UNMASKED

Presented by Team 2

MEET THE TEAM

Mentor - Dr. Anurag Goswami

Anushka Shishodia - E22CSEU1014(Team Leader)
Anurakta Dash - E22CSEU0996
Ishika Singhal - E22CSEU1542
Ridhim Dubey - E22CSEU1009

DEEPFAKE?

1. Deepfakes: Advanced technology replacing faces in videos/images with remarkable accuracy.

2. Implications: Potential for spreading false information and manipulating public perception.

3. Example: A political deepfake could create confusion or sway opinions during an election.

4. Impact: Raises ethical concerns about trust, authenticity, and media manipulation.

PROPOSED SOLUTION: UNMASKED

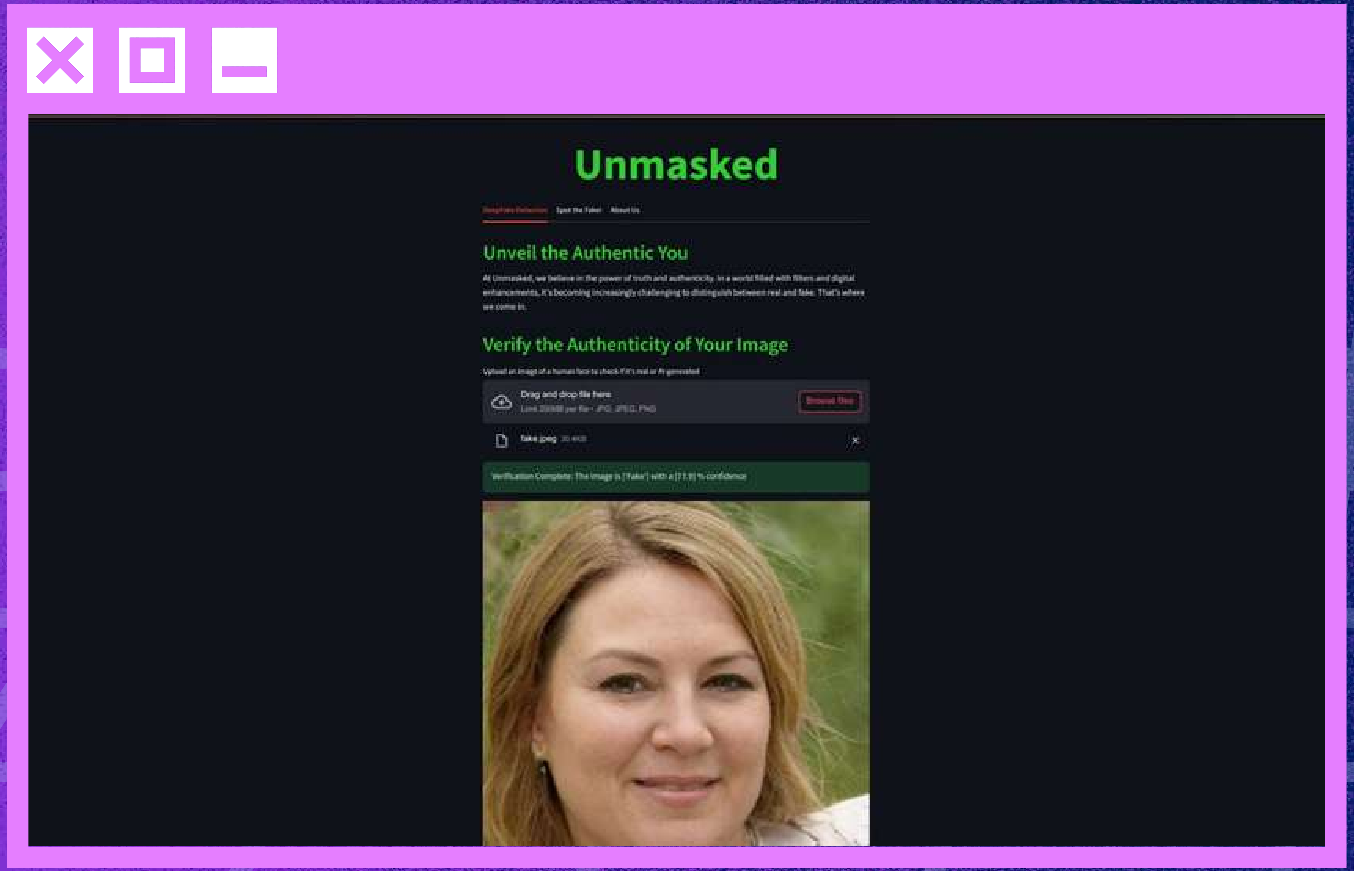
Our solution is a detection system that uses advanced computer vision algorithms by detecting inconsistencies in facial expressions, lighting, and audio quality.

