**Web Design and Development**

User Interface design

The website design uses wireframes made by Adobe XD, with annotations made with GIMP. There is 1 page, with different buttons modifying the DOM or expanding different elements. There is a wireframe for each transition and layout change in the page. There will be media queries for the site included for styling of the website.

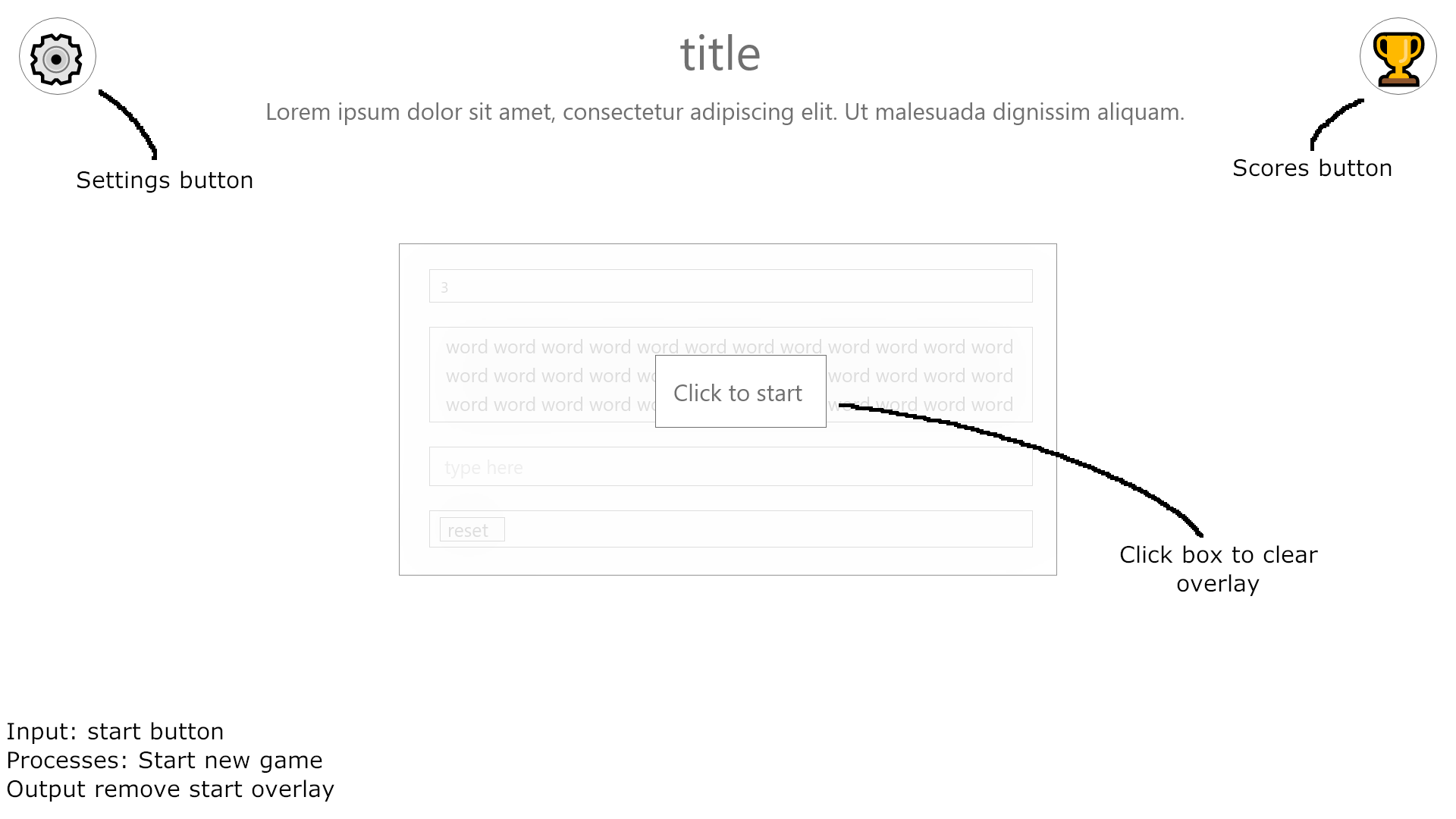
The website will be styled with CSS, and elements created using HTML. Changes in the HTML will be done with JavaScript DOM manipulation.

