

# **Université des Sciences et de la Technologie Houari Boumediene**

## **Faculte d'informatique**



### **Rapport de Projet**

## **Développement d'une Application de Quiz en Pyhton**

Alouache Manel

Aramis Younes

Aziez Hichem Racim

Saheb Med Aymen

## 1. Introduction

Ce rapport technique présente une analyse approfondie du développement d'une application de quiz écrite en Python. L'objectif principal de cette application est de permettre aux utilisateurs de tester leurs connaissances dans divers domaines de l'informatique, tout en leur offrant la possibilité de suivre leurs progrès grâce à un système de suivi des résultats. L'application a été conçue pour être à la fois simple d'utilisation et facilement extensible, afin de pouvoir intégrer de nouvelles fonctionnalités ou catégories de questions sans nécessiter de modifications majeures du code existant.

## 2. Structure du Code

L'application est structurée en deux fichiers principaux, chacun jouant un rôle distinct dans le fonctionnement global du quiz :

- **Questions.py** : Ce fichier constitue le cœur de l'application. Il contient toute la logique métier, y compris la gestion des questions, la validation des réponses, le calcul des scores, et la persistance des données utilisateur. Il est conçu pour être robuste et facilement maintenable, avec une architecture modulaire qui permet d'ajouter de nouvelles fonctionnalités sans perturber le code existant.
- **qst.py** : Ce fichier représente une version simplifiée de l'application, servant de prototype initial. Il a été utilisé pour valider les concepts de base et tester les fonctionnalités essentielles avant de les intégrer dans le fichier principal. Bien que moins complet que Questions.py, il a joué un rôle crucial dans le processus de développement en permettant une itération rapide et une validation précoce des idées.

## 3. Stockage des Données

Le choix du format de stockage des données est un aspect crucial de toute application. Pour ce projet, le format **JSON** a été retenu en raison de ses avantages en termes de simplicité et de lisibilité. JSON permet de structurer les données de manière hiérarchique, ce qui est particulièrement adapté pour organiser les questions par catégories.

- **Structure Hiérarchique des Questions** : Les questions sont organisées en catégories, chacune contenant une liste de questions, les options de réponse associées, et les réponses correctes. Cette structure permet une gestion aisée des questions et facilite l'ajout de nouvelles catégories ou questions sans nécessiter de modifications complexes du code.
- **Persistence des Données Utilisateur** : Les données des utilisateurs, telles que leur historique de sessions de quiz et leurs scores, sont stockées dans un fichier `user_data.json`. Ce fichier est mis à jour à chaque session, permettant ainsi de conserver l'historique des résultats et de suivre les progrès des utilisateurs sur le long terme.

## 4. Organisation des Questions

L'organisation des questions est un élément clé de l'application. Chaque catégorie de questions est structurée de la manière suivante :

- **Liste des Questions** : Chaque catégorie contient une liste de questions, chacune étant identifiée par un identifiant unique et un libellé clair.
- **Options de Réponse** : Pour chaque question, plusieurs options de réponse sont proposées à l'utilisateur. Ces options sont présentées de manière claire et concise, afin de faciliter la prise de décision.
- **Réponses Correctes** : Les réponses correctes sont stockées séparément, permettant à l'application de valider les réponses de l'utilisateur et de calculer son score de manière précise.

Cette structure rend l'application facile à maintenir et à étendre. L'ajout de nouvelles catégories ou questions se fait de manière intuitive, sans nécessiter de modifications complexes du code existant.

## **5. Gestion des Utilisateurs**

La gestion des utilisateurs est un autre aspect important de l'application. Elle permet de personnaliser l'expérience de chaque utilisateur et de suivre ses progrès au fil du temps.

- **Identification par Nom** : Chaque utilisateur est identifié par son nom, ce qui permet de personnaliser l'expérience et de suivre les résultats de manière individuelle.
- **Historique des Sessions de Quiz** : L'application conserve un historique des sessions de quiz pour chaque utilisateur. Cet historique inclut les scores obtenus, les catégories de questions abordées, et les réponses fournies. Cela permet à l'utilisateur de visualiser ses progrès et d'identifier les domaines où il peut s'améliorer.
- **Persistence des Scores** : Les scores des utilisateurs sont persistants, ce qui signifie qu'ils sont conservés d'une session à l'autre. Cela permet aux utilisateurs de suivre leurs progrès sur le long terme et de se fixer des objectifs d'amélioration.

## **6. Interface Utilisateur**

L'interface utilisateur (UI) de l'application a été conçue pour être simple, intuitive et conviviale. Elle joue un rôle crucial dans l'expérience globale de l'utilisateur.

- **Affichage Clair des Questions et des Options** : Les questions et les options de réponse sont présentées de manière claire et lisible, afin de faciliter la prise de décision. L'interface est conçue pour minimiser les distractions et permettre à l'utilisateur de se concentrer sur le contenu du quiz.
- **Validation des Entrées Utilisateur** : L'application inclut un système de validation des entrées utilisateur, afin de s'assurer que les réponses fournies sont valides et conformes aux attentes. Cela permet d'éviter les erreurs et de garantir une expérience utilisateur fluide.
- **Feedback Immédiat sur les Réponses** : Après chaque réponse, l'application fournit un feedback immédiat, indiquant si la réponse était correcte ou incorrecte. Cela permet à l'utilisateur de comprendre rapidement ses erreurs et d'apprendre de ses réponses.

## **7. Défis Rencontrés**

Le développement de cette application a présenté plusieurs défis techniques, notamment en ce qui concerne la gestion de l'historique des utilisateurs et la persistance des données.

- **Maintenir l'Historique des Utilisateurs** : L'un des principaux défis a été de concevoir un système efficace pour maintenir et mettre à jour l'historique des utilisateurs. L'utilisation du format JSON pour le stockage des données a permis de surmonter ce défi, en offrant une structure de données optimisée pour les mises à jour fréquentes.
- **Optimisation de la Structure des Données** : La structure des données a été soigneusement conçue pour permettre des mises à jour rapides et efficaces, tout en garantissant la cohérence et l'intégrité des données. Cela a nécessité une réflexion approfondie sur la manière d'organiser les informations pour minimiser les temps d'accès et de traitement.

## **8. Conclusion**

En conclusion, cette application de quiz en Python atteint avec succès ses objectifs principaux tout en restant extensible et facile à maintenir. La structure modulaire du code permet d'ajouter de nouvelles fonctionnalités ou catégories de questions sans nécessiter de modifications majeures du code existant. L'utilisation du format JSON pour le stockage des données a permis de créer une application à la fois simple et performante, capable de gérer efficacement les données des utilisateurs et de fournir une expérience utilisateur fluide et agréable.

Ce projet a également permis de mettre en lumière l'importance d'une bonne conception de la structure des données et de l'interface utilisateur dans le développement d'applications interactives. Les défis rencontrés ont été surmontés grâce à une approche méthodique et une attention particulière portée à la qualité du code et à l'expérience utilisateur.

Enfin, ce projet ouvre la voie à de futures améliorations, telles que l'intégration de fonctionnalités supplémentaires (par exemple, des quiz chronométrés, des classements, ou des recommandations personnalisées), ou l'extension de l'application à d'autres domaines de connaissances. La base solide établie par ce projet permettra de poursuivre son développement de manière efficace et structurée.