Table of Contents

Part 1 1
Summary 1
Use Case 2
Screenshots 4
Software Requirements 4
Creation Form 5
Update Form 6
Delete Form 7
Logging 8
Log Analytics 8
Browsing & Filtering 9
Progressive Query 12
Keyword Search Auto-complete 17
Filter Ordering 18
API Form (Extra) 19
At Least 50 Records 19
Part 2 20
Configuration Setup (XAMPP) 20
Design Overview of Code 20
Folder Structure 20
Code Structure 22
Design Overview of DB 24
Detailed PHP, HTML5, CSS, JavaScript Code 25
Detailed PHP and HTML5 Code (Layout, User & Admin Content) 25
webpage_user.php 25
webpage_dev.php 26
webpage_user_content.php 27
webpage_admin_content.php 35
Detailed PHP Code (Code involving AJAX response) 52
user_progressiveFacetSearch.php 52
dev_recordCreate_OMDbAPI.php 55
dev_recordRefIDList_Form.php 56
dev_recordTitleList_Form.php 59

dev_recordInfo_Form.php 62
dev_downloadLog.php 63
Detailed PHP Code (Index & PHP-Commons) 65
Index.php 65
Database.php 65
Constants.php 65
Filters.php 67
Functions.php 70
Collection.php 78
Detailed CSS Code 79
cssCommons.css 79
cssUserPage.css 81
cssAdminPage.css 88
Detailed JavaScript Code 97
scriptCommons.js 97
script.js 104

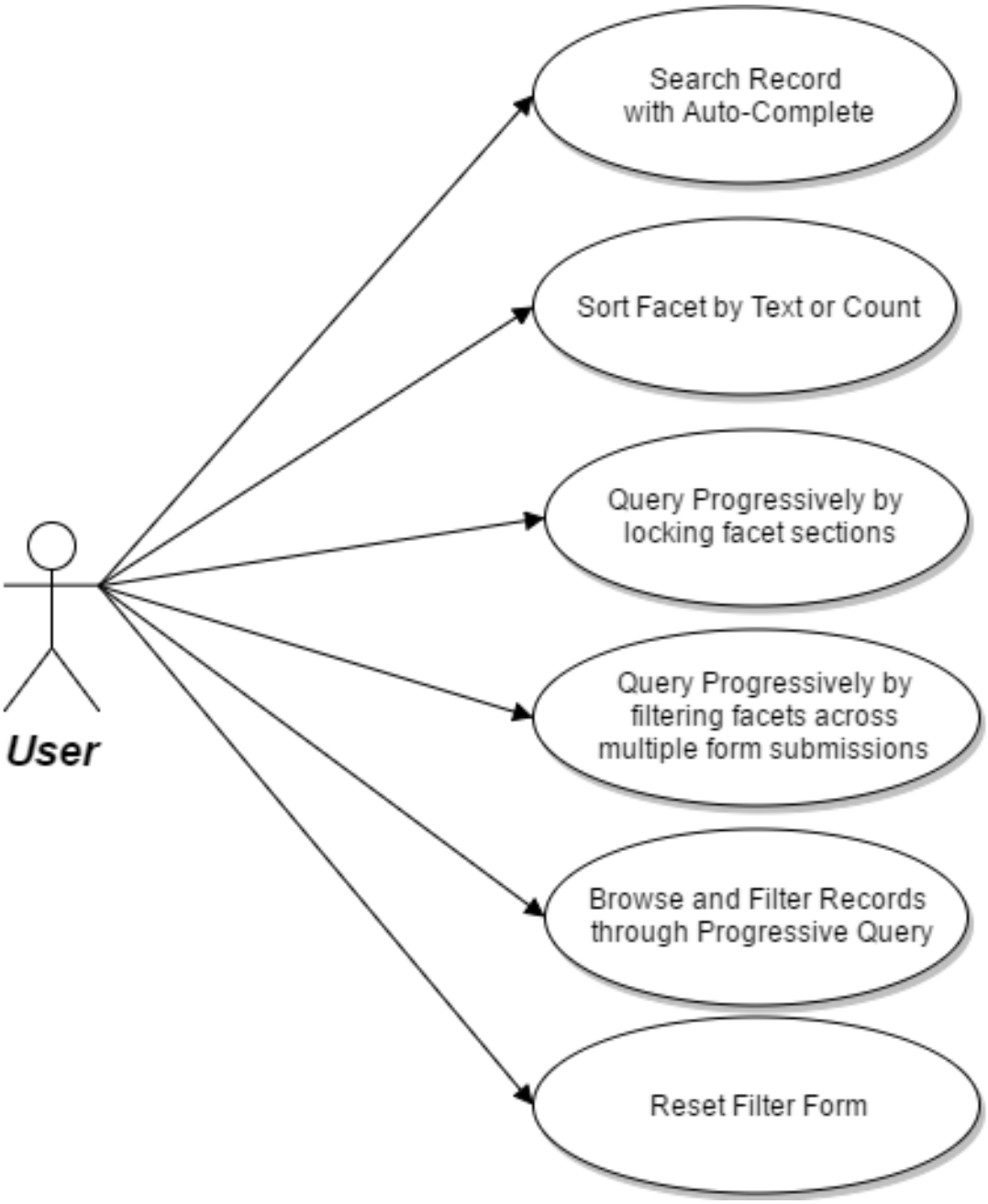
Part 1

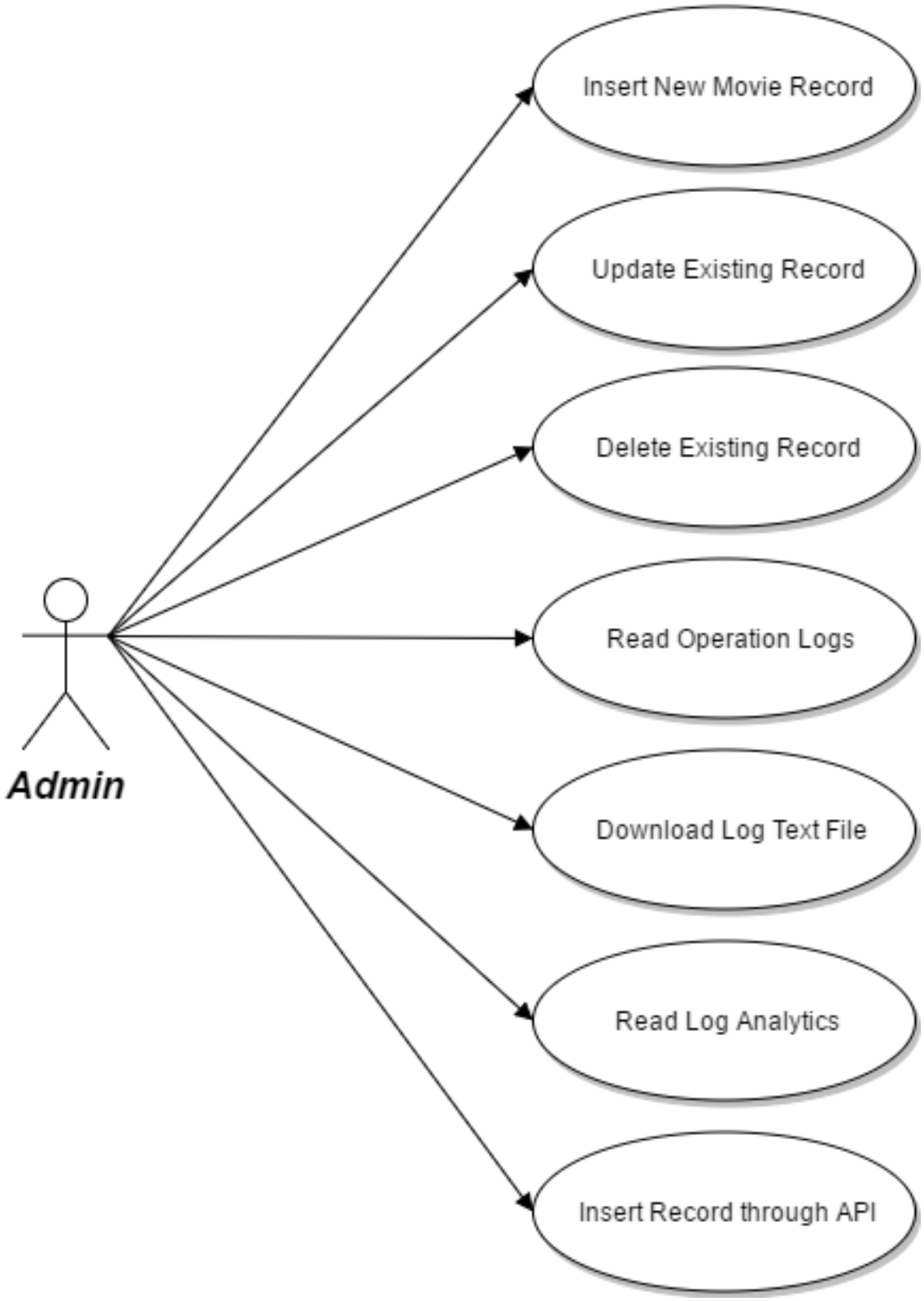
Summary

The web application that was implemented is an application that complements IMDB's website in searching and filtering through multiple movies and series by filtering with facets, which would help users locate the item without recalling the specific details of it. This is done by filtering and browsing the list of movies and series that was obtained through IMDB website's API, OMDbApi.

This process is also supported by an admin interface that allows the admin to insert new records, update and delete existing records, as well as logging the operations made by the admin.

Use Case


Search Record
with Auto-Complete
Sort Facet by Text or Count
Query Progressively by
locking facet sections
Query Progressively by
filtering facets across
multiple form submissions
Browse and Filter Records
through Progressive Query
Reset Filter Form
User


Insert New Movie Record
Update Existing Record
Delete Existing Record
Read Operation Logs
Download Log Text File
Read Log Analytics
Insert Record through API
Admin

Screenshots

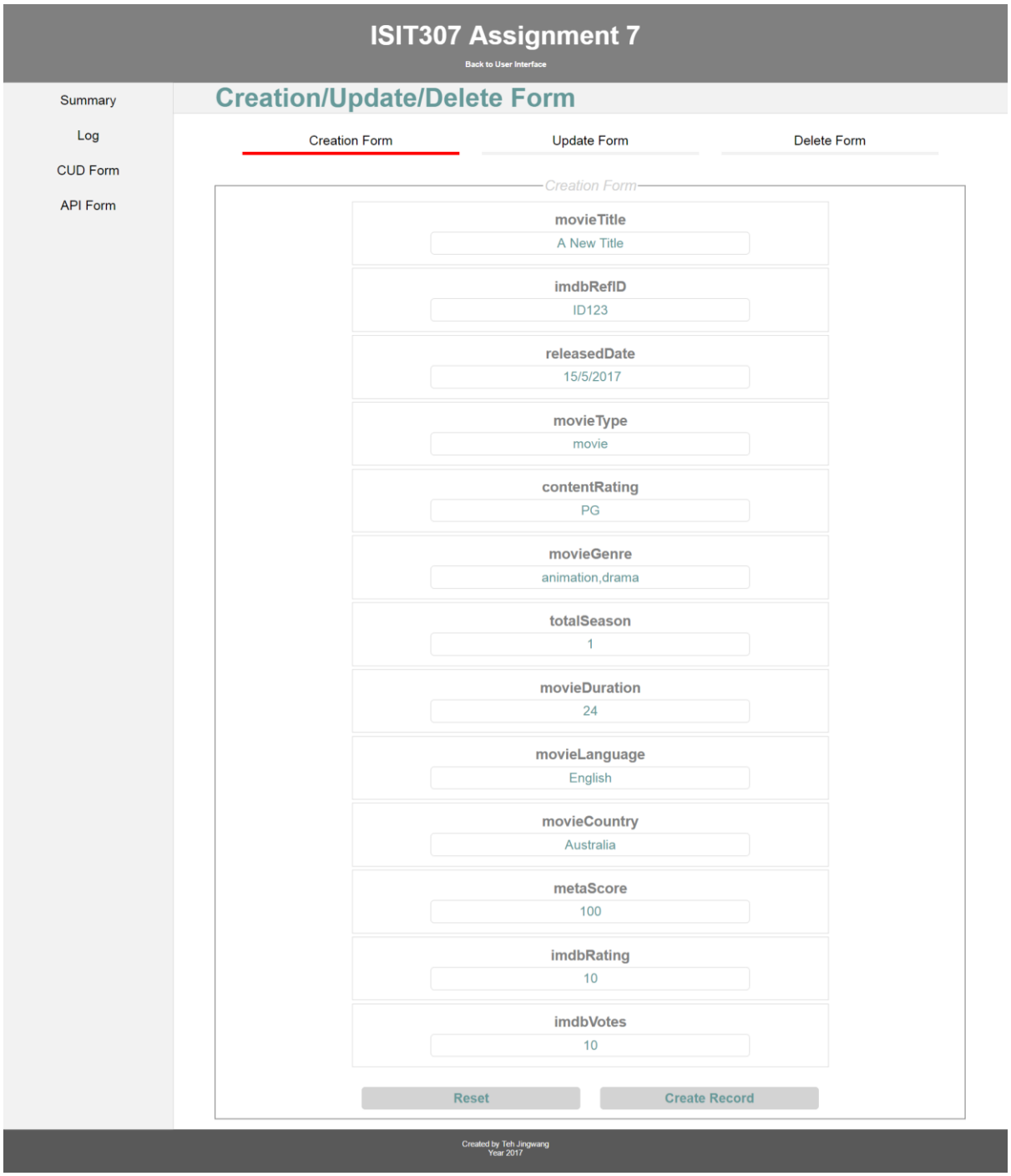
Software Requirements

No.	Requirement	Title	Page
Standard Level			
(a)	Faceted Search Interface (with 3 filters, and intersection/filter & reset controls)	Browsing & Filtering (No Facet Set)	9 – 11
(b)	Faceted Search Count after single query step	Progressive Query (Single-Step Query)	12
(c)	Progressive Query Works (narrowing count values & preserving user prior choices)	Progressive Query (Multi-Step Query – Locking Design & Filter Design)	13 – 16
(d)	Add/Update/Delete items record (with sticky update form)	Creation Form, Update Form, Delete Form	5, 6, 7
(e)	Add/Update/Delete items record log	Logging	8
Advanced Level			
(a)	Integrates keyword search with auto-complete	Keyword Search Auto Complete	17
(b)	Supports numeric interval facet	Browsing & Filtering (No Facet Set)	9 – 11
(c)	Supports 2-level category tree for at least one facet	Browsing & Filtering (No Facet Set)	9 – 11
(d)	Supports analytics over item add/update/delete forms	Log Analytics	8
(e)	Supports change of facet category order (text & count ordering)	Filter Ordering	18

Creation Form

A simple creation form that allows admin to insert data into database.

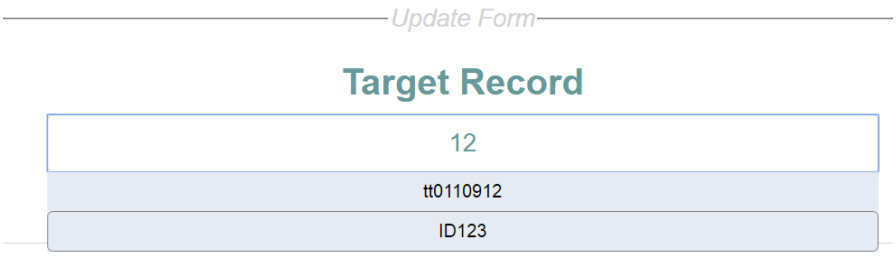
If there are invalid fields, sticky fields will be applied for correct fields, and errors for incorrect fields.

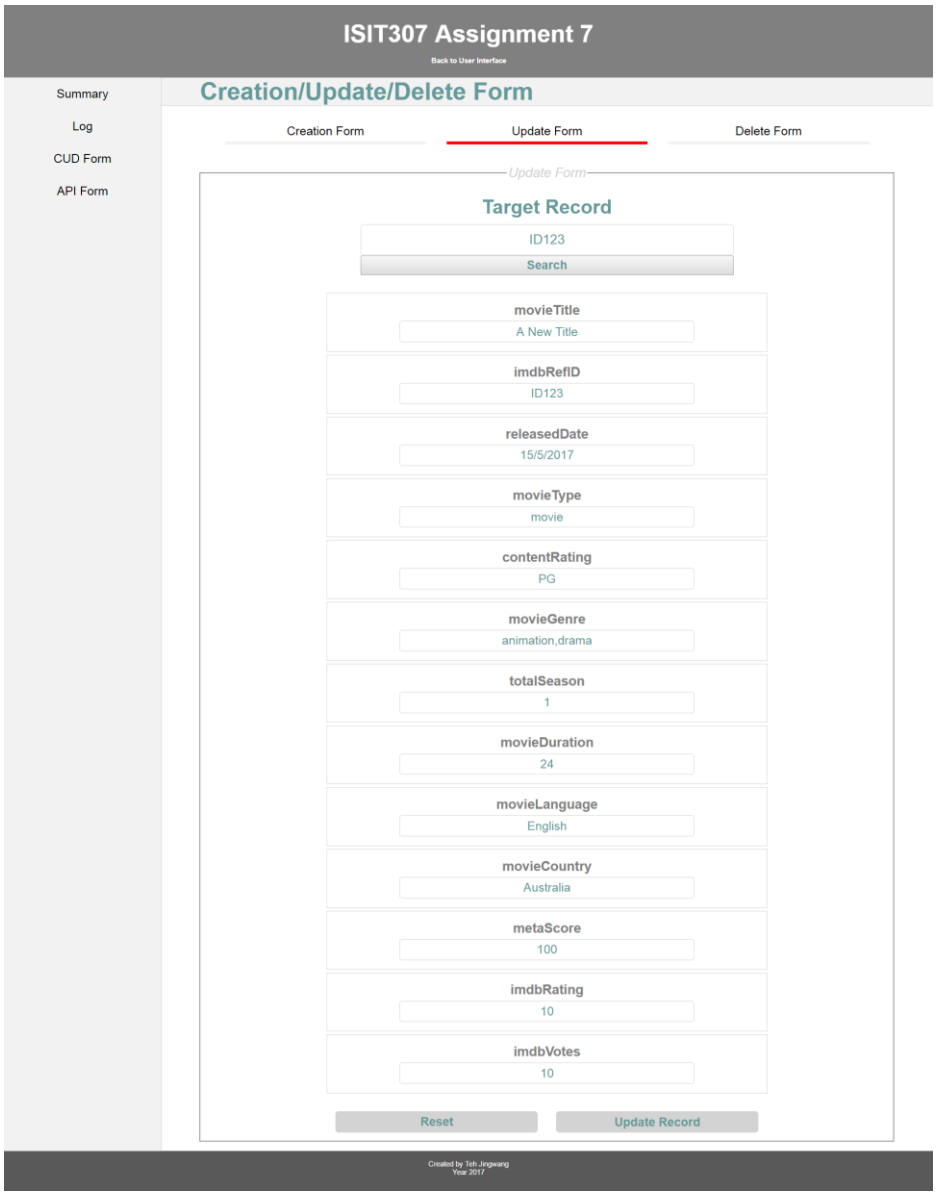


Update Form

An update form that allows admin to search record by IMDbRefID and update the record details into database. After searching record, the fields will be populated by the record values.

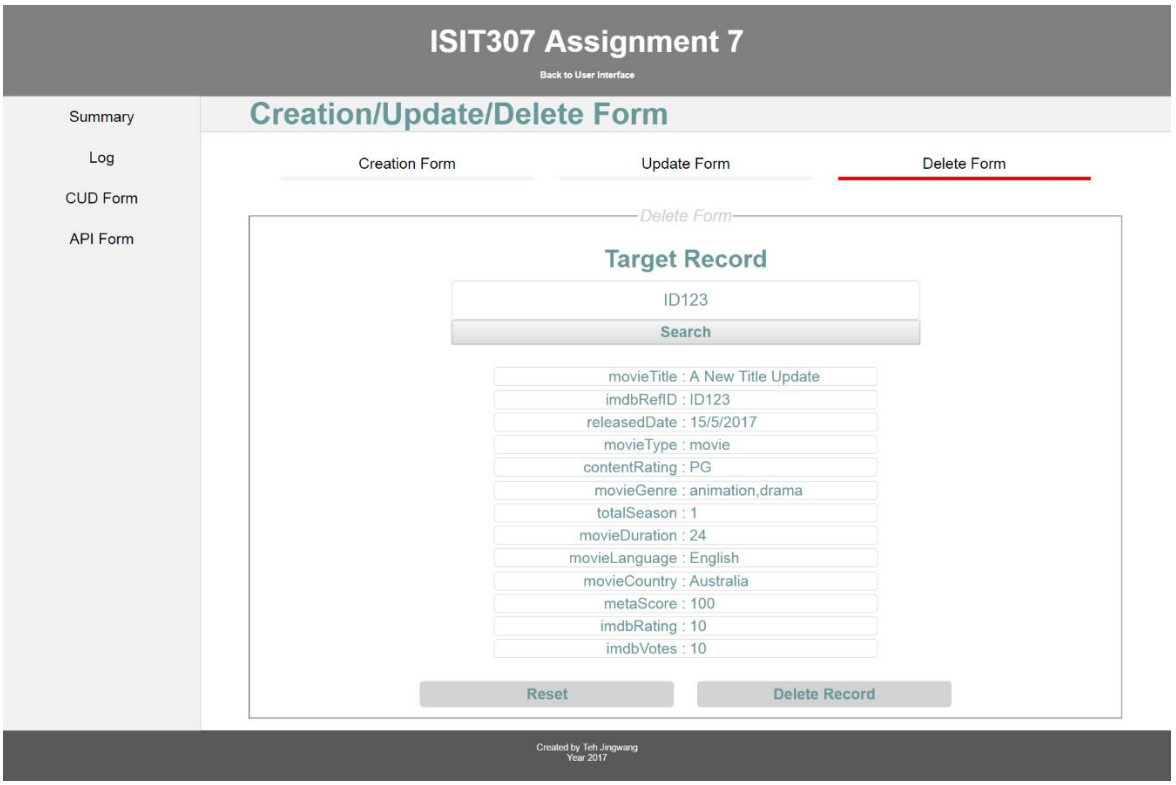
If there are invalid fields, sticky fields will be applied for correct fields, and errors for incorrect fields.





