



Static-Site

November the 11th, 2019

Resumen

Este proyecto trata sobre realizar un quiz, de 20 preguntas en total, sobre HTML, CSS y JavaScript. Las respuestas deben ser guardadas en el localStorage y la puntuación de los usuarios, para mostrarlas al final.

Contenidos

Requerimientos del Proyecto

Tareas

Incidencias

Lecciones aprendidas

Calendario del Proyecto

Cronograma del Proyecto

Medidas de Control de Calidad

Métricas de Calidad

Riesgos

Herramientas usadas

Requerimientos del Proyecto

Para este proyecto, los requerimientos serán los siguientes:

- El cuestionario tendrá que contener al menos 20 preguntas.
- El formato de las respuestas se puede dividir en:
 - o Seleccionar respuesta única de múltiples opciones (radio)
 - o Seleccionar múltiples respuestas (checkbox)
 - o Seleccionar desde un menú desplegable
- Habrá una barra de progreso que medirá el porcentaje de preguntas respondidas respecto al total (esta barra se deberá mostrar en todas las páginas y se deberá actualizar tras contestar cada pregunta).
- La barra de progreso estará dividida en secciones (por ej: HTML, CSS, JS...) y se deberá actualizar de forma dinámica cada vez que se responda a una pregunta mediante una animación de CSS3.
- Tras responder cada pregunta el resultado se almacenará en **localStorage**.
- Debe de existir un call to action para iniciar el cuestionario. Este inicio crea un nuevo registro en **localStorage** para poder almacenar múltiples cuestionarios que se podrán ordenar posteriormente por el scoring final. Es importante que asocies un scoring a cada pregunta para poder obtener el resultado final.
- La primera pantalla que se mostrará como puedes ver en el wireframe pedirá al usuario un nickname el cual se guardará junto al resultado del test en el **localStorage**.
- Las preguntas se repartirán en al menos 3 páginas separadas. Cada vez que se pase a la siguiente página se deberá guardar toda la información de la anterior en **localStorage** (las respuestas y el estado actual del progreso en el cuestionario, que sería el porcentaje de respuestas realizadas respecto al total de respuestas).
- Todo el evolutivo se realizará en local y sin una base de datos, usando HTML, CSS y Javascript (**NO** se pueden usar librerías de terceros)
- Debes hacer uso de GIT
- Escribe commits claros y concisos, no debería extenderse más de 50 caracteres.
- Tiene que ser **responsive** y debes de conseguirlo sin hacer uso de librerías de terceros.
- Al pasar de una pregunta a otra se mostrará una animación hecha con CSS3.
- Al terminar el cuestionario, deberás mostrar una pantalla final con tu resultado y el ranking final indicando tu puntuación obtenida y la de los otros registros que se hayan realizado anteriormente. La tabla del ranking se rellenará con los registros almacenados en el localStorage. Dicha tabla se podrá ordenar de mayor a menor puntuación mediante un botón.

Tareas a realizar

1. Leer el documento descriptivo del proyecto:
 - Prioridad: Alta.
 - Descripción: Leer la descripción del proyecto para entender todos los requisitos.
 - Nivel de dificultad: 1/10
 - Estimación de tiempo: 15 min.
2. Crear repositorio en GitHub
 - Prioridad: Alta.

- Descripción: Crear el repositorio en Github para almacenar todos los contenidos los archivos del proyecto.
 - Nivel de dificultad: 4/10
 - Estimación de tiempo: 10 min.
3. Prototipar:
- Prioridad: Alta.
 - Descripción: Conceptualizar cómo será el quiz, hacer bosquejos de cómo será la página.
 - Nivel de dificultad: 5/10
 - Estimación de tiempo: 30 min
4. Creación index.html
- Prioridad: Alta
 - Descripción: crear las plantillas html a nivel general para cada página del quiz
 - Nivel de dificultad: 1/10
 - Estimación de tiempo: 5 min.
5. Creación de otros html
- Prioridad: Alta.
 - Descripción: Utilizar la plantilla Home como base para el resto de los html's.
 - Nivel de dificultad: 2/10.
 - Estimación de tiempo: 12 min.
6. Detalles a Html's
- Prioridad: Alta
 - Descripción: Darle más detalles (como headings y divs) a todos los htmls.
 - Nivel de dificultad: 5/10.
 - Estimación de tiempo: 10 minutos.
7. Crear Footer
- Prioridad: Baja
 - Descripción: Agregarle un footer a todas las páginas para mejorar la estética del portal.
 - Nivel de dificultad: 2/10.
 - Estimación de tiempo: 5 min.
8. Preguntas de HTML
- Prioridad: Alta.
 - Descripción: Crear 6 preguntas para la sección HTML.
 - Nivel de dificultad: 6/10.
 - Estimación de tiempo: 12 min.
9. Botones Radio

