

IS216 Mini Lab Test

[25 marks]

General Instructions:

- This is a time-bound (1 hour 35 minutes), open-book, open-Internet, and **individual** test.
- You must test your web pages using **Google Chrome Web Browser** Version 103.0.x or later). Your graders will be using only **Google Chrome Web Browser** (Version 103.0.x or later) to test your web pages.
- **No questions will be entertained** by the IS216 teaching team (faculty/instructor/Teaching Assistants) during the test period. If necessary, make your own assumptions and proceed to complete test questions.
- You must **use only standard HTML5, CSS, Bootstrap (Version 5.3), JavaScript and Axios** in your solutions unless the question specifies otherwise. Do not use any other third-party libraries (e.g. Angular, React, or others).
- Use meaningful names for HTML class/id and JavaScript variables and functions. You must indent your code (HTML/CSS/JavaScript) properly. Failure to do so will attract a penalty of up to **20%** of your score for the corresponding question.
- You **MUST** include your name as author in the comments of all your submitted source files. Failure to do so will attract a penalty of up to **20%** of your score for the corresponding question. For example, if your registered name is "KIM Jong Un" and email ID is kim.jongun.2020, include the following comment at the beginning of each source file you write.

HTML files	CSS, JavaScript files
<pre><!-- Name: KIM Jong Un Email: kim.jongun.2021 --></pre>	<pre>/* Name: KIM Jong Un Email: kim.jongun.2021 */</pre>

- You may wish to comment out the parts in your code which cause errors. Commented code will not be marked.

Academic Integrity

- All student submissions will be thoroughly checked by an SMU-approved source code plagiarism checker software program AND an additional external software program. The source code checking will be conducted across all submissions (from 10 sections of IS216).
- Suspected plagiarism cases will be reported immediately to the IS216 faculty in charge and SCIS Dean's Office for further investigation.
- Students in the suspected cases will be informed accordingly by their section faculty, and the incident will be escalated to the SMU University Council of Student Conduct.
- More information about the SMU Student Code of Conduct can be found at this link: <https://smu.sharepoint.com/sites/ucsc/Documents/Code%20of%20Academic%20Integrity.pdf>

Submission Instructions

- **Due Date**

- **13 October 2023 (Friday) 17:35 PM Singapore Time**
- Late submission policy is as follows:

Submit within 5 minutes of set deadline	10% penalty of your entire test's score
Beyond 5 minutes	0 mark

- **Zip** up all files in Q1/Q2/Q3/Q4 folders into <YOUR_SMU_ID>.zip
 - For example, **kim.jongun.2021.zip**
 - **Verify by unzipping this zip file – check the content inside**
 - **Incorrect submission file name** WILL attract a penalty of up to **20%** of your score for the entire test.
- Only **zip** format is accepted.
 - **.7z, rar** or other **compression formats** are **NOT** accepted.
 - Until the correct **zip** format is submitted again by the student, it will be assumed that the student has NOT made the submission and late submission policy will apply.
- Submit the **zip** file to the following location:
 - **IS216-MERGED eLearn** page: <https://elearn.smu.edu.sg/d2l/home/365191>
 - Go to **Assignments** → **Mini Lab Test** → **Submit your ZIP file**
 - **It is your (student's) individual responsibility to ensure that the zip file submission was successful.**
 - **Your section faculty and Teaching Assistants will NOT verify the submission for you.**

Legend



Do NOT edit this given resource file.





Your answer/code goes here into this given resource file.

Q1. Basic CSS

[8 Marks]

Given resources in folder Q1

 q1.html

 q1.css

Instructions

1. You are to style the HTML elements using **vanilla CSS**.
2. Open q1.html in a browser and examine how it looks.
3. You are to edit only q1.css file.

Note: You are to define CSS selectors and styles in the labelled locations e.g. /* part (a) */ within the css file to match the screenshots to the best of your interpretation **where the requirement is not stated explicitly**. This includes margin, padding, font style, font size, font weight, etc.

