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:

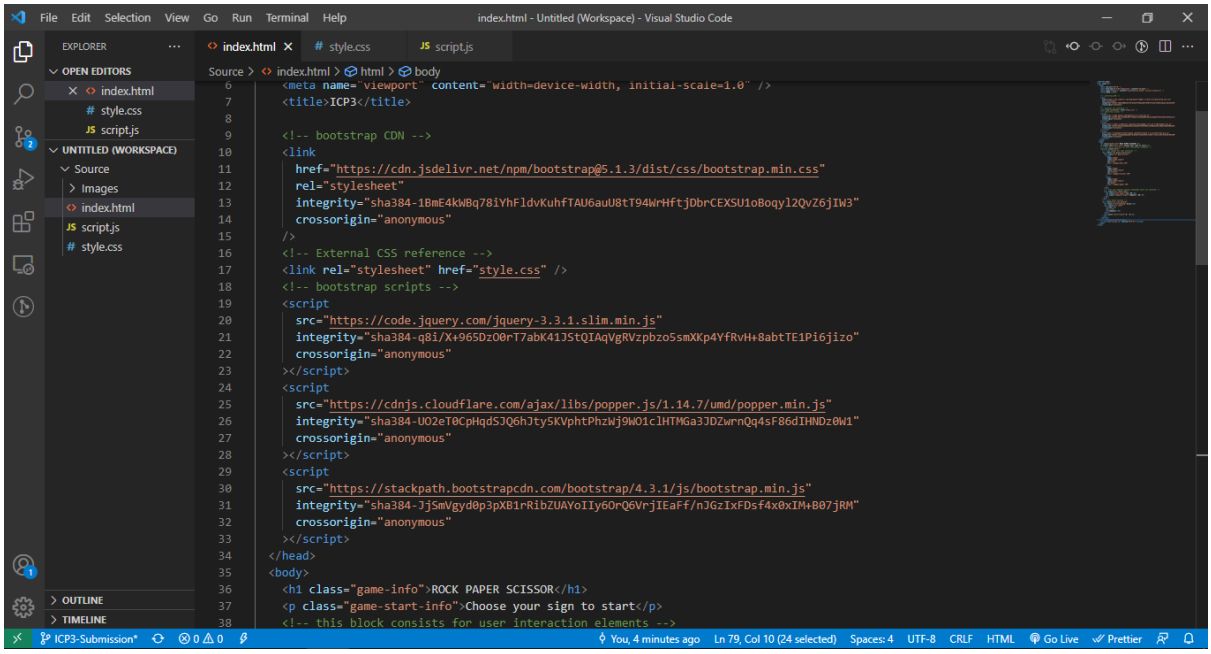
- Created “ICP3-submission” branch in the remote repository and have checkout to that branch using GitHub desktop
- Then required folders for source and documentation was created for icp2 and the source folder was imported in VS code
- Have create index.html, style.css, script.js and images and placed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

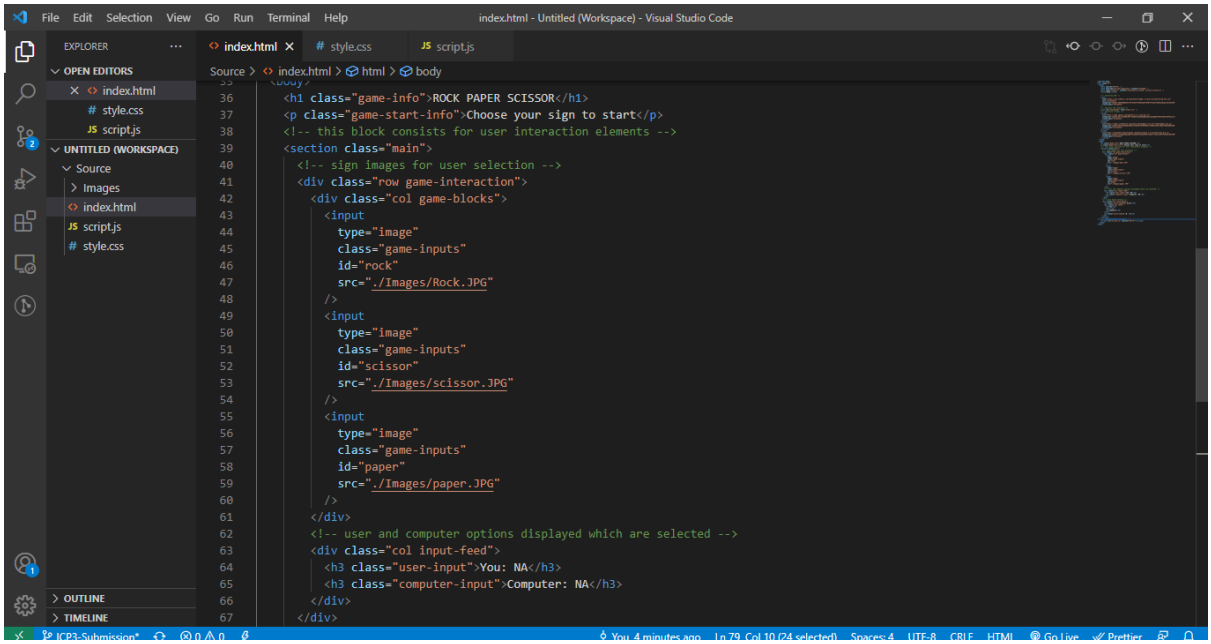
How to play the game:

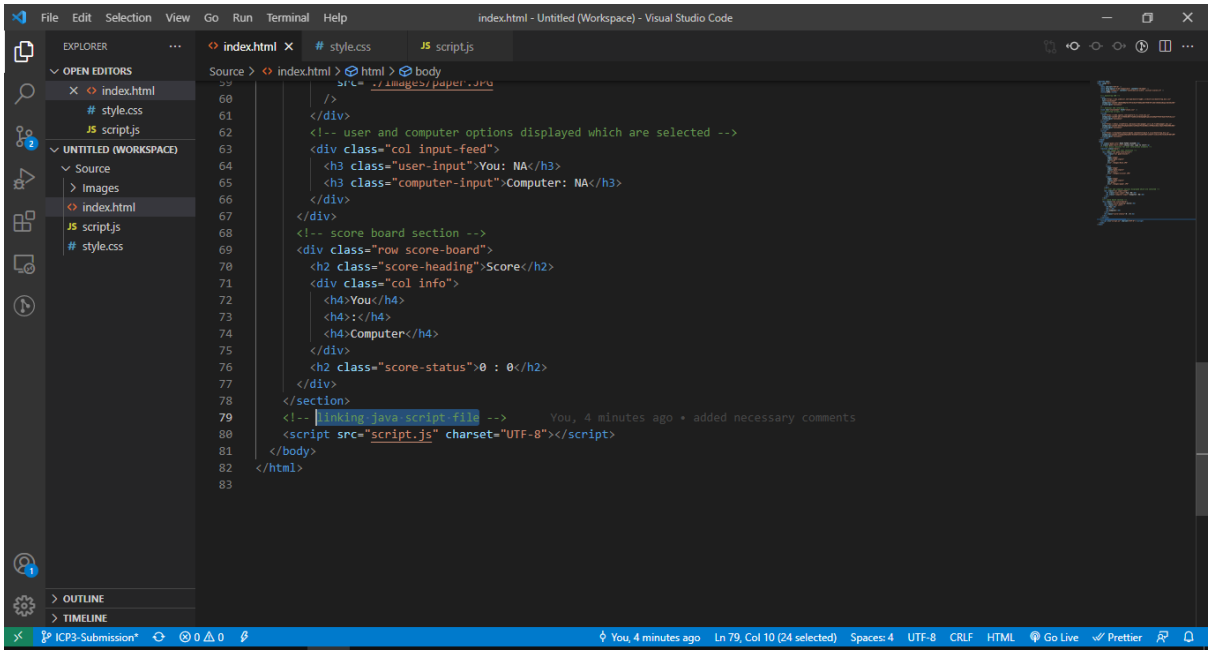
- Once the webpage is loaded user has to click on any of the sign in order to start the game
- And based on the users input and the generated random computer input the winner is decided and the same is displayed at the top of the webpage
- The score streak is also calculated and is displayed the number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






File Edit Selection View Go Run Terminal Help
index.html - Untitled (Workspace) - Visual Studio Code
EXPLORER
OPEN EDITORS
index.html
style.css
script.js
UNTITLED (WORKSPACE)
Source
Images
index.html
script.js
style.css
OUTLINE
TIMELINE
index.html
style.css
script.js
Source > index.html > html > body
/>

