با سلام

پاسخ به تسک های ارسال شده :

1) به زبان ساده مدل های هوش مصنوعی (AI) مانند یک مغز هستند که فقط وقتی از آنها سوال بپرسیم جواب میدهند و برای انجام کارهایی مانند تولید متن یا ترجمه یک متن و پاسخ به سوالات و یا ... استفاده میشوند.
اما AI agent با استفاده از مدل های هوش مصنوعی کاری مانند آن را انجام میدهند با این تفاوت که دارای قابلیت تصمیم گیری و درک هستند و میتواند چند مرحله را خودش انجام دهد .

مثال فرق بین چت بات ساده با دستیار هوشمند:
برای مثال میخواهیم فروش سالیانه را از مجموعه ای از داده ها به دست آوریم وقتی این کار را با استفاده از یک چت بات ساده انجام میدهیم نمیتواند به طور خودکار و مستقل برای ما داده ها را بخواند یا استخراج کند و مبلغ فروش سالیانه را داخل فایلی ساختار یافته مانند یک فایل اکسل ذخیره کند.
اما یک دستیار هوشمند میتواند این کار را به خوبی انجام دهد .

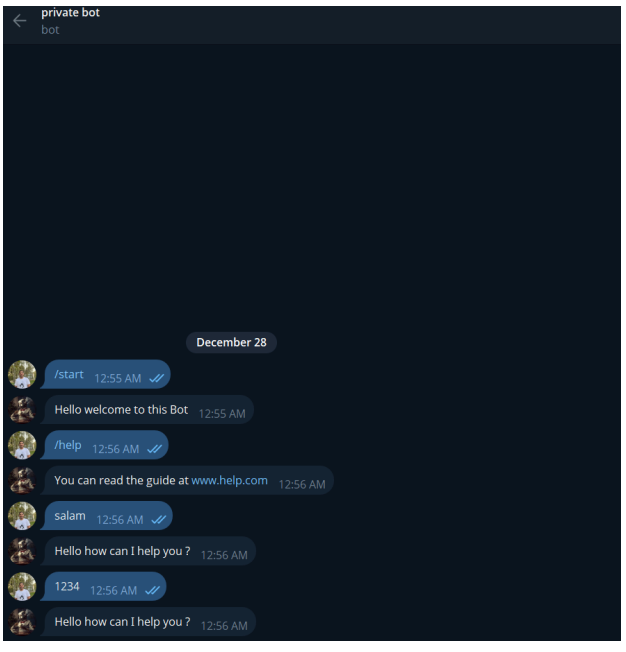
2) معماری یک دستیار هوشمند برای کلینیک پزشکی:
   a) سیستم نوبت دهی به بیماران
   b) سیستم پاسخ هوشمند به سوالات متداول
   c) ماژول یاد آوری نوبت بیماران
   d) سیستم مدیریت پرونده بیماران
   e) سیستم API

لینک گیت هاب برای تسک های عملی:

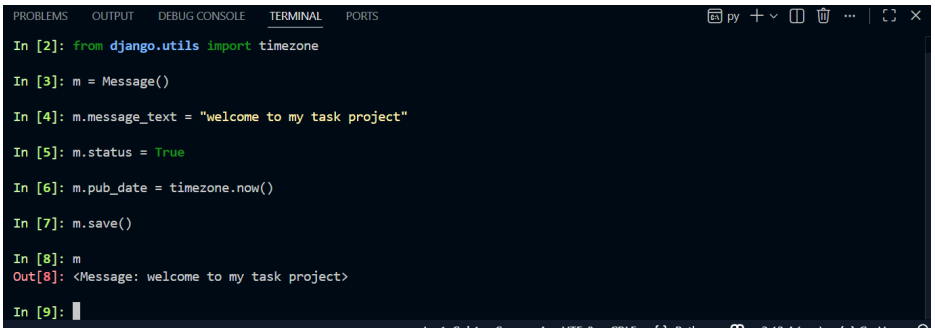
https://github.com/aradsahaviyeh/my_tasks.git

