

yr2-sem1-webdevelopment- assignment1

by Jericho Jermaine Uzzell

FILE	12014_JERICO_JERMAINE_UZZELL_YR2-SEM1-WEBDEVELOPMENT-ASSIGNMENT1_302981_1400154666.DOCX (953.37K)		
TIME SUBMITTED	10-JAN-2021 07:52PM (UTC+0000)	WORD COUNT	4216
SUBMISSION ID	141444752	CHARACTER COUNT	33564

University of Wales Trinity Saint David

School of Computing

Web and Mobile Technologies

ACCA5031

Jericho Uzzell

1702201

09/01/2021



Prifysgol Cymru

Y Drindod Dewi Sant

University of Wales

Trinity Saint David

1. Introduction; [page 4]
2. Index page; [page 4-7]
 - 2.1. Read more component; [page 4]
 - 2.1.1. Screen shots;
 - 2.1.2. Methodology;
 - 2.1.3. Annotated code;
 - 2.2. Graph generator; [page 5-7]
 - 2.2.1. Screen shots; [page 5]
 - 2.2.2. Methodology; [page 5]
 - 2.2.3. Annotated code; [page 6-7]
3. University page; [page 8-20]
 - 3.1. Reload button; [page 8]
 - 3.1.1. Screen shots;
 - 3.1.2. Methodology;
 - 3.1.3. Annotated code;
 - 3.2. Loading data; [page 9-10]
 - 3.2.1. Screen shots; [page 9]
 - 3.2.2. Methodology; [page 9]
 - 3.2.3. Annotated code; [page 9-10]
 - 3.3. Adding data; [page 11-13]
 - 3.3.1. Screen shots; [page 11]
 - 3.3.2. Methodology; [page 12]
 - 3.3.3. Annotated code; [page 12-13]
 - 3.4. Editing data; [page 14-17]
 - 3.4.1. Screen shots; [page 14]
 - 3.4.2. Methodology; [page 14]
 - 3.4.3. Annotated code; [page 15-17]
 - 3.5. Deleting data; [page 18-20]
 - 3.5.1. Screen shots; [page 18]
 - 3.5.2. Methodology; [page 18]
 - 3.5.3. Annotated code; [page 18-20]

4. Spam page; [page 21-27]
 - 4.1. Reload button; [page 21]
 - 4.1.1. Screen shots;
 - 4.1.2. Methodology;
 - 4.1.3. Annotated code;
 - 4.2. Getting data; [page 21-23]
 - 4.2.1. Screen shots; [page 21]
 - 4.2.2. Methodology; [page 21]
 - 4.2.3. Annotated code; [page 22-23]
 - 4.3. Adding data; [page 24-27]
 - 4.3.1. Screen shots; [page 24-25]
 - 4.3.2. Methodology; [page 25]
 - 4.3.3. Annotated code; [page 25-27]
5. Time page; [page 28-34]
 - 5.1. Onload; [page 28-30]
 - 5.1.1. Screen shots; [page 28]
 - 5.1.2. Methodology; [page 28]
 - 5.1.3. Annotated code; [page 28-30]
 - 5.2. Selecting a country; [page 31-32]
 - 5.2.1. Screen shots; [page 31]
 - 5.2.2. Methodology; [page 31]
 - 5.2.3. Annotated code; [page 31-32]
 - 5.3. Updating time; [page 33-34]
 - 5.3.1. Screen shots; [page 33]
 - 5.3.2. Methodology; [page 33]
 - 5.3.3. Annotated code; [page 33-34]
6. Conclusion; [page 35]
7. Bibliography; [page 35]

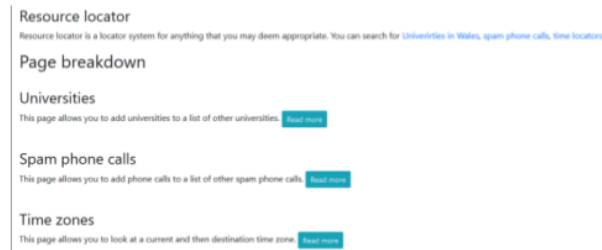
1. Introduction

This document will go over the site that this student developed. They will take a screenshot of the page area which utilises javascript (JS), JQuery and/or JQuery-UI elements, their justification for their way of coding it and why they chose to develop it.

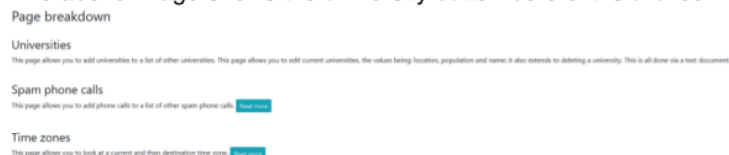
2. Index page

2.1 Read more component

2.1.1. Screen Shots



The above image shows the university button before it is clicked.



The above image shows the university button after it is clicked. Text is shown and the button removed.

2.1.2. Methodology

The student chose to use a fade in when you click a button. This was chosen as the text was not needed to be shown as it would clutter the website. Having a button that allows the users to click to read more about that section is good. The belief is, if the text is not relevant then there is no need to show it.

The student added a fade when the button was clicked, used from the JQuery library, to offer some effects. They used JQuery as the code in pure JS would be a lot.

2.1.3. Annotated Code Listing (paste in code)

```
onclick="$(this).fadeOut();document.getElementById('read_more_unis').classList.remove('hide');"  
onclick="$(this).fadeOut();document.getElementById('read_more_phone_calls').classList.remove('hide');"  
onclick="$(this).fadeOut();document.getElementById('read_more_timezones').classList.remove('hide');"
```

2.2 Graph generator

2.2.1. Screen Shots

Graph results

Want to visualise your data in a graph? Use this cool feature -- the inputs must be numbers.

[Add more fields](#) [Generate](#)

[Remove](#)

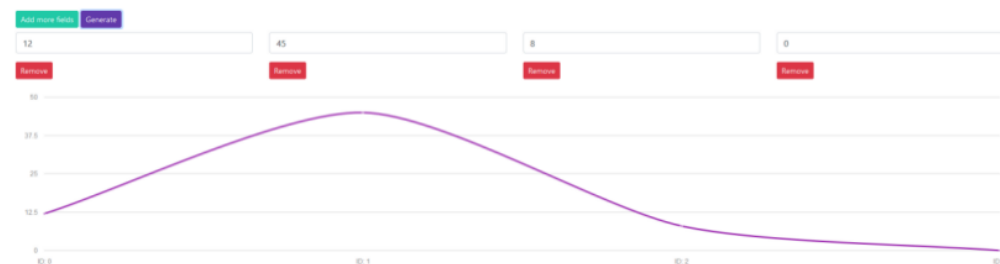
The above image shows the default state when you load the page.

Graph results

Want to visualise your data in a graph? Use this cool feature -- the inputs must be numbers.



The above image shows multiple values being displayed to the user.



The above image shows some of the values being deleted and then regenerated.

2.2.2. Methodology

The student offered to allow the user to generate a graph on the homepage as they thought it would be a nice feature to show people changes in data. I could be seen as out of character, as the other parts the site uses is more focused on adding data however, with data can allow you to track timestamps if desired.

The code used was a mix of JS and JQuery, the graph itself used JS and is called using built in functions. The student used JQuery to append more inputs as they believed it was not a big part of the code, instead the biggest part being the generation.

