Topic	CUSTOM MARKER AR	
Class Description	**Students will learn about pattern marker based augmented reality. Students will also learn to create pattern markers for web based AR.**	
Class	**C168**	
Class time	**45 mins**	
Goal	• Learn about pattern marker augmented reality web apps. • Learn to create a basic web based AR app using pattern markers. • Learn to create pattern markers.	

Resources Required	• Teacher Resources
	  ○ Visual Studio Code Editor
	  ○ laptop with internet connectivity
	  ○ smartphone
	  ○ earphones with mic
	  ○ notebook and pen
	• Student Resources
	  ○ Visual Studio Code Editor
	  ○ laptop with internet connectivity
	  ○ smartphone
	  ○ earphones with mic
	  ○ notebook and pen	

Class structure	**Warm-Up** **Teacher-led Activity** **Student-led Activity** **Wrap-Up**	**5 mins** **15 mins** **20 mins** **5 mins**
WARM-UP SESSION - 10 mins		
CONTEXT • **Web based A-Frame pattern marker AR.**		

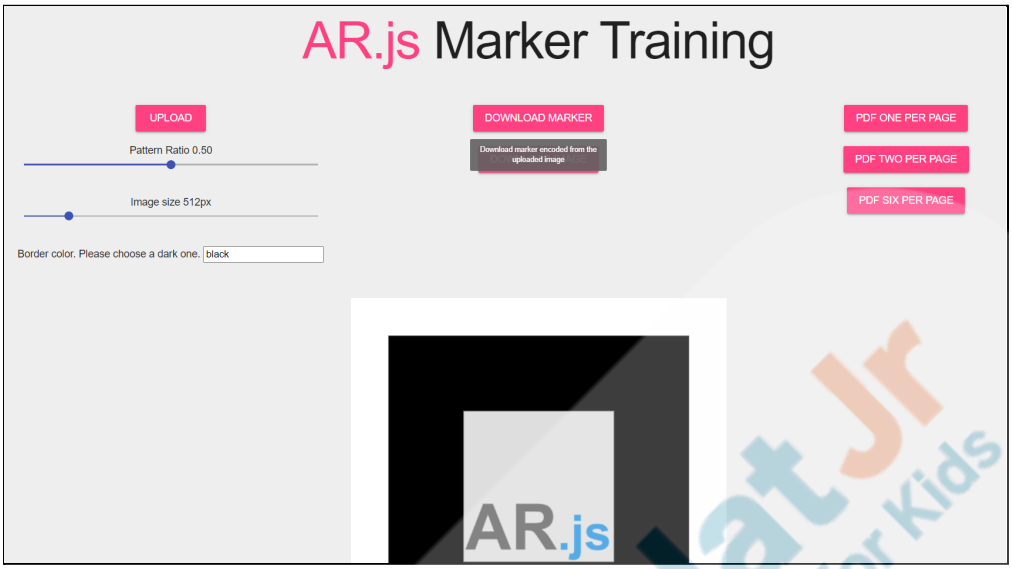
Teacher Starts Slideshow  **Slide 1 to 3** Refer to speaker notes and follow the instructions on each slide.	
Hey <student's name>. How are you? It's great to see you! Are you excited to learn something new today? **Following are the WARM-UP session deliverables:** • Greet the student. • Revision of previous class activities. • Quizzes.	**ESR**: Hi, thanks! Yes I am excited about it! Click on the slide show tab and present the slides
WARM-UP QUIZ Click on In-Class Quiz	
Continue WARM-UP Session  **Slide 4 to 12**	
Following are the session deliverables: • Appreciate the student. • Narrate the story by using hand gestures and voice modulation methods to bring in more interest in students.	

Class Steps	Teacher Action	Student Action
Step 1: Warm-Up (5 mins)	We learned how to use Hiro markers to display content over that, which is a standard marker. We can also create our own marker like we created image trackers. Today we will be learning how we can create our own marker and use it for the AR scenes.	

