

IS216 Sample Large Lab Test (Set A)

[40 marks]

General Instructions:

- This is a time-bound (2 hours), 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 5.2), JavaScript, Axios, and Vue.js (Version 3)** 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 11 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**

- **XX November 2022 (Friday) XX:XX 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/333446>
 - Go to **Assignments → Large 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.

IMPORTANT

Inside **resources** folder, you will find **Bootstrap** files. **Please do NOT edit these files.**

→ Bootstrap\



bootstrap.bundle.min.js




bootstrap.min.css


Q1. [CSS, Bootstrap] Savings

[10 marks]

Given resources:

→ Q1\

◆  savings.html

◆  savings.js

You are to **edit** `savings.html` and `savings.js` **only**. Do not create additional files.

IMPORTANT

You are free to define HTML elements and CSS style required 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.

- `savings.html` must only contain HTML and CSS code. You must write **ALL** JavaScript code inside `savings.js`.
- JavaScript code written inside `savings.html` will **NOT** be considered for grading.

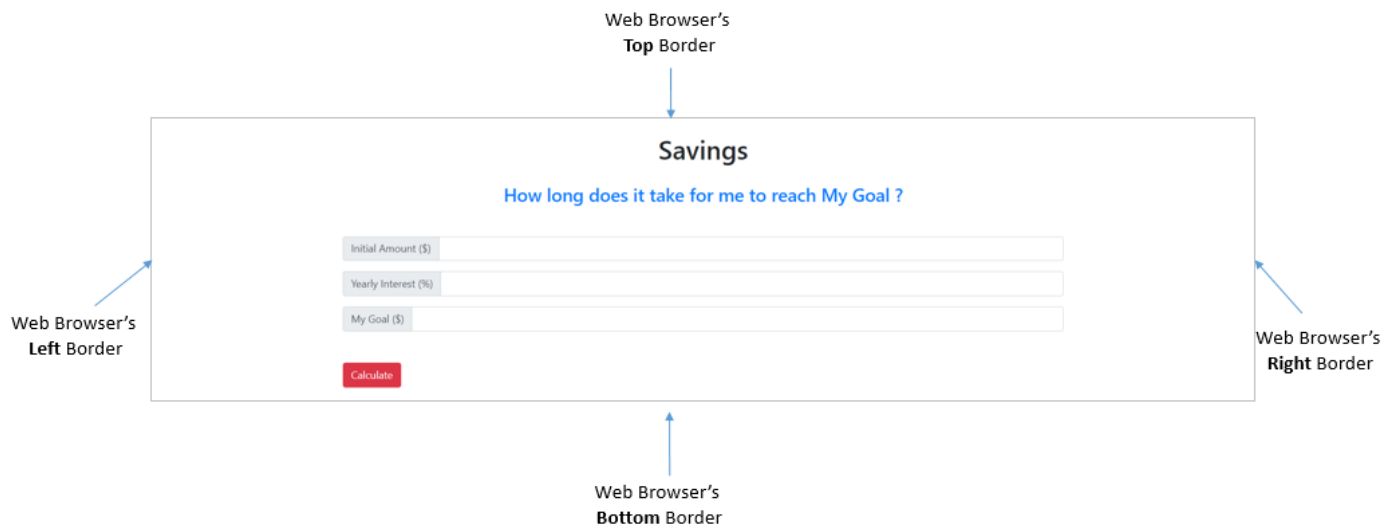
Part A: Complete `savings.html`

Complete implementation of `savings.html` such that when rendered in a web browser, `savings.html` displays the following.

