HTML & Javascript Project,

**Malcolm John Russell B00287827**

**Josh McKie B00288826**

**Lewis Barber-Fleming B00257219**

# 

[**Introduction**](#_aubhnr4me4ga)[**4**](#_aubhnr4me4ga)

[**The Team**](#_fkqn1mlpr29r)[**4**](#_fkqn1mlpr29r)

[**2. Project Specification**](#_y9495aucqca0)[**4**](#_y9495aucqca0)

[2.1 Store Contacts](#_h3diw28jv8tb) [4](#_h3diw28jv8tb)

[2.2 Remove Contacts](#_t772u74wdp1) [4](#_t772u74wdp1)

[2.3 Sort Contacts](#_99jmf2tnqxdw) [5](#_99jmf2tnqxdw)

[2.4 Page Settings](#_qc7728gu7cs8) [5](#_qc7728gu7cs8)

[2.5 Contacts Backup](#_3pa52r8n9ntd) [5](#_3pa52r8n9ntd)

[2.6 Restore Contacts](#_rhg1l6al8gnz) [5](#_rhg1l6al8gnz)

[2.7 Block List](#_ggod7pcpzbos) [5](#_ggod7pcpzbos)

[2.8 About](#_i4qeido3vnx9) [5](#_i4qeido3vnx9)

[2.9 Contact Us](#_glu7sv6hf5o8) [6](#_glu7sv6hf5o8)

[2.10 Give Feedback](#_2bzxtcw8sz0e) [6](#_2bzxtcw8sz0e)

[2.11 Testing](#_62dnp3b7w6ll) [6](#_62dnp3b7w6ll)

[**3. User Interface Example**](#_etne6wf2vmha)[**6**](#_etne6wf2vmha)

[3.2 Styling Choices](#_cz3c7x4ijbut) [9](#_cz3c7x4ijbut)

[**4. Design Documentation**](#_fapvdl3rylj0)[**10**](#_fapvdl3rylj0)

[Objects:](#_rsj5ueuq6bv3) [10](#_rsj5ueuq6bv3)

[Contact Object](#_sz7aj4ill4vj) [10](#_sz7aj4ill4vj)

[Contact Functions](#_lqz129g5teup) [10](#_lqz129g5teup)

[Contact IsBirthDay()](#_xyimotszob8y) [10](#_xyimotszob8y)

[Contact ChangeImage()](#_7keycdtp56hb) [10](#_7keycdtp56hb)

[Contact AddContact()](#_5j2k2wrp1gy3) [10](#_5j2k2wrp1gy3)

[Contact RemoveContact()](#_h6j21dsycapg) [10](#_h6j21dsycapg)

[Contact GetCurLocation()](#_2kgrj8k9lysp) [11](#_2kgrj8k9lysp)

[Contact AddAddress()](#_zhjk5lm4d8cp) [11](#_zhjk5lm4d8cp)

[Contact ChangeAddress()](#_mk0kd3vozt4t) [11](#_mk0kd3vozt4t)

[Contact RemoveAddress()](#_51su17eqk9db) [11](#_51su17eqk9db)

[Contact ViewMapAddress()](#_s3sjwxjmzfhq) [11](#_s3sjwxjmzfhq)

[Contact AddPhoneNo()](#_4bskq5a166i8) [11](#_4bskq5a166i8)

[Contact ChangePhoneNo()](#_wa3wv121uk5z) [11](#_wa3wv121uk5z)

[Contact RemovePhoneNo()](#_dy6tx7qbsywf) [11](#_dy6tx7qbsywf)

[Contact AddEmailAddress()](#_58x0ny1mcu9h) [11](#_58x0ny1mcu9h)

[Contact ChangeEmailAddress()](#_bzdna8a5enas) [11](#_bzdna8a5enas)

[Contact RemoveEmailAddress()](#_rf49vjrcua0r) [11](#_rf49vjrcua0r)

