

```
import streamlit as st
import openai
import json
import time
from datetime import datetime

# =====-
# - الحارس الدلالي - Smart Semantic Guardian
# هكا ثون الابتكار في البيانات - الهيئة العامة للإحصاء
# =====-

st.set_page_config(
    page_title="الحارس الدلالي | Smart Guardian",
    page_icon="🛡️",
    layout="wide",
    initial_sidebar_state="expanded"
)

# CSS تصميم احترافي
st.markdown("""
<style>
    @import url('https://fonts.googleapis.com/css2?family=Tajawal:wght@400;700;900');

    * { font-family: 'Tajawal', sans-serif !important; }

    .main-header {
        background: linear-gradient(135deg, #1a3a6b 0%, #0d6efd 100%);
        padding: 30px;
        border-radius: 15px;
        text-align: center;
        color: white;
        margin-bottom: 30px;
    }

    .main-header h1 { font-size: 2.5rem; font-weight: 900; margin: 0; }
    .main-header p { font-size: 1.1rem; opacity: 0.9; margin-top: 8px; }

    .error-card {
        background: #fff5f5;
        border-right: 5px solid #e53e3e;
        padding: 15px 20px;
        border-radius: 10px;
        margin: 10px 0;
    }

```

```
.warning-card {
    background: #fffbeb;
    border-right: 5px solid #d69e2e;
    padding: 15px 20px;
    border-radius: 10px;
    margin: 10px 0;
}

.success-card {
    background: #f0fff4;
    border-right: 5px solid #38a169;
    padding: 15px 20px;
    border-radius: 10px;
    margin: 10px 0;
}

.score-box {
    text-align: center;
    padding: 25px;
    border-radius: 15px;
    font-size: 3rem;
    font-weight: 900;
}

.form-section {
    background: #f8fafc;
    padding: 25px;
    border-radius: 15px;
    border: 1px solid #e2e8f0;
    margin-bottom: 20px;
}

.rtl { direction: rtl; text-align: right; }

div[data-testid="stButton"] button {
    background: linear-gradient(135deg, #1a3a6b, #0d6efd);
    color: white;
    border: none;
    border-radius: 10px;
    font-size: 1.1rem;
    font-weight: 700;
    padding: 12px 30px;
    width: 100%;
}

</style>
"""", unsafe_allow_html=True)
```

```

# =====
# الذكي للنموذج اللغوي PROMPT الى
# =====

```

صطناعي متخصص في التحقق من جودة استئمارات التعداد والمسح الإحصائي"" = SYSTEM\_PROMPT

نك: تحليل إجابات الاستماره واكتشاف التناقضات الدلالية والمنطقية بين الحقول المختلفة

:أنواع الأخطاء التي تبحث عنها

1. تعارف المسمى الوظيفي مع المؤهل العلمي.
2. تعارف العمر مع سنوات الخبرة (مثلاً: عمر 22 مع 25 سنة خبرة).
3. تعارف العمر مع المؤهل العلمي (مثلاً: عمر 16 مع درجة دكتوراه).
4. تعارف الجنسية مع اللغة الأم.
5. تعارف الحالة الوظيفية مع الراتب (مثلاً: غير موظف مع راتب 10000).
6. تعارض عدد أفراد الأسرة مع الحالة الاجتماعية.
7. أي تناقض منطقي آخر.

:فقط بهذا الشكل بالضبط JSON يجب دائمًا بـ

```

{
    "confidence_score": [رقم من 0 إلى 100],
    "status": ["clean" أو "warning" أو "error"],
    "issues": [
        {
            "severity": ["high" أو "medium" أو "low"],
            "field_1": "اسم الحقل الأول",
            "field_2": "اسم الحقل الثاني",
            "description": "وصف التناقض بالعربية",
            "suggestion": "اقتراح التصحیح"
        }
    ],
    "summary": "ملخص قصير بالعربية"
}

