

---

# Systems Report (Agile)

## Topic: SRC Voting System

## Class: 12SDD5

---

Sprint: 2

Group Number: 5

Group Members: Henil, Aryan

Client: Ms. Mercado

Supervisor: Mr. Montalban

---

A screenshot of a code editor showing a portion of a JavaScript file named index.html. The code defines a function that handles drag-and-drop events on a file drop zone. It uses event listeners for 'dragenter', 'dragleave', 'dragover' (with preventDefault), and 'drop'. The 'dragover' listener adds a 'solid-border' class to the drop zone. The 'drop' listener removes this class and handles files using the DataTransfer API, setting values and attributes for each file and appending them to the drop zone.

# Auto-Table of Contents Page:

<b>Project Information and Overview</b>	<b>2</b>
<b>1. Planning and Designing</b>	<b>4</b>
1.1 - Needs Analysis (Requirements of user - Ms. Mercado)	4
1.2 - Algorithm - flowchart/pseudocode	4
1.3 - Variables and data types	4
1.4 - Interface Design	4
1.5 - Desk Check	5
1.6 - Context Diagram	5
1.7 - Data Flow Diagram	5
1.8 - Gantt Chart	5
<b>2. Implementing</b>	<b>5</b>
2.1 - Screenshots of program working	5
2.2 - Screenshots of code	6
<b>3. Testing and Evaluating Software Solutions</b>	<b>6</b>
3.1 - Evaluation of software against needs analysis	6

## Project Information and Overview

- Project leader - Henil
- Software Development leader - Henil and Aryan
- Documentation leader - Aryan and Henil
- Division of labour:
  - Systems report sections:
    - Needs Analysis (Requirements) - H, A
    - Algorithms (Flowchart/pseudocode) - A
    - Context Diagram - A
    - Data-Flow Diagram - H
    - Variables and Data types - H, A
    - **Interface Design** - H, A
    - **Desk check** - H
    - Gantt chart - H, A

→ Implementation - Henil and Aryan

→ Evaluation of software against needs analysis - Henil and Aryan

- Code - Henil
- Communications with client - Henil and Aryan

- Group rules:

- Respect each other
- Listen actively to each other
- If a problem occurs, then attack the problem, not the person
- Communication and collaboration is key
- Obey correct Workplace Health and Safety (WH&S) and Ergonomics procedures in the workplace

# 1. *Planning and Designing*

## 1.1 - Needs Analysis (Requirements of client - Ms. Mercado)

The purpose of this system is fully based on APHS. The aim of this system is to provide APHS with better software for students and staff to vote for SRC members every year (instead of voting on Google Forms). The needs/requirements of the system are given below in a table according to the client and based on client meetings.

Must Have	Should Have	Could have
A preference and SRC members list that users can select.	A description and speech video of all the SRC members standing for vote.	A database that contains all the SRC members' details (like description, speech video, subjects they do, etc).
Only a student/staff of APHS can vote their preferences for the SRC members.	A prompt box appears for entering a unique key given to the client to ensure that the user is a client.	Member 1/7, 2/7, ... (keep up to time of what member is displayed each time).
The iteration must go on until		A tally system and results

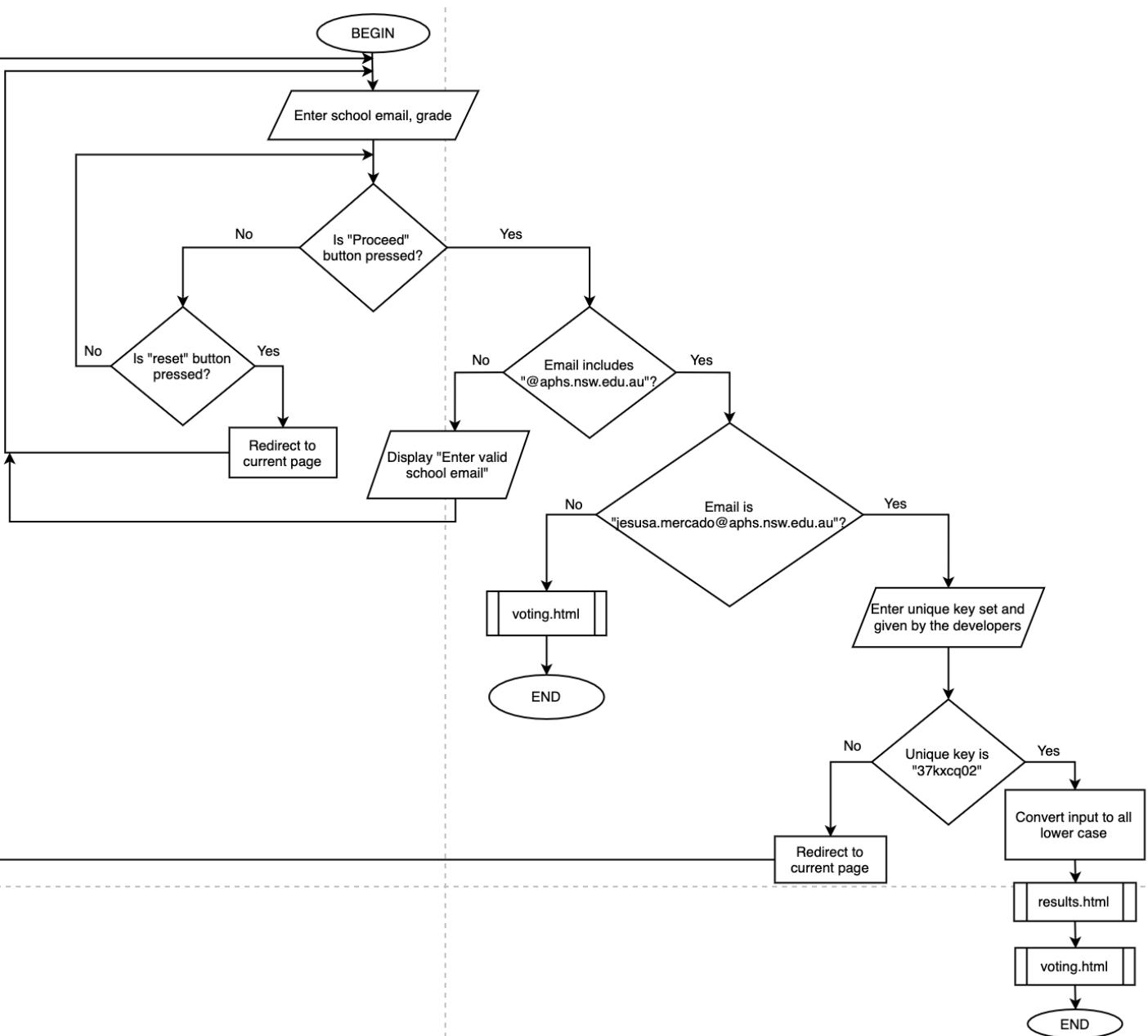
a simple SRC member got the most effective preferences.		table that only gets displayed to Ms. Mercado and no one else.
Each member's container should appear one after other after the user selects his preference and then clicks the "next" button		

Please see both meeting minutes 1 and meeting minutes 2 document file in the folder to get a more depth and detailed view of needs/requirements of the client in regards to this system.

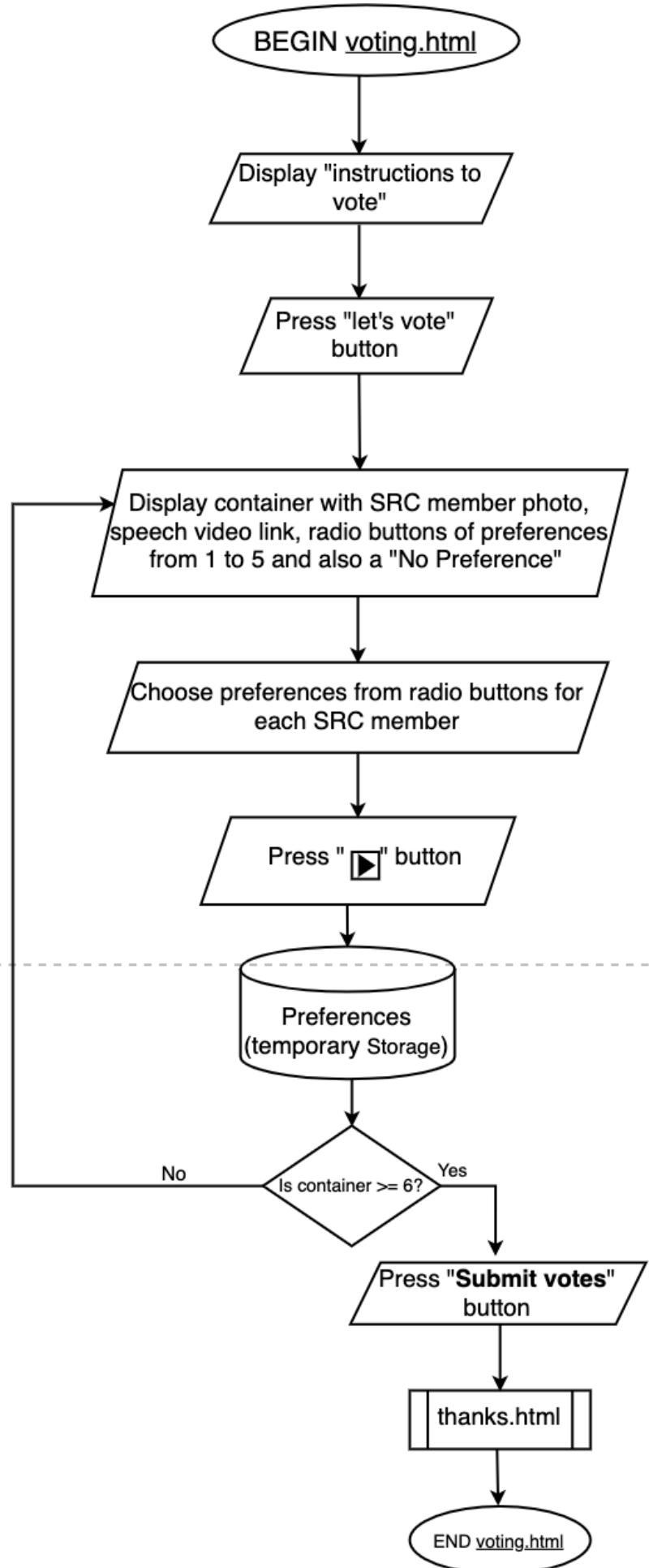
## 1.2 - Algorithm - flowchart/pseudocode

