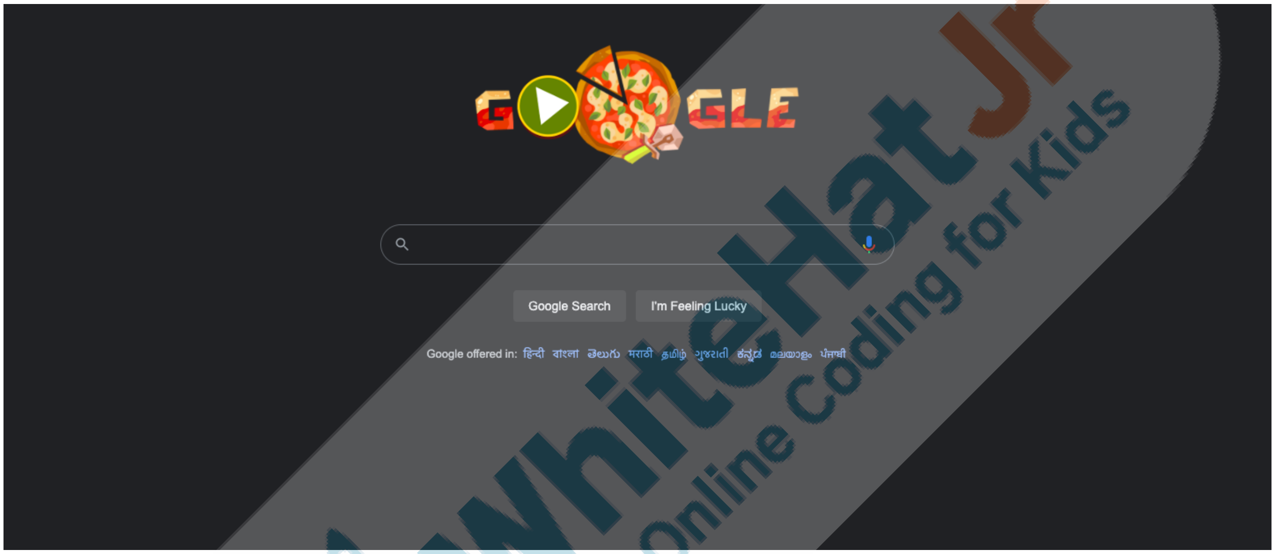
Topic	**Cloning a Web Page**	
Class Description	**Students will learn about how they can fetch any page's frontend code and clone a page to perform a phishing attack.**	
Class	**C-236**	
Class time	**45 mins**	
Goal	● Learn about Cyber Security in Frontend's code ● Learn about cloning a web page	
Resources Required	● Teacher Resources: ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen ○ Visual Studio Code ● Student Resources: ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen ○ Visual Studio Code	
Class structure	**Warm-Up** **Teacher-led Activity 1** **Student-led Activity 1** **Wrap-Up**	**10 mins** **10 mins** **20 mins** **5 mins**

WARM-UP SESSION - 10mins

Teacher Action	**Student Action**
Hey <student's name>. How are you? It's great to see you! Are you excited to learn something new today?	**ESR**: Hi, thanks, yes, I am excited about it!

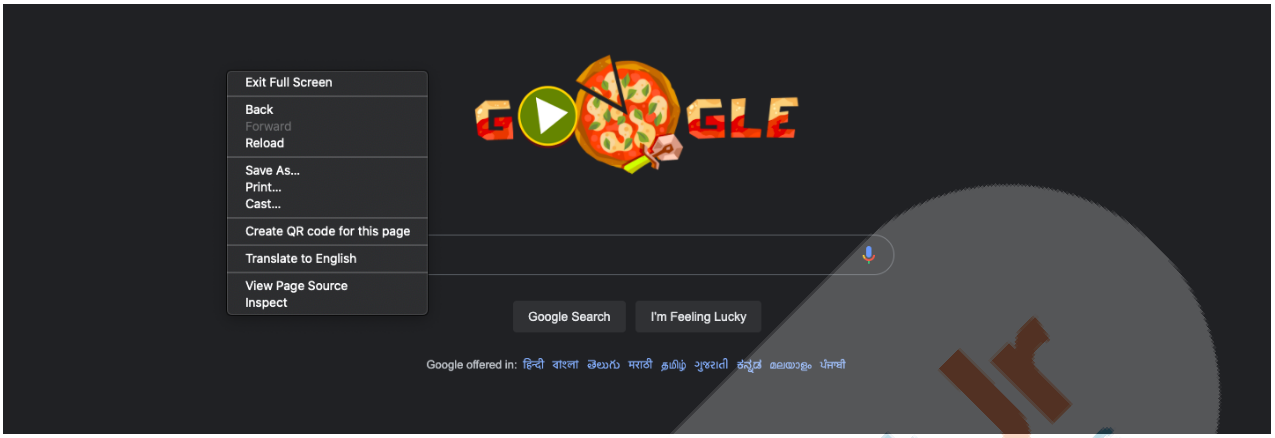
In the last session, we learned about the IDOR attack and saw how we can access unauthorised data by simply tweaking the URL. It is one of the most famous attacks where the attacker just needs to try out different combinations and sequences of the unique ID. Any doubts from the last session? *The teacher clarifies doubts (if any)* *We have performed a phishing attack before, where we created a fake login page and exploited the video chat app's mailer API to send emails on behalf of that application. Today, we will understand about cyber security in Frontend code, and also, how a page's frontend code can simply be cloned.* *Let's get started*	**ESR**: Varied!
Q&A Session	
Question	**Answer**
Which of the following is False? 1. IDOR attack can be used to access unauthorised data 2. IDOR attack can be used to access authorised data 3. IDOR attack is performed by manipulating the API 4. IDOR attack is performed by manipulating the URL	**C**
Which of the following resembles a brute force attack the most? 1. To try all the options one by one until you get the right one 2. To manipulate the URL to fetch data	**A**

3. To try out random options until you get the right one 4. To deceive the victim with a cloned page	
TEACHER-LED ACTIVITY - 10 mins	
Teacher Initiates Screen Share	
ACTIVITY ● **Understanding cyber security in Front-End** ● **To clone the frontend of an HTML page**	
Teacher Action	**Student Action**
Since we want to perform a phishing attack, we know that the first thing that we need is a page with which we can deceive a victim! Do you remember how we performed the phishing attack last time? That's right. Now, if we were to perform a phishing attack, we have to make sure that the page through which we want to deceive a victim should be very relatable as well as realistic so that the user does not question the authenticity of the page and trusts it. For our case, we could clone the profile page of the ecommerce application that we have been working with lately. We know that the profile page of that website is not as secure as it should be, since we have performed both SQL injection as well as an IDOR attack on that page already, therefore it becomes a good option.	**ESR:** We created a fake login page and exploited the mailer API of the video chat app to send our login page through the application.

