**CT4009 Introduction to Web Development**

**Assignment 001**

Technical Report

for the offline store catalogue

**Student Name**: Nathaniel Clarke

**Student Number**: s1906563

**Course**: Computer and Cyber Security

**Submission Date**: Monday, 13 January 2020

**University of Gloucestershire**

Contents

[Introduction 3](#_Toc29043586)

[Website Details 3](#_Toc29043587)

[List of Functional and Non-functional pages 3](#_Toc29043588)

[Testing 4](#_Toc29043589)

[User Manual 5](#_Toc29043590)

[Shortcomings/Things to improve 6](#_Toc29043591)

# Introduction

# Website Details

Username: s1906563

Password: HJ55xH%nMbDv#t9?

Domain name: s1906563-ct4009.uogs.co.uk

Plesk URL for Web Hosting Management: https://uogs.co.uk:8443/login\_up.php

# List of Functional and Non-functional pages

\*JavaScript/IndexedDB only needed for login/nav bar

\*\*functional but nothing on the page except nav bar

|  |  |  |
| --- | --- | --- |
| Page | HTML/CSS | JavaScript/IndexedDB |
| Home – carousel/login\* | ✓ | ✓🗶 |
| Store locator - map | ✓ | ✓ |
| Contact – email/map link\* | ✓ | ✓🗶 |
| Shop - view cat/subcat/items/compare/add to watchlist | ✓ | ✓ |
| Hot Deals\*\* | ✓ | ✓🗶 |
| Watchlist - view | ✓ | ✓ |
| Search Results\*\* | ✓ | ✓🗶 |
| Admin Controls – add cat/add subcat/add item/add location   * Delete & link to update cat/subcat/item/location | ✓ | ✓ |
| Update Map | ✓ | ✓ |
| Update Category | ✓ | ✓ |
| Update Subcategory | ✓ | ✓ |
| Update Item | ✓ | ✓ |

# Testing

All login, IndexedDB tests use /js/lib/indexeddb.js, /js/main.js

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ref.** | **Page Area** | **Scope** | **Expected Results** | **File ref** | **Actual Results** | **Fixes** |
| 001 | Home Login | Login function   1. Login clicked 2. Invalid credentials entered | Login field appears, Access denied, user alerted  Code: respond to incorrect input | /home/HomePage.html  /js/main.js | As expected | N/A |
| 002 | Home Login | Login function admin   1. Correct credentials entered | User redirected to admin page  Code: match valid credentials, redirect to target URL | /home/HomePage.html  /js/main.js | As expected | N/A |
| 003 | Admin Controls Add item, subcategory | Empty selection input   1. User adds item without selecting subcategory or db has no values to display 2. User adds subcategory without selecting category or db has no values to display | Form not submitted; user alerted  Code: handle empty input, prevent form submitting | /login/crud.html  /js/AddFunc/AddItem.js  /js/AddFunc/AddItemDAO.js  /js/AddSubcat/AddSubcategory.js | As expected | N/A |
| 04 | Add Category | New Category with name and description   1. User completes category form | Category added to db, user alerted  Code: enter input into category db store | /login/crud.html  /js/lib/indexeddb.js  /js/AddCat/AddCategory.js | As expected | N/A |
| 05 | Add Subcategory | Subcategory with parent category   1. Select category from selection and submit | Subcategory added to dB, user alerted  Code: enter input into subcategory dB store with according category id | /login/crud.html  /js/lib/indexeddb.js  /js/AddSubcat/AddSubcategory.js | As expected | N/A |
| 006 | Add item | Item with image, subcategory parent id   1. Enter values, select subcategory 2. Upload image | Item added to dB  Code: enter input into item dB store with subcategory id and image file | /login/crud.html  /js/lib/indexeddb.js  /js/AddFunc/AddItem.js  /js/AddFunc/AddItemDAO.js | As expected | N/A |
| 007 | Add location | Add location to db   1. Enter store name 2. Drag marker to desired location and submit | Latitude/longitude of marker saved to dB, user alerted  Code: save data to events store, alert user, handle empty inputs | /login/crud.html  /js/lib/indexeddb.js  /map/InsertEvent.js  /map/InsertEventDAO.js | Location added in centre of map | EventList.js was causing issues being linked in the same page, location list now doesn’t have map with markers |
| 008 | Update/delete locations | List of locations viewable with update/delete buttons | Location list from events dB store viewable  Code: get all values from events store, display with links to update/delete | /login/crud.html  /js/lib/indexeddb.js  /map/listLocations.js | Item values are “undefined” | Function had to be separated from InsertEvent.js file, js link in html needed to be within accordion div |
| 009 | Update/delete locations | Deleting entry from dB   1. Click delete button of desired location | Location deleted from db, user alerted  Code: remove entry with matching id from events store, alert user | /login/crud.html  /js/lib/indexeddb.js  /map/listLocations.js | Data not deleted | Delete function needed to be within function to build html div populated by db |
| 010 | Update/delete categories with subcategories & items | Viewing items indented in subcategories indented in categories   1. Click update categories button | Objects added to db viewable, items under correct subcategories, under correct categories  Code: build html with matching id’s and informating showing | /login/crud.html  /js/lib/indexeddb.js  /js/ListCat/listCat.js | As expected | N/A |
| 011 | Update/delete categories with subcategories & items | Update category, subcategory and item successfully   1. Click update button underneath desired category, subcategory or item | User taken to update page for category, subcategory or item according to which clicked. Current object info filled out, able to edit and submit  Code: Redirect to target URL, obtain correct db entry. Change id values in store | /login/crud.html  /js/lib/indexeddb.js  /js/ListCat/listCat.js  /js/UpdateFunc/updateCat.js  /js/UpdateFunc/updateCatDAO.js  /js/UpdateFunc/updateSubcat.js  /js/UpdateFunc/updateSubcatDAO.js  /js/UpdateFunc/UpdateItem.js  /js/UpdateFunc/UpdateItemDAO.js | 1. Buttons don’t do anything 2. Id values ‘null’ when #1 fixed | 1. Function calls had to be within html build function 2. Needed to call function to get values of objects 3. in array and parse id from array as int |
| 012 | Update/delete Locations | Update store locations successfully   1. Click update button of specific location | User taken to map update page, current info filled out, able to edit and submit  Code: Redirect to target URL, obtain correct db entry. Change id values in store | /login/crud.html  /map/listLocations.js  /js/UpdateFunc/updateMap.js  /js/UpdateFunc/updateMapDAO.js | As expected | N/A |
| 013 | Store Locator | View Store locations | User can see map with markers and titles for each store  Code: retrieve and display all marker information from db | /map/EventList/EventList.html  /map/EventList/EventList.js  /map/EventList/EventListDAO.js | Location in centre of map | Fixed in test #007 |
| 014 | Shop | Navigate to specific category and view subcategories and items   1. Click link to category in nav dropdown | User taken to clicked category on shop page, subcategories and items displayed  Code: redirect to specific element on page, retrieve objects from db and populate html | /CatTemplate/catTemplate.html  /js/ListCat/listCat.js  /js/ListCat/initCompare.js | Images not displaying | Image folder path had to be in image source – images must be uploaded from that folder |
| 015 | Shop | Add 2 items to compare   1. Click compare checkbox on two items | Both items displayed with remove button, without compare or watchlist checkboxes  Code: get correct id and display in compare box without checkboxes, allow no more than two items | /CatTemplate/catTemplate.html  /js/ListCat/listCat.js  /js/ListCat/initCompare.js | As expected | N/A |
| 016 | Shop | Remove items from compare   1. Click remove under compare item | Item deleted from compare,  Code: remove item from div, hide compare div if empty | /CatTemplate/catTemplate.html  /js/ListCat/listCat.js  /js/ListCat/initCompare.js | As expected | Added code to uncheck checkbox when removed |
| 017 | Shop | Add item to watchlist   1. Click add to watchlist checkbox on item | Item added to watchlist db store, user alerted  Code: get item info and add to watchlist store | /CatTemplate/catTemplate.html  /js/ListCat/listCat.js | ID and item info not being found and added | Needed to split id to parse as int |
| 018 | Any | Login to public user page   1. Enter valid credentials and submit | User taken to account/watchlist page, watchlist items viewable  Code: match valid credentials, redirect to target URL | /js/main.js  /WatchList/watch.html  /js/ListFunc/ListItems.js  /js/ListFunc/ListItemsDAO.js | Items not being found | Incorrect object store being named |
| 019 | Watchlist | Delete item from watchlist   1. Click remove button on item | Item removed, user alerted  Code: delete item from db, alert user and refresh page | /WatchList/watch.html  /js/ListFunc/ListItems.js  /js/ListFunc/ListItemsDAO.js | As expected | N/A |
| 020 | Mobile Site | Test site loads on mobile   1. Load site on mobile device | Site should format properly, nav condensed to burger, site functions same as desktop view  Code: distinguish type of device user is using and format | /home/HomePage.html | As expected | N/A |

