

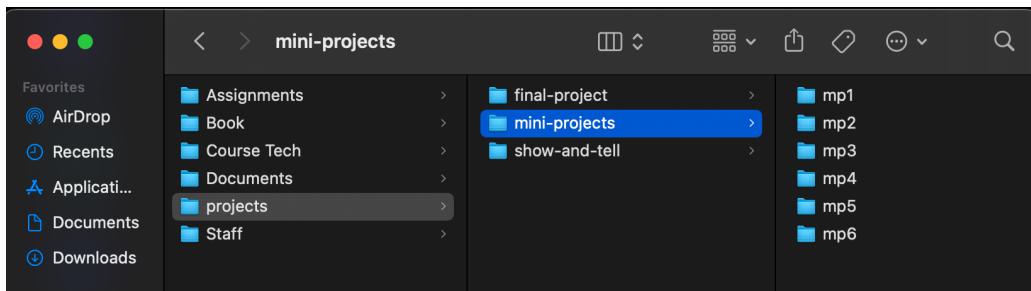
## MP1

In this Mini-Project, you will create a website based on your most-recent **resume**.

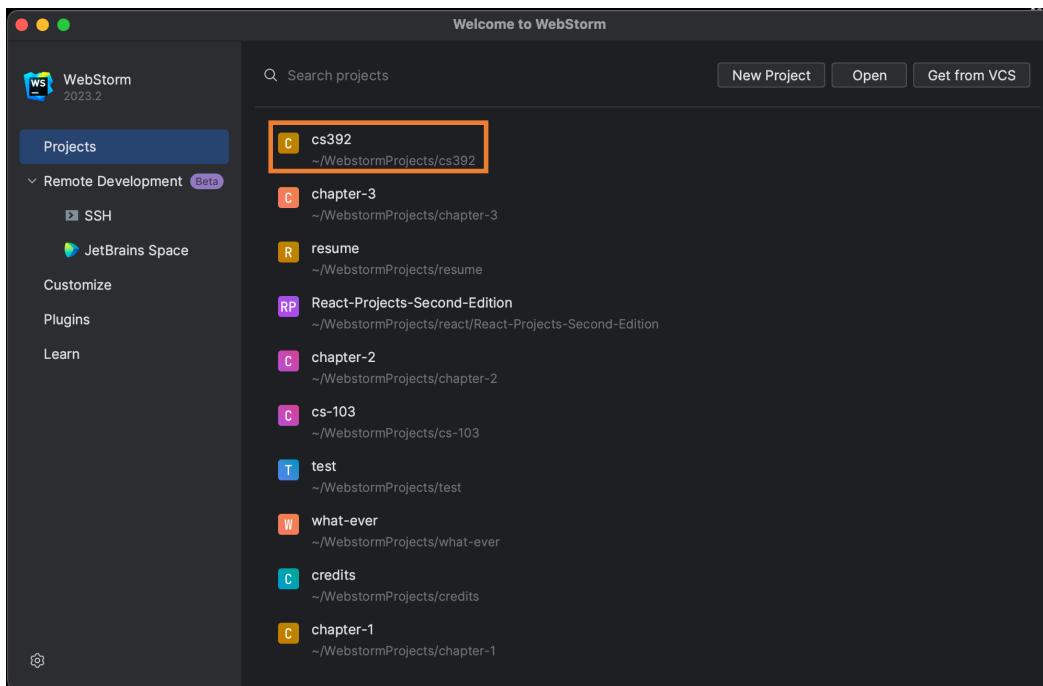
### Task-1:

In CS-391 none of your HTML files and none of your folders should contain spaces and capital letters, (**DO NOT capitalize and NO spaces**).

