



ANNOUNCEMENT WEBSITE MANUAL

JUNE 17, 2020
PATEL, DEV

PREMAWARDENA, C | ICS3U0-C

How to use

This folder features two folders to neatly organized teacher and student code.

To begin, open the "homepage.html" file.

This will bring you the homepage where the user can go to either...

1.Student Options

2.Teacher Login

*Note that you can left click and open in new tab to view both pages at once

The teacher login has a username and password

Username is just any teachers name

***PASSWORD IS "northP123" (case sensitive)

To use the website, first create an announcement using the Teacher Page

Once that's done use the Student Options to enter the options after clicking submit it will take you to Student View to view the announcement

Teacher Login Page

Teacher Announcement Login

North Park Secondary School

Name

Password

Login

```
3
4 ▾ <head>
5     <title>Teacher Login</title>
6     <link rel="stylesheet" href="tLogin.css">
7     <script src="../main.js"> </script>
8 </head>
9
10 ▾ <body>
11
12
13
14 ▾     <header>
15
16         <h1>Teacher Announcement Login</h1>
17         <h3>North Park Secondary School</h3>
18
19     </header>
20
21 ▾     <div class ="LoginBox">
22
23
24
25
26
27
28         <label>Name</label>
29         <input type="text" id="userid" > <!--User enters a username and is set to userid-->
30         <br>
31         <br>
32         <label>Password</label>
33         <input type="password" id="passid"> <!--User enters a password and is set to passid-->
34         <br>
35         <br>
36         <input type="button" onclick= "check()" value="Login" preventDefault()/> <!--When clicked it will run javascript
37         pass check code-->
38
39         <br>
40
41     </div>
42
43 </body>
44 </html>
```

```

function check() { //function to check password

    var pass = document.getElementById("passid").value;
    var user = document.getElementById("userid").value;

    //the following code checkes if password are matching
    if (pass == "northP123") {

        localStorage.setItem("activeTeacher", user) //saves active username
        window.location.href = 'TeacherView.html'; //opens teacherview page if info matches

    }

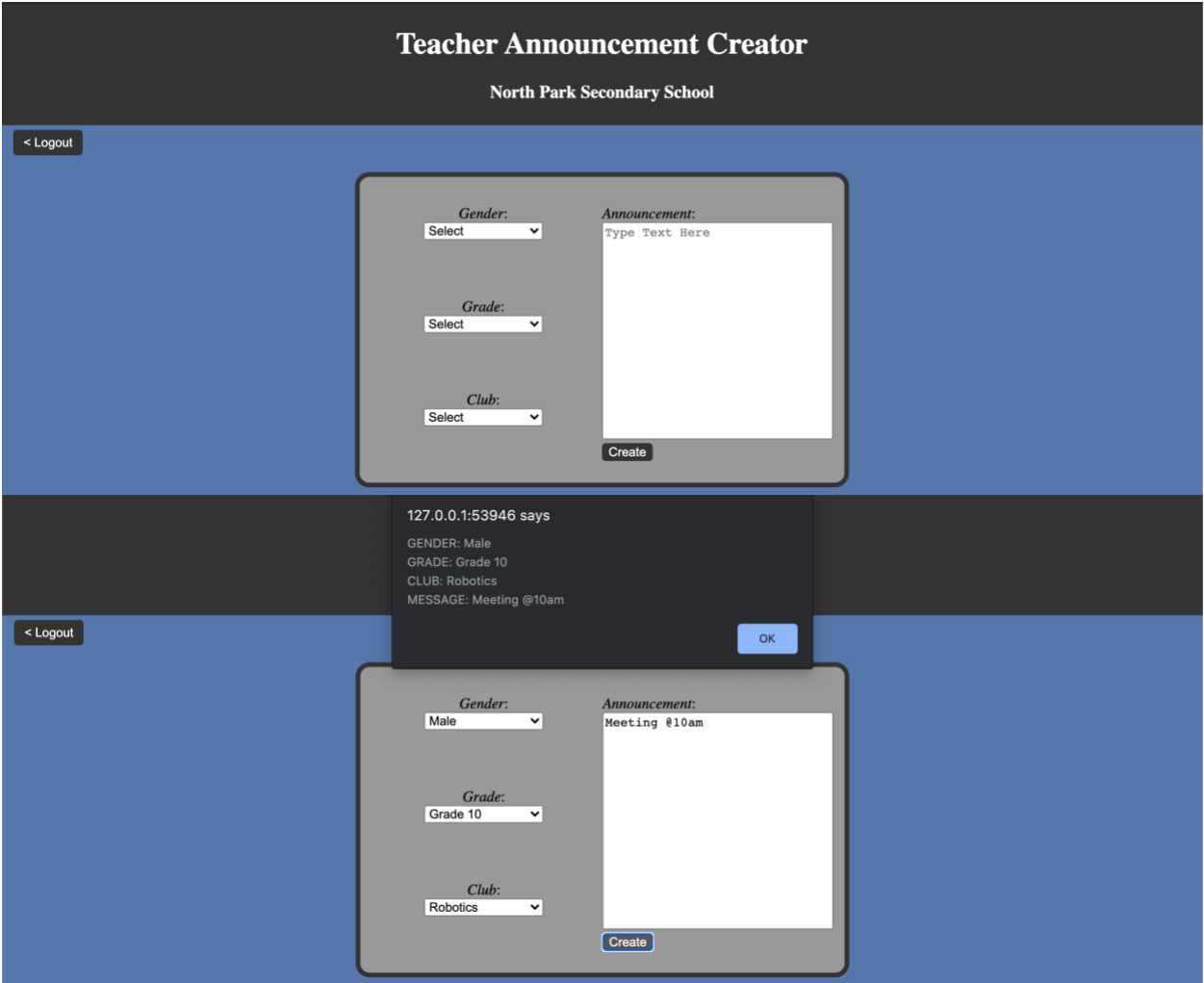
    else
    {
        alert("Incorrect Password");//displays error message
    }

}