Figure 1: Start page

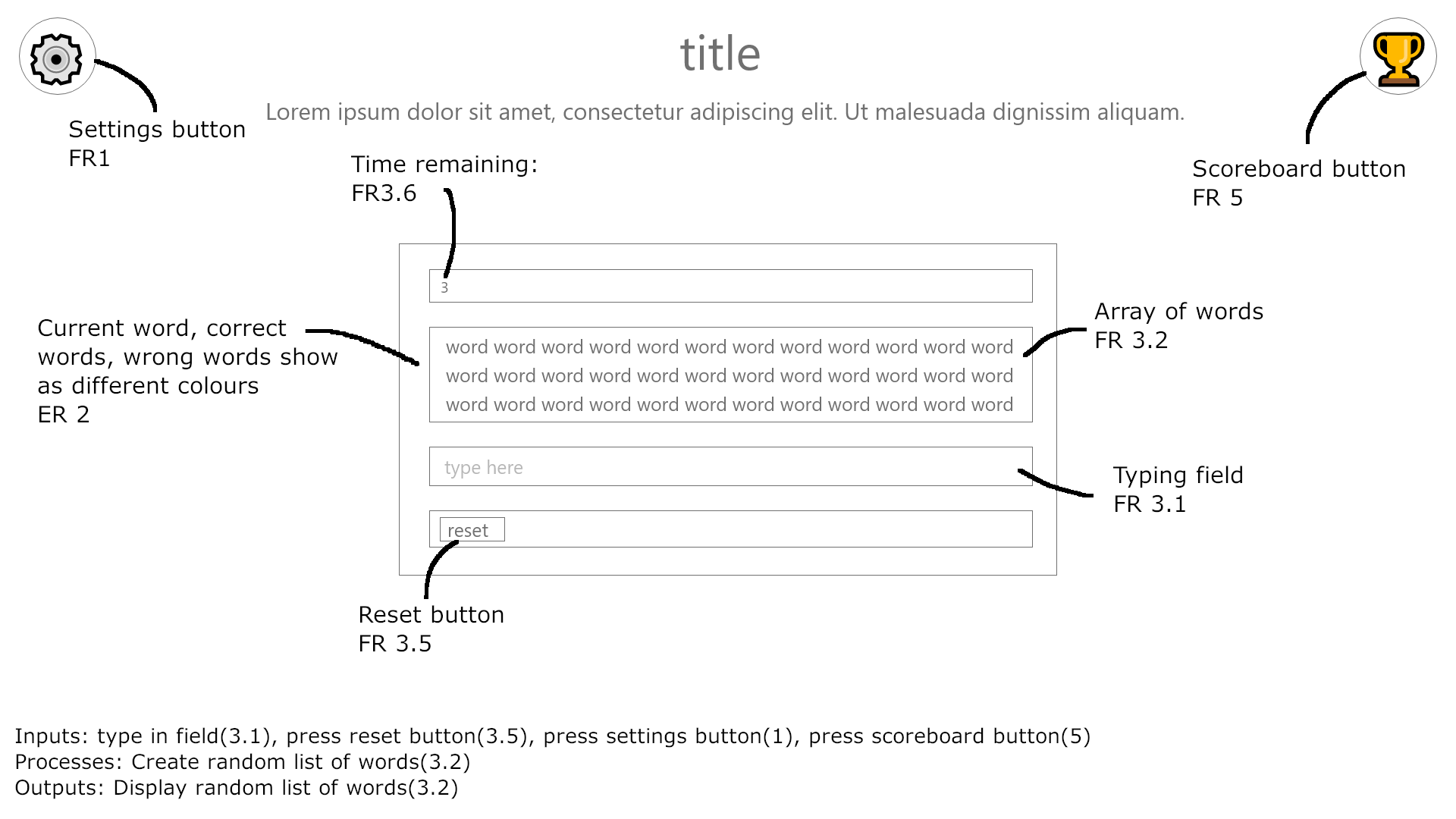


Figure 2: On