Assignment 1: Quiz App

***IMPORTANT**: This assignment must be done individually.*

Read Section A to understand the programming requirements, Section B to understand the programming tasks that you need to carry out and Section C to know what you need to submit as the result.

A. Description

In this assignment, you will create a customizable *one-page* "**Quiz App**"

1. Overall appearance and behavior

The following video shows the look and behavior of the quiz we are asking you to implement.

https://drive.google.com/file/d/1C__g40l3Piti0DUDGQ0io1ygfEydr735/view?usp=sharing

The app has 3 screens, including

- read some information about the quiz → **start quiz** →
- answer quiz questions → **submit your answers** →
- review your answers

Since this is a 10-question **one-page** quiz app, our strategy to organize the page as below:

course name	stack
Quiz name	
Screen 1: Introduction	
Screen 2: Attempt quiz (hidden at the beginning)	
Screen 3: Review quiz (hidden at the beginning)	

One detail that is hard to see in the video:

- To select the answer, we can click the radio box, text or anywhere into the option

- After submitting your answers (review mode), it should *not* be possible to click or change your answer anymore.

B. Task requirements

1. General instructions

IMPORTANT: Strictly follow the use of **tag, text** & naming of **id, class…** as highlighted below, or else no score given for those sections.

1. Download the *starter_pack* for folder structure suggested to you. We expect you will have to make modifications to the following files to complete the assignment:

   - `index.html`: Write your HTML here (`<head>` section has been ***partially*** completed for you)
   - `style.css`: Write your CSS here (pay attention about the mobile layout)
   - `script.js`: Write your JavaScript here.
     - This is the file in which we expect you to implement the quiz behavior.
     - You should define and attach event listeners in this file.

2. HTML: Copy and paste the text of *index-content.txt* into your *index.html* file, then add the HTML tags necessary to style the page. Later you will likely need to make some modifications for the mobile layout and the JavaScript.

Note: create all 3 screens before hiding them using JavaScript

3. CSS: in the `style.css` file,

   - Add appropriate CSS to style your page
   - Add appropriate viewport & CSS media queries to support the mobile view.

Note: style all 3 screens before hiding them using JavaScript

4. JavaScript: in the `script.js` file, to accomplish quiz behavior

   - Screen 1: Introduction
     - Handle user action: starting the quiz & displaying questions on *Screen 2*
   - Screen 2: Attempt quiz

    - Handle user action: selecting an answer
    - Handle user action: changing an answer
    - Handle user action: submitting your answers & display result, review questions on *Screen 3*
- Screen 3: Review quiz
    - Handle user action: restart the quiz

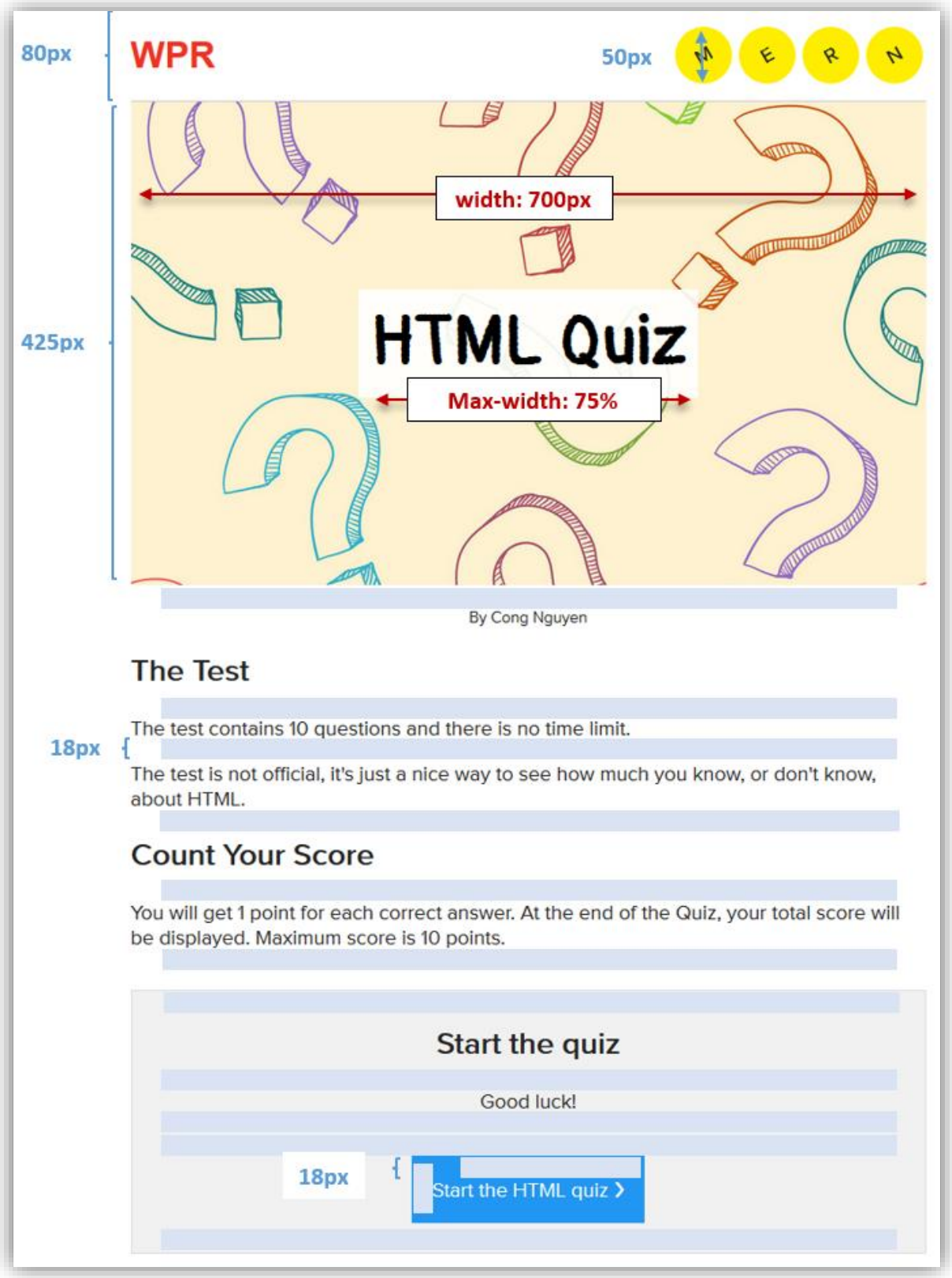
2. Check point 1: HTML/CSS (1) screen 1, 2