The student did a for loop with the max length being the input classes, at least two numbers were required.

2.2.3. Annotated Code Listing (paste in code)

```
$(document).on('click', '#graph-add-more', function() {
    $('#graph-add-div').append('<div class="col-xl-3 col-lg-3 col-md-3 col-sm-3 col-3"><input type="text" lettercountmax nospecials nonumbers class="number_only graph-data-input form-control input-md" /><p></p><input type="button" class="graph-data-delete btn btn-sm btn-danger mb-2" value="Remove" value="" /></div>');
});
$(document).on('click', '#graph-add-generate', function() {
    let datalength=document.getElementsByClassName('graph-data-input').length;
    if(datalength<2) {
        swal({title:'Error in graph',text:'You must have at least two data sources',type:'error'});
        return false;
    }
    let dataraw=document.getElementsByClassName('graph-data-input');
    let data=new Array();
    let localvalue='';
    for(let k=0;k<datalength;k++) {
        localvalue=dataraw[k].value;
        if(localvalue==undefined || !number_con(dataraw[k],false)) {
            dataraw[k].parentNode.querySelector('p').innerText="Must be a number!";
            continue;
        }
        dataraw[k].parentNode.querySelector('p').innerText="";
        data.push({data:'ID: '+k,value:localvalue});
        // console.log(dataraw[k].value);
    }
    if(data.length==datalength) {
        //we can continue;
        document.getElementById('graph-generated').innerHTML='';
        new Morris.Line({
            element: 'graph-generated',
            data: data,
            parseTime:false,
            pointSize: 2,
            hideHover: 'auto',
            behaveLikeLine: true,
            xkey: 'data',
            lineColors: ['rgb(156, 39, 176)'],
            ykeys: ['value'],
            labels: ['Value']
        });
    }
});
$(document).on('click', '.graph-data-delete', function() {
    $(this).parent().remove();
});
$(document).ready(function() {
    document.getElementsByClassName('graph-data-input')[0].value='';
```

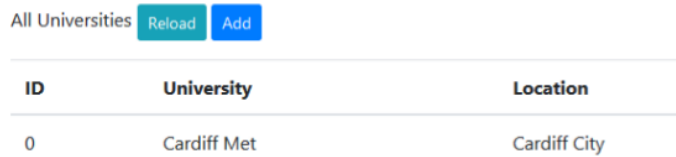
Jericho Uzzell (1702201) – ACCA5031 Web and Mobile Technologies – Sem1 – Assignment
1 – 09/01/2021 - UWTSD

});

3. University page

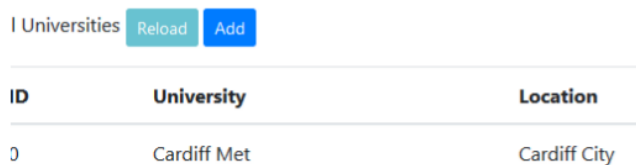
3.1 Reload button

3.1.1. Screen Shots



All Universities Reload Add		
ID	University	Location
0	Cardiff Met	Cardiff City

The above image shows the button before it is clicked.



1 Universities Reload Add		
ID	University	Location
0	Cardiff Met	Cardiff City

The above image shows the button once clicked – it is disabled until the data is loaded.

3.1.2. Methodology

The student decided to have a reload button available to be clicked. They believed that the user would like to see changes without refreshing their page. They also believed that if the ability to refresh the data was given to the user, that the user should not be allowed to continue to click this button whilst the data is being loaded – only once the data is loaded, then they can use the button again. The student used pure JS in this code as it was simple tasks to achieve.

3.1.3. Annotated Code Listing (paste in code)

```
onclick="jsGetUniData();"
. . .
let reloadbutton=document.getElementById('university_reload');
reloadbutton.setAttribute('disabled', true);
reloadbutton.blur();
. . .

setTimeout(function() {
    reloadbutton.removeAttribute('disabled');
},1000);
```

3.2 Loading data

3.2.1. Screen Shots

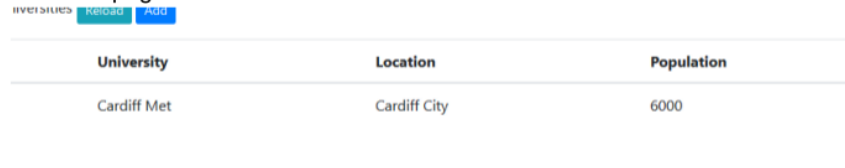


The screenshot shows a web application interface. At the top, there are two buttons: 'Reload' (green) and 'Add' (blue). Below them is a table with the following data:

ID	University	Location	Population
0	Cardiff Met	Cardiff City	5000
1	UWTSD	Swansea	100
2	Oxford Uni	England	6000

To the right of each row, there are two buttons: 'Edit' (teal) and 'Delete' (red).

The above image shows the data being loaded initially. The data is automatically loaded once the page is loaded or the reload button is clicked.



The screenshot shows a web application interface. At the top, there are two buttons: 'Reload' (green) and 'Add' (blue). Below them is a table with the following data:

University	Location	Population
Cardiff Met	Cardiff City	6000

After changing some data, as we can see the population has increased by one thousand from the previous screen shot.

3.2.2. Methodology

The student in this code has tried to write in pure JS when it was not needed. It was somewhat unavoidable when the student was getting the file as the alternative was something called XMLHttpRequest which can be too much and complex. Where as the code seen to get data is one line.

They chose this way of getting data as they believed it should be in a function, the reload button will be disabled during the gathering of the data, it will go through the data and then add it to the document.

3.1.3. Annotated Code Listing (paste in code)

```
function jsGetUniData() {  
    //this function gets the file contents of /resource/university.txt;  
    //returns as text but is turned into json for our needs;  
    let reloadbutton=document.getElementById('university_reload');  
    reloadbutton.setAttribute('disabled', true);  
    reloadbutton.blur();  
  
    // $('#university_reload').attr('disabled',true);  
    random_number=randomNumberGen(); // need to generate a new number as  
    js caches the txt document;  
    $.get('/resource/universitynew.txt?q='+random_number, {},function(data  
    ){  
        console.log(data);  
  
        if(isResultOkay(data, false) [1]==0&&isResultOkay(data, false) [0]==false  
        ) {  
            isResultOkay(data, false);  
            return;  
        }  
        data=JSON.parse(data);  
    })  
}
```

Jericho Uzzell (1702201) – ACCA5031 Web and Mobile Technologies – Sem1 – Assignment
1 – 09/01/2021 - UWTSD

```
// console.log(data.length);
let uniHTML=document.getElementById('university_output');

if(data==undefined||data.length==0) {
    uniHTML.innerHTML = '<tr><td colspan="6" class="text-center text-danger d-3">No data to load!</td></tr>';
    return;
}
let out='';
uniHTML.innerHTML='';
$.each(data,function(k,i) {
    // console.log(i);
    out='';
    out=out+'<tr>';
    out=out+'<td>'+i.id+'</td>';
    out=out+'<td>'+sanitise(i.name)+'</td>';
    out=out+'<td>'+sanitise(i.location)+'</td>';
    out=out+'<td>'+i.population+'</td>';
    out=out+'<td>';
    out=out+'<button class="btn btn-sm btn-block btn-info edit_uni" key="'+i.id+'-'+i.rand+'">Edit</button>';
    out=out+'<button class="btn btn-sm btn-block btn-danger delete_uni" key="'+i.id+'-'+i.rand+'">Delete</button>';
    out=out+'</td>';
    out=out+'</tr>';
    uniHTML.innerHTML += out;
});
});
setTimeout(function(){
    reloadbutton.removeAttribute('disabled');
},1000);

}
function randomNumberGen() {
    return ( Math.random() ).toString().split('.')[1].substring(0,2); //
    need to generate a new number as js caches the txt document -- it will get
    the document with a ?q=random_number so its fresh (we generate each time we
    get it);
}
```