start button press

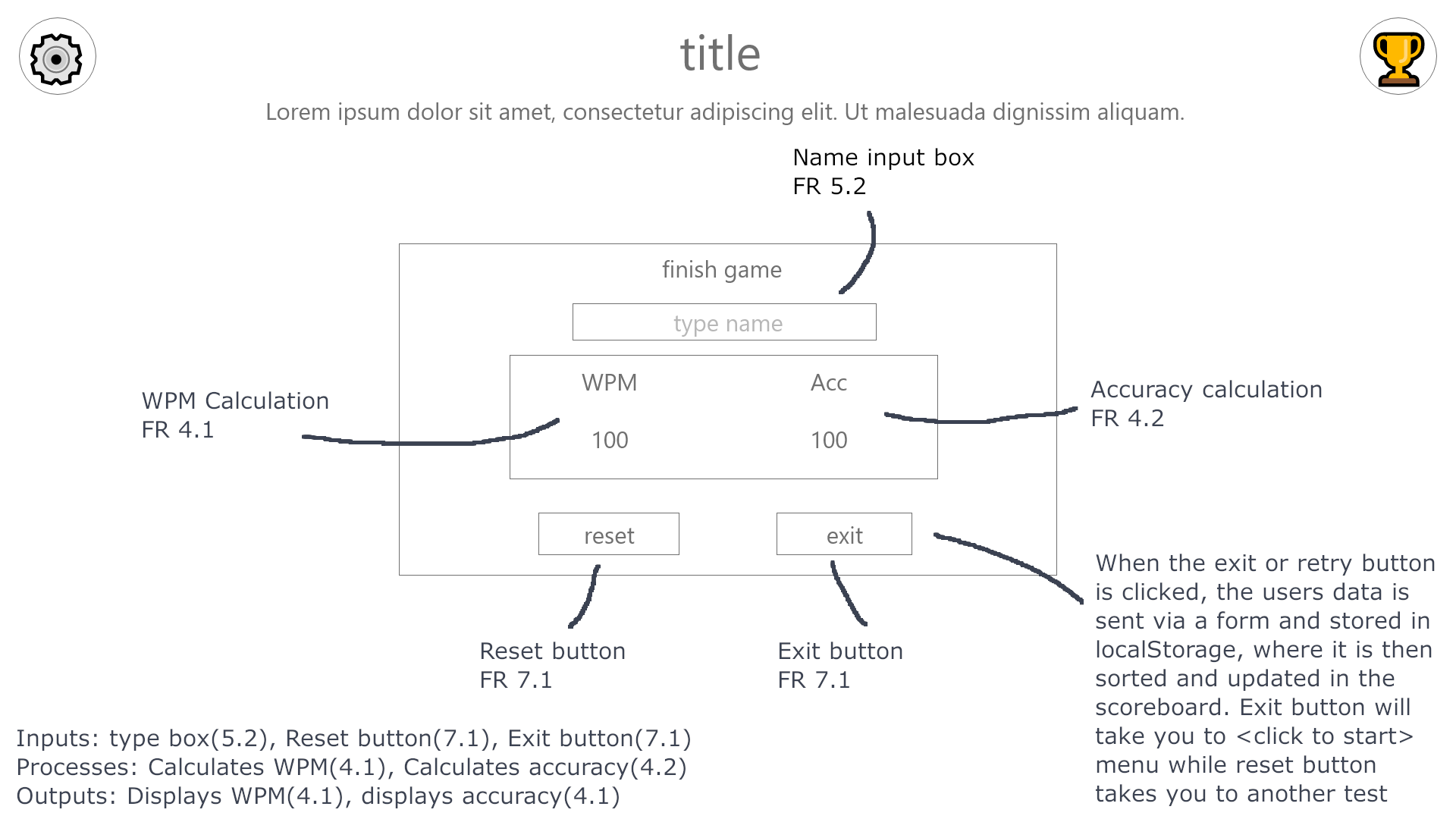


Figure 3: When