

COM601: Rich Internet Applications

Assignment 2

James Johnston(B00730860)
Romano Malocca(B00730863)

Table of Contents

Introduction	2
Background Research	2
Example 1: Mailbook	2
Example 2: Apple Contacts App	3
Example 3: Google Contacts	4
Trello	5
Design Process	5
JSON/Faker Data	6
jQuery Hover - To Display overview	7
Search for Contact- Related Contacts	8
How to access Help on completing a search	10
Easily Accessible and Alphabetical ordered list	10
Easily accessible list	10
List sorted alphabetically	11
SASS/SCSS/CSS	12
GULP	13
Responsive Design	14
Usability Testing	14
Key Points from testing.	15
Response to Negative Issues	17
References	19
Appendices	20
Appendix A: Mailbook	20
Appendix B: Apple Contacts	21
Appendix C: Google Contacts	22
Appendix D: Usability Questionnaires	23
Appendix E: HTML CODE	27
Appendix F: CSS	29
Appendix G: jQuery/JavaScript	40
Appendix H: Trello Activity	47
Appendix I: Node.js & Gulp.js	59
Appendix J: Package.json	63

Introduction

As a group we have been tasked with creating an online address book using JSON as a means to obtain the information on how contact information could be located for a range of people. The information would contain names, addresses and even a phone number which would provide the means to be able to contact the person. Our group consisted of Romano Malocca and James Johnston.

Background Research

Example 1: Mailbook

One example which exists of an online address book is mailbook. An example from this address book can be found in **Appendix A**. This is one good example of the online address book. This is because you can add information to the address book by clicking the add address which means that you can easily add information to the address book within the page. This is a strength within the example because this allows the user to add the address for people to the address book easily online and this means that they can add information to the page which will then show up in the list of addresses on the page.

Another strength that there is within the page is that you can search for a specific contact address and this will be refined within the search and addresses which are shown within the page which would mean that it can be easy to locate the information for the person that they would require easily. Searching would be a good element to include within the address book because this would mean that the time taken to locate an address would be greatly reduced because the search will only display the address for the name which has been included within the address book instead of having to look through all of the address information which is stored.

One weakness within the address book example is that when you select the address elements within the page which would open the address information would mean that the user can edit this information. However this could be classed as a weakness because when you click on the information this could be changed to be viewable only with the additional option to modify the information if it is needed.

So the Address book we are working on would help to improve upon this example because there will be a hover which will display a brief overview of the address information included for that person so that you would be able to check if this is the correct person that you are looking for within the address book before you actually click into the information.

Another way that the address book that we are working on would have a better functionality than the address book which is shown within the example is through the use of related contacts when the contacts within the address book are selected. So this means that it

would reduce the amount of time which is needed to locate the details for people within the address book as the relevant information will be displayed.

Example 2: Apple Contacts App

Another example that we examined was the Apple Contact Book Application which is apple's own address book application. This is the most up to date application which apple have which allows you to keep a record of addresses for people that you may know. This is a good application which will be used for the contact details which would mean that the addresses can be located easily within the application.

This is a strong address book application which will mean that you can easily organise the address information for contacts arranged by alphabetical order which means that the address book has an organised appearance. An example of this can be found in **Appendix B**. This means that you can easily locate the correct address information for the correct person easily as they have been located within the page which would mean that the names are displayed in an organised way as they have been arranged alphabetically within the address book.

There are strengths and weaknesses within the application which would mean that it would be suitable to be used as an application for an address book. One of these is the alphabetical ordering of the contacts which would mean that the addresses could be located easily as they have been arranged by the name which would mean that the page would mean that you would be able to locate the correct contact details as they have been arranged by name and displayed in this way. This also means that the information which would mean that it could be located easily within the page.

Another strength that there is within the application is the search functionality because as you search for the address information for any contact this is refined by the name which has been search which would mean that the information would be refined as the search is completed and this would mean that you can locate the contact details quickly because the search automatically refines the results live within the page.

There is another strength within the application which would allow the user to click the contact within the address book which would mean that the details would be located within the page and this would also mean that the contact information can be displayed within the page which would mean that the details are shown when they select the name from the list within the address book. This could be considered as a strength because it allows you to see all the details within the page. This would mean that they would be able to ensure that the information would stand out within the page.

There is only one weakness which can be identified which would mean that there is no hover to see a brief insight into more information when you hover over the contact details so it would not be easy to locate the correct information without having to constantly click the contact details which are included within the address book. This could be improved by

providing a small description when the user hovers over the information with either a mobile number or an address which means that you can be sure that you are going to select the correct option to begin with.

The address book we are working on would fit in the market with the apple contact application which would mean that you can see the contacts organized in alphabetical order which means that the details have been organised and not just placed into the address book in a random order. So this would allow an organised appearance within the page.

Our address book would also match with the market of online address books because there will be a hover which would appear when the user hovers over the contacts within the address book which would allow them to be able to locate the correct contact without having to constantly open the contacts in the address book to be able to locate the correct person that they would be looking for.

We would also include a search which would help to refine the contacts which are displayed within the page and this would also mean that it is possible to use a search to be able to locate a contact within the address book and we will also include a number of records information which will allow the user to see how many records there are within the address book which match the number of results that had been returned.

So the address book that we are developing would be suitable to be used as an online address book because contacts are displayed in an organised list which is how contacts are displayed within an address book. This will also include features which are not included within the example which has been examined. This is because we are adding a hover to the contacts which would help to reduce the time which would mean that it would be efficient to how to locate the correct contact details. There will also be a search which means that the user can interact with the address book and search for a particular contact within the page. So instead of having to look through a long address book they can be refined by carrying out a search on the contacts included.

Example 3: Google Contacts

Another example of an address book being used online would be Google Contacts. This example can be found in the **Appendix C**. This would be a good address book to use as it is online as well as the user can easily create contact as when entered into the address book there will be a big button that is hard to miss to allow the user to create their contacts adding the information that is needed. As it is Google it is updated regularly so it would be a safe environment for contacts that you may want to keep private or information that you may want to keep as Google would be a reliable place to place information as they have good security.

There are strengths and weaknesses within this online contact application. One of the strengths would be that you can select up to multiple contacts within the application and once the contacts are selected you can then click on the send email icon in which it will open gmail with the selected contacts that the user has then selected. An example would be that a

user that owns a business may want to send an email to certain employee rather than everyone in his company he can just select the employees they want to send the email to and then can send it.

A weakness that we have found is that there was no area that was recommended for the user to put any address in there was a section that the user could but this was under "notes" the user can put the address within that section, but it would have been nice to see a recommended section for this information.

Another weakness would be that there was no way to sort the information as some users like to sort in alphabetical order as it may be easier for the user to find people that they are looking for. As well as another weakness could be that to edit the information the user would have to click the pencil icon but that may be too small for a user to notice especially if they are visually impaired and the colour of the icon may be hard to see as it is a dark grey but when the information is highlighted it is a lighter grey so the colour do kind of blend together.

Trello

To help us organise the tasks that were needed to be completed within the page and also to help to ensure that the tasks were divided up and each person would have a clear understanding of the tasks that they are expected to complete as part of the development of the online address book. This meant that lists could be created containing cards which described each task and had the team member assigned to them so that they had an understanding of what they were to use so that the address book would include the features which would help to enhance its overall design. This also shows the activity that each person has carried out in the process of completing the design.

Design Process

At first we came up with a name that was quick and easy to use and remember, we called it "Fast Addresses" as for the quick and easy way to find addresses. Then we came up with an idea to create a logo that we would display within the website it will be the first thing the user will see when they venture onto our website. In turn these processes then allowed us to look at what features we wanted within the website. Then we drew up ideas on the different features that would be placed throughout the website and where and how they would be interacted with via the users. Then we started to draw up a template that then when we're ready to place out features into the website then the Base of it is already done all we would need to do is to add the features in.

JSON/Faker Data

We used the website of (<https://mockaroo.com/>) to generate the faker data that would be used within the JSON file, this website allowed us to create multiple contact information

without the need of coming up with random contact information. This would allow us to save time and use that time elsewhere in areas that would need a lot of time. This information would be used and spread throughout the address book that we are creating these contact information will help the user to locate that person that they are interested in finding.

```
var faker = require('faker');
var fs = require('fs');

var dataArray = [];

for(var i = 0; i < 15; i++){
    var data = {};
    data.image = faker.image.avatar();
    data.name = faker.name.findName();
    data.phone = faker.phone.phoneNumber();
    data.email = faker.internet.email();
    data.job = faker.name.jobTitle();
    data.streetName = faker.address.streetName();
    data.city = faker.address.city();
    data.zipCode = faker.address.zipCode();
    dataArray.push(data);
}

fs.writeFile('data.json', JSON.stringify(dataArray), (err) =>{
    if(err){
        err += '404 page not found';
        console.log(err);
    }
    console.log('the json file has successfully be generation!');
})
```

After trying the data that was created via mockaroo.com we could not get the data to display out onto a table in fact the data didn't appear at all. Then James and I decided to try another way that could display the data out this is where in one of the lectures we learned more about faker and fs(File Structure) modules that could be used by node.js. we both then decided that this was a way to go when generating the data. These will then help create the json file that we would then store the data then display it out onto the table.

We used the website of "<https://www.npmjs.com/package/faker>" to help ensure the correct extensions that we were going to be used when generating out our faker data, we generated data up to 15 contacts using a for loop as we thought that this would be a substantial amount of data to use but if needed more we could always increase the amount and then regenerate the json data file using node.js the data would then be stored within an array to allow the information to be generated.

Within the fakerfile.js the fs.writeFile is used so that the json file could successfully be created. This will take the dataArray and create it as json data. There is also an error which

will be executed if the file is not successfully created. This was done used the err parameter. This will mean that the user will be able to see a message displayed if they were unable to generate the file and there is also a message displayed to allow the user to understand that the file has been successfully created. This will be shown within the terminal when executing the node command to execute the file.

jQuery Hover - To Display overview

```
        },
        $(document).on('mouseover', 'td', function(){
            var current = $(this).attr("id");
```

The hover was also created through the use of jQuery which will activate once the user

hovers over the td element within the table. This will mean that the function will then be activated once the user hovers over the specified part of the table. Then the variable current will look for the attribute id on the element which the user is currently hovering over. This will mean that it would understand to look for the id of the element that is currently being hovered over within the event.

```
▼ $( function() {
    $(document).tooltip();
});
```

The user hovering over the table will show the hover data within the hover within the application. This means that the user will be able to see the user information

displayed as a tooltip when they have hovered over the table within the page.

```
$(".contact").attr("title", hover_data);
});
```

The code also looks for the element within the class name contact and the attribute title which is what the tooltip will use to display the information. Then the hover_data variable is used which will allow the information to be displayed. This allows the user to be able to hover over the name within the table then the persons data will be displayed to them.

```
▼ $.get("data.json", function(json) {
▼   $.each(json, function(key, value){
      hover_data += value.phone;
      hover_data += value.email;
      hover_data += value.job;
      hover_data += value.streetName;
      hover_data += value.city;
      hover_data += value.zipCode;
      for(var i; i<hover_data.length; i++){
```

There is also a .each which works within the hover that when the user hovers over the elements within the td

which would mean that the variable `hover_data` then has the values for the phone, email, job, streetName, city and zipCode will be included within the page. This means that when the user hovers over the name within the table they should be shown the contact details for the people within the table.

The `$.get("data.json")` will work that it will locate the json file which you are looking for and this will then mean that it will work to load the json data into the variable so that it will be able to successfully locate this within the page.

Search for Contact- Related Contacts

```
130 //information on easily within the address book which has been developed.
131 //search will activate on the key up function which means that as the user enters keys the information should
132 //refine on what has been displayed.
133 var search = $(this).val();
134 //looks at the table and each row of the table to ensure that the data within the table will be refined when
135 //the search is being carried out.
136 $("table tr").each(function(index){
137 //conditional statement will run as long as the index is not equal to zero as this is the heading of the
138 //table.
139 if(index != 0){
140 //assigns the variable name using the this selector which identifies that the name would be used to
141 //show or hide as it would be required.
142 var name = $(this);
143 //will find the name from the td elements within table which is what they are contained within on the
144 //page.
145 name.find('td').each(function(){
146 //reference variable which will be assigned the text name of the contact within the table that is
147 //being searched for.
148 var ref = $(this).text();
149 //conditional statement which is used to use the ref variable and .indexof which uses the search
150 //variable value to check if the number is not the one being search for and this will mean that the
151 //unrequired names will be hidden.
152 if(ref.indexOf(search)<0){
153 name.hide();
154 }
155 //else statement which will display the name which is being searched for. This will mean that the names
156 //will be displayed and refined as the search is carried out.
157 else{
158 name.show();
159 return false;
160 }
161 });
162 });
163 });
164});
```

search which would allow the user to be able to search for the information within the page which would mean that the user would be able to see the search making changes to the table as it is carried out within the page. This will mean that the information would be refined so that when they have completed their search the name which they have searched is the only item which is displayed within the table.

The search function will be activated when the user enters something into the element which has the search id attached to it within the html. Then the .on('keyup') is used to activate the function which will mean that the function will be activated once the user has entered a value to the search bar will be loaded into a search variable which will allow it to look through the table to see if there is any content within the table which matches the search bar value within the address book.

The search will then iterate through the entire table through the use of the \$("table tr") which means that it would mean that the search function should be carried out which would mean that the table could be looked through as the user enters a value into the search bar so that the correct results could be returned.

The table which displays the contact information which has been drawn from the json file which has been generated is also searchable which will allow the user to be able to locate the contact information for the person that they would like to access easily.

This is created as a live

There is a conditional statement within the function which will look at the index and as long as it is not equal to zero then the variable name will be used to identify the item which has been searched and will look for this item within the table through the name.find which is used to look at each of the td elements in the table until it finds one which matches the search. Then within this there is a function which will locate the item within the table and a variable ref is created which will help to identify the item within the search.

