



Project

Advanced programming

2 ème Année Ingénieur

2023-2024



Projet

Advanced programming

Sujet

Application QCM en Python

Réalisé par

DOB Serine

BENTOURI Hasnaa

GUENDOUL Hayat

BENHADDYANoor El Imane

BENSALAH ibtihal



Sommaire

1-Introduction

2-Les deux versions du projet

- Version terminal
- Version graphique

3-Utilisation et explication des fonctionnalités

- Connexion
- Côté Professeur
- *Future vision*

4-la structure de la base de données

5-Conception des interfaces graphiques

- Utilisation de Figma pour le design
- Transition vers Tkinter

6-Ressources et liens



Introduction

Dans le cadre de nos travaux pratiques en programmation Python, nous avons développé une application de Questionnaire à Choix Multiple (QCM) visant à répondre à plusieurs objectifs pédagogiques. Ce projet nous a permis de consolider nos connaissances en programmation tout en explorant des concepts essentiels tels que la gestion des fichiers, l'interaction utilisateur et la création d'interfaces graphiques.

L'objectif principal était de concevoir une application fonctionnelle et intuitive permettant aux utilisateurs de répondre à des questions, de recevoir des retours sur leurs réponses, et de sauvegarder leur progression. Afin d'explorer différentes approches de développement, nous avons réalisé deux versions de l'application :

1. **Une version utilisant le terminal**, mettant l'accent sur l'utilisation des fonctionnalités Python de base, pour une meilleure compréhension des concepts fondamentaux.
2. **Une version avec interface graphique**, conçue avec l'outil Tkinter, pour offrir une expérience utilisateur plus interactive et moderne.

Ce choix de double implémentation nous a permis d'analyser les forces et les limites de chaque approche tout en nous confrontant aux défis pratiques liés au développement d'applications.

Ce rapport présente l'ensemble des étapes du projet, les méthodes utilisées pour concevoir et développer les deux versions, les choix techniques effectués, ainsi que les défis rencontrés et les solutions apportées. Enfin, une réflexion sur les perspectives d'amélioration conclura ce document.



Utilisation et fonctionnalités

1-Connexion et Sécurité des Données :

Lors de l'accès à l'application, l'utilisateur doit se connecter avec ses identifiants. Si l'utilisateur n'a pas de compte, il a la possibilité de créer un nouveau compte via un processus de sign-up. Pour garantir la sécurité et la robustesse des données, une fonction de hachage a été utilisée pour protéger les mots de passe stockés.

Les fonctionnalités affichées après la connexion dépendent du rôle de l'utilisateur : **Professeur** ou bien **Étudiant**

2-Fonctionnalités côté Professeur

1-Affichage des QCM créés :

Le professeur peut visualiser une liste des QCM qu'il a créés, accompagnée d'informations comme le nom du QCM et le nombre de participants.

2-Création d'un nouveau QCM :

Le professeur renseigne :

- Le nom du QCM.
- La catégorie.
- Chaque question et ses réponses associées.
- Indique si chaque réponse est correcte ou incorrecte.
- Une fois le QCM créé, il est affiché pour validation.
- Le professeur peut soit valider directement, soit choisir d'apporter des modifications en accédant aux options suivantes :
- Modifier une question.
- Modifier une réponse dans une question donnée.



3-Fonctionnalités côté Étudiant

1-Répondre à un QCM :

- L'utilisateur choisit une catégorie.
- Une liste de QCM disponibles dans cette catégorie est affichée.
- L'étudiant sélectionne un QCM et commence à répondre.
- Après chaque réponse, l'application indique si la réponse est correcte ou incorrecte.
- À la fin du QCM, l'application affiche :
- Le score final.
- Le temps total passé à répondre au QCM.

2-Voir l'historique des QCM passés :

Cette fonctionnalité affiche une table contenant :

- Le nom des QCM passés.
- Les scores obtenus.
- Le temps passé pour chaque QCM.

4-Fonctionnalités future

1-Gestion des rôles utilisateurs :

Actuellement, l'attribution d'un rôle (étudiant ou professeur) se fait via la ligne de commande. Une future version permettra :

- La transformation d'un compte étudiant en compte professeur via une interface.
- La gestion des utilisateurs (professeurs et étudiants), y compris la possibilité de bannir un compte.

2-Ajout d'un rôle Administrateur

L'administrateur pourra :

- Gérer les comptes utilisateurs (création, suppression, bannissement, modification des rôles).
- Superviser les QCM créés par les professeurs.



