Engineering-4 Group-4 Project

Anne-Marie, Hayley, Megan, Sam, Sara, Stella

Educational Trivia Game

August, 2024

Introduction

Aims and Objectives

This project is about creating a fun and interactive learning game for students who may not be able to attend class due to mental health constraints. It takes the form of a multiple-choice trivia game that covers various curriculum topics for primary school students. It incorporates the use of media to create variation, a sense of enjoyment and achievement through promoting positive engagement to boost the well-being of young people. The objective is to limit the damage that testing can do to the mental health of youth, by making learning fun though an engaging learning platform that is relevant to the curriculum of primary school students. The questions provided in the game platform are age-appropriate and designed to promote self-esteem and boost confidence in students, since they will have the ability to level up as they tackle different quizzes on the game console.

Roadmap of Report

This report is divided into four sections. The first section covers the background of the project, outlining the Educational Trivia Game rules and logic. The second section discusses the project specifications and design, detailing the functional and non-functional requirements along with the project design and architecture. The third section examines the overall implementation and execution of the project including the development approach. The final section provides the analysis drawn from testing and evaluation of the game, reporting the testing strategies and system limitations. The information highlighted in each section contributes to a comprehensive conclusion on the Educational Trivia Game project.

Background

The console app retrieves data about different educational trivia questions, including their subject name and id, difficulty level and an array of the correct and incorrect answer choices.

Students will be able to select their subject choice with questions from the current subjects provided: *English* and *Math*, and then select their difficulty level from the options 'easy', 'medium' or 'hard'. A set of 10 questions for their selected quiz choice will then be generated, and each question has a 30 second timer within which the student will need to provide an answer *A, B, C* or *D*.

Student answers are recorded in a file for the student to be able to track their progress in the app at the end of each question set for further review. At the end of the game, students can see their total score in the console. Points are awarded for correct answers based on the difficulty level and the time taken to answer, whilst incorrect answers gain no points and instead feedback is given on the correct answer. Below is a demonstration of the score key:

Easy: 1-5 points

Medium: 3-7 points

Hard: 5-10 points

Students can earn achievements for completing subjects, reaching score milestones or for their answering streaks.

Specifications and Design

Functional vs Non-Functional Requirements

Functional Requirements

The functional requirements for the Educational Trivia Game are listed below:

- User authentication system for students and teachers, incorporating provision of a username and password to log into the user account
- Quiz selection via console input
- Text-based presentation of questions during game-play
- Thirty-second time limit given for each question to be answered
- Multiple-choice answer selection via console input
- Immediate feedback given to the user on answers during game-play
- Score calculation and tracking during game-play
- Locally stored progress tracking across multiple game sessions
- Teacher functionality for adding custom quiz questions via console input
- Data management system for administrators to manage user account information
- Results and score generation system for teachers to track student progress

Non-Functional Requirements

The non-functional requirements for the Educational Trivia Game are listed below:

- Scalability to allow the system to accommodate an increase in user load
- Usability to ensure a user-friendly interface with intuitive navigation
- Maintainability to facilitate easy maintenance and updates on the system codebase
- Security to protect the privacy of sensitive user data through encryption

Design and Architecture

User Interface (Frontend)

The project currently provides a Python console application with Flask integration for students to play educational trivia games. It allows teachers to create, manage and monitor educational quizzes, and serves as a portal for parents to monitor their child's learning progress. The developmental aim is to switch to a web-based application in the future, possibly incorporating JavaScript, CSS, HTML and React.js for a more dynamic, visual and responsive user experience.

Application Server (Backend)

The application server is built with Flask in Python and it handles user authentication and authorisation, manages game logic and scoring, processes teacher inputs for quiz creation and management. Additionally, it provides APIs for game data and user interactions.

