WPR – Fall 2021

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

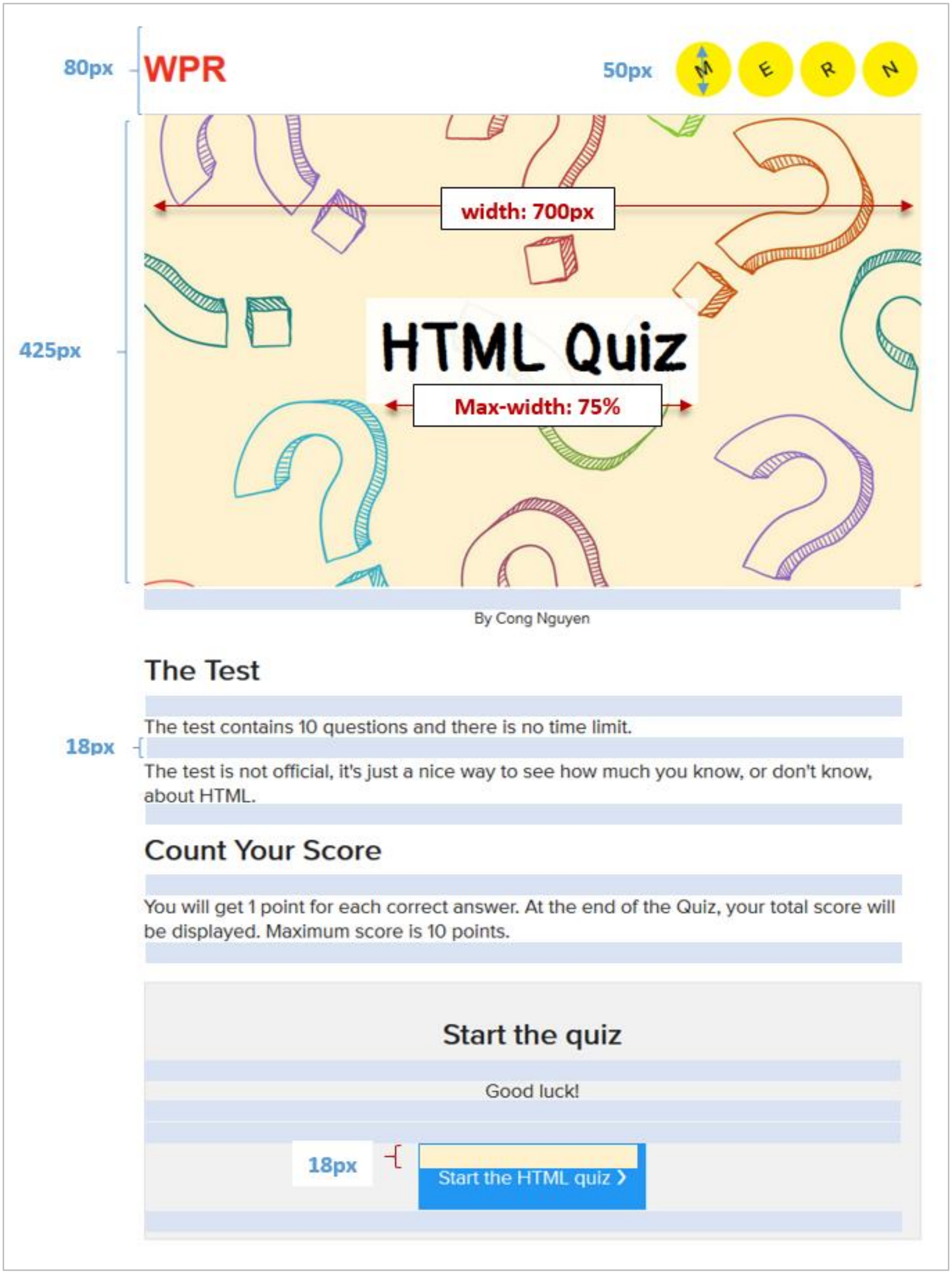
The following video shows the look and behavior of the quiz we are asking you to implement.

