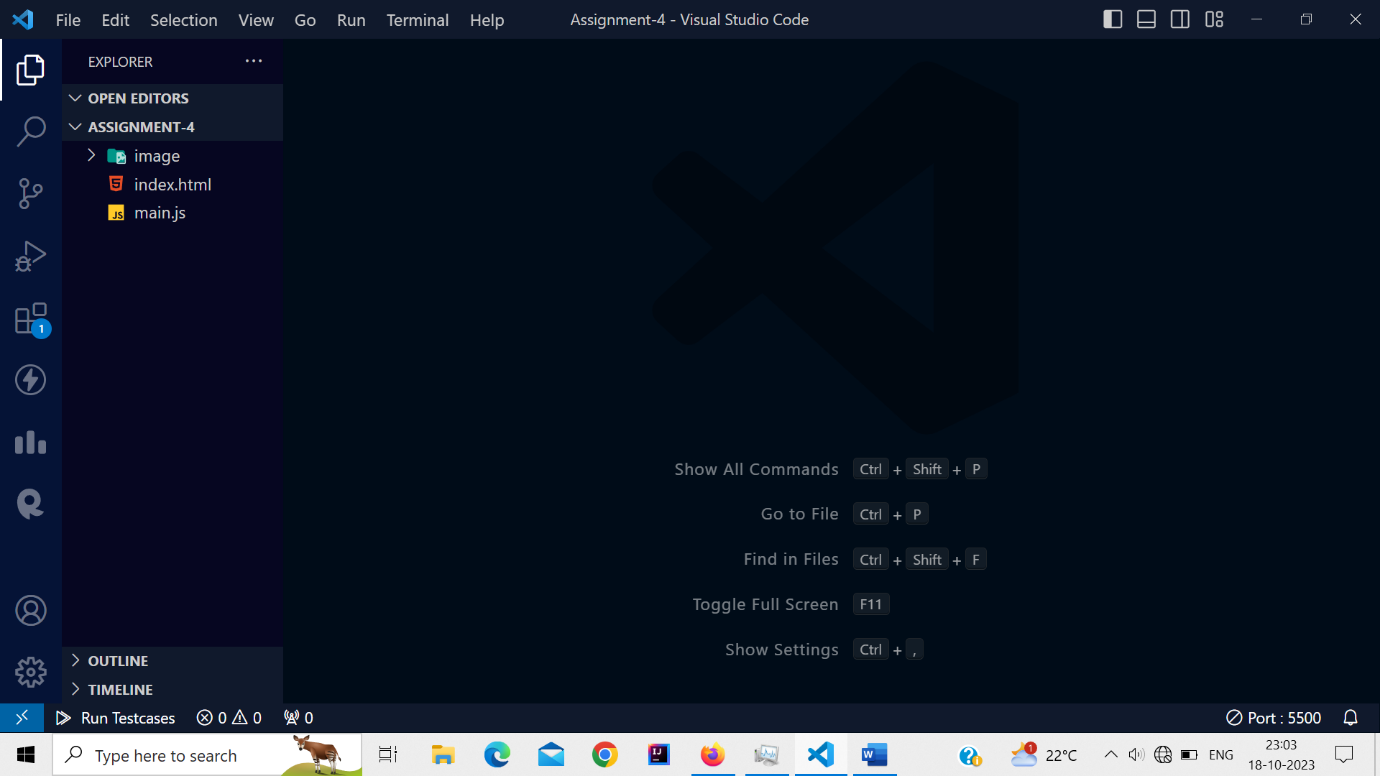
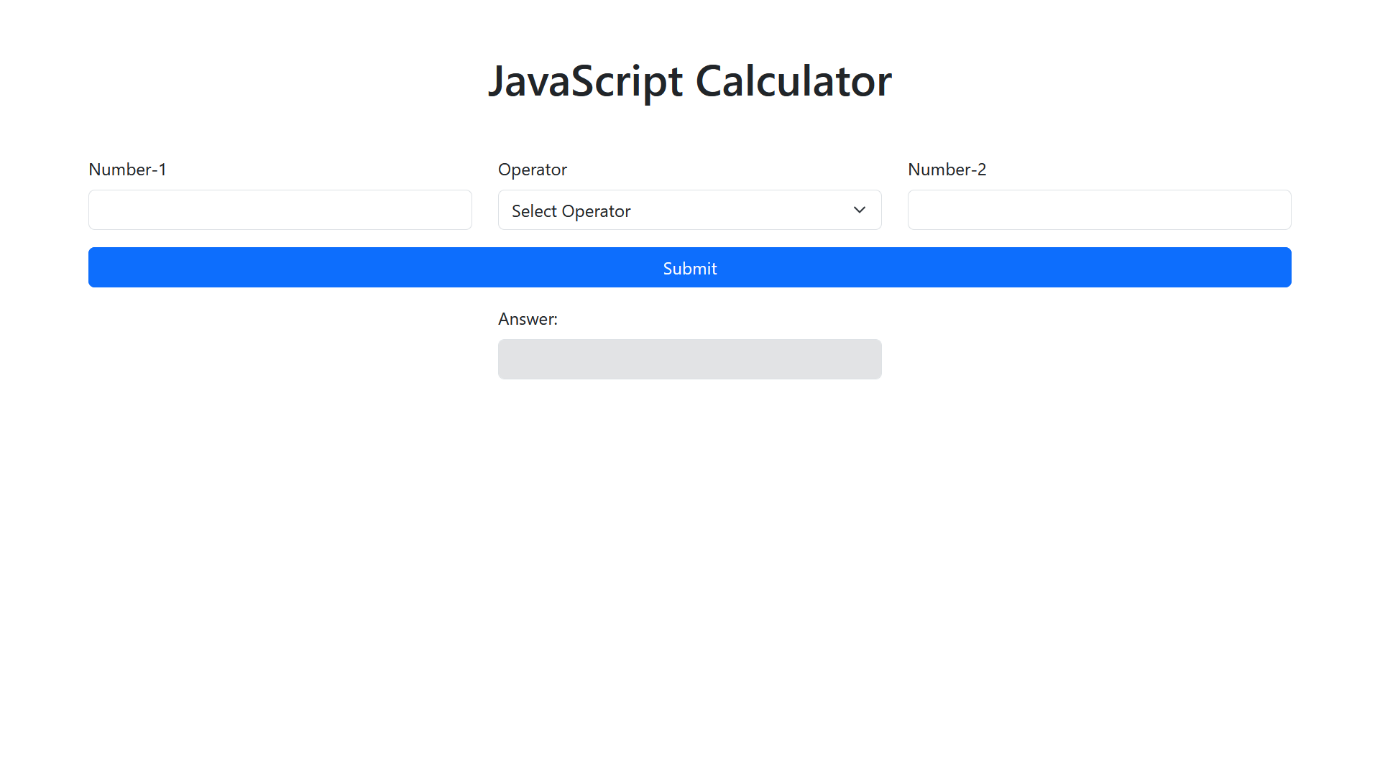
# Thought note on JavaScript Calculator (Module Assignment - 4)

