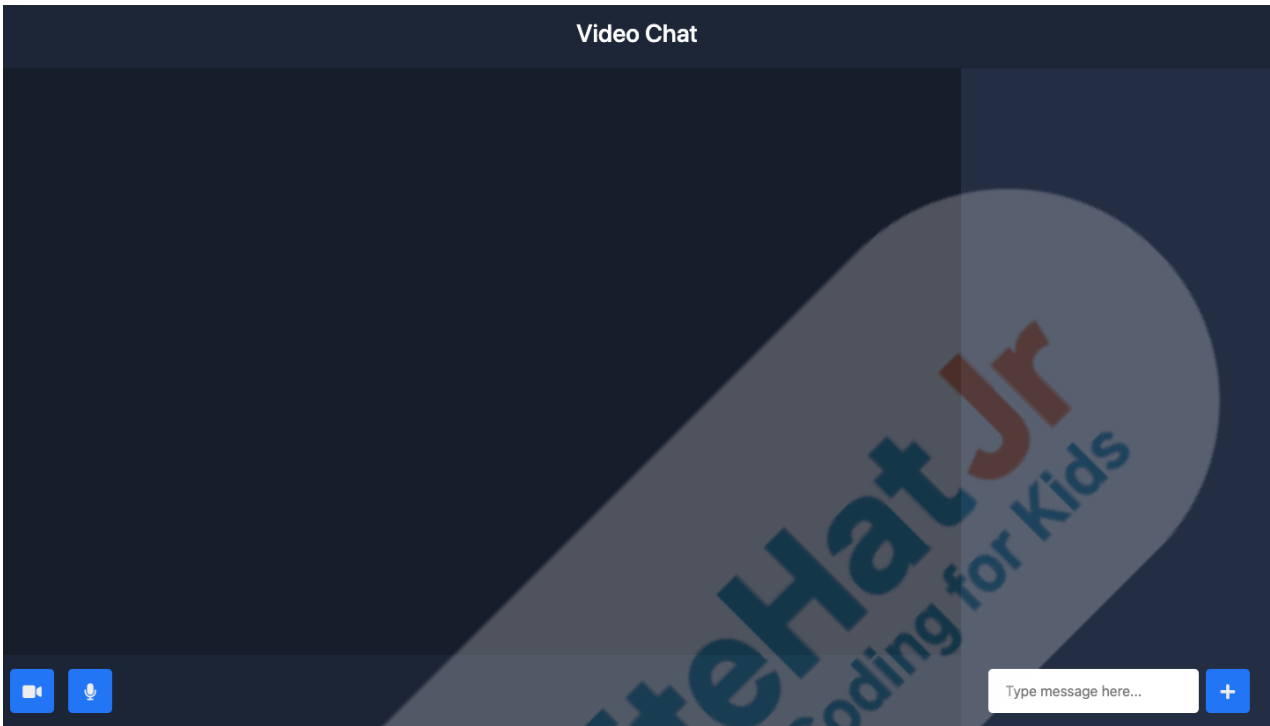


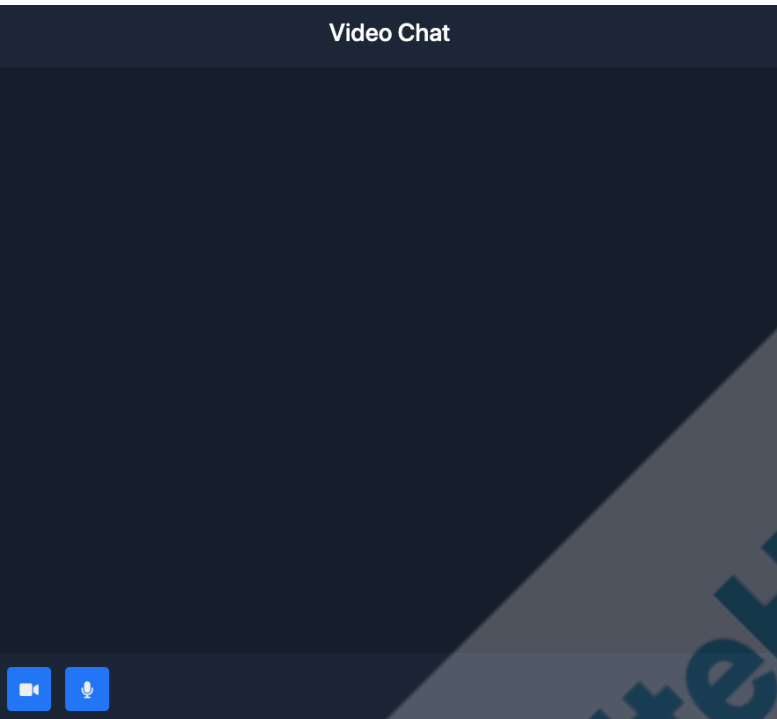
Topic	Video Chat App - UI	
Class Description	Student will start building a video chat app and come up with a responsive UI frontend for it.	
Class	C-214	
Class time	45 mins	
Goal	<ul style="list-style-type: none"> <li>Completing the HTML for the Video Chat App</li> <li>Configuring styles</li> <li>Creating a script to handle responsiveness</li> </ul>	
Resources Required	<ul style="list-style-type: none"> <li>Teacher Resources:               <ul style="list-style-type: none"> <li>Laptop with internet connectivity</li> <li>Earphones with mic</li> <li>Notebook and pen</li> <li>Visual Studio Code</li> </ul> </li> <li>Student Resources:               <ul style="list-style-type: none"> <li>Laptop with internet connectivity</li> <li>Earphones with mic</li> <li>Notebook and pen</li> <li>Visual Studio Code</li> </ul> </li> </ul>	
Class structure	<b>Warm-Up</b> <b>Teacher - led Activity 1</b> <b>Student - led Activity 1</b> <b>Wrap-Up</b>	<b>10 mins</b> <b>15 mins</b> <b>15 mins</b> <b>5 mins</b>
<b>WARM UP SESSION - 10mins</b>		
<b>Teacher Action</b>		<b>Student Action</b>
<i>Hey &lt;student's name&gt;. How are you? It's great to see you!            Are you excited to learn something new today?</i>		<b>ESR:</b> Hi, thanks, yes, I am excited about it!

