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

Année 2024-2025

PROJECT REPORT –TEAM invictus

PROJECT ADVANCED PROGRAMMING

QCM-APPLICATION

Project carried out by :

DENNI Ryad
GHAOUI Wail
KECIRA Abderraouf
SIFI Tarek Mazigh

Project supervised by :

M. MOUHOUN SAID

SUMMARY

Table of contents

Introduction : **3**

Implemented Features: **3**

Technical Choices : **4**

Challenges Encountered and Solutions : **5**

Results Obtained : **5**

Conclusion : **5**

Thanks : **5**

Introduction :

As part of the Python Programming course, we were tasked with developing a MCQ (Multiple Choice Questionnaire) application for computer science students. The main objective of this project was to create a Python application that allows users to answer MCQs, evaluate their answers, display a score, and manage users with a score history. This report presents the technical choices, the challenges encountered, the solutions provided, and the features implemented in our application.

Implemented Features:

Our application meets the project requirements by offering the following features:

1. User Management:

- Users can login or register with a username and password.
- User information is stored in a JSON file (Login.json).
- Existing users can view their MCQ history, including scores and test dates.

2. Management of Questions and Answers:

- Les questions sont stockées dans un fichier JSON (Questions.json) avec des options de réponse et une réponse correcte.
- Les utilisateurs peuvent choisir une catégorie de questions (par exemple, Physique, Mathématiques, Informatique) et un niveau de difficulté (facile, moyen, difficile).

3. Evaluation and Feedback :

- After each MCQ, the application displays the user's final score.
- Feedback is given for each question, indicating whether the answer was correct or incorrect. If the answer is incorrect, the correct answer is displayed.

4. Timer :

- A timer is implemented to limit the response time per question or for the entire MCQ, depending on the chosen difficulty level.

5. Exporting Results :

- Users can export their results to a text or CSV file to save their scores.

6. User Interface:

- The application offers a user-friendly console interface, allowing users to easily navigate between different options (login, registration, category selection, etc.).

Technical Choices :

1. Data Structures :

- Questions and answers are stored in a JSON file (Questions.json), allowing easy and modular data management.
- Users and their histories are stored in a JSON file (Login.json), allowing data persistence between sessions.

2. File Management :

- The application uses Python's json module to load and save user and question data.
- Exporting results is handled via the csv and io modules to create text and CSV files.

3. Loops and Conditionals:

- For and while loops are used to iterate through questions, check answers, and handle user interactions.
- If-else conditions are used to validate user input and handle different use cases.

4. Functions :

- The application is divided into several modular functions, such as register(), login(), play_quiz(), and export_results(), which makes the code easier to maintain and understand.

5. Validation of Entries :

- The application checks that user inputs are valid (e.g. answers must be among the options provided, passwords must match during registration).

Challenges Encountered and Solutions :

1. Data Persistence :

- Handling JSON files to store users and their histories required special attention to avoid read/write errors. We used try-except blocks to handle potential errors.

2. User Interface :

- Creating a user-friendly console interface required a clear organization of menus and options. We used loops and conditions to allow users to navigate easily.

3. Exporting Results :

- Exporting results in different formats (text and CSV) was a technical challenge. We used the csv and io modules to handle this functionality.

Results Obtained :

The app works as expected, providing a smooth and intuitive user experience. Users can log in, choose a question category, answer multiple choice questions, and view their history. The timer adds a dimension of challenge, and exporting results allows users to save their performance.

Conclusion :

This project allowed us to practice Python programming concepts, including file management, data structures, loops, conditionals, and functions. We also learned how to handle technical challenges such as data persistence and time management.

We are satisfied with the final result and hope that this application will be useful for computer science students to test their knowledge in an interactive way.

Thanks :

We would like to thank Mr. MOUHOUN SAID for his support and advice throughout this project.