## Folder and Files:

* 
* The Javascript calculator contains one image folder and two files.
* The image folder contains one .ico image for favicon.
* And the two files are index file and main javascript file.
* The Index file contains the html code along with bootstrap styling.
* And the Javascript file contains the logic behind the Javascript Calculator.

## JavaScript Calculator User Interface (UI)

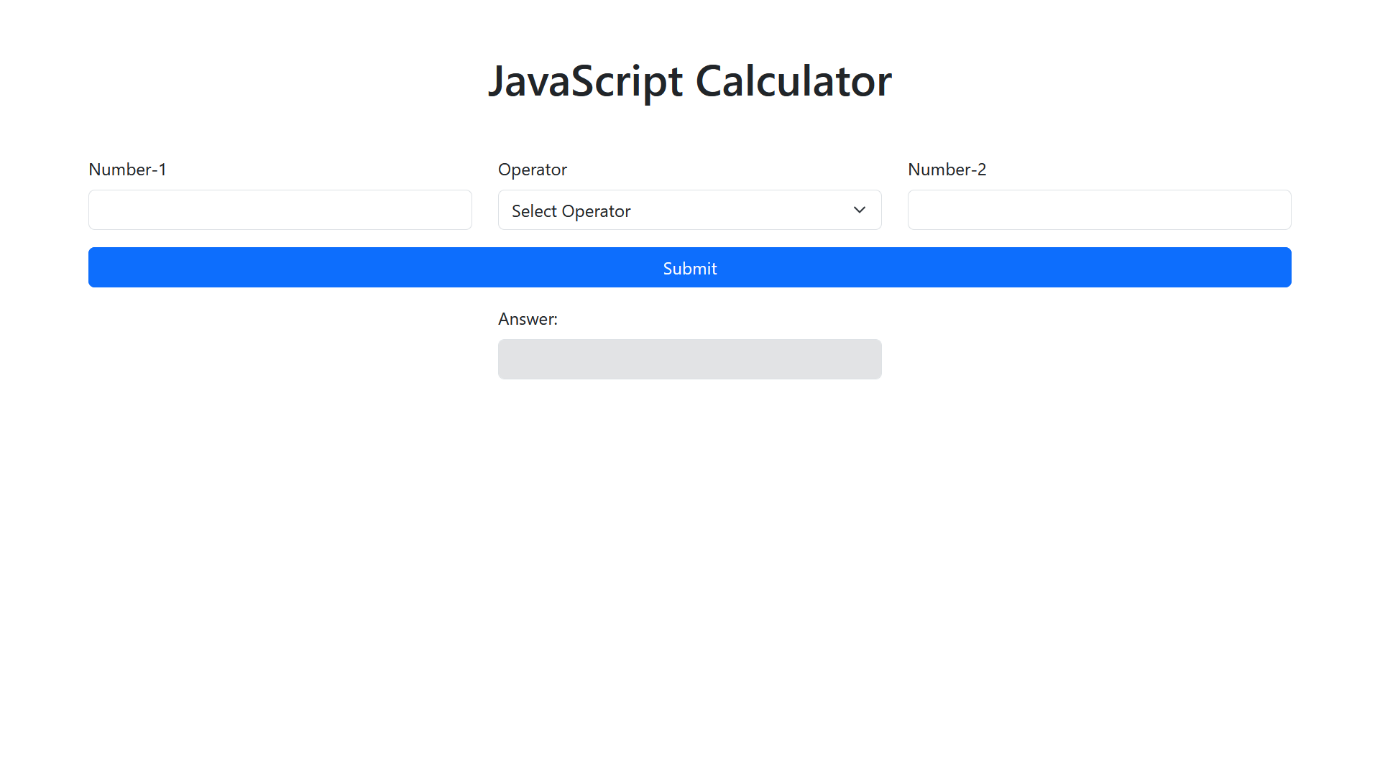
* 
* The Javascript calculator user interface (UI) contains calculator’s heading, two inputs for number, dropdown for selecting operator, submit button and one readonly input for answer.
* The user interface (UI) is builds with the help of bootstrap 5.
* The javascript calculator contains form to submit the input and operator values for mathematical operation.
* Inside the form, three rows are used for better placement of the inputs, dropdown, button and readonly answer input.
* The first row contains three columns, first column is used for number-1 input with type number, second column is used for dropdown operator and third column is used for number-2 input with type number.
* The number arrows in all the input fields of number type are removed using CSS.
* The second row contains only one column, the column is used for the submit button.
* The third row contains only one column, the column is used for the readonly answer input.
* The third column is centered with the help of justify-content-center.
* All this row and column property are achieved with the help of bootstrap.

