***Logical Design***

**Outline:**

**Ro:**

**1. Organizarea Intrebarilor:**

• Intrebarile sunt stocate intr-un array de obiecte.

• Fiecare obiect intrebare contine proprietati pentru ID-ul intrebarii, textul intrebarii, optiuni si raspunsul corect.

**2. Progresul Utilizatorului:**

• Progresul utilizatorului este urmarit folosind un index care indica intrebarea curenta din array.

• Operatiune de amestecare asigura ca intrebarile sunt prezentate intr-o ordine aleatoare de fiecare data cand se ia quiz-ul.

• Odata ce o intrebare a fost raspunsa, sistemul incrementeaza indexul pentru a trece la urmatoarea intrebare.

• Sistemul verifica daca toate intrebarile au fost raspunse pentru a incheia quiz-ul.

**3. Mecanismul de Scor:**

• Fiecare raspuns corect creste scorul utilizatorului.

• Scorul este afisat dupa fiecare raspuns trimis.

• Scorul final este afisat la sfarsitul quiz-ului.

**4. Asigurarea Fara Repetitie:**

• Algoritmul de amestecare garanteaza ca intrebarile nu sunt repetate in cadrul unei singure sesiuni.

• Indexul asigura ca utilizatorul se deplaseaza secvential prin array-ul amestecat, ceea ce previne repetitia.

• Se utilizeaza algoritmul Fisher-Yates

**Eng:**

**1.Question Organization:**

* Questions are stored in an array of objects.
* Each question object contains properties for the question ID, question text, options, and the correct answer.

**2.User Progress:**

* The user's progress is tracked using an index that points to the current question in the array.
* A shuffle operation ensures the questions are presented in a random order each time the quiz is taken.
* Once a question is answered, the system increments the index to move to the next question.
* The system checks if all questions have been answered to end the quiz.

**3.Scoring Mechanism:**

* Each correct answer increments the user's score.
* The score is displayed after each answer is submitted.
* The final score is displayed at the end of the quiz.

**4.Ensuring No Repetition:**

* The shuffle algorithm guarantees that questions are not repeated within a single session.
* The index ensures that the user moves sequentially through the shuffled array, which prevents repetition.
* Fisher-Yates shuffling Algorithm is used

**Pseudocode:**

**Question Shuffling Algorithm:**

Used in both question order and question options order

Start at the end of the array and move backwards

Generate a random index j such that 0 ≤ j ≤ i

Swap the elements at indices i and j

Function shuffleArray(array):

For i from array.length - 1 down to 1:

Let j be a random integer between 0 and i

Swap elements at array[i] and array[j]

End For

End Function

**Initialize quiz:**

Create questions array with question objects {id, question, options, answer}

Shuffle questions array using Fisher-Yates algorithm

Set currentQuestionIndex to 0

Set score to 0

Hide quiz container

Show start container

**Start quiz:**

Get username from input

If username is not empty:

Store username

Hide start container

Show quiz container

Display question at currentQuestionIndex

**Display question:**

If currentQuestionIndex is less than the length of questions array:

Display current question and options shuffled both trough the algorithm

Enable submit button

Disable next question button

Else:

Show results

**Select option:**

If answer has not been submitted:

Highlight selected option

Enable submit button

**Submit answer:**

If selected option matches answer:

Increment score

Highlight selected option as correct

Else:

Highlight selected option as incorrect

Highlight correct option

Disable submit button

Enable next question button

Set answer submitted flag to true

**Next question:**

Unhighlight all options

Increment currentQuestionIndex

Display question at currentQuestionIndex

Set answer submitted flag to false

**Show results:**

Display user's score