WEB DEVELOPMENT 2 REPORT

Tatiana Zihindula

C16339923

DT228

Computer Science

School of Computing
Dublin Institute of technology
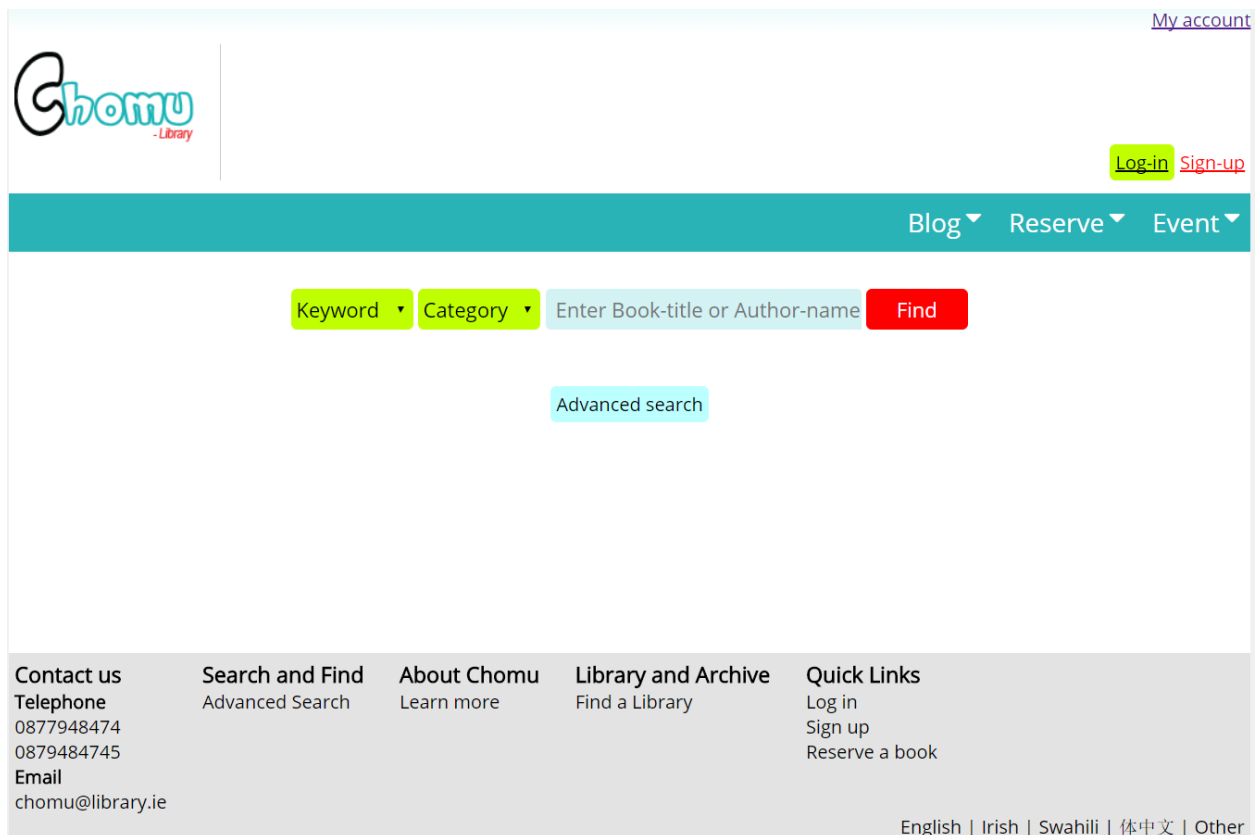
1/12/2017

YEAR 2

Introduction

The following report is about creating a library website model in which the user could perform task such as reservation, book search, log in, log out, and be able to view their details they entered when they log back into the system

Architecture

The website has three types of pages.

