البرمجيات والذكاء الاصطناعي

بناء أول مشروع حقيقي بالذكاء الاصطناعي

انتقل من فكرة متناثرة إلى أداة ذكاء اصطناعي عاملة وموثّقة على GitHub — وهي نفس آلية العمل الكامنة خلف أي عمل برمجي حقيقي يُعرض في ملف الأعمال.

المستوى: مبتدئ إلى متوسط **مدة التنفيذ:** عطلتا نهاية أسبوع **التكلفة التقريبية:** 0–$10

> قبل البدء: يفترض هذا المخطط إلمامًا أساسيًا بلغة Python (المتغيرات، الدوال، الحلقات). إذا كان هذا جديدًا عليك، خصّص بضع ساعات لدورة تمهيدية مجانية أولًا — لست بحاجة لإتقان كامل، بل لراحة أساسية.

01 الهدف

معظم "مشاريع الذكاء الاصطناعي" للمبتدئين هي خلية واحدة في Jupyter تستدعي واجهة برمجية مرة واحدة ثم لا يُعاد لمسها أبدًا. بنهاية هذا المخطط، سيكون لديك شيء مختلف: أداة صغيرة لكنها كاملة مدعومة بالذكاء الاصطناعي — ببنية مشروع حقيقية، وتحكم بالإصدارات، ومعالجة للأخطاء، وواجهة بسيطة، وملف README — يمكنك إرسال رابطها لأي شخص وقول **"هذا ما بنيته"**. الهدف ليس نموذجًا معقدًا، بل إثبات أنك قادر على أخذ فكرة من مجلد فارغ إلى شيء يمكن لشخص غريب تثبيته وتشغيله.

02 المتطلبات الأساسية

- تثبيت Python 3.10 أو أحدث، والراحة في تشغيل السكربتات من الطرفية (terminal)
- حساب مجاني على GitHub — حيث سيُستضاف مشروعك
- الراحة في تثبيت الحزم عبر `pip` — إذا كان هذا جديدًا، تدرّب على حزمة صغيرة واحدة أولًا

03 الأدوات الموصى بها

الأداة	الغرض	ملاحظات
Python 3.10+	اللغة الأساسية	مجانية — python.org
VS Code	محرر الأكواد	مجاني، مع تثبيت إضافة Python
Git + GitHub	التحكم بالإصدارات واستضافة الكود	مجاني للمستودعات العامة
بيئة افتراضية (venv)	تعزل تبعيات المشروع	مدمجة في Python. لا حاجة لتثبيت
مفتاح واجهة برمجية للذكاء الاصطناعي (OpenAI أو Anthropic أو Hugging Face)	يشغّل جزء "الذكاء الاصطناعي" في مشروعك	معظمها يقدّم مستوى مجاني محدود؛ خصّص ٥–١٠$ للاختبار
Streamlit	يحوّل سكربت Python إلى واجهة ويب بسيطة	مجاني، أمر تثبيت واحد فقط