test is finished

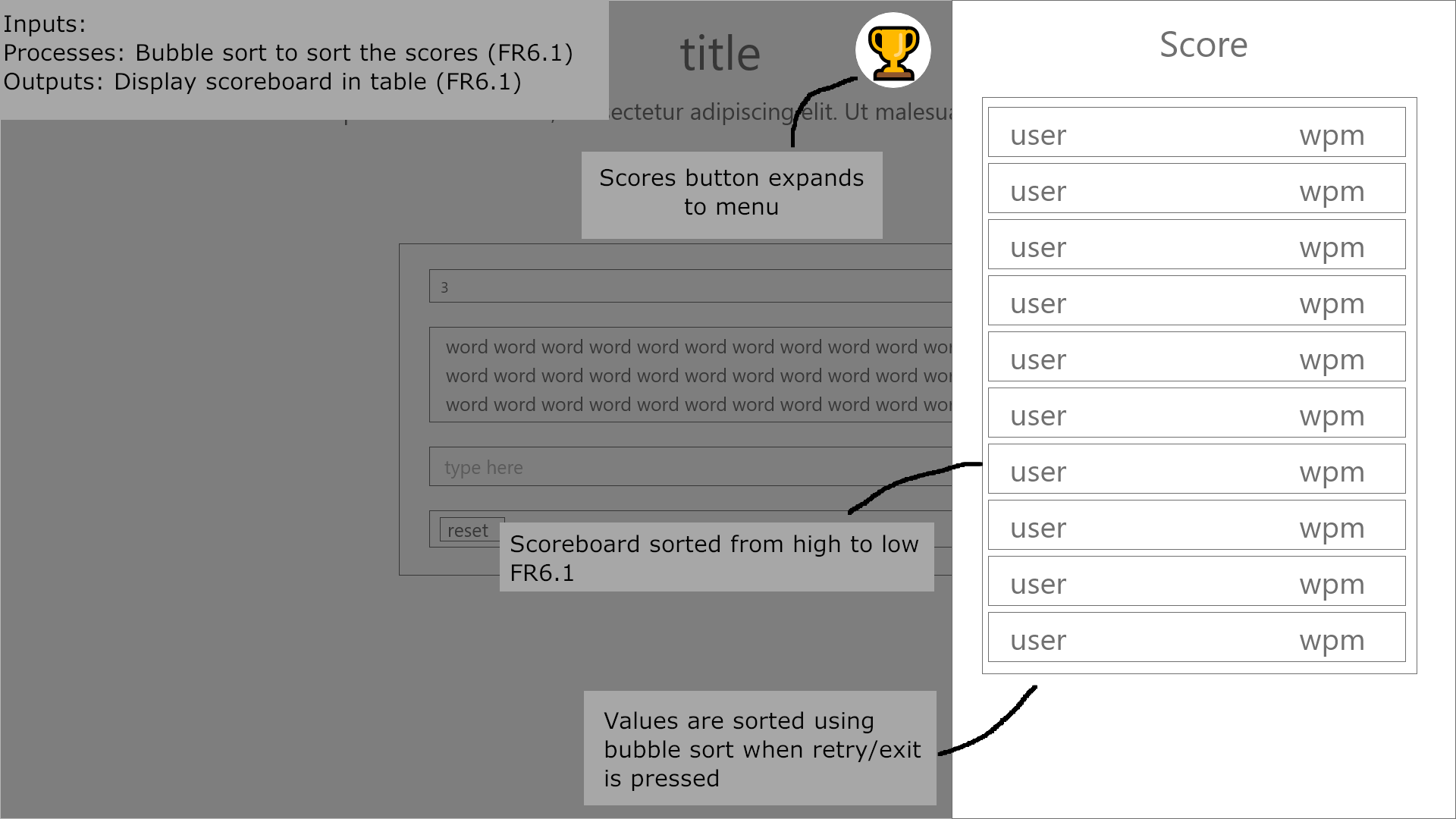


Figure 4: When