Sarath Chandra kunisetty – skkh2@umsystem.edu

GitHub link: https://github.com/kunisettysarath/WebMobileProgramming-Spring22/tree/main/WebDevelopment/ICP's/ICP3

Venu Linga – vl6hw@umsystem.edu

GitHub link: https://github.com/VenubabuLinga/WebDevCourse/tree/main/WEB_DEV/ICP3

ICP 3

Objective:

The objective of this ICP3 is to understand what a responsive webpage and basics of JavaScript is by building a **responsive webpage** and **Rock Paper Scissor game**.

Rock Paper Scissor:

JavaScript (JS):

A typical website which is just built on CSS and HTML will not be interactive, so in order for website to be interactive for the users, the JavaScript is used where the user can provide some details and the website will render the data and will give a response back to the user based on the request.

JavaScript is a programming language which is widely used across the globe and been ranked as most used programing language for web development. It can be used both on client side and server side. Few of the uses cases of using JavaScript are "Drop down menu", "enabling and disabling web elements based on the requirement", etc.

Tasks:

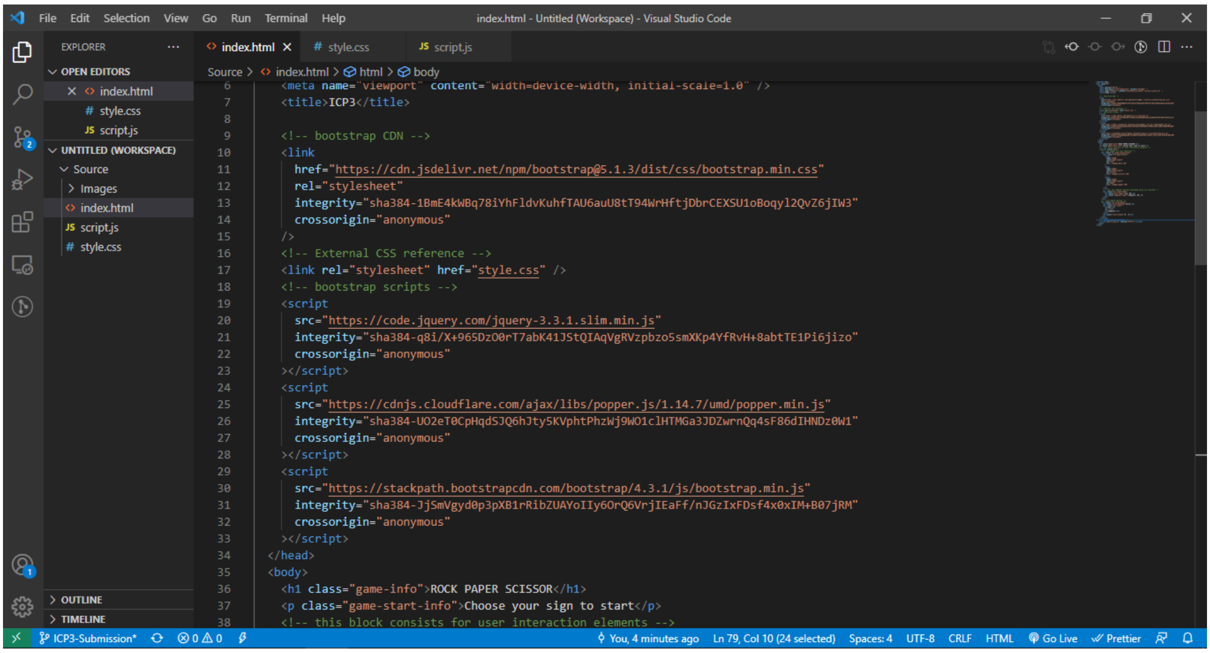
- Created "ICP3-submission" branch in the remote repository and have checkout to that branch using GitHub desktop
- Then required folders for source and documentation was created for icp2 and the source folder was imported in VS code
- Have create index.html, style.css, script.js and images and placed all the images in the folder

How to play the game:

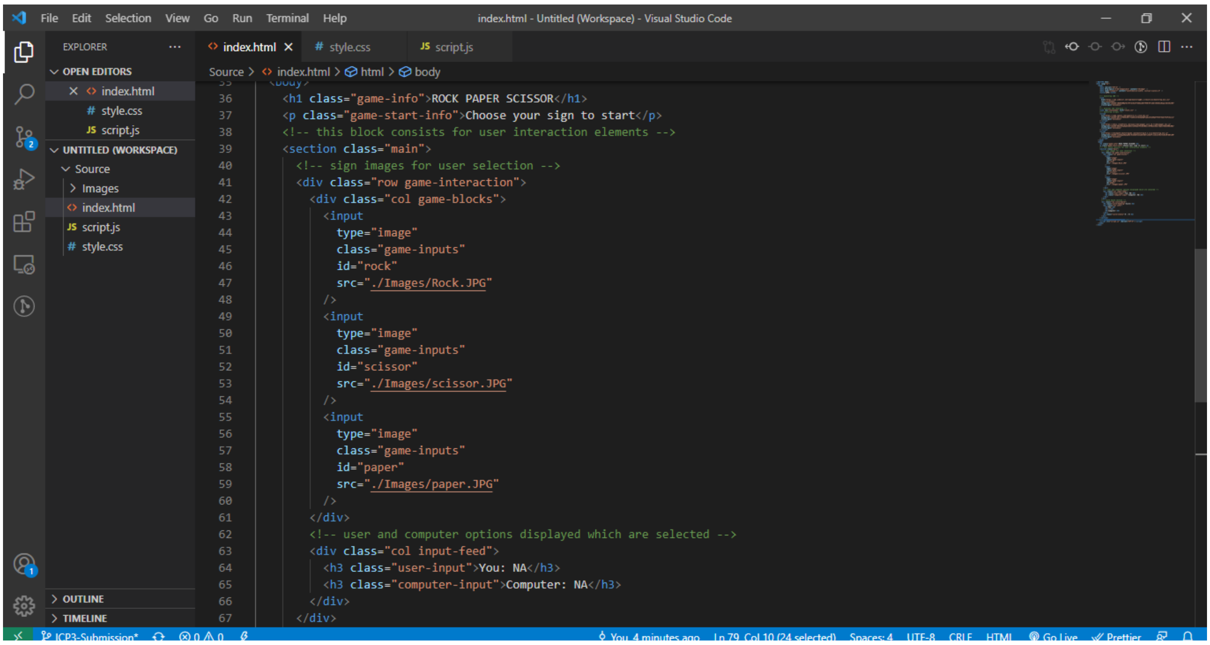
- Once the webpage is loaded user has to click on any of the sign in order to start the game
- And based on the users input and the generated random computer input the winner is decided and the same is displayed at the top of the webpage
- The score streak is also calculated and is displayed the number of wins of the user and computer on the webpage
- User has to refresh the page if the game has to be restarted

Index.html:

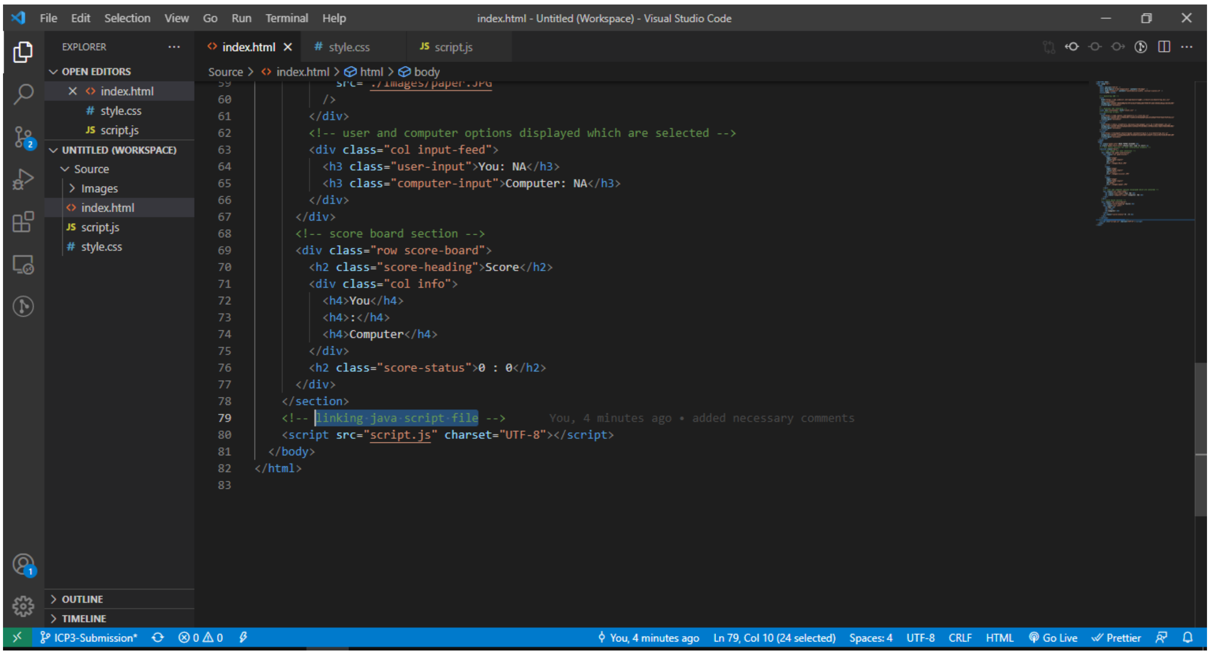
Added all the links for the required resources like external css sheet, bootstrap and javascript(please refer the screenshot below)



```html
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>ICP3</title>

    <!-- bootstrap CDN -->
    <link
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
      rel="stylesheet"
      integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
      crossorigin="anonymous"
    />
    <!-- External CSS reference -->
    <link rel="stylesheet" href="style.css" />
    <!-- bootstrap scripts -->
    <script
      src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
      integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
      crossorigin="anonymous"
    ></script>
    <script
      src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
      integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
      crossorigin="anonymous"
    ></script>
    <script
      src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
      integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
      crossorigin="anonymous"
    ></script>
  </head>
  <body>
    <h1 class="game-info">ROCK PAPER SCISSOR</h1>
    <p class="game-start-info">Choose your sign to start</p>
    <!-- this block consists for user interaction elements -->
```

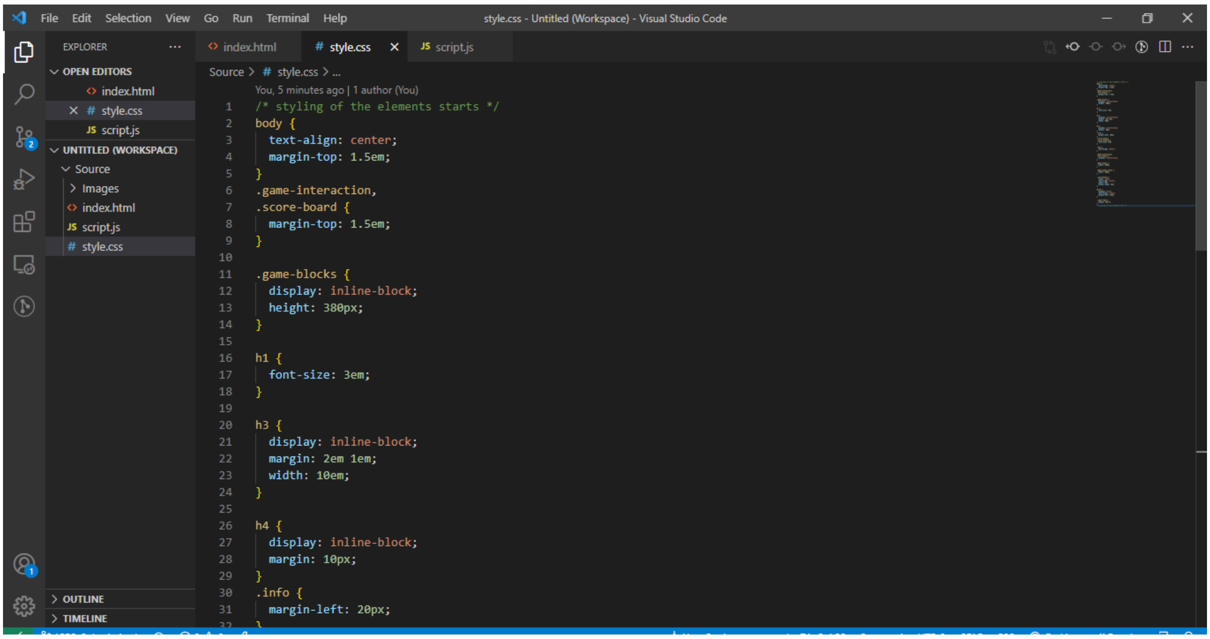


```html
  <body>
    <h1 class="game-info">ROCK PAPER SCISSOR</h1>
    <p class="game-start-info">Choose your sign to start</p>
    <!-- this block consists for user interaction elements -->
    <section class="main">
      <!-- sign images for user selection -->
      <div class="row game-interaction">
        <div class="col game-blocks">
          <input
            type="image"
            class="game-inputs"
            id="rock"
            src="./Images/Rock.JPG"
          />
          <input
            type="image"
            class="game-inputs"
            id="scissor"
            src="./Images/scissor.JPG"
          />
          <input
            type="image"
            class="game-inputs"
            id="paper"
            src="./Images/paper.JPG"
          />
        </div>
        <!-- user and computer options displayed which are selected -->
        <div class="col input-feed">
          <h3 class="user-input">You: NA</h3>
          <h3 class="computer-input">Computer: NA</h3>
        </div>
      </div>
```



```
            src="./images/paper.JPG"
          />
        </div>
        <!-- user and computer options displayed which are selected -->
        <div class="col input-feed">
          <h3 class="user-input">You: NA</h3>
          <h3 class="computer-input">Computer: NA</h3>
        </div>
      </div>
      <!-- score board section -->
      <div class="row score-board">
        <h2 class="score-heading">Score</h2>
        <div class="col info">
          <h4>You</h4>
          <h4>:</h4>
          <h4>Computer</h4>
        </div>
        <h2 class="score-status">0 : 0</h2>
      </div>
    </section>
    <!-- linking java script file -->       You, 4 minutes ago • added necessary comments
    <script src="script.js" charset="UTF-8"></script>
  </body>
</html>
```