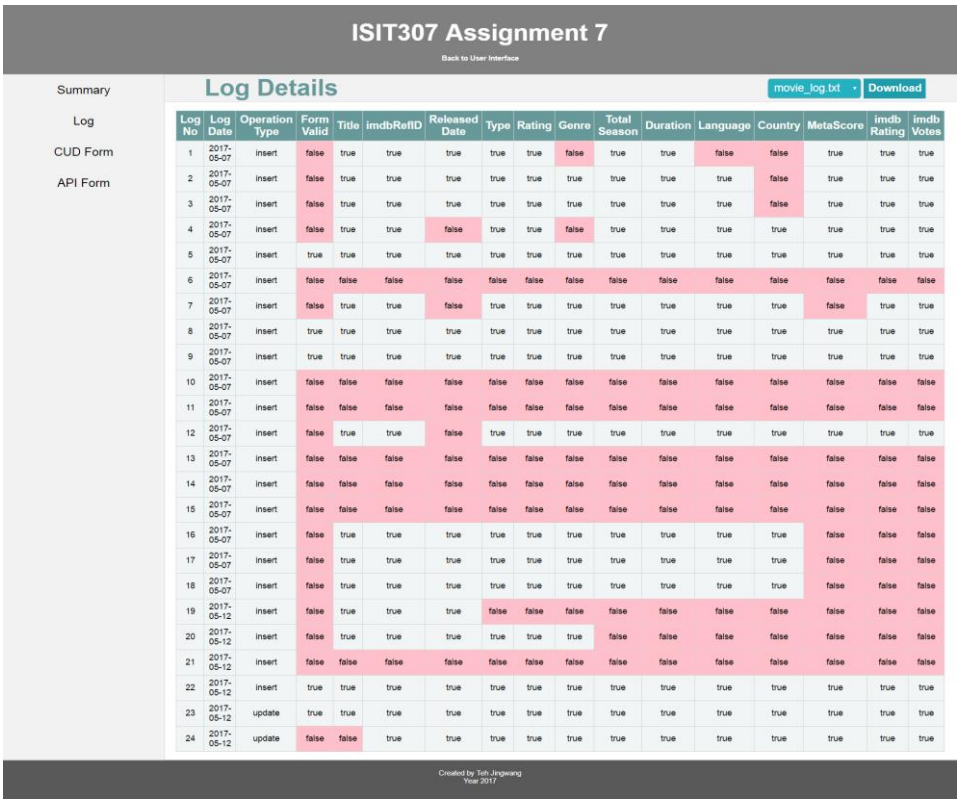
Delete Form

A delete form that allows admin to search record by IMDbRefID and delete the record from database. After searching the record, the record details will be shown below the search input.



Logging

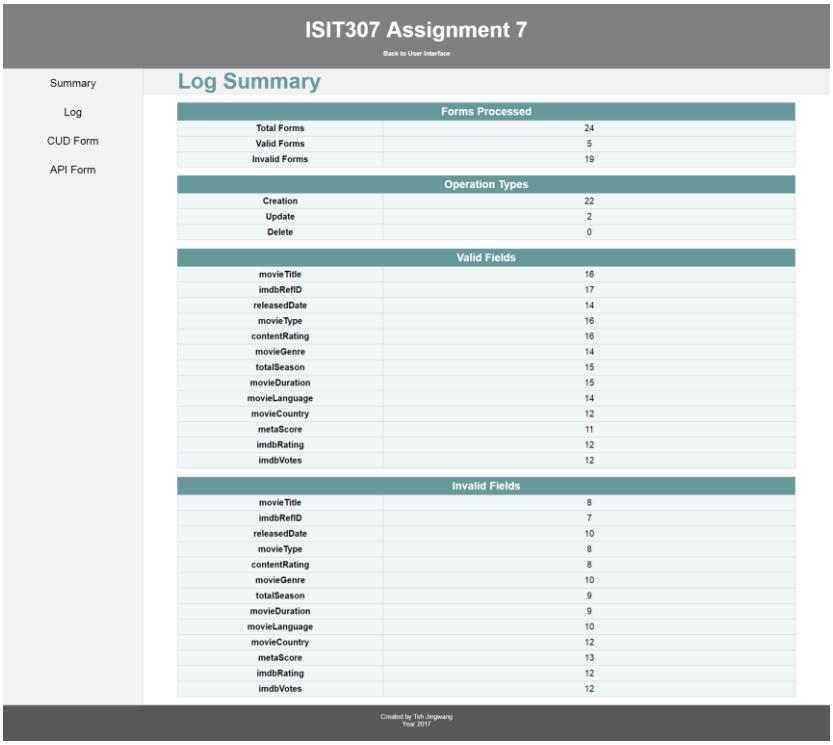
A table that displays log records over the last 7 days from movie_log.txt



Log No	Log Date	Operation Type	Form Valid	Title	imdbRefID	Released Date	Type	Rating	Genre	Total Season	Duration	Language	Country	MetaScore	imdb Rating	imdb Votes
1	2017-05-07	insert	false	true	true	true	true	true	false	true	true	false	false	true	true	true
2	2017-05-07	insert	false	true	true	true	true	true	true	true	true	true	false	true	true	true
3	2017-05-07	insert	false	true	true	true	true	true	true	true	true	true	false	true	true	true
4	2017-05-07	insert	false	true	true	false	true	true	false	true	true	true	true	true	true	true
5	2017-05-07	insert	true	true	true	true	true	true	true	true	true	true	true	true	true	true
6	2017-05-07	insert	false	false	false	false	false	false	false	false	false	false	false	false	false	false
7	2017-05-07	insert	false	true	true	false	true	true	true	true	true	true	true	false	true	true
8	2017-05-07	insert	true	true	true	true	true	true	true	true	true	true	true	true	true	true
9	2017-05-07	insert	true	true	true	true	true	true	true	true	true	true	true	true	true	true
10	2017-05-07	insert	false	false	false	false	false	false	false	false	false	false	false	false	false	false
11	2017-05-07	insert	false	false	false	false	false	false	false	false	false	false	false	false	false	false
12	2017-05-07	insert	false	true	true	false	true	true	true	true	true	true	true	true	true	true
13	2017-05-07	insert	false	false	false	false	false	false	false	false	false	false	false	false	false	false
14	2017-05-07	insert	false	false	false	false	false	false	false	false	false	false	false	false	false	false
15	2017-05-07	insert	false	false	false	false	false	false	false	false	false	false	false	false	false	false
16	2017-05-07	insert	false	true	true	true	true	true	true	true	true	true	true	false	false	false
17	2017-05-07	insert	false	true	true	true	true	true	true	true	true	true	true	false	false	false
18	2017-05-07	insert	false	true	true	true	true	true	true	true	true	true	true	false	false	false
19	2017-05-12	insert	false	true	true	true	false	false	false	false	false	false	false	false	false	false
20	2017-05-12	insert	false	true	true	true	true	true	true	false	false	false	false	false	false	false
21	2017-05-12	insert	false	false	false	false	false	false	false	false	false	false	false	false	false	false
22	2017-05-12	insert	true	true	true	true	true	true	true	true	true	true	true	true	true	true
23	2017-05-12	update	true	true	true	true	true	true	true	true	true	true	true	true	true	true
24	2017-05-12	update	false	false	true	true	true	true	true	true	true	true	true	true	true	true

Log Analytics

Tables that displays a summary of the logs over the last 7 days.



Forms Processed	
Total Forms	24
Valid Forms	5
Invalid Forms	19

Operation Types	
Creation	22
Update	2
Delete	0

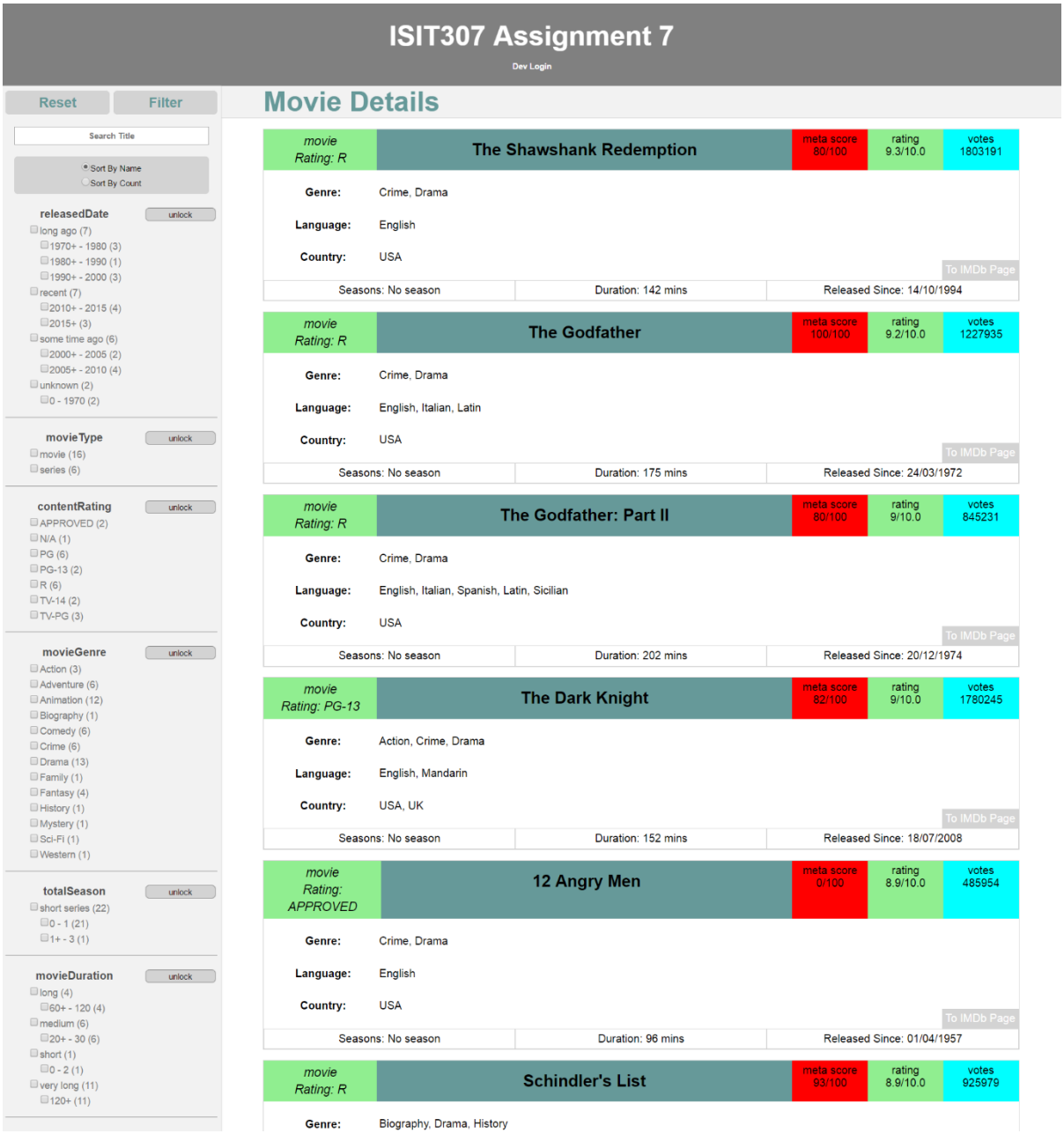
Valid Fields	
movieTitle	16
imdbRefID	17
releasedDate	14
movieType	16
contentRating	16
movieGenre	14
totalSeason	15
movieDuration	15
movieLanguage	14
movieCountry	12
metaScore	11
imdbRating	12
imdbVotes	12

Invalid Fields	
movieTitle	8
imdbRefID	7
releasedDate	10
movieType	8
contentRating	8
movieGenre	10
totalSeason	9
movieDuration	9
movieLanguage	10
movieCountry	12
metaScore	13
imdbRating	12
imdbVotes	12

Browsing & Filtering

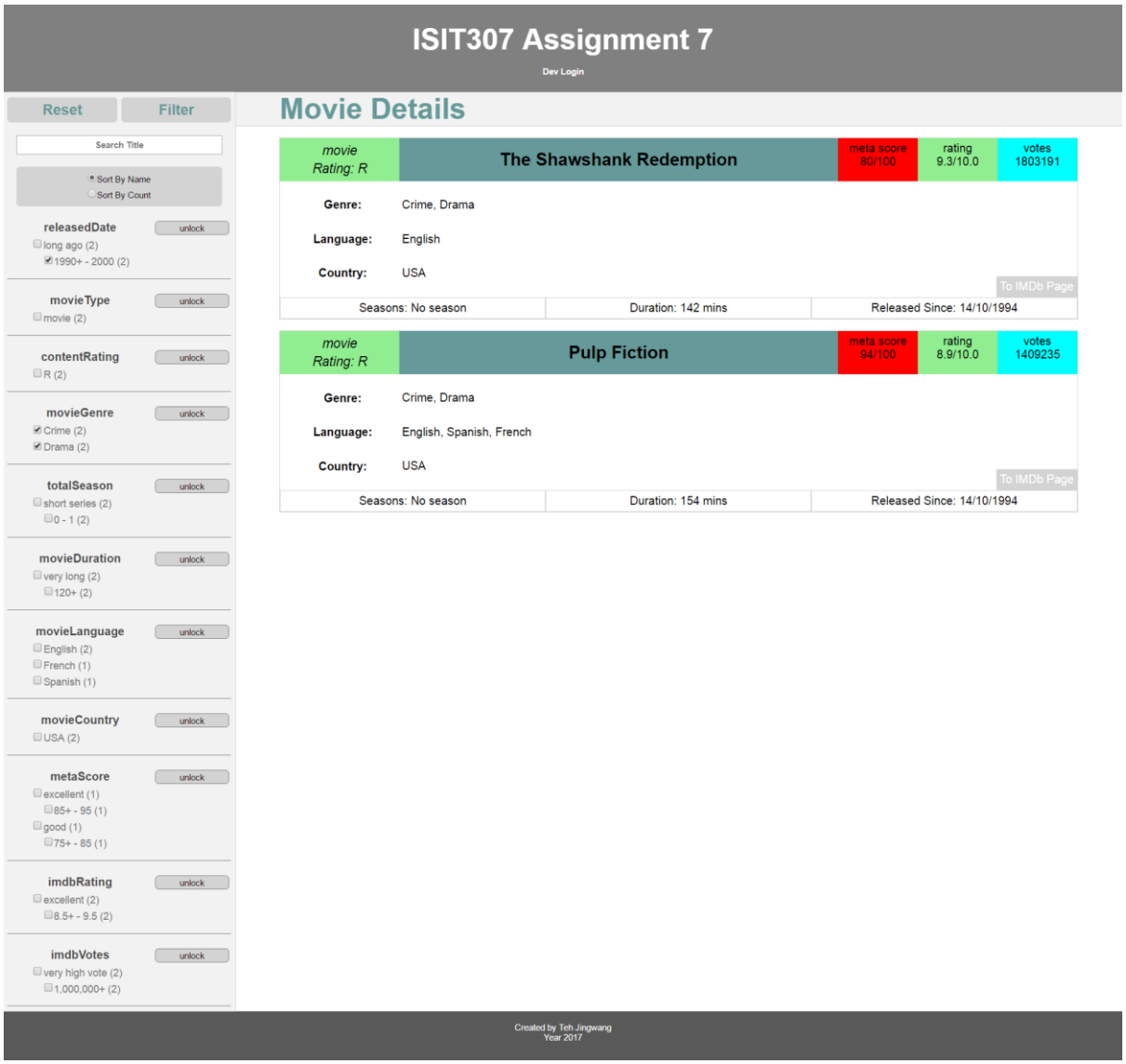
No Filter Set

Filtering with at least 3 filters, filters with 2-level categories, and at least 1 filter with numeric intervals. All records are shown when there are no selections.




0 - 2 (1)
very long (11)
120+ (11)
movieLanguage
unlock
Chinese (1)
English (13)
French (1)
German (1)
Hebrew (1)
Icelandic (1)
Indonesian (1)
Italian (3)
Japanese (9)
Korean (1)
Latin (2)
Malay (1)
Mandarin (2)
OldEnglish (1)
Polish (1)
Portuguese (1)
Quenya (1)
Russian (1)
Sicilian (1)
Sindarin (1)
Spanish (2)
movieCountry
unlock
Australia (1)
Germany (2)
Italy (1)
Japan (9)
NewZealand (1)
Spain (1)
UK (1)
USA (12)
WestGermany (1)
metaScore
unlock
excellent (7)
85+ - 95 (5)
95+ - 100 (2)
good (7)
65+ - 75 (1)
75+ - 85 (6)
terrible (8)
0 - 10 (8)
imdbRating
unlock
excellent (13)
8.5+ - 9.5 (12)
9.5+ - 10.0 (1)
good (9)
6.5+ - 7.5 (1)
7.5+ - 8.5 (8)
imdbVotes
unlock
average vote (5)
1,000+ - 2,500 (1)
2,500+ - 5,000 (3)
5,000+ - 10,000 (1)
high vote (1)
10,000+ - 25,000 (1)
low vote (1)
500+ - 750 (1)
very high vote (14)
1,000,000+ (5)
100,000+ - 500,000 (5)
500,000+ - 1,000,000 (4)
very low vote (1)
0 - 50 (1)
movie Rating: R
Schindler's List
meta score 93/100
rating 8.9/10.0
votes 925979
Genre: Biography, Drama, History
Language: English, Hebrew, German, Polish
Country: USA
To IMDb Page
Seasons: No season
Duration: 195 mins
Released Since: 04/02/1994
movie Rating: R
Pulp Fiction
meta score 94/100
rating 8.9/10.0
votes 1409235
Genre: Crime, Drama
Language: English, Spanish, French
Country: USA
To IMDb Page
Seasons: No season
Duration: 154 mins
Released Since: 14/10/1994
movie Rating: PG-13
The Lord of the Rings: The Return of the King
meta score 94/100
rating 8.9/10.0
votes 1289043
Genre: Adventure, Drama, Fantasy
Language: English, Quenya, OldEnglish, Sindarin
Country: USA, NewZealand
To IMDb Page
Seasons: No season
Duration: 201 mins
Released Since: 17/12/2003
movie Rating: APPROVED
The Good, the Bad and the Ugly
meta score 90/100
rating 8.9/10.0
votes 532413
Genre: Western
Language: Italian
Country: Italy, Spain, WestGermany, USA
To IMDb Page
Seasons: No season
Duration: 161 mins
Released Since: 29/12/1967
movie Rating: PG
Star Wars: Episode V - The Empire Strikes Back
meta score 81/100
rating 8.8/10.0
votes 897240
Genre: Action, Adventure, Fantasy
Language: English
Country: USA
To IMDb Page
Seasons: No season
Duration: 124 mins
Released Since: 20/06/1980
movie Rating: PG
Zootopia
meta score 78/100
rating 8.1/10.0
votes 291412
Genre: Animation, Adventure, Comedy
Language: English
Country: USA
To IMDb Page
Seasons: No season
Duration: 108 mins
Released Since: 04/03/2016
movie Rating: PG
Your Name
meta score 79/100
rating 8.6/10.0
votes 23903
Genre: Animation, Drama, Fantasy
Language: Japanese, Chinese
Country: Japan
To IMDb Page
Seasons: No season
Duration: 106 mins
Released Since: 07/04/2017

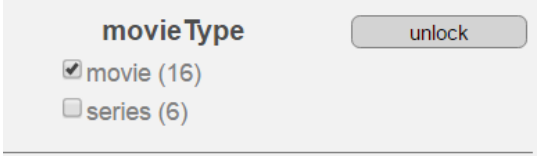
Intersection of the filters controls the observed collection, and clicking reset shows back all the items.



Progressive Query

Single-Step Query

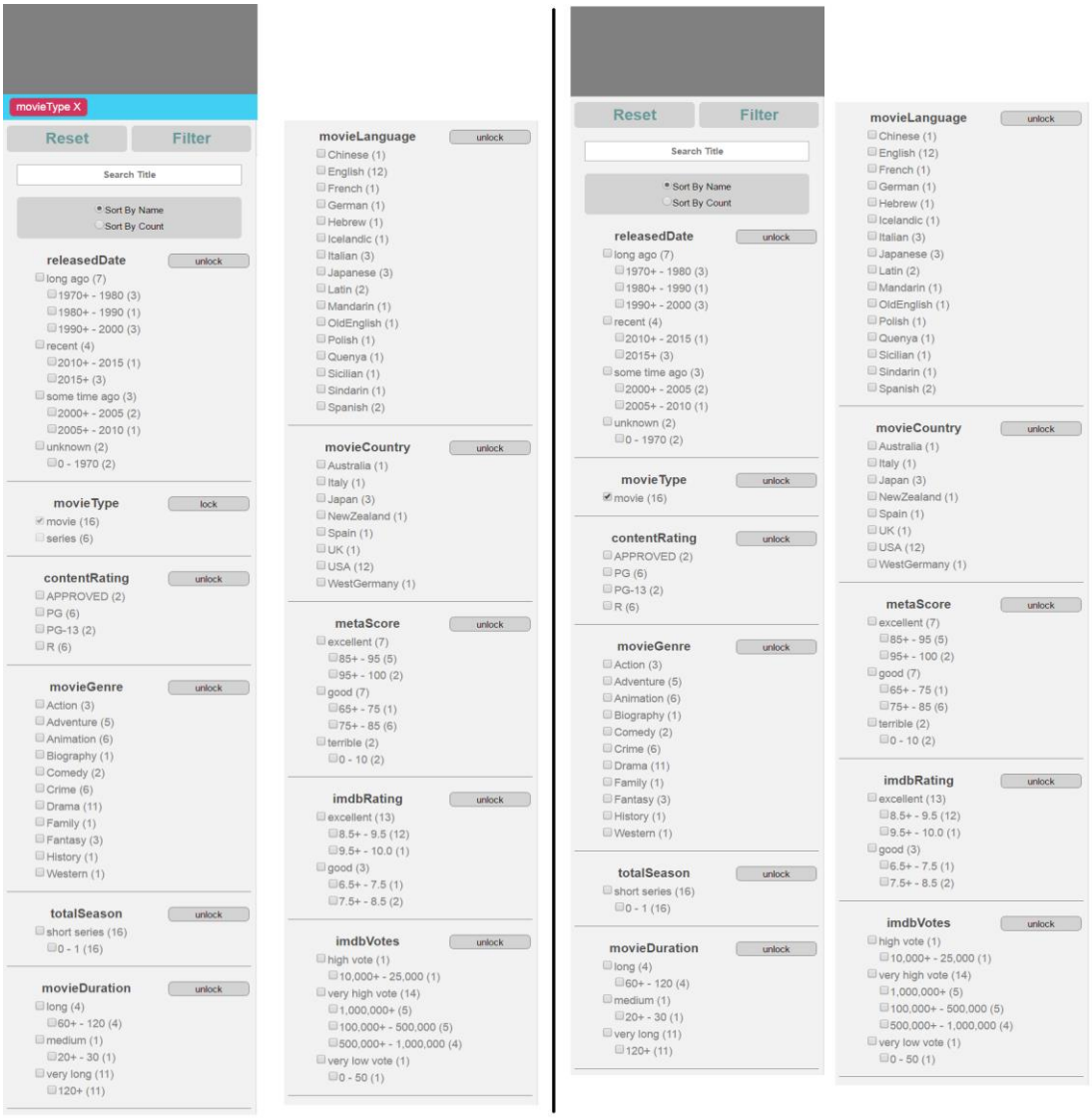
The first single query step example involves selecting movie that currently has a total of 16 records. Progressive Query can be done by locking a specific facet, or by clicking the filter button.



The total counts after a single query step for all facets (except the locked facet) amounts to a number equal to 16 as there are a total of 16 movie records that met the filter selection.

