

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 129.0.x or later. Your graders will be using only **Google Chrome Web Browser** (Version 129.0.x or later) to test your web pages.
- **No questions will be entertained** by the IS216 teaching team (faculty/instructor/invigilators) 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, 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/methods. 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.2022, 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.2022 --></pre>	<pre>/* Name: KIM Jong Un Email: kim.jongun.2022 */</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://academy.smu.edu.sg/code-conduct>

Submission Instructions

- **Due Date**
 - **XX November 2024 (xxx) 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.2022.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/398535>
 - 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.
-

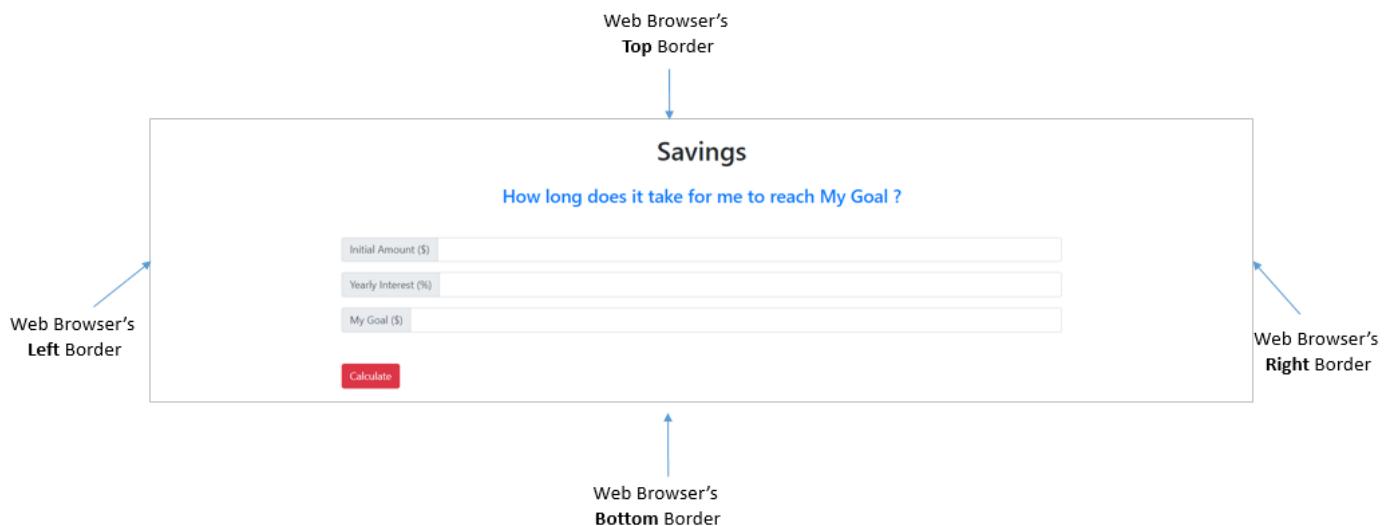
Part A: Complete savings.html

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

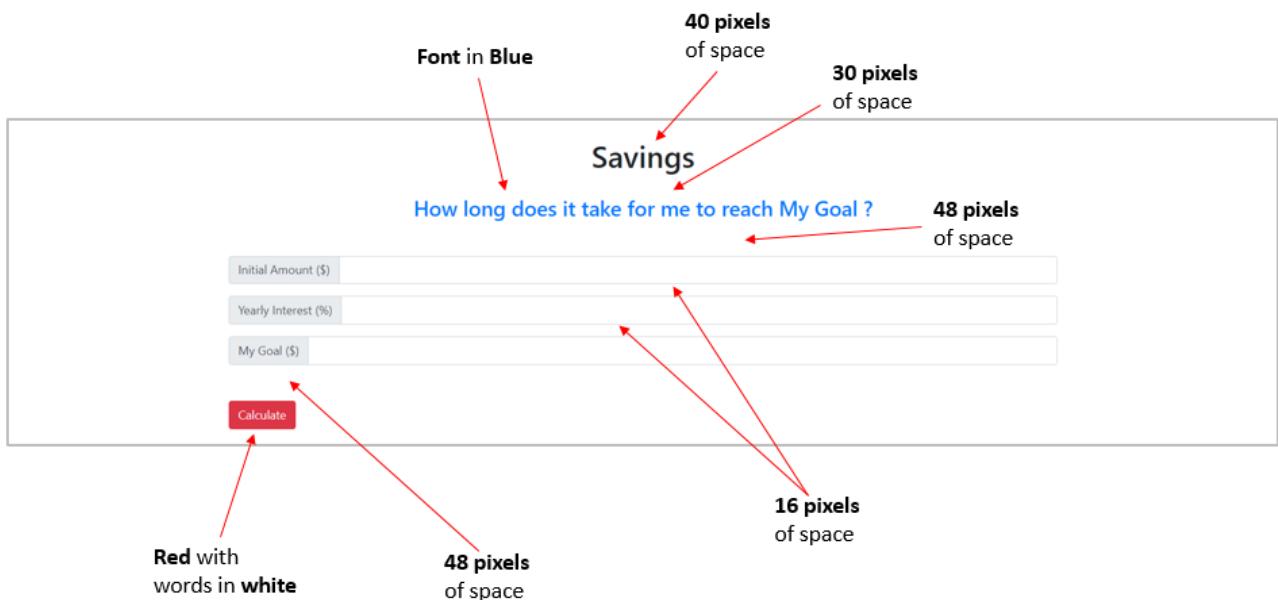
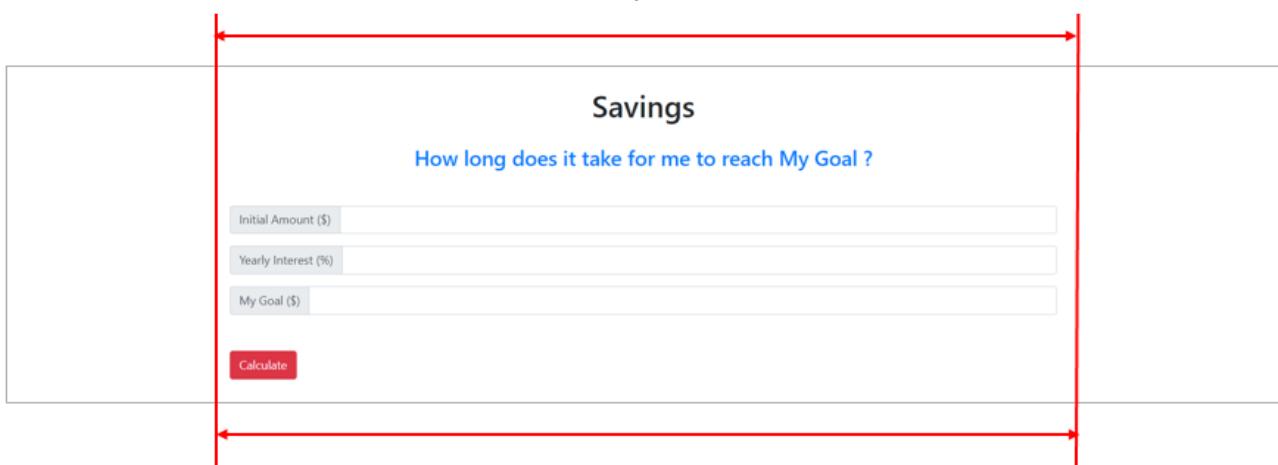
NOTE: The three **input fields** must only allow for **numeric values** (not text). You may **ASSUME** that the user will always enter numeric values greater than 0 (thus, no negative values).

savings.html
<h2>Savings</h2> <p>How long does it take for me to reach My Goal ?</p> <div style="margin-top: 20px;"> <input style="width: 100%; height: 30px; border: 1px solid #ccc; padding: 5px; margin-bottom: 5px; border-radius: 5px; font-size: 14px; font-family: inherit; outline: none; color: inherit; background-color: inherit;" type="text"/>Initial Amount (\$) <input style="width: 100%; height: 30px; border: 1px solid #ccc; padding: 5px; margin-bottom: 5px; border-radius: 5px; font-size: 14px; font-family: inherit; outline: none; color: inherit; background-color: inherit;" type="text"/>Yearly Interest (%) <input style="width: 100%; height: 30px; border: 1px solid #ccc; padding: 5px; margin-bottom: 5px; border-radius: 5px; font-size: 14px; font-family: inherit; outline: none; color: inherit; background-color: inherit;" type="text"/>My Goal (\$) </div> <div style="text-align: center; margin-top: 10px;"> <input style="background-color: #e63366; color: white; border: 1px solid #e63366; padding: 5px 15px; border-radius: 5px; font-size: 12px; font-weight: bold;" type="button" value="Calculate"/> </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)

