Group Assignment –

Multiplication Game Website (Using HTML, CSS, and JavaScript)

Authors: Stephen Burnett, Travis Callam, Kaylen Eastwood, Paris Scott, Iyana Taylor

UNIVERSITY OF TECHNOLOGY, JAMAICA

Faculty of Engineering and Computing

School of Computing and Information Technology

CIT2011: Web Programming

Tutor: Monique Daubon

Due Date: November 30, 2023

Table of Content

Storyboard/ Sitemap

Project Plan

Workplan

XHTML Validation Report

Link to Hosting

Signed Authorization Forms

nav to each section

GAME NAME

Registration Play Game Results

Hello player, Register here

First Name
Last Name
Birth Date
Gender
Age
Email

Register Start

Play Game!

How to play:

Answer status shown here:

Your Answer

Check Next End

Statistics

Your Stats

Percentage Score

All Players' Stats

FN	LN	Age	Gender	Total Questions	Total Correct Answers	Total Incorrect Answers

Charts

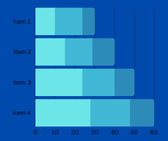


Chart showing (gender)

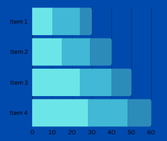


Chart showing (age)

Footer

fonts

Poppins

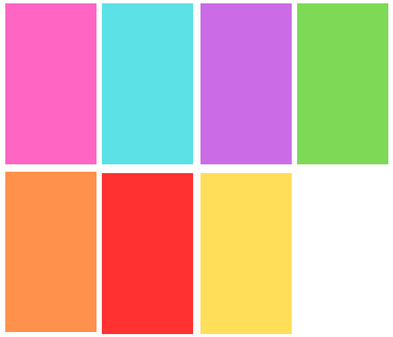
How to play:

Inspiration from Retro Gameboy

icons for each section

colours

chart images



Project Plan

Stephen Burnett

Tasks 1 - 3

DEFINED FUNCTIONS

function register()

function disableForm()

function enableButtons()

Travis Callam

Task 4 - 6

DEFINED FUNCTIONS

function playGame()

function checkAnswer()

function appendSessionStorage()

Paris Scott

Tasks 7 - 9

DEFINED FUNCTIONS

function numOfCheckClicks()

function enableRegForm()

function currentDateTime()

Iyana Taylor

Tasks 10 - 12

DEFINED FUNCTIONS

function findPercentageScore()

function printCurrentPlayerInfo()

function disableplayAreaFormAndButtons()

Kaylen Eastwood

Tasks 13 - 15

DEFINED FUNCTIONS

function showAllStats()

function countFemalePlayers()

function countMalePlayers()

Remaining Tasks 16 - 18 were shared

DEFINED FUNCTIONS

function countGenderPlayers()

function femalePlayerPercent()

function malePlayerPercent()

function countAllAgePlayers()

function agePercents()

Workplan

Stephen Burnett

```
function register()
```

The `register` function handles user registration by retrieving input data, validating key fields, storing user information in session storage, manipulating the registration form, and providing user feedback. It ensures a smooth and controlled registration process with concise steps.

```
function disableForm()
```

The `disableForm` function disables all elements within the "regForm" by retrieving form elements and setting their `disabled` property to `true`. Specifically, it ensures that the submit button ("registerBtn") is also disabled. This functionality is commonly applied post form submission to restrict additional user input.

```
function enableButtons()
```

The `enableButtons` function activates four buttons on the webpage: "start," "end," "checkAnsBtn," and "nextEqBtn," by setting their `disabled` property to `false`. This enables user interaction and is generally invoked after a specific event or condition warrants the activation of these buttons.

