

Web Level 1 - Assignment #5

Overview

In this assignment, you will create a Survey Form about any topic. You will write HTML and CSS to check for simple validation, submit the form, and then style the form.

Example

APP Survey Form

Thank you for your feedback!

Name

Email

Age

Would you recommend APP to a friend?

Definitely Maybe Not Sure

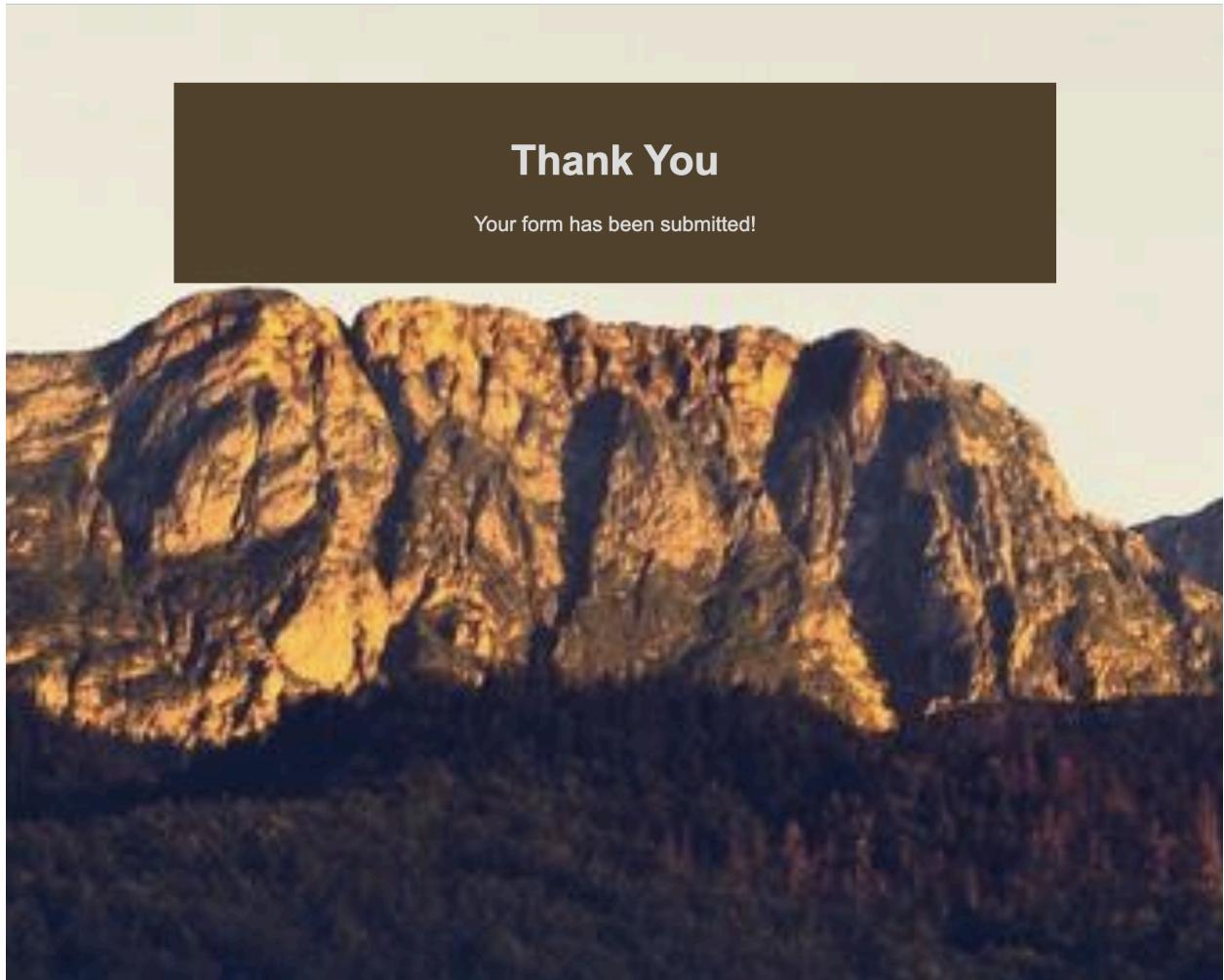
What is your level of education?

Select One

What would you like to see improved?