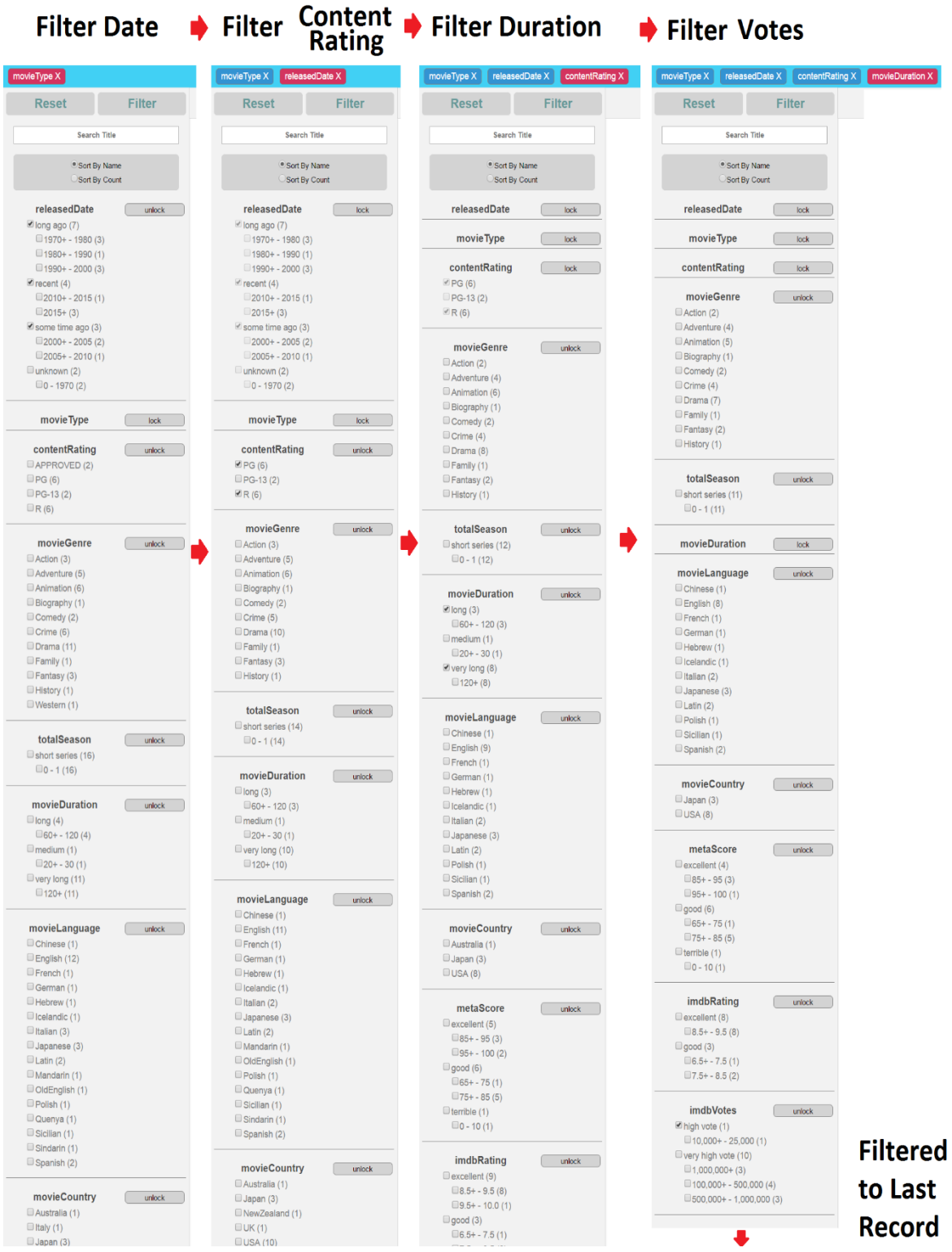
Note: A movie record can consist of multiple movie genre, language, and countries, therefore, they do not amount to the same number of counts as other facets.

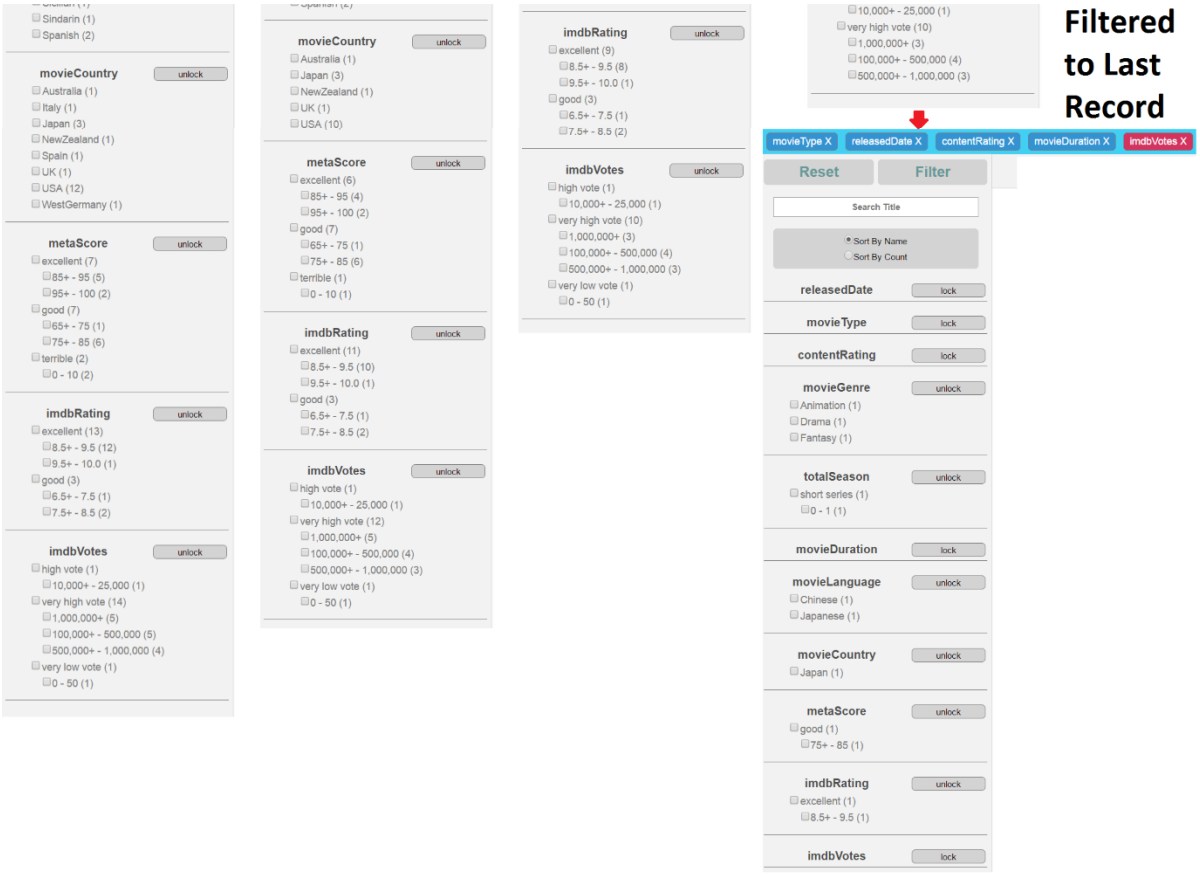
Locking a specific facet (left), clicking filter button (right).



After selecting the movie type, multistep query of narrowing count values can be continued as displayed in the images below, where the user choices are also preserved.

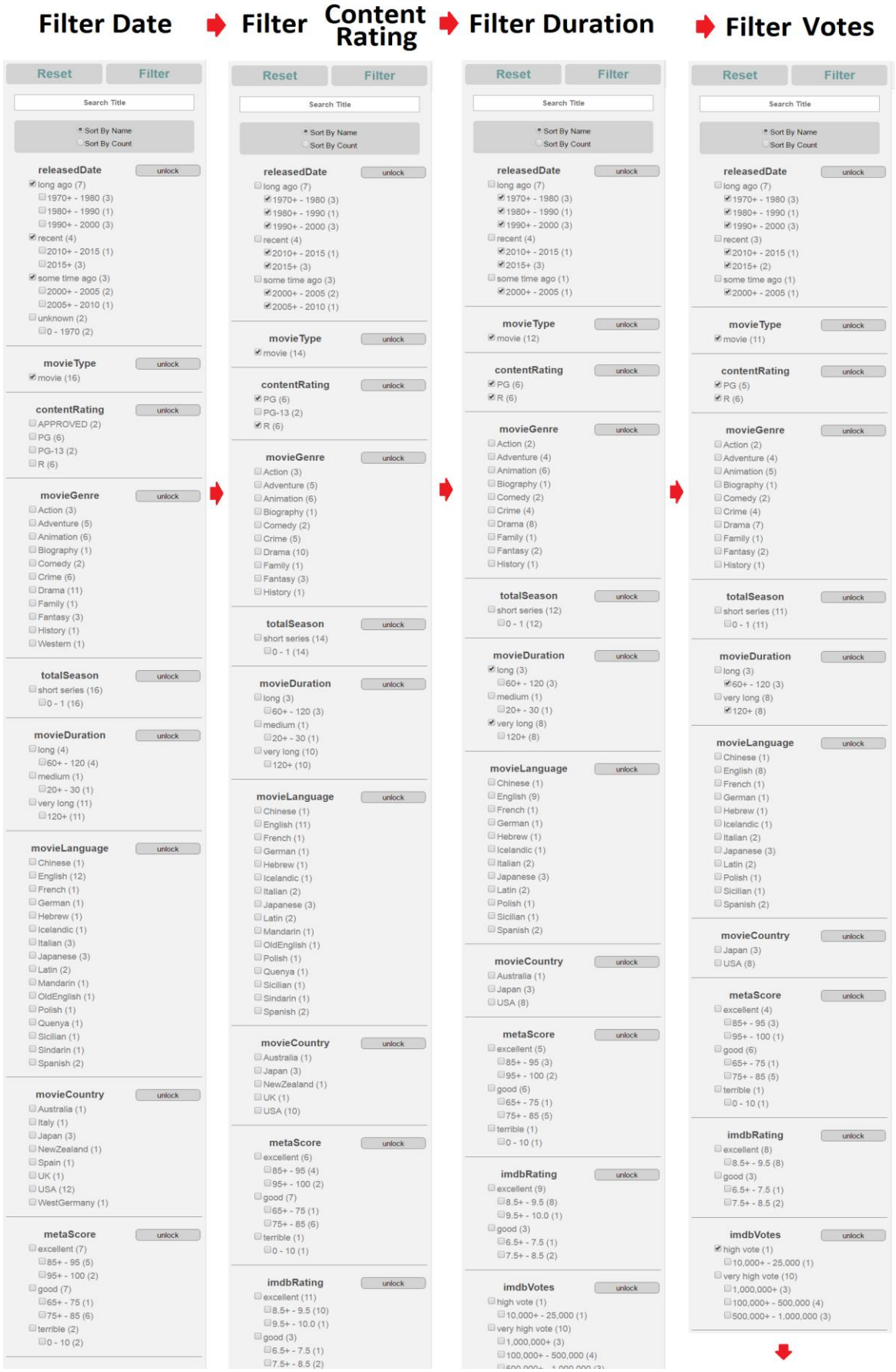
Multi-Step Query (Locking Design)

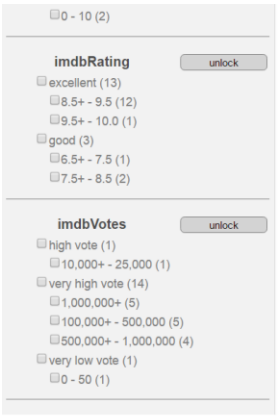


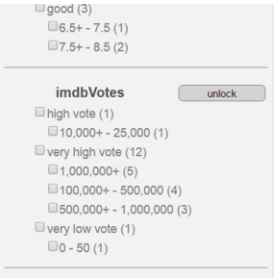


Filtered to Last Record

Multistep query (Filter across Multiple Form Submission Design)







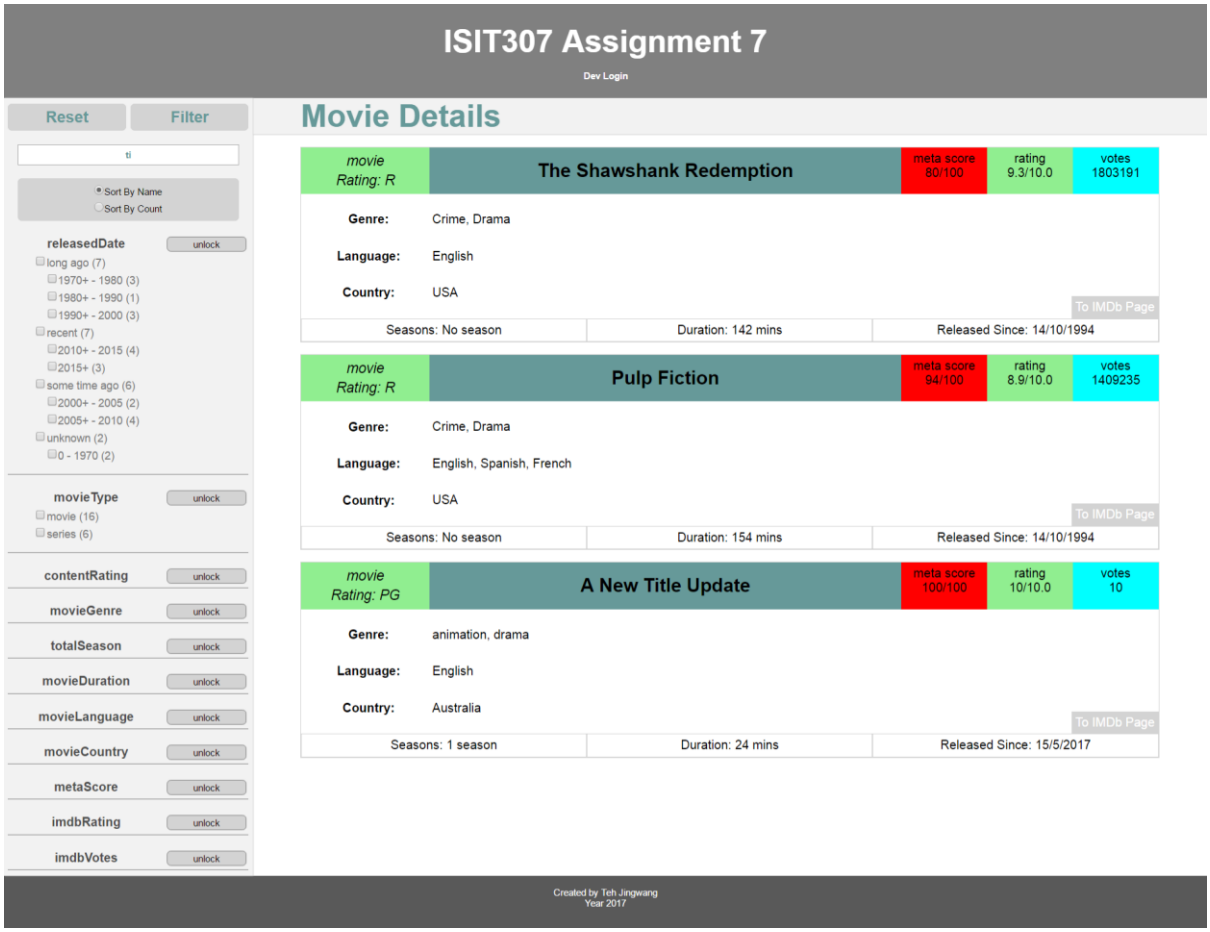
Filtered to Last Record

Keyword Search Auto-complete

Keyword search provides auto-complete for searching based on movie title.

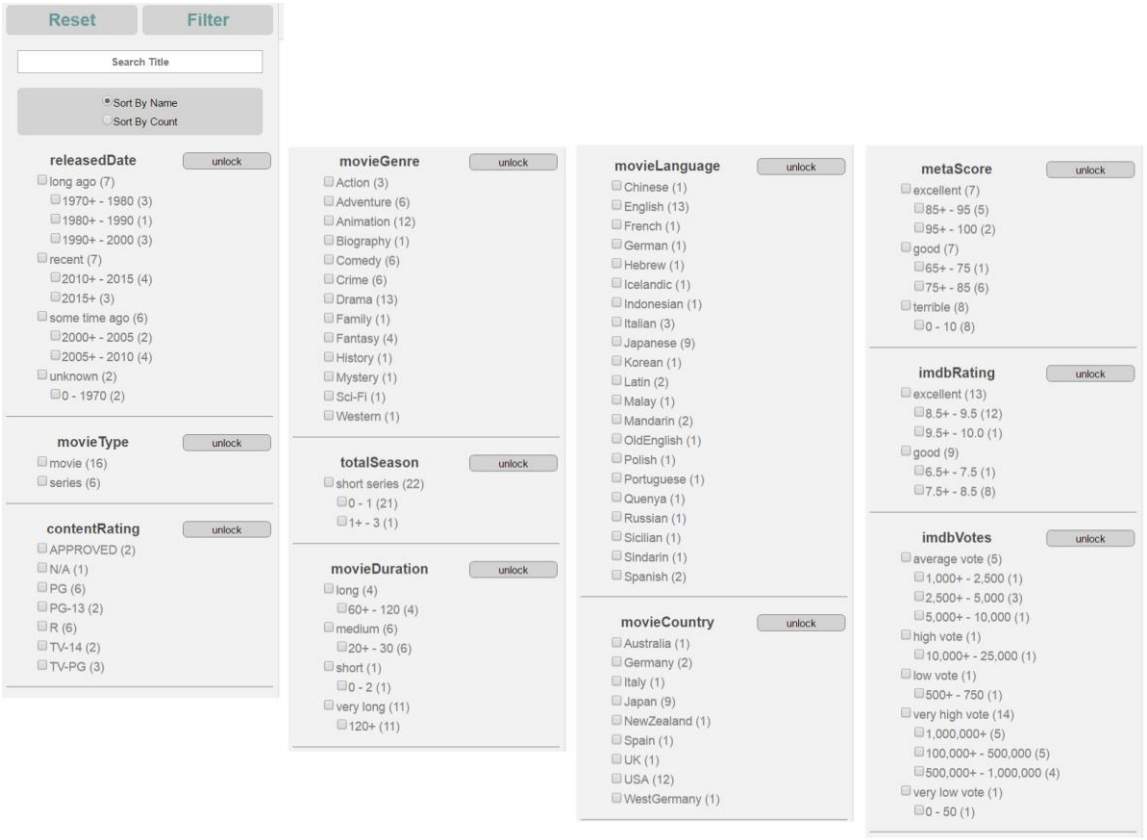
Auto-complete is also filtered based on current filter selections.

Keyword filter is applied along with filter selections when filter button is pressed.

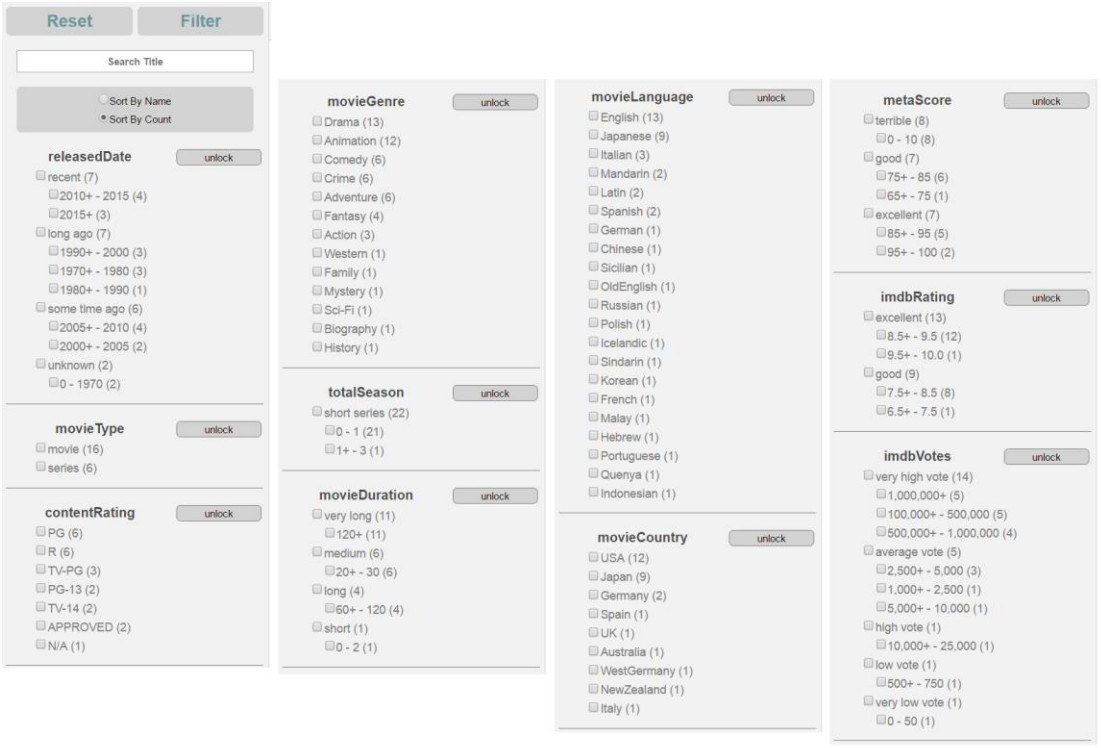


Filter Ordering

Change of Facet Category Order (Default Ordering – Sort by Name/Alphabetical Order)



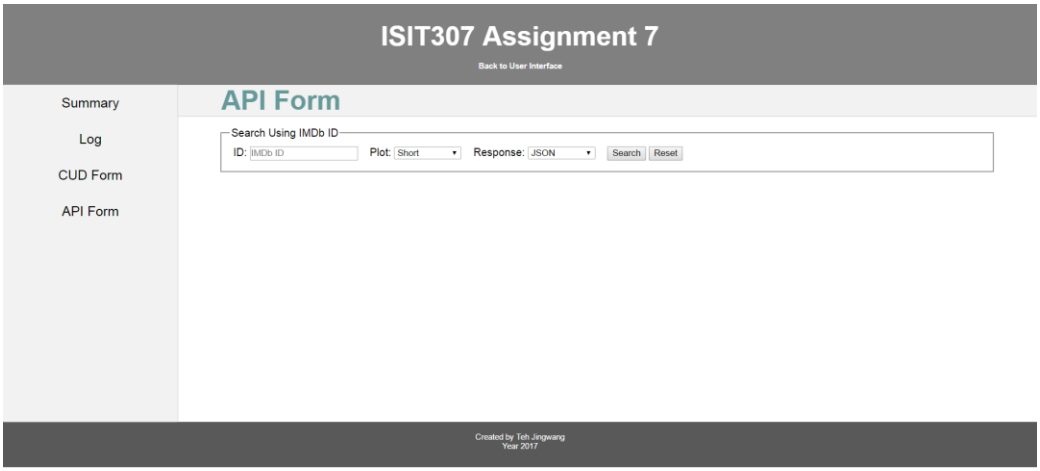
Change of Facet Category Order (Default Ordering – Sort by Count)