The index page

This page greets any user whether they are registered or not, and expose them to various features that they could access once they are logged in.
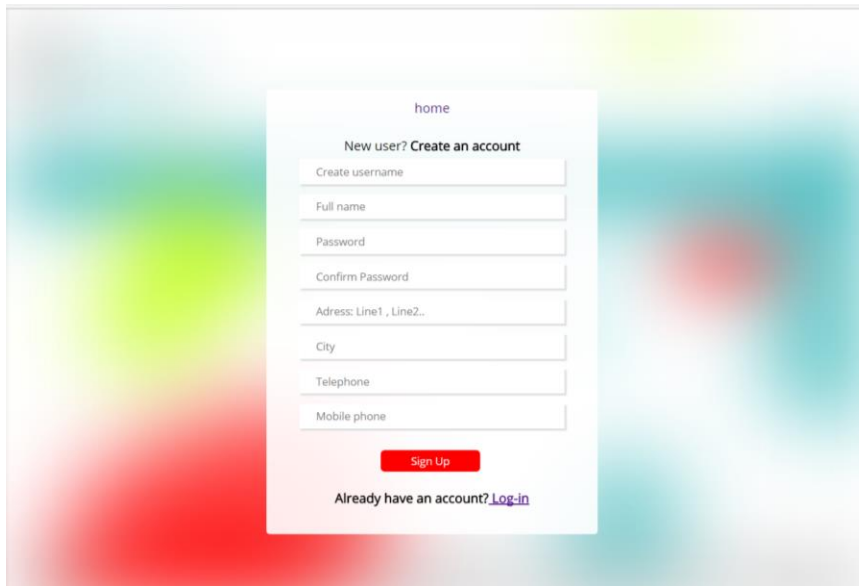

These features include: the ability to search a book, to reserve, and view the catalogue.

If the user is not registered or logged in, they will not be allowed to use any of these features, and will be sent to the login page or signup page.

The signup page takes the information from the user and inserts him/her in the database.

This page is deeply backed with a error checking page called *127.0.0.1/library/signup/sign_up_insert.php.* This page ensures that all the information provided by the user are sane.

If the user enters invalid data, for example enters username that already exists in the database, a short address name, a password that does not have more than eight digits, a phone number that is not made of digits or with password that do not match, they will be shown all the errors and asked to enter the data again.

This will continue until all the credentials are sane.

Below is the type of errors they may get with invalid credentials.

The login page

If the user has already been registered and want to use the library services again, they can do so by logging in the system using either the login button from the home page, via the signup page, and or try to perform task that only registered users are allowed to perform such as reserving a book, viewing he catalogue…

This page receives the input of a user and checks them against the ones currently existent in the database. This page is located under the login route. *127.0.0.1/library/login/* it is backed up with the *127.0.0.1/library/login/login_check.php* page that ensures that the

data provided by the user is correct. If the user enters an invalid password, they get thrown the following error.
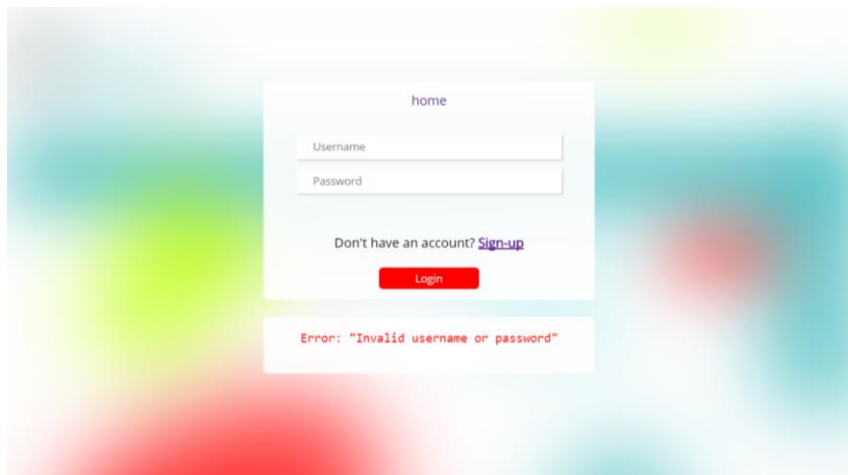
The user page