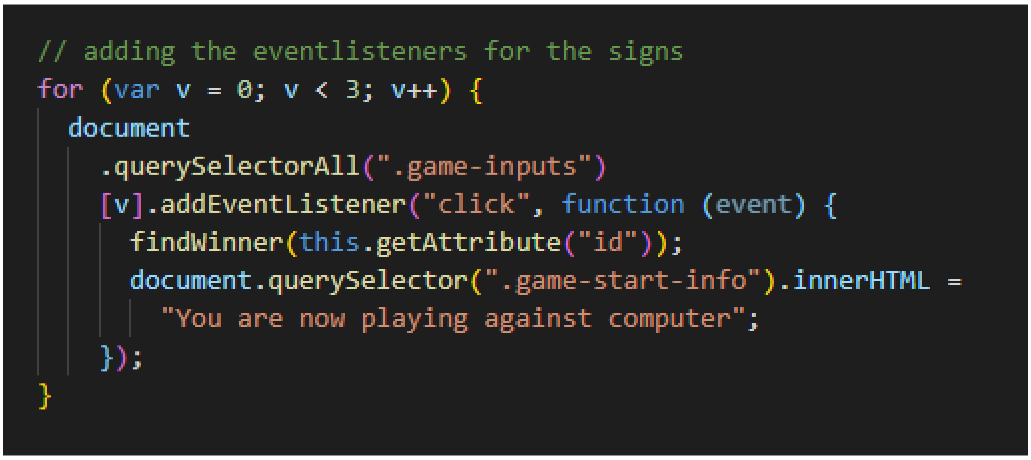
Style.css:



```
You, 5 minutes ago | 1 author (You)
/* styling of the elements starts */
body {
  text-align: center;
  margin-top: 1.5em;
}
.game-interaction,
.score-board {
  margin-top: 1.5em;
}

.game-blocks {
  display: inline-block;
  height: 380px;
}

h1 {
  font-size: 3em;
}

h3 {
  display: inline-block;
  margin: 2em 1em;
  width: 10em;
}

h4 {
  display: inline-block;
  margin: 10px;
}
.info {
  margin-left: 20px;
```

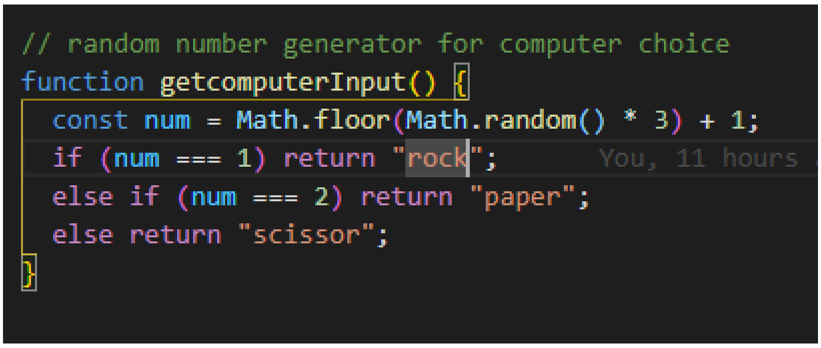
<u>Script.js:</u>

The **for loop** adds the click events for 3 sign images



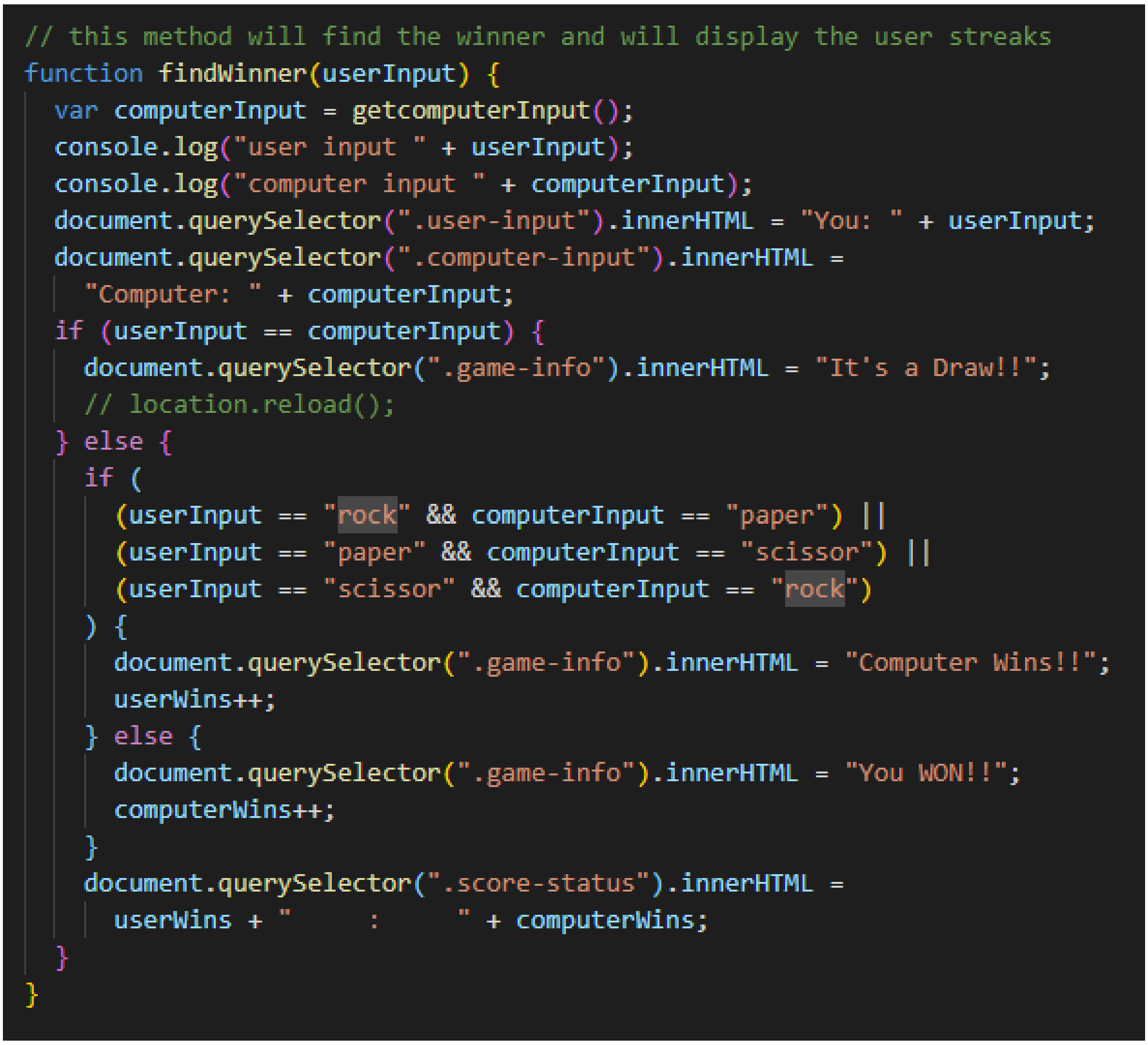
```
// adding the eventlisteners for the signs
for (var v = 0; v < 3; v++) {
  document
    .querySelectorAll(".game-inputs")
    [v].addEventListener("click", function (event) {
      findWinner(this.getAttribute("id"));
      document.querySelector(".game-start-info").innerHTML =
        "You are now playing against computer";
    });
}
```