scores button is pressed

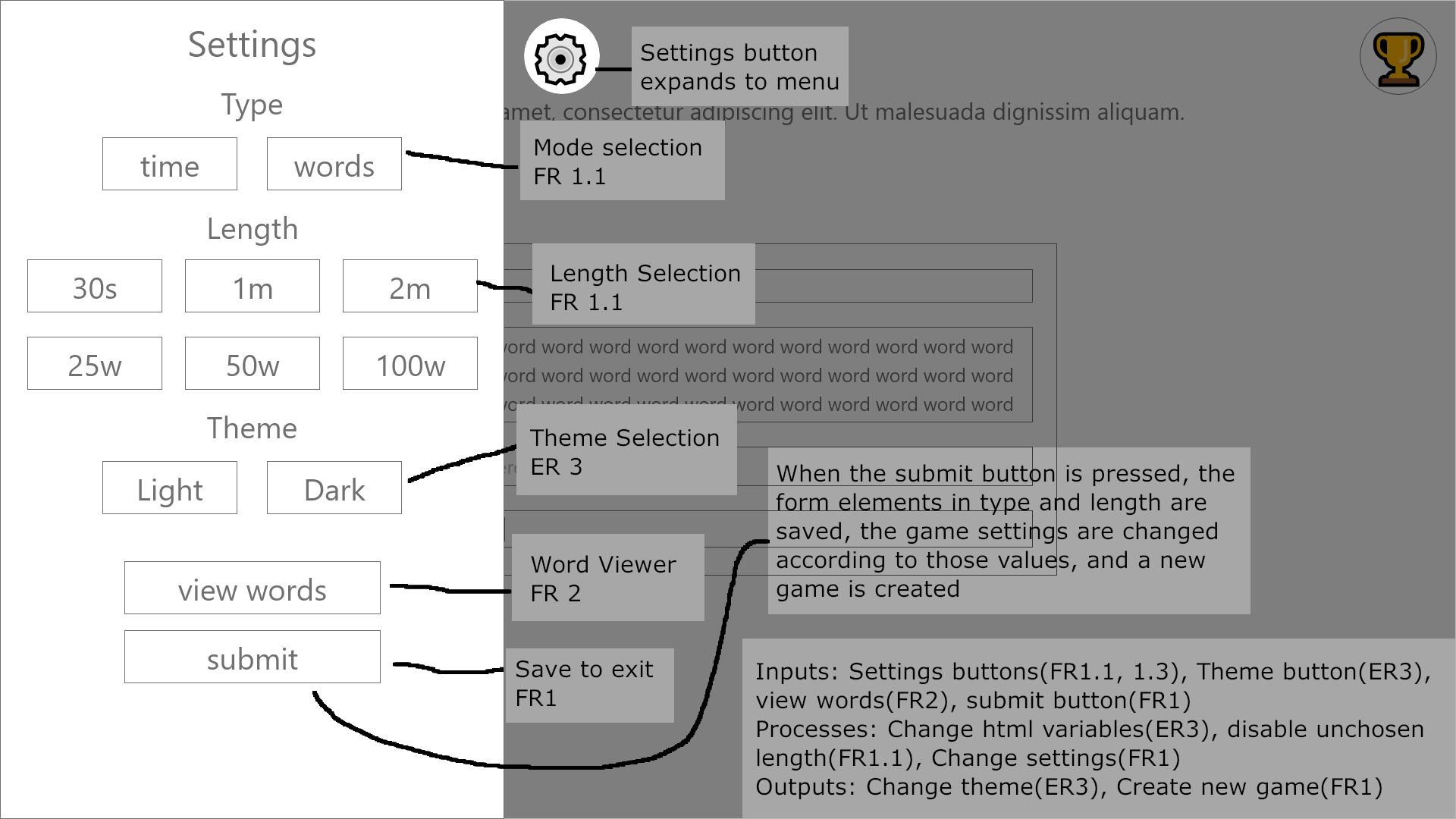