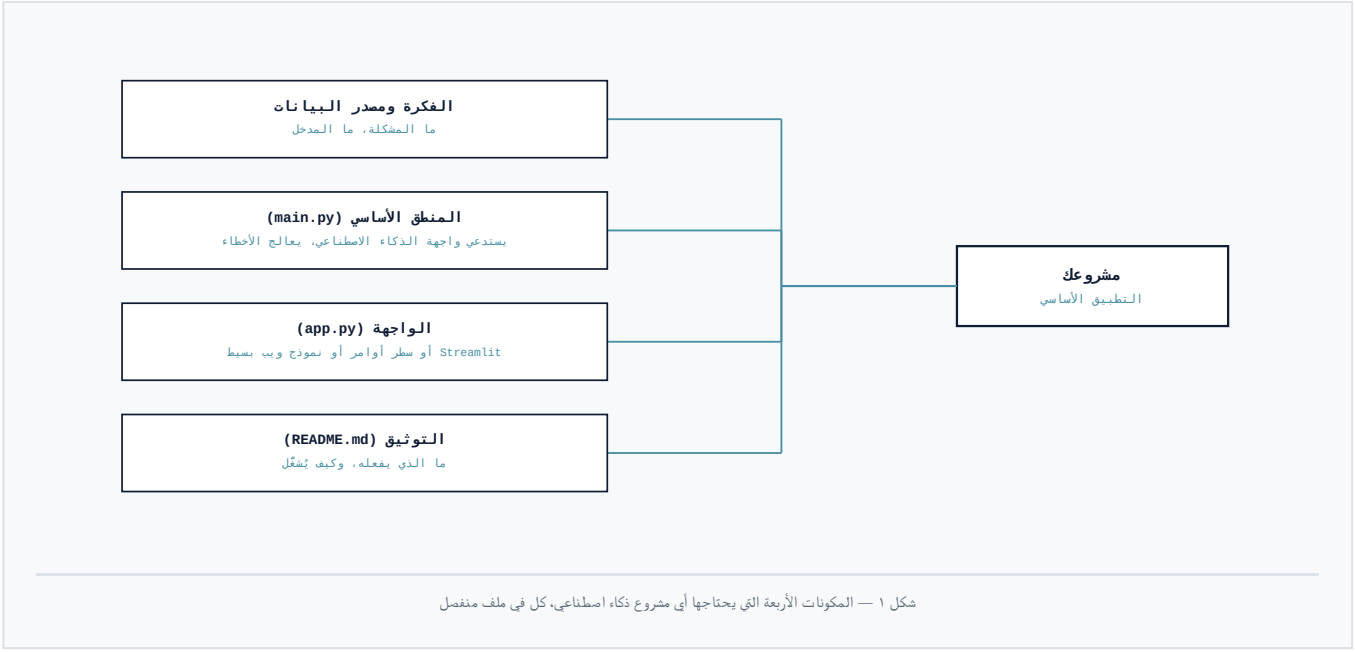
04 الأدوات المطلوبة

- حاسوب مثبّت عليه VS Code و Python
- طرفية (terminal) — مدمجة داخل VS Code

> **استخدام الذكاء الاصطناعي بمسؤولية**
> استخدام مساعد ذكاء اصطناعي (مثل Claude أو ChatGPT) للمساعدة في كتابة أجزاء من الكود أمر معتاد — لكن يجب أن تفهم كل سطر جيدًا بما يكفي لشرحه، وأن تذكر ذلك بوضوح في ملف README ضمن ملاحظة "المساعدة بالذكاء الاصطناعي". لا تُقدّم أبدًا كودًا كتبه الذكاء الاصطناعي في مسابقة أو طلب التحاق دون الإفصاح عن ذلك إن كان الإفصاح مطلوبًا، ولا تدع كودًا مولّدًا يلامس بيانات مستخدمين حقيقية قبل أن تقرأه وتفهمه بالكامل.

05 بنية المشروع

كل مشروع ذكاء اصطناعي — مهما كان صغيرًا — له نفس المكونات الأربعة المتحركة. إبقاؤها في ملفات منفصلة منذ اليوم الأول هو ما يجعل المشروع سهل التصحيح والتوسيع والشرح في مقابلة.



شكل ١ — المكونات الأربعة التي يحتاجها أي مشروع ذكاء اصطناعي، كل في ملف منفصل

المكوّن	الملف	المسؤولية
المنطق الأساسي	main.py / core.py	يتواصل مع واجهة الذكاء الاصطناعي، يعالج الاستجابة، يتعامل مع الإخفاقات
الواجهة	app.py	تستقبل إدخال المستخدم وتعرض الخرج — Streamlit هو الخيار الأسرع للمبتدئين
الإعدادات	.env (غير مرفوعة على GitHub)	تخزّن مفتاح الواجهة البرمجية خارج الكود
التوثيق	README.md	يشرح المشروع لأي شخص يجده، بما في ذلك أنت في المستقبل

06 خطوات البناء

1. **اختر فكرة تستحق البناء.** اختر مشكلة تواجهها فعليًا — أداة دراسة، ملخّص نصوص لقراءتك الخاصة، نظام توصية بسيط. تجنّب "بناء روبوت محادثة" دون مستخدم مستهدف؛ التحديد هو ما يجعل المشروع قابلًا للدفاع عنه لاحقًا.
2. **جهّز مجلد المشروع.** أنشئ مجلدًا، شغّل `python -m venv venv`، فعّله، ثم `git init`.
3. **قم بأول commit قبل كتابة أي كود للذكاء الاصطناعي.** ارفع ملف README مبدئي و `.gitignore` (يتضمن `/venv` و `.env`). هذه العادة وحدها تمنع أشيع خطأ للمبتدئين: تسريب مفتاح واجهة برمجية.
4. **اكتب الدالة الأساسية أولًا، دون واجهة.** احصل على استدعاء واحد يعمل لواجهة الذكاء الاصطناعي يطبع نتيجة في الطرفية. تأكد من عمله قبل بناء أي شيء فوقه.
5. **أضف معالجة الأخطاء.** ضع استدعاء الواجهة البرمجية داخل كتلة try/except وتعامل على الأقل مع: مفتاح مفقود، فشل الشبكة، ومدخل فارغ. هذه هي الفجوة الأكبر بين "عرض يعمل مرة واحدة" و"أداة تعمل فعليًا".
6. **ابنِ واجهة بسيطة.** ضع دالتك الأساسية داخل تطبيق Streamlit — مربع نص، زر، ومنطقة عرض للنتيجة تكفي.
7. **اكتب README بشكل صحيح.** ما الذي يفعله، كيف يُثبّت، كيف يُشغَّل، وسطر واحد عن كيفية استخدام مساعدة الذكاء الاصطناعي إن وُجدت.
8. **انشره.** تقدّم منصات Streamlit Community Cloud أو Render أو Hugging Face Spaces استضافة مجانية — ادفع مستودعك واربطه ببضع نقرات.