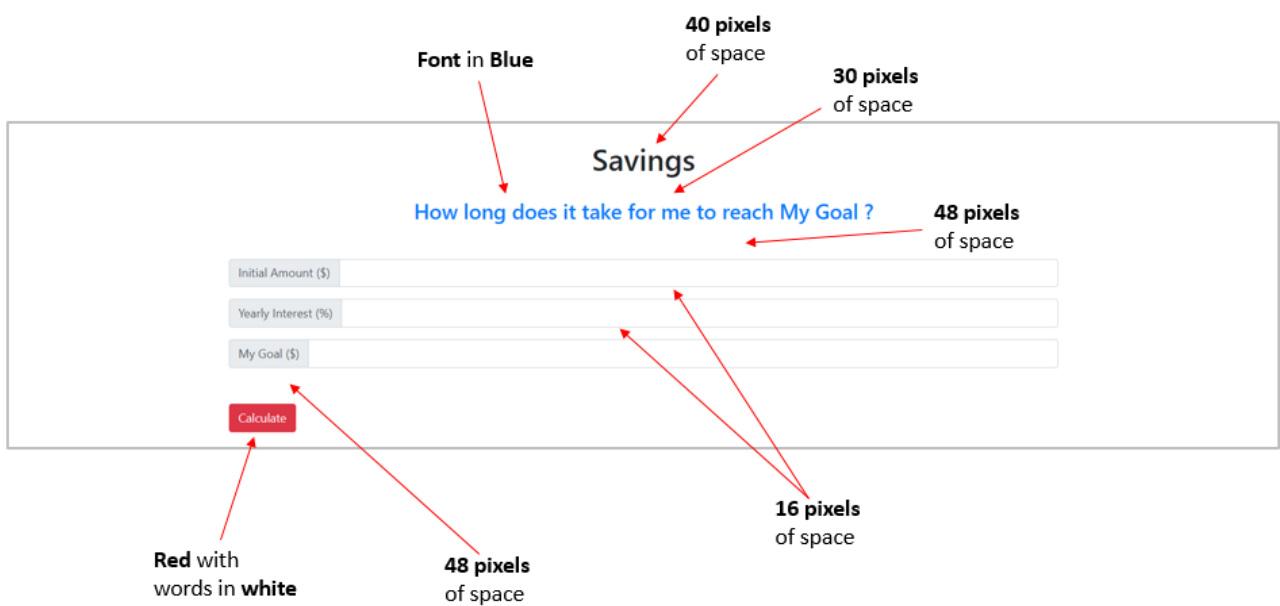
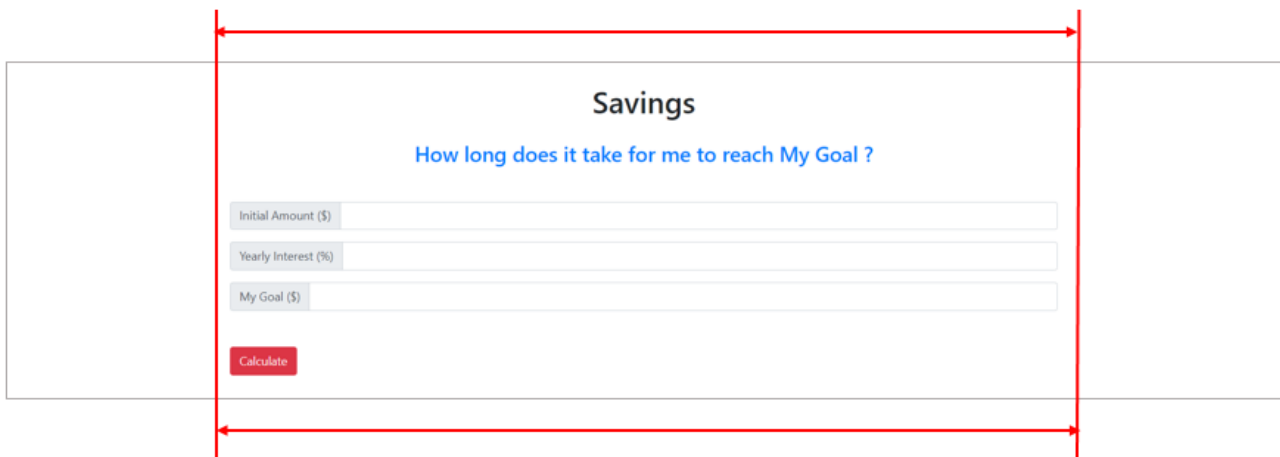
NOTE: The three **input fields** must only allow for **numeric values** (not text). You may **ASSUME** that the user will always enter numeric values greater than 0 (thus, no negative values).