## JavaScript Calculator’s Working:

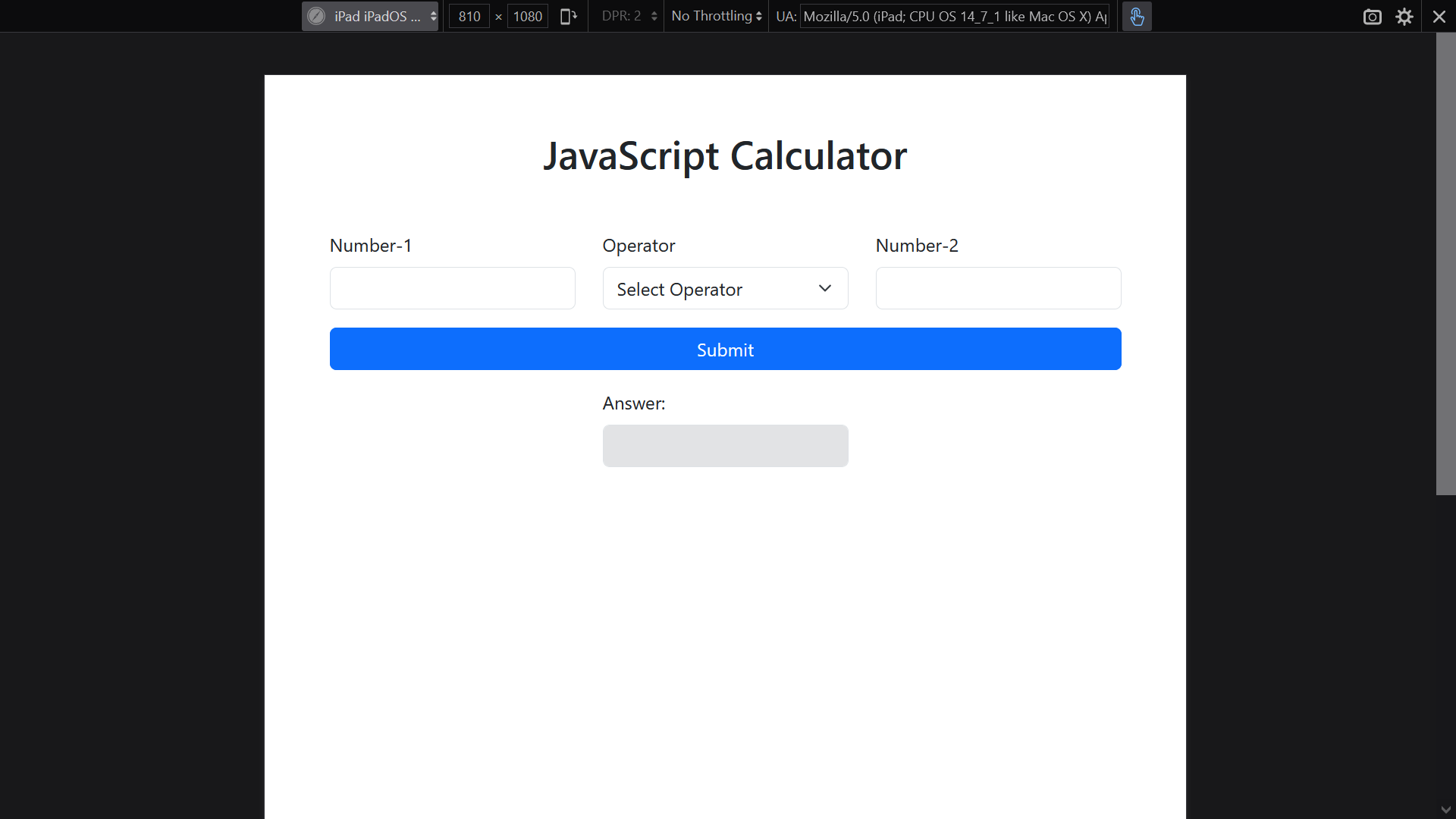
* Whenever the form or value get submitted, the form is prevented from submitting the value into the network tab by using e.preventDefault() (e is known as event), as we are not sending this value to the server. And the submission is happening with the help of addEventListener.
* All the inputs and operator’s value are extracted with the help of document.querySelector(“#id”).value .
* Parse Float has been used to receive the float value and to give the result in float.
* Multiple condition has been applied to check the inputs and operator are valid or not.
* If number-1, number-2 and operator are empty or invalid while submitting, then the alert message will get appeared to the user i.e. “Please enter valid numbers and select operator”.
* If number-1 and number-2 are empty or invalid and operator are selected while submitting the form, then the alert message will get appeared to the user i.e. “Please enter valid numbers in both the field”.
* If number-1 are empty or invalid and number-2 value are provided and operator are also selected while submitting the form, then alert message will get appeared to the user i.e. “Please enter valid number in number-1”.
* If number-2 are empty or invalid and number-1 value are provided and operator are also selected while submitting the form, then alert message will get appeared to the user with the message “Please enter valid number in number-2”.
* If both the input values are provided and operator is not selected while submitting the form, then alert message will get appeared to the user with the message “Please select an operator”.
* If number-1 input value are provided and number-2 value are not and the operator is also not selected while submitting the form, then the alert message will appeared to the user with the message “Please enter valid number in number-2 and select operator”.
* If number-2 input value are provided and number-1 are not and operator is also not selected while submitting the form, then the alert message will display to the user with the message “Please enter valid number in number-1 and select operator”.
* All the possible cases which may occur while submitting the form have been applied in this Javascript calculator.