Link: [Our flowchart \(MODIFIED\) link](#)

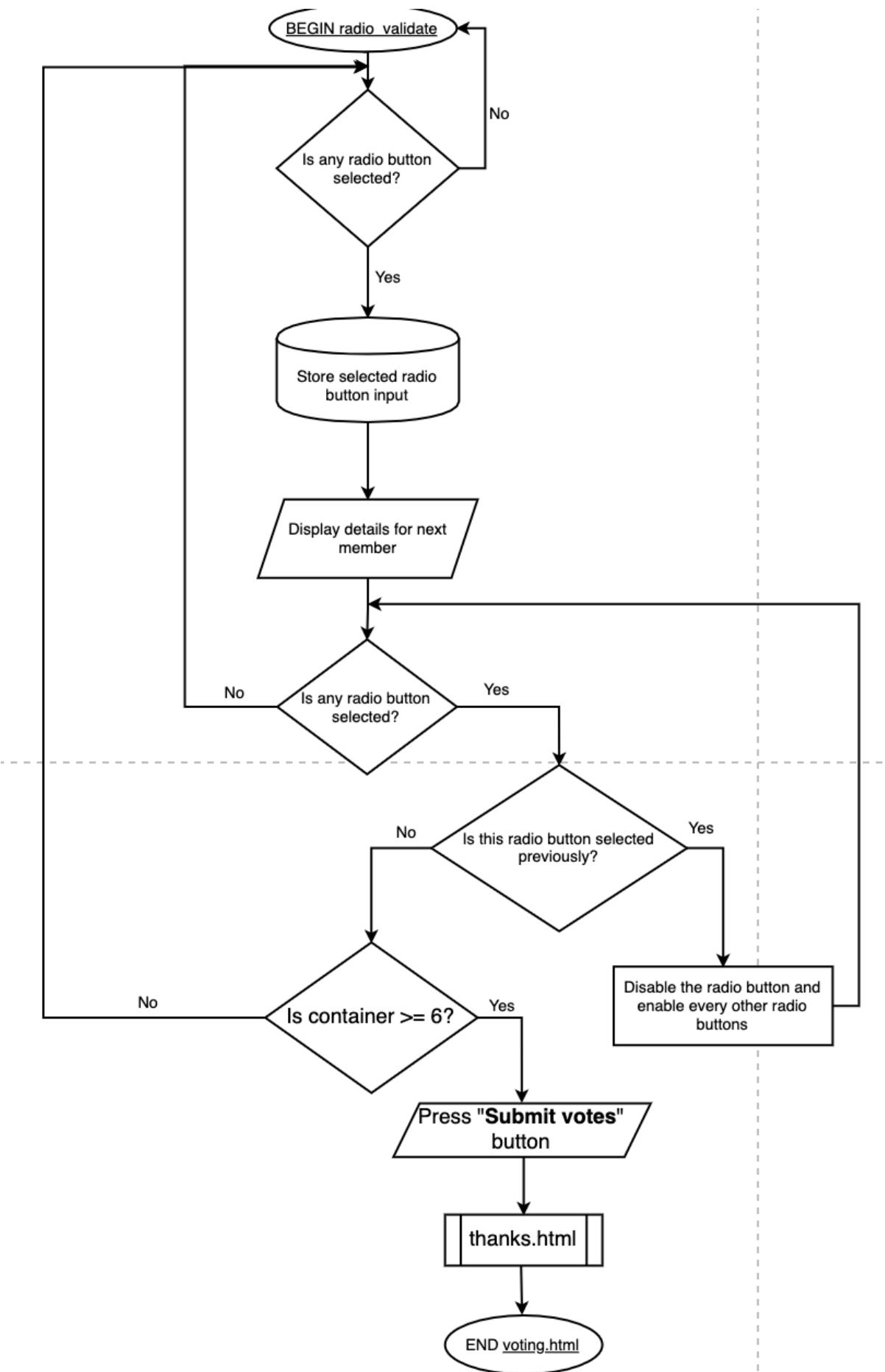
Flowchart 1: [\*index.html\*](#)



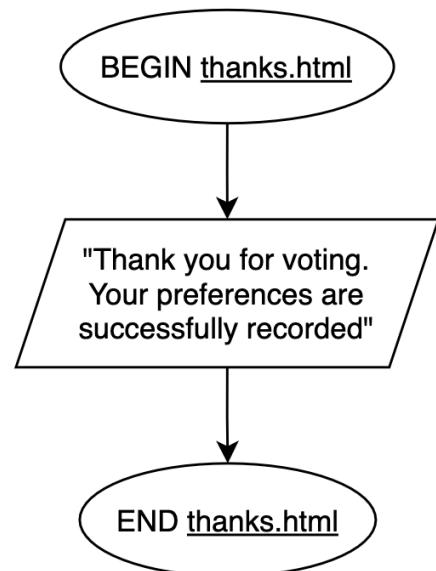
*Flowchart 2: voting.html*



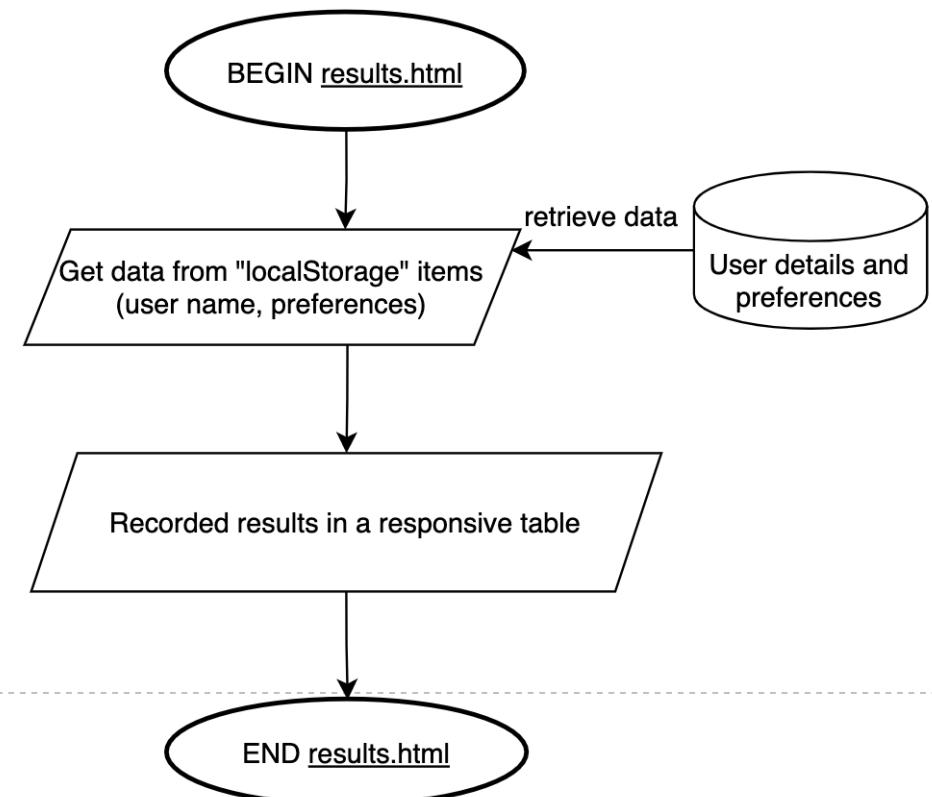
## Flowchart for radio button validation:



*Flowchart 3: thanks.html*



*Flowchart 4: results.html*



## 1.3 - Variables and data types

Variable	Data Type	Description
mail	String	The email of the user who is voting their preferences on SRC members.
grade	Integer	The grade that the user is currently voting in APHS.
key	String	A unique identification key given only to the client to login and access the results.
details	String	
total	Integer	The final totals of each member's 1st, 2nd, 3rd, 4th, 5th and no preferences.
calculate	Integer	The calculations and processes of finding the SRC member who gets the highest 1st preferences in votes.
tally	String	The tallied display of the total of each member's 1st, 2nd, 3rd, 4th, 5th and no preferences.
result	Text	The final display of results of who gets the most 1st preferences and who is the captain so far until the client closes all the responses.
one, two, three, four, five, NA	Boolean	Preferences (from 1 to 5 and No preference) for member 1
one1, two1, three1, four1, five1, NA1	Boolean	Preferences (from 1 to 5 and No preference) for member 2
one2, two2, three2, four2, five2, NA2	Boolean	Preferences (from 1 to 5 and No preference) for member 3
one3, two3, three3, four3, five3, NA3	Boolean	Preferences (from 1 to 5 and No preference) for member 4
one4, two4, three4, four4, five4, NA4	Boolean	Preferences (from 1 to 5 and No preference) for member 5
one5, two5, three5, four5, five5, NA5	Boolean	Preferences (from 1 to 5 and No preference) for member 6