Q&A Session	
Question	Answer
What is the purpose of using the OS module?  A. Display the operating system B. Make your operating system C. To access a file from an operating system D. Path to operating system	<b>C</b>
What is the purpose of using the <b>storbinary()</b> method?  A. Initiates the transfer of a binary file from an FTP client to an FTP server B. It will create a directory for the accessed path C. Display the operating system D. Make your operating system	<b>A</b>
TEACHER-LED ACTIVITY - 15mins	
Teacher Initiates Screen Share	
<p style="text-align: center;"><u><b>ACTIVITY</b></u></p> <ul style="list-style-type: none"> <li>• <b>Understanding the HTML and Bootstrap Code</b></li> <li>• <b>Adding relevant HTML and CSS for responsiveness</b></li> </ul>	
Teacher Action	Student Action
In today's class, we will start building a new Video Chat App!  Now we have already seen how we can communicate between clients through sockets when we first built a chat app, but can't we do the same for video as well?	<p><b>ESR</b> Yes!</p> <p><b>ESR</b></p>

<p>What are the general features of a video chat app?</p>	<p>A video chat app can have</p> <ul style="list-style-type: none"> <li>-</li> <li>1. Video Streaming</li> <li>2. Audio Streaming</li> <li>3. Chat Box</li> <li>4. Functionality to close video or audio</li> <li>5. Share the video call link with someone else</li> </ul>
<p>Great! These are all the features that we will try to build in our video chat application.</p> <p>Before we start writing any of the functionality, we first need a UI which can facilitate all these features, right?</p> <p>Let's start by cloning some pre-written boilerplate code.</p>	
<p><i>Teacher refers to <a href="#">Teacher Activity 1</a> and clones the repository</i></p>	<p><i>Student refers to <a href="#">Student Activity 1</a></i></p>
<p>Okay! Now let's open the <b><i>index.html</i></b> file to see what the output is -</p> <p><i>Teacher opens <b>index.html</b> file and shows the output to the student</i></p>	

 <p>The image shows a video chat interface. At the top, it says 'Video Chat'. Below this is a large dark area representing the video feed. In the bottom left corner, there are icons for video and audio. In the bottom right corner, there is a text input field with the placeholder 'Type message here...' and a blue '+' button.</p>	
<p>That looks like a good UI!</p> <p>Now, one thing that we have to keep in mind is that this video application that we are building would be running on a browser.</p> <p>This means that the users can be using this website from their laptop or their PC, or they could also be using their mobile phones!</p> <p>We know that it looks good on a Desktop, but how do we check if it looks good on a mobile as well?</p>	<p><b>ESR:</b> Varied!</p>
<p>Browsers nowadays offer a solution for that!</p> <p>Right click on the page and click on inspect to inspect the page -</p> <p><i>Teacher inspects the page</i></p>	

### Video Chat



Elements Console Sources Network

```

<!DOCTYPE html>
<html lang="en">
<head>
</head>
<body>
  <div class="row" style="overflow: hidden;">
    <div class="row main">
      <div class="col-sm-12 col-md-12 col-lg-9 left-window">
        <div class="col-sm-12 col-md-12 col-lg-12" style="height: 81vh; background-color: #171e2a;">
          <!-- Video Streams -->
        </div>
      </div>
      <div class="col-sm-12 col-md-12 col-lg-12 options">
      </div>
    </div>
    <div class="col-sm-12 col-md-12 col-lg-3 right-window">
    </div>
  </div>
</body>
</html>

```

html body div.row.main div.col-sm-12.col-md-12.col-lg-9.left-window div.row div...

