# **import os**

import os

import logging

import yt\_dlp

import requests

from aiogram import Bot, Dispatcher, executor, types

# Tokenlar (Render -> Environment Variables ichiga qo'yiladi)

BOT\_TOKEN = os.getenv("BOT\_TOKEN")

logging.basicConfig(level=logging.INFO)

bot = Bot(token=BOT\_TOKEN)

dp = Dispatcher(bot)

# HuggingFace API (bepul model)

HF\_API\_URL = "https://api-inference.huggingface.co/models/facebook/blenderbot-400M-distill"

HF\_API\_KEY = ""  # Agar token bo'lsa qo'shish mumkin, bo'lmasa ham ishlaydi

headers = {"Authorization": f"Bearer {HF\_API\_KEY}"} if HF\_API\_KEY else {}

# /start komandasi

@dp.message\_handler(commands=['start'])

async def start\_cmd(message: types.Message):

    await message.reply("Salom! 馃 Men AI botman.\n\n"

                        "馃挰 Savol-javob qilish uchun yozing\n"

                        "馃柤 Rasm yaratish: /draw <matn>\n"

                        "馃幍 Musiqa nomini topish: audio yuboring\n"

                        "馃摜 Musiqa yuklab olish: /music <nomi>")

# 馃挰 Chat (matnli suhbat)

@dp.message\_handler(lambda msg: not msg.text.startswith(("/",)))

async def chat\_ai(message: types.Message):

    payload = {"inputs": {"text": message.text}}

    try:

        response = requests.post(HF\_API\_URL, headers=headers, json=payload, timeout=60)

        data = response.json()

        if isinstance(data, list):

            answer = data[0]["generated\_text"]

        else:

            answer = data.get("generated\_text", "Javob topilmadi 馃様")

        await message.answer(answer)

    except Exception as e:

        await message.answer("鉂€ Javob olishda xatolik yuz berdi.")

# 馃柤 Rasm yaratish

@dp.message\_handler(commands=['draw'])

async def draw\_cmd(message: types.Message):

    prompt = message.get\_args()

    if not prompt:

        await message.reply("鉂€ Rasm yaratish uchun matn yozing: /draw mushuk kosmosda")

        return

    try:

        url = f"<https://image.pollinations.ai/prompt>/{prompt.replace(' ', '%20')}"

        await bot.send\_photo(message.chat.id, url, caption=f"馃柤 Rasm yaratildi: {prompt}")

    except:

        await message.reply("鉂€ Rasm yaratishda xatolik.")

# 馃幍 Musiqa aniqlash (oddiy demo 鈥€ audio nomini qaytaradi)

@dp.message\_handler(content\_types=types.ContentType.AUDIO)

async def recognize\_music(message: types.Message):

    audio = message.audio

    await message.reply(f"馃幎 Siz yuborgan audio: {audio.title or 'Noma鈥檒um'} 鈥€ {audio.performer or ''}")

# 馃摜 Musiqa yuklab olish (YouTube-dan)

@dp.message\_handler(commands=['music'])

async def download\_music(message: types.Message):

    query = message.get\_args()

    if not query:

        await message.reply("鉂€ Musiqa nomini yozing: /music Sezen Aksu Hadi Bakalim")

        return

    await message.reply("鈴€ Musiqa qidirilmoqda...")

    try:

        ydl\_opts = {

            'format': 'bestaudio/best',

            'noplaylist': True,

            'quiet': True,

            'outtmpl': 'song.%(ext)s',

            'postprocessors': [{

                'key': 'FFmpegExtractAudio',

                'preferredcodec': 'mp3',

                'preferredquality': '192',

            }],

        }

        with yt\_dlp.YoutubeDL(ydl\_opts) as ydl:

            info = ydl.extract\_info(f"ytsearch:{query}", download=True)['entries'][0]

            file\_name = ydl.prepare\_filename(info).replace(".webm", ".mp3")

            await bot.send\_audio(message.chat.id, open(file\_name, 'rb'), title=info.get("title"))

            os.remove(file\_name)

    except Exception as e:

        await message.reply("鉂€ Musiqa yuklab bo鈥榣madi.")

if \_\_name\_\_ == "\_\_main\_\_":

    executor.start\_polling(dp, skip\_updates=True)