```

عد كقائمة فارغة [] و issues إذا لم توجد أخطاء، أعد status: "clean" و score: 95-100"""

```
def analyze_form(api_key: str, form_data: dict) -> dict:
```

```
    """إرسال البيانات للنموذج اللغوي وتلقي التحليل"""
    client = openai.OpenAI(api_key=api_key)
```

```
    form_text = "\n".join([f"- {k}: {v}" for k, v in form_data.items() if v])
```

: حل هذه الاستماره واكتشف أي تناقضات منطقية أو دلالية"" = user\_message = f"""

```
{form_text}
```

ـ أجب بـ """ فقط JSON تذكر:

```
response = client.chat.completions.create(  
    model="gpt-4o",  
    messages=[  
        {"role": "system", "content": SYSTEM_PROMPT},  
        {"role": "user", "content": user_message}  
    ],  
    temperature=0.1,  
    max_tokens=1000  
)  
  
raw = response.choices[0].message.content.strip()  
# تنظيف الرد من markdown  
raw = raw.replace("```json", "").replace("```", "").strip()  
return json.loads(raw)  
  
# ======  
# واجهة المستخدم  
# ======  
  
# الترويسة #  
st.markdown("""  


<h1>الحارس الدلالي</h1>  
    <p>نظام ذكي للتحقق من جودة البيانات الإحصائية لحظياً</p>  
    <small>هكاثون الابتكار في البيانات | الهيئة العامة للإحصاء 2026</small>  
</div>  
""", unsafe_allow_html=True)  
  
# الشريط الجانبي - الإعدادات #  
with st.sidebar:  
    st.markdown("##  الإعدادات")  
    api_key = st.text_input("🔑 OpenAI API Key", type="password", placeholder="sk-...")  
  
    st.markdown("---")  
    st.markdown("### <img alt='Bar chart icon' data-bbox='318 811 348 831"/> إحصائيات الجلسة")  
    if "total_forms" not in st.session_state:  
        st.session_state.total_forms = 0  
        st.session_state.errors_found = 0  
        st.session_state.clean_forms = 0  
  
    col1, col2 = st.columns(2)


```

```

col1.metric("استمرارات فحصت", st.session_state.total_forms)
col2.metric("أخطاء اكتُشفت", st.session_state.errors_found)

st.markdown("---")
st.markdown("###💡")("كيف يعمل النظام؟")
استخدام الذكاء الاصطناعي للكشف عن التناقضات المنطقية قبل حفظ البيانات()
```

المسار 🔗 "المسار الثاني: المعالجة الذكية"()