## Responsiveness:

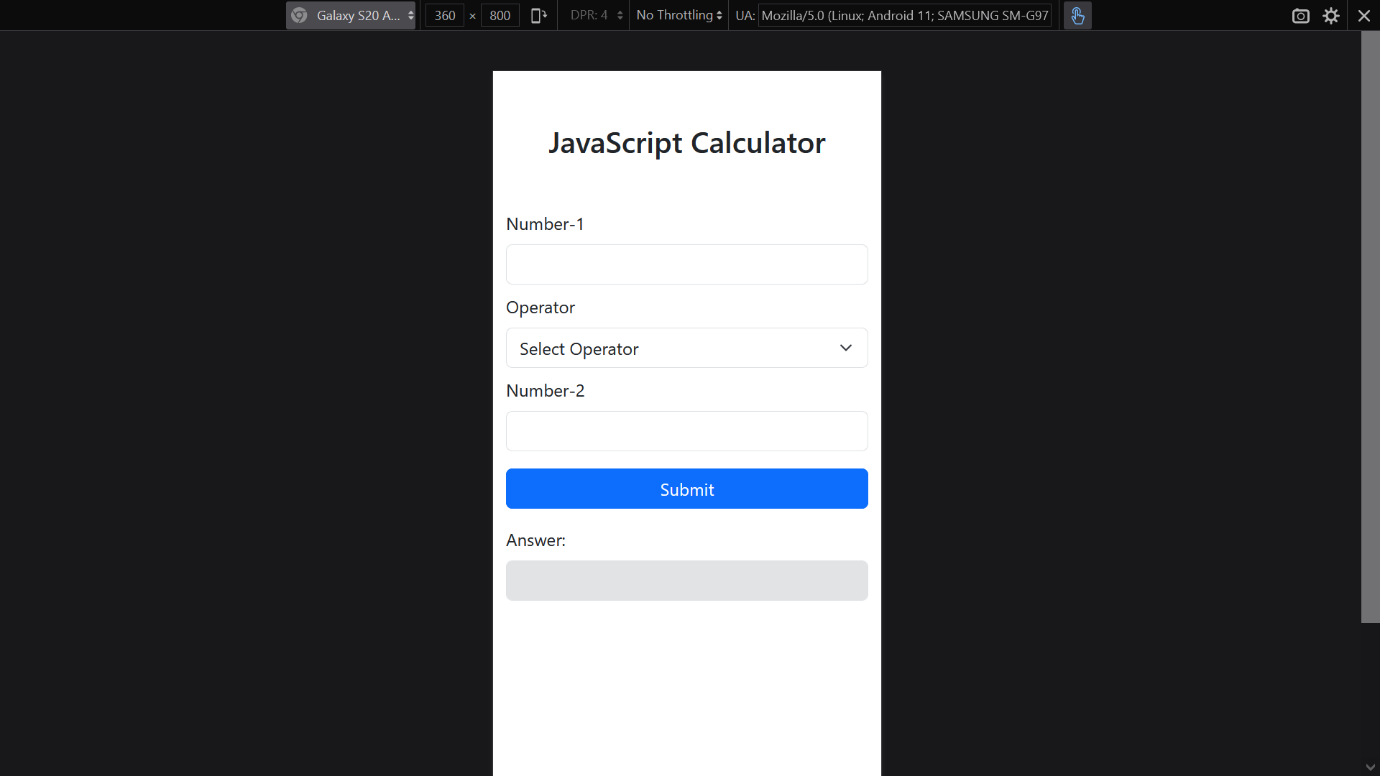
Large Screen:

* 

Medium screen (tablet screen):

* 

Small screen (mobile screen):

* 

Note: Internal CSS has been used for removing number arrows in inputs fields of number type, as I don’t wanted users to make changes (increment or decrement) in input fields through those arrows. And added media query for small screens to add margin bottom in columns.