Topic	SKETCH IDENTIFICATION	
Class Description	**Kids venture into the sea of pretrained neural networks and study them for their possible applications of interest. Learning the neural network methods builds their confidence to sift through complex code and apply only that which is needed to solve a problem. Handling complexity is essential in AI.**	
Class	**ADV-C117**	
Class Time	**55 mins**	
Goals 	• Start learning a new neural network Doodlenet. • Start building sketch Identification WebApp by creating the web page structure.	

Resources Required	• Teacher Resources:
	  ○ Laptop/PC with Webcam
	  ○ Use Gmail login credentials.
	  ○ Earphone with mic
	  ○ Notepad and Pen
	• Student Resources:
	  ○ Laptop/PC with Webcam
	  ○ Use Gmail login credentials.
	  ○ Earphone with mic (optional)
	  ○ Notepad and Pen	

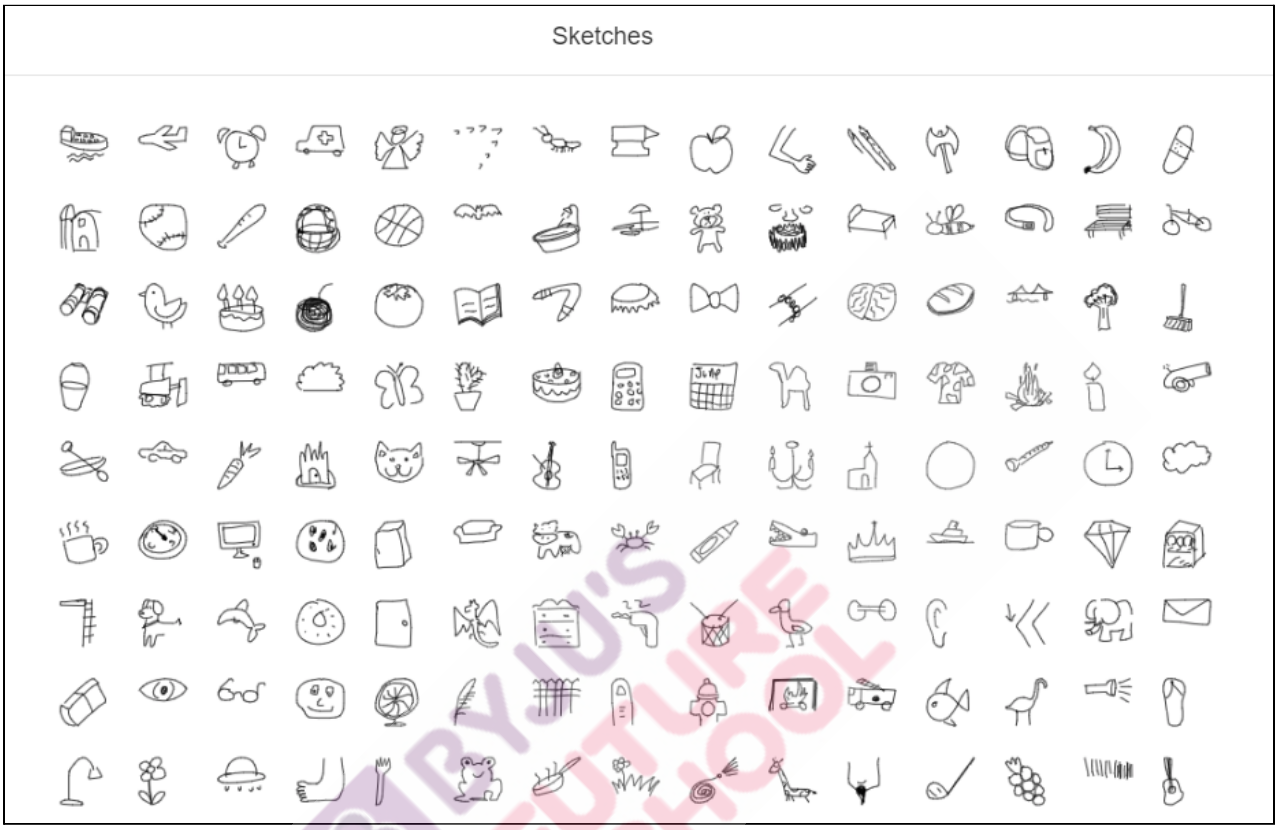
Class Structure	**Warm Up** **Teacher-Led Activity** **Student-Led Activity** **Wrap Up** **Project Pointers and Cues**	**5 Mins** **10 Mins** **30 Mins** **5 Mins** **5 Mins**

WARM UP SESSION - 5 mins	
Teacher starts slideshow  **from slides 1 to 21.** Refer to speaker notes and follow the instructions on each slide.	
TEACHER ACTIVITY  **- 10 mins**	
Teacher Initiates Screen Share	
Say	**Do**
	Remember, you don't have to explain the whole HTML and CSS code. Only explain concepts, as per the document. **Also don't explain the JS code. This should be done in the upcoming classes.** **Please follow the flow for the class -** 1. Download the file from Teacher-Activity-2. 2. First, demo the output to the students. 3. Then, explain the HTML code from Teacher-Activity-2. 4. Then, let the student do the HTML coding.