Part A: Style the header as shown below. (2 marks)

Personal Finance

- Home
- Bank
- CPF

- Set the background color as #151C55
- Set the text color as white
- For “Personal Finance” heading, set
 - top and bottom spacing as 10px
 - left and right spacing as 2px

Part B: Style the menu items in the header as shown below. (3 marks)

Personal Finance

Home

Bank

CPF

- Menu items are displayed in a single row instead of stacked one above the other.
- “Home” menu item is 150px from the left.
- Set text color of menu items as white.
- For each menu item (e.g. “Home”), set
 - top and bottom spacing as 10px.
 - left and right spacing as 30px.

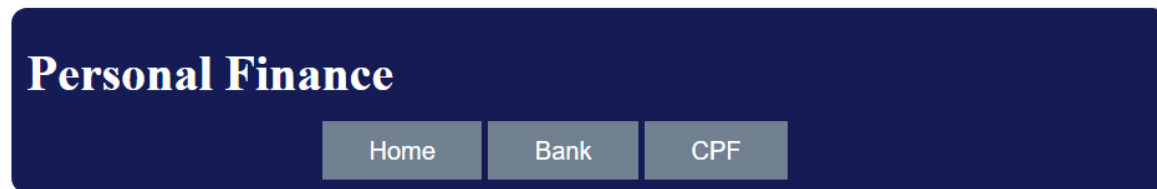
Part C: Style the table as shown below. (3 marks)

Date	Duration	Event
20 Mar 2023	1 hour	Value Investing
1 Apr 2023	2 hours	Stocks trading
4 Apr 2023	8 hours	Options trading

Free events are highlighted in **green**.

- The table border should be single-lined.
- For table heading, the top and bottom spacing should be 15px and left and right spacing should be 10px for each cell.
- The table heading text should be left aligned.
- The entire row should be white text against green background for free events.
- The “green” text of the footnote should be white against green background.

Your final page should look like the one below.



Reasons

- Save for your retirement
- Leave a legacy for your kids

Saving for retirement

So how do you make sure you have enough for your retirement?

Tips

- Spend below your means
- Make your money work for you

Investment Seminars


Date	Duration	Event
20 Mar 2023	1 hour	Value Investing
1 Apr 2023	2 hours	Stocks trading
4 Apr 2023	8 hours	Options trading

Free events are highlighted in **green**.

Q2. Bootstrap: Debug Layout

[5 Marks]

Given resources in folder Q2

 q2.html

Instructions

1. You are to solve the web page layout problem using Bootstrap's *grid and breakpoints*.
2. Open q2.html in a browser and examine how it looks when you change the browser width.
3. You should notice that the given code has already set grid layout and breakpoints for different browser widths but the code is buggy.

Note:

- You are only required to debug and add code to achieve the web page layout at different browser widths specified below.
- You are not required to add/edit other styles, such as margin, padding, font style, font size, etc., not mentioned in the requirements.

Debug and add code in q2.html so that it behaves as follows:

Part A: When the browser width is ≥ 992 px (lg breakpoint), the form should appear as shown below. (1.5 marks)

Registration Form

Name:	Gender:	Age:
<input type="text" value="Enter Your Full Name"/>	<input type="text" value="Male"/>	<input type="text" value="21 and above"/>
Email:		
<input type="text" value="Enter Email"/>		
Password:	Confirm Password:	
<input type="text" value="Password here"/>	<input type="text" value="Password here"/>	
<input type="button" value="Create Account"/>	Address:	
<input type="text"/>		

- Name, Gender, and Age fields occupy **4 columns** of Bootstrap 12-columns grid, each.
- Email field occupies the **entire 12-columns** grid.
- Password and Confirm Password fields occupy **6 columns** of Bootstrap 12-columns grid, each.
- Create Account button and Address field occupy **6 columns** of Bootstrap 12-columns grid, each.

For ease of understanding, the figure in the next page illustrates the above description.

Registration Form

Name: Gender: Age:

Email:

Password: Confirm Password:

Address:

Grid dimensions: 4 columns, 4 columns, 4 columns, 6 columns, 6 columns, 12 columns.