## 1.4 - Interface Diagram

Interface Diagram/Screen Design of index.html

Welcome to the Official School Representative Council Voting Website of  
Arthur Phillip High School



Enter your email

Select your grade

← Animated background

Interface Diagram/Screen Design of index.html (to verify that the actual client is accessing it)

Welcome to the Official School Representative Council Voting Website of

127.0.0.1:5500 says

Enter unique key



Enter your email

Select your grade

Client email entered

Proceed button clicked

← Animated background

## **Instructions to vote:**

- 1.** Choose your preference by selecting any radio button from 1 (your most favourite) to 5 (your least favourite)
- 2.** You can also select "No preference" radio button if you don't feel like or you're confused in which one to select for a particular SRC member shown
- 3.** Once you have chosen your preference for a particular SRC member, you can click the "next member" button to proceed to next member. *NOTE: you cannot proceed to vote for the next member until you select any preference*
- 4.** After reviewing your votes, you can click the "submit votes" button to submit your final preferences for every member
- 5.** Your preferences will then be recorded

Let's vote

**SRC Member 1**

Picture of  
SRC  
Member 1

[MEMBER NAME]  
[MEMBER DESCRIPTION]

[\[MEMBER SPEECH LINK\]](#)

*Preferences*

1	<input type="radio"/>
2	<input type="radio"/>
3	<input type="radio"/>
4	<input type="radio"/>
5	<input type="radio"/>
No Preference	<input type="radio"/>

▶

Interface Diagram /Screen Design of thanks.html



Interface Diagram/Screen Diagram of results.html

User Email	User Grade	Preference for Charminkumar Patel	Preference for Henry Lim	Preference for Malika Azimi	Preference for Simardeep Gill	Preference for Mouna Matri	Preference for Lurcrecia Phung		
kraj@aphs.nsw.edu.au	11	1	2	4	4	3	2		
ecoco@aphs.nsw.edu.au	Teacher	2	3	5	No Preference	4	1		
tom.cooper@aphs.nsw.edu.au	10	3	2	3	4	No Preference	1		

## 1.5 - Desk Check

index.html								
Mail	Email includes == @aphs.nsw.edu.au	Email matches client's email	Enter unique pin given to client	Grade	Press "proceed" button	Proceed to:	Directed to next page	Display
alex.roy@gmail.com	False	False	N/A (Not a client email)	12	True	N/A	False	alert("Invalid email")
tcooper@aphs.nsw.edu.au	True	False	N/A (Not a client email)	11	True	voting.html	True	Takes the user to the next page
jesusa.mercado@aphs.nsw.edu.au	True	True	37KXC Q02	Teacher	N/A	results.html	N/A	N/A

## voting.html

Display (Member Name)	Display (Member Description)	Display (Member photo)	Preference	Speech link	Press "next member" button	Container > 6	Press "Submit" votes button	Proceed to thanks.html
PATEL, Charmin kumar	Studies in 12th grade, plays school cricket, does Modern history, SDD, IPT, Physics, Maths Ext 1...		1	True (For some reason, this member's speech link is not provided by the client to us)	True	False	False	False

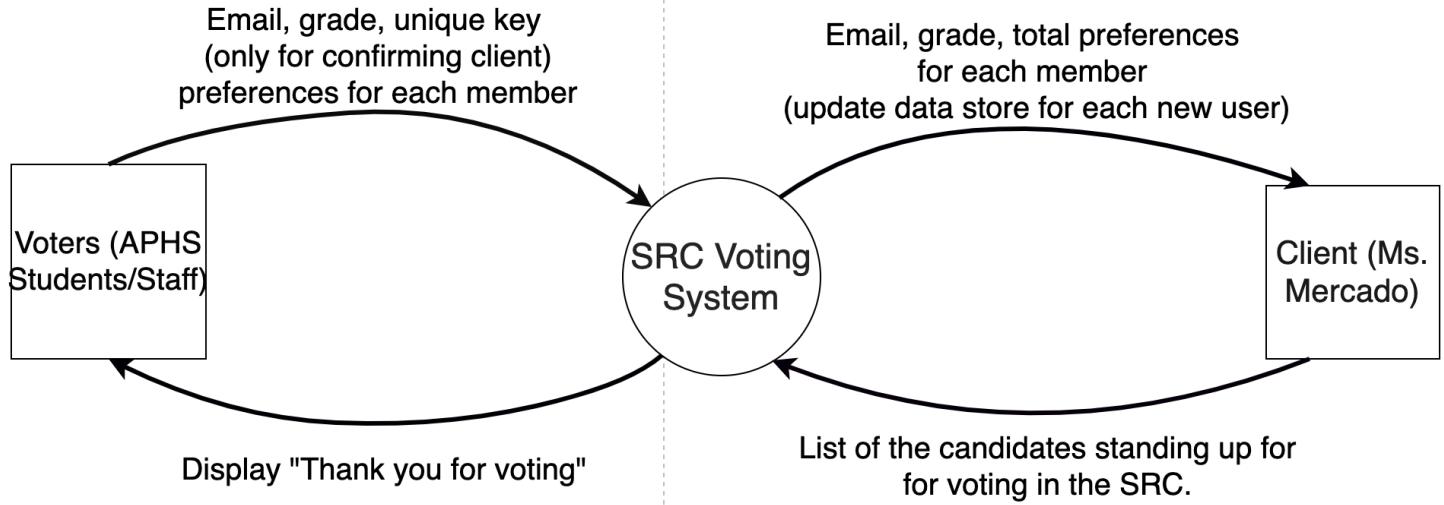
LIM, Henry	Plays school Volleyball, can marry mathematics if he wants ...		2	True ( <a href="#">Henry's speech link</a> )	True	False	False	False
GILL, Simardeep	She likes to participate in extracurricular activities ...		No Preference	True ( <a href="#">Simardeep's speech link</a> )	True	False	False	False
AZIMI, Malika	Ambitious and passionate ...		4	True ( <a href="#">Malika's speech link</a> )	True	False	False	False
MAITRI, Mouna	Very passionate about working hard and the most preferable subject is English ...		5	False ( <a href="#">Mouna's speech link</a> )	True	False	False	False
PHUNG, Lucrecia	Interested in studying business studies ...		3	True ( <a href="#">Lucrecia's speech link</a> )	True	True	True	True

results.html							
User email	User grade	Preference for Charmin	Preference for Henry	Preference for Melika	Preference for Simardeep	Preference for Mouna	Preference for Lucrecia
tom.cooper@aphs.nsw.edu.au	10	3	1	3	2	5	No preference
ecoco@aphs.nsw.edu.au	Teacher	2	1	1	2	3	4
pvraj@aphs.nsw.edu.au	11	2	3	No preference	1	4	5

thanks.html		
email != jesusa@mercado.aphs.nsw.edu.au	Finished voting preference for each SRC member	Display
TRUE	True	Thank you for voting, your preferences are successfully recorded.
FALSE	False	N/A
TRUE	False	N/A