In this check point, you will complete the HTML/CSS for

- shared elements (header, quiz name, etc.) and
- 2 first screens of the app: **introduction**, **attempt quiz**

<table>
<tr><th>Week</th><th>HTML</th><th>CSS</th><th>JS</th></tr>
<tr><td rowspan="2">1</td><td colspan="2">- shared elements (header, quiz name, etc.)
- screen 1: introduction
- screen 2: attempt quiz
+ Question name & options</td><td></td></tr>
<tr><td colspan="3">- test your application (expected elements & styles)</td></tr>
</table>

Measurements

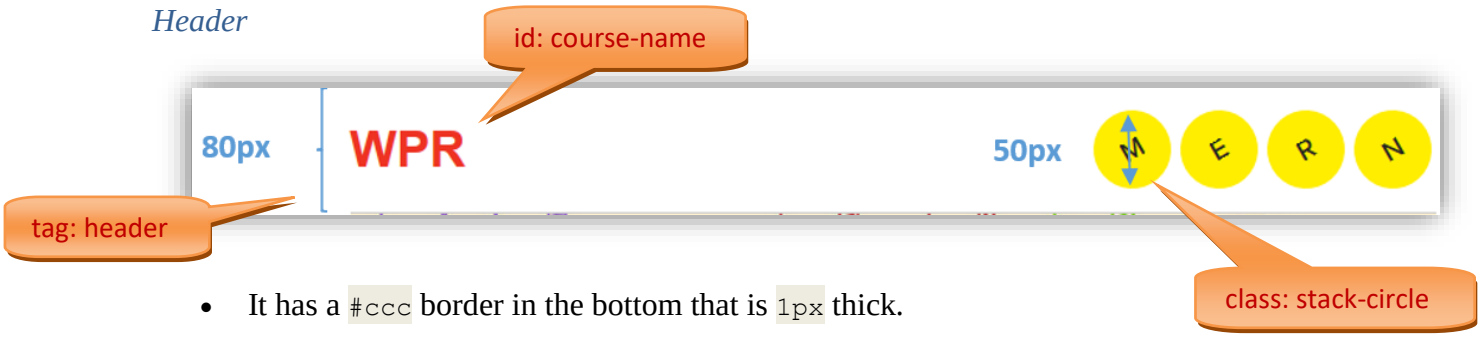


Shared

Common

- The font face is `'Proxima Nova'`, and the fallback fonts are, in order, `'Helvetica'`, `'Arial'`, `sans-serif`.
- The font size is `18px`.
- The font color is `#222222`

Header



- It has a `#ccc` border in the bottom that is `1px` thick.