# User Manual

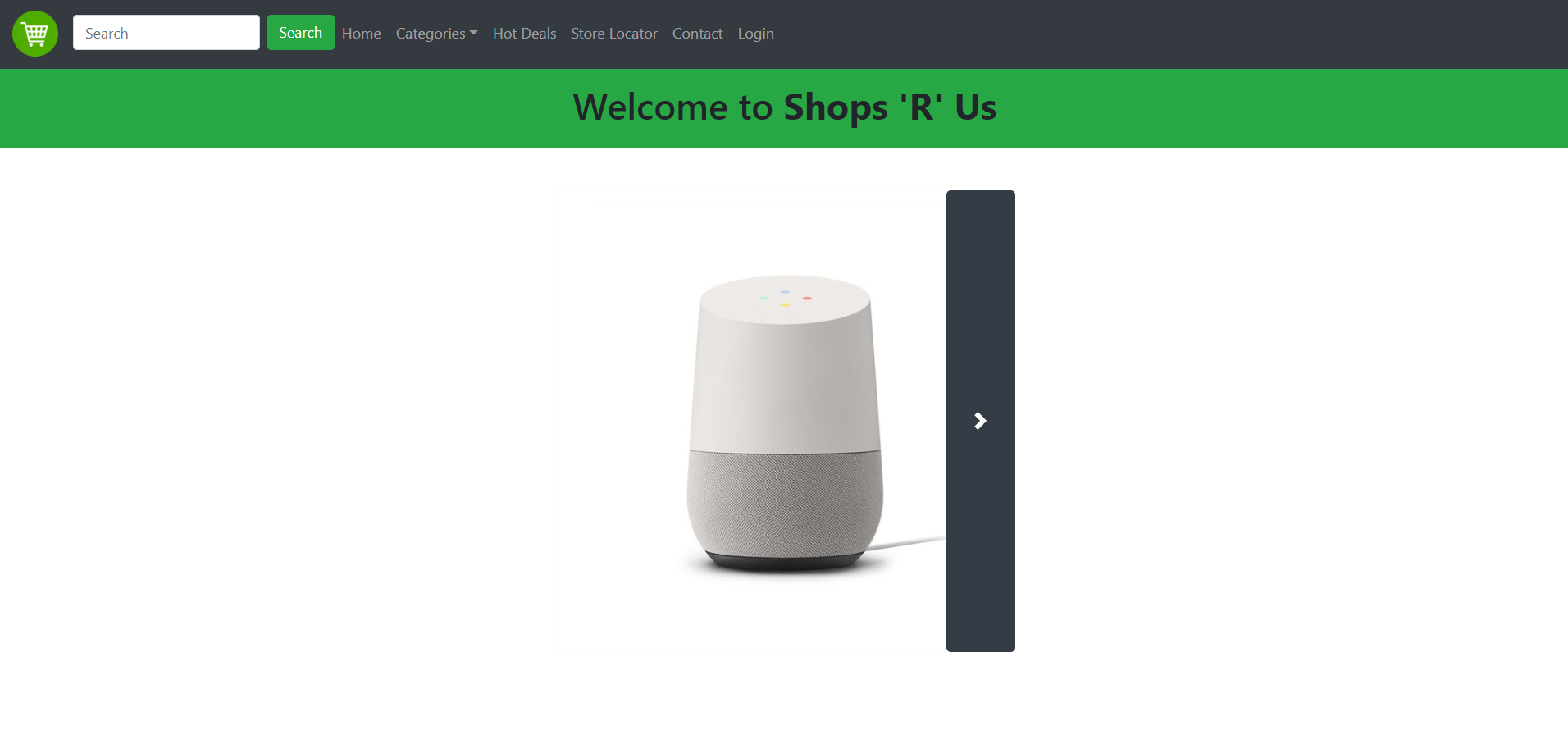
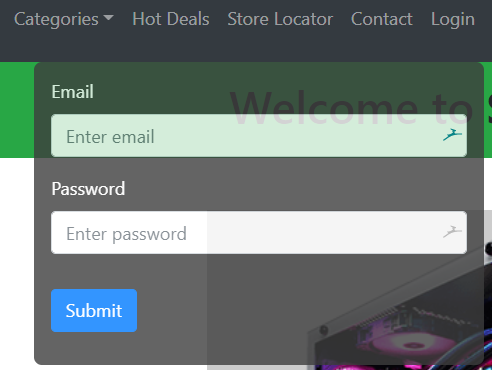
In this manual you will find a detailed explanation of every functionality of the site.

Admin controls: p12 – 17

Contact/Store Locator: p18

Shop/Compare: p19

Watchlist: p20



Click on the login button and this box will appear

The admin login will take a user to the Admin Controls page.

Email: admin@test.com

Password: password123

The public login will take a user to their account or watchlist.

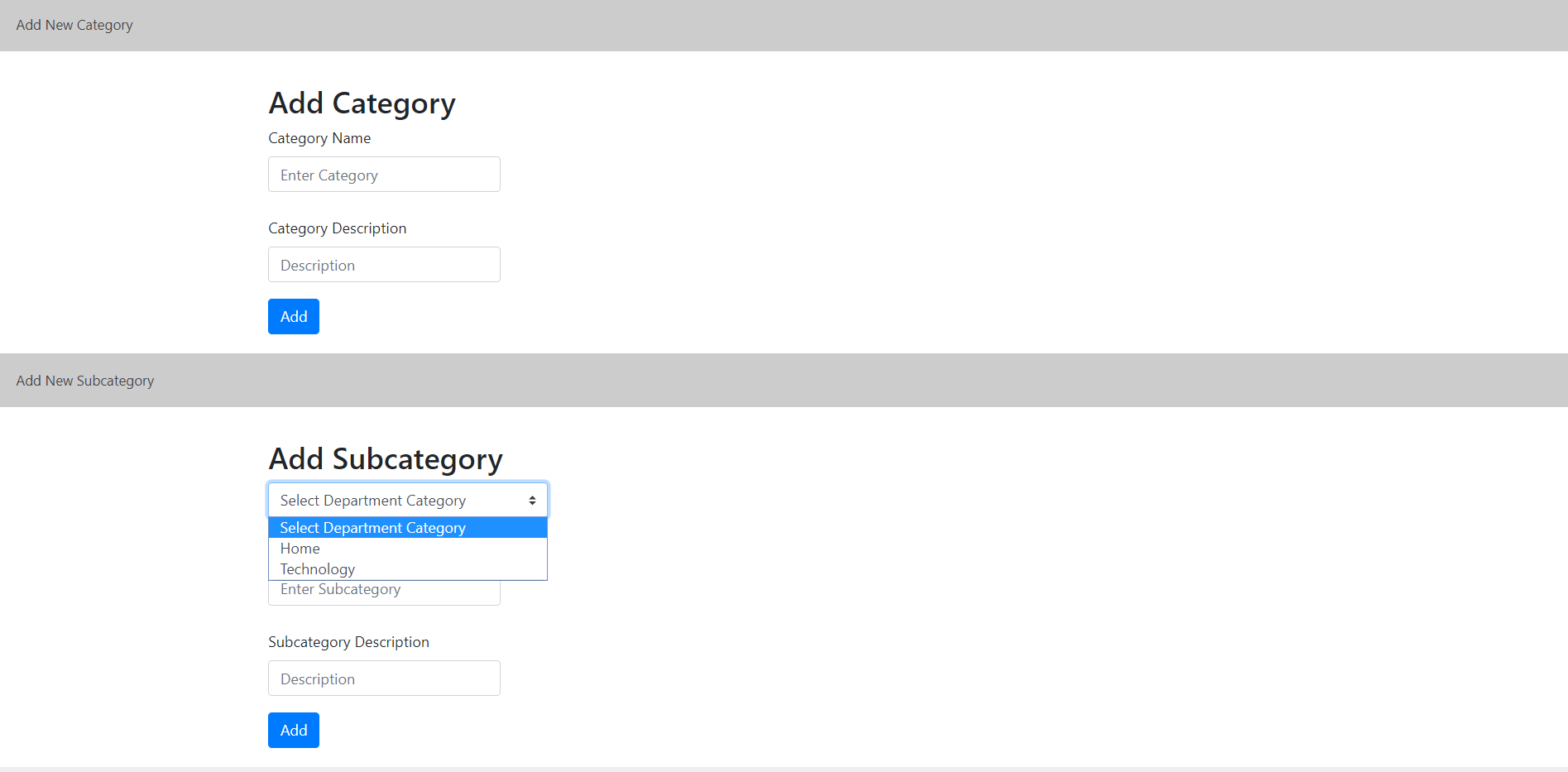
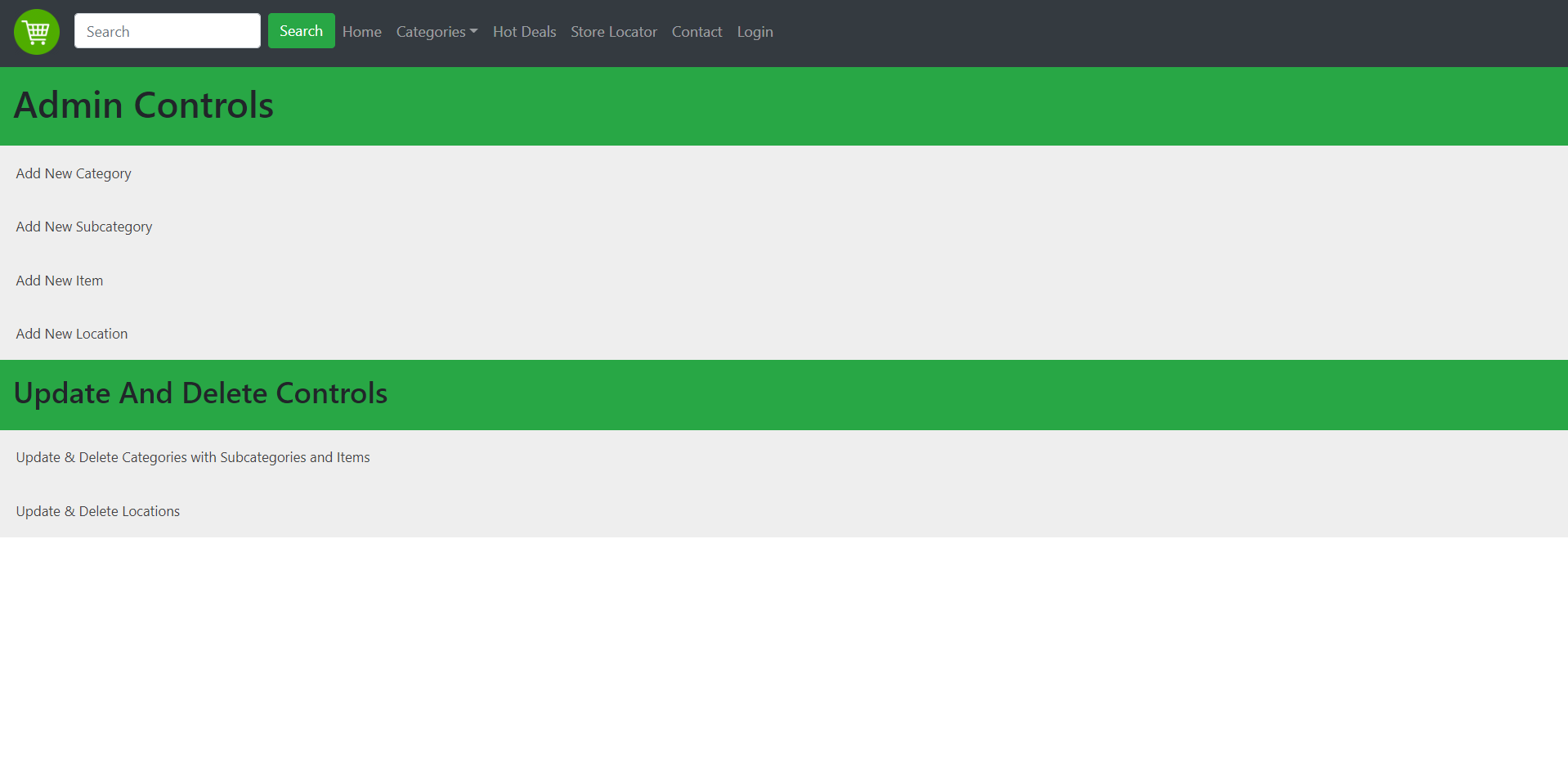
Email: public@test.com

Password: 321password

This is the home page a user will first see.