<!-- user and computer options displayed which are selected -->

<h3 class="user-input">You: NA</h3>
<h3 class="computer-input">Computer: NA</h3>

<!-- score board section -->

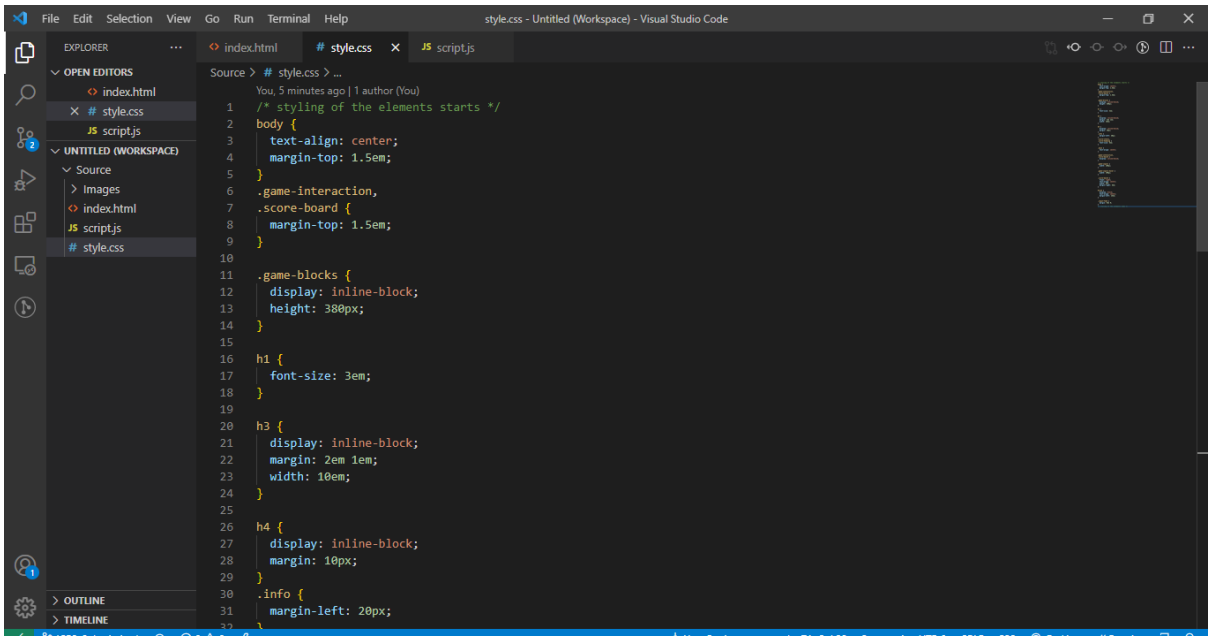
<h2 class="score-heading">Score</h2>

<h4>You</h4>
<h4>:</h4>
<h4>Computer</h4>

<h2 class="score-status">0 : 0</h2>

</section>
<!-- linking java script file -->
You, 4 minutes ago • added necessary comments
<script src="script.js" charset="UTF-8"></script>
</body>
</html>
ICP3-Submission*
You, 4 minutes ago
Ln 79, Col 10 (24 selected)
Spaces: 4
UTF-8
CRLF
HTML
Go Live
Prettier

Style.css:


File Edit Selection View Go Run Terminal Help
style.css - Untitled (Workspace) - Visual Studio Code
EXPLORER
OPEN EDITORS
index.html
style.css
script.js
UNTITLED (WORKSPACE)
Source
Images
index.html
script.js
style.css
OUTLINE
TIMELINE
index.html
style.css
script.js
Source > 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;

Script.js:

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.

