

# Prototype

## Customer Relationship Management System (CRM)

**Kieran Williams**

**The prototype consists of documentation, source code, and a live website that can be used to look at the demo. In the instance that the server is down, you will be able to run the site from source code. A README.txt will also be provided to guide you in running the website locally**

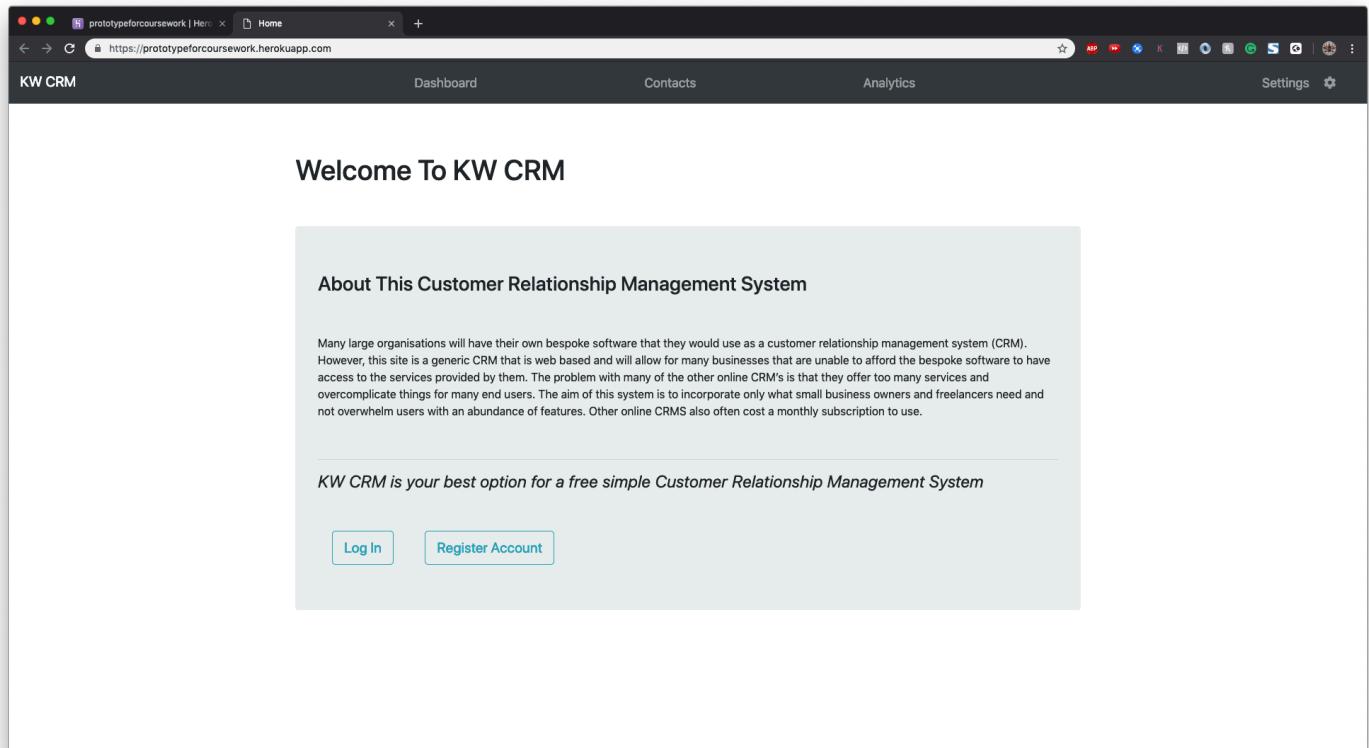
Prototype Link: <https://prototypeforcoursework.herokuapp.com/>

### What I included in the prototype

- \* Most of the webpages are created and made with html and css, along with some javascript for basic client Side functionality. I then turned the HTML files into EJS files so that the server can render them with data so that the site is dynamic
- \* placeholders are used where the data output in the final project will be dynamic but doesn't need to be for the purpose of the prototype
- \* The backend server side code is set up, allowing server side rendering and thus making dynamic webpages possible
- \* I included the ability to add, edit and delete contacts that are stored in csv files locally on the server

# The System

## The Homepage



The purpose of the home page is to provide a welcome screen to the user and to help get them up to speed on what the website is about. It is also used to aid the user down the direction of which routes they almost likely going to want to go down (the login and register routes) upon first accessing the site which has limited functionality without being logged in

Included Functionality	Related Objective	Justification For Inclusion
<b>Site is hosted on a web server</b>	1: The site must be accessible with a domain name typed into a web browser.	Will allow for me to showcase the site to any potential users around the world to get feedback. Also makes it easier to access the site as the server will have all technologies and will make it easy for me to showcase the site to anyone on any desktop computer with an internet connection and chrome web browser. This is the base route and so this is the first webpage that will be displayed
<b>Buttons to redirect to log in and register (requires implementation)</b>	9: Have an easy to navigate through system	Will help aid the user down the direction of which routes they almost likely going to want to go down (the login and register routes) upon first accessing the site which has limited functionality without being logged in. Thus making the system easier to navigate through. Non functional buttons placed there so that the user can see what the finished version is more likely to look like and so that I can get feedback on the style and placement of the buttons

Functionality Not Included	Related Objective	Justification For Exclusion
<b>The log in and register buttons don't redirect anywhere</b>	9: Have an easy to navigate through system	As I did not include any account functionality in the prototype I did not need the buttons to redirect to them pages
<b>No Account Features</b>	2: Account creation must be possible but only one account can be created per person  3: Users with an account should be able to log in to their account.	The account features are not needed to demonstrate the functionality of the CRM directly. Account functionality will be implemented more for the purposes of separating one users data from another.

## The Dashboard

The screenshot shows a web-based CRM application dashboard. At the top, there are tabs for 'KW CRM', 'Dashboard', 'Contacts', 'Analytics', and 'Settings'. The main area is titled 'Dashboard'.

**Schedule Task:**

- Date:** dd/mm/yyyy
- Time:** --::--
- Subject:** [empty input]
- Message:** [large text area]
- Submit** button

**Today:**

- 18:00 Fire Bob @ His office when he has finished his work because he is lazy
- 18:00 Fire Bob @ His office when he has finished his work because he is lazy
- 18:00 Fire Bob @ His office when he has finished his work because he is lazy
- 18:00 Fire Bob @ His office when he has finished his work because he is lazy
- 18:00 Fire Bob @ His office when he has finished his work because he is lazy
- 18:00 Fire Bob @ His office when he has finished his work because he is lazy

**Future:**

- 18:00 Fire Bob @ His office when he has finished his work because he is lazy

## Schedule Task

**date:**

**dd/mm/yyyy** ▼

February 2019 ◀ ▶

Mon	Tue	Wed	Thu	Fri	Sat	Sun
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	1	2	3

**time:**

**21:01** ✖ ▲ ▼

**subject:**

**message:**

The dashboard is just a mock up with placeholders. It will allow the user input scheduled task/appointment in the form on the left. View anything scheduled for the current date in the middle and will display all tasks/appointments scheduled further into the future from the current day.

There is a delete button alongside every list item. The data is just static data to get a UI feel.

Included Functionality	Related Objective	Justification For Inclusion
<b>Dashboard page has been made and set to render at dashboard route</b>	14: Things scheduled for the day should be made clear on the dashboard	Have created the dashboard with both the future and current date task holders. Have done this so that I can show users how I thought an intuitive way for that feature would be and to get feedback on if they agree with it being intuitive to use or if they have any other suggestions.
<b>Static data is displayed in the task holders</b>	n/a	Done so that the user can have a feel for how the page will look and respond to the data in the holders. By this I mean, how the date, time, subject and message will be displayed and how the user will be able to scroll through the holder when there is an overflow of data (more list items than the holder can display within its bounds)
<b>There is a future tasks holder</b>	14: Things scheduled for the day should be made clear on the dashboard	Although there was not a specific objective to have future tasks displayed, I implemented it as it makes sense to me to display future tasks stored in the database rather than only displaying the current day's tasks. Put it into the prototype to get users feedback on what they think about it.

Functionality Not Included	Related Objective	Justification For Exclusion
<b>The data in the holders is not dynamic data.</b>	14: Things scheduled for the day should be made clear on the dashboard	There is no need to get data from the database yet as the purpose of the prototype is to convey the basic idea of how it will look and operate. Through the use of dummy placeholder data, this is still conveyed
<b>The form does not send any data and store anything, it is just a place holder</b>	10: A relational database that is in third normal form with tables: Users, Contacts, Scheduling's, Notes, Conversions, Clients Lost.  13: There must be a way of scheduling tasks/ appointments from the dashboard.	The user will be able to get a good idea of how the system will work. This assumption is made due to the sub system all happening in a single page where there is only really two functions to the page (create a scheduled task and view the scheduled task). As the user should be able to get a good idea of how the sub system will work, the full functionality is redundant for the purposes of a prototype

## The Contacts List Page

The screenshot shows a web application interface for a CRM system. At the top, there's a navigation bar with links for 'Dashboard', 'Contacts', and 'Analytics'. On the far right of the navigation bar are icons for settings and other system functions.

The main content area is titled 'Contacts List'. On the left side of the main area, there is a sidebar containing:

- A 'Search' input field.
- A 'Sort By' section with three radio button options: 'Date', 'City', and 'Business Name'. The 'Date' option is selected.
- An 'Order' dropdown menu set to 'ascending'.
- A green 'Submit' button.
- A blue 'Add Contact' button.
- An 'Import Contacts (.CSV file)' section with a 'Choose file' button showing 'No file chosen'.
- At the bottom of the sidebar are two buttons: 'import Contact' and 'Export Contacts'.

To the right of the sidebar is a table with the following data:

Name	Business	City	Email	Telephone
kieran williams	bright bricks media	Swansea	kwtest@gmail.com	01792782635
james doug	jamesDen	swansea	jdoug@ymail.com	01762 836654
allen roger	bigMerch	Cardiff	alrog@gmail.com	07496437123
sally hughes	dynasty	Neath	dynastyClothing@gmail.com	07496467098
paul smith	paulsmith guitars	Swansea	psGuitars@gmail.com	07436890212
teddy lynn	tedRocks	neath	tedsgraft@gmail.com	07896783856
dan hughes	Dans Tree Services	neath	dannyboi@gmail.com	017927836654
Larry Watson	Dig Dig Diggers	Port Talbot	lazzywatz@gmail.com	07456897564
james brinks	James media	Swansea	jamesbrinks@hotmail.com	01792876785

This page is dynamic. The main purpose of the page is to display all the contacts stored in the database. On the left side there is a menu containing different sections. At the very top, there is a search bar (not functioning yet). When the search bar is implemented, it will only output the matches to the table on the right. Under the search bar there is a sorting form with a drop down menu so that you can sort the table displaying the contacts in both ascending and descending order. Under the sorting menu there is a button for the user to add a contact. When the button is pressed the user will get redirected to the add contact page (implemented). At the bottom of the menu there is an upload file menu that is not implemented yet, along with buttons at the bottom to import a selected file and to export all contacts that are saved in the database (both the buttons have no functionality right now). The Contacts list on the left is a table contains a few of the contacts details fields to help identify one user from the other. The name is also a link that redirects the user to that contacts individual page.

Included Functionality	Related Objective	Justification For Inclusion
<b>Dynamic list of contacts</b>	27: There must be a way of viewing all contacts that are stored in a list on a page	This was done because the main focus of the system is to manage contacts. The best way to convey the flow of this section of system is to implement core functionality. The contacts list needs to be dynamic to be able to showcase the outputs of contacts from the database. It has also been included as I have made the prototype capable of adding, editing and deleting contacts, to help the user visualise them processes happening, they need to be able to see the current contact records stored
<b>Menu UI on the left</b>	n/a	Included so that the user can see what menu features are intended to be implemented in the final solution and so that I can get feedback on if any of them features seem irrelevant, if there should be additional features, if the UI should be laid out differently, etc...
<b>A button to redirect the user to the add contact page</b>	26: There must be a way of adding, modifying and deleting contacts  9: Have an easy to navigate through system	There needs to be a way to access the add contacts route other than just by typing the route name into the url. Added the button with its functionality into the prototype because it was easy to do and the user will be able to tell me if it is in an intuitive position on the screen or if it needs to be changed at all.

Functionality Not Included	Related Objective	Justification For Exclusion
<b>Search bar</b>	24: The contacts in the contact view page must be searchable	Search bars are very common amongst many sites and so it is very clear how it will function and there is not much need to get user input based on how it will search for contacts.
<b>Sorting Menu</b>	22: The contacts in the contact view page must be sortable	The sorting functionality is not a main feature of the system and so the sorting functionality did not seem necessary for the purposes of the prototype
<b>Can not import or export contacts</b>	6: The ability to import and export the contacts in a CSV format.	the import and exporting of contacts is quite self explanatory so the user will be able to imagine how it will work just from the UI. I also wanted to get further input from the users on whether they think that the features are still necessary after seeing how the rest of the system operates.

## The Add Contact Page And The Edit Contact Page

**Create Contact**

**Contact Details**

Business	First Name
<input type="text"/>	<input type="text"/>
Last Name	Email
<input type="text"/>	<input type="text"/>
Telephone	City
<input type="text"/>	<input type="text"/>
Postcode	Address
<input type="text"/>	<input type="text"/>
Contact Status	
<input type="text" value="Lead"/>	
<b>Submit</b>	

**Edit Contact**

**Contact Details**

Business	First Name
<input type="text" value="jamesDen"/>	<input type="text" value="james"/>
Last Name	Email
<input type="text" value="doug"/>	<input type="text" value="jdoug@ymail.com"/>
Telephone	City
<input type="text" value="01762 836654"/>	<input type="text" value="swansea"/>
Postcode	Address
<input type="text" value="sa94bw"/>	<input type="text" value="lane street"/>
Contact Status	
<input type="text" value="Lead"/>	
<b>Submit</b>	

These pages are used to input a contacts details. It is essentially a page that contains a form consisting of labels, text fields, a drop down menu and a submit button. Upon submitting the contacts details the data gets saved into the csv file (either appended to the file or will edit the relevant details of the old file). Made both forms identical to keep consistent HCI

Included Functionality	Related Objective	Justification For Inclusion
<b>Takes user data in the text fields</b>	26: There must be a way of adding, modifying and deleting contacts	As the prototype is going to store user data, there needs to be a way to input that data. I decided to set the contacts status input as a drop down because from the investigation I learned that they are the only contact statuses that my targeted end users would want to use. By showing them this I will also be able to get further feedback on if they agree with my drop down menu input or if they would prefer a text field.
<b>After clicking submit, user details get saved to the database</b>	10: A relational database that is in third normal form with tables: Users, Contacts, Scheduling's, Notes, Conversions, Clients Lost.	Done so that the user can see the flow through the system surrounding the contacts section of the system. The data is stored in 3rd normal form so that my files are containable when it comes to scaling the project, having other data store files, and integrating all the files to give various outputs.
<b>The edit contact page pre fills in the text fields with the selected contact's details</b>	n/a	done because it is how most other systems with edit functionality would implement it.

Functionality Not Included	Related Objective	Justification For Exclusion
<b>No validation on data entered</b>	20: There should be adequate validation for any data input.	Validation at this stage isn't necessary because it will only hinder the demonstration of the system if the user was to put in invalid data and the validations can be discussed once the user has used the system and all the forms
<b>no security considerations. Nothing done to prevent cross site scripting in these fields</b>	12: There must be good website security	Due to this only being a prototype and no sensitive data being stored it is unnecessary to consider security at this stage.
<b>Data is being stored to csv files local to the server rather than to a database such as mysql</b>	n/a	As I am only trying to display a demonstration of the system, the way in which the data is stored does not matter. Instead of implementing mySql it will make sense
<b>The drop down menu in the edit contact page does not get set to the relevant contact status for that contact</b>	n/a	As I am unsure on if this will remain as a drop down menu and am awaiting the users opinions on it, there is no point in spending time on implementing that yet.

## The View Individual Contact Page

james doug

**Business:** jamesDen  
**Email:** jdoug@ymail.com  
**Phone Number:** 01762 836654  
**City:** swansea  
**Postcode:** sa94bw  
**Address:** lane street  
**Contacts Status:** lead

**Extra Notes**

**Add Note**

Date	Notes
21/10/2018	Go intaddafadfa da fdff fd fdfaf afadfa da fdff fd fdfaf da fa dfa a dfa dfa d adfafdafda
21/10/2018	Go intaddafadfa da fdff fd fdfaf afadfa da fdff fd fdfaf da afadfa da fdff fd fdfaf da fa dfa a dfa dfa d adfafdafda
21/10/2018	Go intaddafadfa da fdff fd fdfaf afadfa da fdff fd fdfaf da afadfa da fdff fd fdfaf da fa dfa a dfa dfa d adfafdafda
21/10/2018	Go intaddafadfa da fdff fd fdfaf afadfa da fdff fd fdfaf da afadfa da fdff fd fdfaf da fa dfa a dfa dfa d adfafdafda

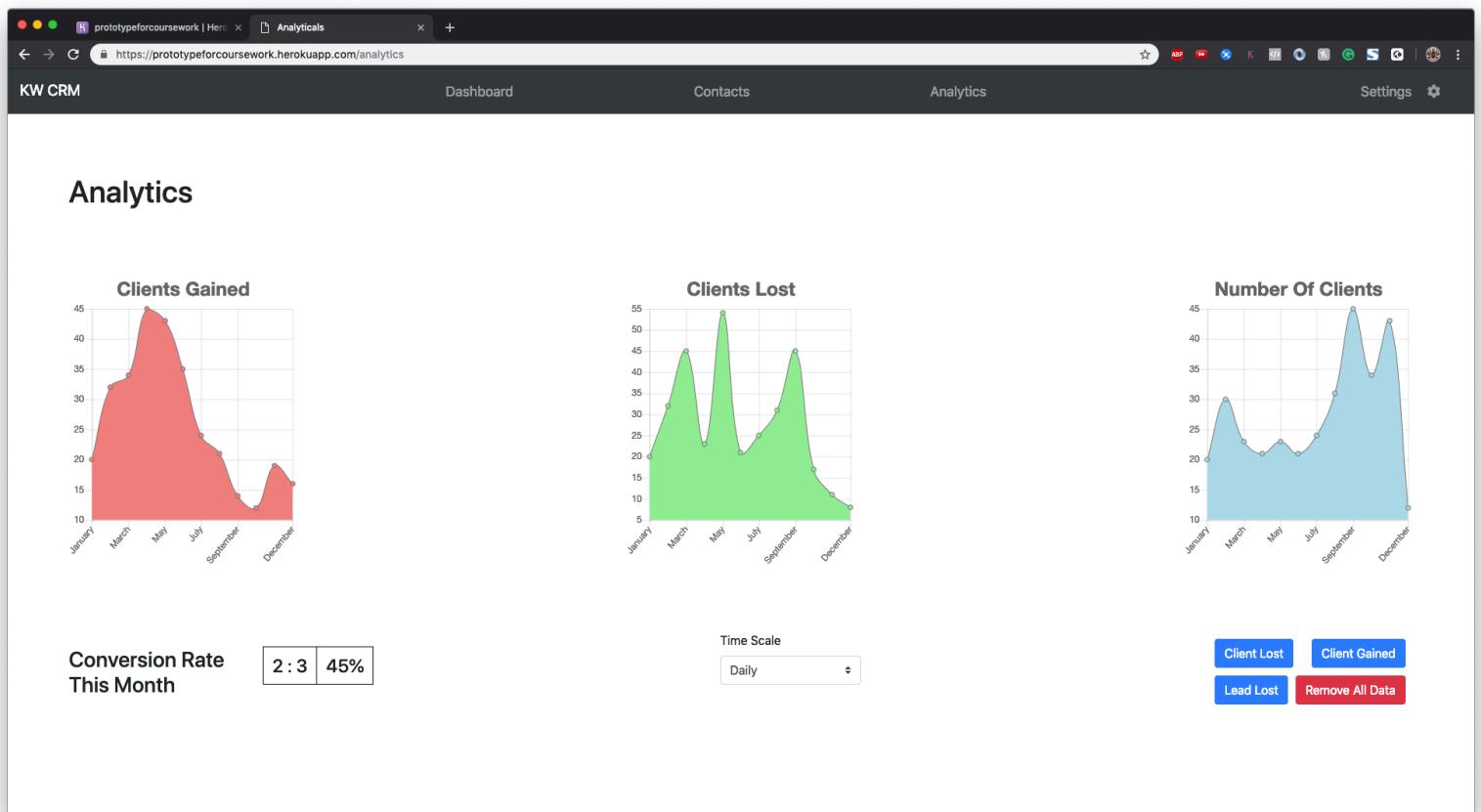
**Edit Contact** **Delete Contact**

The purpose of this page is to display all the details relating to a contact. This page is partially dynamic with other parts static containing mock data. In the top left of the page the contact's details are retrieved from the database and outputted into the screen. Above the details is the full name of the contact that is currently being inspected. Below the contact details in the bottom left is the add notes form which doesn't have any functionality yet but when implemented will take the text from the text area and send it to the database in a notes file with that contacts id as a foreign key. Under the form, at the bottom left is a button for editing a contact as well as a button for deleting a contact. The edit contact page redirects the user to the edit contact page. On the right there is an extra notes section, this is not implemented yet but it will basically be a text area that will contain data and the data can be encrypted if the user wishes to do so with a password. Under the encrypted notes section there is the notes table. This table contains dummy data where there is a date column and then next to that is a column which will take the data that was inputted into the add note section. The date will be the date of which that note was added.