The user page is unique in the sense that the user can only access it if they are logged in. It features their username at the left-hand side, greet them with a search bar, and up to the right-hand side, display features that they can perform such as view the catalogue, view books they have reserved, and log out of the system if they so choose.
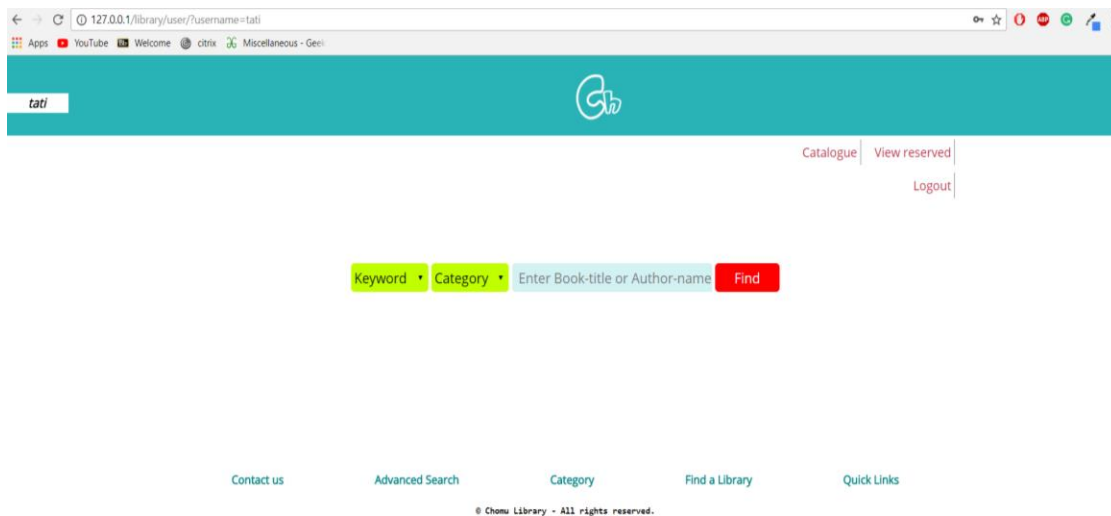
The username also gets displayed in the URL so they can se where in the page hierarchy they are.

This also applies to the rest of the pages. For example the following pages shows the user exactly where they are in the hierarchy.

When the user is logged in, the home pages remembers him and customises itself to the user's information.