API Form (Extra)

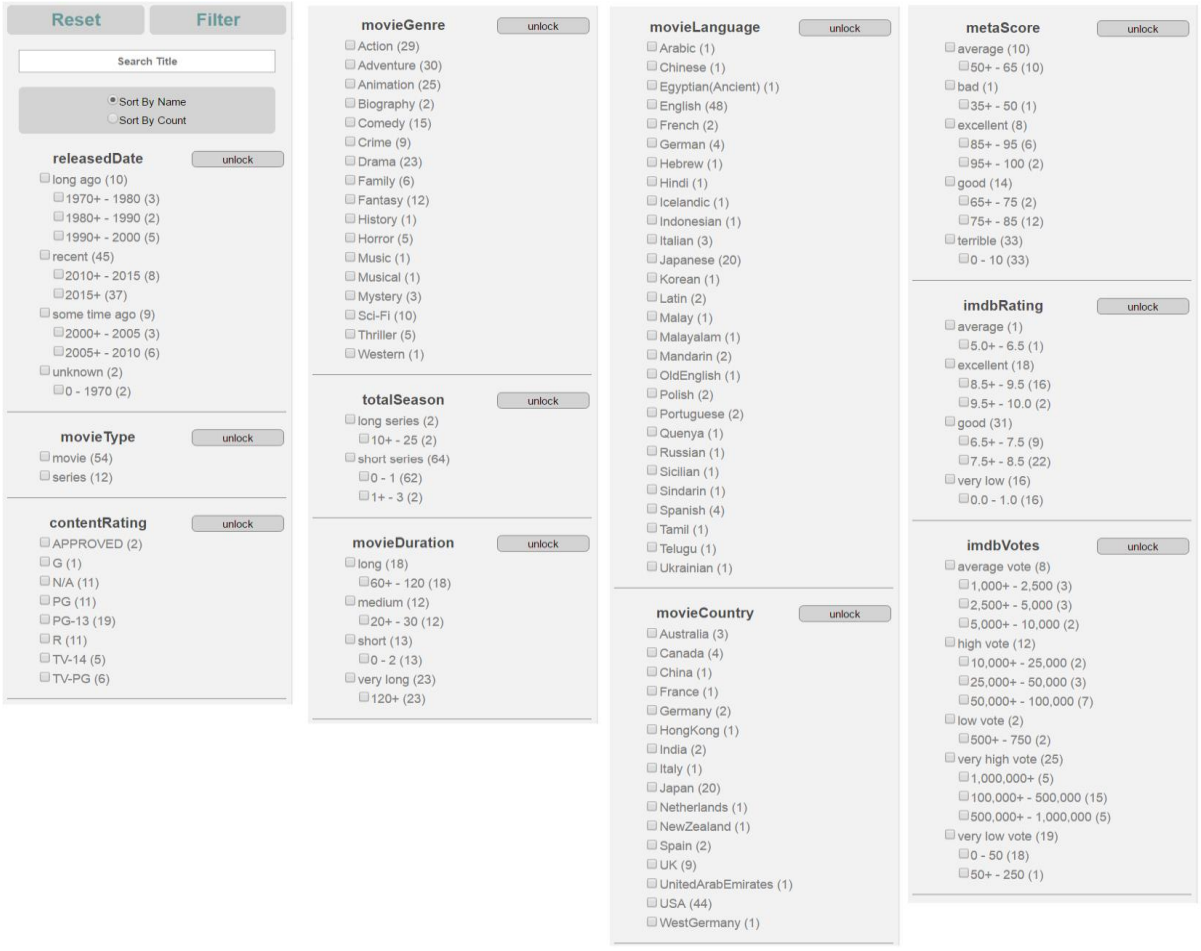
A creation form that gets data from OMDb API by searching using IMDb ID.

The form will directly insert data into database using AJAX.



At Least 50 Records

The counts in each facet is at least 66 for 66 records added, as shown in the image below.



Part 2

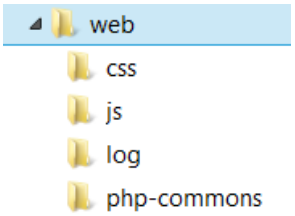
Configuration Setup (XAMPP)

The XAMPP configuration that was used is the default configuration:

- Port Number: 80 (default)
- Document Root & Directory: *<<Project Folder Directory>>*
    - *Index.php* in the project folder will redirect user to the user page.

Design Overview of Code

Folder Structure



The web application code in the project folder is divided into 4 folders and several php files.

The folders include:

- CSS Folder – Stores common **CSS** files for layout, and both CSS files for user and admin interface.
- JS Folder – Stores **JavaScript** files for interface interactivity, and JavaScript functions that also includes all AJAX implementations in the functions.
- Log Folder – Stores the **log text file** for logging record creation/update/delete.
- Php-commons – Stores all common PHP files, such as:
    - (database.php) **database** connection variables
    - (constants.php) global **variables** for extensibility
    - (filters.php) **filter** variable for filter classifications
    - (functions.php) all php **functions**
    - (collection.php) php code that selects all records from database and stores into $collection global variable.

The several php files in the main folder includes:

- Index.php
    - This file receives http request when user accesses the application through the root URL, where it will then **redirect** to the user interface.
- Webpage_user.php
    - This file provides the **layout** for the user interface that will wrap around the user interface content.
- Webpage_user_content.php

    - This file provides **content** for the **user interface**, such as filters and displaying results.
- Webpage_admin.php
    - Similar to webpage_user.php, this file also provides **layout** for wrapping content, but in this case, it is for the admin content.
- Webpage_admin_content.php
    - This file provides **content** for the **admin interface**, such as analytics, logging, creation/update/delete forms, and API form.
- User_progressiveFacetSearch.php
    - This file responds to AJAX call from JavaScript for sorting and progressive filtering.
- Other php files:
    - Dev_downloadLog.php
        - Allows downloading of log text file
    - Dev_recordCreate_OMDbAPI.php
        - Inserts record into database through API form
    - Dev_recordInfo_Form
        - Returns a record based on the imdbRefID specified in update and delete forms
    - Dev_recordRefIDList_Form.php
        - Returns a list of RefID for auto complete
    - Dev_recordTitleList_Form.php
        - Returns a list of Title for auto complete

Code Structure

General Structure and Layout

The structure of the all php code that was written revolves around using variables set in database.php constants.php, and filters.php to dynamically generate code related to database access and facets, which is most of the code written.

On the other hand, JavaScript variables and functions are written in script.js (for user interface related code), and scriptCommons.js (for admin interface related code and generalized javascript code).

All Php code that responds to AJAX calls from JavaScript functions are coded in its own file located in the project root folder, and all PHP functions are written in functions.php.

HTML5 code are integrated with PHP code for generating dynamic content as well as serving as a base for extending layouts. Layouts are stored in webpage_user.php and webpage_admin.php, where the content files, webpage_user_content.php and webpage_admin_content.php, are to be included in.

Interface Structure

The structure for HTML5 code is divided into header, footer, top section, left section, and right section) that are wrapped by the div, *page_container*. Left section and right section divs are wrapped by the div, *content*.

Header and footer sections are located in the layout files, whereas the top section, left section, and right section are located in the content files.

PHP Code Structure (interface)

In the user interface, variables from constants.php (eg. facetInfoArray) is used to display both filters and results by using for loop. The variable allows proper display of facets and results as it also contains boolean variables that specify whether the facet has category-levels, whether it is comma-delimited such as genre, language, and country, and whether it is a date.

The Php code in webpage_user_content.php serves to aggregate and display the filters, receives and processes post request from user, and to display results to the user.

In the admin interface, dbInfoArray from constants.php is used as this variable provides information on the database column names, booleans for validating insert and update forms, as well as other admin-related processes, and also to generate dynamic variable names.

The Php code in webpage_admin_content.php serves to process create/update/delete form submissions, generate dynamic create/update/delete forms, read and display log details, and provide log analytics.

PHP Code Structure (php code in php-commons folder)

In database.php, database connection variables (host, username, password, and database) and log-related directory are stored.

In constants.php, global variables (facetInfoArray, dbInfoArray, and etc) are used for extensibility and ease-of-use when coding in order to quickly access a certain string.

In filters.php, filterClassification variable provides information for categorizing values into intervals as needed in admin form submissions.

In functions.php, all php functions used for the web application is stored here, including functions that aggregates data from database as well as functions that provides a list of records from database that matches user selection, and other miscellaneous functions like sanitize_input, filter_classificationToInterval, filter_valueToInterval, and etc.

As for collection.php, including this file will simply run the php code within in order to select all records from the database and stores it in a global variable, collection, which can be accessed later throughout the file.

PHP Code Structure (Php Files involved in AJAX calls)

In user_progressiveFacetSearch.php, user selections and sort order is received, which is used to do progressive query and also to sort filters. It returns the sorted filter and collection back to the JavaScript function.

Other php files are used to handle admin-side functionality:

- dev_downloadLog.php sends back the log text file
- dev_recordCreate_OMDbAPI.php inserts new record into database from OMDbAPI
- dev_recordInfo_Form.php gets specific record from database based on imdbRefID.
- dev_recordRefIDList_Form.php gets list of imdbRefID from database for auto-complete.
- dev_recordTitleList_Form.php gets list of movie titles from database that matches current user selection for auto-complete.

When handling user selections from JavaScript functions through AJAX calls, the code will first standardize the array as categories and intervals are stored in different arrays in the form before any further processing.

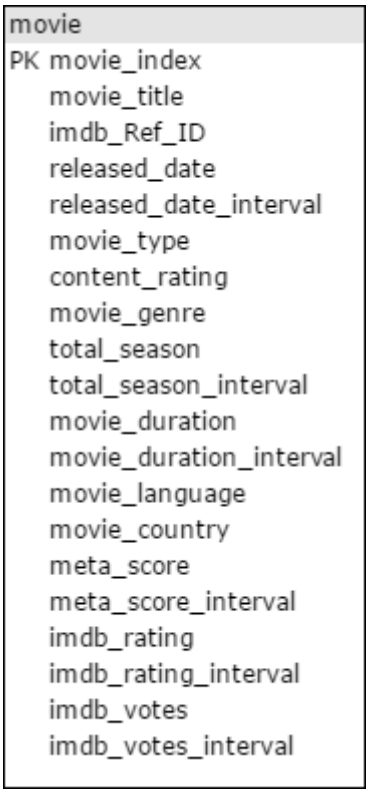
When returning a sorted array from php back to JavaScript function, the code will format the array into a formatted string separated by symbols according to the comments in the code, in order to avoid the JSON object from sorting it again according to array property names.

Design Overview of DB

Movie_index is auto-incremented. Released_date, total_season, movie_duration, meta_score, imdb_rating, and imdb_votes have intervals that are stored in its equivalent interval columns. Imdb_Ref_ID is unique.

When a new record is created, all fields except the movie_index and interval columns will be added into database. After that, all interval strings will be generated through the php function, db_updateIntervals, based on the added columns that have intervals, and then added into the same record according to the imdb_Ref_ID.

Updated intervals can be generated and stored into its respective columns by using the same function again.



Detailed PHP, HTML5, CSS, JavaScript Code

Detailed PHP and HTML5 Code (Layout, User & Admin Content)

webpage_user.php

```
<!DOCTYPE html>
<html>
	<head>
		<meta charset="UTF-8"/>
		<title>assignment 7 User</title>

		<!-- ----------CSS---------- -->
		<link rel="stylesheet" type="text/css" href="css/cssCommons.css">
		<link rel="stylesheet" type="text/css" href="css/cssUserPage.css">

		<!-- ----------JavaScript---------- -->
		<script type="text/javascript" src="js/scriptCommons.js"></script>
		<script type="text/javascript" src="js/script.js"></script>

		<!-- ----------PHP Includes (Commons) ---------- -->
		<!-- Database -->
		<?php include 'php-commons/database.php'; ?>
		<!-- Collection of all records from database -->
		<?php include 'php-commons/collection.php'; ?>

		<!-- Variables -->
		<?php include 'php-commons/constants.php'; ?>
		<!-- Filters -->
		<?php include 'php-commons/filters.php'; ?>

		<!-- Functions -->
		<?php include 'php-commons/functions.php'; ?>
	</head>
	<body>
		<!-- ------------------------------Page Container (Wrapper)------------------------------ -->
		<div class="page_container">

			<!-- ------------------------------Header Section------------------------------ -->
			<div class="header">
				<div>
					<div class="header_content">
						<!-- Header Title -->
						<div class="header_title"><a href="/">ISIT307
Assignment 7</a></div>

						<!-- Header Shortcuts -->
						<div class="header_shortcuts"><a
href="webpage_dev.php">Dev Login</a></div>
					</div>
				</div>
			</div>

			<!-- ------------------------------Content Section------------------------------ -->
			<?php
				include 'webpage_user_content.php';
			?>

			<!-- ------------------------------Footer Section------------------------------ -->
			<div class="footer">
				<div>
					<div class="footer_content">
						<!-- Footer Copyright -->
						<div class="footer_copyright">
							<p>Created by Teh Jingwang</p>
							<p>Year 2017</p>
```

```
                                                    </div>
                                                </div>
                                            </div>
                                        </div>

                    </div>
                    <!-- ------------------------------End of Page Container----------------------------- -->
            </body>
</html>
```

webpage_dev.php

```
<!DOCTYPE html>
<html>
        <head>
                <meta charset="UTF-8"/>
                <title>assignment 7 Dev</title>

                <!-- ----------CSS---------- -->
                <link rel="stylesheet" type="text/css" href="css/cssCommons.css">
                <link rel="stylesheet" type="text/css" href="css/cssAdminPage.css">

                <!-- ----------JavaScript---------- -->
                <script type="text/javascript" src="js/scriptCommons.js"></script>
                <script type="text/javascript" src="js/script.js"></script>

                <!-- ----------PHP Includes (Commons) ---------- -->
                <!-- Database -->
                <?php include 'php-commons/database.php'; ?>
                <!-- Collection of all records from database -->
                <?php include 'php-commons/collection.php'; ?>

                <!-- Variables -->
                <?php include 'php-commons/constants.php'; ?>
                <!-- Filters -->
                <?php include 'php-commons/filters.php'; ?>

                <!-- Functions -->
                <?php include 'php-commons/functions.php'; ?>
        </head>
        <body>
                <!-- ------------------------------Page Container (Wrapper)----------------------------- -->
                <div class="page_container">

                                        <!-- ------------------------------Header Section----------------------------- -->
                                        <div class="header">
                                                <div>
                                                        <div class="header_content">
                                                                <!-- Header Title -->
                                                                <div class="header_title"><a href="/">ISIT307
Assignment 7</a></div>

                                                                <!-- Header Shortcuts -->
                                                                <div class="header_shortcuts"><a
href="webpage_user.php">Back to User Interface</a></div>
                                                        </div>
                                                </div>
                                        </div>

                                        <!-- ------------------------------Content Section----------------------------- -->
                                        <?php
                                                include 'webpage_dev_content.php';
                                        ?>

                                        <!-- ------------------------------Footer Section----------------------------- -->
                                        <div class="footer">
                                                <div>
```

```
                                                <div class="footer_content">
                                                        <!-- Footer Copyright -->
                                                        <div class="footer_copyright">
                                                                <p>Created by Teh Jingwang</p>
                                                                <p>Year 2017</p>
                                                        </div>
                                                </div>
                                        </div>
                                </div>

                        </div>
                        <!-- ------------------------------End of Page Container---------------------------- -->
        </body>
</html>
```