Travis Callam

```
function playGame()
```

The `playGame` function manages the core functionality of the game. It resets user input by clearing the "yourAnswer" input field and the "ans" element. It then generates two random integers and displays them in the appropriate input fields. The product of these numbers is calculated and stored in the `product` variable. The function concludes without returning a specific value.

```
function checkAnswer()
```

The `checkAnswer` function evaluates the user's response in the game by incrementing a click counter (to count the total questions), retrieving the user's input, and checking it against the calculated product. Feedback is provided, and counters (`correctCounter` or `incorrectCounter`) are updated accordingly. The function concludes without returning a specific value and is important for assessing user input and updating game statistics.

```
function appendSessionStorage()
```

The `appendSessionStorage` function combines and updates arrays between session and local storage. It retrieves, validates, and parses the session storage array (`currentPlayer`), as well as the local storage array (`playerRegistrationData`). The arrays are combined, and

the updated array is stored in local storage. The function logs success or absence of arrays to the console, making maintaining user registration data across sessions easy.

Paris Scott

```
function numOfCheckClicks()
```

The `numOfCheckClicks` function tallies clicks on the "Check Answer" button. It initializes a global `clickCounter` variable to monitor button clicks, sets up a click event listener on the button, and increments the counter within the listener, providing a straightforward mechanism to track the number of questions completed by a user.

```
function enableRegForm()
```

The `enableRegForm` function re-enables all elements within the registration form with the ID "regForm." It achieves this by retrieving form elements and iterating through them to set their `disabled` property to `false`. Generally employed post-conditions like those in the `findPercentageScore` function, it restores user interactivity to the registration form.

```
function currentDateTime()
```

The `currentDateTime` function handles the display and logging of the current date and time. It creates a JavaScript `Date` object, converts it to a locale-specific string using `toLocaleString`, and updates the content of the HTML element with the ID "dateTimeArea." Simultaneously, it logs the formatted date and time to the console, providing the current date and time for the user.

Iyana Taylor

```
function findPercentageScore()
```

The `findPercentageScore` function manages various tasks, including disabling/enabling forms, displaying date and time, printing player info, calculating/displaying percentage scores, showing statistics, and setting intervals for certain functions like, `femalePlayerPercent(), malePlayerPercent(), agePercents()`.

```
function printCurrentPlayerInfo()
```

The `printCurrentPlayerInfo` function updates the content of the "yourStatsArea" textarea to display information about the current player, including click, correct, and incorrect counters. It achieves this by appending these counter values to the `currentPlayer` array, converting the array to a JSON string, and storing it in session storage under the key "currentPlayer." The function then retrieves and parses the session storage array, updating the textarea with the player's information. In case the session storage array is not found, a console message is logged.

```
function disableplayAreaFormAndButtons()
```

The `disableplayAreaFormAndButtons` function disables form elements and buttons within the "playAreaForm." It achieves this by retrieving all form elements through the `elements` property, iterating through them, and setting the `disabled` property to `true`. Specifically targeting three buttons ("start," "checkAnsBtn," and "nextEqBtn"), the function ensures these buttons are also disabled. This mechanism serves to restrict user interaction during specific phases or conditions, as indicated by its call within the `findPercentageScore` function.

Kaylen Eastwood

```
function showAllStats()
```

The `showAllStats` function populates an HTML table with player statistics, utilizing data retrieved from local storage. It begins by calling the `appendSessionStorage` function to consolidate information from session and local storage. The HTML content of an element with the ID "showAllPlayers" is reset. It then retrieves and parses the local storage array (`playerRegistrationData`). Subsequently, HTML table rows are generated, each containing 9 elements based on the data from the array. These rows are appended to the HTML table with the ID "showAllPlayers." The function logs a success message to the console if the table is populated successfully and provides a message in case the local storage array is not found. In essence, it dynamically displays player statistics in an organized table format on the webpage.

```
function countFemalePlayers()
```

The `countFemalePlayers` function tallies the number of female players in the local storage array, following a structure like the `countMalePlayers` function. It initiates by calling the `appendSessionStorage` function to update data from session and local storage. Subsequently, it retrieves and parses the local storage array (`playerRegistrationData`). The function then iterates through this array, incrementing a `femaleCounter` variable for each occurrence of the value "female." The final count is logged to the console. In case the local storage array is not found, a corresponding message is logged.

```
function countMalePlayers()
```

The `countMalePlayers` function calculates the number of male players within the local storage array by initially invoking the `appendSessionStorage` function to synchronize and update data. Following this, it retrieves and parses the local storage array (`playerRegistrationData`). The function proceeds to iterate through this array, incrementing a `maleCounter` variable for each instance of the value "male." The final count is then logged to the console. If the local storage array is not found, a corresponding message is logged.

