- Welcome -

Create a ‘welcome.js’ file in the correct folder. In this file, create one jQuery function that selects the HTML element with the ”liveToastBtn” ID and shows the HTML Toast element with the ID of ‘liveToast’.
This function should be run when the user clicks the ‘Like’ button in the HTML header.

Make a copy of the ‘Like’ button element in the HTML file and place it directly after the existing 'Like' button element. Change the button text to "I'm Bored" and the element’s ID to ”boredBtn”. In the same ‘welcome.js’ file, create another function that executes a GET Request to the "**https://www.boredapi.com/api/activity/**" API, once the new "I’m Bored” button is clicked.
The response from this API Request should be logged into the web console.

- Table -

Create a ‘table.js’ file in the correct folder.

Use this file to populate the Table in the ‘personal’ section as follows, **using Vanilla JavaScript**:

- You must have five rows of unique data (the data cannot be the same in each row).
- Populate the table with your chosen data relevant to the table headers.
- The Table needs to be populated from the 'table.js' file
- Icons should be used for the icon column. Here, any icons can be used, but no images are allowed. (You can use bootstrap icons)

The following functionality needs to be implemented in the ‘table.js’ file, **using jQuery**:

- When one of the images in the Table is clicked, the image must be displayed in a new window or tab in your web browser.
- The table’s odd rows need to change font color when the mouse hovers over them, changing font color back to *black* once the mouse stops hovering over them.

- About me -

Create an ‘aboutMe.js’ file in the correct folder.

In this file, you must create **a single function** to update the ‘about me’ content.

Change all the text that says "Something about me text" to something about you when the page loads - This could be anything, such as hobbies, what you like or do not like, your pet's names, or where you live (this information can be real or made up). All the "Something about me text" needs to be changed from the function.

Change the "student name" to your student name in the about me section so that it reads: Hello! I am FJ, the code master. This change should be done **in the same function** as the "something about me text" change.

When you click the content in the "something about me" section that you have updated, the **background color** for the clicked element **needs to change** to *powder-blue*, and the font color needs to change to *white*.

- Contact Information -

Create a 'contact.js' file in the correct folder.

In this file, you must create **a single function** with the following 3 variables:

- email
- telephone_number
- address

In the index.html file, in the 'contact' section, add a Button.

When this button is clicked, the following should happen in the function created above:

1. Ask the user for information.
   You must store this information in the 3 variables created in the contact.js file. Make sure that data is received from the user input, and that it is in the correct format (e.g., make sure the user enters an email address containing both a '@' and '.').
2. Change the current contact information in the HTML file (email, telephone number, and address) to the input data from these 3 new variables.
   This must be done using jQuery selectors, without adding more HTML IDs to the 'contact' section. (The HTML file cannot be changed for this)
3. Print the data entered to the console in the following format:
   From: email@emailaddress.co.za
   Contact: 0458754125
   Address: 12 Smith Road, John Park.

- Accessibility -

Create an 'accessibility.js' file in the correct folder.

In the Accessibility' section, we have a visual aid for users who struggle to read small font in the form of a Toggle Switch. When the user clicks this Toggle Switch to the 'On' position, all text on the web page within <p> tags need to be increased by 20%. When the user clicks the Toggle Switch back to the 'Off' position, all text on the web page within <p> tags needs to be decreased back to the original size.

Create a single function in the file that accomplishes this.