At top is the navigation bar with the links to the shop page, hot deals (non-functional), store locator, contact and login. There is also a search bar, but this functionality has not been implemented and it will take the user to a blank page.

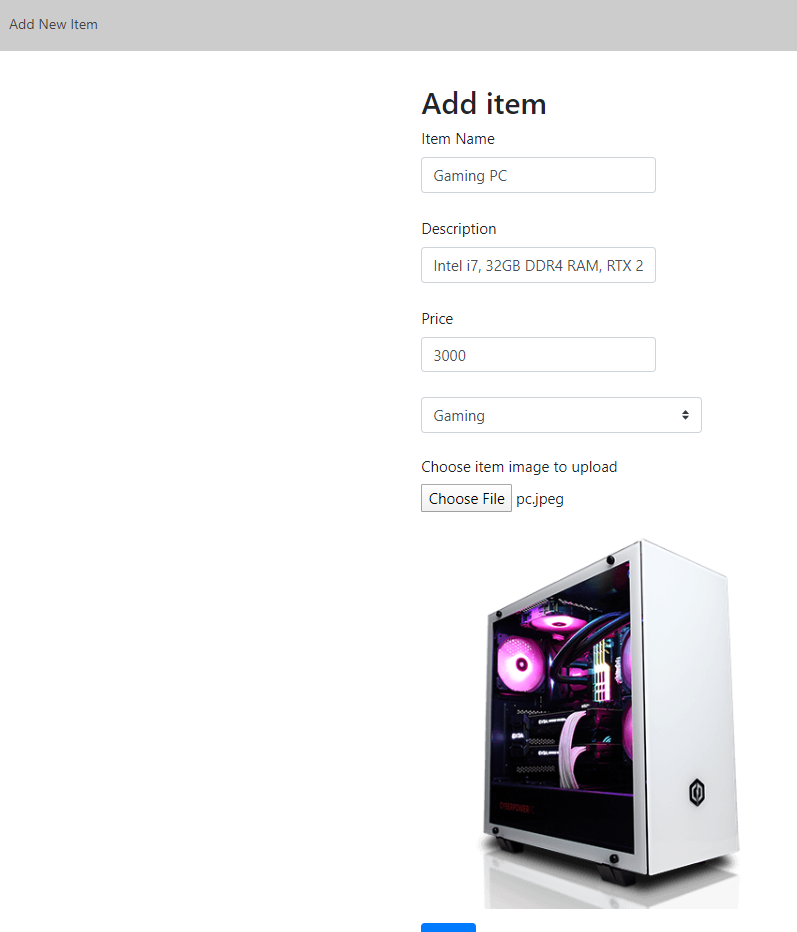
This carousel currently cycles through 3 images however it’s only fed from the file path, not from the database.



This is the page where administrators can add a new category, subcategory and item that all relate to each other. These can all be updated and deleted too. Store locations can also be added which public users can view in the store locator page. These too can be updated and deleted.

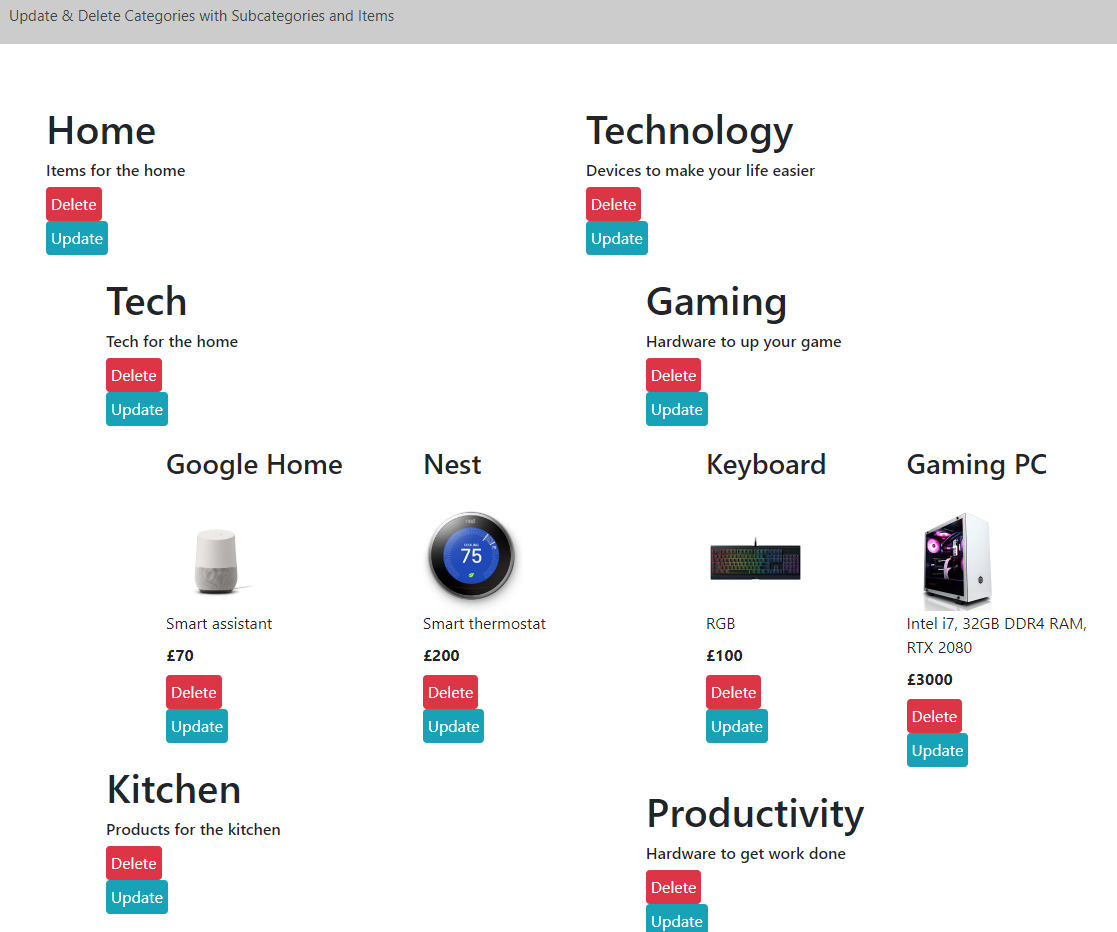
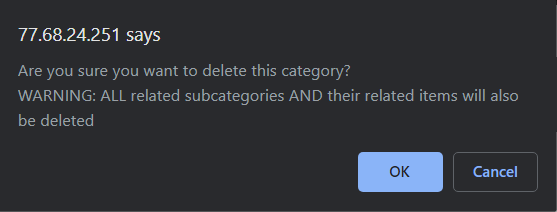
To access a function, click the one you want, and it will expand. This is to keep the site looking clean and tidy.

A category must be added before a subcategory, and a subcategory before an item. This is because the dropdown requires an input that comes from the function preceding it. If a user skips a field and submits, they will be alerted that all fields are required.

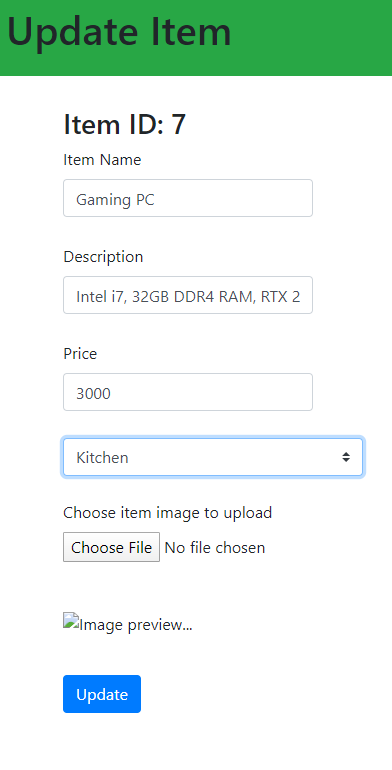


Adding an item is the same as a subcategory except a user must upload an image this time and select a subcategory instead of category

Once an item has been submitted, it’s displayed in the shop page (p19) and the update/delete admin control (p15), with its corresponding subcategory and parent category.