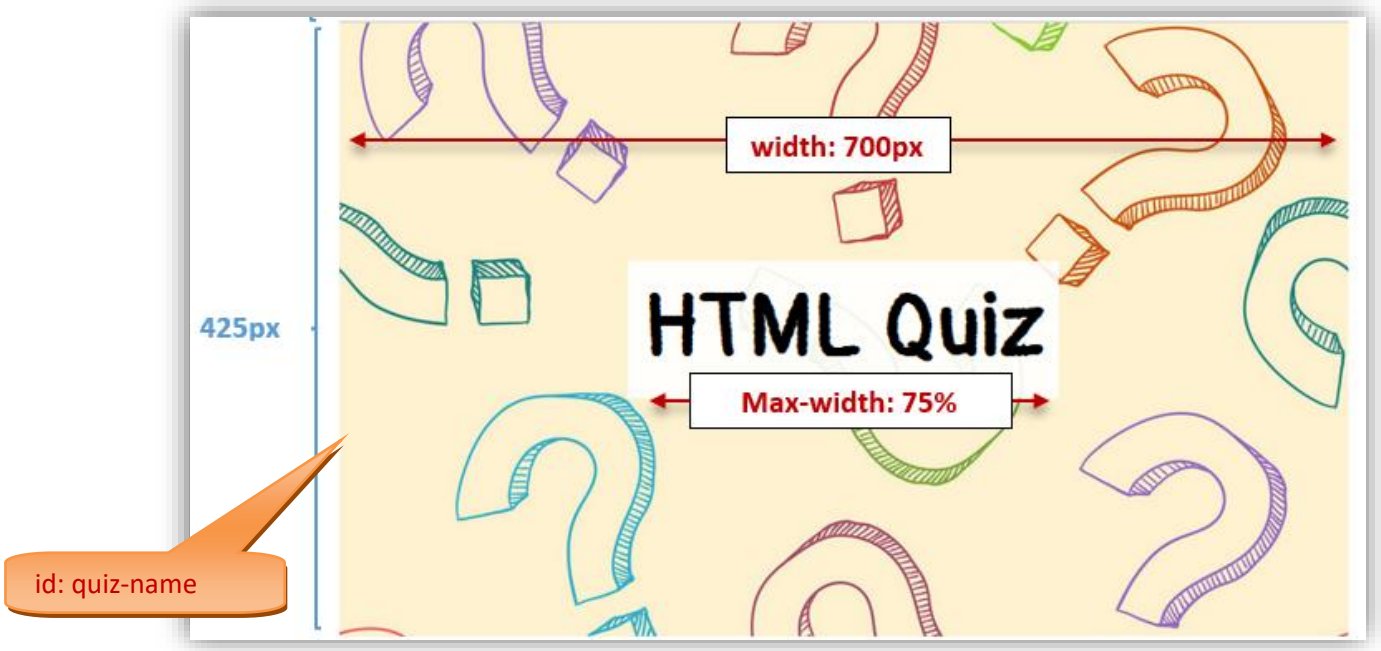
Course name

- The font face is `'Helvetica',` and the fallback font is `sans-serif`.
- The font size is `32px`.
- The font weight is `bold`.
- The font color is `#ee3322`.

MERN Stack (Yellow circles)

- The background color is `#ffee00`
- They have rounded corners with radius `100%`.
- The font weight is `bold`.
- The text is vertical, and center aligned to the circle.
- They are rotated `30 degrees` to the left. **Hint**: see css properties `transform` w3schools

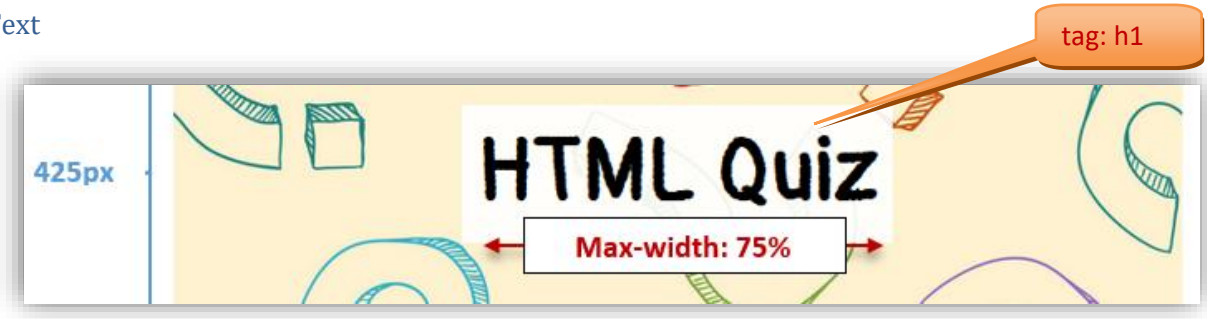
Quiz name



Banner

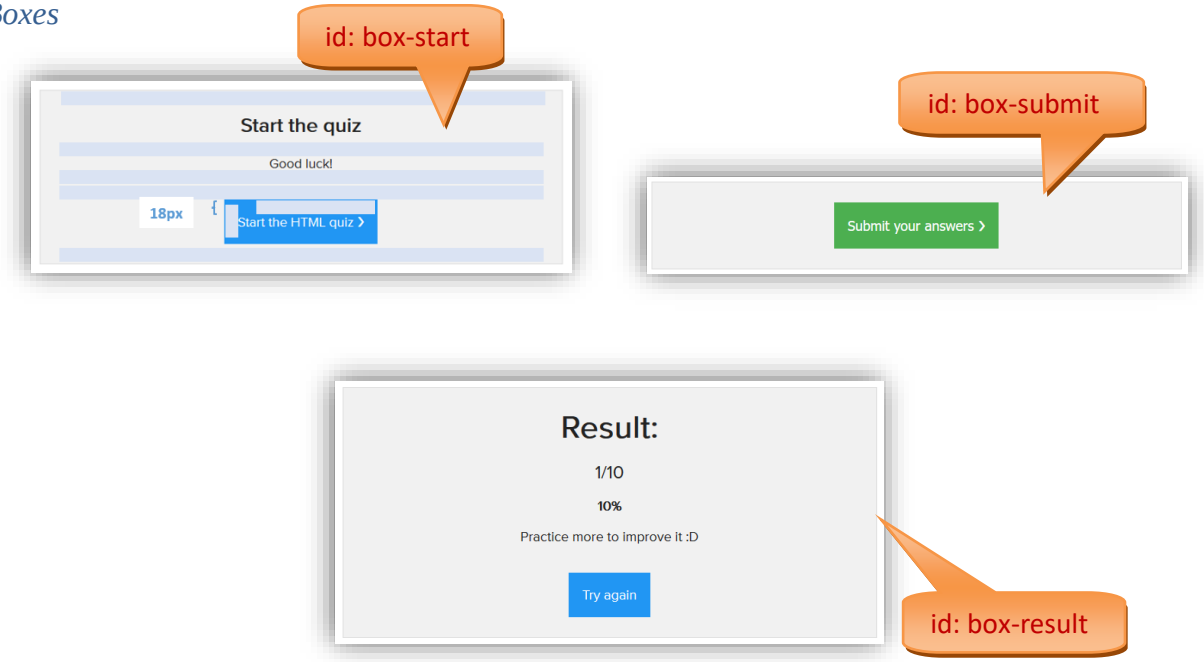
- Quiz banner has a background image (`images/background.jpg`).
- Its height is `425px`.
- The background size is set to cover
- The background position is anchored in the middle