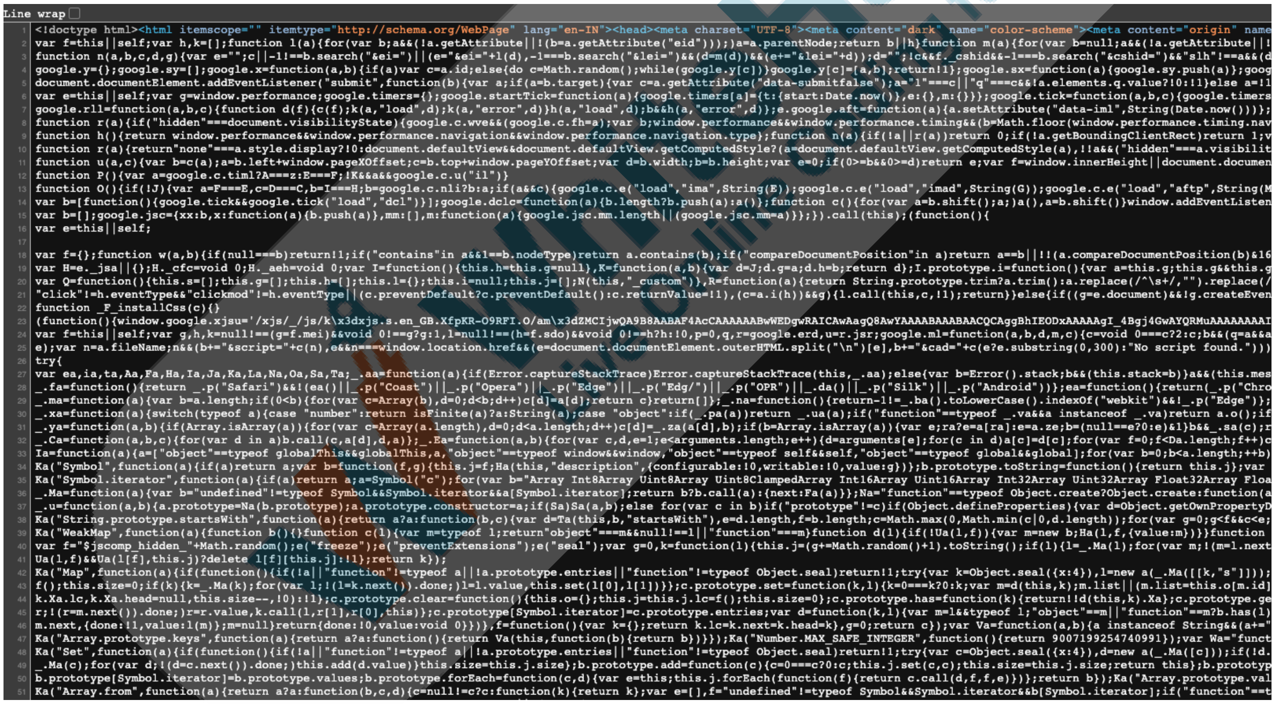
<table>
<tr><td>Before we perform any phishing attack and clone a page, let’s first understand the process.
Please refer to Student Activity 1
Teacher opens the Teacher Activity 1</td><td>*Student opens the Student Activity 1*</td></tr>
<tr><td colspan="2">

Google Search
I'm Feeling Lucky
Google offered in: हिन्दी বাংলা తెలుగు मराठी தமிழ் ગુજરાતી ಕನ್ನಡ മലയാളം ਪੰਜਾਬੀ

</td></tr>
<tr><td>This is Google's search engine’s website. Google is one of the most secure websites out there, and has invested heavily in cyber security.
Let’s try to see the code for this website. For that, we can start by right clicking on the page -</td><td></td></tr>
</table>



We can then click on ***"View Page Source"*** to see the entire code of the website -