Styles Computed Layout Event Listeners DOM Breakpoints Properties

Filter show .cls +

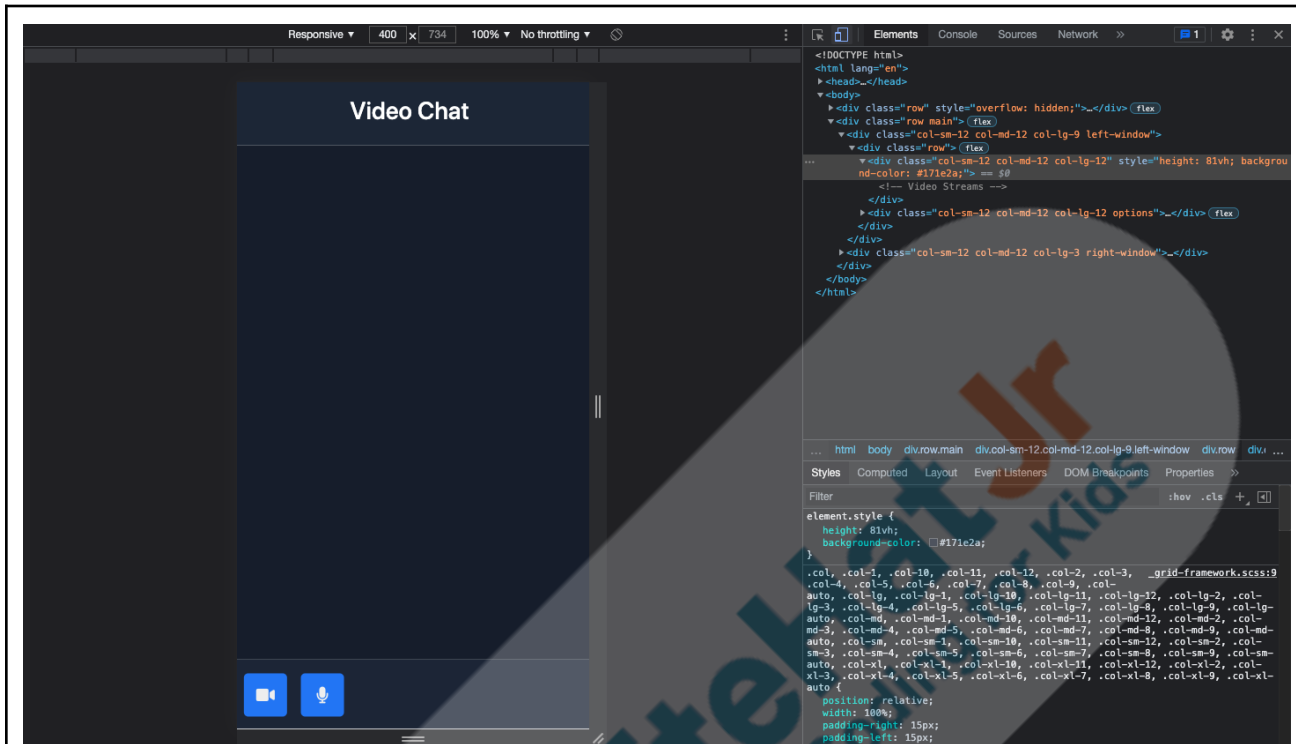
```

element.style {
  height: 81vh;
  background-color: #171e2a;
}
@media (min-width: 768px)
.col-md-12 {
  flex: 0 0 100%;
  max-width: 100%;
}
@media (min-width: 576px)
.col-sm-12 {
  flex: 0 0 100%;
  max-width: 100%;
}
.col, .col-1, .col-10, .col-11, .col-12, .col-2, .col-3, .col-4, .col-5, .col-6, .col-7, .col-8, .col-9, .col-

```

Upon inspecting, on the top left corner of the inspect window, you would see a mobile like icon! Click on that -

*Teacher clicks the icon -*



And there we can see how it looks on a mobile.

Now we can see that it looks all good, but the chat window is completely gone.

What we want to do in today's class is to ensure that our chat window also gets displayed on a mobile as well!

For that, we will have to make some changes to the code, so before we proceed, let's take a quick look at the code we have and try to understand it.

*Teacher opens the project in a VS Code*

In **index.html**, the first thing we will see is that we are importing a few things -

```
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Video Chat App</title>
<link rel="stylesheet" href="style.css" />
<script src="https://kit.fontawesome.com/c939d0e917.js"></script>

<!-- 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://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>
</head>
```

Here, we are importing our **style.css**, and then a library for our icons that we are using from **fontawesome.com**

We are also importing all the libraries for bootstrap from **Teacher Activity 2.**

You can refer to Student Activity 1 for reference.

**Teacher refers to Teacher Activity 2 and clones the repository**

**Student refers to Student Activity 1**

Do you know what bootstrap is and why is it used for?

**ESR:**  
Varied!

Bootstrap is a famous styling library used to make websites responsive. This means that it enables a website to be designed in a way that it looks fine in all kinds of displays, such as in desktop, tablets as well as mobile phones.

Bootstrap follows a box model, and works in **rows** and **columns**. This means that everything that our page consists of is made up of **rows** and **columns**.

One thing however, to always keep in mind while working with bootstrap is that the content should always be inside a **column** instead of directly being inside a **row**.

To understand this better, let's take a look at the beginning of our HTML's body -

```
<body>
  <div class="row" style="overflow: hidden;">
    <div class="col-sm-12 col-md-12 col-lg-12 text-center p-3" style="background-color: #1d2735;">
      <h3 class='text-white'>Video Chat</h3>
    </div>
  </div>
```

Here, we can see that we first have a **<div>** tag, which contains a class called **row**. This defines a bootstrap row.

Inside this div, we have another div tag with class **col-sm-12 col-md-12 col-lg-12 text-center p-3**.

Now, what does all these mean?