الحارس الدلالي\المسار الثاني: المعالجة الذكية()

```
# تبويبات
tab1, tab2, tab3 = st.tabs(["📝 اختبار سجلات جاهزة", "✍ لوحة التحكم فاعلية"])
```

```
# =====
# التبويب الأول: الاستمارة التفاعلية
# =====
with tab1:
    st.markdown("###📝")("استمارة مسح سوق العمل")
    st.markdown(">*أدخل البيانات وسيقوم النظام بفحصها لحظياً*")()
```

```
col1, col2 = st.columns(2)
```

```
with col1:
    st.markdown('<div class="form-section">', unsafe_allow_html=True)
    st.markdown("###👤")("البيانات الشخصية")
    name = st.text_input("الاسم", placeholder="ادخل الاسم...")("الجنس", ["ذكر", "أنثى"])
    age = st.number_input("العمر", min_value=10, max_value=100, value=30)
    gender = st.selectbox("الجنس", ["ذكر", "أنثى"])
    nationality = st.selectbox("الوطنيّة", ["آخر", "باتكستاني", "هندي", "أردني"])
    native_language = st.selectbox("اللغة الأم", ["آخر", "الهندية", "الأردية", "الإنجليزية"])
    st.markdown('</div>', unsafe_allow_html=True)
```

```
with col2:
    st.markdown('<div class="form-section">', unsafe_allow_html=True)
    st.markdown("###💼")("البيانات المهنية")
    education = st.selectbox("المؤهل العلمي", ["بكالوريوس", "ماجستير", "دكتوراه"])
    employment_status = st.selectbox("الحالة الوظيفية", [
        "موظفي القطاع الحكومي", "موظفي القطاع الخاص", "أعمال حرفة", "غير موظف", "طالب"])
    ])
```

```

"متقدعاً"
])
job_title = st.text_input("المسمي الوظيفي", placeholder="المسمي الوظيفي")
years_exp = st.number_input("سنوات الخبرة", min_value=0, max_value=50, value=0)
monthly_salary = st.number_input("الراتب الشهري (₪)", min_value=0, max_value=10000, value=0)
st.markdown('</div>', unsafe_allow_html=True)

col3, col4 = st.columns(2)
with col3:
    st.markdown('<div class="form-section">', unsafe_allow_html=True)
    st.markdown("#### 🏠")
    st.markdown("الحالة الاجتماعية")
    marital_status = st.selectbox(["أعزب", "متزوج", "مطلق", "أرمل"], ["أعزب", "متزوج", "مطلق", "أرمل"])
    family_members = st.number_input("عدد أفراد الأسرة", min_value=1, max_value=10, value=1)
    children_count = st.number_input("عدد الأطفال", min_value=0, max_value=15, value=0)
    st.markdown('</div>', unsafe_allow_html=True)

with col4:
    st.markdown('<div class="form-section">', unsafe_allow_html=True)
    st.markdown("#### 📈")
    st.markdown("بيانات إضافية")
    region = st.selectbox(["المكرمة", "المدينة المنورة", "الشرقية", "آخر"], ["المكرمة", "المدينة المنورة", "الشرقية", "آخر"])
    sector = st.selectbox(["حكومي", "خاص", "غير ربحي", "لا ينطبق"], ["حكومي", "خاص", "غير ربحي", "لا ينطبق"])
    income_source = st.selectbox(["راتب", "أعمال حرة", "استثمارات", "لا يوجد"], ["راتب", "أعمال حرة", "استثمارات", "لا يوجد"])
    st.markdown('</div>', unsafe_allow_html=True)

st.markdown("---")

if st.button("🔍 فحص الاستماراة بالذكاء الاصطناعي", use_container_width=True):
    if not api_key:
        st.error("⚠️ في الشريط الجانبي الرجاء إدخال OpenAI API Key")
    else:
        form_data = {
            "العمر": age,
            "الجنس": gender,
            "الجنسية": nationality,
            "اللغة الأم": native_language,
            "المؤهل العلمي": education,
            "الحالة الوظيفية": employment_status,
            "المسمي الوظيفي": job_title,
            "سنوات الخبرة": years_exp,
            "الراتب الشهري": monthly_salary,
            "الحالة الاجتماعية": marital_status,
            "عدد أفراد الأسرة": family_members,
            "عدد الأطفال": children_count,
            "القطاع": sector,
            "مصدر الدخل": income_source
        }

```

```

with st.spinner("🌐 النظام يحلل الاستماراة ..."):

    try:
        result = analyze_form(api_key, form_data)

        st.session_state.total_forms += 1
        if result.get("issues"):
            st.session_state.errors_found += len(result["issues"])
        else:
            st.session_state.clean_forms += 1

        # عرض النتائج
        st.markdown("---")
        st.markdown("#📊 نتائج الفحص")

        score = result.get("confidence_score", 0)
        status = result.get("status", "error")

        col_s1, col_s2, col_s3 = st.columns(3)

        with col_s1:
            color = "#38a169" if score >= 80 else "#d69e2e" if score
            st.markdown(f"""
<div class="score-box" style="background:{color}22; border:
<div style="color:{color}">{score}</div>
<div style="font-size:1rem; color:#666">درجـة الثقة</div>
</div>""", unsafe_allow_html=True)

        with col_s2:
            issues_count = len(result.get("issues", []))
            st.markdown(f"""
<div class="score-box" style="background:#ebf8ff; border:
<div style="color:#3182ce">{issues_count}</div>
<div style="font-size:1rem; color:#666">مشكلـة مكتشفـة</div>
</div>""", unsafe_allow_html=True)

        with col_s3:
            status_map = {"clean": ("✅ 38#", "نظيفـا169"), "warning":
            s_text, s_color = status_map.get(status, ("❓ 666#", "حددـ"))
            st.markdown(f"""
<div class="score-box" style="background:{s_color}22; border:
<div style="color:{s_color}; font-size:2rem">{s_text}
<div style="font-size:1rem; color:#666">الحالـة</div>
</div>""", unsafe_allow_html=True)

            st.markdown(f"**📝 المـلخص:** {result.get('summary', '')}")

    issues = result.get("issues", [])