savings.html	
<h2>Savings</h2> <p>How long does it take for me to reach My Goal ?</p> <div><div>Initial Amount (\$)</div><input type="text"/></div> <div><div>Yearly Interest (%)</div><input type="text"/></div> <div><div>My Goal (\$)</div><input type="text"/></div> <div><div>Calculate</div></div>	

Please take note of the following **user interface layout** guidelines:



The **main content** is centered and spans **70%** of the web browser **width**.



Part B: Complete function calculate()

Complete the function **calculate()** in **savings.js** such that it calculates the **number of years** it will take for the user to achieve **My Goal** (the user's goal) given:

- 1) an **Initial Amount (\$)**, and
- 2) **Yearly interest (%)**

savings.html (before clicking Calculate button)	
<h2 style="margin: 0;">Savings</h2> <p style="color: blue; margin: 10px 0;">How long does it take for me to reach My Goal ?</p> <div style="margin-bottom: 10px;"> <div style="background-color: #f0f0f0; padding: 2px 5px; display: inline-block;">Initial Amount (\$)</div> <input style="width: 80%; border: 1px solid #ccc;" type="text" value="5000"/> </div> <div style="margin-bottom: 10px;"> <div style="background-color: #f0f0f0; padding: 2px 5px; display: inline-block;">Yearly Interest (%)</div> <input style="width: 80%; border: 1px solid #ccc;" type="text" value="10"/> </div> <div style="margin-bottom: 10px;"> <div style="background-color: #f0f0f0; padding: 2px 5px; display: inline-block;">My Goal (\$)</div> <input style="width: 80%; border: 1px solid #ccc;" type="text" value="8000"/> </div> <div style="text-align: center; margin-top: 20px;"> <div style="background-color: #ff0000; color: white; padding: 5px 15px; border: 1px solid #ccc; cursor: pointer;">Calculate</div> </div>	

1. The user enters Initial Amount = **5000**, Yearly Interest = **10** and a Goal of **8000** in the three input text fields.
2. The user clicks the **Calculate** button.
3. The **results** will be displayed below the **Calculate** button. It will display the **number of years** to achieve the goal (**as an Integer**) and the **amount** you will get in **two decimal places**.

savings.html (after clicking Calculate button)					
<h2 style="margin: 0;">Savings</h2> <p style="color: blue; margin: 10px 0;">How long does it take for me to reach My Goal ?</p> <div style="margin-bottom: 10px;"> <div style="background-color: #f0f0f0; padding: 2px 5px; display: inline-block;">Initial Amount (\$)</div> <input style="width: 80%; border: 1px solid #ccc;" type="text" value="5000"/> </div> <div style="margin-bottom: 10px;"> <div style="background-color: #f0f0f0; padding: 2px 5px; display: inline-block;">Yearly Interest (%)</div> <input style="width: 80%; border: 1px solid #ccc;" type="text" value="10"/> </div> <div style="margin-bottom: 10px;"> <div style="background-color: #f0f0f0; padding: 2px 5px; display: inline-block;">My Goal (\$)</div> <input style="width: 80%; border: 1px solid #ccc;" type="text" value="8000"/> </div> <div style="text-align: center; margin-top: 20px;"> <div style="background-color: #ff0000; color: white; padding: 5px 15px; border: 1px solid #ccc; cursor: pointer;">Calculate</div> </div> <div style="margin-top: 20px;"> <p style="color: green; margin: 0;">Result</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tbody> <tr> <td style="padding: 5px;">You will achieve your goal in (years):</td> <td style="text-align: center; padding: 5px;">5</td> </tr> <tr> <td style="padding: 5px;">You will get (\$):</td> <td style="text-align: center; padding: 5px;">8052.55</td> </tr> </tbody> </table> </div>		You will achieve your goal in (years):	5	You will get (\$):	8052.55
You will achieve your goal in (years):	5				
You will get (\$):	8052.55				

For the “**My Goal**” calculation to work, all three **input field values** are necessary.