The ref is then used within a conditional statement which will look for the ref which has an index less than zero with the searched item added to it so that the index number is less than 0 as the user is searching for the contact from the address book that they desire.

If the item is not relevant to the search item then it will be hidden so that the information will be refined depending on what the user has been searching. This means that the user would be able to see the information hidden as they enter more into the search and this would further refine the content of the address book table.

If the item matches the information which has been entered then the name will be displayed on the table within the address book. This means that the search will be carried out live within the page so as the user enters the letters names which contain these letters would be refined until the final contact shown is the one that you have searched for while the rest of them have been hidden.

The search since it is looking for the specific contact within the table it is case sensitive so it means that the first letter is the capital and the remainder has to be in lowercase. This is because it is carrying out a live search of the table for the information which is being searched.

With having the search for contacts working we then decided to add a numbered search result this is so when the user is to type in any names within their contacts it will display the numbers that are displaying.

The var rowCount will look at the contact table and will look at the tbody within the table then will look for how many rows are shown within the table and then make the number visible to the user when they search through the contacts this is by looking for the ID within the code in which it will know where to display the number.

```
        else{
            name.show();
            return false;
        }
    });
});
var rowCount = $('#contact_tbl>tbody>tr>td:visible').length;
document.getElementById('count').innerHTML=rowCount;
});
```

How to access Help on completing a search

A guided tool tip was created, we discussed how it was going to be used then we came up with the idea of a multi step guide that a user can click through and tell them important information within the address book. Then we decided to have it just appear when the user visits the website that way it is easy to explain to a user and the user will understand it much quicker as they are told straight away, as well as when the user is finished looking at the guide they can just close it down and then a button will appear at the top of the website where if needed they can just click it and it will bring back the guide.

Easily Accessible and Alphabetical ordered list

```
20  </style>
21  <script src= "https://ajax.googleapis.com/ajax/libs/jquery/3.1.0/jquery.min.js"></script>
22  </head>
23 <body>
24 <t><table class="table table-bordered table-striped" id="contact_tbl">
25 <tr>
26 <th>Name</th>
27 </tr>
28 </table>
29
30
31 <script>
32 $(document).ready(function(){
33 $.ajax({
34   url: 'data.json',
35   dataType: 'json',
36   success: function(data){
37     for(var i=0; i<data.length; i++){
38       var row = $(<tr><td>' + data[i].name + '</td></tr>');
39       $('#contact_tbl').append(row);
40     }
41   },
42   error: function(jqXHR, textStatus, errorThrown){
43     alert('Error: ' + textStatus+ ' - ' + errorThrown);
44   });
45 });
46 });
47
```

Easily accessible list

The names of people within the json file will be displayed within a table whenever this loads within the page. This is done using the table which is given the id contact_tbl which is used within the jQuery Ajax call to identify the table so that the data which has been selected could be outputted to the table.

The jQuery .ajax call is used to call the json file which is called data.json within the url so that it understands that this is the table which will be used to extract the data for name and display this within the table. The data type is defined as json so that the ajax call understands that the data it is working with is json data.

Then to ensure that something occurs when either successful or if it fails to load the success and error options are used so that it can measure if this has been successful within the design. If the ajax has been successfully able to locate the json file which it has been assigned then a for loop will be activated which will run for the length of the data which has been assigned to the success function. This means that it will run on a loop until it has located all of the desired data from the data.json file. Then a row will be created for the name data from the json file and this will take the data and index number ie. data[i] from the screenshot and use the name assigned to the corresponding data. This will mean that each row within the table which is displayed will contain only one name. This row will then be appended to the table within the page so that the user would be able to see this information

displayed within the page. The error state will only be activated if the file has not been able to successfully locate the json file. This will then output a message alerting the user to the error that has occurred and why the data has not been shown successfully.

List sorted alphabetically

```

50 $("th").click(function(){
51     //variable which assigns the table to the function so it knows what to use
52     var tbl = $(this).parents("table").eq(0);
53     //variable for contact data in the table. Will be used to sort the information within the table. will then sort
54     //the table using the compare function which has been defined to compare sort the data
55     var cntact_data=tbl.find('tr:gt(0)').toArray().sort(compare($(this).index()));
56     //this will run a for loop as long as the data length is run and will sort the data the length of the table.
57     for (var i = 0; i < cntact_data.length; i++){
58         tbl.append(cntact_data [i]);
59     }
60 }
61 //compare function which will evaluate the data in the table and will sort it into alphabetical order so that the
62 //table would have been sorted into alphabetical order.
63 //internal function which will activate when the compare function has been called. This will ensure that the
64 //data has been correctly sorted so that it will be displayed in alphabetical order.
65 function compare(index) {
66     //variable within the function which will it will use to compare the data which are A and B which are sensible
67     //variables to use which will mean that the data within the table will be sorted. this will take the parameter
68     //index from the compare variable so that it will understand which value has been assigned and the variable A
69     //will take in a from the function which will be returned when this is activated and the B variable will take in
70     //the b parameter.
71     var A = ContactName(a, index), B = ContactName(b, index);
72     //will change the A variable and the B variable to be numeric so that the value can be compared and changed if
73     //B is less than A and then convert back to string and compare to ensure that the data has been arranged to be
74     //alphabetical.
75     return $.isNumeric(A) && $.isNumeric(B) ? A - B : A.toString().localeCompare(B);
76 }
77 //function getCellValue will return the value for the row of the table.
78 function ContactName(row, index){
79     //will return the row child as the element which makes up the row of the table and adds this to the index as
80     //text so that it will be loaded into the compare function so that the table could successfully sort the data
81     //within the table.
82     return $(row).children("td").eq(index).text();
83 }

```

The table was then coded using jQuery so that it could be sorted to be alphabetically displayed within the address book which has been designed.

This would mean that the table can be sorted so that the contact information could be located easily within the page and this would also mean that the

names would be sorted to be alphabetical. This will be carried out once the user clicks the table header within the page which is name it will activate functions which will work to sort and arrange the content of the table in alphabetical order. This allows the user to have the control to sort the information in alphabetical order within the page.

Then when the user clicks the table header they will be able to sort the contacts within the table because it will locate the table within the page and load this to the `tbl` variable which has been defined. Then the contact names are assigned to a variable as long as the value of the index is greater than 0 and then moves this to an array which will then be sorted using the compare function to ensure that the information within the table could be arranged in alphabetical order. Then a for loop will run for the length of the array `cntact_data` which will mean that this will then append the data to the table if they need to be rearranged to be alphabetical. This will mean that the order of the information within the table will be changed.

There is also a compare function which will compare the information in the table and change the order if the information is in the incorrect location and there is another name which should be displayed before the information within the page. This has two variables assigned to it which would mean that the information could be compared with the one below it which would mean that it would change within the design which would mean that the values of variables A and B to be numeric so that they could be compared and exchanged and then changed to string and compare with B to check if the name should be before or after the name which is being compared. The variables A and B will access another function which will retrieve the name from the table within the page.

This will be used to load each of the names into the compare function as this is where the contactName function is called for the names and under the ContatName function this will take the td elements of the table and the index number and will load them into the index parameter which is then called under the compare function. This will allow the names to be loaded so that they could be sorted within the table and will mean that the new list which is displayed will be arranged in alphabetical order.

Within the compare function the A variable is changed to a string again at the .toString() and then this will be compared with the B variable through the .localeCompare(B) so that it can be sorted to be in alphabetical order once the user has activated this through the clicking of the table header.

The .sort(compare(\$this).index()) line will use the function to compare the table and sort the information which has been displayed so that it will be sorted into alphabetical order within the page. This will mean that it will understand how to sort the information within the table and that it is the table that the user wishes to sort because the \$this is used to identify that it is this feature you would like to be sorted within the page.

So the variables A and B will be assigned the names which would mean that the names would be assigned to each of the variables and would be compared and if A is a lesser value than B then nothing will change however if Bs value was smaller than A then this will be changed so that the order within the table would be changed so that the table would then be arranged to be in alphabetical order.

SASS/SCSS/CSS

Using SCSS to help create the CSS file in which will help create and style out an HTML file that would contain all the bits of code that we have created within our assignment. This will help create the look that we imagined whenever creating the proper website. Such as moving it around the web page as well as size and colours of any of the text and can even change the font type.

GULP

Gulp is the build system that we made use of to allow use to compile the SASS/SCSS file into the css which would allow it to create the css file which would contain the styling for the address book which we have been working on. This uses modules which are installed through the use of the terminal and work through the utilisation of the node.js. This helps to automate tasks which would be repetitive to complete. Which makes this good for working with the creation of the styling as there may be multiple elements within this which are the same and this helps to reduce the repetitiveness of the task.

So this will work by located the SCSS file and will then work to compile this into the css file. This will be placed within the styling folder for the application which is being developed. This means that the css styling could easily be located within the page.

```
James-MacBook-Air:assignment_2 jamesjohnston$ gulp style
[16:25:47] Using gulpfile ~/Documents/assignment_2/gulpfile.js
[16:25:47] Starting 'style'...
[16:25:48] Finished 'style' after 1.28 s
James-MacBook-Air:assignment_2 jamesjohnston$
```

Within the terminal this will be used to call the style function which has been created within the gulpfile.js. This will mean that it will look for the file .scss and then will compile this into a minified style.css and will place it into the css folder. This means that the creation of the final css is automated within the development of the application.

```
body,html{background-color:#9db5eb;height:100%;font-family:Avenir,sans-serif}header{height:70px;max-height:300px;width:100%;position:relative;overflow:hidden}curvedbanner{background-color:#093394;background-size:cover;background-attachment:fixed;position:absolute;top:0;left:-30px;right:-30px;bottom:0;border-bottom-right-radius:50px;border-bottom-left-radius:50px;z-index:1}hi{color:#000;text-shadow:1px 1px 1px #666}hi,(position:absolute;left:50%;-webkit-transform:translate(-50%,-50%);transform:translate(-50%, -50%);margin:0;font-size:1.2em}.logo,h2{z-index:1}h2{font-size:1.2em}.logoh2{font-size:1.2em}h2,(position:relative)footer{height:70px;top:100%;max-height:40px;width:100%;overflow:hidden}.footerbox{background-color:#093394;background-size:cover;position: absolute;top:0;left:-30px;right:-30px;bottom:0}h3{font-size:25px;font-weight:100;color:#2c2c2c;margin:20px 25px}.open{(position: absolute; width:100px; height:40px; left:-30px; top:30px; margin-left:-80px; margin-top:-10px; border:2px solid #bdb411; background-color:#093394; border-radius:6px; padding:10px 30px; color:#bdb411).open,.open:hover{ -webkit-transition:all .6s ease-out; transition:all .6s ease-out; open:hover{border:2px solid #aaa; box-shadow:inset 0 0 0px #ccc}}.con{(position: fixed; width:400px; height:238px; left:50%; top:50%; margin-top:-119px; margin-left:-200px; background-color:#f3f3f3; border-radius:10px; box-shadow:0 0 24px rgba(0,0,0,.4); z-index:1}.con:before{content:""; position: absolute; left:
```

This shows the outcome which is a minified css file which can be hard to follow. In Appendix F we have included our full CSS file which has been made easier to read with the spaces but with gulp for the movement from development to production version the css has been minified.

Responsive Design

Making the document responsive is important as you would like the user experience to be enjoyed no matter the device or size that the user is using the website on. We value the users and how they choose to use our web page so that they would continue to use the website wherever they choose to do so.

Having this meta tag that has the width=device-width within the <head> along the top of the HTML will allow the website to know that it is responsive and within the CSS if the scale is set to 100% it will look at the device that it is displaying on and will show with the right dimensions of the device.

```
<title>Fast Addresses</title>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<a href="#">
```

Usability Testing

We carried out some usability testing on the digital address book which had been carried out and this allowed us to see if there were any areas within the development which needed to be improved upon to allow it to be created and improved functionality for an online contact book.

The questionnaires can be found within appendix D;

Questions	Person 1 Response	Person 2 Response	Person 3 Response
Was the Guide helpful?	I felt that the guide was very helpful	Yes, there were 5 steps in order to guide me through the website	Yes, the guide was helpful for me as a user
Did the Search Work?	Yes	Yes	Search worked when searching for name i was looking for
Any ways to improve the search?	Not that I could find	No, the website works very effectively	Does what I need to do to work
Did you try sorting Alphabetically	Yes it showed up in alphabetical order	Yes, works in the correct manor	Yes i tried sorting alphabetically and it worked
Did it Sort Alphabetically?	Yes	Yes	Yes
Did the hover effect show relevant information	Yes but it showed all information rather than one persons	Plenty of information was given about each contact	Yes and it displays all information instead of the one who is hovered
Is the web page responsive?	Yes	Yes, can be minimised	Yes
Is the web page colours eye pleasing?	Yes	They attract a user immediately	Yes
Would you say that the webpage is efficient	Yes i would	Yes	Yes it is

Any Suggestions for the web page?	Maybe an image displayed with the name	Maybe add more names and relative names into the table	Maybe style the list of names to display in the middle.
-----------------------------------	--	--	---

Key Points from testing.

The first key point which we have found from the user testing was that they all found that the guide which was created which would help to explain to them how they would be able to make use of the website. This included a step by step guide which one user said it was helpful because it explained to them how they would be able to make use of the application. This would mean that the user would have a clear understanding of what they have to do to carry out the different tasks within the address book.

Another element which would mean that the user testing would be important would be the comments that have been given for the search within the page which would mean that they would be able to search for the contact that they would prefer. This would mean that the user can easily find the contact that they would desire from the use of the search within the address book which has been created.

Another key element which would mean that the user testing stated was how the table was sorted alphabetically this was clearly shown within the guide how it was possible to sort the table to see the contacts displayed in alphabetical order by clicking on the header of the table which will sort the contacts alphabetically by name.