Figure 5: When settings button is pressed

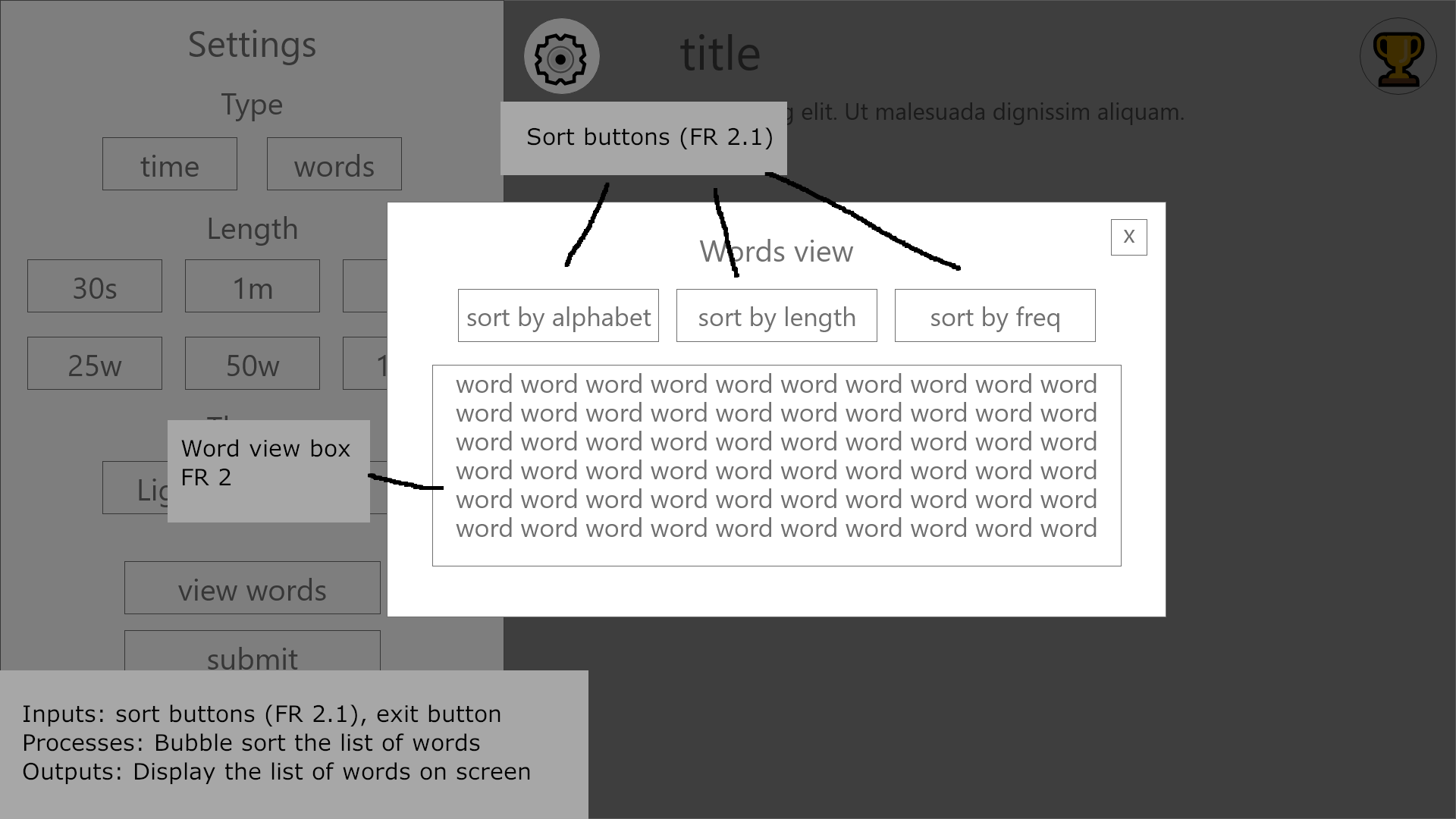
