



Assignment 5

[20 marks]

Due: **Sunday, Nov 28 2021 @23:59**

Note: This is an individual assignment.

Objective

- Practice HTML Form, CSS, JS validation
- Deploy to matrix server and publish the website

Overview

You are going to create an imaginary online store, applying the knowledge of HTML5, CSS, and JavaScript. The task includes two HTML5 pages (**a5Store.html**, **a5ContactUs.html**) and some supporting HTML pages, with proper CSS for layout and styling. JavaScript will be applied for client-side validation. All the pages will be linked (as specified below) to your website on Matrix, which was created in A3 and further improved in later works.

What will your store sell? Your store can sell whatever you want, and you can use this as a way to say something about your personality and interests, such as clothing, jewelry, second-hand clothing, make-up, cakes, tea, and computers. It can be fun, funny, or serious. Ideally, you'll build something in this assignment that you'll be proud to show in interviews when applying for co-op positions.

Working Note:

1. When you work locally, make a copy of A4 (including A3), and make it as the folder for A5. The point is: do not change previous work.
2. When you copy/ upload A5 to matrix, you only upload A5 work with the proper file structure to the existing matrix /public folder.
3. Add links to **a5Store.html**, **a5ContactUs.html** from home page (index.html) menu and won't change anything else. The website menu is updated as Figure 1. You may have different menu items, but make sure the links to A5 are available. Also this will change the timestamp on index.html, which is fine and understandable.

Web Programming Assignment

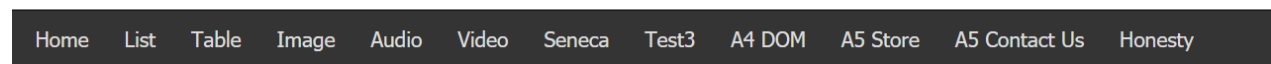


Figure 1: Navigation menu for A5

Specifications:

1. File Structure:

Organize your html files, CSS files, and JavaScript files in separate folders (Figure 2). Yours may be slightly different from that in Figure 2. However the required files (file names starting from a5...) should follow the same structure in Figure 2.

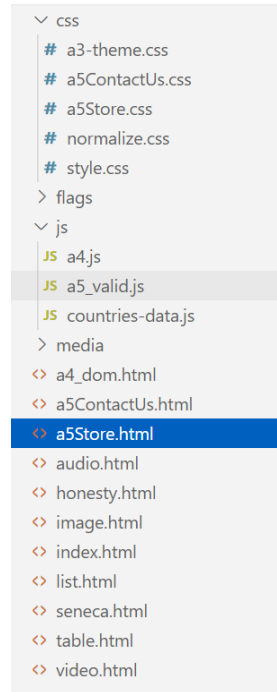


Figure 2: A5 File structure

2. Page a5Store.html

[6 marks]

This is the page you display your items for sale.

- (1) Implement the 3-column layout in Figure 3. The specific height and width of each panel is for you to define. Figure 3 is abstract schematics meant to illustrate the general layout of the pages.
- (2) Your **CSS** for styling and layout should be in external file (**a5Store.css**, Figure 2). You are allowed to use internal and inline css when needed. External CSS is required though. Name it **a5Store.css**, and store it under folder css, refer to Figure 2.
- (3) Suggestion for each panel in Figure 3 follows. However, you are encouraged to design and implement your page in any way you like, as long as following the abstract schematic 3-column layout in Figure 3.
 - a) Header: your company's logo and header
 - b) Navigation Bar: the navigation menu for your website (Figure 1).
 - c) Aside: navigation links to each department, different topics, etc.
 - d) Main: display your items in a **well-organized way**. Display 4 or more items showing your complete design (e.g., grid, list, column, etc.). Each item should have the information with:
 - (i) item picture
 - (ii) item brief description. If user clicks this text, it will bring a page with detailed information for that item. You DO NOT need to write detailed information for that item. However, you DO NEED to create a page for each item's detailed description. You can simply write one line in this page, e.g, "detailed information for item ..."Basically, I will click this brief description and expect to open a page with some information.

- (iii) price.
 - e) Ads: could be your store promotion ads, etc.
 - f) Footer: (1) **Your name**, (2) copy right, and (3) time stamp (**required**).
- Do not hard code time stamp. It should dynamically show:

This page was last modified at "the date and the time you finished/ completed the page."
 Do not update your files after the due date if they are complete because that will change the timestamp and may be considered late.

- (4) the page should have a background of some color;
- (5) the text of the header and footer should be in a different color than the rest of the panels.
- (6) Be creative and have fun!