Another important point which has been made from the user testing is that the colours which have been used within the address book are pleasing to the users eye. This we feel is a key point which has been raised from the completion of user testing which would mean that they are not harsh on the eye. This would be a key point because this would mean that the user would be more likely to make use of the application because the colours are not really harsh on their eye. This was a key consideration in the development as we wanted to be sure that the colours were not harsh or clashed in any way when they have been developed. So to hear that we had been successful in this was good from the user testing that we have carried out.

Another important element that we have achieved is through the use of the hover which the users all said that it was good because it only displays information which would mean that they would be able to see the additional data which is stored for each of the people. We feel that this was a good piece of information to hear back because we wanted to ensure that there was nothing within the hover which is not relevant to the information which has been included within the page.

Another point that was made was that the address book was that it was responsive. This was an important element which we wanted to ensure was because this would mean that the address book would be suitable to be used for by the user on a wide range of devices. This

means that we would have a greater number of user making use of the address book because not everyone makes use of the same type of device to make use of the address book. So it is good to see that the user was able to see that the address book was responsive within the design.

Another key comment within the testing was that the address book was efficient because this allows them to carry out their tasks in an efficient manner. This will also mean that the address book runs at an efficient rate and loading times are not taking a greater length of time within the page. This is a good comment to have received because throughout the development we wanted to ensure that the design would stand out within the page and this also means that there would be a greater user base because from the user testing we have found that it is efficient at loading and allowing the user to carry out tasks as they require within the address book.

There was also some slight negative responses which were good because these will help to improve the overall appearance of the address book which has been created. This would mean that the design and appearance could be improved.

One way that this would mean that the design would mean that the design would be improved is through the inclusion of a greater number of contacts within the address book. This would mean that the address book would have a greater appearance within the design which would mean that the user will be presented with a more detailed address book and it would also mean that they would be able to see a greater level of results which they could then refine. It was also suggested that relative names could be included so there could be people included which would have the same first name so that there could be contacts which have something in common which would mean that they would be able to see this within the page.

Another important piece of feedback that we had completed was that there could be an improve the styling with the positioning of the table to make it more dominant on the page by centering it within the page. Which would mean that this would mean that the design would stand out within the page.

So the user testing was useful within the development of the application which would mean that we were able to see how we have been able to achieve the work to a standard that we had desired. This means that we would be able to see where we have fallen out of this within the design which would mean that if there were any weaknesses within the design this could stand out within the testing and would mean that through carrying it out would allow us to be able to ensure that the final overall design would be suitable.

So we feel that the user testing was helpful because this meant that elements which could do with a little more work on them could be carried out and improve the overall design of the final address book.

Response to Negative Issues

We did a user testing with 3 different users and upon getting these reviews done we got good feedback but as there is good feedback we also got some negative one that in turn would help us create a better product for users.

One user suggested that we fix some issues with how the table was displayed within the product so we went into the CSS and changed how it looked with changing colours to stand out more to display the table in the middle so that it is centered within the website as it would look cleaner.

Another small piece of negative feedback that we had been given was that when the user hovers over the names within the table they are presented with all of the information on the design which would mean that they would only like to see the information for one person at a time. So we took time to look into how this could be achieved and tried out a for loop to see if we could return the individual data within the page.

So we attempted to fix this element within the page which would mean within the address book we would be able to improve the overall appearance because this would mean that the information displayed on hover would be unique to that one single user within the page. We tried this through the use of the console log when the user hovers over the table. This was carried outside the website so that we could test it out without it breaking any other element within the website which was working. This for some reason showed each of the content within the file on separate lines.

This shows the for loop which i used which runs for the length of the hover_data array. And will return the corresponding data item when the user hovers over it.

However when this was modified to the title attribute so that it would display on the page instead of the console. This was displaying each individual element instead of all of the information within the page. This meant that it was only showing one single value instead of the full dataset for each person and was a random number or the first letter of an email. This was not suitable for the address book.

So since this was not working we reverted back to the original code because we were at least able to have the data displayed within the hover. Yes it was showing everyones instead of nothing displaying. This also means that the user would be able to see data displayed when they hover over it within the page.

Another piece of feedback that we received was that there could be more people within the address book. We started with 15 contacts as the development version of the address book. This was generated through the use of faker which would create file.

To increase the number the number 15 was increased to a larger number as shown within the for loop in the image above. This means that when activated through node this would generate the data.json file which will contain a set of 40 data items. However, we changed this to be 20 instead as we felt 40 would be to many.

This shows the node command which runs the fakerfile.js which would create the data.json file. This shows it has been successful because a console.log message has been displayed which says the json file has successfully been generated. This allows you to see it has been successful. This allowed us to easily add additional people to the address book through the use of faker data.

We kept in some of the data and removed some of them as they were messing up the footer of the table. So instead of 20 we ended up with 17 contacts which is still a few more than we had already in. This was because it kept throwing our styling a little.

References

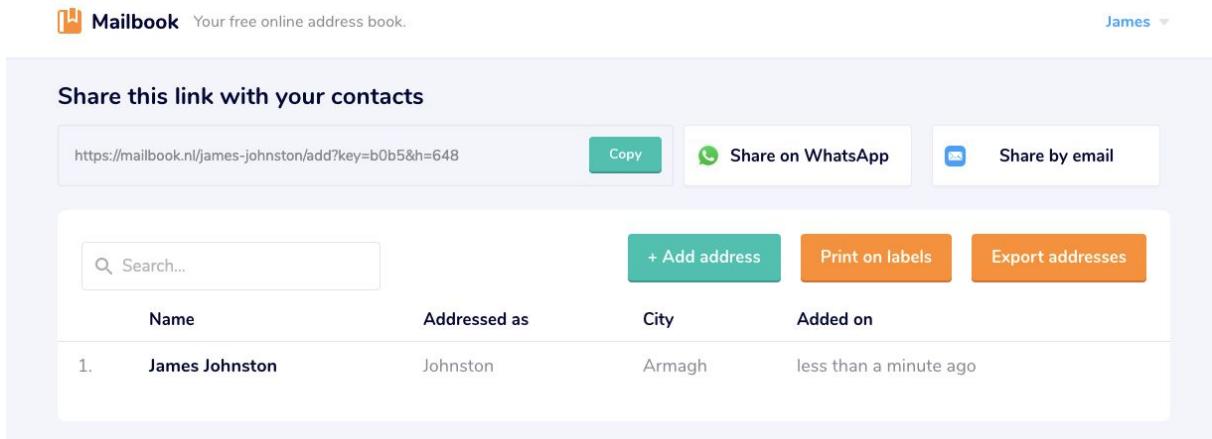
Apple Contacts(2019). *iCloud*. [online] Available at: <https://www.icloud.com/contacts/> [Accessed 29 Oct. 2019].

Google. (2019). *Google Contacts*. Available: <https://contacts.google.com/?hl=en&tab=rC>. Last accessed 29/10/19.

Mailbook. (2019). *James Johnston*. [online] Available at:
<https://mailbook.nl/james-johnston/addressbook> [Accessed 28 Oct. 2019].

Appendices

Appendix A: Mailbook

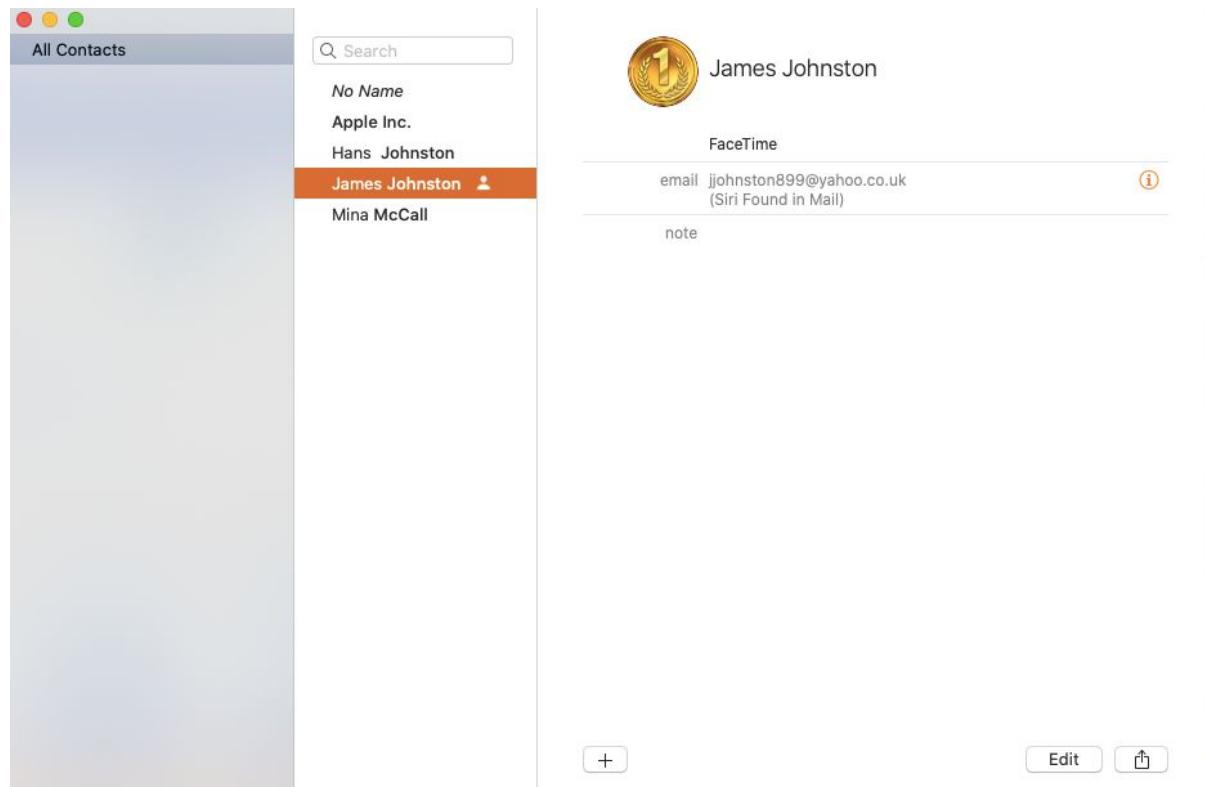


The screenshot shows the Mailbook interface. At the top, there's a header with the Mailbook logo and a link to James. Below the header, a section titled "Share this link with your contacts" displays a URL (<https://mailbook.nl/james-johnston/add?key=b0b5&h=648>) with "Copy", "Share on WhatsApp", and "Share by email" buttons. Below this is a search bar and a row of buttons: "+ Add address", "Print on labels", and "Export addresses". The main content area shows a table with one contact entry:

	Name	Addressed as	City	Added on
1.	James Johnston	Johnston	Armagh	less than a minute ago

(Mailbook, 2019)

Appendix B: Apple Contacts



(Apple, 2019)

Appendix C: Google Contacts



The screenshot shows the Google Contacts interface. At the top left is a navigation bar with three horizontal lines, a person icon, and the word "Contacts". To its right is a search bar with a magnifying glass icon and the word "Search". Below the navigation bar is a button labeled "Create contact" with a plus sign and a person icon. The main area is titled "CONTACTS (2)". It lists two contacts: "James Johnston" and "Romano Malocca". Each contact entry includes a small profile picture, the contact's name, their email address, and their phone number. On the left side of the main area, there are three buttons: "Contacts" (which is highlighted in blue), "Frequently contacted", and "Duplicates".

CONTACTS (2)
 James Johnston james.johnston981@outlook.... +447596325478
 Romano Malocca malocca10@gmail.com +447565423658

(Google Contacts, 2019)

Appendix D: Usability Questionnaires

Questionnaire: Person 1

1. Was the guide helpful?

I felt that the guide was very helpful

2. Did the search work?

Yes

3. Any ways to improve the search?

Not that I could find

4. Did you try sorting by alphabetical?

Yes

5. Did it sort alphabetically?

Yes

6. Did the hover effect show relevant information?

Yes, but it showed all information rather than one persons

7. Is the web page responsive?

Yes

8. Is the web page colours eye pleasing?

Yes

9. Would you say that the web page is efficient?

Yes I would

10. Any suggestions for the web page?

Maybe an image displayed with the name

Signed: Stephen Wilson

Questionnaire: Person 2

1. Was the guide helpful?

Yes, there were 5 steps in order to guide me through the website

2. Did the search work?

Yes

3. Any ways to improve the search?

No, the website works very effectively

4. Did you try sorting by alphabetical?

Yes, works in the correct manor

5. Did it sort alphabetically?

Yes

6. Did the hover effect show relevant information?

Plenty of information was given about each contact

7. Is the web page responsive?

Yes, can be minimised

8. Is the web page colours eye pleasing?

They attract the user immediately

9. Would you say that the web page is efficient?

Yes

10. Any suggestions for the web page?

Maybe add more names and relative names into the table
Signed: Jonathan Helmcke

Questionnaire: Person 3

1. Was the guide helpful?

Yes, the guide was helpful for me as a user

2. Did the search work?

Search worked when searching for name I was looking for

3. Any ways to improve the search?

Does what I needs to do to work.