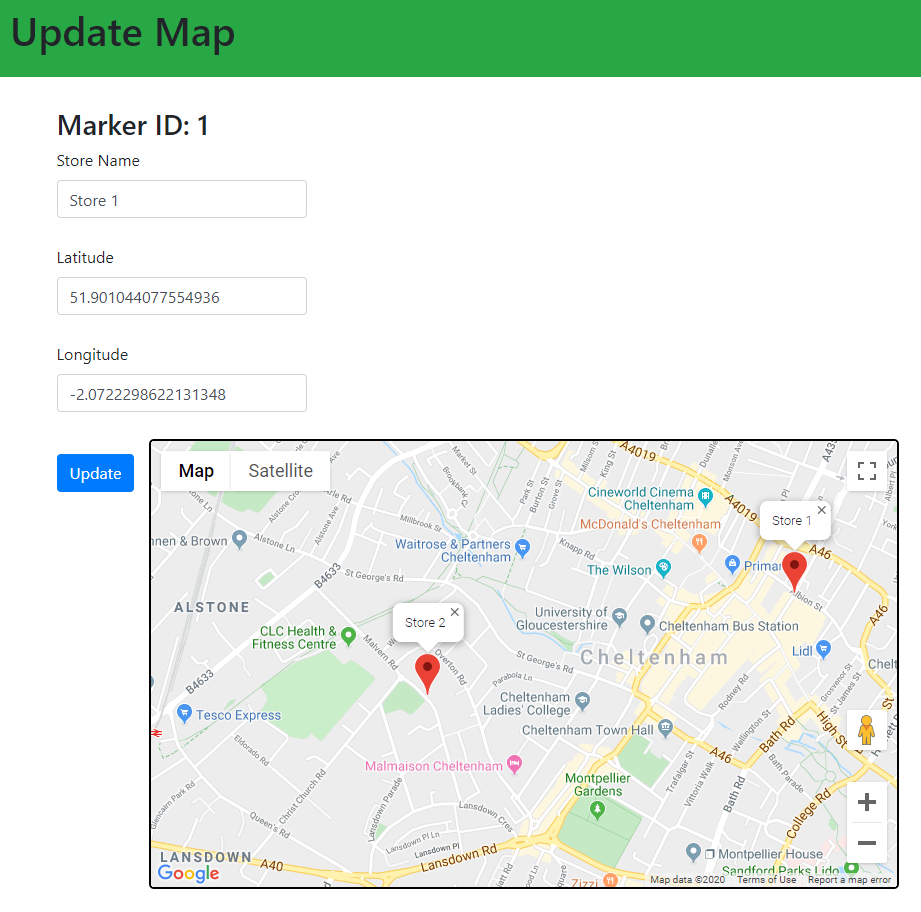
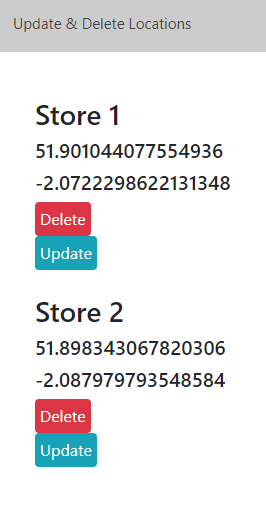
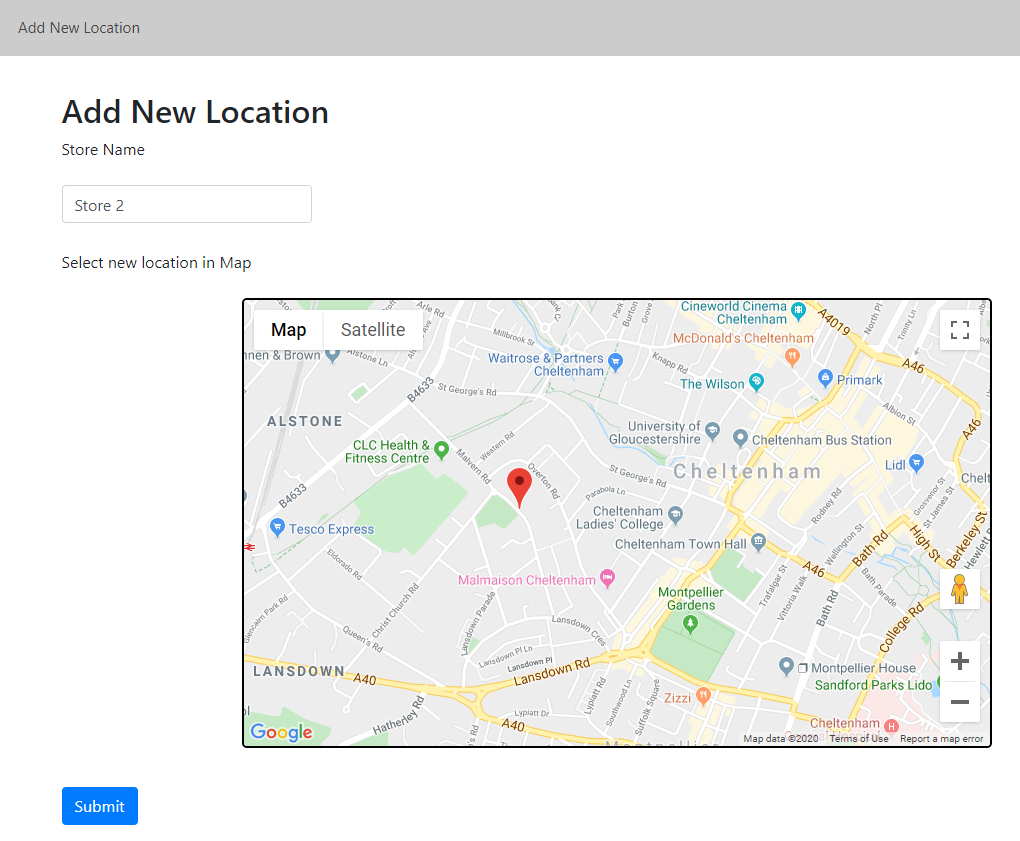


Admins can view items indented under subcategories, under categories, all with buttons to delete and update



Deleting a category will display this message and will delete everything underneath it

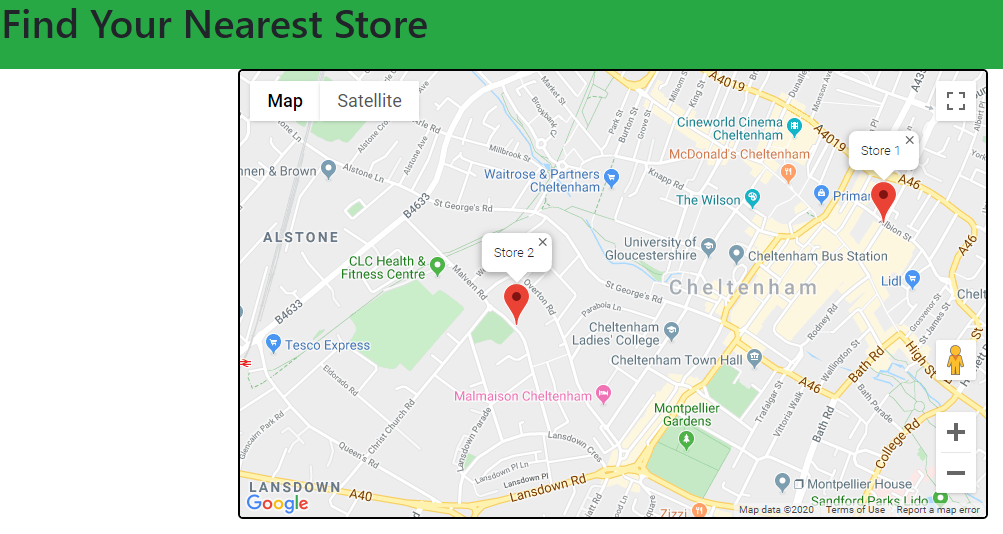
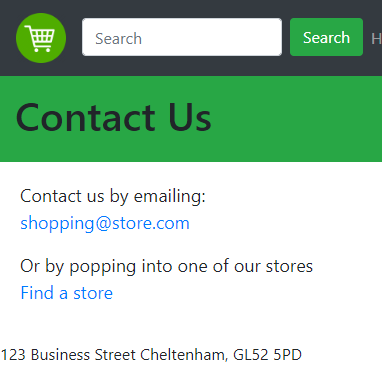
Update buttons takes the user to the update page for that type of element, with all the fields filled out and ready to edit



Users can add a location by dragging the marker, entering a name and submitting

Unfortunately, the users can’t drag the marker to update a location, they must manually change the coordinates

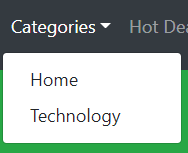
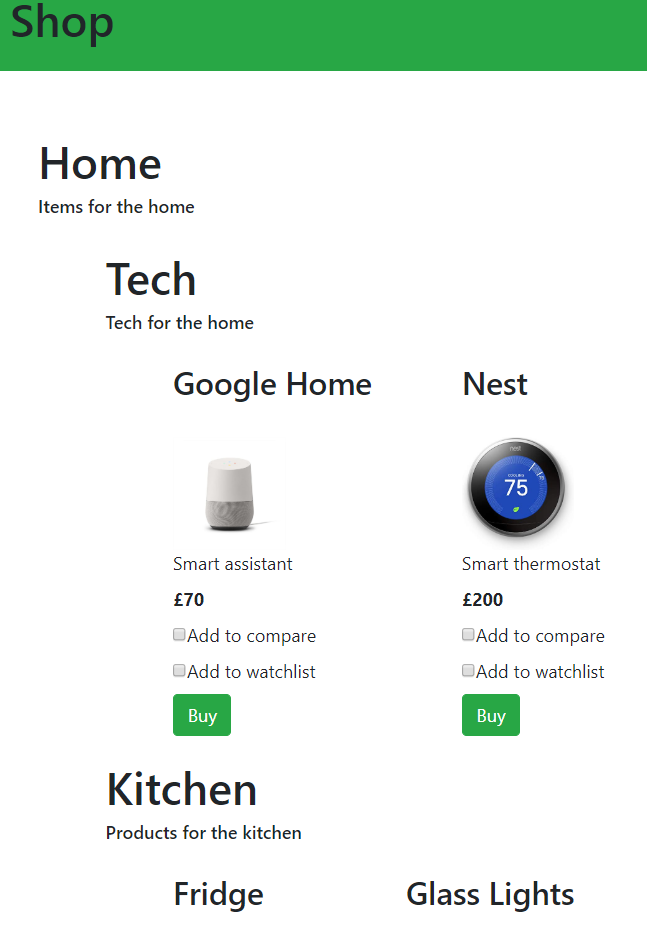
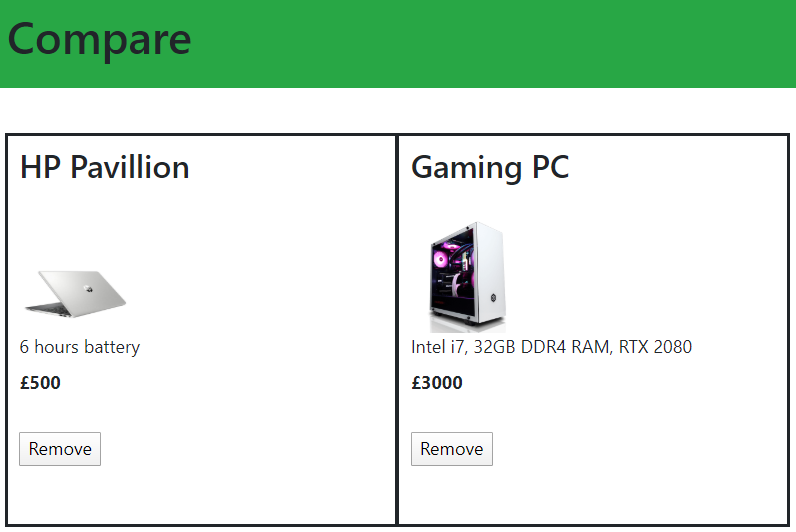
Locations are displayed in the update/delete admin control where they can be deleted and updated



