca IDE principal pentru dezvoltare si debugging, în care am instalat toate librăriile necesare pentru proiect. Toate librăriile utilizate în proiect sunt instalate prin intermediul managerului de pachete **pip**. Comenzile utilizate pentru instalarea librăriilor sunt următoarele:

1. **selenium**: Pentru interacțiunea cu browser-ul Chrome.

pip install selenium

Posibile probleme:

* + Necesită versiunea corectă de ChromeDriver pentru browser-ul instalat.

1. **requests**: Pentru verificarea conexiunii la internet.

pip install requests

Posibile probleme:

* + Probleme de timeout la conexiuni lente.

1. **pyautogui**: Pentru capturi de ecran sau alte interacțiuni automatizate.

pip install pyautogui

Posibile probleme:

* + Necesită rezoluție grafică activă (nu funcționează pe servere fără GUI).

1. **pyaudio**: Utilizat pentru analiza sunetului.

pip install pyaudio

Posibile probleme:

* + Este necesar un driver de sunet compatibil și instalarea fișierului .whl specific platformei.

1. **numpy**: Manipulare matematică și numerică.

pip install numpy

Posibile probleme:

* + Asigurarea compatibilității cu versiunile de Python utilizate.

1. **opencv-python**: Procesarea imaginilor.

pip install opencv-python

Posibile probleme:

* + Necesită suport pentru camere web și hardware specific.

1. **tkinter**: Inclus nativ în instalarea Python pentru dezvoltarea interfețelor grafice. Nu necesită instalare separată.

Posibile probleme:

* + Pe unele sisteme Linux, poate fi necesar să instalați manual biblioteca prin managerul de pachete al sistemului de operare.

1. **mss**: Captură de ecran performantă pentru multi-monitor.

pip install mss

Posibile probleme:

* + Dependență de drivere grafice funcționale.

1. **soundcard**: Captură audio de la dispozitivele sistemului.

pip install soundcard

Posibile probleme:

* + Necesită drivere audio funcționale pe sistemul de operare.

1. **soundfile**: Pentru salvarea fișierelor audio.

pip install soundfile

Posibile probleme:

* + Necesită librăria externă libsndfile pe anumite platforme (în special Linux).

1. **logging**: Gestionarea logurilor pentru aplicație. Este inclus nativ în Python și nu necesită instalare separată.
2. **threading**: Este inclus nativ în Python și este folosit pentru multitasking.
3. **os**: Modul pentru interacțiunea cu sistemul de operare. Este nativ în Python.
4. **time**: Bibliotecă nativă Python pentru gestionarea timpului în execuție.
5. **subprocess**: Permite rularea proceselor externe din Python. Este nativ în Python.

* **Repository GitHub**: Am creat un repository numit[**Project\_SeleniumYTScript**](https://github.com/RazvanUngureanu02/Project_SeleniumYTScript) unde am incarcat progresul proiectului.

**Crearea Structurii Proiectului**  
  
 project/

|-- main.py

|-- youtube\_browser.py

|-- youtube\_gui.py

|-- recorder.py

|-- logs/

|-- application.log  
  
**Analiza situatiilor/problemelor neprevazute**

* **Lipsa conexiunii la internet**:
  + Programul verifică dacă există o conexiune la internet.
  + În caz de eroare, reîncearcă conexiunea și înregistrează problema în fișierul de log.
* **Pop-up-uri neprevăzute pe YouTube**:
  + Programul gestionează pop-up-uri precum: acorduri de cookie-uri, conectare la cont sau sugestii de abonare la YouTube Premium.
  + Pop-up-urile sunt închise automat și înregistrate în log.
* **Erori de încărcare a paginii YouTube**:
  + În cazul erorilor de încărcare sau rețea, programul reîncearcă sau oprește scriptul cu un mesaj corespunzător.
  + Aceste probleme sunt documentate în log.
* **Gestionarea reclamelor din videoclipuri**:
  + **Reclame la începutul videoclipului**:
    - Așteaptă opțiunea de a da skip și o execută sau așteaptă terminarea automata a reclamei.
    - După skip, pornește înregistrarea video și audio.
  + **Reclame în timpul videoclipului**:
    - Detectează începerea reclamei, oprește înregistrarea audio/video.
    - Așteaptă opțiunea de a da skip sau terminarea automata a reclamei.
    - Relansează înregistrarea după terminarea reclamei.
  + Toate aceste situații sunt notate în fișierul de log.
* **Probleme de performanță**:
  + În cazul în care sistemul întâmpină probleme de performanță, cum ar fi încărcarea lentă a paginii sau blocarea aplicației, programul monitorizează utilizarea resurselor și înregistrează timpul de răspuns.
  + În cazuri critice, programul trimite un mesaj de eroare utilizatorului și salvează log-ul corespunzător.
* **Timeout în timpul operațiunilor**:
  + Dacă o operațiune (cum ar fi încărcarea unui videoclip) depășește un timp limită predefinit, programul abandonează acțiunea și încearcă o alternativă sau notifică utilizatorul.
* **Probleme de compatibilitate a driver-ului Chrome**:
  + În cazul în care driver-ul nu este compatibil cu versiunea Chrome instalată, programul detectează această problemă și oferă un mesaj clar utilizatorului pentru actualizare.
  + Erorile sunt înregistrate în log.
* **Întreruperi ale procesului de înregistrare**:
  + În cazul în care procesul de înregistrare este întrerupt (ex. lipsa spațiului pe disc), programul finalizează înregistrarea parțială și notifică utilizatorul despre eroare.
  + Toate incidentele sunt salvate în log pentru diagnosticare ulterioară.
* **Gestionarea mesajelor de eroare YouTube**:
  + Mesaje precum "Video indisponibil" sau "Eroare de redare" sunt gestionate prin afișarea unui mesaj către utilizator și oprirea scriptului în mod controlat.
  + Detaliile problemei sunt salvate în log pentru a putea fi analizate.