Included Functionality	Related Objective	Justification For Inclusion
<b>Displays the contacts details of a contact that has been selected from the contact list page</b>	28: There must be a way of viewing all details of each individual contact on a page	This was implemented as I wanted the user to be how the entire flow adding, editing and deleting contacts would be. It is also quite easy to implement, and seems as I'd be needing to make delete and put routes for deletion and editing of a contact, it made sense to also add in the feature of rendering the contacts details into the page.
<b>The ui of the add notes form, extra notes form and the table of notes</b>	n/a	Designed the ui so that i could get feedback from the user to see if everything the page was intuitive on how they would use it.
<b>Edit contact button</b>	9: Have an easy to navigate through system	This page seems like the most relevant place have the edit and delete buttons. Implemented as they are integral parts of the system I would want to showcase to the users. By showing the user the part of the system they will be using the most (the contacts sections) It is the part of the system that needs to be as useful and intuitive to use as possible.
<b>Delete contact button</b>	26: There must be a way of adding, modifying and deleting contacts	

Functionality Not Included	Related Objective	Justification For Exclusion
<b>The notes</b>	<p>15: A notes section will be on the contact's page where the notes will get saved into the database.</p> <p>16: Notes relating to a contact should get displayed in chronological order</p>	The UI does a good enough job of conveying how the notes will work with the add note form and the table which displays the notes. There was no need to implement it in the prototype as the user might have some suggestions which could make my prototype code redundant
<b>The extra notes with encryption</b>	<p>17: A text area on the contact page that can store data encrypted through the use of a password.</p> <p>18: The password for the encrypted text area must be different to that of the account's login password.</p>	This will be a lot of work and is not a core part of the system. It would have taken a while to implement it and I wanted to also get the users opinion on flow of which the notes will get encrypted, if the encryption is really necessary, or if they even want the notes area to be both encryptable and normal when they already have the notes table.
<b>No validation on data entered</b>	20: There should be adequate validation for any data input.	Validation at this stage isn't necessary because it will only hinder the demonstration of the system if the user was to put in invalid data and the validations can be discussed once the user has used the system and all the forms
<b>No verification when the user clicks the delete contact button</b>	21: There should be adequate verification.	At this stage of the process I just want to convey the basic functionality of the system and so the verification of the user confirming that they want to delete a contact is sort of unnecessary. There will also be no real contacts either so if they get deleted then it's no loss to anyone

## The Analytics Page

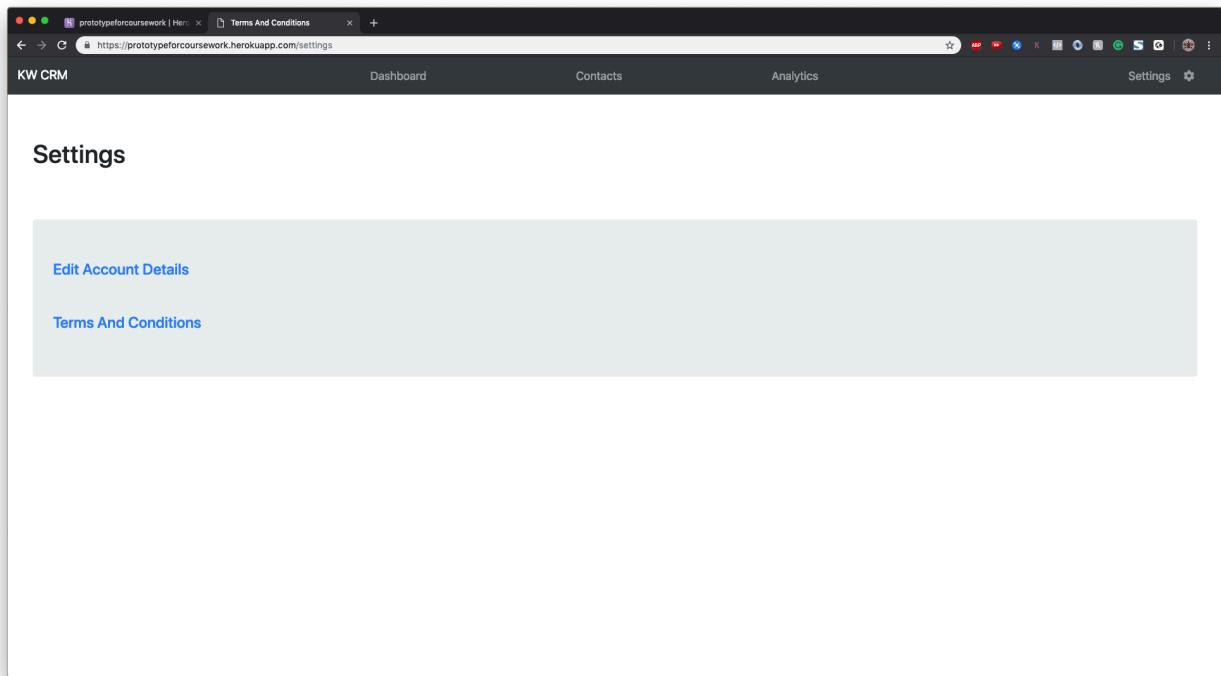


This page has 4 main components to it, all of which are static for the purposes of the prototype. At the top of the page there are 3 graphs which are going to be used to display the data which will be inputted throughout the buttons on the bottom right. In the bottom middle is a drop down menu so that the user can select the time scale of which the data points on the graph will get replotted. On the bottom left is the conversion rate for the month, it'll display the conversion rate for the month in both a ratio form (leads lost : clients gained) in the ratios lowest form, and as a percentage. There will also be a button that will allow the user to delete all data to start again

Included Functionality	Related Objective	Justification For Inclusion
<b>Graphs to present analytical data. Just the UI</b>	7: A graphical representation of clients gained and lost over a period of time	Done so that I can show the user how I thought the best way to display clients gained and lost over time would be and get their input on it.
<b>Conversion rate as a ratio and as a percentage. Just the UI</b>	8: There should be a display of the current conversion rate for the month	done so that I can get input from the user on if they like how the conversion rate is presented and if they still want the conversion rate for only a month.
<b>Buttons to add a client gained, client lost and lead lost</b>	11: There must be a page that allows the user to input that a conversion has been made, a lead was lost or a client has been lost, which gets recorded into the database.	Will need a way of inputting data so I decided to put in the buttons to finish off how a finalised page would look. Also, by putting in the buttons, it allowed me to sort out the CSS to make everything look good and presentable on the page, as the finalised page would look
<b>The time scale drop down menu UI</b>	n/a	As I neglected to find out what time scales they would want their data to be presented to them in, I created a drop down menu with choices: daily, weekly, monthly and yearly

Functionality Not Included	Related Objective	Justification For Exclusion
<b>The graphs are not dynamic</b>	7: A graphical representation of clients gained and lost over a period of time	I only want to see what the users think of my idea for presenting the data. To add in the functionality of the graphs would take a lot of work (dealing with dates, date subtraction, creating a database table, etc..), when I haven't even got any idea on what the user thinks of my design and implementation ideas.
<b>Conversion rate for the month is not dynamic</b>	8: There should be a display of the current conversion rate for the month	I was unsure on if the user would want to see the conversion rate as a percentage or as a ratio, so I did both so that they can decide on what they prefer before I spend time on implementing the functionality.
<b>The time scale drop down menu doesn't change any data presented</b>	n/a	This menu does not change the data presented on the graph as it would be a lot of work to fully implement when I am unsure on what the users want.

## The Settings Page



This page will contain additional links that the user not likely to want to access often and so aren't appropriate to be in the navigation bar. It currently contains 2 links to the edit account page (doesn't redirect anywhere as the users functionality hasn't been implemented yet). The terms and conditions link directs the user to the terms and conditions page.

Included Functionality	Related Objective	Justification For Inclusion
<b>Easy to find page with relevant links</b>	9: Have an easy to navigate through system	Included in the prototype to see if my idea of an easy to navigate through system aligns with how the users would think of an easy to navigate through system. Will be able to get feedback on where page links should be instead, so that the system can be more intuitive for the end users
<b>Terms and conditions link redirects the user to the Terms And Conditions page</b>	n/a	Need a way for the user to be able to access the terms and conditions page besides typing in the url to that page

Functionality Not Included	Related Objective	Justification For Exclusion
<b>Edit account details doesn't redirect anywhere</b>	n/a	I haven't implemented any user account functionality yet so it doesn't make sense to redirect them to any page of that sort at this stage.

## The Terms And Conditions Page

**By Using This Website You Are Agreeing To These Terms And Conditions.**

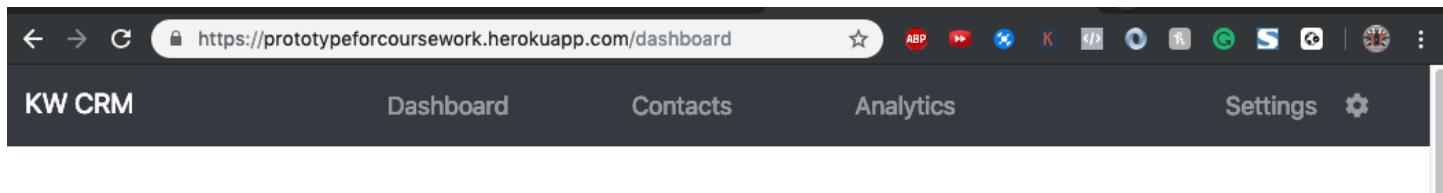
Any data held on your contacts is your responsibility.  
You agree to be liable for any GDPR law breeches that are related to the contacts you have created.

If you wish to receive data that we hold regarding your usage of the system  
you must make a formal request in the form of an email to kieranmarcus@KWCrm.com

This page is there to display the agreement the user must have with the site if they wish to use the system.

Included Functionality	Related Objective	Justification For Inclusion
<b>Displays the terms and conditions for the use of the system</b>	25: The project must be GDPR compliant.	I wanted to get the users opinion on if they think the terms and conditions of using this site are fair or should be modified

## The Navigation Bar



The navigation bar is at the top of every page on the website and is there to help the user navigate around the site.

Included Functionality	Related Objective	Justification For Inclusion
<b>Provides links to the most regularly accessed pages on the site to redirect the user to them pages</b>	9: Have an easy to navigate through system	I wanted to get the users opinion on if they think any additional pages should be put onto the nav bar or if they think any of the links on the nabber should be else where on the website.

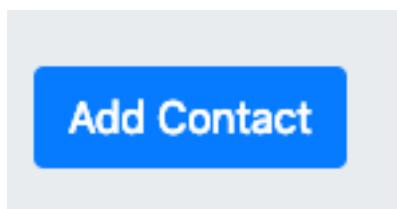
Functionality Not Included	Related Objective	Justification For Exclusion
<b>Nav Bar is not dynamic in the sense that it does not change dependent on If a user is logged in or not</b>	n/a	As there is no other user functionality, there is no reason for it to ever show the user not logged in version of the nav bar.

## The Flow Of Adding, Editing, And Deleting Contacts

### The Original Contacts List

Name	Business	City	Email	Telephone
kieran williams	bright bricks media	Swansea	kwtest@gmail.com	01792782635
james doug	jamesDen	swansea	jdoug@ymail.com	01762 836654
allen roger	bigMerch	Cardiff	alrog@gmail.com	07496437123
sally hughes	dynasty	Neath	dynastyClothing@gmail.com	07496467098
paul smith	paulsmith guitars	Swansea	psGuitars@gmail.com	07436890212
teddy lynn	tedRocks	neath	tedsgraft@gmail.com	07896783856
dan hughes	Dans Tree Services	neath	dannyboi@gmail.com	017927836654
Larry Watson	Dig Dig Diggers	Port Talbot	lazzywatz@gmail.com	07456897564
james brinks	James media	Swansea	jamesbrinks@hotmail.com	01792876785

This list is found on the contacts list page and will allow the user to see the contacts currently stored and click their link in the name column to take them to the contacts individual details page



upon the user clicking the add contact button on the contacts list page they will get redirected to the add contact page.

## The Create Contact Page

KW CRM

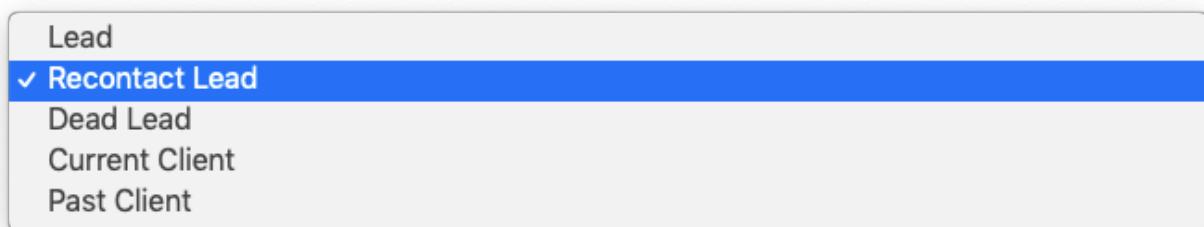
Dashboard Contacts Analytics Settings

Create Contact

Contact Details

Business	First Name
gold chainz	rob
Last Name	Email
sully	robsully@gmail.com
Telephone	City
01792934523	Swansea
Postcode	Address
SA8 4NP	42 Gelligron Rd
Contact Status	
Recontact Lead	
<b>Submit</b>	

User gets redirected to this page and fills in the contacts details into the text fields



User then selects the relevant contact status out of the list and then presses submit which redirects them back to the contacts list page

## The Contacts List With New Contact Addition

Name	Business	City	Email	Telephone
kieran williams	bright bricks media	Swansea	kwtest@gmail.com	01792782635
james doug	jamesDen	swansea	jdoug@ymail.com	01762 836654
allen roger	bigMerch	Cardiff	alrog@gmail.com	07496437123
sally hughes	dynasty	Neath	dynastyClothing@gmail.com	07496467098
paul smith	paulsmith guitars	Swansea	psGuitars@gmail.com	07436890212
teddy lynn	tedRocks	neath	tedsgraft@gmail.com	07896783856
dan hughes	Dans Tree Services	neath	dannyboi@gmail.com	017927836654
Larry Watson	Dig Dig Diggers	Port Talbot	lazzywatz@gmail.com	07456897564
james brinks	James media	Swansea	jamesbrinks@hotmail.com	01792876785
rob sully	gold chainz	Swansea	robsully@gmail.com	01792934523

The contact has been added to the database and is being displayed into the list on the contacts list page. The new contacts details can be fully inspected by clicking on the contacts name (blue text following the typical link HCI found on websites.)

## The New Contacts Individual Page

The screenshot shows a web browser window titled 'Individual Contact' with the URL <https://prototypeforcoursework.herokuapp.com/contacts/viewContactgyd4wnjslwk1u1>. The page is part of a CRM system with a dark header bar containing 'KW CRM', 'Dashboard', 'Contacts', 'Analytics', and 'Settings'. The main content area has a light gray background. At the top left, the contact's name 'rob sully' is displayed. Below it is a gray box containing the following contact details:

- Business:** gold chainz
- Email:** robsully@gmail.com
- Phone Number:** 01792934523
- City:** Swansea
- Postcode:** SA8 4NP
- Address:** 42 Gelligron Rd
- Contacts Status:** recontact-lead

To the right of this box is a section titled 'Extra Notes' with a large empty text area. At the bottom right of this area are two buttons: 'Encrypt Notes' and 'Submit'.

Below the contact details is a section titled 'Add Note' with a large empty text area and a 'Submit' button at the bottom right.

On the far right, there is a table with a light gray header and white rows, showing a list of notes:

Date	Notes
21/10/2018	Go intaddafadfa da fdfd fd fdaf afadfa da fdfd fd fdaf da fa dfd a adf af a da fa daf d adfafadfa
21/10/2018	Go intaddafadfa da fdfd fd fdaf afadfa da fdfd fd fdaf da afadfa da fdfd fd fdaf da afadfa da fdfd fd fdaf da fa dfd a adf af a da fa daf d adfafadfa
21/10/2018	Go intaddafadfa da fdfd fd fdaf afadfa da fdfd fd fdaf da afadfa da fdfd fd fdaf da afadfa da fdfd fd fdaf da fa dfd a adf af a da fa daf d adfafadfa
21/10/2018	Go intaddafadfa da fdfd fd fdaf afadfa da fdfd fd fdaf da afadfa da fdfd fd fdaf da afadfa da fdfd fd fdaf da fa dfd a adf af a da fa daf d adfafadfa

At the bottom left of the page are two buttons: 'Edit Contact' (gray) and 'Delete Contact' (red).

The contact now has a page rendered with his details in it when the user clicks on the link in the contacts list table. Upon clicking the edit button, the user will get redirected to the edit contact page.

**Edit Contact**

## The Edit Contact Page

The screenshot shows a web browser window titled "Edit Contact". The URL is <https://prototypeforcoursework.herokuapp.com/contacts/editgyd4wnjsiwkt1u1>. The page has a header with "KW CRM", "Dashboard", "Contacts", "Analytics", and "Settings". Below the header, the title "Edit Contact" is displayed. The main content is a form titled "Contact Details" with the following fields:

Business	First Name
Gold chainz	Robert
Last Name	Email
Sully	robsully@gmail.com
Telephone	City
01792934523	Swansea
Postcode	Address
SA8 4NP	42 Gelligron Rd
Contact Status	
Current Client	

A blue "Submit" button is located at the bottom of the form.

The fields are all pre-filled with the contacts details. I went in and capitalised some the fields and changed the contact status to lead. Then clicked the submit button. After clicking the submit button, the user gets redirected to the contacts list page and the users details will have been updated in the database.

## Contact Details Have Now Been Modified

Name	Business	City	Email	Telephone
kieran williams	bright bricks media	Swansea	kwtest@gmail.com	01792782635
james doug	jamesDen	swansea	jdoug@ymail.com	01762 836654
allen roger	bigMerch	Cardiff	alrog@gmail.com	07496437123
sally hughes	dynasty	Neath	dynastyClothing@gmail.com	07496467098
paul smith	paulsmith guitars	Swansea	psGuitars@gmail.com	07436890212
teddy lynn	tedRocks	neath	tedsgraft@gmail.com	07896783856
dan hughes	Dans Tree Services	neath	dannyboi@gmail.com	017927836654
Robert Sully	Gold chainz	Swansea	robsully@gmail.com	01792934523
Larry Watson	Dig Dig Diggers	Port Talbot	lazzywatz@gmail.com	07456897564
james brinks	James media	Swansea	jamesbrinks@hotmail.com	01792876785

clicking on the link to the contact again. we get taken to their individual contact page

The screenshot shows the 'Individual Contact' page for 'Robert Sully'. The top navigation bar includes 'KW CRM', 'Dashboard', 'Contacts', 'Analytics', and 'Settings'. The main content area displays contact details: Business: Gold chainz, Email: robsully@gmail.com, Phone Number: 01792934523, City: Swansea, Postcode: SA8 4NP, Address: 42 Gelligron Rd, and Contacts Status: current-client. To the right, there's a section titled 'Extra Notes' with a large text input field and buttons for 'Encrypt Notes' and 'Submit'. Below this is a table of notes with columns 'Date' and 'Notes', showing four entries from 21/10/2018. At the bottom are buttons for 'Edit Contact' and 'Delete Contact'.

Date	Notes
21/10/2018	Go intaddafadfa da fdff fd fdfaf afadfa da fdff fd fdfaf da fa dfd a adf af a da fa daf d adfafafdfa
21/10/2018	Go intaddafadfa da fdff fd fdfaf afadfa da fdff fd fdfaf da fa dfd a adf af a da fa daf d adfafafdfa
21/10/2018	Go intaddafadfa da fdff fd fdfaf afadfa da fdff fd fdfaf da fa dfd a adf af a da fa daf d adfafafdfa
21/10/2018	Go intaddafadfa da fdff fd fdfaf afadfa da fdff fd fdfaf da fa dfd a adf af a da fa daf d adfafafdfa

If the user clicks the delete contact button, all the data related to the currently selected contact gets deleted from the database (csv file storing the data).



Delete Contact

### Contact Has been Removed From The Database

Name	Business	City	Email	Telephone
kieran williams	bright bricks media	Swansea	kwtest@gmail.com	01792782635
james doug	jamesDen	swansea	jdoug@ymail.com	01762 836654
allen roger	bigMerch	Cardiff	alrog@gmail.com	07496437123
sally hughes	dynasty	Neath	dynastyClothing@gmail.com	07496467098
paul smith	paulsmith guitars	Swansea	psGuitars@gmail.com	07436890212
teddy lynn	tedRocks	neath	tedsgraft@gmail.com	07896783856
dan hughes	Dans Tree Services	neath	dannyboi@gmail.com	017927836654
Larry Watson	Dig Dig Diggers	Port Talbot	lazzywatz@gmail.com	07456897564
james brinks	James media	Swansea	jamesbrinks@hotmail.com	01792876785

As you can see, the contact is no longer in the table containing all the contacts stored the contacts file in the database.

## Evaluating The Prototype & Future Improvements

There are many parts of the system that need improving or additional functionality to meet the requirements of the specification. However many of them features were mentioned above. To avoid repeating myself, I will only discuss the additional features and improvements that have either not been discussed in great enough detail or haven't been given the opportunity to be discussed. I will also discuss some of the things that I have learned and will utilise while working towards the final system, along with some of the shortcomings I encountered also.

### User Accounts, And User's Data

Relating Objectives:

- 2: Account creation must be possible but only one account can be created per person
- 3: Users with an account should be able to log in to their account.
- 4: Users must be able to change their passwords at any time
- 10: A relational database that is in third normal form with tables: Users, Contacts, Scheduling's, Notes, Conversions, Clients Lost.

To make my system capable of handling multiple users and their data, I will need to have a way for users to be able to register, log in, and edit their account details. All data that is added by that user will also need to be linked to that user. This brings about another problem as the users data will soon become out of control if stored locally on the server with multiple users all regularly using the system. To solve this issue, I will need to connect the website to a separate server that will store the data in a database. Currently my system is using csv files to store data and my own algorithms which are potentially inadequate in terms of efficiency relating to both memory and time complexity when it comes to manipulating large pieces of data.

## Using The Heroku Service To Store My Web App

Relating Objectives:

1: The site must be accessible with a domain name typed into a web browser.

10: A relational database that is in third normal form with tables: Users, Contacts, Scheduling's, Notes, Conversions, Clients Lost.

Using Heroku to store and serve my web application is great as it will provide me with a free domain name to host my site with, a reasonable number of hours of website up time per month. Heroku also provides me with an easy to use command line interface application to help manage, trace errors (can see the log file), and deploy my website, it also allows manages any of my websites dependencies for me, so that I don't need to manually all the dependencies with the latest versions for security and dependency bug related reasons. Along with all of those features, I am also able add additional functionality to the site through the use of heroku's add on library (add ons include: search engine optimisation tools, security tools, ssl certification, Account management tools, etc...).

By utilising this service, it made it very easy to set up my website onto the internet to be able to easily show potential users my prototype without having to physically open it up and walk them through it.

One of the shortcomings and a downfall of the system moving forward is that due to the Heroku's server's "ephemeral" hard drive, I can write files to the disk, but those files will not persist after the application is restarted (automatically by the Heroku service). The website will remove any additional files or changes to files that have occurred since the website file has been deployed to the server. This means that any contacts added, edited or deleted, will not be persistent and after the server has restarted, the files containing the data will get reverted back to how they were when the website was deployed. In a way this is good though because if a user was to break the site, post restarting the server, the site would be back up online.

To overcome the problem with the ephemeral file system, I will need to either store and manipulate data using amazon web services with amazon s3. Or I could upload my final website to a server and

manage that server myself. Doing this will result in extra work in setting up the server and the possibility of security flaws if dependancies aren't regularly updated and managed.

## Web Browsers And Website In Lower Screen Resolutions

Related Objectives:

9: Have an easy to navigate through system

When Testing the site out on different sized screen resolutions, I realised that I had not optimised the site for varying screen resolutions. To overcome this, I could in the future add some css properties to the each web page's linked css file to move on the page around when the browsers canvas hits a certain resolution to optimise it.

When testing the site out on internet explorer I encountered numerous bugs due to the lack of support that the browser has for many of the later features in javascript and css that many other browsers support.

To overcome these problems I could notify users to ensure that they use the system in a Google Chrome Browser and that they use a standard desktop monitor with the highest resolution that they can use for optimal site performance, viewing, and navigation

I think the overall look of the site is very clean and minimalistic as it only has a few links per page and each webpage has very direct functionality (not multipurpose with lots of things you can do on every page). By keeping the website in the way described, it is very easy to both, use and navigate through. This was one of the main objectives because it is what made this web application in demand: the simplicity and core functionality is the differentiating factor compared to other current systems out there. And I feel as though I managed to achieve that.

