

```
# =====
# 🛡️ مكافحة العدوى الذكي - الإصدار المتكامل
# 💰👤 تطوير: الصيدلانية حنين علي عثمان
# 🛡️ مستشفى السواني بن آدم - قسم مكافحة العدوى
# =====
```

```
import os

import logging

import json

from flask import Flask, request

from telegram import Update, ReplyKeyboardMarkup, KeyboardButton

from telegram.ext import Application, CommandHandler, MessageHandler, filters,
ContextTypes, CallbackContext
```

```
# ===== التهيئة الأساسية =====

logging.basicConfig(
    format='%(asctime)s - %(name)s - %(levelname)s - %(message)s',
    level=logging.INFO
)

logger = logging.getLogger(__name__)
```

```
app = Flask(__name__)

TOKEN = os.environ.get('TELEGRAM_TOKEN')
```

```
# ===== قاعدة المعرفة الطبية الشاملة =====

MEDICAL_KNOWLEDGE = {
    "bacteria": {
        "mrsa": {
            "name_ar": "المكورات العنقودية الذهبية المقاومة (MRSA)",
            "name_en": "Methicillin-Resistant Staphylococcus aureus",
            "type": "بكتيريا موجبة الجرام",

```

```

"symptoms": ["التهابات جلدية", "خراجات", "تسمم الدم", "التهاب رئوي", "حمى", "قشعريرة"],
"diagnostic_tests": ["اختبار الحساسية للمضادات", "صورة دم كاملة", "PCR", "مزرعة دم"],
"antibiotics": {
  "first_line": ["Vancomycin", "Linezolid"],
  "alternatives": ["Daptomycin", "Ceftaroline", "Clindamycin"],
  "dosing": "فانكوميسين 15-20 مجم/كجم كل 8-12 ساعة",
  "monitoring": "ميكروجرام/مل 15-20مستوى الت"
},
"sterilization": {
  "disinfectant": "MRSA ضد EPA مطهرات معتمدة من",
  "concentration": "حسب تعليمات المصنع",
  "contact_time": "دقائق 5-10",
  "dilution": "اتباع تعليمات المنتج"
},
"isolation": "Contact Precautions (احتياطات تلامسية)",
"ppe": "واقى وجه إذا خطر رذاذ gownقفازات",
"notes": "بعد التعامل مع المريض essential غسل اليدين",
},

"c_diff": {
  "name_ar": "المطثية العسيرة (C. diff)",
  "name_en": "Clostridium difficile",
  "type": "بكتيريا موجبة الجرام - مكونة للأبواغ",
  "symptoms": ["إسهال مائي", "ألم بطني", "حمى", "غثيان", "فقدان شهية", "جفاف"],
  "diagnostic_tests": ["مزرعة براز", "GDH EIA", "PCR", "في البراز C. diff سموم"],
  "antibiotics": {
    "first_line": ["Metronidazole", "Vancomycin PO"],
    "alternatives": ["Fidaxomicin"],
    "dosing": "ميترونيدازول 500 مجم 3 مرات/يوم لمدة 10-14 يوم",
    "monitoring": "تحسين الأعراض، تكرار البراز"
  },
},

