

Kira Chan

CSE435 Website instructions (starter):

In this class, you will be creating a home webpage for your project, which will be accessible from the CSE435 course webpage. You will store your project documents as well as the prototype (or a link to download) and update the website as you progress.

You are welcome to use any framework you wish (angular, react...) to make your website, but we must be able to access it through `cse.msu.edu/~netid`. You may be required to migrate it to a different host during the final submit, so make sure your platform supports that migration.

For the first few iteration of the site, a skeletal website is acceptable. But we do expect the website to look progressively more complete and professional by the first meeting with the customer. Please take time to learn how to make websites with external resources.

One suggestion is to use PHPstorm. PHPstorm is an intellij IDE, and it will help with some of the formatting and creating CSS. You can also deploy the page on your localhost to preview it without repeated pushes to confirm how it appears on a browser.

I have provided you with a website template, you may use this as a starting point. But we expect some creativity and different CSS styling as the project is developed. Your submissions **should not** merely be a copy of the template with a few changes.

The following are instruction to create a blank webpage that can be loaded when accessing `cse.msu.edu/~netid`:

- 1) SSH to a CSE server, in your home directory (~) type: `mkdir ~/web`
- 2) `chmod 755 ~/web`
- 3) `cd web/`
- 4) We will now create a document called `index.html`. This is where you will store the webcode to display at your homepage: `touch index.html`. [touch creates an empty file]
- 5) Add this line to it: `echo "Hello World!" >> index.html`
- 6) using a browser, goto `cse.msu.edu/~netid`, you should see a blank page with the text "Hello World".
- 7) You can update the website by changing the content of the `index.html` file. After changing something (particularly if you change CSS), you may need to do a force refresh (ctrl + F5) or (ctrl + reload button) to reload the cache and access the new browser.

The easiest site can be created with html and some css, see:

https://www.w3schools.com/html/html_intro.asp

https://www.w3schools.com/css/css_navbar.asp

For a quick idea on how to make a simple menu bar.

Simple HTML crash course:

HTML is the most basic markup language for creating webpages.

HTML always has this format:

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
<link rel="stylesheet" href="style.css">
</head>
<body>
<div id="name">
<p>hello world</p>
<p>this is my cse website</p>
</div>
//your code here.

</body>
</html>
```

Information surrounded by < > are called elements. For now, assume all pages have this template above.

- Title is what the tab of the page will show on the browser.
- <Head> contains meta information, scripts styles and other site related information.
- <body> is usually where the site code appears. Everything that you want to see should be created here.
- To create a header text, use <h1>, <h2>... (number will adjust size) surrounded by text. You can close an element by doing </h1>, or the respective element name.
- <p></p> forms a block of paragraph, you can put text in between.

- `<div></div>` is a generic *container* for html. You can use these to surround groups of objects that you can then edit style on with CSS.
 - You can name the div bars by doing `<div id="name"></div>` or `<div class="name"></div>`, and then reference this later when you format with CSS".
 - Id is for one object; class is for a bunch of objects with same style.
- Use `` to make surrounding text a hyperlink to something else. It could be neat if you link your linkedin with your team pictures.
 - Note that it should be `<a href..> <p>text</p> ` and not `<p><a href..> </p>`.

- `` tag is used to put an image on the page. The `` tag is a little trickier to use as it takes parameters.
 - To use img, you need to use ``.
 - Create a folder in the same directory as index.html and name it img.
 - You can add your team photos here.
 - Reference these photos by doing `"/img/personname.png"`.
 - `./` means current directory (from index.html)
 - `../` means one directory up.
 - Always use relative path for website creation, absolute paths will cause your website to break on migration or if the resource is relocated.
 - **DO NOT** use absolute path, an example is as `"~/netid/web/img/personname.png"`
 - Difference is with relative path, the file will search for the target in the path given relative to itself, whereas the absolute path will always search at the folder of the absolute path. So if the file no longer exist, it will break your web.
 - `` does not have a closing tag.
 - Be sure that the entire web directory structure has read and execute permissions for world.
 - After you have created all the subdirectories and added the needed files (e.g., photos of the team, prototype, etc.), then type the following commands (first line sets the read/execute permission for world; second line recursively changes the permission for all the subdirectories). As you add more directories and files, you may need to repeat the 2nd command.
 - `chmod 775 web`
 - `chmod -R 775 web/*`

A quick tip for editing websites, you can see your code and change it on the spot by using the *inspect element* function. This will allow you to change your code on the browser and see instant updates. Do

Kira Chan

note that you need to copy over the changes as the original files will not be updated if you edit it through your browser.

Here is a quick example of how I added one image with a title underneath:

```
<div class="profile">
  <a href="https://www.linkedin.com/in/my_profile_link">
    <img src = "./img/person.jpg">
      <p>Name</p>
      <p>Role</p>
  </a>
</div>
```

Cascading Style Sheets (CSS):

A site with plaintext can be enhanced by using CSS to add colours and other style elements to format our site.

To use CSS, add `<link rel="stylesheet" href="Site.css">` to your header, and create Site.css or a file with similar name (must have .css ending) in the same directory as index.html.

CSS can be quite powerful, since you can do a lot of styling to different elements of your site.

You use CSS in the following way. First, reference the element you are typing, say div.profile.

You can then edit the properties (e.g., width of page, alignment, and font) as follows:

```
div.profile {
  width: 200px;
  display: inline-block;
  text-align: center;
  font-family: "Times New Roman", serif;
}
```

Please see https://www.w3schools.com/css/css_intro.asp for more information on how to use CSS.

To make a folder secure access (i.e., requires CSE account to enter page):

1) Create a new subfolder under *web*, you can name this whatever you want. I named mine *protected* for now.

Kira Chan

2) create a page, say *touch groupinfo.html*

[touch creates an empty file]

3) *echo "le toucan has arrived" >> groupinfo.html*

3) *touch .htaccess*

4) the content of .htaccess should contain the following information:

```
AuthType Basic
AuthBasicProvider ldap
AuthName "Roaming Access"
AuthLDAPURL ldap://ds.cse.msu.edu:389/ou=People,dc=cse,dc=msu,dc=edu?uid?sub
AuthLDAPGroupAttribute memberUid
AuthLDAPGroupAttributeIsDN off
Require valid-user
```

5) *chmod 755 .htaccess*

6) Now goto cse.msu.edu/~netid/protected/groupinfo.html, you should see that it requires you to enter a password. You can further restrict the userbase to a username and password combo (hash), follow the last instructions on [this page](#). Make sure you give us access (by providing us a user and password so we can grade if you make it private to a list of users)

7) Tip: if you want to debug the htaccess, you need to clear cache to "Logout".

Helpful links:

<https://www.cse.msu.edu/Resources/Facilities/Howto/CreateWebpage.php>

<https://www.w3schools.com/html/>