

# **CSCE 5320 Scientific Data Visualization**

## **Activity 5**

### **Introduction to Web Technologies**

#### **Submission Guidelines:**

1. Assignment submission is individual.
2. Take screenshots of the entire screen including the date and time.
3. For Task 1 and Task 2, please Submit the screenshots of SVG picture, webpage, alerts and code screenshots and files (html, CSS, JS) for both tutorial and Questions. (Please mention the code filenames as Tutorial.html and Questions.html similarly mention for CSS and JS files)
4. For Task 3, please submit your Viz hub links for both tutorial 3 and Question 3 and explain your understanding.
5. Explain the understanding of every question wherever it is asked. Missing the explanation will result in deduction of marks.
- 6. You should also provide explanation to the tutorials in detail.**
7. The similarity score for your document should be less than 15%.
8. Submission after the deadline is considered as late submission.

# TASK 1

## Tutorial 1: Introduction to SVG and CSS.

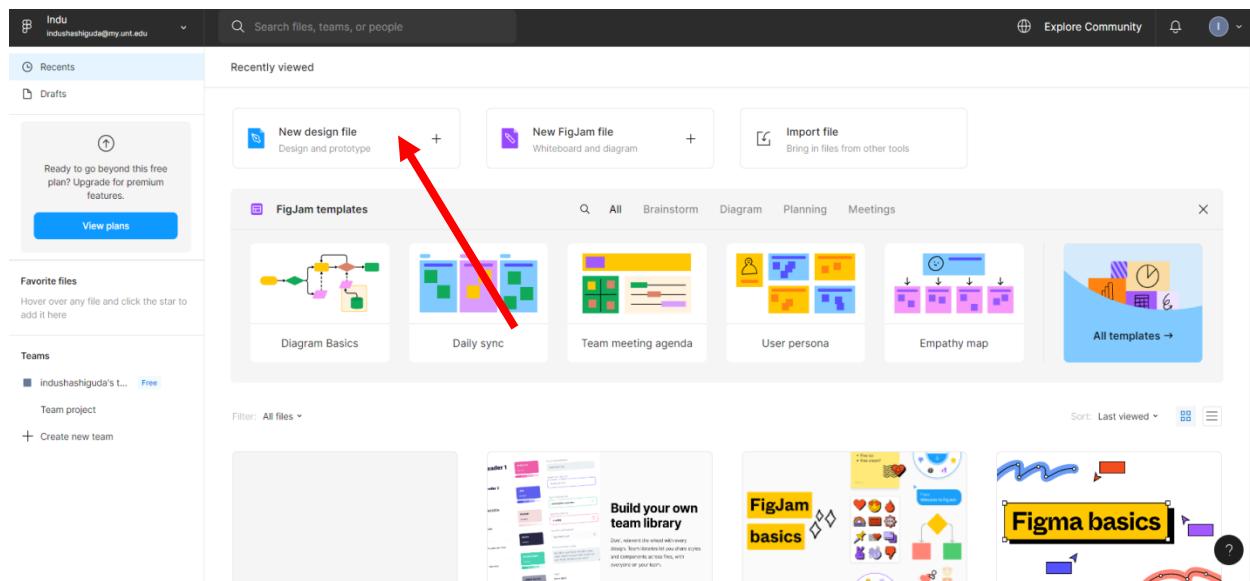
### Figma:

Figma is a collaborative web application for interface design, with additional offline features enabled by desktop applications for macOS and Windows. The feature set of Figma focuses on user interface and user experience design, with an emphasis on real-time collaboration, utilizing a variety of vector graphics editor and prototyping tools.