4. Did you try sorting by alphabetical?

Yes I tried sorting alphabetically and it worked

5. Did it sort alphabetically?

Yes

6. Did the hover effect show relevant information?

Yes and it displays all information instead of the one who is hovered

7. Is the web page responsive?

Yes

8. Is the web page colours eye pleasing?

Yes

9. Would you say that the web page is efficient?

Yes it is

10. Any suggestions for the web page?

Maybe style the list of names to display in the middle. Signed: Jack Reid

Appendix E: HTML CODE

```
1  <!DOCTYPE html>
2 ▼ <html lang="">
3
4 ▼ <head>
5    <meta charset="utf-8">
6    <title>Fast Addresses</title>
7    <meta name="viewport" content="width=device-width, initial-scale=1.0">
8    <link rel="icon" href="images/Fast%20Addresses.png">
9    <link rel = "stylesheet" type = "text/css" href = "jquery-ui.css">
10   <link rel="stylesheet" href="css/style.css">
11   <link rel="stylesheet" href="css/bootstrap-tour.css">
12   <!--<script src="JS/script.js"></script>-->
13   <script src="JS/bootstrap-tour.min.js"></script>
14   <script src='https://cdnjs.cloudflare.com/ajax/libs/jquery/2.1.3/jquery.min.js'></script>
15   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/normalize/5.0.0/normalize.min.css">
16   <script src='https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js'></script>
17   <script
18     src="https://code.jquery.com/jquery-3.4.1.js"
19     integrity="sha256-WpOohJOqMqqyKL9FccASB900KwACQJpFTUBLTYOVvVU="
20     crossorigin="anonymous"></script>
21 </head>
22
23 ▼ <body>
24
25
26 ▼ <header role="banner">
27 ▼   <div class="curvedbanner">
28     <div class="logo"></div>
29   <!--<h1>Fast Addresses</h1>-->
30   </div>
31 </header>
32 <!--Start Of Guide-->
33 ▼ <div class="con">
34   <h3>Helpful Guide</h3>
35   <span class="close"></span>
36 ▼   <div class="slider-con">
37 ▼     <div class="slider-turn">
38       <p>This is your Guided Tool Tip for our Website of Fast Addresses!</p>
39
40       <p>There will be a search bar along the top that will allow you to click on and will allow you to search
41         for someone that you have saved.</p>
```

```

42      <p>To sort by alphabetic click "Name" at the top of the table</p>
43
44      <p>As can be seen there is a search results that will count how many names are viewed at once.</p>
45
46      <p>If the guide is needed again there will be a button along the top of the website with the word "Guide"
47      once clicked on the guide will re-appear</p>
48  </div>
49
50 ▼  <div class="bottom">
51 ▼  <div class="step">
52      <span></span>
53 ▼  <ul>
54      <li data-num="1"></li>
55      <li data-num="2"></li>
56      <li data-num="3"></li>
57      <li data-num="4"></li>
58      <li data-num="5"></li>
59  </ul></div>
60  <button class="btnnxt">Next</button>
61  </div>
62 </div>
63 <button class="open">Guide</button>
64
65
66  <!--Search Bar to allow you to be able to refine the names on the contacts in the table-->
67 ▼ <div class = "search_cnt">
68 ▼ <div class = "searchbar">
69  <input type = "text" id = "search" placeholder = "Live Search">
70  <button type = "submit" class = "search_btn"><i class = "fa fa-search"></i></button>
71 ▼ <div id = "result">Results Found:
72 <div id="count" name="count">
73 </div></div>
74 </div>
75 </div>
76

```

```

77  <!--Table which will be populated with faker data for the contact information displayed within the table-->
78 ▼ <table id="contact_tbl">
79 ▼ <tr>
80 ▼ <thead>
81  <th>Name</th>
82  </thead></tr>
83  <tbody></tbody>
84 </table>
85
86
87
88
89
90 ▼ <footer>
91  <h2>&copy; Fast Addresses INC 2019</h2>
92  <div class="footerbox">
93  </div>
94 </footer>
95 </body>
96
97
98 ▼ <script>
99

```

Appendix F: CSS

```
1 ▼ html, body {
2     background-color: #9db5eb;
3     height: 100%;
4     font-family: 'Avenir', sans-serif;
5 }
6
7 ▼ header {
8     height: 70%;
9     max-height: 300px;
10    width: 100%;
11    position: relative;
12    overflow: hidden;
13 }
14
15 ▼ .curvedbanner {
16     background-color: #093394;
17     background-size: cover;
18     background-attachment: fixed;
19     position: absolute;
20     top: 0px;
21     left: -30%;
22     right: -30%;
23     bottom: 0px;
24     border-bottom-right-radius: 50%;
25     border-bottom-left-radius: 50%;
26     z-index: 1;
27 }
28
29 ▼ h1 {
30     position: absolute;
31     top: 65%;
32     left: 50%;
33     -webkit-transform: translate(-50%, -50%);
34     transform: translate(-50%, -50%);
35     margin: 0;
36     font-size: 3em;
```

```
29 ▼ h1 {
30     position: absolute;
31     top: 65%;
32     left: 50%;
33     -webkit-transform: translate(-50%, -50%);
34     transform: translate(-50%, -50%);
35     margin: 0;
36     font-size: 3em;
37     color: #000;
38     text-shadow: 1px 1px 1px #666;
39 }
40
41 ▼ .logo {
42     position: absolute;
43     top: 50%;
44     left: 50%;
45     -webkit-transform: translate(-50%, -50%);
46     transform: translate(-50%, -50%);
47     margin: 0;
48     font-size: 3em;
49     z-index: 1;
50 }
51
52 ▼ h2 {
53     position: relative;
54     color: #bdb411;
55     text-align: center;
56     font-family: 'Avenir', sans-serif;
57     font-size: .9em;
58     z-index: 1;
59 }
60
61 ▼ footer {
62     height: 70%;
63     top: 100%;
64     max-height: 40px;
```

```
65     width: 100%;  
66     position: relative;  
67     overflow: hidden;  
68   }  
69  
70 ▼ .footerbox {  
71     background-color: #093394;  
72     background-size: cover;  
73     position: absolute;  
74     top: 0px;  
75     left: -30%;  
76     right: -30%;  
77     bottom: 0px;  
78   }  
79  
80 /*Guide*/  
81 ▼ h3 {  
82     font-size: 25px;  
83     font-weight: 100;  
84     color: #2C2C2C;  
85     margin: 20px 25px;  
86   }  
87  
88 ▼ .open {  
89     position: absolute;  
90     width: 100px;  
91     height: 40px;  
92     left: 90%;  
93     top: 300px;  
94     margin-left: -80px;  
95     margin-top: -10px;  
96     border: 2px solid #bdb411;  
97     background-color: #093394;  
98     border-radius: 6px;  
99     padding: 10px 30px;  
100    color: #bdb411;
```

```
101    -webkit-transition: all ease-out 0.6s;
102    transition: all ease-out 0.6s;
103 }
104
105 ▼ .open:hover {
106    border: 2px solid #aaa;
107    -webkit-box-shadow: 0 0 8px #ccc inset;
108        box-shadow: 0 0 8px #ccc inset;
109    -webkit-transition: all ease-out 0.6s;
110        transition: all ease-out 0.6s;
111 }
112
113 ▼ .con {
114    position: fixed;
115    width: 400px;
116    height: 238px;
117    left: 50%;
118    top: 50%;
119    margin-top: -119px;
120    margin-left: -200px;
121    background-color: #F3F3F3;
122    border-radius: 6px;
123    -webkit-box-shadow: 0px 0px 24px rgba(0, 0, 0, 0.4);
124        box-shadow: 0px 0px 24px rgba(0, 0, 0, 0.4);
125    z-index: 1;
126 }
127
128 ▼ .con:before {
129    content: '';
130    position: absolute;
131    left: -14px;
132    top: 28px;
133    border-style: solid;
134    border-width: 10px 14px 10px 0;
135    border-color: rgba(0, 0, 0, 0) #F3F3F3 rgba(0, 0, 0, 0) rgba(0, 0, 0, 0);
136 }
```

```
137
138 ▼ .con p {
139     width: 350px;
140     font-size: 16px;
141     color: #8b8d94;
142     font-weight: 400;
143     line-height: 28px;
144     float: left;
145     margin: 0;
146 }
147
148 ▼ .con .bottom {
149     display: -webkit-box;
150     display: -ms-flexbox;
151     display: flex;
152     width: 100%;
153     bottom: 0;
154     position: absolute;
155 }
156
157 ▼ .con .bottom .step {
158     -webkit-box-flex: 3;
159     flex: 3;
160     -webkit-flex: 3;
161     -ms-flex: 3;
162     width: 100%;
163     height: 54px;
164     background-color: #373942;
165     border-bottom-left-radius: 6px;
166     display: -webkit-box;
167     display: -ms-flexbox;
168     display: flex;
169 }
170
171 ▼ .con .bottom .step span {
```

```
172     -webkit-box-flex: 1;
173         flex: 1;
174     -webkit-flex: 1;
175     -ms-flex: 1;
176     line-height: 54px;
177     color: #fff;
178     margin-left: 25px;
179     font-size: 18px;
180 }
181
182 ▼ .con .bottom .step ul {
183     -webkit-box-flex: 2;
184         flex: 2;
185     -webkit-flex: 2;
186     -ms-flex: 2;
187     list-style: none;
188     height: 10px;
189     margin: 23px 0;
190     padding-left: 15px;
191 }
192
193 ▼ .con .bottom .step ul li {
194     position: relative;
195     height: 7px;
196     width: 7px;
197     margin: 0 10px;
198     float: left;
199     border-radius: 50%;
200     background: none;
201     border: 1px solid #535560;
202 }
203
204 ▼ .con .bottom .step ul li:first-child:before {
205     width: 0;
206 }
207
```

```
208 ▼ .con .bottom .step ul li:before {
209     content: '';
210     position: absolute;
211     width: 20px;
212     border-top: 1px solid #535560;
213     left: -21px;
214     top: 3px;
215 }
216
217 ▼ .con .bottom .step ul li.true {
218     background-color: #7a7d86;
219 }
220
221 ▼ .con .bottom .step ul li.active {
222     background-color: #fff;
223     -webkit-box-shadow: 0 0 6px rgba(255, 255, 255, 0.78);
224             box-shadow: 0 0 6px rgba(255, 255, 255, 0.78);
225 }
226
227 ▼ .close {
228     cursor: pointer;
229 }
230
231 ▼ .close:before, .close:after {
232     content: "";
233     position: absolute;
234     height: 13px;
235     width: 13px;
236     top: 36px;
237     right: 31px;
238     border-top: 2px solid #7c7c7c;
239     -webkit-transform: rotate(-45deg);
240     transform: rotate(-45deg);
241 }
242
243 ▼ .close:before {
```

```
244     right: 40px;
245     -webkit-transform: rotate(45deg);
246     transform: rotate(45deg);
247   }
248
249 ▼ .btnnxt {
250   -webkit-box-flex: 1;
251   -ms-flex: 1;
252   flex: 1;
253   background-color: #6cb5f3;
254   border: 0;
255   margin: 0;
256   padding: 0;
257   text-transform: uppercase;
258   color: #fff;
259   font-weight: bold;
260   border-bottom-right-radius: 6px;
261   cursor: pointer;
262   -webkit-transition: all 0.3s;
263   transition: all 0.3s;
264 }
265
266 ▼ .btnnxt:hover {
267   background-color: #6BA5D6;
268   -webkit-transition: all 0.3s;
269   transition: all 0.3s;
270 }
271
272 ▼ btnnxt:active {
273   background-color: #5F8AAF;
274 }
275
276 ▼ .slider-con {
277   width: 350px;
278   margin: 0 25px;
```

```
279   overflow: hidden;
280 }
281
282 ▼ .slider-turn {
283   width: 1000px;
284 }
285
286 /*Search Table */
287 /*Table which will display the contact details from the json file*/
288 ▼ table {
289   width: 50%;
290   position: absolute;
291   top: 56%;
292   left: 25%;
293 }
294
295 ▼ th {
296   background: #f1f1f1;
297   font-weight: bold;
298   padding: 6px;
299   border: 3px solid #bdb411;
300   cursor: pointer;
301 }
302
303 ▼ td {
304   background: #dcefff;
305   padding: 6px;
306 }
307
308 ▼ td:hover {
309   background: #adc3f7;
310 }
311
312 /*Search bar to allow active searching for the contacts within the address book*/
313 ▼ .searchbar {
```

```
314     width: 100%;  
315     position: relative;  
316     display: -webkit-box;  
317     display: -ms-flexbox;  
318     display: flex;  
319 }  
320  
321 ▼ #search {  
322     width: 100%;  
323     border: 3px solid #093394;  
324     border-right: none;  
325     padding: 5px;  
326     height: 20px;  
327     border-radius: 5px 0 0 5px;  
328     outline: none;  
329     color: #9DBFAF;  
330 }  
331  
332 ▼ #search:focus {  
333     color: #093394;  
334 }  
335  
336 ▼ .search_btn {  
337     width: 40px;  
338     height: 36px;  
339     border: 1px solid #093394;  
340     background: #093394;  
341     text-align: center;  
342     color: #fff;  
343     border-radius: 0 5px 5px 0;  
344     cursor: pointer;  
345     font-size: 20px;  
346 }  
347  
348 ▼ .search_ctnt {  
349     width: 30%;
```

```
350     position: absolute;
351     top: 50%;
352     left: 50%;
353     -webkit-transform: translate(-50%, -50%);
354     transform: translate(-50%, -50%);
355   }
356
357 ▼ #result {
358   position: absolute;
359   left: 50%;
360   top: 100%;
361 }
362
363 ▼ #count {
364   position: absolute;
365   left: 100%;
366   top: 8%;
367 }
368
369 ▼ #details {
370   position: absolute;
371   left: 60%;
372   top: 50%;
373   width: 100px;
374   height: 200px;
375 }
```