La structure de la base de données

Pour ce projet, nous avons choisi d'utiliser SQL pour la gestion des données en raison de ses nombreux avantages, notamment l'intégrité des données, la sécurité et la capacité de gérer des relations complexes entre différentes entités. La base de données est organisée en plusieurs tables pour répondre aux besoins spécifiques du système :

1-Gestion des QCM

- qcm : Contient les informations générales sur chaque QCM (ID, nom, catégorie, créateur).
- qst : Stocke les questions associées à chaque QCM.
- answer : Enregistre les réponses possibles pour chaque question, avec un indicateur si elles sont correctes ou non.

2-Gestion des utilisateurs

- user : Contient les données des utilisateurs (ID, nom, email, rôle, mot de passe haché).

3-Suivi des performances

- qcm_user : Relie les utilisateurs aux QCM qu'ils ont réalisés, avec la note obtenue.
- user_rep : Enregistre les réponses choisies par les utilisateurs pour chaque question.



Conception des interfaces graphiques

1-Utilisation de Figma pour le design

Pour la conception visuelle de l'interface, nous avons utilisé Figma, un outil de design collaboratif, afin de créer des maquettes dynamiques et interactives. Figma nous a permis de concevoir une interface qui répond aux besoins du projet tout en garantissant une expérience utilisateur fluide et moderne. Les maquettes réalisées illustrent l'organisation et l'agencement des éléments visuels que nous souhaitons intégrer dans notre application.

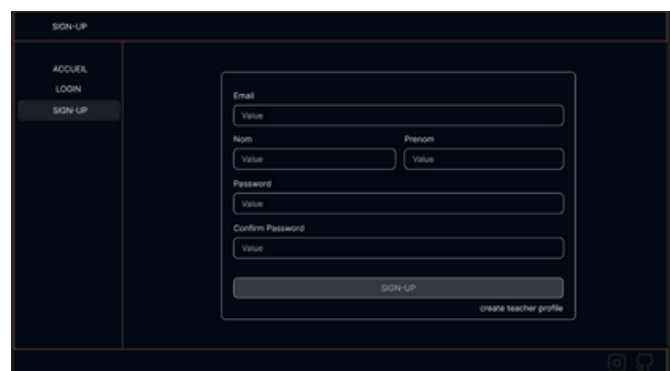
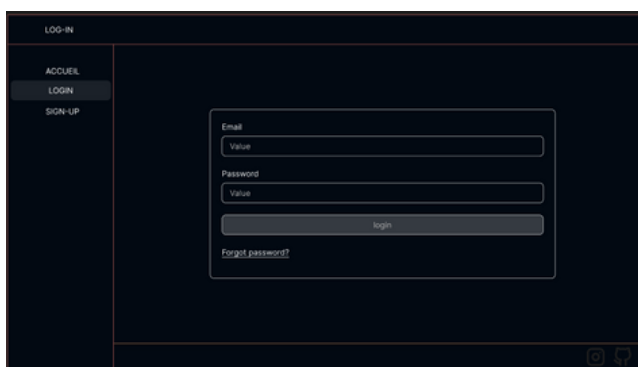
Le lien vers la maquette du design réalisé sur Figma est accessible ici :