In bootstrap, a container can be divided into 12 different sections in terms of width.

- **col** defines a bootstrap column.
- **sm** defines column's width in small screen (mobile)
- **md** defines column's width in medium screen (tablet)
- **lg** defines column's width in large screen (desktop or laptop)
- **text-center** simply means to have all the text in the center of this column.
- **p-3** is for padding. The number **3** here could have been anything from **1-5**.

Therefore here, **col-sm-12** means to have full width of the row in small screen, **col-md-12** means to have full width of the row in medium screen and **col-lg-12** means to have full width of the row in large screen.

Inside this div tag, we have an **h3** tag in which we display **Video Chat**, which is the heading of our screen.

Now let's take a look at the code below it -

**ESR:**  
Varied!



```
<div class="row main">
  <div class="col-sm-12 col-md-12 col-lg-9 left-window">
    <div class="row">
      <div class="col-sm-12 col-md-12 col-lg-12" style="height: 81vh; background-color: #171e2a;">
        <!-- Video Streams -->
      </div>
      <div class="col-sm-12 col-md-12 col-lg-12 options">
        <!-- Icons -->
        <div id="stop_video" class="options_button">
          <i class="fa fa-video-camera"></i>
        </div>
        <div id="mute_button" class="options_button">
          <i class="fa fa-microphone"></i>
        </div>
      </div>
    </div>
  </div>
</div>
```

Below our first div, we again have a div that is a bootstrap row.

Inside this div, we have a column div with class **col-sm-12 col-md-12 col-lg-9**. This time, what it means is that this column would have full width in small and medium size screens, but in large screens, we want to have 9/12 width for this column. We have also given this section a class called **left-window**.

Now if you notice how our page looks in Desktop and Mobile, we can notice that it was not taking full width in desktop but taking full width in the mobile. This is why we were seeing that behaviour.

Now again, in this div, we have a **row** div and then inside it, we have **2 full width columns**, where the first one is for Video Streams that we will display later, and the second one is for the icons.

Right below this code, we have another column -

```
<div class="col-sm-12 col-md-12 col-lg-3 right-window">
  <div class="row" class="main-chat-window">
    <div class="col-sm-12 col-md-12 col-lg-12 messages" style="height: 81vh; background-color: #242f41;">
    </div>
    <div class="col-sm-12 col-md-12 col-lg-12" style="background-color: #242f41;">
      <div class="main_message_container">
        <input id="chat_message" type="text" autocomplete="off" placeholder="Type message here...">
        <div id="send" class="options_button">
          <i class="fa fa-plus" aria-hidden="true"></i>
        </div>
      </div>
    </div>
  </div>
</div>
```

Here, we can see that this column has a class **col-sm-12 col-md-12 col-lg-3**. We also gave this section a class called **right-window**.

Now since our last column had **col-lg-9**, this one has **col-lg-3** which means that this should take up the remaining of the space next to it.

Inside this column, again we have a **row** div inside which we have **2 full width columns**. Note here, that since this container only has 3/12 width of the screen, the full width columns here will also have 3/12 width only.

The first column now is for the messages that the users will type and the second div is for the input, where they can type their messages.

Similarly, in our **style.css**, we have all the relevant styling for this page.

We now understand why the chat box was not visible in the mobile view while it was there in the desktop view, but how do we solve this? How does it happen in other apps?

We have noticed that in mobile, there is a button on the screen through which a chat window opens! Let's try to build that in our app as well.

**ESR:**  
Varied!

The first thing that we want to do for this is to add another icon for messages. Let's do that -

*Teacher writes the code -*

```
<div class="col-sm-12 col-md-12 col-lg-12 options">
  <!-- Icons -->
  <div id="stop_video" class="options_button">
    <i class="fa fa-video-camera"></i>
  </div>
  <div id="mute_button" class="options_button">
    <i class="fa fa-microphone"></i>
  </div>
  <div id="show_chat" class="options_button">
    <i class="fa fa-comment"></i>
  </div>
</div>
```

Here, we added one more icon next to the 2 icons that we already had! Do note that we have given this icon a class called **options\_button** so it looks like the other 2 buttons, and also an **id** called **show\_chat**

Let's check its output -

Video Chat



Type message here... 

Awesome! We can see the icon now!

There is one problem though. We do not want this icon to be visible on the desktop, but only on mobile.

To deal with this, let's understand more about CSS. CSS by default works for desktop screens, so if we give this icon a property called ***display: none***, then it will not be visible to us, but then we also want to make sure it is visible to us on mobile view.

For that, CSS has one special feature called ***media queries***, in which we can write different CSS for different screen sizes! Let's see how!