BEFORE
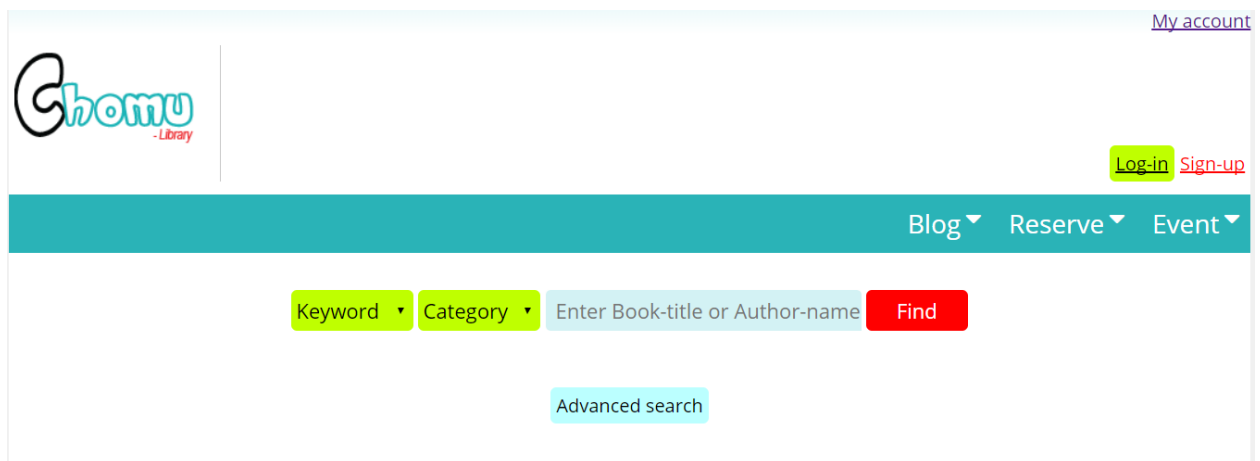
AFTER
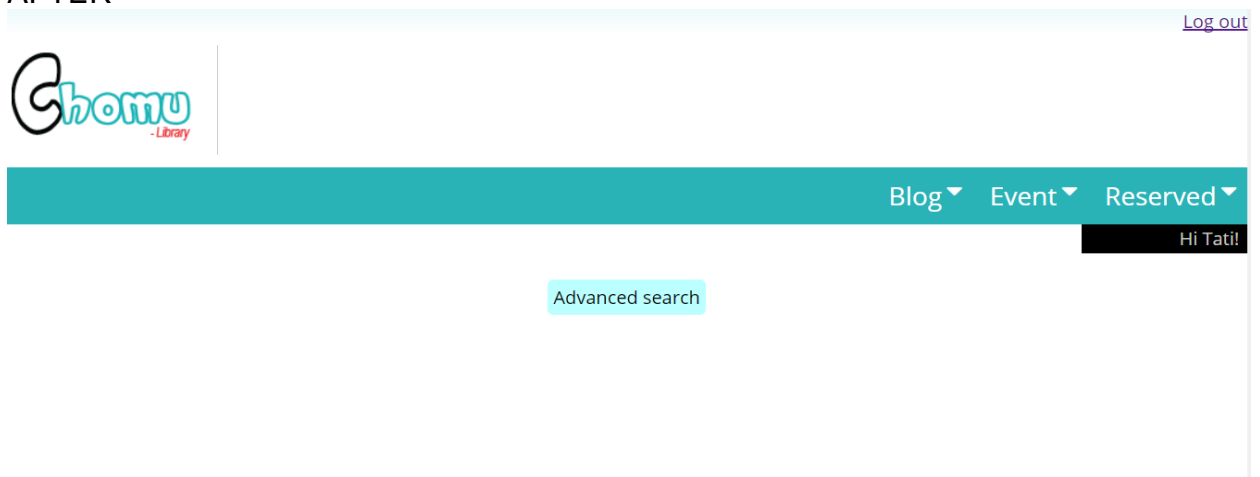
Now that the search bar has been moved to the user page, the user can now perform the advanced search here, if they so choose to do. This feature, was not

asked to be implemented, but it simply shows the types of taks a library user would be performing if they were logged in, the advanced search feature could also be accessed throuough the user page's footer.

This feature would be similar to the one the user could perform but advanced SQL querries such as grouping and unions would be used to ensure the user gets the best experience using the system.


Searching

Searching for a book in the database is done in many different ways.

Searching by book name

This is the default search and the user does not need to select anything to search a book.
They do not even need to enter the full name of the book.
This has been achieved by guessing what the user wants to enter using regular expression in the SQL where clauses.

This was done using a query builder function that would table in the arguments passed and create custom SQL query for it.

```php
else
{

    if($table_name == 'book')
        $query = query_builder("book",
                                $default_cols,
                                array(array("JOIN", "category","category_id")),
                                array("category_id", "=", $category),
                                array(array("AND", "upper(book_title)", "LIKE",
                                      strtoupper($to_search))));
    else if ($table_name == 'author')
        array_push($default_cols,'category_description as CATEGORY');
        $query = query_builder("book",
                                $default_cols,
                                array(array("JOIN", "category","category_id"),
                                      array("JOIN", "author", "author_id")),
                                array("category_id", "=", $category),
                                array(array("AND", "upper(author_name)", "LIKE",
                                      strtoupper($to_search))));
```