The Locations added in the admin page appear here

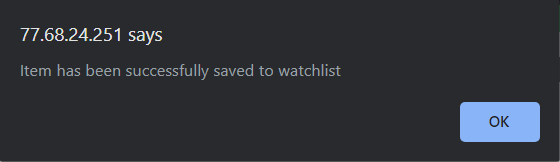
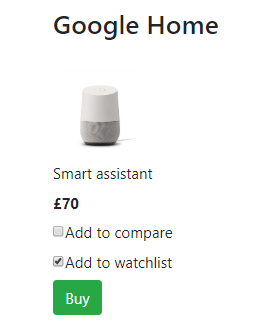
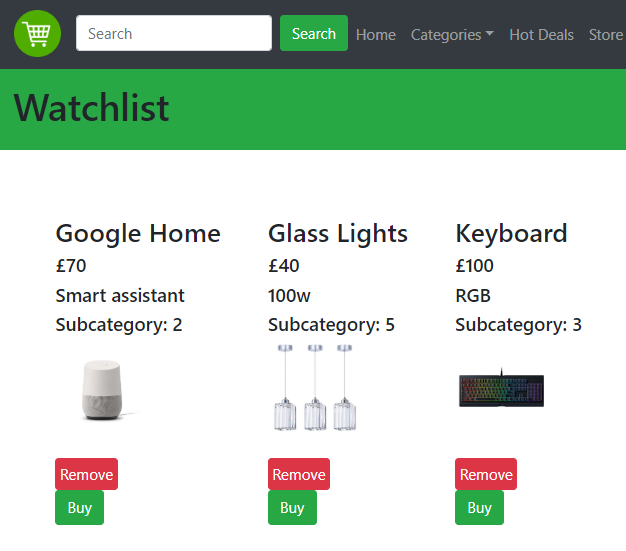
Opens link to user’s email client with the subject: Enquiry

Link to store locator



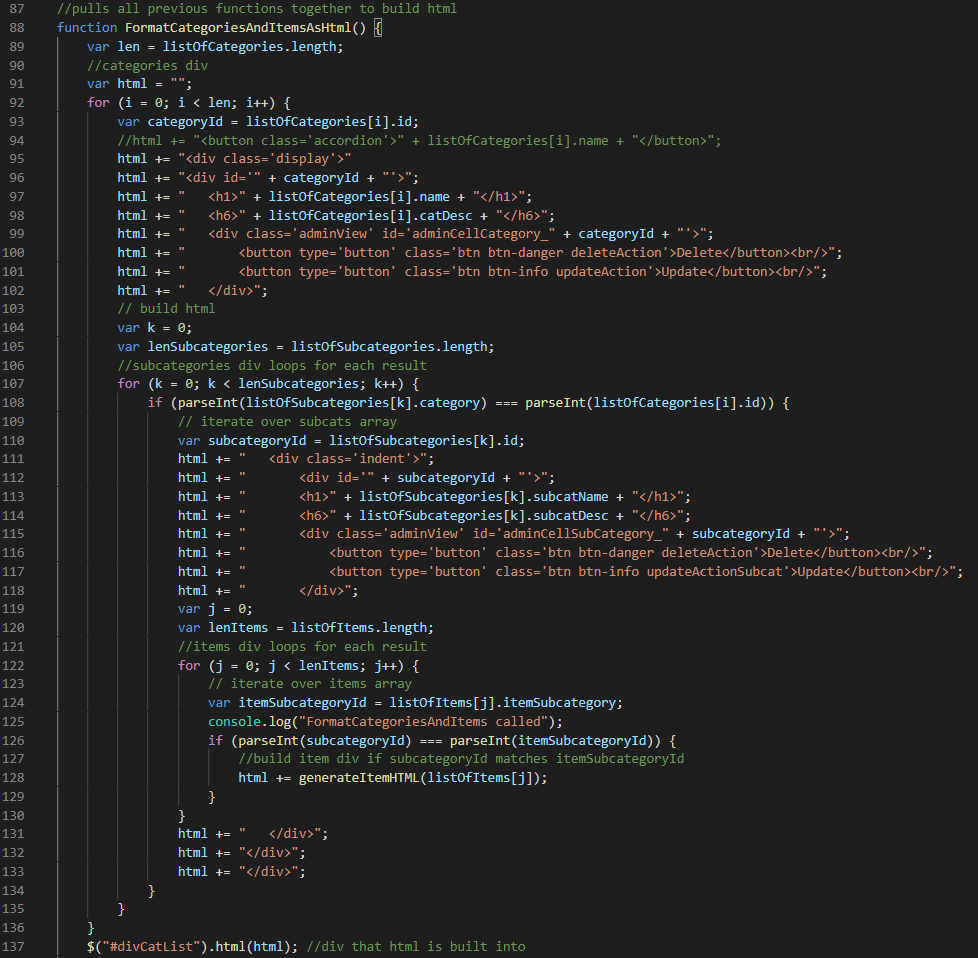
Items have options to add to compare and watchlist and to buy (button takes user to “under construction” page). Adding to compare will display the item in the compare section that appears beneath the shop. A maximum 2 items can be compared. Removing an item will uncheck its checkbox.

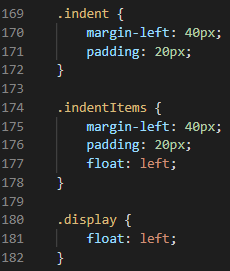
Categories added will appear in the navigation dropdown, user is taken to where that category is on the shop page



Adding an item to watchlist will alert the user and save it to the database. To access the watchlist a user must use the public login (p11). From there they can buy or remove an item. Buying will redirect to the “under construction” page. Removing an item will remove it from the database, but not the shop.

# Interesting Use of HTML/CSS/JavaScript



The interesting use of HTML/CSS/JavaScript in this site all comes from the ‘listCat.js’ JavaScript file. This is the code responsible for assembling the items within the subcategories within both the categories on the shop page and admin page. It decides which buttons or information to display depending on the page URL; the admin page will hide the checkboxes and buy button and the shop page show them and hide the admin update/delete. This code is interesting because it creates divs using JavaScript meaning everything is in one file and you can use your variables directly within the HTML.

The values from the DB are all put into an array first because it works better for obtaining the ID when implementing it in the HTML. Earlier in the code, there is a function to delete objects that cascades to delete all child objects with corresponding ID’s. It does this by call the next function to delete which triggers the one after then and then deletes the ID for its own function.

This makes the page dynamic as everything is built using values from the DB so it’ll only ever display what’s there.

# Shortcomings

While this site is functional, some things could be improved.

1. Images have to use full path to locate them on client-side website. An upload function could be added to add files to a main server to be all in one place.
2. The site uses lots of files making it slower for browser to load. A way to negate this would be to use a single page application where all the HTML is in a main file and uses JavaScript to hide and show elements, giving the illusion of a multi-paged site whilst increasing load times
3. A problem with client-side is you can’t guarantee you have permission to run IndexedDB or JavaScript so it might not work altogether. SQL or other databases are more streamlined and easier to use than IndexedDB
4. Logins can’t be easily encrypted on client-side. You would normally be comparing hash values rather than storing values in plaintext like this site. Login details can currently be accessed through developer tools looking at the DB
5. Hot deals page does nothing – in future the site could add a checkbox in the add item form to add to the hot deals page
6. Carousel isn’t linked to IndexedDB, images come from their file path – in future it would be linked to IndexedDB to show items with some information
7. There are currently no cookies on the site, so the logins only serve to redirect to a specific page. Future implementations would have cookies for users to properly log in and out

# References

**Advice**

UKHO and Clarke, J. (2019) ‘Consultations’.

**IndexedDB wrapper**

CT4009, 2019. *IndexedDB Wrapper API.*[Online]  
Available at: moodle.glos.ac.uk

[Accessed 2019 November 18].

**Images**

*Customize Gamer Dragon Gaming PC* (no date). Available at: https://www.cyberpowerpc.com/system/Gamer-Dragon (Accessed: 6 January 2020).

*Refrigerators in Calgary | Atlas Appliances* (no date). Available at: https://www.atlasappliances.com/en/catalog/product/247656-Samsung-RF26J7500SR (Accessed: 6 January 2020).

*Google Home Speaker | Walmart Canada* (no date). Available at: https://www.walmart.ca/en/ip/google-home-speaker/6000197114649 (Accessed: 6 January 2020).

*Google, T3007ES, Nest Learning Thermostat, 3rd Gen, Smart Thermostat, Stainless Steel, Works With Alexa - - Amazon.com* (no date). Available at: https://www.amazon.com/Nest-T3007ES-Thermostat-Temperature-Generation/dp/B0131RG6VK (Accessed: 6 January 2020).

*Amazon.com: Razer Cynosa Chroma Gaming Keyboard: Customizable Chroma RGB Lighting - Individually Backlit Keys - Spill-Resistant Design - Programmable Macro Functionality: Computers & Accessories* (no date). Available at: https://www.amazon.com/Razer-Cynosa-Chroma-Gaming-Keyboard/dp/B075KMZ4MX?ref\_=fsclp\_pl\_dp\_2 (Accessed: 6 January 2020).

*HP 17.3" Intel Pentium 8GB RAM, 1TB HDD Touchscreen Laptop Bundle - 9187738 | HSN* (no date). Available at: https://www.hsn.com/products/hp-173-intel-pentium-8gb-ram-1tb-hdd-touchscreen-laptop/9187738 (Accessed: 6 January 2020).

