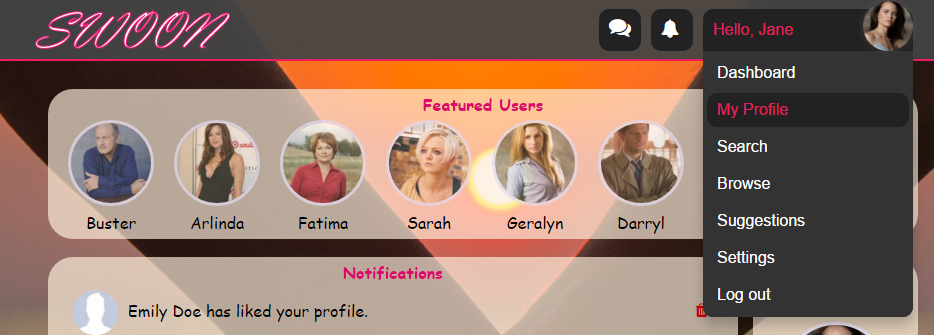
**Page descriptions and functionality:**

**Login/Registration:**

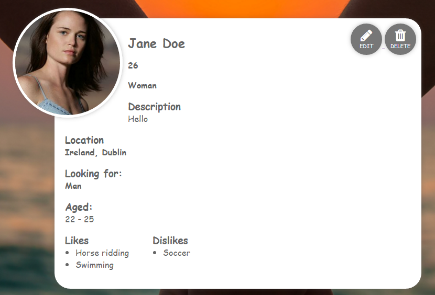
This is where you either login or register for the site. To login you enter your accounts email address and password. To register you click the register button and you enter your email, password, confirm your password, first name, last name, date of birth and sex. Form validation and error messaging is handled with AJAX requests.

**Header:**

 The header display the website logo, messages, notifications, navbar and displays the users profile image. Hovering over the message bar displays who you are messaging. By clicking on the messages it brings up chat for that person, it displays the amount of unseen messages you have as well. Hovering over notifications displays what notifications you have, also it displays the amount of unseen notifications you have. Hovering over the navigation bar displays the pages that you can navigate to. These are generated based on the user’s permissions/role, so an admin will see different options. The admin navigation bar doesn’t show ‘My Profile’, or ‘Suggestions’ because admins can’t have a profile.

**Dashboard:**

The user dashboard has featured users this displays 8 users profile pictures and their names. You can click on their picture to view there profile. You also have your profile overview it displays your profile picture, name and age you can click on the profile to see exactly what other users will see. Notifications displays who liked you, who liked you back and if your ban has been lifted. To set a notification to seen simply click on it and to delete it click on the red bin icon.

 The admin dashboard displays user reports. From here admins can see why certain users have reported others (in a modal window fetched with AJAX) and they can decide what actions to take.

**Profile:**

This page displays your profile this is what other users will see when they click on your profile. It displays your name, age, sex, description, location, sex you are looking for, age range, likes and dislikes. From here you can edit your profile or delete your own profile. Since the site is permission based, if an admin visits this page they are also given the option to edit and delete, as well as to block the user. Other users may like, block or report the user profile. If they have both liked each other, an option to open chat is presented.

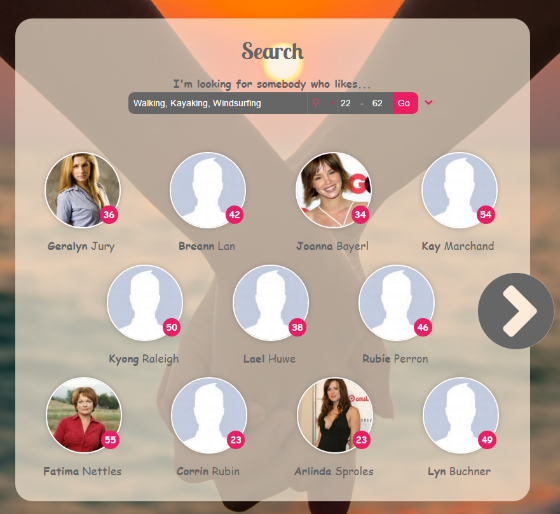
Profile is not displayed to an admin (unless it is that of another user) because they cannot have a profile.

**Edit-profile:**

From here you can edit your profile including your profile picture name, age, sex, description, location, sex you are looking for, age range, likes and dislikes. There is no limit to how many or what you like or dislike. Likes and dislikes are added to the list when the user taps enter using AJAX. The picture uploaded is resized, cropped and thumbnails are generated. The fields are also sanitised and validated.