Savings

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

Initial Amount (\$)	5000
Yearly Interest (%)	10
My Goal (\$)	8000

Calculate

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)

Savings

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

Initial Amount (\$)	5000
Yearly Interest (%)	10
My Goal (\$)	8000

Calculate

Result

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 (\$)	<input type="text"/>
Yearly Interest (%)	<input type="text"/>
My Goal (\$)	<input type="text"/>

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 (\$)	<input type="text"/>
Yearly Interest (%)	<input type="text"/>
My Goal (\$)	<input type="text"/>

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:

Savings

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

Initial Amount (\$)	5000
Yearly Interest (%)	10
My Goal (\$)	8000

30 pixels
of space
Calculate
20 pixels
of space

You will achieve your goal in (years):	5
You will get (\$):	8052.55

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	Result <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 5px;">You will achieve your goal in (years):</td> <td style="padding: 5px;">1</td> </tr> <tr> <td style="padding: 5px;">You will get (\$):</td> <td style="padding: 5px;">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	Result <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 5px;">You will achieve your goal in (years):</td> <td style="padding: 5px;">6</td> </tr> <tr> <td style="padding: 5px;">You will get (\$):</td> <td style="padding: 5px;">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	Result <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 5px;">You will achieve your goal in (years):</td> <td style="padding: 5px;">0</td> </tr> <tr> <td style="padding: 5px;">You will get (\$):</td> <td style="padding: 5px;">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					

		Result
4	Leaving any of the three input fields EMPTY	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

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

Promo code Redeem

payment.html
(Web Browser Width 991 pixels)



Checkout

Billing address

First name

Last name

Address

Country

Postal Code

Credit Card

Name on card

Credit card number

Full name as displayed on card

Expiration

CVV

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

**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

Promo code

Redeem

Billing address

First name

Last name

Address

1234 Main St

Country

Choose... ▾

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; width: fit-content;"> <code>["z",12,"15"]</code> </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()`.

1. The function takes in 3 parameters:
 - o `url1` - URL for a web API
 - o `url2` - URL for another web API
 - o `ele_id` - valid id of an existing HTML element.
2. Retrieve two lists from the web APIs indicated by parameters `url1` and `url2`.
 - o 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.
 - o The given `server/max.php` (described above) is an example of such an API and is used in `max.html` for testing this function.
3. 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]

[error]*

E.g.

no_such_url.php
Error: Request failed with status code 404

4. Find the maximum aka largest number from the 2 lists combined.
 - o Number includes positive, negative integers and decimals.
 - o Note: "15" is a string, not a number.
5. 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]

[url2] : [list from url2]

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

6. 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\
- ◆ ~/src/components/Party.vue
 - ◆ All other files

You are to **edit Party.vue 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.

- You must write **ALL** JavaScript code inside **Party.vue**.
 - There are sections marked “DO NOT EDIT”. You are to leave these code snippets untouched.
-

First, please **carefully inspect** Party.vue.

- The **Vue** component already has **data properties** declared.
 - Please NOTE that you (as a developer) should NOT manually edit data() section.
 - Your HTML and JavaScript code **CAN** access these data properties (e.g. read, edit).
- Vue router has been configured to map the root path to this component. E.g. <http://localhost:5173/>
 - Note that the port number is dependent on your laptop.

Part A: Complete Party component

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

The screenshot shows a web application interface for managing a party invitation list. At the top, a yellow header bar contains the text "Enter email addresses (separated by comma)" and a text input field containing "alice@smu.edu.sg, bob@gmail.com". Below this is a blue button labeled "Add to Directory". To the right is a table with columns "Select", "Email Address", and "Category". The table data is as follows:

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

A dark grey overlay at the top right says "To exit full screen, move mouse away". Below the table is a section titled "Party Invitation" with fields for "Subject" (containing "Year-End KBBQ Party") and "Message" (containing a text message about a party). At the bottom is a red button labeled "Send Invitation".

