```
dist/
*.egg-info/
# Jupyter
.ipynb_checkpoints/
# Secrets & keys (VERY IMPORTANT)
.env.*
*.pem
*.key
*.p12
                                                               قەاعد سىعة:
                                           • folder/ لتجاهل مجلد كامل.
                                          • ".log لتجاهل كل الامتدادات.
                                 • keep.me! لألغاء التجاهل لملف محدد.
                                                • يبدأ / يعنى الجذر فقط.
                                                         أخطاء شائعة + حُلها:
                                            رفعت ملفات قبل إضافة .gitignore
git rm -r --cached.
git add.
git commit -m "Apply .gitignore"
 • الاعتماد على .gitignore لحماية الأسرار غير كافٍّ: لَّا تضعُ الأسرارِ أصلًا داخل
 المشروع. استخدم متغيرات بيئة وملفَ .env محليًا فقطّ، وقدّمَ لَلطلاب ملفًّا
# .env.example
API_KEY=
DB_URL=
                        Add license (إضافة رخصة) 🗸 «شرح كامل»
ملفً يحدد ما المسموح للآخرين فعله بكودك (استخدام/تعديل/توزيع) وشروط ذلك.
 قانونيًا: بدون ترخيص يعني عمليًا All rights reserved (لا استخدام قانوني).

    تربويًا/عمليًا: يتيح للآخرين التعلم والاستخدام وفق شروط واضحة.

                                       • تريد انتشارًا سهلًا وبساطة → MIT

    ▼ تريد حماية براءات الاختراع أيضًا حماية براءات الاختراع أيضًا

    • تريد إلزام المشتقات أن تبقى مفتوحة → GPL-3.0

    مشروع تدریبی بسیط → غالبًا MIT مناسب

    لا تريد استخدامًا خارجيًا → اجعل الربيو Private أو اختر No license

                                        SPDX (سطر مختصر في رأس الملفات):
# SPDX-License-Identifier: MIT
                                            قالب ملف LICENSE (MIT) جاهز:
                     ضع هذا في ملف باسم LICENSE وبدّل الاسم والسنة.
MIT License
Copyright (c) 2025 Qusai Omar AlBahri
```

Permission is hereby granted, free of charge, to any person obtaining a

of this software and associated documentation files (the "Software"), to

in the Software without restriction, including without limitation the

to use, copy, modify, merge, publish, distribute, sublicense, and/or sell

copies of the Software, and to permit persons to whom the Software is

deal

#### Run

python main.py

## **Env (Secrets)**

Create .env from .env.example and fill required keys.

#### License

MIT © 2025 Your Name Add .gitignore

#### (إضافة .gitignore «شرح كامل»

قائمةً بما لا نريد رفعه إلى GitHub: ملفات مؤقتة، إعدادات IDE، بيئات افتراضية،

#### لماذا؟

- يحافظ على المستودع نظيفًا وصغيرًا.
- يمنع تسريب الأسرار (ʾAPl keys, `.env`).
- يتجنّب رفع ملفات جهازك (مثل `idea.` و `vscode.`).
  - أثناء الإنشاء اختر: Add .gitignore → Python
- أو لاحقًا أنشئ ملفًا باسم `gitignore. في جذر المشروع.