## The UI And HCI Of The Website

I put a reasonable amount of consideration into the HCI of my system and designed the UI with care. I often referenced this website to ensure that the site was following common conventions.

Reference Site: <https://www.codecademy.com/articles/ui-design-conventions>

The logo of my site was put into the top left and takes the user to the home page as the website suggests it should.

My website includes a simplistic navigation bar at the top of every page with links that look clickable as it suggests to do. The links are made to look clickable by making the links lighter when the cursor is hovering over them. As shown below (cursor got removed from the screen capture image. On the left there is no hover, on the right is where the cursor is hovering over the link).



All text fields have labels next to them, there was consideration of input types for the data that would be inputted (text fields for data that will be varied, drop down menus where the data input will be finite, radio boxes where the input will be optional but needs to be obvious and check boxes where input is boolean.)

I feel as though the HCI of my website is good upon implementing appropriately, all that was mentioned above, this makes the site very easy to use.

# Source Code For Prototype

## Customer Relationship Management System (CRM)

Kieran Williams

# Root Directory Contents

# Index.js

```
1 const express = require("express");
2 const app = express();
3 const methodOverride = require("method-override");
4 const bodyParser = require("body-parser");
5 const session = require('express-session');
6
7 //requiring routes
8 const
9     indexRoutes = require("./routes/index"),
10    analyticsRoutes = require("./routes/analyticsRoutes"),
11    dashboardRoutes = require("./routes/dashboardRoutes"),
12    contactsRoutes = require("./routes/contactsRoutes");
13
14
15
16
17 // so that I can pass data between the backend and front end without just xhr requests
18 // .urlencoded({extended: true}) allows me to pass nested objects to the front end
19 // and from the front end to the server
20 app.use(bodyParser.urlencoded({extended: true}));
21 app.set("view engine", "ejs"); //sets my view engine to the ejs templating language
22 which will allow me to use js in the mark up
23 app.use('/public', express.static(__dirname + '/public'));
24 app.use(express.static(__dirname + '/public')); // express middleware to serve the
25 static files such as css and js to the client using the express HTTP framework
26 app.use(methodOverride("_method")); // methodOverride will allow me to use more http
27 verbs than html5 forms support
28 app.use(session({secret: 'mySecret', resave: false, saveUninitialized: false})); // will
29 allow me to store data to the session property
30
31 // setting my express http framework to use these url routes as the route paths for
32 // each
33 // of the different types of routes to aid readability with RESTFUL ROUTES for each
34 // of the routes
35 // the routes that extend from their base
36 app.use("/", indexRoutes);
37 app.use("/contacts", contactsRoutes);
38 app.use("/analytics", analyticsRoutes);
39 app.use("/dashboard", dashboardRoutes);
40
41 app.listen(process.env.PORT || 8000, (err) => {
42     if (err) {
43         console.log(err);
44     }
45     else {
46         console.log("server started on port 8000");
47     }
48 });
49
```

# Package-lock.json

```
1  {
2    "name": "kwcrmCourseworkProject",
3    "version": "1.0.0",
4    "lockfileVersion": 1,
5    "requires": true,
6    "dependencies": {
7      "accepts": {
8        "version": "1.3.5",
9        "resolved": "https://registry.npmjs.org/accepts/-/accepts-1.3.5.tgz",
10       "integrity": "sha1-63d99gEXI6OxTopywIBcjoZ0a9I=",
11       "requires": {
12         "mime-types": "2.1.20",
13         "negotiator": "0.6.1"
14       }
15     },
16     "array-flatten": {
17       "version": "1.1.1",
18       "resolved": "https://registry.npmjs.org/array-flatten/-/array-flatten-1.1.1.tgz",
19       "integrity": "sha1-ml9pkFGx5wczKPKgCJaLZOpVdI="
20     },
21     "body-parser": {
22       "version": "1.18.3",
23       "resolved": "https://registry.npmjs.org/body-parser/-/body-parser-1.18.3.tgz",
24       "integrity": "sha1-WykhmP/dVTs6DyDe0FkrlWlVyLQ=",
25       "requires": {
26         "bytes": "3.0.0",
27         "content-type": "1.0.4",
28         "debug": "2.6.9",
29         "depd": "1.1.2",
30         "http-errors": "1.6.3",
31         "iconv-lite": "0.4.23",
32         "on-finished": "2.3.0",
33         "qs": "6.5.2",
34         "raw-body": "2.3.3",
35         "type-is": "1.6.16"
36       }
37     },
38     "bytes": {
39       "version": "3.0.0",
40       "resolved": "https://registry.npmjs.org/bytes/-/bytes-3.0.0.tgz",
41       "integrity": "sha1-0ygVQE1olpn4Wk6k+odV3ROpYEg="
42     },
43     "content-disposition": {
44       "version": "0.5.2",
45       "resolved": "https://registry.npmjs.org/content-disposition/-/content-disposition-0.5.2.tgz"
46       ,
47       "integrity": "sha1-DPaLud318r55YcOoUXjLhdunjLQ="
48     },
49     "content-type": {
50       "version": "1.0.4",
51       "resolved": "https://registry.npmjs.org/content-type/-/content-type-1.0.4.tgz",
52       "integrity": "sha1-512-hIP3EEPs8tB9AT1L+NUqtwoAps4mk2Zob89MWXMHjHWg9milF/j4osnnQLXBCFBk/tvIG/tUc9mOUJiPBhPXAA=="
53     },
54     "cookie": {
55       "version": "0.3.1",
56       "resolved": "https://registry.npmjs.org/cookie/-/cookie-0.3.1.tgz",
57       "integrity": "sha1-5+Ch+e9DtMi6k1xcWpb0BtFoc7s="
58     },
59     "cookie-signature": {
60       "version": "1.0.6",
61       "resolved": "https://registry.npmjs.org/cookie-signature/-/cookie-signature-1.0.6.tgz",
62       "integrity": "sha1-4wOogrNCzD7oylE6eZmXNNqzriw="
63     },
64     "crc": {
65       "version": "3.4.4",
66       "resolved": "https://registry.npmjs.org/crc/-/crc-3.4.4.tgz",
67       "integrity": "sha1-naHpgOO9RPxck79as9ozeNheRms="
68   },
```

```
68 "debug": {
69   "version": "2.6.9",
70   "resolved": "https://registry.npmjs.org/debug/-/debug-2.6.9.tgz",
71   "integrity": "sha512-bC7ElrdJaJnPbAP+1EotYvqZsb3ec15wi6Bfi6BJTUCNowp6cvspg0jXznRTKDjm/E7AdgFBVeAPVMNcKGsHMA==",
72   "requires": {
73     "ms": "2.0.0"
74   }
75 },
76 "depd": {
77   "version": "1.1.2",
78   "resolved": "https://registry.npmjs.org/depd/-/depd-1.1.2.tgz",
79   "integrity": "sha1-m81S4UwJd2PnSbJ0xDRu0uVgtak="
80 },
81 "destroy": {
82   "version": "1.0.4",
83   "resolved": "https://registry.npmjs.org/destroy/-/destroy-1.0.4.tgz",
84   "integrity": "sha1-14hXRCxEJ5CBmE+N5RiBYJqvYA=="
85 },
86 "ee-first": {
87   "version": "1.1.1",
88   "resolved": "https://registry.npmjs.org/ee-first/-/ee-first-1.1.1.tgz",
89   "integrity": "sha1-WQxhFWsK4vTwJVcyoViyZrxWsh0="
90 },
91 "ejs": {
92   "version": "2.6.1",
93   "resolved": "https://registry.npmjs.org/ejs/-/ejs-2.6.1.tgz",
94   "integrity": "sha512-0xy4A/twfrRCnkhfk8ErDi5DqdAsAqeGxht4xkCURsvhhbQNs7E+4jV0CN7+NKIY0aHE72+XvqtBIXzD31ZbXQ=="
95 },
96 "encodeurl": {
97   "version": "1.0.2",
98   "resolved": "https://registry.npmjs.org/encodeurl/-/encodeurl-1.0.2.tgz",
99   "integrity": "sha1-rT/0yG7C0CkyL1oCw6mmBs1bP1k="
100 },
101 "escape-html": {
102   "version": "1.0.3",
103   "resolved": "https://registry.npmjs.org/escape-html/-/escape-html-1.0.3.tgz",
104   "integrity": "sha1-Aljq5NPQwJdN4cFpGI7wBR0dGYg=="
105 },
106 "etag": {
107   "version": "1.8.1",
108   "resolved": "https://registry.npmjs.org/etag/-/etag-1.8.1.tgz",
109   "integrity": "sha1-Qa4u62XvpiJorr/qg6x9eSmbCIC="
110 },
111 "express": {
112   "version": "4.16.3",
113   "resolved": "http://registry.npmjs.org/express/-/express-4.16.3.tgz",
114   "integrity": "sha1-avilAjUNsyRuzEvs9rWjTSL37VM=",
115   "requires": {
116     "accepts": "1.3.5",
117     "array-flatten": "1.1.1",
118     "body-parser": "1.18.2",
119     "content-disposition": "0.5.2",
120     "content-type": "1.0.4",
121     "cookie": "0.3.1",
122     "cookie-signature": "1.0.6",
123     "debug": "2.6.9",
124     "depd": "1.1.2",
125     "encodeurl": "1.0.2",
126     "escape-html": "1.0.3",
127     "etag": "1.8.1",
128     "finalhandler": "1.1.1",
129     "fresh": "0.5.2",
130     "merge-descriptors": "1.0.1",
131     "methods": "1.1.2",
132     "on-finished": "2.3.0",
133     "parseurl": "1.3.2",
134     "path-to-regexp": "0.1.7",
135     "proxy-addr": "2.0.4",
136     "qs": "6.5.1",
137   }
138 }
```

```
137     "range-parser": "1.2.0",
138     "safe-buffer": "5.1.1",
139     "send": "0.16.2",
140     "serve-static": "1.13.2",
141     "setprototypeof": "1.1.0",
142     "statuses": "1.4.0",
143     "type-is": "1.6.16",
144     "utils-merge": "1.0.1",
145     "vary": "1.1.2"
146   },
147   "dependencies": {
148     "body-parser": {
149       "version": "1.18.2",
150       "resolved": "https://registry.npmjs.org/body-parser/-/body-parser-1.18.2.tgz",
151       "integrity": "sha1-h2eKGdhLR9hZuDGZvVm84iKxBFQ=",
152       "requires": {
153         "bytes": "3.0.0",
154         "content-type": "1.0.4",
155         "debug": "2.6.9",
156         "depd": "1.1.2",
157         "http-errors": "1.6.3",
158         "iconv-lite": "0.4.19",
159         "on-finished": "2.3.0",
160         "qs": "6.5.1",
161         "raw-body": "2.3.2",
162         "type-is": "1.6.16"
163       }
164     },
165     "iconv-lite": {
166       "version": "0.4.19",
167       "resolved": "https://registry.npmjs.org/iconv-lite/-/iconv-lite-0.4.19.tgz",
168       "integrity": "sha512-oTZqweIP51xaGPI4uPa56/Pri/480R+mo7SeU+YETByQNhDG55ycFyNLigta9vXhILrxXdmF7ZGhqZIcuN0gJQ=="
169     },
170     "qs": {
171       "version": "6.5.1",
172       "resolved": "https://registry.npmjs.org/qs/-/qs-6.5.1.tgz",
173       "integrity": "sha512-eRzhrN1WSINYCDCbrz796z37LOe3m5tmW7RQf6oBntukAG1nmovJvhnwHHRMAfeoItclm2Hk02WER2aQ/iqs+A=="
174   },
175   "raw-body": {
176     "version": "2.3.2",
177     "resolved": "https://registry.npmjs.org/raw-body/-/raw-body-2.3.2.tgz",
178     "integrity": "sha1-vNYMd9Prk83gBQKVw/N5OJvIj4k=",
179     "requires": {
180       "bytes": "3.0.0",
181       "http-errors": "1.6.2",
182       "iconv-lite": "0.4.19",
183       "unpipe": "1.0.0"
184     },
185     "dependencies": {
186       "depd": {
187         "version": "1.1.1",
188         "resolved": "https://registry.npmjs.org/depd/-/depd-1.1.1.tgz",
189         "integrity": "sha1-V4004cRZ8G+lyif5kfPQbnoxA1k="
190     },
191     "http-errors": {
192       "version": "1.6.2",
193       "resolved": "https://registry.npmjs.org/http-errors/-/http-errors-1.6.2.tgz",
194       "integrity": "sha1-CgAsyFcHGSp+eUb07cERVfYOxzY=",
195       "requires": {
196         "depd": "1.1.1",
197         "inherits": "2.0.3",
198         "setprototypeof": "1.0.3",
199         "statuses": "1.4.0"
200       }
201     },
202     "setprototypeof": {
203       "version": "1.0.3",
```

```
204         "resolved":  
205         "https://registry.npmjs.org/setprototypeof/-/setprototypeof-1.0.3.tgz",  
206         "integrity": "sha1-ZlZ+NwQ+608E2RvWWMDL77VbjgQ="  
207     }  
208 },  
209     "statuses": {  
210         "version": "1.4.0",  
211         "resolved": "https://registry.npmjs.org/statuses/-/statuses-1.4.0.tgz",  
212         "integrity":  
213         "sha512-zhSCtt8v2NDrR1PQpCNtw/heZLtfUDqxBMludqikb/Hbk52LK4nQSwr10u77iopCW5LsyHpuXS0GnEc48mLeew=="  
214     }  
215 },  
216     "express-session": {  
217         "version": "1.15.6",  
218         "resolved":  
219         "https://registry.npmjs.org/express-session/-/express-session-1.15.6.tgz",  
220         "integrity":  
221         "sha512-r0nrHTCYtAMrFwZ0kBzZEa1vtPVrw0dKvGSrKP4dahwBQ1BJpF2/y1Pp4sCD/0kvxV4zzec1yvfmw0B4RMJQA==",  
222         "requires": {  
223             "cookie": "0.3.1",  
224             "cookie-signature": "1.0.6",  
225             "crc": "3.4.4",  
226             "debug": "2.6.9",  
227             "depd": "1.1.2",  
228             "on-headers": "1.0.1",  
229             "parseurl": "1.3.2",  
230             "uid-safe": "2.1.5",  
231             "utils-merge": "1.0.1"  
232         }  
233     },  
234     "filereader": {  
235         "version": "0.10.3",  
236         "resolved": "https://registry.npmjs.org/filereader/-/filereader-0.10.3.tgz",  
237         "integrity": "sha1-x0fUos2PYeVBinwH/hJXpD8KzbE="  
238 },  
239     "finalhandler": {  
240         "version": "1.1.1",  
241         "resolved": "https://registry.npmjs.org/finalhandler/-/finalhandler-1.1.1.tgz",  
242         "integrity":  
243         "sha512-Y1GUDo39ez4aH Aw7MysnUD5JzYX+Waij8I57kO3aEPT1fFRL4sr7mjei97FgnwhAyyzRYmQZaTHb2+9uZ1dPtg==",  
244         "requires": {  
245             "debug": "2.6.9",  
246             "encodeurl": "1.0.2",  
247             "escape-html": "1.0.3",  
248             "on-finished": "2.3.0",  
249             "parseurl": "1.3.2",  
250             "statuses": "1.4.0",  
251             "unpipe": "1.0.0"  
252         }  
253     },  
254     "dependencies": {  
255         "statuses": {  
256             "version": "1.4.0",  
257             "resolved": "https://registry.npmjs.org/statuses/-/statuses-1.4.0.tgz",  
258             "integrity":  
259             "sha512-zhSCtt8v2NDrR1PQpCNtw/heZLtfUDqxBMludqikb/Hbk52LK4nQSwr10u77iopCW5LsyHpuXS0GnEc48mLeew=="  
260         }  
261     },  
262     "forwarded": {  
263         "version": "0.1.2",  
264         "resolved": "https://registry.npmjs.org/forwarded/-/forwarded-0.1.2.tgz",  
265         "integrity": "sha1-mMI9qxFlZXuMBXPozszZGw/xjIQ="  
266     },  
267     "fresh": {  
268         "version": "0.5.2",  
269         "resolved": "https://registry.npmjs.org/fresh/-/fresh-0.5.2.tgz",  
270         "integrity": "sha1-PYyt2Q2XZWn6glqx+OSyOhBWBac="
```

```
267 },
268 "http-errors": {
269   "version": "1.6.3",
270   "resolved": "http://registry.npmjs.org/http-errors/-/http-errors-1.6.3.tgz",
271   "integrity": "sha1-i1VoC7S+KDoLW/TqLjhYC+HZMg0=",
272   "requires": {
273     "depd": "1.1.2",
274     "inherits": "2.0.3",
275     "setprototypeof": "1.1.0",
276     "statuses": "1.5.0"
277   }
278 },
279 "iconv-lite": {
280   "version": "0.4.23",
281   "resolved": "https://registry.npmjs.org/iconv-lite/-/iconv-lite-0.4.23.tgz",
282   "integrity": "sha512-neyTUVFtahjf0mB3dZT77u+800QB89jFdNBkd5P1JgYPbPaia3gXXOVL2fq8VyU2gMMD7SaN7QukTB/pmXYvDA==",
283   "requires": {
284     "safer-buffer": "2.1.2"
285   }
286 },
287 "inherits": {
288   "version": "2.0.3",
289   "resolved": "https://registry.npmjs.org/inherits/-/inherits-2.0.3.tgz",
290   "integrity": "sha1-Yzwsg+PaQqUC9SRmAisA9CCYd4="
291 },
292 "ipaddr.js": {
293   "version": "1.8.0",
294   "resolved": "https://registry.npmjs.org/ipaddr.js/-/ipaddr.js-1.8.0.tgz",
295   "integrity": "sha1-6qM9bd16zo9/b+DJygRA5wZzix4="
296 },
297 "media-typer": {
298   "version": "0.3.0",
299   "resolved": "https://registry.npmjs.org/media-typer/-/media-typer-0.3.0.tgz",
300   "integrity": "sha1-hxDXrwqmJvj/+hzgAWhUUmMlV0g="
301 },
302 "merge-descriptors": {
303   "version": "1.0.1",
304   "resolved": "https://registry.npmjs.org/merge-descriptors/-/merge-descriptors-1.0.1.tgz",
305   "integrity": "sha1-sAqqVW3YtEVoFQ7J0blT8/kMu2E="
306 },
307 "method-override": {
308   "version": "3.0.0",
309   "resolved": "https://registry.npmjs.org/method-override/-/method-override-3.0.0.tgz",
310   "integrity": "sha1-IJ2NNN/mSl9w3kzWB92rcdHpz+HjkxhDJWNDBqSlas+zQdP8wBiJzITPg08M/k2uVvMow7Sk4latndNtt/PHSA==",
311   "requires": {
312     "debug": "3.1.0",
313     "methods": "1.1.2",
314     "parseurl": "1.3.2",
315     "vary": "1.1.2"
316   },
317   "dependencies": {
318     "debug": {
319       "version": "3.1.0",
320       "resolved": "https://registry.npmjs.org/debug/-/debug-3.1.0.tgz",
321       "integrity": "sha1-OX8XqP7/1a9cqkxYw2yXss15f26NKWBpDXQd0/uK/KPqdQhxbPa994hnzjcE2VqQpDslf55723cKPUOGSmMY3g==",
322       "requires": {
323         "ms": "2.0.0"
324       }
325     }
326   }
327 },
328 "methods": {
329   "version": "1.1.2",
330   "resolved": "https://registry.npmjs.org/methods/-/methods-1.1.2.tgz",
331   "integrity": "sha1-VSmklNZUE07cxSZmVoNbD4Ua/O4="
```

```
332 },
333   "mime": {
334     "version": "1.4.1",
335     "resolved": "https://registry.npmjs.org/mime/-/mime-1.4.1.tgz",
336     "integrity": "sha512-KI1+qOZu5DcW6wayYHSzR/tXKCDC5Om4s1z2QJjDULzLcmf3DvzS7oluY4HCTrc+9FiKmWUgeNLg7W3uIQvxtQ=="
337   },
338   "mime-db": {
339     "version": "1.36.0",
340     "resolved": "https://registry.npmjs.org/mime-db/-/mime-db-1.36.0.tgz",
341     "integrity": "sha512-L+xvyD9MkoYMXb1jAmzI/lWYAxAMCPvIBSWur0PZ5nOf5euahRLVqH//FKW9mWp21kqUgYixPgkzfMUfi4zVdW=="
342   },
343   "mime-types": {
344     "version": "2.1.20",
345     "resolved": "https://registry.npmjs.org/mime-types/-/mime-types-2.1.20.tgz",
346     "integrity": "sha512-HrkrPaP9vGuWbLK1B1FfgAkbqNjIuy4eHlIYnFi7kamZyLLrGlo2mpcx0bBmNpKqBtYtAfGbodDddIgddSJC2A==",
347     "requires": {
348       "mime-db": "1.36.0"
349     }
350   },
351   "moment": {
352     "version": "2.24.0",
353     "resolved": "https://registry.npmjs.org/moment/-/moment-2.24.0.tgz",
354     "integrity": "sha512-bV7f+6l2QigeBBZSM/6yTNq4P2fNpSWj/0e7jQcy87A8e7o2nAfP/34/2ky5Vw4B9$446EtI HodAzkFCcR4dQg=="
355   },
356   "ms": {
357     "version": "2.0.0",
358     "resolved": "https://registry.npmjs.org/ms/-/ms-2.0.0.tgz",
359     "integrity": "sha1-VgiurfwAvmwpAd9fmGF4jeDVl8g="
360   },
361   "negotiator": {
362     "version": "0.6.1",
363     "resolved": "https://registry.npmjs.org/negotiator/-/negotiator-0.6.1.tgz",
364     "integrity": "sha1-KzJxhOizIQExeyhWP7XnECrNDKk="
365   },
366   "on-finished": {
367     "version": "2.3.0",
368     "resolved": "https://registry.npmjs.org/on-finished/-/on-finished-2.3.0.tgz",
369     "integrity": "sha1-IPEzzIGwg811M3mSoWlxqi2QaUc=",
370     "requires": {
371       "ee-first": "1.1.1"
372     }
373   },
374   "on-headers": {
375     "version": "1.0.1",
376     "resolved": "https://registry.npmjs.org/on-headers/-/on-headers-1.0.1.tgz",
377     "integrity": "sha1-ko9dD0cNSTQmUepn1LCFFBAGk/c="
378   },
379   "parseurl": {
380     "version": "1.3.2",
381     "resolved": "https://registry.npmjs.org/parseurl/-/parseurl-1.3.2.tgz",
382     "integrity": "sha1-/CidTtizMR1GDBViUyYs3I3mW/M="
383   },
384   "path-to-regexp": {
385     "version": "0.1.7",
386     "resolved": "https://registry.npmjs.org/path-to-regexp/-/path-to-regexp-0.1.7.tgz",
387     "integrity": "sha1-32BBeABfUi8V60SQ5yR6G/qmf4w="
388   },
389   "proxy-addr": {
390     "version": "2.0.4",
391     "resolved": "https://registry.npmjs.org/proxy-addr/-/proxy-addr-2.0.4.tgz",
392     "integrity": "sha512-5erio2h9jp5CHGwcybmxmVqHmnCBZeewlfJ0pex+UW7Qny7OOZXTth56TGNyBizkgiOwhJtMKrVzDTeKcySzWa==",
393     "requires": {
```

```
394     "forwarded": "0.1.2",
395     "ipaddr.js": "1.8.0"
396   },
397   "qs": {
398     "version": "6.5.2",
399     "resolved": "https://registry.npmjs.org/qs/-/qs-6.5.2.tgz",
400     "integrity":
401       "sha512-N5ZAX4/LxJmF+7wN74pUD6qAh9/wnvdQcj9TZjevvXzSUo7bfmw91saqMjzGS2xq91/odN2dW/WO17qQHNDGA=="
402   },
403   "random-bytes": {
404     "version": "1.0.0",
405     "resolved": "https://registry.npmjs.org/random-bytes/-/random-bytes-1.0.0.tgz",
406     "integrity": "sha1-T2ih3Arli9P7lYSMMJDNt11kNgs="
407   },
408   "range-parser": {
409     "version": "1.2.0",
410     "resolved": "https://registry.npmjs.org/range-parser/-/range-parser-1.2.0.tgz",
411     "integrity": "sha1-9JvmtIeJTdxA3M1KMi9hEJLgDV4="
412   },
413   "raw-body": {
414     "version": "2.3.3",
415     "resolved": "https://registry.npmjs.org/raw-body/-/raw-body-2.3.3.tgz",
416     "integrity":
417       "sha512-9esiElv1BrZoI3rCDuOuKCBRbuApGGaDPQfjSf1Gxdy4oyzqghxu6k1EkkVIVBje+FF0BX9coEv8KqW6X/7njw==",
418     "requires": {
419       "bytes": "3.0.0",
420       "http-errors": "1.6.3",
421       "iconv-lite": "0.4.23",
422       "unpipe": "1.0.0"
423     }
424   },
425   "safe-buffer": {
426     "version": "5.1.1",
427     "resolved": "https://registry.npmjs.org/safe-buffer/-/safe-buffer-5.1.1.tgz",
428     "integrity":
429       "sha512-kKvNJn6Mm93gAczWVJg7wH+wGYWNrDHdWvpUmHyEsgCtIwoo3bqPtV4tR5tuPaUhTOo/kvhVwd8XwwOllGYkbg=="
430   },
431   "safer-buffer": {
432     "version": "2.1.2",
433     "resolved": "https://registry.npmjs.org/safer-buffer/-/safer-buffer-2.1.2.tgz",
434     "integrity":
435       "sha512-YZo3K82SD7Riyi0E1EQPojLz7kpepnSQI9IyPbHHg1XXXevb5dJI7tpyN2ADxGcQbHG7vcyRHk0cbwqcQriUtg=="
436   },
437   "send": {
438     "version": "0.16.2",
439     "resolved": "https://registry.npmjs.org/send/-/send-0.16.2.tgz",
440     "integrity":
441       "sha512-E64YFPUsxFHEFBvpbbjr44NCLtI1AohxQ8ZSiJjQLskAdKuriYEP6VyGEsRDH8ScozGpkaX1BGvhanqCwkcEZw==",
442     "requires": {
443       "debug": "2.6.9",
444       "depd": "1.1.2",
445       "destroy": "1.0.4",
446       "encodeurl": "1.0.2",
447       "escape-html": "1.0.3",
448       "etag": "1.8.1",
449       "fresh": "0.5.2",
450       "http-errors": "1.6.3",
451       "mime": "1.4.1",
452       "ms": "2.0.0",
453       "on-finished": "2.3.0",
454       "range-parser": "1.2.0",
455       "statuses": "1.4.0"
456     },
457     "dependencies": {
458       "statuses": {
459         "version": "1.4.0",
460         "resolved": "https://registry.npmjs.org/statuses/-/statuses-1.4.0.tgz",
461       }
462     }
463   }
464 }
```

```
457     "integrity":  
458     "sha512-zhSCtt8v2NDrR1PQpCNtw/heZLtfUDqxBMludqikb/Hbk52LK4nQSwr10u77iopCW5Ls  
459     }  
460   },  
461   "serve-static": {  
462     "version": "1.13.2",  
463     "resolved": "https://registry.npmjs.org/serve-static/-/serve-static-1.13.2.tgz",  
464     "integrity":  
465     "sha512-p/tdJrO4U387R9oMjb1oj7qSMaMfmOyd4j9hOFoxZe2baQszgHcSWjuya/CiT5kgZZKRudHN  
466     OA0pYX018rQ5nw==",  
467     "requires": {  
468       "encodeurl": "1.0.2",  
469       "escape-html": "1.0.3",  
470       "parseurl": "1.3.2",  
471       "send": "0.16.2"  
472     }  
473   },  
474   "setprototypeof": {  
475     "version": "1.1.0",  
476     "resolved": "https://registry.npmjs.org/setprototypeof/-/setprototypeof-1.1.0.tgz",  
477     "integrity":  
478     "sha512-BvE/TwpZX4FXExxOxZyRGQQv651MSwmWKZGqvmPcRIjDqWub67kTKuIMx43cZZrs/cBBzwBc  
479     NDWoFxt2XEFlpQ=="  
480   },  
481   "statuses": {  
482     "version": "1.5.0",  
483     "resolved": "https://registry.npmjs.org/statuses/-/statuses-1.5.0.tgz",  
484     "integrity": "sha1-Fhx9rBd2Wf2YEFQ3cfqZOBR4Yow="  
485   },  
486   "type-is": {  
487     "version": "1.6.16",  
488     "resolved": "https://registry.npmjs.org/type-is/-/type-is-1.6.16.tgz",  
489     "integrity":  
490     "sha512-HRkVv/5qY2G6I8iab9cI7v1b0Idhm94dVjQCPFE1W9W+3GeDOSHmy2EBYe4VTApuzolPcmgF  
491     TN3ftVJRKR2J9Q==",  
492     "requires": {  
493       "media-typer": "0.3.0",  
494       "mime-types": "2.1.20"  
495     }  
496   },  
497   "uid-safe": {  
498     "version": "2.1.5",  
499     "resolved": "https://registry.npmjs.org/uid-safe/-/uid-safe-2.1.5.tgz",  
500     "integrity":  
501     "sha512-KPHm4VL5dDXKz01UuEd88Df+KzynaohSL9fBh096KWAxSKZQDI2uBrVqtvRM4rwrIrRRKsdL  
502     NML/lnaaVSRioA==",  
503     "requires": {  
504       "random-bytes": "1.0.0"  
505     }  
506   },  
507   "uniqid": {  
508     "version": "5.0.3",  
509     "resolved": "https://registry.npmjs.org/uniqid/-/uniqid-5.0.3.tgz",  
510     "integrity":  
511     "sha512-R2qx3X/LYWSdGRalui04dYrPXAJACTqyUjuyXHoJLBUIfmMcnyOyY2d6Y4clZcIz5lK6Zai  
512     OZzmm0cPfsIqzQ=="  
513   },  
514   "unpipe": {  
515     "version": "1.0.0",  
516     "resolved": "https://registry.npmjs.org/unpipe/-/unpipe-1.0.0.tgz",  
517     "integrity": "sha1-sr906FFKrmFltIF4KdIBLvSZB0w="  
518   },  
519   "utils-merge": {  
520     "version": "1.0.1",  
521     "resolved": "https://registry.npmjs.org/utils-merge/-/utils-merge-1.0.1.tgz",  
522     "integrity": "sha1-n5VxD1CiZ5R7LMwSR0HBaoQn5xM="  
523   },  
524   "vary": {  
525     "version": "1.1.2",  
526     "resolved": "https://registry.npmjs.org/vary/-/vary-1.1.2.tgz",  
527   }
```

```
517     "integrity": "sha1-IpnwLG3tMNS1lhsLn3RSShj2NPw="
518 }
519 }
520 }
521 }
```

