

Cody Hager and Theodore Bagley

CSCI 331 Final Project

String Scribe

An AI music transcription app. Users can upload a file and view the generated sheet music.

Tech Stack:

Frontend

- React
- Typescript
- MUI Component Library

Backend

- Python
- FastAPI Web Framework
- Spotify Basic Pitch Algorithm
(AI transcription)

Deployment

- Fly
- Squarespace (for domain name)

<https://stringscribe.com>

String Scribe

An AI music transcription app. Users can upload a file and view the generated sheet music.

Final Project Objectives:

Simple User Account Management

Use Auth0 to provide a way for users to create accounts or sign in with Google.

Paid Subscription Plan

Use Stripe for processing user payments, and link the user's subscription plan to their Auth0 account.

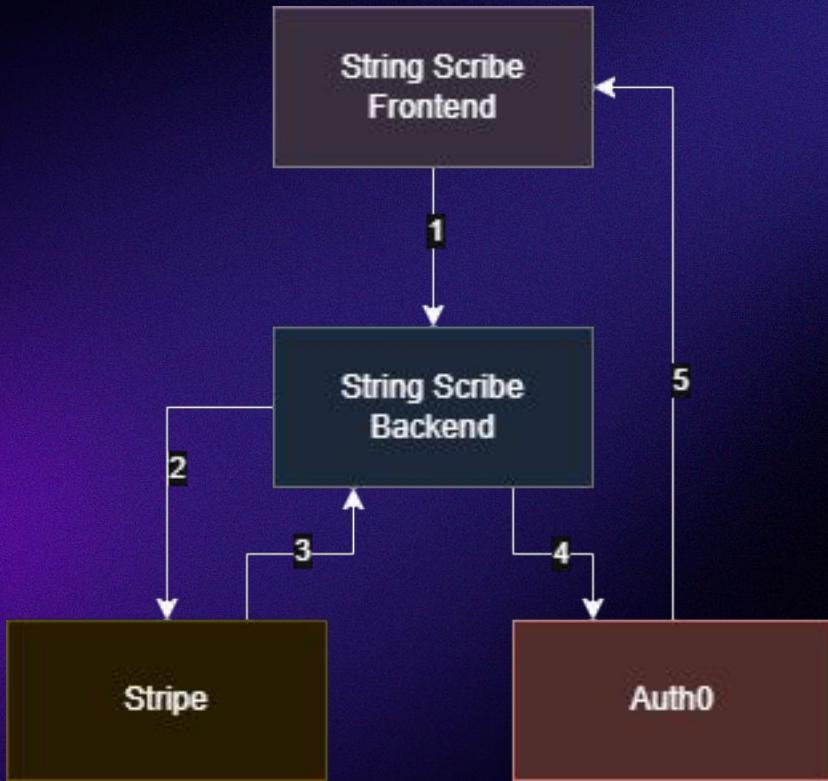
Limit Usage Based on Subscription Plan

Still allow unauthenticated users to transcribe a limited number of files. Use cookies/local storage to determine when they have exceeded the limit.

YouTube Uploads

Implement the ability to upload YouTube links and translating the retrieved audio file related to the uploaded link using youtube-dlp.