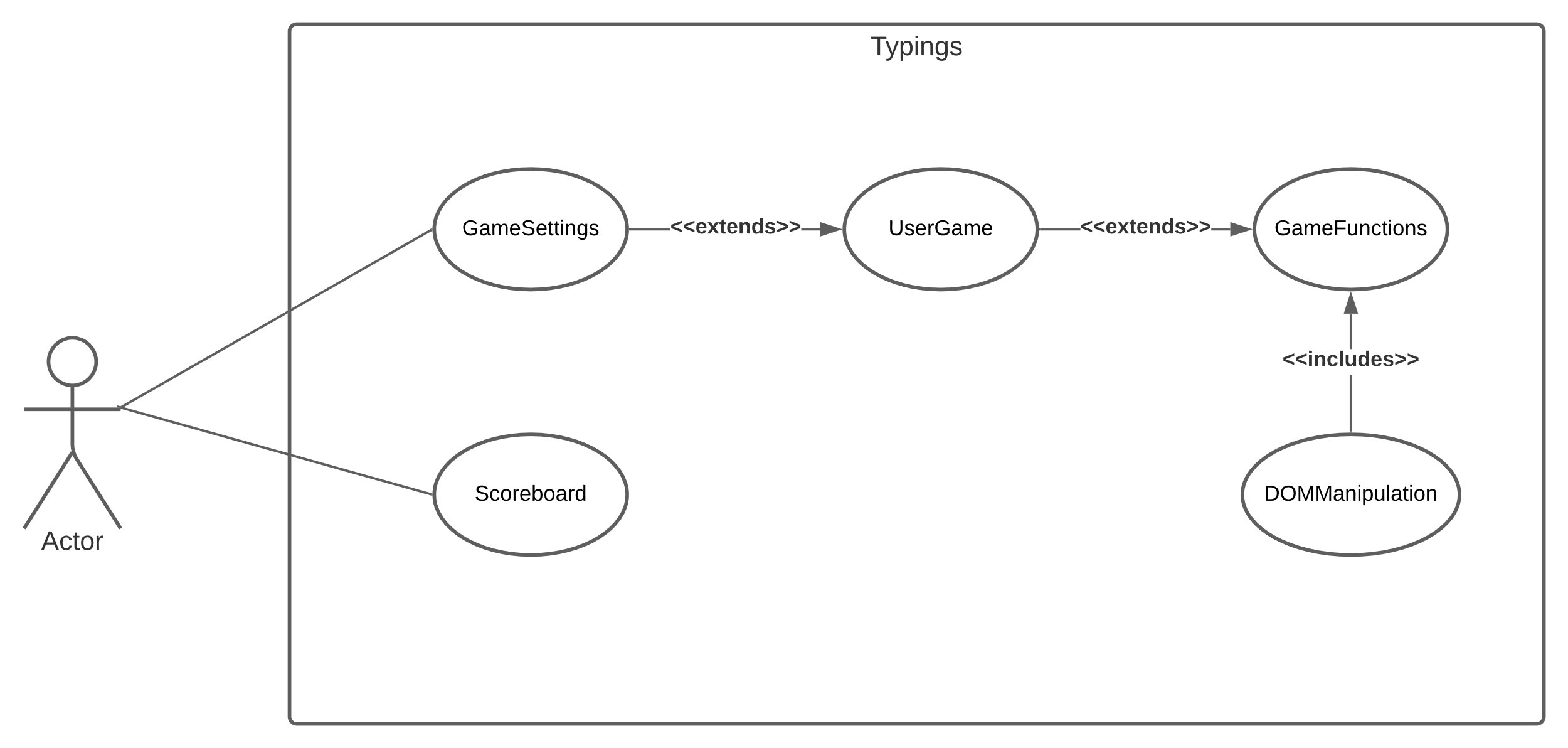
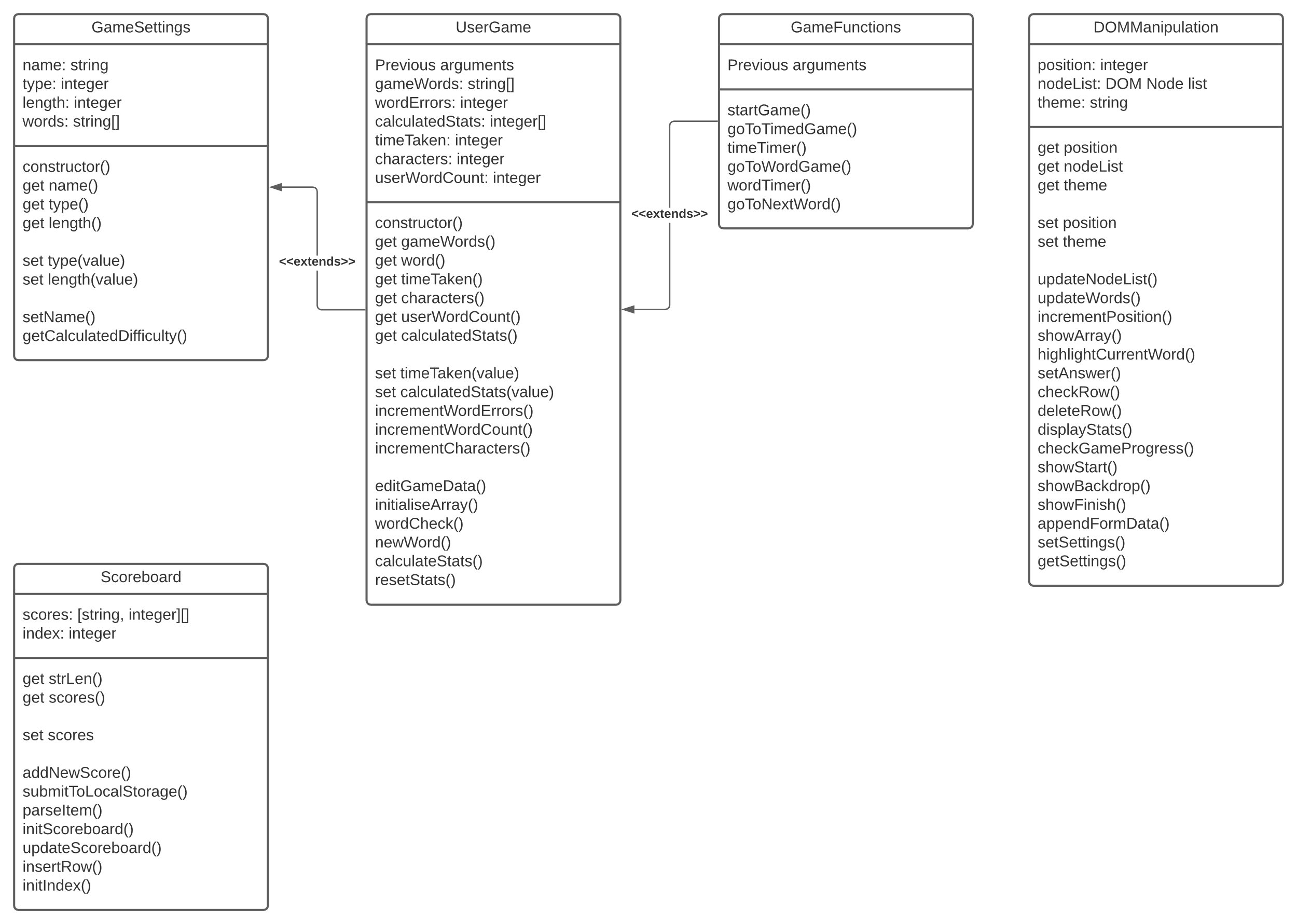