# Package.json

```
1  {
2      "name": "kwcrmCourseworkProject",
3      "version": "1.0.0",
4      "description": "",
5      "main": "index.js",
6      "scripts": {
7          "test": "echo \\\"Error: no test specified\\\" && exit 1",
8          "start": "node index.js"
9      },
10     "author": "kieran williams",
11     "license": "ISC",
12     "dependencies": {
13         "body-parser": "^1.18.3",
14         "ejs": "^2.6.1",
15         "express": "^4.16.3",
16         "express-session": "^1.15.6",
17         "filereader": "^0.10.3",
18         "method-override": "^3.0.0",
19         "moment": "^2.24.0",
20         "uniqid": "^5.0.3"
21     }
22 }
23
```

# Classes

# Directory

# Contents

# CSVHandler.js

```

1 "use strict";
2
3 const
4   fs = require("fs"),
5   path = require('path'),
6   uniqueID = require('uniqid'),
7   Sort = require("./Sort"),
8   Search = require("./Search"),
9   DateHandler = require("./DateHandler");
10
11
12 // way of imitating private methods and variables as the JS does not have a
13 // a way to define them yet. Done by differentiating the way
14 // they are accessed. It isn't, accessible through the object model
15 // of the class like you would expect the method or variable to be.
16 // This is done so that it is clear what is meant to be accessible and
17 // what is not, from outside the class. Even tho the IDE won't
18 // suggest the methods or variables, it is still possible to
19 // access them. And so, this is not perfect but better than nothing.
20
21 const _csvToTxt = Symbol('csvToTxt');
22 const _txtToCsv = Symbol('txtToCsv');
23 const _csvToArray = Symbol('txtInCsvFormatToArray');
24 const _FILE_PATH = Symbol('filePath');
25 const _rewriteDataToFileWithChanges = Symbol('rewriteDataToFileWithChanges');
26 const _generateID = Symbol('generateID');
27 const _makeChangesToRecord = Symbol('makeChangesToRecord');
28 const _getLastRecordItem = Symbol('_getLastRecordItem');
29
30
31 class CSVHandler {
32
33   constructor(filePath) {
34     // private property of the class
35     this[_FILE_PATH] = filePath;
36   }
37
38   // these functions are used to change the
39   // file from a csv file to a txt file or vice versa
40   // these are private methods
41   [_csvToTxt]() {
42     return path.basename(this[_FILE_PATH], '.txt');
43   }
44
45   [_txtToCsv]() {
46     return path.basename(this[_FILE_PATH], '.csv');
47   }
48
49   // will put a txt file that is in a csv format into a 2d array and return it
50   [_csvToArray]() {
51     let file = fs.readFileSync(this[_FILE_PATH], 'utf8');
52     let recordsArray = file.split("\n");
53     let csvInArray = [];
54
55     // there is an empty array element at the end of each record
56     // in this loop every record has the empty element removed and the rest of the
57     // array gets put back into the csvInArray array.
58     for (let i = 0; i < recordsArray.length; i++) {
59       let singleRecordArray = recordsArray[i].split(",");
60       singleRecordArray.pop(); // gets rid of the empty last element in the
61       // array
62       csvInArray.push(singleRecordArray);
63     }
64     return csvInArray;
65   }
66
67   // returns all records stored in the file
68   getAllRecords () {
69     let arrayOfRecords = this[_csvToArray]();
70     arrayOfRecords.pop();
71     return arrayOfRecords;
72   }

```

```

73
74
75 // used to find a record by it's id
76 // requires only the id to retrieve the record
77 selectRecordByID(id) {
78     let array = this[_csvToArray]();
79     array.pop();
80
81     let recordFoundByID = Search.binarySearch2D(array, [0], id); // Stores the
82     // record found.
83
84     // function done on recordFoundByID in the return is done to flatten the array
85     // Output would be a multi dimensional array otherwise containing just one
86     // record.
87     return [].concat.apply([], recordFoundByID)
88 }
89
90 // searches for a record to return based off of a search value
91 // will check all columns entered to see if there is a match in any of them
92 // columnsToSearch parameter should be an array of numbers that represent the
93 // index's of the columns to be searched
94 // the search value should be a sting
95 selectRecordByField(columnsToSearch, searchValue) {
96     let arrayOfRecords = this[_csvToArray]();
97
98     arrayOfRecords.splice(arrayOfRecords.length - 1, 1);
99     return Search.binarySearch2D(arrayOfRecords, columnsToSearch, searchValue)
100 }
101
102 // will return a number for how many matches were found in that column for the
103 // searchValue
104 // columnsToSearch parameter should be an array of numbers that represent the
105 // index's of the columns
106 // to be searched. The search value should be a sting.
107 countMatches (columnsToSearched, searchValue) {
108     return this.selectRecordByField(columnsToSearched, searchValue).length;
109 }
110
111 // returns only the records found that match the searchValue
112 // unlike the selectRecordByField which returns an array of objects
113 getRecordsOnlyByField (columnsToSearch, searchValue) {
114     let arrayOfRecordsAndIndexesObj = this.selectRecordByField(columnsToSearch,
115     searchValue);
116     let arrayOfRecordsOnly = [];
117     for (let i = 0; i < arrayOfRecordsAndIndexesObj.length; i++) {
118         arrayOfRecordsOnly.push(arrayOfRecordsAndIndexesObj[i].record)
119     }
120     return arrayOfRecordsOnly;
121 }
122
123 // The generateUniqueID variable will make this a polymorphic method as it will
124 // run differently dependent on
125 // the argument that gets passed for it.
126 // the data argument should be an array of all the data contained in a record
127 addRecord(data, generateUniqueID, noTimeInTimeStamp) { // no time in time stamp
128     allows the user to make a time stamp with no time elements in it
129
130     let arrayOfRecords = this[_csvToArray]();
131     let arrayHolder = [];
132     let currentDateTime;
133
134
135     // if there is no argument passed in the generateUniqueID parameter
136     // the it's default value will be set to undefined which would be
137     // interpreted as
138     // false by the if statements. Because this method is more likely to be used
139     // to create a unique id when a record is being added, it makes sense for it
140     // to be true when
141     // the user is creating a record and leaves out that argument. Also done to
142     // aid readability as
143     // I could have named it doNotGenerateId but parameters aren't typically
144     // named like that

```

```

134
135     if (generateUniqueID === undefined) {
136         generateUniqueID = true;
137     }
138
139     if (generateUniqueID) {
140         // generate the unique id, has to be declared down here so it does not
141         // interfere with the UniqueIdGenerator class when it accesses the
142         // addRecord method
143         const ID = this[_generateID](); // generates a unique id value
144
145         if (noTimeInTimeStamp === true) { // sets appropriate date time stamp
146             dependent on argument passed in
147             currentDateTime = DateHandler.getCurrentDate();
148         } else {
149             currentDateTime = DateHandler.getCurrentDateTime();
150         }
151
152         let recordWithDateStampAndID = [ID, currentDateTime];
153         // add record
154         data.forEach((field) => {
155             recordWithDateStampAndID.push(field);
156         });
157         // console.log(recordWithDateStampAndID);
158         // removes the empty last element in the array caused by the empty new
159         // line
160         // .pop() resulted in the entire array got cleared for some reason so
161         // this is the
162         // work around
163         arrayOfRecords.splice(arrayOfRecords.length - 1);
164         arrayOfRecords.push(recordWithDateStampAndID);
165         // console.log( arrayOfRecords );
166         this[_rewriteDataToFileWithChanges](arrayOfRecords);
167     }
168
169     // used by the unique id generator private method
170     else {
171         arrayOfRecords.pop();
172         arrayHolder.push(data);
173         // takes the 2D array 'arrayOfRecords' and appends the new record to the
174         // end of it
175         arrayOfRecords.push(arrayHolder);
176         // rewrite the array of records with the appended record to the file
177         this[_rewriteDataToFileWithChanges](arrayOfRecords);
178     }
179
180     deleteRecord(id) {
181         let arrayOfRecords = Sort.bubbleSort2D(this[_csvToArray](), 0);
182         arrayOfRecords.splice(arrayOfRecords.length - 1);
183         //exception handling so program doesn't crash if the delete was not possible
184         try{
185             arrayOfRecords.splice(Search.binarySearch2D(arrayOfRecords, [0],
186             id)[0].index, 1); // deletes
187         }
188         catch(e) {
189             console.error(`record was not able to be deleted: check record with that
190             id ${id} exists\n`);
191             console.error(e)
192         }
193         // arrayOfRecords is parsed back into the file without
194         // the record with the id passed as argument
195         // unless the record was not able to be deleted, then all
196         // records will get parsed back into the original csv file
197         this[_rewriteDataToFileWithChanges](arrayOfRecords);
198     }
199
200     // arrayOfEdits format: [[ "col that needs to be edited", "new data"], ... ,

```

```

["col that needs to be edited", "new data"]]
201 // arrayOfEdits = [["1", "james"], ["2", "doug"]];
202 // example of method calling code: users.editRecord("32", [[{"id": "1", "name": "james"}, {"id": "2", "name": "doug"}]]);
203 // id should be of string type
204 editRecord(id, arrayOfEdits) {
205     // file gets turned to 2d array ad then sorted by the id column
206     let arrayOfRecords = Sort.bubbleSort2D(this[_csvToArray](), 0);
207     arrayOfRecords.splice(arrayOfRecords.length - 1); // removes last element in
208     // array because it is empty
209     // find record by id and returns a javascript object: {record: [record
210     // data], index: index record is in 2d array}
211     let recordObject = Search.binarySearch2D(arrayOfRecords, [0], id)[0];
212     // puts the object values into variables
213     let index = recordObject.index;
214     let record = recordObject.record;
215
216     // calls the makeChangesToRecord private method and then
217     // stores that new record in the place of the old record
218     arrayOfRecords[index] = this[_makeChangesToRecord](record, arrayOfEdits);
219     // rewrites the 2D array to a the csv file.
220     this[_rewriteDataToFileWithChanges](arrayOfRecords);
221 }
222
223 // method that will return the last record in the file
224 [_getLastRecordItem]() {
225     let arrayOfRecords = this[_csvToArray]();
226     let arrLength = arrayOfRecords.length;
227     // to access last record, the element has to be length - 2 because
228     // arrays start from 0 and then the other -1 comes form the fact that it
229     // counts the new line.
230     return arrayOfRecords[arrLength - 2];
231 }
232
233 // takes a 2d array as a argument, will write the 2D Array to the file.
234 // works by deleting all data contained in the file and then writing all the
235 // data in the
236 // 2D array to the text file.
237 [_rewriteDataToFileWithChanges](array2D) {
238
239     // remove all data stored in the file
240     // call back to carry out further tasks after the data has been erased
241     fs.writeFile(this[_FILE_PATH], '', (err) => {
242         if (err) {
243             console.log(err);
244         }
245         else {
246             // rewrite data to file
247             let record = [];
248             //iterates through the length of the array (number of records in the
249             //array)
250             for (let i = 0; i < array2D.length; i++) {
251                 record = array2D[i].join(","); // joins the record into a
252                 // string separated with a comma
253                 // writes a record to the file in format of:
254                 // 'comma separated record data' + "," + new line
255                 // call back to output that the data has been re-written
256                 // successfully or an error
257                 // message outlining the error
258                 fs.appendFileSync(this[_FILE_PATH], `${record}\n`);
259             }
260         }
261     });
262
263     deleteAllRecords () {
264         fs.writeFile(this[_FILE_PATH], '', (err) => {
265             console.log(err);

```

```

265     });
266 }
267
268 // column to sort by should be a number of the index of the field that will get
269 // sorted
270 // ascending should be set to true or false depending you want the 2d array to
271 // be in
272 // ascending order: true or descending order: false
273 // if the ascending parameter has no value passed as argument, it will sort it
274 // in ascending order
275 // as that is how I defined the method in the sort class.
276 sortBy (columnToSortBy, ascending) {
277     let sortedArray = Sort.bubbleSort2D(this[_csvToArray](), columnToSortBy,
278         ascending);
279     sortedArray.pop(); // gets rid of the empty element at the end of the array
280     return sortedArray;
281 }
282
283
284 [_generateID]() {
285     // could use this but there is no point of taking up extra server space so I
286     // am using a trusted unique Id generator package
287     // let idFile = new CSVHandler(_dirname + "/dataStoreFiles/IDs.csv"); // create an instance of the CSVHandler class
288     //         // let lastID = idFile.getLastRecordItem()[0]; // get last id
289     //         // number stored in the csv file
290     //         // let newID = parseInt(lastID) + 1; // set id to that value + 1
291     //         // // if the new id comes back as not a number due to there being
292     //         // no number returned from the file
293     //         // // then make new id = 1;
294     //         // if (isNaN(newID)) {
295     //             // newID = 1;
296     //         // } // only time this module wont work is when the IDs.csv file
297     //         // has no lines in it at all
298     //         // // to fix - press the return key in the csv file to add a
299     //         // blank line
300     //         // idFile.addRecord(newID, false); // add id just generated to
301     //         // the csv file
302     //         // return newID.toString();
303     return uniqueID();
304 }
305
306
307 // this private method makes changes to a single record
308 // record should be an array of a single record
309 // arrayOfEdits should be a 2D array containing arrays of changes
310 // arrayOfEdits format: [[["col that needs to be edited", "new data"], ... ,
311 // ["col that needs to be edited", "new data"]]
312 // example data: record = ["101","kieran","williams"]; edits = [[["1",
313 // "james"], ["2", "doug"]]];
314
315 [_makeChangesToRecord](record, arrayOfEdits) {
316     //initialize the variables
317     let col = "";
318     let newData = "";
319
320     arrayOfEdits.forEach((edit) => {
321         col = parseInt(edit[0]);
322         // makes sure that the column is not 0 because if it is, it will be
323         // possible
324         // to change the id field of a record which is something that should not
325         // ever happen
326         if (col !== 0) {
327             newData = edit[1];
328             record[col] = newData;
329         }
330     });
331     return record;
332 }