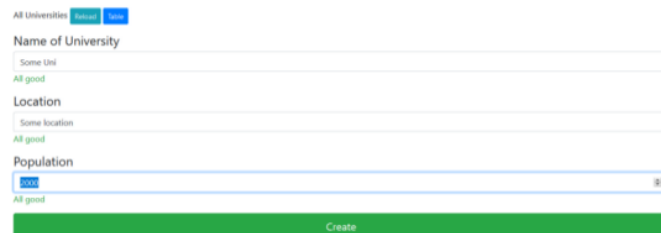
3.3 Adding data

3.3.1. Screen Shots



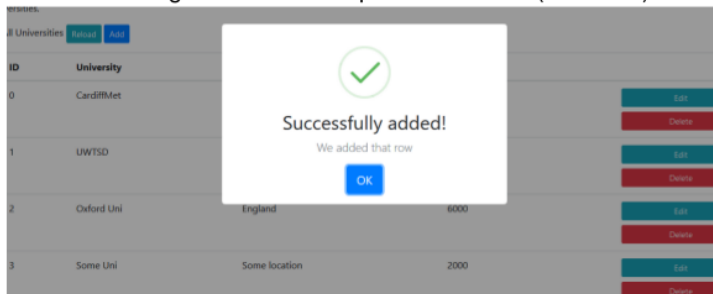
A screenshot of a web form titled 'All Universities' with 'Add' and 'View' buttons. The form has three input fields: 'Name of University' (with a 'Name' placeholder), 'Location' (with a 'Location' placeholder), and 'Population' (with a 'Population' placeholder and a numeric input icon). A green 'Create' button is at the bottom.

The above image shows the input fields being empty.



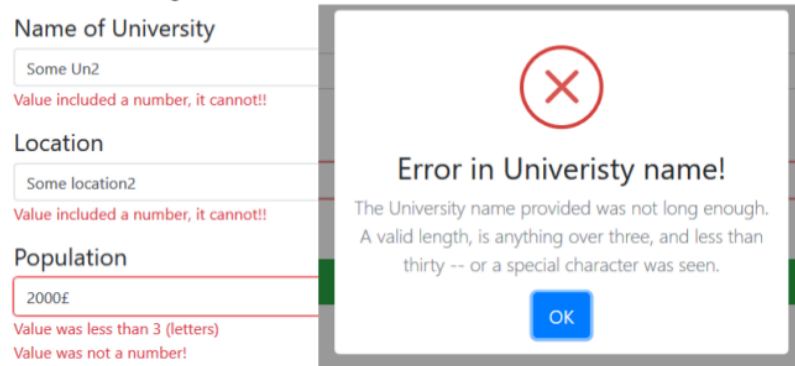
A screenshot of the same web form, now with valid input data. The 'Name of University' field contains 'Some Uni', 'Location' contains 'Some location', and 'Population' contains '1000'. Each field has a green 'All good' validation message below it. The green 'Create' button is still at the bottom.

The above image shows if the input data is valid (no errors).



A screenshot showing a confirmation dialog box with a green checkmark icon. The text inside says 'Successfully added!' and 'We added that row'. There is an 'OK' button. In the background, a table of universities is visible, including rows for CardiffMet, UWTSD, Oxford Uni, and Some Uni.

The above image shows what it is like when the data is submitted.



A screenshot of the web form with validation errors. The 'Name of University' field contains 'Some Un2' with the error message 'Value included a number, it cannot!!'. The 'Location' field contains 'Some location2' with the error message 'Value included a number, it cannot!!'. The 'Population' field contains '2000£' with two error messages: 'Value was less than 3 (letters)' and 'Value was not a number!'. An error popup dialog box is overlaid on the form, featuring a red 'X' icon and the text 'Error in Univeristy name! The University name provided was not long enough. A valid length, is anything over three, and less than thirty -- or a special character was seen.' with an 'OK' button.

The above images show errors if the user does something wrong. If the user clicked create, it will return a popup error.

3.3.2. Methodology

As previously stated regarding fading in and out using JQuery, the student used this as the code to do it would be far less superior in their mind from JQuery. Once you click a button to add data, it will fade the document between two pages, one for adding and one for viewing data.

Adding data is just like gathering it, it uses mostly JS features with some JQuery elements where it was unavoidable. The student programmed it in, where the text fields would be validated one the go whilst the user was typing and any error received was stated to the user. The student built in for input fields to be numbers on, letters only, no special characters, spaces allowed etc. The code will auto generate a random number when requesting a file due to cache problems -- causing old versions being sent (this is done automatically).

3.3.3. Annotated Code Listing (paste in code)

```
function addUni() {
    if( document.getElementById('university_add').textContent == 'Add' )
    {
        $('#university_table_all').fadeOut(500);
        $('#university_table_add').delay(600).fadeIn(500);
        document.getElementById('university_add').textContent =
'Table';
    }else{
        $('#university_table_add').fadeOut(500);
        $('#university_table_all').delay(600).fadeIn(500);
        document.getElementById('university_add').textContent = 'Add';
    }
}
$(document).on('click','#university_table_add_create',function(){
    //this event detects when a button click of id
"university_table_add_create" is clicked. we will then key the key and new
information from edited inputs.
    //we will not search for the name of uni;
    let uni=document.getElementById('university_table_add_name');
    let
location=document.getElementById('university_table_add_location');
    let
population=document.getElementById('university_table_add_name_population');

    if(uni==undefined || !string_con(uni,false) ) {
        //this is to ensure that the length provided is not anything
less than nothing, or is too long that it is unnecessary;
        swal({title:'Error in Univeristy name!',text:'The University
name provided was not long enough. A valid length, is anything over three,
and less than thirty -- or a special character was seen.',type:'error'});
        return;
    }

    if(location==undefined || !string_con(location,false) ) {
        //this is to ensure that the length provided is not anything
less than nothing, or is too long that it is unnecessary;
```

