ICP3 Web Programming

Task1: Rock, Paper, Scissors

In this task initially index.html file is created with the bootstrap css and minified js cdn links added into the head section of the document.

A screenshot of a computer

Description automatically generated

For the page layout a container div is created as below

**Page Layout**

Graphical user interface, text

Description automatically generated with medium confidence

Created a row, in that player and computer scores are displayed as in below snip. Initially these scores are set to 0. Nice css is added along with that for the buttons like padding: 10px

Used display: flex and space-around css to make the buttons on each end of the page.

Col class is used here so that two buttons will be each on 50% width of the page. Col is similar to col-xs-6 here because we used two col divisions.

**Score points at the top of the screen**

Text

Description automatically generated

CSS for the player score buttons is as below:

Text

Description automatically generated

In the second row I created a h3 tag with some header text

**Game play information**

Text

Description automatically generated

Full text and short text are displayed individually based on screen width.

CSS for this as below:

Text

Description automatically generated

**Business logic:**

When user clicks on any of the three buttons findresult method will be called.

Text

Description automatically generated

Alternative: Onclick functionality can be written directly in the javascript using add even listener as below:

Text

Description automatically generated

In this method computer choice will be generated by using Math.random() method. Math.random method generates a random number and It is built in javascript.

To generate random number formula is as below:

Math.random \* (max-min) -min

As we have three choices rock, paper and scissor: Math.random()\*3 will generate a random number between o to 2. With this number we will take an element from the choices array.

Text

Description automatically generated

Based on conditions here again output method is called by passing user choice and computer choice as parameters.

Using these parameters output screen is displayed on the page.

Text

Description automatically generated

If user wins result is displayed in green color. If loses result is displayed on red color. If there is a tie then the text color is shown as black.

There will a increment in individual scores of user and computer whoever wins.

**Output HTML:**

Text

Description automatically generated

**Logic for button bounce animation:**

A screenshot of a computer

Description automatically generated with medium confidence

CSS for the bounce button animation

Text

Description automatically generated

**Output of this task:**

Graphical user interface, application

Description automatically generated

**Mobile screen layout**

Graphical user interface, application, Word

Description automatically generated

**Task2:**

**Responsive web design**

Bootstrap cdn are added to the head section of the document

Text

Description automatically generated

**Layout**

Container is created which will be parent for the layout.

We can use container-fluid also if we want but the page width will be taken 100% here.

Logo is created in the first three columns of the row. And banner information is displayed to the right in the next 9 coumns.

Here is the HTML for this

**Page header and first row**

Text

Description automatically generated

CSS for caption

Text

Description automatically generated

Horizontal rule is created using <hr> tag

Text

Description automatically generated

In the second row image is shown using img-fluid class. This class will make the image use width 100% relative to parent and will be responsive to the parent

Text

Description automatically generated

In the next row featured work text is displayed with the font -family as loto

Chart, text

Description automatically generated with medium confidence

In the next row as I have to display three images in the 12 columns. Divided the 12 columns into 4 columns each so that each group will contain one image.

**Responsive design will be achieved with the help of these container, rowm column layout and with the help of bootstrap css.**

Text

Description automatically generated

Here is the HTML and css for the three images

Text

Description automatically generated

To make the images height equally I have created a img-flg div and in that image height is taken 100%. Because of this if we vary height of img-flg css all the images height will be varied accordingly.

Mobile screen:

Text

Description automatically generated

Laptop screen:

Text, letter

Description automatically generated