**University of Essex International College**

**Module name: Web Development**

**Module code: IY4103**

**Assessment title: Written assignment (individual):**

**Design and create content for a small website**

**Vladislav Veselov**

**Student ID: P400588**

**Tutor’s Name: Mohammadamin Jarrahi**

*I confirm that this assignment is my own work.*

*Where I have referred to external sources, I have provided in-text citations and included a clearly defined source.*

Requirement Analysis

**Identification of users**

The target users of the Essex Typing website are individuals who want to improve their typing skills. These users may be students, office workers, or anyone who frequently uses a keyboard and wants to type faster and more accurately.

**Discussion of user stories or personas**

As a student who frequently writes essays and papers, I want to be able to practice my typing skills so that I can type faster and make fewer mistakes. I want to be able to choose from a variety of typing exercises, track my progress, and see my typing speed and accuracy improve over time.

**Identification and classify requirements**

Using the MoSCoW method of requirement classification, I can priorities these requirements as follows:

* MUST: A variety of typing exercises and tracking progress are essential for the website to meet the needs of its users.
* SHOULD: A user-friendly interface and easy navigation are essential for usability but not as crucial as the website’s core functionality.
* COULD: Additional features such as customisable goals and typing games could be added to enhance the user experience, but they are not necessary for the website’s basic functionality.
* WON'T: Social media integration and other unnecessary features are not relevant to the purpose of the website and can be excluded.

**Plan**

**Our overall project plan for the development of the Essex Typing website is as follows:**

* Initial planning and requirement gathering: In this phase, I will classify user requirements using the MoSCoW method. I will also create user stories and personas to help guide the development of the website.
* Website design and prototyping: In this phase, I will create wireframes and prototypes of the website to determine the layout and functionality of each page. We will also design the user interface and user experience of the website.
* Development: In this phase, I will develop the website using HTML, CSS, and JavaScript. We will also test the website to ensure that it is functional and user-friendly.
* Testing and deployment: In this phase, I will conduct thorough testing of the website to ensure that it meets all requirements and is ready for deployment. We will then deploy the website to a web server and make it available to users.

**Our timeline for the project is as follows:**

* Initial planning and requirement gathering: 1 week
* Website design and prototyping: 1.5 weeks
* Development: 2 weeks
* Testing and deployment: 3-4 days

**Some milestones for the project include:**

* Completion of initial planning and requirement gathering
* Completion of website design and prototyping
* Completion of development
* Deployment of the website

**The project plan is linked back to the requirements for the website in the following way:**

* The initial planning and requirement-gathering phase ensure that I have a clear understanding of the needs and expectations of the users, as identified in the user stories and personas.
* The website design and prototyping phase focus on creating a user-friendly interface and easy navigation, as identified in the MoSCoW requirements classification as "SHOULD" priorities.
* The development phase involves building the website using HTML, CSS, and JavaScript to meet the "MUST" requirement of having a variety of typing exercises and tracking progress.
* The testing and deployment phase ensures that the website is fully functional and meets all requirements before it is made available to users.

**Design**

Here is a wireframe of the initial design for the Essex Typing website:

Diagram

Description automatically generated

Figure 1. Initial design

The initial design (Figure 1) includes a navigation bar at the top of the page, a main content area in the centre, and a footer at the bottom of the page. The navigation bar includes links to different sections of the website, such as "About”, “Contact" and “Mission”. The main content area includes a heading and a call to action for users to "Start Typing". The footer includes social media and additional contact information.

After receiving feedback on the initial design, I made the following changes to the final design:

Text

Description automatically generated

Figure 2. Logo and dropdown menu

* I added a logo to the navigation bar (Figure 2) to give the website a stronger brand identity.
* I added a dropdown menu to the navigation bar for the mobile version of the website to allow users to access different typing exercises easily. (Figure 2)

Graphical user interface, text, application

Description automatically generated

Figure 3."back to top button"

* I added a "back to top" button to the footer to allow users to easily scroll back to the top of the page.

I made these design choices based on the following considerations:

* The logo and dropdown menu help to improve the usability and navigation of the website.
* The "back to top" button is a helpful feature for users who are scrolling through long pages.

Overall, I believe that these design choices will improve the user experience and help the website meet the needs of its users.

**Implementation**

**Implementation process**

The implementation process for this website involved first setting up the HTML structure and content, including the navigation bar and various sections of the page. Next, the CSS was used to style and lay out the page, including styling the navigation bar and making the page responsive for mobile devices. Finally, JavaScript was added to provide interactivity and functionality, such as the "back-to-top" button and the ability to toggle the mobile navigation menu.

**Challenges faced and changes made from designs**

One challenge I faced during the implementation process was ensuring that the page was fully responsive and looked good on all device sizes. To overcome this, I used the w3css library, which provided various responsive design options. We also had to make some changes from the initial designs, such as adjusting the layout of certain sections to fit the responsive design better. Overall, the implementation process was relatively smooth, and I successfully created a functional and visually appealing website for typing and coding practice.