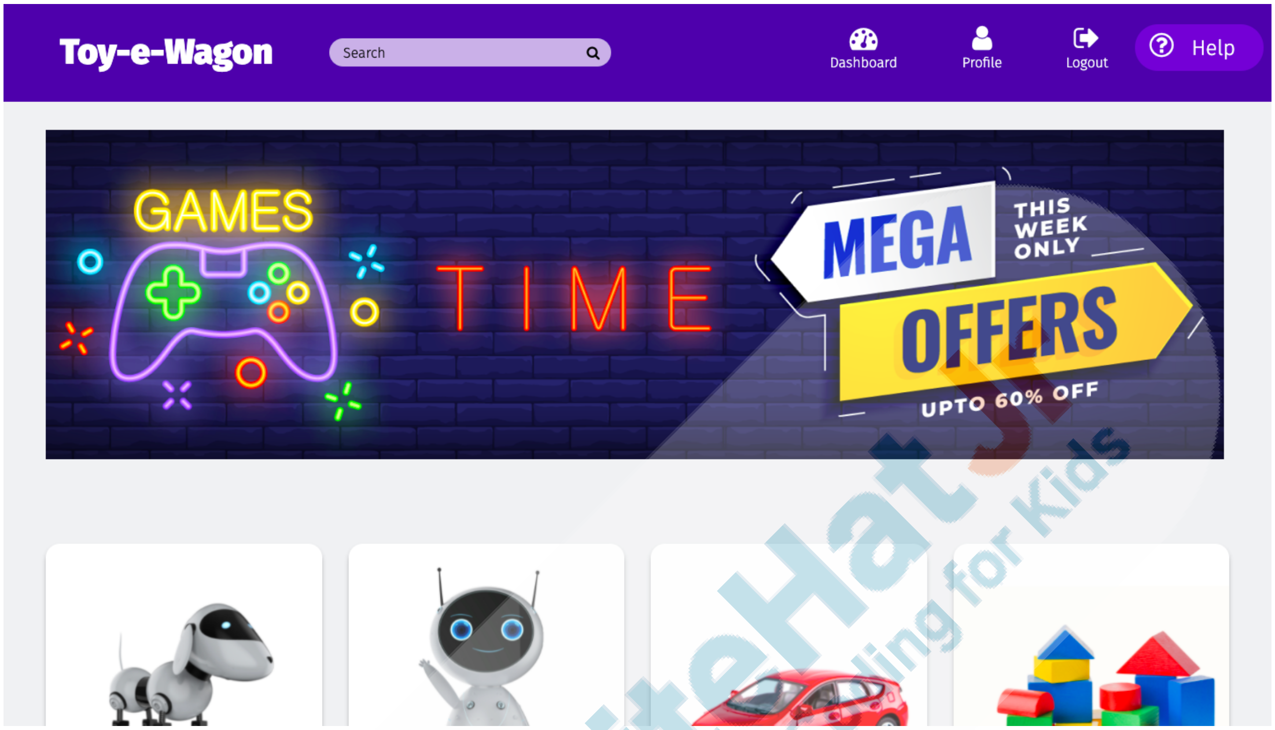


```
<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="en-IN"><head><meta charset="UTF-8"><meta content="dark" name="color-scheme"><meta content="origin" name
var f=this||self;var h,k=[];function l(a){for(var b;a&&(!a.getAttribute||!(b=a.getAttribute("eid")));)a=a.parentNode;return b||h}function m(a){for(var b=null;a&&(!a.getAttribute||!
function n(a,b,c,d,g){var e="";c||-1!==b.search("&ei=")||(e="&ei="+l(d),-1===b.search("&lei=")&&(d=m(d))&&(e+="&lei="+d));d="";!c&&f._cshid&&-1===b.search("&cshid=")&&"slh"!==a&&(d
google.y={};google.sy=[];google.x=function(a,b){if(a)var c=a.id;else{do c=Math.random();while(google.y[c])}google.y[c]=[a,b];return!1};google.sx=function(a){google.sy.push(a)};goog
document.documentElement.addEventListener("submit",function(b){var a;if(a=b.target){var c=a.getAttribute("data-submitfalse");a="1"===c||"q"===c&&!a.elements.q.value?!0:!1}else a=!1
var e=this||self;var g=window.performance;google.timers={};google.startTick=function(a){google.timers[a]={t:{start:Date.now()},e:{},m:{}}};google.tick=function(a,b,c){google.timers
google.rll=function(a,b,c){function d(f){c(f);k(a,"load",d);k(a,"error",d)}h(a,"load",d);b&&h(a,"error",d)};e.google.aft=function(a){a.setAttribute("data-iml",String(Date.now()))};
function r(a){if("hidden"===document.visibilityState){google.c.wve&&(google.c.fh=a);var b;window.performance&&window.performance.timing&&(b=Math.floor(window.performance.timing.nav
function h(){return window.performance&&window.performance.navigation&&window.performance.navigation.type};function n(a){if(!a||r(a))return 0;if(!a.getBoundingClientRect)return 1;v
function r(a){return"none"===a.style.display?!0:document.defaultView&&document.defaultView.getComputedStyle?(a=document.defaultView.getComputedStyle(a),!!a&&("hidden"===a.visibilit
function u(a,c){var b=c(a);a=b.left+window.pageXOffset;c=b.top+window.pageYOffset;var d=b.width;b=b.height;var e=0;if(0>=b&&0>=d)return e;var f=window.innerHeight||document.documen
function P(){var a=google.c.timl?A===z:E===F;!K&&a&&google.c.u("il")}
function O(){if(!J){var a=F===E,c=D===C,b=I===H;b=google.c.nli?b:a;if(a&&c){google.c.e("load","ima",String(E));google.c.e("load","imad",String(G));google.c.e("load","aftp",String(M
var b=[function(){google.tick&&google.tick("load","dcl")}];google.dclc=function(a){b.length?b.push(a):a()};function c(){for(var a=b.shift();a;)a(),a=b.shift()}window.addEventListen
var b=[];google.jsc={xx:b,x:function(a){b.push(a)},mm:[],m:function(a){google.jsc.mm.length||(google.jsc.mm=a)}};}).call(this);(function(){
var e=this||self;

var f={};function w(a,b){if(null===b)return!1;if("contains"in a&&1==b.nodeType)return a.contains(b);if("compareDocumentPosition"in a)return a==b||!!(a.compareDocumentPosition(b)&16
var H=e._jsa||{};H._cfc=void 0;H._aeh=void 0;var I=function(){this.h=this.g=null},K=function(a,b){var d=J;d.g=a;d.h=b;return d};I.prototype.i=function(){var a=this.g;this.g&&this.g
var Q=function(){this.s=[];this.g=[];this.h=[];this.l={};this.i=null;this.j=[];N(this,"_custom")},R=function(a){return String.prototype.trim?a.trim():a.replace(/^\s+/,"").replace(/
"click"!=h.eventType&&"clickmod"!=h.eventType||(c.preventDefault?c.preventDefault():c.returnValue=!1),(c=a.i(h))&&g){l.call(this,c,!1);return}}else{if((g=e.document)&&!g.createEven
function _F_installCss(c){}
(function(){window.google.xjsu='/xjs/_/js/k\x3dxjs.s.en_GB.XfpKR-O9RFI.O/am\x3dZMCIjwQA9B8AABAF4AcCAAAAAABwWEDgwRAICAwAagQ8AwYAAAABAAABAACQCAggBhIEODxAAAAAgI_4Bgj4GwAYQRMuAAAAAAAAI
var f=this||self;var g,h,k=null!==(g=f.mei)&&void 0!==g?g:1,l=null!==(h=f.sdo)&&void 0!==h?h:!0,p=0,q,r=google.erd,u=r.jsr;google.ml=function(a,b,d,m,c){c=void 0===c?2:c;b&&(q=a&&a
e);var n=a.fileName;n&&(b+="&script="+c(n),e&&n===window.location.href&&(e=document.documentElement.outerHTML.split("\n")[e],b+="&cad="+c(e?e.substring(0,300):"No script found.")))
try{
var ea,ia,ta,Aa,Fa,Ha,Ia,Ja,Ka,La,Na,Oa,Sa,Ta;_.aa=function(a){if(Error.captureStackTrace)Error.captureStackTrace(this,_.aa);else{var b=Error().stack;b&&(this.stack=b)}a&&(this.mes
_.fa=function(){return _.p("Safari")&&!(ea()||_.p("Coast")||_.p("Opera")||_.p("Edge")||_.p("Edg/")||_.p("OPR")||_.da()||_.p("Silk")||_.p("Android"))};ea=function(){return(_.p("Chro
_.ma=function(a){var b=a.length;if(0<b){for(var c=Array(b),d=0;d<b;d++)c[d]=a[d];return c}return[]};_.na=function(){return-1!=_.ba().toLowerCase().indexOf("webkit")&&!_.p("Edge")};
_.xa=function(a){switch(typeof a){case "number":return isFinite(a)?a:String(a);case "object":if(_.pa(a))return _.ua(a);if("function"==typeof _.va&&a instanceof _.va)return a.o();if
_.ya=function(a,b){if(Array.isArray(a)){for(var c=Array(a.length),d=0;d<a.length;d++)c[d]=_.za(a[d],b);if(b=Array.isArray(a)){var e;ra?e=a[ra]:e=a.ze;b=(null==e?0:e)&1}b&&_.sa(c);r
_.Ca=function(a,b,c){for(var d in a)b.call(c,a[d],d,a)};_.Ea=function(a,b){for(var c,d,e=1;e<arguments.length;e++){d=arguments[e];for(c in d)a[c]=d[c];for(var f=0;f<Da.length;f++)c
Ia=function(a){a=["object"==typeof globalThis&&globalThis,a,"object"==typeof window&&window,"object"==typeof self&&self,"object"==typeof global&&global];for(var b=0;b<a.length;++b)
Ka("Symbol",function(a){if(a)return a;var b=function(f,g){this.j=f;Ha(this,"description",{configurable:!0,writable:!0,value:g})};b.prototype.toString=function(){return this.j};var
Ka("Symbol.iterator",function(a){if(a)return a;a=Symbol("c");for(var b="Array Int8Array Uint8Array Uint8ClampedArray Int16Array Uint16Array Int32Array Uint32Array Float32Array Floa
_.Ma=function(a){var b="undefined"!=typeof Symbol&&Symbol.iterator&&a[Symbol.iterator];return b?b.call(a):{next:Fa(a)}};Na="function"==typeof Object.create?Object.create:function(a
_.u=function(a,b){a.prototype=Na(b.prototype);a.prototype.constructor=a;if(Sa)Sa(a,b);else for(var c in b)if("prototype"!=c)if(Object.defineProperties){var d=Object.getOwnPropertyD
Ka("String.prototype.startsWith",function(a){return a?a:function(b,c){var d=Ta(this,b,"startsWith"),e=d.length,f=b.length;c=Math.max(0,Math.min(c|0,d.length));for(var g=0;g<f&&c<e;
Ka("WeakMap",function(a){function b(){}function c(l){var m=typeof l;return"object"===m&&null!=l||"function"===m}function d(l){if(!Ua(l,f)){var m=new b;Ha(l,f,{value:m})}}function
var f="$jscomp_hidden_"+Math.random();e("freeze");e("preventExtensions");e("seal");var g=0,k=function(l){this.j=(g+=Math.random()+1).toString();if(l){l=_.Ma(l);for(var m;!(m=l.next
Ua(l,f)&&Ua(l[f],this.j)?delete l[f][this.j]:!1};return k});
Ka("Map",function(a){if(function(){if(!a||"function"!=typeof a||!a.prototype.entries||"function"!=typeof Object.seal)return!1;try{var k=Object.seal({x:4}),l=new a(_.Ma([[k,"s"]]));
f();this.size=0;if(k){k=_.Ma(k);for(var l;!(l=k.next()).done;)l=l.value,this.set(l[0],l[1])}};c.prototype.set=function(k,l){k=0===k?0:k;var m=d(this,k);m.list||(m.list=this.o[m.id]
k.Xa.lc,k.Xa.head=null,this.size--,!0):!1};c.prototype.clear=function(){this.o={};this.j=this.j.lc=f();this.size=0};c.prototype.has=function(k){return!!d(this,k).Xa};c.prototype.ge
r;!(r=m.next()).done;)r=r.value,k.call(l,r[1],r[0],this)};c.prototype[Symbol.iterator]=c.prototype.entries;var d=function(k,l){var m=l&&typeof l;"object"==m||"function"==m?b.has(l)
m.next,{done:!1,value:l(m)};m=null}return{done:!0,value:void 0}}}},f=function(){var k={};return k.lc=k.next=k.head=k},g=0;return c});var Va=function(a,b){a instanceof String&&(a+="
Ka("Array.prototype.keys",function(a){return a?a:function(){return Va(this,function(b){return b})}});Ka("Number.MAX_SAFE_INTEGER",function(){return 9007199254740991});var Wa="funct
Ka("Set",function(a){if(function(){if(!a||"function"!=typeof a||!a.prototype.entries||"function"!=typeof Object.seal)return!1;try{var c=Object.seal({x:4}),d=new a(_.Ma([c]));if(!d.
_.Ma(c);for(var d;!(d=c.next()).done;)this.add(d.value)}this.size=this.j.size};b.prototype.add=function(c){c=0===c?0:c;this.j.set(c,c);this.size=this.j.size;return this};b.prototyp
b.prototype[Symbol.iterator]=b.prototype.values;b.prototype.forEach=function(c,d){var e=this;this.j.forEach(function(f){return c.call(d,f,f,e)})};return b});Ka("Array.prototype.val
Ka("Array.from",function(a){return a?a:function(b,c,d){c=null!=c?c:function(k){return k};var e=[],f="undefined"!=typeof Symbol&&Symbol.iterator&&b[Symbol.iterator];if("function"==t
```