	We'll see how we can create different markers and use them in upcoming classes. Are you excited?	 **ESR:** Yes.
	Let's get started then.	
Teacher Ends Slideshow 		
TEACHER-LED ACTIVITY - 15 mins		
Teacher Initiates Screen Share		
CHALLENGE • **Create pattern marker** • **Create an A-Frame Web AR scene and render objects using a pattern marker.**		
Step 2: Teacher-led Activity (15 mins)	*<The teacher clones the code from the Teacher Activity 1.>* ***[Teacher Activity 1]*** Do you love food? What if you can actually see the food ingredients and how the food will look in AR? That will be pretty cool, right? You will be able to feel the food closely with Augmented reality's help, and that will help you decide whether you want to eat that food or not.	 **ESR:** Yes/No. **ESR:** Yes.

	Suppose you are sitting in a restaurant and you want to order food. What is the first thing that you look for before ordering the food? Yes, you will find a table and you will go through the menu card and find the dish which you can order. To make the hotel menu card content we need to know foods that we want to show on the menu card. In this class, we'll start designing the AR content for the hotel menu card. Almost everyone loves pizza, so we will start with our first food as pizza. To display the food and its ingredients in AR, we are going to use our own markers. These are called **pattern markers.** Now, to create pattern markers we can take an image and create the pattern marker of that. There are a few points that we need to keep in mind while creating pattern markers: • It would be better if we use small size images to create markers.	**ESR:** The menu card**.**

	• The markers should be in square shape. • The black and grey colors are to be used as the background color of the marker. • It's better to avoid transparent, white and any other colored background. Okay, let's get started. We have the pizza image, and we are going to use it to create the pattern marker. For this, we are going to use another AR.js marker creating tool. Link: https://jeromeetienne.github.io/AR.js/three.js/examples/marker-training/examples/generator.html	



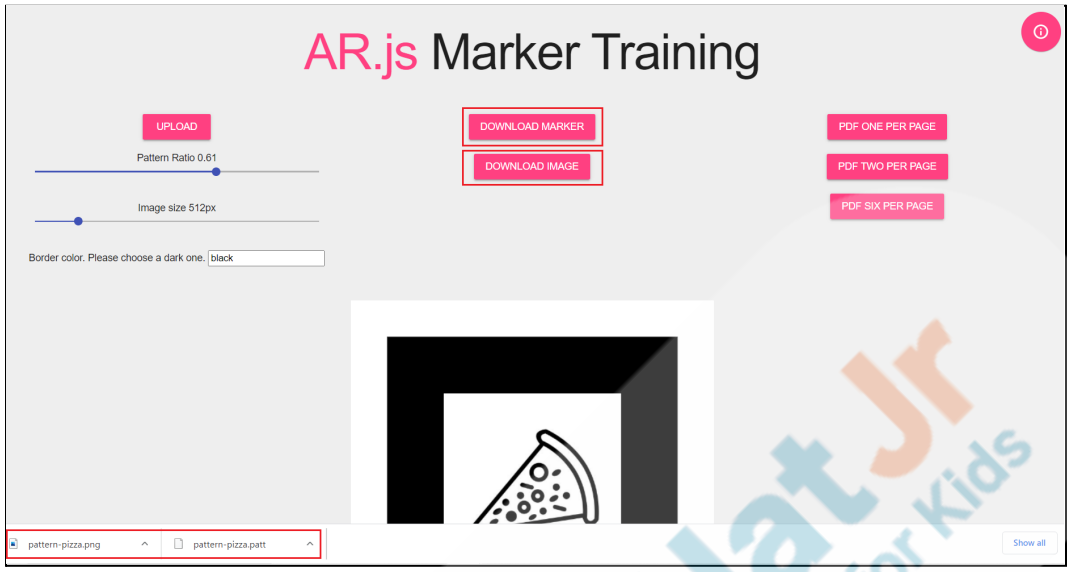
<The teacher opens the link and uploads the image to the tool.>

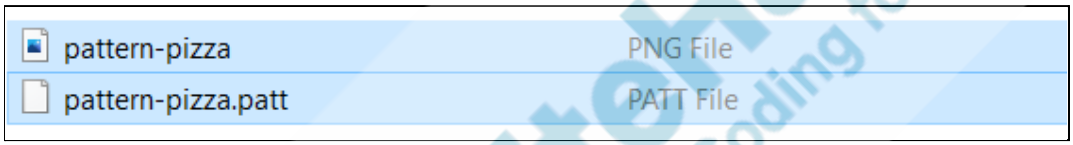
Once the image is uploaded we can set:

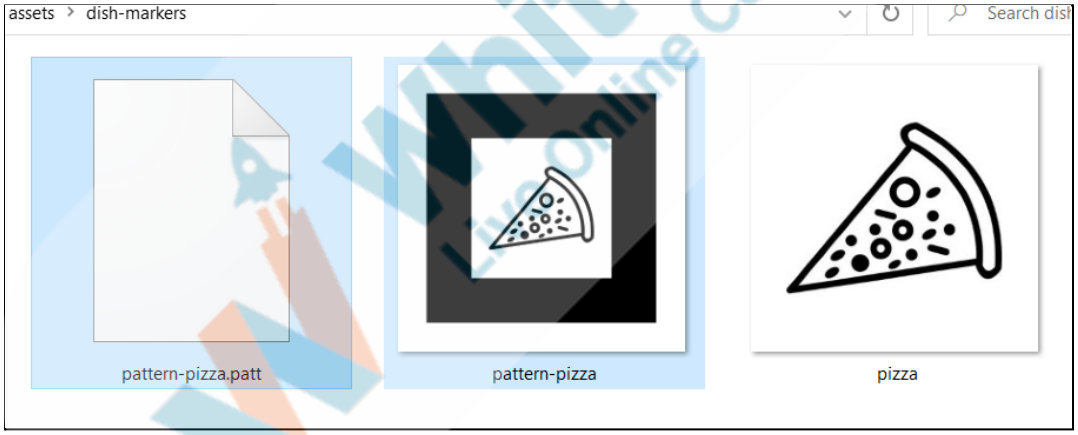
- **Pattern ratio**: this sets the size of the image with respect to the borders.
- **Image size**: size of the image.
- **Background border color**: set to black color.

Click on the download marker and download image.
The marker will be downloaded in the **.patt** file format.

And the new marker image will be downloaded which we can use to detect AR content.







***Note**: You can download the model from the link given in the activity.*

Now let's quickly add the pizza model asset using <a-assets>.

```
<a-assets>
  <a-asset-item id="pizza-model" src="./assets/pizza/scene.gltf"></a-asset-item>
</a-assets>
```

	Now we have to tell the computer to identify the pattern marker. Do you remember what we used to use the Hiro marker? Great! We are going to use <a-marker> again. And how did we tell the computer to make sure that it identifies the Hiro marker? Yes, we use a preset property to tell the computer to identify the Hiro marker because it is one of the default markers. Can you tell me which one is the other default marker? Yes. That's right! But here we are going to set the type property as a pattern to use our custom marker. In <a-marker> we can set : **type**: "pattern" **url**: file path to pattern maker descriptor (including .patt file extension) created before. *<The teacher sets <a-marker> properties.>*	**ESR**: We used <a-marker>. **ESR**: We used the preset property to set the marker type. **ESR**: Kanji marker.

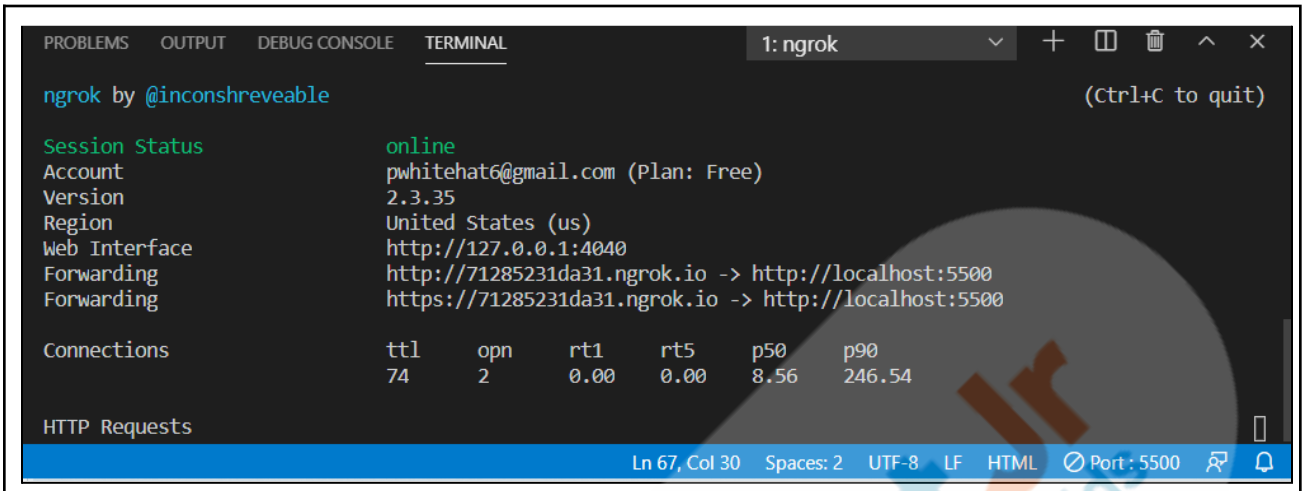
```
<a-marker id="pizza-marker"
  type="pattern"
  url="assets/dish-markers/pattern-pizza.patt"
  cursor="rayOrigin: mouse">


</a-marker>
```

