**Python Quiz Application**

## **Abstract**

## The application is an interactive command-line quiz for students and learners to test their knowledge in various subjects such as DBMS, DSA, and Python. The user registration and login mechanism ensure that all quizzes have scores tracked securely. Every quiz consists of multiple-choice questions that are maintained in different text files for each subject. User credentials and quiz results are also maintained in secure text files. The project is modular, robust, and uses Python file handling for data management.

## **Functionality Overview**

**1.Register  
• Purpose:** Allows new users to create an account by providing a username and password.  
**• Implementation:**  
o Appends the username-password pair to users.txt.  
o Ensures new credentials are stored securely.  
**• Working:**o Accepts input for a new username and password.  
o Writes the credentials to a file for future login. **2. Login  
• Purpose**: Authenticates users to access the quiz.  
**• Implementation:**o Reads from users.txt to validate credentials.  
o Stored data is split using a delimiter (-) and compares inputs.  
**•Working:**o Asks for a username and password.  
o Allows access if the credentials are correct; otherwise, denies login.  
**3. Load Questions  
•Purpose:** Reads questions from a text file dependent upon a subject.  
**•Implementation:**o Loads questions, options, and correct answers from the subject file (DBMS.txt, DSA.txt, Python.txt).  
o STATES that an erroneous entry has been passed.  
**•Working:**o Opens the file relating to the subject.  
o Extracts and structures question data into a list.  
**4. Attempt Quiz  
• Purpose**: Facilitates quiz-taking for logged-in users.  
**• Implementation:**o Calls the load\_questions() function for the selected subject.  
o Tracks correct answers and calculates the score.  
**• Working:**o Displays questions and options to the user.  
o Verifies answers and updates the score.  
o Calls update\_result() to save the score. **5. Update Result  
• Purpose**: Stores or updates the highest score of a user in results.txt.  
**• Implementation:**o Reads existing results from the file.o Updates the score if the new score is higher.  
**• Working:**o Reads the results.txt file.  
o Updates or writes the user's highest score back to the file.  
**6. Show Result  
• Purpose:**Prints out the highest score of a logged-in user. **• Implementation:**o Reads from results.txt.  
o Matches the username to get the score.  
**• Working:**o Searches for the user's score in the results file.  
o Shows the highest score or "No result found".

**Program Workflow  
1.Main Menu:**o Options: Register, Login, Exit.  
o  Based upon the user's choice, routes him.  
**2.Logged-in User Menu:**

o Options-Attempt Quiz, Show Result, Logout.Calls the appropriate functions based on the user's selection.  
**3.Quiz Execution:**

o Prompts the user to choose a subject.  
o Based upon the number of questions for each file, reads the questions one by one and accepts an answer.  
o Calculates and stores the score.

**Data Structure and Storage**

#### **Program Memory :**

**1. Users and Passwords**

* **users:** A list to store registered usernames.
* **pwd:** A dictionary mapping usernames to passwords.

**2. Scores**

* **scores:** A dictionary with nested dictionaries to store subject-wise scores for each user.

**3. Quizzes**

* **quizzes:** A dictionary containing subject names as keys and question sets as values.

**File handling :**• **Users:** Stored in users.txt as username-password  
• **Questions:** Stored in subject-wise files(DBMS.txt, DSA.txt, Python.txt ) in the format:  
• Question 1  
• A. Option 1  
• B. Option 2  
• C. Option 3  
• D. Option 4  
• **Answer:** A  
• **Output:** Saved in results.txt with the format: username-score

**Benefits  
•**Modular design for maintainability.  
•  Straightforward text file processing for data storage.  
•  Question banks are therefore subject-representative and customizable.

**Drawbacks  
•**No encryption of stored passwords.  
•  Only intended for a command-line interface.

**Future Development  
•**Implement encryption of user credentials.  
•  Provide a GUI for improved usability.  
•  Enable feedback to be given real-time during the quiz.