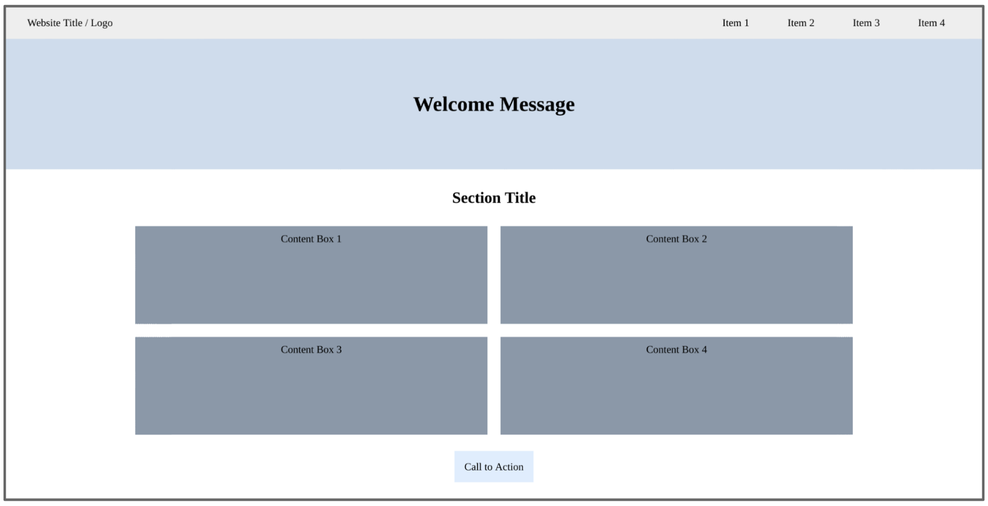
Campus Program #2

Remote Learning Assignment - Week 2

Assignment 1: HTML DOM and Event Handling

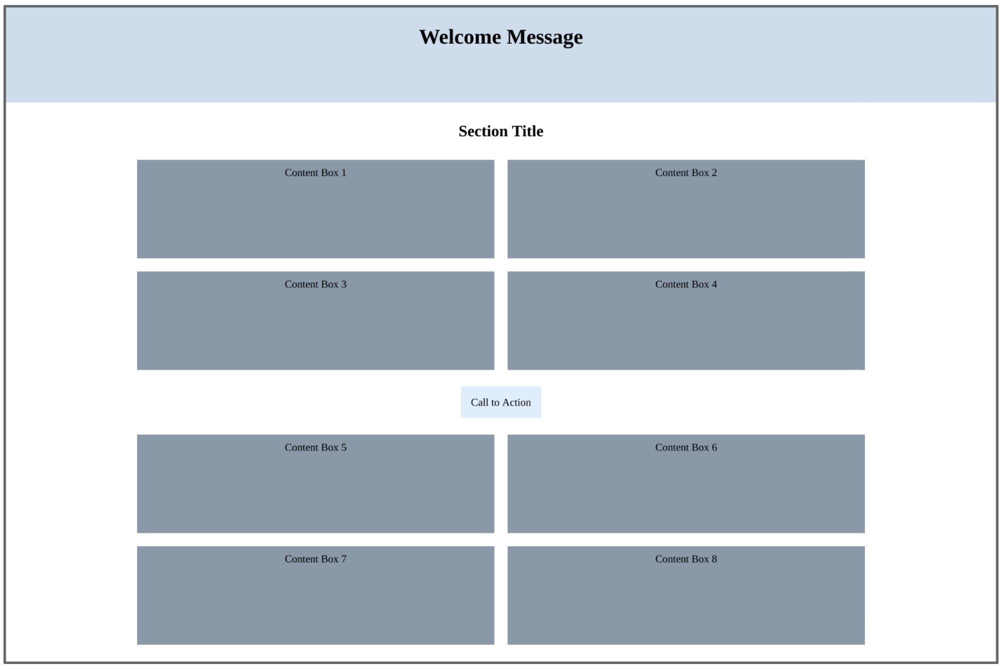
Following the assignment in assignment 1, let's add some effects on it by only pure JavaScript without Bootstrap, JQuery, or any other libraries.