#### قالب عملی مُقترَح (Python + PyCharm/VS Code):

```aitianore # Byte-compiled / cache

\_\_pycache\_\_/
\*.py[cod]

\*\$py.class

# Virtual environments

.venv/

venv/ env/

ENV/

# IDE settings

.idea/

.vscode/

# OS junk

.DS\_Store

Thumbs.db

# Logs & local DBs

\*.log

logs/

db.salite3

# Testing / tooling caches

.pytest\_cache/

.tox/

.nox/

.mypy\_cache/

.pyre/

.pytype/

# Coverage / build artifacts

.coverage\*

htmlcov/ build/

#### فهم كل خيار في صفحة Create a new repository واستخدام أفضل الممارسات: اسم مناسب، README، .gitignorë، وترخيص (License) – ثم رفع الكود بأمان (بدون تسریب أسرار). 🔽 المتطلبات المسبقة • حساب GitHub فعّال. • تثبيت Git (اختياريًا: تهيئة SSH).

الطالب: إنشاء مستودع جديد

## 1) القسم General (عام) (المالك) Owner

الحساب أو المنظمة التي ستملك المستودع. عادةً حسابك الشخصي. Repository name (اسم المستودع) اسم المشروع. اجعله:

- قصيراً وواضحًا: student-grade-system
- بالأحرف الصغيرة مع شرطات بدل المسافات

مشروع جاهز محليًا أو نية لبدء مشروع جديد.

 بدون رموز خاصة أو مسافات في البداية/النهاية 💡 تَلْمَيْحَ: الْاسم يُظْهَر للمستخدمين وللباقة (package) لاحقًا، فليكن

## Description (الوصف)

على GitHub

@ المدف

ملخص قصير عن المشروع. في واجهة GitHub يظهر عدّاد (حتى 350 حرفًا). • مثال: "Simple Python app to calculate and manage student" ".grades

## 2) القسم Configuration (الإعدادات) (الظهور) Choose visibility

- Public: الكل يستطيع رؤية الكود.
- Private: أنت (ومن تمنحه صلاحية) فقط.

1 ملاحظة: Public ≠ Open-Source. الكود علني للرؤية، لكن 1 الاستخدام القانوني يتطلب License.

### Add README (اضافة

- ىفضًل تفعيلها (On).
- README.md هو صفحة المشروع الأمامية: ما هو؟ كيف يُثبّت؟ كيف يُشغَّل؟

### قالب README سريع :

# Student Grade System

Short description: Simple Python app to calculate student GPAs.

## Requirements

- Python 3.11+
- Create a virtual environment

## Setup ```bash python -m venv .venv # activate venv (Windows) .venv\Scripts\activate pip install -r requirements.txt

- الخلط بين Public وPublic → Open-Source أضف LICENSE دائمًا لو أردت السماح بالاستعمال.
- ullet حجم مستودع ضخم بسبب مجلدات venv و ullet أضف ullet.gitignore الصحيح ثم نفّذ تنظيف التتبع (git rm -r --cached ).
  - تاریخ commits غیر منظم → استخدم رسائل احترافیة: feat: add GPA calculation, fix: handle empty grades

## 7) تمارين للطلاب (عملي)

- 1. أنشَىٰ ريبو باسم student-grade-system.
- 2. اجعله Public، أضف Public، أضف README، .gitignore → Python، MIT License
  - 3. أنشئ env.example. وضع فيه:

API\_KEY= DB\_URL=

- 4. ارفع (push) ملف main.py بسيط يطبع "Hello, GitHub!".
- ارتكب خطأ متعمّدًا: أنشئ ملفًا اسمه temp.log ثم تأكد أنه غير مرفوع بسبب .gitignore.
  - 6. أضف سطر SPDX في أعلى main.py.
  - 7. صحّح مسارًا: ارفع ملِّفًا تم تجاهله بالخطأ عبر استعمال قاعدة! في .gitignore وتفسير السبب في README.

## 8) قائمة تحقّق سريعة (Before Create / First (Commit

قبل Create:

- اسم واضح
- وصف مختصر
- Public/Private مناسب
  - README → On •
- aitianore → Pvthon. •
- License مختار (MIT غالبًا)

### قبل أول Push:

- env.example. + موجود محليًا فقط + env.example في الربيو
  - \_\_pycache\_\_ غير متتبَّعة \\_\_pycache\_\_ غير متتبَّعة • رسالة Commit واضحة
- SPDX في أعلى الملفات (اختياري لكنه مُستحسن)

### ملاحق (جاهزة للنسخ)

(کامل gitignore (Python).

موجود في القسم أعلاه – انسخه كما هو.

(قالب مختصر) B) README.md

موجود في القسم أعلاه – عدّل الاسم والتعليمات حسب مشروعك. C) LICENSE (MIT)

موجود في القسم أعلاه – بدّل الاسم والسنة.

furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included

all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO **EVENT SHALL THE** 

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN

THE SOFTWARE.

## Create repository 13 (3

بعد اختبار كل الإعدادات، اضغط Create repository.

## 4) ماذا بعد الإنشاء؟

## أ) بدء مشروع جديد من الصفر (محليًا ثم Push)

echo "# student-grade-system" > README.md

git init

git add

git commit -m "init: first commit"

git branch -M main

git remote add origin

https://github.com/USERNAME/student-grade-system.git

git push -u origin main

- ب) لديك مشروع جاهز مسبقًا انسخ رابط الريموت من GitHub (HTTPS) أو SSH)

git remote add origin https://github.com/USERNAME/REPO.git git branch -M main git push -u origin main

## ج) ملاحظة PyCharm

- من VCS  $\rightarrow$  Share Project on GitHub يمكن إنشاء الريبو والرفع مباشرة.
  - تأكّد أن .env والأسرار غير موجودة في الـ commit (موجودة فقط محليًا ومضافة في .gitignore).

# 5) أمان الأسرار (Secrets) بإيجاز

- لا تحفظ الأسرار في الكود.
- استخدم ملفات . env. محليًا (ومُتجاهلة بـ gitignore).
- في GitHub Actions استخدم Repository Secrets بدلًا من تخزين القيم في YAML.

6) أخطاء شائعة ونصائح
 اسم مستودع غامض → غيّره من Settings لاحقًا لاسم واضح.