[Group object](#_srbiyqa6rv8) [12](#_srbiyqa6rv8)

[Group Object Functions](#_a7d2hdapc6t3) [12](#_a7d2hdapc6t3)

[Group AddGroup()](#_fpk7si1iu8ur) [12](#_fpk7si1iu8ur)

[Group RemoveGroup()](#_f1yz5mfp0si9) [12](#_f1yz5mfp0si9)

[Group ChangeImage()](#_fob6ib72iwhf) [12](#_fob6ib72iwhf)

[Group AddContact()](#_7gqtvpfays10) [12](#_7gqtvpfays10)

[Group RemoveContact()](#_lb3ljs1teatu) [12](#_lb3ljs1teatu)

[Non object functions:](#_3nh9diikojzx) [13](#_3nh9diikojzx)

[Contacts[]](#_o9qg7lehcmr0) [13](#_o9qg7lehcmr0)

[Groups[]](#_74fo4dxphn6b) [13](#_74fo4dxphn6b)

[search()](#_h5lo23vjqovd) [13](#_h5lo23vjqovd)

[setUser()](#_6eywh7i6qqn1) [13](#_6eywh7i6qqn1)

[saveSettings()](#_pc7km1ykr327) [13](#_pc7km1ykr327)

[saveContactsAndGroups()](#_jk5ms8v93wtk) [13](#_jk5ms8v93wtk)

[showContacts()](#_oslbo3i1solq) [13](#_oslbo3i1solq)

[MoveList()](#_c9juczc722nc) [13](#_c9juczc722nc)

[showGroup():](#_lqf11jx044kg) [13](#_lqf11jx044kg)

[login():](#_3hd0xs167kow) [13](#_3hd0xs167kow)

[ContactArrayToString()](#_1zniguzd8sss) [14](#_1zniguzd8sss)

[ContactStringToArray()](#_5z76j1ft3on9) [14](#_5z76j1ft3on9)

[**5. Our future aims**](#_1t13m2v8avit)[**14**](#_1t13m2v8avit)

# 

# Introduction

This report will go over the current content of our web app, descriptions of the current and future functionality of the backend, a description of our current user interface and functionality that we plan to implement in the future.

# The Team

The members of our team and their responsibilities going forward are:

B00288826: CSS/JS styling and Javascript Functionality Coding

B00257219: Documentation and Testing and Javascript Functionality Coding

B00287827: HTML coding and Javascript Functionality Coding and Documentation

# 2. **Project Specification**

We have included a working contacts web app. The app will provide functionality relevant to a contacts book. The purpose of which is to help the user manage and maintain their contacts.

The app will have the following functionality:

## 2.1 **Store Contacts**

The user will have the ability to create new contacts to add to their book. These contacts will contain details such as email address, physical address, home and mobile phone numbers. This is important because the user can create the contact how they want it, with whatever information they want.

## 2.2 **Remove Contacts**

It’s important to be able to edit and remove contacts if the information is no longer correct or relevant. The user will have the ability to delete any contacts they no longer wish to keep from their contacts book.

## 2.3 **Sort Contacts**

The user will have the ability to choose how their contacts will be displayed on the page. The displayed contacts can be sorted by the user’s choosing. This provides a more personal experience for the user. This functionality also allows the user to browse through their contacts as they please.

## 2.4 **Page Settings**

The user will be able to have a more personalised experience on the web app. This is important for accessibility, as users with disabilities need to be able to access the web app as easily as others.

## 2.5 **Contacts Backup**

The user will have the functionality to backup their contacts by exporting them to a document such as a CSV to then download. This allows the user to store their contacts permanently between sessions.

## 2.6 **Restore Contacts**

The user will have the option to import a file such as a CSV that they downloaded with the contacts backup feature, this will then import the data from the CSV into their contacts list. This functionality makes the app great for moving between workstations, as the contacts list can be copied over. It also allows for backup solutions for users data.

