Main.py

import asyncio  
from aiogram import Bot, Dispatcher  
from aiogram.filters import CommandStart, Command  
from aiogram.types import Message, FSInputFile  
from config import TOKEN, API\_KEY  
  
bot = Bot(token=TOKEN)  
dp = Dispatcher()  
  
  
  
async def main():  
 await dp.start\_polling(bot)  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 asyncio.run(main())

import asyncio  
from aiogram import Bot, Dispatcher  
from aiogram.filters import CommandStart, Command  
from aiogram.types import Message, FSInputFile  
from config import TOKEN, API\_KEY  
import os  
import sqlite3  
from datetime import datetime  
  
DB\_PATH = "data/db.sqlite3"  
def ensure\_db():  
 dirpath = os.path.dirname(DB\_PATH)  
 if dirpath:  
 os.makedirs(dirpath, exist\_ok=True)  
  
 with sqlite3.connect(DB\_PATH) as conn:  
 cur = conn.cursor()  
 cur.execute("""  
 CREATE TABLE IF NOT EXISTS users (  
 id INTEGER PRIMARY KEY AUTOINCREMENT,  
 telegram\_id INTEGER UNIQUE,  
 first\_name TEXT,  
 last\_name TEXT,  
 username TEXT,  
 registered\_at TEXT,  
 last\_seen TEXT)""")  
 conn.commit()  
  
ensure\_db()  
  
def upsert\_user(telegram\_id: int, first\_name: str, last\_name: str, username:str):  
 now = datetime.utcnow().isoformat(timespec="seconds")  
 with sqlite3.connect(DB\_PATH) as conn:  
 cur = conn.cursor()  
 try:  
 cur.execute("""  
 INSERT INTO users(telegram\_id, first\_name, last\_name, username, registered\_at, last\_seen)  
 VALUES (?, ?, ?, ?, ?, ?)  
 """, (telegram\_id, first\_name, last\_name, username, now, now))  
 conn.commit()  
 return True  
 except sqlite3.IntegrityError:  
 cur.execute("""  
 UPDATE users SET last\_seen = ? WHERE telegram\_id = ?""", (now, telegram\_id))  
 conn.commit()  
 return False  
  
bot = Bot(token=TOKEN)  
dp = Dispatcher()  
  
  
  
@dp.message(Command('help'))  
async def help(message: Message):  
 await message.answer("Этот бот умеет выполнять команды:\n/start \n/help \n/practice \n/support")  
  
@dp.message(Command(commands=["start"]))  
async def start\_handler(message: Message):  
 user = message.from\_user  
 telegram\_id = user.id  
 first\_name = user.first\_name or ""  
 last\_name = user.last\_name or ""  
 username = user.username or ""  
  
 created = upsert\_user(telegram\_id, first\_name, last\_name, username)  
  
 if created:  
 await message.answer(  
 f"Привет, {first\_name}! Ты успешно зарегистрирован в SpeakSmart. "  
 "Это базовый прототип на этапе 1."  
 )  
 else:  
 await message.answer(  
 f"Привет, {first\_name}! Ты уже зарегистрирован в SpeakSmart."  
 )  
  
async def main():  
 await dp.start\_polling(bot)  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 asyncio.run(main())

config.py

TOKEN = "7703370722:AAGT5ikw2QSFV9xRt7Q3kN1lt8l8Ma1QhR8"  
  
API\_KEY = "\_"