```

The teacher login page has two inputs, one for username and one for password. The password and username is collected in the html file and sent to the java script file by using “.getElementById”. The password is than compared and if the password entered is equal to northP123 the announcement creator site is opened. It also saves the username in local storage which is used for later.

Teacher Announcement Creator Page



Key	Value
activeTeacher	MR. S
storageUsername	["MR. S"]
storageGradeT	["Grade 10"]
storageMsg	["Meeting @10am"]
storageGenderT	["Male"]
storageClubT	["Robotics"]
init	1
i	1
storageDate	["2020-6-17 15:51"]

Local Storage with one announcement

Key	Value
activeTeacher	Mr. Lamar
storageUsername	["MR. S", "Mr. Lamar"]
storageGradeT	["Grade 10", "All"]
storageMsg	["Meeting @10am", "No Practice this week"]
storageGenderT	["Male", "Female"]
storageClubT	["Robotics", "Badminton Team"]
init	1
i	2
storageDate	["2020-6-17 15:51", "2020-6-17 15:53"]

Site was refreshed and a new announcement was added

HTML

```
<!DOCTYPE html>
<html>

<head>
  <title>Teacher Announcement Creator </title>
  <link rel="stylesheet" href="tView.css">
  <script src="../../main.js"></script>
</head>

<body>

  <header>

    <h1>Teacher Announcement Creator</h1>
    <h3>North Park Secondary School</h3>

  </header>

  <!--Returns to Teacher Login Page -->
  <a href="TeacherLogin.html">|
    <button class="logout">&lt; Logout</button>
  </a>

  <br>
  <br>

27 ▼   <div class ="wrapper">
28
29 ▼     <div class="genderBox">
30       <label><em>Gender</em>;</label><br>
31 ▼       <select id="HTMLgender">
32         <option value="Select">Select</option>
33         <option value="Male">Male</option>
34         <option value="Female">Female</option>
35         <option value="All">All</option>
36       </select>
37     </div>
38
39 ▼     <div class="gradeBox">
40       <label><em>Grade</em>;</label><br>
41 ▼       <select id = "HTMLgrade">
42         <option value="Select">Select</option>
43         <option value="Grade 9">Grade 9</option>
44         <option value="Grade 10">Grade 10</option>
45         <option value="Grade 11">Grade 11</option>
46         <option value="Grade 12">Grade 12</option>
47         <option value="All">All</option>
48       </select>
49     </div>
50
51
52 ▼     <div class="clubBox">
53       <label><em>Club</em>;</label><br>
54 ▼       <select id = "HTMLclub" >
55         <option value="Select">Select</option>
56         <option value="DECA">DECA</option>
57         <option value="HOSA">HOSA</option>
58         <option value="Robotics">Robotics</option>
59         <option value="Badminton Team">Badminton Team</option>
60         <option value="Volleyball Team">Volleyball Team</option>
61         <option value="Soccer Team">Soccer Team</option>
62         <option value="Math Club">Math Club</option>
63         <option value="Debate Club">Debate Club</option>
64         <option value="ZONTA">ZONTA</option>
65         <option value="All">All</option>
66       </select>
67     </div>
68     <br>
```