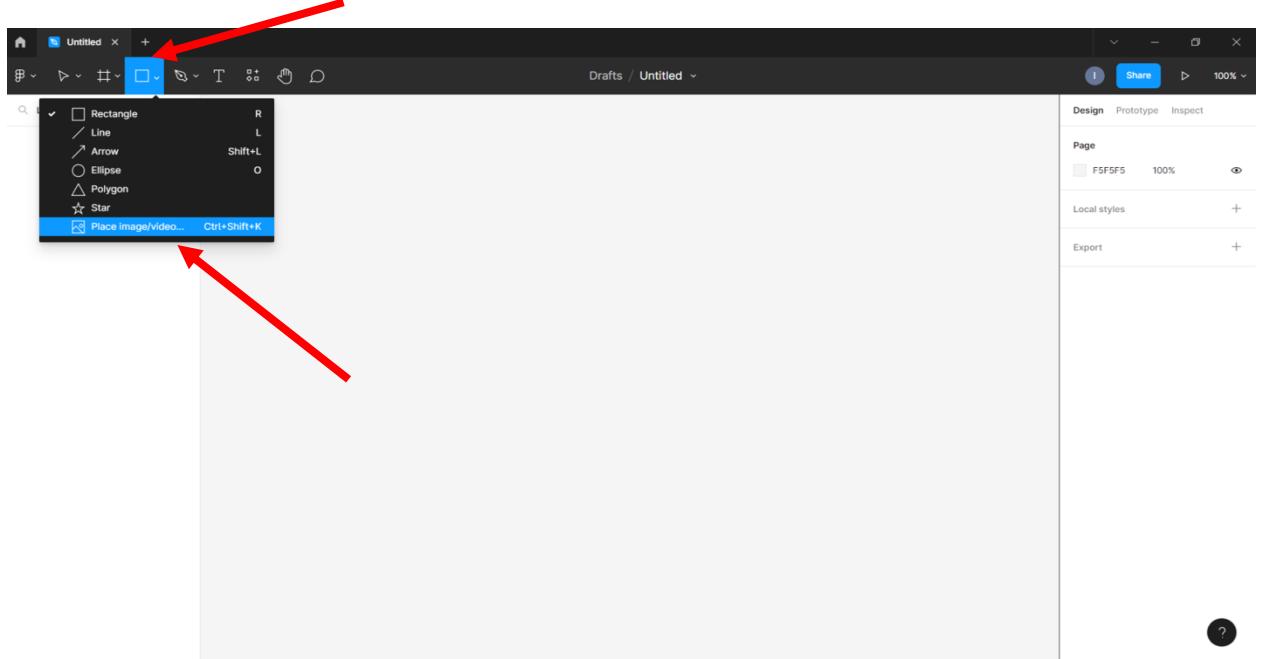
Download Figma from the below given url

<https://www.figma.com/downloads/>

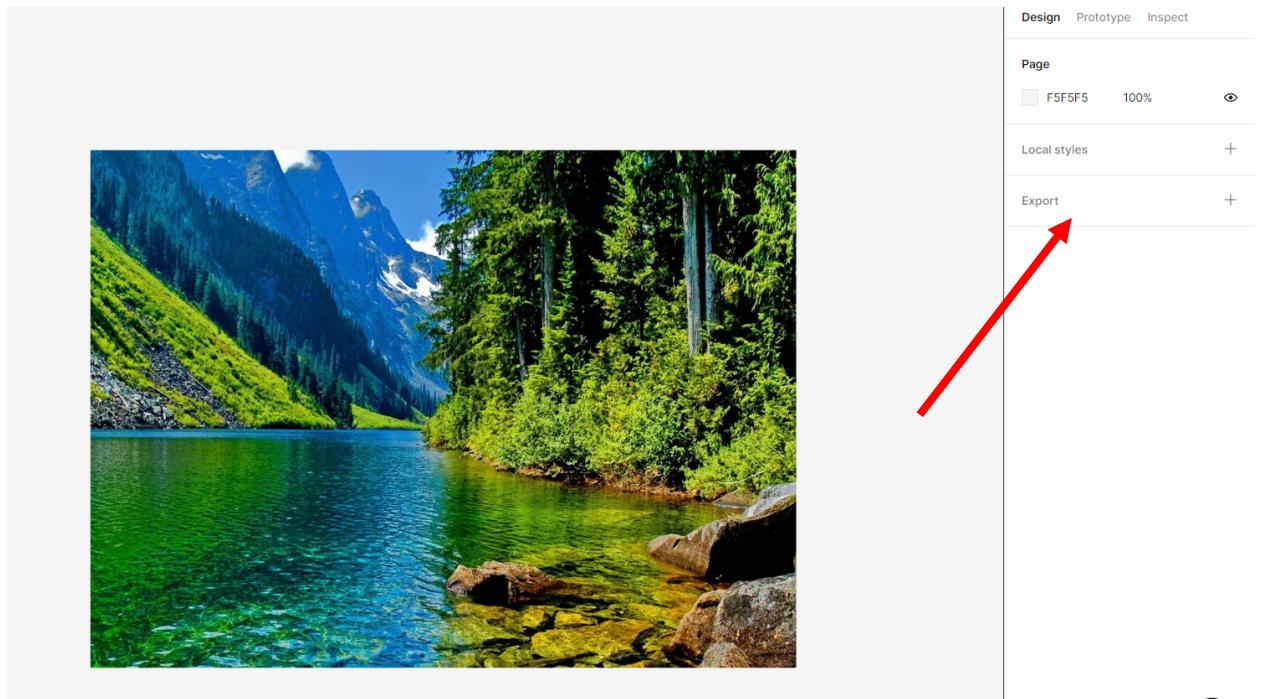
After downloading the Figma, you will get the page as shown below. Click on new Design file. A new page will be opened.



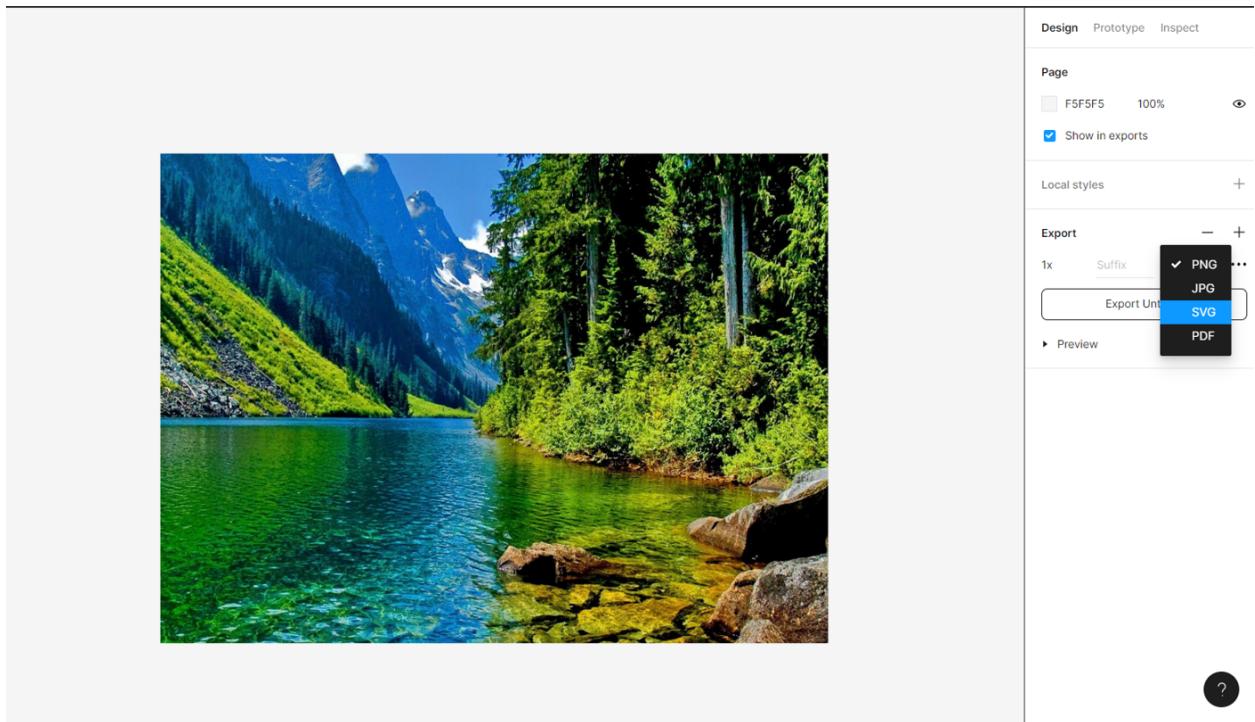
Download the given image(Nature Images.JPG) and click on square symbol which is at the top left corner and select Place image/video.



Place the downloaded image in the workspace and click on export which is on the right side.



After clicking on export, select type as SVG and the image will get exported and an SVG file will be created.

