

Multiple Choice Online system

This project is a **Python-based client-server quiz system**. Users can log in, take a 5-question quiz, and view their scores and the scoreboard. It includes both **CLI** and **GUI (Tkinter)** clients.

Tools and Frameworks

- **Python** , `socket`, `threading`, `pickle`, `tkinter`
- **Persistent data** stored in `users.pkl` using `pickle`
- **TCP/IP** for client-server communication

Features

- User login with password check
- Random quiz generation
- Score calculation and persistent storage
- Scoreboard showing highest scores
- GUI with login, quiz, and result screens

Task Division

- **Hany Osama & Abdelrahman Atef** : Server logic (login, quiz, scoring)
- **Abdulrahman Ali & Hassan Ahmed** : CLI & GUI client (Tkinter)
- **Hany Osama & Abdelrahman Atef**: Testing & integration

Brief Description of Classes and Functions

QuizClient (in GUI)

- **Purpose:** Manages the GUI flow: login, quiz, and results.
- **Key Methods:**
 - **create_login_screen():** Displays login interface.
 - **handle_login():** Sends login info to the server and receives quiz.
 - **show_quiz_screen():** Displays each question and choices.
 - **submit_quiz():** Sends answers, receives results and scoreboard.

Server Functions

- **handle_client():** Main function handling each client connection. It authenticates users, sends quiz, receives answers, calculates scores, and returns results.
- **generate_quiz():** Randomly selects 5 questions.
- **grade_quiz():** Compares user answers with correct answers and calculates the score.
- **scoreboard():** Returns the name(s) of the top scorer(s).
- **load_users() / save_users():** Handle persistent user data using **pickle**.

Client (CLI)

- Simple script that connects to the server, handles login, quiz answering, and displays result in the terminal.