Jericho Uzzell (1702201) – ACCA5031 Web and Mobile Technologies – Sem1 – Assignment
1 – 09/01/2021 - UWTSD

```
        swal({title:'Error in location!',text:'The location provided
was not long enough. A valid length, is anything over three, and less than
thirty -- or a special character was seen.',type:'error'});
        return;
    }

    if(population==undefined || !number_con(population,false)) {
        //this input is an int, make sure the stated is not less than 3
        or 500,000 -- also check that it is a number;
        swal({title:'Error in population!',text:'The population needs
to meeting the following:\nMust be number\nMust be more than 3\nMust be
less than 500,000;',type:'error'});
        return;
    }
    //all good, let us now send us the message to php;

    $.post('/allqueries.php',{type:'adduniversities',post:{name:uni.value
,location:location.value,population:population.value}},function(data){
        console.log(data);

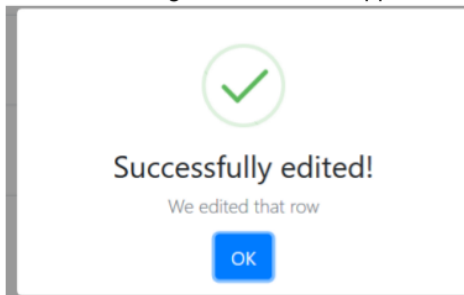
        if(isResultOkay(data,false)[1]==0&&isResultOkay(data,false)[0]==false
) {
            isResultOkay(data,false);
            return;
        }
        data=JSON.parse(data);
        if(data==undefined||data.length==0) {
            swal({title:'Error!',text:'Error in data, could not read
return data.',type:'error'});
            return;
        }
        jsGetUniData();
        addUni();
        swal({title:'Successfully added!',text:'We added that
row',type:'success'});
    });
});
```

3.4 Editing data

3.4.1. Screen Shots

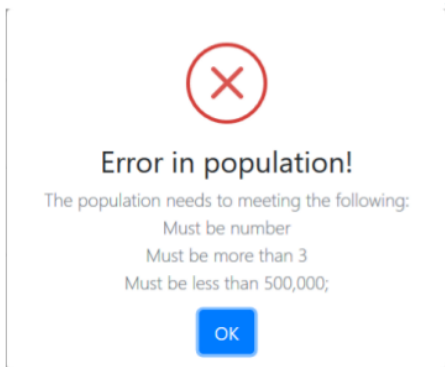
ID	University	Location	Population	
0 127	CardiffMet	CardiffCity	6000	Save Close

The above image shows what happens when you click the edit button.



The above image shows what happens when you click the save button.

Population	
6000E	Save Close
Value was less than 3 (letters) Value was not a number!	



The above images will show errors to the user. If the user clicks save, a more detailed popup will be shown.

3.4.2. Methodology

Editing data appends a row to the desired row and hides the original. Editing data follows on from the initial creation of them, they must follow the requirements such as, numbers only etc. Once saved, the data is sent to server code where a hidden value and an id verifies the row. This is written once again, mostly in JS with some JQuery elements where it was not avoidable.

3.4.3. Annotated Code Listing (paste in code)

```
$(document).on('click','edit_uni',function(){
    //this event detects when a button of class "edit_uni" is clicked. we
    will get the key from the button.
    //we then search the /resource/university.txt for the key, then
    return the row.

    if( !(this.getAttribute('key')).includes('-') ) {return;}

    let key=this.getAttribute('key').split('-')[0];
    let rand=this.getAttribute('key').split('-')[1];
    let row=this;
    random_number=randomNumberGen(); // need to generate a new number as
    js caches the txt document;
    $.get('/resource/universitynew.txt?q='+random_number,{},function(data
    ){
        console.log(data);

        if(isResultOkay(data,false)[1]==0&&isResultOkay(data,false)[0]==false
        ) {
            isResultOkay(data,false);
            return;
        }
        data=JSON.parse(data);
        if(data==undefined||data.length==0) {
            document.getElementById('university_output').innerHTML =
            '<tr><td colspan="6" class="text-center text-danger d-3">No data to
            load!</td></tr>';
            return;
        }
        let out;
        $.each(data,function(k,i){
            if(i.id==key && i.rand==rand) {
                out=i;
                return;
            }
        });
        if(!$_.isPlainObject(out)) {return;}

        $('.editing_row_hidden').show().removeClass('editing_row_hidden');

        $(row).parents('tr').eq(0).hide().addClass('editing_row_hidden');
        $('.editing_row').remove();
        $(row).parents('tr').eq(0).after('<tr
        class="editing_row"><td>'+key+'<br/><small class="text-
        muted">'+out.rand+'</small></td><td><input type="text" class="form-control
        input-md string_con" nonumbers nospecials placeholder="Name"
        value="'+out.name+'" maxlength="60" return_message="editing_row_name" /><p
        id="editing_row_name"></p></td><td><input type="text" class="form-control
        input-md string_con" placeholder="Location"
        return_message="editing_row_location" min="3" max="30" nonumbers nospecials
```

Jericho Uzzell (1702201) – ACCA5031 Web and Mobile Technologies – Sem1 – Assignment
1 – 09/01/2021 - UWTSD

```
value="'+out.location+'" maxlength="60" /><p
id="editing_row_location"></p></td><td><input type="number" class="form-
control input-md number_only" min="3" max="500000"
return_message="editing_row_population" placeholder="Population"
value="'+out.population+'" /><p
id="editing_row_population"></p></td><td><button class="btn btn-block btn-
sm btn-success save_uni" key="'+key+'-'+out.rand+'">Save</button><button
class="btn btn-block btn-sm btn-warning"
onclick="$($.editing_row_hidden\').show().removeClass(\`editing_row_hidden
\`);$($.editing_row\').remove();">Close</button></td></tr>';

    });

});
$(document).on('click','save_uni',function(){
    //this event detects when a button click of class "save_uni" is
    clicked. we will then key the key and new information from edited inputs.
    //we will search the /resource/university.txt for the key, we will
    then update the file if found.
    if( !(this.getAttribute('key')).includes('-') ) {return;}

    let key=this.getAttribute('key').split('-')[0];
    let rand=this.getAttribute('key').split('-')[1];

    let row=document.getElementsByClassName('editing_row')[0]; //get me
    this row;
    let
    rowhidden=document.getElementsByClassName('editing_row_hidden')[0]; //just
    to incase there are more than one -- we'll delete all anyway;

    let uni=row.querySelectorAll('td')[1].querySelector('input');
    let location=row.querySelectorAll('td')[2].querySelector('input');
    let population=row.querySelectorAll('td')[3].querySelector('input');

    if(uni==undefined || !string_con(uni,true) ) {
        //this is to ensure that the length provided is not anything
        less than nothing, or is too long that it is unnecessary;
        swal({title:'Error in Univeristy name!',text:'The University
        name provided was not long enough. A valid length, is anything over three,
        and less than thirty -- or a special character was seen.',type:'error'});
        return;
    }

    if(location==undefined || !string_con(location,true) ) {
        //this is to ensure that the length provided is not anything
        less than nothing, or is too long that it is unnecessary;
        swal({title:'Error in location!',text:'The location provided
        was not long enough. A valid length, is anything over three, and less than
        thirty -- or a special character was seen.',type:'error'});
        return;
    }

    if(population==undefined || !number_con(population,true)) {
```

Jericho Uzzell (1702201) – ACCA5031 Web and Mobile Technologies – Sem1 – Assignment
1 – 09/01/2021 - UWTSD