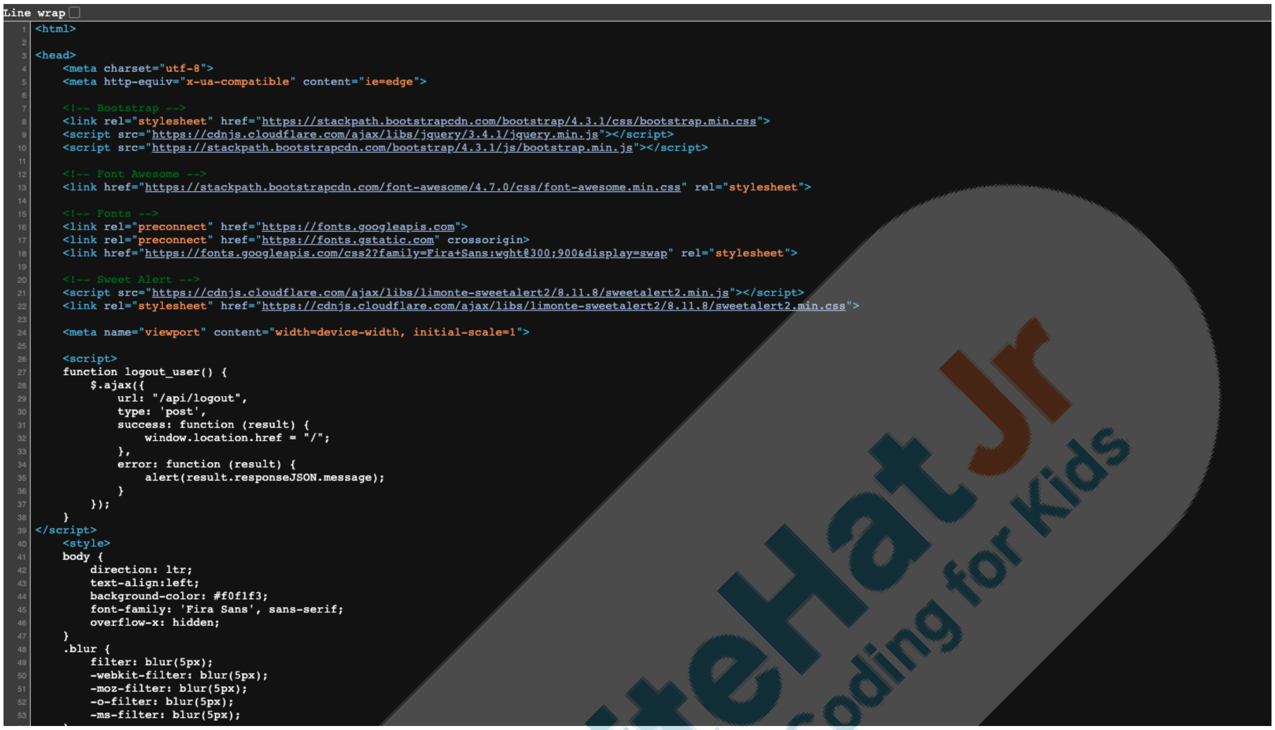
You would notice that this website's code is pretty different from what we normally see.