```

```
323
324 }
325
326 module.exports = CSVHandler;
327
```

# DateHandler.js

```
1  const moment = require('moment');
2
3  const _ddmm-yyyyToStandardDateTimeCheck = Symbol('ddmm-yyyyToStandardDateTimeCheck');
4
5  class DateHandler {
6
7      // creates a time stamp from a date and a time
8      // returns a time stamp that moment.js or the
9      // Date class js has can work with. arguments to be type string
10     // date => dd/mm/yyyy
11     // time is optional, and the seconds value is also optional
12     // time => hh:mm:ss || hh:mm
13     static createTimeStamp (date, time) {
14         let day, month, year, hour, minute, second; // declares variables
15         if (time === undefined) {time = "00:00:00"} // if time is passed as an
16         argument, time is set to "00:00:00"
17         if (time.length === 5) {time = `${time}:00`} // if time is only hh:mm => the
18         ss is set to 00
19         date = date.split("/"); // splits the date string up into a string array
20         ["dd", "mm", "yyyy"]
21         time = time.split(":"); // splits the time string up into a string array
22         ["hh", "mm", "ss"]
23         day = date[0];
24         month = date[1];
25         year = date[2];
26         hour = time[0];
27         minute = time[1];
28         second = time[2];
29         return `${year}-${month}-${day}T${hour}:${minute}:${second}+00:00`;
30     }
31
32     // uses the moment library to get the current date and time
33     // outputs a string in the standard date time stamp
34     static getCurrentDateTime () {
35         return moment().format();
36     }
37
38     // uses the moment library to get the current date and time
39     // outputs a string in the standard date time stamp and then
40     // turns the dateTimeStamp into the standard date time stamp but with
41     // only the date information and the time data will all be set to 0
42     static getCurrentDate() {
43         // returns the result of the current time stamp turned into dd/mm/yyyy,
44         // turned into
45         // the standard date time stamp but with not time information
46         return
47             this.createTimeStamp(this.standardDateTo_Date_DDMMYYYY(this.getCurrentDateTime
48                 ()));
49     }
50
51     // pass a date time stamp as argument format: YYYY-MM-DD'T'HH:MM:SS+MS:mS
52     // eg: '2019-02-02T22:52:49+00:00'
53     // will output the date as dd/mm/yyyy
54     static standardDateTo_Date_DDMMYYYY(dateTimeStamp) {
55         // could use moment(dateTimeStamp).format("DD/MM/YYYY");
56         // but will use my own algorithm to extract the date instead
57
58         // splits argument into an array containing the date at index 0 & the time
59         // at index 1
60         // [0] at the end gets us just the date element, split the date up into an
61         // array ["yyyy", "mm", "dd"]
62         // could use .reverse and then .join("/") which would give me "dd/mm/yyyy"
63         // but am writing them process myself for algorithm marks
64         let date = timeStamp.split("T")[0].split("-");
65
66         // self invoking function that reverses the date and then stores the result in
67         // the dateReversed variable
68         let dateReversed = () => {
69             let temp = "";
70             let j = date.length -1;
71             for (let i = 0; i < date.length / 2; i++, j--) {
72                 temp = date[i];
73                 date[i] = date[j];
74             }
75         }
76
77         dateReversed();
78
79         return date;
80     }
81
82     static standardDateTo_DDMMYYYY(date) {
83         // could use moment(date).format("DD/MM/YYYY");
84         // but will use my own algorithm to extract the date instead
85
86         // splits argument into an array containing the date at index 0 & the time
87         // at index 1
88         // [0] at the end gets us just the date element, split the date up into an
89         // array ["yyyy", "mm", "dd"]
90         // could use .reverse and then .join("/") which would give me "dd/mm/yyyy"
91         // but am writing them process myself for algorithm marks
92         let date = date.split("T")[0].split("-");
93
94         // self invoking function that reverses the date and then stores the result in
95         // the dateReversed variable
96         let dateReversed = () => {
97             let temp = "";
98             let j = date.length -1;
99             for (let i = 0; i < date.length / 2; i++, j--) {
100                 temp = date[i];
101                 date[i] = date[j];
102             }
103         }
104
105         dateReversed();
106
107         return date;
108     }
109 }
```

```

65         date[j] = temp;
66     }
67     return date;
68 })();
69
70     return dateReversed.join("/");
71 }
72
73 // pass a date time stamp as argument format: YYYY-MM-DD'T'HH:MM:SS+MS:mS
74 // eg: '2019-02-02T22:52:49+00:00'
75 // will output the time as hh:mm:ss if the withSeconds argument is == true
76 // if it is not then the time will be outputted as hh:mm
77 static standardDateTo_Time_HHMMSS(dateTimeStamp, withSeconds) {
78
79     // splits argument into an array containing the date at index 0 & the time
80     // at index 1
81     // [1] at the end gets us just the time element. split the time element up
82     // into an array of characters,
83     // the array of characters will be:
84     // ["h","h",":","m","m",":","s","s","+","m","s",":","m","s"]
85     // in terms of format and length. if withSeconds option parameter is set to
86     // true, then all the characters
87     // in the array after the 7th index will get removed. if withSeconds is not
88     // true, elements after 4th index will get removed
89     let time = timeStamp.split("T")[1].split("");
90     let indexToRemoveFrom;
91     if (withSeconds !== true) {indexToRemoveFrom = 5}
92     else { indexToRemoveFrom = 8 }
93     //self invoking function that truncates the array of unwanted data will be
94     // returned
95     return () => {
96         let outputArr = []; // declare output array
97         // append elements from time to outputArray up to the indexToRemoveFrom
98         for (let i = 0; i < indexToRemoveFrom; i++) { outputArr.push(time[i]) }
99         return outputArr.join(""); //joins the array of characters together into
100        // a string
101    }();
102
103 // checks if the dateToCheck is equal to the current date
104 // dateToCheck should be type string in either dd/mm/yyyy format or standard
105 // date time format
106 static isToday (dateToCheck) {
107     // gets the users current date in a time stamp. Then, that time stamp
108     // gets turned into a dd/mm/yyyy format
109     let currentDate =
110         this.standardDateTo_Date_DDMMYYYY(this.getCurrentDateTime());
111     // checks to see if the date is in standard time or dd/mm/yyyy format
112     // if it is in standard time format then the length of the string will not
113     // be = to 10
114     // in that case the dateToCheck is converted into dd/mm/yyyy format
115     if (dateToCheck.length !== 10) { dateToCheck =
116         this.standardDateTo_Date_DDMMYYYY(dateToCheck) }
117     // compares the 2 strings, if the dates are equal true is returned.
118     // Otherwise, false is returned
119     return (dateToCheck === currentDate);
120 }
121
122 // checks if the dateToCheck's month or year is equal to the current month or year
123 // dateToCheck should be type string in either dd/mm/yyyy format or standard
124 // date time format
125 // monthOrYear option should be set to 'month' or 'year'
126 static isThisMonthOrYear (dateToCheck, monthOrYearOption) {
127     // gets the users current date in a time stamp. Then, that time stamp
128     // gets turned into a dd/mm/yyyy format
129     let currentDate =
130         this.standardDateTo_Date_DDMMYYYY(this.getCurrentDateTime());
131     // checks to see if the date is in standard time or dd/mm/yyyy format
132     // if it is in standard time format then the length of the string will not
133     // be = to 10

```

```

123     // in that case the dateToCheck is converted into dd/mm/yyyy format
124     dateToCheck = this[_ddmmyyyyToStandardDateTimeCheck] (dateToCheck);
125
126     // splits the datetime by "T" and then take index 0 of the array to get
127     // yyyy-mm-dd
128     // split the yyyy-mm-dd by "-" to get array of ["yyyy", "mm", "dd"]
129     dateToCheck = dateToCheck.split("T") [0].split("-");
130     currentDate = currentDate.split("T") [0].split("-");
131
132     // if the the option 'year' is not passed in as an argument the
133     // months will be compared by default
134     if (monthOrYearOption !== 'year') {
135         // returns the evaluation result of the equality between the month data
136         // from both arrays
137         return dateToCheck[1] === currentDate[1];
138     }
139     else {
140         // returns the evaluation result of the equality between the year data
141         // from both arrays
142         return dateToCheck[2] === currentDate[2];
143     }
144
145
146
147     // can pass in date as standard date time format or as dd/mm/yyyy as type string
148     // objectContainingTheAddingToMake should be type object can be a negative
149     // number to subtract days
150     // objectContainingTheAddingToMake: examples: {days : 6} or {months: 9} or
151     // {years : 9}
152     // can even do any combination: examples: {days : 7, months : 9} or {days : 8,
153     // months: 9, years : 2}
154     static addToDate (date, objectContainingTheAddingToMake) {
155         // if date is in dd/mm/yyyy format it gets changed into the standard date
156         // time format the moment.js library can work with
157         date = this[_ddmmyyyyToStandardDateTimeCheck] (date);
158         // uses the moment library to make relevant changes to the date. The result
159         // then gets formatted to the standard
160         // date format by using the .format()
161         return moment(date).add(objectContainingTheAddingToMake).format();
162     }
163
164
165
166     // can pass in date as standard date time format or as dd/mm/yyyy as type string
167     // objectContainingTheAddingToMake should be type object can be a negative
168     // number to subtract days
169     // objectContainingTheAddingToMake: examples: {days : 6} or {months: 9} or
170     // {years : 9}
171     // can even do any combination: examples: {days : 7, months : 9} or {days : 8,
172     // months: 9, years : 2}
173     static subtractFromDate (date, objectContainingTheSubtractionsToMake) {
174         // if date is in dd/mm/yyyy format it gets changed into the standard date
175         // time format the moment.js library can work with
176         date = this[_ddmmyyyyToStandardDateTimeCheck] (date);
177         // uses the moment library to make relevant changes to the date. The result
178         // then gets formatted to the standard
179         // date format by using the .format()
180         return moment(date).subtract(objectContainingTheSubtractionsToMake).format();
181     }

```

```
182
183
184
185 // method checks to see if the dateToBeChecked is in the past or in the future.
186 // dateToBeChecked should type string
187 // The argument should be in the standard date format for greater
188 // accuracy as time will also be accounted for then but dd/mm/yyyy format will
189 // work
190 // the method will return true if the date has passed and false if the date is
191 // in the future
192 static hasDatePassed (dateToBeChecked) {
193     let currentDateTime = this.getCurrentDateTime();
194     // if date is in dd/mm/yyyy format it gets changed into the standard date
195     // time format the moment.js library can work with
196     dateToBeChecked = this[_ddmmyyyyToStandardDateTimeCheck] (dateToBeChecked);
197
198     // returns the evaluation of the dates
199     return currentDateTime > dateToBeChecked;
200 }
201
202 // private method to check if a date is dd/mm/yyyy or standard date time stamp
203 // if the date is dd/mm/yyyy format then the date gets changed to the equivalent
204 // in
205 // the standard date time stamp
206 static [_ddmmyyyyToStandardDateTimeCheck] (date) {
207     if (date.length === 10) { date = DateHandler.createTimeStamp(date) }
208     return date;
209 }
210
module.exports = DateHandler;
```

# Search.js

```

1  const Sort = require("./Sort"); // allows the use of sort methods in this file
2
3  function Search(){}
4
5
6  // creates private method
7  const _binarySearch2DFinder = Symbol('binarySearch2DFinder');
8  const _checkBoundsForMatches = Symbol('checkBoundsForMatches');
9
10
11
12 // searches for a value and returns the index the element was found at
13 // or returns -1 if the search value was not found in the array
14 // ONLY WORKS ON SORTED ARRAYS
15 Search.binarySearch = (array, searchValue) => {
16     // makes sure that the array is sorted before the search is done
17     array = Sort.bubbleSort(array);
18     let first = 0;
19     let last = array.length;
20     let middle = Math.floor((first + last)/2);
21     let found = false;
22     while(!found || (first <= last)){
23         if(array[middle] === searchValue){
24             found = true;
25             return middle;
26         }
27         if(searchValue < array[middle]){
28             last = middle - 1;
29             middle = Math.floor((first + last)/2);
30         }
31         if(searchValue > array[middle]){
32             first = middle + 1;
33             middle = Math.floor((first + last)/2);
34         }
35     }
36     return -1;
37 };
38
39 Search[_checkBoundsForMatches] = (array, middle, col, searchValue) => {
40     let matchAtBoundsFoundFlag = true; // set to true so that the loop will start
41     let recordsFound = [];
42     let upperBounds = middle;
43     let lowerBounds = middle;
44
45     // the algorithm works in the binary search as any other matches will be found
46     // either side of the initial match
47     // due to a binary search only working on sorted arrays which will result in the
48     // matches in conjunction in the array
49
50     // loop through upper values
51     while (matchAtBoundsFoundFlag === true){
52
53         // set to false so that if no matches are found on the upper bounds from the
54         // first match
55         // the loop will end.
56         matchAtBoundsFoundFlag = false;
57
58         // checks that if the upperBounds was to get increased by one,
59         // the program will not crash due to the record not being in the bounds of
60         // the array
61         if ((upperBounds + 1) > (array.length - 1)) {
62             break; // terminate the loop
63         }
64
65         if (array[upperBounds + 1][col] === searchValue) { // checks to see if the
66             array element above is a match to the search value
67             matchAtBoundsFoundFlag = true; // set back to true so that the loop will
68             run again to check for more matches.
69             upperBounds++; // increase value stored in upperBounds variable by 1
70
71             // match to the search value was found so the recordsFound
72             // array gets the record and index of record in a javascript object
73             // appended to the end of it

```

```

67         recordsFound.push({ record : array[upperBounds], index : upperBounds });
68     }
69 }
70
71 matchAtBoundsFoundFlag = true; // reset the flag value so the next loop can run
72
73 // loop through lower values
74 while (matchAtBoundsFoundFlag === true) {
75
76     // set to false so that if no matches are found on the lower bounds of the
77     // first match
78     // the loop will end.
79     matchAtBoundsFoundFlag = false;
80
81     // checks that if the lowerBounds was to get decreased by one,
82     // the program will not crash due to the record not being in the bounds of
83     // the array
84     if ((lowerBounds - 1) < (0)) {
85         break; // terminate the loop
86     }
87
88     if (array[lowerBounds - 1][col] === searchValue) {
89         matchAtBoundsFoundFlag = true; // set back to true so that the loop will
90         // run again to check for more matches.
91         lowerBounds--; // increase value stored in lowerBounds variable by 1
92
93     }
94 }
95
96 return recordsFound;
97 };
98
99 // purpose of method is to find a match with the searchValue in a set column
100 Search[_binarySearch2DFinder] = (array, col, searchValue) => {
101     let first = 0;
102     let last = array.length;
103     let middle = Math.floor((first + last)/2);
104     let upperAndLowerBoundMatches = [];
105     let recordsFound = [];
106
107
108
109     // loop that will keep going until last > first or the matching records are found
110     // the array[middle][col] --> middle is the row that is currently under
111     // inspection, col is a constant
112     while (first <= last) {
113
114         // to avoid a crash, the program will check to see if middle is greater than
115         // the index of the last
116         // record and that if it is a negative value. In either instance the loop
117         // will be terminated
118         if (middle > array.length - 1 || middle < 0){
119             break;
120         }
121
122         if(array[middle][col] === searchValue){
123             //pushes the array and an index contained in a javascript object type
124             recordsFound.push({ record : array[middle], index : middle }); // adds
125             the record into records found array
126
127             // finds any matches in the upper and lower bounds of the
128             upperAndLowerBoundMatches = Search[_checkBoundsForMatches](array,
129             middle, col, searchValue);
130
131             upperAndLowerBoundMatches.forEach((match) => {
132                 recordsFound.push(match);
133             });
134             break; // gets out of loop
135         }
136     }
137 }
```

```

131
132     // terminates the loop if the middle value is 0 as it will cause a crash
133     // crash is due to the fact that middle will turn into a negative value
134     if (middle === 0) {
135         break;
136     }
137
138     if(searchValue < array[middle][col]){
139         last = middle - 1;
140         middle = Math.floor((first + last)/2);
141     }
142     if(searchValue > array[middle][col]){
143         first = middle + 1;
144         middle = Math.floor((first + last)/2);
145     }
146 }
147
148 if (recordsFound.length > 0){
149     return recordsFound;
150 }
151 };
152
153
154
155
156 // returns a 2D array of matching records
157 // columnsToBeSearched should be an array of integers
158 // that determines what columns are to be searched
159 Search.binarySearch2D = (array, columnsToBeSearched, searchValue) => {
160
161     let recordsFoundInAllColumnsSearched = [];
162     let recordsFoundInOneColumn = [];
163     let col = 0;
164
165     // if there is only one record in the array then there is no point in running
166     // any more code
167
168     for (let i = 0; i < columnsToBeSearched.length; i++) { // goes through all
169     // columns that need to be searched
170
171         col = columnsToBeSearched[i]; // col is made equal to the i index of the
172         // columnsToBeSearched array
173
174         // sorts the array with regard to the column that is being searched on
175         array = Sort.bubbleSort2D(array, col);
176
177         recordsFoundInOneColumn = Search[_binarySearch2DFinder](array, col,
178         searchValue);
179
180         // if statement to check that what got returned is an
181         // array of records found and not just undefined
182         if (recordsFoundInOneColumn !== undefined){
183             // goes through every record in the recordsFoundInOneColumnArray
184             recordsFoundInOneColumn.forEach((currentRecord) => {
185                 recordsFoundInAllColumnsSearched.push(currentRecord);
186             });
187         }
188     }
189 }
190
191 return recordsFoundInAllColumnsSearched;
192
193
194 //searches for a value and returns the index the element was found at
195 //or returns -1 if the search value was not found in the array
196
197 Search.linearSearch = (array, searchValue) => {
198     let found = false;
199     let i = 0;

```

```
200 let length = array.length;
201 while((!found) || (i < length)){
202     if(array[i] === searchValue){
203         found = true;
204         return i;
205     } else {
206         i++;
207     }
208 }
209 return -1;
210 };
211
212 Search.getRecords = (ArrayOfObjectsContainingRecordsWithIndexes) => {
213     let arrayOfRecords = [];
214     ArrayOfObjectsContainingRecordsWithIndexes.forEach((recordWithIndex) => {
215         arrayOfRecords.push(recordWithIndex.record);
216     });
217     return arrayOfRecords;
218 };
219
220 module.exports = Search;
221
222
223
```

# Sort.js

```

1
2
3 function Sort (){}
4
5 const _comparisonOperator = Symbol('comparisonOperator');
6
7
8 // private function used so that dynamic comparisons can be made
9 // params should take a < or > type string in the comparissonOp
10 // value1 and value2 should be the two values that are going to be compared.
11 Sort[_comparisonOperator] = (comparisonOp, value1, value2) => {
12   if(comparisonOp === "<") { return value1 < value2 }
13   if(comparisonOp === ">") { return value1 > value2 }
14 };
15
16 // these methods can be considered an imitation of static methods
17
18 // sorts the array in ascending or descending order
19 // depending on if the ascending variable is set to true
20 // or false, and returns the sorted array
21 // THE ASCENDING PARAMETER COULD BE CONSIDERED POLYMORPHISM AS THE METHODS
22 // FUNCTIONALITY DIFFERS DEPENDENT ON
23 // WHETHER THE ARGUMENT PASSED SETS IT TO FALSE OR IS LEFT ALONE/ SET TO TRUE
24 Sort.bubbleSort = (array, ascending) => {
25
26   if (ascending === undefined) ascending = true;
27   // if no ascending argument is passed, method sorts array in ascending order
28
29   let n = array.length;
30   let sorted = false;
31   let temp = 0;
32   let comparisonOp = "";
33
34   if (ascending) comparisonOp = ">";
35   if (!ascending) comparisonOp = "<";
36
37   while(!sorted){
38
39     sorted = true;
40     for(let i = 0; i < n; i++){
41
42       // if(array[i] > array[i+1]) is for ascending
43
44       // uses the comparisonOperator method to make this
45       // method dynamic in sorting both ascending and descending
46       if (Sort[_comparisonOperator](comparisonOp,
47         array[i], array[i+1])) {
48
49         temp = array[i];
50         array[i] = array[i+1];
51         array[i+1] = temp;
52         sorted = false;
53       }
54     }
55
56   return array;
57 };
58
59 // works by comparing a set lot of columns from different
60 // records against each other. Swaps the records if the condition
61 // for a swap to be made is met (condition is determined by the boolean
62 // argument "ascending")
63 // column to sort by should be type number
64 // ascending should be boolean or not entered
65 Sort.bubbleSort2D = (array2D, columnToSortBy, ascending) => {
66
67   if (ascending === undefined) ascending = true;
68   // if no ascending argument is passed, method sorts array in ascending order
69
70   let n = array2D.length;
71   let sorted = false;
72   let temp = [];

```

```
73 let comparisonOp = "";
74
75 if (ascending) comparisonOp = ">";
76 if (!ascending) comparisonOp = "<";
77
78
79 while(!sorted){
80     sorted = true;
81     for(let i = 0; i < n -1 ; i++){
82
83         // if(array[i] > array[i+1]) is for ascending
84
85         // uses the comparisonOperator method to make this
86         // method dynamic in sorting both ascending and descending
87         if (Sort[_comparisonOperator](comparisonOp,
88             array2D[i][columnToSortBy], array2D[i+1][columnToSortBy])) {
89
90
91             temp = array2D[i];
92             array2D[i] = array2D[i+1];
93             array2D[i+1] = temp;
94             sorted = false;
95
96
97         }
98     }
99 }
100
101 return array2D;
102 };
103
104
105
106 module.exports = Sort;
107
108
109
```

# Public Directory Contents

# Scripts

# Directory

# Contents

# AnalyticsScripts.js

```

1 // store the graph canvas elements into variables
2 let clientsGained = document.querySelector('#clients-gained').getContext('2d');
3 let clientsLost = document.querySelector('#clients-lost').getContext('2d');
4
5
6 // // dummy test data for the graphs
7 // let clientsGainedLabels = ['January', 'February', 'March', 'April', 'May',
8 // 'June', 'July', 'August', 'September', 'October', 'November', 'December'];
9 // let clientsGainedDataPoints = [20, 32, 34, 45, 43, 35, 24, 21, 14, 12, 19, 16];
10 // //
11 // let clientsLostLabels = ['January', 'February', 'March', 'April', 'May', 'June',
12 // 'July', 'August', 'September', 'October', 'November', 'December'];
13 // let clientsLostDataPoints = [20, 32, 45, 23, 54, 21, 25, 31, 45, 17, 11, 8];
14
15
16 // when time scale option gets selected the form's action gets changed to the
17 // selected option's id
18 function changeAction() {
19     let timeScale = $('#time-scale').val();
20     console.log(timeScale);
21     document.querySelector("#time-scale-form").action = "/analytics/" + timeScale;
22     // line that changes the action attribute of the time scale form
23 }
24
25
26
27 // creates the graph for clients gained using the chart.js library
28 let clientsGainedGraph = new Chart(clientsGained, {
29     type: 'line',
30     data: {
31         labels: clientsGainedLabels,
32         datasets: [{
33             label: 'Number Of Clients Gained',
34             data: clientsGainedDataPoints,
35             backgroundColor: 'lightgreen',
36             borderWidth: 1,
37             borderColor: '#777',
38         }]
39     },
40     options: {
41         responsive: true,
42         maintainAspectRatio: false,
43         title: {
44             display: true,
45             text: 'Clients Gained',
46             fontSize: 25,
47         },
48         legend: {
49             display: false
50         },
51     },
52 });
53
54 // creates the graph for clients lost using the chart.js library
55 let clientsLostGraph = new Chart(clientsLost, {
56     type: 'line',
57     data: {
58         labels: clientsLostLabels,
59         datasets: [{
60             label: 'Number Of Clients Lost',
61             data: clientsLostDataPoints,
62             backgroundColor: 'lightcoral',
63             borderWidth: 1,
64             borderColor: '#777',
65         }]
66     },
67     options: {
68         responsive: true,
69         maintainAspectRatio: false,

```

```
70     title: {
71         display: true,
72         text: 'Clients Lost',
73         fontSize: 25,
74     },
75     legend:{  
76         display: false  
77     },  
78 }  
79 );  
80  
81 // // creates the graph for number of clients using the chart.js library  
82 // let numberOfClientsGraph = new Chart(numberOfClients , {  
83 //     type: 'line',  
84 //     data: {  
85 //         labels: numberOfClientsLabels,  
86 //         datasets:[{  
87 //             label:'Number Of Clients',  
88 //             data: numberOfClientsDataPoints,  
89 //             backgroundColor: 'lightblue',  
90 //             borderWidth: 1,  
91 //             borderColor: '#777',  
92 //         }]  
93 //     },  
94 //     options: {  
95 //         responsive: true,  
96 //         maintainAspectRatio: false,  
97 //         title: {  
98 //             display: true,  
99 //             text: 'Number Of Clients',  
100 //             fontSize: 25,  
101 //         },  
102 //         legend:{  
103 //             display: false  
104 //         },  
105 //     }  
106 // });
```