Part B: When the browser width is ≥ 768 px (md breakpoint) but < 992 px (lg breakpoint), the form should appear as shown below. (1.5 marks)

Registration Form

Name: Gender:

Age: Email:

Password: Confirm Password:

Address:

- Name and Gender fields occupy **6 columns** of Bootstrap 12-columns grid, each.
- Age and Email fields occupy **6 columns** of Bootstrap 12-columns grid, each.
- Password and Confirm Password fields occupy **6 columns** of Bootstrap 12-columns grid, each.
- Create Account button and Address field occupy **6 columns** of Bootstrap 12-columns grid, each.

Part C: When the browser width is < 768 px (md breakpoint), the form should appear as shown below. (2 marks)

Registration Form

Name:

Gender:

Age:

Email:

Password:

Confirm Password:


Address:


- Each field occupies the **entire 12-columns** grid.
- Create Account button is displayed below Address field.

Q3. JavaScript DOM

[5 Marks]

Given resources in folder Q3

 q3.html

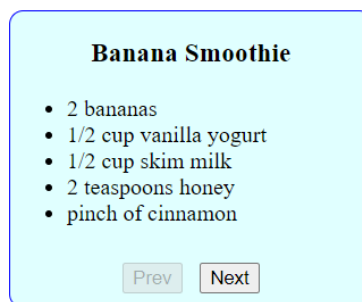
 q3.js

Instructions

1. You are to edit only q3.js
 - a. debug changeRecipe and updateButtonState function in q3.js
 - b. complete displayRecipe function in q3.js
2. No hardcoding. The contents to be displayed should be retrieved from the recipes array given in q3.js. The values of recipes array will be changed when grading.

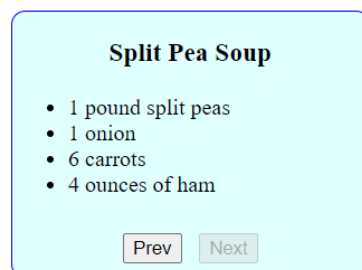
Debug and Add Code in q3.js so that q3.html behaves as follows.

Part A: When q3.html is open in the browser for the first time, it should show the name of the first recipe, from the recipes array, and its ingredients as an unordered list. The “Prev” button should also be disabled as shown below. (2.5 marks)



Part B: When the “Next” button is clicked, the details for the next recipe on the recipes list should be displayed. Similarly, when the “Prev” button is clicked, the details for the previous recipe on the recipes list should be displayed. (1 mark)






Part C: When the last recipe is displayed, the “Next” button should be disabled as shown below. (1.5 mark)



Q4: JavaScript AXIOS & JSON

[7 Marks]