Admins can edit any profile they want.

**Search:**

From here you can search through users. You can specify the age range you are looking for, sex, your interests, location and name (age range and sex are loaded by default based on your interests if you have a profile). You can enter a list of interests either with or without commas. The users who have more in common with you in terms of interests will be at the top of the search results.

A score is determined on matching other user’s likes (1 if they like it too, -0.5 if they don't). A user is NOT required to have a match for all likes listed, but if their match score falls below zero, they are no longer listed. The other search fields are strict in that the users returned must match the location, sex and age range provided.

**Browse:**

Here all the users from the website are displayed.

**Manage Blocked Users:**

Here a user may browse through users that they have blocked, in case they want to unblock them.

**Suggestions:**

These are all the users who we feel are best matched to your profile. Like in search, a score is determined where the score is determined on matching other user’s likes (1 if they like it too, -0.5 if they don't) but also other user’s dislikes (1 if they dislike it too, -0.5 if they don't).

Suggestions are not displayed to an admin because they cannot have a profile.

**Users can paginate through results left and right. Pagination is handled with AJAX and the previous page is cached and the next page is preloaded. A fall back is also provided if there is no JavaScript. (Search, Browse, Manage Blocked Users and Suggestions share the same base, search.php)**

**When on a user’s profile:**

From the results, the user can click on a profile and this will display the users name, age, sex, description, location, sex they are looking for, age range, likes and dislikes. You can like there page, block them, report them and if you have both liked each other you can chat with them. When you click block their profile will never show up on search results and they will no longer be able to see your profile or contact you again unless you unblock them via the settings page. When you click on report you will be prompted to fill out a form saying why you are reporting this user.

**Settings:**

From her you can change your account settings. You can change your email by entering your new email twice. You can enter a new password by entering your new password twice. To change either of the above you have to enter your current password. The data is validated and error messages are displayed.

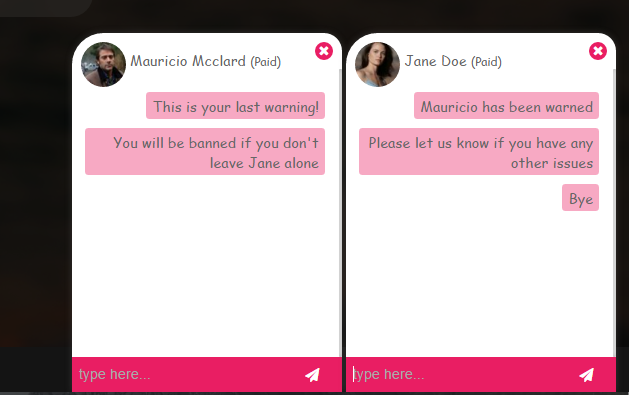
There is a button ‘manage blocked users’ from here you can unblock users that you’ve blocked.

When an admin is logged in a button is displayed saying ‘Add another admin’.

**Add-admin:**

When an admin is logged in the admin can add another admin by typing in their email, the admins password, confirming the admin password, first name and last name.

**Payment:**

This page is where you upgrade your profile from a free account to a paid account. You enter your full name and card details to pay. The data is validated and error messages are displayed using AJAX. Once the payment has been accepted you are instantly upgraded to a paid account.

**Chat:**

When to users have both liked each other they are able to chat with each in real time using a Facebook like chat system. You can have several chat windows up at the same time. Messages are refreshed every 10 seconds and they are sent using AJAX.

Admins can chat with any user.

**User functionality:**

**Super Admins:**

Can’t be deleted. They have the same functionality as normal admins, only they can also add admins.

**Admins:**

Admins can’t have a profile. Admins can delete, ban and edit other user’s profiles, they can chat with any other user.

**Paid users:**

Paid users can create a profile and edit there profile. They can interact with other profiles this includes liking, blocking, reporting and chatting.

**Free users:**

Have to become paid users to create a profile. They will be prompted to upgrade when trying to access certain features.

**Banned users:**

Can’t login till there ban has expired.

**Blocked/deleted users:**

These users will never be allowed to login again.

**List of functions and scripts:**

**AJAX:**

**Add\_remove\_interests.php**

Facilitates AJAX requests for adding or deleting an interest (like/dislike) using the interest's ID

**Delete\_notification.php**

Facilitates AJAX requests for deleting a notification with a specified ID

**Get\_chat.php**

Facilitates AJAX requests for loading the chat, getting the messages in the chat and sending messages. The chat will only load if the users have permission to chat with each other.