# DashboardScripts.js

Kieran Marcus Williams

```
1 // function that prevents the enter/ return key from being used
2 function returnKeyBlocker(e) {
3     // sets e to be equal to either the event passed as argument or the window.event
4     // object for IE support
5     e = e || window.event;
6     // gets the key of e but if that is deprecated or isn't supported by the browser
7     // then then charCode is used
8     let keyCode = e.keyCode || e.charCode;
9     return keyCode !== 13;
}
```

# IndividualContactScripts.js

```
1 // function that prevents the enter/ return key from being used
2 function returnKeyBlocker(e) {
3     // sets e to be equal to either the event passed as argument or the window.event
4     // object for IE support
5     e = e || window.event;
6     // gets the key of e but if that is deprecated or isn't supported by the browser
7     // then then charCode is used
8     let keyCode = e.keyCode || e.charCode;
9     return keyCode !== 13;
}
```

# Stylesheets

## Directory

### Contents

# AnalyticsStyles.css

```
1 .top-half-of-analytics {
2     height: 450px;
3     display: flex;
4     justify-content: space-around;
5 }
6
7 .bottom-half-of-analytics {
8     display: flex;
9     justify-content: space-between;
10}
11
12 .middle-bottom-analytics {
13     width: 250px;
14 }
15
16 .custom-select {
17     width: 150px;
18 }
19
20 .time-scale-separator {
21     display: flex;
22     justify-content: space-between;
23 }
24
25 .conversion-rate-holder {
26     padding-top: 20px;
27     width: 400px;
28     display: flex;
29     justify-content: space-between;
30 }
31
32
33 .remove-margin {
34     margin: 0;
35 }
36
37 .conversion-rate-table {
38     width: 100%;
39     height: 75px;
40     border: 1px solid black;
41 }
42
43 .row-of-buttons {
44     width: 250px;
45     margin: 10px;
46     display: flex;
47     justify-content: space-between;
48 }
49
50 .button-holder-analytics {
51     width: fit-content;
52 }
53
54 td {
55     border: 1px solid black;
56     text-align: center;
57     padding: 10px;
58 }
59
60 body {
61     margin: 40px 80px;
62 }
63
64
65 canvas {
66     width: 35vw;
67 }
68
69
70
71
72
73
```

# ContactsListStyles.css

```
1 body {
2     margin: 40px 40px;
3 }
4
5 #modify-list-results-form {
6
7     width: fit-content;
8     padding: 10px;
9 }
10
11 #contact-management {
12     margin-top: 30px;
13     width: fit-content;
14     padding: 10px;
15 }
16
17 #left-side-container {
18
19     height: 700px;
20     margin: 0;
21     display: flex;
22     flex-direction: column;
23     justify-content: space-between;
24 }
25
26 #right-side-container {
27     margin-right: 2%;
28     height: 600px;
29     width: 1000px;
30     overflow: scroll;
31 }
32
33
34 .table {
35     position: inherit;
36     top: 0;
37     bottom: 0;
38     left: 0;
39     right: 0;
40     width: 100%;
41 }
42
43
44 .holder {
45     margin: 30px 0;
46     width: 100%;
47     display: flex;
48     justify-content: space-between;
49 }
50
51 .submit-col {
52     display: flex;
53 }
```

# CreateContactsStyles.css

```
1 body {
2     margin: 40px 40px;
3 }
4
5 .container {
6     width: 800px;
7 }
8
9 .form {
10    margin-top: 20px;
11    padding: 30px;
12 }
13
14 .form-control {
15     border: 1px solid grey;
16 }
17
18 .custom-select {
19     border: 1px solid grey;
20     height: 40px;
21 }
22
23 .btn-submit {
24     position: relative;
25     margin-top: 40px;
26     margin-left: 20%;
27     margin-right: 20%;
28     width: 60%;
29     height: 50px;
30 }
```

# DashboardStyles.css

```
1 #dashboard-separator {
2   display: flex;
3   flex-direction: row;
4   justify-content: space-between;
5 }
6
7 #schedule-task-form {
8   margin: 15px;
9 }
10
11 #today {
12   margin: 15px;
13 }
14
15 #future {
16   margin: 15px;
17 }
18
19 .scheduled-display {
20   display: flex;
21 }
22
23 .trash-button {
24   color: darkgray;
25 }
26
27 .trash-button:hover {
28
29   color: grey;
30 }
31
32
33
34 .dashboard-slots {
35   border: solid 1px grey;
36   padding: 30px 25px;
37   width: 400px;
38   height: 700px;
39   overflow-y: scroll;
40   word-wrap: break-word;
41 }
42
43
44 ul {
45   list-style: none;
46   margin: 0;
47   padding: 0;
48 }
49
50 textarea {
51   resize: none;
52 }
53
54 body {
55   margin: 40px 80px;
56 }
```

# EditContactStyles.css

```
1 body {
2     margin: 40px 40px;
3 }
4
5 .container {
6     width: 800px;
7 }
8
9
10
11 .form-control {
12     border: 1px solid grey;
13 }
14
15 .custom-select {
16     border: 1px solid grey;
17     height: 40px;
18 }
19
20 .btn-submit {
21     position: relative;
22     margin-top: 40px;
23     margin-left: 20%;
24     margin-right: 20%;
25     width: 60%;
26     height: 50px;
27 }
```

# HomeStyles.css

```
1 body {
2     margin: 20px 40px;
3 }
4
5 h1 {
6     margin-top: 60px;
7 }
8
9 .link-separator {
10    margin-left: 20px;
11 }
12
13 #login-btn {
14     margin-right: 40px;
15 }
16
17
18
19
20
21
22
```

# IndividualContactStyles.css

```
1 #halfway-separator {
2     display: flex;
3     justify-content: space-between;
4 }
5
6 #left-side {
7     display: flex;
8     flex-direction: column;
9     justify-content: space-around;
10 }
11
12 .contact-details{
13     width: fit-content;
14 }
15
16 .add-note-section {
17     width: fit-content;
18 }
19
20 .btn-submit-add-note {
21     margin-top: 10px;
22     float: right;
23 }
24
25 .btn-container-extra-notes {
26     float: right;
27     margin-top: 10px;
28     width: 40%;
29 }
30
31 #extra-notes-btn-group {
32     display: flex;
33     justify-content: space-evenly;
34 }
35
36 .btn-submit-extra-notes {
37 }
38
39 .btn-submit-extra-notes-encrypt {
40 }
41
42 span {
43     font-weight: bold;
44     font-size: x-large;
45     margin-right: 1em;
46 }
47
48 .contact-management-buttons {
49     margin-top: 30px;
50     width: 40%;
51     display: flex;
52 }
53
54 .btn-edit-contact {
55     margin-right: 20%;
56 }
57
58 .table {
59     position: inherit;
60     top: 0;
61     bottom: 0;
62     left: 0;
63     right: 0;
64     width: 100%;
65 }
66
67 .td-borededer {
68     border-left: 1px solid lightgrey;
69 }
70
71 .td-borededer {
72     border-left: 1px solid lightgrey;
73 }
```

```
74  
75 .thead-grey {  
76     background-color: lightgrey;  
77 }  
78  
79  
80 #table-holder-notes{  
81     width: 680px;  
82     height: 400px;  
83     overflow: scroll;  
84 }  
85  
86 textarea {  
87     resize: none;  
88 }  
89  
90 body {  
91     margin: 40px 60px;  
92 }  
93
```

# NavbarLoggedInStyles.css

```
1  .separator {
2      width: 100%;
3      display: flex;
4      justify-content: space-between;
5  }
6
7
8  .nav-item{
9      font-size: large;
10     margin: 0 20px;
11 }
12
13 .small-margin-right {
14     margin-left: 20px;
15 }
16
17 .mid-nav{
18     width: 50%;
19 }
20
21 .spread-nav-items{
22     width: 100%;
23     display: flex;
24     justify-content: space-around;
25 }
```

# SettingsStyles.css

```
1 body {  
2     margin: 20px 40px;  
3 }  
4  
5 h1 {  
6     margin-top: 60px;  
7 }
```

# Routes

# Directory

# Contents

# contactsRoutes.js

```

1 const
2   express = require("express"),
3   router = express.Router(),
4   CSVHandler = require("../classes/CSVHandler"),
5   StringUtility = require('../utilityMethods/StringUtility'),
6   DateHandler = require("../classes/DateHandler"),
7   Sort = require("../classes/Sort"),
8   Encryption = require("../classes/Encryption"),
9   PasswordValidation = require("../classes>PasswordValidation"),
10  ContactsValidation = require("../classes/ContactsValidation"),
11  AddNoteValidation = require("../classes/AddNoteValidation");
12
13
14 ///////////////////////////////////////////////////////////////////
15 ///////////////////////////////////////////////////////////////////
16 // GET ROUTES
17
18 ///////////////////////////////////////////////////////////////////
19
20
21
22
23
24 //renders the contacts list page that outputs all the contacts in a table
25 router.get("/", (req, res) => {
26   let contactsFile = new CSVHandler(__dirname + '/dataStoreFiles/contacts.csv');
27   let allContactRecords = contactsFile.getAllRecords(); // gets all the contact
28   records and stores in a 2D array
29   let message = req.session.message;
30   let errorMessage = req.session.errorMessage;
31
32   // an object with a property containing the 2D Array is passed to the ejs template
33   // searchValue: undefined allows me to use the same ejs page in the
34   .post("/search") route
35   res.render("ContactsList", {contactRecords: allContactRecords, titleAddition:
36   undefined, flashMessage: message, errorMessage: errorMessage},
37   function (err, html) {
38     req.session.message = undefined;
39     req.session.errorMessage = undefined;
40     res.send(html);
41   });
42 });
43
44
45 // downloads the contacts file then redirects to the base route
46 router.get("/download", (req, res) => {
47
48   res.download(__dirname + '/dataStoreFiles/contacts.csv',
49   'contactRecords' + ':' + DateHandler.getCurrentDateTime() + '.csv', // sets
50   the name to be contacts records with the current date in with the title
51   function(err){
52     if (err) {
53       console.log(err); // logs the error to the console
54     } else {
55       console.log("downloaded: " + __dirname + '/dataStoreFiles/contacts.csv');
56     }
57   });
58 });
59
60
61
62 // renders the individual contact page for the contact with the id in the url
63 router.get("/viewContact:id", (req, res) => {
64
65   let contactsFile = new CSVHandler(__dirname + '/dataStoreFiles/contacts.csv');
66   let notesFile = new CSVHandler(__dirname + '/dataStoreFiles/contactsNotes.csv');
67   let extraNotesFile = new CSVHandler(__dirname +
68   '/dataStoreFiles/contactsExtraNotes.csv');
69   let id = req.params.id;
70   // find contact by id to get the contacts record

```

```

67 let contactDetailsRecord = contactsFile.selectRecordByID(id);
68
69
70 // gets all the notes relating to the current contact that is being viewed
71 // stored in the contactNotes variable as a 2D array with each inner array being
72 // a record from the notes csv file
73 let contactsNotes = notesFile.getRecordsOnlyByField([3], id);
74
75
76 let contactsExtraNotes = extraNotesFile.getRecordsOnlyByField([2], id); // gets
77 // the contacts extra notes
78
79 contactsNotes = Sort.bubbleSort2D(contactsNotes, [1], false); // sorts the notes
80 // ascending by their date time stamp
81
82
83 if (contactsExtraNotes.length !== 0) { // if there is data in the contacts extra
84 // notes 2d array
85 contactsExtraNotes = contactsExtraNotes[0][3]; // gets the data from the
86 // extra notes data column
87 }
88
89 if (req.session.decryptedData !== undefined) {
90 contactsExtraNotes = req.session.decryptedData;
91 }
92
93
94
95 let flashMessage = req.session.message;
96 let errorMessage = req.session.errorMessage;
97
98 res.render("ViewIndividualContact",
99 {
100   contactDetails: contactDetailsRecord[0].record,
101   flashMessage: flashMessage,
102   errorMessage: errorMessage,
103   contactNotes: contactsNotes,
104   extraNotes: contactsExtraNotes
105 },
106 (err, html) => {
107   req.session.message = undefined;
108   req.session.errorMessage = undefined;
109   req.session.decryptedData = undefined;
110   res.send(html);
111 }
112 );
113
114 //renders the create contact page
115 router.get("/new", (req, res) => {
116   let message = req.session.message; // if there is a message in the session
117   // object then it will be outputted to the user
118   let errorMessage = req.session.errorMessage;
119
120   let businessName = req.session.businessName;
121   let firstName = req.session.firstName;
122   let lastName = req.session.lastName;
123   let email = req.session.email;
124   let telephone = req.session.telephone;
125   let city = req.session.city;
126   let postcode = req.session.postcode;
127   let address = req.session.address;
128
129   let valuesStoredInSessionFromBody = [businessName, firstName, lastName, email,
130   telephone, city, postcode, address];
131
132   res.render("CreateContact",
133   {

```



```

200
201 // post route to add a record to the contacts page
202
203 router.post("/", (req, res) => {
204     let contactsFile = new CSVHandler(__dirname + '/dataStoreFiles/contacts.csv');
205
206     // stores the values from the text fields into variables
207     // the some of the values get capitalized to aid the data maintainability
208     let businessName = StringUtility.capitalizeString(req.body.businessName);
209     let firstName = StringUtility.capitalizeString(req.body.firstName);
210     let lastName = StringUtility.capitalizeString(req.body.lastName);
211     let email = req.body.email;
212     let telephone = req.body.telephone;
213     let city = StringUtility.capitalizeString(req.body.city);
214     let postcode = req.body.postcode;
215     let address = StringUtility.capitalizeString(req.body.address);
216     let status = req.body.status;
217
218     // do validation here
219
220     let Validation = new ContactsValidation(businessName,firstName, lastName, email,
221         telephone, city, postcode, address, status);
222     let validationResult = Validation.checkFieldsAreValid(); // this will either
223     store the boolean value true or a message of what needs to be fixed
224
225     if (validationResult === true) {
226         // if it passes validation, add record and redirect to contacts list page
227         contactsFile.addRecord([businessName, firstName, lastName, email, telephone,
228             city, postcode, address, status]);
229         req.session.message = "Successfully Created A Contact";
230         res.redirect("/contacts/");
231     }
232
233     else {
234         // if something does not pass the validation then the user will get
235         // redirected to the
236         // create contacts page and a req.session.message will get set to the
237         // validationResult output
238         // this can then be accessed at the /contacts/new route to notify the user
239         // where they went wrong
240
241         req.session.message = validationResult; // validation message
242
243         // there will also be other properties attached to the session memory
244         // property containing the entered form data
245         // so that the fields don't go blank if any of the user's inputs did not
246         // pass the validation
247         req.session.businessName = businessName;
248         req.session.firstName = firstName;
249         req.session.lastName = lastName;
250         req.session.email = email;
251         req.session.telephone = telephone;
252         req.session.city = city;
253         req.session.postcode = postcode;
254         req.session.address = address;
255
256         res.redirect("/contacts/new")
257     }
258 }
259 });
260
261 //renders the contacts list page that outputs any contacts that match the search value
262 router.post("/search", (req, res) => {
263     let contactsFile = new CSVHandler(__dirname + '/dataStoreFiles/contacts.csv');
264
265     let searchBarValue = req.body.searchBar; // gets the search value from the form

```

```

265 // finds all matches of the search value by looking in every column for one
266 let matchingRecords =
267 contactsFile.getRecordsOnlyByField([0,1,2,3,4,5,6,7,8,9,10], searchValue);
268
269 // if there is no match then do the search again but with the search value
270 // capitalized
271 if (matchingRecords.length === 0) {
272     searchValue = StringUtility.capitalizeString(searchValue);
273     matchingRecords =
274     contactsFile.getRecordsOnlyByField([0,1,2,3,4,5,6,7,8,9,10], searchValue);
275
276     // if there is still no match, the user gets redirected to the contacts list
277     // page with all the contacts
278     if (matchingRecords.length === 0) {
279         req.session.errorMessage = "No Matches found";
280         res.redirect("/contacts/");
281     } else {
282
283         res.render("ContactsList", {
284             contactRecords: matchingRecords,
285             titleAddition: "Searched for: " + searchValue,
286             flashMessage : matchingRecords.length.toString() + " Matches Found",
287             errorMessage: undefined
288         });
289     }
290 } else {
291     res.render("ContactsList", {
292         contactRecords: matchingRecords,
293         titleAddition: "Searched for: " + searchValue,
294         flashMessage : matchingRecords.length.toString() + " Matches Found",
295         errorMessage: undefined
296     });
297
298
299
300
301
302 router.post("/sortBy", (req, res) => {
303     let contactsFile = new CSVHandler(__dirname + '/dataStoreFiles/contacts.csv');
304     let sortByValue = req.body.sortBy; // will return undefined if no value is
305     selected
306     let ascOrDec = req.body.order; // will be either ascending or descending
307     let sortedArray = [];
308
309     // turns the names gotten back from the sort form into the index that the
310     // records will need to be sorted by
311     switch (sortByValue) {
312         case "date":
313             sortByValue = {index: 1, label: "Date"}; // date is stored at index 1 in
314             the contacts csv file
315             break;
316         case "city":
317             sortByValue = {index: 7, label: "City"}; // city is stored at index 7 in
318             the contacts csv file
319             break;
320         case "businessName":
321             sortByValue = {index: 2, label: "Business Name"}; // business name is at
322             index 2 in the contacts csv file
323             break;
324         default:
325             // if undefined is returned because there is no value selected in the form
326             // then the sortByValue will be set to the value 3 which is the index of
327             // the first name
328             sortByValue = {index: 3, label: "First Name"};
329     }
330
331     // turns the ascending/ descending option into a boolean value which can be used
332     // by the sortBy method
333     switch (ascOrDec) {

```

```

327     case "ascending":
328         ascOrDec = true;
329         break;
330     case "descending":
331         ascOrDec = false;
332         break;
333     default:
334         ascOrDec = false;
335     }
336
337     // sort the array by the sortByValue in ascending or descending order dependent
338     // on the ascOrDec boolean argument that
339     // gets passed in. the sorted array gets stored into the sortedArray variable
340     sortedArray = contactsFile.sortBy(sortByValue.index, ascOrDec);
341
342     res.render("ContactsList",
343     {
344         contactRecords: sortedArray,
345         titleAddition: "Sorted By: " + sortByValue.label,
346         flashMessage: undefined,
347         errorMessage: undefined
348     });
349 }
350
351
352
353
354
355
356
357 // this route is used to add a note to the contactNotes file
358 router.post("/addnote:id", (req, res) => {
359     let contactsNotes = new CSVHandler(__dirname +
360         '/dataStoreFiles/contactsNotes.csv');
361
362     let contactId = req.params.id; // gets the contacts id from the url
363     let note = req.body.addNote; // gets the data in the text area and stores it
364     // gets the current date and then converts it into dd/mm/yyyy format
365     let date =
366         DateHandler.standardDateTo_Date_DDMMYYYY(DateHandler.getCurrentDateTime());
367
368     // do validation on the note
369     let validation = new AddNoteValidation(note);
370     let validationValue = validation.checkFieldsAreValid();
371
372     if (validationValue === true) { // if it passes the validation then add the record
373         // add note to the file as: noteId, dateTimeStamp, contactId, note
374         contactsNotes.addRecord([date, contactId, note]);
375         req.session.message = "Successfully Added A Note";
376         res.redirect("/contacts/viewContact" + contactId);
377     }
378
379     else {
380         // if the data does not pass the validation the user will get sent back to
381         // the same page with a
382         // message telling where they went wrong
383         req.session.errorMessage = validationValue;
384         res.redirect("/contacts/viewContact" + contactId);
385     }
386
387 }
388
389
390
391
392
393 // will take the user to the password page so that the user can enter a password to
394 // encrypt or decrypt the
395 // Extra Notes text area data

```

```

395
396 router.post("/viewContact:id/encryptNotes_:encOrDec", (req, res) => {
397   let id = req.params.id; // get id from the url
398   let encryptOrDecrypt = req.params.encOrDec; // stores if the user wants to
399   encrypt or decrypt the data
400   let message = req.session.message;
401
402   if (req.body.extraNotes !== undefined) {
403     req.session.extraNotesData = req.body.extraNotes; // get the extra notes
404     data from the text area and store in the session memory
405   }
406
407
408
409
410 // preform encryption on the data and append a new record to the database if there
411 // is no record containing the contact's id
412 // if a record is found with the users id then the record will be edited to contain
413 // the new encrypted data
414 router.post("/viewContact:id/encrypt", (req, res) => {
415
416   let id = req.params.id;
417   let extraNotesFile = new CSVHandler(__dirname +
418     '/dataStoreFiles/contactsExtraNotes.csv');
419
420
421   // does password validation and gets either true or a string value returned
422   let passwordValidation = new PasswordValidation(req.body.password,
423     req.body.confirmPassword);
424   let validationResult = passwordValidation.checkFieldsAreValid();
425   let encryptedData;
426   let extraNotesRecord;
427
428   // do password validation, if it passes validation then continue
429   if (validationResult === true) {
430
431     let extraNotesData = req.session.extraNotesData; // stores the
432     extraNotesData from the session memory
433     req.session.extraNotesData = undefined; // removes the property from the
434     session memory
435
436     // encrypt data with the password
437
438     encryptedData = Encryption.xorEncrypt(extraNotesData, req.body.password);
439
440     // look for a record with an id of the contacts
441     // id found ==> edit the data to store the encrypted data
442     // id not found ==> add record : [id, timestamp, contact id, data]
443
444     // search for a match for the contacts id in the contactsExtraNotes file
445
446     extraNotesRecord = extraNotesFile.getRecordsOnlyByField([2], id);
447     if(extraNotesRecord.length === 0){ // if no match is found
448       extraNotesFile.addRecord([id, encryptedData]); // add record
449       req.session.message = "Successfully Encrypted Data";
450       res.redirect("/contacts/viewContact" + id);
451     }
452
453     // match is found
454     else {
455       // edit the data in that record
456       extraNotesFile.editRecord(extraNotesRecord[0][0], [[3, encryptedData]]);
457       req.session.message = "Successfully Changed The Encrypted Data";
458       res.redirect("/contacts/viewContact" + id);
459     }
460
461   }
462
463   else {