```

```

if issues:
    st.markdown("### !: المشكلات المكتشفة")
    for i, issue in enumerate(issues, 1):
        severity = issue.get("severity", "medium")
        card_class = "error-card" if severity == "high" else
        icon = "🔴" if severity == "high" else "🟡"

        st.markdown(f"""
<div class="{card_class}">
<strong>{icon} {i}: {issue.get('field_1',
📍 {issue.get('description', '')})}<br>
💡 <em>{issue.get('suggestion', '')}</em>
</div>""", unsafe_allow_html=True)
else:
    st.markdown("""
<div class="success-card">
<strong>✓ لم يتم اكتشاف أي تناقضات
البيانات المدخلة متسقة ومنطقية.
</div>""", unsafe_allow_html=True)

# حفظ في سجل الجلسة
if "history" not in st.session_state:
    st.session_state.history = []
st.session_state.history.append({
    "time": datetime.now().strftime("%H:%M:%S"),
    "score": score,
    "issues": len(issues),
    "status": status
})

except json.JSONDecodeError:
    st.error("خطأ في تحليل رد النموذج اللغوي. حاول مرة أخرى")
except Exception as e:
    st.error(f"خطأ: {str(e)}")

# =====
# التبويب الثاني: اختبار سجلات جاهزة
# =====
with tab2:
    st.markdown("### 📁 اختبار سجلات من واقع العمل الميداني")
    st.markdown("*سجلات تحتوي على أخطاء منطقية لاختبار قدرة النظام*")

test_records = [
{
    "الاسم": "أحمد محمد",
    "العمر": 19,
}

```

```

        "المؤهل العلمي": "دكتوراه",
        "المسمى الوظيفي": "طبيب متخصص",
        "سنوات الخبرة": "15",
        "الحالة الوظيفية": "موظفي القطاع الخاص",
        "الراتب الشهري": "25000",
        "الحالة الاجتماعية": "متزوج",
        "عدد الأطفال": "5",
        "الجنسية": " سعودي",
        "اللغة الأم": "الإنجليزية"
    },
    {
        "الاسم": "فاطمة علي",
        "العمر": "35",
        "المؤهل العلمي": "بكالوريوس",
        "المسمى الوظيفي": "سائق شاحنة",
        "سنوات الخبرة": "10",
        "الحالة الوظيفية": "غير موظف",
        "الراتب الشهري": "8000",
        "الحالة الاجتماعية": "أعزب",
        "عدد الأطفال": "4",
        "الجنسية": " سعودي",
        "اللغة الأم": " العربية"
    },
    {
        "الاسم": "خالد السالم",
        "العمر": "45",
        "المؤهل العلمي": "بكالوريوس هندسة",
        "المسمى الوظيفي": "مهندس مدني",
        "سنوات الخبرة": "20",
        "الحالة الوظيفية": "موظفي القطاع الحكومي",
        "الراتب الشهري": "18000",
        "الحالة الاجتماعية": "متزوج",
        "عدد الأطفال": "3",
        "الجنسية": " سعودي",
        "اللغة الأم": " العربية"
    }
]

labels = [
    "سجل 1: عمر 19 + دكتوراه + 15 سنة خبرة",
    "سجل 2: غير موظف + راتب 8000 + سائق",
    "سجل 3: مهندس - سجل نظيف"
]

selected_idx = st.selectbox("اختر سجل للفحص", range(len(labels)), format_fun)

selected_record = test_records[selected_idx]

