IT312 Web Applications Engineering
Second Trimester of 2022-2023
Duration: 3:00 hours

جـــامــــعــة
الملك سعود
King Saud University

كلية علوم الحاسب والمعلومات
قسم تقنية المعلومات

Final Lab /20

General Instructions:

- Download all the provided files: **index.html**, **style.css**, and **javascript.js**.

- Before reading the questions, open all these files and **read the code**.

- Write your name and ID in the beginning of each file, then start modifying each file as described below in each part.

- Allowed materials: Lecture Slides only (i.e. Week 7 DOM external examples are not allowed), W3 Schools and W3C Validator.

- Write only inside the marked area in the files, and always refer to the exam paper and the provided output for details.

Add your code in the provided files: **index.html**, **style.css**, and **javascript.js**. To add your code to their corresponding files, follow the requirements listed in each part (Part 1: HTML, Part 2: CSS, and Part 3: JavaScript).

You will be developing a word guessing game. The game gives the player six rounds to guess a specific word (a 5-letter word). In each round, the result of each guessed letter is shown in a table above, where:

- If the letter is guessed correctly, and in the right position, then this letter tile will be colored green.

- If the letter is guessed correctly, but not in the correct position, then this letter tile will be colored gold.

- If the letter is not in the word, then this letter tile will be colored dark gray.

    For example: If the correct word was "HUMAN", and the user guess was "HEART", then the result will be shown as follows:
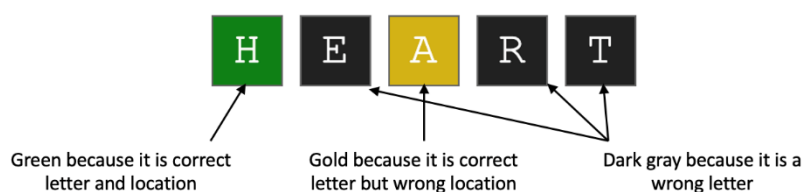
- In the beginning, the game randomly selects a specific word from a list of words.

- The game allows the player to reset at any time using RESET button, which causes the game to randomly select a new word and start-over.

Part 1: HTML   /5

Modify <u>**index.html**</u> file by filling each "HTMLQ#" part. Complete the form, and the footer of the page:

Modify Form:

1. **HTMLQ1:**

   - Add name to the form.

2. **HTMLQ2:**

   - Modify the form to include:

     1. Five text inputs with class (txt) that accept only one character.
     2. An input as a button (ENTER) with ID.
     3. An input as a button (RESET).

Modify Page Footer:

3. **HTMLQ3:**

   - Add an unordered list with these three items:

     1. "Final Lab Exam 2023".
     2. "My Name is (Add Your Name)".
     3. "ID: (Add Your Student ID)" .

Part 2: CSS   /5

Modify <u>**style.css**</u> by filling each "CSSQ#" part. The below **wireframe** clarifies the <u>positioning</u> of the elements:

Modify HTML to link CSS file:

1. **CSSQ1:**

   - Link **style.css** file.

Modify CSS file:

2. **CSSQ2:**

   - Apply the provided style declaration to the header, footer, main and aside.

3. **CSSQ3:**

   - Add a specific style rule to the aside:
     - It stands to the left of `<main>` element (see the wireframe below).
     - It has a width that is proportionate with the `<main>` width (main width is 82%).
     - The padding is 15px.

4. **CSSQ4:**

   - Add a specific style declaration to the footer:
     - Make the footer bolder.
     - Add background image to the footer: "bg.png".

5. **CSSQ5:**

   - The table should be in the center of the `<main>` element.

6. **CSSQ6:**

   - Without using **id** or **class**, apply the provided style declaration to the `<td>` and the button `<input>` when **the mouse goes over them.**

7. **CSSQ7:**
   - Apply the provided style declaration to class "txt".

8. **CSSQ8:**
   - Apply the following style to the element with id "button":
     o Make the font's size large.

9. **CSSQ9:**
   - Without using **id** or **class**, apply the following styles to the button <input>**:**
     o Using a percentage value, make the width occupies the whole width of the parent element.
     o Make it half transparent.
     o The border corners are rounded by 0.625rem.
     o The line's height is 2rem.