	Now what should we do to display the content of the marker? Yeah! Amazing! Now we can set the **model** and **dish name** with a **list of ingredients** in the dish. We can use <a-plane> to display the text entities over the plane. *<The teacher sets the pizza model entity and the text entities for the name of the dish with ingredients as the child entity of <a-marker> in the scene.>* As an extra feature for the pizza model, we can use the **gesture-handler** component, which is a part of this library, to rotate and scale up/down the model. (Make sure the **gesture-detector** component is attached to the <a-scene>.) Link: https://raw.githack.com/fcor/arjs-gestures/master/dist/gestures.js	ESR: We should add the content as the child of the <a-marker>.

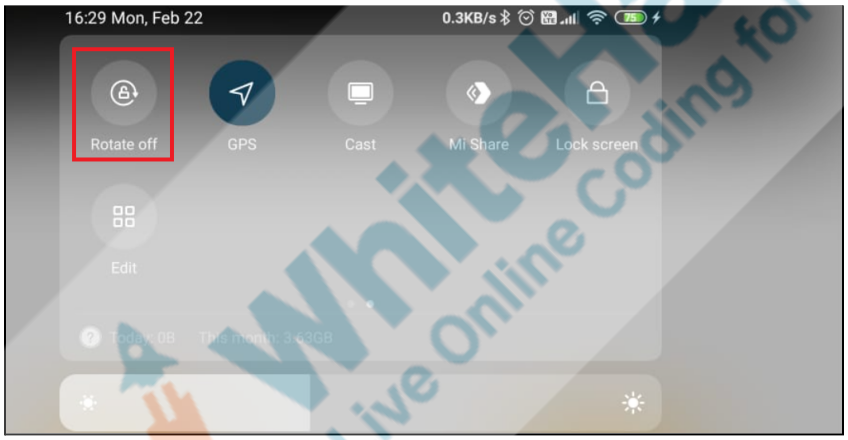
	***Note** : The positions and scale values can be updated to set the orientation properly according to the user.*	

