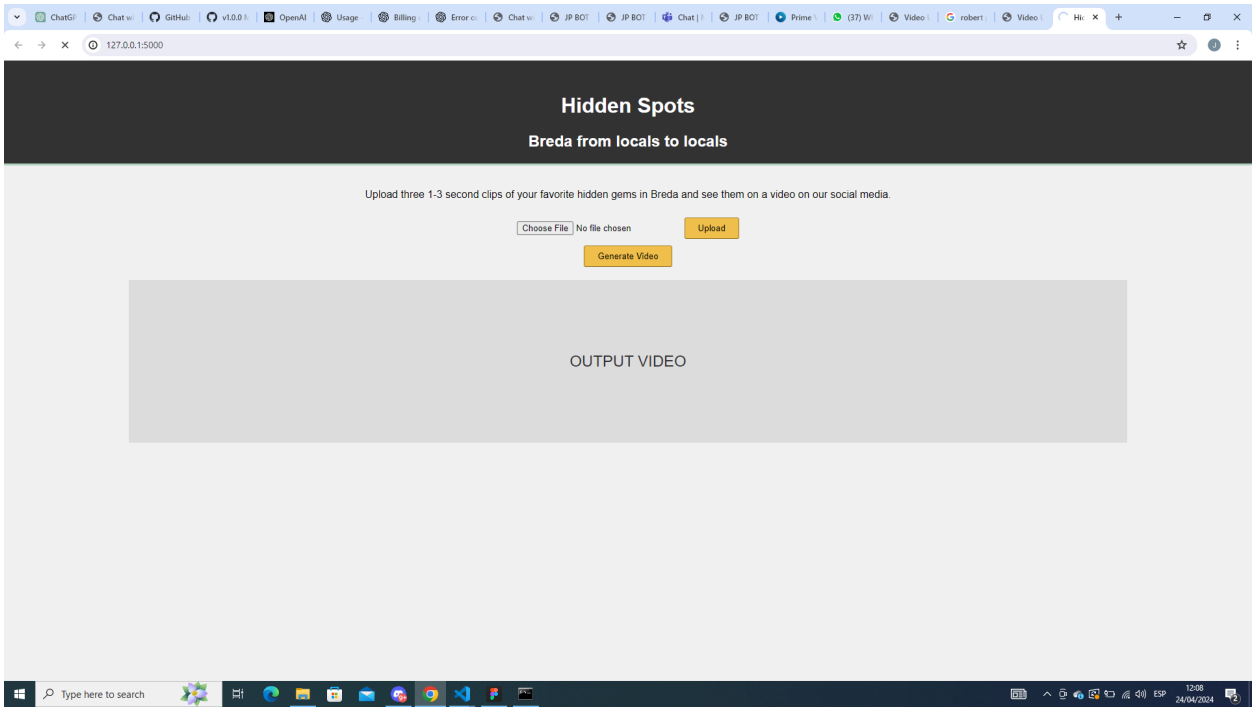
Hidden Spots First attempt:

Side Info - first attempt:
This project had a large process, i think it pertains to development to mention that before coding this site for project 3, i tried to finish coding it to present this prototype to the clients in project 2, and for this i tried to code the functionality in python, a language i'm much more familiar with, so i coded the local upload part, and also i had this idea that having the site make the video compilations alone would generate much more content and make the life of the CMs much easier, so i also did that. (story behind this idea in project 2, LO3)
This idea stayed as an idea and i didnt work on it any further since i did not finish it on time for the presentation and then pivoted to the other concept of simply receiving videos.



what the flask app did was:

Setup:

- It sets up a place to save uploaded video files.
- Install relevant python libraries.
- It allows only certain types of video files (like .mp4, .avi, and .mov). (the final project 3 one only accepted .mov files)
- Sets secret key for security.

Upload:

- upload video files on main page.
- The app checks if the file is a valid video type.
- If valid, the video is saved in the uploads folder.

Generate video:

- There's a button or link to create a video montage from the uploaded videos (uses all videos in the uploads folder).
- The app combines the uploaded videos into one single video. (if i continued with this idea i would have put a clip limit or time limit for the final video)
- The combined video is saved, and displayed on the site for the user to download.

Resetting:

- There's an option to reset, which clears the page to upload videos again.

There were many faults and things to work but i think is a nice idea to present here as an extra.

lll only show the code for the flask app sinse the HTML was just simple to see functionality and no design at all.

```
from flask import Flask, request, redirect, url_for, render_template,
session
from werkzeug.utils import secure_filename
import os
from video_combiner import create_video_montage

app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = 'static/uploads/'
app.config['ALLOWED_EXTENSIONS'] = {'mp4', 'avi', 'mov'}
app.secret_key = '12345678'

def allowed_file(filename):
    return '.' in filename and filename.rsplit('.', 1)[1].lower() in
app.config['ALLOWED_EXTENSIONS']

@app.route('/', methods=['GET', 'POST'])
def upload_file():
    video_file = None
    if request.method == 'POST':

        session.pop('video_file', None)

        if 'file' not in request.files:
```

```
            return redirect(request.url)
        file = request.files['file']

        if file.filename == '' or not allowed_file(file.filename):
            return redirect(request.url)

        if file and allowed_file(file.filename):
            filename = secure_filename(file.filename)
            file_path = os.path.join(app.config['UPLOAD_FOLDER'],
filename)
            file.save(file_path)
            return redirect(url_for('upload_file'))

    video_file = session.get('video_file', None)

    return render_template('index.html', video_file=video_file)

@app.route('/generate', methods=['GET'])
def generate_video():
    video_path = create_video_montage(app.config['UPLOAD_FOLDER'])

    session['video_file'] = video_path

    return redirect(url_for('upload_file'))

@app.route('/download/<filename>', methods=['GET'])
def download(filename):

    return send_from_directory(app.config['UPLOAD_FOLDER'], filename,
as_attachment=True)

@app.route('/reset', methods=['GET'])
def reset():

    session.pop('video_file', None)
    return redirect(url_for('upload_file'))

if __name__ == "__main__":
    app.run(debug=True)
```