

# Project Documentation

## Quiz Application

### Problem Statement

The objective is to develop an interactive quiz application that:

- Presents users with multiple-choice questions.
- Tracks the user's score and displays it at the end of the quiz.
- Includes a timer for each question.
- Provides immediate feedback on the correctness of the user's answer.

### Solution Overview

The solution is a web-based quiz application built using HTML, CSS, and JavaScript. The key functionalities include:

- A start button to initiate the quiz.
- A container to display questions and multiple-choice answers.
- A timer for each question.
- Score tracking to display the user's results at the end of the quiz.
- Immediate feedback for the user's selected answer.

### Features

- **Interactive User Interface:** The quiz interface is designed for ease of use with clear instructions and responsive design.
- **Question Timer:** Each question has a time limit, adding a level of challenge.
- **Immediate Feedback:** Users receive immediate feedback on their answers.
- **Score Tracking:** The application tracks and displays the user's score at the end of the quiz.
- **Responsive Design:** The layout adjusts for various screen sizes, ensuring a consistent user experience on different devices.

### Challenges and Resolutions

1. **Timer Synchronization**
  - **Challenge:** Ensuring the timer accurately resets and counts down for each question.
  - **Resolution:** Implemented `setInterval` and `clearInterval` functions to manage the timer effectively, resetting it with each new question.
2. **Handling User Interactions**
  - **Challenge:** Managing user inputs for selecting and submitting answers.
  - **Resolution:** Added event listeners for answer selection and the next button to handle user inputs and provide immediate feedback.
3. **Dynamic Question and Answer Display**
  - **Challenge:** Dynamically updating the displayed question and choices as the user progresses through the quiz.

- **Resolution:** Utilized JavaScript to dynamically update the DOM elements with the current question and choices, and manage the progression of the quiz.

## Instructions

### Setup and Usage

#### 1. File Structure

- `ind.html`: The main HTML file.
- `sty.css`: The CSS file for styling.
- `app.js`: The JavaScript file containing the quiz logic.

#### 2. Setup

- Ensure all files are in the same directory.
- Open `ind.html` in a web browser to run the application.

### Detailed Instructions

#### 1. Starting the Quiz

- Open `ind.html` in a web browser.
- Click the "Start" button to begin the quiz.

#### 2. Answering Questions

- A question with multiple-choice answers will be displayed.
- Click on the choice you believe is correct. Your selected choice will be highlighted.

#### 3. Submitting Answers

- After selecting an answer, click the "Next" button.
- Immediate feedback will be displayed indicating whether your selected answer is correct or incorrect.
- The next question will be displayed automatically.

#### 4. Timer Functionality

- A timer is displayed for each question.
- The timer counts down from 15 seconds.
- If the timer reaches zero, you will be prompted to restart the quiz.

#### 5. Viewing Results

- After answering all questions, your score will be displayed.
- The score shows the number of correct answers out of the total number of questions.
- You can choose to restart the quiz by clicking the "Play Again" button.

### Example of Use

#### 1. Start the Quiz

- Click the "Start" button.
- The first question appears with multiple-choice answers and a countdown timer.

2. **Answer a Question**

- Select an answer by clicking on one of the choices.
- The selected choice is highlighted.
- Click the "Next" button to submit the answer.

3. **Receive Feedback**

- An alert appears, indicating whether the answer was correct or incorrect.
- The next question is displayed automatically.

4. **Complete the Quiz**

- After answering all questions, the user's score is displayed.
- Click "Play Again" to restart the quiz.