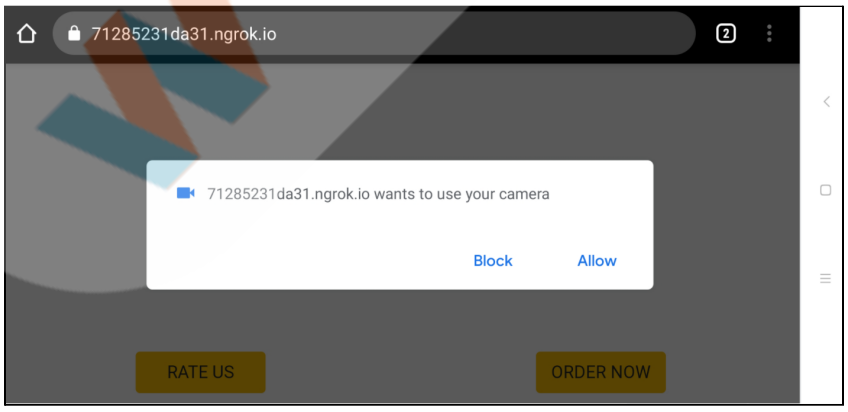
```
<!--Pattern Marker-->
<a-marker id="pizza-marker" type="pattern" url="assets/dish-markers/pattern-pizza.patt" cursor="rayOrigin: mouse">

  <a-entity id="pizza-3D-model" position="-2 0 0" scale="0.05 0.05 0.05"
    gltf-model="#pizza-model"
    gesture-handler>
  </a-entity>


  <!-- Ingredients -->
  <a-plane position="0 0 0" width="1.5" height="1.5" rotation="-90 0 0">
    <a-plane position="0 0.89 0.02" width="1.49" height="0.3" rotation="0 0 0" color="#F0C30F">
      <a-entity position="0 0 0.1" rotation="0 0 0"
        text="font:monoid;value:PIZZA MARGHERITA;color:black;width: 1.8;height:1;align:center;"></a-entity>
    </a-plane>

    <a-entity position="0 0 0.1" rotation="0 0 0"
      text="color: black; align: center; width: 2; font: monoid;
            value: Tomato sauce\n\nMozzarella cheese\n\nParmesan cheese (optional)\n\nFresh Basil\n\nOlive oil\n\nBlack pepper (optional)">
    </a-entity>
  </a-plane>

</a-marker>
```

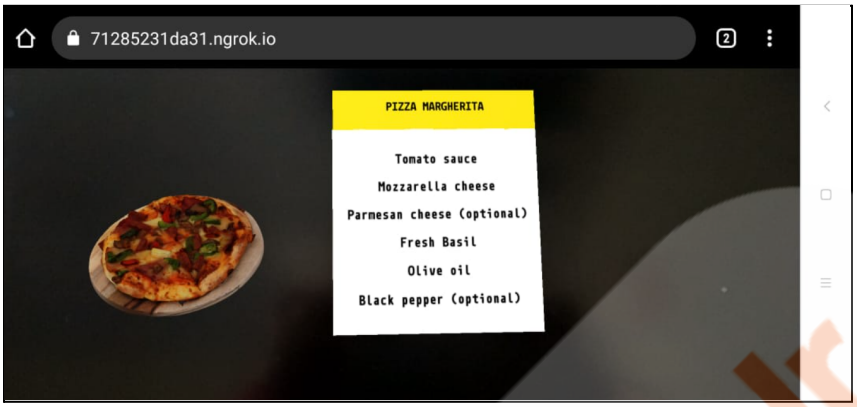
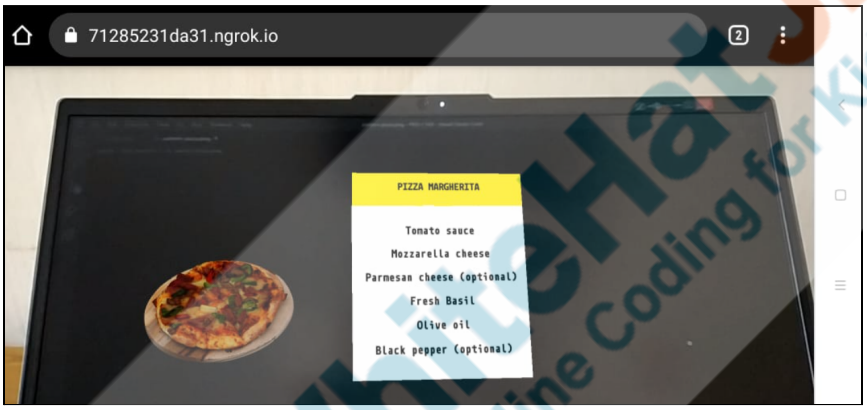
	Now let's test the output: To see the output: • Use **ngrok** to run the application. • Open **HTTPS URL** in your smartphone/laptop and give permission to use the camera. • Open the **marker image** that was downloaded from the tool and point the camera towards it.	



Note: Make the rotation mode is off while using the phone in landscape mode.