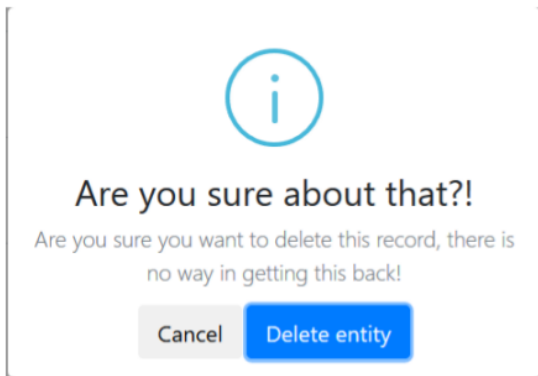
```
        //this input is an int, make sure the stated is not less than 3
        or 500,000 -- also check that it is a number;
        swal({title:'Error in population!',text:'The population needs
        to meeting the following:\nMust be number\nMust be more than 3\nMust be
        less than 500,000;',type:'error'});
        return;
    }
    //all good, let us now send us the message to php;

    $.post('/allqueries.php',{type:'updateuniversities',post:{key:key,ran
    d:rand,name:uni.value,location:location.value,population:population.value}}
    ,function(data){
        console.log(data);

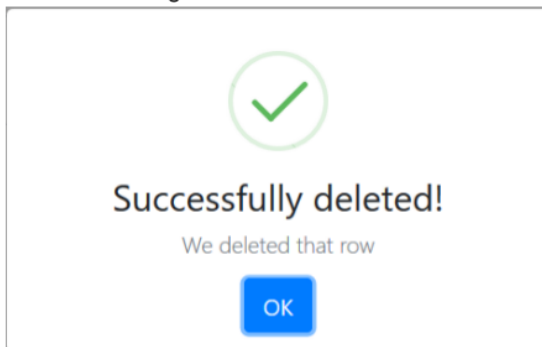
        if(isResultOkay(data,false)[1]==0&&isResultOkay(data,false)[0]==false
        ) {
            isResultOkay(data,false);
            return;
        }
        data=JSON.parse(data);
        if(data==undefined||data.length==0) {
            document.getElementById('university_output').innerHTML =
            '<tr><td colspan="6" class="text-center text-danger d-3">No data to
            load!</td></tr>';
            return;
        }
        jsGetUniData();
        swal({title:'Successfully edited!',text:'We edited that
        row',type:'success'});
    });
});
```

3.5 Deleting data

3.5.1. Screen Shots



The above image is what is shown if the user clicks delete.



If the user is sure about deleting an entity, they will click the previous delete entity button, this message will be shown.

2	Oxford Uni	England	6000	Delete
				Edit
				Delete

Here you can see the entity has been deleted.

3.5.2. Methodology

When deleting data, the student wanted the user to be absolutely sure that they wanted to perform this action. Once the delete button is clicked, it will call a plugin known as sweetalerts (swal), where the user must click confirm or deny. The rest of the code follows on from adding and editing data however, only the id and unique number is sent to server code, where it will delete the needed entity.

3.5.3. Annotated Code Listing (paste in code)

```
$(document).on('click', '.delete_uni', function() {  
    //this event detects when a button of class "edit_uni" is clicked. we  
    will get the key from the button.  
    //we then search the /resource/university.txt for the key, then  
    return the row.  
  
    if( !(this.getAttribute('key')).includes('-') ) {return;}  
}
```

```
let key=this.getAttribute('key').split('-')[0];
let rand=this.getAttribute('key').split('-')[1];
let row=this;

$.get('/resource/universitynew.txt?q='+random_number, {}, function(data)
){
    console.log(data);

    if(isResultOkay(data, false) [1]==0&&isResultOkay(data, false) [0]==false
) {
        isResultOkay(data, false);
        return;
    }
    data=JSON.parse(data);
    if(data==undefined||data.length==0) {
        document.getElementById('university_output').innerHTML =
'<tr><td colspan="6" class="text-center text-danger d-3">No data to
load!</td></tr>';
        return;
    }
    let out;
    $.each(data, function(k,i) {
        if(i.id==key) {
            out=i;
            return;
        }
    });
    if(!$.isPlainObject(out)) {return;}

    swal({
        title:'Are you sure about that?!',
        text:'Are you sure you want to delete this record, there
is no way in getting this back!',
        type:'info',
        showCancelButton:true,
        confirmButtonColor:'#779999',
        confirmButtonText:'Delete entity',
        closeOnConfirm:false
    },function() {deleteUni(key, rand);});

});
});
function deleteUni(key, rand) {
    if(key==undefined||key=='') {return false;}
    if(rand==undefined||rand=='') {return false;}
    $.post('/allqueries.php', {type:'updateuniversities', post:{key:key, ran
d:rand, name:null, location:null, population:null, delete:true}}, function(data)
{
        console.log(data);

        if(isResultOkay(data, false) [1]==0&&isResultOkay(data, false) [0]==false
) {
```

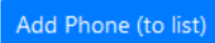
Jericho Uzzell (1702201) – ACCA5031 Web and Mobile Technologies – Sem1 – Assignment
1 – 09/01/2021 - UWTSD

```
        isResultOkay(data, false);  
        return;  
    }  
    data=JSON.parse(data);  
    if(data==undefined||data.length==0) {  
        document.getElementById('university_output').innerHTML =  
'<tr><td colspan="6" class="text-center text-danger d-3">No data to  
load!</td></tr>';  
        return;  
    }  
    jsGetUniData();  
    swal({title:'Successfully deleted!',text:'We deleted that  
row',type:'success'});  
    });  
}
```

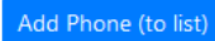
4. Spam page

4.1 Reload button

4.1.1. Screen Shots

All spam phones (list)  

The above image shows the reload button has loaded data, it can be clicked.

All spam phones (list)  

The above image shows the reload button has been clicked or is loading data already, it is disabled for the moment.

4.1.2. Methodology

The reload button uses pure JS as the JQuery equivalent is one line -- disabling the reload button is needed to stop spamming from the user, it is also an indication of data being retrieved.

4.1.3. Annotated Code Listing (paste in code)

```
onclick="jsGetPhoneData();" ; "
```

4.2 Getting data

4.2.1. Screen Shots

ID	Phone number	Location	Affiliates
1	07964391301 Phone type: Mobile Region: 7	ssss	ss
2	07964391301 Phone type: Mobile Region: 7	ssss	ss
3	07964391302 Phone type: Mobile Region: 93	Bagdad	No affiliates
4	01792517754 Phone type: Land line	Swansea	ellow
5	1234567891 Phone type: Land line	some location	

The above image shows all the data being loaded in.

4.1.2. Methodology

Gathering data from the phone spam list follows on from the previous page (adding universities). The only difference, the data is only viewable, once added it cannot be edited. The concept was, the user could add logs under the phone numbers for spam. The user is given information, on whether it was a home telephone or if it was a mobile. Mobiles have extra details to ascertain the region it called from. This code is mostly written in JS with JQuery elements where it was needed.

