

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
بوت تيليغرام للبحث في الأنظمة والقوانين السعودية
Saudi Legal Research Telegram Bot
"""

import logging
import requests
import json
from telegram import Update
from telegram.ext import Application, CommandHandler, MessageHandler, filters, Co

# ===== إعدادات المفاتيح =====
TELEGRAM_TOKEN = "8048665417:AAEUWd5RCFZ3hEgOcHjnr4MRyKyvANf4-qs"
GEMINI_API_KEY = "AIzaSyA7hRs1veViDmfDqEkMZLkODp774jD6ZUE"
GEMINI_URL = f"https://generativelanguage.googleapis.com/v1beta/models/gemini-1.5

# ===== إعداد السجل =====
logging.basicConfig(
    format='%(asctime)s - %(name)s - %(levelname)s - %(message)s',
    level=logging.INFO
)
logger = logging.getLogger(__name__)

# ===== دالة الاتصال بـ Gemini =====
def ask_gemini(user_question: str) -> str:
    """والحصول على تحليل قانوني Gemini AI إرسال السؤال إلى"""

    system_prompt = """مستشار قانوني متخصص في الأنظمة والقوانين واللوائح السعودية
مهمتك هي:
1. البحث في الأنظمة السعودية ذات الصلة بسؤال المستخدم
2. ذكر النص القانوني المحدد من النظام الرسمي
3. تحديد الجهة المختصة (وزارة العدل، هيئة الخبراء، إلخ)
4. تحليل ما إذا كان وضع المستخدم يتوافق مع النظام
5. تقديم توصية واضحة

:التنسيق المطلوب في ردك
_____
📄 **:الأنظمة ذات الصلة:**
[اذكر اسم النظام ورقم المادة]

📖 **:النص القانوني:**
[اذكر النص الحرفي أو ما يقاربه من النظام]
```



**\*\*الجهة المختصة\*\***

[اسم الجهة الحكومية]



**\*\*التحليل\*\***

[هل الوضع يتوافق مع النظام أم لا؟ ولماذا؟]



**\*\*التوصية\*\***

[ماذا يجب على المستخدم أن يفعل؟]



**\*\*تنبيه:\*\*** هذا تحليل استرشادي، يُنصح باستشارة محامٍ متخصص للحالات المعقدة.

:الأنظمة التي تغطيها

- نظام الشركات
  - نظام العمل والعمال
  - نظام التجارة الإلكترونية
  - نظام حماية المستهلك
  - نظام مكافحة الجرائم المعلوماتية
  - نظام الإقامة
  - نظام الأحوال الشخصية
  - نظام الملكية الفكرية
  - نظام البناء والتشييد
  - نظام الاستثمار الأجنبي
  - نظام ضريبة القيمة المضافة
  - نظام الزكاة والدخل
  - أنظمة هيئة السوق المالية
  - نظام مكافحة الغش التجاري
  - نظام التوثيق
- ""وجميع الأنظمة السعودية الأخرى

```
payload = {
  "contents": [
    {
      "parts": [
        {
          "text": f"{system_prompt}\n\nسؤال المستخدم: {user_questio
```

```

try:
    response = requests.post(GEMINI_URL, json=payload, timeout=30)
    response.raise_for_status()
    data = response.json()

    if "candidates" in data and len(data["candidates"]) > 0:
        return data["candidates"][0]["content"]["parts"][0]["text"]
    else:
        return "❌ لم أتمكن من الحصول على إجابة. حاول مرة أخرى."

except requests.exceptions.Timeout:
    return "⌚ انتهت مهلة الاتصال. حاول مرة أخرى."
except requests.exceptions.RequestException as e:
    logger.error(f"Gemini API error: {e}")
    return "❌ حدث خطأ في الاتصال. حاول مرة أخرى."

# ===== أوامر البوت =====
async def start(update: Update, context: ContextTypes.DEFAULT_TYPE):
    """رسالة الترحيب"""
    welcome_message = """
    🏛️ **مرحباً بك في بوت الأنظمة القانونية السعودية**

    : أنا مساعدك القانوني الذكي! يمكنني مساعدتك في

    🔍 **البحث في:**
    • أنظمة العمل والتوظيف
    • أنظمة الشركات والتجارة
    • أنظمة الإقامة والجنسية
    • أنظمة الملكية الفكرية
    • أنظمة التجارة الإلكترونية
    • حماية المستهلك
    • الأنظمة الجزائية
    • وجميع الأنظمة السعودية