Text



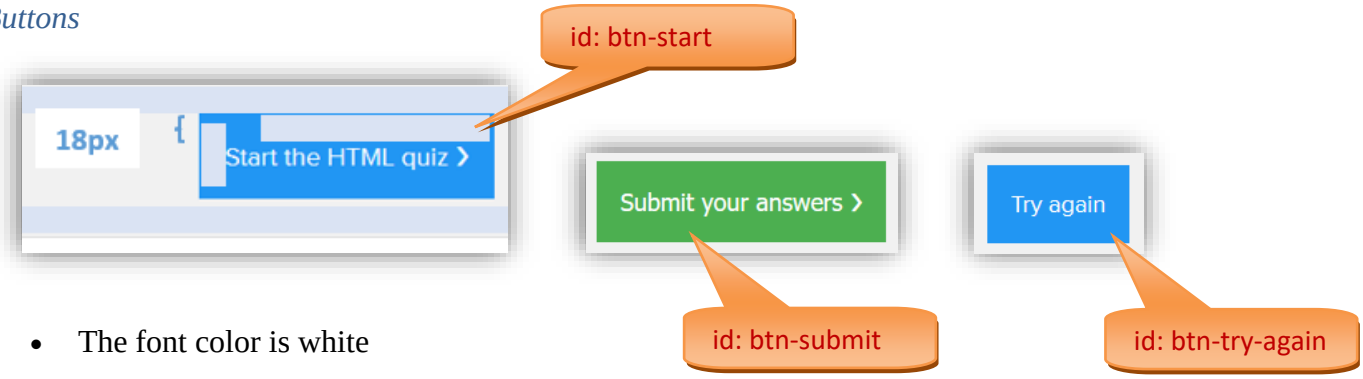
- The back ground color is semi-transparent, which is `rgba(255, 255, 255, 0.9)`
- The font face is 'Pangolin', and the fallback fonts are `'Trebuchet MS'`, `cursive`
- The font size is `60px`
- The font color is `black`
- Space between the border and the content of the element is `10px`.
- The quiz name is vertical, and center aligned to the quiz image.

Boxes



- The background color is `#f1f1f1`
- It has a solid border with `#dcdcdc` color of that is `1px` thick