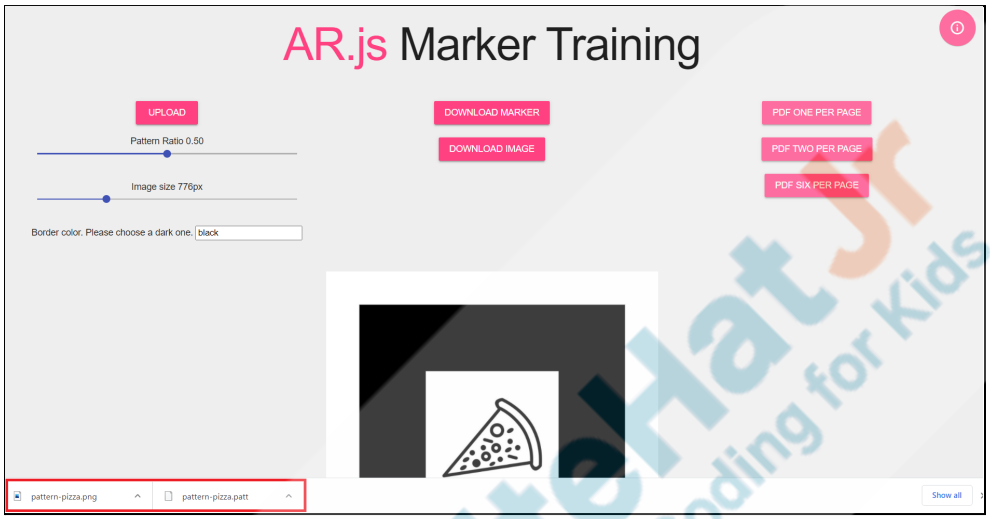




 		
	That was very interesting! Now you will also create the pattern marker and display content over that. You will also add two buttons to allow the user to place orders and give ratings. Are you excited?	**ESR**: Yes!
Teacher Stops Screen Share		
	Now it's your turn. Please share your screen with me.	

Teacher Starts Slideshow  **Slide 13 to 15** Refer to speaker notes and follow the instructions on each slide.		
We have one more class challenge for you. Can you solve it? Let's try. I will guide you through it.		
Teacher Ends Slideshow 		
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.** • **Teacher gets into fullscreen.**		
ACTIVITY • **Create a pattern marker based A-Frame Web AR scene.** • **Render objects in the AR scene using pattern markers.** • **Add buttons elements and use Bootstrap for styling**		
Step 3: Student-Led Activity (20 mins)	*The teacher guides the student to clone the code from Student Activity 1.* ***[Student Activity 1]*** ***Note**: The student will repeat some of the activity performed by the teacher.*	
	How should we start making the AR scene?	**ESR**: First we need to create the pattern marker.

Guide the student to create the pattern marker files and add those in the working directory.



*Guide the student to add the **model and text content** in the <a-marker> entity.*

```
<!--Pattern Marker-->
<a-marker id="pizza-marker" type="pattern" url="assets/dish-markers/pattern-pizza.patt" cursor="rayOrigin: mouse">

  <a-entity id="pizza-3D-model" position="-2 0 0" scale="0.05 0.05 0.05"
    gltf-model="#pizza-model"
    gesture-handler>
  </a-entity>


  <!-- Ingredients -->
  <a-plane position="0 0 0" width="1.5" height="1.5" rotation="-90 0 0">
    <a-plane position="0 0.89 0.02" width="1.49" height="0.3" rotation="0 0 0" color="#F0C30F">
      <a-entity position="0 0 0.1" rotation="0 0 0"
        text="font:monoid;value:PIZZA MARGHERITA;color:black;width: 1.8;height:1;align:center;"></a-entity>
    </a-plane>

    <a-entity position="0 0 0.1" rotation="0 0 0"
      text="color: black; align: center; width: 2; font: monoid;
            value: Tomato sauce\n\nMozzarella cheese\n\nParmesan cheese (optional)\n\nFresh Basil\n\nOlive oil\n\nBlack pepper (optional)">
    </a-entity>
  </a-plane>

</a-marker>
```

Now let's add buttons to allow the user to order and rate the project.

We can add the tag and write a component to add the button elements.