Appendix G: jQuery/JavaScript

```
1 //jQuery code will activate once the document has successfully loaded.
2 $(document).ready(function(){
3     //ajax call for the data.json file this will create a table for the length of the data which has been created.
4     //This will generate the table data which would then be output onto the screen. This means that the user would be
5     //able to see the names displayed within the online address book. the loop will execute until it gets to the
6     //maximum number within the json file and then outputted within the table on the page.
7     $.ajax({
8         url: 'data.json',
9         dataType: 'json',
10        success: function(data){
11            for(var i=0; i<data.length; i++){
12                var row = $('<tr><td class = "contact" title = "name">' + data[i].name + '</td></tr>');
13                $('#contact_tbl').append(row);
14            }
15        },
16        error: function(jqXHR, textStatus, error){
17            alert('Error:Please connect to server to display the json data');
18        }
19    });
20    //code to sort the table alphabetically. This will allow the user to see the details arranged within the page.
21    //This will allow the user to sort the contact information which loads to be in alphabetical order when they click
22    //the heading of the table.
23    $('th').click(function(){
24        //variable which assigns the table to the function so it knows what to use
25        var tbl = $(this).parents('table').eq(0);
26        //variable for contact data in the table. Will be used to sort the information within the table. will then
27        //sort the table using the compare function which has been defined to compare sort the data
28        var contact_data=tbl.find('tr:gt(0)').toArray().sort(compare($(this).index()));
29        //this will run a for loop as long as the data length is run and will sort the data the length of the table.
30        for (var i = 0; i < contact_data.length; i++){
31            tbl.append(contact_data [i]);
32        }
33    })
34}
```

```
30    //compare function which will evaluate the data in the table and will sort it into alphabetical order so that
31    //the table would have been sorted into alphabetical order.
32    function compare(index) {
33        //internal function which will activate when the compare function has been called. This will ensure that the
34        //data has been correctly sorted so that it will be displayed in alphabetical order.
35        return function(a, b) {
36            //variable within the function which will use to compare the data which are A and B which are
37            //sensible variables to use which will mean that the data within the table will be sorted. this will take the
38            //parameter index from the compare variable so that it will understand which value has been assigned and the
39            //variable A will take in a from the function which will be returned when this is activated and the B variable
40            //will take in the b parameter.
41            var A = ContactName(a, index), B = ContactName(b, index);
42            //will change the A variable and the B variable to be numeric so that the value can be compared and changed
43            //if B is less than A and then convert back to string and compare to ensure that the data has been arranged to
44            //be alphabetical.
45            return $.isNumeric(A) && $.isNumeric(B) ? A - B : A.toString().localeCompare(B);
46        }
47    }
48    //function getCellValue will return the value for the row of the table.
49    function ContactName(row, index){
50        //will return the row child as the element which makes up the row of the table and adds this to the index as
51        //text so that it will be loaded into the compare function so that the table could successfully sort the data
52        //within the table.
53        return $(row).children('td').eq(index).text();
54    }
55
56    //live search the table to allow the user to be able to locate the person that they would like to access
57    //information on easily within the address book which has been developed.
58    $('#search').on('keyup', function(){
59        //search will activate on the key up function which means that as the user enters keys the information should
60        //refine on what has been displayed.
61        var search = $(this).val().replace('[A-Z a-z 0-9]');
62        //looks at the table and each row of the table to ensure that the data within the table will be refined when
63        //the search is being carried out.
64        $("table tr").each(function(index){
65            //conditional statement will run as long as the index is not equal to zero as this is the heading of the
66            //table.
67            if(index != 0){
68                //assigns the variable name using the this selector which identifies that the name would be used to
69                //show or hide as it would be required.
70                var name = $(this);
71                //will find the name from the td elements within table which is what they are contained within on the
72                //page.
73            }
74        })
75    })
76}
```

```

57 ▼    page;
58      name.find('td').each(function(){
59          //reference variable which will be assigned the text name of the contact within the table that is
60          //being searched for.
61          var ref = $(this).text();
62          //conditional statement which is used to use the ref variable and .indexof which uses the search
63          //variable value to check if the number is not the one being search for and this will mean that the
64          //unrequired names will be hidden.
65          if(ref.indexOf(search)<0){
66              name.hide();
67          }
68          //else statement which will display the name which is being searched for. This will mean that the
69          //names will be displayed and refined as the search is carried out.
70          else{
71              name.show();
72          }
73      });
74      //will count the number of results visible on the screen and will display a number to the user. Allows them to
75      //see the number of results given.
76      var rowCount = $('#contact_tbl>tbody>tr>td:visible').length;
77      document.getElementById('count' ).innerHTML=rowCount;
78  });
79
80  //hover on the table to reveal more contact details using a jquery ajax call.
81  $('#contact_tbl>tbody>tr>td').click(function(event) {
82      $.get("data.json")
83          .done(function(e) {
84              $(e).appendTo('textarea');
85              console.log("Success");
86          })
87          .fail(function() {
88              alert("Unforunately we are unable to process your request at this time." + "<br />" + "Please ensure you are
89              connected to a server and try again");
90      });
91  });
92  $('#postData').click(function(){
93      newKey = $('#key').val();
94      newValue = $('#value').val();
95
96      $.post('/postData',
97          {
98              key: newKey,
99              value: newValue
100         },function(data){
101             console.log('Data successfully posted -' + newKey + '!' + newValue);
102             alert('Data Posted');
103         });
104     });
105

```

```

91 ▼  $('#postData').click(function(){
92     newKey = $('#key').val();
93     newValue = $('#value').val();
94     $.post('/postData',
95         {
96             key: newKey,
97             value: newValue
98         },function(data){
99             console.log('Data successfully posted -' + newKey + '!' + newValue);
100            alert('Data Posted');
101        });
102
103    });
104
105

```

```

100
101    $(document).ready(function () {
102        var len = $('.con p').length;
103        var wid = parseInt($('.con p').css("width"));
104        var max = (len - 1) * wid;
105        $("ul li[data-num='1']").addClass('active');
106        $('.step span').html('Step 1');
107
108    $('body').on('click', '.bttnxt', function(){
109        var marL = parseInt($('.slider-turn').css('margin-left'));
110        var modulo = marL%wid;
111        if (-marL < max && modulo == 0) {
112            marL -= wid;
113
114            $('.slider-turn').animate({
115                'margin-left':marL
116            },300);
117            $('ul li.active').addClass('true').removeClass('active');
118            var x = -marL/wid +1;
119            $('ul li[data-num="'+x+'"]').addClass('active');
120            $('.step span').html("Step "+x);
121        }
122        else {}
123    });
124
125    $('body').on('click', '.close',function(){
126        $('.con').animate({
127            'opacity':0
128        },600);
129        $('.con').animate({
130            'top':-1200
131        }, {
132            duration: 2300,
133            queue: false
134        });
135        $('.open').animate({
136            'top':'50%'
137        });
138    });
139

```

```

139
140    $('body').on('click', '.open', function() {
141        $('.open').animate({
142            'top': -1000
143        });
144        $('.con').animate({
145            'opacity': 1
146        }, 400);
147        $('.con').animate({
148            'top': '50%'
149        }, {
150            duration: 800,
151            queue: false
152        });
153    });
154 });
155
156
157
158 //jQuery code will activate once the document has successfully loaded.
159 $(document).ready(function(){
160     //ajax call for the data.json file this will create a table for the length of the data which has been created. This
161     //will generate the table data which would then be output onto the screen. This means that the user would be able to
162     //see the names displayed within the online address book. the loop will execute until it gets to the maximum number
163     //within the json file and then outputted within the table on the page.
164
165     var hover_data = '';
166     var contact_data = [];
167     $.getJSON( "data.json")
168     .done(function(json) {
169         $.each(json, function(key, value){
170             contact_data.push("<tr>");
171             contact_data.push("<td id ='" + key + "' class = 'contact' title = 'name'>" + value.name +"</td>");
172         });
173
174     .fail(function( jqxhr, textStatus, error ) {
175         alert("Connection unsuccessful. Please ensure you are connected to a server before trying again")
176     });
177     //hover to display the details for each person

```

```

179 ▼ $( function() {
180     $(document).tooltip();
181 });
182
183
184 ▼ $.get("data.json", function(json) {
185     $.each(json, function(key, value){
186         hover_data += value.phone;
187         hover_data += value.email;
188         hover_data += value.job;
189         hover_data += value.streetName;
190         hover_data += value.city;
191         hover_data += value.zipCode;
192     ▼ for(var i; i<hover_data.length; i++){
193         var number = hover_data[i];
194         console.log(number);
195     };
196 });
197 ▼ $(document).on('mouseover','td',function(){
198     var current = $(this).attr("id");
199     console.log(current);
200     $(".contact").attr("title", hover_data);
201 });
202 });
203
204
205 //code to sort the table alphabetically. This will allow the user to see the details arranged within the page. This
206 //will allow the user to sort the contact information which loads to be in alphabetical order when they click the
207 //heading of the table.
208 ▼ $('#th').click(function(){
209     //variable which assigns the table to the function so it knows what to use
210     var tbl = $(this).parents('table').eq(0);
211     //variable for contact data in the table. Will be used to sort the information within the table. will then sort
212     //the table using the compare function which has been defined to compare sort the data
213     var contact_data=tbl.find('tr>td').toArray().sort(compare($(this).index()));
214     //this will run a for loop as long as the data length is run and will sort the data the length of the table.
215     ▼ for (var i = 0; i < contact_data.length; i++){
216         tbl.append(contact_data [i]);
217     }
218 });
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243

```

```

244 ▼      name.find('td').each(function(){
245          //reference variable which will be assigned the text name of the contact within the table that is
246          //being searched for.
247          var ref = $(this).text();
248          //conditional statement which is used to use the ref variable and .indexof which uses the search
249          //variable value to check if the number is not the one being search for and this will mean that the
250          //unrequired names will be hidden.
251          if(ref.indexOf(search)<0){
252              name.hide();
253          }
254          //else statement which will display the name which is being searched for. This will mean that the
255          //names will be displayed and refined as the search is carried out.
256          else{
257              name.show();
258          }
259      });
260      //will count the number of results visible on the screen and will display a number to the user. Allows them to see
261      //the number of results given.
262      var rowCount = $('#contact_tbl>tbody>tr>td:visible').length;
263      document.getElementById('count' ).innerHTML=rowCount;
264      //closes all the scripting
265  });
266
267
268 </script>
269
270 </html>

```

```

2 //variables which will require the node faker and fs modules.
3 var faker = require('faker');
4 var fs = require('fs');
5 //array which will contain the relevant information.
6 var dataArray = [];
7
8 //for loop which will run 15 times until all the necessary data has been generated.
9 ▼ for(var i = 0; i < 20; i++){
10     var data = {};
11     data.image = faker.image.avatar();
12     data.name = faker.name.findName();
13     data.phone = faker.phone.phoneNumber();
14     data.email = faker.internet.email();
15     data.job = faker.name.jobTitle();
16     data.streetName = faker.address.streetName();
17     data.city = faker.address.city();
18     data.zipCode = faker.address.zipCode();
19     dataArray.push(data);
20 }
21 //will write the data which is stored within the dataArray to a json file called data.json.
22 ▼ fs.writeFile('data.json', JSON.stringify(dataArray), (err) =>{
23     //conditional statement which checks for errors and if there is so alerts the user that it could not be located
24     //and if not then will tell them the file has successfully been generated.
25     if(err){
26         err += '404 page not found';
27         console.log(err);
28     }
29     console.log('the json file has successfully be generation');
}

```

```
1 //require gulp module
2 var gulp = require('gulp');
3 //require the sass module.
4 var sass = require('gulp-sass');
5 var concat = require('gulp-concat');
6
7 var postcss = require('gulp-postcss');
8 var autoprefixer = require('autoprefixer');
9 var cssnano = require('cssnano');
10 ▼ function style(){
11     return(gulp.src("assignment_2/css/.scss")
12         .pipe(sass())
13         .on("error", sass.logError)
14         .pipe(postcss([autoprefixer(), cssnano()])))
15         .pipe(concat('style.min.css'))
16         .pipe(gulp.dest("assignment2/css")));
17 }
18 exports.style = style;
```

Appendix H: Trello Activity

	<p>James Johnston moved <u>Hover Over a person and be presented with an overview</u> from In progress to Completed a minute ago</p>
	<p>James Johnston completed Code in Appendix on <u>Report Work</u> a minute ago</p>
	<p>James Johnston completed Main Content on <u>Report Work</u> 19 minutes ago</p>
	<p>James Johnston completed Responses on <u>James Video</u> yesterday at 22:30</p>
	<p>James Johnston completed Construct Json/FAKER on <u>James Video</u> yesterday at 18:16</p>
	<p>James Johnston moved <u>James Video</u> from Comments to In progress yesterday at 18:10</p>
	<p>malocca10 attached A2.zip to <u>HTML/CSS/Javascript</u> yesterday at 10:59</p>
	<p>James Johnston added Video items to <u>James Video</u> 10 Dec at 15:02</p>
	<p>James Johnston added Video Items to <u>Romano Video</u> 10 Dec at 15:01</p>
	<p>James Johnston added <u>James Video</u> to Comments 10 Dec at 15:01</p>
	<p>James Johnston added <u>Romano Video</u> to Comments 10 Dec at 15:01</p>
	<p>malocca10 attached A2.zip to <u>HTML/CSS/Javascript</u> 10 Dec at 13:46</p>
	<p>malocca10 moved <u>Usability Testing X3 people</u> from In progress to Completed 10 Dec at 13:32</p>

	<p>malocca10 completed Search on <u>Romano Video</u> yesterday at 16:32</p>
	<p>malocca10 completed Help Guide on <u>Romano Video</u> yesterday at 16:32</p>
	<p>malocca10 completed SASS/SCSS on <u>Romano Video</u> yesterday at 16:32</p>
	<p>malocca10 moved <u>Romano Video</u> from Comments to Tasks to Complete yesterday at 16:14</p>
	<p>malocca10 attached A2.zip to <u>HTML/CSS/Javascript</u> yesterday at 16:03</p>
	<p>malocca10 deleted the A2.zip attachment from <u>HTML/CSS/Javascript</u> yesterday at 16:03</p>
	<p>malocca10 completed User testing table on <u>Report Work</u> yesterday at 11:39</p>

	<p>James Johnston attached data.json to <u>data file</u> yesterday at 16:54</p>
	<p>James Johnston deleted the data.json attachment from <u>data file</u> yesterday at 16:54</p>
	<p>James Johnston attached data.json to <u>data file</u> yesterday at 16:54</p>
	<p>James Johnston attached data.json to <u>data file</u> yesterday at 16:48</p>
	<p>James Johnston added <u>data file</u> to Comments yesterday at 16:48</p>
	<p>malocca10 moved <u>Romano Video</u> from In progress to Completed yesterday at 16:33</p>
	<p>malocca10 moved <u>Romano Video</u> from Tasks to Complete to In progress yesterday at 16:33</p>
	<p>malocca10 completed User Testing on <u>Romano Video</u> yesterday at 16:33</p>