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

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 the quiz is completed (i.e. review mode - after submitting your answers), it should *not* be possible to change your answer anymore.

Section measurements



By Cong Nguyen

The Test

The test contains 10 questions and there is no time limit.

The test is not official, it's just a nice way to see how much you know, or don't know, about HTML.

Count Your Score

You will get 1 point for each correct answer. At the end of the Quiz, your total score will be displayed. Maximum score is 10 points.

Start the quiz

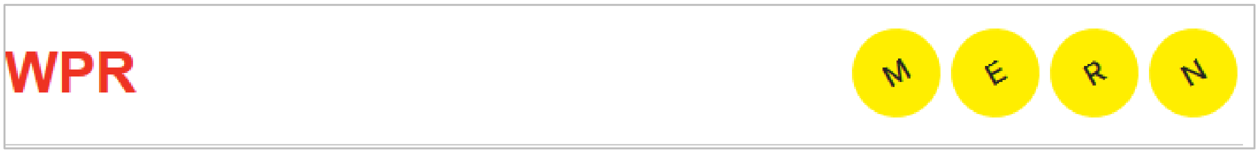
Good luck!

Font faces, size, and colors:

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.

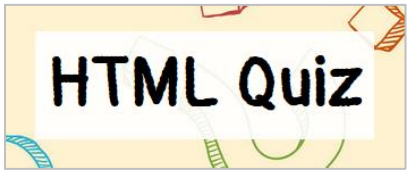
Course name

- 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]

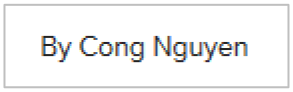
Quiz name



- 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.

