*Cole Siegel – 3 Features*

1) Overall layout design

**Description**

* + This feature encompasses the overall layout design for the website which includes the header.php, footer.php, index.php and home.php page. I also set up the Bootstrap and jQuery references for the project.
  + Header contains a navigation bar which uses JavaScript to scroll to the appropriate section when a link is clicked. The current page is also dynamically indicated on the navigation bar as the user scrolls down through the page. The navbar is responsive (layout adjusts accordingly for mobile devices).
  + Home.php front page uses a large carousel to display marketing images, as well as a few blurbs with links to get the users’ attention.
  + Basic footer contains copyright info; more links may be added in the future – possibly for social media.
  + As new pages are added I will need to make changes to the navigation bar and re-evaluate the overall UX design periodically. The logic may need to change if we incorporate a framework such as Laravel.

**Programming Logic**

* + Index.php simply contains references (include and require) to the other .php pages, as well as the necessary JavaScript, CSS, jQuery, and Bootstrap files.
  + This design is easily maintainable and extensible as new sections have their own separate .php file which can included in the index.php file as needed.
  + Navigation bar is in header.php. Using the data-spy attribute and JavaScript, when the user clicks a link it will scroll to the appropriate section and activate that link to indicate the user has reached that area of the website. Navbar-fixed-top class ensures the navigation is always visible no matter how far you scroll down the page.
  + User profile/login links use “data-toggle=modal” as opposed to being included in the main website (added by another group member).
  + This design is very mobile-friendly as the user can scroll up and down through most of the website content without needing to click links/load new pages. Using Bootstrap navbar classes, the navigation bar is adjusts accordingly for mobile devices.

2) Contact Us

**Description**

* + Contact.php page includes a form to submit enquiries to the website staff.
  + Validation is performed using JavaScript and PHP.
  + Visual cues use Bootstrap classes to indicate invalid/valid data entry.
  + On submission, the form contents are e-mailed to the website administrator using PHP.
  + Address information and Google Maps panel is also included in this section.

**Programming Logic**

* + User input/view for the form is in contact.php file.
  + Form processing is kept in separate file, formProcessing.php, which also utilizes our validation-library.php functions.
  + Required fields are marked red initially using Bootstrap classes. Once the user exits a field, it is re-evaluated. If the input is correct, the input tags are changed to the Bootstrap valid class using jQuery (green indicator).
  + If user selects “Other” from dropdown menu, then JavaScript enables the “If other, specify” field.
  + After the form is submitted, it is validated using JavaScript. If invalid, it will stop the form submission and display an error message.
  + If it passes JavaScript validation, the form contents will be posted and validated using PHP. If invalid, an error message displays and the form contents are retained. If valid, the form contents are e-mailed to the administrator (must be deployed on web server for this feature to work).
  + Possible future additions – make the name/e-mail address field fill in automatically if the user is logged in.
  + Also uses Google Maps API to display a map to the Perfect Shades’ office (Humber College). This map is held in a flexible container class to ensure it is responsive for mobile devices.

3) View price information (web scraping/APIs)

**Description**

* + This feature has not yet been implemented
  + After the user selects a pair of sunglasses, they will have the option to view prices
  + Using web scraping and APIs, a few different online marketplace websites will be searched to find the price of the sunglasses.
  + Different order page links / prices will be presented to the user.

**Programming Logic**

* + User selects a pair of sunglasses and clicks view prices (or possibly search merchants).
  + Using PHP, access AWS (Amazon Web Services) API to retrieve the price information in XML format for the selected item. Our database will store the ASIN (Amazon Standard Identification Number) or URL of the selected item which will be passed as an identifier to the AWS API.
  + Using PHP, access eBay’s Finding API to get the listing with the lowest price for the user’s selected sunglasses. The information for this listing will be returned by the API in XML format.
  + Possibly access other websites/online marketplaces to find more prices for the chosen sunglasses.
  + Convert the XML data into PHP objects, sort the objects by lowest price first, and display the list to the user. Links will be provided to access the marketplace listing and place an order.