In our ***style.css***, let's first add ***display: none*** to this icon -

```
#show_chat {
  display: none;
}
```

And next, we will create a media query for screen sizes below 700px, as that's generally true for all the mobile screens -

```
#show_chat {
  display: none;
}

@media (max-width: 700px) {
  #show_chat {
    display: flex;
  }
}
```

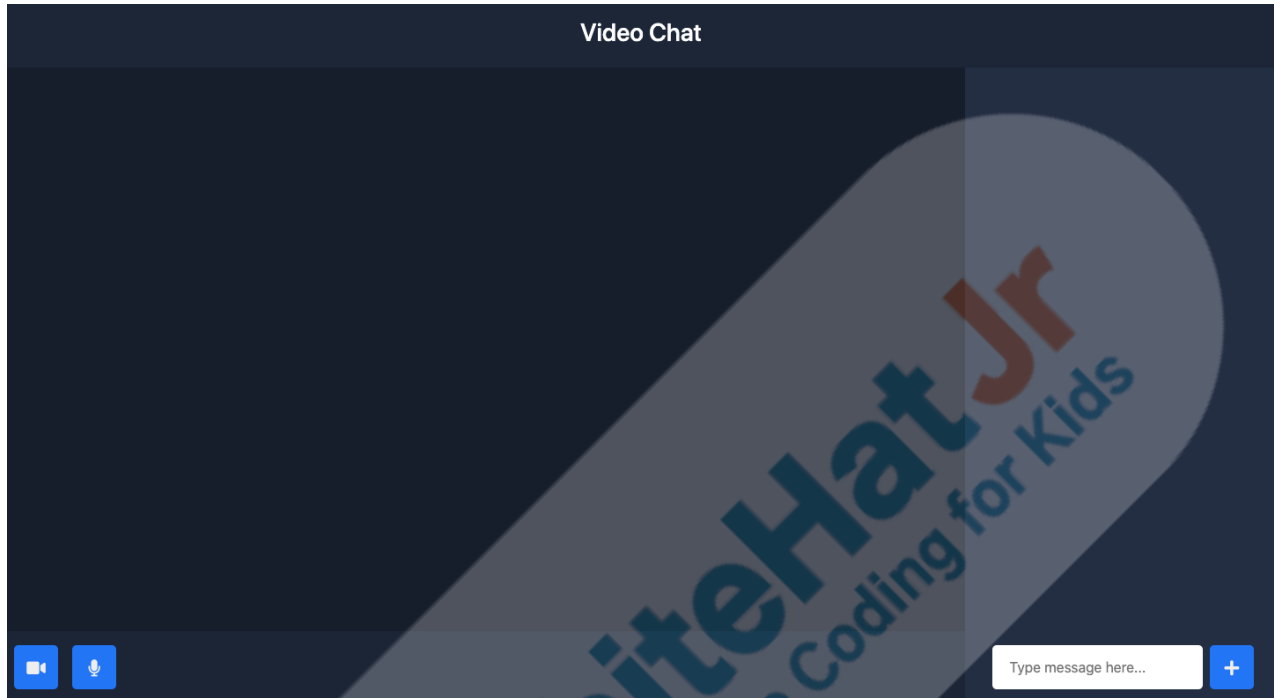
Here, we have written a media query by using **@media** keyword, and defined it's condition that the **max-width** for it to work shall be **700px**.

Inside this media query, we have again added a styling for our **#show\_chat** button, but this time around, we have kept **display** to **flex**.

Do note that media queries should **ALWAYS** go at the last of the CSS file, or your styles may not reflect properly.

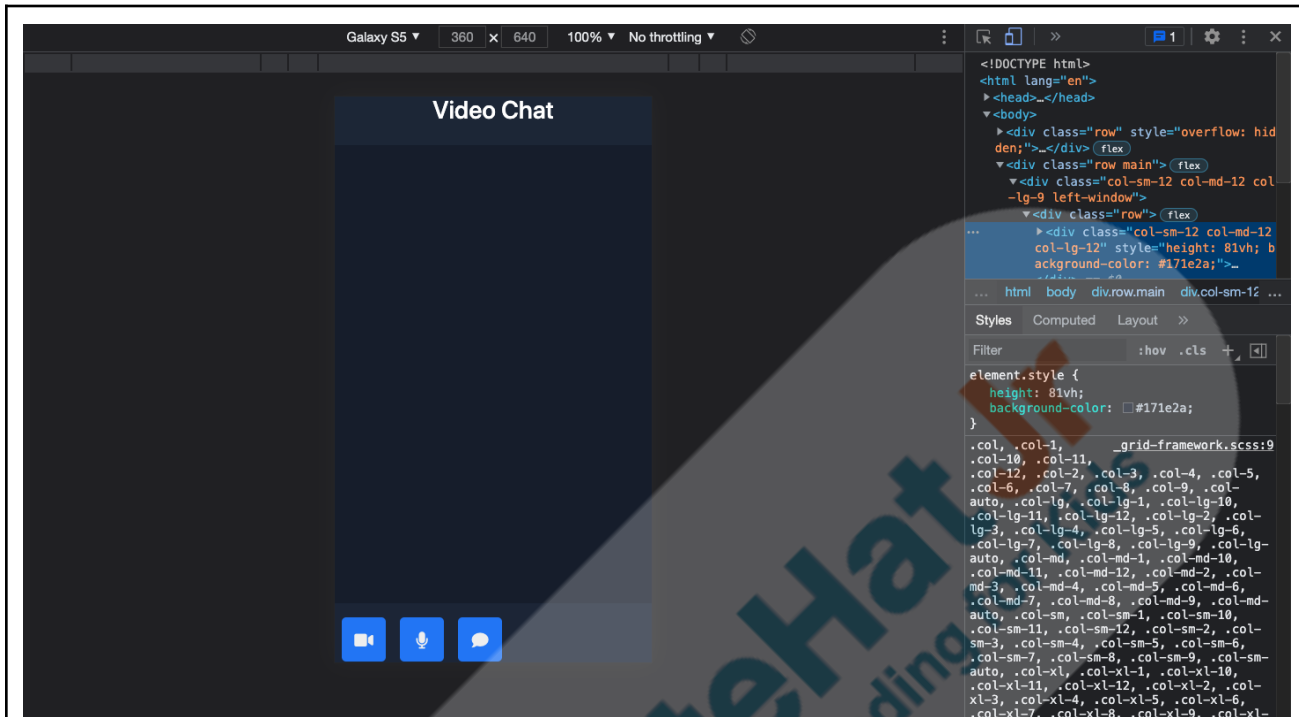
Now let's check the output for both **Desktop** as well as