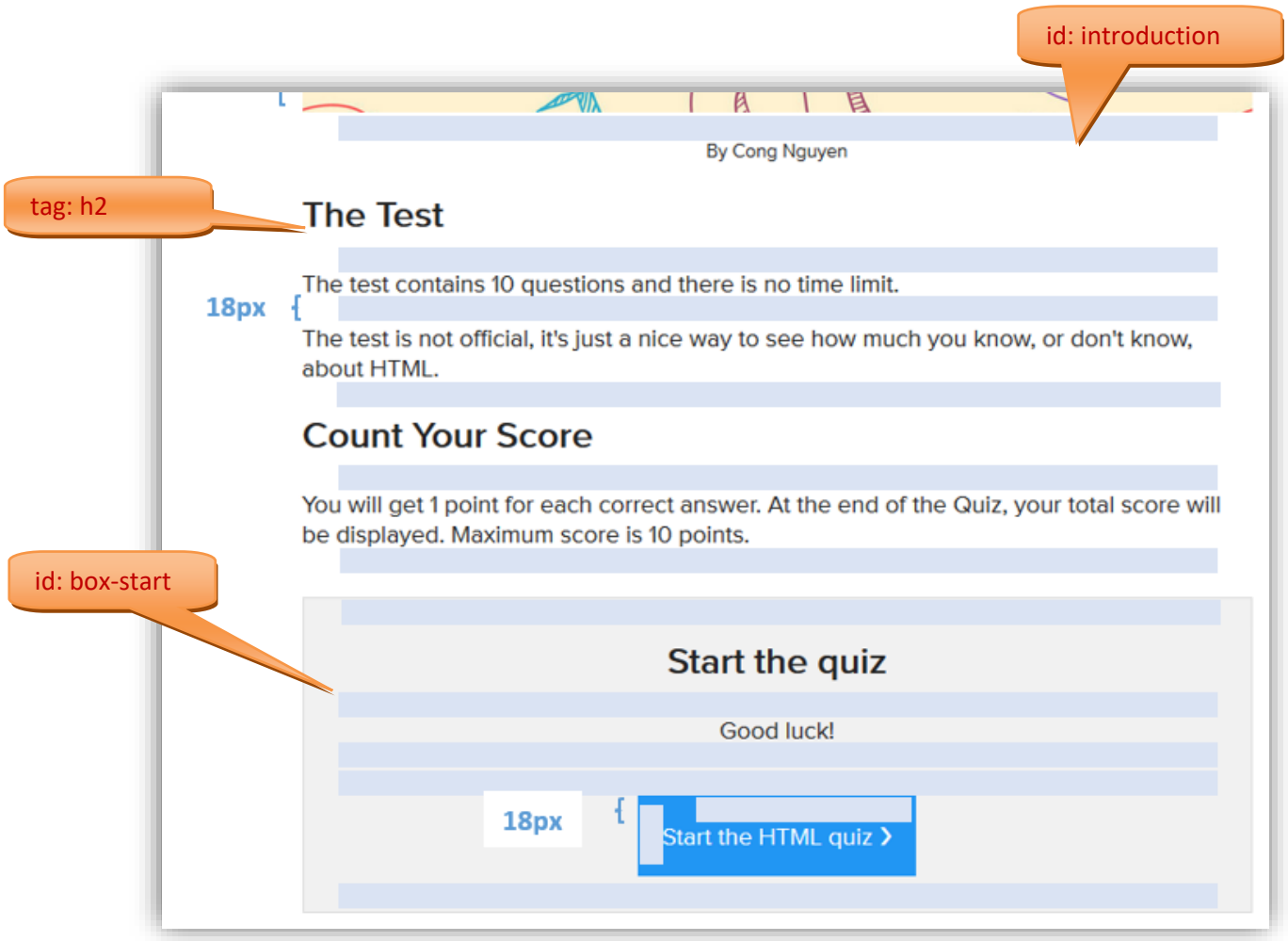
Buttons



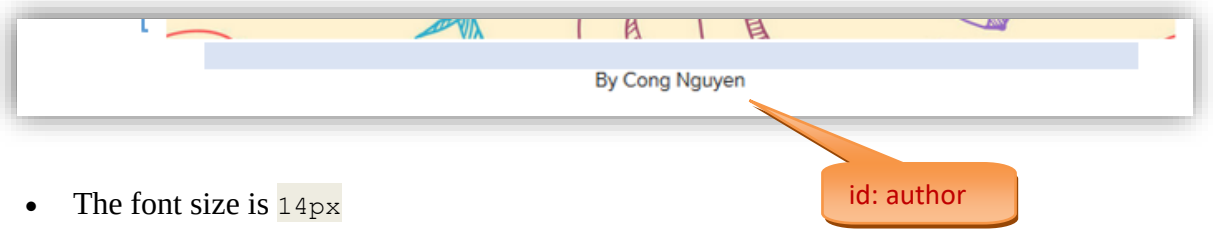
- The font color is white
- Blue buttons:
  - The background color is `#2196F3`
  - When users move the mouse over, the background color is `#0d8bf2`
- Green buttons:
  - The background color is `#4CAF50`
  - When users move the mouse over, the background color is `#46a049`

Screen 1: Introduction

This is the 1st screen shown to the users with some information before starting the quiz.