We can identify it as some JavaScript that is written in a

weird way, without any line breaks or code indentations, and the code is not readable at all. The job of a security engineer is to make it as difficult as it can be for a person to hack or even clone the source code of the page and make it work elsewhere. Therefore, what they do is, they create the entire frontend code of the website purely in JavaScript, that then creates relative HTML and CSS code for the page through functions, and this HTML and CSS is displayed in the browser using **<iframe>** tags. Iframe tags are used to create a browser inside a browser, and therefore it becomes extremely difficult for any hacker to simply clone the website, let alone misuse it to deceive their victims. In our case, however, we know that the website is not so secure. In the last class, we were simply able to fetch data from the profile page for any user we wanted. Let's refer back to our website, and login with the following credentials - ***Email -*** john.doe@gmail.com ***Password -*** hello_john *Teacher refers to Teacher Activity 2 and logins*	*Student refers to Student Activity 2 and logins*



Now let’s navigate to the profile page and view the page source *Teacher navigates to the profile page, rights click on the page and clicks on view page source*	*Student observes*

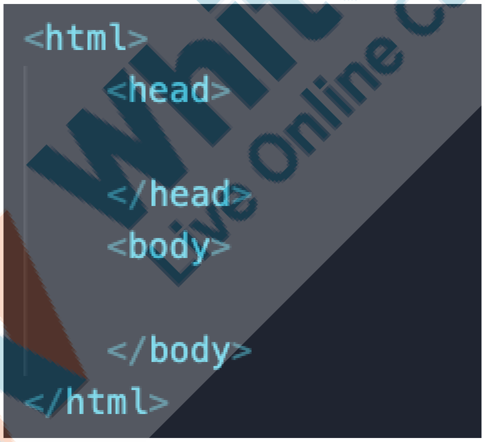


```
Line wrap □
<html>

<head>
    <meta charset="utf-8">
    <meta http-equiv="x-ua-compatible" content="ie=edge">

    <!-- Bootstrap -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>

    <!-- Font Awesome -->
    <link href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" rel="stylesheet">

    <!-- Fonts -->
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?family=Fira+Sans:wght@300;900&display=swap" rel="stylesheet">

    <!-- Sweet Alert -->
    <script src="https://cdnjs.cloudflare.com/ajax/libs/limonte-sweetalert2/8.11.8/sweetalert2.min.js"></script>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/limonte-sweetalert2/8.11.8/sweetalert2.min.css">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <script>
    function logout_user() {
        $.ajax({
            url: "/api/logout",
            type: 'post',
            success: function (result) {
                window.location.href = "/";
            },
            error: function (result) {
                alert(result.responseJSON.message);
            }
        });
    }
</script>
    <style>
    body {
        direction: ltr;
        text-align:left;
        background-color: #f0f1f3;
        font-family: 'Fira Sans', sans-serif;
        overflow-x: hidden;
    }
    .blur {
        filter: blur(5px);
        -webkit-filter: blur(5px);
        -moz-filter: blur(5px);
        -o-filter: blur(5px);
        -ms-filter: blur(5px);
```

Here, we can notice that the structure is pretty different from what we saw in Google's page source. We can clearly see all the CDNs imported, the styling created and even the JS functions that the page uses. Can you identify the HTML structure that we usually follow?	**ESR:** Yes
Awesome! Now, if we wanted to display this page on this website to deceive the user, do you think it would make more sense to just copy and paste the page's data? That's right! There are 2 ways in which this can be done - 1. Manually copying and pasting all the code of this web page and making the fixes. 2. Fetching the HTML of this page through JavaScript code, tweaking it with some functions and then displaying it.	**ESR:** Yes!

While both of the above approaches would do the same thing, the second approach can be virtually used to clone most of the pages without investing much time!	
Let's give it a thought. Do you think, with the help of JavaScript, we can make a GET request to the profile page, and fetch it's entire page source?	**ESR:** Yes!
Okay, and since jQuery is used for DOM manipulation, do you think we can manipulate it's code with some simple jQuery before displaying it in our page?	**ESR:** Yes!
Great! There is one more real advantage of this approach. Can you identify and tell me what it is?	**ESR:** Varied!
The thing is, if we copy-paste the code manually, it might be the case that sometime in future, it's design may change, and our cloned page would then no longer incorporate the design changes, making it difficult to deceive anyone with the page. If we, however, use JavaScript to clone the page, any design changes that happen later in time would also be incorporated in our page, since it will always fetch fresh code. Now, let's spend some time cloning this page, and make it look realistic so that it can be used to deceive other people with the help of JavaScript and jQuery!	
STUDENT-LED ACTIVITY - 20 mins	
● Ask the student to press the ESC key to come back to the panel. **● Guide the student to start Screen Share.** **● The teacher gets into Full Screen.**	

ACTIVITY ● **Clone the page with AJAX and JavaScript** ● **Manipulate the page with jQuery**	
Teacher Action	**Student Action**
Okay, so the very first thing that we need is an HTML file! Let's open VS Code and create a new file called ***index.html*** and also create the basic structure of the HTML with ***html, head and body*** tags. *Teacher guides the student to open VS Code and create the new file **index.js** and create the basic HTML template*	*Student opens the VS Code and creates a new file **index.js** and creates the basic HTML structure in it.*