```

```

st.markdown("**:بيانات السجل**")
cols = st.columns(3)
items = list(selected_record.items())
for i, (k, v) in enumerate(items):
    cols[i % 3].info(f"**{k}:** {v}")

if st.button("🔍 فحص هذا السجل", use_container_width=True):
    if not api_key:
        st.error("⚠️ في الشريط الجانبي OpenAI API Key الرجاء إدخال")
    else:
        with st.spinner("🤖 جاري التحليل..."):
            try:
                result = analyze_form(api_key, selected_record)

                st.session_state.total_forms += 1

                score = result.get("confidence_score", 0)
                issues = result.get("issues", [])
                status = result.get("status", "error")

                color = "#38a169" if score >= 80 else "#d69e2e" if score >= 6

                st.markdown(f"""
                <div style="background:{color}22; border:3px solid {color}; padding: 10px;>
                    <h2 style="color:{color}> درجة الثقة: {score}/100</h2>
                    <p>{result.get('summary', '')}</p>
                </div>""", unsafe_allow_html=True)

                if issues:
                    st.session_state.errors_found += len(issues)
                    st.markdown(f"### ❌ تم اكتشاف {len(issues)} مشكلة")
                    for issue in issues:
                        severity = issue.get("severity", "medium")
                        card_class = "error-card" if severity == "high" else "warning-card"
                        st.markdown(f"""
                        <div class="{card_class}">
                            <strong>⚠️ {issue.get('field_1', '')} ↳ {issue.get('description', '')}<br>
                            <em>💡 {issue.get('suggestion', '')}</em>
                        </div>""", unsafe_allow_html=True)
                else:
                    st.success("✅ لا توجد تناقضات في هذا السجل")
            except Exception as e:
                st.error(f"خطأ: {str(e)}")

```

```

# =====
# التبويب الثالث: لوحة التحكم
# =====

with tab3:
    st.markdown("### 🔍 (لوحة متابعة جودة البيانات)

col1, col2, col3, col4 = st.columns(4)
col1.metric("📋 إجمالي الاستمرارات", st.session_state.total_forms)
col2.metric("🔴 أخطاء مكتشفة", st.session_state.errors_found)
col3.metric("✅ استمرارات نظيفة", st.session_state.clean_forms)
error_rate = round((st.session_state.errors_found / max(st.session_state.tota
col4.metric("📊 معدل الخطأ", f"{error_rate}%")

if "history" in st.session_state and st.session_state.history:
    st.markdown("## 📊 (سجل الفحوصات الأخيرة)
    import pandas as pd
    df = pd.DataFrame(st.session_state.history)
    df.columns = ["الوقت", "درجة الثقة", "المشكلات", "الحالة"]
    st.dataframe(df, use_container_width=True)

    st.markdown("## ⏲ (تطور درجة الثقة)
    st.line_chart(df[["درجة الثقة"]])

else:
    st.info("ابدأ بفحص استمرارات لعرض الإحصائيات هنا")
    st.markdown("---")
    st.markdown("##💡 (القيمة المضافة للنظام)

col_a, col_b, col_c = st.columns(3)
with col_a:
    st.markdown("""
        <div style="background:#ebf8ff; padding:20px; border-radius:10px; text-align:center;>
            <h2>⌚</h2>
            <h3> توفير الوقت </h3>
            <p>/> اكتشاف الأخطاء لحظياً بدلاً من المعالجة اليدوية اللاحقة</p>
        </div>""", unsafe_allow_html=True)
with col_b:
    st.markdown("""
        <div style="background:#f0fff4; padding:20px; border-radius:10px; text-align:center;>
            <h2>📊</h2>
            <h3> جودة البيانات </h3>
            <p>/> ضمان دقة وموثوقية البيانات الإحصائية الوطنية</p>
        </div>""", unsafe_allow_html=True)
with col_c:
    st.markdown("""
        <div style="background:#fffbeb; padding:20px; border-radius:10px; text-align:center;>

```

```
<h2>💡</h2>
<h3>ذكاء اصطناعي</h3>
<p>تجاوز القواعد الجامدة نحو الفهم الدلالي العميق</p>
</div>"", unsafe_allow_html=True)

# Footer
st.markdown("---")
st.markdown("""
<div style="text-align:center; color:#999; padding:10px">
الحارس الدلالي | هكاثون الابتكار في البيانات | الهيئة العامة للإحصاء 2026
</div>"", unsafe_allow_html=True)
```