**Get\_profile.php**

Facilitates AJAX requests for loading a user’s profile (for display in a modal window) based on a specified ID. If the user\_id is set and the profile exists the user\_id is passed into get\_profile().

**Get\_report.php**

Facilitates AJAX requests for loading a user report (for display in a modal window) based on a specified ID.

**Report\_ban\_user.php**

Facilitates AJAX requests for loading a user report/admin ban form (for display in a modal window) and processes reports/bans. An admin can only ban somebody with a role weight lower than theirs. A notification is sent to the admins if a report is made.

**Set\_notification.php**

Facilitates AJAX requests for setting a notification as seen based on a specified ID.

**Set\_relationship.php**

Facilitates AJAX requests for setting a relationship status with another user based on a specified user ID.

**Validate\_form.php**

Facilitates AJAX requests for validation of a specified form field. Errors are returned in JSON format. If no errors are returned, 'success' is returned.

**Core:**

**DB:**

**Connect.php**

**-** Connects to the database using mysqli.

**PopulateDB.php**

- A script that populates the database with users.

- Generates 200 random BELIEVABLE users with names matching gender and a random set of likes/dislikes.

**Func:**

**Functions.php**

* Get\_header()

– Displays the site header

* Get\_footer()

– Displays the site footer

* Get\_profile\_image($size, $user\_id = null)

– gets the profile image of a user the function takes the size (e.g. THUMB, MEDIUM, LARGE) and the user id, and returns the image for that user.

* Check\_password($pass)

– Checks if the password for the current user is correct. We hash the users password for security. The function returns a Boolean value whether the password is correct or not.

**Image-resize.php**

Crops and down-samples an image of type png or jpg to a specified height and width

* imageResize($target, $newcopy, $w, $h, $ext)

– This resizes the image to the specific width and height that is pass into the function it also keeps the aspect ratio of the image that the user provided.

**Image-upload.php**

Processes image upload, checks for errors such as file size limit and extension and generates various image sizes

* This is called when the user trys to upload the image and it makes sure that the user uploads an image with the correct extension, it is under 5mb and there is no errors uploading the image from there imageResize function is called.

**Interests.php**

Functions for adding, removing and getting a users like/dislikes

* add\_interest($user\_id, $likes, $content)

– This function is called to add a like/dislike to a profile. It checks the user’s permission to see if he/she can edit it.

* remove\_interest($user\_id, $interests\_id)

– This function is used when a user removes a like/dislike from a profile.

* get\_interests($user\_id, $likes = null)

–The user id passed in likes is a Boolean value true for likes and false for dislikes a query is then used to get the likes/dislikes for the specified user id. The likes/dislikes are then returned in an array.

**Messaging.php**

Chat functions for sending, getting and setting messages as seen

* send\_message($user\_id, $content)

– takes in the user id (receiver\_id) and the content of the message and adds the data to the database.

* get\_messages($user\_id, $seen = null)

– gets the messages for a user from another user binds the results of the query to messages array and returns the array.

* set\_messages\_from\_user\_seen($sender\_id)

– Sets the messages sent to the current user as seen for a particular chat by updating the bit value to TRUE (1) in the database.

* get\_latest\_messages()

- Gets all the latest messages, with the other users name and the amount of unread messages.

**Navigation.php**

* \_\_construct($title, $link=null, $class=null, $counter=null, $extra\_html=null, $display=null)

- Creates a menu object with a display title, link, class list, counter and extra html such as an icon.

* create\_navigation\_menu\_items($menu\_items)

- Recursively generates multilevel menus from a list of menu objects

**Notifications.php**

* Get\_unseen\_notification\_count($user\_id)

– Counts the number for notifications that haven’t been seen by a user and returns the count.

* delete\_notification($notification\_id)

– Deletes the notification from the database the where notification id is passed into the function and returns true or false.

* set\_notification\_seen($notification\_id)

– Sets the notification to seen by updating the bit value to TRUE(1) in the database.

* get\_notifications($user\_id)

– Gets the notifications for a user for the last 30 days the results are put into notifications array the array is returned.

* remove\_old\_notifications()

– Deletes notification from a database if they are older than 30 days.

* create\_notification($receiver\_user\_id, $content, $type)

– Takes the receiver id the content and the type and uses a switch case statement to change the content of the notification if needed. All the relevant information is then put into the database.