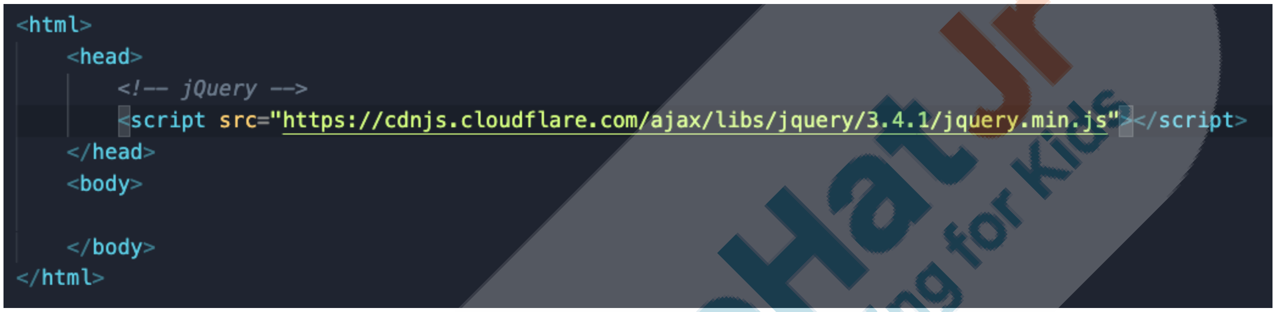

`<html>`
`    <head>`
`    </head>`
`    <body>`
`    </body>`
`</html>`	

Awesome! Now, the next thing that we want to do is to fetch the HTML of the profile page from the ecommerce website. For that, we will have to make a GET request on the URL with the help of AJAX. AJAX is a jQuery function, so let's include it in our **<head>** tag!	

Teacher gives the student the following line to copy and paste.

Student copy pastes the following line to add jQuery into the HTML in the head tag.

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



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

    </body>
</html>
```

Awesome! Now, all we have to do is to make the AJAX call.

Before making the AJAX call, we need to ensure that our page has loaded properly. For that, do you remember which function can we use in jQuery?

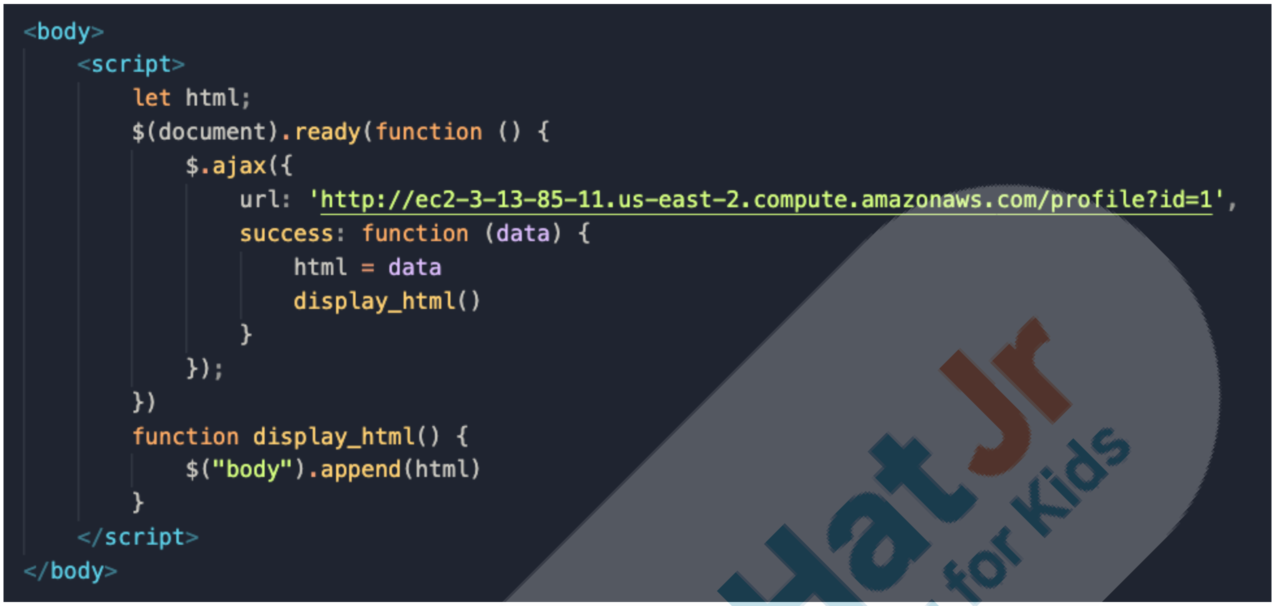
We can use the ***$(document).ready()*** function!

Let's create a ***script*** tag inside the ***body*** tag and make the AJAX call to fetch the page -

Teacher helps the student in writing the code

ESR:
Varied!

Student writes the code