```

```

459     // if not, redirect to the /contacts/viewContact:id/encryptNotes_encrypt
460     // with a message of what went wrong
461     req.session.message = validationResult;
462     // redirects to the post route /viewContact:id/encrypt
463     res.redirect(307, "/contacts/viewContact" + id + "/encryptNotes_encrypt");
464   }
465
466
467
468
469
470
471
472 });
473
474
475
476
477 // preform decryption on the data stored in req.session.extraNotesData
478 // if req.session.extraNotesData === undefined then redirect to the individual
479 contacts page
480 router.post("/viewContact:id/decrypt", (req, res) => {
481   // do password validation, if it passes validation then continue
482   // if not, redirect to the /contacts/viewContact:id/encryptNotes_decrypt
483   // with a message of what went wrong
484
485   let id = req.params.id;
486   let extraNotesFile = new CSVHandler(__dirname +
487     '/dataStoreFiles/contactsExtraNotes.csv');
488
489   // does password validation and gets either true or a string value returned
490   let passwordValidation = new PasswordValidation(req.body.password,
491     req.body.confirmPassword);
492   let validationResult = passwordValidation.checkFieldsAreValid();
493   let extraNotesRecord;
494   let encryptedData;
495   let decryptedData;
496
497   // do password validation, if it passes validation then continue
498   if (validationResult === true) {
499
500     // decrypt data with the password
501
502     // look for a record with an id of the contacts
503     // id found ==> edit the data to store the encrypted data
504     // id not found ==> add record : [id, timestamp, contact id, data]
505
506     // search for a match for the contacts id in the contactsExtraNotes file
507     extraNotesRecord = extraNotesFile.getRecordsOnlyByField([2], id)[0];
508
509     if(extraNotesRecord.length === 0){ // if no match is found
510       req.session.errorMessage = "No Data To Decrypt";
511       res.redirect("/contacts/viewContact" + id);
512     }
513
514     // match is found
515     else {
516       // decrypt the encrypted data and put that into session memory
517       encryptedData = extraNotesRecord[3];
518       decryptedData = Encryption.xorDecrypt(encryptedData, req.body.password);
519       req.session.message = "Successfully Decrypted The Data";
520       req.session.decryptedData = decryptedData;
521       res.redirect("/contacts/viewContact" + id);
522     }
523
524   }
525
526   else {
527     // if not, redirect to the /contacts/viewContact:id/encryptNotes_encrypt
528     // with a message of what went wrong

```

```

529     req.session.message = validationResult;
530     // redirects to the post route /viewContact:id/encrypt
531     res.redirect(307, "/contacts/viewContact" + id + "/encryptNotes_decrypt");
532   }
533
534
535
536 });
537
538
539
540
541
542
543
544
545
546 //////////////////////////////////////////////////////////////////
547 //////////////////////////////////////////////////////////////////
548 //                                     PUT ROUTES
549
550 //////////////////////////////////////////////////////////////////
551
552
553
554
555 // route for updating files containing data
556 router.put("/:id", (req, res) => {
557   let contactsFile = new CSVHandler(__dirname + '/dataStoreFiles/contacts.csv');
558   let record = req.body; // stores the data from the edit contact form in an object
559   let recordArray = Object.values(record); // turns the record object into an array
560
561   let arrayOfEdits = [];
562
563   // populate an array of changes to be made
564   for (let i = 0; i < recordArray.length; i++) {
565     // capitalizes the values stored in the array at the bellow index's
566     if (i === 0 || i === 1 || i === 2 || i === 5 || i === 7) {
567       recordArray[i] = StringUtil.capitalizeString(recordArray[i]);
568     }
569     // puts the data into the format needed for the edit record method to work
570     let recordChange = [(i+2).toString(), recordArray[i]];
571     arrayOfEdits.push(recordChange);
572   }
573
574   // do validation here
575
576   let Validation = new ContactsValidation(record.businessName, record.firstName,
577   record.lastName,
578   record.email, record.telephone, record.city, record.postcode,
579   record.address, record.status);
580
581   let validationResult = Validation.checkFieldsAreValid(); // this will either
582   // store the boolean value true or a message of what needs to be fixed
583
584   if (validationResult === true) {
585     // update all fields in the record that has been updated
586     contactsFile.editRecord(req.params.id, arrayOfEdits);
587     req.session.message = "Contact Successfully Edited";
588     res.redirect("/contacts/"); // takes user back to the contact list page
589   }
590
591   else {
592     // if something does not pass the validation then the user will get
593     // redirected to the
594     // create contacts page and a req.session.message will get set to the
595     // validationResult output
596     // this can then be accessed at the /contacts/new route to notify the user
597     // where they went wrong
598
599     req.session.message = validationResult; // validation message

```

```

594
595
596     // if validation not met then a message gets generated and the user gets
597     // sent to the edit contact page
598     // with that message stored in the session memory object
599
600     res.redirect("/contacts/edit" + req.params.id); // takes user back to the
601     // contacts list page
602 }
603
604
605
606 ///////////////////////////////////////////////////////////////////
607 ///////////////////////////////////////////////////////////////////
608 //          DELETE ROUTES
609 ///////////////////////////////////////////////////////////////////
610 ///////////////////////////////////////////////////////////////////
611
612
613 // delete route to remove a record
614 // when a contact is deleted, all of its notes and its extra notes records should
615 // also get deleted
616 router.delete("/delete:id", (req, res) => {
617     let contactsFile = new CSVHandler(__dirname + '/dataStoreFiles/contacts.csv');
618     let contactsExtraNotesFile = new CSVHandler(__dirname +
619         '/dataStoreFiles/contactsExtraNotes.csv');
620     let id = req.params.id; // gets the contacts id from the url
621     let contactsExtraNoteRecord;
622
623     let contactsNotesTempFileForDelete;
624
625     contactsFile.deleteRecord(id); // deletes the contact record from the
626     // contacts.csv file
627
628     // gets the record with the contacts id at index 2 Then returns a 2D array with
629     // the record in the only array
630     // within the 2D array. record accessed with the [0] at the end
631     contactsExtraNoteRecord = contactsExtraNotesFile.getRecordsOnlyByField([2],
632     id)[0];
633     contactsExtraNotesFile.deleteRecord(contactsExtraNoteRecord[0]); // the [0] gets
634     // the data at the 0 index which is the id
635
636     res.redirect("/contacts/");
637 }
638
639
640
641 module.exports = router;

```

# index.js

```
1 const express = require("express");
2 const router = express.Router();
3
4 router.get("/", (req, res) => {
5   res.render("Home");
6 });
7
8
9 router.get("/settings", (req, res) => {
10   res.render("Settings");
11 });
12
13 router.get("/termsAndConditions", (req, res) => {
14   res.render("TermsAndConditions");
15 });
16
17
18
19 module.exports = router;
```

# Views Directory Contents

# AnalyticsView.ejs

Kieran Marcus Williams

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Analytics</title>
6
7      <link rel="stylesheet"
8          href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
9          integrity="sha384-Gn5384xqQlaoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm
10         " crossorigin="anonymous">
11
12      <link rel="stylesheet"
13          href="https://use.fontawesome.com/releases/v5.6.3/css/all.css"
14          integrity="sha384-UHRTzLI+pbxtHCWp1t77Bi1L4ZtiqrqD80Kn4Z8NTSRyMA2Fd33n5dQ81WUE00s/
15         " crossorigin="anonymous">
16
17      <script
18          src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.7.3/Chart.bundle.min.js"></
19      script>
20
21      <link rel="stylesheet" type="text/css" href="/stylesheets/AnalyticsStyles.css">
22      <link rel="stylesheet" type="text/css"
23          href="/stylesheets/NavBarLoggedInStyles.css">
24      <link rel="stylesheet" type="text/css"
25          href="/stylesheets/NavBarNotLoggedInStyles.css">
26
27  </head>
28  <body>
29
30
31  <% include partials/NavBarLoggedIn.ejs %>
32  <br><br>
33
34  <h1>Analytics</h1>
35  <br><br><br>
36
37  <!--top section with the graphs-->
38
39  <div class="split-screen-horizontal">
40
41      <div class="top-half-of-analytics">
42          <div class="first-graph">
43              <canvas id="clients-gained"></canvas>
44          </div>
45          <div class="second-graph">
46              <canvas id="clients-lost"></canvas>
47          </div>
48          <!--<div class="third-graph">-->
49              <!--<canvas id="number-of-clients"></canvas>-->
50          <!--</div>-->
51      </div>
52  <br><br><br>
53
54
55  <!--bottom section for conversion rate and the input analytical data buttons-->
56
57  <div class="bottom-half-of-analytics">
58
59      <div class="left-side-bottom-analytics">
60
61          <div class="conversion-rate-holder">
62              <h3>Conversion Rate This Month</h3>
63              <table class="conversion-rate-table">
64                  <tr>
65                      <td><h4
66                          class="remove-margin"><%=conversionRate.ratio%></h4></td>
67                      <td><h4
68                          class="remove-margin"><%=conversionRate.percentage%></h4></td>
69                  </tr>
70              </table>
71          </div>
72
73      </div>
74
75

```

```
61
62 <div class="middle-bottom-analytics">
63   <label for="time-scale">Time Scale</label>
64   <form id="time-scale-form" action="/analytics/" method="get">
65     <div class="time-scale-separator">
66       <select onchange="changeAction()" class="custom-select"
67         name="time-scale" id="time-scale">
68
69           <!--ejs tags help determine which of the options should be
70             made selected -->
71
72           <% if (timeScale === "daily") { %>
73             <option selected class="time-scale-option"
74               value="daily">Daily</option>
75           <% } else { %>
76             <option selected class="time-scale-option"
77               value="daily">Daily</option>
78           <% } %>
79
80           <% if (timeScale === "weekly") { %>
81             <option selected class="time-scale-option"
82               value="weekly">Weekly</option>
83           <% } else { %>
84             <option class="time-scale-option"
85               value="weekly">Weekly</option>
86           <% } %>
87
88           <% if (timeScale === "monthly") { %>
89             <option selected class="time-scale-option"
90               value="monthly">Monthly</option>
91           <% } else { %>
92             <option class="time-scale-option"
93               value="monthly">Monthly</option>
94           <% } %>
95
96         </select>
97         <button id="time-scale-btn" class="btn
98           btn-success">Submit</button>
99       </div>
100    </form>
101  </div>
102
103 <div class="right-side-bottom-analytics">
104
105   <div class="button-holder-analytics">
106
107     <div class="row-of-buttons">
108
109       <a class="btn btn-primary" id="client-lost-btn"
110         href="/analytics/clientLost">Client Lost</a>
111
112       <a class="btn btn-primary" id="client-gained-btn"
113         href="/analytics/clientGained">Client Gained</a>
114
115     </div>
116
117     <div class="row-of-buttons">
118
119       <a class="btn btn-primary" id="lead-lost-btn"
120         href="/analytics/leadLost">Lead Lost</a>
121
122       <a class="btn btn-danger" onclick="return confirm('Are You Sure
123         You Want To Delete ALL Analytics Data?')"
124         id="remove-all-data-btn"
125         href="/analytics/removeAllData">Remove All Data</a>
126
127     </div>
128   </div>
129 </div>
```

```
119     </div>
120 
121 </div>
122 
123 
124 
125 
126 
127 
128 <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-KJ3o2DKtIkYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>
129 <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>
130 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-JZR6Spejh4U02d8j0t6vLEHfe/JQGiRRSQQxSffFWpilMquVdAyjUar5+76PVCmYl"
crossorigin="anonymous"></script>
131 
132 
133 
134 
135 <!-- sets the variables for the graphs to use - data points and labels-->
136 <script>
137     let clientsGainedLabels = "<%=graphData.clientsGained.labels%>".split(", ");
138     let clientsGainedDataPoints =
139         "<%=graphData.clientsGained.dataPoints%>".split(", ");
140 
141     let clientsLostLabels = "<%=graphData.clientsLost.labels%>".split(", ");
142     let clientsLostDataPoints = "<%=graphData.clientsLost.dataPoints%>".split(", ");
143 
144 </script>
145 <script src="/scripts/AnalyticsScripts.js"></script>
146 
147 </body>
148 </html>
149 
```

# ContactsListView.ejs

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Contacts</title>
6      <link rel="stylesheet"
7          href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
8          integrity="sha384-Gn5384xqQlaoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm
9          " crossorigin="anonymous">
10     <link rel="stylesheet"
11         href="https://use.fontawesome.com/releases/v5.6.3/css/all.css"
12         integrity="sha384-UHRTzLI+pbxtHCWplt77Bi1L4ZtiqrqD80Kn4Z8NTSRyMA2Fd33n5dQ81WUE00s/
13         " crossorigin="anonymous">
14     <link rel="stylesheet" href="/stylesheets/ContactsListStyles.css">
15     <link rel="stylesheet" href="/stylesheets/NavBarLoggedInStyles.css">
16     <link rel="stylesheet" href="/stylesheets/NavBarNotLoggedInStyles.css">
17 </head>
18 <body>
19
20
21
22
23
24
25
26
27
28
29
30
31    <!--page for displaying the users contacts in a table-->
32
33 <h1>Contacts List
34     <% if(titleAddition !== undefined) { %>
35         -   <em style="margin-left: 15px;"><%= titleAddition %></em>
36     <% } %>
37 </h1>
38
39    <!--page is split up vertically-->
40
41 <div class="holder">
42
43     <!--left side will contain the searching and sorting options along with options
44         to add a contact, import contacts with a csv file
45         and export contacts as a csv file-->
46
47     <div id="left-side-container">
48         <div class="jumbotron">
49             <div id="modify-list-results-form">
50
51                 <form action="/contacts/search" method="post">
52                     <label for="searchBar">Search</label><br>
53                     <input name="searchBar" id="searchBar" type="text">
54                     <button class="btn btn-sm btn-outline-dark"><i class="fas fa-search"></i></button>
55
56                 </form>
57                 <br>
58
59                 <form action="/contacts/sortBy" method="post">
60
61                     <h5>Sort By</h5>
62
63                     <input type="radio" name="sortBy" id="date" value="date"> <label
64                         for="date">Date</label> <br>
65
66                     <input type="radio" name="sortBy" id="city" value="city"> <label

```

```

64         for="city">City</label> <br>
65
66         <input type="radio" name="sortBy" id="business-name"
67           value="businessName"> <label for="business-name">Business
68           Name</label> <br><br>
69
70           <label for="order">Order</label>
71           <select class="custom-select" name="order" id="order">
72             <option value="ascending">ascending</option>
73             <option value="descending">descending</option>
74           </select> <br><br>
75
76           <input class="btn btn-success" type="submit">
77         </form>
78       </div>
79
80       <hr>
81
82       <div id="contact-management">
83
84         <a class="btn btn-primary" id="add-contact" href="/contacts/new">Add
85           Contact</a>
86
87         <hr>
88
89         <!-- UNCOMMENT TO BE ABLE IMPORT CONTACTS HTML WHEN IT IS TO GET
90           IMPLEMENTED -->
91
92         <!-->
93         <!--<form>-->
94           <!--<div class="form-group">-->
95             <!--<label for="import-contacts">Import Contacts (.CSV
96               file)</label>-->
97             <!--<input type="file" class="form-control-file"
98               id="import-contacts">-->
99             <!--</div>-->
100            <!--</form>-->
101            <!--<br>-->
102
103
104           <!--<form action="#">-->
105             <!--<button class="btn btn-secondary"
106               id="import-contact-btn">import Contact</button>-->
107           <!--</form>-->
108
109           <br>
110
111           <form action="/contacts/download" method="get">
112             <button class="btn btn-secondary" id="export-contact-btn">Export
113               Contacts</button>
114           </form>
115
116           </div>
117
118           </div>
119
120           <!--on the right side there is a table containing all the contacts with a link
121             to the contact's individual page in the name field-->
122           <div id="right-side-container">
123             <table class="table">
124               <thead class="thead-dark">
125                 <tr>
126                   <th>Name</th>
127                   <th>Business</th>
128                   <th>City</th>
129                   <th>Email</th>
130                   <th>Telephone</th>
131                 </tr>

```

```
126      </thead>
127
128      <tbody>
129
130          <!--gets the contacts details and puts it into the table row by row-->
131          <% contactRecords.forEach((record) => { %>
132              <tr>
133                  <td><a
134                      href="/contacts/viewContact<%=record[0]%>"><%=record[3] + %
135                          " + record[4]></a></td>
136                  <td><%=record[2]> </td>
137                  <td><%=record[7]></td>
138                  <td><%=record[5]></td>
139                  <td><%=record[6]></td>
140              </tr>
141          <%});%>
142
143      </tbody>
144  </table>
145  </div>
146</div>
147
148 <script src="/scripts/ContactListScripts.js"></script>
149 <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-KJ3o2DKtIkYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>
150 <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>
151 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-JZR6Spejh4U02d8j0t6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
crossorigin="anonymous"></script>
152 </body>
153 </html>
```

# CreateContactView.ejs

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Create Contact</title>
6      <link rel="stylesheet"
7          href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
8          integrity="sha384-Gn5384xqQlaoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm
9          crossorigin="anonymous">
10     <link rel="stylesheet" href="/stylesheets/CreateContactStyles.css">
11     <link rel="stylesheet" href="/stylesheets/NavbarLoggedInStyles.css">
12     <link rel="stylesheet" href="/stylesheets/NavbarNotLoggedInStyles.css">
13     <script src="/scripts/CreateContactScripts.js"></script>
14 </head>
15 <body>
16
17
18 <%if (flashMessage !== undefined){%>
19 <div class="alert alert-danger">
20     <%= flashMessage %>
21 </div>
22 <%}%>
23
24 <!--basic form using bootstrap that will be used as a data capture to make create a
contact-->
25
26 <h1>Create Contact</h1>
27 <div class="container">
28     <div class="jumbotron">
29         <form class="form" action="/contacts" method="post">
30             <h3>Contact Details</h3><br>
31             <div class="row">
32                 <div class="col">
33                     <label for="business">Business</label>
34                     <input class="form-control" id="business" name="businessName"
35                         value="<%=savedBodyValue[0]%>" type="text"><br>
36                 </div>
37                 <div class="col">
38                     <label for="first-name">First Name</label>
39                     <input class="form-control" id="first-name" name="firstName"
40                         value="<%=savedBodyValue[1]%>" type="text"><br>
41                 </div>
42             <div class="row">
43                 <div class="col">
44                     <label for="last-name">Last Name</label>
45                     <input class="form-control" id="last-name" name="lastName"
46                         value="<%=savedBodyValue[2]%>" type="text"><br>
47                 </div>
48                 <div class="col">
49                     <label for="email">Email</label>
50                     <input class="form-control" id="email" name="email"
51                         value="<%=savedBodyValue[3]%>" type="text"><br>
52                 </div>
53             <div class="row">
54                 <div class="col">
55                     <label for="telephone">Telephone</label>
56                     <input class="form-control" id="telephone" name="telephone"
57                         value="<%=savedBodyValue[4]%>" type="text"><br>
58                 </div>
59                 <div class="col">
60                     <label for="city">City</label>
61                     <input class="form-control" id="city" name="city"
62                         value="<%=savedBodyValue[5]%>" type="text"><br>
63                 </div>
64             </div>
65         <div class="row">
66             <div class="col">

```

```

63         <label for="postcode">Postcode</label>
64         <input class="form-control" id="postcode" name="postcode"
65             value="<%>=savedBodyValue[6]<%>" type="text"><br>
66     </div>
67     <div class="col">
68         <label for="address">Address</label>
69         <input class="form-control" id="address" name="address"
70             value="<%>=savedBodyValue[7]<%>" type="text"><br>
71     </div>
72 </div>
73 <div class="row">
74     <div class="col">
75         <label for="status">Contact Status</label>
76         <select class="custom-select" name="status" id="status">
77             <option value="lead">Lead</option>
78             <option value="recontact-lead">Recontact Lead</option>
79             <option value="dead-lead">Dead Lead</option>
80             <option value="current-client">Current Client</option>
81             <option value="past-client">Past Client</option>
82         </select>
83     </div>
84     <input class="btn btn-primary btn-lg btn-submit" type="submit">
85 </form>
86 </div>
87
88
89
90
91
92 <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
93     integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
94     crossorigin="anonymous"></script>
95 <script
96     src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
97     integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
98     crossorigin="anonymous"></script>
99 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
100    integrity="sha384-JZR6Spejh4U02d8j0t6vLEHfe/JQGiRRSQxSffFWpi1MquVdAyjUar5+76PVCmYl"
101    crossorigin="anonymous"></script>
</body>
</html>

```

# DashboardView.ejs

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Dashboard</title>
6
7      <link rel="stylesheet"
8          href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
9          integrity="sha384-Gn5384xqQlaoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
10         crossorigin="anonymous">
11     <link rel="stylesheet"
12         href="https://use.fontawesome.com/releases/v5.6.3/css/all.css"
13         integrity="sha384-UHRTZLI+pbxtHCWp1t77Bi1L4ZtiqrqD80Kn4Z8NTSRyMA2Fd33n5dQ81WUE00s/"
14         crossorigin="anonymous">
15     <link rel="stylesheet" href="/stylesheets/NavbarLoggedInStyles.css">
16     <link rel="stylesheet" href="/stylesheets/NavbarNotLoggedInStyles.css">
17     <link rel="stylesheet" href="/stylesheets/DashboardStyles.css">
18
19
20     <% include partials/NavbarLoggedIn.ejs %>
21
22     <%if (flashMessage !== undefined){%>
23         <div class="alert alert-success">
24             <%= flashMessage %>
25         </div>
26     <%}%>
27
28     <%if (errorMessage !== undefined){%>
29         <div class="alert alert-danger">
30             <%= errorMessage %>
31         </div>
32     <%}%>
33
34     <h1>Dashboard</h1>
35
36     <!-- The Dashboard is split into 3 sections -->
37     <div id="dashboard-separator">
38
39         <!--on the left there is the form to create a scheduling -->
40
41         <div id="schedule-task-form">
42
43             <h3>Schedule Task</h3>
44
45
46             <form class="dashboard-slots" title="Schedule Task" action="/dashboard/"
47                 method="post" id="Schedule Task">
48                 <label for="date">date:</label> <br>
49                 <input id="date" name="date" title="date" type="text"
50                     placeholder="dd/mm/yyyy"> <br><br>
51
52                 <label for="time">time:</label><br>
53                 <input id="time" name="time" title="time" type="text" placeholder="24
54                     hour time: hh:mm"> <br><br>
55
56                 <label for="subject">subject:</label><br>
57                 <input id="subject" name="subject" type="text"> <br><br>
58
59                 <label for="message">message:</label> <br>
60                 <textarea onkeypress="return returnKeyBlocker(event)" id="message"
61                     title="message" name="message" cols="30" rows="10"></textarea> <br><br>
62
63                 <input class="btn btn-success" type="submit">
64             </form>
65         </div>