<head> is the header of the page. This html file is linked to the main.js file and the tView.css file

This button sends the user back to the login page

These are all the drop-down menus for gender grade and club. Each option has a value which is then received by the JavaScript code

```

69
70
71 ▾      <div class= "messageBox">
72          <label><em>Announcement</em>:</label>
73
74          <!-- Inputbox -->
75          <textarea placeholder="Type Text Here" id="inputBox" name="text" cols="45" rows="15"></textarea>
76          <br>
77          <button class ="createButton" onclick="saveAnnouncement()">Create</button>
78
79      </div>
80
81  </div>
82
83
84
85
86
87 </body>
88 </html>

```

This is the code for the text box where teachers can write their announcements. The button then calls the “saveAnnouncement()” function

Java Script

```

function saveAnnouncement(){ //function to save TEACHER announcement data

    //Takes the value of each HTML element and sets them to a JS variable
    tGender[i] = document.getElementById("HTMLgender").value;
    tGrade[i] = document.getElementById("HTMLgrade").value;
    tClub[i] = document.getElementById("HTMLclub").value;
    tMsg[i] = document.getElementById("inputBox").value;

    //gets active teachers name
    tUsername = localStorage.getItem("activeTeacher");

    //Collects current date and time as of posting
    var today = new Date();
    var date = today.getFullYear()+'-'+(today.getMonth()+1)+'-'+today.getDate();
    var time = today.getHours() + ":" + today.getMinutes() ;
    var dateTime = date+' '+time;

    //adds date and teacher name to an array
    datePosted[i] = dateTime;
    postUsername[i] = tUsername;

    //Takes JS values and puts in in local storage
    localStorage.setItem("storageGenderT", JSON.stringify(tGender));
    localStorage.setItem("storageGradeT", JSON.stringify(tGrade));
    localStorage.setItem("storageClubT", JSON.stringify(tClub));
    localStorage.setItem("storageMsg", JSON.stringify(tMsg));
    localStorage.setItem("storageDate", JSON.stringify(datePosted));
    localStorage.setItem("storageUsername", JSON.stringify(postUsername));
}

```

First the data entered by the teacher is saved into an array by using “document.getElementById”. Arrays are like a list and have positions/index’s. The value of i determines where in the array the data will go, in this case it goes on the end. Next the active username is retrieved from local storage and set to a variable. Then the date and time as of posting is collected. The username and time are saved as an array with the same position as the announcement. Finally, all arrays are saved in local storage. To save an array into local storage it must be converted into a string by using “JSON.stringify”

```
alert("GENDER: "+tGender[i]+ "\nGRADE: "+ tGrade[i]+ "\nCLUB: "+tClub[i]+ "\nMESSAGE: "+tMsg[i])

i++
localStorage.setItem("i",JSON.stringify(i)); //changes local storage value for i

init = "1"; //chages init to 1 meaning that data has been entered in local storage
localStorage.setItem("init", init) //stores init value
```

An alert message then pops up showing all the details. The value of i is increased so the next time details are entered it will go into the next position. i is also saved in local storage to prevent it from being reset. Finally, the variable init is used to determine if the program has been run before, so once an announcement has been created the value of init is changed to 1 and it is saved in local storage.