Given resources in folder Q4

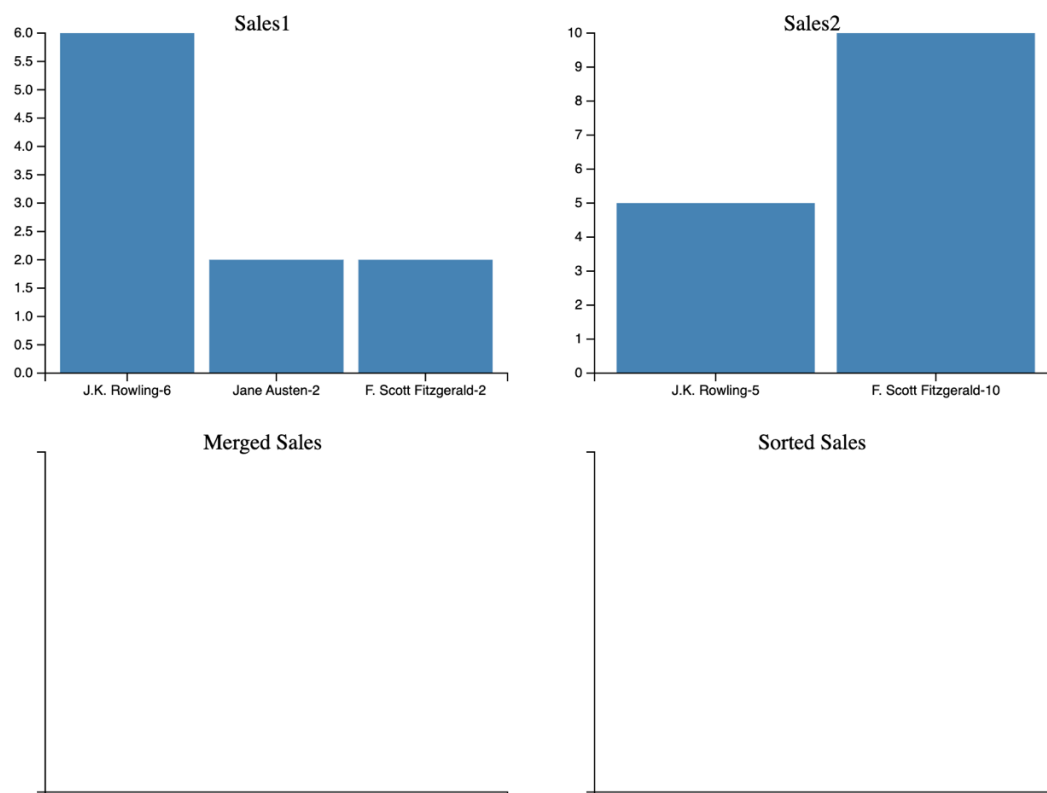
-  q4.js
-  q4.html
-  draw.js
-  sales1.json
-  sales2.json

q4.html displays the book sales of various authors from sales1.json and sales2.json retrieved via asynchronous requests. It then displays the *merged sales data* and *the sales of the best-selling authors* (authors who have sold at least five books).

Instructions

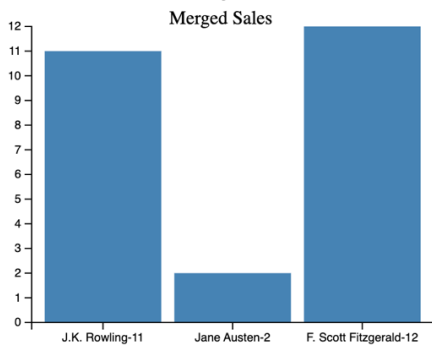
- You are to edit only q4.js
 - complete mergeSales function in q4.js
 - complete sortSales function in q4.js
- No hardcoding. The data to be displayed should be retrieved from sales1.json and sales2.json. The data of those JSON files will be changed when grading. But you may assume that the data structure of sales1.json and sales2.json is always the same.
- Note that your solution will be graded based on the correct Browser output. No partial mark will be given for incomplete/incorrect solution.

When q4.html is open in the browser, it will show the following web page initially:

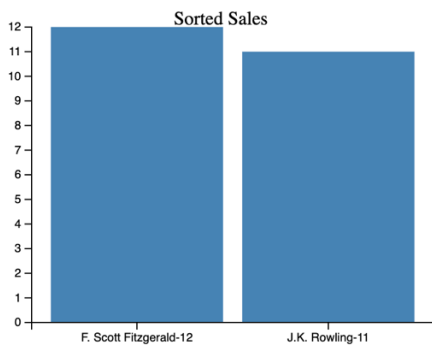


Notice that “Merged Sales” and “Sorted Sales” charts have not been populated yet.

Part A: Add code in `mergeSales` function in `q4.js` so that the web page displays the bar chart shown below. The function should merge the two sales data from `sales1.json` and `sales2.json` and produce the chart showing the total sales of each author. (4 marks)



Part B: Add code in `sortSales` function in `q4.js` so that the web page displays the bar chart shown below. The function should merge the two sales data from `sales1.json` and `sales2.json`; sort the authors according to their total sales; and produce the chart showing the total sales of each author who has sold at least five books, in descending order. (3 marks)



Notice that the bar chart is sorted in descending order according to sales. Only the authors who have sold *at least five books* are displayed in the chart.

If done correctly, the final output of `q4.html` should be