	5. Then, if you have time, do the Additional Activities.

Concept & Understanding

In today's class, we are going to use a sketch identification neural network model, which is called the Doodlenet neural network model.
Do you know what a doodle is?
So, basically, the doodle is a scribbled representation, or a design, or simply we can say that it is a sketch of anything or an object. For example: A sketch of an alarm clock, a cat, a dog, a bottle, etc.



Now, let's see what a doodle net is and how we are going to use this neural network model?

So, Doodlenet is a pre-trained neural network model developed by Google, which classifies and labels sketches that are hand-drawn by users. Now, these sketches are classified in 345 categories out of which each category has 50 million doodles. So, we can say how sophisticated, yet efficient and easy to use this neural network model can be.

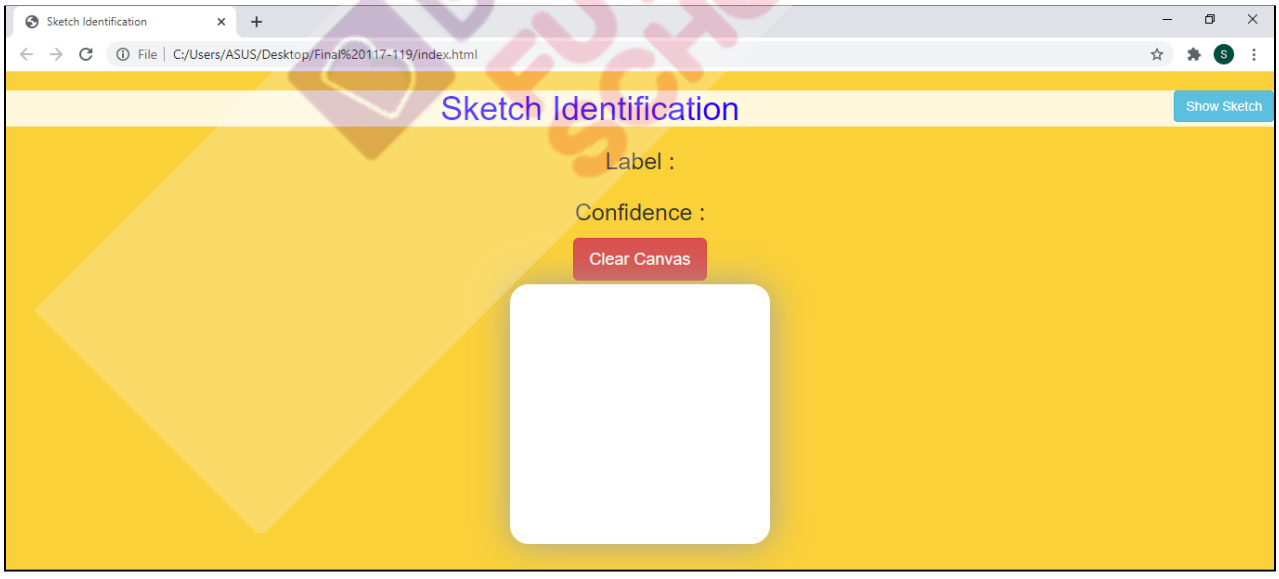
So, we will be using this neural network model to build a web app where we will draw a sketch on canvas and pass it to this neural network model, and this neural network model will recognise this sketch by comparing the sketch with already-present doodles in the neural network model and will return what exactly the sketch is all about - its name and the highest accuracy.

So, let's have a quick demo on how our web app will work.

Getting Started: DEMO

Run [Teacher-Activity-1.]