Random generator is used to get a random number from 1 to 3 and based on that I have assigned a sign which will be referred as a computer choice.



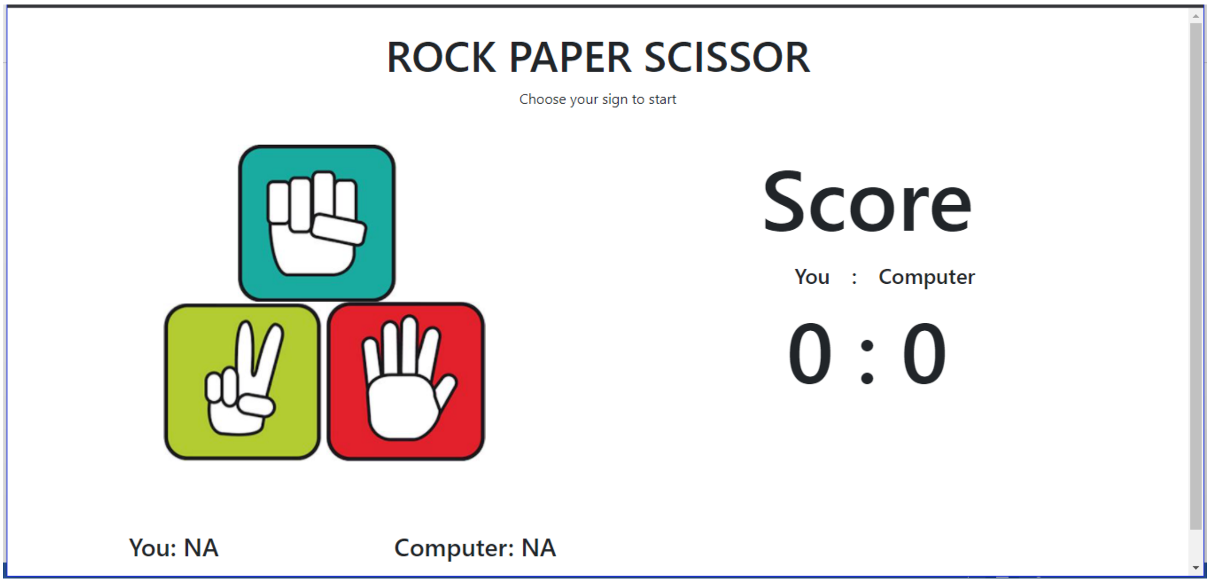
```
// random number generator for computer choice
function getcomputerInput() {
  const num = Math.floor(Math.random() * 3) + 1;
  if (num === 1) return "rock";        You, 11 hours
  else if (num === 2) return "paper";
  else return "scissor";
}
```

The main logic is enclosed in this method, which is invoked based on the events recorded. And the winner is decided and the respective winner is passed to the webpage and score is calculated.



```javascript
// this method will find the winner and will display the user streaks
function findWinner(userInput) {
  var computerInput = getcomputerInput();
  console.log("user input " + userInput);
  console.log("computer input " + computerInput);
  document.querySelector(".user-input").innerHTML = "You: " + userInput;
  document.querySelector(".computer-input").innerHTML =
    "Computer: " + computerInput;
  if (userInput == computerInput) {
    document.querySelector(".game-info").innerHTML = "It's a Draw!!";
    // location.reload();
  } else {
    if (
      (userInput == "rock" && computerInput == "paper") ||
      (userInput == "paper" && computerInput == "scissor") ||
      (userInput == "scissor" && computerInput == "rock")
    ) {
      document.querySelector(".game-info").innerHTML = "Computer Wins!!";
      userWins++;
    } else {
      document.querySelector(".game-info").innerHTML = "You WON!!";
      computerWins++;
    }
    document.querySelector(".score-status").innerHTML =
      userWins + "     :     " + computerWins;
  }
}
```

Output:



ROCK PAPER SCISSOR

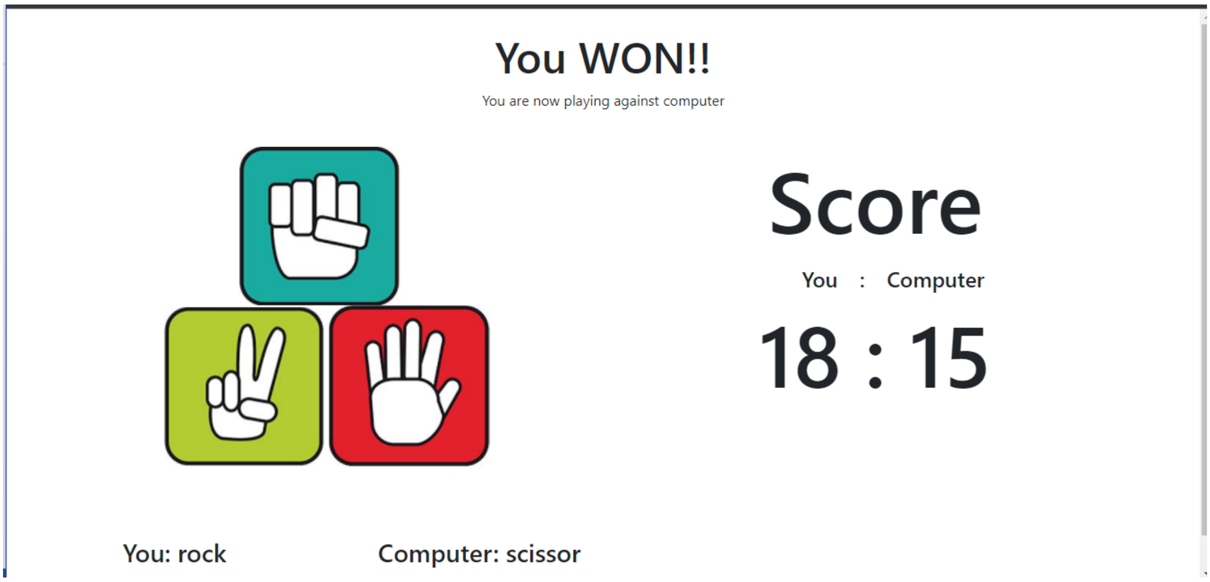
Choose your sign to start

Score

You : Computer

0 : 0

You: NA Computer: NA



You WON!!

