SCAD Webdev Test Project

Specifications

Thank you for your interest in the Web Developer II position. This test project will enable us to evaluate your fundamental web development skills, ability to follow instructions and attention to detail. This project focuses primarily on front-end development, but also includes some back-end development. Unless otherwise specified in this document, the tools, techniques, frameworks and any third-party scripts you use to accomplish this project are at your discretion.

We strongly recommend you review this document in its entirety before beginning this project.

Objectives

- Create a one-page test website according to the mockups provided and adhering to the functional specifications in this document. This includes all HTML, CSS and JavaScript files necessary to meet requirements.
- Create functional PHP script(s) to create the database necessary for storing email signups.
- Create functional PHP script(s) to manage the intake and output of email sign-up form data.
- Create a basic administrative web page to display the results of email sign-up form submissions according to the specifications in this document.
- Create a README.md file in your repository containing any feedback, the amount of time required to complete this project, and explanation of why you chose the techniques and frameworks you used.

Note: You do not need to host the website -- when you are finished, commit the entire website folder to a public repository on your Github account. We will download the repository and install the website locally to evaluate it.

When the finished product is available on Github, email the link to the repository to Erin Findlay at efindlay@scad.edu. If you have any questions about these directions or the specifications, please contact Erin.

The deadline for product delivery is end of day Monday, June 24. If you are unable to submit a finished product by this deadline, submit what you *have* completed with an explanation of what is unfinished and why.

Assumptions

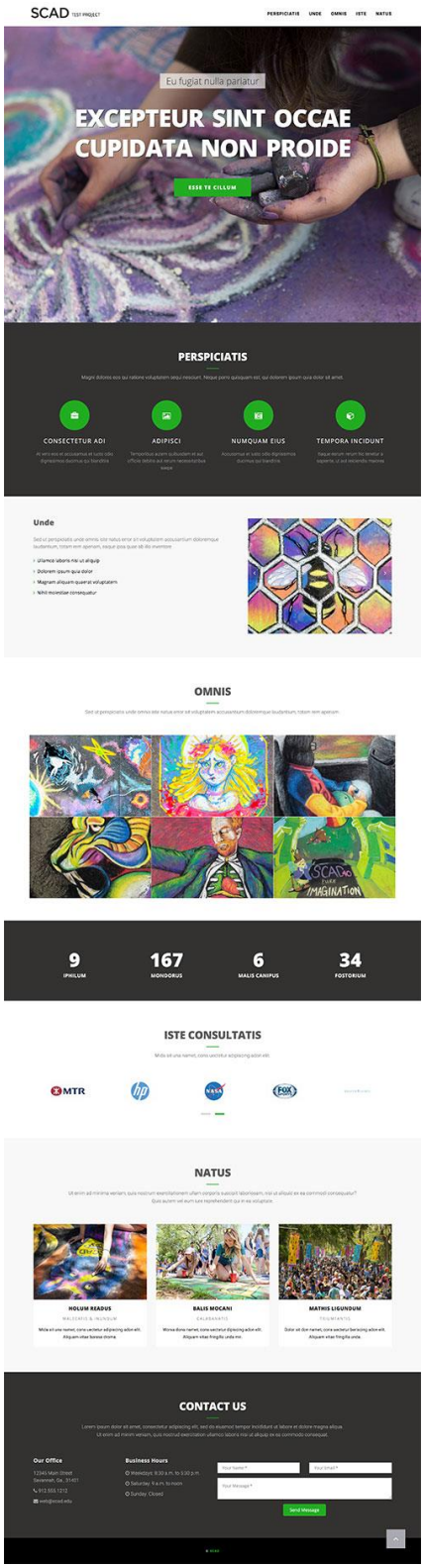
- The assets have been provided. You may modify images as necessary to meet requirements.
- Use HTML5 semantic elements when possible.
- The website should be responsive, displaying well and functioning properly at all breakpoints.
- The website should display and function as specified in all recent versions of Chrome, Firefox and Edge on a PC; on Chrome, Firefox, Safari on a Mac; and on iOS and Android devices.
- The website should conform to web accessibility standards, specifically WCAG 2.0 Level AA.
- There should be no error messages in the browser developer console log.
- The total in-browser page load size of the website including HTML, images, CSS and JavaScript should be less than 5MB. This does not include back-end scripts or the administrative page. You may optimize images to meet this requirement, but a high image quality should be maintained.
- You are encouraged to use subtle CSS animations that activate as the user scrolls. Such animation is often used to introduce motion as the page elements subtly move into position as they appear on the page. Example: https://atomicobject.com. You do not need to use the same CSS/JavaScript libraries as the example site.
- If you use third-party scripts that are available by referencing a CDN (e.g., Google Fonts, Bootstrap, jQuery, etc.), use the CDN. Do not include those scripts in your repository unless a CDN-hosted version is unavailable.
- Do not minify or otherwise obfuscate your custom JavaScript, HTML, CSS or PHP files.
- You may use a CSS pre-processor (SASS, LESS, etc.) but it is not required.
- How you structure the contents and sub-folders of your docroot folder is at your discretion.
- Do not use a content management system.

Evaluation criteria

Your work will be evaluated by the following criteria:

- Does the finished product visually match the mockups?
- Does the website function as requested?
- Is your custom code clean, understandable and following basic coding standards?
- Is the website cross-browser/cross-platform compatible?
- Does the website display properly at all breakpoints?
- Does the site yield console errors?
- Does the PHP successfully create the database as requested?
- Does all JavaScript function as requested?
- Does the administration page function as requested?
- Did you follow all instructions in this document?

