Description

Short Description (used on project cards, visible to students even if unpublished)

Web forms appear everywhere online: forms allow users to order books, sign up for web sites, and post to Facebook. This project challenges you to build a responsive, mobile-first registration form using a variety of HTML form elements.

Long Description (used on details page)

In this project you'll build a responsive, mobile-friendly registration form using a wide variety of HTML form input types and attributes. Using the supplied mockup files, you'll build a mobile and desktop version of the form using media queries, and a "mobile-first" approach.

NOTE: The Front End Web Development Techdegree is meant to train you in HTML, CSS and JavaScript, and let you practice and show your mastery of these fundamental building blocks of the web. Because of that, please avoid using frameworks like Bootstrap, Foundation, Skeleton, and so on for this project. Even though you may end up using frameworks like these professionally, you still need to know and be able to implement designs with your own knowledge of HTML, CSS and JavaScript.

In addition, please avoid submitting any projects that rely on a server-side technology like PHP or Ruby on Rails.

8.0 hours to complete

Prerequisites

This is how prerequisites will be displayed:

- **As with the previous projects, you'll submit your finished working using GitHub. If you need a reminder on how to use GitHub and GitHub desktop to create a new repository check out this workshop: Share Your Projects WIth GitHub Desktop**
- **Download the project files. We've supplied two files for you to use:**
  - mobile-form.png is an example of how the layout should look on a mobile phone that’s 320 pixels wide.
  - desktop-form.png is an example of how the layout should look on tablet and desktop screens that are at least 768 pixels wide.

Skills

- Responsive Design
- HTML forms
  - Variety of form fields
  - Form labels, fieldsets and legends
  - Mobile input fields
  - Placeholder text
- CSS
- Web Fonts

Instructions

This is how instructions will be displayed:

- **For this project we will be checking the design in Google Chrome version 48 or higher. Different browsers display forms slightly differently so please make sure it looks correct in Google Chrome before submitting.**
- **Build the layout using a mobile first design:**
  - Make sure the HTML file includes the viewport meta tag in the head of the document, see Configuring the Viewport to understand why and how to add this tag.
  - Look at the provided mockup (mobile-form.png) and add the same content into your index.html file.
- **NOTE:**
  - It's okay if your checkboxes and radio buttons don't match the look of those in the mockups. It's perfectly fine to use the standard default radio buttons and checkboxes.
- **Create the form structure:**
- Only use one `<form>` tag. The `<form>` tag should contain all the form elements. Add a fieldset and legend for each of the following sections:
  - "Contact Information" section of the page, and
  - The "Newsletter" section of the page
- **Note:** You don't need to make a functioning form -- that is, it doesn't have to do anything when the form is submitted. To do that, you'd need to add some server-side programming to actually process the user's input.

- **Make sure you include the following form field types:**
    - text input
    - email input
    - telephone input
    - select menu
    - checkboxes
    - radio buttons
    - textarea
    - submit button
- **Form fields should include the following attributes:**
    - **`input`**: should include `id`, `type` and `name` attributes.
    - **`select`** and **`textarea`**: should include `id` and `name` attributes.
- **NOTE:**
    - Checkboxes should have identical `name` attributes but unique value attributes
    - Radio Buttons should have identical `name` attributes but unique value attributes
- **Add labels to each form element using the HTML `<label>` tag. The text inside the labels should match the names of the form fields in the mockups.**
    - Make sure you properly pair each `<label>` element with its corresponding form control via the `for` attribute. See the link above for an example. And don't forget about the textarea element. That needs a functioning label too.
- **NOTE:** Remember that to associate a label with a form input element, the label's "for" attribute should match the input's *unique* `id`.
- **Use the input field's `placeholder` attribute to add the text "required" to:**
    - the Full Name field
    - the Email address field
- **Once you have everything in place for the mobile layout, use a media query to add a breakpoint to adjust the layout for wider tablet and desktop screens.**
    - Match the design as it should look on a tablet or desktop that is 768px wide using the desktop-form.png mockup.
    - Use a mobile-first approach by writing your media queries using the `min-width` property in your CSS.
- **Once all your breakpoints are in place, double check your layout matches both the mockups.**
    - Check that the label text position matches both mockups:
        - *Mobile*: Text should be above the form field.
        - *Desktop*: Text should be to the left side of the form field.
- **Use a Google Font for the text. The design uses the "Merriweather" font but, you can use any Google Font that you'd like.**

- **Add `:focus` states to the form for when a user clicks or tabs into a text field.**
- **Make sure to check your code is valid by running it through an HTML and CSS validator.**
  - Links to the validators can be found in the Project Resources. This will help you spot any errors that might be in your code.
  - There are a few exceptions that you don't need to fix:
    - Don't worry about any warnings, we just need you to check any errors that might be there.
    - If CSS validator flags use of calc, vendor prefixes or pseudo-elements/pseudo-classes these errors should be ignored.
    - If HTML validator flags use of pipe ('|') in Google font links/URLs this error can also be ignored.
- **Before you submit your project for review, make sure you can check off all of the items on the Student Project Submission Checklist. The checklist is designed to help you make sure you've met the grading requirements and that your project is complete and ready to be submitted!**
-

- **NOTE:** A good practice is to check your project for cross browser compatibility. Making sure that it looks and functions correctly in multiple (at least three) browsers is an important part of being a top-notch developer. If you want, leave a comment to the project reviewer about which browser(s) the project was checked to ensure they are seeing things as you have designed them.
- Some browser options:
  - Google Chrome
  - Mozilla Firefox
  - Internet Explorer/Edge
  - Safari
-

Extra Credit

This is how Extra Credit will be displayed:

- **Additional placeholder text for other text fields.**
- **Use the `required` attribute to add HTML5 validation to make sure that required fields are filled out and input is formatted correctly.**

- **Add at least the following additional styling enhancements to the form, feel free to add extra styling but leave the basic layout the same as the mockup.:**
    - Change the background color for at least ONE of the main sections of the site.
    - Uses CSS transitions for focus states.
- **Use CSS transition effects for highlighting the look of the form fields.**
    - For example, make a background color fade into view when the user clicks into a text field, and fade out when the user clicks or tabs out of the field.
- ---
- **NOTE:**
    - To get an "Exceeds Expectations" grade for this project, you'll need to complete **each** of the items in this section. See the rubric in the "**How You'll Be Graded**" tab above for details on how you'll be graded.
    - If you’re shooting for the "Exceeds Expectations" grade, it is recommended that you mention so in your submission notes.
    - Passing grades are final. If you try for the "Exceeds Expectations" grade, but miss an item and receive a “Meets Expectations” grade, you won’t get a second chance. Exceptions can be made for items that have been misgraded in review.
- ---