## 2.7 **Block List**

The user will be able to keep track of contacts they do not wish to hear from, such as spammers. This allows the user to keep any unwanted people from getting in touch with them or viewing their location via the map integration.

## 2.8 **About**

The about section will display meta information on the web app, such as version number. It will also feature information about the creators. There will also be information on what the app is for, an FAQ and a feature of instructions on how to best utilise the app.

## 2.9 **Contact Us**

This section will allow the user to get in touch with the creators via email, this is helpful if they have any questions regarding the web app.

## 2.10 **Give Feedback**

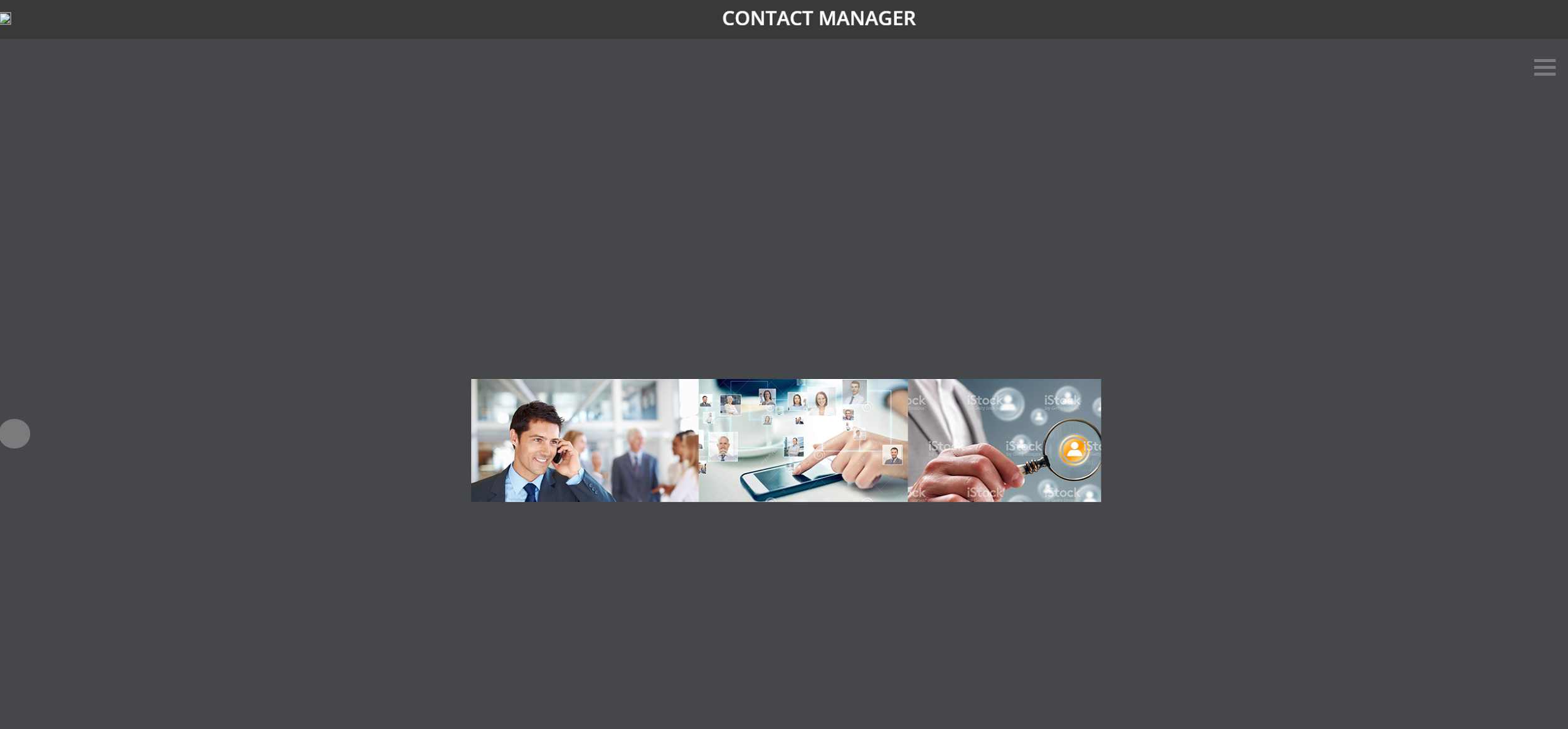
The feedback section is in place so that we can improve the web app based on user input. Being able to change features and display of the web app based on user experience will help make the app more user friendly and popular.