You are now playing against computer

Score

You : Computer

18 : 15

You: rock Computer: scissor

Responsive Web page:

Responsive Web design refers to a design of page or application that responds to user in which it is viewed. It has a various CSS and HTML code and essential techniques.

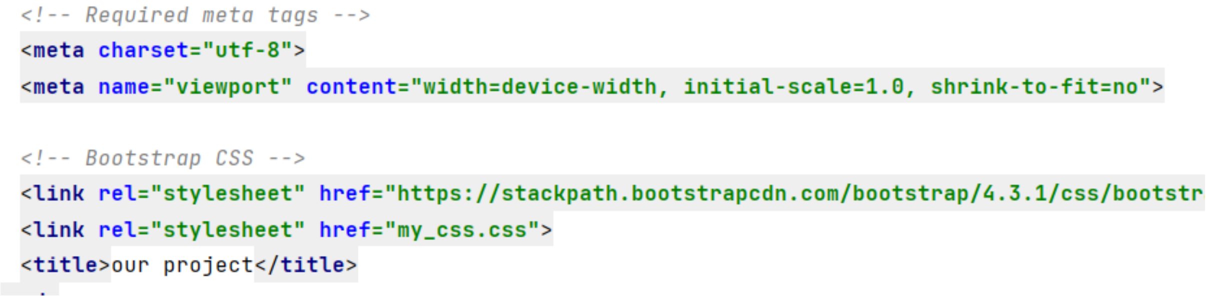
We have created an webpage with the help of RWD techniques. In which we have divided the page into 3 division mainly

1.Profile

2.Wrapper

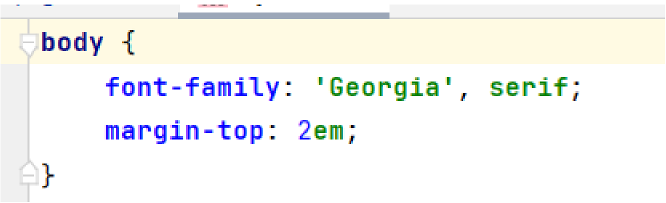
3.Image column

Firstly, we have imported the required libraries to our html page Added Shrink to fit as no in view meta tag.



```
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0, shrink-to-fit=no">

<!-- Bootstrap CSS -->
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstr
<link rel="stylesheet" href="my_css.css">
<title>our project</title>
```

Added body level front and margin at top of page.

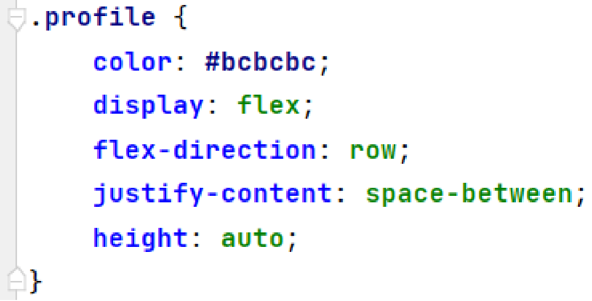


```
body {
    font-family: 'Georgia', serif;
    margin-top: 2em;
}
```

1. **Profile section**

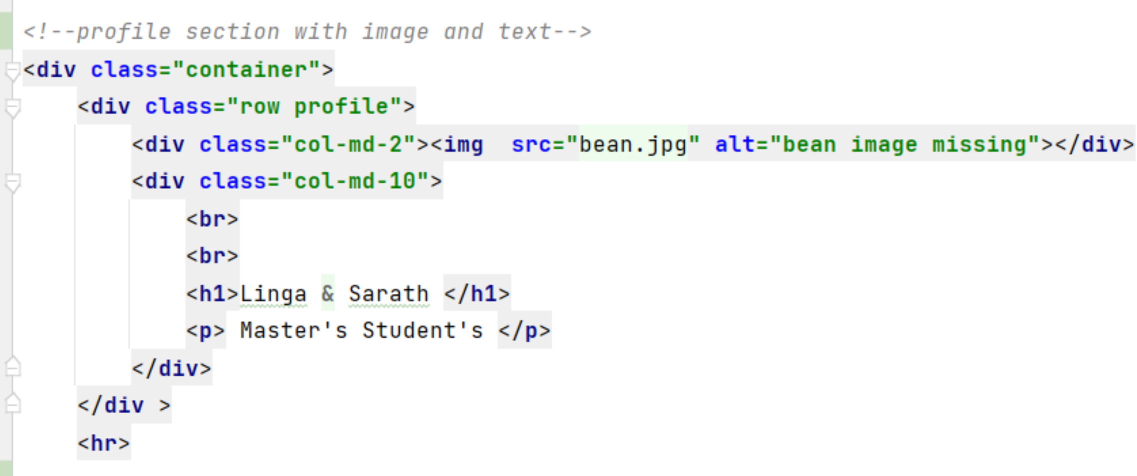
   Here we used class container with the row and column 1 and column 11, inserted the image to right and text to left of the page. And added the horizontal line to the bottom of the profile block.

Css code:



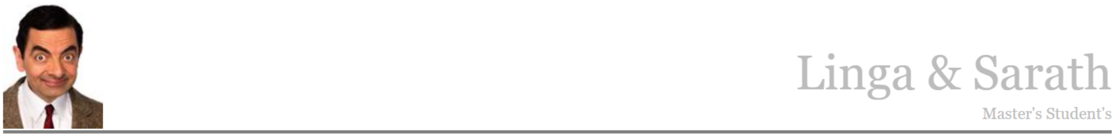
```css
.profile {
     color: #bcbcbc;
     display: flex;
     flex-direction: row;
     justify-content: space-between;
     height: auto;
}
```

HTML code:



```html
<!--profile section with image and text-->
<div class="container">
    <div class="row profile">
        <div class="col-md-2"><img  src="bean.jpg" alt="bean image missing"></div>
        <div class="col-md-10">
            <br>
            <br>
            <h1>Linga & Sarath </h1>
            <p> Master's Student's </p>
        </div>
    </div >
    <hr>
```

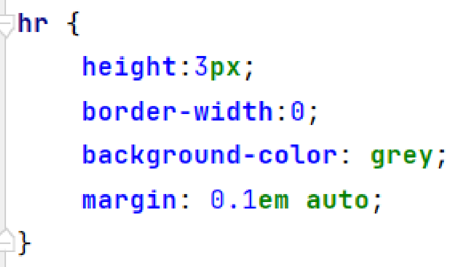
Output:



Linga & Sarath

Master's Student's

Horizontal line CSS code:

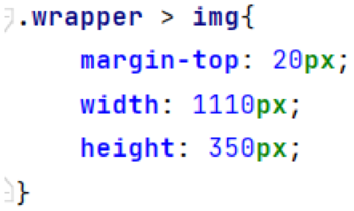


```css
hr {
    height:3px;
    border-width:0;
    background-color: grey;
    margin: 0.1em auto;
}
```