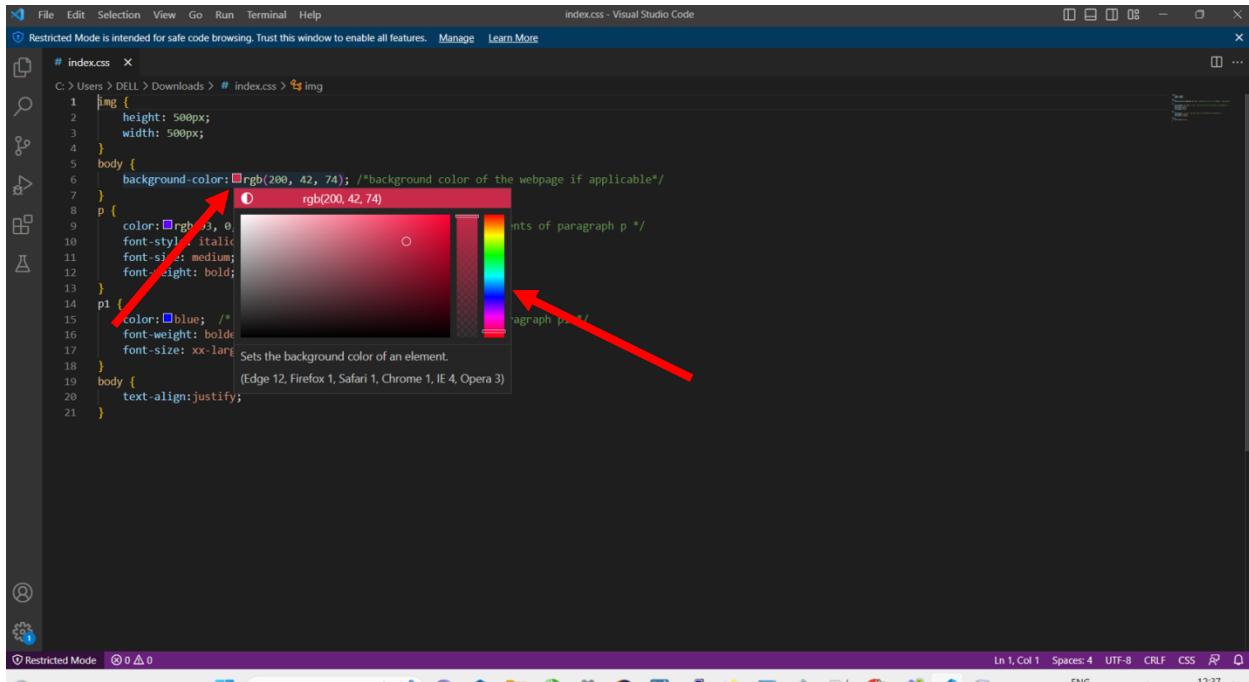


Now we have to create our own webpage by using CSS and html.

## Index.css

```
# index.css  X  ◊ Index.html  JS example.js
C: > Users > DELL > Downloads > # index.css > p1
1  img {
2    height: 500px;
3    width: 500px;
4  }
5  body {
6    background-color: #rgb(200, 42, 163); /*background color of the webpage if applicable*/
7  }
8  p {
9    color: #rgb(55, 0, 255); /* color and font style of the contents of paragraph p */
10   font-style: italic;
11   font-size: medium;
12   font-weight: bold;
13 }
14 p1 {
15   color: #blue; /* color and font style of the contents of paragraph p1 */
16   font-weight: bolder;
17   font-size: xx-large;
18 }
19 body {
20   text-align:justify;
21 }
```

Above is the CSS code, in which we have created the webpage by giving the background color of web page and color and font style of the paragraphs and the alignment of the text.



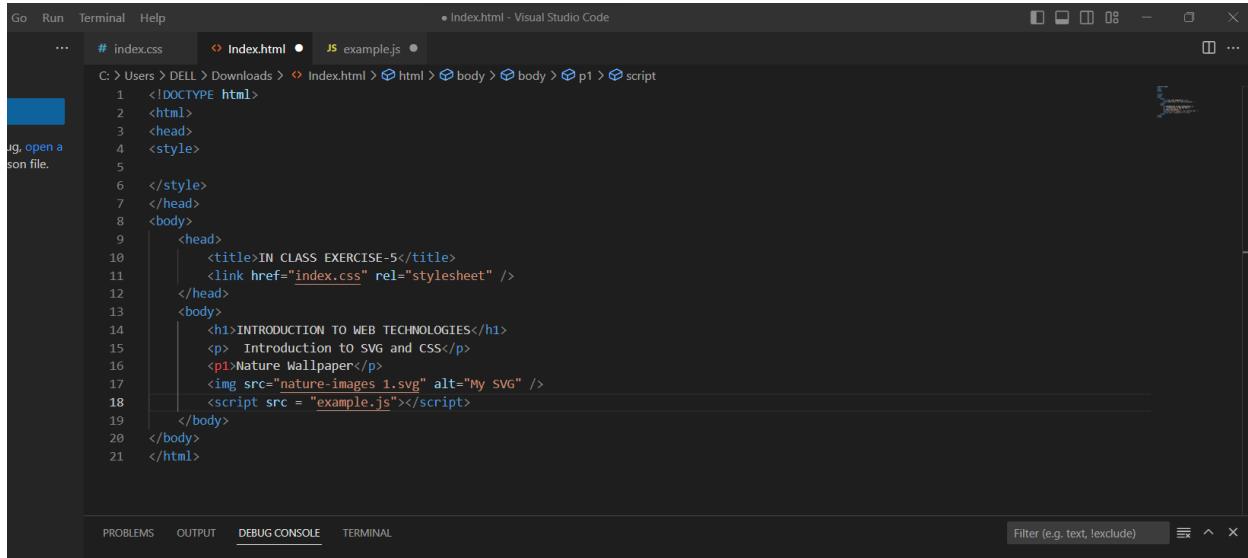
In the above screenshot, you can see by clicking on the square box, we can select different colors and we can change the background color and the text color.

#### Index.css

```
img {  
    height: 500px;  
    width: 500px;  
}  
body {  
    background-color:rgb(200, 42, 163); /*background color of the webpage if applicable*/  
}  
p {  
    color:rgb(55, 0, 255); /* color and font style of the contents of paragraph p */  
    font-style: italic;  
    font-size: medium;  
    font-weight: bold;  
}  
p1 {  
    color:blue; /* color and font style of the contents of paragraph p1 */  
    font-weight: bolder;  
    font-size: xx-large;  
}  
body {  
    text-align:justify;  
}
```

Above is the CSS code, you can keep this code in Visual studio code or in any code editor.