## 2.11 **Testing**

The methods for testing used will be making sure each button expands to show only the content that it is supposed to. Testing the creation of contacts objects in the future will be done by making sure only correct data can be input into a field, e.g. you can’t put “hello” in as a phone number. All tests will be carried out over multiple browsers, (IE, Firefox, Chrome, Safari).

# 3. **User Interface Example**

The screenshot shows the two menu buttons positioned at the far left and right of the screen, when clicked, these buttons will expand the menus from either side of the screen. Upon clicking the circle div on the left side, it will expand a div that contains contact functionality and information, and at the same time, the circle div will move the same amount to the right. The right hand side menu is expanded by clicking on the button in the top right which will hide the middle line(div) and transform the top and bottom lines(also divs) to form a red cross. As the button transforms into a cross, a div containing the menu functions and options will slide out from the right hand side of the screen. The missing image at the top left of the page is a placeholder for what will be a logo for the web app in the future.

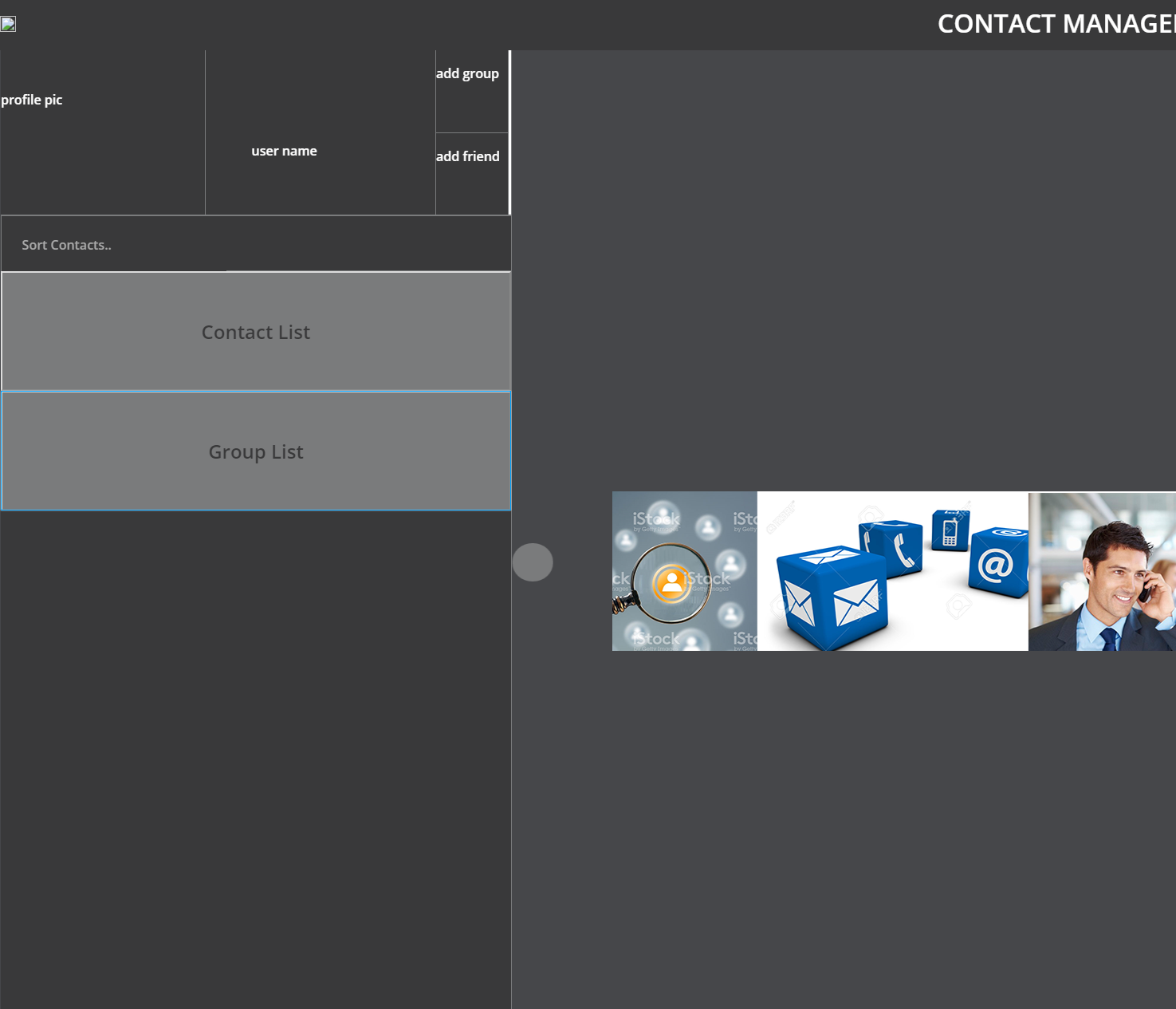


The left hand menu (below) shows the contact and group lists, if the user clicks on the contact list, it expands towards the bottom of the page, moving the group list down the same amount and at the same speed. The expanded contacts list div will show the user’s currently stored contacts. The contact list div can be clicked on again to shrink back up.

The groups list can be expanded like the contacts list by clicking on the group list div, this will expand a hidden div underneath it containing group information that is stored/created by the user.

The three coloured divs at the top of the contact page are placeholders for a profile picture, the currently logged in user's name and an add contact button. The add contact button will bring up a new div to add the new contact’s details in.

The left circular button can be clicked on again to hide the left contacts div. The div and button both move left at the same speed and to the same relative position.

The right hand menu (below) will show options and functions that the user can perform; adding contacts, removing contacts, blocking contacts.

Clicking page settings will bring up a div in the middle of the page that will contain the options for the page. The “img” text next to it is a placeholder for an image to be put in at a later date.

Clicking on back up contacts will bring up a div in the middle of the page that will allow the user to export the contact list to a CSV file. The user will then be able to download this CSV file to their own computer.

Clicking on restore contacts will open a div in the middle of the screen, in this div the user will be able to upload a CSV file that will have existing contacts saved on it. The contact manager web app will then create the contacts in the app from this CSV file.

Manage Block list will open a div in the middle of the screen that displays currently blocked contacts.