10. **CSSQ10:**
    - Apply the following style to <ul>:
      o Make the text dark gray with 0.4 opacity.
      o Set the items marker to square.
      o Set the items marker to the inside of the list.

```
HEADER
┌──────────────────────────────────────────────┐
│ H1                                           │
└──────────────────────────────────────────────┘
ASIDE        MAIN
                  ┌─────────────────────────────┐
                  │ TABLE                       │
                  │                             │
                  │                             │
                  │                             │
                  └─────────────────────────────┘
                  ┌─────────────────────────────┐
                  │ FORM                        │
                  │ ┌─────┐┌─────┐┌─────┐┌─────┐┌─────┐│
                  │ │INPUT││INPUT││INPUT││INPUT││INPUT││
                  │ └─────┘└─────┘└─────┘└─────┘└─────┘│
                  │ ┌───────────────────────────┐│
                  │ │         ENTER             ││
                  │ └───────────────────────────┘│
                  │ ┌───────────────────────────┐│
                  │ │         RESET             ││
                  │ └───────────────────────────┘│
                  └─────────────────────────────┘
FOOTER
┌──────────────────────────────────────────────┐
│ UL                                           │
└──────────────────────────────────────────────┘
```

Part 3: JavaScript /10

Modify **index.html** and **javascript.js** by filling each "JSQ#" part:

Modify index.html to link JS file and call JS functions:

1. **JSQ1:**

   - Link **javascript.js** file.

2. **JSQ2:**

   - Call a function `setName()` when the page is loaded.

3. **JSQ3:**

   - Call a function `play()` when ENTER button is clicked.

   - Call a function `resetGame()` when RESET button is clicked.

Modify javascript.js:

Function `setName()`:

4. **JSQ4:**

   Write the content of the function `setName()` to do the following:

   - Pop up a window to ask the player to enter his/her name and store it in `pName`.

   - In the aside, add the name and a horizontal line.

Function `play()`:

This function is called when ENTER button is clicked, and it performs the following in order:

- It checks every letter in the word entered by the player and displays the guessing results in the above table. For each letter:
  - It takes the letter from the form input.
  - Put it in the corresponding cell in the table.
  - If it is the right letter in the right position, it will color the cell background to green and increment `winCounter`.
  - If not, it will test if the letter is found anywhere else in the word, then it will color the cell background to gold, else it means the letter is not in the word, so it will color the cell background to dark gray.

- It checks if all the letters were guessed correctly (the player wins and the game ends):
  - It checks `winCounter`. If it reached 5, it means the player won the game.

- It checks if the player used up all the six rounds (the player lost and the game ends).
  - It will check `roundCounter`. If it reached 6, it means the player lost the game.

5. **JSQ5:**

   Read the entered letter and display it in the corresponding table cell, to do this:

   - Store the ith text input value (as an uppercase) in `currentLetter`.

   - Store the ith `<td>` object in `targetTD` (hint: you may need `roundCounter` to index the table cell).

   - Copy the letter from `currentLetter` to the inside of `targetTD`.

6. **JSQ6:**

    Check if the entered letter matches the `choosenWord` at the corresponding position, to do this:

    - Compare the `currentLetter` to the ith letter of the `choosenWord`.
    - If they are identical:
        a. set the background color for `targetTD` to "green".
        b. increment `winCounter`.

7. **JSQ7:**

    Check if the entered letter is in any other position of `choosenWord`, to do this:

    - Test if the `currentLetter` is in the `choosenWord`.
    - If yes, set the background color for `targetTD` to gold (#d2b201).
    - else, set the background color for `targetTD` to dark gray (use hex color).

8. **JSQ8:**

    Check if the player won the game, to do this:

    - Checks `winCounter`. If it reached 5, it means the player won the game.
    - Pop up a window displaying "YOU WIN" and in a new line displays the guessed word.
    - Disable the ENTER button.

9. **JSQ9:**

    Check if the player lost the game (if `roundCounter` reached 6), to do this:

    - In the aside, add "GAME OVER :(" with italic font under the horizontal line.
    - Disable the ENTER button.

Function `resetGame()`:

This function is called when RESET button is clicked, and it performs the following in order:

- Both **roundCounter** and **winCounter** are reset to Zero. (Code for this is already given).
- A new random word will be stored in **chosenWord.** (Code for this is already given).
- The ENTER button will be enabled.
- The aside content will be updated to include only the name and the horizontal line.
- All table cells will be cleared from their content and their background colors.
- All input texts will be cleared from their content (Code for this is already given).

10. **JSQ10:**

    - Enable the ENTER button.
    - In the aside, update its content to include only the name and the horizontal line.

11. **JSQ11:**

    - Using a loop, clear all `<td>`s from their contents, and return their background color to the 'initial' color.

Part 4: Handing your final lab solution

When you finish, make sure to:

1. Create folder in your **Desktop** called "**S#_Name_IT312_Final_Lab**"
2. Put all modified HTML, CSS and JS files into this folder.
3. Compress the folder.
4. Open your **LMS** account then **upload** the compressed folder.

Part 5: Output print screens

After entering the player's name:

After playing 6 rounds and the player didn't guess the right word (PIXIE):

After RESET:

If the player guessed the right word:

Good Luck.