Remaining Functions

```
function countGenderPlayers()
```

The `countGenderPlayers` function counts players, irrespective of gender, within a local storage array by initiating the `appendSessionStorage` function for data synchronization. It retrieves and parses the local storage array (`playerRegistrationData`), then iterates through the array, incrementing a `genderCounter` variable for each occurrence of "male" or "female." The ultimate count of players, regardless of gender, is logged to the console. In the absence of the local storage array, a corresponding message is logged.

```
function femalePlayerPercent()
```

The `femalePlayerPercent` function calculates the percentage of female players among all players by utilizing the counts obtained from the `countFemalePlayers` and `countGenderPlayers` functions. It then dynamically adjusts the width of an HTML container with the ID "showChartsForGirls" to visually represent this percentage. The function creates a new variable (`newWidth`) to store the calculated female percentage, retrieves the target HTML container, and updates its content with an image tag, setting the width to the calculated percentage. Overall, the function provides a visual representation of the female player percentage based on the gender distribution in the array.

```
function malePlayerPercent()
```

The `malePlayerPercent` function calculates the percentage of male players among all players by leveraging the counts obtained from the `countMalePlayers` and `countGenderPlayers` functions. It dynamically adjusts the width of an HTML container with the ID "showChartsForBoys" to visually represent this percentage. The function creates a new variable (`newWidth`) to store the calculated male percentage, retrieves the target HTML container, and updates its content with an image tag, setting the width to the calculated percentage. Overall, the function provides a visual representation of the male player percentage based on the gender distribution in the array.

```
function countAllAgePlayers()
```

The `countAllAgePlayers` function counts the number of players falling within the age range of 8 to 12 years old by utilizing the counts obtained from the `appendSessionStorage` function and iterating through the local storage array. It increments the `allAgeCounter` variable for each player within the specified age range and logs the final count to the console. The function assumes that the age information is represented by numeric values in the local storage array and handles logging for both the presence and absence of the array.

```
function agePercents()
```

The `agePercents` function calculates the percentage of players within the ages 8 to 12 range, leveraging counts obtained from the `countAllAgePlayers` function and iterating through the local storage array. It dynamically adjusts the width of containers, visually representing these percentages using images. The function iterates through the age values using a `switch` statement, increments specific age counters, calculates percentages, and

updates container content with image tags. The function handles both the presence and absence of the local storage array, logging appropriately.

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for index.html

Checker Input

Show ☐ source ☐ outline ☐ image report Options…

Check by file upload Choose File No file chosen

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

Check

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Message Filtering

1. **Error** **Attribute** `name` **not allowed on element** `meta` **at this point.**

   From line 5, column 9; to line 5, column 95

   ```
   >↩        <meta charset="utf-8" name="viewport" content="width=device-width,
   initial-scale=1.0"/>↩
   ```

   Attributes for element `meta`:

   Global attributes

   `name` — Metadata name

   `http-equiv` — Pragma directive

   `content` — Value of the element

   `charset` — Character encoding declaration

   `media` — Applicable media

2. **Error** **Attribute** `content` **not allowed on element** `meta` **at this point.**

   From line 5, column 9; to line 5, column 95

   ```
   >↩        <meta charset="utf-8" name="viewport" content="width=device-width,
   initial-scale=1.0"/>↩
   ```

   Attributes for element `meta`:

   Global attributes

   `name` — Metadata name

   `http-equiv` — Pragma directive

   `content` — Value of the element

`charset` — Character encoding declaration

`media` — Applicable media

3. **Info** **Trailing slash on void elements has no effect and interacts badly with unquoted attribute values.**