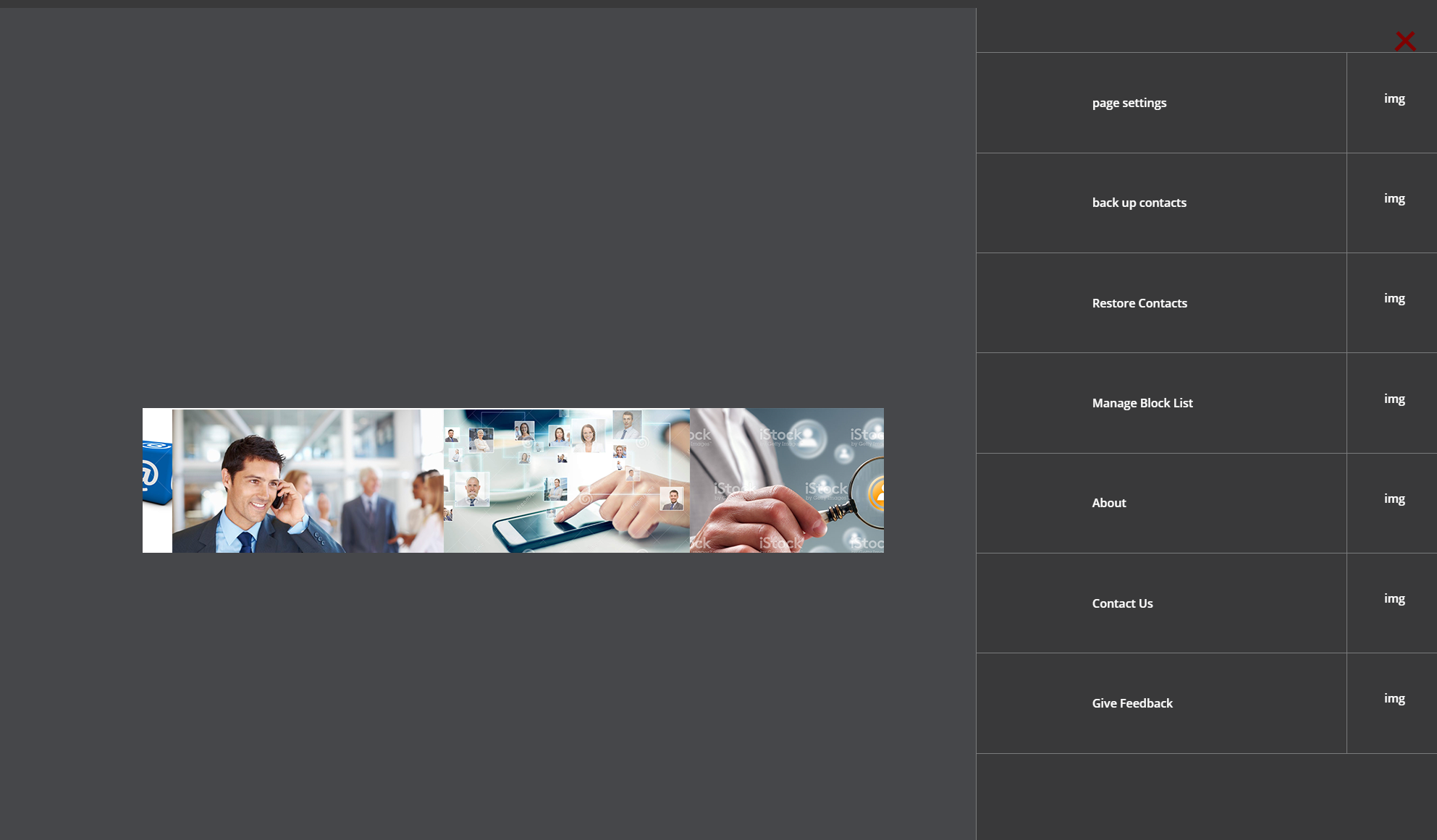
Clicking on about will open a div in the middle of the screen to display information about the web app.

Clicking on contact us will open a mailto link to email the creators.

Clicking on give feedback will open a div in the middle of the screen with options to provide feedback.

All of the “img” divs are placeholders for images that will be implemented at a later date.

In the middle of the screen there is a scrolling image.

On the bottom-middle of the screen there will be the cookie disclaimer.

## 3.2 **Styling Choices**

We decided to go for a modern, yet minimalistic theme for our web app. The menus that are initially hidden allow for a dynamic user experience, what is displayed can be only what is needed to perform each action.

The colour scheme we decided to go for what that monochromatic grey, we find that this helps the text (white) stand out against the background (greys). The borders on the divs that make up the left and right pop out menus fit with this grey/white colour scheme and look.

The reason there aren’t very many links is to keep with the minimalist theme.

We have decided to use percentage scaling divs for most of our content to provide accessibility, as most of the modern world has access to the internet over many mediums such as smartphones, tablets, laptops and computers. There are too many different sizes of screens used in the modern day to have everything sized to exact pixels. This scaling ability can also help the vision impaired, as the display can be scaled up for easier viewing.

