

مشروع "تحسين" لإدارة البلاغات عبر
واتساب بوت

• وصف المشروع:

مشروع "تحسين" هو نظام إلكتروني يتيح للمستخدمين إرسال بلاغات حول النظافة والصيانة في المرافق العامة من خلال مسح باركود باستخدام تطبيق واتساب. يتم استقبال هذه البلاغات وتسجيلها في قاعدة بيانات لمتابعتها وإدارتها.

• كيفية عمل المشروع

١. يقوم المستخدم بمسح الباركود المثبت في المرفق العام باستخدام كاميرا هاتفه، مما يفتح محادثة في تطبيق واتساب.
٢. يرسل المستخدم رسالة تتضمن تفاصيل البلاغ، مثل نوع المشكلة ووصفها.
٣. يقوم النظام المطور باستخدام (Flask) باستقبال البلاغ وتحليله.
٤. يتم تخزين البلاغ في قاعدة بيانات SQLite، حيث يتم تعيينه إلى حالة "قيد المعالجة".
٥. يمكن لموظفي الإدارة الاطلاع على البلاغات عبر API والتحقق من حالتها.
٦. عند حل المشكلة، يتم تحديث حالة البلاغ إلى "تم الحل"، ويتم إرسال إشعار للمستخدم.

• كيفية تشغيل المشروع

١. تثبيت المتطلبات
٢. تشغيل التطبيق عبر " `pip install flask` "
٣. الوصول إلى النظام:
٤. لزيارة الصفحة الرئيسية `http://127.0.0.1:5000/`
٥. لإرسال بلاغ جديد استخدم " `POST /receive_report` "
٦. لعرض البلاغات استخدم " `GET /reports` "
٧. لتحديث حالة البلاغ استخدم `PUT /update_report/<report_id>`

• شرح الكود البرمجي:

١. إعداد المشروع وتهيئة قاعدة البيانات

- يستخدم المشروع (Python) مع إطار (Flask) لإنشاء (API)
- يتم إنشاء قاعدة بيانات (SQLite) لتخزين البلاغات باستخدام الدالة `init_db()` التي تقوم بإنشاء الجدول إذا لم يكن موجودًا مسبقًا.

٢. صفحة الترحيب (Home Route)

- عند زيارة <http://127.0.0.1:5000/> " يعرض التطبيق رسالة ترحيبية تشير إلى أن النظام يعمل بشكل صحيح.

٣. استقبال البلاغات

- عند إرسال طلب إلى [POST /receive_report](#) " يتم استلام بيانات البلاغ من المستخدم ثم يتم التحقق منها وإضافتها إلى قاعدة البيانات، إذا كانت البيانات ناقصة، يتم إرسال رسالة خطأ.

٤. عرض جميع البلاغات

- عند إرسال طلب إلى [GET /reports](#) " يتم جلب جميع البلاغات المخزنة في قاعدة البيانات وإرجاعها بتنسيق [JSON](#).

٥. تحديث حالة البلاغ

- عند إرسال طلب [PUT /update_report/<report_id>](#) " يمكن تغيير حالة البلاغ إلى "تم الحل" أو أي حالة أخرى يحددها يتم تحديدها من قبل الجهة المستخدمة.

الكود البرمجي: app.py

```
from flask import Flask, request, jsonify
import sqlite3
import json

# إنشاء التطبيق وإضافة اسم "تحسين"
app = Flask(__name__)
app.config["APP_NAME"] = "تحسين"

# إنشاء قاعدة بيانات SQLite
def init_db():
    conn = sqlite3.connect('reports.db')
    cursor = conn.cursor()
    cursor.execute('''CREATE TABLE IF NOT EXISTS reports (
                        id INTEGER PRIMARY KEY AUTOINCREMENT,
                        user_phone TEXT,
                        category TEXT,
                        description TEXT,
                        status TEXT DEFAULT ' قيد المعالجة',
                        created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP)''')
    conn.commit()
    conn.close()

init_db()

# الصفحة الرئيسية لعرض رسالة ترحيب بالتطبيق
@app.route('/')
def home():
    message = {"message": f"مرحبًا بك في تطبيق {app.config['APP_NAME']} لإدارة البلاغات!"}

    # تحويل Unicode Escape ومنع UTF-8 باستخدام JSON
    response = app.response_class(
        response=json.dumps(message, ensure_ascii=False).encode('utf-8'),
        status=200,
        mimetype="application/json"
    )

    return response

# استقبال البلاغات عبر API
@app.route('/receive_report', methods=['POST'])
def receive_report():
    data = request.json
    user_phone = data.get("user_phone")
    category = data.get("category")
    description = data.get("description")
```

الكود البرمجي: app.py

```
if not all([user_phone, category, description]):
    return jsonify({"error": f"يجب إدخال جميع البيانات لتطبيق {app.config['APP_NAME']}"}, 400)

conn = sqlite3.connect('reports.db')
cursor = conn.cursor()
cursor.execute("INSERT INTO reports (user_phone, category, description) VALUES (?, ?, ?)",
               (user_phone, category, description))
conn.commit()
conn.close()

return jsonify({"message": f"تم استقبال البلاغ في تطبيق {app.config['APP_NAME']}"}, 200, {"!نجاح!"})

# عرض جميع البلاغات
@app.route('/reports', methods=['GET'])
def get_reports():
    conn = sqlite3.connect('reports.db')
    cursor = conn.cursor()
    cursor.execute("SELECT * FROM reports")
    reports = cursor.fetchall()
    conn.close()

    return jsonify([{"id": r[0], "user_phone": r[1], "category": r[2], "description": r[3], "status": r[4], "created_at": r[5]} for r in reports])

# تحديث حالة البلاغ
@app.route('/update_report/<int:report_id>', methods=['PUT'])
def update_report(report_id):
    data = request.json
    new_status = data.get("status")

    if not new_status:
        return jsonify({"error": "يجب تحديد الحالة الجديدة"}), 400

    conn = sqlite3.connect('reports.db')
    cursor = conn.cursor()
    cursor.execute("UPDATE reports SET status = ? WHERE id = ?", (new_status, report_id))
    conn.commit()
    conn.close()

    return jsonify({"message": f"تم تحديث حالة البلاغ رقم {report_id} في {app.config['APP_NAME']} إلى {new_status}!"})

# تشغيل التطبيق
if __name__ == '__main__':
    app.run(debug=True)
```