4.1.3. Annotated Code Listing (paste in code)

```
var random_number=randomNumberGen();
function jsGetPhoneData() {
    //this function gets the file contents of /resource/university.txt;
    //returns as text but is turned into json for our needs;
    let reloadbutton=document.getElementById('phone_reload'); //WE NEED
    TO GET THE RELOAD BUTTON;
    reloadbutton.setAttribute('disabled', true); //DISABLE IT TO STOP
    SPAMMING FOR THE MOMENT;
    reloadbutton.blur(); //STOP IS FROM BEING FOCUSED;
    random_number=randomNumberGen(); // need to generate a new number as
    js caches the txt document;

    $.get('/resource/spam-phones.txt?q='+random_number, {}, function(data) {
    //QUERY TXT FILE;
        console.log(data);

        if(isResultOkay(data, false) [1]==0&&isResultOkay(data, false) [0]==false
        ) {
            isResultOkay(data, false);
            return;
        }
        data=JSON.parse(data);
        // console.log(data.length);
        let uniHTML=document.getElementById('phone_email_output');

        if(data==undefined||data.length==0) {
            uniHTML.innerHTML = '<tr><td colspan="6" class="text-
center text-danger d-3">No data to load!</td></tr>';
            return;
        }
        let out='';
        uniHTML.innerHTML='';
        $.each(data, function(k,i) {
            // console.log(i);
            out='';
            out=out+'<tr>';
            out=out+'<td>'+i.id+'</td>';
            out=out+'<td>'+sanitise(i.phonenumber)+'<br/>Phone
type: '+(i.phonetype==0?'Mobile<br/>Region: '+(i.region:'Land
line')+'</td>';
            out=out+'<td>'+sanitise(i.location)+'</td>';
            out=out+'<td>'+i.affilitates+'</td>';
            out=out+'</tr>';
            uniHTML.innerHTML += out;
        });
    });
    setTimeout(function() {
        reloadbutton.removeAttribute('disabled');
    }, 1000);
}
```

```
function randomNumberGen() {  
    return ( Math.random() ).toString().split('.')[1].substring(0,2); //  
    need to generate a new number as js caches the txt document -- it will get  
    the document with a ?q=random_number so its fresh (we generate each time we  
    get it);  
}
```

4.3 Adding data

4.3.1. Screen Shots

Image

All spam phones (list) [Reload](#) [View list of phones](#)

Type of number
Select

Region
(+7) Abkhazia

Phone number
Phone number

Location
Can be empty
Location

Affiliates
Can be empty
Affiliates -- can be a company, names of potential people/person

Create

The above image shows the way you can add a spam phone number.

Type of number
Mobile

Region
(+7) Abkhazia

Phone number
aa

Value was less than 10 (letters)
Value was not a number!

The above image shows what happens if you have an error within the data.



Error in phone number!

The phone number needs to be within 10 and 12
numbers!



A more details error is return if the user clicks Create.

Type of number
Home
Region
(+7) Abkhazia
Phone number

The above image shows the transition between Home and Mobile number and how that interacts with the region.



Successfully added!

We added that row



The above image shows what happens once a phone number is added.

5	1234567891	some location
	Phone type: Land line	

The above image shows the data we just created.

4.3.2. Methodology

Adding the for phone numbers must have certain data depending on what is selected. If a mobile is selected, the region is required, if a landline is selected then we do not want to require the region. This is validated in pure JS in the website and on the server code -- the rest of the code that requires the number, location, affiliates and type are sent with their own requirements (numbers only etc).

4.3.3. Annotated Code Listing (paste in code)

```
function jsAddPhonesList() {  
    if( document.getElementById('phone_email_phones').textContent == 'Add  
Phone (to list)' ) {  
        $('#phone_email_table_all').fadeOut(500);  
        $('#phone_email_table_add').delay(600).fadeIn(500);  
        document.getElementById('phone_email_phones').textContent =  
'View list of phones';  
  
        document.getElementById('phone_email_table_add_typednumber').value='nu  
ll';  
  
        document.getElementById('phone_email_table_add_region').value='7';  
  
        document.getElementById('phone_email_table_add_phonenumber').value=''  
;  
  
        document.getElementById('phone_email_table_add_location').value='';  
  
        document.getElementById('phone_email_table_add_affiliates').value='';
```

Jericho Uzzell (1702201) – ACCA5031 Web and Mobile Technologies – Sem1 – Assignment
1 – 09/01/2021 - UWTSD

```
    }else{
        $('#phone_email_table_add').fadeOut(500);
        $('#phone_email_table_all').delay(600).fadeIn(500);
        document.getElementById('phone_email_phones').textContent =
        'Add Phone (to list)';
    }
}
$(document).on('click','#phone_email_table_add_affiliates_create',function()
){
    //this event detects when a button click of id
    "phone_email_table_add_affiliates_create" is clicked. we will then key the
    key and new information from edited inputs.
    let
    type_number=document.getElementById('phone_email_table_add_typednumber');
    let region=document.getElementById('phone_email_table_add_region');
    let
    phonenumber=document.getElementById('phone_email_table_add_phonenumber');
    let
    location=document.getElementById('phone_email_table_add_location');
    let
    affiliatates=document.getElementById('phone_email_table_add_affiliates');

    if(type_number==undefined || type_number.value=='null' ) {
        //this error happens when the user fails to select a type of
        number, mobile=0 or home number=1
        swal(
            {title:'Error in Type of number!',text:'You must select a type
of number, home or mobile!',type:'error'},
            function(){setTimeout(function(){type_number.focus();},200);}
        //after timeout, we will select the relivent box that error'd. timeout
        needed as the user clicks a confirm button;
        );
        return;
    }

    if(phonenumber==undefined || !number_con(phonenumber,false)) {
        //this input is an int, make sure the stated is not less than 3
        or 500,000 -- also check that it is a number;
        swal(
            {title:'Error in phone number!',text:'The phone number needs to
be within 10 and 12 numbers!',type:'error'},
            function(){setTimeout(function(){phonenumber.focus();},200);}
        //after timeout, we will select the relivent box that error'd. timeout
        needed as the user clicks a confirm button;
        );
        return;
    }

    if(location==undefined || !string_con(location,false) ) {
        //this is to ensure that the length provided is not anything
        less than nothing, or is too long that it is unnecessary;
        swal(
```

Jericho Uzzell (1702201) – ACCA5031 Web and Mobile Technologies – Sem1 – Assignment
1 – 09/01/2021 - UWTSD

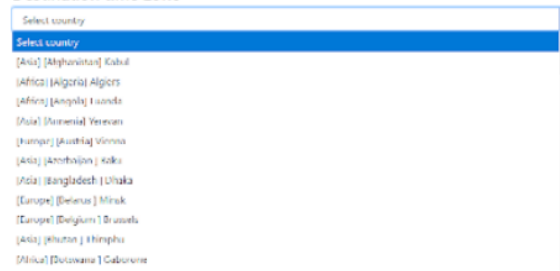
```
        {title:'Error in location!',text:'The location provided was not  
long enough. A valid length, is anything over three, and less than thirty -  
- or a special character was seen.',type:'error'},  
        function(){setTimeout(function(){location.focus();},200);}  
    );  
    return;  
}  
  
//all good, let us now send us the message to php;  
  
$.post('/allqueries.php',{type:'addspamphone',post:{phonetype:type_nu  
mber.value,region:region.value,phonenumber:phonenumber.value,location:locat  
ion.value,affilitates:affilitates.value}},function(data){  
    console.log(data);  
  
    if(isResultOkay(data,false)[1]==0&&isResultOkay(data,false)[0]==false  
) {  
        isResultOkay(data,false);  
        return;  
    }  
    data=JSON.parse(data);  
    if(data==undefined||data.length==0) {  
        swal({title:'Error!',text:'Error in data, could not read  
return data.',type:'error'});  
        return;  
    }  
    jsGetPhoneData();  
    jsAddPhonesList();  
    swal({title:'Successfully added!',text:'We added that  
row',type:'success'});  
    });  
});  
$(document).on('change','#phone_email_table_add_typednumber',function(){  
    let val=this.value;  
    if( val==0 )  
{document.getElementById('phone_email_table_add_region').removeAttribute('d  
isabled');}  
    else  
{document.getElementById('phone_email_table_add_region').setAttribute('disa  
bled',true);}  
});
```

5 Time page

5.1 Onload

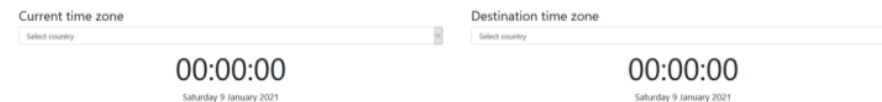
5.1.1. Screen Shots

Destination time zone



The above image shows on loading of the page, all the countries it will generate.