## Index.html



```
Go Run Terminal Help
... # index.css Index.html example.js
C: > Users > DELL > Downloads > Index.html > html > body > body > p1 > script
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <style>
5
6  </style>
7  </head>
8  <body>
9    <head>
10   <title>IN CLASS EXERCISE-5</title>
11   <link href="index.css" rel="stylesheet" />
12 </head>
13 <body>
14   <h1>INTRODUCTION TO WEB TECHNOLOGIES</h1>
15   <p> Introduction to SVG and CSS</p>
16   <p1>Nature Wallpaper</p>
17   
18   <script src = "example.js"></script>
19 </body>
20 </body>
21 </html>
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Filter (e.g. text, !exclude) ▾ ×

Above is the html code, in this we have given the title name of our web page and embedded the SVG file(which is generated through figma), and we have also given the heading and the text that should be included in our web page.

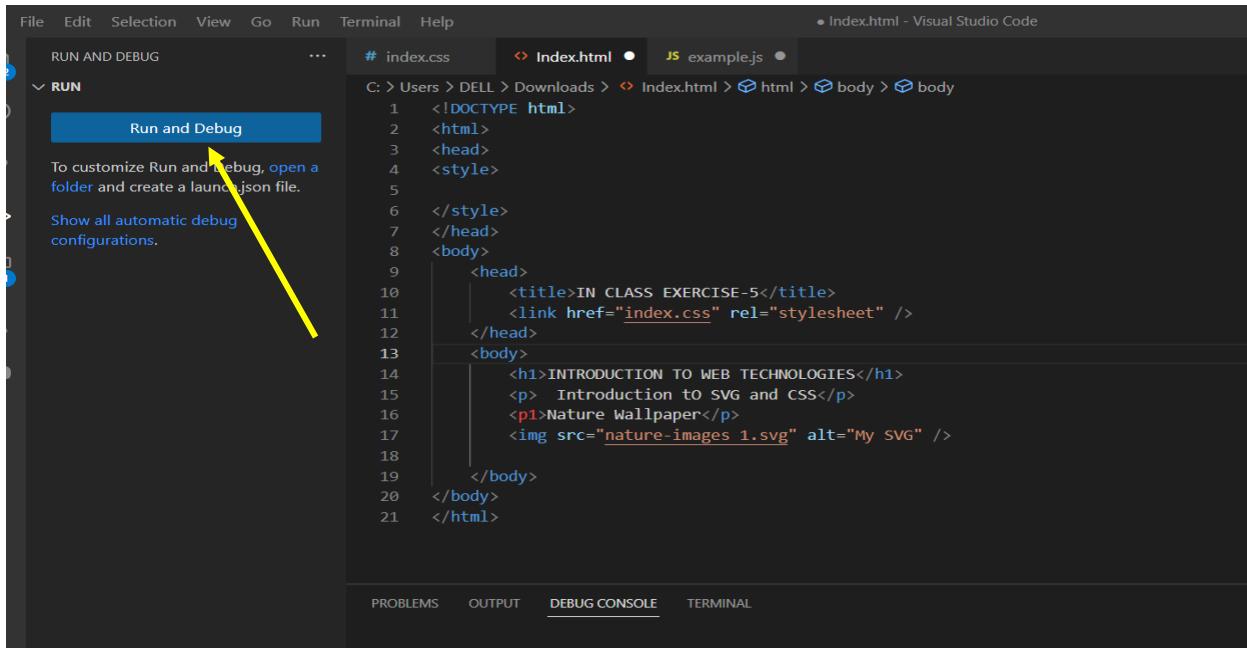
```
<!DOCTYPE html>
<html>
<head>
<style>

</style>
</head>
<body>
  <head>
    <title>IN CLASS EXERCISE-5</title>
    <link href="index.css" rel="stylesheet" />
  </head>
  <body>
    <h1>INTRODUCTION TO WEB TECHNOLOGIES</h1>
    <p> Introduction to SVG and CSS</p>
    <p1>Nature Wallpaper</p>
    

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

Above is the html code, you can keep this code in Visual studio code. Here, we can see title as “IN CLASS EXERCISE -5” and heading as “Introduction to web technologies” and paragraph content as “Introduction to SVG and CSS” and next paragraph content as “Nature Wallpaper”.

The highlighted part indicates that svg file is embedded with html.



The screenshot shows the Visual Studio Code interface with the following details:

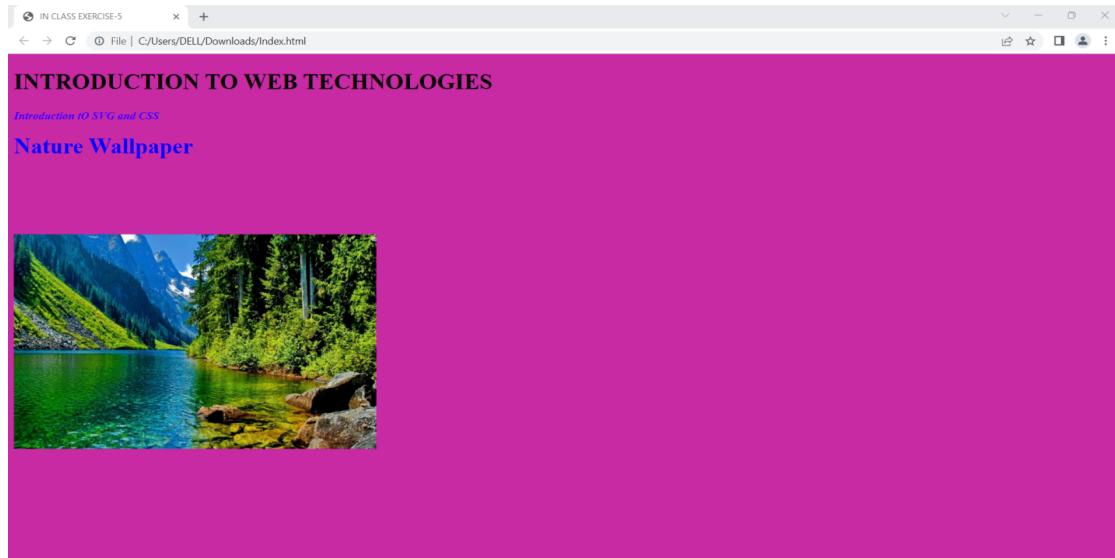
- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help
- Active File:** # index.css, Index.html (highlighted), example.js
- Project Explorer:** RUN AND DEBUG, RUN (highlighted with a yellow arrow)
- Code Editor:** HTML code for index.html:

```
<!DOCTYPE html>
<html>
<head>
<style>
</style>
</head>
<body>
<title>IN CLASS EXERCISE-5</title>
<link href="index.css" rel="stylesheet" />
</head>
<body>
<h1>INTRODUCTION TO WEB TECHNOLOGIES</h1>
<p> Introduction to SVG and CSS</p>
<p>Nature Wallpaper</p>

</body>
</body>
</html>
```
- Bottom Navigation:** PROBLEMS, OUTPUT, DEBUG CONSOLE (highlighted with a yellow arrow), TERMINAL

Open the index.html file and click on Run and Debug, then the code will be executed.

You can see the output in any browser like chrome, edge. It depends on your selection before running the code.



Above is the generated web page based on our html, CSS and svg code.

## **TASK-2**

### **Tutorial 2: Introduction to JavaScript**

```
<!DOCTYPE html>
<html>
<head>
<style>

</style>
</head>
<body>
    <head>
        <title>IN CLASS EXERCISE-5</title>
        <link href="index.css" rel="stylesheet" />
    </head>
    <body>
        <h1>INTRODUCTION TO WEB TECHNOLOGIES</h1>
        <p> Introduction to SVG and CSS</p>
        <p>Nature Wallpaper</p>
        
        <script src = "example.js"></script>
    </body>
</body>
</html>
```

In your html code, you should keep an extra line to get an alert before generating a webpage.