```

```

"sterilization": {
  "disinfectant": "مبيض مخفف (هيبوكلوريت الصوديوم)",
  "concentration": "جزء في المليون 1000-5000",
  "contact_time": "دقائق 10",
  "dilution": "مبيض إلى ماء (لـ 1000 جزء في المليون) 1:10",
},
"isolation": "Contact Precautions (احتياطات تلامسية صارمة)",
"ppe": "غسل يدين بالصابون والماء، غطاءات،",
"notes": "الكحول غير فعال ضد الأبواغ - استخدم الصابون والماء فقط",
},

"vre": {
  "name_ar": "المكورة المعوية المقاومة للفاونكومايسين (VRE)",
  "name_en": "Vancomycin-Resistant Enterococcus",
  "type": "بكتيريا موجبة الجرام",
  "symptoms": "[تسمم الدم، التهابات مسالك بولية، التهابات جروح، التهاب شغاف القلب]",
  "diagnostic_tests": "[مزرعة دم، مزرعة بول، اختبار الحساسية]",
  "antibiotics": {
    "first_line": ["Linezolid", "Daptomycin"],
    "alternatives": ["Quinupristin/Dalfopristin", "Tigecycline"],
    "dosing": "الينزوليد 600 مجم كل 12 ساعة",
    "monitoring": "عدد كريات الدم، وظائف الكبد",
  },
},
"sterilization": {
  "disinfectant": "EPA مطهرات معتمدة من",
  "concentration": "حسب تعليمات المصنع",
  "contact_time": "دقائق 10",
},
"isolation": "Contact Precautions",
"ppe": "غطاءات، غطاءات",
}

```

},

"viruses": {

"norovirus": {

"name_ar": "النوروفيروس",

"name_en": "Norovirus",

"type": "فيروس غير مغلف",

"symptoms": ["إسهال مائي", "قيء", "غثيان", "ألم بطني", "حمى خفيفة", "آلام عضلية"],

"diagnostic_tests": ["PCR", "فحص البراز بالإلكتروني", "EIA"],

"treatment": "رعاية داعمة، ترطيب، محاليل فموية أو وريدية",

"sterilization": {

"disinfectant": "مبيض مخفف",

"concentration": "جزء في المليون 1000-5000",

"contact_time": "دقيقة 10-20",

"dilution": "إلى 1:50 مبيض إلى ماء 1:10"

},

"isolation": "نظافة صارمة + Contact Precautions",

"ppe": "كمامة إذا خطر رذاذ gownقفازات",

"notes": "فيروس شديد العدوى، ينظف الأسطح جيداً"

},

"influenza": {

"name_ar": "فيروس الإنفلونزا",

"name_en": "Influenza Virus",

"type": "فيروس مغلف",

"symptoms": ["حمى مفاجئة", "سعال", "ألم حلق", "سيلان أنف", "آلام عضلية", "صداع", "تعب"],

"diagnostic_tests": ["اختبار سريع للإنفلونزا", "مزرعة فيروسية", "PCR"],

"treatment": "رعاية داعمة Oseltamivir، Zanamivir",

"sterilization": {

"disinfectant": "كحول 70% أو مطهرات معتمدة",

"concentration": "كحول 70%",

```

        "contact_time": "ثانية - 2 دقيقة 30",
    },
    "isolation": "Droplet Precautions (احتياطات قطرات)",
    "ppe": "كمامة جراحية، نظارات واقية إذا خطر رذاذ",
    "notes": "التطعيم السنوي للوقاية"
}
}
}

```

DISINFECTANT_DATABASE = {

```

    "sodium_hypochlorite": {
        "name_ar": "هيبوكلوريت الصوديوم (مبيض)",
        "dilution_guides": {
            "1000_ppm": "مل مبيض + 980 مل ماء 20",
            "5000_ppm": "مل مبيض + 900 مل ماء 100",
            "10000_ppm": "مل مبيض + 800 مل ماء (لتنظيف الدم) 200"
        },
        "contact_time": "دقيقة 10-30",
        "effective_against": ["C. diff", "Norovirus", "MRSA", "VRE", "السل"],
        "limitations": "يسبب تآكل المعادن، رائحة قوية، يحتاج تهوية"
    },

```

```

    "alcohol_70": {
        "name_ar": "%كحول إيثيلي 70",
        "preparation": "%مل كحول 95 + 270 مل ماء = 1 لتر 70 730",
        "contact_time": "ثانية - 2 دقيقة 30",
        "effective_against": ["معظم البكتيريا"، "الفطريات", "VRE", "MRSA", "فيروس الإنفلونزا"],
        "limitations": "غير فعال ضد الأبواغ وبعض الفيروسات غير المغلفة"
    }
}