1. Header `<h1>` text
 - a. Please do NOT hardcode the header text.
 - b. HINT: Which Vue directive can you use to retrieve the information from an existing Vue **data property**?
2. Enter email addresses
 - a. Please do NOT hardcode the email addresses shown above.
 - b. HINT: Which Vue directive can you use to retrieve the information from an existing Vue **data property**?
 - c. 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 **data property**.
3. Table of email addresses
 - a. Please do NOT hardcode the table data inside `<tbody>...</tbody>`.
 - b. HINT: Which Vue directives can you use to retrieve the information from an existing Vue **data property**?
4. Party Invitation
 - a. Subject
 - i. Please do NOT hardcode the input field value.
 - ii. HINT: Which Vue directive can you use to retrieve the information from an existing Vue **data property**?
 - iii. Later, when the user edits the input field value, your web application should be able to take this **new data** and update the corresponding Vue **data property**.

b. Message

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

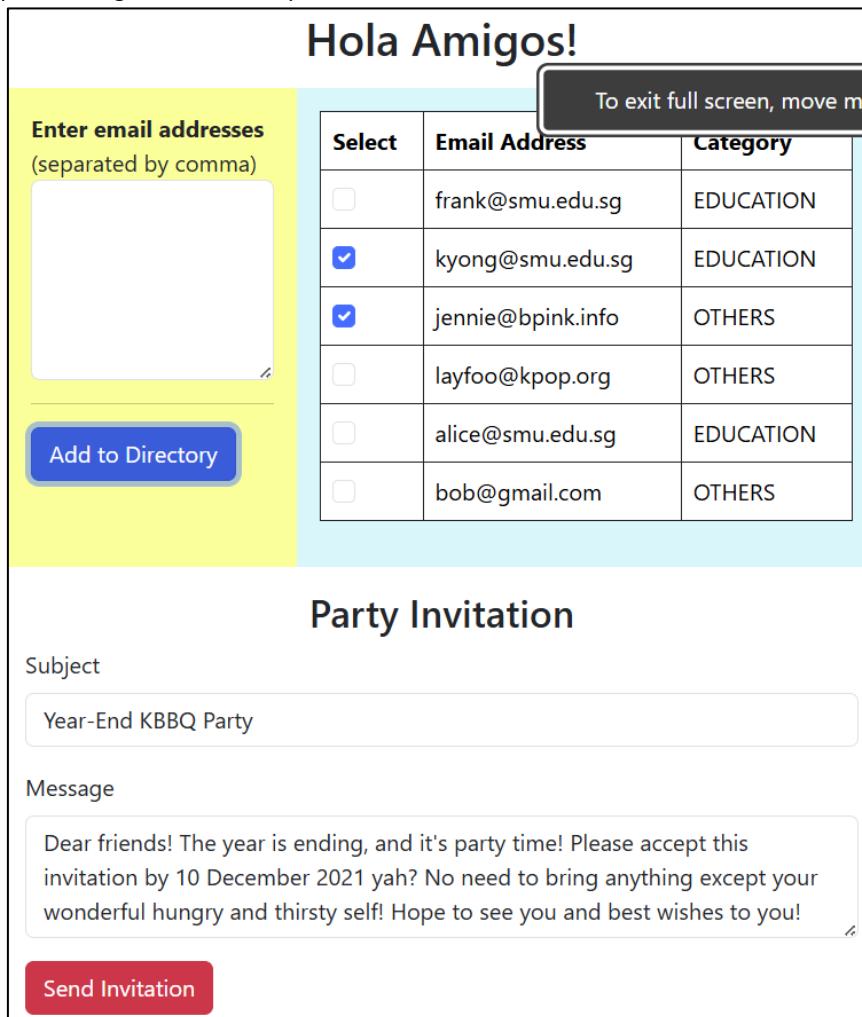
Part B: Complete `add_to_directory()`

Complete `Party.vue` 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. The page shows that the new email addresses have been added to the **Table (on the right-hand side)** as shown below:



The screenshot shows a web application interface. On the left, there is a yellow sidebar with the title "Enter email addresses (separated by comma)" and a text area containing "alice@smu.edu.sg, bob@gmail.com". Below this is a blue "Add to Directory" button. On the right, there is a main content area titled "Hola Amigos!". It contains a table with the following data:

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

Below the table, there is a section titled "Party Invitation" with fields for "Subject" (containing "Year-End KBBQ Party") and "Message" (containing "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!"). At the bottom is a red "Send Invitation" button.

NOTE: If an email address 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.vue` 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 be invited 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**.

And he edited the **Message** content.

(BEFORE clicking on Send Invitation button)

Hola Amigos!

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

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

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

An **alert** pops up (to simulate sending of email) as shown below:

(AFTER clicking on Send Invitation button)



HINT: When creating a String, you can use `\n` (newline character) to create new lines to separate texts or use template string delimited with backtick (`) characters.