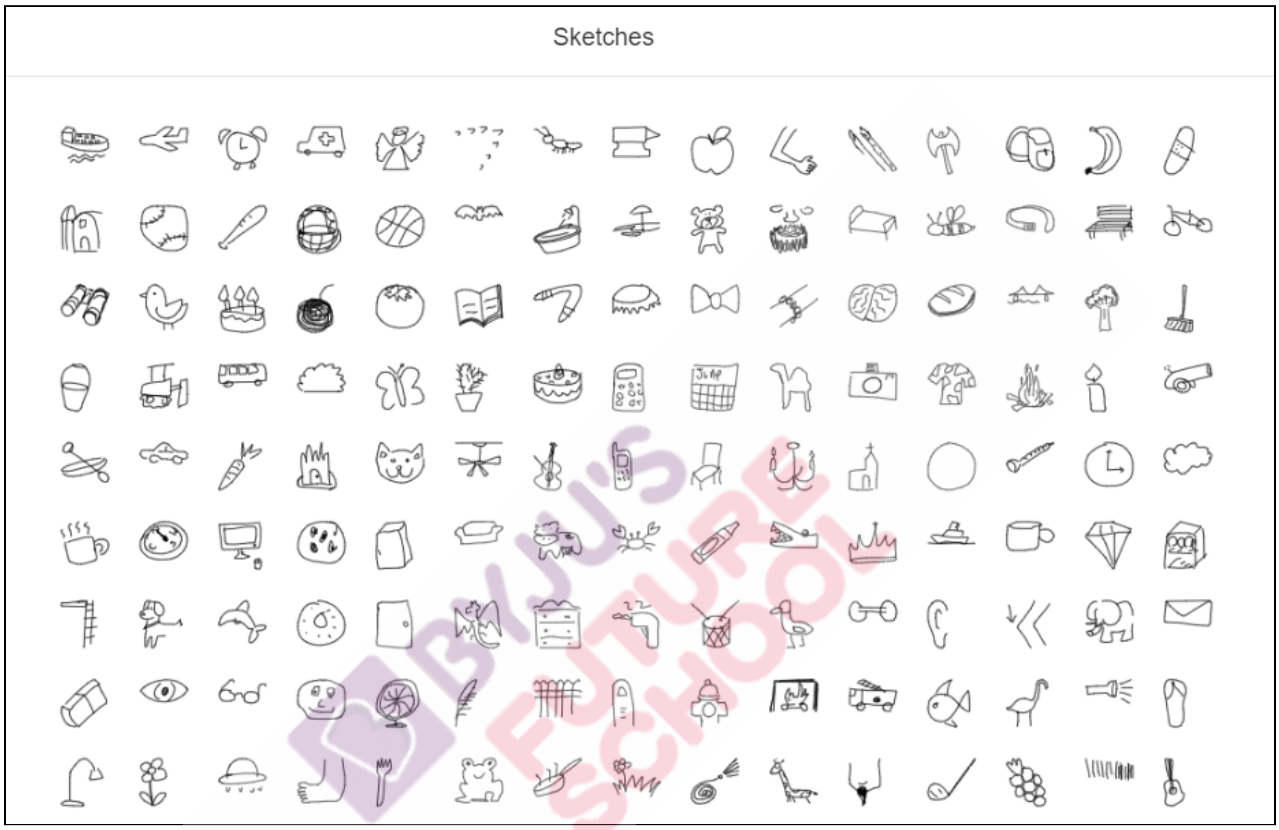
1. This is the UI of the application.



2. If you click on the **Show Sketch** button 

, you will find that there are

many sketch images, and on these sketch images, the Doodlenet neural network model has been trained by Google.

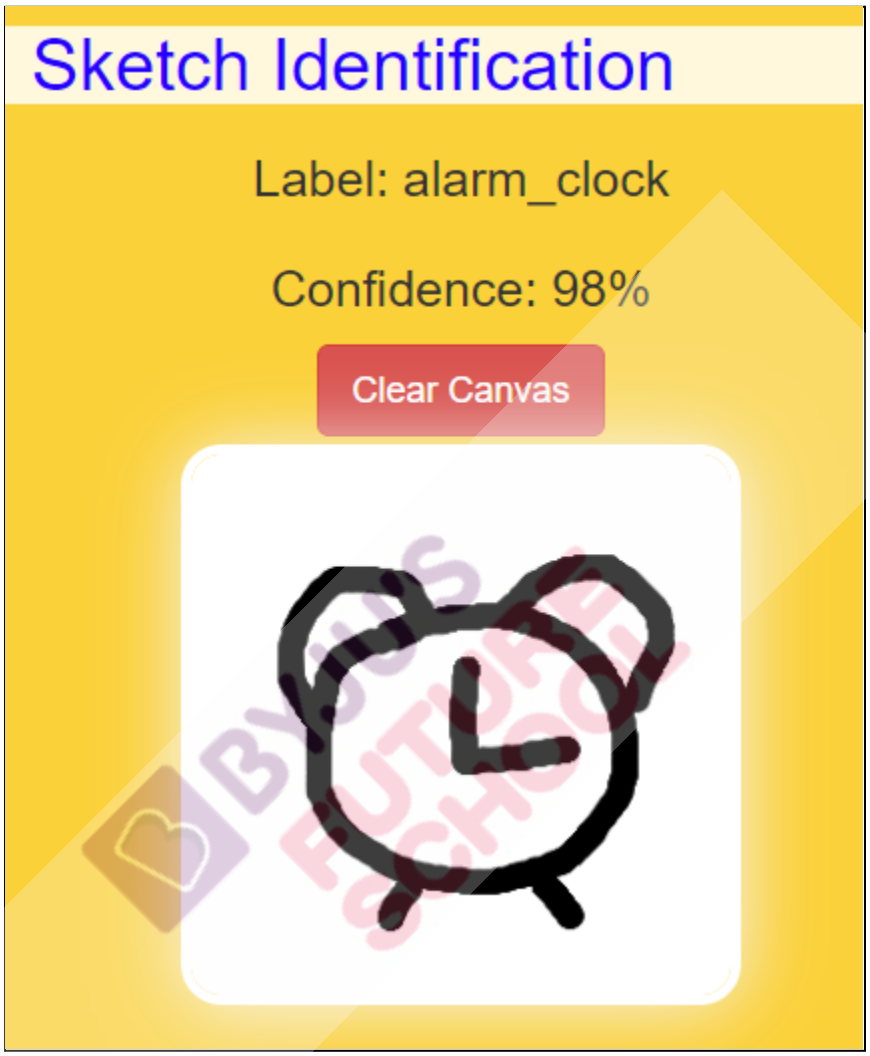


So, this pop-up should open.