Let's begin!

FUNCTIONALITIES

Our Homepage

Here we have an upload button where we upload the image and it predicts whether the image is fake or real.



Unmasked

Unveil the Authentic You

At Unmasked, we believe in the power of truth and authenticity. In a world filled with filters and digital enhancements, it's becoming increasingly challenging to distinguish between real and fake. That's where we come in.

Verify the Authenticity of Your Image

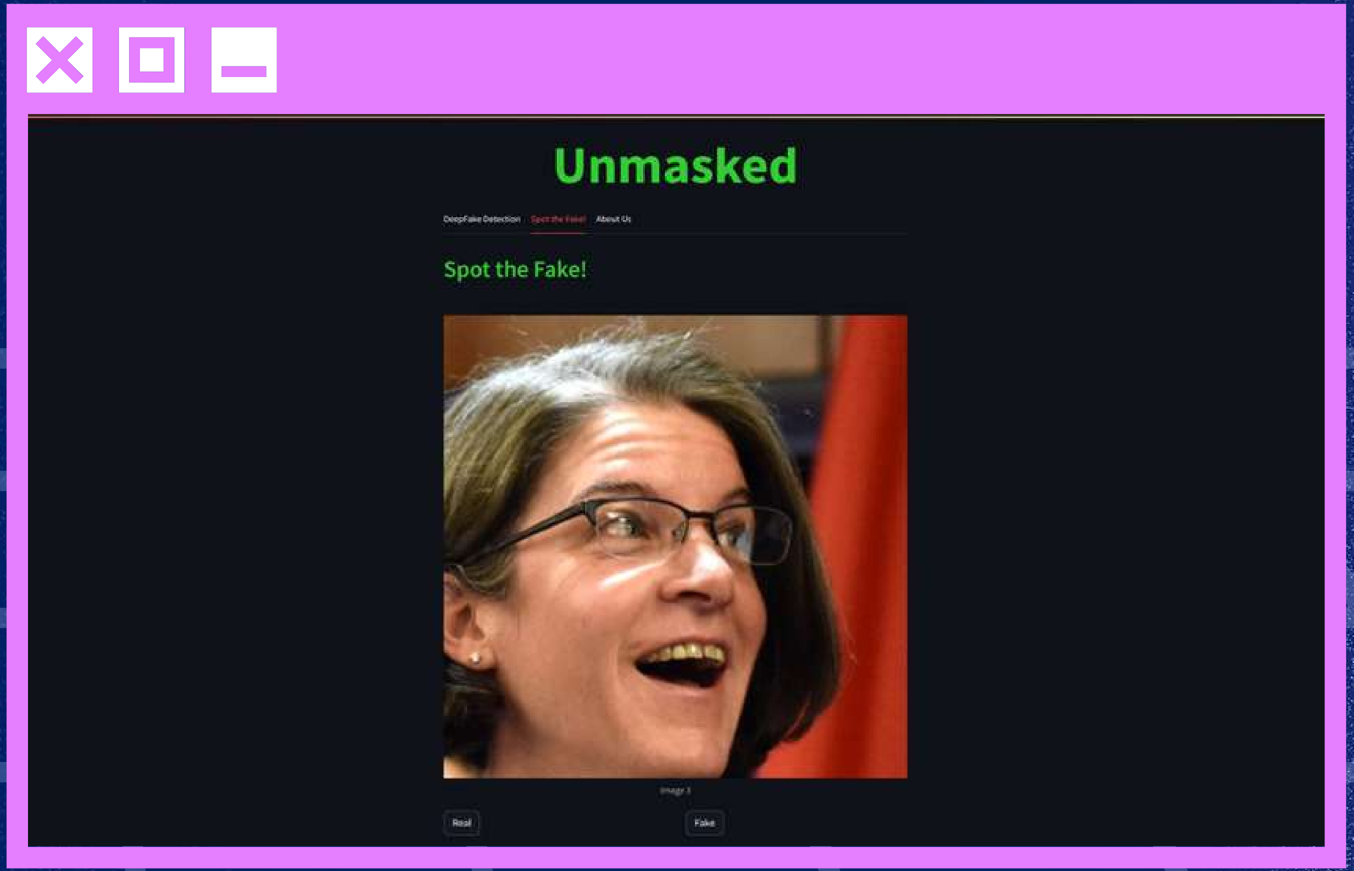
Upload an image of a human face to check if it's real or AI-generated