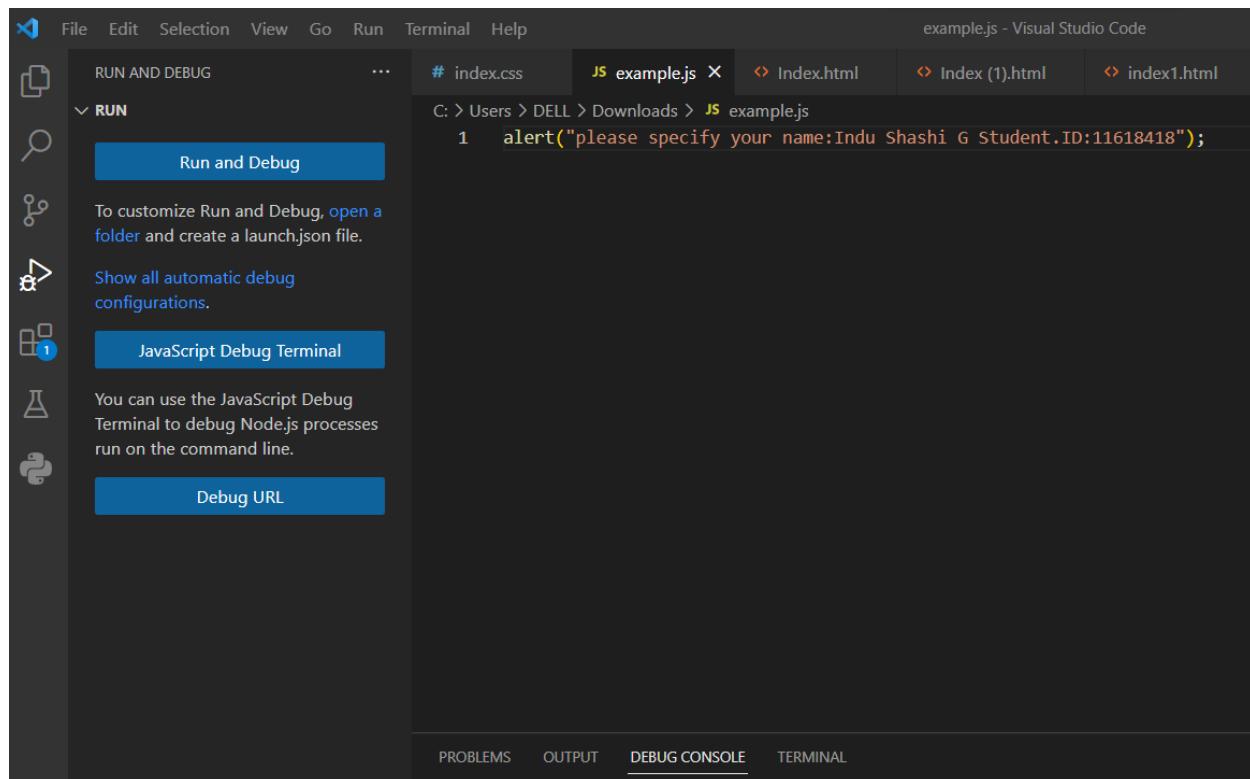
i.e, <script src = "example.js"></script>

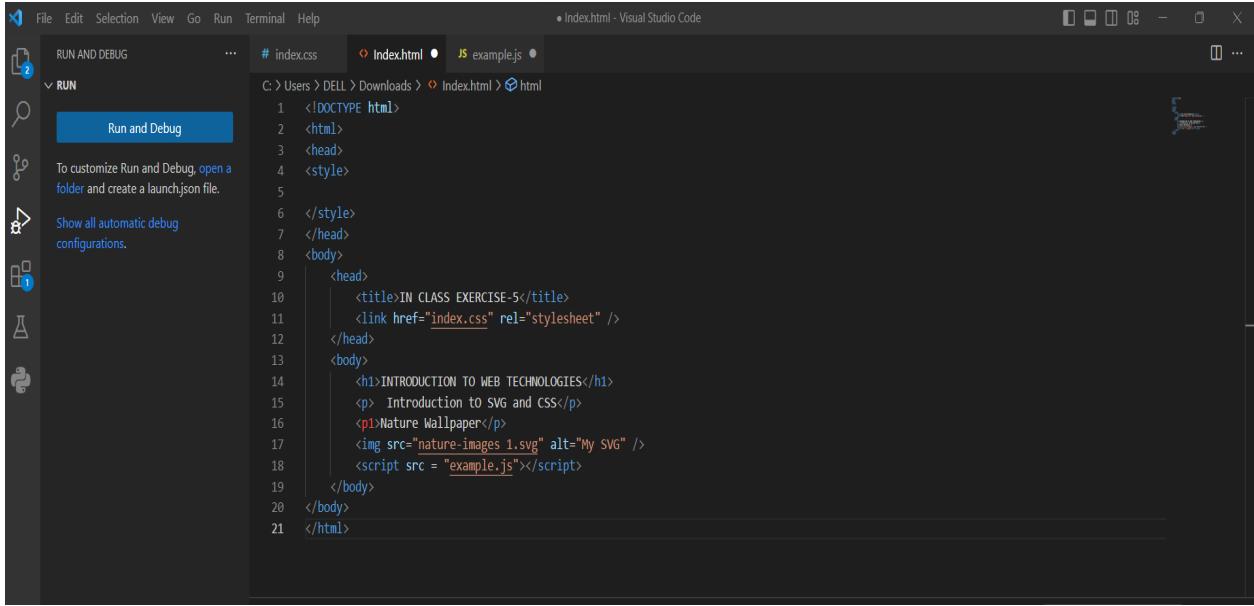
Here, example.js is a java script file. You have to create the new file in visual studio, and you have to create an alert and save it with.js extension.

### example.js

```
alert("please specify your name:Indu Shashi G Student.ID:11618418");
```

In the above code “you have to specify your name and student Id”. You must change your name and Student Id.



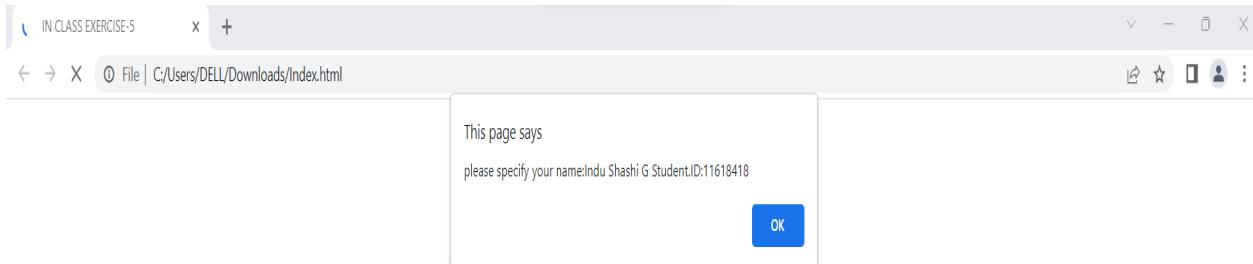


The screenshot shows the Visual Studio Code interface. On the left, the 'RUN AND DEBUG' sidebar is open, with 'Run and Debug' selected. It includes instructions to customize configurations by opening a folder and creating a launch.json file. The main editor area displays the content of 'index.html'. The code includes a title, a link to 'index.css', and a script tag pointing to 'example.js'.

```
<!DOCTYPE html>
<html>
<head>
<style>
</style>
</head>
<body>
<title>IN CLASS EXERCISE-5</title>
<link href="index.css" rel="stylesheet" />
</head>
<body>
<h1>INTRODUCTION TO WEB TECHNOLOGIES</h1>
<p> Introduction to SVG and CSS</p>
<p>Nature Wallpaper</p>

<script src = "example.js"></script>
</body>
</body>
</html>
```

After adding the example.js in html page, run the html again by clicking on run and debug, you will get the alert as shown in below screenshot.

