

DecibelIT

Editor Audio-Video

Bianca-Ioana Braha
Diana-Marinela Buliga
Alberto-Ionuț Toșcariu
Grupa 1409B
Anul universitar 2021-2022

Cuprins:

1. Descrierea Tematicii
2. Obiectivele Proiectului
3. Descrierea Tehnică a Aplicației
4. Modul de Utilizare a Aplicației
5. Rezultatele Execuției
6. Prezentarea Dependințelor Aplicației
7. Rolurile în Echipă

1. Descrierea Tematicii

Există o multitudine de programe menite să faciliteze editarea audio-video a materialelor .mp4, .mp3 sau .wav astfel încât să satisfacă anumite preferințe din punct de vedere vizual sau anumite cerințe astfel încât materialele să poată fi folosite în diverse domenii precum film, publicitate, jocuri etc. Cele mai folosite funcții ale programelor de editare audio-video deja existente pe piață sunt tăierea, îmbinarea și adăugarea de elemente. Aceste funcții sunt incluse în programe complexe, care necesită componente de bună calitate și avansate pentru a rula fără erori sau întreruperi pe computer. Acest lucru devine o problemă atunci când utilizatorii doresc să editeze la un nivel de bază anumite materiale, fără a avea nevoie de funcționalități complexe, ce consumă memoria RAM. O altă problemă relevantă pentru acest subiect este perioada îndelungată de învățare a unui astfel de program de editare. Cu multiplele funcționalități, poate deveni greoi pentru utilizator să înțeleagă fiecare funcție accesibilă.

Astfel, pentru a veni în ajutorul persoanelor care își doresc a se folosi de funcționalitățile de bază a unui editor audio-video, însă nu beneficiază de un calculator destul de puternic pentru a rula programele existente pe piață sau de timpul necesar pentru a învăța cum să le utilizeze, ne-am propus să realizăm DecibelIT. Scopul este de a oferi publicului larg spre utilizare un program simplu și rapid de folosit în domeniul manipulării audio-video. Pentru a folosi DecibelIT, este esențial ca utilizatorii să dețină doar cunoștințe minime de manevrare a computerului.

2. Obiectivele Proiectului

În faza incipientă a proiectului, ne-am propus să realizăm un editor audio-video care să faciliteze:

- Introducerea de către utilizator fișiere cu extensii de tipul .mp3, .wav, .mp4
- Derularea elementelor audio și video
- Modificarea vitezei de redare
- Îmbinarea mai multor fișiere audio
- Decuparea fișierelor audio și video
- Vizualizarea efectelor de undă

Prin realizarea tuturor acestor obiective, utilizatorii vor avea ocazia de a edita materiale video și audio în scop personal într-un mod simplist și rapid.

Pe parcursul realizării proiectului, am întâmpinat diverse dificultăți de implementare la care nu ne așteptam și a căror rezolvare nu am reușit să o găsim în timp util. Din acest motiv, obiectivele pe care am reușit să le îndeplinim pentru proiectul DecibelIT sunt:

- Încărcarea de fișiere de tipul .mp3, .mp4 și .wav
- Derularea elementelor audio și video
- Modificarea vitezei de redare
- Decuparea fișierelor audio și video
- Vizualizarea unui playlist cu fișierele încărcate
- Salvarea fișierul editat
- Extinderea ferestrei în modul fullscreen

3. Descrierea Tehnică a Aplicației

Programul are la bază 2 fișiere Python, main.py și VideoWindowClass.py. Fișierul main.py pornește aplicația, aici regăsindu-se începutul fiecărui proiect Python, `__main__`. În funcția main se inițializează atât aplicația, cât și fereastra care va pune la dispoziție interfața aferentă.

Fișierul VideoWindowClass.py conține crearea și funcționalitatea butoanelor din fereastra aplicației. Pentru crearea acestora, s-au folosit funcții special create din librăriile PyQt5.