Wrapper section :

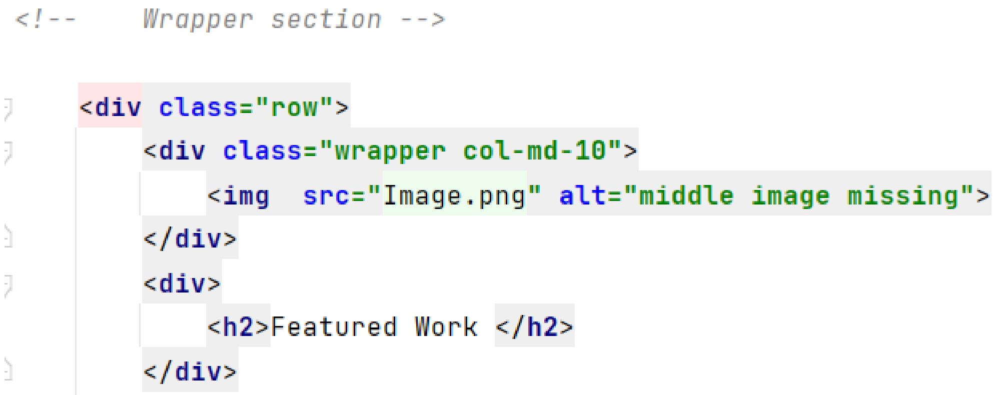
Here we have used an image tag in the next row class and made column to medium level display. And added the image width and height attributes to image.

CSS code:



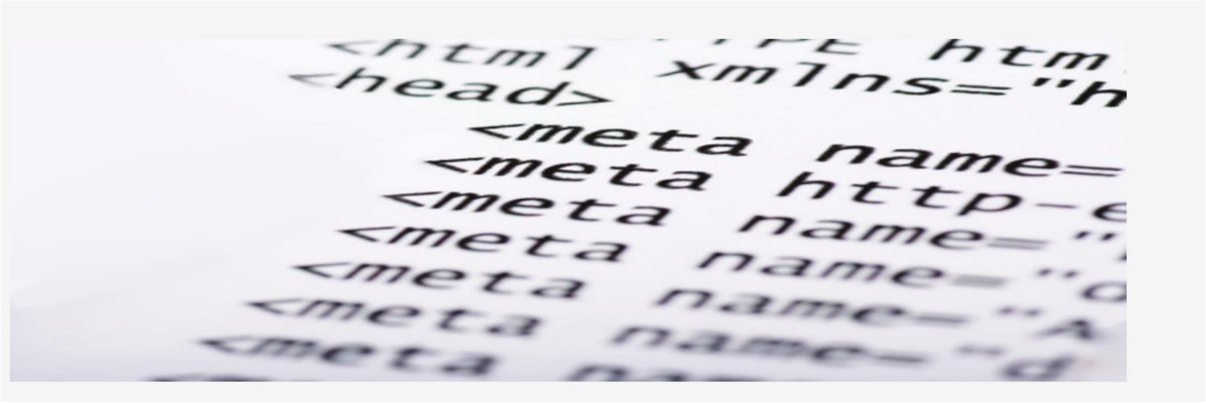
```
.wrapper > img{
     margin-top: 20px;
     width: 1110px;
     height: 350px;
}
```

HTML Code:



```
<!--     Wrapper section -->

     <div class="row">
          <div class="wrapper col-md-10">
               <img  src="Image.png" alt="middle image missing">
          </div>
          <div>
               <h2>Featured Work </h2>
          </div>
```

Output :

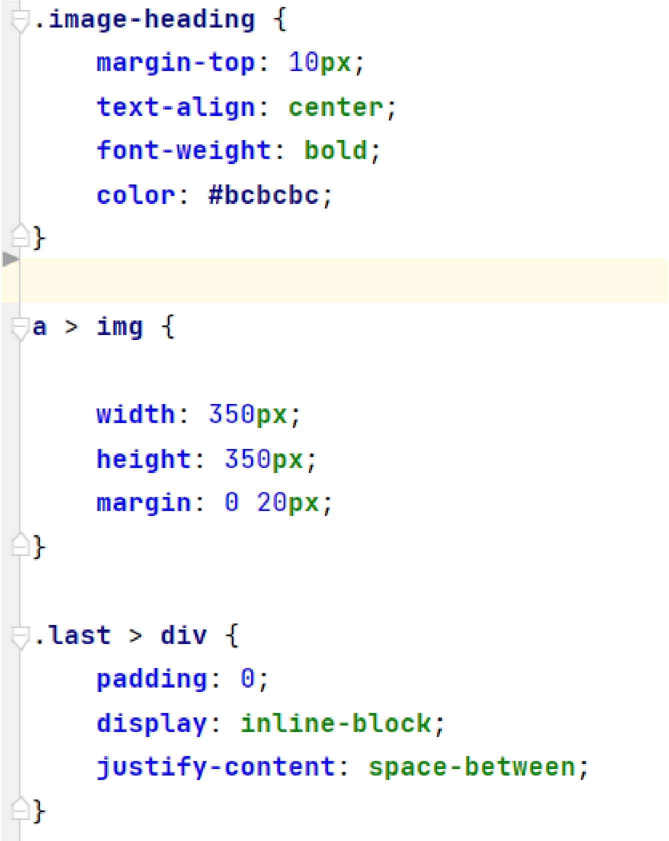


3.Image column:

In this section we have divided the row (last_set) into 3 parts and assigned the images with the hyperlink to the same.

