

```

from aiogram import Bot, Dispatcher, types
from aiogram.types import Message
from aiogram.filters import Command
import asyncio
import torch
from transformers import pipeline

# Telegram API Token
API_TOKEN = "7736783071:AAEf15d0_mo8ZVzsUGdH1EOKQs3UEBXo15g"

# Load TinyLlama-Model
pipe = pipeline(
    "text-generation",
    model="TinyLlama/TinyLlama-1.1B-Chat-v1.0",
    torch_dtype=torch.bfloat16,
    device_map="auto"
)

# Initialise bot and dispatcher
bot = Bot(token=API_TOKEN)
dp = Dispatcher()

# Welcome handler
@dp.message(Command("start"))
async def welcome_handler(message: Message):
    await message.answer(f"Hi {message.from_user.first_name}. I am an AI chat assistant. How can I help you?")

# Thanks handler
@dp.message(Command("thanks"))
async def thanks_handler(message: Message):
    await message.answer("You are welcome. I am glad I could help.")

# General message handler with TinyLlama logic
@dp.message()
async def message_handler(message: Message):
    try:
        # User input
        user_message = message.text

        # Prepare inputs for TinyLlama
        messages = [
            {
                "role": "system",
                "content": "You are a friendly chatbot who always responds in the style of a pirate",
            },
            {"role": "user", "content": user_message},
        ]

        # Generate prompt for the model
        prompt = pipe.tokenizer.apply_chat_template(messages, tokenize=False, add_generation_prompt=True)

        # Generate response
        outputs = pipe(prompt, max_new_tokens=256, do_sample=True, temperature=0.7, top_k=50, top_p=0.95)
        generated_text = outputs[0]["generated_text"]

        # Get just actual response
        # Search for assistant marker and delete part before
        if "<|assistant|>" in generated_text:
            response = generated_text.split("<|assistant|>")[-1].strip()
        else:
            response = generated_text.strip()

        # Send response
        await message.answer(response)
    except Exception as e:
        # Error handler
        await message.answer(f"An error occurred: {str(e)}")

```

```

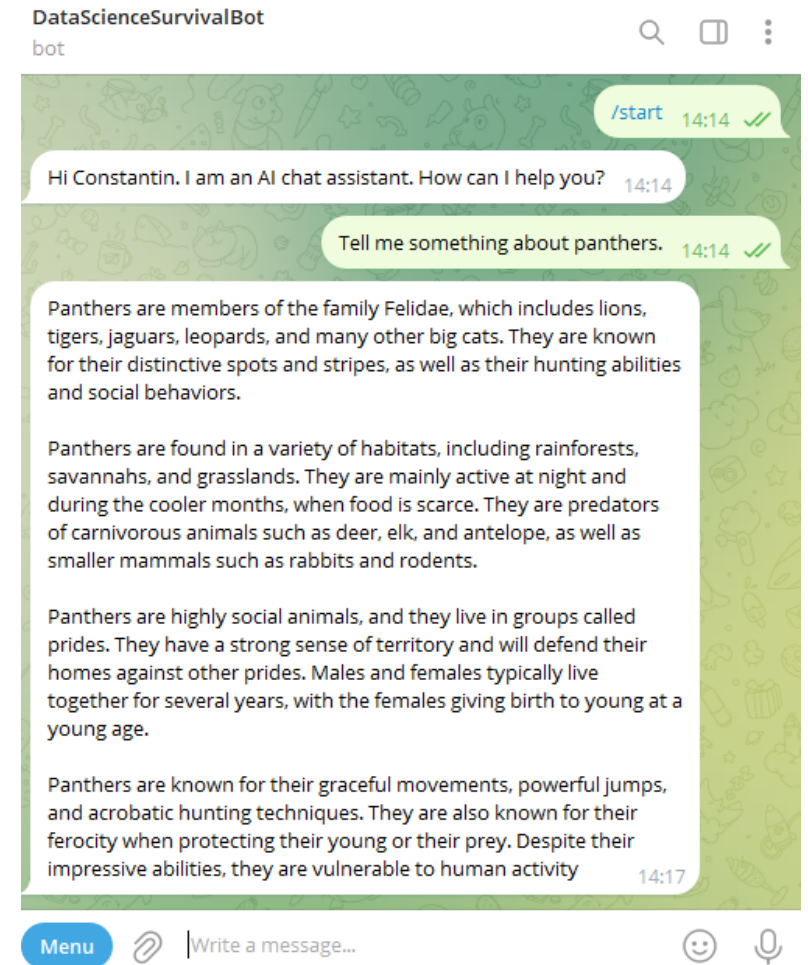
# Main function
async def main():
    print("Bot is running...")
    await dp.start_polling(bot)

if __name__ == "__main__":
    asyncio.run(main())

```

Link to Telegram Bot:
t.me/DataScienceSurvivalBot

Git repository link:
https://github.com/Wolff11912/dsss_homework9.git



Name: Constantin Wolff
Mat. Num.: 22442020
IdM: lu11synu