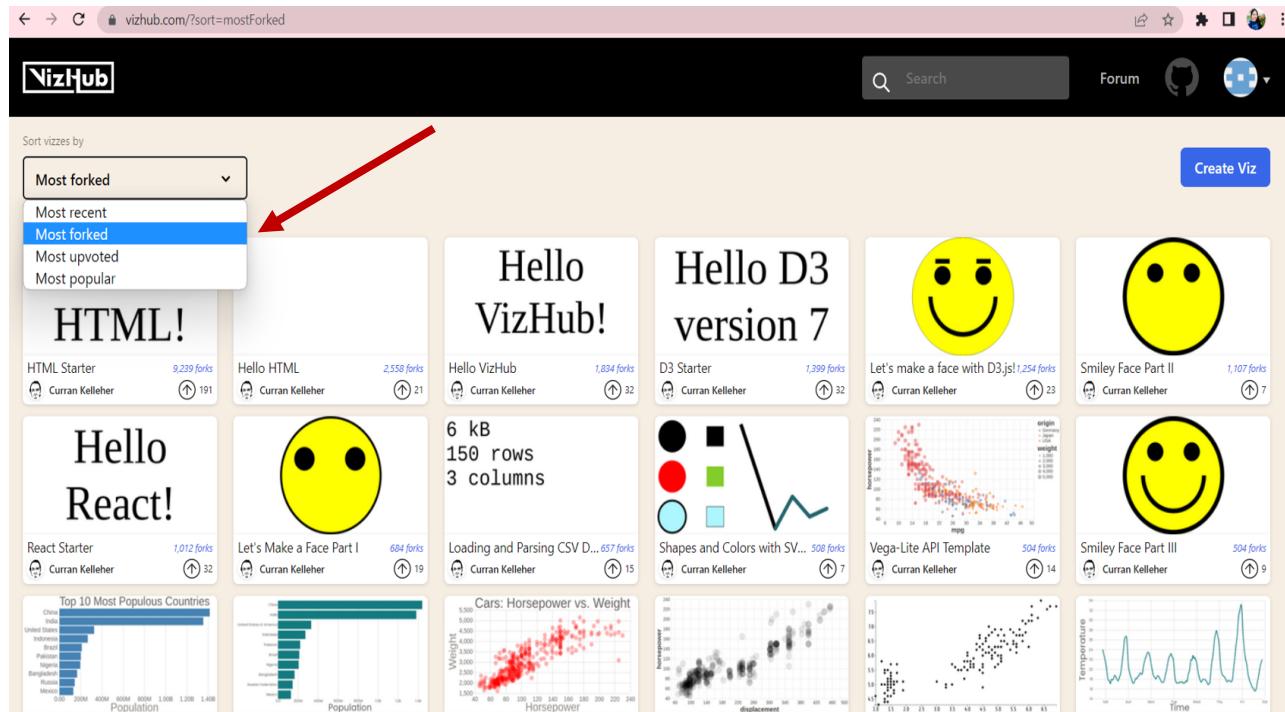


Click on Ok. Then the webpage will be generated.

## TASK-3

### Tutorial 3: Introduction to VizHub and Making a Face with D3.js

Click on <https://vizhub.com/>, log in to it and connect it with your GitHub. Then select a project from most forked/ most popular.



The screenshot shows the VizHub homepage with a dropdown menu titled "Sort vizzes by" set to "Most forked". A red arrow points to the "Most forked" option in the dropdown menu. The page displays a grid of visualization projects. Some visible titles include "Hello VizHub!", "Hello D3 version 7", "Smiley Face Part II", "Hello React!", "Let's Make a Face Part I", "6 kB 150 rows 3 columns", "Shapes and Colors with SV...", "Vega-Lite API Template", "Smiley Face Part III", and "Top 10 Most Populous Countries". Each project card includes a thumbnail, title, description, forks count, and author information.

In search box type Buddha and search for Buddha images.

The screenshot shows the VizHub platform interface. At the top, there's a search bar with the query "buddha". Below the search bar, a button labeled "Create Viz" is visible. The main area features a grid of cards, each representing a different visualization or project. Some of the visible cards include:

- Hello D3 version 7
- Fork of Simley Face
- VizHub Cumulative Unique ...
- MoMA Streamgraph
- Mexico Map
- Anv\_ICE8\_Argentina\_(Topojs...)
- Tree chart
- vcTimeGraph Test
- World Map
- Making A Bar Chart
- World\_Map\_ICE\_8
- Reproduction of A. Grigoryev
- MY Tree ICE-8
- Basic Cost Of Netflix
- Europe map
- Scatter Plot

Each card includes a small thumbnail image, the project title, the author's name, and the number of forks (a measure of popularity).

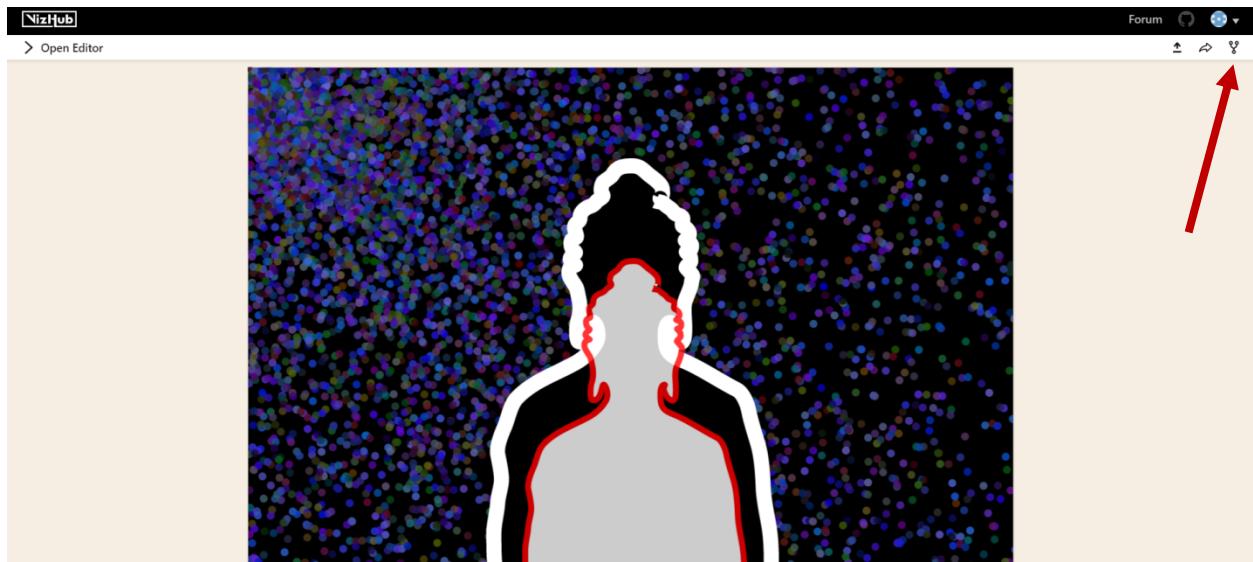
Here, you can see different images of Buddha with different backgrounds.