Locate the time zone for major cities.



The above image shows the current date when the page is loaded.

5.1.2. Methodology

It is important when the page is loaded that code is executed for the dropdown to select times. JQuery detects when the page has loaded, it will then check in JS certain things, such as if the cities array has data and if there are dropdowns on the page to place this data. The last of this code generates an array for the day names when converted into a time. We used JQuery to detect when the page was ready as the code to do this in JS was not necessary.

5.1.3. Annotated Code Listing (paste in code)

```
$(document).ready(function(){
    if(cities.length>0) { //IN CASE CITIES ARRAY HAS NOTHING IN IT;
        let
        class_gencountries=document.getElementsByClassName('generate-countries');
        //THE BELOW WILL GO THROUGH ALL CLASSES WITH THE CLASS NAME OF
        'generate-countries';
        for(let classes=0;classes<class_gencountries.length;classes++)
        {
            class_gencountries[classes].innerHTML = '<option
            value="null">Select country</option>';
            for(let i=0;i<cities.length;i++) {
                class_gencountries[classes].innerHTML += '<option
                value="'+i+'">['+cities[i][2]+'] ['+cities[i][0]+']
                '+cities[i][1]+'</option>'; //PUT THIS CITY INTO THE OPTION LIST;
            }
        }
        let time_date_setup=document.getElementsByClassName('timeformat
        date');
        if(time_date_setup.length>0) {
            let timenow=new Date();
```

Jericho Uzzell (1702201) – ACCA5031 Web and Mobile Technologies – Sem1 – Assignment
1 – 09/01/2021 - UWTSD

```
        var dayofweek =
["Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday"];
        var months =
["January","February","March","April","May","June","July","August","September",
"October","November","December"];

        for(classes=0;classes<time_date_setup.length;classes++) {
            time_date_setup[classes].innerHTML =
dayofweek[timenow.getDay()]+ ' '+timenow.getDate()+ '
'+months[timenow.getMonth()]+ ' '+timenow.getFullYear();
        }

    }

    setInterval(function(){updatetime();},100); //CHECKS THE TIME EVERY
100MS (1000MS IS EQUAL TO 1s)
});
var cities=[
    ["Afghanistan","Kabul","Asia"],
    ["Algeria","Algiers","Africa"],
    ["Angola","Luanda","Africa"],
    ["Armenia","Yerevan","Asia"],
    ["Austria","Vienna","Europe"],
    ["Azerbaijan ","Baku","Asia"],
    ["Bangladesh ","Dhaka","Asia"],
    ["Belarus ","Minsk","Europe"],
    ["Belgium ","Brussels","Europe"],
    ["Bhutan ","Thimphu","Asia"],
    ["Botswana ","Gaborone","Africa"],
    ["Bulgaria ","Sofia","Europe"],
    ["Croatia ","Zagreb","Europe"],
    ["Cyprus ","Nicosia","Asia"],
    ["Denmark ","Copenhagen","Europe"],
    ["Djibouti ","Djibouti","Africa"],
    ["Egypt ","Cairo","Africa"],
    ["Eritrea ","Asmara","Africa"],
    ["Estonia ","Tallinn","Europe"],
    ["Finland ","Helsinki","Europe"],
    ["Gabon ","Libreville","Africa"],
    ["Georgia ","Tbilisi","Asia"],
    ["Germany ","Berlin","Europe"],
    ["Ghana ","Accra","Africa"],
    ["Greece ","Athens","Europe"],
    ["Guinea ","Conakry","Africa"],
    ["Guinea-Bissau ","Bissau","Africa"],
    ["Hungary ","Budapest","Europe"],
    ["Indonesia ","Jakarta","Asia"],
    ["Iran ","Tehran","Asia"],
    ["Iraq ","Baghdad","Asia"],
    ["Ireland ","Dublin","Europe"],
    ["Israel ","Jerusalem","Asia"],
    ["Italy ","Rome","Europe"],
    ["Japan ","Tokyo","Asia"],
    ["Jordan ","Amman","Asia"],
```

Jericho Uzzell (1702201) – ACCA5031 Web and Mobile Technologies – Sem1 – Assignment
1 – 09/01/2021 - UWTSD

```
["Kenya ", "Nairobi", "Africa"],  
["Latvia ", "Riga", "Europe"],  
["Lebanon ", "Beirut", "Asia"],  
["Lesotho ", "Maseru", "Africa"],  
["Liberia ", "Monrovia", "Africa"],  
["Liechtenstein ", "Vaduz", "Europe"],  
["Lithuania ", "Vilnius", "Europe"],  
["Luxembourg ", "Luxembourg", "Europe"],  
["Mali ", "Bamako", "Africa"],  
["Mauritania ", "Nouakchott", "Africa"],  
["Moldova ", "Chisinau", "Europe"],  
["Monaco ", "Monaco", "Europe"],  
["Mongolia ", "Ulaanbaatar", "Asia"],  
["Montenegro ", "Podgorica", "Europe"],  
["Mozambique ", "Maputo", "Africa"],  
["Namibia ", "Windhoek", "Africa"],  
["Nepal ", "Kathmandu", "Asia"],  
["Niger ", "Niamey", "Africa"],  
["Norway ", "Oslo", "Europe"],  
["Oman ", "Muscat", "Asia"],  
["Philippines ", "Manila", "Asia"],  
["Poland ", "Warsaw", "Europe"],  
["Romania ", "Bucharest", "Europe"],  
["Rwanda ", "Kigali", "Africa"],  
["Senegal ", "Dakar", "Africa"],  
["Serbia ", "Belgrade", "Europe"],  
["Singapore ", "Singapore", "Asia"],  
["Slovenia ", "Ljubljana", "Europe"],  
["Spain ", "Madrid", "Europe"],  
["Sudan ", "Khartoum", "Africa"],  
["Sweden ", "Stockholm", "Europe"],  
["Taiwan ", "Taipei", "Asia"],  
["Tajikistan ", "Dushanbe", "Asia"],  
["Thailand ", "Bangkok", "Asia"],  
["Togo ", "Lome", "Africa"],  
["Tunisia ", "Tunis", "Africa"],  
["Turkmenistan ", "Ashgabat", "Asia"],  
["Uganda ", "Kampala", "Africa"],  
["Ukraine ", "Kiev", "Europe"],  
["Uzbekistan ", "Tashkent", "Asia"],  
["Yemen ", "Sana'a", "Asia"],  
["Zambia ", "Lusaka", "Africa"],  
["Zimbabwe ", "Harare", "Africa"]  
];
```

5.2 Selecting a country

5.2.1. Screen Shots

Image



The screenshot shows a web interface with two dropdown menus. The first, labeled 'Current time zone', has a selected value of '[Europe] [Ireland] Dublin'. The second, labeled 'Destination time zone', has a selected value of 'Select country'. Below the first dropdown, the time '10:46:46' is displayed in large font, with 'Saturday, 9 January 2021' below it in smaller font. Below the second dropdown, the time '00' is displayed in large font, with 'Saturc' below it in smaller font.