Author

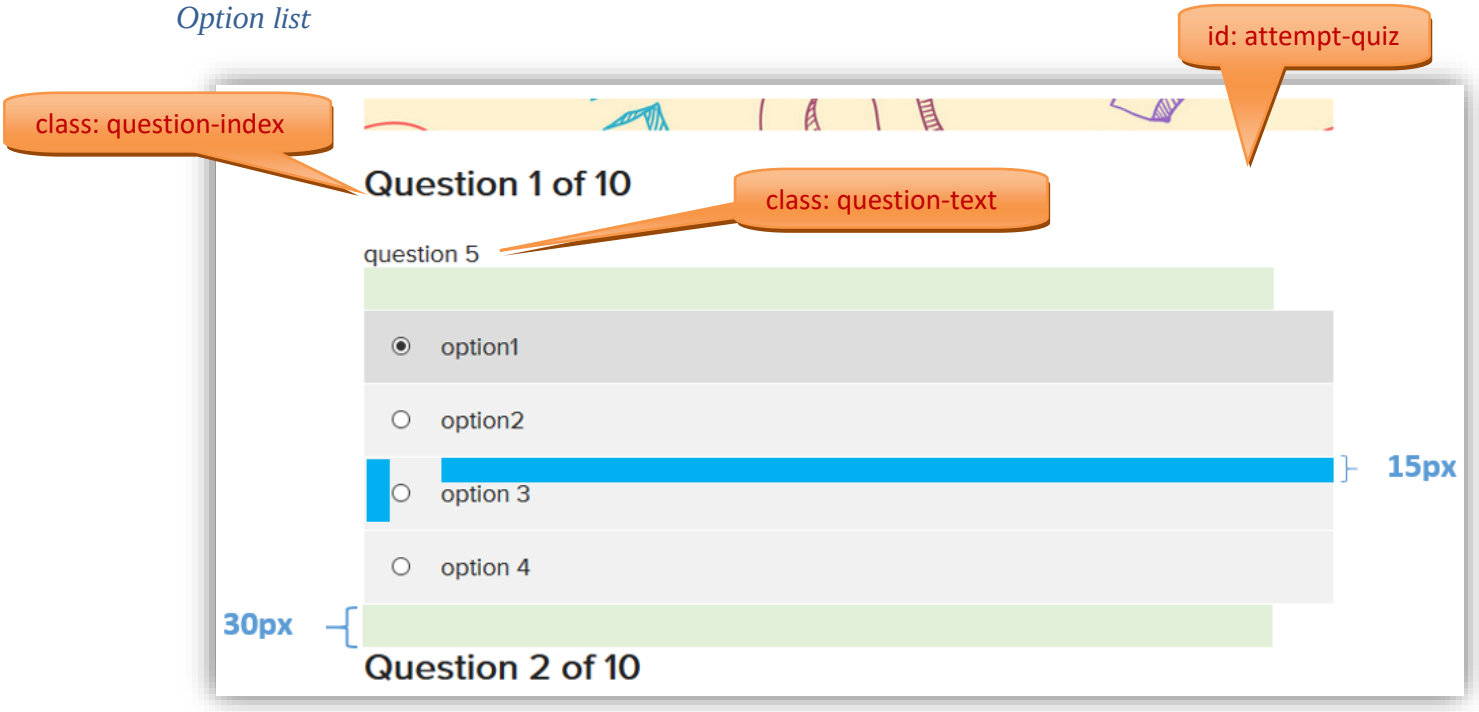


- The font size is `14px`
- The author is center aligned to the page
- **Note**: You must replace "Cong Nguyen" by ***your name*** as the author of your assignment.

Screen 2: Attempt Quiz

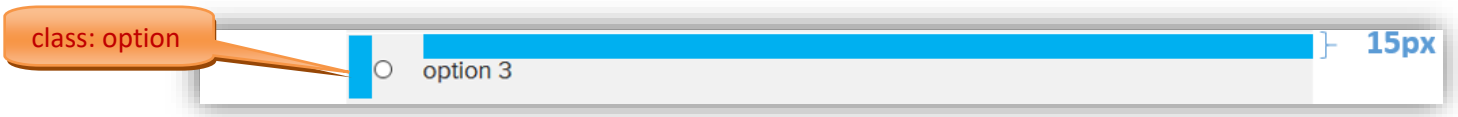
This is the 2nd screen shown to the users with questions & their answers to select.

Option list



- There is `1px` between each answer

*Option**: un-selected***



- Space between radio & text is `15px`
- The background color is `#f1f1f1`
- When users move the mouse over, the background color is `#ddd`

*Option: **selected***

- When users selected the answer, the background color is `#ddd`

3. Check point 2: HTML/CSS (2) screen 3, mobile layout

In this check point, you will complete the HTML/CSS for

- the remaining screen of the app: **review quiz**
- the mobile layout
- hide screen 2, 3 (at the beginning of the quiz)

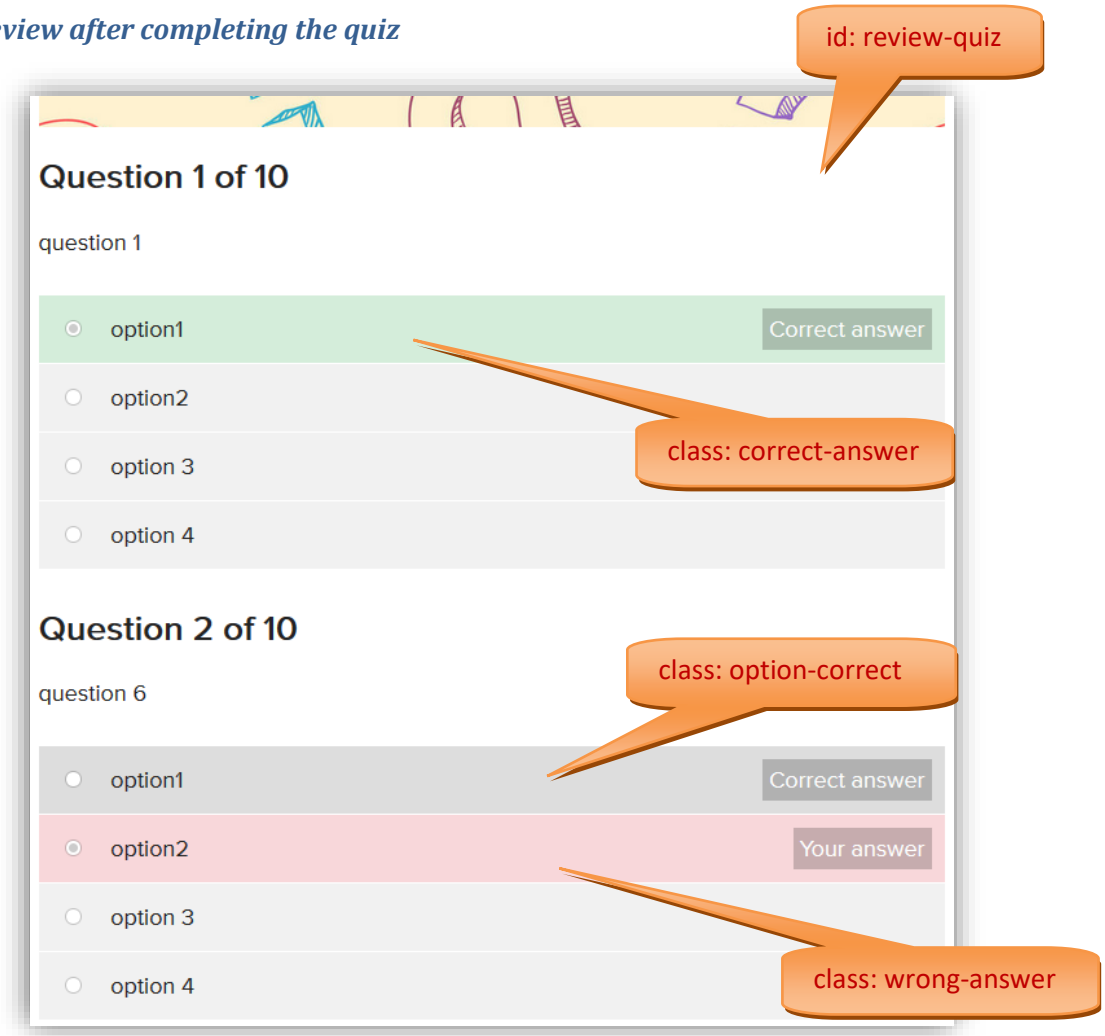
<table>
<tr><th>Week</th><th>HTML</th><th>CSS</th><th>JS</th></tr>
<tr><td rowspan="3">2</td><td colspan="2">- screen 3: review quiz
+ Your answer/ Correct answer
+ Correct or Wrong
+ result view</td><td rowspan="2"></td></tr>
<tr><td></td><td>- mobile layout
- hide screen 2, 3</td></tr>
<tr><td colspan="3">- test your application (expected elements, styles, mobile layout)</td></tr>
</table>