Q- Do you remember what it is called?
A- Yes, it is a bootstrap modal which we have used many times in previous classes.

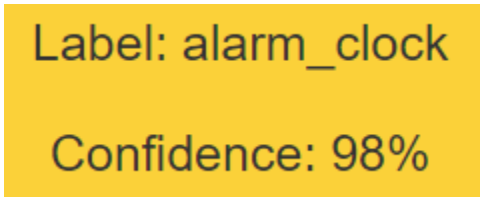
NOTE - While you are demoing the sketch identification app by drawing sketches on the canvas, try replicating some of these mentioned sketches and show the results to the student.

3. On the canvas, draw any sketch, like this:

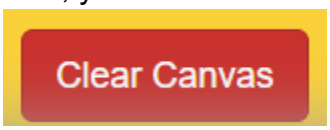


The below output shows what sketch it is and gives the confidence which is the accuracy on how much percentage the neural network model has identified the sketch.

Also, as we have used the speech synthesiser, it will keep on converting the text of the label which holds the name of the object drawn on the canvas, into the speech. Hence, the system will call out the name of the object we will be drawing on the canvas.

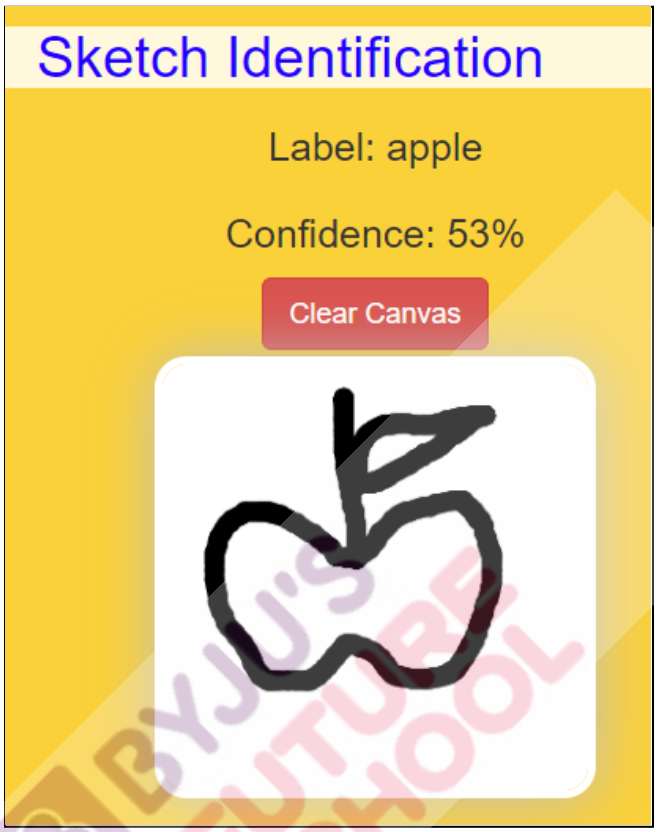


So, the neural network model detects the above drawn sketch as an alarm_clock and gives its confidence as 98%. This means that the model is 98% sure that it is an alarm clock. And we can see that it is correct.

4. Now, you can sketch another doodle. For that, you need to click on the 

button first, to clear the canvas.

5. Now, use another doodle from the doodles mentioned on the website and draw it on the canvas. As soon as you start drawing a different object, the name of the object drawn will appear automatically and the system will call out the name of the object drawn on the canvas.



Note: Don't expect the result to be very accurate every time. Sometimes, an approximate value of the object will be displayed.

6. Now, do it again using a different object which is mentioned on the Bootstrap modal.

So you saw that the website is identifying the sketch in real-time. So, we can say that this is a real-time sketch identified by a web app.
This advanced version of sketch identification is good enough to be built in today's class.

Code & Explanation

So, we'll build this web app in parts. In today's class, we will start with the HTML and CSS code.

NOTE FOR TEACHERS -
DO NOT explain the whole HTML and CSS code of the web app in this class. Just explain the HTML code as per the document.

You have to download the **sketch-Identification** folder from **Student-Activity-2.** The folder has:

- An **index.html** file - This file has some prewritten HTML code and the student has to complete it.
- A **stye.css** file - This file is empty and the student has to complete it.
- A **main.js** file - This file is empty and the student has to complete it in the upcoming classes.
- 2 images which need to be used as sketches.