* find\_type\_id($type)

– This function queries the database and finds out the type id of the notification.

* get\_report\_notifications($notification\_id = null)

– Gets all the report notifications this is used for the admin dashboard. The results are put into a reports array and that array is returned.

* truncate($string, $length, $dots = "...")

– This function takes the string and the length and truncates the string with a “…” should the string exceed the specified length.

* get\_unseen\_report\_notification\_count()

– Used for the admin dashboard and counts the unseen report notifications and returns that count.

**Profiles.php**

* \_\_construct($user\_id)

- Initialises a profile object for a user ID.

* error\_push($error)

– Stores errors for later access if a problem should occur.

* create\_profile($DOB\_day, $DOB\_month, $DOB\_year, $sex)

– Creates profile and inserts user\_id, DOB, sex, looking\_for and date and time updated into profiles.

* fetch()

- Fetches the profile of the user. The result is stored in the Profile object.

* submit()

– Updates a profile like first name and last name, DOB, sex, description, location, looking for, min age and max age.

* get\_profile($user\_id)

– This gets the profile of a user and returns it.

* exists\_profile($user\_id)

– This takes the $user\_id and checks if the profile for this user exists and returns TRUE or FALSE.

* delete\_profile($user\_id)

– Deletes the profile of a specified user. It first checks the user has permission to do this.

* query\_add($query, $stmt\_part, $param\_value = null, $param\_type = null, $join\_part = null, $end\_part = null)

- Helper function which builds query parts ready for use with prepared statements.

* get\_profiles($query\_stmt\_parts, $query\_param\_values, $query\_param\_types, $query\_join\_parts, $query\_end\_parts)

- Gets profiles based on the query parts passed. An SQL injection safe, variable parameter length, query is built using prepared statements. A list of users with names, DOB and location are returned. This is used by search, browse and suggestions to build their queries.

* get\_random\_profiles($user)

– This function is used for the user dashboard and gets 8 random profiles and returns the user\_id and the first\_name in rUsers array.

**Roles.php**

Functions for user permissions/roles (where $user\_id=null, the default is the current user)

* user\_can($permission, $user\_id = null)

– Checks if the user has a permission to do whatever is specified in $permission (e.g ban\_users, edit\_profile) the database is queried and a bit value is returned.

* get\_user\_role\_name($user\_id = null)

– Returns the name of the role of the users.

* get\_user\_role\_weight($user\_id = null)

– Gets a user’s role weight.

* get\_role\_weight($role)

– Gets the weight of a role.

* user\_is\_role($role, $user\_id = null)

– Checks if a user has a certain role.

* user\_is\_at\_least\_role($role, $user\_id = null)

– Checks if a user has at least a certain role.

* set\_user\_role($role, $user\_id = null)

– Updates the user to a role which is passed into the function.

* ban\_user($reason, $duration, $user\_id)

– Bans a user for a duration (changes role to BANNED and logs the reason). It calls set\_user\_role and add\_ban\_details.

* delete\_user($reason, $user\_id)

– Deletes a user and their profile forever (changes role to DELETED and logs the reason).

* add\_ban\_details($reason, $duration, $user\_id)

– Logs details of a ban. Inserts the user id, duration and reason into the database.

* get\_ban\_details()

– Gets the details of a ban (reason, date and time) and returns an object.

* can\_message\_each\_other($user\_id1, $user\_id2)

– Checks if the two users passed into the function can message each other returns true or false.

- Anybody can message an admin and an admin can message anybody

- Two users must mutually like each other otherwise

**Session.php**

* logout($timeout = null)

– Logs the current user out, redirects the user to the main page and stores a redirect if they are logged out by a timeout.

* login($email, $password)

- Logs a user in and redirects them to their dashboard if a redirect is not set

Removes a ban if period has passed

* register($email, $password, $first\_name, $last\_name, $DOB\_day, $DOB\_month, $DOB\_year, $sex)

- Registers a user. Passwords are stored as a hash (sha256). If the email provided already exists in the DB, an error is returned.

* verify\_login()

– Verifies if the user is logged in and redirects the user if they are not. The timeout is 30 minutes

* is\_user\_logged\_in()

– Checks if the user is logged in returns true or false.

**Users.php**

* get\_user\_name($target\_user\_id)

– Gets the first and last name of the target\_user\_id returns them in an object array.

* set\_relationship($target\_user\_id, $status)

– Sets the relationship between the current user and another user such as Block, Dislike and Like. If two users like each other a notification is sent.