Tipuri de widget-uri folosite în aplicație, precum și crearea lor:

- Menu, apelându-se `menuBar()` pentru crearea meniului
- Action, apelându-se funcția `QAction()` pentru butoanele de Upload, Save, Exit
- Layout, apelându-se funcțiile `QHBoxLayout()`, `QVBoxLayout()` pentru o organizare intuitivă
- Button, apelându-se funcția `QPushButton()` pentru realizarea butoanelor de redare a melodiei, afișarea playlistului, afișarea pe întreg ecranul a fișierului rulat, tăierea acestuia, ieșire din aplicație și salvare
- ComboBox, apelându-se funcția `QComboBox()` pentru schimbarea vitezei de redare
- Slider, apelându-se funcția `QSlider()` pentru afișarea timpului de redare și pentru schimbarea minutului de redare
- Table, apelându-se funcția `QTableWidget()` pentru crearea tabelului în care sunt afișate fișierele încărcate
- Text, apelându-se funcția `QLineEdit()` pentru crearea căsuțelor în care se vor introduce timpii necesari pentru tăierea fișierului
- StatusBar, care este creat prin funcția `QStatusBar()` și este actualizat în funcție de starea programului. Va afișa detalii despre butoanele din meniu sau fișierul media care rulează în momentul curent.

Funcții create:

- `getSettingsValues` - preia dimensiunile ferestrei în care este afișată interfața
- `showPlaylist` - apelată atunci când butonul de „Show Playlist” este apăsat, afișează sau ascunde layoutul în care este playlistul, în funcție de starea sa anterioară
- `speed` - preia valoarea selectată de utilizator din lista cu viteze
- `updateSpeed` - schimbă viteza de redare a fișierului deschis, fiind apelată la selectarea uneia dintre opțiunile puse la dispoziție din lista aferentă comboBox-ului
- `playMusic` - schimbă starea media playerului, redând sau oprind melodia selectată
- `mediaStateChanged` - schimbă iconul aferent butonului de play în funcție de starea media playerului
- `positionChanged` - modifică poziția sliderului atunci când este mutat din poziția sa actuală din interfață
- `setPosition` - setează noua poziție a sliderului
- `timeDisplayMax` - setează durata fișierului redat
- `timeDisplayMin` - afișează în timp real timpul de redare
- `loadMedia` - redă în videoPlayer fișierul selectat
- `openFile` - încarcă un singur fișier și îl pune pe acesta în playlist
- `openMediaFolder` - preia din fișierul selectat doar fișierele de tip audio sau video și le adaugă în playlist
- `createTable` - creează și actualizează tabelul aferent playlistului
- `createTableRow` - crează o nouă intrare în playlist
- `clickedItem` - rulează în videoPlayer fișierul selectat din playlist

- cropPressed - creează un subclip în funcție de fișierul curent redat
- saveFile - se salvează rezultatul în urma editării
- exitCall - oprește aplicația și închide interfața

4. Modul de Utilizare a Aplicației

După deschiderea aplicației DecibelIT, fereastra care va apărea pe ecranul utilizatorului va conține:

- Butoanele:
 - File
 - Loaded
 - Fullscreen
 - Exit
 - Show Playlist
 - Crop File
 - Save
 - Pauză
- Drop-down pentru modificarea vitezei
- Spațiu pentru derularea video-ului

În momentul în care utilizatorul dă click pe butonul File, se deschide un meniu de tip drop-down cu următoarele variante:

- Load folder
- Load file
- Save
- Exit

În cazul în care utilizatorul dorește încărcarea a mai multor fișiere în același timp, este recomandată selectarea butonului Load Folder, moment în care se va deschide o nouă fereastră prin care utilizatorul poate selecta directorul dorit.

În cazul în care utilizatorul nu are colecționate toate fișierele pe care dorește să le utilizeze în cadrul procesului de editare, acesta va fi nevoit să se folosească de funcția Load File. La fel ca în cazul Load Folder, se va deschide o nouă fereastră în care se va putea naviga prin directoarele computerului, ca apoi să fie selectat fișierul dorit. Dacă se dorește încărcarea mai multor fișiere din mai multe directoare diferite, este necesară utilizarea repetată a butonului Load File.

Odată ce utilizatorul a adăugat fișierele dorite, acesta poate să le vizualizeze apăsând butonul Show Playlist. Astfel va apărea în fereastra deja deschisă o nouă secțiune în care sunt enumerate fișierele încărcate, alături de durata acestora exprimată în minute. Din playlist se poate selecta, rând pe rând, fișierul pe care dorește utilizatorul să îl modifice sau deruleze pentru a ajunge la rezultatul finit.

Pentru a modifica viteza de derulare a videoclipului sau a fișierului audio, utilizatorul are la dispoziție viteze de derulare cuprinse între x0.5 și x1.5. Viteza by default, pe care este menținut orice video sau fișier audio este cea de x1.0.

Pentru a decupa o secvență din fișierele video sau cele audio, este necesară selectarea fișierului din cadrul playlist-ului, urmată de acționarea asupra butonului Crop și selectarea secvenței ce se dorește a fi păstrată.