Figure 6: On view words button pressed

**Software Design and Development**

In the Software design section, there will be a UML class diagam of all the different classes included in the program. These will include any variables, functions and inheritance of each of the classes. There will be a basic layout of the localStorage structure, which will be used to store user data and user settings. There will also be pseudocode for 2 programs – the first one being purely the classes and game element. This will have clases for the program itself, and classes for how the program integates with the website. The second pseudocode is for the rest of the website, which includes global DOM variables, event listeners and other events.

UML Class Diagram  


Local Storage

User settings will be stored in the website LocalStorage.

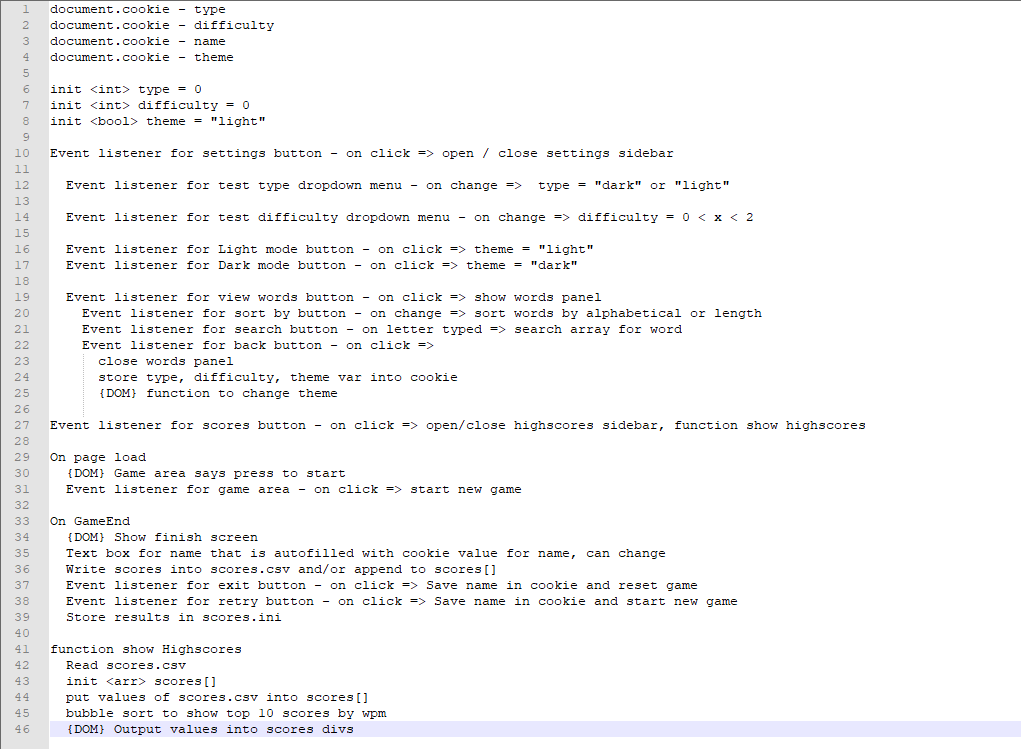
Scores will also be stored in localStorage, and will be updated each time a user submits a score

e.g [[“type”, “1”], [“length”, “1”], [“theme”, “dark”], [“Leon”, “93”]]

Pseudocode(Program)



Pseudocode – Website



|  |  |
| --- | --- |
| Requirement | Is Requirement met? |
| 1. You will be able to customise the settings of the game. |  |
| * 1. The program will let you change the length and the mode of the program. When the page loads, the default settings are timed game with a length of 1, which is a 1-minute timed test. |  |
| * 1. The number of words in the array will be the top 200 words. |  |
| * 1. The mode can be selected from specified time (2 min, 1 min, 30 secs) or specified words (25 words, 50 words, 100 words) |  |
| 1. The program will let you display the words used in a list. |  |
| * 1. User will be able to choose if the words will be displayed alphabetically, by frequency or by length using a bubble sort. |  |
| 1. The program will play a typing game that measures your WPM and accuracy. |  |
| * 1. The program will detect when you start typing in the input box and start the program. |  |
| * 1. The program will get an array of random words from a file containing the stored words. |  |
| * 1. The program will detect when a space bar is pressed, move onto the next word, and generate a new value in the array. |  |
| * 1. The program will check whether the word you typed is correct and will change word’s text colour to red if it is wrong. |  |
| * 1. There will be a reset button where you can restart the test with a new set of words |  |
| * 1. There is a counter that will count down time remaining or words remaining, depending on the selected type |  |
| 1. When the program finishes, the user’s data will be calculated. |  |
| * 1. WPM (Words Per minute) – The number of words over the time taken in minutes rounded to nearest % |  |
| * 1. Accuracy – The number of correct words divided by the total number of words typed rounded to nearest % |  |
| 1. When the program finishes, the user will have the option to save their score to the leader board. |  |
| * 1. Scores will be saved in an array of records in JavaScript. |  |
| * 1. The user will also have the option to save their name to the leader board. |  |
| * + 1. If the user does not input a name, the name shows as blank. |  |
| * + 1. Checkbox to save your name for next rounds, which will be stored in a session variable. |  |
| * + 1. Any further rounds, the name box will be automatically inputted. |  |
| 1. The top 10 scores will be shown on a scoreboard. |  |
| * 1. The scores will be calculated using a bubble sort to put the value in. |  |
| 1. When the program finishes, the user has the option to retry or exit. |  |
| * 1. The retry button will start a new game. |  |
| * 1. The exit button will return the game to its starting state. |  |

|  |  |
| --- | --- |
| 1. Clean interface that is intuitive to work with. |  |
| 1. Interactivity with text colour changes while doing a test (green for pass, red for fail) |  |
| 1. Ability to change theme to fit the user’s needs (light/dark) |  |
| 1. Must be able to be played on most standard size computer monitors. |  |