For example, below, we demonstrate a scenario where the user leaves all three input fields EMPTY.

savings.html (before clicking Calculate button)

Savings

How long does it take for me to reach My Goal ?

Initial Amount (\$)

Yearly Interest (%)

My Goal (\$)

Calculate

1. The user leaves all three **input fields** empty (no values).
2. The user clicks the **Calculate** button.

savings.html (after clicking Calculate button)

Savings

How long does it take for me to reach My Goal ?

Initial Amount (\$)

Yearly Interest (%)

My Goal (\$)

Calculate

Result

You will achieve your goal in (years):	0
You will get (\$):	0.00

3. The **results** will be displayed below the **Calculate** button (indicating zero).

Please take note of the following **user interface layout** guidelines:

The diagram illustrates the UI layout for a 'Savings' calculator. At the top, the title 'Savings' is centered, followed by the question 'How long does it take for me to reach My Goal ?' in blue. Below this are three input fields: 'Initial Amount (\$)' with value 5000, 'Yearly Interest (%)' with value 10, and 'My Goal (\$)' with value 8000. A red 'Calculate' button is positioned below the inputs. To the left of the button, a red arrow points to the space between the button and the result, labeled '30 pixels of space'. Below the button, the word 'Result' is displayed in green. A red arrow points to this text, labeled 'Font in Green'. Below 'Result' is a table with two rows: 'You will achieve your goal in (years):' with value 5, and 'You will get (\$):' with value 8052.55. A red arrow points to the space between the button and the table, labeled '20 pixels of space'. A red arrow points to the width of the table, labeled 'Table width of 500 pixels'.

You may refer to the following **test cases** to test your code.

	User Input	Result after clicking Calculate button				
1	Initial Amount: 1000 Yearly Interest: 10 My Goal: 1100	<div>Result</div> <table><tr><td>You will achieve your goal in (years):</td><td>1</td></tr><tr><td>You will get (\$):</td><td>1100.00</td></tr></table>	You will achieve your goal in (years):	1	You will get (\$):	1100.00
You will achieve your goal in (years):	1					
You will get (\$):	1100.00					
2	Initial Amount: 1500 Yearly Interest: 5.2 My Goal: 2000	<div>Result</div> <table><tr><td>You will achieve your goal in (years):</td><td>6</td></tr><tr><td>You will get (\$):</td><td>2033.23</td></tr></table>	You will achieve your goal in (years):	6	You will get (\$):	2033.23
You will achieve your goal in (years):	6					
You will get (\$):	2033.23					
3	Initial Amount: 1000.50 Yearly Interest: 7.8 My Goal: 800	<div>Result</div> <table><tr><td>You will achieve your goal in (years):</td><td>0</td></tr><tr><td>You will get (\$):</td><td>1000.50</td></tr></table>	You will achieve your goal in (years):	0	You will get (\$):	1000.50
You will achieve your goal in (years):	0					
You will get (\$):	1000.50					

4	Leaving any of the three input fields EMPTY	<div>Result</div> <table><tr><td>You will achieve your goal in (years):</td><td>-</td></tr><tr><td>You will get (\$):</td><td>0.00</td></tr></table>	You will achieve your goal in (years):	-	You will get (\$):	0.00
You will achieve your goal in (years):	-					
You will get (\$):	0.00					


Q2. [Bootstrap & Responsive Page] Payment

[10 marks]

Given resources:

→ Q2\

◆  logo.jpg

◆  payment.html

You are to **edit payment.html only**. Do not edit the other files. Do not create additional files.

