My application/website is one that takes in information about the weather at particular place from the user and outputs an array of clothing choices that the person can wear that day.

The first service is the login functionality. The login/create account functionality let’s any new user create an account and log himself into the website. This allows the user to access his account and save particular outfits that he/she likes to his catalogue in case he wants to view it at a future date. This functionality works by inserting new users into the database and using session global variables to make sure that the user can view all of his personal information between pages.

The second service is the outfit maker functionality. This service uses the weather API and shopstyle clothing store API to pass information to javascript files to process and php files to transfer information between the server and save information to the database. The user inputs a city and a state. AJAX then fetches the weather information from the weather underground API. This information is sent to php by Javascript using AJAX and POST. Php then refers to the database to see what clothing categories are suitable for these weather conditions. Some conditions considered are temperature highs and lows, precipitation and wind chill. PHP then outputs the suitable clothing categories. Php then uses AJAX to access shopstyle api to return specific articles of clothing. These are stored in an array called outfit and returned to the user.

The third functionality is the ability of a user to save the outfit to his catalogue and access it at a later date. The save functionality is only available to those users who are logged in. This outfit is then saved to the user’s personal profile called catalogue and the user may view these t a later date.

One of my biggest challenges in doing this was to pass information between different kinds of files, different formats and different locations. For example, passing information between javascript and php was especially difficult because php executes before html and javascript after the body is executed.

Also, when php is embedded in javascript, the file needs to be of the extension .php. However, this means that event handlers did not recognize events being triggered because the javascript was not included as a script on the page but as a php file. Since php is server side, it does not handle client side events. The solution was to make different script files for event handlers, pass the information to a script on the html page and then pass the information into php so that it could make database calls, the return the data back to a js file.

Another problem I faced was that session variables weren’t beng saved between pages. This was because of a combination of using foundation forms and javascript to pass information. In javascript, session variables do not exist. I had to change the workflow of how the files connected to each other as a work around.

Login credentials:

Username: user

Password: password