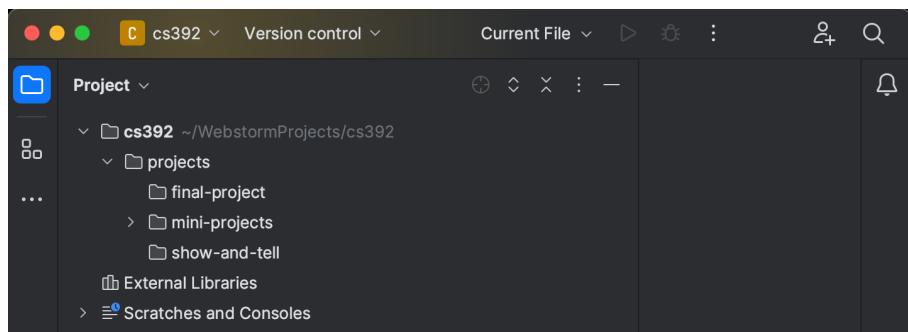
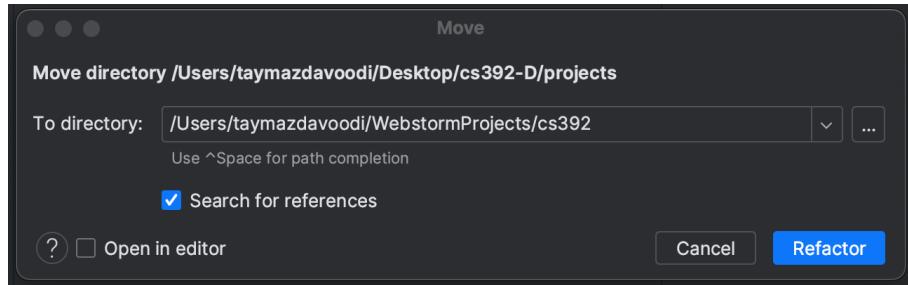
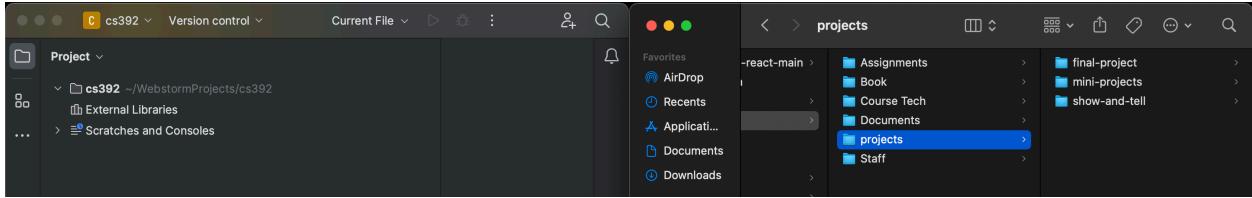
- Create a folder on your computer, name it **projects**, and inside create 3 sub-folders and name them **mini-projects**, **show-and-tell**, and **final-project**. In the **mini-project** folder, create 6 additional subfolders and name them **mp1** through **mp6**



- Open **WebStorm** and click on **cs-391**.



- Drag and drop your **projects** folder, from your computer, into your **cs-391** folder, inside the WebStorm, and click “**Refactor**”.



- In **WebStorm** right-click on **mp1**, select “**New**” and then “**HTML File**”.
- Name the new file something **unique** but similar to “**home-resume.html**”, and start with the home-page of your resume.

### **Requirements:**

- Your online resume **must include at least 6 internal webpages**, (based on sections of your resume). For example, an internal page about your education, another about your work experiences, another one about your certifications, etc.
- One of the internal webpages must be **Projects**. You must have this internal page, even if you do not have a Projects section in your resume. If you do have a Projects section in your resume, you may add your content to this page but please leave some space at the bottom because we will add a **JavaScript Calculator** at the bottom of this internal page, which should serve a JS project.

# HTML

All internal webpages and your website's **Home** page must have the 4 main tags (**header, nav, main, and footer**):

## The <header></header>:

- Should include a title (<h1>) for the webpage and subtitle (<p>) describing it.



## The <nav></nav>:

- Must include at least 6 links (<a>) to internal webpages, but may be more than 6, (e.g., Home, Education, Experiences, Certifications, References, Achievements, Documents, etc.).
- Each link must have an “href” attribute:
- Remember that those links would have to be nested inside <li> tags, and <li> tags must be nested inside a <ul> tag.

- [Home](#)
- [Education](#)
- [Employment](#)
- [Achievements](#)
- [References](#)
- [Documents](#)

## The <main></main>:

- Each internal page (including **Home**) must have a unique content based on one of the sections of your resume. The content must occupy at least half of the screen.

Home

My name is Taymaz Davoodi, and I am a part-time Computer Science lecturer at Boston University. I would like to further my academic and professional involvement in teaching, research, or administrative capacity, while I continue to advance towards a PhD or a full-time teaching position.

Welcome to my website, here you will find my [Educational](#) and my [Employment](#) history, as well as other information that might be of interest to you.

## The <footer></footer>:

- Should include a **copyrights statement** (<p>).
- The **copyrights statement** should include a link (<a>) named Credits and a **copyrights logo** (&#169;).