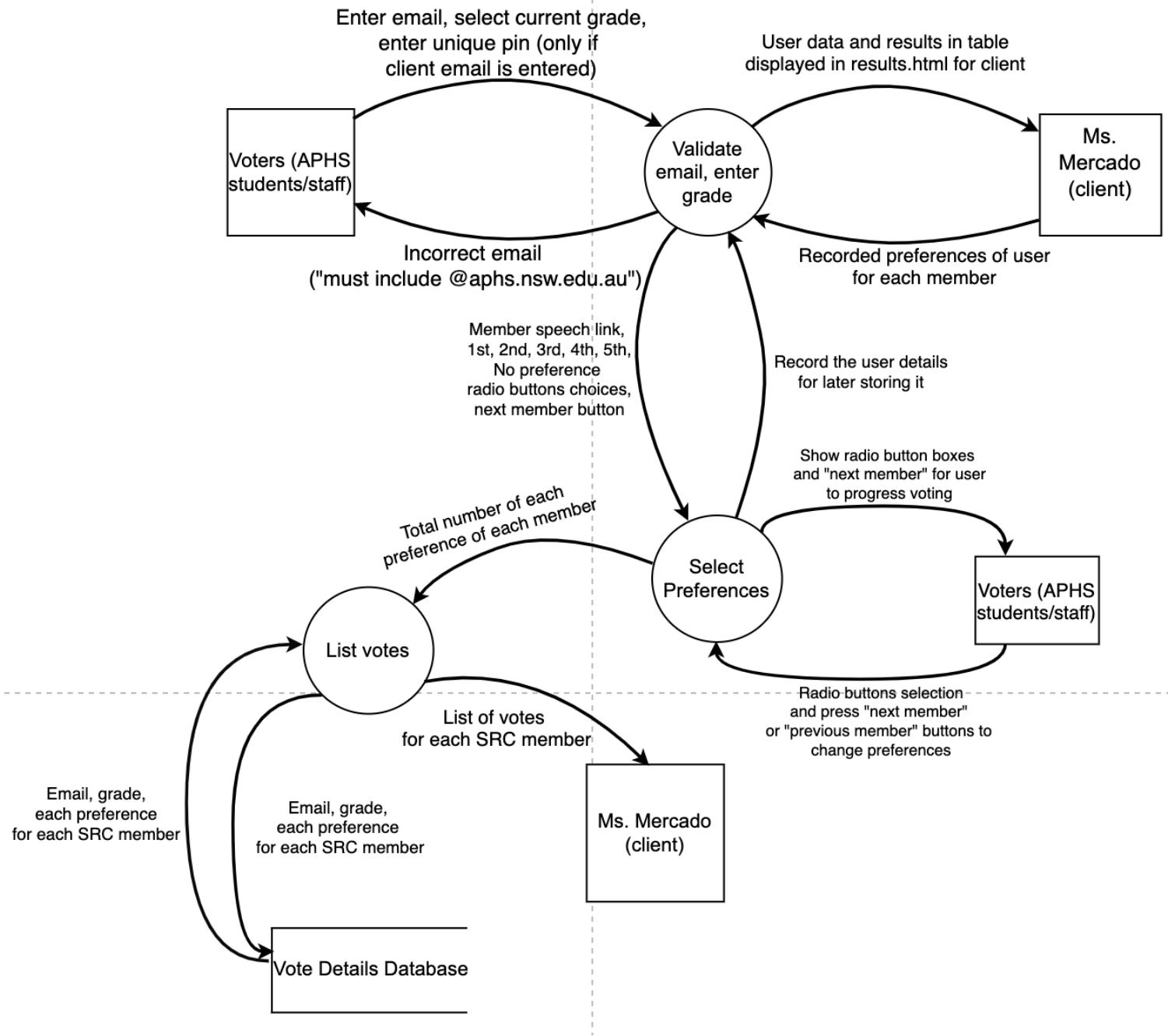
## 1.6 - Context Diagram

### 1.6 - Context Diagram for SRC Voting System



## 1.7 - Data Flow Diagram

### 1.7 - Data-Flow Diagram for SRC Voting System



## 1.8 - Gantt Chart

PROJECT TITLE		SRC Voting System for APHS					COMPANY NAME		APHS Tech							
PROJECT MANAGER		Henil and Aryan					DATE OF PRODUCTION START		25/02/21							
TASK NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION (Days)	TASK COMPLETION	SPRINT ON									
							WEEK 6			WEEK 7						
1	Planning and Designing						Mon	Tue	Wed	Thur	Fri	Mon	Tue	Wed	Thur	Fri
1.1	Needs Analysis	Henil, Aryan	12/03/21	15/03/21	3	100%										
1.2	Algorithm - flowchart/pseudocode	Aryan	15/03/21	16/03/21	1	100%										
1.3	Variables and data types	Henil	15/03/21	21/03/21	6	100%										
1.4	Interface Design	Henil, Aryan	16/03/21	22/03/21	6	100%										
1.5	Desk Check	Henil	18/03/21	22/03/21	4	100%										
1.6	Context Diagram	Aryan	19/03/21	22/03/21	3	100%										
1.7	Data Flow Diagram	Aryan	22/03/21	23/03/21	1	100%										
1.8	Gantt Chart	Henil, Aryan	23/03/21	23/03/21	0	100%										
2	Implementing															
2.1	Screenshots of program working	Henil, Aryan	24/03/21	28/03/21	4	100%										
2.2	Screenshots of code	Henil	28/03/21	28/03/21	3	100%										
2.3	User Documentation (Installation Guide)	Henil	29/05/21	29/05/21	0	100%										
3	Testing and Evaluating															

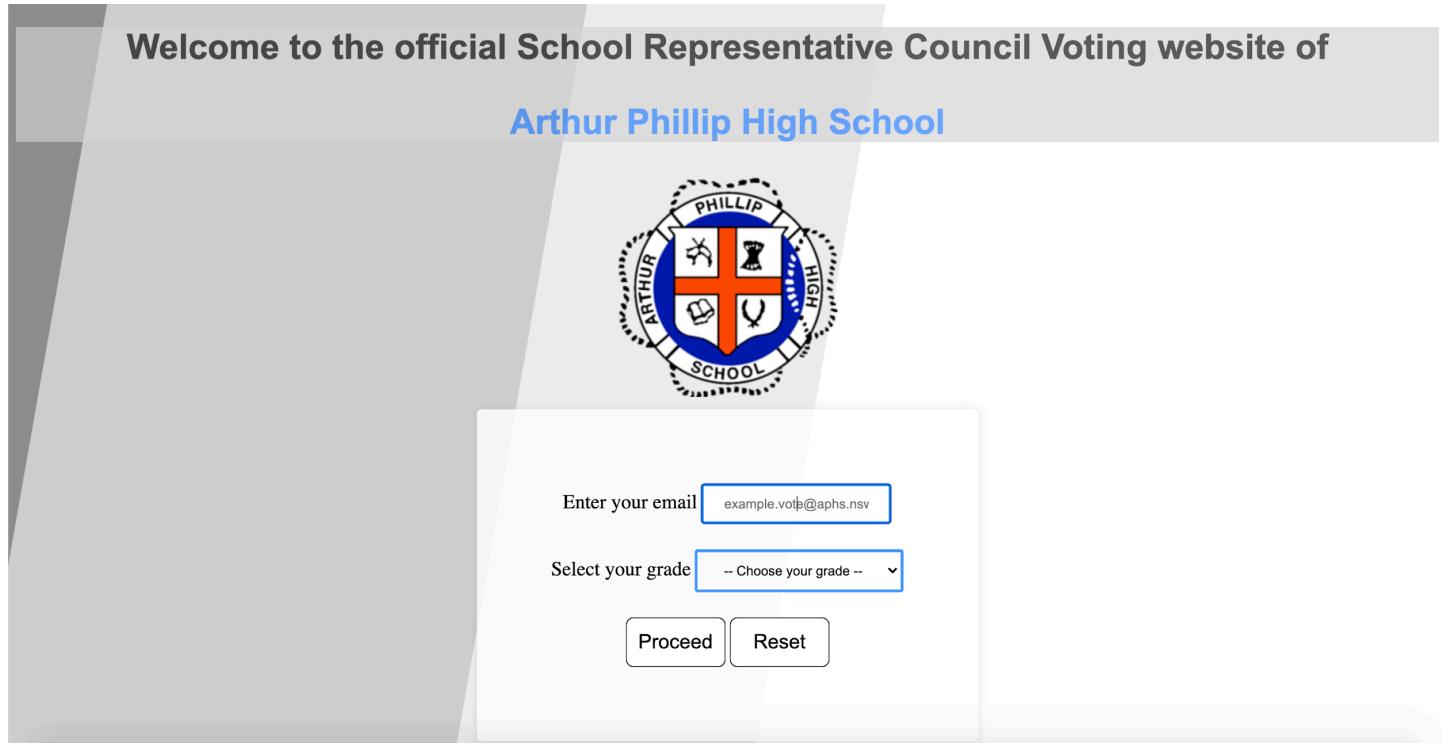
Link to access the Gantt Chart

⇒ [Our Gantt Chart link](#)

## 2. *Implementing*

### 2.1 - Screenshots of program working

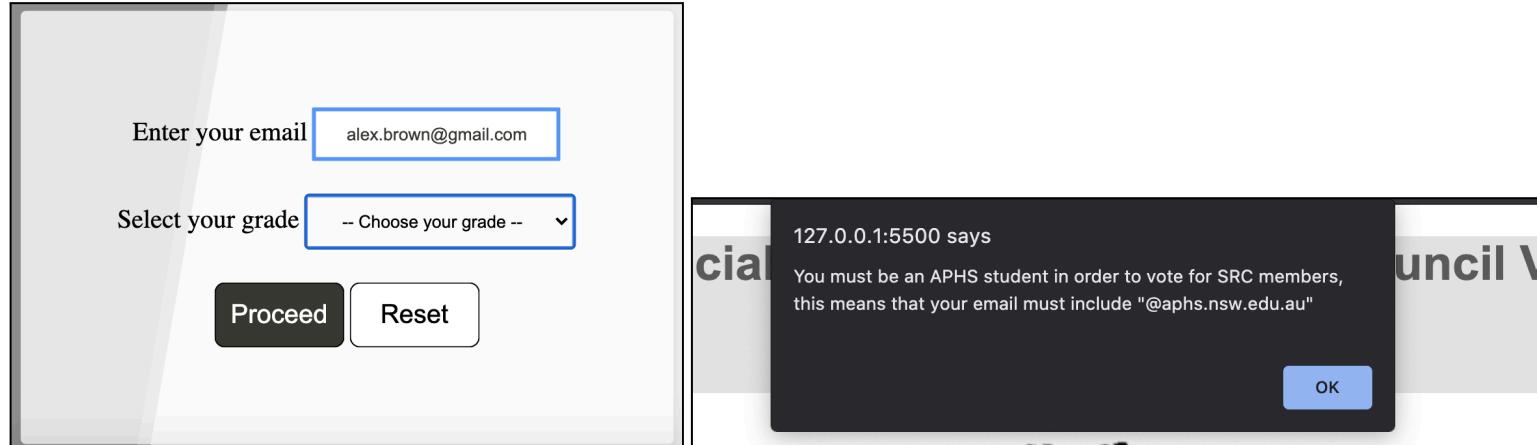
Screenshot 1: index.html (beginning page to user/client):



Explanation of above screenshot 1:

This is the title page of the SRC voting system. This page has a title and an 'Arthur Phillip High School' logo. This page is used as the first page for user login and verification. The main reason is that the client can easily access information by entering her email on this page. The user first enters his/her email address and then selects their grade and presses the submit button. The reason for choosing such a live background with a neutral gray and a cool white colour to become an attraction to the user.

