

Task 1:

```
from telegram import Update
from telegram.ext import ApplicationBuilder, CommandHandler, MessageHandler, ContextTypes, filters
from llm_handler import generate_response

# Function to handle the /start command
async def start(update: Update, context: ContextTypes.DEFAULT_TYPE):
    await context.bot.send_message(chat_id=update.effective_chat.id, text="Hello! I am a bot that can help you with your daily tasks.")

async def process(update: Update, context: ContextTypes.DEFAULT_TYPE) -> None:
    """Process the user message."""
    user_message = update.message.text # Get the user's message
    print(f"Received message: {user_message}") # Log the message

    llm_response = generate_response(user_message)

    # Clean the response
    assistant_tag = "<|assistant>"
    if assistant_tag in llm_response:
        llm_response = llm_response.split(assistant_tag, 1)[1].strip()

    await context.bot.send_message(chat_id=update.effective_chat.id, text=llm_response)

def main() -> None:
    print("Hello")
    API_TOKEN = "7811652764:AAHsGZovSBazla9io-cIqZdikgGygabc1b4"

    application = ApplicationBuilder().token(API_TOKEN).build()
    print("Application created")

    application.add_handler(CommandHandler("start", start))
    application.add_handler(MessageHandler(filters.TEXT & (~filters.COMMAND), process))
    print("Handlers added")

    print("Running polling")
    application.run_polling()

if __name__ == "__main__":
    print("Running main now")
    main()
```

Github link : <https://github.com/rohitpotdukhe01/HW9>

Bot link : t.me/ropo_ai_bot

Rohit Potdukhe, M. N.: 22985091, IdM: yx49uxym

Task 2:

```
import torch
from transformers import pipeline

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

def generate_response(user_message: str) -> str:
    """Generate a response to the user's message."""
    messages = [
        {
            "role": "user",
            "content": "You are a friendly chatbot who always responds in a fun and informative way",
        },
        {"role": "user", "content": user_message},
    ]

    # Apply the chat template
    prompt = pipe.tokenizer.apply_chat_template(messages, tokenize=False, add_generation_prompt=True)
    # Generate the response
    outputs = pipe(prompt, max_new_tokens=256, do_sample=True, temperature=0.7, top_k=50, top_p=0.95)
    # Extract and return the generated text
    generated_text = outputs[0]["generated_text"]

    #return response
    return generated_text
```