The above image shows what happens as soon as the user selects a country.

5.2.2. Methodology

When selecting a country, this code must check what type of dropdown this is, once known it will replace the time variable (current time or destination time) with its value (if it is null or an integer). If an integer is passed, it will get the country and continent, then it will print out the time using JS. Some elements such as the value and setting the time text on page could have been coded in JS.

5.2.3. Annotated Code Listing

```
$(document).on('change','generate-countries',function(){
    //when either current time or destination is changed, execute this;
    if($(this).val()=='null') {
        //this {THIS} value is null, reset this element;

        if(this.getAttribute('currenttime')!=undefined&&this.getAttribute('currenttime')!='') {autoupdater.current=null;}
        else {autoupdater.destination=null;}
        return false;
    }

    if(this.getAttribute('currenttime')!=undefined&&this.getAttribute('currenttime')!='') {autoupdater.current=$(this).val();}
    else {autoupdater.destination=$(this).val();}

    let city=cities[$(this).val()]; //GET CITY INDEX;

    let options = {
        timeZone: city[2]+'/'+city[1],
        hour: 'numeric',
        minute: 'numeric',
        second: 'numeric',
    },
    formatter = new Intl.DateTimeFormat([], options);
    let newdate=formatter.format(new Date());

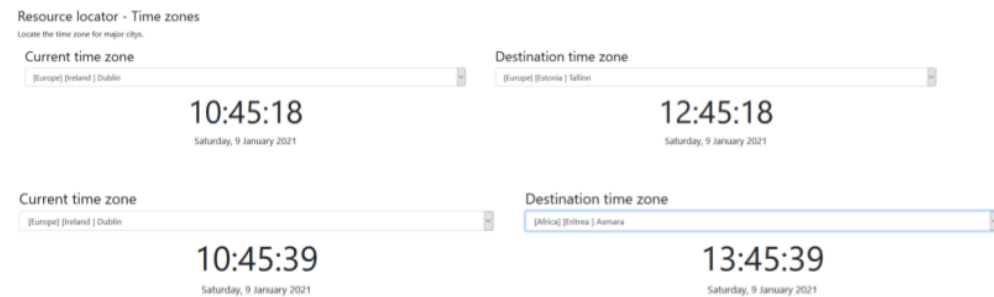
    $(this).parent().parent().find('.timeformat.time').eq(0).text(
    newdate );
    //    year: 'numeric',
    //    month: 'numeric',
    //    day: 'numeric',
    //    hour: 'numeric',
    //    minute: 'numeric',
```

Jericho Uzzell (1702201) – ACCA5031 Web and Mobile Technologies – Sem1 – Assignment
1 – 09/01/2021 - UWTSD

```
        // second: 'numeric',  
    });
```

5.3 Update time - Every second

5.3.1. Screen Shots



The above image shows the update function working. Updating the time for the two locations.

5.3.2. Methodology

The update time function is called every 100 millisecond, it will check if the current or destination time have an integer value within it (this represents an array index). If there is once, it will call the time for that index -- if the value is null for either or, the code does nothing. This code is written in JS as JQuery was not required. The student used a function as this code is called within an interval function and is also when the time dropdowns are changed.

5.3.3. Annotated Code Listing (paste in code)

```
function updatetime() {  
    //https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Intl/DateTimeFormat/DateTimeFormat  
    //THE ABOVE IS DOCUMENTATION ON HOW TO USE THIS...HOW TO GET FORMATS ETC;  
  
    //IF autoupdater.current (current time) is empty do not update it -- we could make it print but "please select a country";  
    if(autoupdater.current!=null) {  
  
        let time_date_setup=document.getElementsByClassName('timeformat time')[0];  
        let  
time_date_setu_2=document.getElementsByClassName('timeformat date')[0];  
  
        let city=cities[autoupdater.current];  
        time_date_setup.innerHTML =  
getDateTimeFormat(autoupdater.current,0);  
        time_date_setu_2.innerHTML =  
getDateTimeFormat(autoupdater.current,1);  
  
    }  
  
    //IF autoupdater.destination (destination time) is empty do not update it -- we could make it print but "please select a country";  
    if(autoupdater.destination!=null) {
```

Jericho Uzzell (1702201) – ACCA5031 Web and Mobile Technologies – Sem1 – Assignment
1 – 09/01/2021 - UWTSD

```
        let time_date_setup=document.getElementsByClassName('timeformat
time')[1];
        let
time_date_setu_2=document.getElementsByClassName('timeformat date')[1];
        time_date_setup.innerHTML =
getDateTImeFormat (autoupdater.destination,0);
        time_date_setu_2.innerHTML =
getDateTImeFormat (autoupdater.destination,1);

    }
}
```

356. Conclusion

Concluding this document, the student used a lot of pure JS code. The pure JS code was commonly used from JQuery, likely over 65%. The site the student created is required to run on a locally hosted computer that has PHP as a server side language for files within this document to be saved. This site in theory can read the files without server sided language however, it is not advised.

7. Bibliography

[https://developer.mozilla.org/en-](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Intl/DateTimeFormat/DateTimeFormat)

[US/docs/Web/JavaScript/Reference/Global_Objects/Intl/DateTimeFormat/DateTimeFormat](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Intl/DateTimeFormat/DateTimeFormat)

<https://morrisjs.github.io/morris.js/>

<https://jquery.com/>

<https://sweetalert2.github.io/>

yr2-sem1-webdevelopment-assignment1

GRADEMARK REPORT

FINAL GRADE

82/100

GENERAL COMMENTS

Instructor

WEBSITE

Range of Javascript code demonstrated - A decent amount of JQuery used and fits into the pages well. RawJS and JQuery has been mixed in some areas - I don't have any objection to that, I just note that this is acceptable (for the benefit of the second marker) Good use of validation - some of the code is a bit repetitive and could be slimmed down with user of parameters (lots of your if statements have an almost identical function) Time zones are good - might be a little over-engineered for what it is, but nevertheless you have shown some good skills 54/60 -

Overall integration of website - Nice, clean layout and good feedback to the user from the validation. I'm not entirely clear what the subject of the site was - it seems that it just demonstrates disparate JS functions?! 8/10

The pathing problem I think is this for example '/index.php' which takes you back to the root. You could use up a level to make it relative, for example '../index.php' This makes it easier to port your site between servers (as it may not always be in the root directory of a server, but perhaps in a subdomain or similar)

REPORT

Analysis of technical development - be careful on the style of your writing - don't use an active voice such as "The student chose" or "They used jQuery". ('The student' and 'They' are active) You need to use the passive voice - "xxx was chosen" "jQuery was used for xxx" You have written about what you did quite well, but you could have had a bit more of a technical analysis - not just what you did, but whether you think

your method was efficient and if you think you could improve it if you were doing it again. In hindsight, could you have approached it in a different way?

11/15 -

Annotated code listing - I think you need a bit more annotation in place - some areas are good, but in others the code comments are minimal 9/ 15 -

A good assignment overall and you have pushed your knowledge in the JS area. Tighten up your academic report writing style, this will be particularly important for the Independent Project in Year 3.

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35