IMPORTANT

Edit `payment.html` as per the instructions and visual illustrations below using **only Bootstrap classes**.


You are free to define HTML elements required to match the screenshots to the best of your interpretation **where the requirement is not stated explicitly**.

- `payment.html` should not contain any internal or inline CSS.
 - `payment.html` should not reference your own external CSS files.
 - Any internal, inline, or custom CSS will **NOT** be considered for grading.
-

Edit `payment.html` as per the instructions and visual illustrations below using **only Bootstrap classes**.

When done correctly, the web browser will display the following expected outputs:

payment.html
(Web Browser Width 992 pixels)


Checkout

Billing address

First name

Last name

Address

1234 Main St

Country

Choose... ▾

Postal Code

Credit Card

Name on card

Credit card number

Full name as displayed on card

Expiration

CVV

Continue to checkout

Your cart 3

Bathroom Detergent 3-bundle pack	\$30
Black Coffee 20 sachets	\$20
Kangaroo Facial Wash Clearance Sale 60% off	\$10
Promo code XMAS2021	-\$10
Total (USD)	\$50

Promo code

Redeem

payment.html
(Web Browser Width 991 pixels)



Checkout

Billing address

First name

Last name

Address

1234 Main St

Country

Choose...



Postal Code

Credit Card

Name on card

Credit card number

Full name as displayed on card

Expiration

CVV

Continue to checkout

Your cart

3

Bathroom Detergent

3-bundle pack

\$30

Black Coffee

20 sachets

\$20

Kangaroo Facial Wash

Clearance Sale 60% off

\$10

Promo code

XMAS2021

-\$10

Total (USD)

\$50

Promo code

Redeem

payment.html
(Web Browser Width 767 pixels)



Checkout

Your cart

3

Bathroom Detergent 3-bundle pack	\$30
Black Coffee 20 sachets	\$20
Kangaroo Facial Wash Clearance Sale 60% off	\$10
Promo code XMAS2021	-\$10
Total (USD)	\$50

Redeem

Billing address

First name

Last name

Address

Country

Postal Code

Credit Card

Name on card

Full name as displayed on card

Credit card number

Expiration

CVV

[Continue to checkout](#)

payment.html
(Web Browser Width 575 pixels)



Checkout

Your cart

3

Bathroom Detergent 3-bundle pack	\$30
Black Coffee 20 sachets	\$20
Kangaroo Facial Wash Clearance Sale 60% off	\$10
Promo code XMAS2021	-\$10
Total (USD)	\$50

Redeem

Billing address

First name

Last name

Address

Country



Postal Code

Credit Card

Name on card

Full name as displayed on card

Credit card number

Expiration

CVV

[Continue to checkout](#)

NOTE: First name and Last name fields are no longer *side-by-side*.

Q3. [JavaScript, Axios] Get maximum number from two lists**[10 marks]**

Given resources:

→ Q3\

- ◆  server/max.php
- ◆  axios.js
- ◆  max.html
- ◆  max.js

You are to **edit max.js only**. Do not edit the other files. Do not create additional files.

You may assume the following 3 files are correct.

axios.js	Local axios library.
server/max.php	<ul style="list-style-type: none"> • Optional GET HTTP parameter <code>id</code> • Displays <ul style="list-style-type: none"> ○ a list of items. The items can be numbers, strings, boolean, etc. ○ OR an empty list. <p>E.g. <code>http://localhost/.../Q3/server/max.php?id=3</code></p> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;">["z",12,"15"]</div> <p>The values will be changed during grading.</p>
max.html	<p>Access this from a WAMP/MAMP server; NOT from your file system.</p> <p>Run multiple test cases for function <code>get_max()</code> in file <code>max.js</code>.</p>

Edit the given file `max.js` to complete the function `get_max()`.

