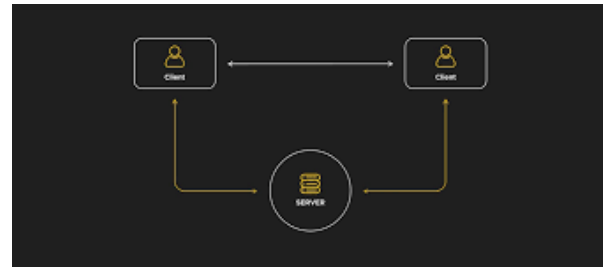## Cloning a Web Page

What is our GOAL for this CLASS?

In this class, we will be cloning a web page purely with JavaScript and jQuery without writing any HTML and CSS

What did we ACHIEVE in the class TODAY?

- Understanding cyber security in Frontend Code
- Cloning a web page purely with HTML
- Manipulating HTML with jQuery

Which CONCEPTS/ CODING BLOCKS did we cover today?

- JavaScript
- jQuery
- Cloning
- Cyber Security

How did we DO the activities?

Activity:

1. We understand how big companies like Google protect their pages from getting cloned and make it hard for hackers by viewing the page source of google.com
2. They invest hugely in security and the developers there ensure that the code is not readable and the entire HTML and CSS is created purely with JavaScript.

3. We login into the ecommerce store here and view the page source of the profile page that we want to clone.

```
Line wrap
1  <html>
2
3  <head>
4      <meta charset="utf-8">
5      <meta http-equiv="x-ua-compatible" content="ie=edge">
6
7      <!-- Bootstrap -->
8      <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
9      <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
10     <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>
11
12     <!-- Font Awesome -->
13     <link href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" rel="stylesheet">
14
15     <!-- Fonts -->
16     <link rel="preconnect" href="https://fonts.googleapis.com">
17     <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
18     <link href="https://fonts.googleapis.com/css2?family=Fira+Sans:wght@300;900&display=swap" rel="stylesheet">
19
20     <!-- Sweet Alert -->
21     <script src="https://cdnjs.cloudflare.com/ajax/libs/limonte-sweetalert2/8.11.8/sweetalert2.min.js"></script>
22     <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/limonte-sweetalert2/8.11.8/sweetalert2.min.css">
23
24     <meta name="viewport" content="width=device-width, initial-scale=1">
25
26     <script>
27     function logout_user() {
28         $.ajax({
29             url: "/api/logout",
30             type: 'post',
31             success: function (result) {
32                 window.location.href = "/";
33             },
34             error: function (result) {
35                 alert(result.responseJSON.message);
36             }
37         });
38     }
39 </script>
40     <style>
41     body {
42         direction: ltr;
43         text-align:left;
44         background-color: #f0f1f3;
45         font-family: 'Fira Sans', sans-serif;
46         overflow-x: hidden;
47     }
48     .blur {
49         filter: blur(5px);
50         -webkit-filter: blur(5px);
51         -moz-filter: blur(5px);
52         -o-filter: blur(5px);
53         -ms-filter: blur(5px);
```

4. We discuss how this page can be cloned by either manually copy-pasting the code or writing pseudo code that fetches fresh HTML every time the page is opened. The advantages for that is -
   a. It saves a lot of time
   b. You mostly do not have to understand the code yourself
   c. It will automatically inherit any changes that the real website undergoes in terms of page design

5. We open VS Code editor and create a new file called *index.js* and create the basic HTML structure in it -

```
<html>
    <head>

    </head>
    <body>

    </body>
</html>
```

6. We import jQuery in the head tag

<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

```html
<html>
    <head>
        <!-- jQuery -->
        <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    </head>
    <body>

    </body>
</html>
```

7. We create a script tag inside the body tag and then with the help of AJAX, fetch the HTML of the profile page and add it into the body tag with the **append()** function

```html
<body>
    <script>
        let html;
        $(document).ready(function () {
            $.ajax({
                url: 'http://ec2-3-13-85-11.us-east-2.compute.amazonaws.com/profile?id=1',
                success: function (data) {
                    html = data
                    display_html()
                }
            });
        })
        function display_html() {
            $("body").append(html)
        }
    </script>
</body>
```

