Name: Ticio Victoriano

Report

When I started analyzing the project requirements while reading the project description, I was trying to visualize what would be the best approach to come up with a layout that would satisfy all the requirements of the project. I came up with so many designs on my head but none of them was good enough to be implemented as the design of this particular project. Then I took a step back and tried to break down what really means to have a laid out interface and an attractive design to keep students engaged. As result, I came to realize that a website with a laid out interface and attractive design would be a website that:

- It is easy to use: The user can easily browse the different parts of the website without having to think much about how the website works.
- It is visually appealing: The user can easily engage with the website because of its beauty.
- It is interactive: The user can easily engage with the website because it provides interactivity and keeps the user engaged while browsing on the website.
- It is accessible: The user can access the website from devices with different screens in terms of size.
- It is simple: The website provides simple and effective user journeys which allow the users to easily each achieve their goals on the website.

Then I tried to put myself in the users' shoes. I started asking myself: If I would be a user of this website, How the website site should look like so that it can be usable, accessible, simple and visually appealing to me? What additional things, rather than quizzes, should the website have to help me as a student to maximize my learning in javascript?

To make the website easy to use I reduced the number of steps that I user could have taken to achieve something through the website.

To make the website visually appealing, I used images as background of elements and applied a slight darkness on them using the linear-gradient and rgba(0,0,0, 0.6). In addition to that, I created white elements (navigation, quizzes, contact form, etc) to create synchrony between dark colors and white color.

To make the website accessible on devices of different screen sizes I used bootstrap to make the website responsive.

Finally, to make the website simple I created a simple layout for the index, I reused that layout on the other pages only changing small things.

Regarding the quizzes, I tried to use the same design so that the user can easily get familiar with it after a small number of attempts. This allows the user to fully focus on the quizzes rather than trying to understand, for example, how he can exit the quiz, how he can go to the next question or how he can check a question and what happens when he checks a question.

For the first and second quizzes, I have defined the questions in the javascript. I used javascript objects to store all the information related to the question such as the question statement, the different options, the feedback when the selected option is correct and the feedback when the selected option is incorrect. On the first quiz, I have implemented the javascript to block a user to move to the next question if he didn't select and check if the option he selected is correct or not. On the second quiz, I have implemented the javascript to block the user to move back to a previous question.

For the third three, I have defined the questions within the HTML code. I decided to do that, because, the way the questions are structured and displayed vary from one another. For instance, I have created drop-down, autocomplete, select and memory game question. For each one of the questions, there is a javascript associated file which adds interactivity to the question.

On this project, I am not implementing a database structure, login, registration, authentication, and dashboards, because the project's requirements do not address it. In case, I would be required to build the website to store students' scores and track their progress, it would be necessary to create a database structure that would allow us to store the students' records. This database would require as minimum tables the following:

- **students table:** To store student's information like name, student id, student age, student email address.
- **quiz table:** To store information about the different quizzes like quiz id, quiz name, etc.
- **studentScore table:** To store information about the scores of the students and track their performance.

Alongside with the database, the website site would have to require a student to register and login before attempting a quiz. On the registration process, the website would get the student information and send it to be stored in the student table in the database. Data information would be used to log in the user into the system and allow him to attempt the quizzes that are in the system. Additional things like dashboard could also be implemented. A dashboard would help a student to keep track of his performance.