from fastapi import FastAPI, HTTPException, Request, UploadFile, File

import stripe

import json

import os

import smtplib

from email.mime.text import MIMEText

from email.mime.multipart import MIMEMultipart

from dotenv import load\_dotenv

from diffusers import AnimateDiffPipeline, StableVideoDiffusionPipeline

from PIL import Image

import shutil

from firebase\_admin import credentials, initialize\_app, storage

load\_dotenv()

app = FastAPI()

# Stripe API Key

stripe.api\_key = os.getenv("STRIPE\_SECRET\_KEY")

STRIPE\_WEBHOOK\_SECRET = os.getenv("STRIPE\_WEBHOOK\_SECRET")

# Firebase Storage Setup

cred = credentials.Certificate("firebase\_credentials.json")

initialize\_app(cred, {"storageBucket": os.getenv("FIREBASE\_BUCKET")})

# AI Models

i2v\_pipe = AnimateDiffPipeline.from\_pretrained("ByteDance/AnimateDiff").to("cuda")

t2v\_pipe = StableVideoDiffusionPipeline.from\_pretrained("stabilityai/stable-video-diffusion").to("cuda")

UPLOAD\_DIR = "uploads"

os.makedirs(UPLOAD\_DIR, exist\_ok=True)

@app.post("/image-to-video/")

async def image\_to\_video(file: UploadFile = File(...)):

filepath = f"{UPLOAD\_DIR}/{file.filename}"

with open(filepath, "wb") as buffer:

shutil.copyfileobj(file.file, buffer)

image = Image.open(filepath).convert("RGB")

video = i2v\_pipe(image, num\_inference\_steps=50, num\_frames=150)

output\_video = "output\_video.mp4"

video[0].save(output\_video)

bucket = storage.bucket()

blob = bucket.blob(f"videos/{file.filename}.mp4")

blob.upload\_from\_filename(output\_video)

video\_url = blob.public\_url

return {"video\_url": video\_url}