   From line 5, column 9; to line 5, column 95

   ```
   >↩        <meta charset="utf-8" name="viewport" content="width=device-width,
   initial-scale=1.0"/>↩
   ```

4. **Warning** **The `type` attribute is unnecessary for JavaScript resources.**

   From line 10, column 9; to line 10, column 59

   ```
   >↩        <script type="text/javascript" src="new-script.js"></scri
   ```

5. **Error** **The element `label` must not appear as a descendant of the `label` element.**

   From line 67, column 21; to line 67, column 38

   ```
            <label for="male">Male</
   ```

6. **Error** **The `label` element may contain at most one `button`, `input`, `meter`, `output`, `progress`, `select`, or `textarea` descendant.**

   From line 69, column 21; to line 69, column 90

   ```
            <input type="radio" id="female" name="gender" value="female"
   required>↩
   ```

7. **Warning** **`label` element with multiple labelable descendants.**

   From line 64, column 17; to line 64, column 23

   ```
            <label>↩
   ```

8. **Error** **The element `label` must not appear as a descendant of the `label` element.**

   From line 70, column 21; to line 70, column 40

   ```
            <label for="female">Female
   ```

Document checking completed.

Used the HTML parser.

Total execution time 10 milliseconds.

About this checker • Report an issue • Version: 23.11.28

Link to Hosting

https://iyanataylor1.github.io/circuitcruncher/

UNIVERSITY OF TECHNOLOGY, JAMAICA
Declaration of Authorship

FACULTY: Faculty of Engineering and Computing

School/Department: School of Computing and Information Technology

Course Code & Title: CIT2011 Web Programming

Submitted To: Mrs. Monique Daubon
(Lecturer/Supervisor)

Submitted By: Stephen Burnett
Student's name

2210182
ID Number

Zinnea Circle, Rosemount Gardens, Mount Salem, Montego Bay,St.James
Address

876-229-2189
Contact telephone numbers (home, work, cell)

Date of Submission: Thursday, November 30, 2023

Title of Assignment: Web Propgramming Group Project

****

Declaration: I certify that I am the author of this paper and that any assistance I received in its preparation is fully acknowledged and disclosed in the paper. I have also cited all sources from which I used visuals, data, ideas or words, either quoted directly or paraphrased. I also certify that this paper was prepared by me specifically for this course. I also understand that a grade will not be assigned without the submission of this agreement.

Student's Signature: S.Burnett

****

Lecturer's/Supervisor's Grade for Assignment: _____

Lecturer's/Supervisor's Comments: _____

Note: For group assignments each student is required to complete a separate Declaration of Authorship.

***Ref: Regulation 5**: Conditions and Procedures Governing Student Academic Misconduct*

UNIVERSITY OF TECHNOLOGY, JAMAICA
Declaration of Authorship

FACULTY: Faculty of Engineering and Computing

School/Department: University of Technology, Jamaica

Course Code & Title: CIT2011, Web Programming

Submitted To: Mrs. Monique Daubon
(Lecturer/Supervisor)

Submitted By: Travis Callam
Student's name

2209431
ID Number

Lambs Rivier, Bethel Town P.O., Westmoreland
Address

(876) 458-7998
Contact telephone numbers (home, work, cell)

Date of Submission: Thursdays November 30, 2023.

Title of Assignment: Web Programming Group Project

_____

****

Declaration: I certify that I am the author of this paper and that any assistance I received in its preparation is fully acknowledged and disclosed in the paper. I have also cited all sources from which I used visuals, data, ideas or words, either quoted directly or paraphrased. I also certify that this paper was prepared by me specifically for this course. I also understand that a grade will not be assigned without the submission of this agreement.

Student's Signature: T. Callam

****

Lecturer's/Supervisor's Grade for Assignment: _____

Lecturer's/Supervisor's Comments: _____

_____

Note: For group assignments each student is required to complete a separate Declaration of Authorship.