Student Options Page

Student Announcement Options

North Park Secondary School

Gender:

Select

Grade:

Select

Club:

Select

Submit

Student Announcement Options

North Park Secondary School

Gender:

Male

Grade:

Grade 10

Club:

Robotics

Submit

storageGradeS	Grade 10
storageGenderS	Male
storageClubS	Robotics

HTML

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <title>Student Login</title>
6   <link rel="stylesheet" href="Login.css">
7   <script src="../main.js"></script>
8 </head>
9
10 <body>
11
12   <header>
13     <h1>Student Announcement Options</h1>
14     <h3>North Park Secondary School</h3>
15   </header>
```

<head> is the header of the page. This html file is linked to the main.js file and the Login.css file

```
18 <div class="options">
19   <label for="Gender"><em>Gender</em>:</label><br>
20   <select id="studentGender">
21     <option value="Select">Select</option>
22     <option value="Male">Male</option>
23     <option value="Female">Female</option>
24   </select>
25
26   <br>
27   <br>
28
29   <label for="Grade"><em>Grade</em>:</label><br>
30   <select id = "studentGrade">
31     <option value="Select">Select</option>
32     <option value="Grade 9">Grade 9</option>
33     <option value="Grade 10">Grade 10</option>
34     <option value="Grade 11">Grade 11</option>
35     <option value="Grade 12">Grade 12</option>
36   </select>
37
38   <br>
39   <br>
40
41   <label for="Club"><em>Club</em>:</label><br>
42   <select name="ClubSelect" id = "studentClub">
43     <option value="Select">Select</option>
44     <option value="DECA">DECA</option>
45     <option value="HOSA">HOSA</option>
46     <option value="Robotics">Robotics</option>
47     <option value="Badminton Team">Badminton Team</option>
48     <option value="Volleyball Team">Volleyball Team</option>
49     <option value="Soccer Team">Soccer Team</option>
50     <option value="Math Club">Math Club</option>
51     <option value="Debate Club">Debate Club</option>
52     <option value="ZONTA">ZONTA</option>
53   </select>
54
55   <br>
56   <br>
57   <!-- Button leads to Student Viewer and runs getData function -->
58   <a href="StudentView.html">
59     <button onClick="getData()">Submit</button>
60   </a>
```

Similar to the teacher page, the student page also has options for gender, grade and club

This button send the student to the announcement view page and runs the getData() function

Java Script

```
function getData(){  
  
    //Gets STUDENT filter options  
    sGender = document.getElementById("studentGender").value;  
    sGrade = document.getElementById("studentGrade").value;  
    sClub = document.getElementById("studentClub").value;  
  
    //Puts student options in local storage  
    localStorage.setItem("storageGenderS", sGender);  
    localStorage.setItem("storageGradeS", sGrade);  
    localStorage.setItem("storageClubS", sClub);  
  
}
```

First the options entered by the student are saved into a variable by using
“document.getElementById”

The variables are then saved in local storage under different keys.

Student Viewer Page

Student Announcement Viewer

North Park Secondary School

< End Session

Mr. Lamar

Please bring in your forms by this week!

2020-6-17 16:34

MR. S



Meeting @ 10am

2020-6-17 15:51

Tweets by @NorthParkSSca

North Park Secondary
@NorthParkSSca

Congratulations to our North Park MIP WINNERS for Junior Boys Volleyball - Ashvin and MIP Senior Boys Volleyball - Jashan. @PeelSchools @NPSS_Athletics

JUNIOR BOYS VOLLEYBALL
MIP WINNER - ASHVIN

SENIOR BOYS VOLLEYBALL
MIP WINNER - JASHAN

18h

Each message is displayed in its own box/div

Each box contains the teacher who posted it, the announcement and the date and time of posting

On the left side of the screen is a twitter feed

Student Announcement Viewer

North Park Secondary School

< End Session

Sorry No Announcements

If no announcements are found this message pops up

HTML

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <title>Student View</title>
6   <link rel="stylesheet" href="View.css">
7   <script src="../../main.js"></script>
8 </head>
9
10
11
12 <body onload="display()">
13
14   <header>
15     <h1>Student Announcement Viewer</h1>
16     <h3>North Park Secondary School</h3>
17   </header>
18
19   <!--Returns to student login -->
20   <a href="StudentLogin.html">
21     <button class="logout">&lt; End Session</button>
22   </a>
23
24
25   <div class="wrapper">
26
27     <div id="msgBox">
28
29
30
31     </div>
32
33     <!--Twitter Feed -->
34     <div class="twitter">
35
36       <a class="twitter-timeline" data-height="1500" href="https://twitter.com/NorthParkSSca?ref_src=twsrc%SEtfw">Tweets
37       by NorthParkSSca</a>
38       <script async src="https://platform.twitter.com/widgets.js" charset="utf-8"></script>
39
40     </div>
```

When the page loads the “display()” function will run automatically.