HTML CODE in **index.html:**

```
<html>

<head>
  <meta charset="UTF-8">
  <title>Sketch Identification</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

  <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.9.0/p5.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.9.0/addons/p5.dom.min.js"></script>

  <script src="https://unpkg.com/ml5@latest/dist/ml5.min.js" type="text/javascript"></script>


  <link rel="stylesheet" type="text/css" href="style.css">
</head>

<body>
  <script src="main.js"></script>
</body>

</html>
```

Bootstrap Links

Imported ml5.js link

Imported p5.js link

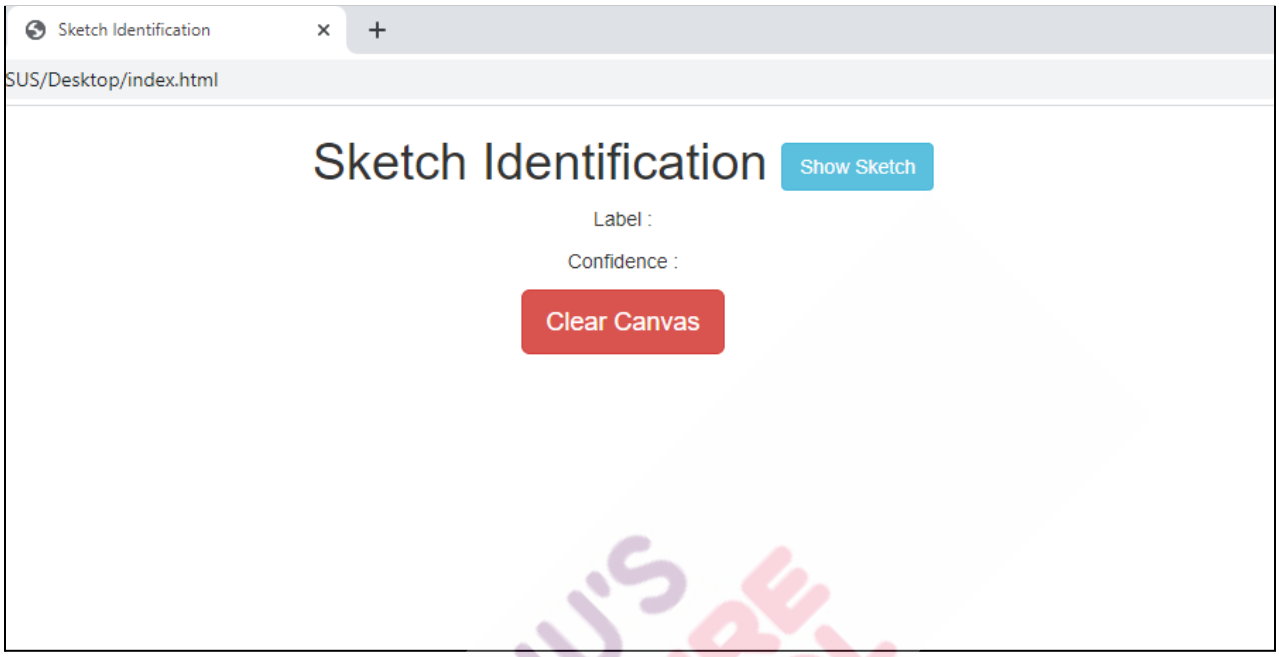
Our style file link

Our JS file link

The above HTML code has:

- **Bootstrap** links
- **p5.js** link
- **ml5.js** link
- Our **style.css** file link
- Our **main.js** file link

Now, let's start adding the HTML elements in **index.html**.
After adding the code in the **index.html** file, our output will look like this:



Complete HTML Code:

```
<html>

<head>
  <meta charset="UTF-8">
  <title>Sketch Identification</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

  <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.9.0/p5.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.9.0/addons/p5.dom.min.js"></script>

  <script src="https://unpkg.com/ml5@latest/dist/ml5.min.js" type="text/javascript"></script>



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

```
<body>
  <center>
<!-- Modal -->
<div id="myModal" class="modal fade">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal">&times;</button>
        <h4 class="modal-title">Sketches</h4>
      </div>

      <div class="modal-body">
        <img src="sketch1.png">
        <img src="sketch2.png">
      </div>

    </div>
</div>

<h1>Sketch Identification
<button id="show_sketch" class="btn btn-info" data-toggle="modal" data-target="#myModal">Show Sketch</button>
  </h1>
  <p id="label">Label : </p>
  <p id="confidence">Confidence : </p>
  <button class="btn btn-danger btn-lg" onclick="clearCanvas()">Clear Canvas</button>
</center>

<script src="main.js"></script>
</body>

</html>
```

1. First, we will add the doodle reference images to our application, for referring the sketches to be drawn and identified which are also pretrained. For this, inside the **body** tag, we will add the **center** tag `<center>`, and after this, create a modal to show all doodles of the doodlenet model. As you know that a modal is a style container which looks like a window on the webpage.

2. Then, we will add a **div** tag. This div will be the main div that will be holding the bootstrap modal (pop-up). And for this, we will be using many bootstrap classes.
   The advantage of using bootstrap classes is that we just have to write the class name and the style for it and it will be included automatically.
   For this, create a tag, and inside it, add **id** for the modal and give the class name as **modal fade**, like this:

```
<body>
  <center>
<!-- Modal -->
<div id="myModal" class="modal fade">
```

- ➔ Define the div tag and give **id** as "**myModal**".
- ➔ Then, give a bootstrap class **modal** - It helps to make the background opacity low, when the modal is open.
- ➔ Also, give a bootstrap class "**fade**" - The fade class adds animation effects when this modal is opened and closed.

3. Then, again add the tag, and inside it, add a class as **modal-content**.