- The function takes in 3 parameters:
 - `url1` - URL for a web API
 - `url2` - URL for another web API
 - `ele_id` - valid id of an existing HTML element.
- Retrieve two lists from the web APIs indicated by parameters `url1` and `url2`.
 - You may assume that the web API will return
 - a list of items. The items can be numbers, strings, boolean, etc.
 - OR an empty list.
 - The given `server/max.php` (described above) is an example of such an API and is used in `max.html` for testing this function.
- If any of the AJAX calls fails, in the HTML element with id indicated by the parameter `ele_id`, display the failed URL (`url1` or `url2`), and the error in the following format:

```
[failed AJAX url]<br>
[error]
```

E.g.

```
no_such_url.php
Error: Request failed with status code 404
```

- Find the maximum aka largest number from the 2 lists combined.
 - Number includes positive, negative integers and decimals.
 - Note: "15" is a string, not a number.
- In the HTML element with id indicated by the parameter `ele_id`, display the parameters `url1` and `url2`, the 2 lists and the maximum number in the following format:

```
[url1]: [list from url1]<br>
[url2]: [list from url2]<br>
Max : [maximum number]
```

E.g.

```
server/max.php?id=1 : [6,-1,9.5,0]
server/max.php?id=2 : [3.8,5,-8]
Max : 9.5
```

- If both lists do not contain any number, display "No number".

```
server/max.php?id=4 : ["a"]
server/max.php : []
Max : No number
```


If done correctly, `max.html` should display the following output in the web browser.

1. lists contain only numbers

```
server/max.php?id=1 : [6,-1,9.5,0]
server/max.php?id=2 : [3.8,5,-8]
Max : 9.5
```

2. lists may contain string

```
server/max.php?id=3 : ["z",12,"15"]
server/max.php?id=2 : [3.8,5,-8]
Max : 12
```

3. empty list and list with no numbers

```
server/max.php?id=4 : ["a"]
server/max.php : []
Max : No number
```

4. Failed AJAX call

```
no_such_url.php
Error: Request failed with status code 404
```

5. Failed AJAX call

```
unknown_url.php
Error: Request failed with status code 404
```

DO NOT HARDCODE!

The arguments passed in to `get_max()` and the retrieved lists' values will be changed for grading.

Q4. [Vue.js] Party

[10 marks]

Given resources:

→ Q4\

- ◆  vue.global.js
- ◆  party.html
- ◆  party.js

You are to **edit** `party.html` and `party.js` **only**. Do not edit the other files. Do not create additional files.

IMPORTANT

You are free to define HTML elements and CSS style required 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.

- `party.html` must only contain HTML and CSS code. You must write **ALL** JavaScript code inside `party.js` (including Vue.js code).
 - JavaScript code written inside `party.html` will **NOT** be considered for grading.
-

First, please **carefully inspect** `party.js`.

- It creates a new **Vue** app and binds to an HTML element with `id="app"`.
- The **Vue** app already has **data properties** declared.
 - Please NOTE that you (as a developer) should NOT manually edit `data()` section inside `party.js`.
 - Your HTML and JavaScript code **CAN** access these data properties (e.g. read, edit).

Part A: Complete party.html

Edit `party.html` such that when rendered in a web browser, the web page displays the following:

party.html

Hola Amigos!

Enter email addresses
(separated by comma)

alice@smu.edu.sg,
bob@gmail.com

Add to Directory

Select	Email Address	Category
<input type="checkbox"/>	frank@smu.edu.sg	EDUCATION
<input checked="" type="checkbox"/>	kyong@smu.edu.sg	EDUCATION
<input checked="" type="checkbox"/>	jennie@bpink.info	OTHERS
<input type="checkbox"/>	layfoo@kpop.org	OTHERS

Party Invitation

Subject

Year-End KBBQ Party

Message

Dear friends! The year is ending, and it's party time! Please accept this invitation by 10 December 2021 yah? No need to bring anything except your wonderful hungry and thirsty self! Hope to see you and best wishes to you!

