

CREATE A FORM THAT CALCULATES THE SUM OF 2 NUMBERS (FRONT END)

In this exercise, you will need to create two HTML forms where the users can enter two numbers and submit the form to a backend server. If the request is successful, the server will respond with an HTML page containing the value of the two numbers along with their sum (total).

PREREQUISITES

- You must have [Node.JS](#) installed
- You must have [npm](#) version **5.2.0** or greater (npm comes bundled with [Node.JS](#), you don't need to install it separately)
 - Check the [npm](#) version by running:

```
npm --version
```

INSTRUCTIONS

1. Download and unzip the exercise material from the following URL:
<https://courses.socialhackersacademy.org/wp-content/uploads/calculator-form-exercise.zip>
2. Open [VSCode](#), then open a new file (**File > New File**) and select **File > Add Folder to Workspace** choosing the folder containing the exercise material.
3. Once you have your Workspace set up, open the terminal, ensure that it is pointing to the current folder (the one containing the exercise contents and specifically the [calculator-server.js](#) file) and execute the following command:

```
npx nodemon calculator-server.js
```

You can always stop the server by pressing **Ctrl + C** on the terminal running the above command.

4. After running the command, you should be seeing the message **SERVER LISTENING AT PORT 8080** on your terminal
5. Open the file [calculator-form.html](#) in [VSCode](#) and add the required code in order for the following requirements to be met:

5.1) The HTML page must include a link to the following CSS file (served from a CDN) so that our page is styled according to a nice dark theme:

```
https://cdn.jsdelivr.net/gh/kognise/water.css@latest/dist/dark.min.css
```

5.2) Make sure that the two forms point to (submit their content to) the local server's address:

`http://127.0.0.1:8080`

5.3) Make sure that the first form is using the **HTTP GET** method to submit its content to the server and the 2nd one is using the **HTTP POST** method.

5.4) Each form must contain 3 input fields. Two numeric fields for entering the first and second number for calculating the sum (these input fields must not accept negative numbers), and a third **checkbox** input field which the users must check in order to accept the software terms (see screenshot below). All 3 fields are **required** before submitting the form.

IMPORTANT: The server requires that the two accepted input values (remember the **?key=value&key=value** pattern?) use the following key names:

number_a and **number_b**

FORM using HTTP GET

Calculator: Add 2 Numbers

Enter the first number:

Enter the second number:



Please fill in this field. I agree with the terms of the software agreement.

Calculate

FORM using HTTP POST

Calculator: Add 2 Numbers

Enter the first number:

Enter the second number:

☐

I agree with the terms of the software agreement.

Calculate

5.5) All input fields must be accompanied by their respective **labels**.

5.6) Visit the following tweet and style the checkbox input fields accordingly (see screenshot). You can pick an appropriate emoji character from this list: <https://www.emoji.com/emoji-cheat-sheet/smileys-people>

<https://twitter.com/sulco/status/1311283794817802247>

5.7) When both forms are submitted, a web page showing the submitted values and their sum should be displayed on the web page (see screenshot).

Congratulations! Calculation: $56 + 34 = 90$

Input **number_a**: **56**

Input **number_b**: **34**

5.8) Run your HTML code through the following services and ensure that no errors or warnings are present:

[Web Accessibility Checker](#)

[W3C Validator](#)

Make sure that your web page looks as close as possible to the screenshots!