**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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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 | /js/main.js | As expected | N/A |
| 002 | Home Login | Login function admin   1. Correct credentials entered | User redirected to admin page  Code: redirect to target URL | /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 | Elements 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 element 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 elements 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  /js/lib/indexeddb.js  /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 filled out | /CatTemplate/catTemplate.html  /js/ListCat/listCat.js  /js/ListCat/initCompare.js |  |  |
| 015 |  |  |  |  |  |  |
| 016 |  |  |  |  |  |  |
| 017 |  |  |  |  |  |  |

# User Manual

# Interesting Use of CSS/HTML/JavaScript

# Shortcomings/Things to improve

Images have to use full path to locate them on client-side website. Need an upload function to add files to main server to be all in one place.

Watchlist items only work for one user – grabs all values set to 1.

Lots of files – slower for browser to load – benefits of SPA.

Problem with client side is you can’t guarantee you have permission to run IndexedDB or JavaScript. SQL or other databases are more streamlined and easy to use than IndexedDB.

Can’t encrypt logins on client side easily. Would normally be comparing hash values. Login details can also be accessed through developer tools looking at the db.

Hot deals page does nothing – in future add a checkbox in add item form to add to hot deals page.

# References

**Home Page**

**Nav Bar -**

<https://www.w3schools.com/bootstrap4/bootstrap_navbar.asp>

<https://www.w3schools.com/bootstrap4/tryit.asp?filename=trybs_navbar_dropdown&stacked=h>

**Login Form –**

<https://www.w3schools.com/bootstrap4/tryit.asp?filename=trybs_form_basic&stacked=h>

**Search Box –**

<https://www.w3schools.com/bootstrap4/tryit.asp?filename=trybs_navbar_form&stacked=h>

**Carousel**

<https://www.w3schools.com/bootstrap4/tryit.asp?filename=trybs_ref_js_carousel2>

**Indexeddb wrapper**

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

[Accessed 2019 November 18].

**Other indexeddb stuff (main.js)**

Help from my dad

<https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API/Using_IndexedDB>

<https://dev.to/andyhaskell/build-a-basic-web-app-with-indexeddb-38ef>

<https://www.vanamco.com/2014/11/14/indexeddb-fundamentals-plus-a-indexeddb-example-tutorial/>

**Accordion**

<https://www.w3schools.com/howto/howto_js_accordion.asp>

**Mail to on contact page**

<https://www.w3schools.com/tags/tryit.asp?filename=tryhtml_link_mailto>

**Misc. useful**

<https://stackoverflow.com/questions/17650776/add-remove-html-inside-div-using-javascript>

<https://www.w3schools.com/jsref/dom_obj_checkbox.asp>

<https://stackoverflow.com/questions/13405129/javascript-create-and-save-file>

https://stackoverflow.com/questions/42635126/bootstrap-4-navbar-with-2-rows

<https://getbootstrap.com/docs/4.0/components/navbar/>

<https://www.w3schools.com/bootstrap/bootstrap_buttons.asp>

    https://www.raymondcamden.com/2012/08/10/Searching-for-array-elements-in-IndexedDB

equality: == vs ===  see :

https://codeburst.io/javascript-double-equals-vs-triple-equals-61d4ce5a121a?gi=a7a512abed2a

cookies and js see :

<https://www.w3schools.com/js/js_cookies.asp>

**Checkboxes**

<https://www.w3schools.com/tags/att_input_type_checkbox.asp>

**Images**

<https://www.iconfinder.com/icons/536262/buy_cart_circle_ecommerce_green_shopping_trolley_icon>