Screen 3: Review quiz

This is the 3rd screen shown to the users with the score + feedback text (`scoreText`) & also reviewing the selected answer with the correct one.

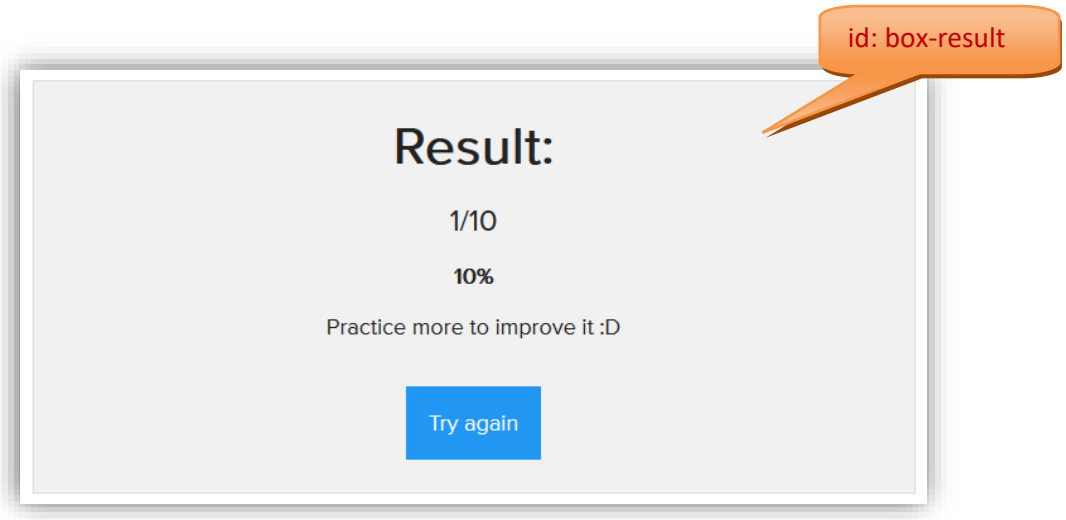
- The page should remain in this "completed" state until the users click "Try again" or refresh the page.

Option: ***review after completing the quiz***



- The radio boxes are disabled
- Correct answer (user selected) has background `#d4edda`
- Correct answer (user un-selected) has background `#d4edda`
- Wrong answer has background `#f8d7da`
- The label: *Correct answer/ Your answer*
    - The position is at the far end of answer (https://www.w3schools.com/css/css_positioning.asp may be helpful)
    - The background color is semi-transparent `rgba(0,0,0, 0.2)`

Result view



- The font size of the score (number of corrected answers - 1/10) is `24px`

Mobile layout

You need to also modify the CSS and HTML if necessary to implement support a mobile view. The video below shows an example of how it should look and behave:

https://drive.google.com/file/d/1aTYG8DEfBZpJ-ocwA5w3pYJt-eqBs-HW/view?usp=sharing

- If the page is viewed on mobile:
  - The viewport should be set to zoom-level `100%`, and the width should be the device width
- If the device screen size is less than `700px` wide:
  - The width of the page content should be `95%` instead of `700px`
- If the device screen size is less than `500px` wide:
  - The yellow circles in the page header should not appear

4. Check point 3: JS (1) user interactions

In this check point, you will start adding behavior to the quiz app including

- User navigation between screens
- User actions on quiz questions

Week	HTML	CSS	JS
3			- handle users click buttons to show/ hide (navigate between) screens

			- handle users click an answer / change answer
	- test your application		

Navigating between screens

When users click “Start the HTML quiz”,

- Hide *Screen 1: Introduction* & show *Screen 2: Attempt quiz*

When users click “Submit your answers”,

- Confirm if user really want to finish attempt or not, if yes
- Hide *Screen 2: Attempt quiz* & show *Screen 3: Review quiz*

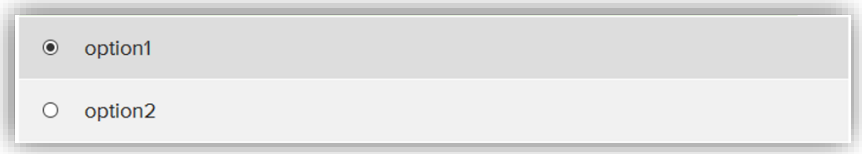
When users click “Try again”,

- Hide *Screen 3: Review quiz* & show *Screen 1: Introduction*

Whenever switching screen, you should also scroll to the top of the page. You can call `element.scrollIntoView();` to do this. See mdn for more details.