This screenshot shows a subset of the VizHub content related to the search term "buddha". The cards in this grid focus on different artistic representations of Buddha figures:

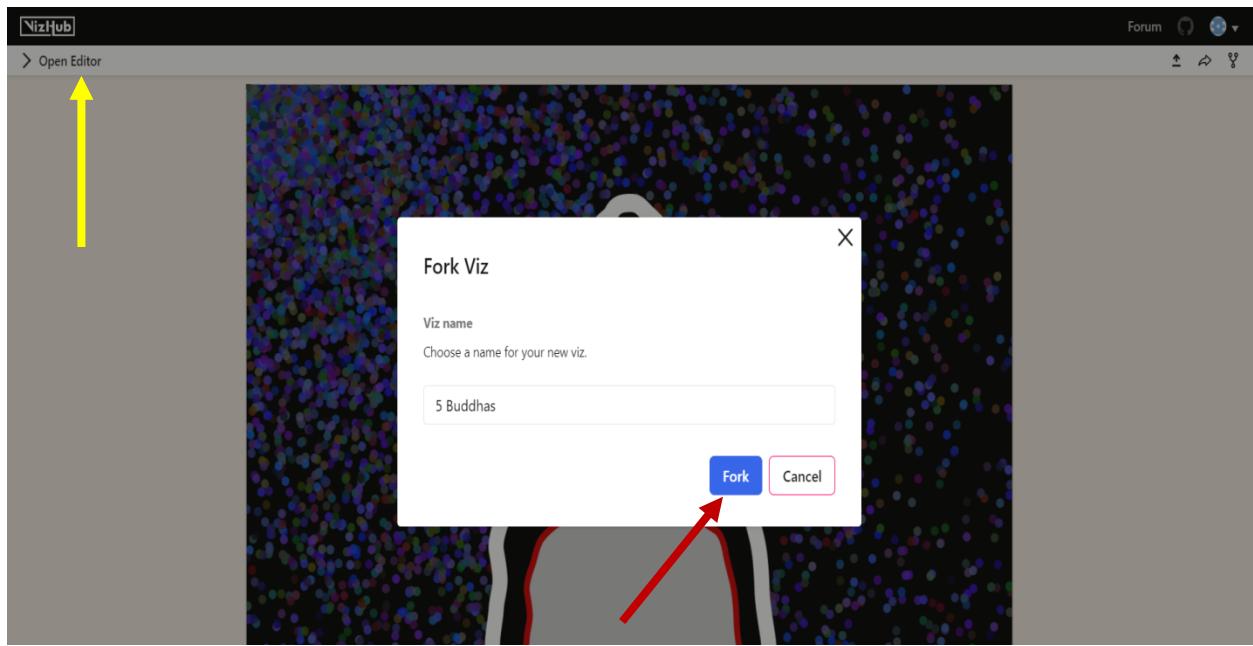
- Fork of Fork of Fork of Fork ...
- Fork of Fork of Buddha in co...
- Fork of Fork of Fork of Budd...
- Buddha in colors - strokes
- Buddha in colors - white back...
- Buddha in colors
- Buddha Mask
- Fork of Fork of Buddha Imag...
- Fork of Buddha in colors -op...
- Fork of Fork of Budd...
- Fork of Fork of Fork of For...

Each card provides a preview of the visualization, the project title, the author's name, and the number of forks.

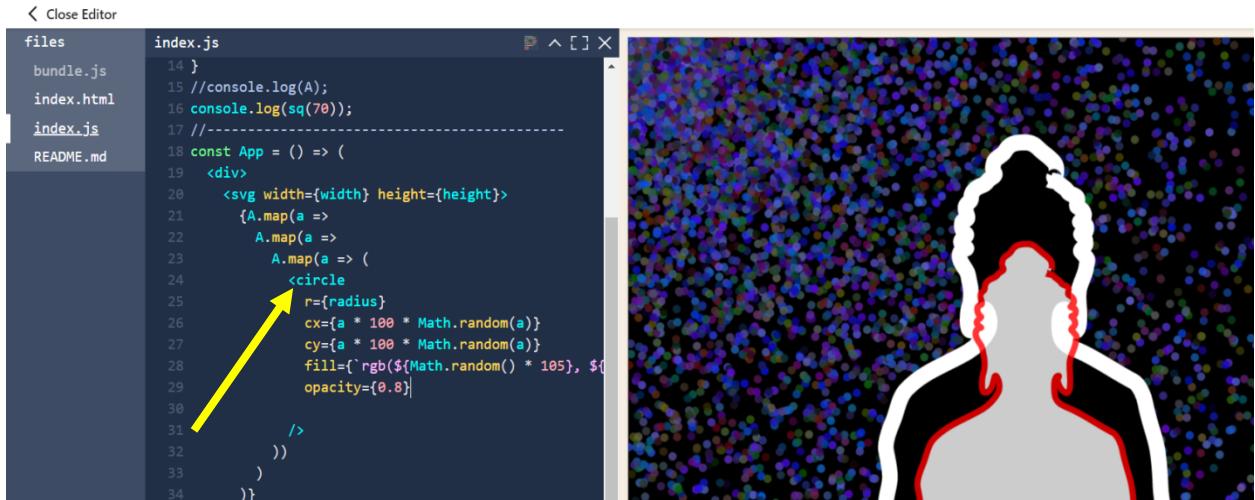
Select the first image of Buddha, which has two reflections, and the background is with dots.



Click on fork symbol and give the name and click on fork button.