```
<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>
```

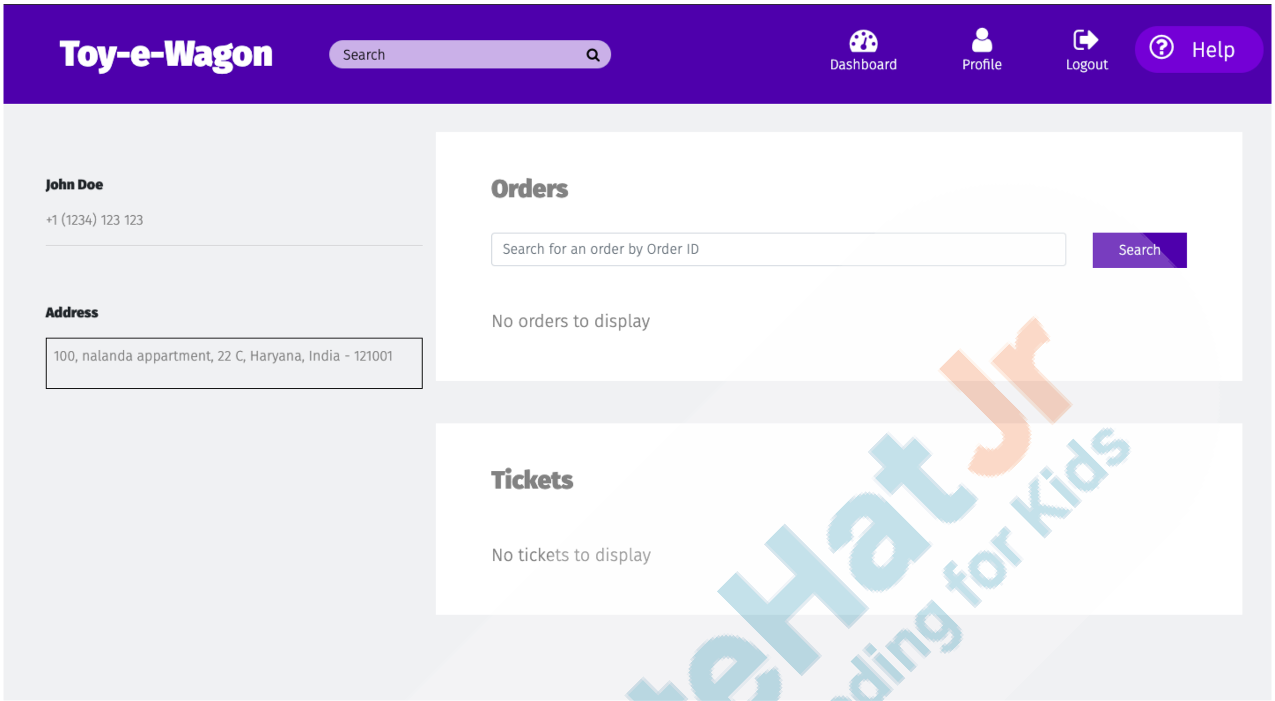
Here, inside of the script tag, we have created a variable called ***html*** to store our HTML when we retrieve it.

Next, we have our ***$(document).ready(function())*** which will wait for the page to load, and then make the AJAX call.

We have provided our Profile page's URL to make the AJAX call, and in its success handler function, we have saved the data (which will be the profile page's HTML) inside the variable HTML.

Once we are done, we are finally calling a function called ***display_html()*** which is defined just below it, and will append the HTML in the body.

If we try to open the ***index.html*** page in the browser, we would see the following -



John Doe

+1 (1234) 123 123

Address

100, nalanda appartment, 22 C, Haryana, India - 121001

Orders

Search for an order by Order ID

No orders to display

Tickets

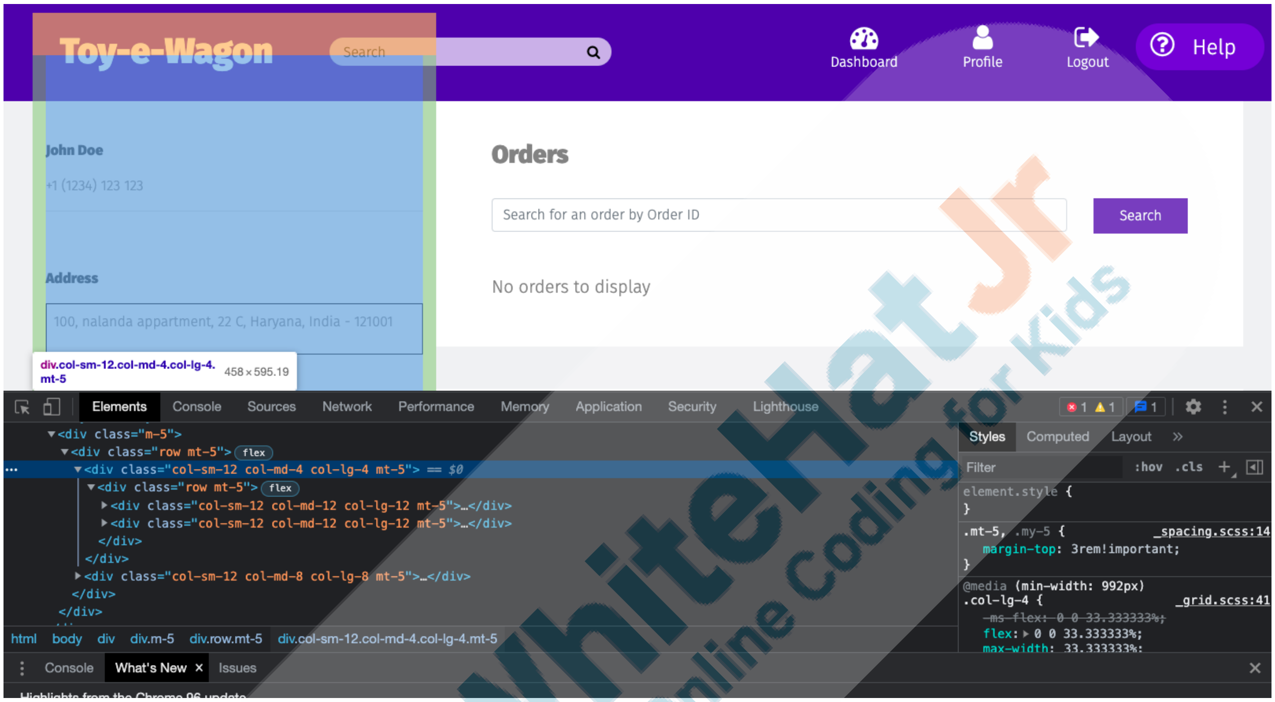
No tickets to display

Looks completely like the profile page, isn't it? We just did it with a few lines of code!	**ESR:** Yes!
Now, most of the things are sorted for this page, but there is still one problem. Can you identify that? We still have the name, mobile number and address of a user, which may make it suspicious to the victim, since it is not their details! The best way to deal with this scenario, is to remove that entire section. For that, we would have to tweak our HTML page a little. Let's first inspect our page by right clicking on it and clicking	**ESR:** Varied!

on inspect in the browser -

Teacher guides the student

Student inspects the page



Toy-e-Wagon

John Doe

+1 (1234) 123 123

Address

100, nalanda appartment, 22 C, Haryana, India - 121001

Orders

Search for an order by Order ID

No orders to display

div.col-sm-12.col-md-4.col-lg-4.mt-5 458 × 595.19

```
▼<div class="m-5">
  ▼<div class="row mt-5"> flex
    ▼<div class="col-sm-12 col-md-4 col-lg-4 mt-5"> == $0
      ▼<div class="row mt-5"> flex
        ▶<div class="col-sm-12 col-md-12 col-lg-12 mt-5">…</div>
        ▶<div class="col-sm-12 col-md-12 col-lg-12 mt-5">…</div>
       </div>
     </div>
    ▶<div class="col-sm-12 col-md-8 col-lg-8 mt-5">…</div>
   </div>
 </div>
```