webpage_user_content.php

```
<!-- Facet Filter : Step 1 - aggregate unique -->
<?php
        /*
         * aggregate each facet and get unique value-count associative array for display
         */
        $facetUnique = array();
        foreach ($facetInfoArray as $facetInfo)
        {       $facetUnique[] = db_uniqueFacet_all($facetInfo, true);      }

        $sortOrder = '';
        $ac_search_Val = '';
?>

<!-- Facet Filter : Step 2 - User Input Criteria -->
<?php
        /*
         * Get user-submitted faceted filter selections
         * to obtain an array of matching indexes for each facet
         */
        if ($_SERVER['REQUEST_METHOD'] === 'POST')
        {
                // prepare array for making sql query to get results based on user selections
                // & get corresponding intervals for selected categories
                $userSelections = array();
                foreach ($facetInfoArray as $facetInfo)
                {       // has no category (select)
                        if (!$facetInfo[2]){
                                if (array_key_exists($facetInfo[0].'Select', $_POST))
                                { $userSelections[$facetInfo[1]] = $_POST[$facetInfo[0].'Select']; }
                        }
                        // has category (super & sub)
                        else if ($facetInfo[2]){
                                if (array_key_exists($facetInfo[0].'Select_super', $_POST) ||
array_key_exists($facetInfo[0].'Select_sub', $_POST))
                                {
                                        if (array_key_exists($facetInfo[0].'Select_sub', $_POST))
                                        { $userSelections[$facetInfo[1]] = $_POST[$facetInfo[0].'Select_sub']; }
                                        if (array_key_exists($facetInfo[0].'Select_super', $_POST))
                                        {       // convert category(super) to its corresponding intervals(sub)
                                                $super = $_POST[$facetInfo[0].'Select_super'];
                                                foreach ($super as $category)
                                                {
                                                        $super = filter_classificationToInterval($facetInfo[0],
$category);

                                                        foreach ($super as $interval){
$userSelections[$facetInfo[1]][] = $interval; }
                                                }
                                        }
                                }
                        }
```

```
		}

		$ac_search_Val = (inputNotEmpty($_POST['ac_search']))? sanitize_input($_POST['ac_search']):null;

		// get user selected results from database
		$queryMatch = "";
		$table_resultDisplay = db_matchResults($userSelections, $ac_search_Val);

		$sortOrder = (!empty($_POST['sortOrder']))? sanitize_input($_POST['sortOrder']):null;
		if (!empty($userSelections))
		{	$facetUnique = array();
			/* unique facet on full collection *
			if (empty($sortOrder) || $sortOrder != 'count'){
				foreach ($facetInfoArray as $facetInfo)
				{	$facetUnique[] = db_uniqueFacet_all($facetInfo, true);	}
			} else {
				foreach ($facetInfoArray as $facetInfo)
				{	$facetUnique[] = db_uniqueFacet_all($facetInfo, false);	}
			}
			/* unique facet on filtered collection */
			if (empty($sortOrder) || $sortOrder != 'count'){
				foreach ($facetInfoArray as $facetInfo)
				{	$facetUnique[] = db_matchResults_unique($facetInfo, $queryMatch,
true);	}
			} else {
				foreach ($facetInfoArray as $facetInfo)
				{	$facetUnique[] = db_matchResults_unique($facetInfo, $queryMatch,
false);	}
			}
		}
	}
?>

<!-- ------------------------------Top Content Section------------------------------ -->
<div class="top_content"><div id="lockedSequenceDisplay"></div></div>

<!-- ------------------------------Content Section------------------------------ -->
<div class="content">
	<!-- ------------------------------Left Content Section------------------------------ -->
	<div class="left_content pullDiv" onmouseout="setPushPull('left_content')">

		<!-- Faceted Filters Form -->
		<div class="left_content_facetedFilter">
			<form id="facetedFilterForm" action="<?php echo
htmlspecialchars($_SERVER['PHP_SELF']); ?>" method="post">

				<!-- Buttons -->
				<div class="btns_container">
					<div><a class="btns resetLink" href="<?php echo
htmlspecialchars($_SERVER['PHP_SELF']); ?>">Reset</a></div>
					<div><input class="btns submitBtn" type="submit"
value="Filter"/></div>
					<input type="hidden" name="filter_form" value="filter_form"/>
				</div>

				<!-- Auto Complete Search -->
				<div class="left_content_facetedSearch">
					<div>
						<input type="text" id="ac_search" name="ac_search"
							placeholder="Search Title"
value="<?=$ac_search_Val?>"

onkeyup="getData_FormAutoComplete_Filtered('ac_search','autocomplete_s')" />
						<div class="autocomplete_popup" id="autocomplete_s"
							onmouseleave="popup_disappear('autocomplete_s')"
onclick="popup_disappear('autocomplete_s')"></div>
					</div>
```

```
                                        </div>

                                        <!-- Sort -->
                                        <div class="left_content_facetedSort">
                                                <div>
                                                        <label><input type="radio" id="btn_sortUnique"
name="sortOrder" value="name"
                                                                                <?= (empty($sortOrder) ||
$sortOrder != 'count')? "checked='checked'":"?>

onclick="sortUniqueOrCount(true)"/>Sort By Name</label><br/>
                                                        <label><input type="radio" id="btn_sortCount"
name="sortOrder" value="count"
                                                                                <?= ($sortOrder == 'count')?
"checked='checked'":"?>

onclick="sortUniqueOrCount(false)"/>Sort By Count</label>
                                                </div>
                                        </div>

                                        <!-- Filters -->
                                        <div id="facetedFilterForm_Content"><?php
                                                //  FacetUniqueInfo requires: unique values/intervals(with categories),
facet name, interval name, and bool hasCategory
                                                $facetUniqueInfoArray = array();
                                                for ($i=0; $i<count($facetInfoArray); $i++)
                                                { $facetUniqueInfoArray[] = array($facetUnique[$i],
$facetInfoArray[$i][0], $facetInfoArray[$i][1], $facetInfoArray[$i][2]); }

                                                // output facets
                                                foreach ($facetUniqueInfoArray as $facet_info)
                                                {
                                                        $facetSection_sectionID =
'facetedFilterForm_'.$facet_info[1].'_Section';
                                                        $facetSection_labelID =
'assignment6Form_'.$facet_info[1].'_Label';
                                                        $facetSection_labelText = $facet_info[1];

                                                        // Section
                                                        echo "<div class='facetedFilterFormSection extendDiv'
                                                                        id='$facetSection_sectionID'

        onclick='clickDisplayID(\"$facetSection_sectionID\")'>";
                                                                /* **************************Section
Label************************** */
                                                                echo "<div class='facetedFilterFormSection_Label'

        id='$facetSection_labelID'>";
                                                                        // Label
                                                                        echo
"<div><label>$facetSection_labelText</label></div>";
                                                                        // LockText
                                                                        echo "<div><span class='lockText'
onclick='ajaxFacetFilter(\"$facetSection_sectionID\")'>

        $lockText_default

        </span></div>";
                                                                echo "</div>";
                                                                /* **************************Section
Input************************** */
                                                                echo "<div class='facetedFilterFormSection_Input'>";
                                                                        if(!empty($facet_info[0])){
                                                                                // if has categories
                                                                                if ($facet_info[3])
                                                                                {               // Set text strings
```

```
        $facetSection_inputClass_super      = 'facetedFilterFormSection_inputSuper';
        $facetSection_inputClass_sub        = 'facetedFilterFormSection_inputSub';
        $facetSection_inputName_super       = $facet_info[1].'Select_super[]';
        $facetSection_inputName_sub         = $facet_info[1].'Select_sub[]';
                                                                                    foreach ($facet_info[0]
as $classification => $intervalGroup)
                                                                                    {
                                                                                        // Additional
text strings for classification
        $facetSection_inputClass_super_classification = $facet_info[1].'_super_'.$classification;
        $facetSection_inputClass_sub_classification = $facet_info[1].'_sub_'.$classification;
                                                                                        // Get total
number of unique_count (category count) from all intervals in each interval group
        $categoryCount = 0;
                                                                                        foreach
($intervalGroup as $categoryCount_interval => $categoryCount_count)
                                                                                        {
$categoryCount += $categoryCount_count; }
                                                                                        // Display
current category/classification
                                                                                        echo
"<label>";
                                                                                            echo
"<input type='checkbox'
                                class='$facetSection_inputClass_super $facetSection_inputClass_super_classification'
                                name='$facetSection_inputName_super'
                                value='$classification'
                                onclick='toggleFacetCheck(\"$facet_info[1]\", \"super\", \"$classification\")' />";
                                                                                            echo
$classification ." (". $categoryCount .") ";
                                                                                        echo
"</label><br/>";
                                                                                        // Display all
intervals for current category/classification
                                                                                        foreach
($intervalGroup as $interval => $count)
                                                                                        {
                                                                                            echo
"<label>";
        echo "<input type='checkbox'
                                    class='$facetSection_inputClass_sub $facetSection_inputClass_sub_classification'
                                    name='$facetSection_inputName_sub'
                                    value='$interval'
                                    onclick='toggleFacetCheck(\"$facet_info[1]\", \"sub\", \"$classification\")' ";
                                    if (!empty($userSelections[ $facet_info[2] ])){
                                        if (in_array($interval, $userSelections[ $facet_info[2] ]))
```

```
                                        echo "checked='checked' ";
                                }
        echo "/>";
        echo $interval ." (". $count .") ";
                                                                                                                                        echo
"</label><br/>";
                                                                                                                                                }
                                                                                                                                        }
                                                                                                                                }
                                                                                                                                // if no categories
                                                                                                                                else
                                                                                                                                {       // set text strings
        $facetSection_inputClass_select = 'facetedFilterFormSection_inputSuper';
        $facetSection_inputName_select          = $facet_info[1].'Select[]';
                                                                                                                                        // Display all intervals
                                                                                                                                        foreach ($facet_info[0]
as $unique_value => $unique_count)
                                                                                                                                        {
                                                                                                                                                echo
"<label>";
                                                                                                                                        echo
"<input type='checkbox'
                                 class='$facetSection_inputClass_select'
                                 name='$facetSection_inputName_select'
                                 value='$unique_value' ";
                                if (!empty($userSelections[ $facet_info[2] ])){
                                        if (in_array($unique_value, $userSelections[ $facet_info[2] ]))
                                                echo "checked='checked' ";
                                }
                                                                                                                                        echo
"/>";
                                                                                                                                        echo
$unique_value ." (". $unique_count .") ";
                                                                                                                                                echo
"</label><br/>";
                                                                                                                                        }
                                                                                                                                }
                                                                                                                        } else { echo
"<label><span>Nothing</span></label><br/>"; }
                                                                                                        echo "</div>";
                                                                                                echo "</div>";
                                                                                        } // End of Section
                                        ?></div>
                                </form>
                                <script>
                                        // JavaScript : Function to add event listeners (mouse over, out, and click onto
lockTextElements)
                                        var lockTextElements = document.getElementsByClassName("lockText");
                                        for (var i = 0; i < lockTextElements.length; i++)
                                        {       // mouse over
                                                lockTextElements[i].addEventListener("mouseover", function(event){
                                                        var lockTextElement = event.target || event.srcElement;
                                                        mouseOverLock(lockTextElement);
```

```
                                        });
                                        // mouse out
                                        lockTextElements[i].addEventListener("mouseout", function(event){
                                                var lockTextElement = event.target || event.srcElement;
                                                mouseOutLock(lockTextElement);
                                        });
                                }
                                // JavaScript : Function to change text when mouse over lockTextElement
                                function mouseOverLock(lockTextElement)
                                {
                                        if (lockTextElement.innerHTML == text_unlock)
                                        { lockTextElement.innerHTML = text_toLock; }
                                        else if (lockTextElement.innerHTML == text_lock)
                                        { lockTextElement.innerHTML = text_toUnlock; }
                                        // default to "lock?" if text changed by user
                                        else { lockTextElement.innerHTML = text_toLock; }
                                }
                                // JavaScript : Function to change text when mouse out lockTextElement
                                function mouseOutLock(lockTextElement)
                                {
                                        if (lockTextElement.innerHTML == text_toLock)
                                        { lockTextElement.innerHTML = text_unlock; }
                                        else if (lockTextElement.innerHTML == text_toUnlock)
                                        { lockTextElement.innerHTML = text_lock; }
                                        // default back to "unlock" text if text changed by user
                                        else if (lockTextElement.innerHTML != text_lock &&
lockTextElement.innerHTML != text_unlock)
                                        { lockTextElement.innerHTML = text_unlock; }
                                }
                        </script>
                </div>

        </div>
        <!-- End of Left Content Section -->

        <!-- ------------------------------Right Content Section------------------------------ -->
        <div class="right_content">

                <!-- Movie Results according to Faceted Filters -->
                <div class="right_content_resultDisplay">
                        <div class="right_content_title"><h1>Movie Details</h1></div>

                        <div class='right_content_resultDisplay_content' id="results_section">
                        <?php
                                // if there is no records in movie collection
                                if (empty($collection))
                                { echo "<div>No Movie Content</div>"; }
                                // if user selection filter returns a result
                                else if (!empty($table_resultDisplay))
                                { // display selected results
                                        foreach ($table_resultDisplay as $record)
                                        {       echo "<div>";
                                                        echo "<div>";
                                                                echo
"<div><span>",$record['movieType'],"<br/>Rating: ",$record['contentRating'],"</span></div>";
                                                                echo
"<div><span>",$record['movieTitle'],"</span></div>";
                                                                echo "<div>meta
score<br/>",$record['metaScore'],"/100</div>";
                                                                echo
"<div>rating<br/>",$record['imdbRating'],"/10.0</div>";
                                                                echo
"<div>votes<br/>",$record['imdbVotes'],"</div>";
                                                        echo "</div>";

                                                        echo "<div>";
                                                                echo "<br/><div><span>Genre: </span>";
```

```
                                                $prefix = "";
                                                foreach($record['movieGenre'] as $genre){
                                                        echo (empty($prefix))?"":$prefix;
                                                        echo $genre;
                                                        $prefix = ", ";
                                                }
                                                echo "</div>";

                                                echo "<br/><div><span>Language:
</span>";
                                                $prefix = "";
                                                foreach($record['movieLanguage'] as
$language){
                                                        echo (empty($prefix))?"":$prefix;
                                                        echo $language;
                                                        $prefix = ", ";
                                                }
                                                echo "</div>";

                                                echo "<br/><div><span>Country: </span>";
                                                $prefix = "";
                                                foreach($record['movieCountry'] as
$country){
                                                        echo (empty($prefix))?"":$prefix;
                                                        echo $country;
                                                        $prefix = ", ";
                                                }
                                                echo "</div><br/>";

                                                echo "<a target='_blank'
href='http://www.imdb.com/title/",$record['imdbRefID'],"'><div class='ext_link_imdb'>";
                                                        echo "To IMDb Page";
                                                echo "</div></a>";
                                        echo "</div>";

                                        echo "<div>";
                                                echo "<div>Seasons:
",($record['totalSeason'] == 0)?'No':$record['totalSeason']," season</div>";
                                                echo "<div>Duration:
",$record['movieDuration']," mins</div>";
                                                echo "<div>Released Since:
",$record['releasedDate'],"</div>";
                                        echo "</div>";
                                echo "</div>";
                        }
                }
                // display all movie records if all selection is at default
                else if (empty($userSelections) && empty($ac_search_Val))
                { // display all records
                        foreach ($collection as $record)
                        {       echo "<div>";
                                        echo "<div>";
                                                echo
"<div><span>",$record['movieType'],"<br/>Rating: ",$record['contentRating'],"</span></div>";
                                                echo
"<div><span>",$record['movieTitle'],"</span></div>";
                                                echo "<div>meta
score<br/>",$record['metaScore'],"/100</div>";
                                                echo
"<div>rating<br/>",$record['imdbRating'],"/10.0</div>";
                                                echo
"<div>votes<br/>",$record['imdbVotes'],"</div>";
                                        echo "</div>";

                                        echo "<div>";
                                                echo "<br/><div><span>Genre: </span>";
                                                $prefix = "";
```

```
                                                                    foreach($record['movieGenre'] as $genre){
                                                                        echo (empty($prefix))?"":$prefix;
                                                                        echo $genre;
                                                                        $prefix = ", ";
                                                                    }
                                                                    echo "</div>";

                                                                    echo "<br/><div><span>Language:
</span>";
                                                                    $prefix = "";
                                                                    foreach($record['movieLanguage'] as
$language){
                                                                        echo (empty($prefix))?"":$prefix;
                                                                        echo $language;
                                                                        $prefix = ", ";
                                                                    }
                                                                    echo "</div>";

                                                                    echo "<br/><div><span>Country: </span>";
                                                                    $prefix = "";
                                                                    foreach($record['movieCountry'] as
$country){
                                                                        echo (empty($prefix))?"":$prefix;
                                                                        echo $country;
                                                                        $prefix = ", ";
                                                                    }
                                                                    echo "</div><br/>";

                                                                    echo "<a target='_blank'
href='http://www.imdb.com/title/",$record['imdbRefID'],"'><div class='ext_link_imdb'>";
                                                                        echo "To IMDb Page";
                                                                    echo "</div></a>";
                                                                echo "</div>";

                                                                echo "<div>";
                                                                    echo "<div>Seasons:
",($record['totalSeason'] == 0)?'No':$record['totalSeason']," season</div>";
                                                                    echo "<div>Duration:
",$record['movieDuration']," mins</div>";
                                                                    echo "<div>Released Since:
",$record['releasedDate'],"</div>";
                                                                echo "</div>";
                                                            echo "</div>";
                                                        }
                                                    }
                                                    // if user's selection does not match any record in collection
                                                    else if (empty($table_resultDisplay))
                                                    { echo "<div>No Matching Movie Content</div>"; }
                                                ?>
                                                </div>
                                            </div>
                                            <!-- End of Result Display -->

                                    </div>
                                    <!-- End of Right Content Section -->
                            </div>
                            <!-- End of Content Section -->
```