 **James Johnston** attached [Questionnaire.docx](#) to [Usability Testing X3 people](#) 10 Dec at 13:31

 **James Johnston** attached [index.html](#) to [Up to date hover](#) 10 Dec at 11:59

 **James Johnston** added [Up to date hover](#) to Comments 10 Dec at 11:59

 **James Johnston** attached [index.html](#) to [Hover Tooltip all data](#) 9 Dec at 15:26

 **James Johnston** added [Hover Tooltip all data](#) to Comments 9 Dec at 15:26

 **James Johnston** on [General Comments](#) 6 Dec at 13:58

Looking at creating a variable and loading it to the hover

 **James Johnston** moved [Usability Testing X3 people](#) from Tasks to Complete to In progress 4 Dec at 17:33

 **malocca10** attached [Questionnaire.docx](#) to [Usability Testing X3 people](#) 4 Dec at 17:31

 **James Johnston** attached [index.html](#) to [Inprogress hover tooltip](#) 4 Dec at 17:20

 **James Johnston** on [General Comments](#) 4 Dec at 13:06

Working on the hover

 **James Johnston** on [General Comments](#) 3 Dec at 15:52

Hover is beginning to get there

 **James Johnston** attached [index.html](#) to [Inprogress hover tooltip](#) 3 Dec at 15:51

 **James Johnston** added [Inprogress hover tooltip](#) to Comments 3 Dec at 15:51

 **James Johnston** on [General Comments](#) 2 Dec at 17:39

Looking at the hover at the moment

 **malocca10** on [General Comments](#) 1 Dec at 21:49

i uploaded a file with the css and html with the progress that we have achieved so far if you wouldnt mind checking if that is ok

 **malocca10** attached [A2.zip](#) to [HTML/CSS/JS](#) 1 Dec at 21:48

 **malocca10** moved [HTML/CSS/JS](#) from Tasks to Complete to In progress 1 Dec at 21:47

 **James Johnston** on [General Comments](#) 1 Dec at 18:33

Taking a look at the hover yet again

 **James Johnston** on [General Comments](#)
30 Nov at 15:36

Still looking into the html hover effect.
Still not getting anywhere with it

 **malocca10** on [Hover Over a person and be presented with an overview](#) 30 Nov at 10:56

could this help:
<https://stackoverflow.com/questions/40109087/insert-json-data-to-a-hover-div-matching-an-employee>

 **malocca10** on [General Comments](#)
29 Nov at 18:00

still looking into click to show more relative information

 **James Johnston** on [General Comments](#)
29 Nov at 10:18

Still looking at the hover just cant get anywhere which is annoying

 **James Johnston** attached [list.html](#) to [General Comments](#) 28 Nov at 14:11

 **James Johnston** on [General Comments](#)
28 Nov at 13:14

Looking into the hover and possibly changing it to be on a list

 **James Johnston** on [General Comments](#)
27 Nov at 20:37

Still not getting that far with the hover.
Will look at it again fresh in the morning

 **malocca10** attached [index3.html](#) to [General Comments](#) 27 Nov at 20:15

 **James Johnston** attached [index.html](#) to [General Comments](#) 27 Nov at 14:35

 **James Johnston** deleted the [index.html](#) attachment from [General Comments](#)
27 Nov at 14:35

 **malocca10** moved [Clicking on the item Should display result in the provision of more detailed description that is only loaded on this request](#) from Tasks to Complete to In progress
27 Nov at 14:02

 **James Johnston** attached [index.html](#) to [General Comments](#) 27 Nov at 13:52

 **James Johnston** deleted the [index.html](#) attachment from [General Comments](#)
27 Nov at 13:52

 **James Johnston** attached [index.html](#) to [General Comments](#) 27 Nov at 13:22

 **James Johnston** deleted the [index.html](#) attachment from [General Comments](#)
27 Nov at 13:21

 **James Johnston** attached [index.html](#) to [General Comments](#) 27 Nov at 13:14

 **James Johnston** deleted the [index.html](#) attachment from [General Comments](#)
27 Nov at 13:11

 **malocca10** moved [Access Help on how to perform a search](#) from In progress to Completed
27 Nov at 13:01

 **James Johnston** attached [index.html](#) to [General Comments](#) 27 Nov at 12:52

 **James Johnston** deleted the [index.html](#) attachment from [General Comments](#)
27 Nov at 12:52

 **James Johnston** on [General Comments](#)
27 Nov at 12:52

td class name is .contact for use within the hover function

 **James Johnston** attached [index.html](#) to [General Comments](#) 27 Nov at 12:51

 **James Johnston** deleted the [index.html](#) attachment from [General Comments](#)
27 Nov at 12:51

 James Johnston attached index.html to [General Comments](#) 27 Nov at 12:35

 James Johnston attached [index2.html](#) to [General Comments](#) 27 Nov at 12:32

 James Johnston deleted the index2.html attachment from [General Comments](#) 27 Nov at 12:30

 James Johnston attached [server.js](#) to [General Comments](#) 27 Nov at 12:29

 malocca10 moved [Add Data to the json file](#) from Tasks to Complete to In progress 27 Nov at 12:27

 James Johnston attached index2.html to [General Comments](#) 27 Nov at 12:22

 James Johnston on [General Comments](#) 27 Nov at 12:01

Still working on the hover

 James Johnston attached index.html to [General Comments](#) 27 Nov at 12:35

 James Johnston attached [index2.html](#) to [General Comments](#) 27 Nov at 12:32

 James Johnston deleted the index2.html attachment from [General Comments](#) 27 Nov at 12:30

 James Johnston attached [server.js](#) to [General Comments](#) 27 Nov at 12:29

 malocca10 moved [Add Data to the json file](#) from Tasks to Complete to In progress 27 Nov at 12:27

 James Johnston attached index2.html to [General Comments](#) 27 Nov at 12:22

 James Johnston on [General Comments](#) 27 Nov at 12:01

Still working on the hover

 malocca10 on [Hover Over a person and be presented with an overview](#) 26 Nov at 20:00

This may help.
<https://stackoverflow.com/questions/25464562/load-json-data-into-a-bootstrap-modal>

 James Johnston on [General Comments](#) 26 Nov at 19:50

Still looking into the modal and also looking at the delete data from the json file part of the assignment work

 James Johnston on [General Comments](#) 26 Nov at 15:34

Taking a break from the hover and looking at the delete data from the json file

 James Johnston on [General Comments](#) 26 Nov at 10:49

Looking into the hover function again today

 James Johnston on [General Comments](#) 24 Nov at 21:28

will get another look into the hover again tomorrow still not getting anywhere with it

 James Johnston on [General Comments](#) 24 Nov at 19:27

It not the hover thats the problem its the getting a modal to display json data on hover

 malocca10 on [General Comments](#) 24 Nov at 15:49

if your still having trouble with the hover that is



malocca10 attached [tablehover.html](#) to [General Comments](#) 24 Nov at 15:46



James Johnston on [General Comments](#) 23 Nov at 14:25

Having a look to see if the pop up modal could be used on hover



James Johnston on [General Comments](#) 21 Nov at 15:16

Still looking into this at the moment



James Johnston on [General Comments](#) 21 Nov at 12:14

Report updated with the trello detail added.

Back to looking at the hover to display more detail part now



James Johnston on [General Comments](#) 21 Nov at 11:53

Added into the report about the way we are using trello to help with the organisation of the tasks



James Johnston on [General Comments](#) 20 Nov at 18:41

Will get looking at the hover again tomorrow night



James Johnston on [General Comments](#) 20 Nov at 15:01

looking into the tooltip-hover to see more detail from the json data file



James Johnston on [Hover Over a person and be presented with an overview](#)

19 Nov at 13:27

This Link may help with the development of this.
<https://stackoverflow.com/questions/45741102/populate-tooltip-with-json-based-on-element-data-tooltip-value>



James Johnston on [Clicking on the item Should display result in the provision of more detailed description that is only loaded on this request](#) 19 Nov at 12:55

This might help with this part
<https://codepen.io/kyledws/pen/osify>



malocca10 on [General Comments](#) 19 Nov at 10:30

i had updated the report to include the numbered search results



James Johnston joined [Delete Data from the json file](#)

19 Nov at 10:23



James Johnston added [malocca10 to Add Data to the json file](#)

19 Nov at 10:22



James Johnston added [Delete Data from the json file to Tasks to Complete](#)

19 Nov at 10:22



James Johnston added [Add Data to the json file to Tasks to Complete](#)

19 Nov at 10:22

 **James Johnston** attached [Screenshot 2019-11-19 at 10.19.08.png](#) to [Installing express and body_parser](#) 19 Nov at 10:19

```
James-MacBook-Air:Documents JamesJohnston$ npm install express --g
npm WARN saveError Missing write access to /usr/local/lib/node_modules
code EACCES
 errno -38
 syscall access
 path /usr/local/lib/node_modules
fstat: EACCES: permission denied, access '/usr/local/lib/node_modules'
fstat: EACCES: permission denied, access '/usr/local/lib/node_modules' { 
  errno: -38,
  code: 'EACCES',
  syscall: 'access',
  path: '/usr/local/lib/node_modules'
}

The operation was rejected by your operating system.
It is likely you do not have the permissions to access this file as the current user.
If you believe this might be a permissions issue, please double-check the
permissions of the file and its containing directories, or try running
the command again as root/Administrator.

A complete log of this run can be found in:
  /Users/jamesjohnston/.npm/_logs/2019-11-19T20_37_04_969Z-debug.log
James-MacBook-Air:Documents JamesJohnston$ rm install express
rm: node-test@0.0: No such file or directory.

+ express@4.17.1
added 58 packages from 37 contributors and audited 138 packages in 0.913s
Found 0 vulnerabilities
James-MacBook-Air:Documents JamesJohnston$ rm install body-parser
rm: node-test@0.0: No such file or directory.

+ body-parser@0.0.0
updated 1 package and audited 138 packages in 1.963s
Found 0 vulnerabilities
James-MacBook-Air:Documents JamesJohnston$
```

James Johnston added [Installing express and body_parser](#) to [Assignment Brief](#) 19 Nov at 10:19

 **James Johnston** added [Add to and delete from json](#) to [Assignment Brief](#) 19 Nov at 10:06

 **James Johnston** on [General Comments](#) 18 Nov at 19:08

@malocca10

 **malocca10** deleted the index.html attachment from [Search and Retrieve Information](#) 18 Nov at 18:50

 **malocca10** moved [General Comments](#) from [Assignment Brief](#) to [Comments](#) 18 Nov at 18:47

 **malocca10** added [Comments](#) to this board 18 Nov at 18:47

 **malocca10** moved [Search and Retrieve Information](#) from [In progress](#) to [Completed](#) 18 Nov at 18:47

 **malocca10** on [General Comments](#) 18 Nov at 18:47

i believe that is the numbered results done

 **malocca10** completed Display number of results on [Search and Retrieve Information](#) 18 Nov at 18:46

 **malocca10** attached [index.html](#) to [Search and Retrieve Information](#) 18 Nov at 18:46

 **malocca10** added Tasks to [HTML/CSS/JS](#) 18 Nov at 14:03

 **malocca10** added [James Johnston](#) to [HTML/CSS/JS](#) 18 Nov at 14:03

 **malocca10** joined [HTML/CSS/JS](#) 18 Nov at 14:03

 **malocca10** added [HTML/CSS/JS](#) to Tasks to Complete 18 Nov at 14:02

 **James Johnston** on [General Comments](#) 17 Nov at 16:24

Search information now added to the report

 **James Johnston** on [General Comments](#) 17 Nov at 15:05

Adding the item on search to the final report work

 **James Johnston** on [General Comments](#) 17 Nov at 09:47

Next task is to work on the hover for the table to display more detail on the contacts as the user hovers over each of the contacts within the address book

 **malocca10** on [General Comments](#) 16 Nov at 15:02

Looking into the number of results on the search

-  **James Johnston** attached index.html to [Search and Retrieve Information](#)
16 Nov at 15:02

 **James Johnston** on [General Comments](#)
16 Nov at 14:33

Faker.js File is now commented and uploaded

 **James Johnston** attached fakerfile.js to [Have an Easily Accessible and alphabetically sorted list](#) 16 Nov at 14:33

 **James Johnston** on [General Comments](#)
16 Nov at 14:03

Searching the table now operational and allows the user to be able to search for the names of those easily within the page

 **James Johnston** completed Refine the results as it is carried out on [Search and Retrieve Information](#)
16 Nov at 13:51

 **James Johnston** completed Retrieve and display Json Data in a table on [Search and Retrieve Information](#)
16 Nov at 13:51

 **James Johnston** added malocca10 to [Search and Retrieve Information](#)
16 Nov at 13:42

 **James Johnston** moved [Search and Retrieve Information](#) from Tasks to Complete to In progress
16 Nov at 13:41

 **James Johnston** on [General Comments](#)
16 Nov at 13:41

Looking into the search functionality

 **James Johnston** on [General Comments](#)
16 Nov at 11:40

Planning today to have a look into the hover and the search
-  **James Johnston** added [General Comments](#) to Assignment Brief
16 Nov at 11:40

 **James Johnston** on [Have an Easily Accessible and alphabetically sorted list](#)
14 Nov at 14:06

Report is updated to include the alphabetically sorted information

 **James Johnston** moved [Hover Over a person and be presented with an overview](#) from Tasks to Complete to In progress
14 Nov at 12:29

 **James Johnston** moved [Search and Retrieve Information](#) from In progress to Tasks to Complete
14 Nov at 12:29

 **James Johnston** moved [Search and Retrieve Information](#) from Tasks to Complete to In progress
14 Nov at 11:01

 **James Johnston** deleted the index.html attachment from [Have an Easily Accessible and alphabetically sorted list](#)
14 Nov at 10:57

 **James Johnston** deleted the search.html attachment from [Have an Easily Accessible and alphabetically sorted list](#)
14 Nov at 10:56