**Mobile view -**



© 2021 - WhiteHat Education Technology Private Limited.

Note: This document is the original copyright of WhiteHat Education Technology Private Limited.  
Please don't share, download or copy this file without permission.



Here, we can see that now our chat icon is visible in the mobile view but not in the desktop view!

Awesome!

Now let's give it some thought! As soon as the user clicks the chat button, we will open the chat box, but how will the user get back to the main screen? There should be a back button for the same, right?

Let's add that as well -

```
<div class="col-sm-12 col-md-12 col-lg-12 text-center p-3" style="background-color: #1d2735;">
  <div class="header_back">
    <i class="fas fa-angle-left"></i>
  </div>
  <h3 class='text-white'>Video Chat</h3>
</div>
```

Our video chat heading was visible in both the desktop and mobile, so it's better we add the back icon there as reflected in the screenshot above. Do note that we have given it a class called **header\_back**. We will also add it's relevant styling in the **style.css** file -

```
.header_back {
  display: none;
  position: absolute;
  font-size: 1.3rem;
  top: 17px;
  left: 28px;
  color: ■ #fff;
}
```

Lastly, just to be sure, we will add styles for our **left-window** and **right-window** in our media query, to ensure that by default when the page loads on a mobile, only the **left-window** is visible and the **right-window** is hidden, so our UI doesn't breaks -



```
@media (max-width: 700px) {
  #show_chat {
    display: flex;
  }

  .right_window {
    display: none;
  }

  .left_window {
    display: flex;
  }
}
```

Awesome! Now our page is ready but our chat button and the back button that we just added are still not functional. For that, you will be writing some JavaScript and jQuery!

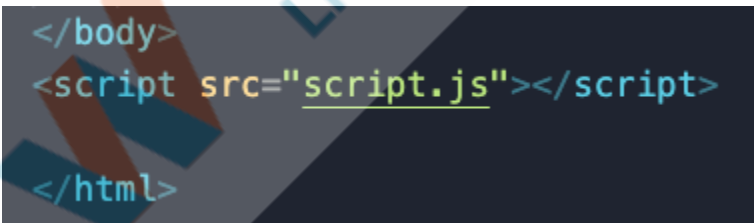
### Teacher Stops Screen Share

### STUDENT-LED ACTIVITY - 15 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

- Add jQuery code to make chat button functional
- Add jQuery code to make back button functional

Teacher Action	Student Action
<p><i>Guide the student to get the boilerplate code from <a href="#">Student Activity 2</a></i></p>	<p><i>Student clones the code from <a href="#">Student Activity 2</a></i></p>
<p>Do you remember jQuery, and how we used it to create event handlers for things like button clicks, etc. in the previous module?</p> <p>Don't worry if you don't remember it much! We will work it out together!</p> <p>First thing that we need to do is to create a new file called <b>script.js</b>.</p> <p><i>Student creates a new file</i></p> <p>Now let's import this file into our <b>index.html</b> below our <b>body</b> tag</p> <p><i>Teacher guides the student to import <b>script.js</b> in <b>index.html</b></i></p>	<p><b>ESR:</b> Varied!</p> <p><i>Student creates the file</i></p>
 <pre> &lt;/body&gt; &lt;script src="script.js"&gt;&lt;/script&gt; &lt;/html&gt; </pre>	
<p>Okay! Now to create event handlers, we will create a \$ function. Remember that in jQuery, all the event handlers must be placed inside a \$ function or else they won't work.</p> <p><i>Teacher helps the student in creating a \$ function inside <b>script.js</b></i></p>	<p><i>Student writes the code</i></p>

```
$(function () {  
  
})
```

Okay! Now the first event handler is for our **#show\_chat** button.

What should happen when it is clicked?

**ESR:**

The **left-window** should be hidden and the **right-window** and **header\_back** button should be displayed.

Good! Let's do that -

*Teacher helps the student in writing the code*

*Student writes the code*

```
$(function () {  
    $("#show_chat").click(function () {  
        $(".left-window").css("display", "none")  
        $(".right-window").css("display", "block")  
        $(".header_back").css("display", "block")  
    })  
})
```

Okay, now here, we have first creating a function for the **click()** event on the **#show\_chat** button, inside which, we are altering the **css()** of the **.left-window**, **.right-window** and the **.header\_back** button.

1. **.left-window** display property is changing to **none**
2. **.right-window** display property is changing to **block**
3. **.header\_back** display property is changing to **block**

Do note that our chat button only gets displayed on mobile, so this code is only for mobile too.