În urma modificărilor realizate asupra materialelor video și audio, se poate salva proiectul pentru a înregistra progresul. Acest lucru se poate realiza prin apăsarea butonului Save din partea de jos a ferestrei sau din cadrul meniului de tip drop-down afișat în urma apăsării butonului File din partea stânga-sus a ferestrei. Prin acționarea unui click asupra

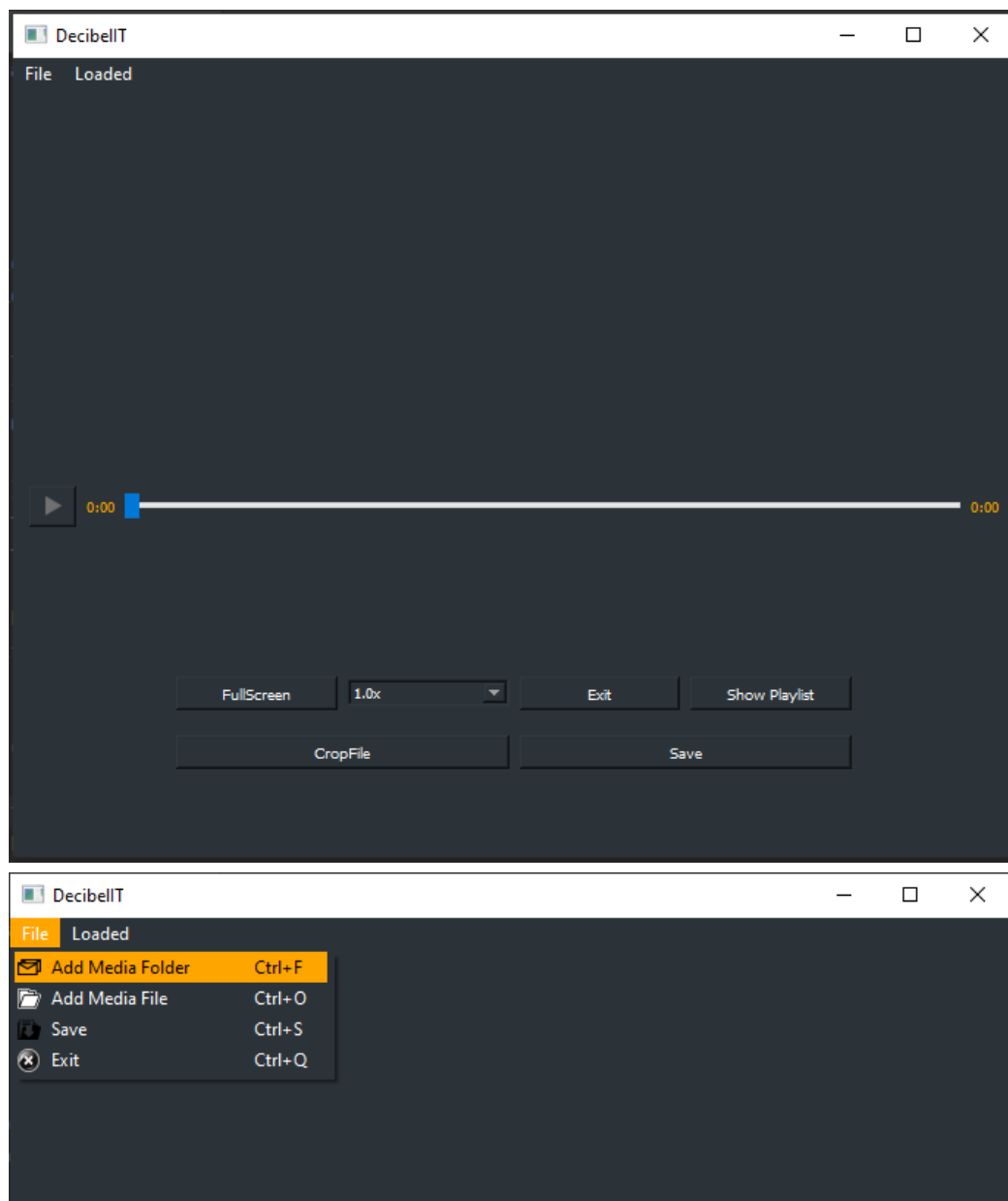
butonului Save(indiferent care dintre cele două), în partea de jos a ferestrei vor apărea două câmpuri ce trebuie completate cu informațiile necesare:

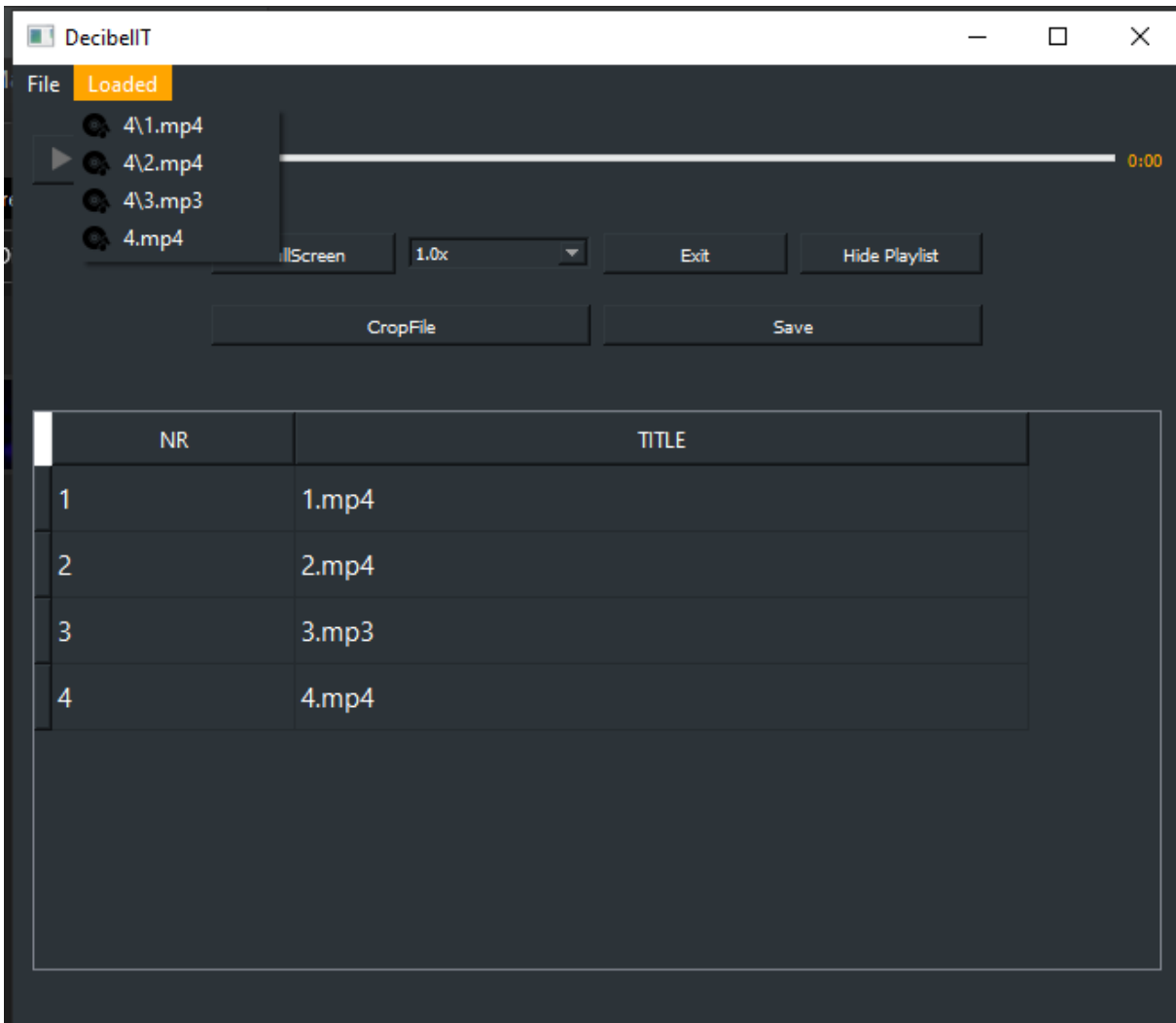
- Numele sub care se dorește salvarea fișierului editat
- Directorul în care se dorește salvarea fișierului editat, prin indicarea path-ului