 **James Johnston** attached index.html to [Have an Easily Accessible and alphabetically sorted list](#) 14 Nov at 10:56

 **James Johnston** deleted the index.html attachment from [Have an Easily Accessible and alphabetically sorted list](#)
14 Nov at 10:56

 **James Johnston** attached index.html to [Have an Easily Accessible and alphabetically sorted list](#) 14 Nov at 10:46

 **James Johnston** moved [Have an Easily Accessible and alphabetically sorted list](#) from In progress to Completed
14 Nov at 10:38

-  **James Johnston** on [Have an Easily Accessible and alphabetically sorted list](#)
14 Nov at 10:15

User can sort the table by clicking the th of the table and this will rearrange the data to be in alphabetical order. Allows interactivity within the table.
-  **James Johnston** on [Have an Easily Accessible and alphabetically sorted list](#)
13 Nov at 22:48

Will return to working on this tomorrow getting there with the alphabetic display
-  **James Johnston** on [Have an Easily Accessible and alphabetically sorted list](#)
13 Nov at 21:49

At the moment the table will be sorted alphabetically when the user clicks on it
-  **James Johnston** on [Have an Easily Accessible and alphabetically sorted list](#)
13 Nov at 11:10

Working on alphabetically sorting the table today
-  **James Johnston** on [Construct Json Based Data](#)
12 Nov at 13:54

<https://www.npmjs.com/package/faker>
Link which had been used for Faker Data to be generated to be displayed in the table
-  **James Johnston** on [Have an Easily Accessible and alphabetically sorted list](#)
12 Nov at 13:14

Finally got the data to display in the table
-  **James Johnston** on [Have an Easily Accessible and alphabetically sorted list](#)
13 Nov at 20:29

getting there with the alphabetically sorted list
-  **James Johnston** on [Have an Easily Accessible and alphabetically sorted list](#)
13 Nov at 17:12

currently looking to see how to sort the table
-  **James Johnston** on [Have an Easily Accessible and alphabetically sorted list](#)
13 Nov at 13:33

Still working through the alphabetical sorted table
-  **James Johnston** completed Table list of the Data from the Json File on [Have an Easily Accessible and alphabetically sorted list](#)
12 Nov at 13:14
-  **James Johnston** on [Have an Easily Accessible and alphabetically sorted list](#)
12 Nov at 12:57

Currently working on the display json data names in a table for the address book
-  **James Johnston** on [Have an Easily Accessible and alphabetically sorted list](#)
12 Nov at 12:57

<https://www.kodingmadesimple.com/2016/07/display-json-data-in-html-table-using-jquery-ajax.html> link which should help with the creation of the table to display the data

 malocca10 moved [Access Help on how to perform a search](#) from Tasks to Complete to In progress
12 Nov at 12:29

 James Johnston on [Construct Json Based Data](#) 12 Nov at 10:46

File used to generate the faker data has now also been uploaded

 James Johnston attached [fakerfile.js](#) to [Construct Json Based Data](#) 12 Nov at 10:46

 James Johnston on [Construct Json Based Data](#) 12 Nov at 10:36

Updated the json file through the use of node to generate it there this morning.

 James Johnston attached [data.json](#) to [Construct Json Based Data](#) 12 Nov at 10:35

 James Johnston attached [Screenshot 2019-11-11 at 21.06.01.png](#) to [Node.js](#) 11 Nov at 21:06

 James Johnston on [Code for json generation](#) 11 Nov at 20:59

<https://www.npmjs.com/package/faker>
Link for the Node.js Faker npm

 James Johnston on [Code for json generation](#) 11 Nov at 20:36

Working on producing the Javascript file which would generate the json file using faker through node.js

 James Johnston attached [Screenshot 2019-11-11 at 21.06.01.png](#) to [Node.js](#) 11 Nov at 21:06



 James Johnston added [Node.js](#) to [Assignment Brief](#) 11 Nov at 21:06

 James Johnston on [Code for json generation](#) 11 Nov at 21:03

Thats the javascript file created which will be used to generate the faker data using node.js
This will be completed in the morning

 James Johnston attached [fakerfile.js](#) to [Code for json generation](#) 11 Nov at 21:02

 James Johnston added [Code for json generation](#) to [Assignment Brief](#) 11 Nov at 18:07

 James Johnston on [Assistance with alphabetically sorting table](#) 11 Nov at 12:33

Plan for Monday 11th November 2019:
Try and finalise the sorting jquery code

 James Johnston on [Have an Easily Accessible and alphabetically sorted list](#) 10 Nov at 19:53

Currently looking into the sorting the table option at the moment have been unable to get it to work successfully so far

 James Johnston on [Assistance with alphabetically sorting table](#) 10 Nov at 19:52

Tutorial option for sorting the table:
<https://www.youtube.com/watch?v=Ne2t-Wp7rdQ>

 **James Johnston** on [Assistance with alphabetically sorting table](#) 10 Nov at 18:52

Still looking into this

 **James Johnston** on [Assistance with alphabetically sorting table](#) 10 Nov at 11:41

Looking in the Sorting table code
Sunday 10am start

 **James Johnston** on [Assistance with alphabetically sorting table](#) 8 Nov at 12:19

<https://www.geeksforgeeks.org/how-to-sort-a-list-alphabetically-using-jquery/>
another link which might help with this task

 **James Johnston** on [Assistance with alphabetically sorting table](#) 8 Nov at 12:03

Will have a look into it this weekend to help progress the assignment work further

 **malocca10** attached [table.html](#) to [Have an Easily Accessible and alphabetically sorted list](#) 6 Nov at 22:46

 **James Johnston** on [Have an Easily Accessible and alphabetically sorted list](#) 6 Nov at 12:59

Been working on this all morning and cant get it to display the table

 **James Johnston** deleted the myscript.js attachment from [Have an Easily Accessible and alphabetically sorted list](#) 6 Nov at 12:59

 **James Johnston** attached search.html to [Have an Easily Accessible and alphabetically sorted list](#) 6 Nov at 12:58

 **James Johnston** attached myscript.js to [Have an Easily Accessible and alphabetically sorted list](#) 6 Nov at 12:58

 **James Johnston** attached index.html to [Have an Easily Accessible and alphabetically sorted list](#) 6 Nov at 12:58

 **James Johnston** on [Assistance with alphabetically sorting table](#) 8 Nov at 12:03

this link may help with sorting the table alphabetically

 **James Johnston** added [Assistance with alphabetically sorting table](#) to Assignment Brief

8 Nov at 12:03

 **James Johnston** on [Have an Easily Accessible and alphabetically sorted list](#) 8 Nov at 12:00

Having this piece of work for the alphabetically sorted list will allow us to be able to include this when we figure out why the table is not being displayed.

 **James Johnston** on [Have an Easily Accessible and alphabetically sorted list](#) 8 Nov at 10:58

Plan this weekend look for alphabetically sorted jquery code

 **James Johnston** attached [contactinfo.json](#) to [Have an Easily Accessible and alphabetically sorted list](#) 6 Nov at 12:58

 **James Johnston** added Tasks to [Have an Easily Accessible and alphabetically sorted list](#)

4 Nov at 19:53

 **James Johnston** moved [Have an Easily Accessible and alphabetically sorted list](#) from Tasks to Complete to In progress

4 Nov at 19:52

 **James Johnston** added [Mockaroo Json Generator](#) to Assignment Brief

29 Oct at 14:43

 **James Johnston** changed the background of this board

29 Oct at 14:41

 **James Johnston** added Tasks to [Search and Retrieve Information](#)

29 Oct at 14:40

-  **James Johnston** moved [Report Work](#) from Tasks to Complete to In progress
29 Oct at 14:40
-  **malocca10** moved [Construct Json Based Data](#) from Tasks to Complete to Completed
29 Oct at 14:39
-  **malocca10** completed Construct JSON on [Construct Json Based Data](#)
29 Oct at 14:38
-  **malocca10** attached [contactinfo.json](#) to [Construct Json Based Data](#) 29 Oct at 14:38
-  **malocca10** added Task to [Construct Json Based Data](#)
29 Oct at 14:36
-  **James Johnston** moved [Background Research](#) from In progress to Completed
29 Oct at 10:27
-  **malocca10** completed Romano x1 Example on [Background Research](#)
29 Oct at 08:13

-  **James Johnston** completed James x2 examples on [Background Research](#)
28 Oct at 20:45
-  **James Johnston** completed Contents Page on [Report Work](#)
28 Oct at 20:45
-  **James Johnston** completed Cover Page on [Report Work](#)
28 Oct at 20:45
-  **James Johnston** added Tasks to [Background Research](#)
28 Oct at 16:56
-  **James Johnston** moved [Background Research](#) from Tasks to Complete to In progress
28 Oct at 16:55
-  **James Johnston** added **malocca10** to [Usability Testing X3 people](#)
28 Oct at 16:48
-  **James Johnston** joined [Usability Testing X3 people](#)
28 Oct at 16:48

-  **James Johnston** added **malocca10** to [Usability Testing X3 people](#)
28 Oct at 16:48
-  **James Johnston** joined [Usability Testing X3 people](#)
28 Oct at 16:48
-  **James Johnston** attached [COM601 assignment 2\(1\).pdf](#) to [Brief](#)
28 Oct at 16:44
-  **James Johnston** added [Brief](#) to [Assignment Brief](#)
28 Oct at 16:44
-  **James Johnston** added Assignment Brief to this board
28 Oct at 16:43
-  **James Johnston** added Tasks to complete to [Report Work](#)
28 Oct at 16:42
-  **James Johnston** added **malocca10** to [Report Work](#)
28 Oct at 16:42

-  **James Johnston** added **malocca10** to [Construct Json Based Data](#)
28 Oct at 16:41
-  **James Johnston** added **malocca10** to [View Additional Related people whereby content is accessed from JSON formatted using jQuery](#)
28 Oct at 16:41
-  **James Johnston** joined [Have an Easily Accessible and alphabetically sorted list](#)
28 Oct at 16:41
-  **James Johnston** joined [Search and Retrieve Information](#)
28 Oct at 16:41
-  **James Johnston** added **malocca10** to [Access Help on how to perform a search](#)
28 Oct at 16:41
-  **James Johnston** joined [Access Help on how to perform a search](#)
28 Oct at 16:41

-  **James Johnston** added [Construct Json Based Data](#) to Tasks to Complete
28 Oct at 16:38
-  **James Johnston** added [View Additional Related people whereby content is accessed from JSON](#) formatted using jQuery to Tasks to Complete
28 Oct at 16:38
-  **James Johnston** added [Have an Easily Accessible and alphabetically sorted list](#) to Tasks to Complete
28 Oct at 16:37
-  **James Johnston** added [Search and Retrieve Information](#) to Tasks to Complete
28 Oct at 16:37
-  **James Johnston** added [Access Help on how to perform a search](#) to Tasks to Complete
28 Oct at 16:36
-  **James Johnston** added Completed to this board
28 Oct at 16:36

-  **James Johnston** added [Access Help on how to perform a search](#) to Tasks to Complete
28 Oct at 16:36
-  **James Johnston** added Completed to this board
28 Oct at 16:36
-  **James Johnston** added In progress to this board
28 Oct at 16:36
-  **James Johnston** added Tasks to Complete to this board
28 Oct at 16:36
-  **James Johnston** added **malocca10** to this board
28 Oct at 16:36
-  **James Johnston** added this board to **J&R Addresses**
28 Oct at 16:35
-  **James Johnston** created this board
28 Oct at 16:35

-  **James Johnston** added **malocca10** to [Background Research](#)
28 Oct at 16:41
-  **James Johnston** joined [Background Research](#)
28 Oct at 16:41
-  **James Johnston** added [Usability Testing X3 people](#) to Tasks to Complete
28 Oct at 16:40
-  **James Johnston** added [Report Work](#) to Tasks to Complete
28 Oct at 16:40
-  **James Johnston** added [Background Research](#) to Tasks to Complete
28 Oct at 16:40
-  **James Johnston** added [Clicking on the item Should display result in the provision of more detailed description that is only loaded on this request](#) to Tasks to Complete
28 Oct at 16:39

