Université des Sciences et de la
Technologie Houari Boumediene Faculté
d'Informatique

RAPPORT APPLICATION DE QCM

Examen TP Programmation avancée

Réalisé par:

- **Tiril Souhila**
- **Layadi Meriem**
- **Ghersi Meriem**

Introduction :

Ce projet consiste à développer une application de QCM (Questionnaire à Choix Multiples) en Python, destinée aux étudiants en informatique. L'application permet aux utilisateurs de répondre à des questions, d'évaluer leurs réponses, et de consulter leur historique de scores. L'interface est conviviale et basée sur une console, avec une gestion des utilisateurs et des questions via des fichiers Excel.

Objectif du projet :

- Permettre aux utilisateurs de répondre à des QCM.
- Évaluer les réponses et afficher un score.
- Offrir une interface conviviale en console.
- Gérer les utilisateurs et leur historique de scores.

Fonctionnalités Implémentées :

Gestion des Utilisateurs :

- Enregistrement des utilisateurs : Les utilisateurs peuvent se connecter avec leur nom. Si l'utilisateur n'existe pas, il peut en créer un nouveau.
- Historique des scores : L'application enregistre l'historique des scores pour chaque utilisateu dans un sheet dédiée (history), avec la date et le score obtenu.
- Stockage des données : Les informations des utilisateurs sont stockées dans un fichier Excel (qcm_data.xlsx), dans une feuille dédiée (users).

Gestion des Questions et Réponses :

- Stockage des questions : Les questions et leurs options sont stockées dans une feuille Excel (questions).
- Validation des réponses : L'application vérifie les réponses des utilisateurs et affiche un feedback immédiat (bonne ou mauvaise réponse)

- Calcul du score : Le score final est calculé en fonction du nombre de bonnes réponses.

Interface Utilisateur :

- Affichage de l'historique : l'affichage de la date le score et le thème du qcm

- Affichage des questions : Les questions sont affichées une par une, avec des options de réponse sous forme de boutons radio.

- Feedback immédiat : Après chaque réponse, l'application indique si la réponse était correcte ou incorrecte.

Fonctionnalités Avancées (Optionnelles) :

- Timer :Un compte à rebours a été implémenté pour afficher le temps restant en minutes et secondes. La boucle while met à jour l'affichage chaque seconde en utilisant divmod pour convertir le temps restant en format MM:SS. Une fois le temps écoulé, un message "Temps écoulé !" est affiché.
- Thèmes de questions : Les questions sont organisées par thèmes grâce a une colonne ajoutée dans le fichier excel des questions (Python, Réseaux, Algorithmes, etc.).
- Exportation des résultats : Les résultats peuvent être exportés dans udeux formats (CSV et TXT) et propose des boutons de téléchargement via Streamlit.

Technologies Utilisées :

- Language de programmation : python.
- Bibliothèques Python :
  - ✔ **Pandas** : Nous avons utilisé Pandas pour charger et manipuler facilement les données des utilisateurs, des questions et de l'historique depuis le fichier Excel (qcm_data.xlsx). Les DataFrames de Pandas simplifient le traitement des données structurées, ce qui nous permet de gérer et de filtrer les informations de manière efficace.

- ✔ **Streamlit** : Nous avons utilisé Streamlit pour créer l'interface utilisateur de notre application. Grâce à sa simplicité d'utilisation, nous avons pu concevoir rapidement un QCM interactif où les utilisateurs peuvent répondre aux questions, voir leur score en temps réel, et avoir une expérience fluide. Streamlit nous permet d'intégrer facilement des composants comme des boutons, des zones de texte, des graphiques, et des chronomètres, le tout sans nécessiter de code HTML ou JavaScript complexe.
- ✔ **Datetime** : Pour gérer le temps (timer) et enregistrer la date des QCM.
- ✔ **Time** :our mesure Pr le temps écoulé pendant le QCM et gérer les timers associés à chaque question.

- **Stockage des Données dans un Fichier Excel :** Les données sont stockées dans un fichier Excel (qcm_data.xlsx) avec trois feuilles :
  - **users** : Pour les informations des utilisateurs.
  - **history** Pour l'historique des scores.
  - **questions** : Pour les questions et les réponses.

Défis et solutions :

- **Gestion de l'état de l'application** : Nous avons rencontré des difficultés pour maintenir l'état de l'application (score, questions, etc.) au fur et à mesure des interactions. Cependant, nous avons appris à utiliser efficacement les *session_state* de Streamlit, ce qui nous a permis de garder les données entre les différentes étapes du QCM, offrant une expérience utilisateur fluide.
- **Intégration des Chronomètres et du Timer** :Gérer le temps tout en maintenant la pression des examens a été un défi. Nous avons appris à intégrer des timers pour chaque question, tout en offrant une limite de temps réaliste, mais cela nous a appris à mieux gérer le contrôle du temps dans les applications interactives.
- **Gestion du Temps sous Pression des Examens :** Le projet coïncidait avec nos examens, ce qui a rendu difficile la gestion du temps. Nous

avons appris à prioriser les tâches et à travailler sous pression pour respecter les délais.