SF Dev Workshop - 2024

Assignment – 2 | Javascript Task

Dynamic Quiz App

INTRODUCTION

Are you a beginner web developer looking to enhance your skills and create a challenging project that incorporates various JavaScript functionalities? This project is designed to help you build a dynamic quiz application using HTML, CSS, and JavaScript. Throughout this project, you will encounter a series of progressively challenging tasks that will test and strengthen your web development skills.

PROJECT OBJECTIVES

- Create a responsive web page structure using HTML and CSS to display the quiz.
- Use JavaScript to manage quiz questions, options, and user interactions.
- Incorporate features like timers, score tracking, and dynamic question loading.
- Ensure a visually appealing and user-friendly interface with CSS styling.

PROJECT TASKS

1. SETUP THE PROJECT

- Create an HTML file for the quiz app.
- Create a basic CSS file to style the app.
- Link the HTML and CSS files together.
- Include a JavaScript file and link it to the HTML.

2. DESIGN THE QUIZ INTERFACE

- Create a header for the quiz title.
- Design a section to display the quiz questions.
- Add an area for multiple-choice options.
- Include a timer element.

3. DEFINE QUIZ QUESTIONS

- Create an array or object to store quiz questions, options, and correct answers.
- Populate the quiz with at least 20-30 questions on various topics.
- Alternatively, you can use a public Quiz API and fetch questions from it.
- You can try to find the public API online, some fro example would be - open trivia database.

4. DISPLAY QUESTION AND OPTIONS

- Use JavaScript to display the first question and its options when the quiz starts.
- Implement a timer that counts down while the user answers each question.

5. HANDLE USER INPUT

- Capture and validate user input when an option is selected.
- Keep track of the user's score based on correct and incorrect answers.
- Display feedback to the user after each question.

6. DYNAMICALLY LOAD QUESTIONS

- Create a function to load the next question when the user answers the current one.
- Ensure that the quiz ends after all questions are answered.

7. ADD A TIMER

- Implement a countdown timer for each question.
- Automatically move to the next question when the timer expires.

8. DISPLAY THE FINAL SCORE

- Show the user's final score and a congratulatory message at the end of the quiz.
- Allow the user to see the correct answer to the questions after the quiz is completed. The app should show both the selected input and the correct answer
- Provide an option to restart the quiz.

9. STYLE AND POLISH

- Apply CSS to make your quiz visually appealing.
- Ensure responsive design for various screen sizes.
- Add animations or transitions for a polished look.

10. BONUS CHALLENGE (OPTIONAL)

- Implement a high-score tracking system using localStorage.
- Add a lifeline feature, such as a "Skip Question" option.
- Include a question bank with a wider variety of questions, by integrating with an external extensive public API

This project will challenge you to apply your HTML, CSS, and JavaScript knowledge creatively. Take your time, break down the tasks, and don't hesitate to seek help from online resources or communities when needed. Good luck, and have fun building your dynamic quiz app!

RESPONSIVENESS

As a number of different devices keep coming up, the variety of screen types and screen sizes keep increasing. As a developer, it is important to make sure that the website UI is just as properly formatted, well proportional and usable, on each screen ranging from mobile phones to tablets to laptops to monitor screens and all the different available screen sizes within these as well.

Now, if we start testing and modifying our website for each of the commercially available screen sizes, then that won't be feasible at all. This is where responsiveness comes into play. It helps in making sure that the website does not break (overflowing texts, overlapping components, too tightly packed content or too much blank space, unnecessary scrollbars and so on) for a large range of screen sizes.

After developing the website for your laptop screen, try to make it responsive. Learn the different ways to make each of the different types of components in a website responsive. Understand and choose the ways which fit your application and implement them after completing your webpage. Learning how to use media queries is very critical to make a website responsive.

You may want to create a number of breakpoints in your website, but it is recommended to use breakpoints only when you want to change the structuring of the content. For other cases, try using responsive units: rem, %, vh, vw, etc. for dimensions and position, and rem, em for font sizes. Using grid and flexbox wherever possible and playing around with their properties can lead to very highly responsive components.

At this point, you aren't expected to come up with a perfectly responsive webpage in your first try.

Just try to add as much responsiveness as you can.

Important Instructions:

- Most importantly, DO NOT COPY THE CODE from anywhere, you can take reference, but try to write every line on your own – to learn the most. We won't be releasing any grade cards at the end of the assignment or the workshop. Whatever you do in these assignments, stays between you and us. So, just use this opportunity to learn and improve. As for the SF Tech Team selections, copying isn't going to get you anywhere…because we'll know.

- While writing CSS, don't limit yourself to just using pixels (px) as a measurement unit. Read about other units like rem, em, %, vh, vw and understand how using particular units in particular places can help you design better and make your website highly responsive.

- Try to add interesting animations, creative styling and patterns wherever you can. Also, try to choose your color combinations wisely and make sure that your website looks neat with proper spacing, alignment and sizing of all components (we're talking margins and padding). Show off those creative and artistic skills!

- We understand a lot of you would be starting from scratch with no or minimal prior knowledge. So, the assignment might look a bit overwhelming at first glance and that is fine. Don't think about the website as a whole, rather focus on one section at a time. Start building and working upon the components in each section one-by-one. Slowly, seeing the sections come to life visually should motivate you to go further. Also, it's okay if you're not able to fully complete the assignment. Try to do as much as you can in the given time.

- Lastly, web development is a vastly interesting and highly sought-after field that can open up a plethora of opportunities for you, in web dev as well as other software domains. With everything shifting online and digital, the demand for skilled and proficient developers is enormous. And this is your chance to get a head-start in this widely potential field. So, give it your best shot because who knows where it might lead you to.