Button returns to the student option page.

Twitter feed was found from twitter’s website on how to include a timeline on to a html site.

Java Script

```
function display(){  
  
    noAnnouncement = true; //resets no announcements to true each time function is called  
  
    //gets local storage data for student  
    sGender = localStorage.getItem("storageGenderS");  
    sGrade = localStorage.getItem("storageGradeS");  
    sClub = localStorage.getItem("storageClubS");  
  
    //gets local storage data for teacher  
    tGender = JSON.parse(localStorage.getItem("storageGenderT"));  
    tGrade = JSON.parse(localStorage.getItem("storageGradeT"));  
    tClub = JSON.parse(localStorage.getItem("storageClubT"));  
    tMsg = JSON.parse(localStorage.getItem("storageMsg"));  
  
    //gets date and username of teacher  
    datePosted = JSON.parse(localStorage.getItem("storageDate"));  
    postUsername = JSON.parse(localStorage.getItem("storageUsername"));
```

The var noAnnouncement is used to determine if there are any announcements to show or not (will be used later)

First the code gets all the local storage data from the students and then teacher. The teacher data needs to be parsed since it is a string and needs to be converted back into an array. The date and time are also retrieved from local storage.

```
//loop runs for each item in an array  
for(i = 0; i < tGender.length; i++){  
  
    //compares if student selection equals to teacher selection  
    if((tGender[i] == sGender || tGender[i] == "All") && (tGrade[i] == sGrade || tGrade[i] == "All") && (tClub[i] == sClub || tClub[i] == "All")){
```

The for loop has a counter variable called i, the loop will run as long as i is less than the length of the tGender array, each time the loop completes 1 value is added to i.

The if statement compares the student options to all the teacher entries, if a match is found the code inside the if statement will run, if not the loop will run again and will compare the next item in the array.

```

//creates new elements each time a match is found
var div = document.createElement("div") //this parent div element houses the msg, teacher name and time
var name = document.createElement("B")
var msg = document.createElement("P")
var time = document.createElement("P")

//this div already exists in the student html and each new div will be added in here
var items = document.getElementById("msgBox")

//creates a class name for msg and time...used for css styling
div.className = "msg";
time.className = "time"

//puts the current username inside of the div element
name.innerHTML = postUsername[i];
div.appendChild(name)

//puts the current mesaage inside of the div element
msg.innerHTML = tMsg[i];
div.appendChild(msg)

//puts the current time inside of the div element
time.innerHTML = datePosted[i];
div.appendChild(time);

//appends div to parent div "msgBox"
items.insertBefore(div, items.childNodes[0]);

noAnnouncement = false; //since there is an announcement that means this is false

```

If a match is found a new message box is created and the announcement is shown. This handles multiple announcements.

When a match is found, 4 new elements are created using document.createElement()

1. Div - this is a divider tag that is used to house all the other elements, as well I assigned it a class so I can add css to it to give it the box look
- 2.Name - this is a bold text tag and by using innerHTML I can put the current teachers name in
- 3.Msg - this is a paragraph tag and by using innerHTML I can put the current msg name in it
- 4.Time - this is also paragraph tag and by using innerHTML I can put the current date and time in it

```

//creates new elements each time a match is found
var div = document.createElement("div") //this parent div element houses the msg, teacher name and time
var name = document.createElement("B")
var msg = document.createElement("P")
var time = document.createElement("P")