Author



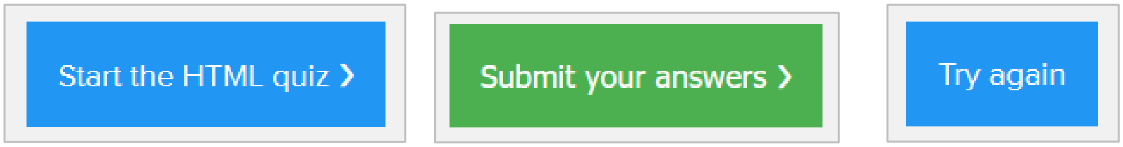
By Cong Nguyen

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

Images

- Quiz image
  - Quiz name 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

Buttons

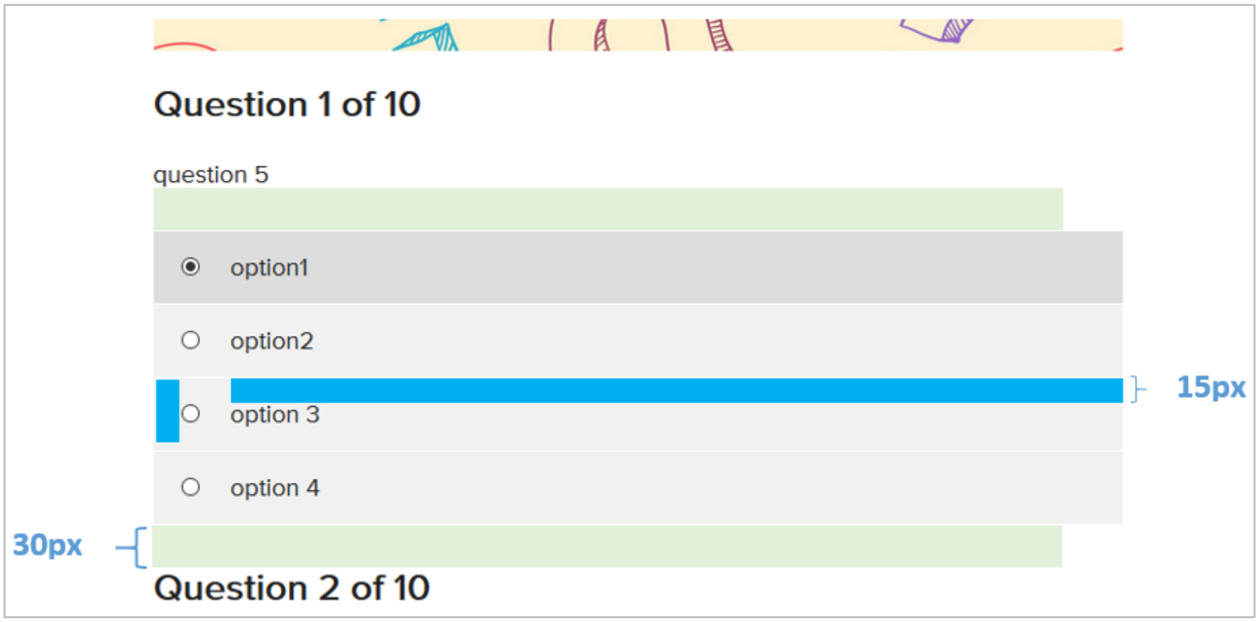


- The font color is white
- Blue buttons:
  - The background color is `#2196F3`
  - When users move the mouse over, the background color is `#0d8bf2`
- Green buttons:
  - The background color is `#4CAF50`
  - When users move the mouse over, the background color is `#46a049`

2. Choice list

You need to finish styling the question and answers.