```
element.style {
}
.mt-5, .my-5 {                    _spacing.scss:14
   margin-top: 3rem!important;
}
@media (min-width: 992px)
.col-lg-4 {                          _grid.scss:41
   ~~-ms-flex: 0 0 33.333333%;~~
   flex: ▶ 0 0 33.333333%;
   max-width: 33.333333%;
```

html body div div.m-5 div.row.mt-5 div.col-sm-12.col-md-4.col-lg-4.mt-5

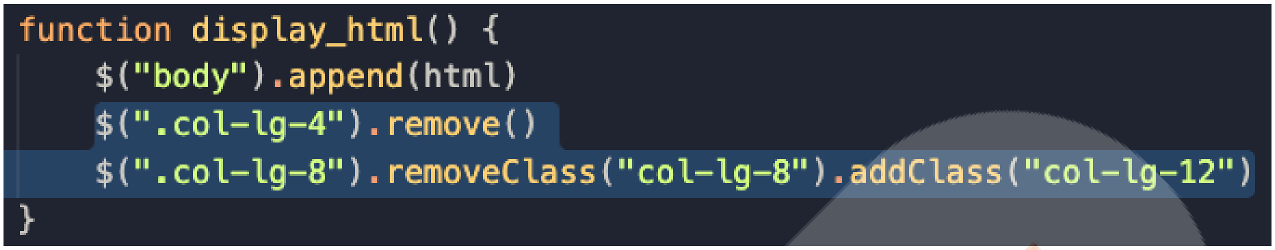
If you look closely, you would see a pattern of bootstrap. Inside the very first row, there are 2 columns -

1. First one for the left panel with width 4/12
2. Second one for the right panel with width 8/12

If we manage to remove the first div (with class ***col-lg-4***) and make the second div have class ***col-lg-12*** instead of ***col-lg-8***, then we will eliminate the left section and would only be left with the right section.

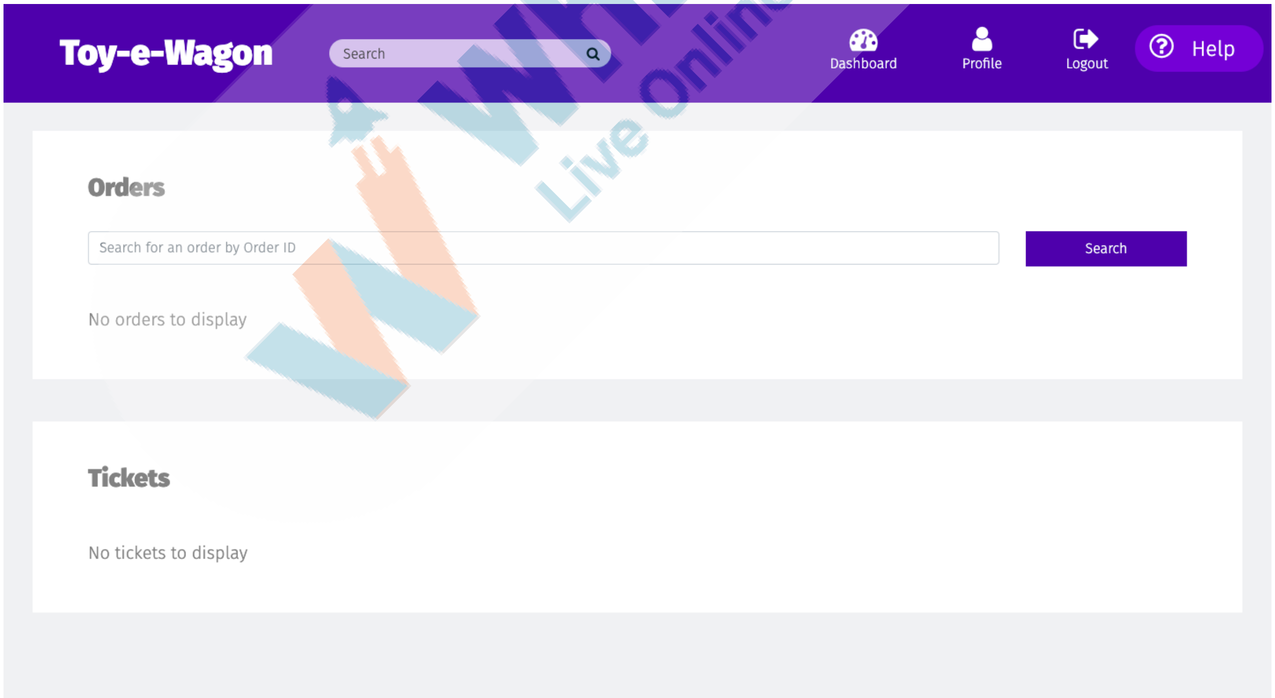
Let's make those changes!

Teacher guides the student in writing the code	*Student writes the code*



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

Here, we are using jQuery to first find a div with class ***col-lg-4*** and we are calling the ***remove()*** function to remove it entirely, and then we are finding a div with class ***col-lg-8*** and we are using the ***removeClass()*** function to remove the ***col-lg-8*** class from it. We are then using the ***addClass()*** function to add another class to it, which is ***col-lg-12***. With this, our page would look like -	



Orders

Search for an order by Order ID

No orders to display

Tickets

No tickets to display

Now this does not contain any sensitive data for the user to get suspicious. Since the design is almost similar, the user may feel that the website has changed it's design a bit and is no longer displaying the information it used to earlier. With just a couple of lines of code, we managed to clone an entire page exactly as it is in real time, and also managed to tweak it's code. This cloning technique is used by a lot of hackers, to create an illusion for the victim that they are in the right place, and they end up giving up their information to the attacker unknowingly by trusting the page. In the next class, we will build this further to demonstrate a potential phishing attack due to one of the vulnerabilities in this website!	

Teacher Guides Student to Stop Screen Share

WRAP UP SESSION - 5 Mins

Quiz time - Click on in-class quiz

Question	Answer
Which function is used to remove a class? 1. remove() 2. removeclass() 3. classremove() 4. removeClass()	**D**
Which function runs as soon as the page is ready? 1. $(document).ready() 2. document.ready() 3. $(document).load()	**A**

4. document.load()	
Which of the following can be used to remove an entire element from an HTML? 1. removeClass() 2. removeElement() 3. remove() 4. Remove()	**C**
End the quiz panel	
FEEDBACK • **Appreciate the students for their efforts in the class.** • **Ask the student to make notes for the reflection journal along with the code they wrote in today's class.**	
Teacher Action	**Student Action**
You get Hats off for your excellent work! In the next class we will learn about SQL Union	*Make sure you have given at least 2 Hats Off during the class for:* Creatively Solved Activities +10 Great Question +10 Strong Concentration +10
Project Discussion You have been asked by a client that they would like to see if anyone could clone their website's login page and if it would still work with the regular credentials on the	

website, as the hacker could get access to the credentials and misuse them later. Your task is to clone the page and login to check if it works after cloning or not.	
Teacher Clicks	
ADDITIONAL ACTIVITIES	
Additional Activities *Encourage the student to write reflection notes in their reflection journal using markdown.* Use these as guiding questions: ● What happened today? ○ Describe what happened. ○ The code I wrote. ● How did I feel after the class? ● What have I learned about programming and developing games? ● What aspects of the class helped me? What did I find difficult?	*The student uses the markdown editor to write her/his reflections in the reflection journal.*

ACTIVITY LINKS		
Activity Name	**Description**	**Link**
Teacher Activity 1	Google	https://www.google.com/
Teacher Activity 2	Ecommerce Website	http://ec2-3-108-196-161.ap-south-1.compute.amazonaws.com/
Teacher Activity 3	Reference Code	https://github.com/pro-whitehatjr/PRO-C236-Reference-Code
Student Activity 1	Google	https://www.google.com/
Student Activity 2	Ecommerce Website	http://ec2-3-108-196-161.ap-south-1.compute.amazonaws.com/