Guide the student to add the element.

```
<!-- Button Main Div -->
<div id="button-div"></div>
```

*Guide the student to register the"**create-buttons**" component and attach the component to the <a-scene> entity.*

***Note**: Add the src to the index.html.*

```
AFRAME.registerComponent("create-buttons", {
  init: function() {
    // 1. Create the button
    var button1 = document.createElement("button");
    button1.innerHTML = "RATE US";
    button1.setAttribute("id", "rating-button");
    button1.setAttribute("class", "btn btn-warning");

    // 2. Create the button
    var button2 = document.createElement("button");
    button2.innerHTML = "ORDER NOW";
    button2.setAttribute("id", "order-button");
    button2.setAttribute("class", "btn btn-warning");

    // 2. Append button elements
    var buttonDiv = document.getElementById("button-div");
    buttonDiv.appendChild(button1);
    buttonDiv.appendChild(button2);
  }
});
```

```
<a-scene
  vr-mode-ui="enabled: false"
  embedded
  arjs="sourceType: webcam;
  sourceWidth:1280; sourceHeight:960;
  displayWidth: 1280; displayHeight: 960;
  debugUIEnabled: false;"
  gesture-detector
  create-buttons>
```

```
<!--JS Files-->
<script src="./js/addButtons.js"></script>
```

Guide the student to create a .css file and add the styling to the button element.

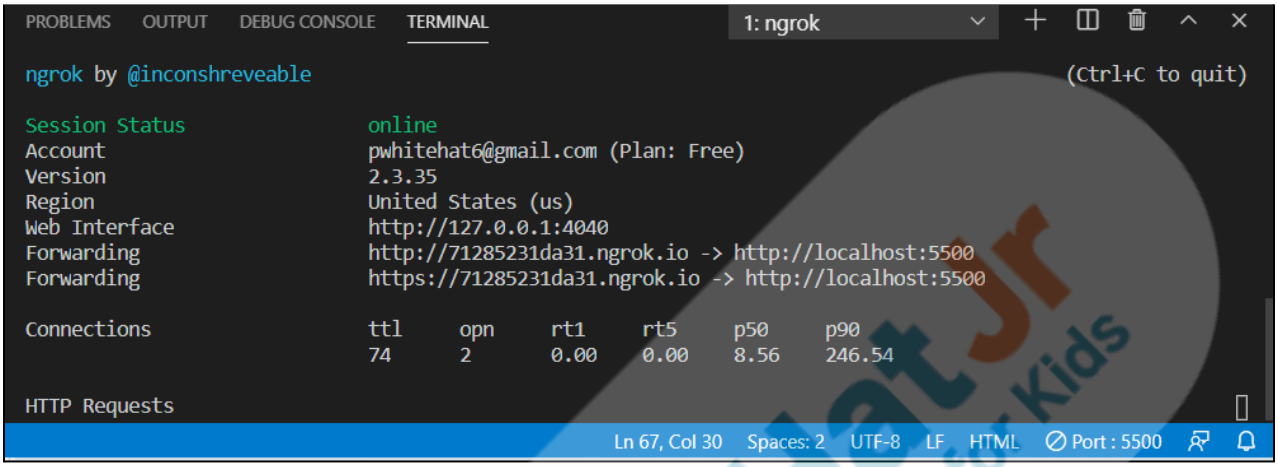
***Note 1**: We are using the Bootstrap CSS library to style the buttons.*

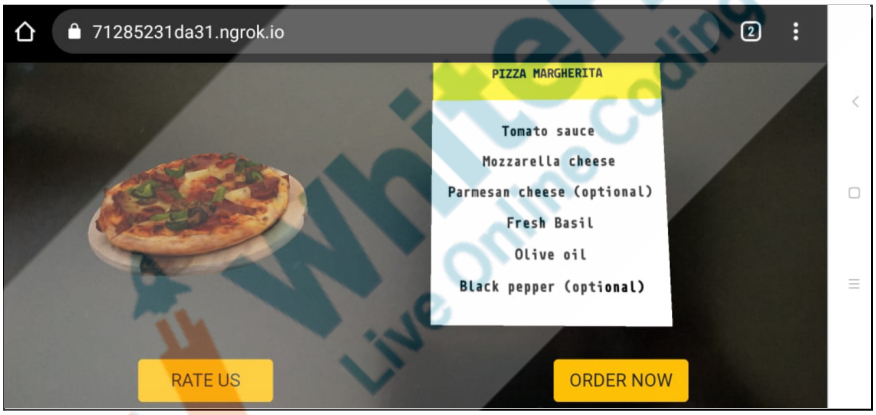
***Note 2**: Add the src to the index.html*