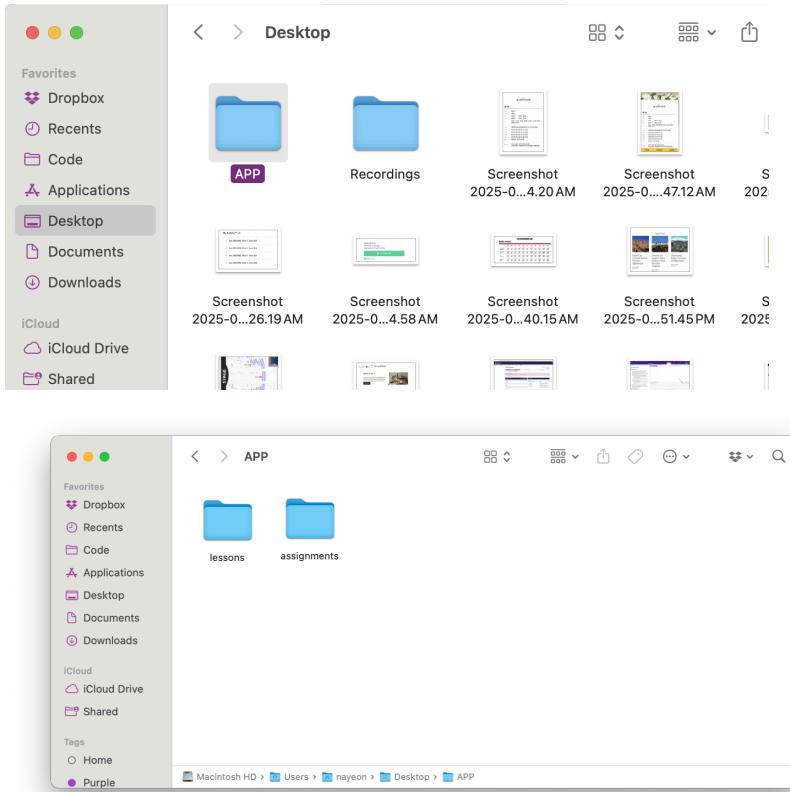
Lectures Labs Assignments Community

Comments



Requirements

1. Pick a topic that you want to create a survey about. It can be any topic that you are curious about.
2. In the **APP folder** you created on your **Desktop** in Lesson #1, make sure you have a folder named **assignments**. In that folder, create another folder named **assignment05**. You can do this on your File Explorer or through VS Code.



3. In VS Code, open up the same **APP** folder that you've been using for the lessons and previous assignments. Create a new file named **index.html** in the **assignment05** folder you just created.
4. Create another file named **confirmation.html** in the **assignment05** folder. This file will be the page shown after a user submits the form.
5. Create a CSS file named **styles.css**. Write any shared CSS in this file. You can still write CSS inside a HTML file for that specific file.
6. Link the **styles.css** file in both HTML files so that both HTML files use the CSS file.
7. Write HTML code so that the browser's tab displays “Assignment 5” in both HTML files.

Form Structure (index.html):

1. Create a **<div>** tag. This element will be the container element that contains the heading text, description, and form. Give it an id **container**.
2. In this container, display the survey title and a short description.

3. In this container, create a form.
4. In the form:
 - a. Display at least one text input where users can type text.
 - b. Display at least one email input where users can type their email address.
 - c. Display at least one number input where user can type in a number only.
 - d. Display at least three radio buttons.
 - e. Display at least three checkboxes.
 - f. Display at least one dropdown that has at least three options to select from.
 - g. Display at least one textarea that allows users to write multi-line text.
5. All input elements must have the appropriate `name` attribute.
6. Radio buttons, checkboxes, and dropdown options must have the `value` attribute.
7. All text describing the input must have a `label` tag around it and be linked with its input by using the correct `for` and `id` attributes.
8. For the text and email input fields, set `placeholder` text that gives an example for each field.
9. Display two buttons - one to `submit` the form and another to `reset` the form.
10. When the form is submitted, `validate` the inputs:
 - a. The text input and email inputs are **required**. An error is shown if they are not filled out.
 - b. The email input must have at least **three** letters. An error is shown if the email has less than three letters.
 - c. The number input must not be below **zero**. An error is shown if a negative number is entered.

Confirmation Page Structure (confirmation.html)

1. Hint: You can use a lot of the same code for this page from the main form.
2. Create a `<div>` tag. This element will be the container element that contains the heading text, description, and form. Give it an id `container`.
3. In this container, display a heading and a short description.
4. When the form is submitted with all inputs entered correctly, show this page.

Styling the form (index.html)

1. The `index.html` and `confirmation.html` will share some CSS. Review the assignment example to see what CSS will be shared. Write shared CSS into the shared `styles.css` file.
2. Add a **background image** to the browser. You can find free beautiful images at: <https://unsplash.com/>
3. Add a **background color** on the container `<div>` (the one with an id `container`).
4. The background color of the container `<div>` is slightly **transparent**. But make sure the text color is **not** transparent.
5. Set the container `<div>` to have a **width of 600px**.
6. The container `<div>` is **horizontally centered**.
7. Add some spacing **inside** the container `<div>`.
8. Change the **width** of the text, email, and number input tags to be wider than the default.
9. Like in the example, display the label of name, email, the number input, the dropdown, and the textarea **on its own line, above their input fields**.
10. Like in the example, display the question of the radio buttons **on its own line, above all the radio buttons**. But each individual radio button's labels must be **next to** the radio button.
11. Like in the example, display the question of the checkboxes **on its own line, above all the checkboxes**. But each individual checkbox's labels must be **next to** the checkbox.
12. All elements and text need to be readable (i.e. no dark text colors on dark backgrounds).