```

```
# ===== نظام الذكاء الاصطناعي المحسن =====
```

```
def intelligent_medical_assistant(user_message):
```

```
    """مساعد طبي ذكي يفهم مختلف صيغ الأسئلة"""
```

```
    message = user_message.lower().strip()
```

```
    # تنظيف النص من التشكيل والكلمات الشائعة
```

```
    message = clean_arabic_text(message)
```

```
    # البحث الذكي في جميع القواعد
```

```
    response = generate_comprehensive_response(message)
```

```
    return response
```

```
def clean_arabic_text(text):
```

```
    """تنظيف النص العربي من التشكيل والكلمات الشائعة"""
```

```
    # إزالة التشكيل
```

```
    tashkeel = [',', '،', '،', '،', '،', '،', '،']
```

```
    for char in tashkeel:
```

```
        text = text.replace(char, "")
```

```
    # إزالة علامات الاستفهام والنقاط
```

```
    punctuation = ['?', '!', '!', '!', '!', '!', '!']
```

```
    for char in punctuation:
```

```
        text = text.replace(char, "")
```

```
    return text
```

```
def generate_comprehensive_response(message):
```

```
    """توليد رد شامل بناء على تحليل النص"""
```

```
    # كلمات مفتاحية موسعة مع مرادفات
```

```
keyword_patterns = {  
    # MRSA ومشتقاته  
    'mrsa': 'mrsa',  
    'مكورات عنقودية': 'mrsa',  
    'عنقودية ذهبية': 'mrsa',  
    'جراثيم عنقودية': 'mrsa',  
    'staph': 'mrsa',  
    'staphylococcus': 'mrsa',
```

```
    # C. diff ومشتقاته  
    'مطثية عسيرة': 'c_diff',  
    'c.diff': 'c_diff',  
    'c diff': 'c_diff',  
    'كلوستريديوم': 'c_diff',  
    'clostridium': 'c_diff',  
    'اسهال بعد مضاد': 'c_diff',  
    'اسهال مائي': 'c_diff',
```

```
    # VRE  
    'vre': 'vre',  
    'مكورة معوية': 'vre',  
    'انتروكوكس': 'vre',  
    'enterococcus': 'vre',
```

```
    # النوروفيروس  
    'norovirus': 'norovirus',  
    'نورو': 'norovirus',  
    'norovirus': 'norovirus',  
    'قيء واسهال': 'norovirus',  
    'ترجيع واسهال': 'norovirus',  
    'عدوى قلبية': 'norovirus',
```

الإنفلونزا

'influenza': 'إنفلونزا',
'influenza': 'نزلة برد',
'influenza': 'influenza',
'flu': 'influenza',
'influenza': 'حمى وسعال',

تعقيم وتطهير

'sterilization': 'تعقيم',
'sterilization': 'تطهير',
'sterilization': 'مطهر',
'sterilization': 'تنظيف',
'sterilization': 'غسل',
'sterilization': 'disinfect',
'sterilization': 'steril',

علاجات

'treatment': 'علاج',
'treatment': 'دواء',
'treatment': 'مضاد',
'treatment': 'antibiotic',
'treatment': 'ادوية',

جرعات

'dosing': 'جرعة',
'dosing': 'dosage',
'dosing': 'حاسبة',
'dosing': 'حساب',
'dosing': 'كم جرعة',

أعراض

'اعراض': 'symptoms',
'أعراض': 'symptoms',
'شكوى': 'symptoms',
'علامات': 'symptoms',
'symptoms': 'symptoms',
'عرض': 'symptoms',

تحاليل

'تحليل': 'tests',
'فحص': 'tests',
'مختبر': 'tests',
'تحاليل': 'tests',
'lab': 'tests',
'test': 'tests',

