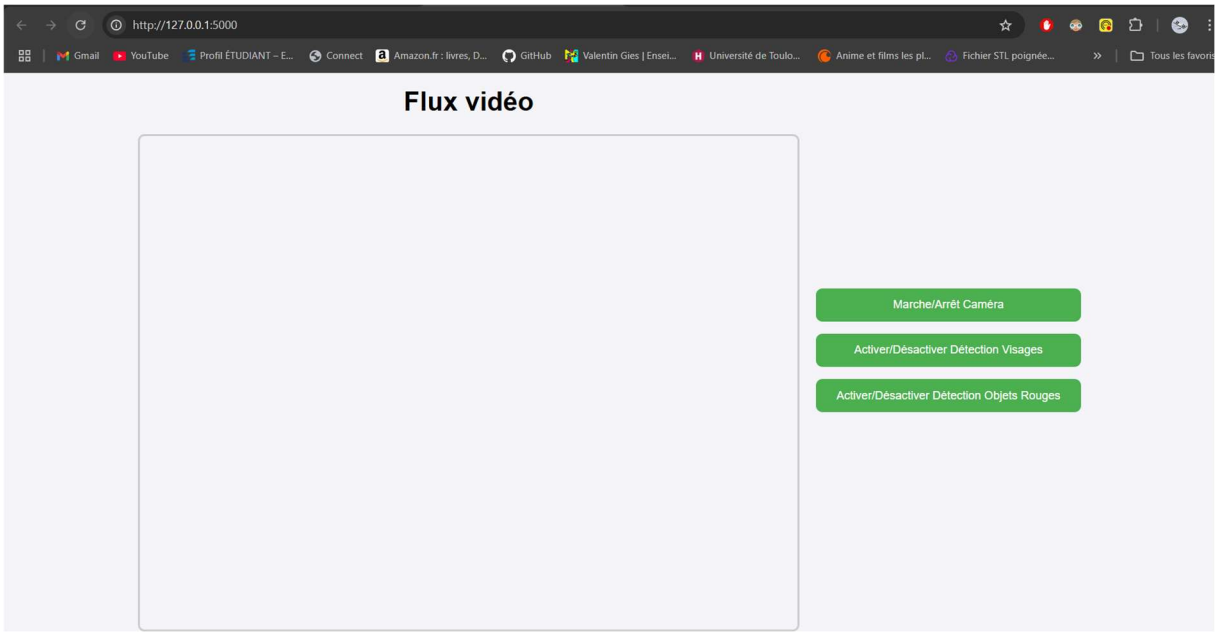


```python
#La résolution de la vidéo a été réduite à 640x480.
#La compression JPEG a été appliquée à 80% pour réduire la taille des images transmises et améliorer la
from flask import Flask, Response, render_template_string
import cv2
import numpy as np
import threading
import os
import signal
import webbrowser

app = Flask(__name__)

# Variables de contrôle pour les détections
detect_faces = True
detect_red_objects = True
camera_running = False
camera = None
camera_index = 2  # Caméra USB à l'index 2 (selectionner la camera utilisé)

# Charger le modèle de détection de visages
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + "haarcascade_frontalface_default.xml")

def detect_red_objects_and_faces(frame):
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)

    # Détection des objets rouges
    if detect_red_objects:
        lower_red1 = np.array([0, 120, 70], dtype=np.uint8)
        upper_red1 = np.array([10, 255, 255], dtype=np.uint8)
        lower_red2 = np.array([170, 120, 70], dtype=np.uint8)
        upper_red2 = np.array([180, 255, 255], dtype=np.uint8)

        mask_red1 = cv2.inRange(hsv, lower_red1, upper_red1)
        mask_red2 = cv2.inRange(hsv, lower_red2, upper_red2)
        mask_red = mask_red1 + mask_red2

        kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (11, 11))
        mask_red = cv2.erode(mask_red, kernel, iterations=2)
        mask_red = cv2.dilate(mask_red, kernel, iterations=2)
        mask_red = cv2.GaussianBlur(mask_red, (3, 3), 0)
```