*SHENGQINGTOP Modern Crystal Pendant Light for Kitchen Island, Bedroom, Living Room (Cuboid Pendant Light) - - Amazon.com* (no date). Available at: https://www.amazon.com/ShengQing-Crystal-Pendant-1-Light-Lighting/dp/B07CNJL9ZP?th=1 (Accessed: 6 January 2020).

*Buy, cart, circle, ecommerce, green, shopping, trolley icon* (no date). Available at: https://www.iconfinder.com/icons/536262/buy\_cart\_circle\_ecommerce\_green\_shopping\_trolley\_icon (Accessed: 8 January 2020).

*Searching for array elements in IndexedDB* (no date). Available at: https://www.raymondcamden.com/2012/08/10/Searching-for-array-elements-in-IndexedDB (Accessed: 8 January 2020).

*Bootstrap Buttons* (no date). Available at: https://www.w3schools.com/bootstrap/bootstrap\_buttons.asp (Accessed: 8 January 2020).*Navbar · Bootstrap* (no date). Available at: https://getbootstrap.com/docs/4.0/components/navbar/ (Accessed: 8 January 2020).

*HTML DOM Input Checkbox Object* (no date). Available at: https://www.w3schools.com/jsref/dom\_obj\_checkbox.asp (Accessed: 8 January 2020).*Tryit Editor v3.6* (no date). Available at: https://www.w3schools.com/tags/tryit.asp?filename=tryhtml\_link\_mailto (Accessed: 8 January 2020).

**Accordion**

*How To Create an Accordion* (no date). Available at: https://www.w3schools.com/howto/howto\_js\_accordion.asp (Accessed: 8 January 2020).

**IndexedDB**

*IndexedDB fundamentals - Plus a indexeddb example - tutorial* (no date). Available at: https://www.vanamco.com/2014/11/14/indexeddb-fundamentals-plus-a-indexeddb-example-tutorial/ (Accessed: 8 January 2020).

*Build a basic web app with IndexedDB - DEV Community 👩‍💻👨‍💻* (no date). Available at: https://dev.to/andyhaskell/build-a-basic-web-app-with-indexeddb-38ef (Accessed: 8 January 2020).

*Using IndexedDB - Web APIs | MDN* (no date). Available at: https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB\_API/Using\_IndexedDB (Accessed: 8 January 2020).

**Carousel**

*Tryit Editor v3.6* (no date). Available at: https://www.w3schools.com/bootstrap4/tryit.asp?filename=trybs\_ref\_js\_carousel2 (Accessed: 8 January 2020).

**Navbar**

*Tryit Editor v3.6* (no date). Available at: https://www.w3schools.com/bootstrap4/tryit.asp?filename=trybs\_navbar\_form&stacked=h (Accessed: 8 January 2020).

*Bootstrap 4 Navigation Bar* (no date). Available at: https://www.w3schools.com/bootstrap4/bootstrap\_navbar.asp (Accessed: 8 January 2020).

*Tryit Editor v3.6* (no date). Available at: https://www.w3schools.com/bootstrap4/tryit.asp?filename=trybs\_navbar\_dropdown&stacked=h (Accessed: 8 January 2020).

**Login Form**

*Tryit Editor v3.6* (no date). Available at: https://www.w3schools.com/bootstrap4/tryit.asp?filename=trybs\_form\_basic&stacked=h (Accessed: 8 January 2020).