عزل

'عزل': 'isolation',
'عزلة': 'isolation',
'isolation': 'isolation',
'احتياطات': 'isolation',

وقاية

'وقاية': 'ppe',
'وقائي': 'ppe',
'ppe': 'ppe',
'معدات وقاية': 'ppe',
'كمادات': 'ppe',
'قفازات': 'ppe'

}

```

# البحث عن الأنماط
found_patterns = []
for pattern, pattern_type in keyword_patterns.items():
    if pattern in message:
        found_patterns.append(pattern_type)

# توليد الرد المناسب
if not found_patterns:
    return get_intelligent_help_response(message)

return create_detailed_response(found_patterns, message)

def create_detailed_response(patterns, original_message):
    """إنشاء رد مفصل بناء على الأنماط المكتشفة"""

    # تحديد النمط الأساسي
    primary_pattern = patterns[0]

    # البحث في قاعدة البيانات
    for category, pathogens in MEDICAL_KNOWLEDGE.items():
        for pathogen_id, data in pathogens.items():
            if pathogen_id == primary_pattern:
                return format_pathogen_response(data, patterns, original_message)

    # إذا كان طلباً عاماً
    if primary_pattern == 'sterilization':
        return get_comprehensive_sterilization_guide()
    elif primary_pattern == 'treatment':
        return get_treatment_overview()
    elif primary_pattern == 'symptoms':
        return get_symptoms_guide()

```

```

elif primary_pattern == 'dosing':

    return get_dosing_calculator_guide()

return get_intelligent_help_response(original_message)

def format_pathogen_response(pathogen_data, patterns, original_message):
    """تنسيق رد مفصل عن الممرض"""

    response = f"🦠 **{pathogen_data['name_ar']}**\n"
    response += f"*({pathogen_data['name_en']})*\n\n"

    # إضافة المعلومات حسب الأنماط المكتشفة

    if any(p in ['symptoms', 'treatment', 'tests'] for p in patterns) or any(word in
original_message for word in ['تحليل', 'علاج', 'اعراض']):

        response += f"📋 **الأعراض:** {' '.join(pathogen_data['symptoms'])}\n\n"
        response += f"📋 **التحاليل:** {' '.join(pathogen_data['diagnostic_tests'])}\n\n"

    if 'treatment' in patterns or 'antibiotic' in original_message:

        if 'antibiotics' in pathogen_data:

            abx = pathogen_data['antibiotics']

            response += f"💊 **العلاج:** {' '.join(abx['first_line'])}\n"

            if 'dosing' in abx:

                response += f"💊 **الجرعة:** {abx['dosing']}\n"

            if 'monitoring' in abx:

                response += f"🔍 **المتابعة:** {abx['monitoring']}\n"

            response += "\n"

    if 'sterilization' in patterns or any(word in original_message for word in ['تطهير', 'تعقيم', 'مطهر']):

        ster = pathogen_data['sterilization']

        response += f"🧼 **التعقيم:** {ster['disinfectant']}\n"
        response += f"🕒 **وقت التلامس:** {ster['contact_time']}\n"

```

if 'dilution' in ster:

response += f"🧪 **طريقة التحضير:** {ster['dilution']}\n"

response += "\n"

response += f"⚠️ **إجراءات العزل:** {pathogen_data['isolation']}\n\n"

response += f"🧤 **معدات الوقاية:** {pathogen_data['ppe']}\n\n"

if 'notes' in pathogen_data:

response += f"💡 **ملاحظات مهمة:** {pathogen_data['notes']}\n\n"

return response

def get_comprehensive_sterilization_guide():

"""دليل التعقيم الشامل"""

guide = """📖 **الدليل الشامل للتعقيم والتطهير**

● **المطثية العسيرة (C. diff):**

- المطهر: مبيض مخفف
- التركيز: 1000-5000 جزء في المليون
- الوقت: 10 دقائق
- التحضير: 1:10 مبيض إلى ماء
- الكحول غير فعال ⚠️