- Prioridad: Alta.
 - Descripción: Encontrar la manera de usar botones “radio” para las preguntas de elegir la mejor respuesta
 - Nivel de dificultad: 6/10.
 - Estimación de tiempo: 30 min.
10. Contenido preguntas HTML
- Prioridad: Alta.
 - Descripción: Creación del contenido para las preguntas Html’s
 - Nivel de dificultad: 6/10.
 - Estimación de tiempo: 30 min
11. Tutoriales LocalStorage
- Prioridad: Alta.
 - Descripción: Ver tutoriales sobre localStorage y entenderlo bien, ya que es crucial para el desarrollo del proyecto
 - Nivel de dificultad: 9/10.
 - Estimación de tiempo: 120 min
12. Cambio de Pregunta
- Prioridad: Alta.
 - Descripción: lograr que la ventana modal cambie de pregunta, al hacerle click.
 - Nivel de dificultad: 2/10.
 - Estimación de tiempo: 120 min.
13. Grabar datos en localStorage
- Prioridad: Alta.
 - Descripción: lograr que el localStorage funcione para guardar los datos.
 - Nivel de dificultad: 8/10.
 - Estimación de tiempo: 60 min..
14. questions.js
- Prioridad: Alta.
 - Descripción: Intentar crear un archivo Questions.js siguiendo un tutorial para que las preguntas se pasen al JavaScript
 - Nivel de dificultad: 8/10.
 - Estimación de tiempo: 60 min.
15. Crear funciones
- Prioridad: Alta.
 - Descripción: Probar diferentes funciones

- Nivel de dificultad: 8/10.
- Estimación de tiempo: 12 min.

Registro de Incidencias

- La primera incidencia que tuve fue al crear el repo de Git y al intentar acceder el folder del Proyecto, debí haber escrito Desktop/final-quiz/final-quiz, pero en cambio traté de acceder solamente a "Desktop/final-quiz", lo cual causó una confusión al Git, así que tuve que pedirle ayuda a Alaa.
- Después del almuerzo del 11/6/2019 he notado que pierdo el enfoque de dónde estaba y qué era la próximo que debía hacer.
- En el inicio tuve dificultades para crear los botones radio para las preguntas de html, ya que tenía que encontrar cuál era la mejor opción, por ejemplo si usar tag submit, tag input o tag button.
- Al colocar los progress bars, el progreso no se registra bien. Se borra después de darle al botón.
- Entender cómo funciona el localStorage me ha tomado bastante tiempo.
- Tuve mucha dificultad a la hora de crear funciones. No sabía ni por dónde empezar.

Lecciones aprendidas

- Al usar Git, debes ser muy preciso al momento de hacer referencia a un folder en particular. Si no escribes bien el “path”, el programa se puede confundir. Debo evaluar la opción de simplemente crear los repositorios desde Git Bash de ahora en adelante, en vez de usar github.com.
- A la hora de prototipar, es buena idea conversar con el equipo para recibir feedback de diferentes formas en que se puede maquetar un proyecto.
- A la hora de preparar la documentación, creo que es mejor utilizar Google Docs, ya que guarda todo automáticamente.
- Es buena idea crear un css para poder visualizar los borders de todas las secciones que se van a utilizar en la página.
- En visual studio code, se puede utilizar por ejemplo, “link.css” y se crea automáticamente un link especial para css.
- Debes utilizar < para hacer “<” y > para hacer “>”.
- Debes revisar los requisitos del informe del proyecto constantemente para asegurar que estás haciendo lo que debes hacer.
- El tag <meter> es muy útil para crear barras de progreso, pero parece que no funciona bien con Internet Explorer 11.
- La propiedad Element.innerHTML devuelve o establece la sintaxis HTML describiendo los descendientes del elemento.
- Evitar usar variables Globales porque retrasan la descarga de la página, y no es un buen hábito a largo plazo.
- Utilizar Let es más recomendable que Var a la hora de declarar variables porque te permite trabajar con variables de forma más segura.
- Evitar usar variables con “_”, mejor usar camelCase.
- La organización de todo es super importante. Tanto del tiempo como los archivos, tu propio ordenador, etc.
- Es mejor que las funciones cumplan una sola función, en vez de intentar meter todas en una sola función.
- Evitar comentarios sobre las variables o funciones. Si tienes que usarlos, es porque el nombre que has utilizado para ellas no es tan bueno.

- Me falta mucho estudio, o sea la parte teórica para poder trabajar mejor con javascript y entender mejor los conceptos para luego poder aplicar la parte práctica.
- Alt+click: te permite elegir varias cosas a la vez dentro de VSC.
- Se puede partir el código en varios archivos, para que sea más ligera la página.

Calendario

- Miércoles
 - ❖ Preparar a nivel básico la maquetación de las páginas principales.
 - ❖ Agregarle diseño a las páginas principales, con CSS, por ejemplo márgenes.
- Jueves:
 - ❖ Trabajar en funcionalidades de la primera página de preguntas.
 - ❖ Estudiar y entender sobre Modales, para poder cambiar las preguntas.
 - ❖ Ver tutoriales de YouTube para entender sobre el localStorage.
 - ❖ Reunirme con Rafael para ver el tema de cómo crear las barras de progreso con html y css
- Viernes:
 - ❖ Trabajar con Javascript
 - ❖ Entender más cómo funciona el localStorage.
 - ❖ Intentar que el contenido del array de preguntas pueda reemplazarse al html con la propiedad innerhtml
- Sábado:
 - ❖ Seguir intentando crear funciones con javascript.
 - ❖ Mejoras generales a la organización del proyecto (creación de folders para archivar).
 - ❖ Conceptualizar las tareas a seguir

Cronograma