* get\_relationship($target\_user\_id, $user\_id = null)

– Gets the relationship status between the current user and another.

* user\_is\_blocked\_by($target\_user\_id, $user\_id = null)

- Checks if the current user (default) is blocked by another user

**validation.php**

* validate\_name($name, $field\_name)

– The name entered and is searched for errors if errors are found the appropriate error(s) are set.

* validate\_text($text, $field\_name)

– The text entered and is searched for errors if errors are found the appropriate error(s) are set.

* validate\_email($email, $field\_name)

– The email entered and is searched for errors if errors are found the appropriate error(s) are set.

* validate\_card\_number($number, $field\_name)

– The card number is entered and is searched for errors if errors are found the appropriate error(s) are set.

* validate\_card\_cvc($cvc, $field\_name)

– The cvc number is entered and is searched for errors if errors are found the appropriate error(s) are set.

* validate\_password($password, $field\_name)

– The password is entered and is searched to make sure the criterial for a password is matched. If not the appropriate message(s) are set.

* validate\_date\_of\_birth($day, $month, $year, $field\_name)

– The date of birth is entered and is searched for errors if errors are found the appropriate error(s) are set.

* validate\_card\_expiry\_date($month, $year, $field\_name)

– The card expiry date is entered and is searched for errors if errors are found the appropriate error(s) are set.

* verify\_card($cardholder\_name, $card\_number, $card\_cvc, $card\_expiry\_date)

– The cardholder name, card number, card cvc and card expiry date is given to the credit card server and is accepted of declined.

* get\_form\_field\_status($field\_name)

- Gets the status of a field based on the errors set.

* get\_form\_field\_message($field\_name)

- Gets the success/error message for a field

**Templates:**

**browse-profiles.php**

Browse template for search.php

Loads all users, allowing pagination by limiting the returned results.

Hides users that have blocked the current user.

Hides all users the current user has blocked, unless $\_GET['blocked'] is set, in which case only blocked users are shown (for unblocking)

**search-profiles.php**

Search template for search.php

Queries users based on supplied attributes, allowing pagination by limiting the returned results.

Hides users that have blocked the current user.

**suggest-profiles.php**

Suggestions template for search.php

Queries users based on attributes from the current users profile, allowing pagination.

Hides users that have blocked the current user.

**chat.php**

Displays messages in a chat box, optionally just the messages and not the chat box.

**edit-profile-intests.php**

Displays likes and dislikes, input for a new like/dislike and a remove button for each

**edit-profile-single.php**

Template for editing a profile.

Displays the saved information in an editable form.

**profile-single.php**

Template for a viewable profile.

Displays a profile and users can like, unlike, block, unblock, report and ban the user (depending on role and permissions) through AJAX

**init.php**

Initialises every page

- Defines commonly used constants

- Includes commonly required files such as connect.php, session.php, etc

**JS:**

**Modal.js**

show\_modal(content, style\_class) – When called the modal is shown with the appropriate style and content that has been past into the function. When the user clicks on the close button of outside the modal the modal is closed.

**Scroll-lock.js**

Locks the scroll to a particular element and prevents the rest of the page from scrolling. Locks the scroll on items such as notifications and chat where a scroll to bottom/top should not cause the rest of the page to move

**Validation.php**

* set\_validation(on)

**–** Sets whether validation is on or not. Validation is disabled on the login form.

* validate\_field(el, field\_value, field\_name, validation\_type)

**–** Sends an AJAX request to validate a field using the elements passed into the function.

* check\_same\_password (el, field\_name, pass1, pass2

**–** Checks whether pass words match and update\_field\_message is called if the passwords don’t match and ‘Passwords do not match’ is passed through this function for display.

* update\_field\_message(el, valid, field\_name, field\_message)

- Displays valid/invalid messages under a field.

**References to copied materials:**

Profile images:

<http://www.buddytv.com/actors-all.aspx>

Default profile image:

<https://www.thinkprocurement.com/assets/img/gallery/1.jpg>

Background image:

<https://img1.goodfon.ru/original/4272x2848/3/bd/priroda-more-zakat-lyubov-ruki.jpg>

Data sources for populating DB:

<http://listofrandomnames.com/>

<http://www.notsoboringlife.com/list-of-hobbies/>

**References to resources:**

JQuery library:

<http://jquery.com/>

FontAwesome icons:

<https://fortawesome.github.io/Font-Awesome/icons/>

StackOverflow.com, php.net, w3schools.com

**Final Database Tables**