07 شرح الكود

هذه نسخة مبسّطة وصادقة من دالة أساسية للذكاء الاصطناعي — وليست نسخة تجريبية تخفي أنماط فشلها. لاحظ الأمور الثلاثة التي يتجاهلها سكربت المبتدئين عادة: قراءة المفتاح من بيئة النظام (وليس مكتوبًا مباشرة في الكود)، تعليق توضيحي (docstring) يشرح ما تفعله الدالة، ومعالجة صريحة للأخطاء.

```
# core.py — الدالة التي يستدعيها كل شيء آخر
import os
from anthropic import Anthropic

client = Anthropic(api_key=os.environ.get("ANTHROPIC_API_KEY"))

def summarize(text: str) -> str:
    """Return a 2-sentence summary of `text`, or raise a clear error."""
    if not text.strip():
        raise ValueError("Input text is empty.")
    if not os.environ.get("ANTHROPIC_API_KEY"):
        raise EnvironmentError("Missing ANTHROPIC_API_KEY — set it in your .env file.")
```

```
try:
    response = client.messages.create(
        model="claude-sonnet-5",
        max_tokens=200,
        messages=[{"role": "user", "content": f"Summarize in 2 sentences: {text}"}]
    )
    return response.content[0].text
except Exception as e:
    # Fail loudly and clearly instead of silently returning None
    raise RuntimeError(f"AI request failed: {e}")
```

ملاحظة: يُقرأ مفتاح الواجهة البرمجية من متغير بيئي، ولا يُكتب أبدًا داخل الملف نفسه.

08 الاختبار واستكشاف الأخطاء

العارض	السبب المحتمل	الحل
GitHub يحذّر من مفتاح سري / المفتاح توقف عن العمل	تم رفع مفتاح الواجهة البرمجية إلى المستودع	ألغِ المفتاح فورًا، أنشئ مفتاحًا جديدًا، وأضف `.env` إلى `.gitignore` قبل المتابعة
التطبيق يتعطل دون رسالة مفيدة	لا توجد معالجة أخطاء حول استدعاء الواجهة البرمجية	ضع الاستدعاء داخل try/except واطبع رسالة محددة لكل نوع فشل
النشر يفشل لكنه يعمل محليًا	ملف `requirements.txt` مفقود	شغّل `pip freeze > requirements.txt` قبل النشر
المُراجِع لا يستطيع تشغيله	README لا يتضمن خطوات الإعداد	أضف أوامر التثبيت والتشغيل بدقة — افترض أن القارئ لا يعرف شيئًا
ملف واحد بطول ٣٠٠ سطر، يصعب شرحه	لا فصل بين المنطق والواجهة	قسّمه إلى core.py و app.py كما في القسم ٠٥

09 الخطوة التالية

من أداة عاملة إلى عمل في ملف الإنجازات

مشروع منشور بملف README واضح هو بالفعل أكثر مما يعرضه معظم المتقدمين — لكن الخطوة التي تجعله لا يُنسى هي كتابة مختصرة عن المشكلة التي يحلّها وما ستحسّنه لاحقًا، مثبّتة على ملفك الشخصي في GitHub. اجمع هذا المخطط مع **SA-002 (نشر وتوسيع تطبيقات الذكاء الاصطناعي)** لتتعلم كيفية التعامل مع مستخدمين حقيقيين وحدود الاستخدام عندما يبدأ أشخاص غيرك باستخدامه — وراجع وحدة دليل الجامعات والمنح لمعرفة كيفية وصف هذا المشروع في طلبات الالتحاق.