**Documentarea Codului**

**Explicația Bucăților Importante din Cod**

Clasa YouTubeBrowser

-**check\_internet\_connection:**

A computer screen with text

Description automatically generated  
  
 **-accept\_consent:**

A white background with black text

Description automatically generated

-**handle\_ads:**

A white background with black text

Description automatically generated  
  
**Modulul recorder.py**

* **Funcționalități**:
  + Utilizează mss pentru capturarea video și soundcard pentru captură audio.
  + Include analiza nivelului de sunet și exportă datele într-un fișier text.

**Exemple din cod:**

**-Captură video**:

A close-up of a screen

Description automatically generated  
  
**-Captură audio**:

A computer screen shot of a computer code

Description automatically generated

**Modulul main.py**

A screen shot of a computer code

Description automatically generated

**Logging-ul**

Toate erorile și acțiunile sunt salvate într-un fișier .log, ceea ce facilitează diagnosticarea problemelor și urmărirea performanței aplicației.

**Formatul log-urilor**

[YYYY-MM-DD HH:MM:SS] | LEVEL: Message

**Exemple de log-uri**

* **INFO**:

[2024-11-18 10:00:00] | INFO: Conexiune la internet stabilită.

[2024-11-18 10:01:00] | INFO: Videoclipul a fost încărcat cu succes.

* **ERROR**:

[2024-11-18 10:01:30] | ERROR: Lipsă conexiune la internet.

[2024-11-18 10:02:00] | ERROR: Eșec la deschiderea paginii YouTube.

* **WARNING**:

[2024-11-18 10:03:00] | WARNING: Timp de răspuns lent al paginii.

Log-urile sunt salvate într-un fișier centralizat și pot fi trimise automat prin email pentru analiză rapidă în cazul erorilor critice.

**Direcții Viitoare**

**Extinderea Funcționalității**

1. **Algoritmi de Recunoaștere Vocală**:
   * Implementarea unui sistem de transcriere automată a audio-ului din videoclipuri, util pentru generarea de subtitrări sau căutare semantică.
2. **Descărcare Automată de Videoclipuri**:
   * Adăugarea opțiunii de descărcare a videoclipurilor în diferite formate și rezoluții, inclusiv doar audio.

**Optimizări de Performanță**

1. **Reducerea Timpului de Așteptare**:
   * Optimizarea utilizării Selenium pentru gestionarea mai rapidă a elementelor dinamice, cum ar fi pop-up-urile.
2. **Teste Unitare**:
   * Crearea unui set complet de teste unitare pentru a asigura funcționalitatea corectă și pentru a facilita debugging-ul.

**Compatibilitate Cross-platform**

1. **Suport pe Linux și macOS**:
   * Extinderea aplicației pentru a funcționa pe diferite sisteme de operare prin adaptarea modulelor de drivere și interfață.
2. **Driver Universal**:
   * Implementarea unui mecanism de detectare automată și configurare a browserului activ.

**Interfață Mai Complexă**

1. **Migrare la Framework Modern**:
   * Trecerea de la tkinter la un framework mai avansat, precum PyQt, pentru o interfață mai prietenoasă și scalabilă.
2. **Control Avansat**:
   * Posibilitatea de a selecta regiunea de captură a ecranului și de a seta calitatea fișierelor înainte de înregistrare.

**Integrare cu API-uri de Notificare**

1. **Notificări în Timp Real**:
   * Notificări push pentru a informa utilizatorii despre finalizarea proceselor sau despre erorile întâmpinate.
2. **Generare Rapoarte**:
   * Crearea de rapoarte automate în format PDF/Excel despre utilizarea aplicației și performanțele înregistrate.