**The implementation process of JavaScript code in index page**

The JavaScript code for this website is used to provide interactivity and functionality to the page. Some of the key features implemented using JavaScript include:

* The "back-to-top" button smoothly scrolls the user back to the top of the page when clicked. This is implemented using the scroll To function, which animates the scroll position of the document.
* The toggle function for the mobile navigation menu allows the user to open and close the menu when viewing the website on a mobile device. This is implemented using the classList property, which will enable us to add or remove classes from an element.
* The typing practice functionality, which allows the user to type out a given phrase and check their input for accuracy. This is implemented using a combination of addEventListener to listen for user input and innerHTML to update the page with the result message.

Overall, the JavaScript code plays a crucial role in enhancing the website’s user experience by providing various interactive features.

**Changes made from designs and bugs. A screenshot of a computer

Description automatically generated**

Figure 4. Incorrect accuracy and initial design of typing page

In the initial stage of development, I had an error with calculating the accuracy percentage due to the fact that each negative was shown to me and the percentage was greater than the maximum, which was -127% ( Figure 4)

**This problem was related to two errors in my code:**

* The variables used in the accuracy calculation are not initialized correctly and contain the expected values.
* My initial variable had an initial percentage of 100 which did not allow it to have any other value except -127%

**The implementation process of JavaScript typing code**

The implementation process for this typing and coding project involved creating variables to store the start time and the correct paragraph and making an HTTP GET request to the Quotable API to retrieve a random quote. This quote is then displayed in the paragraph element and stored in the correctParagraph variable.

The user's input is taken from the typing area and compared to the correct paragraph using a loop that checks each letter. The input letter is highlighted in red if the letters don't match.

The user can then check their answer by pressing the "Check Answer" button, which compares the paragraph and the typing area. If they match, a success message is displayed along with the user's typing speed and accuracy. If they don't match, an error message is shown.

The user can also restart the typing test by pressing the "Restart Typing Test" button, which resets the start time and retrieves a new quote from the Quotable API.

One challenge faced during the implementation process was ensuring that the typing speed and accuracy calculations were accurate. This required careful testing and debugging to ensure that the correct values were being displayed.

Another challenge was making the user's input match the correct paragraph as they were typing, which required using the input event listener and the checkAnswer function. Despite these challenges, the implementation was successful, and the typing and coding project is now functional.

**Testing and Integration**

**Testing procedure**

To test and integrate the JavaScript code, a series of tests were conducted to ensure that all the functions were working correctly and that the user experience was smooth. The first step was to test the fetch request to the Quotable API to retrieve a random quote. This involved verifying that a quote was being displayed in the paragraph element and that the correctParagraph variable was being correctly set to the quote.

Next, the checkTyping function was tested to ensure that the input letters were being correctly compared to the correct letters and that any mismatches were being highlighted in red. The checkAnswer function was then tested to ensure that the correct message was being displayed and that the typing speed and accuracy calculations were correct.

**Existing and repaired bugs**

During the testing process, several bugs were discovered and repaired. One issue was that the input event listener was causing the checkAnswer function to be called too frequently, which resulted in the typing speed and accuracy calculations being inaccurate. This issue was resolved by adding a debounce function to the input event listener to limit the number of times it was called. Another issue was that the typing speed and accuracy calculations were incorrect when the user started typing before the quote was displayed in the paragraph element. This was fixed by adding a delay to the startTime variable so that it was not set until after the quote was displayed.

**One of the existing bug** which I’ve found is incorrect letter is not being highlighted in red when the user is typing

* One possible cause could be an issue with the check Typing function, which is responsible for comparing the input letters to the correct letters and highlighting any mismatches in red
* Another possible cause of this issue could be an error in the HTML or CSS of the webpage, which could be preventing the red color from being applied to the incorrect letters.

But After my code review, I couldn’t fix this problem.

Overall, the testing and integration process of the JavaScript code was functioning nearly correctly and the user experience was smooth. The discovered bugs were nearly repaired, and the code is now ready.

**Maintenance**

**Maintenance procedure for future development and possible future work.**

Depending on my target audience and the site's purpose, I can include many features on a typing or coding website. Some potential features might include the following:

* One main procedure for future development is adding additional programming languages because now available only three languages and one random quote typer.
* Typing lessons: Users can take typing lessons to learn how to type more efficiently or to learn a new keyboard layout.
* Coding challenges: Users can practice coding by completing challenges and exercises on the site.
* Coding tutorials: Users can learn how to code by following step-by-step tutorials on the site.
* Personalized progress tracking: Users can track their progress and set goals for their typing or coding skills.
* Leaderboards: Users can compete against each other to see who has the highest typing speed or completed the most coding challenges.