webpage_admin_content.php

```
<!-- Process Form Submission -->
<?php
	// Initialize Empty Sticky Fields and Error Values
	foreach ($dbInfoArray as $dbInfo){
		// (Creation Form) Sticky Field Variables, Error Variables, isFacetValid Variables
		${'CForm_'.$dbInfo[4].'_Val'} = '';
		${'CForm_'.$dbInfo[4].'_Err'} = '';
		${'CForm_is'.$dbInfo[4].'Valid'} = false;

		// (Update Form) Sticky Field Variables, Error Variables, isFacetValid Variables
		${'UForm_'.$dbInfo[4].'_Val'} = '';
		${'UForm_'.$dbInfo[4].'_Err'} = '';
		${'UForm_is'.$dbInfo[4].'Valid'} = false;
	}
	$UForm_search_Val = $DForm_search_Val = '';

	/*
	 * Handle Form Submission (Creation/Update/Delete)
	 */
	$isSubmittedDataValid = true; // if any input has error, submission becomes invalid (false)
	// Creation Form
	if (array_key_exists("creation_form", $_POST))
	{	$text_formType	= 'CForm_';

		// Sanitize & Validate Input
		foreach ($dbInfoArray as $dbInfo)
		{	$text_inputName = $text_formType.$dbInfo[4];

			/* Sanitize Input */
			$_POST[$text_inputName] = (inputNotEmpty($_POST[$text_inputName]))?
	sanitize_input($_POST[$text_inputName]) : null;

			/*
			 * Validate Input, and
			 * If Invalid:
			 * - Set Error Strings
			 * - Set Boolean for Invalid Field
			 */
			// isEmpty
			if (!inputNotEmpty($_POST[$text_inputName])){
				${$text_formType.$dbInfo[4].'_Err'} = "Please fill in the field for ".$dbInfo[4].".";
				$isSubmittedDataValid = false;
			}
			// isCommaDelimited
			else if ($dbInfo[5]){
				$_POST[$text_inputName] = str_replace(' ', '', $_POST[$text_inputName]);
				// Invalid Comma Delimited Text
				if( in_array("", explode(',', $_POST[$text_inputName])) ){
					${$text_formType.$dbInfo[4].'_Err'} = "Please provide a valid comma
delimited text";
					$isSubmittedDataValid = false;
				} // Limit comma delimited
				else if (count(explode(',', $_POST[$text_inputName])) > 5){
					${$text_formType.$dbInfo[4].'_Err'} = "Please provide less than or equal
to 5 ".$dbInfo[4];
					$isSubmittedDataValid = false;

				} else { ${$text_formType.'is'.$dbInfo[4].'Valid'} = true; }
			}
			// isDate
			else if ($dbInfo[6])
			{	// Invalid Date Format
				if (count(explode('/', $_POST[$text_inputName])) != 3){
					${$text_formType.$dbInfo[4].'_Err'} = "Date format should be
DD/MM/YYYY.";
```

```
                                        $isSubmittedDataValid = false;
                                } // Invalid Type
                                else if (!is_numeric(explode('/', $_POST[$text_inputName])[1]) ||
                                                !is_numeric(explode('/', $_POST[$text_inputName])[0]) ||
                                                !is_numeric(explode('/', $_POST[$text_inputName])[2]) ){
                                        ${$text_formType.$dbInfo[4].'_Err'} = "Please provide a valid date.";
                                        $isSubmittedDataValid = false;
                                } // Invalid Date
                                else if (! (checkdate( explode('/', $_POST[$text_inputName])[1],
                                                                        explode('/',
$_POST[$text_inputName])[0],
                                                                        explode('/',
$_POST[$text_inputName])[2] )) ){
                                        ${$text_formType.$dbInfo[4].'_Err'} = "Please provide a valid date.";
                                        $isSubmittedDataValid = false;
                                } else { ${$text_formType.'is'.$dbInfo[4].'Valid'} = true; }
                        }
                        // isNum
                        else if ($dbInfo[7])
                        {       // isNumeric
                                if (is_numeric($_POST[$text_inputName])){
                                        // isInt
                                        if ($dbInfo[10]){
                                                if (filter_var($_POST[$text_inputName],
FILTER_VALIDATE_INT) || filter_var($_POST[$text_inputName], FILTER_VALIDATE_INT) === 0){
                                                        // hasRange
                                                        if ($dbInfo[12]){
                                                                // inRange
                                                                if ($_POST[$text_inputName] < $dbInfo[13]
|| $_POST[$text_inputName] > $dbInfo[14]){

        ${$text_formType.$dbInfo[4].'_Err'} = $dbInfo[4]." needs to be within range of ".$dbInfo[13]." to
".$dbInfo[14].".";
                                                                        $isSubmittedDataValid = false;
                                                                } else {
${$text_formType.'is'.$dbInfo[4].'Valid'} = true; }
                                                        } // not valid number (isNegative)
                                                        else if ($_POST[$text_inputName] < 0){
                                                                ${$text_formType.$dbInfo[4].'_Err'} =
$dbInfo[4]." needs to be a positive number.";
                                                                $isSubmittedDataValid = false;
                                                        } else { ${$text_formType.'is'.$dbInfo[4].'Valid'} =
true; }
                                                } // not Int
                                                else {
                                                        ${$text_formType.$dbInfo[4].'_Err'} = $dbInfo[4]."
needs to be an integer number.";
                                                        $isSubmittedDataValid = false;
                                                }
                                        } // isFloat
                                        else if ($dbInfo[11]){
                                                // hasRange
                                                if ($dbInfo[12]){
                                                        // inRange
                                                        if ($_POST[$text_inputName] < $dbInfo[13] ||
$_POST[$text_inputName] > $dbInfo[14]){
                                                                ${$text_formType.$dbInfo[4].'_Err'} =
$dbInfo[4]." needs to be within range of ".$dbInfo[13]." to ".$dbInfo[14].".";
                                                                $isSubmittedDataValid = false;
                                                        } else { ${$text_formType.'is'.$dbInfo[4].'Valid'} =
true; }
                                                } // not valid number (isNegative)
                                                else if ($_POST[$text_inputName] < 0){
                                                        ${$text_formType.$dbInfo[4].'_Err'} = $dbInfo[4]."
needs to be a positive number.";
                                                        $isSubmittedDataValid = false;
                                                } else { ${$text_formType.'is'.$dbInfo[4].'Valid'} = true; }
```

```
                                        } else { ${$text_formType.'is'.$dbInfo[4].'Valid'} = true; }
                                } // not a number
                                else {
                                        ${$text_formType.$dbInfo[4].'_Err'} = $dbInfo[4]." is not a number.";
                                        $isSubmittedDataValid = false;
                                }
                        } else { ${$text_formType.'is'.$dbInfo[4].'Valid'} = true; }
                }

                /*
                 * Store entries and log into file if no errors
                 */
                if ($isSubmittedDataValid)
                {       /*
                         * Store Entries
                         */

                        // Connect to database
                        $DB_connect = @new mysqli ($db_connect_var['host'], $db_connect_var['username'],
$db_connect_var['password'], $db_connect_var['database']);
                        // Database Connection Errors
                        if ($DB_connect->connect_errno > 0){ die();     }

                        /* Query : Insert Query String */
                        // Query First Half (Columns)
                        $prefix = '';
                        $queryString = "INSERT INTO movie (";
                        foreach ($dbInfoArray as $dbInfo){
                                $queryString .= (empty($prefix))? ":$prefix;
                                $queryString .= $dbInfo[1];
                                $prefix = ', ';
                        }
                        $prefix = '';
                        $queryString .= ")";
                        // Query Second Half (Values)
                        $queryString .= " VALUES (";
                        foreach ($dbInfoArray as $dbInfo){
                                $queryString .= (empty($prefix))? ":$prefix;
                                $queryString .= "". $_POST[$text_formType.$dbInfo[4]] . "";
                                $prefix = ', ';
                        }
                        $queryString .= ")";

                        // Execute Insert Query
                        if ($DB_connect->query($queryString) === TRUE){
                                // update intervals for the record specified by imdbRefID
                                db_updateIntervals($_POST[$text_formType.$dbInfoArray[1][4]]);

                                /*
                                 * Store Log
                                 * Format : operation type, formValid & formInvalid, fieldsValid & fieldsInvalid
                                 */

                                // Set Log Values
                                $log_time = date("Y-m-d");
                                $log_operationType = "insert";
                                $log_formValid = "true";

                                // Store Log
                                $prefix = '';
                                $logString = $log_time .','. $log_operationType .','. $log_formValid .',';
                                foreach ($dbInfoArray as $dbInfo){
                                        $logString .= empty($prefix)?":$prefix;
                                        $logString .= 'true';
                                        $prefix = ',';
                                }
```

```
                                        // add in a new log for SUCCESSFUL form submission into log file
                                        is_dir($dir_root.'/'.$dir_log_folder) || @mkdir($dir_root.'/'.$dir_log_folder) ||
die("Can't Create folder");
                                        file_put_contents($dir_root.'/'.$dir_log_folder.'/'.$logFileName,
stripslashes($logString)."\r\n", FILE_APPEND);

                                        $globalMsg = "<span class='success'>Creation Form Submission
Completed</span>";
                                } else {
                                        $globalMsg = "<span class='fail'>Creation Form Submission Incomplete - Database
Insert Error (".$DB_connect->error.")</span>";

                                        // Update Sticky Fields with error messages for user input
                                        foreach ($dbInfoArray as $dbInfo)
                                        {       $text_inputName = $text_formType.$dbInfo[4];
                                                ${$text_formType.$dbInfo[4].'_Val'}    =
(${$text_formType.'is'.$dbInfo[4].'Valid'})?        $_POST[$text_inputName]: '';
                                        }
                                }

                                // Close database connection
                                $DB_connect->close();
                        }
                        else
                        {
                                // Update Sticky Fields with error messages for user input
                                foreach ($dbInfoArray as $dbInfo)
                                {       $text_inputName = $text_formType.$dbInfo[4];
                                        ${$text_formType.$dbInfo[4].'_Val'}    =
(${$text_formType.'is'.$dbInfo[4].'Valid'})?        $_POST[$text_inputName]: '';
                                }

                                // Set Log Values
                                $log_time = date("Y-m-d");
                                $log_operationType = "insert";
                                $log_formValid = "false";

                                // Store Log
                                $prefix = '';
                                $logString = $log_time .','. $log_operationType .','. $log_formValid .',';
                                foreach ($dbInfoArray as $dbInfo){
                                        $logString .= empty($prefix)?'':$prefix;
                                        $logString .= (${$text_formType.'is'.$dbInfo[4].'Valid'})? 'true':'false';
                                        $prefix = ',';
                                }

                                // add in a new log for UNSUCCESSFUL form submission into log file
                                is_dir($dir_root.'/'.$dir_log_folder) || @mkdir($dir_root.'/'.$dir_log_folder) || die("Can't Create
folder");
                                file_put_contents($dir_root.'/'.$dir_log_folder.'/'.$logFileName, stripslashes($logString)."\r\n",
FILE_APPEND);

                                $globalMsg = "<span class='fail'>Creation Form Submission Invalid</span>";
                        }
                }
                // Update Form
                else if (array_key_exists("update_form", $_POST))
                {       $text_formType      = 'UForm_';

                        // Sanitize & Validate Input
                        foreach ($dbInfoArray as $dbInfo)
                        {       $text_inputName = $text_formType.$dbInfo[4];

                                /* Sanitize Input */
                                $_POST[$text_inputName] = (inputNotEmpty($_POST[$text_inputName]))?
                sanitize_input($_POST[$text_inputName]) : null;
```

```
                        /*
                         * Validate Input, and
                         * If Invalid:
                         * - Set Error Strings
                         * - Set Boolean for Invalid Field
                         */
                        // isEmpty
                        if (!inputNotEmpty($_POST[$text_inputName])){
                                ${$text_formType.$dbInfo[4].'_Err'} = "Please fill in the field for ".$dbInfo[4].".";
                                $isSubmittedDataValid = false;
                        }
                        // isCommaDelimited
                        else if ($dbInfo[5]){
                                $_POST[$text_inputName] = str_replace(' ', '', $_POST[$text_inputName]);
                                // Invalid Comma Delimited Text
                                if( in_array("", explode(',', $_POST[$text_inputName])) ){
                                        ${$text_formType.$dbInfo[4].'_Err'} = "Please provide a valid comma
delimited text";
                                        $isSubmittedDataValid = false;
                                } // Limit comma delimited
                                else if (count(explode(',', $_POST[$text_inputName])) > 5){
                                        ${$text_formType.$dbInfo[4].'_Err'} = "Please provide less than or equal
to 5 ".$dbInfo[4];
                                        $isSubmittedDataValid = false;

                                } else { ${$text_formType.'is'.$dbInfo[4].'Valid'} = true; }
                        }
                        // isDate
                        else if ($dbInfo[6])
                        {       // Invalid Date Format
                                if (count(explode('/', $_POST[$text_inputName])) != 3){
                                        ${$text_formType.$dbInfo[4].'_Err'} = "Date format should be
DD/MM/YYYY.";
                                        $isSubmittedDataValid = false;
                                } // Invalid Type
                                else if (!is_numeric(explode('/', $_POST[$text_inputName])[1]) ||
                                                !is_numeric(explode('/', $_POST[$text_inputName])[0]) ||
                                                !is_numeric(explode('/', $_POST[$text_inputName])[2]) ){
                                        ${$text_formType.$dbInfo[4].'_Err'} = "Please provide a valid date.";
                                        $isSubmittedDataValid = false;
                                } // Invalid Date
                                else if (! (checkdate( explode('/', $_POST[$text_inputName])[1],
                                                                        explode('/',
$_POST[$text_inputName])[0],
                                                                        explode('/',
$_POST[$text_inputName])[2] )) ){
                                        ${$text_formType.$dbInfo[4].'_Err'} = "Please provide a valid date.";
                                        $isSubmittedDataValid = false;
                                } else { ${$text_formType.'is'.$dbInfo[4].'Valid'} = true; }
                        }
                        // isNum
                        else if ($dbInfo[7])
                        {       // isNumeric
                                if (is_numeric($_POST[$text_inputName])){
                                        // isInt
                                        if ($dbInfo[10]){
                                                if (filter_var($_POST[$text_inputName],
FILTER_VALIDATE_INT) || filter_var($_POST[$text_inputName], FILTER_VALIDATE_INT) === 0){
                                                        // hasRange
                                                        if ($dbInfo[12]){
                                                                // inRange
                                                                if ($_POST[$text_inputName] < $dbInfo[13]
|| $_POST[$text_inputName] > $dbInfo[14]){

        ${$text_formType.$dbInfo[4].'_Err'} = $dbInfo[4]." needs to be within range of ".$dbInfo[13]." to
".$dbInfo[14].".";
                                                                        $isSubmittedDataValid = false;
```

```
                                                                        } else {
${$text_formType.'is'.$dbInfo[4].'Valid'} = true; }
                                                                } // not valid number (isNegative)
                                                                else if ($_POST[$text_inputName] < 0){
                                                                        ${$text_formType.$dbInfo[4].'_Err'} =
$dbInfo[4]." needs to be a positive number.";
                                                                        $isSubmittedDataValid = false;
                                                                } else { ${$text_formType.'is'.$dbInfo[4].'Valid'} =
true; }
                                                        } // not Int
                                                        else {
                                                                ${$text_formType.$dbInfo[4].'_Err'} = $dbInfo[4]."
needs to be an integer number.";
                                                                $isSubmittedDataValid = false;
                                                        }
                                                } // isFloat
                                                else if ($dbInfo[11]){
                                                        // hasRange
                                                        if ($dbInfo[12]){
                                                                // inRange
                                                                if ($_POST[$text_inputName] < $dbInfo[13] ||
$_POST[$text_inputName] > $dbInfo[14]){
                                                                        ${$text_formType.$dbInfo[4].'_Err'} =
$dbInfo[4]." needs to be within range of ".$dbInfo[13]." to ".$dbInfo[14].".";
                                                                        $isSubmittedDataValid = false;
                                                                } else { ${$text_formType.'is'.$dbInfo[4].'Valid'} =
true; }
                                                        } // not valid number (isNegative)
                                                        else if ($_POST[$text_inputName] < 0){
                                                                ${$text_formType.$dbInfo[4].'_Err'} = $dbInfo[4]."
needs to be a positive number.";
                                                                $isSubmittedDataValid = false;
                                                        } else { ${$text_formType.'is'.$dbInfo[4].'Valid'} = true; }
                                                } else { ${$text_formType.'is'.$dbInfo[4].'Valid'} = true; }
                                        } // not a number
                                        else {
                                                ${$text_formType.$dbInfo[4].'_Err'} = $dbInfo[4]." is not a number.";
                                                $isSubmittedDataValid = false;
                                        }
                                } else { ${$text_formType.'is'.$dbInfo[4].'Valid'} = true; }
                        }

                        /*
                         * Update record and log into file if no errors
                         */
                        $_POST['UForm_search'] = (inputNotEmpty($_POST['UForm_search']))?
                sanitize_input($_POST['UForm_search']) : null;
                        if ($isSubmittedDataValid)
                        {       /*
                                 * Update Record
                                 */

                                // Connect to database
                                $DB_connect = @new mysqli ($db_connect_var['host'], $db_connect_var['username'],
$db_connect_var['password'], $db_connect_var['database']);
                                // Database Connection Errors
                                if ($DB_connect->connect_errno > 0){ die();     }

                                if (inputNotEmpty($_POST['UForm_search'])){
                                        /* Query : Select Record by IMDbRefID to check if it exists */
                                        $query = "SELECT * FROM movie WHERE
".$dbName_searchRefID."='".$_POST['UForm_search']."'";;
                                        if ($pQuery = $DB_connect->prepare($query)){
                                                $pQuery->execute();

                                                // my_sqli result Object
                                                $result = $pQuery->get_result();
```

```
                                        // array of records
                                        $table = $result->fetch_all();

                                        if (!empty($table)){
                                                /* Query : Update Query String */
                                                $prefix = '';
                                                $queryString = "UPDATE movie SET ";
                                                foreach ($dbInfoArray as $dbInfo){
                                                        $queryString .= (empty($prefix))? ":$prefix;
                                                        $queryString .=
$dbInfo[1]."='".$_POST[$text_formType.$dbInfo[4]]."'";
                                                        $prefix = ', ';
                                                }
                                                $queryString .= " WHERE
".$dbName_searchRefID."='".$_POST['UForm_search']."'";

                                                // Execute Update Query
                                                if ($DB_connect->query($queryString) === TRUE){
                                                        // update intervals for the record specified by
imdbRefID

        db_updateIntervals($_POST[$text_formType.$dbInfoArray[1][4]]);

                                                        /*
                                                         * Store Log
                                                         * Format : operation type, formValid & formInvalid,
fieldsValid & fieldsInvalid
                                                         */

                                                        // Set Log Values
                                                        $log_time = date("Y-m-d");
                                                        $log_operationType = "update";
                                                        $log_formValid = "true";

                                                        // Store Log
                                                        $prefix = '';
                                                        $logString = $log_time .','. $log_operationType .','.
$log_formValid .',';
                                                        foreach ($dbInfoArray as $dbInfo){
                                                                $logString .= empty($prefix)?":$prefix;
                                                                $logString .= 'true';
                                                                $prefix = ',';
                                                        }

                                                        // add in a new log for SUCCESSFUL form
submission into log file
                                                        is_dir($dir_root.'/'.$dir_log_folder) ||
@mkdir($dir_root.'/'.$dir_log_folder) || die("Can't Create folder");

        file_put_contents($dir_root.'/'.$dir_log_folder.'/'.$logFileName, stripslashes($logString)."\r\n", FILE_APPEND);

                                                        $globalMsg = "<span class='success'>Update Form
Submission Completed</span>";
                                                } else {
                                                        $globalMsg = "<span class='fail'>Update Form
Submission Incomplete - Database Update Error (".$DB_connect->error.")</span>";

                                                        // Update Sticky Fields with error messages for user
input
                                                        foreach ($dbInfoArray as $dbInfo)
                                                        {       $text_inputName =
$text_formType.$dbInfo[4];
                                                                ${$text_formType.$dbInfo[4].'_Val'}    =
(${$text_formType.'is'.$dbInfo[4].'Valid'})?        $_POST[$text_inputName]: '';
                                                        } $UForm_search_Val = $_POST['UForm_search'];
                                                }
                                        } // No Record to Delete
```

```
                                        else {
                                                $globalMsg = "<span class='fail'>Update Form Incomplete -
Target Record to Update Does Not Exist</span>";

                                                // Update Sticky Fields with error messages for user input
                                                foreach ($dbInfoArray as $dbInfo)
                                                {       $text_inputName = $text_formType.$dbInfo[4];
                                                        ${$text_formType.$dbInfo[4].'_Val'}    =
(${$text_formType.'is'.$dbInfo[4].'Valid'})?    $_POST[$text_inputName]: '';
                                                } $UForm_search_Val = $_POST['UForm_search'];
                                        }
                                }
                        } // Target Record not Specified
                        else {
                                $globalMsg = "<span class='fail'>Update Form Submission Incomplete - Missing
Detail(RefID) for Target Record to Update</span>";

                                // Update Sticky Fields with error messages for user input
                                foreach ($dbInfoArray as $dbInfo)
                                {       $text_inputName = $text_formType.$dbInfo[4];
                                        ${$text_formType.$dbInfo[4].'_Val'}    =
(${$text_formType.'is'.$dbInfo[4].'Valid'})?    $_POST[$text_inputName]: '';
                                } $UForm_search_Val = $_POST['UForm_search'];
                        }

                        // Close database connection
                        $DB_connect->close();
                }
                else
                {
                        // Update Sticky Fields with error messages for user input
                        foreach ($dbInfoArray as $dbInfo)
                        {       $text_inputName = $text_formType.$dbInfo[4];
                                ${$text_formType.$dbInfo[4].'_Val'}    =
(${$text_formType.'is'.$dbInfo[4].'Valid'})?    $_POST[$text_inputName]: '';
                        } $UForm_search_Val = $_POST['UForm_search'];

                        // Set Log Values
                        $log_time = date("Y-m-d");
                        $log_operationType = "update";
                        $log_formValid = "false";

                        // Store Log
                        $prefix = '';
                        $logString = $log_time .','. $log_operationType .','. $log_formValid .',';
                        foreach ($dbInfoArray as $dbInfo){
                                $logString .= empty($prefix)?'':$prefix;
                                $logString .= (${$text_formType.'is'.$dbInfo[4].'Valid'})? 'true':'false';
                                $prefix = ',';
                        }

                        // add in a new log for UNSUCCESSFUL form submission into log file
                        is_dir($dir_root.'/'.$dir_log_folder) || @mkdir($dir_root.'/'.$dir_log_folder) || die("Can't Create
folder");
                        file_put_contents($dir_root.'/'.$dir_log_folder.'/'.$logFileName, stripslashes($logString)."\r\n",
FILE_APPEND);

                        $globalMsg = "<span class='fail'>Update Form Submission Invalid</span>";
                }
        }
        // Delete Form
        else if (array_key_exists("delete_form", $_POST))
        {       $text_formType      = 'DForm_';

                // Sanitize Input
                $_POST['DForm_search'] = (inputNotEmpty($_POST['DForm_search']))?
        sanitize_input($_POST['DForm_search']) : null;
```

```
		/*
		 * Delete record and log into file if no errors
		 */

		// Connect to database
		$DB_connect = @new mysqli ($db_connect_var['host'], $db_connect_var['username'],
$db_connect_var['password'], $db_connect_var['database']);
		// Database Connection Errors
		if ($DB_connect->connect_errno > 0){ die();	}

		if (inputNotEmpty($_POST['DForm_search']))
		{
			/* Query : Select Record by IMDbRefID to check if it exists */
			$query = "SELECT * FROM movie WHERE
".$dbName_searchRefID."='".$_POST['DForm_search']."';;
			if ($pQuery = $DB_connect->prepare($query)){
				$pQuery->execute();

				// my_sqli result Object
				$result = $pQuery->get_result();
				// array of records
				$table = $result->fetch_all();

				if (!empty($table)){
					/* Query : Delete Query String */
					$prefix = '';
					$queryString = "DELETE FROM movie WHERE ";
					$queryString .= $dbName_searchRefID."='".$_POST['DForm_search']."'";

					// Execute Update Query
					if ($DB_connect->query($queryString) === TRUE){
						/*
						 * Store Log
						 * Format : operation type, formValid & formInvalid, fieldsValid
& fieldsInvalid
						 */

						// Set Log Values
						$log_time = date("Y-m-d");
						$log_operationType = "delete";
						$log_formValid = "true";

						// Store Log
						$prefix = '';
						$logString = $log_time .','. $log_operationType .','.
$log_formValid .',';
						foreach ($dbInfoArray as $dbInfo){
							$logString .= empty($prefix)?'':$prefix;
							$logString .= 'true';
							$prefix = ',';
						}

						// add in a new log for SUCCESSFUL form submission into log
file
						is_dir($dir_root.'/'.$dir_log_folder) ||
@mkdir($dir_root.'/'.$dir_log_folder) || die("Can't Create folder");
						file_put_contents($dir_root.'/'.$dir_log_folder.'/'.$logFileName,
stripslashes($logString)."\r\n", FILE_APPEND);

						$globalMsg = "<span class='success'>Delete Form Submission
Completed</span>";
					} else {
						$globalMsg = "<span class='fail'>Delete Form Submission
Incomplete - Database Delete Error (".$DB_connect->error.")</span>";
						$DForm_search_Val = $_POST['DForm_search'];
					}
```

```
                                        } // No Record to Delete
                                        else {
                                                $globalMsg = "<span class='fail'>Delete Form Incomplete - Target Record
to Delete Does Not Exist</span>";
                                                $DForm_search_Val = $_POST['DForm_search'];
                                        }
                                }
                        } // Target Record not Specified
                        else {
                                $globalMsg = "<span class='fail'>Delete Form Incomplete - Missing Detail(RefID) for Target
Record to Delete</span>";
                                $DForm_search_Val = $_POST['DForm_search'];
                        }

                        // Close database connection
                        $DB_connect->close();
                }
?>

<!-- read from log file to get summary of log contents -->
<?php

                // initial form summary values
                $logCount = $formValid = $formInvalid = 0;
                // initial operation type summary values
                $insertCount = $updateCount = $deleteCount = 0;
                // valid & invalid field summary values
                foreach ($dbInfoArray as $dbInfo){
                        ${$dbInfo[4].'_validCount'} = 0;
                        ${$dbInfo[4].'_invalidCount'} = 0;
                }

                // update form and field summary values if log exists (no log if no previous form submission)
                if (file_exists($dir_log_folder.'/'.$logFileName))
                {       // get log file details
                        $logFile = file($dir_log_folder.'/'.$logFileName, FILE_IGNORE_NEW_LINES);

                        // get log summary details
                        foreach ($logFile as $logFileLine){
                                if (!empty($logFileLine)){
                                        $logFileLineArray = explode(',', $logFileLine);

                                        if ( (count($logFileLineArray)-3) == count($dbInfoArray)){
                                                if (count(explode('-', $logFileLineArray[0])) == 3){
                                                        if ( (checkdate( explode('-', $logFileLineArray[0])[1],
                                                                                explode('-',
$logFileLineArray[0])[2],
                                                                                explode('-',
$logFileLineArray[0])[0] )) )
                                                        {
                                                                // look into each logFileLineArray Element
                                                                // constraint: <= 7 days
                                                                $daysDiff = (new DateTime($logFileLineArray[0]))-
>diff(new DateTime());
                                                                if ($daysDiff->days <= 7){

                                                                        if ($logFileLineArray[1] == 'insert')
                $insertCount++;
                                                                        else if ($logFileLineArray[1] == 'update')
                $updateCount++;
                                                                        else if ($logFileLineArray[1] == 'delete')
                $deleteCount++;
                                                                        ($logFileLineArray[2] === 'true')?
                        $formValid++                            : $formInvalid++;
                                                                        // update valid & invalid counts
                                                                        for ($lfl_i = 3; $lfl_i <
count($logFileLineArray); $lfl_i++){
```

```
                                                                ($logFileLineArray[$lfl_i] ===
'true')?
                                                                ${$dbInfoArray[$lfl_i-
3][4].'_validCount'}++ : ${$dbInfoArray[$lfl_i-3][4].'_invalidCount'}++;
                                                        }

                                                        // update form count
                                                        $logCount++;
                                                }
                                        }
                                }
                        }
                } // end of summary details
        } // log exists but empty content
} //else { echo $dir_log_folder.'/'.$logFileName; }

?>

<!-- ------------------------------Top Content Section------------------------------ -->
<div class="top_content"><div id="top_content_msg"><?= empty($globalMsg)?"":$globalMsg?></div></div>

<!-- ------------------------------Content Section------------------------------ -->
<div class="content">

        <!-- ------------------------------Left Content Section------------------------------ -->
        <div class="left_content">
                <div>
                        <!-- Navigation Tabs -->
                        <nav class="nav_devTabs">
                                <button onclick="tabSelect('right_content_summary',
'devTabs')">Summary</button>
                                <button onclick="tabSelect('right_content_log', 'devTabs')">Log</button>
                                <button onclick="tabSelect('right_content_form', 'devTabs')">CUD Form</button>
                                <button onclick="tabSelect('right_content_api', 'devTabs')">API Form</button>
                        </nav>
                </div>
        </div>
        <!-- End of Left Content Section -->

        <!-- ------------------------------Right Content Section------------------------------ -->
        <div class="right_content">

                <div class="devTabs" id="right_content_summary">
                        <!-- Summary Title -->
                        <div class="right_content_title"><h1>Log Summary</h1></div>

                        <!-- Tables for Form Summary, Operation Types, Field Valid Summary, and Field Invalid
Summary -->
                        <div id="right_content_summary_content">
                                <table>  <tr>     <th colspan="2">Forms Processed</th>
                                                </tr>
                                                <tr>     <td>Total Forms</td>
        <td><?=$logCount?></td>                                 </tr>
                                                <tr>     <td>Valid Forms</td>
        <td><?=$formValid?></td>                                <tr>
                                                <tr>     <td>Invalid Forms</td>
        <td><?=$formInvalid?></td>                              </tr>     </table>

                                <table>  <tr>     <th colspan="2">Operation Types</th>
                                                </tr>
                                                <tr>     <td>Creation</td>
        <td><?=$insertCount?></td>                              </tr>
                                                <tr>     <td>Update</td>
        <td><?=$updateCount?></td>                              </tr>
                                                <tr>     <td>Delete</td>
        <td><?=$deleteCount?></td>                              </tr>     </table>
```

```
                                    <table>   <tr>      <th colspan="2">Valid Fields</th>
                                                        </tr><?php
                                    foreach ($dbInfoArray as $dbInfo)
                                    { echo "<tr>        <td>$dbInfo[4]</td>
        <td>".${$dbInfo[4].'_validCount'}."</td>        </tr>"; }
                                    ?></table>

                                    <table>   <tr>      <th colspan="2">Invalid Fields</th>
                                                        </tr><?php
                                    foreach ($dbInfoArray as $dbInfo)
                                    { echo "<tr>        <td>$dbInfo[4]</td>
        <td>".${$dbInfo[4].'_invalidCount'}."</td>      </tr>"; }
                                    ?></table>
                                </div>
                        </div>

                        <div class="devTabs" id="right_content_log">
                                <!-- Log Title -->
                                <div class="right_content_title">
                                        <h1>Log Details</h1>

                                        <!-- Download Log Form -->
                                        <div id="right_content_download_log">
                                                <form action="dev_downloadLog.php" method="POST">
                                                        <select name="fileToDownload">
                                                                <option value="<?php echo $logFileName ?>"> <?php
echo $logFileName ?> </option>
                                                        </select>

                                                        <input type="submit" value="Download"/>
                                                </form>
                                        </div>
                                </div>

                                <!-- Table for Log File Entries -->
                                <div class="shortenDiv" id="right_content_log_FileDetails"
onclick="toggleDisplayHorizontal('right_content_log_FileDetails')">

                                        <!-- display log file details -->
                                        <?php
                                        echo "<table>";
                                                echo "<tr>
                                                        <th>Log No</th>                 <th>Log
Date</th>                       <th>Operation Type</th>     <th>Form Valid</th>
                                                        <th>Title</th>
        <th>imdbRefID</th>                      <th>Released Date</th>
                                                        <th>Type</th>
        <th>Rating</th>                         <th>Genre</th>
                                                        <th>Total Season</th>           <th>Duration</th>
        <th>Language</th>                       <th>Country</th>
                                                        <th>MetaScore</th>              <th>imdb
Rating</th>          <th>imdb Votes</th>
                                                  </tr>";

                                                if (file_exists($dir_log_folder.'/'.$logFileName))
                                                {       // get log file details
                                                        $logFile = file($dir_log_folder.'/'.$logFileName,
FILE_IGNORE_NEW_LINES);

                                                        // read each line in log file
                                                        $logFileLineCount = 0;
                                                        foreach ($logFile as $logFileLine){
                                                                if (!empty($logFileLine)){
                                                                        $logFileLineArray = explode(',',
$logFileLine);

                                                                        if ( count($logFileLineArray) == 16){
```

```
                                                                                    if (count(explode('-',
$logFileLineArray[0])) == 3){
                                                                                        if ( (checkdate(
explode('-', $logFileLineArray[0])[1],
                                explode('-', $logFileLineArray[0])[2],
                                explode('-', $logFileLineArray[0])[0] )) )
                                                                                        {
                                                                                            // look into
each logFileLineArray Element
                                                                                            // constraint:
<= 7 days
                                                                                            $daysDiff =
(new DateTime($logFileLineArray[0]))->diff(new DateTime());
                                                                                            if ($daysDiff-
>days <= 7){
                        $logFileLineCount++;
                                                                                                echo
"<tr>";
                        echo "<td>$logFileLineCount</td>";
                        foreach ($logFileLineArray as $logFileLineElement){
                                if ($logFileLineElement == 'false') {
                                        echo "<td style='background-color: pink;'>$logFileLineElement</td>";
                                } else { echo "<td>$logFileLineElement</td>"; }
                        }
                                                                                                echo
"</tr>";
                                                                                            }
                                                                                        }
                                                                                    }
                                                                                }
                                                                            } // end of summary details
                                                                        } // log exists but empty content
                                                                        if ($logFileLineCount === 0){ echo "<tr><td colspan='17'>No
Log Content</td></tr>"; }
                                                                    } else { echo "<tr><td colspan='17'>No Log</td></tr>"; }
                                                                echo "</table>";
                                                                ?>
                                </div>
                        </div>
                        <!-- Form -->
                        <div class="devTabs" id="right_content_form">
                                <!-- Form Title -->
                                <div class="right_content_title"><h1>Creation/Update/Delete Form</h1></div>
                                <!-- Form Tabs -->
                                <div class="nav_formTabs">
                                        <button class="formTabs_btn formTabs_btn_active"
onclick="tabSelect('right_content_creation_form', 'formTabs', this, 'formTabs_btn', true, 'formTabs_btn_active')">Creation
Form<hr/></button>
                                        <button class="formTabs_btn" onclick="tabSelect('right_content_update_form',
'formTabs', this, 'formTabs_btn', true, 'formTabs_btn_active')">Update Form<hr/></button>
                                        <button class="formTabs_btn" onclick="tabSelect('right_content_delete_form',
'formTabs', this, 'formTabs_btn', true, 'formTabs_btn_active')">Delete Form<hr/></button>
                                </div>
```

```
<!-- Form Content (Creation/Update/Delete Forms) -->
<div id="right_content_forms">

	<!-- Creation Form Div -->
	<div class="formTabs" id="right_content_creation_form">
		<!-- Creation Form Content -->
		<div id="creation_form_content">

<form action="<?php echo
htmlspecialchars($_SERVER['PHP_SELF']); ?>" method="POST">
<fieldset>
<legend
id="creation_form_legend">Creation Form</legend>

<div id="creation_form_inputs_container">
<?php
foreach ($dbInfoArray as $dbInfo)
{
$text_label =
$dbInfo[4];
$text_inputName =
'CForm_'.$dbInfo[4];
$text_inputID =
$text_inputName;
$text_spanID =
'CForm_'.$dbInfo[4].'_Error';
$val_input =
inputNotEmpty(${'CForm_'.$dbInfo[4].'_Val'})?${'CForm_'.$dbInfo[4].'_Val'}:'';
$val_span =
(!empty(${'CForm_'.$dbInfo[4].'_Err'}))?${'CForm_'.$dbInfo[4].'_Err'}:'';

echo "<div
class='creation_form_inputs_section'>";

echo "<label
for='$text_inputID'>$text_label</label><br/>";
echo "<input type='text'
name='$text_inputName' id='$text_inputID' value='".$val_input."' />";
echo "<span
id='$text_spanID'>",$val_span,"</span>";

echo "</div>";
}
?>

</div>

<!-- Reset & Submit -->
<div class="forms_btns_container">
<div><a class="forms_btns
resetLink" href="<?php echo htmlspecialchars($_SERVER['PHP_SELF']); ?>">Reset</a></div>
<div><input type="submit"
class="forms_btns formSubmitBtn" id="" value="Create Record"/></div>
<input type="hidden"
name="creation_form"  value="creation_form"/>
</div>
</fieldset>
</form>

	</div>
</div>

<!-- Update Form Div -->
<div class="formTabs" id="right_content_update_form">
	<!-- Update Form Content -->
	<div id="update_form_content">
```

```
                                                <form action="<?php echo
htmlspecialchars($_SERVER['PHP_SELF']); ?>" method="POST"
                                                    onkeypress="return event.keyCode != 13;">
                                                <fieldset>
                                                    <legend id="update_form_legend">Update
Form</legend>

                                                    <div id="update_form_search">
                                                        <label
for="UForm_search">Target Record</label><br/>
                                                        <input type="text"
id="UForm_search" name="UForm_search"
                                                            placeholder="Search
by IMDbRefID" value="<?=$UForm_search_Val?>"
onkeyup="getData_FormAutoComplete('UForm_search','autocomplete_u')"
onchange="getData_FormData('u')"><br/>
                                                        <button type="button"
onclick="getData_FormData('u')">Search</button>
                                                        <div class="autocomplete_popup"
id="autocomplete_u"
onmouseleave="popup_disappear('autocomplete_u')" onclick="popup_disappear('autocomplete_u')"></div>
                                                    </div>

                                                    <div id="update_form_inputs_container">
                                                    <?php
                                                    foreach ($dbInfoArray as $dbInfo)
                                                    {
                                                        $text_label         =
$dbInfo[4];
                                                        $text_inputName     =
'UForm_'.$dbInfo[4];
                                                        $text_inputID       =
$text_inputName;
                                                        $text_spanID        =
'UForm_'.$dbInfo[4].'_Error';
                                                        $val_input          =
inputNotEmpty(${'UForm_'.$dbInfo[4].'_Val'})?${'UForm_'.$dbInfo[4].'_Val'}:'';
                                                        $val_span           =
(!empty(${'UForm_'.$dbInfo[4].'_Err'}))?${'UForm_'.$dbInfo[4].'_Err'}:'';

                                                        echo "<div
class='update_form_inputs_section'>";

                                                            echo "<label
for='$text_inputID'>$text_label</label><br/>";
                                                            echo "<input type='text'
name='$text_inputName' id='$text_inputID' value='".$val_input."' />";
                                                            echo "<span
id='$text_spanID'>",$val_span,"</span>";

                                                        echo "</div>";
                                                    }
                                                    ?>
                                                    </div>

                                                    <!-- Reset & Submit -->
                                                    <div class="forms_btns_container">
                                                        <div><a class="forms_btns
resetLink" href="<?php echo htmlspecialchars($_SERVER['PHP_SELF']); ?>">Reset</a></div>
                                                        <div><input class="forms_btns
formSubmitBtn" type="submit" value="Update Record"/></div>
                                                        <input type="hidden"
name="update_form"  value="update_form"/>
                                                    </div>
```

```
                                        </fieldset>
                                    </form>

                                </div>
                            </div>

                            <!-- Delete Form Div -->
                            <div class="formTabs" id="right_content_delete_form">
                                <!-- Delete Form Content -->
                                <div id="delete_form_content">

                                    <form action="<?php echo
htmlspecialchars($_SERVER['PHP_SELF']); ?>" method="POST"
                                          onkeypress="return event.keyCode != 13;">
                                        <fieldset>
                                            <legend id="delete_form_legend">Delete
Form</legend>

                                            <div id="delete_form_search">
                                                <label
for="DForm_search">Target Record</label><br/>
                                                <input type="text"
id="DForm_search" name="DForm_search"
                                                       placeholder="Search
by IMDbRefID" value="<?=$DForm_search_Val?>"

onkeyup="getData_FormAutoComplete('DForm_search','autocomplete_d')"

onchange="getData_FormData('d')"><br/>
                                                <button type="button"
onclick="getData_FormData('d')">Search</button>
                                                <div class="autocomplete_popup"
id="autocomplete_d"

onmouseleave="popup_disappear('autocomplete_d')" onclick="popup_disappear('autocomplete_d')"></div>
                                            </div>

                                            <div id="delete_form_inputs_container">
                                            <?php
                                            foreach ($dbInfoArray as $dbInfo)
                                            {       $text_inputID       =
'DForm_'.$dbInfo[4];

                                                echo "<div
class='delete_form_inputs_section'>";
                                                    echo "<div
id='$text_inputID'></div>";
                                                echo "</div>";
                                            }
                                            ?>
                                            </div>

                                            <!-- Reset & Submit -->
                                            <div class="forms_btns_container">
                                                <div><a class="forms_btns
resetLink" href="<?php echo htmlspecialchars($_SERVER['PHP_SELF']); ?>">Reset</a></div>
                                                <div><input class="forms_btns
formSubmitBtn" type="submit" value="Delete Record"/></div>
                                                <input type="hidden"
name="delete_form" value="delete_form"/>
                                            </div>
                                        </fieldset>
                                    </form>

                                </div>
                            </div>
```

```
                </div> <!-- End of Form Content (Creation/Update/Delete Forms) -->
            </div> <!-- End of Right Content Form -->

            <!-- API Form (OMDb API Form) -->
            <div class="devTabs" id="right_content_api">
                <!-- API Form Title -->
                <div class="right_content_title"><h1>API Form</h1></div>

                <!-- Form to get record information from OMDB API -->
                <div id="right_content_api_form">

                    <form action="http://www.omdbapi.com/" method="GET" onsubmit="return
false;">
                        <fieldset>
                            <legend id="api_form_legend">Search Using IMDb
ID</legend>

                            <div id="api_form_inputs_container">
                                <label for="i">ID:</label>
                                <input type="text" id="i" name="i"
placeholder="IMDb ID">
                                  

                                <label>Plot:</label>
                                <select name="plot" style="width: 100px;">
                                    <option value=""
selected="">Short</option>
                                    <option value="full">Full</option>
                                </select>
                                  

                                <label>Response:</label>
                                <select name="r" style="width: 100px;">
                                    <option value="">JSON</option>
                                    <option value="xml">XML</option>
                                </select>
                                  

                                <button type="button"
onclick="getData_OMDbAPI()">Search</button>
                                <button type="reset">Reset</button>
                            </div>
                        </fieldset>
                    </form>

                </div>
            </div>

        </div>
        <!-- End of Right Content Section -->

</div>
<!-- End of Content Section -->
```