● **المكورات العنقودية (MRSA):**

- EPA المطهر: مطهرات معتمدة من
- الوقت: 5-10 دقائق
- %البديل: كحول 70

🦠 **النوروفيروس:**

- المطهر: مبيض مخفف

- التركيز: 1000-5000 جزء في المليون
- الوقت: 10-20 دقيقة
- التنظيف الشامل للأسضر

❑ **السل**

- المطهر: مطهرات معتمدة للسل
- الوقت: 10+ دقائق
- التهوية المناسبة

○ **للأجهزة الطبية**

- الأجهزة الحرجة: تعقيم البخار
- الأجهزة شبه الحرجة: تطهير عالي المستوى
- ""الأجهزة غير الحرجة: تنظيف وتطهير

return guide

def get_treatment_overview():

 ""نظرة عامة على العلاجات""

 return ""📌 **نظرة عامة على العلاجات**""

❑ **المطثية العسيرة**

- الخط الأول: ميترونيدازول 500 مجم 3 مرات/يوم
- الحالات الشديدة: فانكومايسين فموي 125 مجم 4 مرات/يوم
- المدة: 10-14 يوم

❑ **MRSA) المكورات العنقودية**

- الخط الأول: فانكومايسين 15-20 مجم/كجم كل 12 ساعة
- البديل: لينزوليد 600 مجم كل 12 ساعة
- ميكروجرام/مل 15-20 rough المتابعة: مستوى الت

❑ **VRE) المكورة المعوية**

- الخط الأول: لينزوليد 600 مجم كل 12 ساعة
- البديل: دابتوميسين 6-8 مجم/كجم يومياً

****النوروفيروس**** ?

- العلاج: رعاية داعمة، ترطيب
- الوقاية: نظافة صارمة، تعقيم

*****راجعى الطبيب للجرعات الدقيقة والمراقبة** □ \$

def get_symptoms_guide():

"""دليل الأعراض التفاضلي"""

return """Q **دليل الأعراض التفاضلي**

◆ ****إسهال مائي + حمى****

- المطثية العسيرة (C. diff)
- النوروفيروس
- بكتيريا أخرى مسببة للإسهال

****التهابات جلدية + خراجات**** ?

- المكورات العنقودية (MRSA)
- بكتيريا أخرى

****قيء + إسهال**** ?

- النوروفيروس
- تسمم غذائي
- فيروسات أخرى

☹ ****سعال مزمن + نقص وزن****

- السل
- التهابات رئوية أخرى

****تسمم الدم + حمى**** 📋

- المكورات العنقودية
- المكورة المعوية
- بكتيريا أخرى

🔗 *******التحاليل المخبرية تؤكد التشخيص****

def get_dosing_calculator_guide():

 "دليل حاسبة الجرعات"

 return *******حاسبة الجرعات الذكية****

لحساب الجرعات الدقيقة، أرسلني:

- اسم الدواء
- وزن المريض (كجم)
- مستوى الكرياتينين في الدم
- العمر

****الأدوية المتاحة****

- فانكومايسين
- جنتاميسين
- ميرروبينيم

****مثال****

"احسب جرعة فانكومايسين لوزن 70 كجم، كرياتينين 1.2، عمر 45"

"/calculate_vancomycin" أو استخدم الأمر

def get_intelligent_help_response(user_message):

 "رد مساعدة ذكي يتكيف مع السؤال"

help_responses = {

'default': ""

****لم أفهم سؤالك بشكل كامل**** ?

****لحصول على أفضل إجابة، جربي هذه الصيغ**** 📝

****:للأعراض**** 🔍

- "مريض عنده إسهال مائي وحرارة"
- "أعراض التهاب الجلد والخراجات"
- "شكوى سعال مزمن مع نقص وزن"

****:للميكروبات**** ?

- "معلومات عن المكورات العنقودية"
- "C. diff كيف أتعامل مع مريض"
- "بروتوكول النوروفيروس"