Screenshot 2: index.html (email validation):

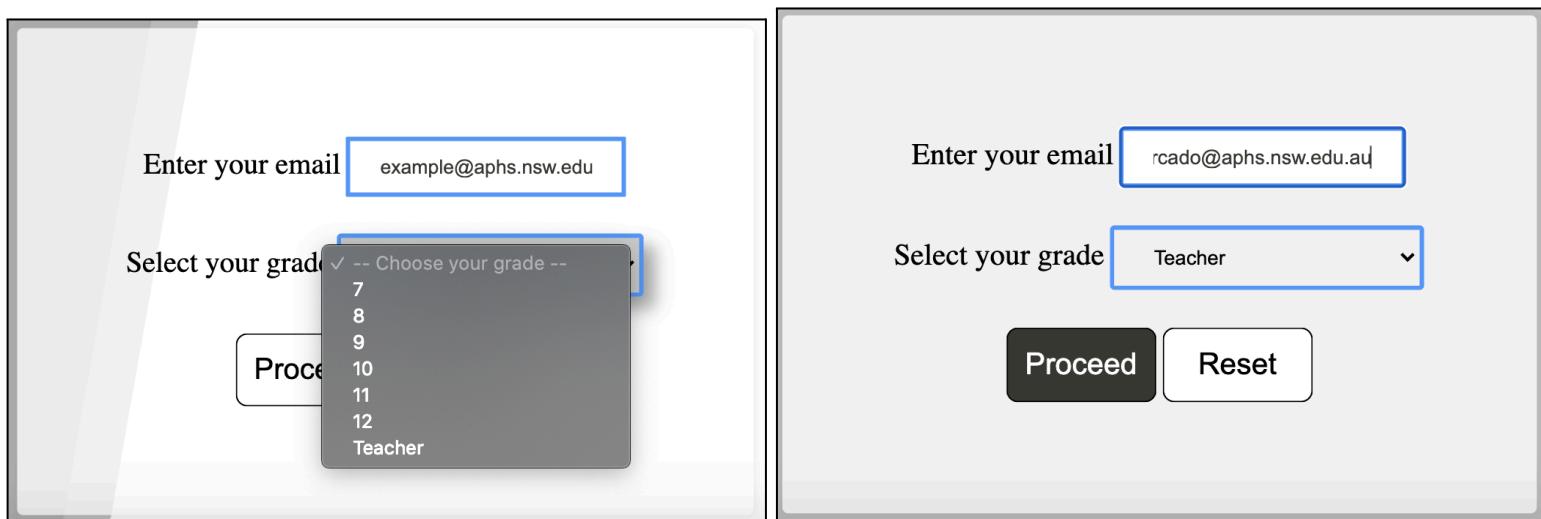


Explanation of above screenshots:

The image above shows how the user has inputted their email address and left the grade blank, this is because the system itself sets the grade to 'Grade 7' which is a bug, the grade if not entered should also have a alert box showing that 'The grade was not entered'

There are two buttons under the 'Select your grade option' where both the buttons have a black and white hovering effect. The proceed button guides the user to the next page of the system, whereas the reset button will reset the email and grade to a blank space. As we saw in the image beside the email entered was 'alex.brown@gmail.com' whereas the email should contain '@aphs.nsw.edu.au'. The reason for alerting the user is that the user who is voting must be a member of our school as per the clients requirement, so that a user outside APHS cannot access and vote for the SRC members.

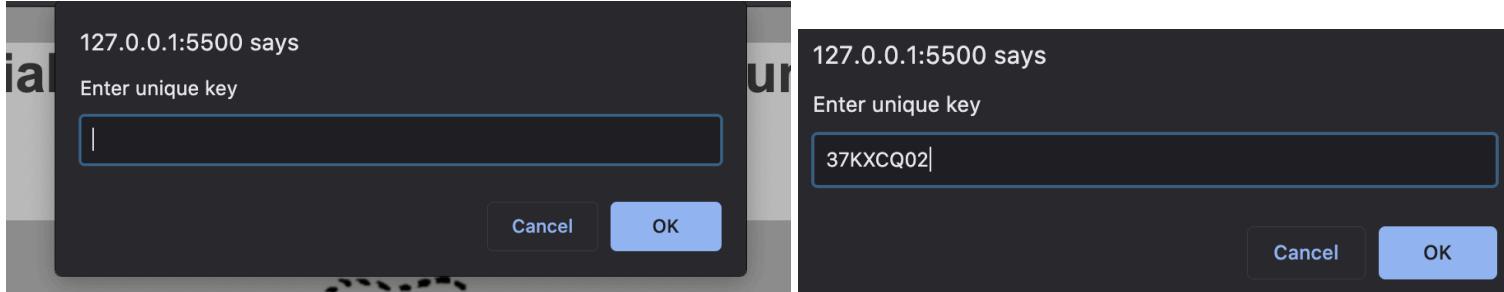
Screenshot 3: index.html (client email entered):



Explanation of above screenshots:

The above images show that the client is about to request the system to access the voting system results. She enters her email address and adds as a 'Teacher' in the 'Enter your grade' field. Then she presses the proceed button. The reason for doing so was that it made it easier for the client to access the results instead of going through the whole voting process and then accessing the results.

Screenshot 4: index.html (unique key identification):



Explanation of above screenshots:

After the client enters her email address and the system checks her email address so that it is a valid email. The images shown are alert boxes which appear as soon as the teacher puts her email address in 'Enter you email', then the system will ask the client to 'Enter a unique key' this will help the system to cross check that it is actually the client who is accessing the results. The reason for adding this security was to protect the results as anyone other than the client can enter the client's email and access the results which would make the results public.

Screenshot 5: voting.html (instructions):

### Instructions for voting:

1. Choose your preference by selecting any radio button from 1 (your most favourite) to 5 (your least favourite)
2. You can also select "No preference" radio button if you don't feel like or you're confused in which one to select for a particular SRC member shown
3. Once you have chosen your preference for a particular SRC member, you can click the "next member" button to proceed to next member. NOTE: you cannot proceed to vote for the next member until you select any preference, you will be able to change your choice later
4. After reviewing your votes, you can click the "submit votes" button to submit your final preferences for every member
5. Your preferences will then be recorded

Let's vote

Explanation of above screenshot 5:

The image above shows that after the user inputs his/her details and they are validated the user is redirected to the next page that is this page. The reason for making this page before directly displaying the candidates. The user reads all the information for the voting process, and then has to press the 'Lets Vote' button. The button and all the instructions disappear and displays the final process of voting for the candidates.

Screenshot 6: voting.html (SRC members containers):

### SRC Member 1



PATEL, Charminkumar

Charminkumar Patel studies in 12th grade and is diligent and tries his best to achieve well in all his subjects and HSC.

[Member's Speech link](#)

**Preference:**

1	<input checked="" type="radio"/>
2	<input type="radio"/>
3	<input type="radio"/>
4	<input type="radio"/>
5	<input type="radio"/>
No Preference <input type="radio"/>	



Explanation of above screenshot 6:

This page is the main page in the SRC voting system, users have to give preferences to each candidate and if they do not wish to provide a given candidate a preference then he/she can click on 'No Preference'. The reason for giving almost all the information to the user is that he/she can make a better decision in choosing their leader. The user then has to press the arrow button to proceed to the next member and give a preference for the next candidate. The user is provided with a photo of the candidate, his full name and a short description of him/her. Users are also provided with the candidates speech link, which then redirects the user to the speech link, where they can listen to the whole speech, this idea was chosen to better help the user to listen to their speech if he/she has missed listening to their speech

## 2.2 - Screenshots of code

### Index.html (before entering details)

```
<body>

    <!-- Greeting to user -->
    <h1>Welcome to the official School Representative Council Voting website of <br><p
id="aphs">Arthur Phillip High School</p></h1>

    <!-- APHS School logo -->
    

    <!-- For background animation purposes -->
    <div class="bg"></div>
    <div class="bg bg2"></div>
    <div class="bg bg3"></div>
    <div class="content"> <!-- A small container in which the form input fields and buttons will get
displayed -->

        <form>
            <!-- Input for getting user email -->
            <div>
                <label for="mail" class="user_mail">Enter your email</label>
                <input id="mail" name="mail" type="email" placeholder="example.vote@aphs.nsw.edu.au"
autofocus required>
            </div>

            <!-- Drop-down box for selection of user's grade -->
            <div>
                <label for="grade" class="voter_grade">Select your grade</label>
                <select name="grade" id="grade" required>
                    <option value="ChooseYear" selected disabled>-- Choose your grade --</option>
                <!-- Default value of drop-down box -->
                    <option value="7" label="7"></option><br>
                    <option value="8" label="8"></option><br>
                    <option value="9" label="9"></option><br>
                    <option value="10" label="10"></option><br>
                    <option value="11" label="11"></option><br>
                    <option value="12" label="12"></option><br>
                    <option value="Teacher" label="Teacher"></option><br> <!-- Even teacher/staff
can vote for SRC members -->
                </select>
            </div>

            <!-- For security purposes, a unique key to confirm that client's email is not
unauthorisedly accessed -->
            <div id="unique_key">
```

```

        <label for="unique" class="uniquekey">Enter the unique identification key given to
you</label>
        <input id="unique" name="unique" type="text" placeholder="1-EX-AM-PLE-07">
</div>

<div>
    <!-- "proceed" button to go to next page according to validation and "reset" button
to reset the details inputted -->
    <button type="submit" id="submit" onclick="next()">Proceed</button>
    <button type="reset" id="reset" onclick="restart()">Reset</button>
</div>

</form>
</div>
</body>

```

Explanation of code above:

The above code snippet is creating a HTML page and context of input fields for user email, and a drop-down box for users selecting their grades (also Teachers are allowed to vote).

### Index.html (after entering normal APHS student details)

```

else {
    // For school security //
    if (email.includes("@aphs.nsw.edu.au")) {

        window.open("./voting.html"); // Direct the user to voting.html if his/her email
includes @aphs.nsw.edu.au //
    }

    else {
        alert("You must be an APHS student in order to vote for SRC members, this means that
your email must include \@aphs.nsw.edu.au\\"");
        location.href = "#"; // Reload user's page and don't let him/her go to the next page
if email doesn't include @aphs.nsw.edu.au //
    }
}

```

EXPLANATION OF CODE ABOVE:

This section of code identifies that a normal user from APHS has entered in his/her details and can go further or not go further if the email entered does not include "@aphs.nsw.edu.au" at the end of their email.