Choice list



Question 1 of 10

question 5

● option1

○ option2

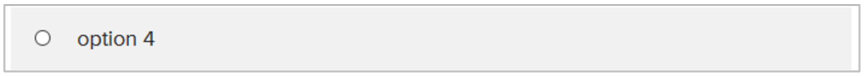
○ option 3

○ option 4

Question 2 of 10

- There is `1px` between each answer

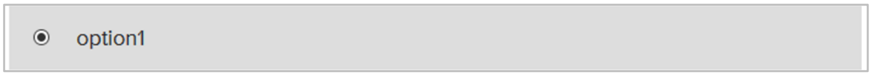
Choice: un-selected



○ option 4

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

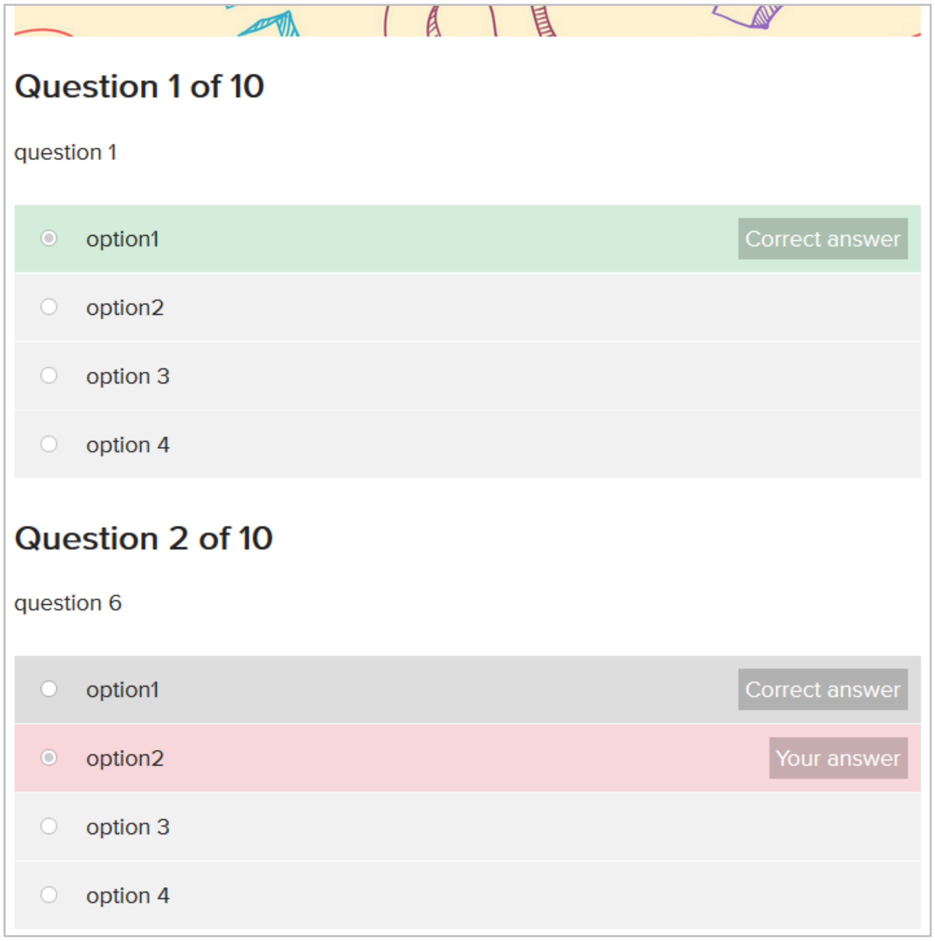
Choice: selected



● option1

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

Choice: review after complete the quiz



Question 1 of 10

question 1

● option1 Correct answer

○ option2

○ option 3

○ option 4

Question 2 of 10

question 6

○ option1 Correct answer

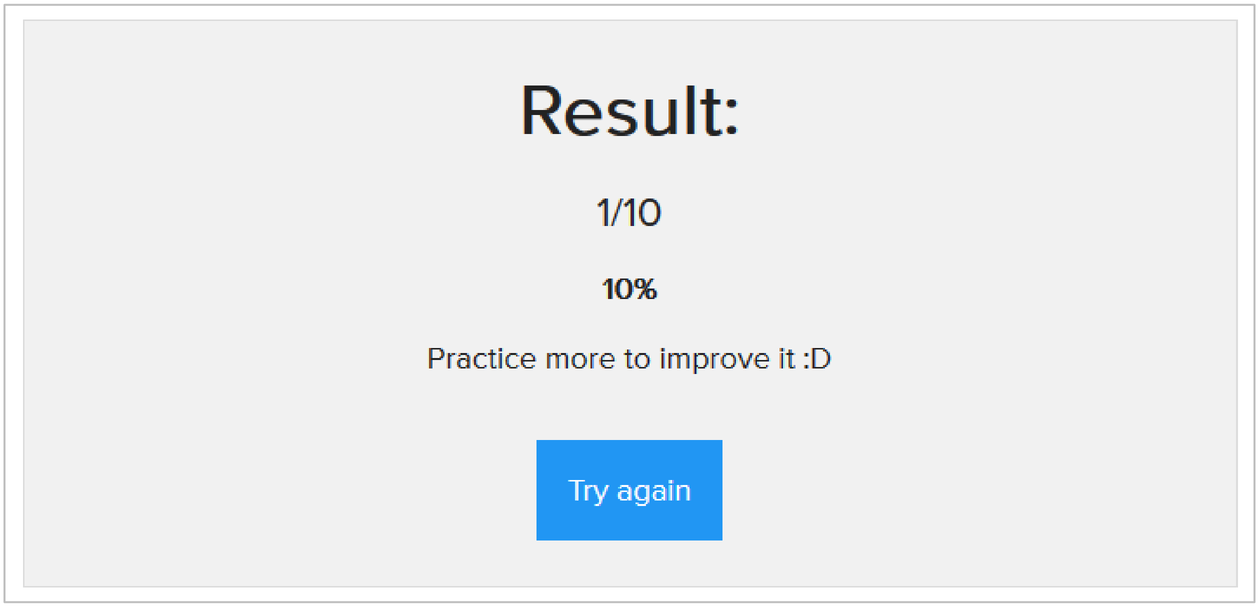
● option2 Your answer

○ option 3

○ option 4

- The radio boxes are disabled
- Correct answer 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)`