The function creating the querry is **127.0.0.1/library/find.php**

```php
190        }
191
192        function query_builder($table_name, $colum_name, $joins, $wheres, $and_or)
193        {
194            $join_builder= array();
195            $where_builder = array();
196            $and_or_builder = array();
197
198            # building the joins
199            if(count($joins) > 0)
200                foreach ($joins as $build)
201                    array_push($join_builder, " ". $build[0]." " . $build[1]." USING(".$build[2].") ");
202
203            # building the where clauses
204            if(count($wheres) > 0)
205                array_push($where_builder, " WHERE ". $wheres[0]." " . $wheres[1]." '".$wheres[2]."' ");
206
207            if(count($and_or) > 0)
208                foreach($and_or as $cond)
209                    array_push($and_or_builder, $cond[0]." " . $cond[1]." ".$cond[2]." '".$cond[3]."'" );
210
211
212            $query ="SELECT  ". implode(", ", $colum_name).
213                      " FROM ". $table_name . implode(" ", $join_builder).
214                      implode(" ", $where_builder). implode(",", $and_or_builder);
215
216            return $query;
217        }
218
219
```

Reserve a book.

The user can choose to reserve a book, Only after they are logged in and if the book is not already reserved.
The books available for reservation are styled in green, and those not available are shown in red.

BOOK	AUTHOR	YEAR	EDITION	STATUS	CATEGORY
Business Strategy	Joe Peppard	1997	2	Reserved	Health
A guide to nutrition	John Thorpe	1997	2	Reserved	Health
My life in picture	Dan Brown	2004	8	Reserved	Health
My ranch in Texas	John Smith	2005	1	Reserved	Health
My life in bits	Cleo Blair	2003	1	Reserve	Health
Shooting History	Alicia Oneill	2003	1	Reserve	Health

Result
12

If the user reserves a book that has been reserved by a different user they get the following error.

Keyword ▾ Category ▾ Enter Book-title or Author-name Find

Sorry, book already reserved

Contact us   Advanced Search   Category   Find a Library   Quick Links

© Chomu Library - All rights reserved.

Cancel a reservation

If the user is the one who has reserved the book, they can choose to cancel it by clicking on the cancel button, or go the page that enables them to view reserved pages and delete all the reservations they have in the system if they so choose to do.

Catalogue | View reserved

Logout

Keyword ▾ Category ▾ Enter Book-title or Author-name Find

Result
1

BOOK	AUTHOR	YEAR	EDITION	STATUS	CATEGORY
DaVinci Code	Jamie Oliver	2003	1	Cancel	Fiction
Tara Road	Jon Snow	2002	4	Reserve	Fiction

To unreserve a book, the user simply has to click on the cancel button.

TO view all the books they have reserved in the system, they can click on the view reserved in the top right corner.

Catalogue | View reserved

Logout

Tis feature will bring them to this page, where they can view all the pages they have reserved.
Below are all the books currently reserved by the user **tati**.

Logout

Finally when the user logs out, they are redirected to the home page, and all the transactions they effectuated in the database stay saved for them until the next session.

Pages hierarchy.

📁 search

  ✳ index.php

  ✳ search_form.php

📁 signup

  📁 css

    </> signup.css

  📁 media

    🖼 background.png

  ☰ index.html

  ✳ login_footer.php

  ✳ sign_up_insert.php

  ✳ signup_view.php

📁 user

  📁 css

    </> user.css

  ✳ index.php

  ✳ logout.php

  ✳ reserve.php

Line 1, Column 1

⊞ Search Windows

  ✳ index.php

  ✳ logout.php

  ✳ reserve.php

  ✳ reserve_form.php

  ✳ user_footer.php

  ✳ user_header.php

✳ advanced_search.php

✳ connect.php

✳ find.php

✳ footer.php

✳ functions.php

✳ get.php

✳ head.php

✳ header.php

</> lab12.sql

</> library.sql

✳ rect.php

✳ WEB DEVELOPMENT 2 REPOR