fake.jpeg 30.4KB

Verification Complete: The image is ['Fake'] with a [71.9] % confidence

Spot the fake!

We have added an extra functionality as a game to make the users aware about how a Deepfake image looks like and increase their knowledge about the same



Unmasked

Spot the Fake!

Image 3

TECHNOLOGY STACK

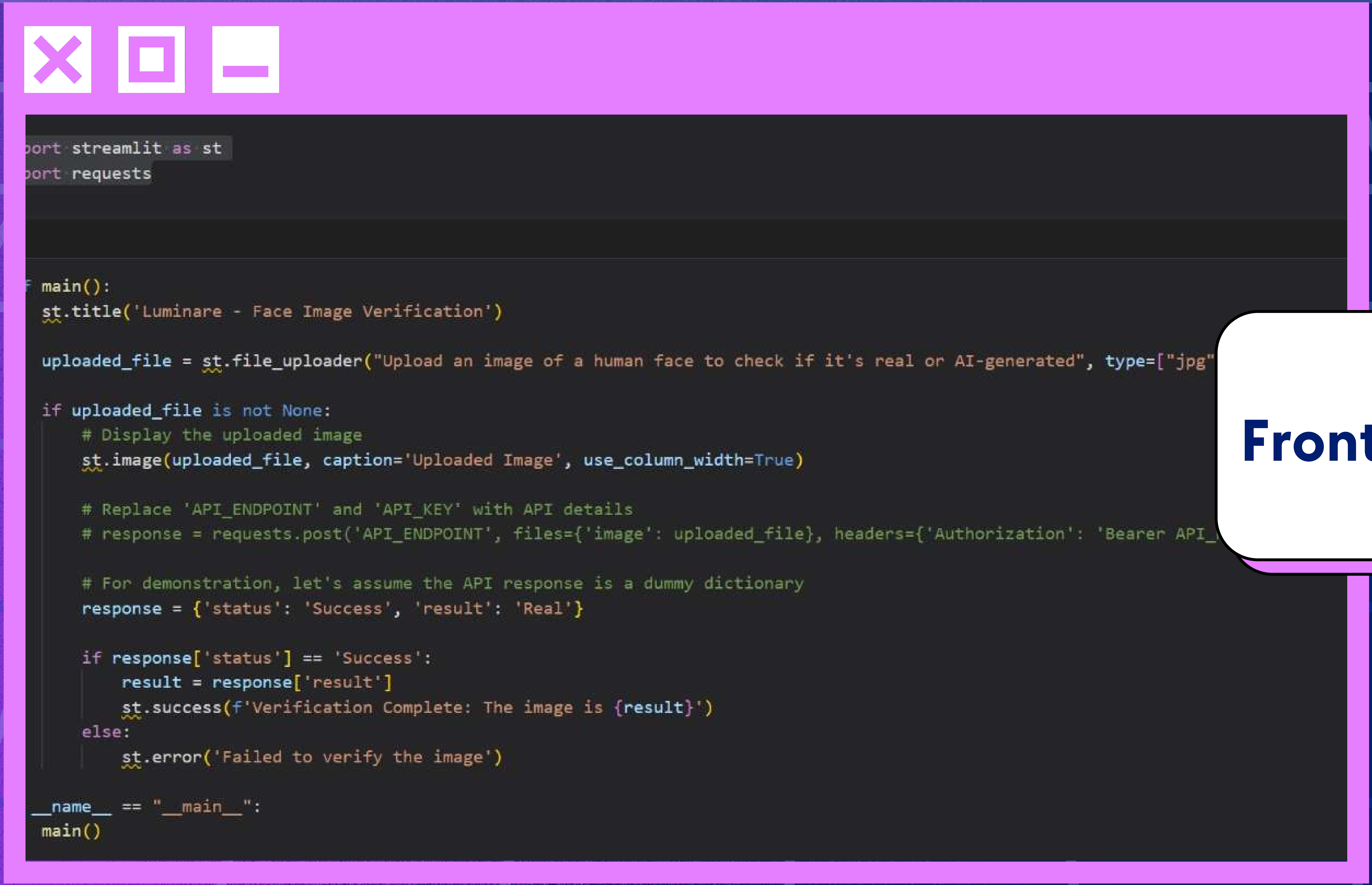
- Python Programming Language- provides open extensive libraries.
- TensorFlow aids advanced machine learning and model development.
- Matplotlib, Plotly visualize data comprehensively for in-depth analysis.
- Flask - web framework in python that handles HTTP requests.
- CNN and its architechtures like VGG16 and RESTful50 for image classification.