```

```

63
64    <!-- in the middle there will be an array that shows anything that is
65    scheduled.csv for the current date -->
66    <div id="today">
67        <h3>Today</h3>
68        <div class="dashboard-slots">
69
70
71
72
73        <%scheduledToday.forEach((schedule) => { %>
74
75            <tr>
76                <td>
77
78                    <div class="schedule-info">
79                        <div class="time-display-today"><%=schedule[3]%></div>
80                        <div class="date-display-today"><%=schedule[4]%></div>
81                        <span style="font-weight: bold"><div
82                            class="subject-display-today"><%=schedule[5]%></div></span
83                            >
84                            <div class="message-display-today"><%=schedule[6]%></div>
85                        </div>
86
87                </td>
88
89                <td>
90                    <form
91                        action="/dashboard/delete<%=schedule[0]%>?_method=DELETE"
92                        method="post">
93                        <button style="float: right; margin-bottom: 10px;
94                            font-size: x-large;" class="btn"><i class="trash-button
95                                fas fa-trash-alt"></i></button>
96                    </form>
97
98                </td>
99
100            </tr>
101            <br><br><hr>
102
103        <%}); %>
104
105        </div>
106    </div>
107
108
109
110
111
112
113    <!--on the right all the scheduled.csv tasks further on into the future from
114    just the the current day will be put into this list -->
115
116    <div id="future">
117        <h3>Future</h3>
118        <div class="dashboard-slots">
119
120            <%scheduledFuture.forEach((schedule) => { %>
121
122                <tr>
123                    <td>
124
125                        <div class="schedule-info">
126                            <div class="time-display-today"><%=schedule[3]%></div>
127                            <div class="date-display-today"><%=schedule[4]%></div>
128                            <span style="font-weight: bold"><div
129                                class="subject-display-today"><%=schedule[5]%></div></span>
130                            <div class="message-display-today"><%=schedule[6]%></div>
131                        </div>
132
133                    </td>
134
135                    <td>
136                        <form action="/dashboard/delete<%=schedule[0]%>?_method=DELETE"
137                        method="post">
138                            <button style="float: right; margin-bottom: 10px; font-size:

```

```
125         <x-large;" class="btn"><i class="trash-button fas fa-trash-alt"></i></button>
126     </form>
127   </td>
128 </tr>
129 <br><br><hr>
130
131
132
133   <%});%>
134   </div>
135 </div>
136 </div>
137
138
139
140 <script src="/scripts/DashboardScripts.js"></script>
141 <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-KJ3o2DKtIkYIJK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>
142 <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPsvkvXusvfa0b4Q"
crossorigin="anonymous"></script>
143 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-JZR6Spejh4U02d8j0t6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
crossorigin="anonymous"></script>
144 </body>
145 </html>
```

# EditContactView.ejs

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Edit Contact</title>
6      <link rel="stylesheet"
7          href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
8          integrity="sha384-Gn5384xqQlaoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm
9          crossorigin="anonymous">
10     <link rel="stylesheet" href="/stylesheets/EditContactStyles.css">
11     <link rel="stylesheet" href="/stylesheets/NavbarLoggedInStyles.css">
12     <link rel="stylesheet" href="/stylesheets/NavbarNotLoggedInStyles.css">
13     <script src="/scripts/EditContactScripts.js"></script>
14 </head>
15 <body>
16
17 <% include partials/NavbarLoggedIn.ejs %>
18
19 <%if (flashMessage !== undefined){%>
20     <div class="alert alert-danger">
21         <%= flashMessage %>
22     </div>
23 <%}%>
24
25 <!--form containing a contacts data in the text fields-->
26 <h1>Edit Contact</h1>
27 <div class="container">
28
29     <div class="jumbotron">
30         <!--action url is set to put to have the correct http verb for updating db-->
31         <form class="form" action="/contacts/<%= contact.record[0] %>?_method=PUT"
32             method="post">
33             <h3>Contact Details</h3><br>
34             <div class="row">
35                 <div class="col">
36                     <label for="business">Business</label>
37                     <input class="form-control" id="business" type="text"
38                         name="businessName" value="<%= contact.record[2] %>"><br>
39                 </div>
40                 <div class="col">
41                     <label for="first-name">First Name</label>
42                     <input class="form-control" id="first-name" type="text"
43                         name="firstName" value="<%= contact.record[3] %>"><br>
44                 </div>
45             </div>
46             <div class="row">
47                 <div class="col">
48                     <label for="last-name">Last Name</label>
49                     <input class="form-control" id="last-name" type="text"
50                         name="lastName" value="<%= contact.record[4] %>"><br>
51                 </div>
52                 <div class="col">
53                     <label for="email">Email</label>
54                     <input class="form-control" id="email" type="text" name="email"
55                         value="<%= contact.record[5] %>"><br>
56                 </div>
57             </div>
58             <div class="row">
59                 <div class="col">
60                     <label for="telephone">Telephone</label>
61                     <input class="form-control" id="telephone" type="text"
62                         name="telephone" value="<%= contact.record[6] %>"><br>
63                 </div>
64                 <div class="col">
65                     <label for="city">City</label>
66                     <input class="form-control" id="city" type="text" name="city"
67                         value="<%= contact.record[7] %>"><br>
68                 </div>
69             </div>
70             <div class="row">
71                 <div class="col">

```

```

63         <label for="postcode">Postcode</label>
64         <input class="form-control" id="postcode" type="text"
65           name="postcode" value="<% contact.record[8] %>"><br>
66     </div>
67     <div class="col">
68       <label for="address">Address</label>
69       <input class="form-control" id="address" type="text"
70         name="address" value="<% contact.record[9] %>"><br>
71     </div>
72   </div>
73   <div class="row">
74     <div class="col">
75       <label for="status">Contact Status</label>
76       <select class="custom-select" name="status" id="status">
77
78         <!--the bellow conditionals check to see the value of the
79         clients current status and makes the
80         the right one the selected value out of the drop down
81         menu-->
82
83         <%if(contact.record[10] === "lead") { %>
84           <option selected value="lead">Lead</option>
85         <%} else { %>
86           <option value="lead">Lead</option>
87         <%}%>
88
89         <%if(contact.record[10] === "recontact-lead") { %>
90           <option selected value="recontact-lead">Recontact
91             Lead</option>
92         <%} else { %>
93           <option value="recontact-lead">Recontact Lead</option>
94         <%}%>
95
96         <%if(contact.record[10] === "dead-lead") { %>
97           <option selected value="dead-lead">Dead Lead</option>
98         <%} else { %>
99           <option value="dead-lead">Dead Lead</option>
100        <%}%>
101
102
103         <%if(contact.record[10] === "current-client") { %>
104           <option selected value="current-client">Current
105             Client</option>
106         <%} else { %>
107           <option value="current-client">Current Client</option>
108         <%}%>
109
110         </select>
111     </div>
112   </div>
113   <input class="btn btn-primary btn-lg btn-submit" type="submit">
114 </form>
115 </div>
116 </div>
117
118
119
120
121
122 <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
123   integrity="sha384-KJ3o2DKtIkVYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
124   crossorigin="anonymous"></script>
125 <script
126   src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
127   integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
128   crossorigin="anonymous"></script>

```

```
124 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
125   integrity="sha384-JZR6Spejh4U02d8j0t6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
126   crossorigin="anonymous"></script>
</body>
</html>
```

# EncryptNotesView.ejs

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Encrypt Extra Notes</title>
6      <link rel="stylesheet"
7          href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
8          integrity="sha384-Gn5384xqQlaoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm
9          crossorigin="anonymous">
10     <link rel="stylesheet" href="/stylesheets/ChangePasswordStyles.css">
11     <link rel="stylesheet" href="/stylesheets/NavBarLoggedInStyles.css">
12     <link rel="stylesheet" href="/stylesheets/NavBarNotLoggedInStyles.css">
13     <script src="/scripts/EncryptionNotesPasswordScripts.js"></script>
14 </head>
15 <body>
16
17
18 <%if (flashMessage !== undefined){%>
19 <div class="alert alert-danger">
20     <%= flashMessage %>
21 </div>
22 <%}%>
23
24
25 <h1>Encryption Password</h1>
26 <div class="container">
27
28
29
30     <!--basic form to change the password-->
31     <div class="jumbotron">
32         <form class="form"
33             action="/contacts/viewContact<%=contactId%>/<%=encryptOrDecrypt%>"
34             method="post">
35             <div class="row">
36
37                 <div class="col">
38                     <label for="password">Password</label>
39                     <input class="form-control" name="password" id="password"
40                         type="password"><br>
41
42                 <div class="col">
43                     <label for="confirmPassword">Confirm Password</label>
44                     <input class="form-control" name="confirmPassword"
45                         id="confirmPassword" type="password"><br>
46
47             </div>
48
49             <div class="row">
50                 <div class="col">
51                     <div class="form-check">
52                         <input class="form-check-input" id="see-password"
53                             type="checkbox" onclick="showPassword();">
54                         <label class="form-check-label" for="see-password">See
55                             Password</label>
56
57                     </div>
58
59                 </div>
60
61             <small id="password-help" class="form-text text-muted">
62                 Your password must be longer than 8 characters, contain upper and
63                 lower case letters, numbers,
64                 and must not contain spaces, special characters, or emoji.
65             </small>
66             <input class="btn btn-primary btn-lg btn-submit" type="submit">
67
68         </form>
69
70     </div>
71 </div>

```

```
63
64
65
66
67
68 <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
       integrity="sha384-KJ3o2DKtIkYIK3UENzm7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
       crossorigin="anonymous"></script>
69 <script
       src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
       integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
       crossorigin="anonymous"></script>
70 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
       integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
       crossorigin="anonymous"></script>
71 </body>
72 </html>
```

# HomeView.ejs

Kieran Marcus Williams

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Home</title>
6      <link rel="stylesheet"
7          href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
8          integrity="sha384-Gn5384xqQlaoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm
9          crossorigin="anonymous">
10     <link rel="stylesheet"
11         href="https://use.fontawesome.com/releases/v5.6.3/css/all.css"
12         integrity="sha384-UHRTzL1+pbtHCWp1t77Bi1L4ZtiqrqD80Kn4Z8NTSRyMA2Fd33n5dQ81WUE00s/
13         crossorigin="anonymous">
14     <link rel="stylesheet" href="/stylesheets/HomeStyles.css">
15     <link rel="stylesheet" href="/stylesheets/NavbarLoggedInStyles.css">
16     <link rel="stylesheet" href="/stylesheets/NavbarNotLoggedInStyles.css">
17     <script src="/scripts/HomeScripts.js"></script>
18 </head>
19 <body>
20
21     <% include partials/NavbarLoggedIn.ejs %>
22
23     <!--contains a description of the site and links to register an account or log in-->
24
25     <div class="container">
26
26         <h1>Welcome To KW CRM</h1>
27
28         <br><br>
29
30         <div class="jumbotron">
31             <h3>About This Customer Relationship Management System</h3><br><br>
32             <p>Many large organisations will have their own bespoke software that they
33                 would use as a customer relationship management system (CRM). However, this
34                 site is
35                     a generic CRM that is web based and will allow for many businesses that
36                     are unable to afford the bespoke software to have access to the services
37                     provided by them.
38
39                     The problem with many of the other online CRM's is that they offer too
40                     many services and overcomplicate things for many end users. The aim of
41                     this system is to incorporate
42
43                     only what small business owners and freelancers need and not overwhelm
44                     users with an abundance of features.
45
46                     Other online CRMS also often cost a monthly subscription to use.</p><br>
47
48             <hr>
49             <br>
50             <h4><em>KW CRM is your best option for a free simple Customer Relationship
51                 Management System</em></h4><br><br>
52         </div>
53     </div>
54
55
56
57     <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
58         integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
59         crossorigin="anonymous"></script>
60     <script
61         src="https://cdn.jsdelivr.net/npm/@popperjs/core@1.12.9/dist/umd/popper.min.js"
62         integrity="sha384-ApNbgh9B+y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
63         crossorigin="anonymous"></script>
64     <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
65         integrity="sha384-JZR6Spejh4U02d8j0t6vLEHfe/JQGiRRQQxSffFWpi1MquVdAyjUar5+76PVCmYl"
66         crossorigin="anonymous"></script>
67
68 </body>
69 </html>

```

# IndividualContactView.ejs

Kieran Marcus Williams

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Individual Contact</title>
6
7      <link rel="stylesheet"
8          href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
9          integrity="sha384-Gn5384xqQlaoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
10         crossorigin="anonymous">
11
12     <link rel="stylesheet"
13         href="https://use.fontawesome.com/releases/v5.6.3/css/all.css"
14         integrity="sha384-UHRTZLI+pbxtHCWp1t77Bi1L4ZtiqrqD80Kn4Z8NTSRyMA2Fd33n5dQ81WUE00s/"
15         crossorigin="anonymous">
16
17     <link rel="stylesheet" href="/stylesheets/ViewIndividualContactStyles.css">
18     <link rel="stylesheet" href="/stylesheets/NavBarLoggedInStyles.css">
19     <link rel="stylesheet" href="/stylesheets/NavBarNotLoggedInStyles.css">
20
21 </head>
22 <body>
23
24 <% include partials/NavBarLoggedIn.ejs %>
25
26
27 <%--if any error or success messages need to be outputted to the user then they will
28 be entered in here--%>
29
30 <%if (flashMessage !== undefined){%>
31     <div class="alert alert-success">
32         <%= flashMessage %>
33     </div>
34 <%}%>
35
36 <%if (errorMessage !== undefined){%>
37     <div class="alert alert-danger">
38         <%= errorMessage %>
39     </div>
40 <%}%>
41
42
43 <%--this page will show all of the details of an individual contact--%>
44 <%--the page is vertically split in half--%>
45 <div id="halfway-separator">
46
47     <%--contacts main details are here--%>
48     <div id="left-side">
49         <h2><%= contactDetails[3] + " " + contactDetails[4]></h2><br>
50         <div class="contact-details">
51             <div class="jumbotron">
52                 <table>
53                     <tr>
54                         <td>
55                             <h5><span>Business: </span></h5>
56                         </td>
57                         <td>
58                             <h5><%= contactDetails[2]></h5>
59                         </td>
60                         <td>
61                             <h5><span>Email: </span></h5>
62                         </td>
63                         <td>
64                             <h5><%= contactDetails[5]></h5>
65                         </td>
66                     </tr><tr>
67                         <td>
68                             <h5><span>Phone Number: </span></h5>
69                         </td>
70                         <td>
71                             <h5><%= contactDetails[6]></h5>
72                         </td>
73                     </tr><tr>
74                         <td>
75                             <h5><span>Address: </span></h5>
76                         </td>
77                         <td>
78                             <h5><%= contactDetails[7]></h5>
79                         </td>
80                     </tr>
81
82     </div>
83
84 </div>
85 </body>
86
```

```

66
67             <td>
68                 <h5><span>City: </span></h5>
69             </td>
70                 <h5><%= contactDetails[7] %></h5>
71             </td>
72         </tr><tr>
73             <td>
74                 <h5><span>Postcode: </span></h5>
75             </td>
76                 <h5><%= contactDetails[8] %></h5>
77             </td>
78         </tr><tr>
79             <td>
80                 <h5><span>Address: </span></h5>
81             </td>
82                 <h5><%= contactDetails[9] %></h5>
83             </td>
84         </tr>
85
86
87
88     <!-- below code changes the data from the csv file for client
89     status into something that looks better
90     eg. current-client ===> Current
91     Client
92     -->
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123             </div>
124         </div>
125
126     <!--this text area form allows for the user to add a note which will get
127     displayed in the notes table to the left-->
128     <div class="add-note-section">
129
130         <form action="/contacts/addnote<%=contactDetails[0] %>" method="post">
131
132             <h3><label for="addNote">Add Note</label><br></h3>
133             <textarea name="addNote" onkeypress="return returnKeyBlocker(event)" id="addNote" cols="60" rows="7"></textarea>

```

```

133
134         <br>
135         <input class="btn-submit-add-note btn btn-primary" type="submit">
136     </form>
137
138     </div>
139
140     <!--edit contact button will redirect the user to the edit contact page with
141     that contacts details filled in-->
142     <!--the delete contact will ask for verification before deleting the
143     contact-->
144
145     <div class="contact-management-buttons">
146
147         <a class="btn btn-secondary btn-edit-contact"
148             href="/contacts/edit<%=contactDetails[0]%>">Edit Contact</a>
149
150         <form action="/contacts/delete<%=contactDetails[0]%>?_method=DELETE"
151             onsubmit="return confirm('Are You Sure You Want To Delete This
152             Contact');" method="post">
153             <button class="btn btn-danger btn-delete-contact">Delete
154             Contact</button>
155         </form>
156
157     </div>
158
159
160     <!--on the right side there is another text area that can be used to add, edit
161     data and view data-->
162
163     <div id="right-side">
164         <div class="extra-notes">
165             <br><br><br>
166
167             <!--the place holder will display instructions as to how to save,
168             encrypt and decrypt the data as
169             it is not super intuitive at first -->
170             <form
171                 action="/contacts/viewContact<%=contactDetails[0]%>/encryptNotes_encrypt"
172                 method="post">
173                 <h3><label for="extraNotes">Extra Notes</label></h3>
174                 <textarea name="extraNotes" id="extraNotes" onkeypress="return
175                 returnKeyBlocker(event)" cols="80" rows="10"
176                 placeholder="">
177
178                 REMEMBER THE PASSWORD! There Is No Way Of Getting The Data Back If
179                 You Forget It. Type Data into text area and press save & encrypt to save the data to
180                 the file with xor encryption done to
181                 to it. this data can be changed by first decrypting it, changing the data and then
182                 pressing the save & encrypt button again.
183                 When data is decrypted it will not save anything back to the file. If the wrong
184                 decrypt password is entered, the decrypted text will
185                 not be the original text
186             ><%=extraNotes%></textarea><br>
187
188             <!--when the save and encrypt btn is pressed, the data in the text
189             field will get encrypted and saved to the database-->
190             <!--when the decrypt button is pressed the data will get retrieved
191             from the database and outputted to the text area-->
192             <!--a password page will appear before the encryption or decryption
193             happens though-->
194             <div class="btn-container-extra-notes">
195
196                 <div id="extra-notes-btn-group">
197
198                     <input class="btn-submit-extra-notes btn btn-secondary"
199                         type="submit" value="Decrypt"
200                         formaction="/contacts/viewContact<%=contactDetails[0]%>
201                         /encryptNotes_decrypt" formmethod="post">
202
203                     <input class="btn-submit-extra-notes btn btn-primary"
204                         type="submit" value="Save & Encrypt">
205
206             </div>
207         </div>
208
209     </div>
210
211 
```

```

185                         </div>
186
187                         </div>
188                     </form>
189                 </div>
190
191             <br><br><br><br><br>
192
193             <!--this is a table that will display all the notes and the date that each
194             note was created next to it -->
195             <div id="table-holder-notes">
196                 <table class="table ">
197                     <thead class="thead-grey">
198                         <tr>
199                             <th>Date</th>
200                             <th>Notes</th>
201                         </tr>
202                     </thead>
203
204                     <tbody>
205
206                         <%contactNotes.forEach( (noteRecord) => {%
207
207                             <tr>
208                                 <!-- the notes creation date gets put into here-->
209                                 <td class="td-borededer">
210                                     <%=noteRecord[2]>
211                                 </td>
212                                 <!--the note data will be in here-->
213                                 <td class="td-borededer">
214                                     <%=noteRecord[4]>
215                                 </td>
216
217                             </tr>
218
219                         <%});%>
220
221                     <tbody>
222                 </table>
223             </div>
224
225
226         </div>
227
228     </div>
229
230
231     <script src="/scripts/ViewIndividualContactScripts.js"></script>
232     <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
233         integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
234         crossorigin="anonymous"></script>
235     <script
236         src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
237         integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4O"
238         crossorigin="anonymous"></script>
239     <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
240         integrity="sha384-JZR6Spejh4U02d8j0t6vLEHfe/JQGiRRSQQxSffFWpi1MquVdAyjUar5+76PVCmYl"
241         crossorigin="anonymous"></script>
242
243     </body>
244
245 </html>

```

# SettingsView.ejs

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Terms And Conditions</title>
6      <link rel="stylesheet"
7          href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
8          integrity="sha384-Gn5384xqQlaoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm
9          crossorigin="anonymous">
10     <link rel="stylesheet"
11         href="https://use.fontawesome.com/releases/v5.6.3/css/all.css"
12         integrity="sha384-UHRTZLI+pbxtHCWplt77Bi1L4ZtiqrqD80Kn4Z8NTSRyMA2Fd33n5dQ81WUE00s/
13         crossorigin="anonymous">
14     <link rel="stylesheet" href="/stylesheets/SettingsStyles.css">
15     <link rel="stylesheet" href="/stylesheets/NavbarLoggedInStyles.css">
16     <link rel="stylesheet" href="/stylesheets/NavbarNotLoggedInStyles.css">
17     <script src="/scripts/SettingsScripts.js"></script>
18 </head>
19 <body>
20
21 <% include partials/NavbarLoggedIn.ejs %>
22
23 <!--This page will display links to the edit account page and to the Terms and
24 Conditions Page-->
25 <h1>Settings</h1>
26 <br><br><br>
27 <div class="jumbotron">
28
29     <!-- UNCOMMENT WHEN IMPLEMENTATION OF USERS IS ADDED IN -->
30     <!--<h4><a href="#">Edit Account Details</a></h4>-->
31
32     <br>
33
34     <h4><a href="/termsAndConditions">Terms And Conditions</a></h4>
35 </div>
36
37 <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
38     integrity="sha384-KJ3o2DKtIkY1K3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
39     crossorigin="anonymous"></script>
40 <script
41     src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
42     integrity="sha384-ApNbgh9B+y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
43     crossorigin="anonymous"></script>
44 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
45     integrity="sha384-JZR6Spejh4U02d8j0t6vLEHfe/JQGiRRQQxSffFWpi1MquVdAyjUar5+76PVCmYl"
46     crossorigin="anonymous"></script>
47 </body>
48 </html>

```

# TermsAndConditionsView.ejs

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Terms And Conditions</title>
6      <link rel="stylesheet"
7          href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
8          integrity="sha384-Gn5384xqQlaoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm
9          " crossorigin="anonymous">
10     <link rel="stylesheet"
11         href="https://use.fontawesome.com/releases/v5.6.3/css/all.css"
12         integrity="sha384-UHRTZL1+pbxtHCWp1t77Bi1L4ZtiqrqD80Kn4Z8NTSRyMA2Fd33n5dQ81WUE00s/
13         " crossorigin="anonymous">
14     <link rel="stylesheet" href="/stylesheets/TermsAndConditionsStyles.css">
15     <link rel="stylesheet" href="/stylesheets/NavbarLoggedInStyles.css">
16     <link rel="stylesheet" href="/stylesheets/NavbarNotLoggedInStyles.css">
17 </head>
18 <body>
19
20 <% include partials/NavbarLoggedIn.ejs %>
21
22 <br><br>
23 <h1>Terms And Conditions</h1>
24 <br><br>
25 <div class="jumbotron">
26     <h5>By Using This Website You Are Agreeing To These Terms And
27     Conditions.</h5>
28     <br>
29     <p>
30         Any data held on your contacts is your responsibility. <br>
31         You agree to be liable for any GDPR law breeches that are related to the
32         contacts you have created. <br><br>
33
34         If you wish to receive data that we hold regarding your usage of the system
35         <br>
36         you must make a formal request in the form of an email to
37             kieranmarcus@KWCrm.com
38     </p>
39
40 </div>
41
42
43
44 <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
45         integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
46         crossorigin="anonymous"></script>
47 <script
48     src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
49     integrity="sha384-ApNbgh9B+y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
50     crossorigin="anonymous"></script>
51 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
52     integrity="sha384-JZR6Spejh4U02d8j0t6vLEHfe/JQGiRRSQQxSffFWpi1MquVdAyjUar5+76PVCmYl"
53     crossorigin="anonymous"></script>
54
55 </body>
56 </html>
```