

1. HTML – Structure & Elements

- Structure:

```
<div class="container">  
  <div class="screen" id="start-screen">...</div>  
  <div class="screen" id="quiz-screen">...</div>  
  <div class="screen" id="result-screen">...</div>  
</div>
```
- <div>: Generic container for grouping elements.
- class & id:
 - class for styling multiple elements.
 - id for uniquely identifying elements to manipulate in JS.
- Screens: 3 screens (Start, Quiz, Result). Only one visible at a time using .active class.
- Dynamic Content Placeholders:

```
<span id="current-question">1</span>  
<span id="score">0</span>  
<span id="time-left">15</span>
```
- Updated dynamically by JavaScript.
- Buttons:

```
<button>Start Quiz</button>  
<button>Select Answer</button>  
<button>Restart Quiz</button>
```
- Progress Bar:

```
<div class="progress-bar">  
  <div id="progress" class="progress"></div>  
</div>
```
- Inner <div> width updated dynamically in JS.

2. CSS – Styling & Layout

- Reset & Layout:

```
* { margin:0; padding:0; box-sizing:border-box; }  
body { display:flex; justify-content:center; align-items:center; min-height:100vh; }
```
- Reset: Removes default browser margins/paddings.
- Flexbox: Centers .container vertically and horizontally.
- box-sizing: border-box: Simplifies width/padding calculations.
- Container:

```
.container {  
  background-color: rgba(237,230,232,0.6);  
  border-radius: 1rem;  
  box-shadow: 0 10px 30px rgba(0,0,0,0.28);  
  width: 100%;  
  min-width: 600px;  
}
```
- border-radius: Rounded corners.
- box-shadow: Floating card effect.
- min-width: Prevents shrinking on small screens.
- Screens:

```
.screen { display:none; text-align:center; padding:2rem; }  
.screen.active { display:block; }
```
- .active toggles visibility between screens.
- Buttons:

```
.answer-btn {  
  background-color: rgb(236,234,217);  
  color: rgb(15,31,31);  
  border: 2px solid dimgray;  
  border-radius:10px;  
  padding:1rem;  
  cursor:pointer;  
  transition: all 0.3s ease;  
}
```

```
.answer-btn.correct { background-color: lightgreen; color: darkgreen; }
.answer-btn.incorrect { background-color: lightcoral; color: darkred; }
- .correct / .incorrect dynamically added for answer feedback.
- Responsive Design:
  @media (max-width:500px){
    #start-screen h1 { font-size: 2rem; }
    .answer-btn { padding: 12px; }
  }
- Adjusts layout/font sizes for small screens.
```

3. JavaScript – Functionality & Logic

```
- Selecting Elements:
  const startButton = document.getElementById('start-btn');
  const questionText = document.getElementById('question-text');
- Use getElementById for unique elements.
- Use querySelector or getElementsByClassName for multiple elements.
- Event Listeners:
  startButton.addEventListener("click", startQuiz);
- addEventListener listens for events like click, input, mouseover.
- Quiz Data Structure:
  const quizQuestions = [
    { question:"Capital of France?", answers:[{text:"Paris", correct:true}, ...] }
  ];
- Array of objects: Stores questions and answers.
- Each answer has text and correct properties.
- Shuffling Questions & Answers:
  function shuffleArray(array){
    for(let i=array.length-1;i>0;i--){
      const j=Math.floor(Math.random()*(i+1));
      [array[i], array[j]] = [array[j], array[i]];
    }
  }
- Fisher-Yates shuffle algorithm.
- Makes quiz unpredictable.
- Showing Questions:
  function showQuestion(){
    const currentQuestion = quizQuestions[currentQuestionIndex];
    questionText.textContent = currentQuestion.question;
    answersContainer.innerHTML = "";
    shuffleArray(currentQuestion.answers);

    currentQuestion.answers.forEach(answer=>{
      const button = document.createElement("button");
      button.textContent = answer.text;
      button.dataset.correct = answer.correct;
      button.addEventListener("click", selectAnswer);
      answersContainer.appendChild(button);
    });
  }
- Dynamically creates answer buttons.
- Uses dataset to store if the answer is correct.
- Answer Selection:
  function selectAnswer(event){
    const isCorrect = event.target.dataset.correct === "true";
    if(isCorrect) score++;
    // Highlight buttons
    Array.from(answersContainer.children).forEach(btn=>{
      if(btn.dataset.correct=== "true") btn.classList.add("correct");
      else if(btn===event.target) btn.classList.add("incorrect");
    });
  }
```

```

    });
  }
- Updates score.
- Applies CSS classes .correct / .incorrect.
- Timer:
  let timer;
  const timePerQuestion = 15; // seconds

  function startTimer(){
    let time = timePerQuestion;
    timeLeftSpan.textContent = time;
    timer = setInterval(()=>{
      time--;
      timeLeftSpan.textContent = time;
      if(time<=0){
        clearInterval(timer);
        selectAnswer(null); // move to next question automatically
      }
    }, 1000);
  }
- setInterval: Executes a function repeatedly every second.
- clearInterval: Stops timer.

```