CHALLENGES IN DEEPFAKE DETECTION

Hardware Requirement: GPUs or specialised hardware system to train and deploy.Finding a wide range of devices and hardware configurations was challenging.

Data Collection and Quality: Acquiring a large dataset of high quality images to train the deep learning models is crucial. This process can be time consuming and expensive.

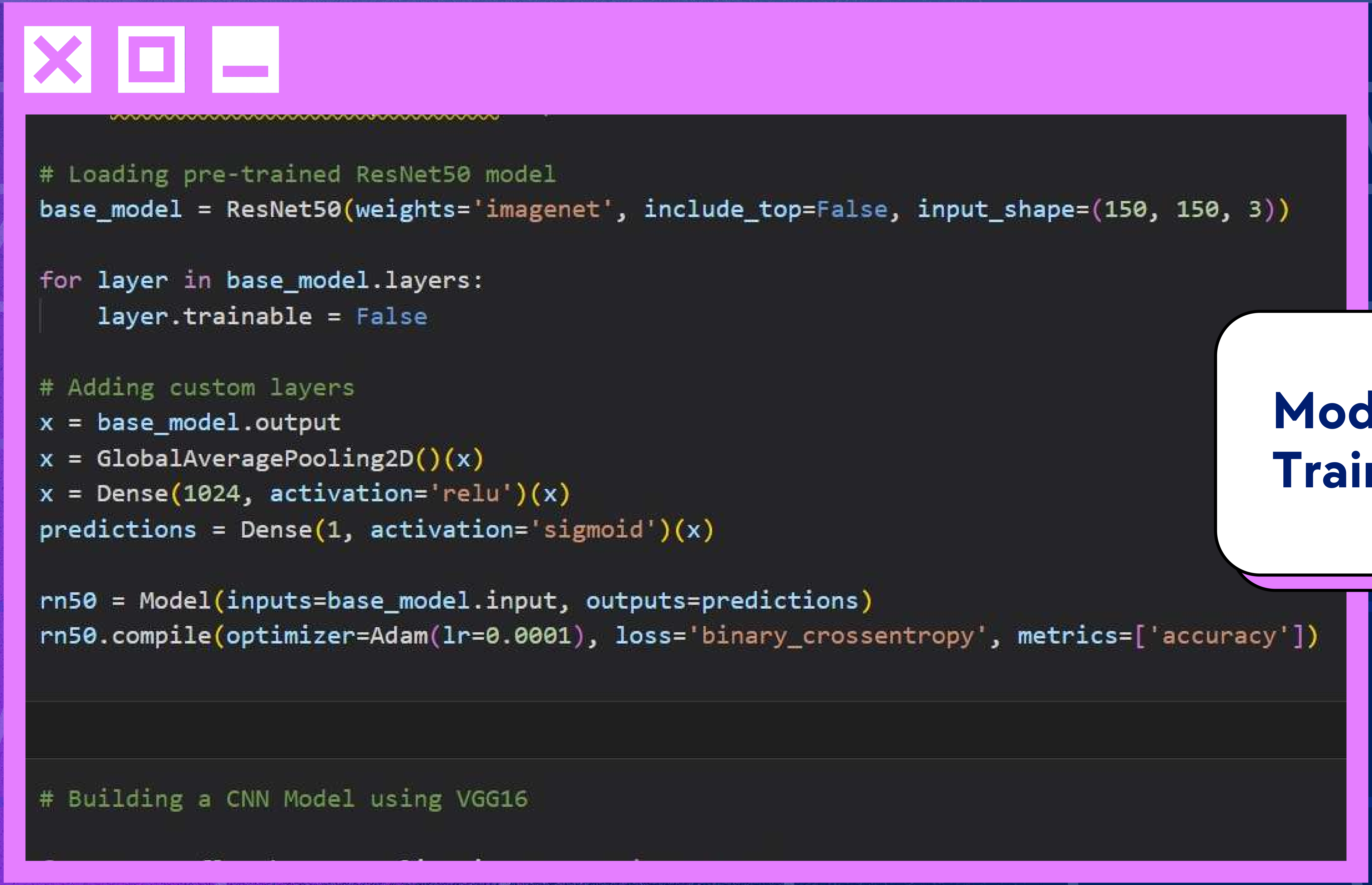
Algorithmic complexity : Developing a high end algorithms capable of accurately swapping faces while it requires expertise in ML ,computer vision.



```python
port streamlit as st
port requests


f main():
    st.title('Luminare - Face Image Verification')

    uploaded_file = st.file_uploader("Upload an image of a human face to check if it's real or AI-generated", type=["jpg"

    if uploaded_file is not None:
        # Display the uploaded image
        st.image(uploaded_file, caption='Uploaded Image', use_column_width=True)

        # Replace 'API_ENDPOINT' and 'API_KEY' with API details
        # response = requests.post('API_ENDPOINT', files={'image': uploaded_file}, headers={'Authorization': 'Bearer API_

        # For demonstration, let's assume the API response is a dummy dictionary
        response = {'status': 'Success', 'result': 'Real'}

        if response['status'] == 'Success':
            result = response['result']
            st.success(f'Verification Complete: The image is {result}')
        else:
            st.error('Failed to verify the image')

__name__ == "__main__":
    main()
```