```

The words in the end bracket declare what type of tag it is. So for example <div> is a div tag, bold text, <P> paragraph.

```

//puts the current username inside of the div element
name.innerHTML = postUsername[i];
div.appendChild(name)

//puts the current mesaage inside of the div element
msg.innerHTML = tMsg[i];
div.appendChild(msg)

//puts the current time inside of the div element
time.innerHTML = datePosted[i];
div.appendChild(time);

```

"Div.appendChild()" basically adds element 2,3 and 4 inside the div element

```

//appends div to parent div "msgBox"
items.insertBefore(div, items.childNodes[0]);

```

"items.insertBefore(div, items.childNodes[0])" adds the new div element inside "item" which is basically the "msgBox" div ← This was created in student viewer

insertBefore(__,childNodes[0]) allows the div to be inserted before the 1'st child node in msgBox meaning announcements will appear in order of posting.

```

msgBox(PARENT NODE)
  div(CHILD NODE)
    name(CHILD NODE)
    msg(CHILD NODE)
    time(CHILD NODE)

```

By using this method, I don't have to create one big string using a text holder variable. Instead I can give each announcement its own box


```

        noAnnouncement = false; //since there is an announcement that means this is false
    }
}

if (noAnnouncement == true){ //if there is no announcemet this code will run

    //creates the message box
    var div = document.createElement("div")
    var msg = document.createElement("p")
    var items = document.getElementById("msgBox")

    div.className = "msg";

    msg.innerHTML = "Sorry No Announcements";
    div.appendChild(msg);

    items.insertBefore(div, items.childNodes[0]);

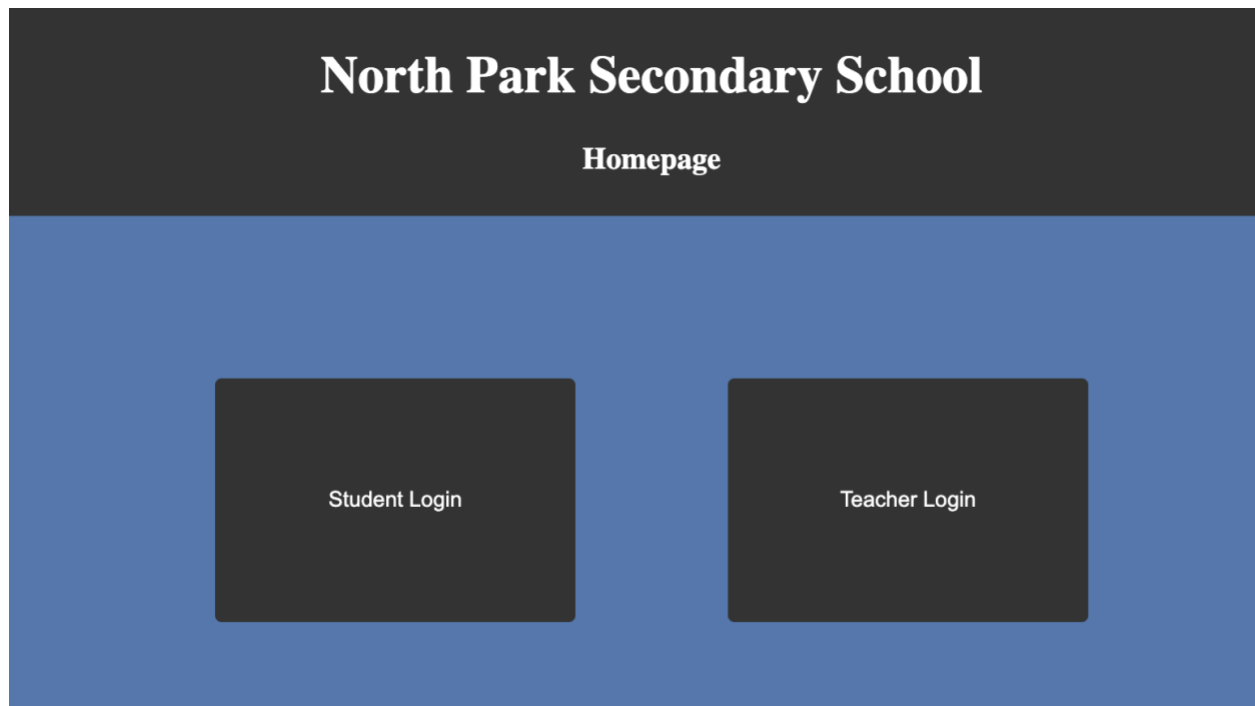
}