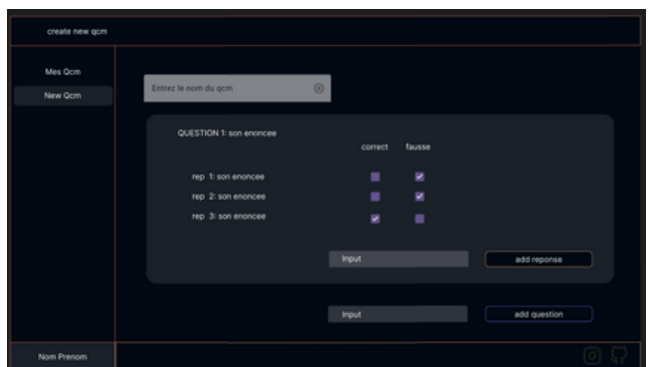
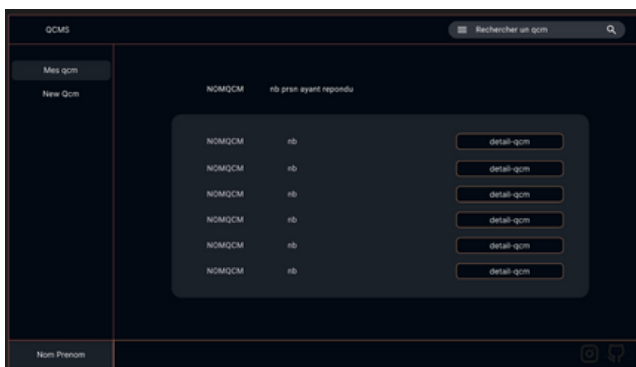
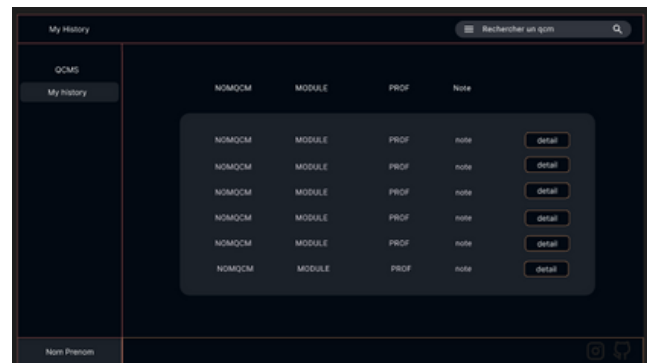
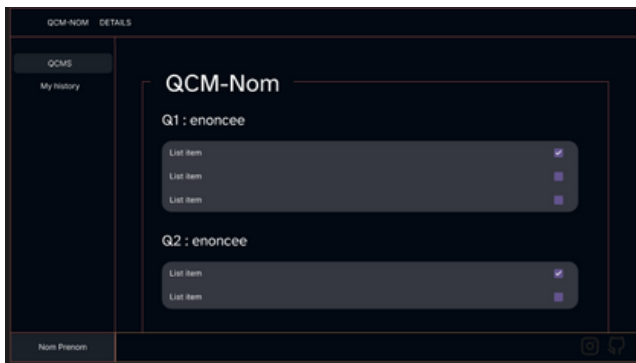
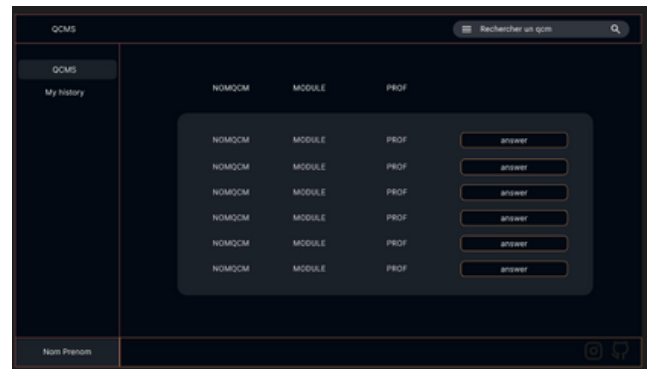
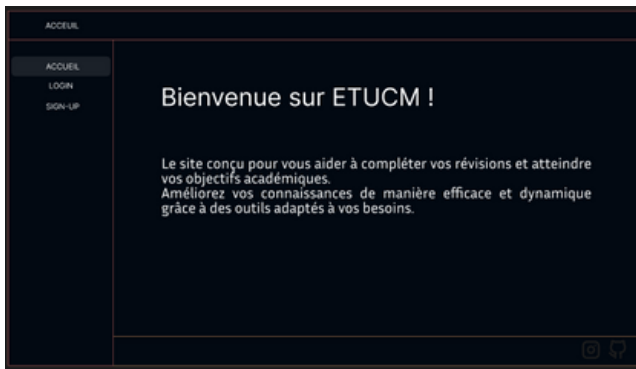
[lien vers design Figma.](#)

Ci-dessous, nous présentons les captures d'écran des différentes pages de l'interface

2-Transition vers Tkinter

Une fois les maquettes créées sur Figma, nous avons procédé à leur implémentation dans notre application en utilisant la bibliothèque Tkinter. Tkinter, étant une bibliothèque standard de Python pour la création d'interfaces graphiques utilisateur (GUI), nous a permis de donner vie aux maquettes en créant des interfaces interactives et fonctionnelles. Grâce à Tkinter, nous avons pu recréer fidèlement les éléments visuels tout en ajoutant les fonctionnalités nécessaires pour rendre l'application dynamique et réactive.





Ressources et liens

Dans cette section, vous trouverez tous les liens pertinents associés à notre projet. Ces ressources sont essentielles pour accéder aux différentes parties du projet, notamment le code source, les maquettes de design, ainsi que les vidéos et documents partagés.

Code source sur GitHub : [Lien vers le repository GitHub](#)

Maquettes de design sur Figma : [Lien vers la maquette Figma](#)

Vidéo de présentation du projet : [Lien vers la vidéo](#)

(si lien a la video fonctionne pas vous la trouverez sur drive)

Documents et ressources partagées sur Google Drive : [Lien vers Google Drive](#)