

Summative Assessment

Task

Design and development are two core skills for all software developers and HTML, CSS and JavaScript are core web development languages. In your assignment, you will show us your understanding and use of the Bootstrap CSS Library.

In your assignment, we'll combine all the knowledge and **the best practices** we've learned in JavaScript, CSS, and HTML to create a more polished web application. We'll start being creative as we introduce external services (like a mailing API) into our project.

Once you've decided what you want your web page to do, you'll need to create a basic design for functionality and layout. Please use **Figma**, or draw.io to create the designs (wireframe) for the **Checkout** and **Products** page.

Please note - Your assignment grades will be based on **code quality, code + file organization, HTML/CSS** (flexbox / grids), the **overall presentability**, etc.

Please download this file in the materials tab to work on this project (required).

Also, download the Marking Criteria in the materials tab to see how the project will be graded.

Here are the requirements:

1. Shop Page

1. Please change all the goblin images so each item has a unique image
2. Create a button below each item with the text "Add item".
Once clicked, let the user know the item has been added using [Bootstrap toast](#)
3. Modify the code necessary so that the localStorage stores the item image

2. Shopping Cart Modal

1. In shoppingCart.js create jQuery logic to decrease item quantity.
2. When the modal is opened, please modify the necessary HTML so that:
The **item image** is displayed in the modal - in addition to the item name, price, and the total quantity
3. **Bug** - Currently when we subtract the item quantity, the number goes below 0 and a negative number is displayed. Fix this bug so that the subtract button is **disabled** when the item quantity is 1
4. Create a field at the bottom that calculates the total cost of all items (each item price * item quantity)

3. Checkout Page

1. Create a checkout page that displays all the item details that are in the shopping cart (please follow folder structure as explained in #6 Fetch Exercise)
2. Create a form using [Bootstrap](#) with the fields below:
 1. First and last name
 2. Email address
 3. Phone number
 4. Address
 5. Credit card number
 6. Expiry
 7. 3-digit security code
3. Each form field must have a **validator** (take a look at [regex](#)). If one of the fields is incorrectly inputted, let the user know which field is incorrect.
4. When every form field is correctly inputted and the user submits the form, trigger an emailer API (4th requirement)

4. Emailer Service

1. Find an external (3rd party - [example](#)) API that sends an email confirmation when the user submits the form
2. The email should contain all the information provided by the user - including the items that were purchased
 1. For credit card information, please blur out the first 12 digits and display the last 4 (ex: "1234-5678-9999-8511" should be "**** * 8511")

5. Database Service

1. Create an account for Google's Firebase Service ([reference here](#))
2. Select and create their "**Real-time Database**" and integrate their API into your project
3. Store the user's purchase data in the database when the user makes a purchase at the checkout page

6. Fetch Exercise (Using Async/Await)

1. Create a new Products page and add the new products link to the Header and footer both in the index.html and in the jQuery header & footer (along the Shop, Contact, About)
2. The Products page needs its own CSS file (Please use **flexbox**)
3. [Use this API \(https://raw.io\)](https://raw.io) for a list of games (at least 5):

1. Product name
2. Product release date
3. Product genre

At least **two** more details about the product (picture of game would be nice).

4. Please note: use the **async/await keyword** when using fetch and handling JSON. **Do not** use: `.then()` `.catch()` or `.finally()` chains. Reference [here](#) to see how one function handles the call and then we use another function to process our data.
5. Please continue the folder structure so products.html will go in the pages folder, and products.css will go in the styles folder, and finally the products.js should be in the scripts folder.

7. Pages (products & checkout) should have a minimum of three breakpoints (desktop/iPad/mobile)

8. Create a folder named extras and include:
 - Screenshots of received Email containing product data (Please include "Sent From" "Sent To" section as well)
 - Firebase Real-time Database containing product data
 - Please use **Figma**, or draw.io to create the designs (wireframe) for the **Checkout** and **Products** page

To Submit Project:

- Make a copy of your working folder and then rename with your first and last name.
- Delete git folder (hidden folder)
- Please Zip all the contents and name the file **<student_name>.zip**

Need Help?

Please look at all the links and references provided above. A large part of software development is doing one's own research / Googling. Look at how things/logic were implemented by developers worldwide on websites like Stack Overflow, Medium, Mozilla, GeeksForGeeks, Quora, Reddit, etc.

Feel free to reach out on discord where you can chat with instructors and your peers as well.

Best of luck!