```
<body>
  <center>
<!-- Modal -->
<div id="myModal" class="modal fade">
    <div class="modal-content">
```

The bootstrap class **modal-content** adds white as the background color, black as the font color, a box-shadow, and a border-radius to the modal (pop-up) so that the modal looks good. This **div** will hold all the content of the modal.

4. Now, add one more div. This div tag will be used to hold the **head** part of the modal and give a bootstrap class **modal-header**. This class adds padding to the header of the modal, so that header of the modal looks good.

```
<body>
  <center>
<!-- Modal -->
<div id="myModal" class="modal fade">
    <div class="modal-content">
      <div class="modal-header">
```

5. Then, inside this modal-header tag, we will create a button, which will close the modal (pop-up). For this:

   ➔ Inside this div tar, first add a cross button (), that can be used to close this modal.

   ➔ Define the button tag and give class "**close**". This will make the style of the default button from this -  to this -  .

   ➔ Then, add **data-dismiss="modal"**. This will add the functionality of closing the modal when this cross button is pressed.

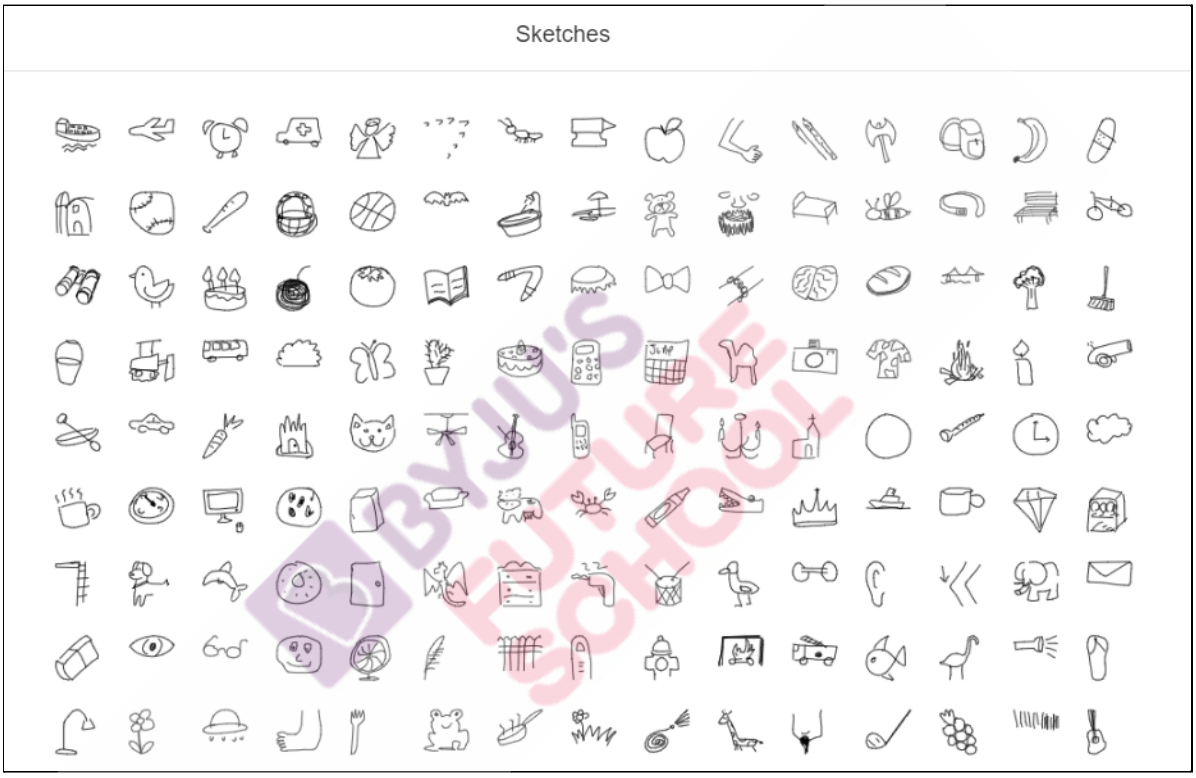
   ➔ To get the bootstrap cross icon  , write **×** as button text.

