**Synopsis**



**Quiz Management System**



Project Id: 05

Student Name: Sukhmani Kaur Ahluwalia

University Roll No.: 2025505

Section: L

Class Roll No.: 18

* **Problem Description:**

Create a quiz management system using C programming language that enables users to add questions, conduct quizzes, and assess results. Participants should be able to take quizzes, answer multiple-choice questions, and get their scores. The system must keep a record of questions, monitor participants' performance, and allow users to retrieve results by their names.

The quiz management system has following applications in real life:

1. It is used to conduct online exams and quizzes, along with applying a time constraint to finish it, which allows a convenient way to assess student’s knowledge.
2. Corporate sectors use them to assess employee’s knowledge and his capability to work in their company.
3. Educational institutions use them to deliver engaging quizzes to the students, having various levels, which helps the student to learn and apply knowledge in an enthralling way.
4. Companies use them to gather customer feedbacks and information.
5. These are used in game development to implement fun and engaging quizzes for players.

* **Roadmap:**

User

Yes

Creates his own username and password.

New User

Authenticate Function is invoked

Quiz Management System

No

User enters his credentials

His credentials are stored in secured file

Login

The input credentials are matched with his created (stored) credentials.

Program Terminated

No

Match Found

Yes

Allows the user to add questions using add question function and retrieve quiz results by name using result retrieval function.

Program Terminated

Allows user to answer multiple choice questions by invoking conduct function, calculates his score with the help of result evaluator function and displays it.

* **Proposed Modules:**

1. User Authentication (Authenticate Function): It will ask the user to create his own username and password for logging in to ensure secure access and store it in a secure file. Whenever the user will login, it will compare the inputted credentials with the created/stored login credentials of the user and block access if unmatched or any of the credentials are not inputted.
2. Quiz question adder function: It will be invoked when the user chooses to add questions in the quiz. It will allow user to input questions to be added and append it to the respective text file (named questions) which will store the quiz questions.
3. Quiz Attempt (Attempt()) Function: It will be invoked when the user chooses the quiz attempt option. It will allow the user to answer multiple choice questions by inputting the answer number (a,b,c or d) to the questions which will be read from the file storing quiz questions and all the user’s answers will be stored in an array.
4. Quiz Conduction (Conduct()) Function: It will be invoked when the user chooses quiz conduction option. It will give choice to user to add questions using question adder function and to retrieve results of all or any specific participant using result retrieval function.
5. Result Evaluator Function: It will be invoked at the end of quiz attempt function. It will receive character array of user’s answers as parameter and compare it with the correct answers stored in the respective file of quiz’s answers (named correct answers). For each correct answer, the value of counter variable, which will be initialized by zero at the beginning, will be updated by 1 and the final value of counter variable after checking all the answers will be displayed as the user’s score.
6. Participant Performance Tracker (and Record Handling) Function: It will be invoked after result evaluation function. It will receive the name and the score of the participant as parameters. It will determine the participant’s performance remark as outstanding, satisfactory, etc. by comparing the participant’s score against specified conditions. After that, it will create a structure named user which will have two char array variables, name and remark and an integer variable named score. After this, it will append the values of all these variables to the respective file storing the participant’s information (named participant\_info).
7. Result Retrieval Function: It will allow the user (who will conduct the quiz) to retrieve the results and performance remarks of all the participants and even the result and remark of any specific participant by his name.
8. String Manipulation Module: This section will contain all the string functions necessary for working on strings such as copy (to copy one string to other string), compare (to compare two strings and find whether they are equal or not) and length (to find the length/number of characters of a string).
9. File Handling Module: This section will contain functions for reading data from a file (named readfile( )) and write/append data to file (named writefile( )).
10. Main Driver Module ( main ( ) function ): It is the starting point of the execution of the program, ensuring the smooth flow of control from one function to other, and offer choice to the user to attempt or conduct quiz or exit, along with inputs if necessary.

* **Required topics from the subject:**

1. 1-D Arrays: Arrays are blocks of continuous memory allocation which hold values of same datatype. 1-D array is essentially a linear list of elements, like a single row of values. It is used in storing linear data such as lists, recording daily temperatures and in string manipulation where a string is stored and treated as a character array terminated with null value.
2. Strings: String is a one-dimensional array of characters terminated with a null value. The only difference between a string and a 1-D character array is a character array can be or be not null terminated whereas for a character array to be treated as a string, it always has to be null terminated.
3. Pointers: Pointer is a kind of variable which stores the address of the other variable. It has to be initialized by the address of a variable by user. It can of a specific primitive data type, like integer pointer which holds address of an integer variable, or a void pointer which can hold address of a variable of any data type.
4. File Handling: File Handling refers to interacting with files on your system such as creating a file, writing data to a file, reading(retrieving) data from a file, appending data to a file, etc. File Handling allows us to store the data from the program permanently on our system after the end of the whole program which would otherwise be deleted as and when the program ends.

1. Structures: A structure is a single variable which holds values of variables of multiple datatypes. You can think of it like a box (or a user defined type) which holds for example three things, your name (string), your section (character datatype) and your class roll number (integer datatype). It is like a record where the record name is the name of structure variable and the fields in it are the variables which the structure holds.

* **Platform Required:**

Code Blocks

* **Books and Link Sources:**