8. We open the **index.html** page in browser and see the following output, i.e. a page that looks similar to the profile page -
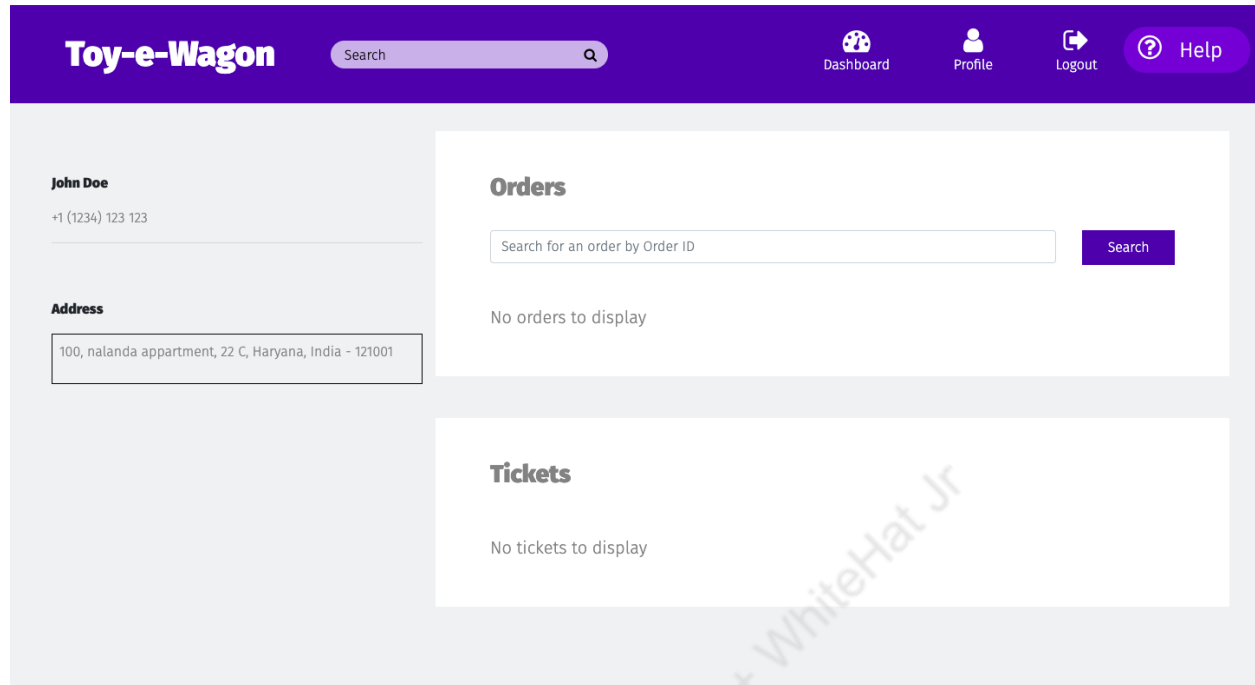
9. We identify that any user might notice the name, number and address details in the left section that does not belong to them and might get suspicious about this page.

