

1. Architecture en couches

Une architecture en couches permet de séparer les responsabilités et d'améliorer la maintenabilité. Voici les principales couches pour votre projet :

1.1. Couche Présentation (Frontend)

- Responsable de l'interaction avec l'utilisateur.
- Peut inclure une interface web ou une interface utilisateur console.
- **Technologies possibles :**
 - **Web** : Flask/Django (avec HTML/CSS/JavaScript).
 - **Console** : Interface texte en Python.

1.2. Couche Application (Backend)

- Gère la logique métier et les traitements.
- Exemple de responsabilités :
 - Validation des réponses de l'utilisateur.
 - Calcul des scores.
 - Gestion des QCM, questions et réponses.
- **Technologies possibles :**
 - Python avec un framework web comme Flask ou FastAPI.

1.3. Couche Données (Database)

- Gère le stockage et la récupération des données.
- **Contenu :**
 - Utilisateurs : noms, emails, etc.
 - QCM : titres, descriptions.
 - Questions : texte, type (choix unique/multiple).
 - Réponses : texte, booléen (correct ou incorrect).
 - Tentatives et résultats : historique des réponses.
- **Technologies possibles :**
 - SQLite (simple et léger pour les petits projets).
 - PostgreSQL/MySQL (pour des projets plus complexes).

3. Exemple de technologies

Voici un aperçu technologique pour notre projet :

Couche	Technologie	Exemple d'outil
Présentation	Flask / Django	HTML + CSS + JS
API (Backend)	FastAPI / Flask	Python
Base de données	SQLite / PostgreSQL	SQLAlchemy (ORM)