```
<body>
  <center>
<!-- Modal -->
<div id="myModal" class="modal fade">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal">&times;</button>
```

6. Then, we will write **Sketches** as the title to this modal (pop-up). Like this:

```
<h4 class="modal-title">Sketches</h4>
```

The output will look like this:




```
<body>
  <center>
<!-- Modal -->
<div id="myModal" class="modal fade">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal">&times;</button>
        <h4 class="modal-title">Sketches</h4>
      </div>
```

7. Then, add one more **div** tag. This div will be used to hold the **body** part of the modal and give a bootstrap class **modal-body**. This class adds padding to the body of the modal, so that the body of the modal looks good.
   The body of the modal will hold two images of sketches from the doodle neural network model.
   So, for this, add the **.png** images which contain the doodles for the objects, like this:

```
<div class="modal-body">
  <img src="sketch1.png">
  <img src="sketch2.png">
</div>
```

   Code till now:

```
<body>
  <center>
<!-- Modal -->
<div id="myModal" class="modal fade">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal">&times;</button>
        <h4 class="modal-title">Sketches</h4>
      </div>

      <div class="modal-body">
        <img src="sketch1.png">
        <img src="sketch2.png">
      </div>

    </div>
</div>
```

8. Then, we will add the heading to our web page which is:

Sketch Identification

For this, write **Sketch Identification** in <h1> tag, like this:

```
<h1>Sketch Identification
```

Now, we will not close the <h1> tag as we need to add a button (that opens the sketch doodle images) in this header.

```
<h1>Sketch Identification
<button id="show_sketch" class="btn btn-info" data-toggle="modal" data-target="#myModal">Show Sketch</button>
  </h1>
```

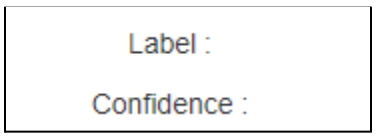
This should show the output as below:

9. Then, we will add the paragraph tags, and inside it, we will store the labels and confidence.
    - **Label** - This is the identified sketch name which is drawn on the canvas.
    - **Confidence** - This shows the accuracy in percentage of the identified sketch.

    So inside the <p> tag, write **Label** and **Confidence** as the ids to add the text (Label and Confidence) to appear on the web page, like this:

```
<h1>Sketch Identification
<button id="show_sketch" class="btn btn-info" data-toggle="modal" data-target="#myModal">Show Sketch</button>
  </h1>
  <p id="label">Label : </p>
  <p id="confidence">Confidence : </p>
```

This will show the output as below when any sketch is identified:



10. Then, we will create a button. The functionality of this button is to clear the canvas output so that another sketch can be drawn on it.
So, to create the button, use the **button** tag and assign bootstrap classes to this button tag. This button is from **btn btn-danger btn-lg**, which fills the button with red color.

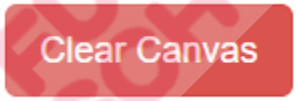
Then, add an **onclick** event and call the function "**clearCanvas()**".

```
<h1>Sketch Identification
<button id="show_sketch" class="btn btn-info" data-toggle="modal" data-target="#myModal">Show Sketch</button>
  </h1>
  <p id="label">Label : </p>
  <p id="confidence">Confidence : </p>
  <button class="btn btn-danger btn-lg" onclick="clearCanvas()">Clear Canvas</button>
</center>
```

```
<button class="btn btn-danger btn-lg" onclick="clearCanvas()">Clear Canvas</button>
```

We will define the **clearCanvas()** function in the upcoming classes.

Output of this:



Great!
We will add style (coding CSS) to this in the next class.

Teacher Stops Screen Share

STUDENT ACTIVITY **- 30 mins**

Now it's your turn. Please share your screen with me.

- **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 Fullscreen.**

Student Starts Screen Share	
Say	**Do**
Download the predefined code from the **Student-Activity-1-** and start coding in this file.	**Student-Activity-1- PREDEFINED CODE**
Start HTML code for **index.html** file. Write the HTML code to design the web page for Sketch Identification: - Add the heading tags. - Add the images using img tag. - Add the button elements to show the sketches. - Add the onclick events to clear canvas.	**Student-Activity-2- CODE DIAGRAM**
We will code the **style.css**, **main.js** in the upcoming classes.	The student has to download the **Sketch_ Identification** folder from **Student-Activity-1.** The folder has: • An **index.html** file - This file has some prewritten HTML code and the student has to complete it. • A **stye.css** file - This file is empty and the student has to complete it. • A **main.js** file - This file is empty and the student has to complete it in the upcoming classes. • 2 images which need to be used as sketches.

Keep the code files safe as we will be uploading these files on github when we complete building this website.	**If time permits, encourage the student to design freely.**

Complete Code

```
<html>

<head>
  <meta charset="UTF-8">
  <title>Sketch Identification</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

  <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.9.0/p5.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.9.0/addons/p5.dom.min.js"></script>

  <script src="https://unpkg.com/ml5@latest/dist/ml5.min.js" type="text/javascript"></script>



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

```
<body>
  <center>
<!-- Modal -->
<div id="myModal" class="modal fade">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal">&times;</button>
        <h4 class="modal-title">Sketches</h4>
      </div>

      <div class="modal-body">
        <img src="sketch1.png">
        <img src="sketch2.png">
      </div>

    </div>
</div>

<h1>Sketch Identification
<button id="show_sketch" class="btn btn-info" data-toggle="modal" data-target="#myModal">Show Sketch</button>
  </h1>
  <p id="label">Label : </p>
  <p id="confidence">Confidence : </p>
  <button class="btn btn-danger btn-lg" onclick="clearCanvas()">Clear Canvas</button>
</center>

<script src="main.js"></script>
</body>

</html>
```

Teacher Guides Student to Stop Screen Share

Teacher Initiates Screen Share

WRAP UP SESSION - 5 mins

Teacher starts slideshow  **from slides 22 to 24.**
Refer to speaker notes and follow the instructions on each slide.