Now similarly, what should happen when the **.header\_back** button is clicked?

Correct! Let's code that too -

*Teacher helps the student in completing the code*

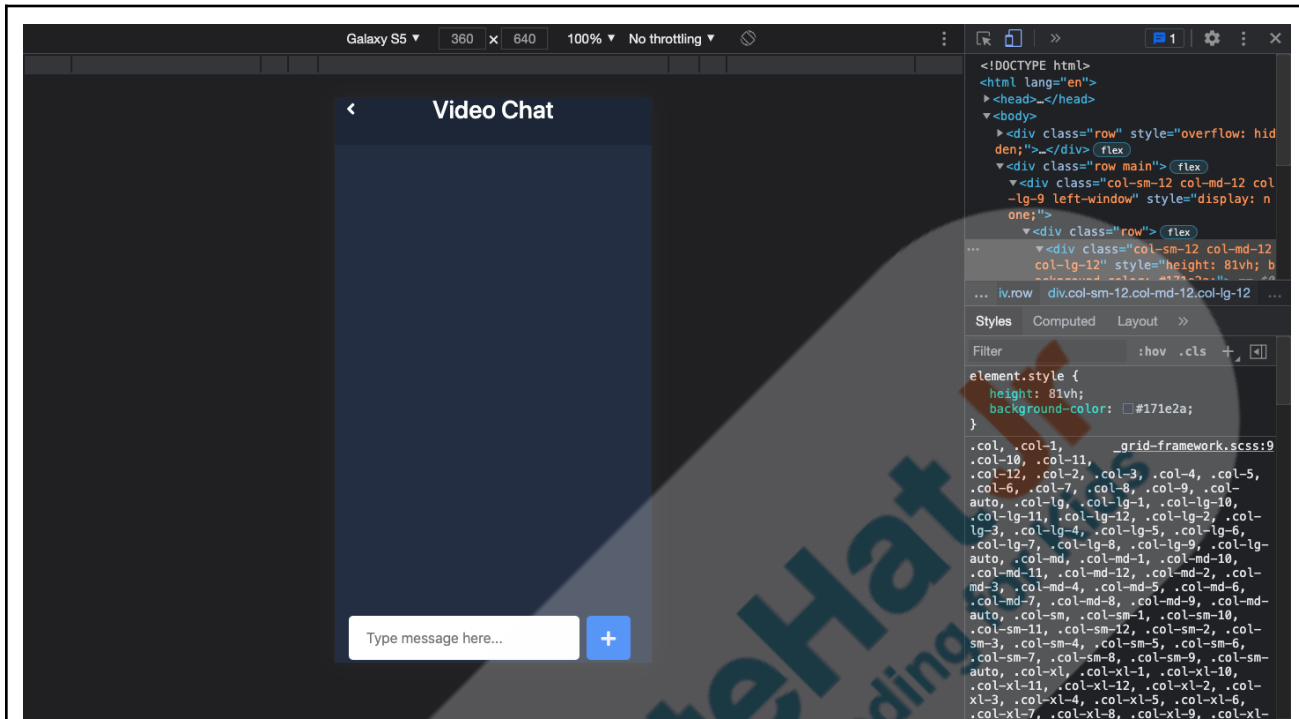
**ESR:**

The **.left-window** is displayed while the **.right-window** and **.header\_back** button is hidden.

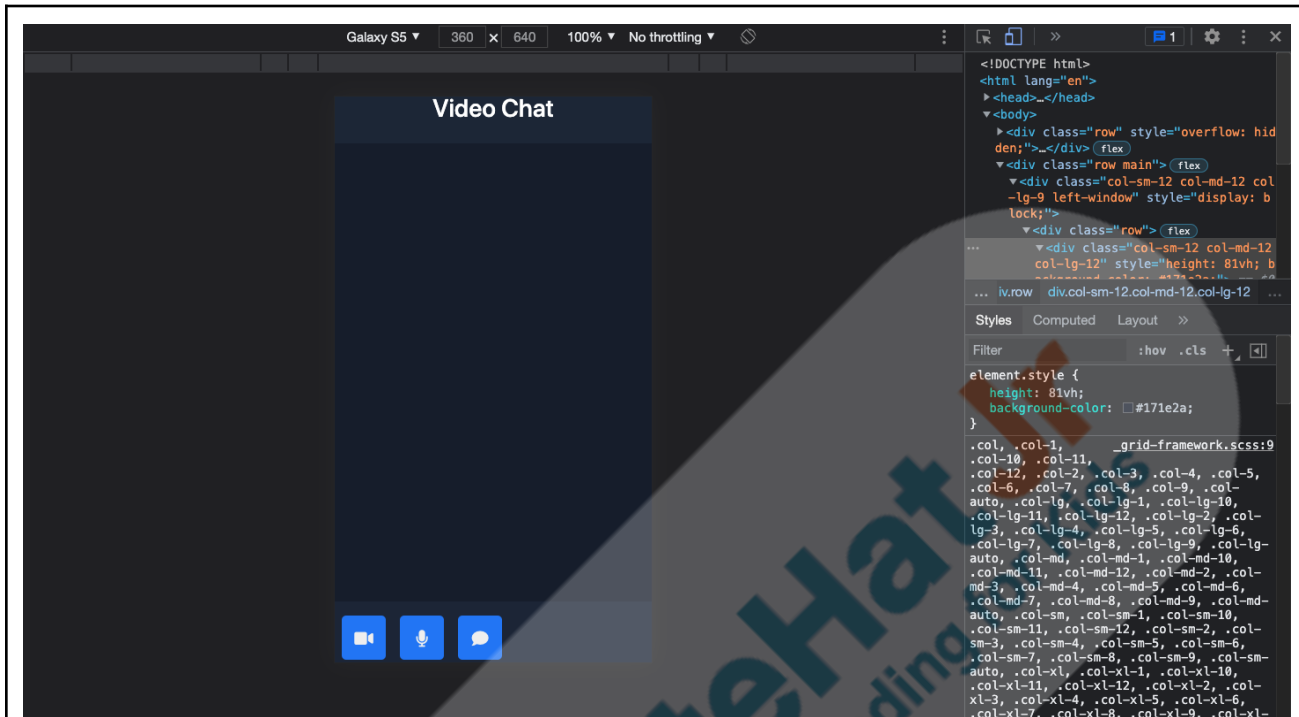
*Student completes the code*

```
$(function () {
  $("#show_chat").click(function () {
    $(".left-window").css("display", "none")
    $(".right-window").css("display", "block")
    $(".header_back").css("display", "block")
  })
  $(".header_back").click(function () {
    $(".left-window").css("display", "block")
    $(".right-window").css("display", "none")
    $(".header_back").css("display", "none")
  })
})
```