****:للعلاجات**** 💊

- "علاج المطثية العسيرة"
- "بدائل الفانكوميسين"
- "جرعة الميترونيدازول"

****:للتعقيم**** ?

- "C. diff كيف أعقم غرفة مريض"
- "أفضل مطهر للجراثيم"
- "طريقة تخفيف المبيض"

****:لحساب الجرعات**** 📊

- "احسب جرعة فانكوميسين"

****أو استخدمي الأزرار في الأسفل للخيارات السريعة**** ⌨

"" ,

'short': "...اكتبي اسم الميكروب أو الأعراض للحصول على المعلومات Q",

'creator': "👤\$□ باستخدام المطورة باحثي عن معلومات /creator"
}

اختيار الرد المناسب بناء على طول الرسالة

if len(user_message) < 5:

return help_responses['short']

elif any(word in user_message for word in ['مطور', 'حنين', 'creator', 'صانعة']):

return help_responses['creator']

else:

return help_responses['default']

===== حاسبة الجرعات المتقدمة =====

class AdvancedDoseCalculator:

@staticmethod

def calculate_vancomycin(weight, scr, age, gender="female"):

"""حساب جرعة الفانكوميسين مع تعديل وظائف الكلى"""

try:

حساب معدل الترشيح الكبيبي

crcl = AdvancedDoseCalculator.calculate_crcl(weight, scr, age, gender)

تحديد الجرعة والفاصل الزمني

if crcl > 50:

dose = weight * 15

interval = "كل 12 ساعة"

frequency = "Q12H"

elif crcl >= 30:

dose = weight * 15

interval = "كل 24 ساعة"

frequency = "Q24H"

else:

dose = weight * 10

interval = "كل 48 ساعة"

frequency = "Q48H"

return {

"dose": round(dose),

"interval": interval,

"frequency": frequency,

"crcl": round(crcl, 1),

"target_trough": "15-20 مل/ميكروجرام",

"monitoring": "فحص مستوى الفانكوميسين والكرياتينين بانتظام"

}

except Exception as e:

return {"error": f"خطأ في الحساب: {str(e)}"}

@staticmethod

def calculate_crcl(weight, scr, age, gender):

"""(Cockcroft-Gault) حساب معدل الترشيح الكبيبي"""

factor = 0.85 if gender == "female" else 1.0

return ((140 - age) * weight) / (scr * 72) * factor

===== واجهة المستخدم التفاعلية =====

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

"""بدء المحادثة مع واجهة تفاعلية محسنة"""

keyboard = [

[

KeyboardButton("أعراض إسهال Q"),

KeyboardButton("مكورات عنقودية Q")

],

```
[
    KeyboardButton("🩺 علاجات شائعة"),
    KeyboardButton("📖 دليل التعقيم"),
],
[
    KeyboardButton("💊 حاسبة الجرعات"),
    KeyboardButton("🏠 إجراءات العزل"),
],
[
    KeyboardButton("👤 عن المطورة"),
    KeyboardButton("🆘 المساعدة")
]
]

reply_markup = ReplyKeyboardMarkup(keyboard, resize_keyboard=True)

welcome_message = ""

**مرحباً بك في البوت الذكي لمكافحة العدوى** 🤖

تطوير: **الصيدلانية حنين علي عثمان** 🏥
المستشفى: **السواني بن آدم - قسم مكافحة العدوى**

**كيف يمكنني مساعدتك اليوم؟** 🙋

تشخيص الأمراض المعدية** بناء على الأعراض** 🔍
معلومات مفصلة** عن الميكروبات المقاومة** 📄
اختيار المضادات الحيوية** المناسبة** 🩹
بروتوكولات التعقيم** المثلى لكل حالة** 📋
حساب الجرعات** بدقة مع تعديل وظائف الكلى** 💊
إجراءات العزل والوقاية** الشخصية** 🏠
```