Request 1: Click to Change Text.
When the user clicks on the "Welcome Message" block, change the text to "Have a Good Time!".



Request 2: Click to Show More Content Boxes.
There are some more content boxes waiting to show. When the user clicks the Call-to-Action button, show those hidden content boxes.

Hint: all content boxes are already there, they are just set to <u>display: none</u> at the beginning.



Campus Program #2

Remote Learning Assignment - Week 2

Assignment 2: Callback Function

Complete the function below to show a delayed result in the console.

```
function delayedResult(n1, n2, delayTime, callback) {
 // your code here
}
delayedResult(4, 5, 3000, function (result) {
 console.log(result);
}); // 9 (4+5) will be shown in the console after 3 seconds

delayedResult(-5, 10, 2000, function (result) {
 console.log(result);
}); // 5 (-5+10) will be shown in the console after 2 seconds
```

Hint: You should use setTimeout() for time scheduling.

Assignment 3: Your First Web Server

To build your first web server for development, follow the steps below:

1. Install Node.js
2. Create a Node.js project by NPM
3. Install Express module in your Node.js project by NPM
4. Write a simple web server program and start it
5. Show an HTML page when you enter http://localhost:3000/ in a browser's address bar (For example a simple page including "Hello, My Server!" is an acceptable result.)

You may refer to getting started document in Express official website to complete this assignment.

Reminders:

1. You have to learn how to use a command-line interface on your computer.
2. Set up your GitHub repository to **ignore folder node_modules**, which includes all the modules installed in your Node.js project. Refer to Ignoring Files document.
3. All the assignments this week should continue with the same Node.js project built in this assignment.
4. You don't need to split folders for each assignment this week, your folder structure could be like remote-assignments/Week2/Assignments

Campus Program #2

Remote Learning Assignment - Week 2

Assignment 4: Build Backend API for Front-End

Now, try to modify your code executed on the server-side to build a simple API. Your server should fulfill the following client requests:

1. When the user enters http://localhost:3000/data in a browser's address bar, show a "Lack of Parameter" message on the page.
2. When the user enters http://localhost:3000/data?number=xyz in a browser's address bar, shows a "Wrong Parameter" message on the page. (xyz means any non-integer value)
3. When the user enters http://localhost:3000/data?number=5, they should get the result of 1+2+....+5 on the page.
4. Generally speaking, when the user enters http://localhost:3000/data?number=N, they can get the result of 1+2+....+N on the page. (N is any positive integer)

Note:

1. handle HTTP GET method and parameters with Node.js and Express on the server-side.
2. (Optional) Think about what will happen when N is very large?

Campus Program #2

Remote Learning Assignment - Week 2

Assignment 5: Connect to Backend API by AJAX

You have built your first API in the backend, then let's get back to the front-end. Follow the steps below to send an HTTP request to your backend API by AJAX.

1. Update your Express project to serve static files. You can refer to this document.
2. Serve a static HTML file named sum.html. It means you can enter http://localhost:3000/sum.html in a browser's address bar to get this HTML page.
3. Write some JavaScript code in sum.html to make an HTTP request by AJAX to http://localhost:3000/data?number=10, and get the result 55 from the server.
4. Write a simple user interface to let users enter a number and get results from the server. (For a simple example, a text input and a button.)

Hint: refer to W3Schools or MDN for learning more about AJAX.

Campus Program #2

Remote Learning Assignment - Week 2

Assignment 6: Promise & Async / Await (Advanced Optional)

If you still have time, you can try to learn a very important concept called Asynchronous. Read the article and finish the following functions.

Basically, There are three ways to implement the asynchronous function, which are:

1. callback
2. promise
3. async / await

The delayedResult function in assignment 2 is implemented by **callback**, try to implement it again using **promise** this time. It should look like:

```
function delayedResultPromise(n1, n2, delayTime) {
 // your code here
}
delayedResultPromise(4, 5, 3000).then(console.log);
// 9 (4+5) will be shown in the console after 3 seconds
```

You can also try the third way and implement it using async/await this time.

```
async function main() {
 // your code here, you should call delayedResultPromise here and
get the result using async/await.
}
main(); // result will be shown in the console after <delayTime>
seconds
```