### Voting.html (Instructions to vote)

```

<h1 id="instruction">Instructions for voting: </h1>

<div id="instr">
```

```

<ol>
    <li>Choose your preference by selecting any radio button from 1 (your most favourite) to
5 (your least favourite)</li><br>
    <li>You can also select "No preference" radio button if you don't feel like or you're
confused in which one to select for a particular SRC member shown</li><br>
    <li>Once you have chosen your preference for a particular SRC member, you can click the
"next member" button to proceed to next member. NOTE: you cannot proceed to vote for the next
member until you select any preference</li><br>
    <li>You can click the "submit votes" button to submit your final preferences for every
member</li><br>
    <li>Your preferences will then be recorded</li><br>
</ol>

<style>
#instr {
    text-align:left;
}
</style>
</div>

<button id="move" onclick="move()">Let's vote</button>

```

#### EXPLANATION OF CODE ABOVE:

The above code basically displays the instructions for the users to vote and steps.

#### Voting.html (radio buttons and details for a particular SRC member):

```

<div id="container">
    <h2>SRC Member 1</h2>
    
    <p>PATEL, Charminkumar</p>
    <p>Charminkumar Patel studies in 12th grade and is diligent and tries his best to achieve
well in all his subjects and HSC.</p>
    <a href="#">Member's Speech link</a>

    <div class="form">
        Preference:
        <div class="inputGroup">
            <input value="1" id="one" name="radio" type="radio"/>
            <label for="one">1</label>
        </div>

        <div class="inputGroup">
            <input value="2" id="two" name="radio" type="radio"/>
            <label for="two">2</label>
        </div>
    </div>

```

```

<div class="inputGroup">
    <input value="3" id="three" name="radio" type="radio"/>
    <label for="three">3</label>
</div>

<div class="inputGroup">
    <input value="4" id="four" name="radio" type="radio"/>
    <label for="four">4</label>
</div>

<div class="inputGroup">
    <input value="5" id="five" name="radio" type="radio"/>
    <label for="five">5</label>
</div>

<div class="inputGroup">
    <input value="No Preference" id="NA" name="radio" type="radio"/>
    <label for="NA">No Preference</label>
</div>
</div>

```

#### EXPLANATION OF CODE ABOVE:

The above code basically generates after the “let’s vote” button is pressed after reading the instructions of how to vote. This code runs in a div container where SRC member’s details like (member name, description, SRC photo, SRC speech video link) so that users can get clear information about the members. Radio buttons are displayed to users to select one preference for the SRC member.

#### Voting.html (“>” buttons for proceeding to next member)

```

<div id="member_1">
    <!-- <h2 id="proceed">Next Member</h2> -->
    <button id="member1" onclick="member1()"><i class="far fa-caret-square-right">
    </i></button><br>
</div>

```

#### EXPLANATION OF CODE ABOVE:

The above code defines that if the “previous member” button is pressed, then the user will go back to the previous member. Also the “next member” button is pressed, then the user will go back to the next member.

#### style.css (for radio buttons styling and positioning and clicking):

```

.inputGroup {
    background-color: #fff;
    display: block;
}

```

```

        margin: 10px 0;
        position: relative;
    }

.inputGroup label {
    padding: 12px 20px;
    width: 100%;
    display: block;
    text-align: left;
    color: #3C454C;
    cursor: pointer;
    position: relative;
    z-index: 2;
    transition: color 200ms ease-in;
    overflow: hidden;
}

.inputGroup label:before {
    width: 10px;
    height: 10px;
    border-radius: 50%;
    content: "";
    background-color: #5562eb;
    position: absolute;
    left: 50%;
    top: 50%;
    transform: translate(-50%, -50%) scale3d(1, 1, 1);
    transition: all 300ms cubic-bezier(0.4, 0, 0.2, 1);
    opacity: 0;
    z-index: -1;
    /* border: 2px solid black;
    border-radius: 8px; */
}

.inputGroup label:after {
    width: 32px;
    height: 32px;
    content: "";
    border: 2px solid #D1D7DC;
    background-color: #fff;
    background-image: url("data:image/svg+xml,%3Csvg width='32' height='32' viewBox='0 0 32 32' xmlns='http://www.w3.org/2000/svg'%3E%3Cpath d='M5.414 11L4 12.414L5.414 5.414L20.828 6.414 19.414 51-10 10z' fill='%23fff' fill-rule='nonzero'/%3E%3C/svg%3E ");
    background-repeat: no-repeat;
    background-position: 2px 3px;
    border-radius: 50%;
    z-index: 2;
}

```

```

        position: absolute;
        right: 30px;
        top: 50%;
        transform: translateY(-50%);
        cursor: pointer;
        transition: all 200ms ease-in;
    }

.inputGroup input:checked ~ label {
    color: #fff;
}

.inputGroup input:checked ~ label:before {
    transform: translate(-50%, -50%) scale3d(56, 56, 1);
    opacity: 1;
}

.inputGroup input:checked ~ label:after {
    background-color: #54E0C7;
    border-color: #54E0C7;
}

.inputGroup input {
    width: 32px;
    height: 32px;
    order: 1;
    z-index: 2;
    position: absolute;
    right: 30px;
    top: 50%;
    transform: translateY(-50%);
    cursor: pointer;
    visibility: hidden;
}

.form {
    padding: 0 16px;
    max-width: 550px;
    margin: 50px auto;
    font-size: 18px;
    font-weight: 600;
    line-height: 36px;
}

body {
    background-color: #D1D7DC;
    font-family: "Fira Sans", sans-serif;
}

```

```

}

*,
*::before,
*::after {
    box-sizing: inherit;
}

html {
    box-sizing: border-box;
}

code {
    background-color: #9AA3AC;
    padding: 0 8px;
}

.photo{
    width: auto;
    height: 200px;
}

```

Explanation of above code:

The above code styles, positions, and aligns all the radio buttons for all the SRC members and also implements the selection effects of when the user selects each radio button.

### Thanks.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <script language="javascript" type="text/javascript" src="vote.js"></script>

    <title>Thank you</title>
</head>
<body>

<style>
    div {
        font-size: 50px;
        text-align: center;
        transform: translateY(80%);
    }
    /* reset */

```

```

*,  

*::before,  

*::after {  

margin: 0;  

padding: 0;  

box-sizing: border-box;  

}  
  

body {  

display: flex;  

flex-direction: column;  

justify-content: space-around;  

align-items: center;  
  

min-height: 100vh;  

padding: 20px;  
  

font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell,  

'Open Sans', 'Helvetica Neue', sans-serif;  

}  
  

/* other */  

.info {  

margin: 20px 0;  

text-align: center;  

}  
  

p {  

color: #2e2e2e;  

margin-bottom: 20px;  

}  
  

/* block-$ */  

.block-effect {  

font-size: calc(8px + 6vw);  

}  
  

.block-reveal {  

--t: calc(var(--td) + var(--d));  
  

color: transparent;  

padding: 4px;  
  

position: relative;  

overflow: hidden;
}

```

```

animation: revealBlock 0s var(--t) forwards;
}

.block-reveal::after {
content: '';
width: 0%;
height: 100%;
padding-bottom: 4px;

position: absolute;
top: 0;
left: 0;

background: var(--bc);
animation: revealingIn var(--td) var(--d) forwards, revealingOut var(--td) var(--t) forwards;
}

/* animations */
@keyframes revealBlock {
100% {
color: #0f0f0f;
}
}

@keyframes revealingIn {
0% {
width: 0;
}

100% {
width: 100%;
}
}

@keyframes revealingOut {
0% {
transform: translateX(0);
}

100% {
transform: translateX(100%);
}
}

```

```

}

.abs-site-link {
position: fixed;
bottom: 20px;
left: 20px;
color: rgba(0, 0, 0, 0.6);
font-size: 16px;
}
</style>

<h1 class="block-effect" style="--td: 1.2s">
    <div class="block-reveal" style="--bc: #4040bf; --d: .1s">Thank you for voting!</div>
    <div class="block-reveal" style="--bc: #bf4060; --d: .5s">Your preferences have been
successfully recorded</div>
</h1>

<div class="info">

</div>

<a href="https://abubakersaeed.netlify.app/designs/d12-block-revealing-effect"
class="abs-site-link" rel="nofollow noreferrer"

```

### EXPLANATION OF CODE ABOVE:

This code displays thank you to the user after the user completes the whole process for voting. The CSS in this code is mainly used for designing the background of the page to make it more attractive.

### Results.html

```

<body id="results">
    <h1>SRC Members Vote Results</h1>

    <h1 id="headers">User Email - Charminkumar Patel - Melika Azimi - Henry Lim - Simardeep Gill -
Lucrecia Phung - Mouna Maitri</h1>

    <style>
        #headers {
            font-weight: normal;
            font-size: 15px;
        }
    </style>

```

```

<!-- <button onclick="update()" id="update">View results</button> -->



</div>


</div>


</div>

<style>
body {
    padding: 0;
    margin: 0;
    font-family: Verdana, Geneva, Tahoma, sans-serif;
}

table {
    position: absolute;
    left: 50%;
    top: 50%;
    transform: translate(-50%, -50%);
    border-collapse: collapse;
    width: 1300px;
    height: auto;
    border: 1px solid black;
    box-shadow: 2px 2px 12px rgba(0,0,0,0.2), -1px -1px 8px rgba(0,0,0,0.2);
    text-align: center;
}

tr {
    transition: all 0.2s ease-in;
}

th,td {
    padding: 20px;
    text-align: left;
    border-bottom: 1px solid black;
    border-color: black;
    border: 1px solid black;
}