💡 ****أسأليني مباشرة مثل ****

- "مريض عنده إسهال مائي بعد مضاد"
- "في الجروح MRSA كيف أتعامل مع"
- "علاج المطثية العسيرة"
- "طريقة تعقيم النوروفيروس"

💰 ****المعلومات من مصادر طبية معتمدة****

- Mandell, Douglas, and Bennett's Principles
- Sanford Guide to Antimicrobial Therapy
- CDC Infection Control Guidelines
- APIC Standards

📝 ****اختر من الأزرار أو اكتب سؤالك****

=====

await update.message.reply_text(welcome_message, reply_markup=reply_markup)

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

"""معلومات شاملة عن المطورة"""

creator_message = """

👤 ****المطورة: الصيدلانية حنين علي عثمان****

🎓 ****المؤهلات العلمية****

- تخصص الميكروبيولوجيا السريرية - (PharmD) دكتور صيدلة -
- أخصائية مكافحة عدوى معتمدة -
- باحثة في مقاومة المضادات الميكروبية -

🏥 ****الخبرة المهنية****

- أخصائية صيدلة إكلينيكية - مستشفى السواني بن آدم -
- مسؤولة برنامج مكافحة العدوى -
- مدربة معتمدة للكوادر الطبية -

مستشارة أنظمة الرعاية الصحية الذكية -

التخصصات الدقيقة

- الميكروبيولوجيا السريرية والتشخيص
- إدارة المضادات الميكروبية
- بروتوكولات التعقيم والتطهير
- أنظمة ترصد العدوى
- الجودة والسلامة الصحية

الرؤية والرسالة

تحويل الخبرة السريرية إلى حلول رقمية ذكية تمكن الكوادر"
"الطبية من تقديم رعاية أكثر أماناً وفعالية

الإنجازات

- تطوير نظام ترصد العدوى الآلي -
- %خفض معدلات العدوى المكتسبة 40 -
- تدريب 200+ من الكوادر الطبية -
- نشر أبحاث في مجلات طبية محكمة -

للاستشارات المتخصصة

يمكن التواصل عبر البوت أو زيارة مستشفى السواني بن آدم

"شعار: ** "علم يخدم الحياة، وتقنية تمكن الرعاية**"

=====

```
await update.message.reply_text(creator_message)
```

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

```
    """حاسبة جرعة الفانكوميسين"""
```

```
    try:
```

```
        if context.args and len(context.args) >= 3:
```

```
weight = float(context.args[0])
```

```
scr = float(context.args[1])
```

```
age = int(context.args[2])
```

```
result = AdvancedDoseCalculator.calculate_vancomycin(weight, scr, age)
```

```
if "error" in result:
```

```
    response = f"❌ {result['error']}"
```

```
else:
```

```
    response = f"📋 **توصيات جرعة الفانكوميسين**"
```

- **الوزن:** {weight} كجم
- **الكرياتينين:** {scr} مجم/ديسيلتر
- **العمر:** {age} سنة
- **مل/دقيقة {result['crcl']} :معدل الترشيح الكبيبي**

💡 **الجرعة الموصى بها**

- مجم {result['dose']} :الجرعة**
- {result['interval']} ({result['frequency']}) :الفاصل الزمني**
- {result['target_trough']} :الهدف العلاجي**

🔍 **نصائح المتابعة**

```
{result['monitoring']}
```

💰 **راجعى الصيدلي للتأكد من الجرعة**

```
else:
```

```
    response = f"💡 استخدام حاسبة الفانكوميسين**"
```

/calculate_vancomycin [الوزن] [الكرياتينين] [العمر]

****مثال:****

/calculate_vancomycin 70 1.2 45

****سيحسب:****

- الجرعة المناسبة -

- الفاصل الزمني -

""تعديل وظائف الكلى -

```
except (ValueError, IndexError):
```

```
    response = "❌ أدخل أرقام صحيحة \nمثال: /calculate_vancomycin 70 1.2 45"
```

```
await update.message.reply_text(response)
```

```
# ===== المعالج الرئيسي للرسائل =====
```