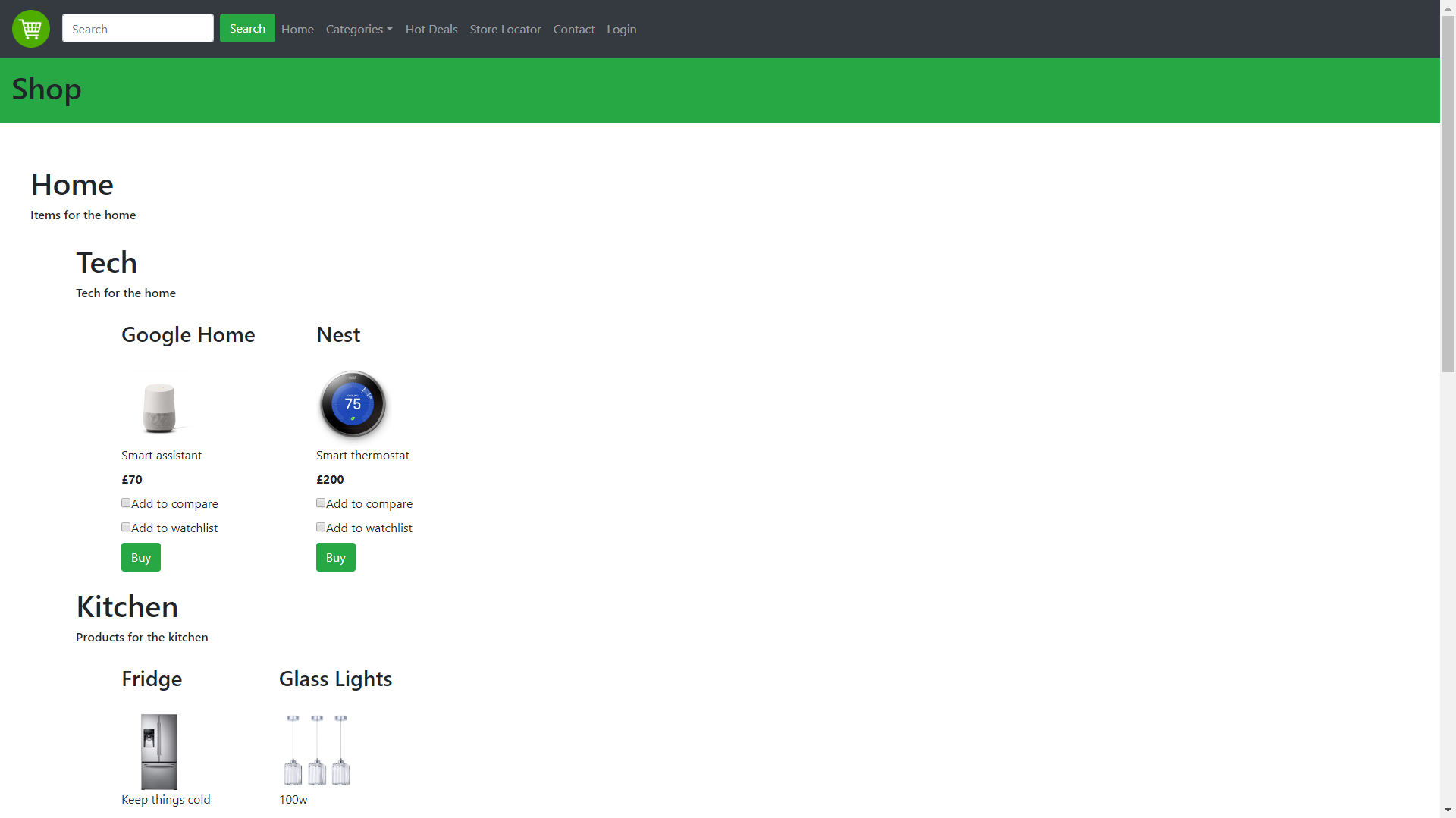
**Checkbox**

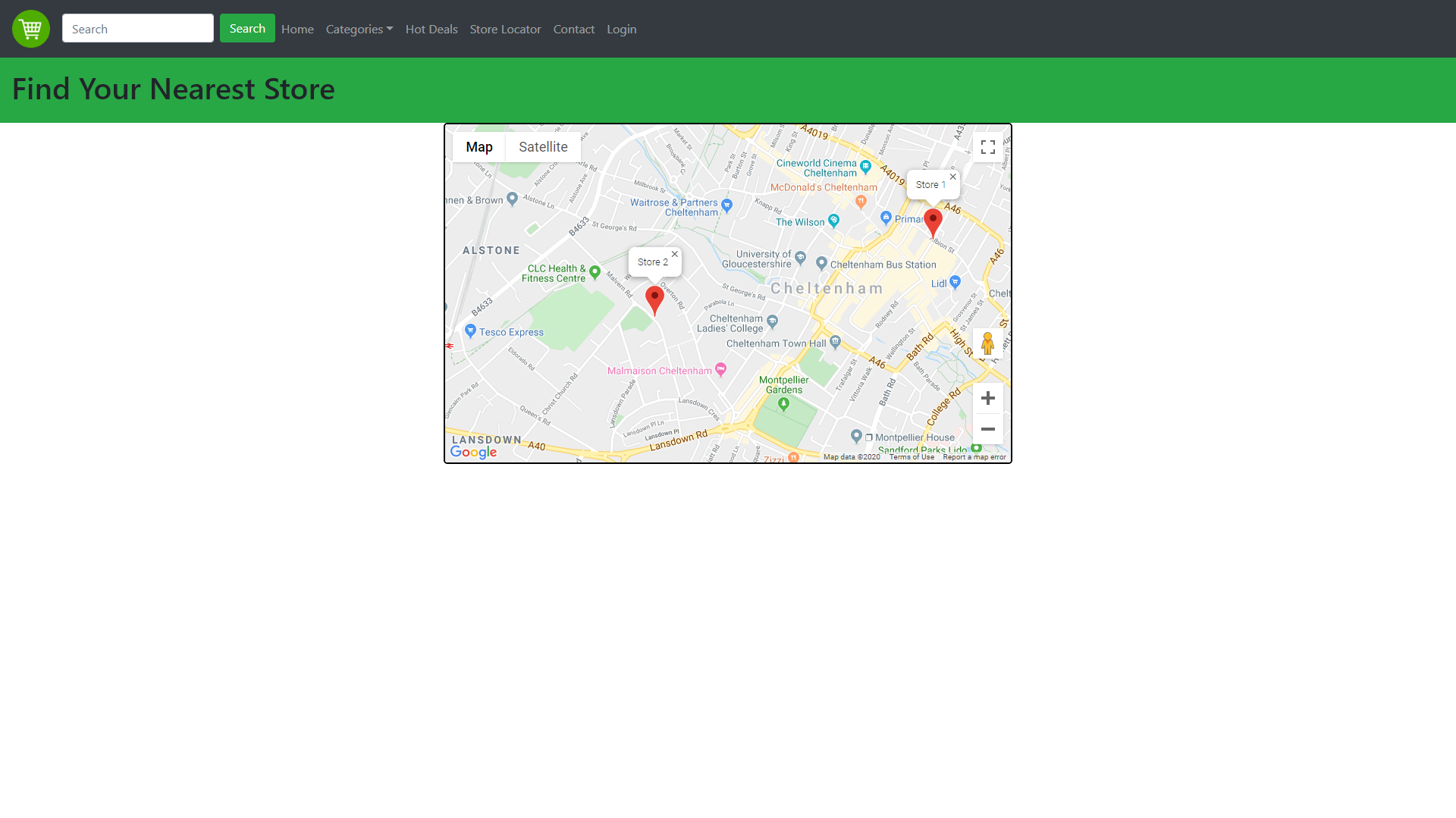
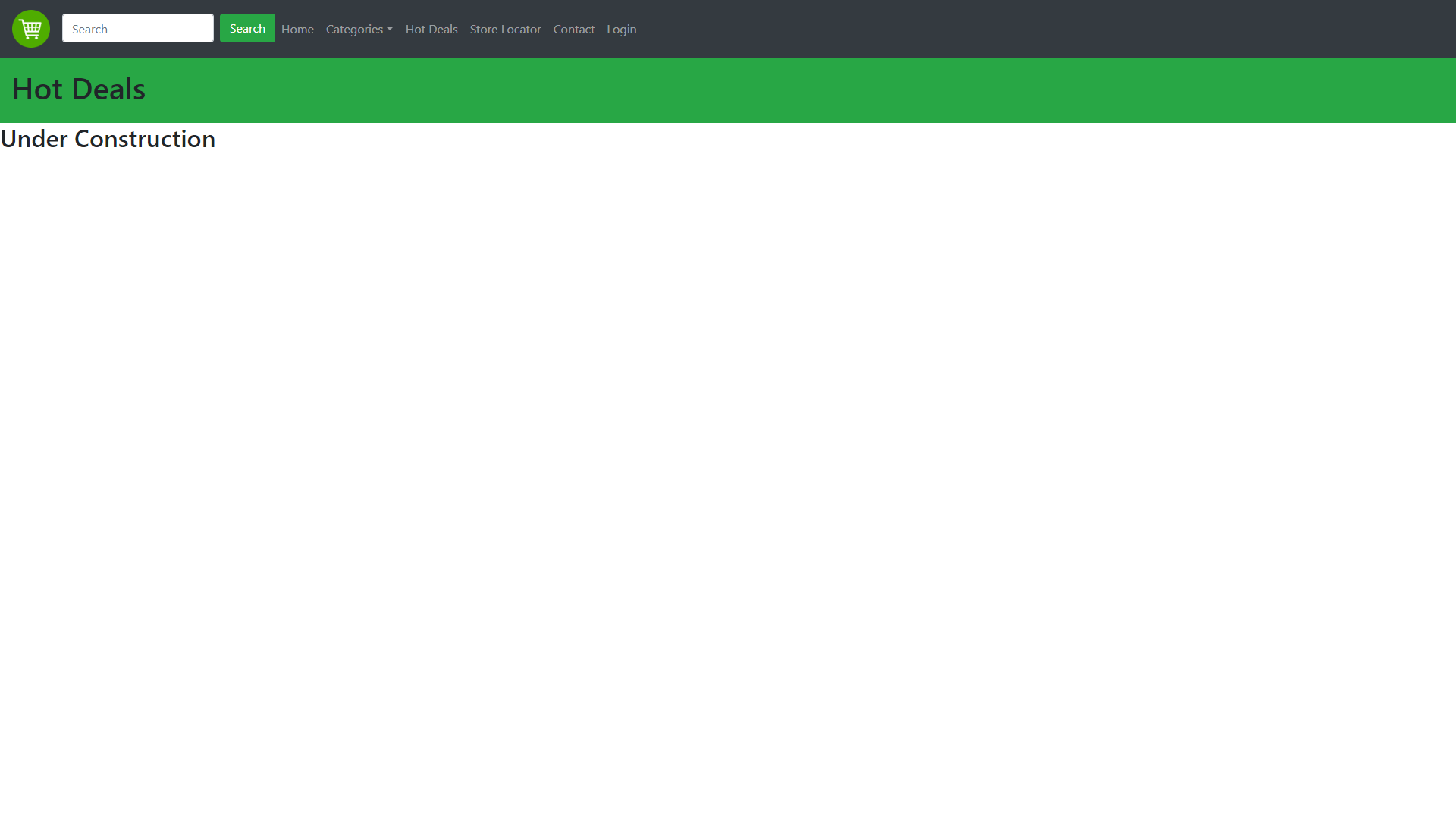
*HTML input type="checkbox"* (no date). Available at: https://www.w3schools.com/tags/att\_input\_type\_checkbox.asp (Accessed: 8 January 2020).

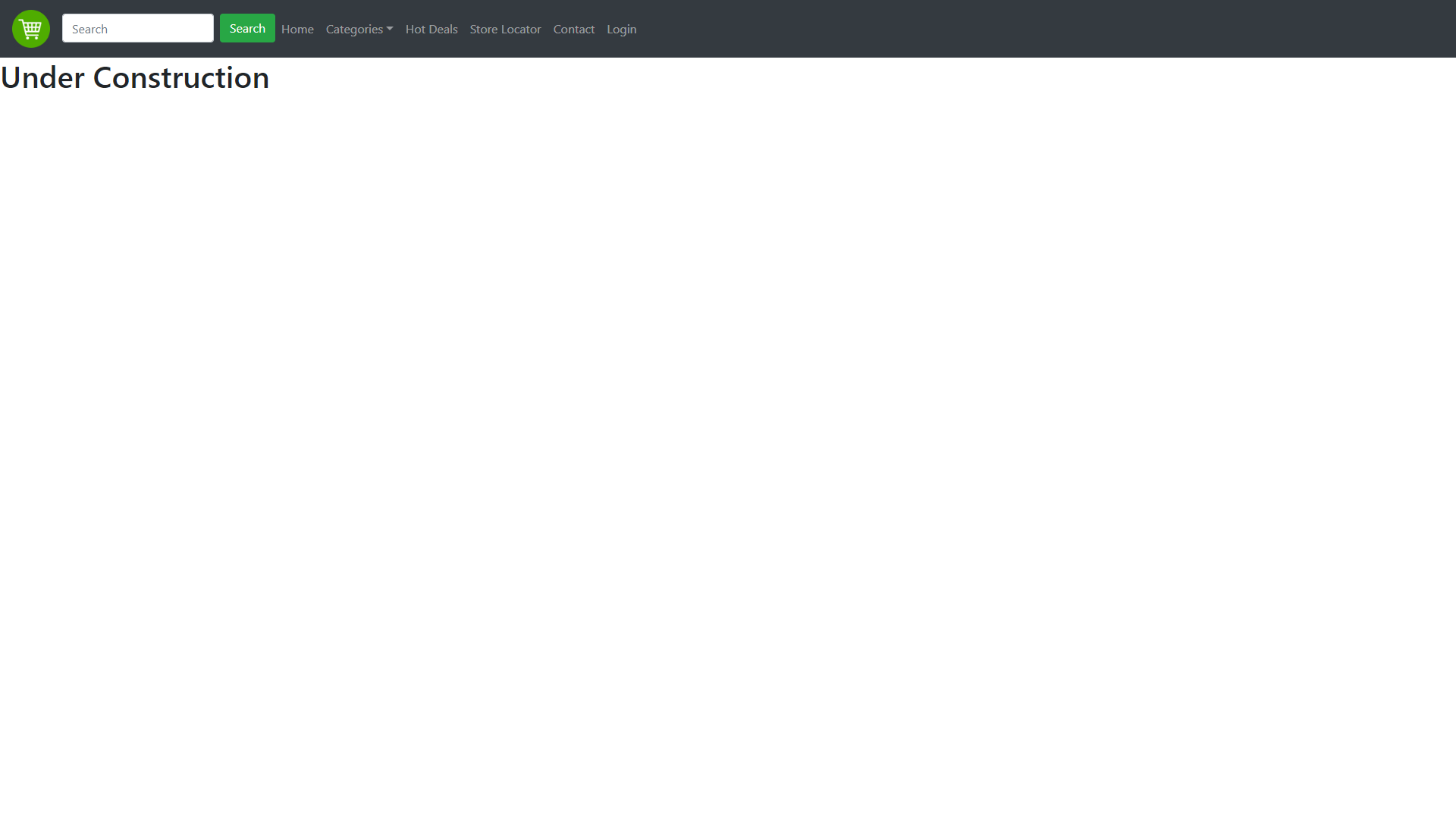
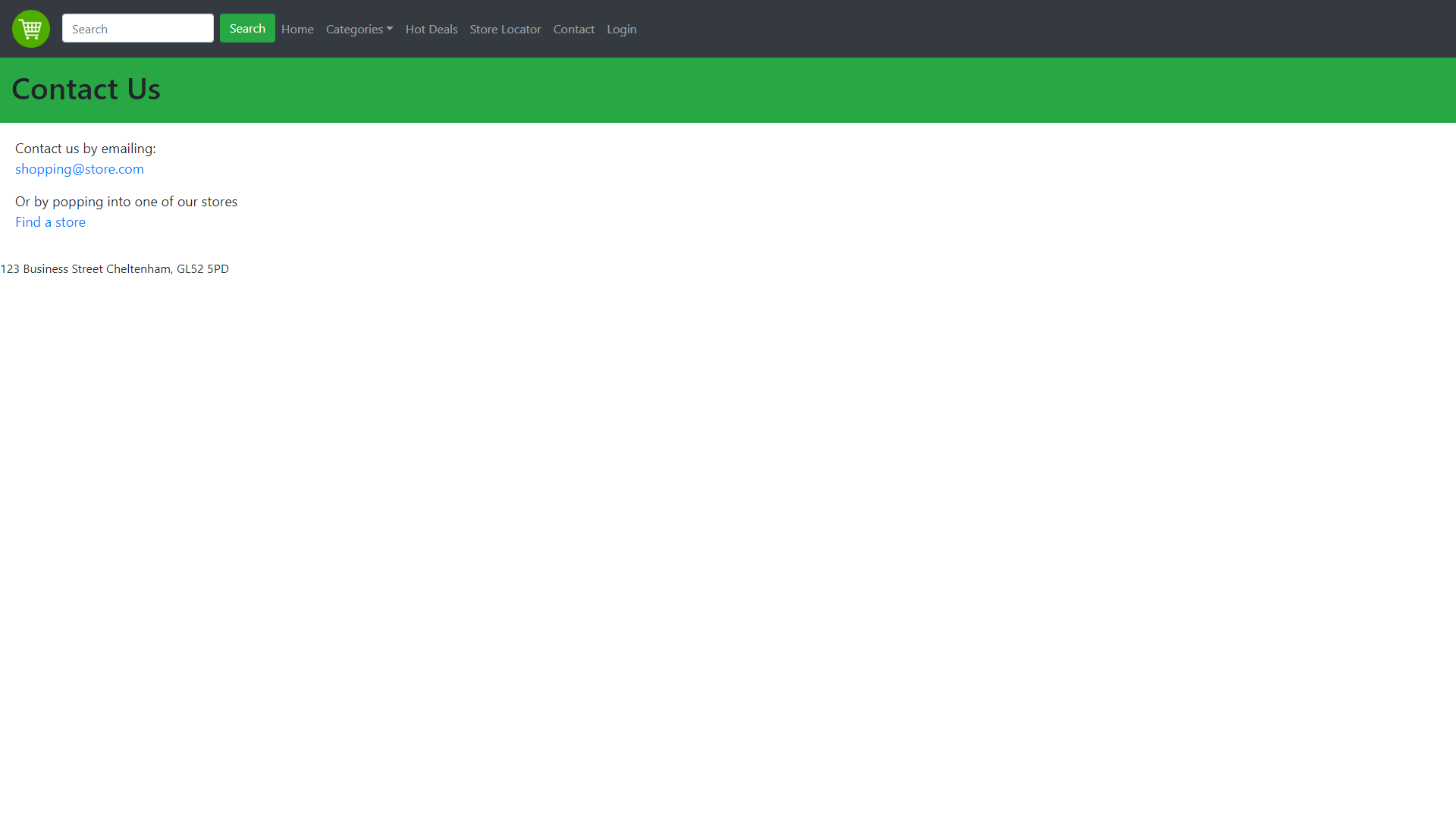
**== vs. ===**