All Rights Reserved by Tamaz Davoodi [Credits](#)

Ex:



## Taymaz Davoodi

My Online Resume

- [Home](#)
- [Education](#)
- [Employment](#)
- [Achievements](#)
- [References](#)
- [Documents](#)

## Home



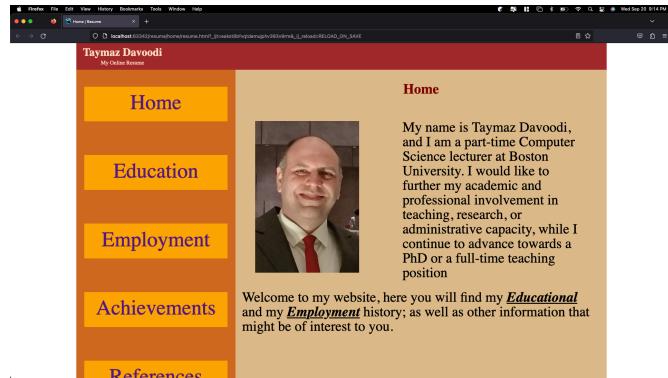
My name is Taymaz Davoodi, and I am a part-time Computer Science lecturer at Boston University. I would like to further my academic and professional involvement in teaching, research, or administrative capacity, while I continue to advance towards a PhD or a full-time teaching position

Welcome to my website, here you will find my Educational and my Employment history; as well as other information that might be of interest to you.

All Rights Reserved by Tamaz Davoodi [Credits](#)

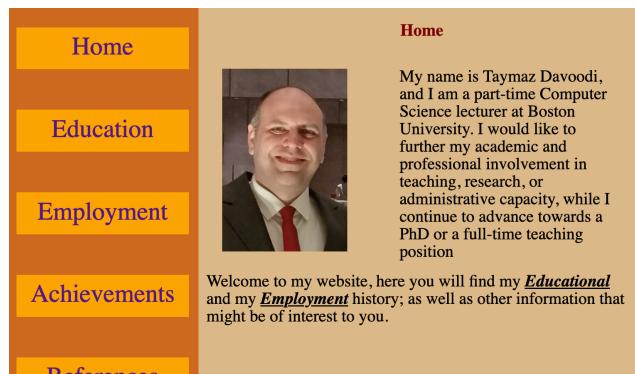
# CSS

All internal webpages, including the **Home** page, **must be responsive**, wrapped, squeezed with margins on both sides, and centered:

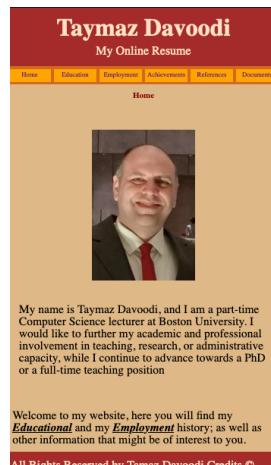


## The nav{}:

- For screens bigger than 750-px: should not exceed 30% of the width of its parent, and must be vertical.



- For screens smaller than 750-px: should occupy 100% of the width of its parent and appear horizontal.



## The header{}:

- For screens bigger than 750-px: the content should be aligned left.

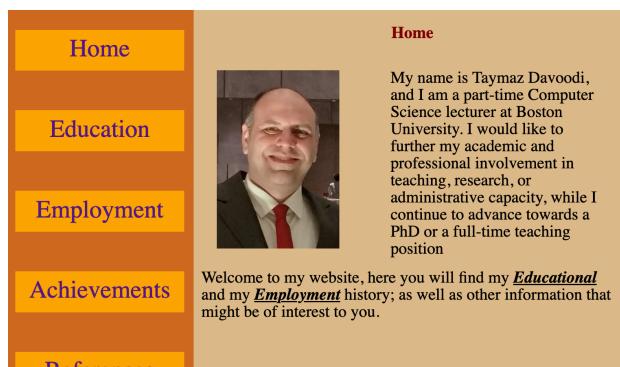


- For screens smaller than 750-px: the content should be aligned center.

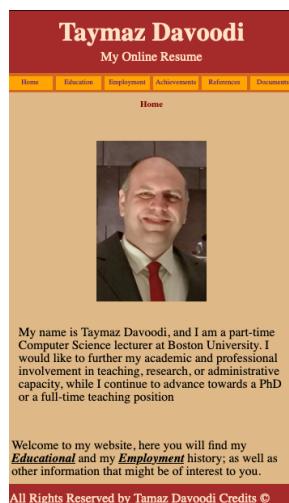


## The main{}:

- For screens bigger than 750-px: should not exceed 70% of the width of its parent.