3. Boxes & Result view



Result:

1/10

10%

Practice more to improve it :D

- The background color is `#f1f1f1`
- It has a solid border with `#dcdcdc` color of that is `1px` thick
- The font size of the score (number of corrected answers - 1/10) is `24px`

4. 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

5. Quiz behavior

This is a 10-question **one-page** quiz app. Write the code necessary to implement the quiz behavior as detailed below.

Even though you will mostly be writing JavaScript for this part, you may need to also update the HTML or CSS in order to implement the behavior as described.

The app has 3 screens (modes), including (1) read some information about the quiz → **start quiz** → (2) answer quiz questions → **submit your answers** → (3) review your answer

Screen 1: Introduction

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

Starting the quiz

When the users click "Start the X quiz", the webpage need to:

- Calling 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)`

- Displaying *received questions* on the *Screen 2: Attempt quiz*

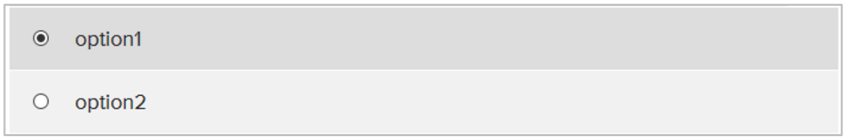
Screen 2: Attempt quiz

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

- You should also make the page 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 choice (radio box, text or anywhere into the option), the answer choices should update in the following way:



● option1

○ option2

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

Changing an answer

If the user has not completed the quiz (i.e. not click submit yet), they should be able to change their answer to a question by clicking a different answer.

After the user has submitted the answers, the answer choices should lock and it should no longer be possible to change an answer, until the user clicks “Try again” or refreshes the page.

Submitting your answers

When user click “Submit your answers”, the webpage need to:

- Confirming if user really want to finish attempt or not, if yes
- Calling 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 {questionID: userSelectedAnswerIndex}`

```
e.g. {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!!)`

- Displaying ***received questions***, *user selected answers*, *correct answers & the score* on the *Screen 3: Review quiz*

NOTE: received questions - not the dumpy text as in the demo screenshot

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 user refreshes the page or clicks “Try again”.

Restarting quiz

When user click “Try again”, the webpage need to show back the *Screen 1: Introduction*

- You should also make the page scroll to the top of the page.
  - You can call `element.scrollIntoView();` to do this. See mdn for more details.

B. Task requirements

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 as described in section ***A.2***
- Add appropriate viewport & CSS media queries to support the mobile view. (see section ***A.4***)

Note: style all 3 screens before hiding them using JavaScript

4. JavaScript: in the `script.js` file, to accomplish quiz behavior as described in section ***A.5***

- 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

5. Weekly plan:

You have to schedule yourself a weekly plan to complete your tasks, an example is given below.

<table>
<tr><th>Week</th><th>HTML</th><th>CSS</th><th>JS</th></tr>
<tr><td>1</td><td colspan="2">- header
- quiz title, author
- screen 1: introduction
- screen 2: attempt quiz
+ Question name & option</td><td></td></tr>
<tr><td rowspan="2">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>3</td><td></td><td></td><td>- handle user click buttons to show/ hide (navigate between) screens
- handle user click an answer / change answer</td></tr>
<tr><td>4</td><td></td><td></td><td>- handle user click: start the quiz (call api → populate questions & show screen 2)
- handle user click: submit the quiz (call api → populate review questions</td></tr>
</table>

			& show screen 3) - restart the quiz
	- test your application (mobile layout, correct result corresponding to clicked items, restart quiz)		

C. Submission

You must submit a single zip file containing the application to the portal by the due date. The zip file name must be of the form `a1_Sid.zip`, 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.

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!