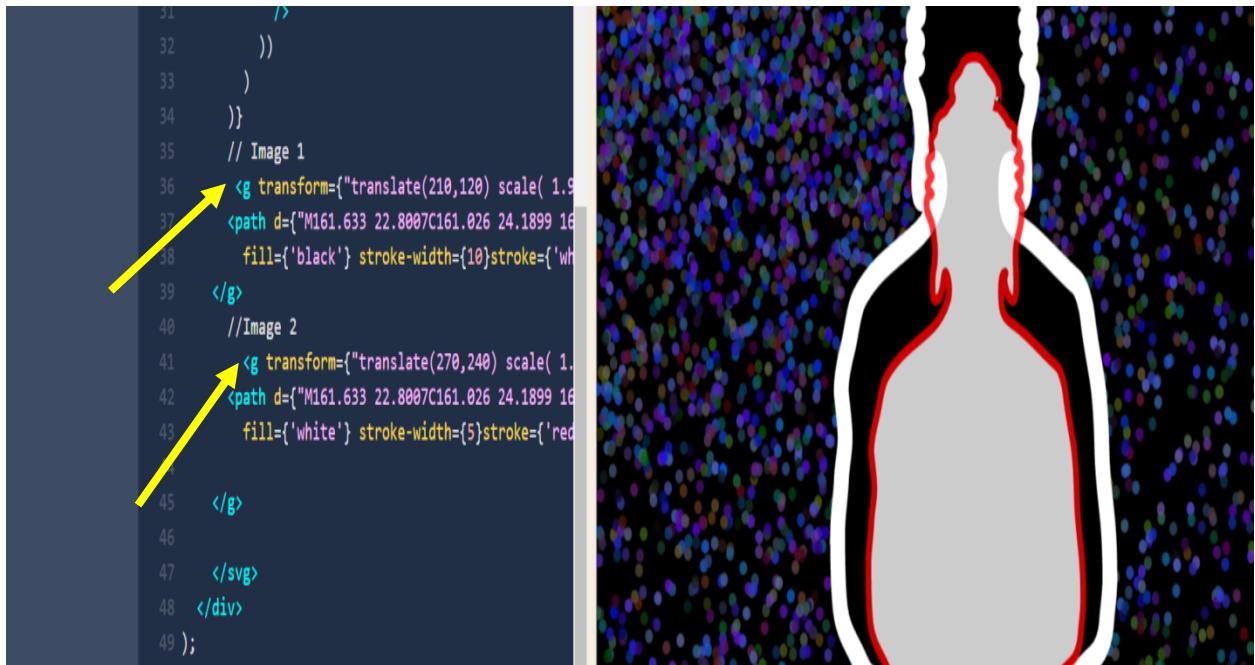


Click on open editor and select index.js, a java script file will be opened.



The image shows a code editor window with a sidebar labeled "files" containing "bundle.js", "index.html", "index.js", and "README.md". The main editor area displays the "index.js" file. A yellow arrow points to the line of code: "25 <circle r={radius} cx={a \* 100 \* Math.random(a)} cy={a \* 100 \* Math.random(a)} fill={'rgb(\${Math.random() \* 105}, \${Math.random() \* 105}, \${Math.random() \* 105})'} opacity={0.8} />". To the right of the editor is a visualization of a silhouette of a person standing, filled with a pattern of small, semi-transparent colored circles.

In the above image you can see background is with small circles, so in java script file we can see the tag name as circle.



The image shows a code editor window with the same file structure as the previous screenshot. The "index.js" file contains code for two images. A yellow arrow points to the first image definition: "36 <g transform='translate(210,120) scale( 1.9)'>". Another yellow arrow points to the second image definition: "41 <g transform='translate(270,240) scale( 1.9)'>". To the right of the editor is a visualization of a silhouette of a person standing, with two distinct reflections or images overlaid on it, each consisting of a black outline and a white fill.

As there are two reflections of Buddha, we can see in java script file there is description about two images. i.e. Image 1 and Image 2.

Here, we need to transform the picture from 2 reflections to 5 reflections and we should change the background.

```
14 }
15 //console.log(A);
16 console.log(sq(70));
17 //-----
18 const App = () => (
19   <div>
20     <svg width={width} height={height}>
21       {A.map(a =>
22         A.map(a =>
23           A.map(a => (
24             <polygon
25               points={`${a * 100 * Math.random()} ${a * 100 * Math.random()}`}
26               fill={`rgb(${Math.random() * 500}, ${Math.random() * 500}, ${Math.random() * 500})`}
27               opacity={2}
28               transform={`rotate(45 ${a * 100 * Math.random()})`})
29           )
30         )
31       />
32     )));
33   )
34 )}
```



In the above picture, we can see the background is changed to polygons from small dots.

We need to replace the code from circle to polygon.



As we have to get five reflections of Buddha and size should be decreased from one over the other, we should keep details of 5 images accordingly in java script file instead of 2 images. In the above screenshot, you can see there is description of 5 images. i.e. Image 1, Image 2, Image 3, Image 4, Image 5.

```

index.js
11 let A = [];
12 for (let i = 0; i < number + 1; i++) {
13   A[i] = i;
14 }
15 //console.log(A);
16 console.log(sq(70));
17 //-----
18 const App = () => (
19   <div>
20     <svg width={width} height={height}>
21       {A.map(a =>
22         A.map(a =>
23           A.map(a => (
24             <polygon
25               points={`${a * 100 * Math.random(a)}, ${a * 100 * Math.random(a) - radius}, ${a * 100 * Math.random(a) + radius}, ${a * 100 * Math.random(a)}`}
26               fill={`rgb(${Math.random() * 500}, ${Math.random() * 500}, ${Math.random()*355})`}
27               opacity={2}
28               transform={`rotate(45 ${a * 100 * Math.random(a)} ${a * 100 * Math.random(a)})`}
29
30             />
31           ))
32         )
33       )}
34     )

```

```

index.js
35   // Image 1
36   <g transform="translate(210,120) scale( 1.9)">
37     <path d="M161.633 22.8007C161.026 24.1899 163.146 21.5968 163.146 19.776C163.146 17.2713 163.24 15.9871 161.745 14.1187C159.539 11.3608 157.107 9
38       fill={'violet'} stroke-width={6}stroke={'white'} opacity={1}/>
39   </g>
40   //Image 2
41   <g transform="translate(270,240) scale( 1.5)">
42     <path d="M161.633 22.8007C161.026 24.1899 163.146 21.5968 163.146 19.776C163.146 17.2713 163.24 15.9871 161.745 14.1187C159.539 11.3608 157.107 9
43       fill={'green'} stroke-width={4}stroke={'red'} opacity={0.8}/>
44   </g>
45   //Image 3
46   <g transform="translate(320,360) scale( 1.2)">
47     <path d="M161.633 22.8007C161.026 24.1899 163.146 21.5968 163.146 19.776C163.146 17.2713 163.24 15.9871 161.745 14.1187C159.539 11.3608 157.107 9
48       fill={'blue'} stroke-width={3}stroke={'white'} opacity={0.4}/>
49   </g>
50   //Image 4
51   <g transform="translate(350,440) scale( 1.0)">
52     <path d="M161.633 22.8007C161.026 24.1899 163.146 21.5968 163.146 19.776C163.146 17.2713 163.24 15.9871 161.745 14.1187C159.539 11.3608 157.107 9
53       fill={'green'} stroke-width={3}stroke={'yellow'} opacity={0.4}/>
54   </g>
55   //Image 5
56   <g transform="translate(390,540) scale( 0.7)">
57     <path d="M161.633 22.8007C161.026 24.1899 163.146 21.5968 163.146 19.776C163.146 17.2713 163.24 15.9871 161.745 14.1187C159.539 11.3608 157.107 9
58       fill={'yellow'} stroke-width={1}stroke={'red'} opacity={0.4}/>
59   </g>
60 
```

The above screenshots are the code of Java script file to get 5 reflections of Buddha with one over the other and background with different polygons.

<https://vizhub.com/INDU1998-GIT/f285af37794f460d808e5d085346feab?edit=files&file=index.js>

Above is the viz hub link of 5 reflections of Buddha. You can open that link and go through it for the JavaScript file to transform 2 reflections of Buddha to 5 images of Buddha along with change in background.