#header {
    background-color: rgb(255, 113, 57);
    color: #fff;
}

h1 {
    font-weight: 600px;
}


```

```

        text-align: center;
        color: black;
        padding: 15px 0px;

    }

tr:hover {
    cursor: pointer;
    background-color: darkgrey;
    transform: scale(1.01);
    box-shadow: 2px 2px 12px rgba(0,0,0,0.2), -1px -1px 8px rgba(0,0,0,0.2);
}

/* When you resize and play around with the web page size, some of the table parts get
covered, to prevent this issue and make the whole table appear in any size of the web page */

@media only screen and (max-width: 768px) {
    table {
        width: 90%;
    }
}

</style>

<p id="display"></p>

```

### **Explanation of above code:**

The above code just creates a responsive web page to retrieve and display the results of each user vote.

### Javascript for email validation:

```

function next() {
    email = document.getElementById('mail').value;

    // Nested if statements //
    if (email == "jesusa.mercado@aphs.nsw.edu.au") {
        // For client security //

        var unique = prompt("Enter unique key"); // User input of given unique key to client to
confirm it is the client //
        var lower_unique = unique.toLowerCase();

        if (lower_unique == "37kxcq02") { // given only to client to access results //
            window.open("results.html"); // Direct the user (client) to results.html page //
        }

        else {
            location.href = "#"; // Reload user's page if he/she doesn't enter the correct pin
as per client security purposes //
        }
    }
}

```

```

        }

    }

    else {
        // For school security //
        if (email.includes("@aphs.nsw.edu.au")) {

            window.open("./voting.html"); // Direct the user to voting.html if his/her email
includes @aphs.nsw.edu.au //
        }

        else {
            alert("You must be an APHS student in order to vote for SRC members, this means that
your email must include \@aphs.nsw.edu.au\");
            location.href = "#"; // Reload user's page and don't let him/her go to the next page
if email doesn't include @aphs.nsw.edu.au //
        }
    }
}

```

#### **Explanation of above code:**

The above javascript code is a very important part of this system as it validates the email input field and secures an unauthorised access in order to vote. It states that if a random user enters a client's email (Ms. Mercado's email), then he/she will have to enter a unique key that is given to the client only.

#### Javascript for going to next member after each member preference:

```

function member1() {
    var one = document.getElementById('one');
    var two = document.getElementById('two');
    var three = document.getElementById('three');
    var four = document.getElementById('four');
    var five = document.getElementById('five');
    var none = document.getElementById("NA");

    if (one.checked == true || two.checked == true || three.checked == true || four.checked == true
|| five.checked == true || none.checked == true) {
        document.getElementById('container').style.display = "none";
        document.getElementById('container_1').style.display = "block";
        document.getElementById('member1').style.display = "none";
        document.getElementById('member2').style.display = "block";
        document.getElementById('icon2').style.display = "block";
        // document.getElementById('icon_2').style.display = "block";
    } else {
        alert("Please select any one preference to go to next member");
    }
}

```

```
}
```

### Explanation of above code:

This JS code checks and verifies if any preference is selected for each member. If the user has pressed the ">" button without selecting any preference, then an alert message will appear on his/her screen saying "Please select any one preference to go to the next member".

### Javascript for preference validation according to previous member preference selected:

```
if (document.getElementById('one').checked) {
    document.getElementById("one1").disabled = true;
    // document.getElementById('one1', 'one2', 'one3', 'one4', 'one5').innerHTML =
"<style>(background-color: grey)</style>";
    document.getElementById('one2').disabled = true;
    document.getElementById('one3').disabled = true;
    document.getElementById('one4').disabled = true;
    document.getElementById('one5').disabled = true;
}

if (document.getElementById('two').checked) {
    document.getElementById("two1").disabled = true;
    // document.getElementById('one1', 'one2', 'one3', 'one4', 'one5').innerHTML =
"<style>(background-color: grey)</style>";
    document.getElementById('two2').disabled = true;
    document.getElementById('two3').disabled = true;
    document.getElementById('two4').disabled = true;
    document.getElementById('two5').disabled = true;
}

if (document.getElementById('three').checked) {
    document.getElementById("three1").disabled = true;
    // document.getElementById('one1', 'one2', 'one3', 'one4', 'one5').innerHTML =
"<style>(background-color: grey)</style>";
    document.getElementById('three2').disabled = true;
    document.getElementById('three3').disabled = true;
    document.getElementById('three4').disabled = true;
    document.getElementById('three5').disabled = true;
}

if (document.getElementById('four').checked) {
    document.getElementById("four1").disabled = true;
    // document.getElementById('one1', 'one2', 'one3', 'one4', 'one5').innerHTML =
"<style>(background-color: grey)</style>";
    document.getElementById('four2').disabled = true;
    document.getElementById('four3').disabled = true;
```

```

document.getElementById('four4').disabled = true;
document.getElementById('four5').disabled = true;
}

if (document.getElementById('five').checked) {
    document.getElementById("five1").disabled = true;
    // document.getElementById('one1', 'one2', 'one3', 'one4', 'one5').innerHTML =
"<style>(background-color: grey)";
    document.getElementById('five2').disabled = true;
    document.getElementById('five3').disabled = true;
    document.getElementById('five4').disabled = true;
    document.getElementById('five5').disabled = true;
}

if (document.getElementById('NA').checked) {
    document.getElementById("NA2").disabled = true;
    // document.getElementById('one1', 'one2', 'one3', 'one4', 'one5').innerHTML =
"<style>(background-color: grey)";
    document.getElementById('NA3').disabled = true;
    document.getElementById('NA4').disabled = true;
    document.getElementById('NA5').disabled = true;
    document.getElementById('NA6').disabled = true;
}

```

### Explanation of above code:

This code validates the current preferences left for the user to select according to the preference selected for the previous member. For example, if the user selected 1st preference for the previous member, he/she cannot select 1st preference for any other member.

### Javascript for storing user preference of each SRC member:

```

const grade = document.getElementById('grade').value;

localStorage.setItem("email", email);
localStorage.setItem(email, '{}');
localStorage.setItem("grade", grade);
const emails = localStorage.getItem("emails");
if (emails) {
    const parsed = JSON.parse(emails)
    const found = false;
    parsed.forEach(element => {
        if (element.email === email) {
            // parsed[email] = { "email": email }
            element = { "email": email }
            found = true;
        }
    });
}

```

```

        if (!found) {
            parsed.push({ "email": email });
        }
        localStorage.setItem("emails", JSON.stringify(parsed));
    } else {
        localStorage.setItem("emails", JSON.stringify([{ "email": email }]));
    }

    location.reload();
}

```

#### **Explanation of above code:**

This code stores the user detail inputs like email, grade, radio buttons clicked (preferences selected for each member). It's stored in a Javascript storage method and "localStorage" and it is permanently stored so no data/result of any member can get removed unless the client does it manually.

#### Javascript for retrieving and displaying user preference for each SRC member:

```

const emails = localStorage.getItem("emails")

if (emails) {
    const parsed = JSON.parse(emails)

    parsed.forEach(e => {
        console.log(e.email)
        const d = document.getElementById("display")
        const div = document.createElement("div")
        div.innerHTML = `<span>${e.email} - ${e.member1} - ${e.member2} - ${e.member3} -
${e.member4} - ${e.member5}</span>`
        d.appendChild(div)
    })
}

```

#### **Explanation of above code:**

This code retrieves and displays all the user data and results stored in localStorage. The data and details that get retrieved include the user email, grade, each preference selected for each member.

#### Javascript code for creating a function to store user selected radio buttons:

```

function storeVote(email, pref, score) {
    const emailStore = JSON.parse(localStorage.getItem(email))
    emailStore[pref] = score;
}

```

```

localStorage.setItem(email, JSON.stringify(emailStore))

const parsed = JSON.parse(localStorage.getItem("emails"))
parsed.forEach(element => {
  if (element.email === email) {
    element[pref] = score
  }
});
localStorage.setItem("emails", JSON.stringify(parsed))

}

```

Explanation of above code:

The above code creates a function that stores all the votes for all the preferences selected by the user for all the members

Javascript code for storing user selected radio buttons:

```

const currentEmail = localStorage.getItem("email")
  storeVote(currentEmail, "member1",
document.querySelector('input[name="radio"]:checked').value)

```

Explanation of above code:

The above code stores the selected radio button for the specific SRC member that the user is voting for in the function created above the code.

### 3. *Testing and Evaluating Software Solutions*

#### 3.1 - Evaluation of software against needs analysis

Number	Met or Not Met	Requirements and needs of Software
1	Met	<p>The email validates which means that it only accepts emails including “@aphs.nsw.edu.au”. Whenever a user inputs his/her email, then that email should end with “@aphs.nsw.edu.au”, this helps secure the system as people from outside the school cannot access the voting system. This also allows collection of validated data to be sent to the client.</p> <p>Therefore this requirement was met efficiently.</p>
2	Not met (it was not a requirement anymore)	<p>The grade drop-down box validates well so that users will have to select that grade other than the default which is “-- Select your grade --”. This is to collect data and see which student from which grade has voted in the system. This is not working in the system and the user can only go further in the system after entering his/her email, which is not valid.</p> <p>Therefore this requirement was not met efficiently, as the code did not work for this part of the system.</p>
3	Met	<p>The reset button and proceed button function as expected on the first page according to the event handlers. The reset button is mainly used by the user to reset his/her details in the system. This also helps the UI (user interface) to look better and attractive which would please the user to vote further. The reset button resets the details on the main page that are the users email and their grade.</p> <p>Therefore this requirement was met efficiently.</p>
4	Met	<p>The client's email that is '<a href="mailto:jesusa.mercado@aphs.nsw.edu.au">jesusa.mercado@aphs.nsw.edu.au</a>' validates and then asks for a pin to confirm that the client's email is not accessed unauthorisedly. The pin is a string of characters which protects the system. <b>NOTE: The unique pin is given to the client by the developers.</b> This makes the system more secure and safe for the client.</p> <p>Therefore this requirement was met efficiently.</p>
5	Met	<p>The user data also gets stored (email, grade, preferences for each member) in both the <b>page1_js</b> and <b>vote.js</b> file with the help of localStorage method in Javascript.</p> <p>Therefore this requirement was met efficiently.</p>
6	Met	<p>SRC Members' Picture and their Speech Link get displayed in each member's container in their appropriate place. All the important details of each SRC member like their SRC school photo, SRC speech, description and names get displayed in their appropriately positioned places on the user screen.</p> <p>Therefore, this requirement was met.</p>
8	Met	<p>The position of the '&gt;' button is in their appropriate place where the arrow must be sharp on the left and vice versa. When the '&lt;' button is pressed by the user, the previous member that he/she voted for will appear and the user will be able to change his/her preference for that SRC member. Alternatively, when the '&gt;' button is pressed by the user, he/she will go to vote for the next SRC member.</p>

		Therefore, this requirement was met successfully
9	Met	If the user presses the “>” button before even selecting a preference (radio button) for the current SRC member he/she is voting for, an alert message will pop up on the user’s screen saying “Please select any one preference to go to the next member”. So in this case, the user will first need to select any of the preferences for the current SRC member they are voting for. Overall, this requirement was met successfully
10	Met	The instructions disappear as soon as the user presses “Let’s vote”. When the user finishes reading the instructions of ‘how to vote’ and understands it, he/she will press the “Let’s vote” button. After the user presses that button, the instructions disappear and voting for each SRC member will begin. The main aim for making this process was to make sure that the user doesn’t skip a candidate while voting, which would become unfair for the candidate as well as the user. Therefore, this requirement was met successfully.
11	Met	For each next member preference, if the previous SRC member preference is the same, then disable the radio button (preference) for that current SRC member that the user is voting for. For example, for the first SRC member, the user votes preference 1, then for any of the next SRC members, the user won’t be able to select the radio button of preference 1 as the user selected preference 1 prior to voting for the current SRC member. With the help of many ‘if’ and ‘else if’ statements, this was successfully implemented. Therefore, this requirement was met successfully.
12	Met	Extended width in every single web page This issue was later solved by simply adding a property and “overflow-x” in the CSS file and putting its value as “hidden” so this prevents the horizontal bar from appearing and scrolling horizontally is disabled. Therefore, this small issue of the project was solved successfully. It was not really a requirement, but good User Experience (UX) is crucial.
13	Met	The “thank you for voting” page. After the user finishes voting for the last SRC member, a button called “Submit votes” displays beside the last SRC member container. Once the user presses that button, he/she goes to the “thanks.html” web page where they can see cool background animation and text appearing in front of the web page saying “Thank you for voting. Your preferences are successfully recorded”. Hence, this requirement was met successfully.

Therefore, we met almost all requirements of the client, except that one requirement which is to run the code in any device, even the device that doesn’t have the code/system folder in it. This requirement was not met as the economic feasibility of the project was not given and so we couldn’t spend money on buying a server that could run the system on any machine/device without the code being downloaded in it.



Components 1: JavaScript Program and Code (/20)					
Grade	A Outstanding	B High	C Sound	D Basic	E Limited
Component					
Mark Boundary	17 - 20	13 - 16	9 - 12	5 - 8	1 - 4
JavaScript Program (/20)	<p>Performs <b>all</b> of the functions correctly to an <b>outstanding</b> level</p> <p><b>Outstanding</b> use of control structures</p> <p><b>Outstanding</b> evidence of input control and user feedback</p> <p><b>Outstanding</b> use of correct data types and structures</p> <p><b>Outstanding</b> internal documentation – variable names and comments</p> <p><b>Outstanding</b> inclusion of structure and arrays</p> <p>Both structures and arrays have appropriate names</p> <p><b>Outstanding</b> matching between variables and data types</p>	<p>Performs <b>most</b> of the functions correctly to a <b>high level</b></p> <p><b>High level</b> use of control structures</p> <p><b>High level</b> evidence of input control and user feedback</p> <p><b>High level</b> use of correct data types and structures</p> <p><b>High level</b> internal documentation – variable names and comments</p> <p><b>High</b> inclusion of structure and arrays</p> <p>Both structures and arrays have names</p> <p><b>High level of</b> matching between variables and data types.</p>	<p>Performs <b>some</b> of the functions correctly to a <b>sound</b> level</p> <p><b>Sound</b> use of control structures</p> <p><b>Sound</b> evidence of input control and user feedback</p> <p><b>Sound</b> use of data types and structures</p> <p><b>Sound</b> internal documentation – variable names and comments</p> <p><b>Sound</b> inclusion of structure or arrays</p> <p>Structures or arrays have names</p> <p><b>Sound</b> matching of variables to data types.</p>	<p>Performs a <b>few</b> of the functions to a <b>basic</b> level</p> <p><b>Basic</b> use of control structures</p> <p><b>Basic</b> evidence of input control and user feedback</p> <p><b>Basic</b> use of data types and structures</p> <p><b>Basic</b> internal documentation – variable names and comments</p> <p><b>Basic</b> inclusion of structure and arrays</p>	<p>Performs <b>one or two</b> of the functions to a <b>limited</b> level</p> <p><b>Limited</b> use of control structures</p> <p><b>Limited</b> evidence of input control and user feedback</p> <p><b>Limited</b> use of data types and structures</p> <p><b>Limited</b> or no internal documentation – variable names and comments</p>

## Component 2: Systems Report (/15)

Grade	A Outstanding	B High	C Sound	D Basic	E Limited
Component					

Mark Boundary	5	4	3	2	1
Planning and Designing Software Solutions (/5)	<p><b>Outstanding Needs Analysis</b></p> <p><b>Outstanding use of algorithms</b></p> <p><b>Outstanding analysis of variables and Data Types</b></p> <p><b>Outstanding Interface Design - as screen mockups or storyboards</b></p> <p><b>Outstanding Context Diagram</b></p> <p><b>Outstanding Data Flow Diagram</b></p> <p><b>Outstanding Gantt Chart or PERT Chart</b></p>	<p><b>High level Needs Analysis</b></p> <p><b>High level use of algorithms</b></p> <p><b>High level explanation of variables and Data Types</b></p> <p><b>High level Interface Design - as screen mockups or storyboards</b></p> <p><b>High level Context Diagram</b></p> <p><b>High level Data Flow Diagram</b></p> <p><b>High level Gantt Chart or PERT Chart</b></p>	<p><b>Sound level Needs Analysis</b></p> <p><b>Sound use of algorithms</b></p> <p><b>Sound description of variables and/or Data Types</b></p> <p><b>Sound Interface Design - as screen mockups or storyboards</b></p> <p><b>Sound Context Diagram</b></p> <p><b>Sound Data Flow Diagram</b></p> <p><b>Sound Gantt Chart or PERT Chart</b></p>	<p><b>Basic level Needs Analysis</b></p> <p><b>Basic use of algorithms</b></p> <p><b>Basic definition of variables or Data Types</b></p> <p><b>Basic Interface Design - as screen mockups or storyboards</b></p> <p><b>Basic Context Diagram</b></p> <p><b>Basic Data Flow Diagram</b></p> <p><b>Basic Gantt Chart or PERT Chart</b></p>	<p><b>Limited Needs Analysis</b></p> <p><b>Limited use of algorithms</b></p> <p><b>Limited statement of variables or Data Types</b></p> <p><b>Limited use of one or more of the following:</b></p> <ul style="list-style-type: none"> <li>• Structure Charts</li> <li>• Interface Designs</li> <li>• Context Diagram</li> <li>• Data Flow Diagram</li> <li>• Gantt Chart or PERT Chart</li> </ul>

Mark Boundary	5	4	3	2	1
Implementing Software Solutions (/5)	<p><b>Screenshots of your JavaScript program in action</b></p> <p><b>Explanations of each screenshot</b></p> <p><b>Examples of JS Code with comments and proper indentation</b></p> <p><b>Explanations of JS Code examples</b></p>	<p><b>Screenshots of your JavaScript program in action</b></p> <p><b>Description of each screenshot</b></p> <p><b>Examples of JS Code with comments and proper indentation</b></p> <p><b>Description of JS Code examples</b></p>	<p><b>Screenshots of your JavaScript program in action</b></p> <p><b>Examples of JS Code with comments and proper indentation</b></p>	<p>Examples of JS Code</p>	<p>Screenshots of your JavaScript program</p>

Mark Boundary	5	4	3	2	1
Testing and Evaluating Software Solutions (/5)	<p><b>Outstanding use of:</b></p> <ul style="list-style-type: none"> <li>• Test Data Tables</li> </ul> <p><b>Evaluation of the solution against Needs Analysis</b></p>	<p><b>High level of use of:</b></p> <ul style="list-style-type: none"> <li>• Test Data Tables</li> </ul> <p><b>Analysis of the solution against Needs Analysis</b></p>	<p><b>Sound use of:</b></p> <ul style="list-style-type: none"> <li>• Test Data Tables</li> </ul> <p><b>Explanation of the solution against Needs Analysis</b></p>	<p><b>Limited use of:</b></p> <ul style="list-style-type: none"> <li>• Test Data Tables</li> </ul> <p><b>Description of the solution against Needs Analysis</b></p>	<p><b>Basic use of:</b></p> <ul style="list-style-type: none"> <li>• Test Data Tables</li> </ul> <p><b>Definition of the solution against Needs Analysis</b></p>