```

Finally, at the very end of the if statement noAnnouncements is set to false since there is now an announcement is created

Outside of the for loop is another if statement. If there is no announcements to be displayed (meaning the code inside previous if statement was never run) then a new message box is created and inside the text box a message shows “Sorry No Announcements”

Home Page



```
<!DOCTYPE html>
<html>

  <head>
    <title>North Park Home Page</title>
    <link rel="stylesheet" href="homepage.css">
  </head>

  <body>

    <header>
      <h1>North Park Secondary School</h1>
      <h3>Homepage</h3>
    </header>
    <div class="wrapper">
      <div class="btn">
        <a href="Student/StudentLogin.html">
          <button>Student Login</button>
        </a>
      </div>

      <div class="btn">
        <a href="Teacher/TeacherLogin.html">
          <button>Teacher Login</button>
        </a>
      </div>
    </div>

  </body>

</html>
```

To add to the user friendliness of the program a home page was created. This is the site that should be handed out to both teachers and students. Teachers can enter through to teacher login page and students can use the student login page.

Retaining Data After Refreshing

```
3 //all variables that aren't saved over session...will refresh each time page reloads
4 var sGender;
5 var sGrade;
6 var sClub;
7 var tUsername;
8 var init;
9 var noAnnouncement; //this is used if there are no announcements to show
10
11
12
13 init = localStorage.getItem("init"); //retrives value for init from local storage
14 //if the code hasn't been intialized init will equal null
15
16
17 if(init === "1"){ //if init has a value it will retrieve all previously local stored data
18     tGender = JSON.parse(localStorage.getItem("storageGenderT"));
19     tGrade = JSON.parse(localStorage.getItem("storageGradeT"));
20     tClub = JSON.parse(localStorage.getItem("storageClubT"));
21     tMsg = JSON.parse(localStorage.getItem("storageMsg"));
22
23     datePosted = JSON.parse(localStorage.getItem("storageDate"));
24     postUsername = JSON.parse(localStorage.getItem("storageUsername"));
25
26     i = parseInt(localStorage.getItem("i")); //retrives value of i from local storage
27 }
28 else{ //will ONLY run when the website is run for the first time
29     var tGender = []; //these variables should only be declared once to avoid resetting them
30     var tGrade = [];
31     var tClub = [];
32     var tMsg = [];
33     var datePosted = [];
34     var postUsername = [];
35     var i = 0;
36 }
37
38 }
```

First what was the issue?

BEFORE

Every time a page is refreshed main.js file is being called too.

The code is then read from the top where I declare all my variables.

This means all the arrays are set to "[]" and the value of "i" is set back to 0

This is not good since its going to reset all the local storage values

AFTER

What I did is create a variable called "init" (short for initialize) which checks if the code has been run/initialized before, meaning that the variables have already been declared

If the code has been initialized before each variable will be set back to what has been said in local storage

If the code hasn't been initialized then another code will run to initialize the program, what this does is basically declare all the arrays and the var i

*note this method was created before the screen cast session so I devised my own method that worked every time the page was refreshed.

Test Cases

Test case	Description of Input	Expected Result	Actual Result	Pass/Fail Conclusion
Test case 1	Male, Grade 10, HOSA	HOSA meeting for grade 10 boys next week Monday @lunch	HOSA meeting for grade 10 boys next week Monday @lunch	Pass
Test case 2	Male, Grade 10, HOSA	Meeting next week is cancelled	Meeting next week is cancelled HOSA meeting for grade 10 boys next week Monday @lunch	Pass Multiple announcements shown in correct order
Test case 3	Gender ALL, Grade 9, DECA	Congratulations to ALL grade 9 DECA students	Congratulations to ALL grade 9 DECA students	Pass Same announcement shown for male and female in grade 9 DECA
Test case 4	Male, Grade ALL, DECA	ALL male DECA students will stay in the 12 th floor in the hotel	ALL male DECA students will stay in the 12 th floor in the hotel	Pass Same announcement shown males in deca for all grades
Test case 5	ALL, ALL, ALL	NO SCHOOL TOMORROW	NO SCHOOL TOMORROW	Pass Announcement shows for any student option
Test case 6	TEACHER: Grade ALL, Gender ALL, Robotics STUDENT: Grade 9, Male, Zonta	Teacher Input: Please have all robotics work finished by Monday Student output: No Announcements	No Announcements	Pass Even though grade and gender match, club doesn't match so no announcements show
Test case 7	TEACHER: Grade 9, Male, Soccer STUDENT: Grade 11, Female, Badminton	Teacher Input: Soccer game next week Tuesday Student output: No Announcements	No Announcements	Pass None of the categories match so no announcements