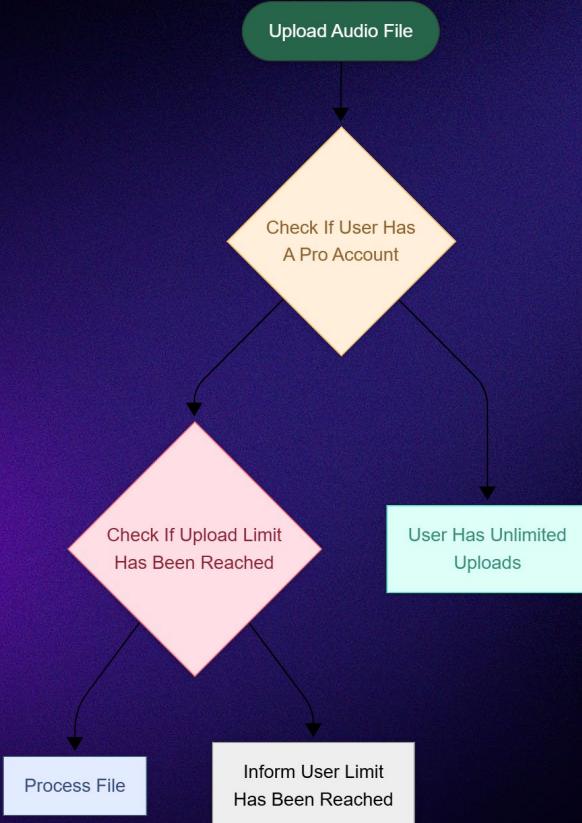
Auth0 and Stripe Integration



Processing Subscription: Order of Operations

1. Frontend makes POST to backend
2. Backend redirects to Stripe
3. When payment is complete, Stripe hits Webhook in backend
4. Backend adds "Pro" role to user in Auth0
5. Frontend can view new role on user

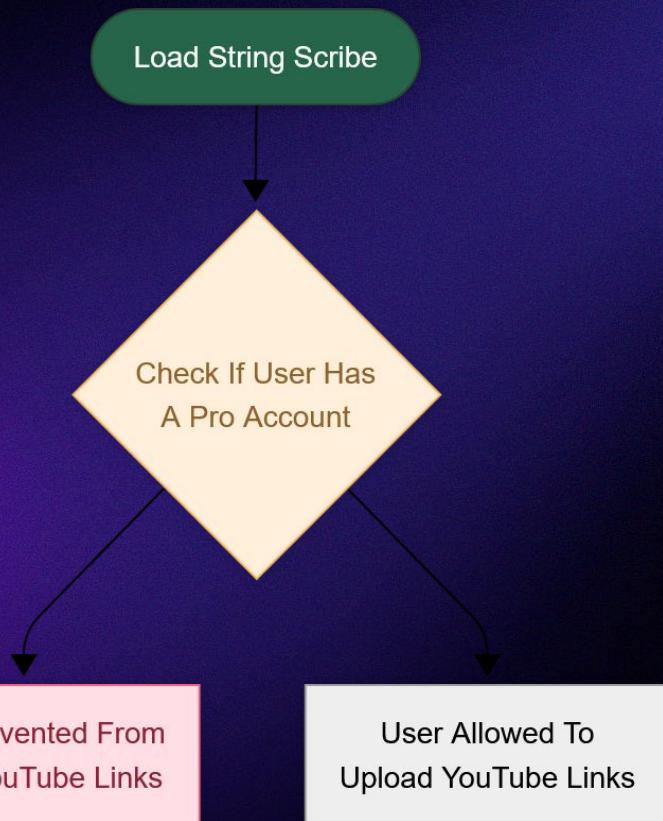
Paywall Implementation: Upload Limit



Paywall Interaction:

- Frontend sends POST to the Backend
- Backend checks if the user is a pro member
 - If they are, they have unlimited uploads
- If user is not a pro member, the Backend checks if a cookie session has started
 - If not, it starts one
- Backend translates audio file
- Backend increments counter tracking upload amount
- Frontend displays sheet music

YouTube Upload Feature



YouTube Upload Interaction:

- Frontend sends POST with link url and user ID
- Backend extracts url and user ID
- Backend checks if it is a pro user
- Backend downloads the audio file from YouTube DLP
- Backend does the translation process
- Backend sends a JSON file containing the sheet music generated
- Frontend displays sheet music

Takeaways

Cody:

- Auth0 is the most simple access management platform out of the ones I have tried for small projects like this
- Stripe is more complicated than I was hoping and required lots of reading docs/Googling.

Teddy:

- Setting up the cookie tracking was more difficult than I expected. Interesting considering I believed it to be the easiest way to set up a paywall system
- Setting up YouTube audio downloading required two components: the yt-dlp Python library for fetching videos from YouTube, and FFmpeg (a separate multimedia processing tool). This made it very simple to set up.

Live Demo!

Site: <https://stringscribe.com>

GitHub:
<https://github.com/CodyHager/string-scribe>

References/Dependencies

- <https://fastapi.tiangolo.com/>
- <https://mui.com/>
- <https://fly.io>
- <https://github.com/spotify/basic-pitch>
- <https://auth0.com/>
- <https://stripe.com/>
- <https://github.com/opensheetmusicdisplay/opensheetmusicdisplay>