Send Invitation

1. Header <h1> text

- Please do NOT hardcode the header text.
- HINT: Which Vue directive can you use to retrieve the information from an existing Vue app **data property**?

2. Enter email addresses

- Please do NOT hardcode the email addresses shown above.
- HINT: Which Vue directive can you use to retrieve the information from an existing Vue app **data property**?
- Later on, when the user keys in other email addresses, your web application should be able to take this **new data** and update the corresponding Vue app **data property**.

3. Table of email addresses

- Please do NOT hardcode the table data inside `<tbody>...</tbody>`.
- HINT: Which Vue directives can you use to retrieve the information from an existing Vue app **data property**?

4. Party Invitation**a. Subject**

- Please do NOT hardcode the input field value.
- HINT: Which Vue directive can you use to retrieve the information from an existing Vue app **data property**?
- Later on, when the user edits the input field value, your web application should be able to take this **new data** and update the corresponding Vue app **data property**.

b. Message

- Please do NOT hardcode the header text.
- HINT: Which Vue directive can you use to retrieve the information from an existing Vue app **data property**?
- Later on, when the user edits the textarea value, your web application should be able to take this **new data** and update the corresponding Vue app **data property**.

Part B: Complete add_to_directory()

Complete `party.html` and `party.js` such that when the user clicks on **Add to Directory** button, your web application will add the user input to the **Table** (on the right-hand side).

For example, see the above screenshot where the user keyed in **2 new email addresses** in the **textarea**:

alice@smu.edu.sg, bob@gmail.com

Next, user clicked on **Add to Directory** button. `party.html` shows that the new email addresses have been added to the **Table** (on the right-hand side) as shown below:

party.html

Hola Amigos!

Enter email addresses
(separated by comma)

Add to Directory

Select	Email Address	Category
<input type="checkbox"/>	frank@smu.edu.sg	EDUCATION
<input checked="" type="checkbox"/>	kyong@smu.edu.sg	EDUCATION
<input checked="" type="checkbox"/>	jennie@bpink.info	OTHERS
<input type="checkbox"/>	layfoo@kpop.org	OTHERS
<input type="checkbox"/>	alice@smu.edu.sg	EDUCATION
<input type="checkbox"/>	bob@gmail.com	OTHERS

Party Invitation

Subject

Year-End KBBQ Party

Message

Dear friends! The year is ending, and it's party time! Please accept this invitation by 10 December 2021 yah? No need to bring anything except your wonderful hungry and thirsty self! Hope to see you and best wishes to you!

Send Invitation

NOTE: If an email addresses contains **edu** as part of the **domain name** (e.g. smu.edu.sg, nyu.edu, etc.), the **email** is **categorized** into **“EDUCATION”**. Otherwise, we classify the email into **“OTHERS”** category.

Part C: Complete send_party_invitation()

Complete **party.html** and **party.js** such that when the user clicks on **Send Invitation** button, your web application will pop up an alert message in the web browser.

Suppose that the user selected **3 people** to invite to his party. He used checkbox buttons in the **Table** to make the selections:

- jennie@bpink.info
- layfoo@kpop.org
- alice@smu.edu.sg

The user edited the **Subject** to **Year-End KBBQ Party at Club XYZ**.
Further, he edited the **Message** content.

party.html
(BEFORE clicking on Send Invitation button)

Enter email addresses
(separated by comma)

Add to Directory

Select	Email Address	Category
<input type="checkbox"/>	frank@smu.edu.sg	EDUCATION
<input type="checkbox"/>	kyong@smu.edu.sg	EDUCATION
<input checked="" type="checkbox"/>	jennie@bpink.info	OTHERS
<input checked="" type="checkbox"/>	layfoo@kpop.org	OTHERS
<input checked="" type="checkbox"/>	alice@smu.edu.sg	EDUCATION
<input type="checkbox"/>	bob@gmail.com	OTHERS