We are going to attempt to add keyboard and mobile accessibility where possible.

We chose the font “Open Sans” because it also fits with our modern design.

# 4. Design Documentation

## **Objects:**

### **Contact Object**

The properties of the contact object are:

Contact display name: this is the name that is used to identify the contact

Contact name[]: this array will hold the contact's full name

Contact address[][]: this multidimensional array will store the full address, and any additional address of the contact.

Contact Image: this is to represent the contact in image form it will be set by the contact themselves, if no image is set it will be the default guess image.

Contact D.o.B: this is the date of birth of the contact

Contact Email[]: this array will store all the email addresses of the contact

Contact phone No.[]: this array will store all the phone numbers given of the contact

Contact Notes: this is where the user can store notes on the contact

### **Contact Functions**

The functions of contact are:

#### **Contact IsBirthDay()**

This on the load of the webpage will compare the current date with the D.o.B and returns if it is the contact's birthday.

#### **Contact ChangeImage()**

This will change the user image

#### **Contact AddContact()**

This is to add a new contact to array

#### **Contact RemoveContact()**

This is to remove a contact from the array

#### **Contact GetCurLocation()**

This will return the contacts current position if available

#### **Contact AddAddress()**

This will add a new address to the array.

#### **Contact ChangeAddress()**

This will allow the user to update and change existing address in the array

#### **Contact RemoveAddress()**

This will remove an address chosen from the array

#### **Contact ViewMapAddress()**

This will bring a map for the addressed that is clicked

#### **Contact AddPhoneNo()**

This will add a new phone number to the array.

#### **Contact ChangePhoneNo()**

This will Allow the user to update or change an existing phone number in the array

#### **Contact RemovePhoneNo()**

This will remove a chosen phone number from the array

#### **Contact AddEmailAddress()**

This will Add a new Email Address to the array

#### **Contact ChangeEmailAddress()**

This will allow the user to update or change an existing Email Address within the array

#### **Contact RemoveEmailAddress()**

User can delete a chosen email address within the array.

### **Group object**

The properties of the group object are:

Group Name: this is the name of the group

Group Contacts[]: this array is to store all the contacts that belong to this group

Group Image: this is the image that is assigned to the group by the user.

### **Group Object Functions**

The Functions of group are:

#### **Group AddGroup()**

This adds a new group to the Group array

#### **Group RemoveGroup()**

This removes a group from the array

#### **Group ChangeImage()**

This changes the image for the group

#### **Group AddContact()**

This is to add a contact to array

#### **Group RemoveContact()**

This is to remove a contact from the array

### 

### 

### 

### **Non object functions:**

#### **Contacts[]**

This is an array to store all the contact objects

#### **Groups[]**

This is an array to store all the group objects

#### **search()**

This will compare the input from the search form to each contact name in the contacts array and then each member of each group which will return a new search results array with the required information.

#### **setUser()**

When the page loads this will set the user if the user is logged in or not. By default it will be set to guest.

#### **saveSettings()**

This will save the user settings such as name changes etc.

#### **saveContactsAndGroups()**

When the paged is closed this will save any changes to the groups or contacts such as if a new contact is added.

#### **showContacts()**

This expands the contact drop down menu

#### **MoveList()**

This moves the divs in the contacts menu down

#### **showGroup():**

This expands the group drop down menu

#### **login():**

This will be used to load the user's personal settings and reload the page

#### **ContactArrayToString()**

This will save the contact array to string file

#### **ContactStringToArray()**

This will import the file and read the string and store the contact in the contact array

# 5. Our future aims

In future versions of our web app we plan to implement more features. The website will handle personalised user functionality such as a login system. The user will be able to create contacts objects to store. They will be able to search for their contact's home address on a map. We plan on integrating Google Maps into a div that will pop up when the user clicks on a contact, the div will show the contact’s current location.