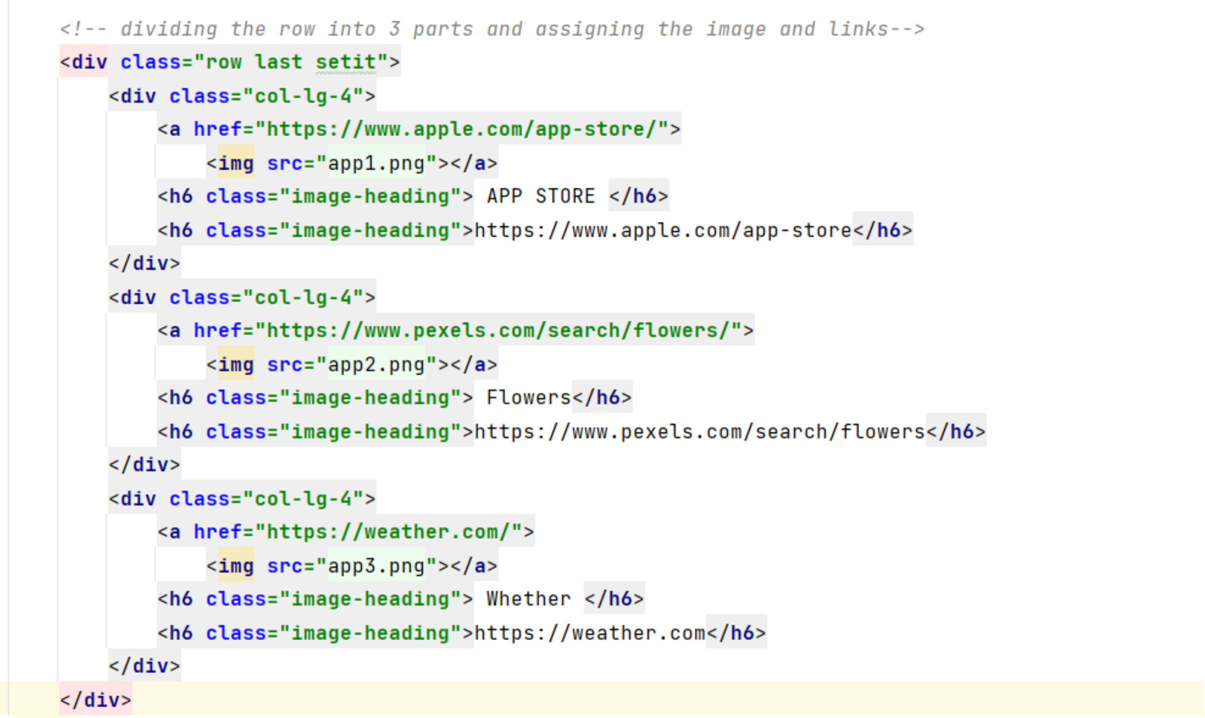
CSS code:

For all of the images under the anchor tag, we have added the same height and width and a left, right margin inorder to maintain space between the images



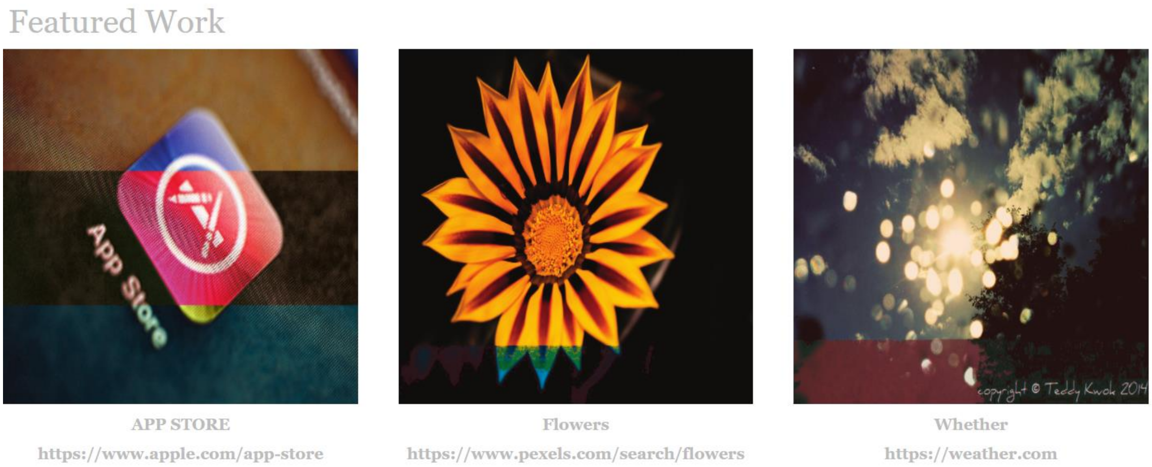
```css
.image-heading {
    margin-top: 10px;
    text-align: center;
    font-weight: bold;
    color: #bcbcbc;
}

a > img {

    width: 350px;
    height: 350px;
    margin: 0 20px;
}

.last > div {
    padding: 0;
    display: inline-block;
    justify-content: space-between;
}
```

<u>HTML code:</u>



```html
<!-- dividing the row into 3 parts and assigning the image and links-->
<div class="row last setit">
    <div class="col-lg-4">
        <a href="https://www.apple.com/app-store/">
            <img src="app1.png"></a>
        <h6 class="image-heading"> APP STORE </h6>
        <h6 class="image-heading">https://www.apple.com/app-store</h6>
    </div>
    <div class="col-lg-4">
        <a href="https://www.pexels.com/search/flowers/">
            <img src="app2.png"></a>
        <h6 class="image-heading"> Flowers</h6>
        <h6 class="image-heading">https://www.pexels.com/search/flowers</h6>
    </div>
    <div class="col-lg-4">
        <a href="https://weather.com/">
            <img src="app3.png"></a>
        <h6 class="image-heading"> Whether </h6>
        <h6 class="image-heading">https://weather.com</h6>
    </div>
</div>
```

OutPut:



Featured Work

APP STORE

https://www.apple.com/app-store

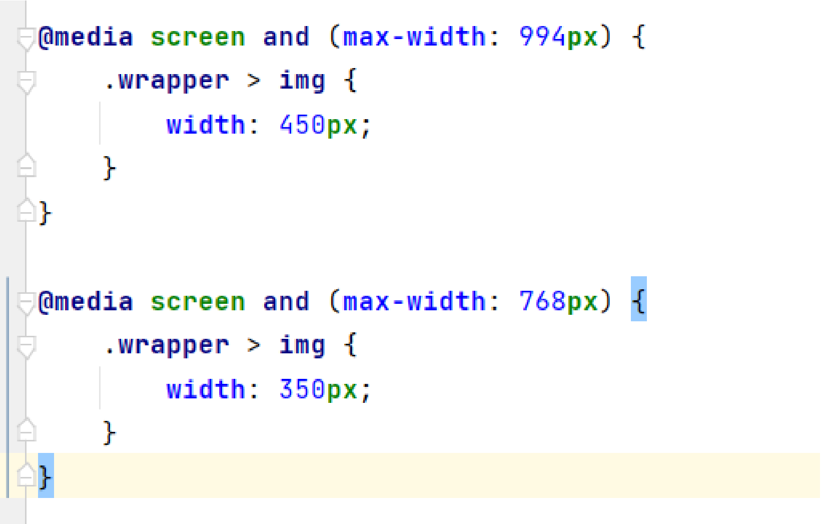
Flowers

https://www.pexels.com/search/flowers

Whether

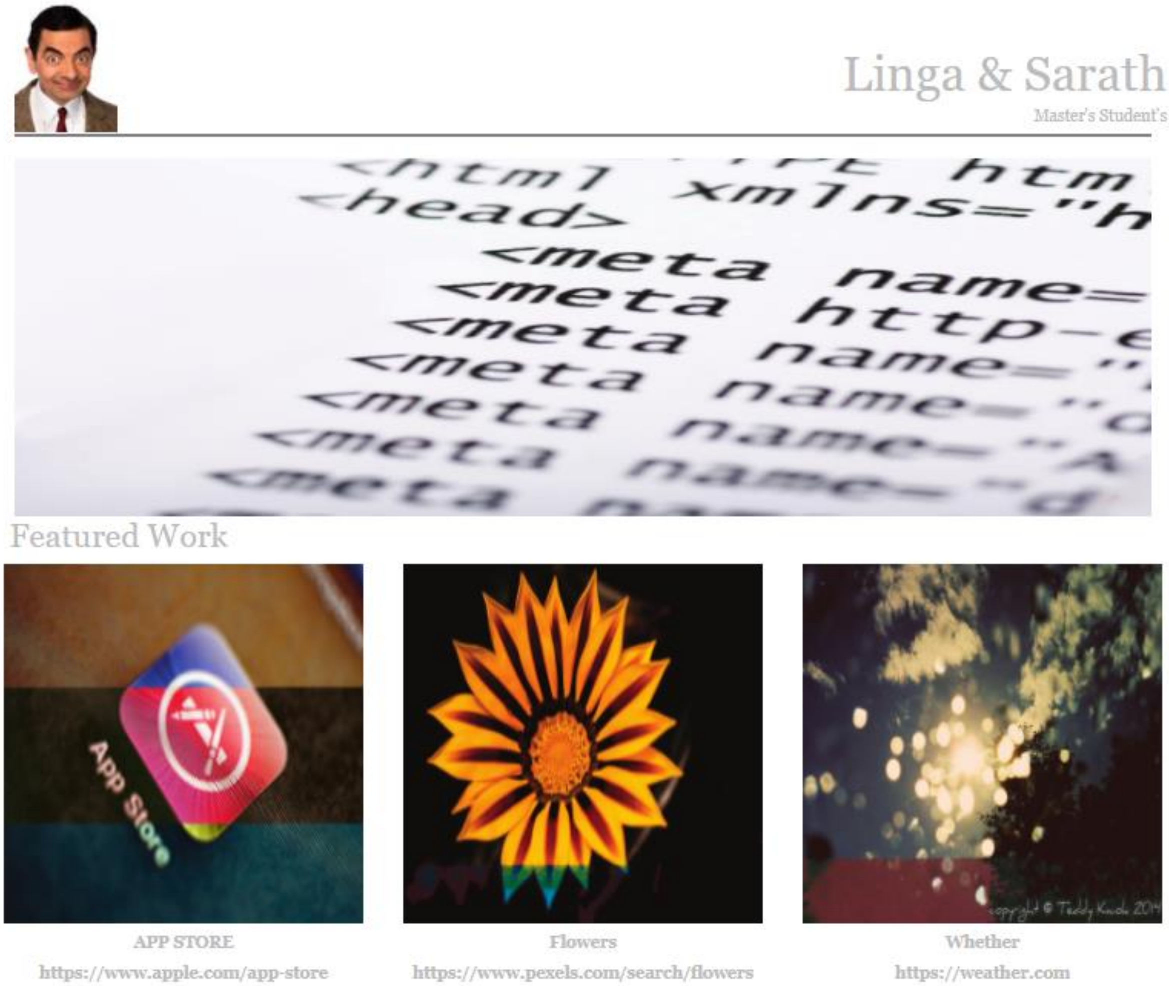
https://weather.com

<u>Media Query:</u> We have used media query inorder to add the responsiveness for the large image, two resolutions has been specified where the size of the image will be changed for resolutions less than 994px and 768px which is 450px and 350px respectively



```
@media screen and (max-width: 994px) {
    .wrapper > img {
        width: 450px;
    }
}

@media screen and (max-width: 768px) {
    .wrapper > img {
        width: 350px;
    }
}
```

Final Output:



Linga & Sarath

Master's Student's

Featured Work

APP STORE

https://www.apple.com/app-store

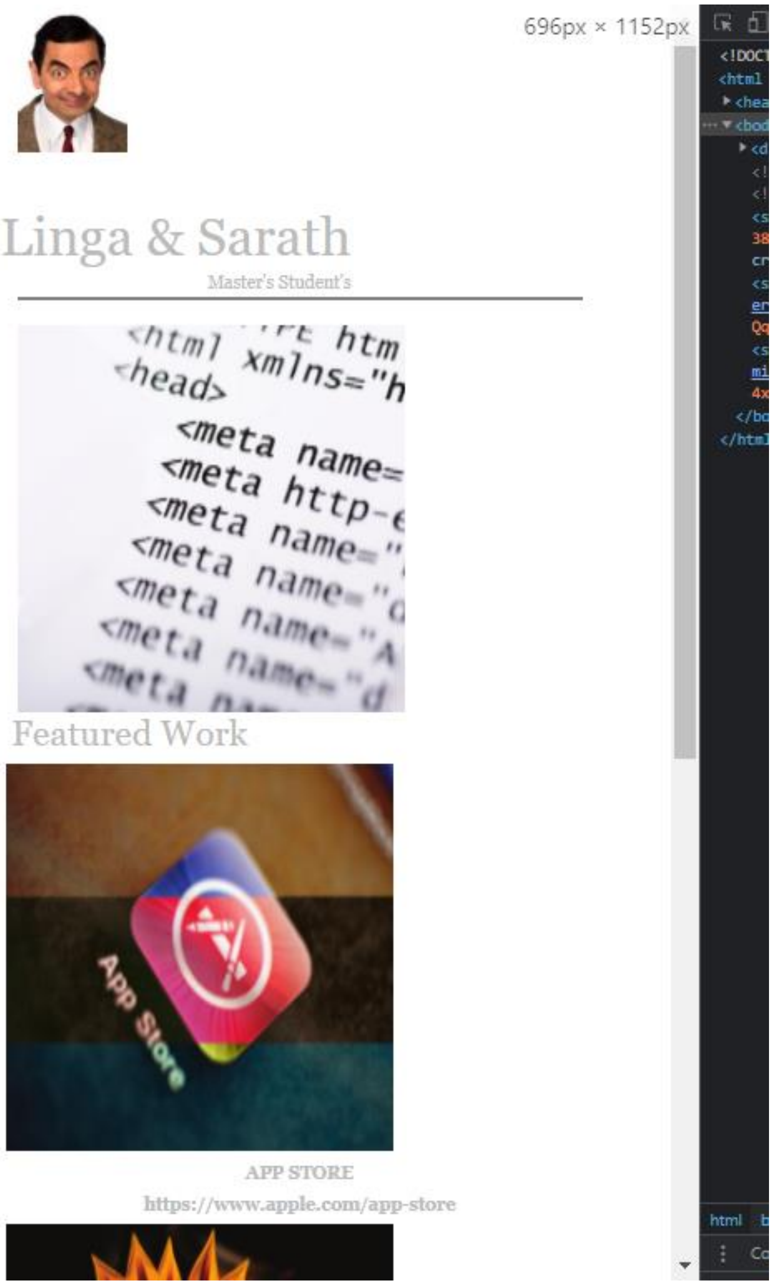
Flowers

https://www.pexels.com/search/flowers

Whether

https://weather.com

Responsive Output:



696px × 1152px

Linga & Sarath

Master's Student's

Featured Work

APP STORE

https://www.apple.com/app-store