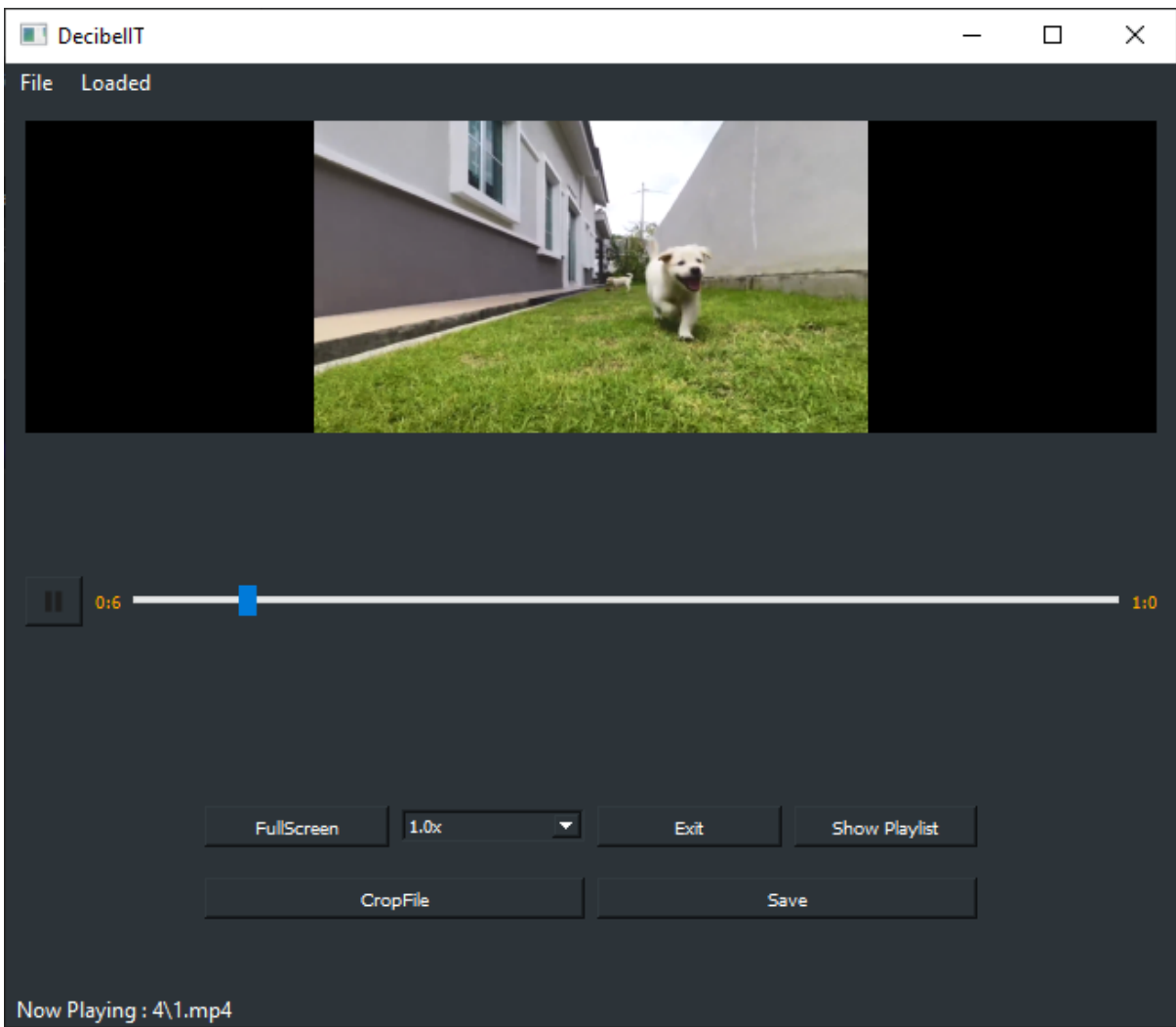
După realizarea tuturor acestor etape, utilizatorul va putea considera procesul creativ de editare audio-video încheiat cu succes și se va putea bucura de rezultatele obținute într-un timp mult mai redus și mai puțin costisitor din punct de vedere al resurselor consumate față de programele de editoarele existente deja pe piață.