10. We inspect the page to identify that there are 2 columns with classes **col-lg-4 and col-lg-8** respectively for the left and the right section.
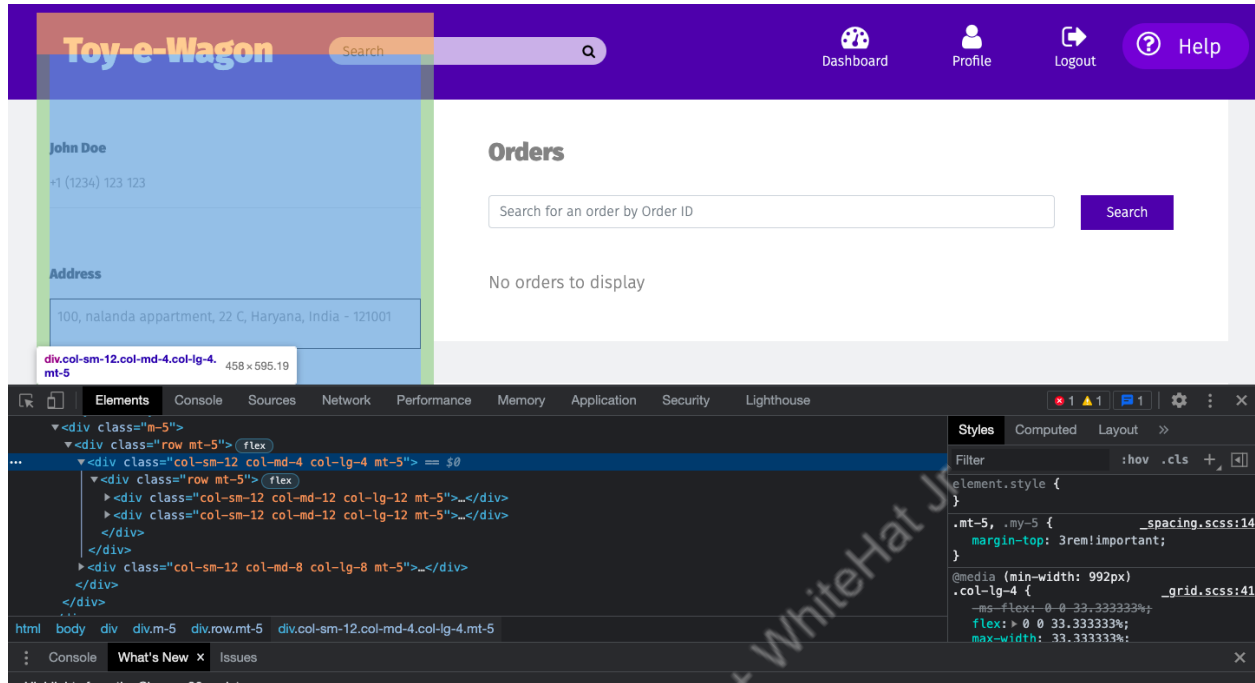
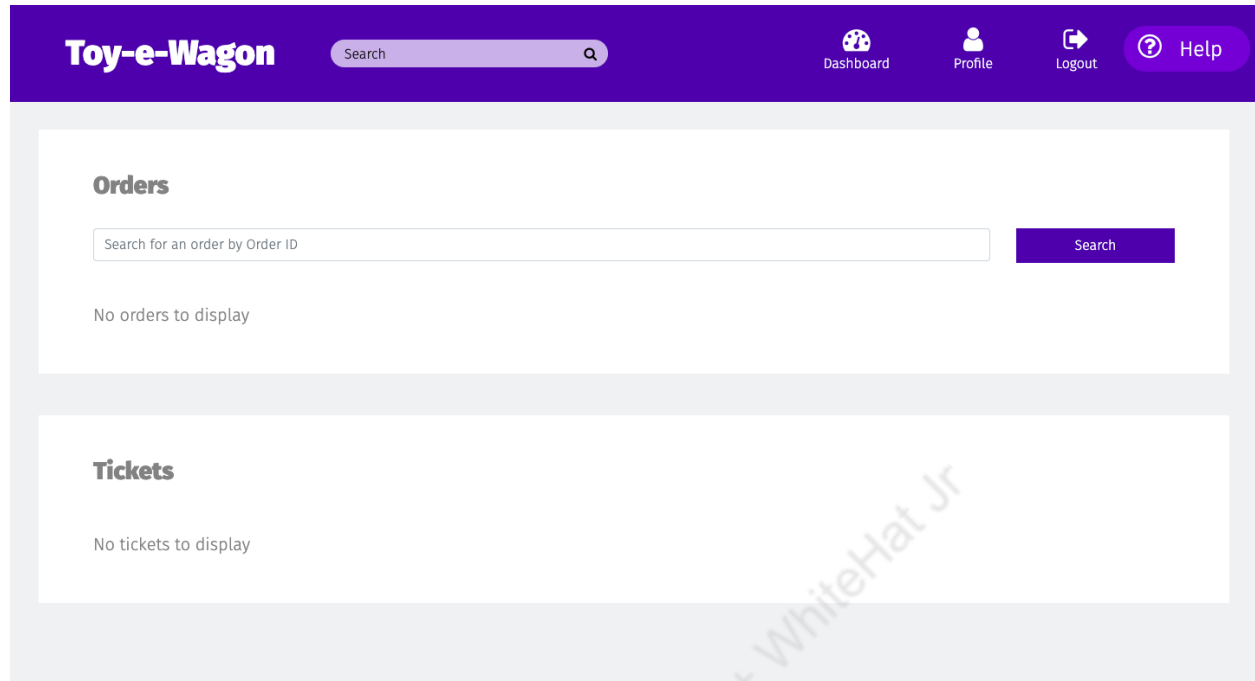11. We remove the **div** tag with the **col-lg-4** class, and replace the **col-lg-8** class with **col-lg-12** class so that our right section occupies the entire width of the screen.

```
function display_html() {
    $("body").append(html)
    $(".col-lg-4").remove()
    $(".col-lg-8").removeClass("col-lg-8").addClass("col-lg-12")
}
```

12. We observe that the victim might think that the website changed it's design, and achieve the final output as we check our **index.html** in the browser.

Orders

Search for an order by Order ID

Search

No orders to display

Tickets

No tickets to display

What's NEXT?

In the next class, we will add some more tweaks to this output and demonstrate a phishing attack by exploiting another vulnerability of this website.

Expand Your Knowledge:

Learn more about Phishing attacks and how to prevent them here.