PROJECT POINTERS AND CUES - 5 mins

Teacher starts slideshow  at slide 25. Refer to speaker notes and follow the instructions on each slide.	
Say	**Do**
## QUICK DRAW WEBAPP - 1 **Goal of the Project:** Today, you have started making a Sketch Identification web app. You have also started designing the web application. For this project, you have to make a quick-draw application. And in this project, you have to use the Doodlenet model which is the same one that you have used in the Class. For this project, you have to design the web application and complete the HTML and CSS as per the instructions. **Story:** SWork is launching a drawing app for kids under the age group of 5-8, and now they want an application, which will be a doodle game, in which they want you to create an application, which will allow the kids to draw a specific sketch which is being instructed by the app. And also, this game app will be based on time constraints and this will give the score for each right sketch drawn by the kid.	

So your task is to create a beautiful web page and also add some JavaScript functionalities to it. So let's start by designing the web app. Are you ready for the upcoming Capstone class? As always, we expect you to strive harder and push your creative limits. In the capstone class, we will continue to enhance our Real-Time Image Identification web application. Our focus will be to work on its HTML and CSS. Good Luck!	
ADDITIONAL ACTIVITY	
Teacher Initiates Screen Share	
Teacher starts slideshow  **from Slide 26 to 30.**	
STUDENT ADDITIONAL ACTIVITY 	
Student Initiates Screen Share	
NOTE FOR TEACHERS - **For the solution of all the Additional Activities, open Teacher-Activity-4 and navigate to class number C117.** **Additional Activity 1 -** **Run Student-Activity-3- from the panel.**	

The **TASK** and **HINTS** are mentioned on the website itself.

Additional Activity 2 -
Run **Student-Activity-4** from the **panel.**
The **TASK** and **HINTS** are mentioned on the website itself.

Additional Activity 3 -
Run **Student-Activity-5** from the **panel.**
The **TASK** and **HINTS** are mentioned on the website itself.

Additional Activity 4 -
Run **Student-Activity-6** from the **panel.**
The **TASK** and **HINTS** are mentioned on the website itself.

Additional Activity 5 -
Run **Student-Activity-7** from the **panel.**
The **TASK** and **HINTS** are mentioned on the website itself.

Teacher ends slideshow  **at slide 31.**
Refer to speaker notes and follow the instructions on each slide.

Teacher Stops Screen Share

Teacher Clicks ✖ End Class

Activity	Activity Name	Links
Teacher Activity 1	WEBSITE LINK	https://shravantihable.github.io/Web_Designing/
Teacher Activity 2	COMPLETE CODE	https://curriculum.whitehatjr.com/ADV+Asset/ADV-C117-C119.zip
Teacher Activity 3	CODE DIAGRAM	https://docs.google.com/document/d/e/2PACX-1vTBo9TOMCARweTgciq3ej1V5_j4lR5Vyxmol-G96Iodi4Cw_Nx7gWrmK37rxHG4xZPU7gie5vOcyzuj/pub
Teacher Activity 4	ADDITIONAL ACTIVITIES SOLUTIONS	https://docs.google.com/spreadsheets/d/e/2PACX-1vRehzZcZVoeWCJ-tTN3JhfOJFJTlJ7v5fh7ImWZ7SV4zkvzNoER2R4vk6y7QskuvnC-gxmL7wDREZHY/pubhtml
Student Activity 1	PREDEFINED CODE	https://curriculum.whitehatjr.com/ADV+Asset/Sketch_Identification.zip
Student Activity 2	CODE DIAGRAM	https://docs.google.com/document/d/e/2PACX-1vTBo9TOMCARweTgciq3ej1V5_j4lR5Vyxmol-G96Iodi4Cw_Nx7gWrmK37rxHG4xZPU7gie5vOcyzuj/pub
Student Reference Activity 1	BOOTSTRAP MODAL	https://www.w3schools.com/bootstrap/bootstrap_modal.asp
Project Solution	QUICK DRAW WEBAPP - 1	https://drive.google.com/file/d/1ZWgDhEWFDVI7h2KwbHJoQjTUvCcrsvZw/view?usp=sharing This is the complete output student will create at the end of C119, for know just showcase the design of the webapp
Teacher Reference Visual aid link	Visual aid link	https://s3-whjr-curriculum-uploads.whjr.online/a958d658-49c7-4708-824c-b924d1d09c5b.html

Teacher Reference In-class quiz	In-class quiz	https://s3-whjr-curriculum-uploads.whjr.online/bb25b3d7-219e-4a10-9037-55b51f3d0c5f.pdf