Styling the confirmation page (confirmation.html)

1. The `index.html` and `confirmation.html` will share some CSS. Review the assignment example to see what CSS will be shared. Write shared CSS into the shared `styles.css` file.
2. The browser has the same **background image** as it does in `index.html`.
3. The container `<div>` has the same **background color** as it does in `index.html`.
4. The background color of the container `<div>` is slightly **transparent**. But make sure the text color is **not** transparent.
5. The container `<div>` has the same **width** and **spacing** as it is in `index.html`.
6. The container `<div>` is **horizontally centered**.

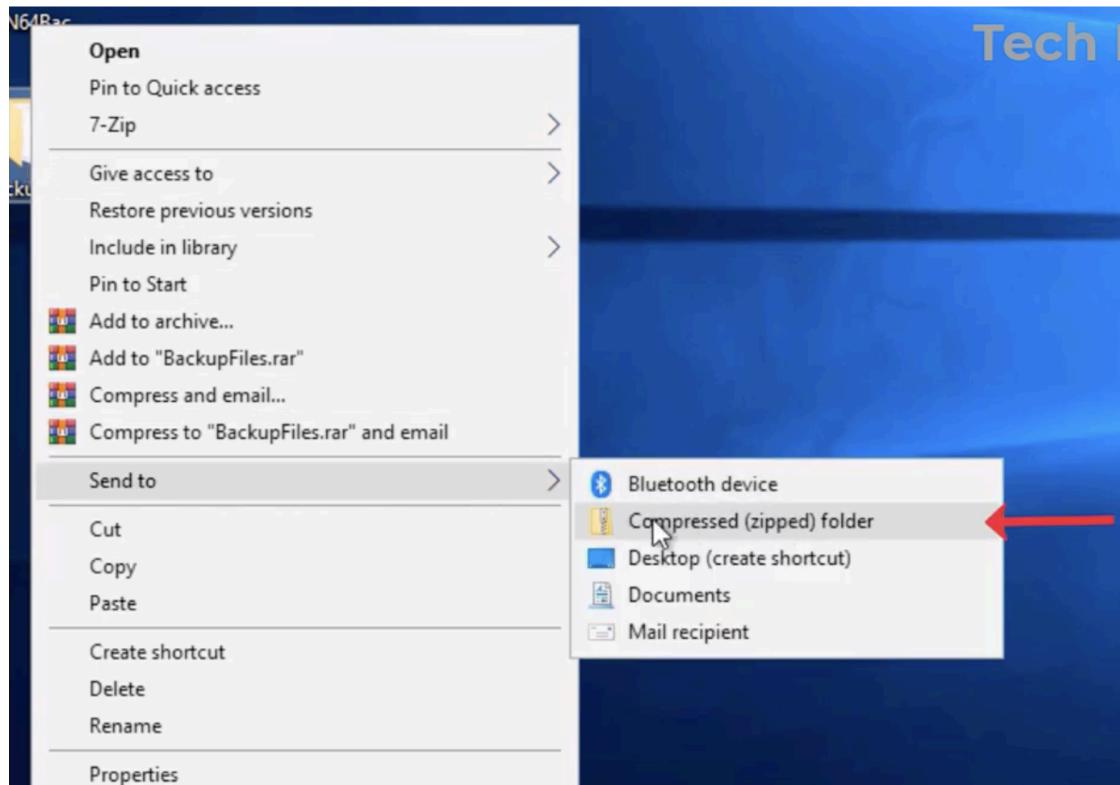
HTML Validation

Even if everything looks correct, it's possible that the code still has some errors. After completing the assignment, make sure that:

1. The code is indented properly. A general rule of thumb for indentation is:
 1. Indent whenever an HTML tag is inside (nested) in another tag. (e.g. when a `<h1>` tag is nested inside a `<header>` tag, indent the `<h1>` tag)
 2. Indent the lines after the curly brace `{` when writing a CSS rule.
2. Copy paste your code into a HTML validator like https://validator.w3.org/#validate_by_input to check for any HTML validation errors and fix them. Warnings are ok, but make sure to fix any errors.

Submission

1. When everything looks good, it's time to **submit your assignment!** To submit this assignment, open your File Explorer and navigate to the `assignment05` folder that contains your assignment file.
2. Right-click on the `assignment05` folder and click **Send to**, then click on **Compressed (zipped) folder**. This will compress your folder into a `.zip` file.



3. Rename this **.zip** file to add your name to it, for example **nayeon-kim-assignment05.zip**. Send this **.zip** file to your LA on WhatsApp. Do not share with the subgroup. Send it directly to only your LA. Alternatively, you can email this file to your LA. Contact your LA to get their email address if you prefer to email it to them. That's it. Congratulations on finishing your first assignment!
4. Also share a screenshot of your assignment (not the code, just what is shown on the browser) to your LA WhatsApp subgroup to show your work! 😊