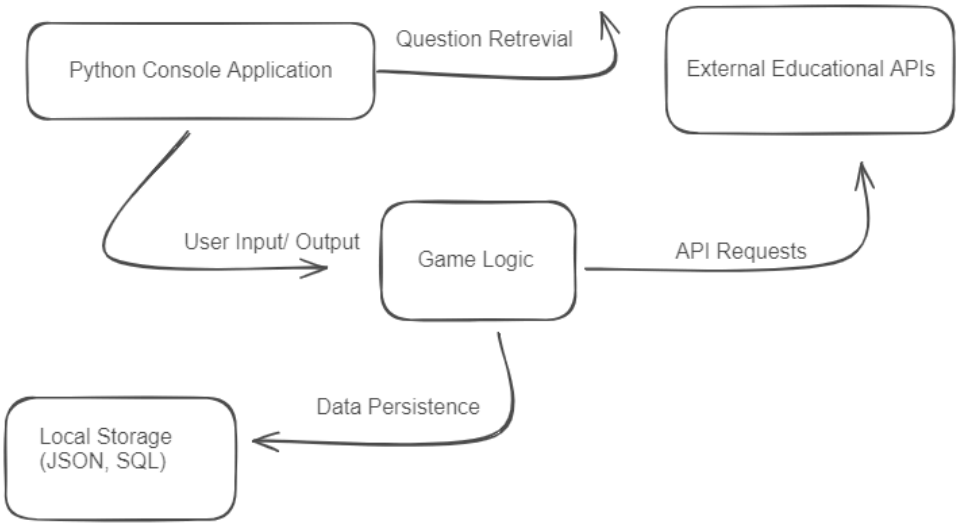
Databases

The project facilitates data management by incorporating the use of MySQL databases to store user data, namely that of students, parents and teachers. Separate databases are also used for keeping track of quizzes, questions and lastly for maintaining records of student progress, achievements and scores.

Analytics and Reporting

The project aims to provide real-time tracking of students' progress, and quiz results generation for teachers and parents to gain insight into performance and engagement.

Architecture Diagram



Implementation and Execution

Development Approach and Team Member Roles

This project was developed using a Kanban development approach. Excalidraw was used as the initial visual workflow management tool, followed by Trello as the project management tool of choice. 'Backlog', 'Tickets', 'Working On', 'Reviewing' and 'Done' cards were created on the shared Trello board, and each team member assigned themselves to different tickets throughout the development of the project. Collaboration was maintained through regular reviewing of team member tickets and feedback provision.

Tools and Libraries

Tools

- Git & GitHub for version control
- PyCharm and Visual Studio Code as the Integrated Development Environments
- Trello and Excalidraw for project and visual workflow management respectively
- Unittest for project testing

Libraries

- MySQL for database management
- Python getpass and hashlib for authentication and security

Implementation Process

The team was able to successfully implement most of the initial planned features such as the Student, Teacher and Parent user stories, along with the main functional requirements such as user authentication, progress tracking, multiple choice question display with timer, and score calculation and tracking.

The challenges faced included building up the project with enough time to be able to upgrade it to a web-based application and implementing media as part of the question options. As a result, the design plan had to be changed to maintain a terminal-based console app that implements only text-based questions. Further challenges included successfully integrating the code into a working project that could work consistently across multiple devices.

Agile Development

The group implemented code integration between different code versions during the testing and file combination stage. Code reviews were given amongst group members that participated in running the files, and peer-programming was implemented to troubleshoot during different stages of the project building.

Testing and Evaluation

Testing Strategy

Testing was done throughout the project development, mainly to identify bugs and ensure optimal performance and security. Further details are outlined below.

Functional and User Testing

Functional

During the individual file creation, tests were run on the different functions using python logging, error catching and print statements. Unit tests were created to test the final completed functions in the app.py, game.py, main.py, question.py and user.py files.

User Testing

Group members pulled the files from the GitHub testing branches to test the different user functionality on their individual devices.

System Limitations

The inbuilt Python getpass did not work on some of the group member devices, and needed to be replaced with a regular input statement for testing purposes. A few database connectivity issues were also faced by some group members during the testing stages. Integrating code between the different testing branches helped to mitigate some of the system limitations.

Conclusion

The Educational Trivia Game project successfully created an engaging learning platform for primary school students, particularly those facing mental health challenges. Despite some challenges, such as transitioning from a planned web-based platform to a console-based application, the project met its objectives by implementing key features like user authentication, score tracking, and interactive quizzes. Testing ensured functionality, and the project lays a strong foundation for future enhancements.