The Design



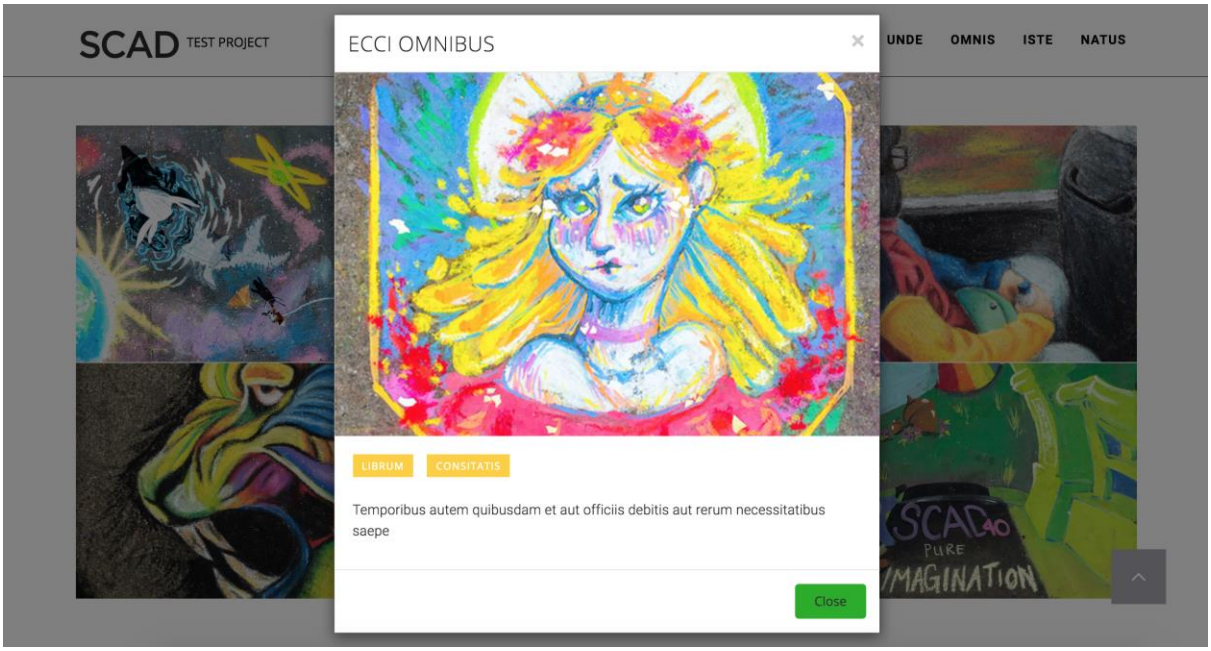
The mockups for desktop and mobile are in the *assets* folder.

Details

- This is a full-width layout. The hero image should extend the full width of the window. The content below the hero should be constrained to 1180px, although the background colors should extend full width.
- All fonts are available from Google Fonts. They are:
    - Open Sans
    - Roboto
- It is up to you to deduce font sizes, font colors, background colors, margins, padding and other CSS based on the mockups.
- The header should be sticky, persisting at the top of the browser window as the user scrolls.
- The nav items in the header are anchor links. Clicking on them should cause the browser to scroll smoothly to the corresponding section of the page.
- Icons throughout the page can be obtained from the Font Awesome library. They are font icons.
- The green button in the hero region should have a rollover state (the hover design is at your discretion). Clicking on it should cause the page to scroll smoothly to the "Perspiciatis" section.
- The green circles in the "Perspiciatis" section should be created via CSS. They do not link to anything.
- The image in the “Unde” section should be a carousel of multiple images. The forward and back buttons are from the Font Awesome library. You can build this carousel from scratch or you can use an existing JavaScript library.
- The images in the “Omnis” section have hover states and clicking on them should trigger a modal to appear (see next page for the style of the hover state and the layout of the modal).
- The “Iste Consultatis” logos should be part of a carousel. The two dashes below the logos should allow the user to advance between logo sets.
- The cards in the “Natus” section are static and do not link to anything.
- The gray box with the up arrow at the bottom right corner of the page should be sticky to that position in the document window. Clicking on the box should cause the page to scroll smoothly to the top of the page.

“Omnis” image hover state example

“Omnis” modal example



Email Sign Up Form

The email sign-up form in the footer should be a working form.

Instructions

- Use JavaScript to validate (1) that all fields have values before the form is submitted, and (2) that the email address is well-formed. If the values do not validate successfully, display an error message on the form, indicating the specific validation error(s).
- The form data should be sent to the PHP script via AJAX. There should not be a page reload or a separate confirmation page. Instead, upon successfully submitting the data, display a confirmation message above the form.

Back-end scripts:

- Create a PHP script that, when run, will create a MySQL database with a table that includes the following columns:
    - **SID** – This should be a unique integer (a "sign-up ID") that auto increments.
    - **name**
    - **email**
    - **created** – This should be the datetime when the record was created.
- Create a PHP script that will receive the form data, validate it, sanitize it and record the data to the database.

Admin page

Create a basic admin page that displays all submitted email sign ups in a table. This page should live at */signups*. The page layout is at your discretion. You do not need to apply any authentication measures for a user to view this page.