*Michael Corrales - 3 features*

1. Resources

**Description**  
- a basic addition to the nav bar to cite resources for the images of the products used in the website and database

**Programming logic**  
-basic use of drop down using animations from bootstrap and href tags to provide links to the owners of the images

2. Recommend to a friend

**Description**  
-users who find the website can suggest it to a friend who are interested on finding the perfect pair of shades that they need

**Programming Logic**

-users can enter the valid email of their friends and be able to add a few short description about the website and perfect shades will send the email for the user

3. Comments and Review Section

**Description**

-currently not implemented (but form is already existing)  
-users will be able to rate, comment and share their opinions about the product

**Programming logic**

-a dedicated form handles the comments and suggestion page and requires user to input ther email to be validated

-a sub-feature that is already working is a star rating where you can already click the number of stars you want to give a product. It also has the option for the user to reset and start the rating all over again

4. Upload user photo

-currently not implemented (but form is already existing)

-this is the place where user will upload their photo and choose a category for their face shape

Programming logic

-the form offers user to upload a file of the image of their face

-form must validate the file size limit and file type

Man Luo – 3 Features

1) Admin UI Design  
**Description**

-          This feature encompasses the Admin UI design for the website which includes: admin.php, tables\_category.php, tables\_color.php, tables\_face\_pairing.php, tables\_face\_shapes.php,tables\_favourite\_product.php, tables\_glasses.php, tables\_manufacturers.php, tables\_users.php, tables\_web\_resources.php as well as editor pages includes: editor\_category.php, editor\_color.php,editor\_face\_pairing.php, editor\_face\_shapes.php,editor\_favourite\_products.php, editor\_glasses.php, editor\_manufacturers.php, editor\_users.php, editor\_web\_resources.php page.

- admin portal contains a side navigation bar which uses JavaScript to go to the appropriate section when a link is clicked. It also contains a top navigation bar which display the message as well as a menu allow administrator to check profile and log out. The admin.php page also includes a dashboard which display the Statistics Overview to the administrator, such as New User, New Comments and New Items. The navbar is responsive.

-          All pages with tables\_ prefix includes a table which display table data from database for each table respectively as well as two action button, Edit and Delete.

-          Edit button provides the function to display a modal form and allow administrator to edit the item and save the changes to database.

-           Delete button provides the function to allow administrator to delete the item from database.

-          All pages can be modified per database design.

-          All pages are populated with static data for now and will be populated with data from database dynamically in latter phase.

**Programming Logic**

-          After administrator logged in, the admin.php page will be displayed. admin.php contains a side navigation bar and top navigation bar which use php to populate dynamically. Will be populated with data from database in latter phase.

-          Side navigation bar contains a dashaboard and tables section. Dashboard indicates the statistics overview. It will display the incrementation of user, item and comments as well as an Activity Panel. The activity panel will display activities performed recently.

  -       Tables section in side navigation bar is a dropdown list which include all links to each page with tables\_ prefix. When user click the link, it will go to the specific page and all the data of this table will display along with two buttons, Edit and Delete. When click Edit button for an item, it will open a modal form to allow user to edit the data of this item and save the update to database. When click delete button it will perform deletion of this item from database.

2) Log In

**Description**

-          login.php page includes a form for user to enter email and password to login as well as a checkbox to allow user to save email and password.

**Programming Logic**

-          After user enter their email, password and click submit button then check if user checked the RememberMe checkbox. If no, check if the combination of email and password exists in database or not. If it exists then login successfully and the menu item of user profile menu will be changed. If it doesn’t exist, then there will be an error message to indicate that user may enter a wrong email/password.

-         If user checked the RememberMe checkbox when they are logging in, use cookie to save the user email and password.

-          This feature is not fully functional yet. It will be completed in latter phase.

3) Forget Your Password?

**Description**

-          forget\_password.php includes a form to allow user to submit email to reset password.

-          After the user submit the valid email, they will receive an email with a link to another page to reset the password.

**Programming Logic**

-          After user click submit button, check if it’s an email address. If so, then check if it’s existed in database or not. If so, then send an email with the reset password link to this email address.

-         If user input is not an email address, then display an error message to ask for another input.

-         If user input doesn’t exist in database, then display an error message to tell user that email address can’t be identified.

-         This feature is not fully functional yet. It will be completed in latter phase.