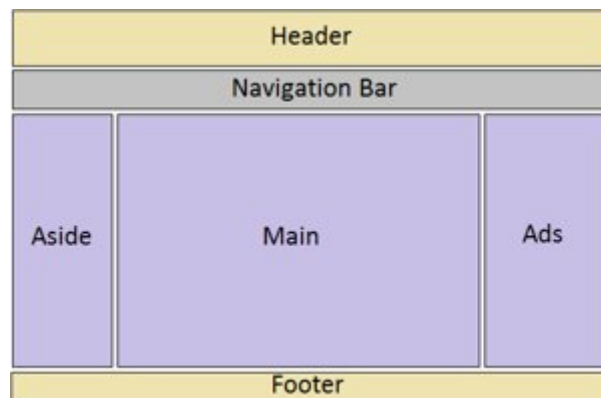


Figure 3: 3-column layout for a5Store.html

3. Page a5ContactUs.html

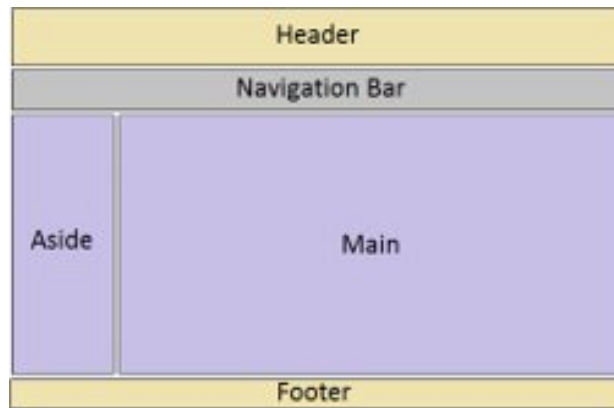


Figure 4: 2-column layout for a5ContactUs.html

Page **a5ContactUs.html** provides your store contact information and allows user to fill in his/her information.

- (1) It should be designed as the abstract layout in Figure 4. The specific height and width of each panel is for you to define.
- (2) Your **CSS** for styling and layout should be in external file (**a5ContactUs.css**, Figure 2). You are allowed to use internal and inline css when needed. External CSS is required though. Name it **a5Store.css**, and store it under folder css, refer to Figure 2.

- (3) Suggestion for each panel in Figure 4 follows. However, you are encouraged to design and implement your page in any way you like, as long as following the abstract schematic 2-column layout in Figure 4. [4 marks]

- a) Header: your company's logo and header
 - b) Navigation Bar: the navigation menu for your website (Figure 1).
 - c) Aside: Display your company's contact information, e.g., phone, address, etc.
 - d) Main: a form to collect user's input. Will be specified below in "(4) Form Requirements" section.
 - e) Footer: (1) **Your name**, (2) copy right, and (3) time stamp (**required**).
- Do not hard code time stamp. It should dynamically show:

This page was last modified at "the date and the time you finished/ completed the page."

Do not update your files after the due date if they are complete because that will change the timestamp and may be considered late.

(4) Form Requirements:

[10 marks]

The main panel in Figure 4 is a form to take users' input. You can layout and style the form as you wish. You need to implement the form fields and client-side validation as follows. If some features or attributes are not specified, you can determine it by yourself.

- a) Provide a heading for this form (above the form), such as "Contact Us".
- b) All fields are required. Also display this information (i.e., All fields are required) below the form heading and above the form.
- c) The form will have the following fields:

(i) **Your Name:** this is a hidden field. Set your real name as value for this field. E.g.,

```
<input type="hidden" name="fullName" value="Your real name" />
```

(ii) **Student Number:** this is a hidden field. Set your real Student Number as value for this field.

(iii) **User Name:** text. **Validation:** Must start with a cap and only alphabet allowed.

Note: validate using JavaScript, not HTML5 pattern attribute.

(iv) **Gender:** radio. The value must choose from one of the following: Male, Female, Prefer not to share

(v) **Age:** number. Must be in between 0 - 999.

(vi) **Province:** Dropdown list with options of provinces and territories in Canada. Default is Ontario.

(vii) **Phone number:** text. **Validation:** in the format: (NNN)NNN-NNNN. E.g., (416)111-2222. **Note: validate using JavaScript, not HTML5 pattern attribute.**

(viii) **Email:** use html5 input type for default validation.

(ix) **Comment:** textarea. **Validation:** Cannot be blank or only white space.

Note: validate using JavaScript, not HTML5 pattern attribute.

d) Client-side Validation:

(i) Client-side validation is implemented in external JavaScript file **a5Valid.js** (Figure 2).

(ii) When a field input is invalid, **highlight this field** (e.g., set red border) and display error messages **beside** or **below** the field. Windows alert() message box will NOT be accepted for error message display.

(iii) You can display the error messages all at once when submit the form or

(iv) Display the error messages immediately after the field is changed (if applicable).

Hint: use the onchange or onfocusout etc. event handler to call JavaScript functions.

(v) Error messages must be meaningful, which will help user understand what was wrong and how to fix it.

E.g., good error message like,

“Display phone number in the format: (NNN)NNN-NNNN. E.g., (416)111-2222.”

As the following error message will be considered bad,

“invalid input”.

Because it’s not clear what’s wrong and how to fix it. Users get confused.

(vi) When no errors, submit the form to <https://httpbin.org/post> .

Other Requirements

(1) Your code will be evaluated for correctness of syntax and appropriate usage

(2) You will be evaluated based on creativity and effort.

(3) Validate your HTML code using <https://validator.w3.org/>

(4) Validate your CSS using <https://jigsaw.w3.org/css-validator>

Submission:

You do not need explicitly submit the work. Leave it on Matrix account, which will be checked after due date. **DO NOT need to email me or submit to blackboard.**

Late submission:

10% penalty of the assignment each day, up to 5 school days. It will not be accepted after 5 school days late and will be marked zero for this assignment.