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

```
    ""المعالج الذكي لجميع الرسائل""
```

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

```
# معالجة خاصة للأزرار
```

```
if user_message == "🔍 أعراض إسهال":
```

```
    response = intelligent_medical_assistant("إسهال مائي")
```

```
elif user_message == "🦠 مكورات عنقودية":
```

```
    response = intelligent_medical_assistant("MRSA")
```

```
elif user_message == "🩺 علاجات شائعة":
```

```
    response = get_treatment_overview()
```

```
elif user_message == "🧼 دليل التعقيم":
```

```
    response = get_comprehensive_sterilization_guide()
```

```
elif user_message == "📊 حاسبة الجرعات":
```

```
    response = get_dosing_calculator_guide()
```

```
elif user_message == "⚠️ إجراءات العزل":
```

```
    response = "⚠️ استخدم اسم الميكروب لمعرفة إجراءات العزل المحددة ⚠️"
```

```

elif user_message == "👤$□ المطورة □":
    await creator_info(update, context)
    return
elif user_message == "👤□ المساعدة":
    response = get_intelligent_help_response("مساعدة")
else:
    # استخدام المساعد الذكي للرسائل النصية
    response = intelligent_medical_assistant(user_message)

# إضافة التوقيع في نهاية كل رد
response += "\n\n---\n**حنين علي عثمان**"
response += "\n🏥" مستشفی السواني بن آدم - قسم مكافحة العدوى
response += "\n$□ استشيری الطبيب للتشخيص النهائي"

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

# ===== إعداد وتشغيل البوت =====
def setup_bot():
    """تهيئة البوت وإضافة المعالجات"""
    application = Application.builder().token(TOKEN).build()

    # إضافة معالجات الأوامر
    application.add_handler(CommandHandler("start", start))
    application.add_handler(CommandHandler("creator", creator_info))
    application.add_handler(CommandHandler("help", start))
    application.add_handler(CommandHandler("calculate_vancomycin",
    calculate_vancomycin_dose))

    # إضافة معالج الرسائل النصية
    application.add_handler(MessageHandler(filters.TEXT & ~filters.COMMAND,
    handle_message))

```



```
return application
```

```
def main():
```

```
    """الدالة الرئيسية لتشغيل البوت"""
```

```
    try:
```

```
        # تهيئة البوت
```

```
        bot_app = setup_bot()
```

```
        # رسالة بدء التشغيل
```

```
        print("=" * 60)
```

```
        print("بدء تشغيل البوت الذكي لمكافحة العدوى 🛡️")
```

```
        print("المطورة: الصيدلانية حنين علي عثمان 👩💻")
```

```
        print("المستشفى: السواني بن آدم 🏥")
```

```
        print("الإصدار: المتكامل الشامل 📋")
```

```
        print("المميزات: قاعدة معرفة شاملة + ذكاء اصطناعي 🧠")
```

```
        print("=" * 60)
```

```
        # بدء التشغيل
```

```
        bot_app.run_polling()
```

```
    except Exception as e:
```

```
        print(f"❌ فشل تشغيل البوت {e}")
```

```
        print("💡 تأكد من")
```

```
        print("environment variables تكون البوت في -")
```

```
        print("اتصال الإنترنت -")
```

```
        print("تنصيب المكتبات المطلوبة -")
```

```
# ===== تشغيل التطبيق =====
```

```
if __name__ == '__main__':
```

تشغيل سيرفر فلاكس للتوافق مع Replit/Railway

import threading

def run_flask():

app.run(host='0.0.0.0', port=5000, debug=False, use_reloader=False)

flask_thread = threading.Thread(target=run_flask, daemon=True)

flask_thread.start()

تشغيل البوت

main()