- For screens smaller than 750-px: should occupy 100% of the width of its parent



- Example:

## Home

## References

## Education

Etc.

**Note:**

- Make good use of **margins** and **paddings**. Leave open spaces and do not clutter.
- Use **calc( + )** for font-sizes.
- Chose a background color for your main tags (**header**, **nav**, **main**, **footer**), but maintain a sense of uniformity.
- Use customized fonts, like google fonts.
- You may use Bootstrap or Flex-box with Media Queries.

## JS

In your internal **Projects** webpages, we are going to add a **JavaScript Calculator**. Remember that this page is an internal page and just like all the other internal pages must have the 4 main tags (header, nav, main, and footer). If you do have worked on other projects, and have shared them on GitHub you may post links to those repositories in your Projects webpage, but a some space at the bottom of the page must be reserved for our calculator.

**Our calculator should include:**

- 2 **input** boxes.
- 6 **buttons** marked (+, -, \*, /, \*\*, **Clear**).
- 1 **output** textual tag (e.g., <p></p>, <h1></h1>...<h6></h6>).

The user should be able to enters a number into the first and the second **input** box, and click on one of the **buttons**, to see the result of the operation displayed in the **output** box.

- The **input** tag specifies an input field where the user can enter data. It is a stand-alone element. This element contains only attributes and it does not have any content.

*Ex:*

```
<input type="text" id="first-number">
```

- The **button** tag defines a clickable event. Inside a button element you can put content, like text or images.

*Ex:*

```
<button onclick="addition()"> + </button>
```

**Note:**

For the `onclick=“ ”` attribute of your addition button, **DO NOT** write “**add()**”. **add()** is a pre-existing function in Js. and we don’t want to use that one.

- The **output** should be displayed in textual tag using the `.innerHTML()` function.

Ex:

.html  
<h3 id="output"></h3>

.js  
document.getElementById("output").innerHTML=String(result);

Ex:



First Number:  Second Number:  + - \* /

Addition:



First Number:  Second Number:  + - \* /

5

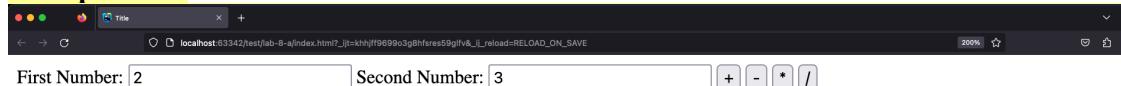
Subtraction



First Number:  Second Number:  + - \* /

-1

Multiplication



First Number:  Second Number:  + - \* /

6

Division



First Number:  Second Number:  + - \* /

0.6666666666666666

- In your .html file:

1. Add a **power** button; when it is pushed it should raise the value of the first input to the value of the second input.
2. Add a **clear** button; when it is pushed, it should erase the contents of the 2 input boxes and the output.

- In your .js file:

1. For the **power()** function you must use a **for()** loop. **If you implement your power() function using anything but a for() loop, (like Math.pow() or \*\*), you will get a 1/3dedcution on MP1's grade.**

*hint:*

<https://javascript.plainenglish.io/javascript-algorithm-power-2cbedf59f40c>

2. For the **clear()** function you can assign an empty “” to the value of the input boxes and the **.innerHTML** of the output tag.
3. For your results, regardless of which button is pushed, If the value of the result is **negative**, the result should be displayed with either **red** text or a **red background**.

A screenshot of a web browser window displaying a calculator application. The URL in the address bar is `localhost:83342/lec-9-supplements/cs-103-lab-8-a/index.html?_t=eq04cgfHNgSpI4BsoohQa6DnA_ll_reload=RELOAD_ON_SAVE`. The page contains two input fields labeled "First Number" and "Second Number", both containing the value "3". Below these are several buttons: "+", "-", "\*", "/", "\*\*", and "Clear". To the left of the buttons, the result "-1" is displayed in red text. The browser interface includes standard window controls (minimize, maximize, close) and a toolbar at the top.

