

Project Instructions

Open the project files in your favorite text editor and preview them in the browser. Also, it's a smart idea to test your project in multiple browsers!

Github

- Have a GitHub account and create a new repo for this project.
- Create a README.md file for your repo that explains what the projects is and anything your user or fellow developers might need to know to use the project.

Download the project files

- You've been supplied several files for this project:
 - ***index.html*** - contains the initial HTML page. Ideally, you want to make as few changes to this file as possible. You'll have to add the necessary script tags to the link your JS, and you'll need to modify the HTML for the "Job Role" task below. But other than that, try to do everything you can with JavaScript, rather than modifying the HTML.
 - ***css/style.css*** - contains the styles for this project. Feel free to play around with some of the styles here to make it your own. You can experiment with things like the background color, font, transitions, animations, box shadows and text shadows. But the basic layout and positioning of elements should not be changed.

Add the necessary files

- Create a folder and name it js.
- js/script.js - Create a folder named js and add a script.js file to it. Then link that file to index.html with a script tag.
- jquery - Link index.html to the latest version of jQuery either by including the jQuery files directly in the "js" folder, or by linking to a jQuery CDN. You should be able to find both with a good Google search, but if you get stuck or have questions, be sure to reach out on Slack.
- css/reset.css or css/normalize.css - This is not a project requirement, but using a reset or normalize CSS file is an important common standard in web development as it helps zero out the initial styles that different browsers add to their page by default. So you are encouraged to take this extra step. You should be able to find what you need with a Google search.

jQuery

- Add jQuery to your project by either including the jQuery source files in your directory or by using a CDN link.
- Utilize jQuery while coding the functionality for your form.

Note: You don't have to use jQuery for every single aspect of your project, but at the very least, your project should include and utilize at least some jQuery. Although, you should keep in mind that

consistency is important. Typically, projects are built with either a jQuery or a "vanilla" JavaScript approach. And it's not a great idea to continuously bounce back and forth between the two in a single project.

Set focus on the first text field

- When the page first loads, the first text field should be in focus by default.

"Job Role" section

- Include a text field that will be revealed when the "Other" option is selected from the "Job Role" drop down menu.
- Give the field an id of "other-title," and add the placeholder text of "Your Job Role".

NOTE: You'll need to add the "Other" job role input directly into the HTML and hide it initially with JS in order to get this feature to work when JS is disabled, which is a requirement below.

"T-Shirt Info" section

- Until a theme is selected from the "Design" menu, no color options appear in the "Color" drop down and the "Color" field reads "Please select a T-shirt theme".
- For the T-Shirt "Color" menu, after a user selects a theme, only display the color options that match the design selected in the "Design" menu.
 - If the user selects "Theme - JS Puns" then the color menu should only display "Cornflower Blue," "Dark Slate Grey," and "Gold."
 - If the user selects "Theme - I ♥ JS" then the color menu should only display "Tomato," "Steel Blue," and "Dim Grey."
- When a new theme is selected from the "Design" menu, both the "Color" field and drop down menu is updated.

"Register for Activities" section

- Some events are at the same day and time as others. If the user selects a workshop, don't allow selection of a workshop at the same day and time -- you should disable the checkbox and visually indicate that the workshop in the competing time slot isn't available.
- When a user unchecks an activity, make sure that competing activities (if there are any) are no longer disabled.
- As a user selects activities, a running total should display below the list of checkboxes. For example, if the user selects "Main Conference", then Total: \$200 should appear. If they add 1 workshop, the total should change to Total: \$300.

"Payment Info" section

- Display payment sections based on the payment option chosen in the select menu.

- The "Credit Card" payment option should be selected by default. Display the #credit-card div, and hide the "PayPal" and "Bitcoin" information. Payment option in the select menu should match the payment option displayed on the page.
- When a user selects the "PayPal" payment option, the PayPal information should display, and the credit card and "Bitcoin" information should be hidden.
- When a user selects the "Bitcoin" payment option, the Bitcoin information should display, and the credit card and "PayPal" information should be hidden.

NOTE: The user should not be able to select the "Select Payment Method" option from the payment select menu, because the user should not be able to submit the form without a chosen payment option.

Form validation

- If any of the following validation errors exist, prevent the user from submitting the form:
 - Name field can't be blank.
 - Email field must be a validly formatted e-mail address (you don't have to check that it's a real e-mail address, just that it's formatted like one: dave@teamtreehouse.com for example).
 - User must select at least one checkbox under the "Register for Activities" section of the form.
 - If the selected payment option is "Credit Card," make sure the user has supplied a Credit Card number, a Zip Code, and a 3 number CVV value before the form can be submitted.
 - *Credit Card field* should only accept a number between 13 and 16 digits.
 - *The Zip Code* field should accept a 5-digit number.
 - *The CVV* should only accept a number that is exactly 3 digits long.

NOTE: Don't rely on the built in HTML5 validation by adding the required attribute to your DOM elements. You need to actually create your own custom validation checks and error messages.

NOTE: Avoid using snippets or plugins for this project. To get the most out of the experience, you should be writing all of your own code for your own custom validation.

NOTE: Make sure your validation is only validating Credit Card info if Credit Card is the selected payment method.

Form validation messages

- Provide some kind of indication when there's a validation error. The field's borders could turn red, for example, or even better for the user would be if a red text message appeared near the field.
- The following fields should have some obvious form of an error indication:
 - Name field
 - Email field
 - Register for Activities checkboxes (at least one must be selected)
 - Credit Card number (Only if Credit Card payment method is selected)
 - Zip Code (Only if Credit Card payment method is selected)
 - CVV (Only if Credit Card payment method is selected)

NOTE: Error messages or indications should not be visible by default. They should only show upon submission, or after some user interaction.

NOTE: Avoid use alerts for your validation messages.

NOTE: If a user tries to submit an empty form, there should be an error indication or message displayed for the name field, the email field, the activity section, and the credit card fields if credit card is the selected payment method.

Form works without JavaScript – Progressive Enhancement

- The user should still have access to all form fields and payment information if JS isn't working for whatever reason. For example, when the JS is removed from the project:
 - The “Other” text field under the "Job Role" section should be visible.
 - All information for Bitcoin, PayPal or Credit Card payments should be visible.

CSS styles

- It is not required, but you are encouraged to experiment with things like the color, background color, font, transitions, animations, box shadows and text shadows. So dive into the CSS file and see if you can make this project your own with a few adjustments to the styles. But the basic layout and positioning of elements should not be changed.

Add code comments

Cross-Browser consistency

- Google Chrome has become the default development browser for most developers. With such a selection of browsers for users to choose from, it's a good idea to get in the habit of testing your projects in all modern browsers.

NOTE: Sometimes getting started is the hardest part

- It's not uncommon to feel overwhelmed and confused when beginning to build a project. If you feel this way, try not to get too focused on seeing the project as a whole. Instead, just take it one small piece at a time. After familiarizing yourself with the instructions, start by downloading the source files, and creating a GitHub repo to store them. That is always a great place to start. Then just start tackling the project one small step at a time. Remember, your first attempt isn't likely to be perfect, and it doesn't have to be. As coders, we focus first on creating something that works. And then we refactor and optimize on later iterations.