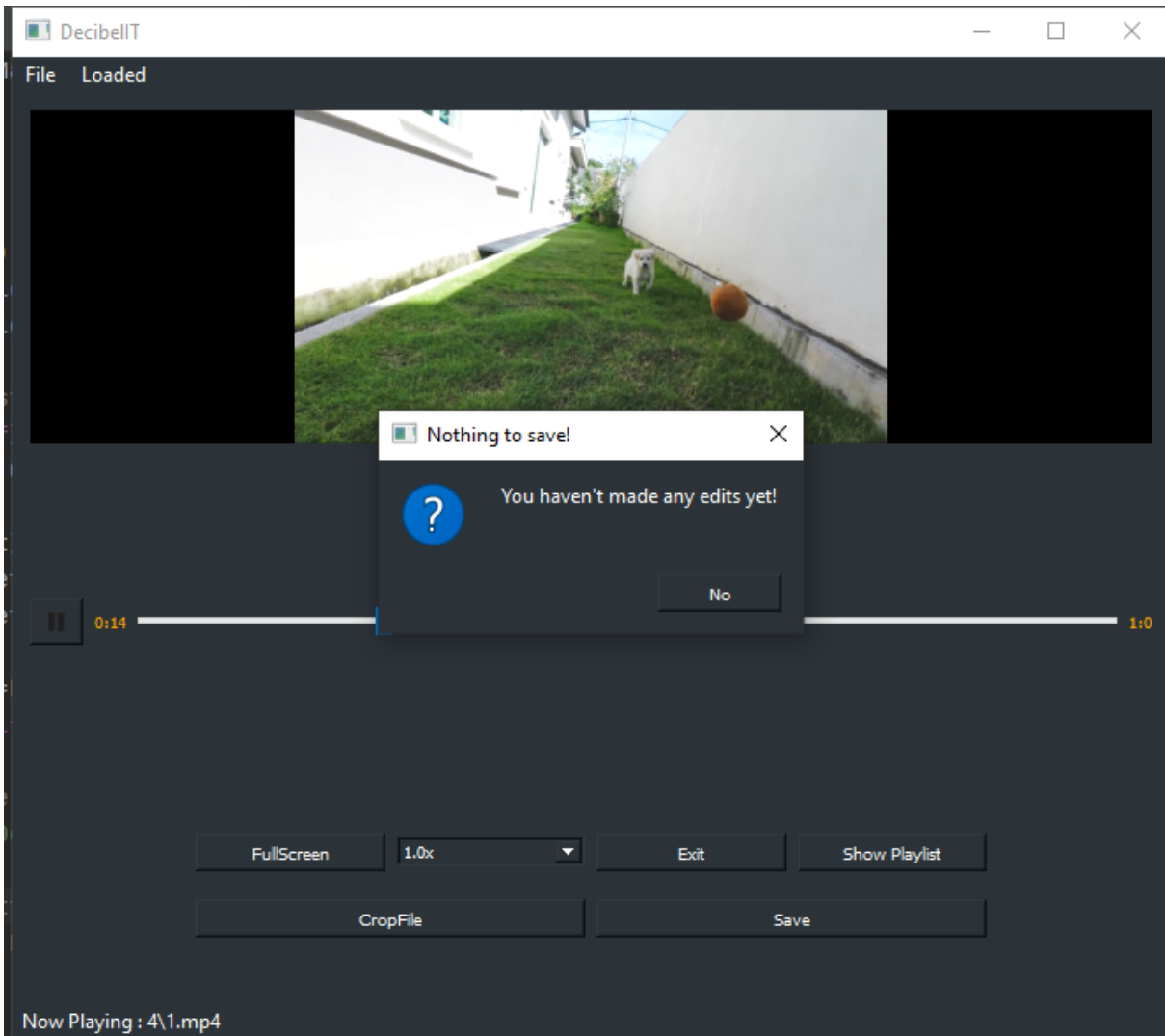
5. Rezultatele Execuției

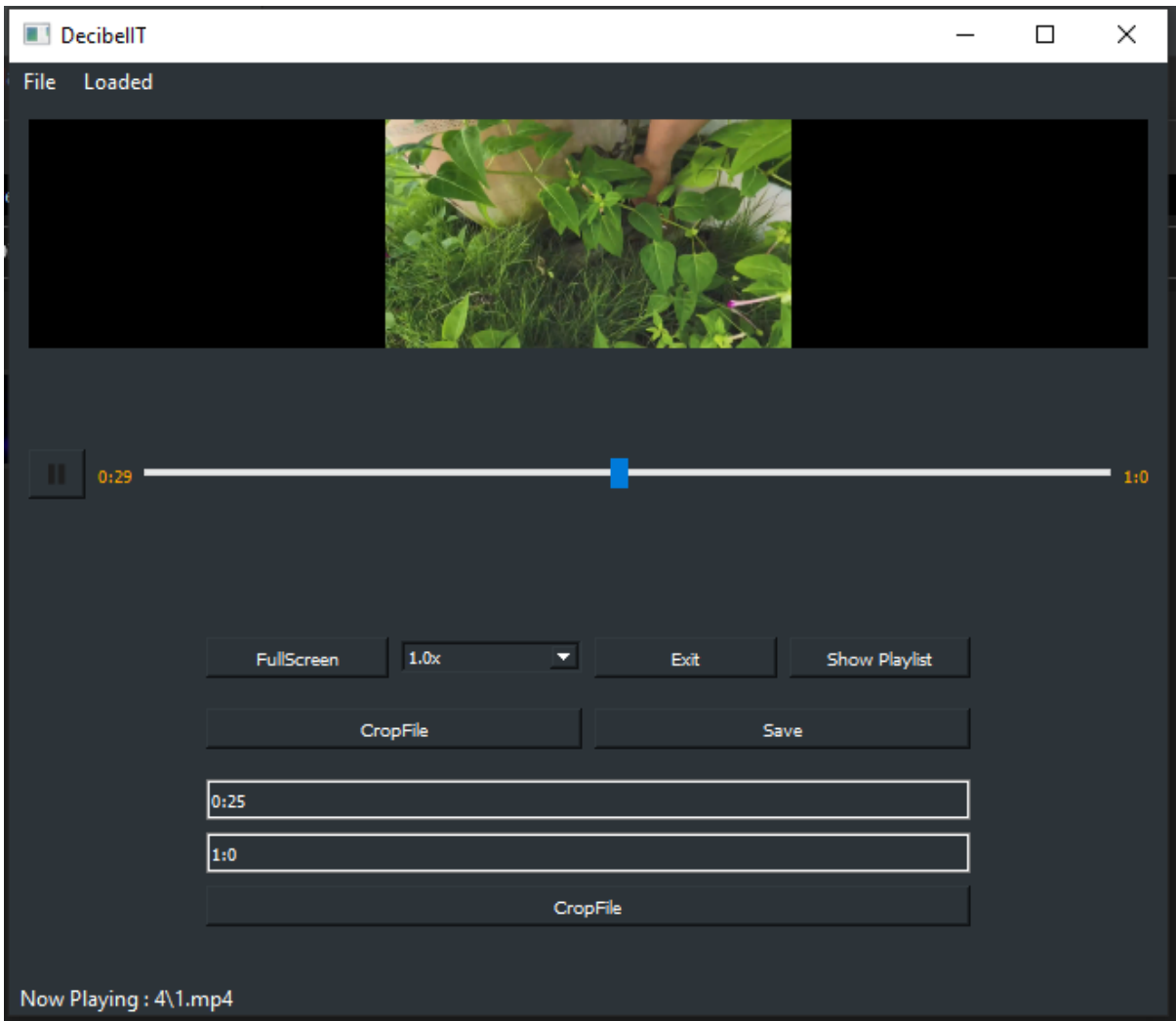
-

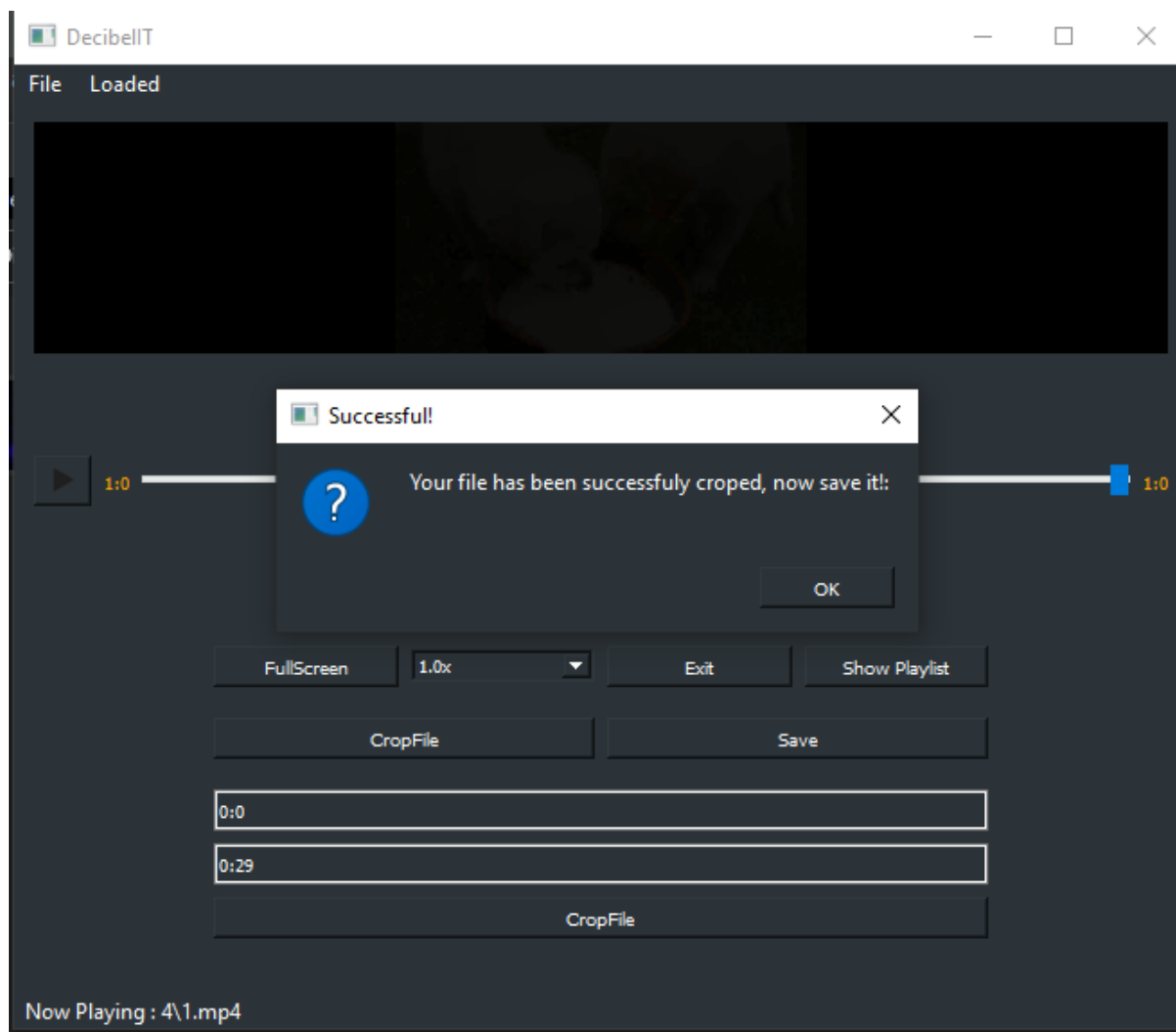












6. Prezentarea Dependințelor Aplicației

Pentru a putea reda sau edita videoclipurile din cadrul aplicației este necesară instalarea unui codec (pentru comprimare și decompimare) pe sistemul de operare. Pentru a rula programul cu succes, s-au folosit K-Lite Codec Pack 16.7.0 pentru Windows 10 și clementine pentru Linux Ubuntu 20.04.

Pentru implementarea aplicației s-au folosit limbajul de programare Python, atât framework-ul PyQt5, cât și librăria MoviePy.

Versiunea de Python utilizată este **Python 3.8**, însă se pot folosi versiuni mai noi precum 3.9 sau 3.10, disponibile pe site-ul oficial python.org, în secțiunea Downloads.