## Style your calculator!!

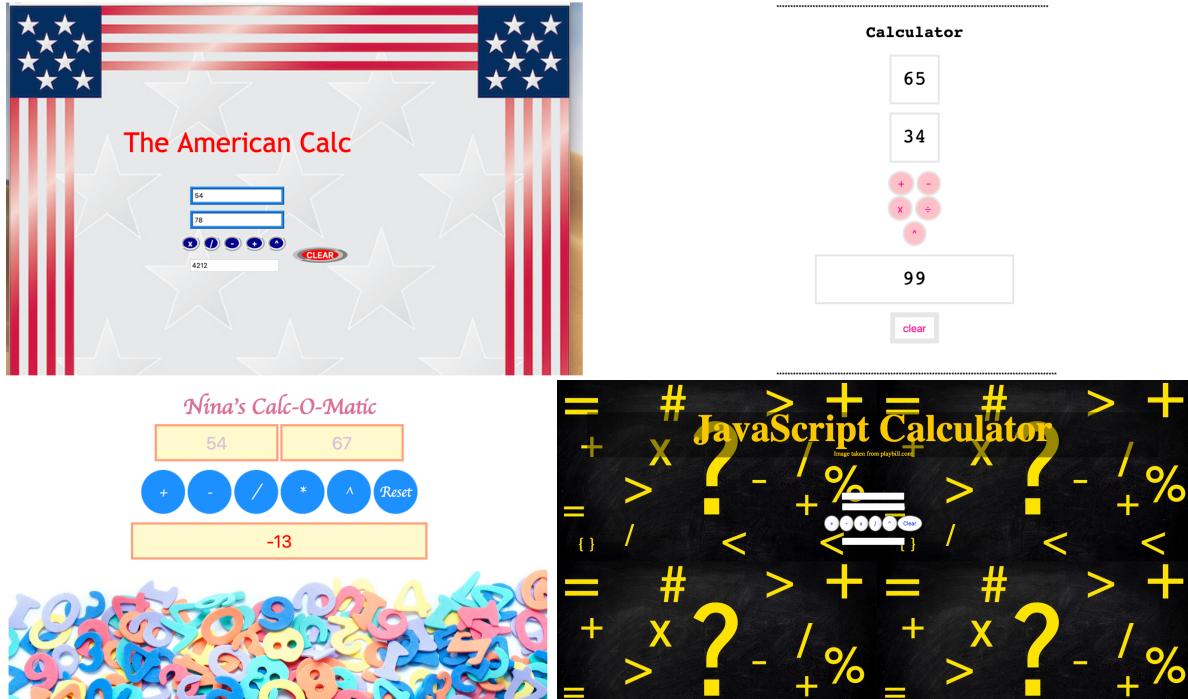
- At this point you should have a working multifunction calculator, but it might not be the prettiest thing on the web.
- In this part you'll apply some creative CSS to snap things up a bit.

This means that you should:

1. **Use div elements to group and align the inputs, buttons, result, and for any styling that you want.**
2. **Make good use of margins and paddings. Leave open spaces and do not clutter.**
3. **Use calc( + ) for font-sizes.**
4. **Use customized fonts, like google fonts.**
5. **Use Flex-box with Media Queries.**
6. **Make sure the content doesn't exceed the limitations of the screen or their parent tag, even when the screen is at smallest size (mobile device).**
7. **Make sure the text remains readable even if the screen is at the smallest size (mobile device).**
8. **Make sure margins and paddings shrink when the screen gets smaller.**

**9. Make sure you apply max-width for any image, so that it becomes responsive to the screen size.**

The following images are some examples of calculators (from past semesters):



**Task-2:**

**Assignment Deployment:**

- When you are done, open **BU's Server**, from Tools >> Deployment >> Browse Remote Host.
- Where it says **<None>**, to its right there is an arrow, click on it and select **BU's Server**.
- Drag your **resume (file and folders)** from the left-panel and drop it into the same folder cs391 >> projects >> mini-projects >> mp1, on the right-panel.
- Check your work before you go to the next task.
- Go to:

**cs-people.bu.edu/YOUR-BU-USERNAME/cs391/projects/mini-projects/mp1/  
YOUR\_RESUME.HTML**

if you ended up looking at the page you just created, then you have successfully finished **Deployed** your assignment on BU's Server, otherwise retrace your steps or seek help (from the instructor or the graders).

**Assignment Submission:**

- Navigate to Blackboard >> Mini-Project-1 >> Create Submission
- Paste this link in the text field:  
**cs-people.bu.edu/YOUR-BU-USERNAME/cs391/projects/mini-projects/mp1/  
YOUR\_RESUME.HTML**
- **Your Link might be slightly different from the one above, so make sure that it opens to your assignment before you submit it.**
- **Duplicate** (make a copy of it) your project and **Compress/Zip** the **Duplicated Project**.
- Navigate to Blackboard >> Mini-Project-1 >> Upload Files
- Upload a **zipped/compress** version of your **Duplicated Project**.
- Add comments if you like, and click **submit**.