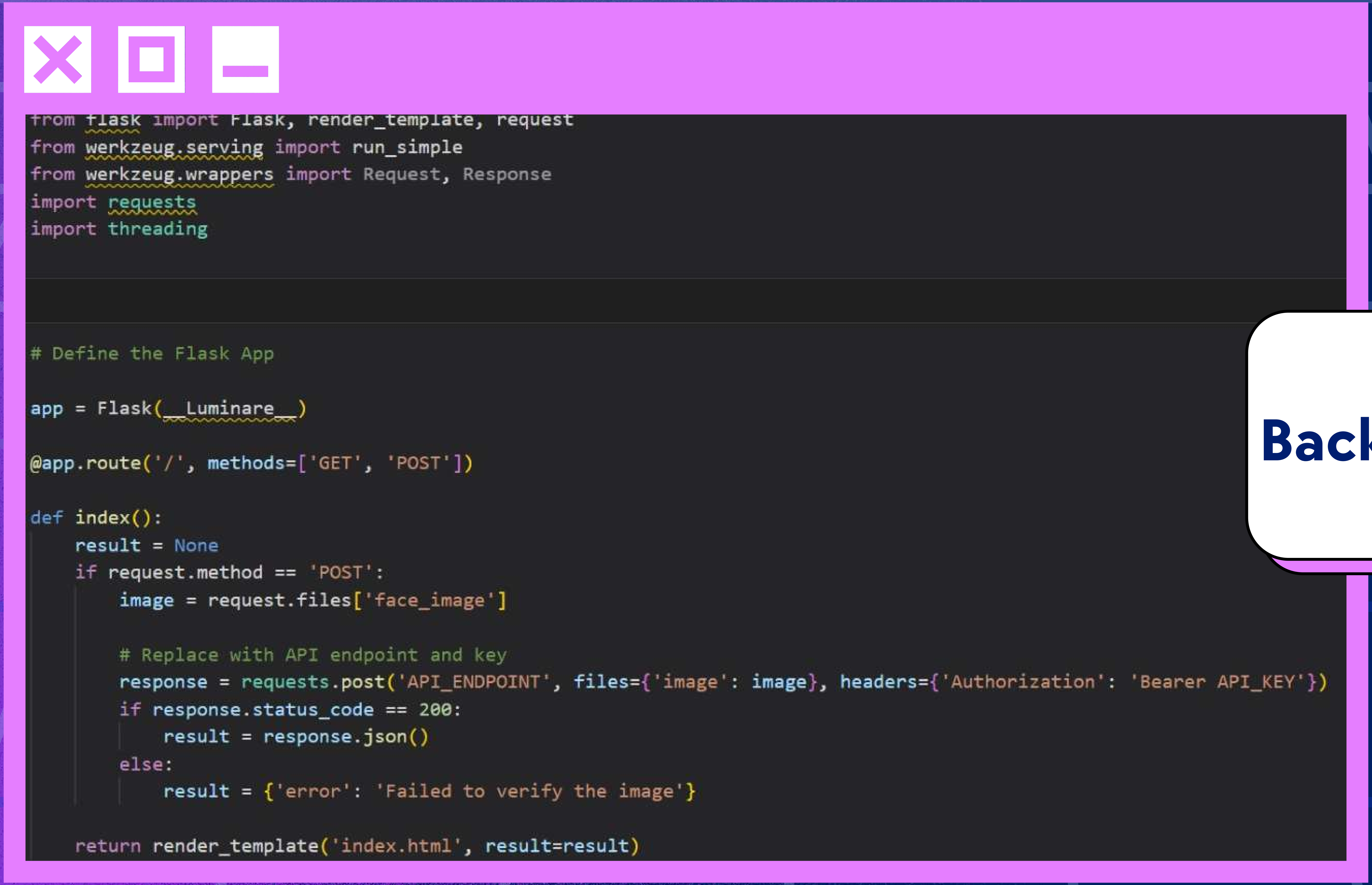
Front end

Model Training



```
# Loading pre-trained ResNet50 model
base_model = ResNet50(weights='imagenet', include_top=False, input_shape=(150, 150, 3))

for layer in base_model.layers:
    layer.trainable = False

# Adding custom layers
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(1024, activation='relu')(x)
predictions = Dense(1, activation='sigmoid')(x)

rn50 = Model(inputs=base_model.input, outputs=predictions)
rn50.compile(optimizer=Adam(lr=0.0001), loss='binary_crossentropy', metrics=['accuracy'])
```

```
# Building a CNN Model using VGG16
```



```
from flask import Flask, render_template, request
from werkzeug.serving import run_simple
from werkzeug.wrappers import Request, Response
import requests
import threading



# Define the Flask App

app = Flask(__Luminare__)

@app.route('/', methods=['GET', 'POST'])

def index():
    result = None
    if request.method == 'POST':
        image = request.files['face_image']

        # Replace with API endpoint and key
        response = requests.post('API_ENDPOINT', files={'image': image}, headers={'Authorization': 'Bearer API_KEY'})
        if response.status_code == 200:
            result = response.json()
        else:
            result = {'error': 'Failed to verify the image'}

    return render_template('index.html', result=result)
```

Back-end

FUTURE ENHANCEMENTS

Deep Learning Models

Advanced deep learning architechtures like BERT or GPT which may offer improved performance in detecting subtle cues and patterns that indicates deepfake manipulation.

Multi-Modal fusion

Explore techniques for integrating information from multiple modalities (audio, text, video) to improve the robustness and accuracy of deepfake detection systems.

Continual Learning

Develop algorithms for continual learning to adapt to evolving deepfake generation techniques and maintain detection effectiveness over time.

THANK YOU!