    📝 **كيف تستخدمني؟**
    عك أو سؤالك بالتفصيل وسأبحث لك في الأنظمة وأخبرك هل يتوافق مع القانون السعودي أم لا

    **مثال:**
    "أريد فتح مطعم في الرياض، ما هي الاشتراطات القانونية المطلوبة؟"

    ⚠️ هذا البوت للأغراض البحثية والاسترشادية فقط.
    """
    await update.message.reply_text(welcome_message, parse_mode='Markdown')

async def help_command(update: Update, context: ContextTypes.DEFAULT_TYPE):
    """أمر المساعدة"""
    help_text = """

```

 **\*\* دليل الاستخدام \*\***

**\*\* الأوامر المتاحة \*\***

/start - بدء البوت  
/help - عرض المساعدة  
/about - معلومات عن البوت

**\*\* طريقة الاستخدام \*\***

: اكتب سؤالك أو وضعك بوضوح مثل


- ✓ "هل يجوز فصل موظف بدون سبب؟"
- ✓ "ما هي حقوق المستهلك عند شراء منتج معيب؟"
- ✓ "ما هي اشتراطات تأسيس شركة ذات مسؤولية محدودة؟"
- ✓ "هل العمل عن بُعد مع شركة أجنبية مسموح به؟"


```
await update.message.reply_text(help_text, parse_mode='Markdown')
```


```
async def about_command(update: Update, context: ContextTypes.DEFAULT_TYPE):
```

```
    """معلومات عن البوت"""
```

```
    about_text = """
```


 **\*\* عن البوت \*\***

 بوت البحث في الأنظمة القانونية السعودية

 يستند إلى الأنظمة الرسمية الصادرة عن:

- هيئة الخبراء بمجلس الوزراء
- وزارة العدل
- الجريدة الرسمية (أم القرى)
- الجهات الحكومية المختصة

 ⚡ Gemini AI من Google مدعوم بتقنية

 ⚠ **\*\* تنبيه مهم \*\***

. هذا البوت للأغراض البحثية والاسترشادية فقط.  
للحالات القانونية المعقدة يُنصح باستشارة محام مرخص

```
    """
```

```
    await update.message.reply_text(about_text, parse_mode='Markdown')
```

```
async def handle_message(update: Update, context: ContextTypes.DEFAULT_TYPE):
```

```
    """معالجة رسائل المستخدم"""
```

```
    user_message = update.message.text
```

```
    user_name = update.message.from_user.first_name
```

```
    # رسالة انتظار
```

```
    waiting_msg = await update.message.reply_text(
```

```
        f"⌚ جاري البحث في الأنظمة السعودية لسؤالك يا {user_name}...\n
```

```
)
```

```

logger.info(f"User {user_name} asked: {user_message}")

# الحصول على الإجابة من Gemini
response = ask_gemini(user_message)

# حذف رسالة الانتظار
await waiting_msg.delete()

# إرسال الإجابة
# تقسيم الرسالة إذا كانت طويلة (تليغرام يسمح بـ 4096 حرف فقط)
if len(response) > 4000:
    parts = [response[i:i+4000] for i in range(0, len(response), 4000)]
    for part in parts:
        await update.message.reply_text(part, parse_mode='Markdown')
else:
    await update.message.reply_text(response, parse_mode='Markdown')

async def error_handler(update: Update, context: ContextTypes.DEFAULT_TYPE):
    """معالجة الأخطاء"""
    logger.error(f"Error: {context.error}")
    if update and update.message:
        await update.message.reply_text("حدث خطأ غير متوقع. حاول مرة أخرى ❌")

# ===== تشغيل البوت =====
def main():
    """الدالة الرئيسية لتشغيل البوت"""
    print("🚀 جاري تشغيل بوت الأنظمة القانونية السعودية")

    # إنشاء التطبيق
    application = Application.builder().token(TELEGRAM_TOKEN).build()

    # إضافة معالجات الأوامر
    application.add_handler(CommandHandler("start", start))
    application.add_handler(CommandHandler("help", help_command))
    application.add_handler(CommandHandler("about", about_command))

    # معالج الرسائل النصية
    application.add_handler(MessageHandler(filters.TEXT & ~filters.COMMAND, handl

    # معالج الأخطاء
    application.add_error_handler(error_handler)

    print("✅ للإيقاف Ctrl+C البوت يعمل الآن! اضغط")

    # تشغيل البوت
    application.run_polling(allowed_updates=Update.ALL_TYPES)

```

```
if __name__ == '__main__':  
    main()
```