Party Invitation

Subject

Year-End KBBQ Party at Club XYZ

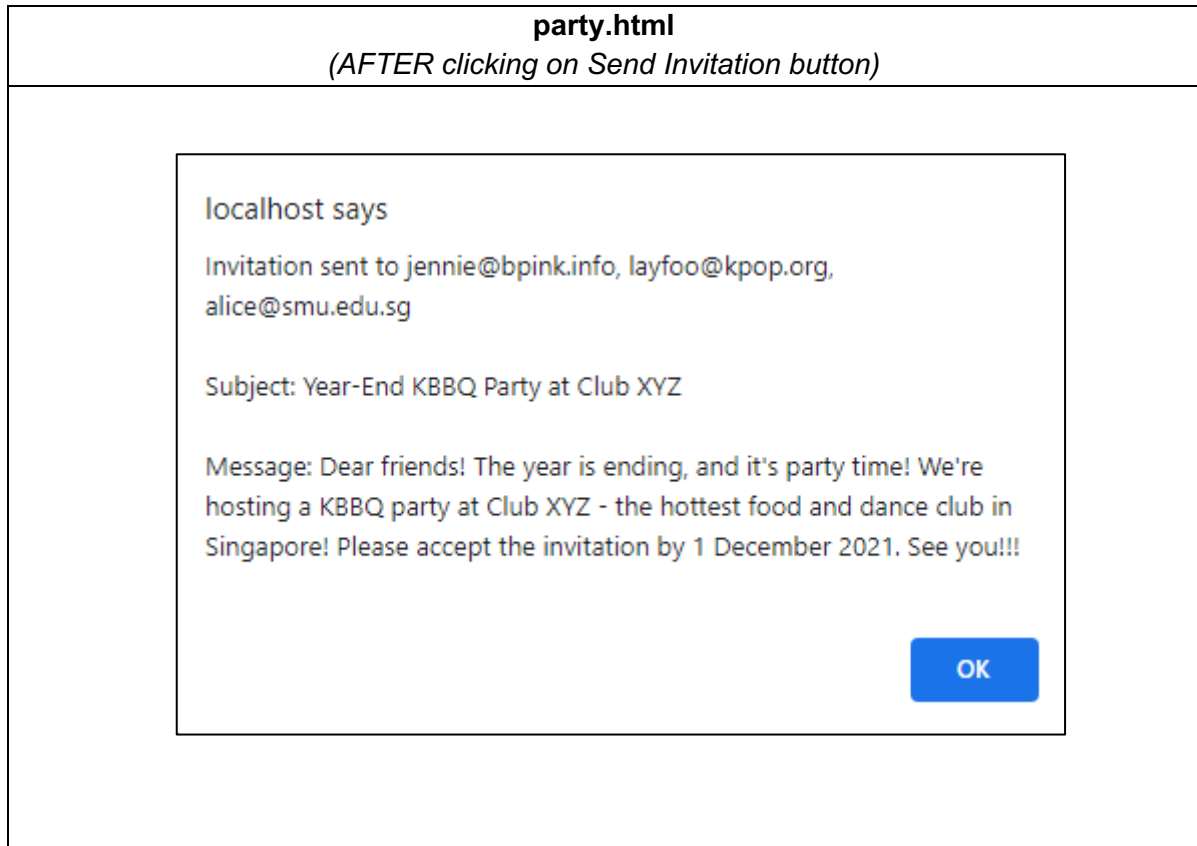
Message

Dear friends! The year is ending, and it's party time! We're hosting a KBBQ party at Club XYZ - the hottest food and dance club in Singapore! Please accept the invitation by 1 December 2021. See you!!!

Send Invitation

All right! The **invitation** is ready to go out.
The user clicks on **Send Invitation** button.

`party.html` pops up an **alert** as shown below:



HINT: When creating a String, you can use `\n` (newline character) to create new lines to separate texts.