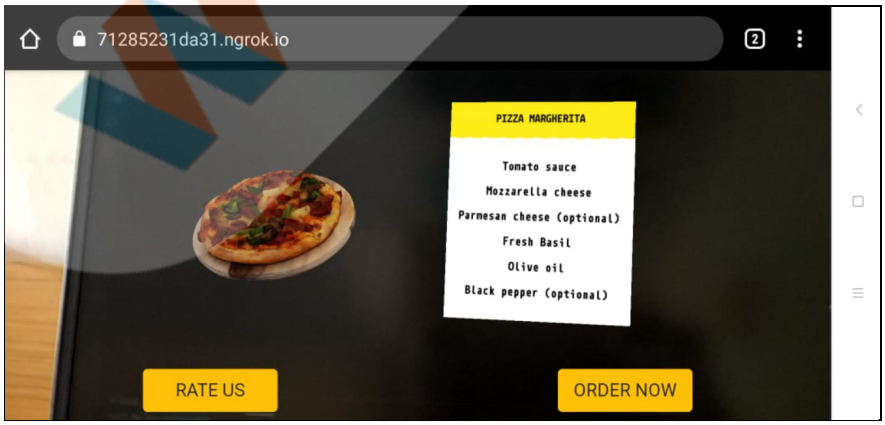
```
button {
  width:120px;
}

#button-div {
  display: flex;
  align-items: center;
  justify-content: space-around;
  position: fixed;
  bottom: 10px;
  width:100%;
  z-index: 1;
  border: 2px soild black;
}
```

```
<link href="./style.css" rel="stylesheet" />
```

Guide the student to run and test the application using the https ngrok URL.







Teacher Guides Student to Stop Screen Share

• **WRAP UP SESSION - 5 mins**
Teacher Starts Slideshow **Slide 16 to 20**
Activity details **Following are the WRAP-UP session deliverables:** • Appreciate the student. • Revise the current class activities. • Discuss the quizzes.
WRAP-UP QUIZ Click on In-Class Quiz
Continue WRAP-UP Session **Slide 21 to 26**
Activity Details **Following are the session deliverables:** • Explain the facts and trivia • Next class challenge • Project for the day • Additional Activity (Optional)
FEEDBACK • **Compliment the student for her/his effort in the class.** • **Encourage the student to think and come up with their own solutions.**

	You get a “hats-off”. Alright. See you in the next class.	*Make sure you have given at least 2 Hats Off during the class for:* Creatively Solved Activities +10

		 
PROJECT OVERVIEW DISCUSSION Refer the document below in Activity Links Sections		
Teacher Clicks 		
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 their reflections in a reflection journal.*

Activity	Activity Name	Links
Teacher Activity 1	Boilerplate Code	https://github.com/whitehatjr/PRO-C168-Boilerplate

Teacher Activity 2	Teacher Reference Code	https://github.com/whitehatjr/PRO-C168
Teacher Activity 3	Output Reference	https://curriculum.whitehatjr.com/PRO+Asset/PRO+168+Output+Ref+(1).mp4
Teacher Activity 4	AR.js Marker Training Link	https://jeromeetienne.github.io/AR.js/three.js/examples/marker-training/examples/generator.html
Student Activity 1	Boilerplate Code	https://github.com/whitehatjr/PRO-C168-Boilerplate
Teacher Reference 1	Ngrok Updates	https://docs.google.com/document/d/1dlMry188llEJl6rHEc3AkBashQSOwGQ40HQft29S8vQ/edit?usp=sharing
Teacher Reference 2	Project Document	https://s3-whjr-curriculum-uploads.whjr.online/6367b51c-79e9-4ca7-a390-066e4f23287f.pdf
Teacher Reference 3	Project Solution	https://github.com/whitehatjr/PRO-C168-AR
Teacher Reference 4	Visual-Aid	https://s3-whjr-curriculum-uploads.whjr.online/aa5b2ea3-9738-4f36-9882-c235794bf87a.html
Teacher Reference 5	In-Class Quiz	https://s3-whjr-curriculum-uploads.whjr.online/8ee6dee0-5c41-4718-bd84-3d7f0416e81d.pdf