Test case 1

Gender:

Male ▼

Grade:

Grade 10 ▼

Club:

HOSA ▼

Announcement:

HOSA meeting for grade 10 boys
next week Monday @lunch

Create

< End Session

Mr. Lamar

HOSA meeting for grade 10 boys next week Monday @lunch

2020-6-17 17:27

Test case 2

Gender:

Male ▼

Grade:

Grade 10 ▼

Club:

HOSA ▼

Announcement:

Meeting next week is cancelled

Create

< End Session	
Mr. Lamar Meeting next week is cancelled	2020-6-17 17:29
Mr. Lamar HOSA meeting for grade 10 boys next week Monday @lunch	2020-6-17 17:27

Test case 3

Gender:

All ▼

Grade:

Grade 9 ▼

Club:

DECA ▼

Create

Announcement:

Congratulations to ALL grade 9 DECA students

Gender:

Male ▼

Grade:

Grade 9 ▼

Club:

DECA ▼

Submit

Gender:

Female ▼

Grade:

Grade 9 ▼

Club:

DECA ▼

Submit

Mr. Lamar

Congratulations to ALL grade 9 DECA students

2020-6-17 17:31

Test case 7

Gender:

Male ▼

Grade:

Grade 9 ▼

Club:

Soccer Team ▼

Create

Announcement:

Soccer game next week Tuesday

Gender:

Female ▼

Grade:

Grade 11 ▼

Club:

Badminton Tea ▼

Submit

< End Session

Sorry No Announcements

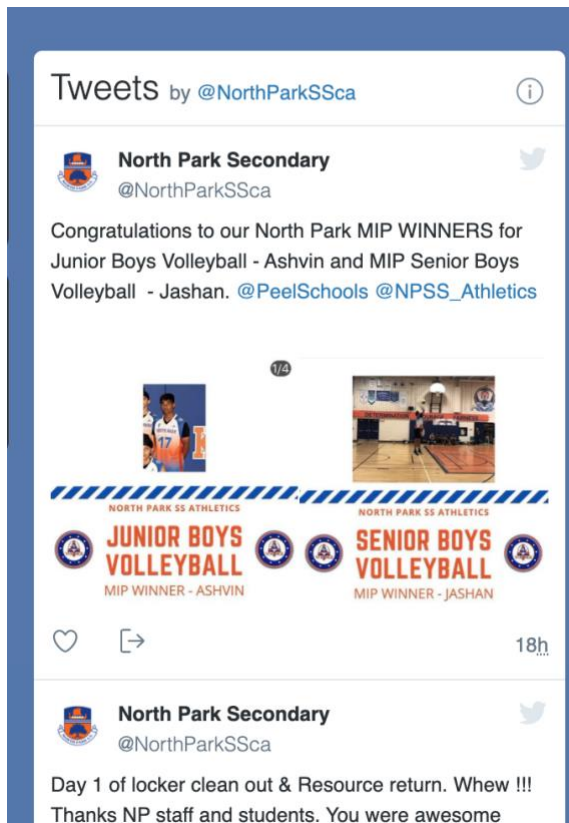
Additional Features



All buttons on the site react to a mouse when they are hovered over.

Mr. Lamar Please bring in your forms by this week!	2020-6-17 16:34
MR. S Meeting @10am	2020-6-17 15:51

Each post has the Teachers name, so the student knows who posted it. Also, the date and time are posted so students know if a post was made recently. Finally, the most recent posts are shown on top.



Twitter Feed shown on left side of screen to keep students up to date.