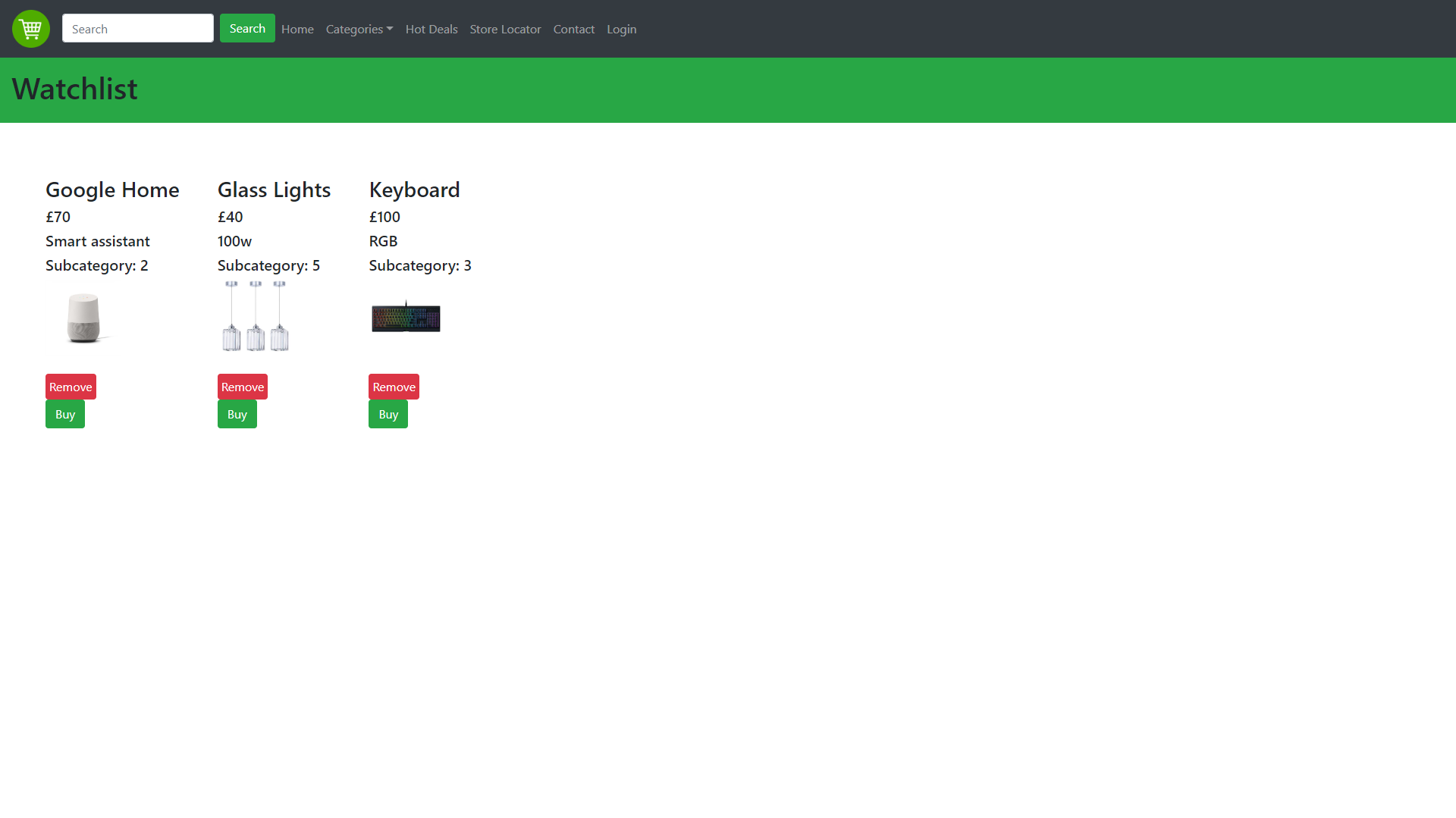
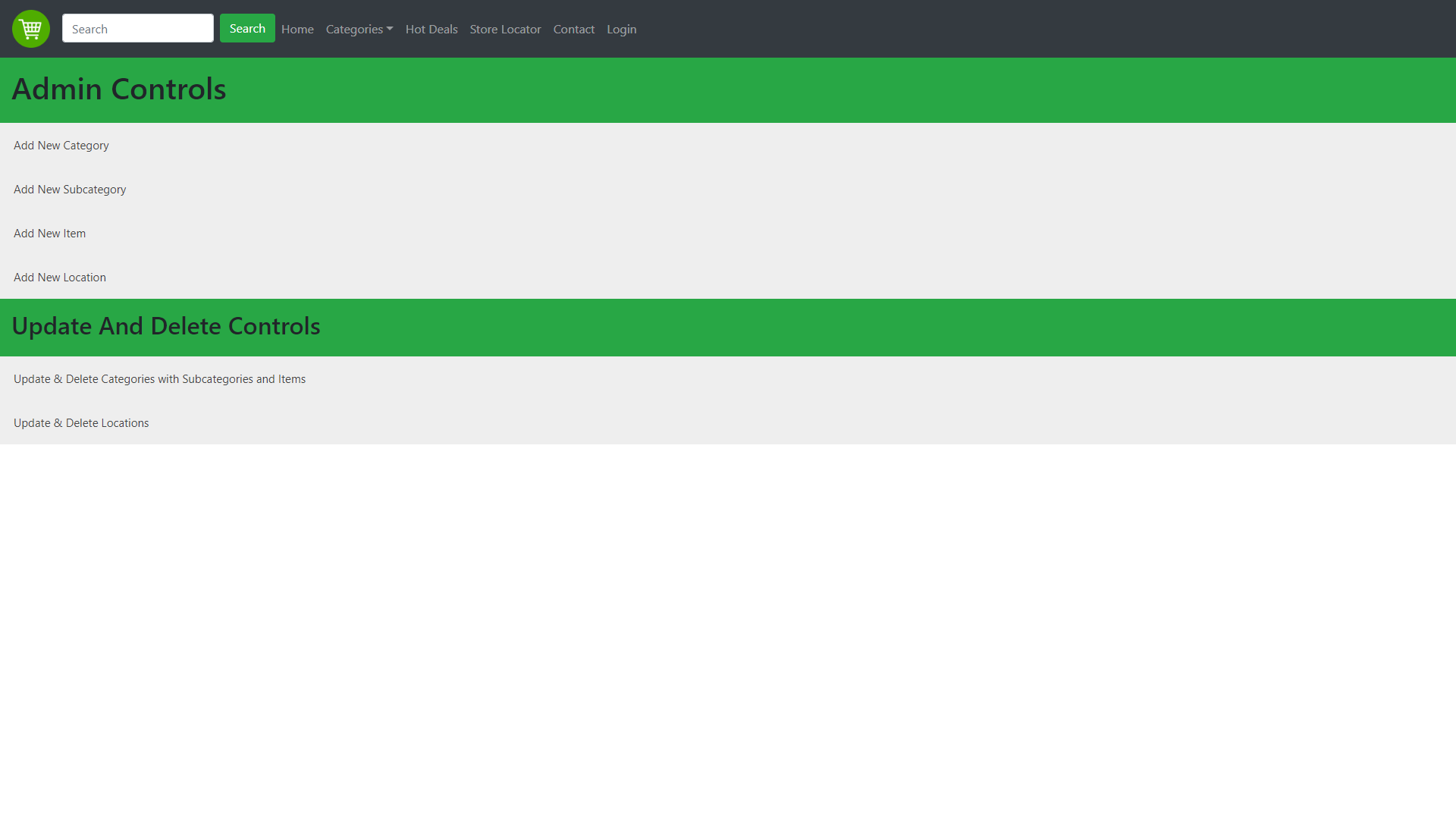
*JavaScript — Double Equals vs. Triple Equals - codeburst* (no date). Available at: https://codeburst.io/javascript-double-equals-vs-triple-equals-61d4ce5a121a (Accessed: 8 January 2020).

# Appendix 1







# Appendix 2