Now let's test it on mobile display! First, on clicking the chat button, we can see that the chat window is displayed along with the back button -



And on click the back button, we go back to our **left-window**






Awesome! With this, the UI of our App is ready. In the next few classes, we will be building the backend server for this frontend, and we will also learn some exciting things like how to send emails from an app to invite users, and we will also be deploying this app so that it can be accessed on the internet from a URL!

**Teacher Guides Student to Stop Screen Share**

**WRAP UP SESSION - 5 Mins**

**Quiz time - Click on in-class quiz**

Question	Answer
<p>What is the purpose of .css?</p> <p>A. For styling B. For Information C. For data D. None of the above</p>	A

What do you mean by responsiveness in coding?  A. Display according to device B. Alignment according to device C. Design as per device D. All of the above	D
What is the purpose of Bootstrap?  A. Make website responsive B. Will access styling sheet C. Helps to design websites faster D. All of the above	D
End the quiz panel	
<b>FEEDBACK</b> <ul style="list-style-type: none"> <li>• <b>Appreciate the students for their efforts in the class.</b></li> <li>• <b>Ask the student to make notes for the reflection journal along with the code they wrote in today's class.</b></li> </ul>	
<b>Teacher Action</b>	<b>Student Action</b>
You get Hats off for your excellent work!  In the next class	<p><i>Make sure you have given at least 2 Hats Off during the class for:</i></p> <div>           Creatively Solved Activities  +10         </div> <div>           Great Question  +10         </div> <div>           Strong Concentration  +10         </div>

<b>Project Discussion</b>	
<div>Teacher Clicks</div> <div>✕ End Class</div>	
<b>ADDITIONAL ACTIVITIES</b>	
<b>Additional Activities</b> <i>Encourage the student to write reflection notes in their reflection journal using markdown.</i>  Use these as guiding questions: <ul style="list-style-type: none"> <li>• What happened today?             <ul style="list-style-type: none"> <li>◦ Describe what happened.</li> <li>◦ The code I wrote.</li> </ul> </li> <li>• How did I feel after the class?</li> <li>• What have I learned about programming and developing games?</li> <li>• What aspects of the class helped me? What did I find difficult?</li> </ul>	<i>The student uses the markdown editor to write her/his reflections in the reflection journal.</i>

<b>ACTIVITY LINKS</b>		
<b>Activity Name</b>	<b>Description</b>	<b>Link</b>
Teacher Activity1	Teacher Boilerplate Code	<a href="https://github.com/pro-whitehatjr/PRO-C214-TeacherBoilerplate">https://github.com/pro-whitehatjr/PRO-C214-TeacherBoilerplate</a>
Teacher Activity 2	Bootstrap Documentation	<a href="https://getbootstrap.com/docs/4.3/getting-started/introd">https://getbootstrap.com/docs/4.3/getting-started/introd</a>

© 2021 - WhiteHat Education Technology Private Limited.

Note: This document is the original copyright of WhiteHat Education Technology Private Limited.

Please don't share, download or copy this file without permission.



		<a href="#">uction/</a>
Teacher Activity 3	Reference Code	<a href="https://github.com/pro-whitehatjr/PRO-C214-ReferenceCode">https://github.com/pro-whitehatjr/PRO-C214-ReferenceCode</a>
Student Activity 1	Bootstrap Documentation	<a href="https://getbootstrap.com/docs/4.3/getting-started/introduction/">https://getbootstrap.com/docs/4.3/getting-started/introduction/</a>
Student Activity 2	Boilerplate Code	<a href="https://github.com/pro-whitehatjr/PRO-C214-StudentBoilerplate">https://github.com/pro-whitehatjr/PRO-C214-StudentBoilerplate</a>