Ref: Regulation 5*: Conditions and Procedures*

Governing Student Academic Misconduct

Division of Student Services & Registry
Ac. Brd. Sept 23, 2004

UNIVERSITY OF TECHNOLOGY, JAMAICA
Declaration of Authorship

FACULTY: Engineering and Computing

School/Department: Computing and Information Technology

Course Code & Title: CIT2011 Web Programming

Submitted To: Ms. Monique Daubon
(Lecturer/Supervisor)

Submitted By: Kaylen Eastwood
Student's name

22010194
ID Number

Chantilly R.d., Savanna la Mar P.O., Westmoreland.
Address

876 559 - 7734
Contact telephone numbers (home, work, cell)

Date of Submission: November 29, 2023.

Title of Assignment: CIT2011 Group Project SEM1 AY23-24

_____

****

Declaration: I certify that I am the author of this paper and that any assistance I received in its preparation is fully acknowledged and disclosed in the paper. I have also cited all sources from which I used visuals, data, ideas or words, either quoted directly or paraphrased. I also certify that this paper was prepared by me specifically for this course. I also understand that a grade will not be assigned without the submission of this agreement.

Student's Signature: K. Eastwood

****

Lecturer's/Supervisor's Grade for Assignment: _____

Lecturer's/Supervisor's Comments : _____

_____

Note: For group assignments each student is required to complete a separate Declaration of Authorship.

***Ref: Regulation 5**: Conditions and Procedures Governing Student Academic Misconduct*

UNIVERSITY OF TECHNOLOGY, JAMAICA
Declaration of Authorship

FACULTY: Faculty of Engineering and Computing

School/Department: School of Computing and Information Technology

Course Code & Title: CIT2011 Web Programming

Submitted To: Mrs. Monique Daubon
(Lecturer/Supervisor)

Submitted By: Paris Scott
Student's name

22110191
ID Number

Lot 821 Barret Hall, Lilliput, Montego Bay, St. James
Address

876-825-3933
Contact telephone numbers (home, work, cell)

Date of Submission: Thursday, November 30, 2023

Title of Assignment: Web Programming Group Project

****

Declaration: I certify that I am the author of this paper and that any assistance I received in its preparation is fully acknowledged and disclosed in the paper. I have also cited all sources from which I used visuals, data, ideas or words, either quoted directly or paraphrased. I also certify that this paper was prepared by me specifically for this course. I also understand that a grade will not be assigned without the submission of this agreement.

Student's Signature: P. Scott

****

Lecturer's/Supervisor's Grade for Assignment: _____

Lecturer's/Supervisor's Comments: _____

_____

Note: For group assignments each student is required to complete a separate Declaration of Authorship.

***Ref: Regulation 5**: Conditions and Procedures*

Governing Student Academic Misconduct

Division of Student Services & Registry
Ac. Brd. Sept 23, 2004

UNIVERSITY OF TECHNOLOGY, JAMAICA
Declaration of Authorship

FACULTY: Faculty of Engineering and Computing

School/Department: School of Computing and Information Technology

Course Code & Title: CIT2011 Web Programming

Submitted To: Mrs. Monique Daubon
(Lecturer/Supervisor)

Submitted By: Iyana Taylor
Student's name

2209566
ID Number

Burnt Ground, Ramble P.O., Hanover
Address

876-213-8437
Contact telephone numbers (home, work, cell)

Date of Submission: Thursday November 30, 2023

Title of Assignment: Web Programming Group Project - Multiplication Website

****

Declaration: I certify that I am the author of this paper and that any assistance I received in its preparation is fully acknowledged and disclosed in the paper. I have also cited all sources from which I used visuals, data, ideas or words, either quoted directly or paraphrased. I also certify that this paper was prepared by me specifically for this course. I also understand that a grade will not be assigned without the submission of this agreement.

Student's Signature: I. Taylor

****

Lecturer's/Supervisor's Grade for Assignment: _____

Lecturer's/Supervisor's Comments: _____

_____

Note: For group assignments each student is required to complete a separate Declaration of Authorship.

***Ref: Regulation 5**: Conditions and Procedures Governing Student Academic Misconduct*

Division of Student Services & Registry
Ac. Brd. Sept 23, 2004