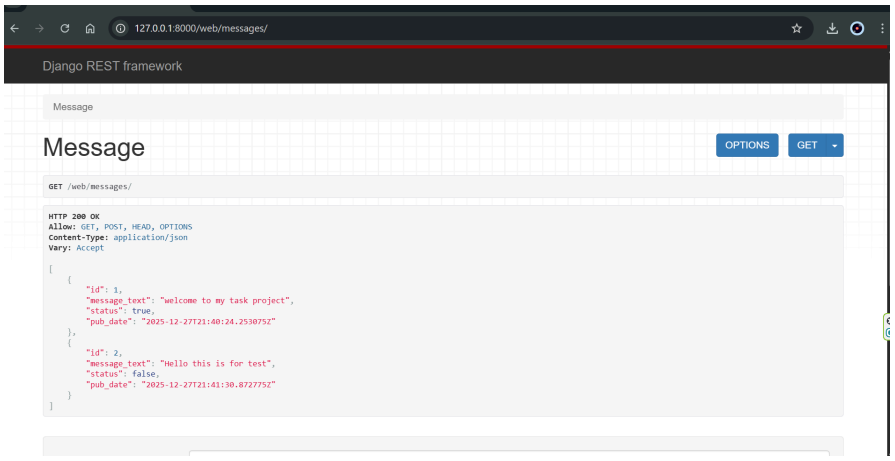
ربات تلگرامی

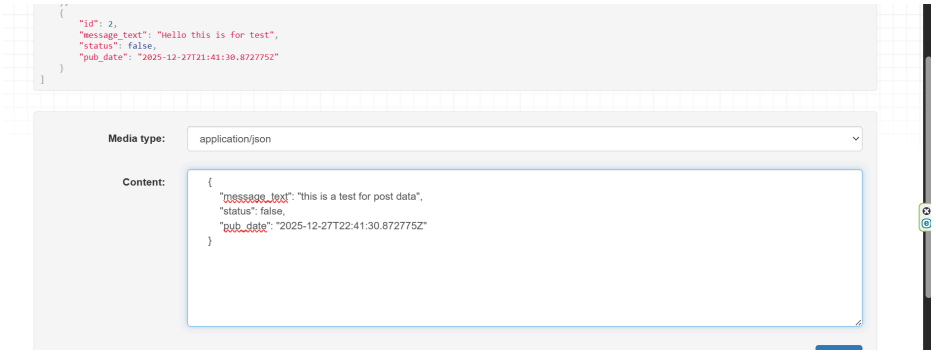
```python
from telegram import Update
from telegram.ext import (
    ApplicationBuilder,
    ContextTypes,
    MessageHandler,
    CommandHandler,
    filters,
)
from os import environ
from dotenv import load_dotenv

load_dotenv()
_token = environ["TOKEN"]


# for start command
async def start_command(update: Update, context: ContextTypes.DEFAULT_TYPE):
    await update.message.reply_text("Hello welcome to this Bot")


# for help command
async def help_command(update: Update, context: ContextTypes.DEFAULT_TYPE):
    await update.message.reply_text("You can read the guide at www.help.com")


# for any text
async def static_message(update: Update, context: ContextTypes.DEFAULT_TYPE):
    await update.message.reply_text("Hello how can I help you ?")


if __name__ == "__main__":
    app = ApplicationBuilder().token(_token).build()

    # commands
    app.add_handler(CommandHandler("start", start_command))
    app.add_handler(CommandHandler("help", help_command))

    # messages
    app.add_handler(MessageHandler(filters.TEXT & (~filters.COMMAND), static_message))
```

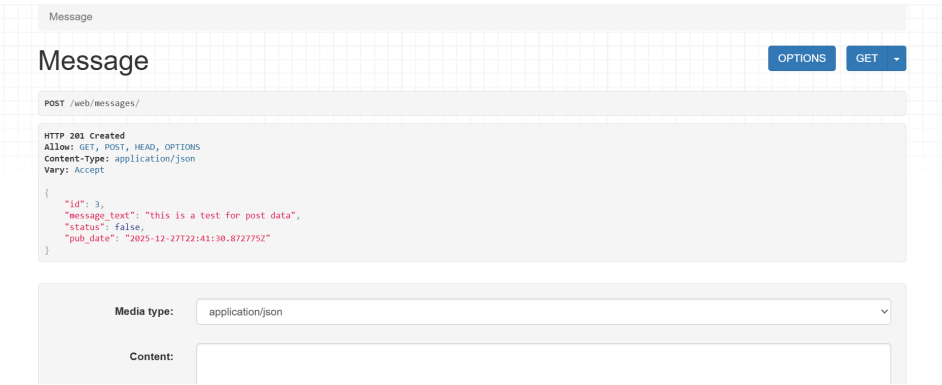


ساخت API با استفاده از django rest framework









تنظیمات فنی ربات تلگرامی:
برای ساخت ربات تلگرامی، ابتدا با استفاده از BotFather یک ربات ایجاد و توکن مربوطه دریافت شد. جهت افزایش امنیت، توکن در فایل .env ذخیره گردید.
سپس یک virtual environment ایجاد و کتابخانه‌های python-telegram-bot و python-dotenv نصب شدند. در ادامه، توابع لازم برای پاسخ به پیام‌ها و درخواست‌های کاربران پیاده‌سازی شد.

تنظیمات فنی DRF:
برای پیاده‌سازی API با django rest framework، ابتدا یک virtual environment برای ایزوله‌سازی پروژه ایجاد و فعال شد. سپس django نصب و پروژه با دستور django-admin startproject ساخته شد. بعد از آن یک اپلیکیشن با نام web ایجاد گردید.
django rest framework نصب و در فایل settings.py به لیست اپلیکیشن‌ها اضافه شد. مسیرهای مربوط به اپلیکیشن در فایل urls.py پروژه و اپ مدیریت شدند.
برای ساخت APIها، فایل serializers.py ایجاد و serializer های موردنیاز پیاده‌سازی شد و در فایل views.py مورد استفاده قرار گرفت. با استفاده از متدهای GET و POST امکان دریافت داده‌ها به صورت JSON و ارسال اطلاعات معتبر به دیتابیس فراهم شد.

با تشکر
آراد ساهویه