Framework-ul **PyQt5** este un set de librării C++ folosite în special pentru crearea de interfețe cu utilizatorul. Pentru acest proiect s-a folosit ultima versiune care este disponibilă din Octombrie 2021. Instalarea acesteia se face folosind linia de comandă cu drepturi de administrator, folosind pachetul PIP. Se rulează sintaxa: *pip install pyqt5*.

Librăria MoviePy este un modul Python utilizat pentru editare video. Este un software Open Source, scris inițial de Zulko, însă putându-se aduce modificări. Aceasta se poate instala într-un mod facil din linia de comandă cu drepturi de administrator, utilizând instrucțiunea: *pip install moviepy*.

7. Rolurile în Echipă

Diana Buliga:

- Încărcarea unui singur fișier
- Stilizarea aplicației
- Realizare Slider și salvarea dinamică a timestamp-ului unui fișier
- Realizare buton de stabilirea vitezei de redare
- Realizare funcționalitate fullscreen

Alberto Toscaru:

- Încărcarea unui director de fișiere
- Realizare funcționalitate status bar
- Realizare funcționalitate playlist
- Reținerea dimensiunii aplicației dintr-o rulare anterioară
- Realizare buton de save și funcționalitatea acestuia

Bianca Braha:

- Designul inițial al aplicației
- Încărcarea unui director de fișiere
- Realizarea buton de corp și funcționalitatea acestuia
- Realizare playlist
- Realizare funcționalitate buton de loaded media pentru rulare fără playlist

Toata lumea:

- Quality Assurance permanent