Detailed PHP Code (Code involving AJAX response)

user_progressiveFacetSearch.php

```
<?php
	include 'php-commons/database.php';
	include 'php-commons/constants.php';
	include 'php-commons/filters.php';
	include 'php-commons/functions.php';

	/************************* Facet Filter (AJAX): Respond to AJAX call *************************/

	// get json_encoded userSelections from POST, and Decode it into an array
	$userSelections_array = json_decode($_POST['userSelections'], true);

	// default sort order = sort by unique value/interval, if false, sort by count
	$toSortUnique = (array_key_exists('sortOrder', $_POST))? $_POST['sortOrder']:'true';
	if ($toSortUnique === 'false') $toSortUnique = false;
	else if ($toSortUnique === 'true') $toSortUnique = true;
	else $toSortUnique = true;

	$indexed_filtered_facetArray_unique = array();
	if (!empty($userSelections_array))
	{	/*
		 * standardize super and sub arrays into a single array (sub)
		 */
		foreach ($userSelections_array as $section => $array_of_super_or_sub_or_select_group)
		{	// in each section, find which section it is so that the correct facetName can be provided
			$currentFacetName = section_facetName($section);
			// count whether there is only select/super/sub, or there are both super and sub
			$super_or_sub_or_select_groupKeys = array_keys($array_of_super_or_sub_or_select_group);

			// get corresponding intervals for each category and add it to the sub name's array
			if (count($super_or_sub_or_select_groupKeys) == 1) {
				if (strpos($super_or_sub_or_select_groupKeys[0], "super")) {
					// get corresponding sub name of super
					$string_sub =
str_replace("super","sub",$super_or_sub_or_select_groupKeys[0]);

					// get all categories selected by user

	foreach($userSelections_array[$section][$super_or_sub_or_select_groupKeys[0]] as $category)
					{
						// get all intervals of each category
						$selectedIntervals =
filter_classificationToInterval($currentFacetName, $category);
						// add intervals to sub
						foreach($selectedIntervals as $interval){
$userSelections_array[$section][$string_sub][] = $interval; }
					}
					// delete current category array

	unset($userSelections_array[$section][$super_or_sub_or_select_groupKeys[0]]);
				}
			}
			else if (count($super_or_sub_or_select_groupKeys) == 2) {
				if (strpos($super_or_sub_or_select_groupKeys[0], "super")) {
					// get corresponding sub name of super
					$string_sub =
str_replace("super","sub",$super_or_sub_or_select_groupKeys[0]);

					// get all categories selected by user

	foreach($userSelections_array[$section][$super_or_sub_or_select_groupKeys[0]] as $category)
					{
						// get all intervals of each category
```

```
                                                $selectedIntervals =
filter_classificationToInterval($currentFacetName, $category);
                                                // add intervals to sub
                                                foreach($selectedIntervals as $interval){
$userSelections_array[$section][$string_sub][] = $interval; }
                                        }
                                        // delete current category array

        unset($userSelections_array[$section][$super_or_sub_or_select_groupKeys[0]]);
                                }
                                else if (strpos($super_or_sub_or_select_groupKeys[1], "super")) {
                                        // get corresponding sub name of super
                                        $string_sub =
str_replace("super","sub",$super_or_sub_or_select_groupKeys[1]);

                                        // get all categories selected by user

        foreach($userSelections_array[$section][$super_or_sub_or_select_groupKeys[1]] as $category)
                                        {
                                                // get all intervals of each category
                                                $selectedIntervals =
filter_classificationToInterval($currentFacetName, $category);
                                                // add intervals to sub
                                                foreach($selectedIntervals as $interval){
$userSelections_array[$section][$string_sub][] = $interval; }
                                        }
                                        // delete current category array

        unset($userSelections_array[$section][$super_or_sub_or_select_groupKeys[1]]);
                                }
                        }
                }

                // Format userSelections_array to a simpler array with dbColumnNames as key
                $userSelections = array();
                foreach($userSelections_array as $section => $array_of_sub_or_select_group){
                        // get dbcolumn name based on section name
                        $dbColumnName = section_facetName($section, true);

                        // loop through each intervalGroup (only 1 key-value pair)
                        foreach($array_of_sub_or_select_group as $sub_or_select => $intervalGroup)
                        { $userSelections[$dbColumnName] = $intervalGroup; }
                }

                /* Get Records based on user selections and aggregate for filters */
                $queryMatch = "";
                $collection = db_matchResults($userSelections);

                $filtered_facetArray_unique = array();
                if ($toSortUnique){             // sort by key
                        foreach ($facetInfoArray as $facetInfo)
                        {       $filtered_facetArray_unique[$facetInfo[0]] = db_matchResults_unique($facetInfo,
$queryMatch, true);}
                }
                else {                                                  // sort by count
                        foreach ($facetInfoArray as $facetInfo)
                        {       $filtered_facetArray_unique[$facetInfo[0]] = db_matchResults_unique($facetInfo,
$queryMatch, false);            }
                }

        }
        else if (empty($userSelections_array))
        {       /* aggregate facets and sort the unique arrays */
                $filtered_facetArray_unique = array();
                if ($toSortUnique){             // sort by key
                        foreach ($facetInfoArray as $facetInfo)
```

```
                        {       $filtered_facetArray_unique[$facetInfo[0]] = db_uniqueFacet_all($facetInfo, true);
        }
                }
                else {                                                  // sort by count
                        foreach ($facetInfoArray as $facetInfo)
                        {       $filtered_facetArray_unique[$facetInfo[0]] = db_uniqueFacet_all($facetInfo, false);
        }
                }

                include 'php-commons/collection.php';
        }

        /*
         * Convert Associative array into single dimension Indexed array, delimited by sep array
         * note: JSON automatically sort associative array by alphabetical order on keys,
         *                       which prevents using sorting with associative array before passing back to AJAX
         */
        $sep = array('--','|||','~','||','|');
        foreach($filtered_facetArray_unique as $facetName => $facetUnique_array)
        {
                $facet_joinedString = $facetName . $sep[0];

                foreach($facetUnique_array as $categoryOrValue => $intervalOrCount)
                {
                        $facet_joinedString .= empty($prefix_sep_1)? ":$prefix_sep_1;

                        // if $intervalOrCount is an array of intervals-count pairs
                        // section : category~ interval|count||interval|count  |||  category~ interval|count
                        if (is_array($intervalOrCount))
                        {
                                $facet_joinedString .= $categoryOrValue . $sep[2];

                                foreach($intervalOrCount as $interval => $count)
                                {
                                        $facet_joinedString .= empty($prefix_sep_2)? ":$prefix_sep_2;
                                        $facet_joinedString .= $interval . $sep[4] . $count;

                                        $prefix_sep_2 = $sep[3];
                                }
                                $prefix_sep_2 = "";
                        }
                        // if $intervalOrCount is only count
                        // section : value|count   |||   value|count
                        else
                        {
                                $facet_joinedString .= $categoryOrValue . $sep[4];
                                $facet_joinedString .= $intervalOrCount;
                        }

                        $prefix_sep_1 = $sep[1];
                }

                $indexed_filtered_facetArray_unique[] = $facet_joinedString;
                $prefix_sep_1 = "";
        }

        // pass the "sep-delimited" array back to AJAX
        echo json_encode(array($indexed_filtered_facetArray_unique,$collection));
?>
```

dev_recordCreate_OMDbAPI.php

```
<?php
	include 'php-commons/database.php';
	include 'php-commons/constants.php';
	include 'php-commons/filters.php';
	include 'php-commons/functions.php';

	/* Connect to database */
	$DB_connect = @new mysqli ($db_connect_var['host'], $db_connect_var['username'],
$db_connect_var['password'], $db_connect_var['database']);
	/* Database Connection Errors */
	if ($DB_connect->connect_errno > 0){ die();	}

	/* get json_encoded api_response from POST, and Decode it into an array */
	$record_array = json_decode($_POST['api_response'], true);

	/* API Input Validation */
	$inputValues = array();
	foreach ($dbInfoArray as $dbInfo)
	{
		// isCommaDelimited
		if ($dbInfo[5]){
			$inputValues[$dbInfo[1]] = str_replace(' ', '', $record_array[$dbInfo[0]]);
			if (count(explode(',', $inputValues[$dbInfo[1]])) > 5) {
				$cd_array = explode(',', $inputValues[$dbInfo[1]]);
				$inputValues[$dbInfo[1]] = '';
				$prefix = '';
				for ($cd_array_i = 0; $cd_array_i < 5; $cd_array_i++){
					$inputValues[$dbInfo[1]] .= (empty($prefix))?"":$prefix;
					$inputValues[$dbInfo[1]] .= $cd_array[$cd_array_i];
					$prefix = ',';
				}
			}
		}
		// isDate
		else if ($dbInfo[6]){ $inputValues[$dbInfo[1]] = date('d/m/Y', strtotime( $record_array[$dbInfo[0]] )); }
		// isNum
		else if ($dbInfo[7])
		{
			// default num
			if ( empty($record_array[$dbInfo[0]]) || ($record_array[$dbInfo[0]] == "N/A") )
			{ $inputValues[$dbInfo[1]] = 0; }
			// isNumSpaceSep
			else if ($dbInfo[8])
			{ $inputValues[$dbInfo[1]] = explode(" ", $record_array[$dbInfo[0]])[0]; }
			// isNumCommaSep
			else if ($dbInfo[9])
			{	// get rid of comma
				if( strpos($record_array[$dbInfo[0]], ',') !== false )
				{ $inputValues[$dbInfo[1]] = str_replace(',', '', $record_array[$dbInfo[0]]); }
				else { $inputValues[$dbInfo[1]] = $record_array[$dbInfo[0]]; }
			}
			else { $inputValues[$dbInfo[1]] = $record_array[$dbInfo[0]]; }
		}
		else { $inputValues[$dbInfo[1]] = $record_array[$dbInfo[0]]; }

		// intervals
		if ($dbInfo[2])
		{	// isDate
			if ($dbInfo[6]){ $inputValues[$dbInfo[3]] = filter_valueToInterval($dbInfo[4], explode('/',
$inputValues[ $dbInfo[1] ])[2]); }
			else { $inputValues[$dbInfo[3]] = filter_valueToInterval($dbInfo[4], $inputValues[
$dbInfo[1] ]); }
		}
	}
```

```
		/* Query : Insert Query String */
		// Query First Half (Columns)
		$prefix = '';
		$queryString = "INSERT INTO movie (";
		foreach ($inputValues as $key => $value){
			$queryString .= (empty($prefix))? ":$prefix;
			$queryString .= $key;
			$prefix = ', ';
		}
		$prefix = '';
		$queryString .= ")";
		// Query Second Half (Values)
		$queryString .= " VALUES (";
		foreach ($inputValues as $key => $value){
			$queryString .= (empty($prefix))? ":$prefix;
			$queryString .= "". $value . "";
			$prefix = ', ';
		}
		$queryString .= ")";

		// Execute Insert Query
		if ($DB_connect->query($queryString) === TRUE) { echo "true"; }
		else { echo "Error: " . $DB_connect->error;		}

		// Close database connection
		$DB_connect->close();
?>
```

dev_recordRefIDList_Form.php

```
<?php
		include 'php-commons/database.php';
		include 'php-commons/constants.php';
		include 'php-commons/filters.php';
		include 'php-commons/functions.php';

		/* Connect to database */
		$DB_connect = @new mysqli ($db_connect_var['host'], $db_connect_var['username'],
$db_connect_var['password'], $db_connect_var['database']);
		/* Database Connection Errors */
		if ($DB_connect->connect_errno > 0){ die();	}

		/* get RefID from POST */
		$refID = (inputNotEmpty($_POST['RefID']))? sanitize_input($_POST['RefID']) : null;
		// get json_encoded userSelections from POST, and Decode it into an array
		if (array_key_exists('userSelections', $_POST)){ $userSelections_array = json_decode($_POST['userSelections'],
true); }

		if (!empty($refID)){
			if (empty($userSelections_array)){
				/* Query : Get Details for Record specified by RefID */
				$query = "SELECT ";
				foreach ($dbInfoArray as $dbInfo){
					$query .= (empty($prefix))? ":$prefix;
					$query .= $dbInfo[1];
					$prefix = ', ';
				} $prefix = '';
				$query .= " FROM movie WHERE ".$dbName_searchRefID." LIKE '%".$refID."%'" LIMIT
10";

				if ($pQuery = $DB_connect->prepare($query)){
					$pQuery->execute();

					// my_sqli result Object
					$result = $pQuery->get_result();
					// array of records
					$table = $result->fetch_all();
```

```
                                        if(count($table) > 0){
                                                $listOfID = array();
                                                foreach($table as $record)
                                                { $listOfID[] = $record[1]; }

                                                echo json_encode($listOfID);
                                        } else { echo ""; }

                                        // Close Connection
                                        $pQuery->free_result();
                                        $pQuery->close();
                                }
                        }
                        else if (!empty($userSelections_array))
                        {       /*
                                 * standardize super and sub arrays into a single array (sub)
                                 */
                                foreach ($userSelections_array as $section => $array_of_super_or_sub_or_select_group)
                                {       // in each section, find which section it is so that the correct facetName can be
provided
                                        $currentFacetName = section_facetName($section);
                                        // count whether there is only select/super/sub, or there are both super and sub
                                        $super_or_sub_or_select_groupKeys =
array_keys($array_of_super_or_sub_or_select_group);

                                        // get corresponding intervals for each category and add it to the sub name's array
                                        if (count($super_or_sub_or_select_groupKeys) == 1) {
                                                if (strpos($super_or_sub_or_select_groupKeys[0], "super")) {
                                                        // get corresponding sub name of super
                                                        $string_sub =
str_replace("super","sub",$super_or_sub_or_select_groupKeys[0]);

                                                        // get all categories selected by user

        foreach($userSelections_array[$section][$super_or_sub_or_select_groupKeys[0]] as $category)
                                                        {
                                                                // get all intervals of each category
                                                                $selectedIntervals =
filter_classificationToInterval($currentFacetName, $category);
                                                                // add intervals to sub
                                                                foreach($selectedIntervals as $interval){
$userSelections_array[$section][$string_sub][] = $interval; }
                                                        }
                                                        // delete current category array

        unset($userSelections_array[$section][$super_or_sub_or_select_groupKeys[0]]);
                                                }
                                        }
                                        else if (count($super_or_sub_or_select_groupKeys) == 2) {
                                                if (strpos($super_or_sub_or_select_groupKeys[0], "super")) {
                                                        // get corresponding sub name of super
                                                        $string_sub =
str_replace("super","sub",$super_or_sub_or_select_groupKeys[0]);

                                                        // get all categories selected by user

        foreach($userSelections_array[$section][$super_or_sub_or_select_groupKeys[0]] as $category)
                                                        {
                                                                // get all intervals of each category
                                                                $selectedIntervals =
filter_classificationToInterval($currentFacetName, $category);
                                                                // add intervals to sub
                                                                foreach($selectedIntervals as $interval){
$userSelections_array[$section][$string_sub][] = $interval; }
                                                        }
                                                        // delete current category array
```

```
	unset($userSelections_array[$section][$super_or_sub_or_select_groupKeys[0]]);
                                        }
                                        else if (strpos($super_or_sub_or_select_groupKeys[1], "super")) {
                                                // get corresponding sub name of super
                                                $string_sub =
str_replace("super","sub",$super_or_sub_or_select_groupKeys[1]);

                                                // get all categories selected by user

	foreach($userSelections_array[$section][$super_or_sub_or_select_groupKeys[1]] as $category)
                                                {
                                                        // get all intervals of each category
                                                        $selectedIntervals =
filter_classificationToInterval($currentFacetName, $category);
                                                        // add intervals to sub
                                                        foreach($selectedIntervals as $interval){
$userSelections_array[$section][$string_sub][] = $interval; }
                                                }
                                                // delete current category array

	unset($userSelections_array[$section][$super_or_sub_or_select_groupKeys[1]]);
                                        }
                                }
                        }

                        // Format userSelections_array to a simpler array with dbColumnNames as key
                        $userSelections = array();
                        foreach($userSelections_array as $section => $array_of_sub_or_select_group){
                                // get dbcolumn name based on section name
                                $dbColumnName = section_facetName($section, true);

                                // loop through each intervalGroup (only 1 key-value pair)
                                foreach($array_of_sub_or_select_group as $sub_or_select => $intervalGroup)
                                { $userSelections[$dbColumnName] = $intervalGroup; }
                        }

                        /* Query : Get Details for Record specified by RefID AND userSelections */
                        $query = "SELECT ";
                        foreach ($dbInfoArray as $dbInfo){
                                $query .= (empty($prefix))? ":$prefix;
                                $query .= $dbInfo[1];
                                $prefix = ', ';
                        } $prefix = '';
                        $query .= " FROM movie WHERE (".$dbName_searchRefID." LIKE '%".$refID."%') AND ";

                        foreach($userSelections as $section => $intervalGroup){
                                $query .= empty($prefixAND)? "":$prefixAND;
                                $query .= "(";

                                // get each interval
                                foreach ($intervalGroup as $interval)
                                {       // check if column is comma delimited
                                        if (!(isDBColumn_commaDelimited($section)))
                                        {
                                                $query .= empty($prefixOR)? "":$prefixOR;
                                                $query .= $section ." = '". $interval ."'";

                                                // OR
                                                $prefixOR = ' OR ';
                                        }
                                        else if (isDBColumn_commaDelimited($section))
                                        {
                                                $query .= empty($prefixOR)? "":$prefixOR;
                                                $query .= $section ." LIKE '%". $interval ."%'";
```

```
                                                // OR (changed to AND for dbColumns with multiple values
(comma-delimited)
                                                $prefixOR = ' AND ';
                                        }
                                }
                                $query .= ")";

                                // AND
                                $prefixAND = " AND ";
                                $prefixOR = "";
                        }

                        if ($pQuery = $DB_connect->prepare($query)){
                                $pQuery->execute();

                                // my_sqli result Object
                                $result = $pQuery->get_result();
                                // array of records
                                $table = $result->fetch_all();

                                if(count($table) > 0){
                                        $listOfID = array();
                                        foreach($table as $record)
                                        { $listOfID[] = $record[1]; }

                                        echo json_encode($listOfID);
                                } else { echo ""; }

                                // Close Connection
                                $pQuery->free_result();
                                $pQuery->close();
                        }
                }
        } else { echo ""; }

        // Close database connection
        $DB_connect->close();
?>
```