Appendix I: Node.js & Gulp.js

```
James-MacBook-Air:Documents jamesjohnston$ npm install sass
npm WARN node-test@1.0.0 No repository field.

+ sass@1.23.7
added 1 package from 1 contributor and audited 8821 packages in 3.635s
found 0 vulnerabilities

James-MacBook-Air:Documents jamesjohnston$ npm install gulp-sass
> node-sass@4.13.0 install /Users/jamesjohnston/Documents/node_modules/node-sass
> node scripts/install.js

Downloading binary from https://github.com/sass/node-sass/releases/download/v4.13.0/darwin-x64-79_binding.node
Download complete! 🎉
Binary saved to /Users/jamesjohnston/Documents/node_modules/node-sass/vendor/darwin-x64-79_binding.node
Caching binary to /Users/jamesjohnston/.npm/node-sass/4.13.0/darwin-x64-79_binding.node
> node-sass@4.13.0 postinstall /Users/jamesjohnston/Documents/node_modules/node-sass
> node scripts/build.js

Binary found at /Users/jamesjohnston/Documents/node_modules/node-sass/vendor/darwin-x64-79_binding.node
Testing binary
Binary is fine
npm WARN node-test@1.0.0 No repository field.

+ gulp-sass@4.0.2
added 114 packages from 112 contributors and audited 9385 packages in 11.292s
found 0 vulnerabilities

James-MacBook-Air:Documents jamesjohnston$ npm install gulp-concat
npm WARN node-test@1.0.0 No repository field.

+ gulp-concat@2.6.1
added 3 packages from 38 contributors and audited 9417 packages in 8.867s
found 0 vulnerabilities

James-MacBook-Air:Documents jamesjohnston$
```

```
James-MacBook-Air:~ jamesjohnston$ pd
-bash: pd: command not found
James-MacBook-Air:~ jamesjohnston$ pwd
/Users/jamesjohnston
James-MacBook-Air:~ jamesjohnston$ cd /Users/jamesjohnston/Documents
James-MacBook-Air:Documents jamesjohnston$ npm install gulp
npm WARN deprecated fsevents@1.2.9: One of your dependencies needs to upgrade to fsevents v2: 1) Proper nodejs v10+ support 2) No more fetching binaries from AWS, smaller package size
> fsevents@1.2.9 install /Users/jamesjohnston/Documents/node_modules/fsevents
> node install

node-pre-gyp WARN Using needle for node-pre-gyp https download
node-pre-gyp WARN Tried to download(404): https://fsevents-binaries.s3-us-west-2.amazonaws.com/v1.2.9/fse-v1.2.9-node-v79-darwin-x64.tar.gz
node-pre-gyp WARN Pre-built binaries not found for fsevents@1.2.9 and node@13.0.1 (node-v79 ABI, unknown) (falling back to source compile with node-gyp)
  SOLINK_MODULE(target) Release/.node
  CXN(target) Release/obj.target/fse/fsevents.o
  SOLINK_MODULE(target) Release/fse.node
  COPY /Users/jamesjohnston/Documents/node_modules/fsevents/lib/binding/Release/node-v79-darwin-x64/fse.node
  TOUCH Release/obj.target/action_after_build.stamp
npm WARN node-test@1.0.0 No repository field.

+ gulp@4.0.2
added 388 packages from 238 contributors and audited 6673 packages in 21.211s
found 0 vulnerabilities

James-MacBook-Air:Documents jamesjohnston$ npm install sass
npm WARN node-test@1.0.0 No repository field.

+ sass@1.23.7
added 1 package from 1 contributor and audited 8821 packages in 3.635s
found 0 vulnerabilities
```

```
Last login: Mon Nov 11 18:07:44 on ttys000

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
[James-MacBook-Air:~ jamesjohnston$ npm install faker
npm WARN saveError ENOENT: no such file or directory, open '/Users/jamesjohnston/package.json'
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN enoent ENOENT: no such file or directory, open '/Users/jamesjohnston/package.json'
npm WARN jamesjohnston No description
npm WARN jamesjohnston No repository field.
npm WARN jamesjohnston No README data
npm WARN jamesjohnston No license field.

+ faker@4.1.0
added 1 package from 1 contributor and audited 1 package in 1.495s
found 0 vulnerabilities

James-MacBook-Air:~ jamesjohnston$ npm install fs
npm WARN saveError ENOENT: no such file or directory, open '/Users/jamesjohnston/package.json'
npm WARN enoent ENOENT: no such file or directory, open '/Users/jamesjohnston/package.json'
npm WARN jamesjohnston No description
npm WARN jamesjohnston No repository field.
npm WARN jamesjohnston No README data
npm WARN jamesjohnston No license field.

+ fs@0.1-security
added 1 package and audited 2 packages in 0.642s
found 0 vulnerabilities

James-MacBook-Air:~ jamesjohnston$
```

```
[Jamess-MacBook-Air:Documents jamesjohnston$ npm gulp-postcss
Usage: npm <command>
where <command> is one of:
  access, adduser, audit, bin, bugs, c, cache, ci, cit,
  clean-install, clean-install-test, completion, config,
  create, ddp, dedupe, deprecate, dist-tag, docs, doctor,
  edit, explore, get, help, help-search, hook, i, init,
  install, install-ci-test, install-test, it, link, list, ln,
  login, logout, ls, org, outdated, owner, pack, ping, prefix,
  profile, prune, publish, rb, rebuild, repo, restart, root,
  run, run-script, s, se, search, set, shrinkwrap, star,
  stars, start, stop, t, team, test, token, tst, un,
  uninstall, unpublish, unstar, up, update, v, version, view,
  whoami

npm <command> -h  quick help on <command>
npm -l       display full usage info
npm help <term>  search for help on <term>
npm help npm   involved overview

Specify configs in the ini-formatted file:
  /Users/jamesjohnston/.npmrc
or on the command line via: npm <command> --key value
Config info can be viewed via: npm help config

npm@6.12.0 /usr/local/lib/node_modules/npm

[Jamess-MacBook-Air:Documents jamesjohnston$ npm install gulp-postcss
npm WARN node-test@1.0.0 No repository field.

+ gulp-postcss@8.0.0
added 4 packages from 7 contributors and audited 10116 packages in 4.404s
found 0 vulnerabilities

[Jamess-MacBook-Air:Documents jamesjohnston$ npm install autoprefixer
npm WARN node-test@1.0.0 No repository field.

+ autoprefixer@9.7.3
added 4 packages from 4 contributors and audited 10144 packages in 4.189s
found 0 vulnerabilities

Jamess-MacBook-Air:Documents jamesjohnston$ ]
```

```
[James-MacBook-Air:Documents jamesjohnston$ npm install node-sass --save-dev

> node-sass@4.13.0 install /Users/jamesjohnston/Documents/node_modules/node-sass
> node scripts/install.js

Cached binary found at /Users/jamesjohnston/.npm/node-sass/4.13.0/darwin-x64-79_
binding.node

> node-sass@4.13.0 postinstall /Users/jamesjohnston/Documents/node_modules/node-
sass
> node scripts/build.js

Binary found at /Users/jamesjohnston/Documents/node_modules/node-sass/vendor/dar
win-x64-79/binding.node
Testing binary
Binary is fine
npm WARN node-test@1.0.0 No repository field.

+ node-sass@4.13.0
updated 1 package and audited 10673 packages in 9.585s
found 0 vulnerabilities
```

James-MacBook-Air:Documents jamesjohnston\$

```
James-MacBook-Air:Documents jamesjohnston$ sudo npm install --global gulp
Password:
Sorry, try again.
Password:
Sorry, try again.
Password:
npm WARN deprecated fsevents@1.2.9: One of your dependencies needs to upgrade to fsevents v2: 1) Proper nodejs v10+ support 2) No more fetching binaries from AWS, smaller package size
/usr/local/bin/gulp -> /usr/local/lib/node_modules/gulp/bin/gulp.js

> fsevents@1.2.9 install /usr/local/lib/node_modules/gulp/node_modules/fsevents
> node install

node-pre-gyp WARN Using need for node-pre-gyp https download
node-pre-gyp WARN Pre-built binaries not installable for fsevents@1.2.9 and node@13.0.1 (node-v79 ABI, unknown) (falling back to source compile with node-gyp)
node-pre-gyp ERR! Tried to download "https://node-pre-gyp/dist/install/shell/gyp/mkdir '/usr/local/lib/node_modules/gulp/node_modules/fsevents/lib'", but failed with error: Error: EACCES: permission denied, mkdir '/usr/local/lib/node_modules/gulp/node_modules/fsevents/lib'
gyp ERR! Error is: EACCES current user ("nobody") does not have permission to access the dev dir "/Users/jamesjohnston/Library/Caches/node-gyp/13.0.1"
gyp ERR! Error is: EACCES attempting to reinstall using temporary dev dir "/usr/local/lib/node_modules/gulp/node_modules/fsevents/.node-gyp"
gyp ERR! install got an error, rolling back install
gyp ERR! install got an error, rolling back install
gyp ERR! configure error
gyp ERR! stack Error: EACCES: permission denied, mkdir '/usr/local/lib/node_modules/gulp/node_modules/fsevents/.node-gyp'
gyp ERR! System Darwin 19.0.0
gyp ERR! command "node" "/usr/local/lib/node_modules/npm/node_modules/node-gyp/bin/node-gyp.js" "configure" "--fallback-to-build" "--module=/usr/local/lib/node_modules/gulp/node_modules
/fsevents/lib/binding/Release/node-v79-darwin-x64/fse-node" "--module_name=fse" "--module_path=/usr/local/lib/node_modules/gulp/node_modules/fsevents/lib/binding/Release/node-v79-darwin-x64" "--napi_v
ersion=5" "--node_abi_napi=napi" "--napi_build_version=0" "--node_napi_label=node-v79"
gyp ERR! cwd /usr/local/lib/node_modules/gulp/node_modules/fsevents
gyp ERR! node -v v13.0.1
gyp ERR! node-gyp -v V5.0.5
gyp ERR! stack Error: EACCES: permission denied, mkdir '/usr/local/lib/node_modules/gulp/node_modules/fsevents/lib/binding/Release/node-v79-darwin-x64/fse-node'
node-pre-gyp ERR! build error
node-pre-gyp ERR! stack Error: Failed to execute '/usr/local/bin/node' '/usr/local/lib/node_modules/npm/node_modules/node-gyp/bin/node-gyp.js' configure --fallback-to-build --module=/usr/local/lib/node_m
odules/gulp/node_modules/fsevents/lib/binding/Release/node-v79-darwin-x64/fse-node' --module_name=fse --module_path=/usr/local/lib/node_modules/gulp/node_modules/fsevents/lib/binding/Release/node-v79-d
arwin-x64 --napi_version=5 --node_abi_napi=napi --napi_build_version=0 --node_napi_label=node-v79' (1)
node-pre-gyp ERR! stack at ChildProcess. (/usr/local/lib/node_modules/gulp/node_modules/fsevents/node_modules/node-pre-gyp/lib/util/compile.js:83:29)
node-pre-gyp ERR! stack at ChildProcess.emit (events.js:218:5)
node-pre-gyp ERR! stack at maybeClose (internal/child_process.js:1028:16)
node-pre-gyp ERR! stack at Process.ChildProcess._handle.onexit (internal/child_process.js:283:5)
node-pre-gyp ERR! System Darwin 19.0.0
node-pre-gyp ERR! command "/usr/local/bin/node" "/usr/local/lib/node_modules/gulp/node_modules/fsevents/node_modules/node-pre-gyp/bin/node-pre-gyp" "install" "--fallback-to-build"
node-pre-gyp ERR! cwd /usr/local/lib/node_modules/gulp/node_modules/fsevents
node-pre-gyp ERR! node -v v13.0.1
node-pre-gyp ERR! node-pre-gyp -v v0.12.0
node-pre-gyp ERR! stack Error: EACCES: permission denied, mkdir '/usr/local/lib/node_modules/gulp/node_modules/fsevents/lib/binding/Release/node-v79-darwin-x64/fse-node'
Failed to execute '/usr/local/bin/node' '/usr/local/lib/node_modules/npm/node_modules/node-gyp/bin/node-gyp.js' configure --fallback-to-build --module=/usr/local/lib/node_modules/gulp/node_modules/fse
ts/lib/binding/Release/node-v79-darwin-x64/fse-node' --module_name=fse --module_path=/usr/local/lib/node_modules/gulp/node_modules/fsevents/lib/binding/Release/node-v79-darwin-x64 --napi_version=5 --no
de_abi_napi=napi --napi_build_version=0 --node_napi_label=node-v79' (1)
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@0.1.29 (node_modules/gulp/node_modules/fsevents):
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@0.1.29 install: "node install"
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: Exit status 1
```

```
+ gulp@4.0.2
added 314 packages from 217 contributors in 15.357s
James-MacBook-Air:Documents jamesjohnston$ npm install gulp
npm WARN node-test@1.0.0 No repository field.

+ gulp@4.0.2
updated 1 package and audited 10673 packages in 6.424s
found 0 vulnerabilities

James-MacBook-Air:Documents jamesjohnston$ npm install gulp-util
npm WARN deprecated gulp-util@3.0.8: gulp-util is deprecated - replace it, following the guidelines at https://medium.com/gulpjs/gulp-util-ca3b1f9f9ac5
npm WARN node-test@1.0.0 No repository field.

+ gulp-util@3.0.8
added 35 packages from 33 contributors and audited 10742 packages in 5.269s
found 0 vulnerabilities

James-MacBook-Air:Documents jamesjohnston$ npm install gulp-coffee
npm WARN node-test@1.0.0 No repository field.

+ gulp-coffee@3.0.3
added 2 packages from 2 contributors and audited 10768 packages in 4.136s
found 0 vulnerabilities

James-MacBook-Air:Documents jamesjohnston$ npm install gulp-uglify
npm WARN node-test@1.0.0 No repository field.

+ gulp-uglify@3.0.2
added 6 packages from 41 contributors and audited 10800 packages in 4.495s
found 0 vulnerabilities

James-MacBook-Air:Documents jamesjohnston$ npm install gulp-connect
npm WARN node-test@1.0.0 No repository field.

+ gulp-connect@5.7.0
added 30 packages from 37 contributors and audited 10906 packages in 6.48s
found 0 vulnerabilities

James-MacBook-Air:Documents jamesjohnston$
```

```
at Tuesday, 17 October 2017, 16:25:47 +0100
James-MacBook-Air:assignment_2 jamesjohnston$ gulp style
[16:25:47] Using gulpfile ~/Documents/assignment_2/gulpfile.js
[16:25:47] Starting 'style'...
[16:25:48] Finished 'style' after 1.28 s
James-MacBook-Air:assignment_2 jamesjohnston$
```

Appendix J: Package.json

```
1 ▼ {  
2     "name": "node-test",  
3     "version": "1.0.0",  
4     "description": "Hello World",  
5     "main": "main.js",  
6     "scripts": {  
7         "test": "echo \"Error: no test specified\" && exit  
8             1"  
9     },  
10    "author": "James Johnston",  
11    "license": "ISC",  
11 ▼ "dependencies": {  
12        "autoprefixer": "^9.7.3",  
13        "body-parser": "^1.19.0",  
14        "cssnano": "^4.1.10",  
15        "express": "^4.17.1",  
16        "faker": "^4.1.0",  
17        "fs": "0.0.1-security",  
18        "gulp": "^4.0.2",  
19        "gulp-concat": "^2.6.1",  
20        "gulp-postcss": "^8.0.0",  
21        "gulp-sass": "^4.0.2",  
22        "path": "^0.12.7",  
23        "postcss": "^7.0.24",  
24        "sass": "^1.23.7"  
25    },  
26    "devDependencies": {  
27        "node-sass": "^4.13.0"  
28    }  
29}  
30
```