Clicking an answer

When the users click an answer option (radio box, text or anywhere into the option), the answer options should update in the following way:



- For the selected item:
  - The radio box should change from unchecked to checked
  - Background color is `#ddd`

Changing an answer

The users should be able to change their answer to a question by clicking a different answer.

5. Final submission: JS (2) fetching API

Finally, you will complete the quiz app functionalities by interacting with server API

- Create a new attempt, populate questions & display to user

- Submit user answers, populate review questions & display score to user

Week	HTML	CSS	JS
4			- handle users click: start the quiz (call api → populate questions & show screen 2) - handle users click: submit the quiz (call api → populate review questions & show screen 3) - restart the quiz
	- test your application		

Starting quiz

When users click "Start the HTML quiz",

- Call the provided API to get the attempt ID & an array of questions

`POST: https://wpr-quiz-api.herokuapp.com/attempts`

  - `Request data: none`
  - `JSON Response: status (201) + the new attempt (contains attempt ID & an array of questions)`

Example request & response: https://wpr-quiz-api.herokuapp.com/

- Populate *received questions* on the *Screen 2: Attempt quiz,* handle event to *click an answer* and *change an answer* (use check point 3)
- Hide *Screen 1: Introduction* & show *Screen 2: Attempt quiz* (done in check point 3)

Submitting your answers

When user click "Submit your answers",

- Confirm if user really want to finish attempt or not, if yes (done in check point 3)
- Call the *Submit API* to send user selected answers & received back the score

`POST: https://wpr-quiz-api.herokuapp.com/attempts/:id/submit`

  - `Request data:`

    - `Route param: id of the attempt`
    - `JSON body: an object of user selected answers in format {"userAnswers": {questionID: userSelectedAnswerIndex, …}}`

```
e.g. {"userAnswers": {5f61d4ec7c0f0531a4cedf41: 0, …}}
// user selected the first answer (index = 0) of question with id 5f61d4ec7c0f0531a4cedf41
```

- **`JSON Response:`**
  - `status (200) + the attempt (contains questions, user selected answers, correct answers of the same format, score e.g. 9/10, scoreText e.g. Perfect!!)`

Example request & response: https://wpr-quiz-api.herokuapp.com/

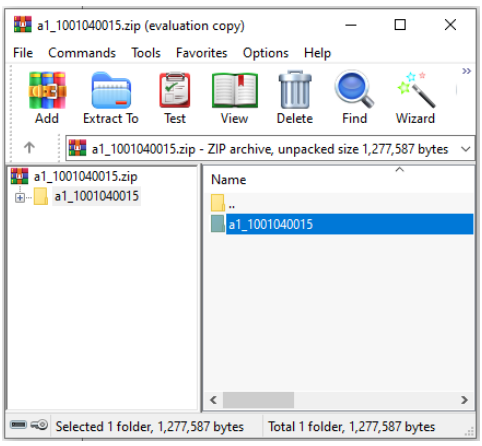
- Populate ***received questions***, *user selected answers, correct answers & the score* on the *Screen 3: Review quiz*
- Hide *Screen 2: Attemp quiz* & show *Screen 3: Attempt quiz* (done in check point 3)

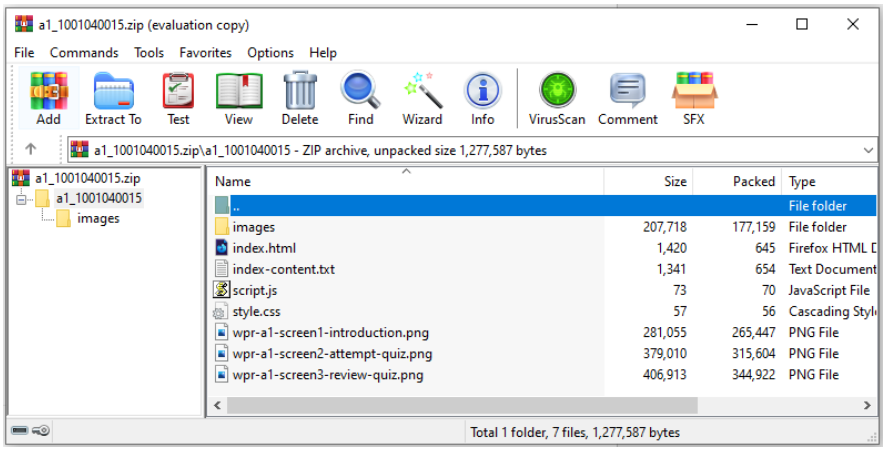
NOTE: (1) *received questions* - not the dumpy text as in the demo screenshot

(2) after submitting the answers, the user no longer be able to select or change an answer, until the users click "Try again" or refresh the page.

C. Submission

Rename *starter_pack* into `a1_Sid`, where *Sid* is your student identifier (the remaining bits of the file name must not be changed!). For example, if your student id is 1801040001 then your zip file must be named a1_1801040001. **Zip** this folder with file name in the form of `a1_Sid.zip` and submit the portal by the due date. Expectedly opening the file,





IMPORTANT: failure to name the file as shown will result in no marks being given!

NO PLAGIARISM: if plagiarism is detected, 0 mark will be given!