UNIVERSITY OF LEICESTER

School of Computing and Mathematical Sciences

CO7201 Individual Project

Interim Report

An AI Multiple-Choice Programming Exercises Tutor

Sai Raghava Vemuri

Srv10@student.le.ac.uk

249034443

Project Supervisor: Dr. Wentao Li
Principal Marker: Dr. Martina Palusa

Word Count: 657
25-07-2025

Contents

1. Overview:

This interim report outlines the progress made on the project titled "An AI Multiple-Choice Programming Exercises Tutor " The project aims to develop an intelligent system capable of dynamically generating and evaluating MCQs based on user performance, enhancing personalized learning experiences.

Overall, approximately 75% of the system functionality has been implemented. The core backend pipeline (fetcher → generator → validator → adaptive engine) is complete, and a working Flask-based web interface has been integrated. Users are able to answer questions interactively, with bonus scoring, live stats, and adaptive difficulty implemented. Remaining work involves finalizing user dashboard and profile features, and deployment.

2. Progress

2.1 Essential:

These are the main features required for the core functionality of the quiz application. These were initially developed and tested in the Command Line Interface (CLI) version for ease of testing. The progress made:

Requirement	Description	Progress
Dataset creation	Created curated datasets of 100 high-quality, difficulty level and language labelled MCQs foreach of three programming languages: Python, Java, and C	Done
MCQ Presentation	The dataset and also Ai generated questions are in JSON format so created a reliable method to display it to the users in readable text.	Done
Answer Checking	Implemented answer checking method, which takes User inout and validate it against the correct answer.	Done
Questions Generation	Developed AI-based generator.py using example-based prompts to produce contextual MCQ [questions, options and correct option] and Explanation.	Done
AI Explanation		
Adaptive Difficulty	Built the adaptive engine that adjusts difficulty level based on user performance and recent answers	Done

2.2 Recommended:

These mainly focus on the User Interface and experience.

Requirement	Description	Progress
Web Interface	Implemented web interface using Flask to present quiz and receive answers	Done
User Profiles	User profile system to track scores and progress across sessions is currently under development.	Work in Progress
Log incorrect	Store incorrect responses for future review.	Done

2.3 Optional

These are advanced, innovative features that can significantly enhance the interactivity of the system.

Requirement	Description	Progress
Reward System	Integrated a timer-based bonus scoring mechanism (bonus for answers within 30 seconds) in addition to the usual scoring.	Done
Personalized Dashboard	Dynamic personalized dashboard to visualize progress.	To Do
Chat Bot	Allows users to follow up and understand topics interactively.	To Do

3. Changes:

- While the core architecture has remained consistent, a few scheduling changes were made for workflow efficiency
- The scoring system and gamification features were started earlier than originally planned to help visualize quiz flow and improve testability.
- As a result, user profile/dashboard implementation has been shifted to a later week.

4. Challenges Faced

- Dataset creation and difficulty labelling: Designing high-quality MCQs across three languages and tagging them by difficulty has been a bit time-consuming.
- AI hallucinations: Some generated questions had incorrect answers or formatting, which led to the development of a dedicated AI validator
- Testing adaptive logic: Ensuring the generator and validator responded accurately to changing user input required careful testing through CLI and UI.

5. Timeline Comparison: (Till now)

Milestone	Planned Completion	Actual Completion
Dataset Creation	Week 2	Week 2
Fetcher + Generator	Week 3	Week 3
Validator	Week 4	Week 4
Adaptive Engine	Week 5	Week 5
Flask UI	Week 6	Week 6
Gamification Features	Week 8	Week 7 (Early)
Profile	Week 7	Moved to Week 8

Dashboard	Week 9	To Do
Implement other Optional Requirements	Week 10	To Do
Refine Complete Application	Week 11	To Do
Evaluation, Bug Fixing, Peer Testing and feedback		
Final Documentation & Supervisor Review	Week 12	To Do

What's Left

- Finalize user dashboard and profile handling.
- Other optional Features, Cloud deployment and real-user testing.
- Final documentation, UI polishing, and report preparation.