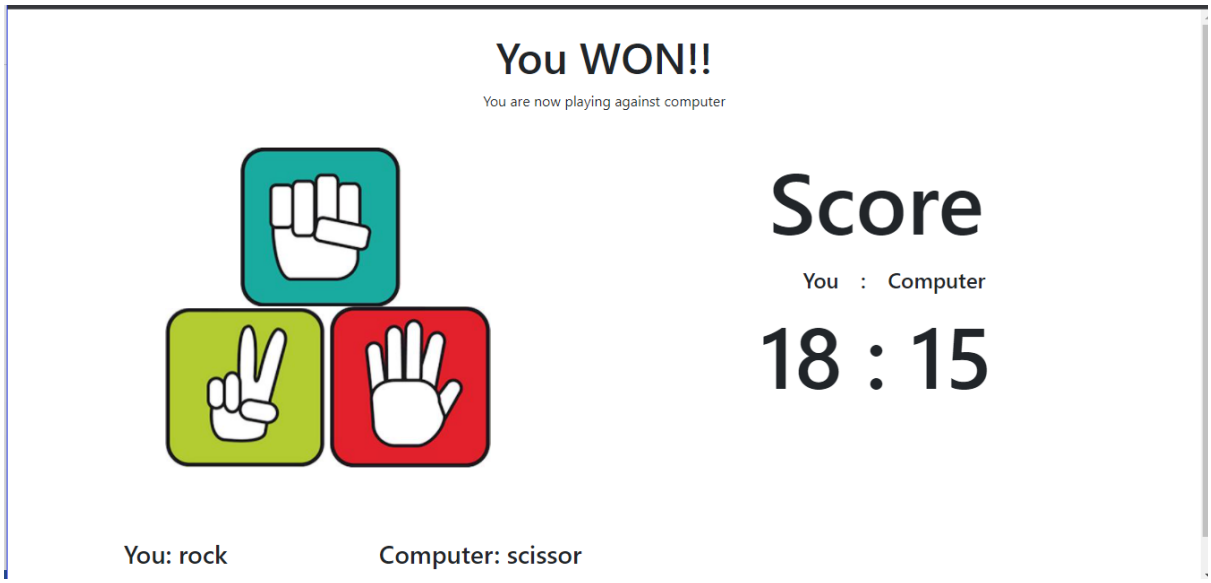
```
// random number generator for computer choice
function getcomputerInput() {
  const num = Math.floor(Math.random() * 3) + 1;
  if (num === 1) return "rock";       You, 11 hours
  else if (num === 2) return "paper";
  else return "scissor";
}
```

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.

```
// 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 and made body background colour to the WhiteSmoke colour.

```
<!doctype html>
<html lang="en">
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

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

Added the **media Query** to make page to the viewer as per the device screen.

```
/*the media query*/
@media screen and (min-width: 1000px) {
body {
    width: 100%;
       background: whitesmoke;
    box-sizing: border-box;
     margin: 0;
    padding: 0;
}
}
```

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:

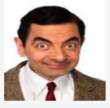
```css
/*profile block */
.profile {
    height: 110px;
    margin-top: 2px;
    color: rgb(224, 193, 43);
}
.profile-pic {
    height: 110px;
    width: 100%;
}
```

HTML code:

```html
<!--here we used container-->
<div class="container">

<!--the profile section with image to left and text to right
-->
    <div class="row profile">
        <div class="col-md-1">
            <img  src="Documentation/bean.jpg"
                class="img-responsive" alt="bean image missing" width="100" height="100">
        </div>
        <div class="col-md-11">
            <div class="profile text-right">
                <h1>Linga &&Sarath </h1>
                <p> Master's Student's </p>
            </div>
        </div>
    </div>
    <hr style="height:2px;border-width:0;color:gray;background-color:#fdec3b">
```

Output:



Linga &&Sarath

Master's Student's

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:

```css
/* Wrapper block */
.wrapper {
    height: 400px;
       background-size: cover;
    display: flex;
    justify-content: center;
    align-items: center;
    flex-direction: column;
    margin-bottom: 10px;
}
```

HTML Code:

```html
<!-- the wrapper section with an image-->
<div class="row">
    <div class="col-md">
        <div class="wrapper">
            <img  src="Documentation/Image.png"
                  class="img-responsive" alt="middle image missing" width="1140" height="350">
        </div>
    </div>
</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. On clicking the images takes to the respective tabs.

CSS code:

```css
/*the bottom images positioning and text front sizes*/

.col .image {
    width: 100%;
    height: 200px;
}
.image-heading {
    margin-top: 10px;
    text-align: center;
    font-weight: bold;
}

.last_set {
    margin-bottom: 10px;}
```

HTML code:

```html
<!-- dividing the row into 3 parts and assigning the image and links-->
<div class="row last_set">
    <div class="col-md">
        <a href="https://www.apple.com/app-store/">
            <img class="img-thumbnail"  width="555" height="300" src="Documentation/app1.png"></a>
        <h6 class="image-heading"> APP STORE </h6>

    </div>
    <div class="col-md">
        <a href="https://www.pexels.com/search/flowers/">
            <img class="img-thumbnail" width="555" height="300" src="Documentation/app2.png"></a>
        <h6 class="image-heading"> Flowers</h6>

    </div>
    <div class="col-md">
        <a href="https://weather.com/">
            <img class="img-thumbnail"  width="555%" height="300px" src="Documentation/app3.png"></a>
        <h6 class="image-heading"> Whether </h6>

    </div>
</div>
iv>
```

OutPut:

Sample links with images.

APP STORE

Flowers

Whether