dev_recordTitleList_Form.php

```
<?php
        include 'php-commons/database.php';
        include 'php-commons/constants.php';
        include 'php-commons/filters.php';
        include 'php-commons/functions.php';

        /* Connect to database */
        $DB_connect = @new mysqli ($db_connect_var['host'], $db_connect_var['username'],
$db_connect_var['password'], $db_connect_var['database']);
        /* Database Connection Errors */
        if ($DB_connect->connect_errno > 0){ die();     }

        /* get RefID from POST */
        $refID = (inputNotEmpty($_POST['RefID']))? sanitize_input($_POST['RefID']) : null;
        // get json_encoded userSelections from POST, and Decode it into an array
        if (array_key_exists('userSelections', $_POST)){ $userSelections_array = json_decode($_POST['userSelections'],
true); }

        if (!empty($refID)){
                if (empty($userSelections_array)){
                        /* Query : Get Details for Record specified by RefID */
                        $query = "SELECT ";
                        foreach ($dbInfoArray as $dbInfo){
                                $query .= (empty($prefix))? ":$prefix;
                                $query .= $dbInfo[1];
                                $prefix = ', ';
```

```
                        } $prefix = '';
                        $query .= " FROM movie WHERE ".$dbName_searchTitle." LIKE '%".$refID."%'";
                        $query .= " ORDER BY ".$dbName_searchTitle." ASC LIMIT 10";

                        if ($pQuery = $DB_connect->prepare($query)){
                                $pQuery->execute();

                                // my_sqli result Object
                                $result = $pQuery->get_result();
                                // array of records
                                $table = $result->fetch_all();

                                if(count($table) > 0){
                                        $listOfID = array();
                                        foreach($table as $record)
                                        { $listOfID[] = $record[0]; }

                                        echo json_encode($listOfID);
                                } else { echo ""; }

                                // Close Connection
                                $pQuery->free_result();
                                $pQuery->close();
                        }
                }
                else if (!empty($userSelections_array))
                {       /*
                         * standardize super and sub arrays into a single array (sub)
                         */
                        foreach ($userSelections_array as $section => $array_of_super_or_sub_or_select_group)
                        {       // in each section, find which section it is so that the correct facetName can be
provided
                                $currentFacetName = section_facetName($section);
                                // count whether there is only select/super/sub, or there are both super and sub
                                $super_or_sub_or_select_groupKeys =
array_keys($array_of_super_or_sub_or_select_group);

                                // get corresponding intervals for each category and add it to the sub name's array
                                if (count($super_or_sub_or_select_groupKeys) == 1) {
                                        if (strpos($super_or_sub_or_select_groupKeys[0], "super")) {
                                                // get corresponding sub name of super
                                                $string_sub =
str_replace("super","sub",$super_or_sub_or_select_groupKeys[0]);

                                                // get all categories selected by user

        foreach($userSelections_array[$section][$super_or_sub_or_select_groupKeys[0]] as $category)
                                                {
                                                        // get all intervals of each category
                                                        $selectedIntervals =
filter_classificationToInterval($currentFacetName, $category);
                                                        // add intervals to sub
                                                        foreach($selectedIntervals as $interval){
$userSelections_array[$section][$string_sub][] = $interval; }
                                                }
                                                // delete current category array

        unset($userSelections_array[$section][$super_or_sub_or_select_groupKeys[0]]);
                                        }
                                }
                                else if (count($super_or_sub_or_select_groupKeys) == 2) {
                                        if (strpos($super_or_sub_or_select_groupKeys[0], "super")) {
                                                // get corresponding sub name of super
                                                $string_sub =
str_replace("super","sub",$super_or_sub_or_select_groupKeys[0]);

                                                // get all categories selected by user
```

```
		foreach($userSelections_array[$section][$super_or_sub_or_select_groupKeys[0]] as $category)
{
// get all intervals of each category
$selectedIntervals = filter_classificationToInterval($currentFacetName, $category);
// add intervals to sub
foreach($selectedIntervals as $interval){ $userSelections_array[$section][$string_sub][] = $interval; }
}
// delete current category array
		unset($userSelections_array[$section][$super_or_sub_or_select_groupKeys[0]]);
}
else if (strpos($super_or_sub_or_select_groupKeys[1], "super")) {
// get corresponding sub name of super
$string_sub = str_replace("super","sub",$super_or_sub_or_select_groupKeys[1]);

// get all categories selected by user
		foreach($userSelections_array[$section][$super_or_sub_or_select_groupKeys[1]] as $category)
{
// get all intervals of each category
$selectedIntervals = filter_classificationToInterval($currentFacetName, $category);
// add intervals to sub
foreach($selectedIntervals as $interval){ $userSelections_array[$section][$string_sub][] = $interval; }
}
// delete current category array
		unset($userSelections_array[$section][$super_or_sub_or_select_groupKeys[1]]);
}
	}
}

// Format userSelections_array to a simpler array with dbColumnNames as key
$userSelections = array();
foreach($userSelections_array as $section => $array_of_sub_or_select_group){
	// get dbcolumn name based on section name
	$dbColumnName = section_facetName($section, true);

	// loop through each intervalGroup (only 1 key-value pair)
	foreach($array_of_sub_or_select_group as $sub_or_select => $intervalGroup)
	{ $userSelections[$dbColumnName] = $intervalGroup; }
}

/* Query : Get Details for Record specified by RefID AND userSelections */
$query = "SELECT ";
foreach ($dbInfoArray as $dbInfo){
	$query .= (empty($prefix))? ":$prefix;
	$query .= $dbInfo[1];
	$prefix = ', ';
} $prefix = '';
$query .= " FROM movie WHERE (".$dbName_searchTitle." LIKE '%".$refID."%') AND ";

foreach($userSelections as $section => $intervalGroup){
	$query .= empty($prefixAND)? "":$prefixAND;
	$query .= "(";

	// get each interval
	foreach ($intervalGroup as $interval)
	{	// check if column is comma delimited
		if (!(isDBColumn_commaDelimited($section)))
		{
			$query .= empty($prefixOR)? "":$prefixOR;
```

```
                                        $query .= $section ." = '". $interval ."'";

                                        // OR
                                        $prefixOR = ' OR ';
                                }
                                else if (isDBColumn_commaDelimited($section))
                                {
                                        $query .= empty($prefixOR)? "":$prefixOR;
                                        $query .= $section ." LIKE '%". $interval ."%'";

                                        // OR (changed to AND for dbColumns with multiple values
(comma-delimited)
                                        $prefixOR = ' AND ';
                                }
                        }
                        $query .= ")";

                        // AND
                        $prefixAND = " AND ";
                        $prefixOR = "";
                }
                $query .= " ORDER BY ".$dbName_searchTitle." ASC LIMIT 10";

                if ($pQuery = $DB_connect->prepare($query)){
                        $pQuery->execute();

                        // my_sqli result Object
                        $result = $pQuery->get_result();
                        // array of records
                        $table = $result->fetch_all();

                        if(count($table) > 0){
                                $listOfID = array();
                                foreach($table as $record)
                                { $listOfID[] = $record[0]; }

                                echo json_encode($listOfID);
                        } else { echo ""; }

                        // Close Connection
                        $pQuery->free_result();
                        $pQuery->close();
                }
        }
} else { echo ""; }

// Close database connection
$DB_connect->close();
?>
```

dev_recordInfo_Form.php

```
<?php
        include 'php-commons/database.php';
        include 'php-commons/constants.php';
        include 'php-commons/filters.php';
        include 'php-commons/functions.php';

        /* Connect to database */
        $DB_connect = @new mysqli ($db_connect_var['host'], $db_connect_var['username'],
$db_connect_var['password'], $db_connect_var['database']);
        /* Database Connection Errors */
        if ($DB_connect->connect_errno > 0){ die();     }

        /* get RefID from POST */
        $refID = (inputNotEmpty($_POST['RefID']))? sanitize_input($_POST['RefID']) : null;
```

```
	if (!empty($refID)){
		/* Query : Get Details for Record specified by RefID */
		$query = "SELECT ";
		foreach ($dbInfoArray as $dbInfo){
			$query .= (empty($prefix))? ":$prefix;
			$query .= $dbInfo[1];
			$prefix = ', ';
		} $prefix = '';
		$query .= " FROM movie WHERE ".$dbName_searchRefID."='".$refID."'";

		if ($pQuery = $DB_connect->prepare($query)){
			$pQuery->execute();

			// my_sqli result Object
			$result = $pQuery->get_result();
			// array of records
			$table = $result->fetch_all();

			if(!empty($table[0])){

				$record = array();
				for ($dbi_i = 0; $dbi_i < count($dbInfoArray); $dbi_i++)
				{ $record[$dbInfoArray[$dbi_i][4]] = $table[0][$dbi_i]; }

				echo json_encode($record);
			}
			else { echo ""; }

			// Close Connection
			$pQuery->free_result();
			$pQuery->close();
		}
	} else { echo ""; }

	// Close database connection
	$DB_connect->close();
?>
```

dev_downloadLog.php

```
<?php
	include 'php-commons/functions.php';
	include 'php-commons/database.php';

	/* Get file name of the file to download */
	if (array_key_exists("fileToDownload", $_POST)) {
		$fileToDownload = (!empty($_POST['fileToDownload']))? sanitize_input($_POST['fileToDownload']) :
null;

		// check if file name is not empty, and file is readable
		if (!empty($fileToDownload)){
			if(is_readable($dir_log_folder . '/' . $fileToDownload))
			{	// build download document
				header ("Content-type: text/plain");
				header ("Content-Disposition: attachment; filename='$fileToDownload'");

				// place file in download document
				readfile($dir_log_folder . '/' . $fileToDownload);
				// redirect back to previous page
			} else {
				?>

				<!DOCTYPE html>
				<html>
					<head>
						<meta charset="UTF-8"/>
						<title>File Download Error</title>
```

```
                                        </head>
                                        <body>
                                                <p>ERROR: <?php echo $fileToDownload ?> does not exist, or
is not readable.</p>
                                                <a href="assignment6dev.php">Go Back</a>
                                        </body>
                                </html>

                                <?php
                        }
                } else {
                        // Invalid File Name (empty)
                        ?>

                        <!DOCTYPE html>
                        <html>
                                <head>
                                        <meta charset="UTF-8"/>
                                        <title>File Download Error</title>
                                </head>
                                <body>
                                        <p>ERROR: Invalid Name</p>
                                        <a href="assignment6dev.php">Go Back</a>
                                </body>
                        </html>

                        <?php
                }
        } else {
                // Invalid POST
                ?>

                <!DOCTYPE html>
                <html>
                        <head>
                                <meta charset="UTF-8"/>
                                <title>File Download Error</title>
                        </head>
                        <body>
                                <p>ERROR: Invalid Path</p>
                                <a href="assignment6dev.php">Go Back</a>
                        </body>
                </html>

                <?php
        }
?>
```

Detailed PHP Code (Index & PHP-Commons)

Index.php

```
<?php
	header('Location: /webpage_user.php');
?>
```

Database.php

```
<?php
	/* Root Directory */
	$dir_root = realpath($_SERVER["DOCUMENT_ROOT"]);

	/* Log Directory & File Name */
	$dir_log_folder = "log";
	$logFileName = "movie_log.txt";

	$db_connect_var = array(
		'host' => 'localhost',
		'username' => 'adminU',
		'password' => 'adminUser',
		'database' => 'php_assessment_major7'
	);
?>
```

Constants.php

```
<?php
	// default text string
	$lockText_default = "unlock";

	/*
	 * Strings for column names of the collection
	 */
	// facet name, db name, isCategory, isCommaDelimited, isDate
	$facetInfoArray = array(
		array('releasedDate', 'released_date_interval'	, true	, false	, true),
		array('movieType'		, 'movie_type'				, false	, false	, false),
		array('contentRating'	, 'content_rating'			, false	, false	, false),
		array('movieGenre'		, 'movie_genre'				, false	, true	, false),
		array('totalSeason'		, 'total_season_interval'	, true	, false	, false),
		array('movieDuration'	, 'movie_duration_interval'	, true	, false	, false),
		array('movieLanguage'	, 'movie_language'			, false	, true	, false),
		array('movieCountry'	, 'movie_country'			, false	, true	, false),
		array('metaScore'		, 'meta_score_interval'		, true	, false	, false),
		array('imdbRating'		, 'imdb_rating_interval'	, true	, false	, false),
		array('imdbVotes'		, 'imdb_votes_interval'		, true	, false	, false)
	);

	$facetName_releasedDate = $facetInfoArray[0][0];
	$facetName_totalSeason = $facetInfoArray[4][0];
	$facetName_movieDuration = $facetInfoArray[5][0];
	$facetName_metaScore = $facetInfoArray[8][0];
	$facetName_imdbRating = $facetInfoArray[9][0];
	$facetName_imdbVotes = $facetInfoArray[10][0];

	// imdbRefID db column name for querying in update form
	$dbName_searchTitle = "movie_title";
	$dbName_searchRefID = "imdb_Ref_ID";

	// db name, interval db name, facet name
	$valueAndIntervalArray = array(
```

```
                array('released_date'           , 'released_date_interval'      , 'releasedDate'),
                array('total_season' , 'total_season_interval'      , 'totalSeason'),
                array('movie_duration'          , 'movie_duration_interval'     , 'movieDuration'),
                array('meta_score'              , 'meta_score_interval'                 , 'metaScore'),
                array('imdb_rating'             , 'imdb_rating_interval'        , 'imdbRating'),
                array('imdb_votes'              , 'imdb_votes_interval'                 , 'imdbVotes')
        );

        /*
         * Array for Form validation
         */
        // (0-4)    api name, db name, isCategory/hasInterval, interval db name, facet name, ...
        // (5-9)    ...booleans : isCommaDelimited, isDate, isNum, isNumSpaceSep, isNumCommaSep, ...
        // (10-14) .............isInt, isFloat, hasRange, min, max
        $dbInfoArray = array(
                array('Title'                           , 'movie_title'                 , false     , ''
                                , 'movieTitle'
                        , false     , false     , false     , false     , false
                        , false     , false     , false     , 0                 , 0
                ),
                array('imdbID'                          , 'imdb_Ref_ID'             , false     , ''
                                , 'imdbRefID'
                        , false     , false     , false     , false     , false
                        , false     , false     , false     , 0                 , 0
                ),
                array('Released'                , 'released_date'       , true      , 'released_date_interval'      ,
$facetInfoArray[0][0]
                        , false     , true      , false     , false     , false
                        , false     , false     , false     , 0                 , 0
                ),
                array('Type'                            , 'movie_type'              , false     , ''
                                , $facetInfoArray[1][0]
                        , false     , false     , false     , false     , false
                        , false     , false     , false     , 0                 , 0
                ),
                array('Rated'                           , 'content_rating'      , false     , ''
                        , $facetInfoArray[2][0]
                        , false     , false     , false     , false     , false
                        , false     , false     , false     , 0                 , 0
                ),
                array('Genre'                           , 'movie_genre'             , false     , ''
                                , $facetInfoArray[3][0]
                        , true      , false     , false     , false     , false
                        , false     , false     , false     , 0                 , 0
                ),
                array('totalSeasons' , 'total_season'       , true      , 'total_season_interval'       , $facetInfoArray[4][0]

                        , false     , false     , true      , false     , false
                        , true      , false     , false     , 0                 , 0
                ),
                array('Runtime'                         , 'movie_duration'  , true      , 'movie_duration_interval'     ,
$facetInfoArray[5][0]
                        , false     , false     , true      , true      , false
                        , true      , false     , false     , 0                 , 0
                ),
                array('Language'                , 'movie_language'  , false     , ''
                , $facetInfoArray[6][0]
                        , true      , false     , false     , false     , false
                        , false     , false     , false     , 0                 , 0
                ),
                array('Country'                         , 'movie_country'   , false     , ''
                        , $facetInfoArray[7][0]
                        , true      , false     , false     , false     , false
                        , false     , false     , false     , 0                 , 0
                ),
                array('Metascore'               , 'meta_score'                  , true      , 'meta_score_interval'
        , $facetInfoArray[8][0]
```

```
                        , false     , false     , true      , false     , false
                        , true      , false     , true      , 0                 , 100
                ),
                array('imdbRating'          , 'imdb_rating'             , true      , 'imdb_rating_interval'        ,
$facetInfoArray[9][0]
                        , false     , false     , true      , false     , false
                        , false     , true      , true      , 0                 , 10
                ),
                array('imdbVotes'           , 'imdb_votes'              , true      , 'imdb_votes_interval'
        , $facetInfoArray[10][0]
                        , false     , false     , true      , false     , true
                        , true      , false     , false     , 0                 , 0
                )
        );
?>
```

Filters.php

```
<?php
        /*
         * Associative array of filters classification for multiple facets
         */
        $filterClassification = array(
                $facetName_releasedDate      => array(
                        'interval'                      => array(
                                'unknown'                       => array(
                                        array(
                                                array('0 - 1970', -1)
                                        ), -1
                                ),
                                'long ago'              => array(
                                        array(
                                                array('1970+ - 1980', 1970),
                                                array('1980+ - 1990', 1980),
                                                array('1990+ - 2000', 1990)
                                        ), 1970
                                ),
                                'some time ago'=> array(
                                        array(
                                                array('2000+ - 2005', 2000),
                                                array('2005+ - 2010', 2005)
                                        ), 2000
                                ),
                                'recent'                => array(
                                        array(
                                                array('2010+ - 2015', 2010),
                                                array('2015+', 2015)
                                        ), 2010
                                )
                        )
                ),
                $facetName_totalSeason       => array(
                        'interval'                      => array(
                                'short series'          => array(
                                        array(
                                                array('0 - 1', -1),
                                                array('1+ - 3', 1)
                                        ), -1
                                ),
                                'average series'=> array(
                                        array(
                                                array('3+ - 5', 3),
                                                array('5+ - 10', 5)
                                        ), 3
                                ),
                                'long series'           => array(
                                        array(
```

```
                                        array('10+ - 25', 10),
                                        array('25+', 25)
                                ), 10
                        )
                )
        ),
        $facetName_movieDuration => array(
                'interval'                      => array(
                        'short'                 => array(
                                array(
                                        array('0 - 2', -1),
                                        array('2+ - 5', 2),
                                        array('5+ - 7', 5),
                                        array('7+ - 10', 7)
                                ), -1
                        ),
                        'medium'                => array(
                                array(
                                        array('10+ - 20', 10),
                                        array('20+ - 30', 20)
                                ), 10
                        ),
                        'long'                  => array(
                                array(
                                        array('30+ - 60', 30),
                                        array('60+ - 120', 60)
                                ), 30
                        ),
                        'very long'             => array(
                                array(
                                        array('120+', 60)
                                ), 120
                        )
                )
        ),
        $facetName_metaScore            => array(
                'interval'                      => array(
                        'terrible'              => array(
                                array(
                                        array('0 - 10', -1),
                                        array('10+ - 25', 10)
                                ), -1
                        ),
                        'bad'                                   => array(
                                array(
                                        array('25+ - 35', 25),
                                        array('35+ - 50', 35)
                                ), 25
                        ),
                        'average'               => array(
                                array(
                                        array('50+ - 65', 50)
                                ), 50
                        ),
                        'good'                                  => array(
                                array(
                                        array('65+ - 75', 65),
                                        array('75+ - 85', 75)
                                ), 65
                        ),
                        'excellent'=> array(
                                array(
                                        array('85+ - 95', 85),
                                        array('95+ - 100', 95)
                                ), 85
                        )
                )
```

```
),
$facetName_imdbRating       => array(
        'interval'                      => array(
                'very low'              => array(
                        array(
                                array('0.0 - 1.0', -0.1),
                                array('1.0+ - 2.5', 1.0)
                        ), -0.1
                ),
                'bad'                           => array(
                        array(
                                array('2.5+ - 3.5', 2.5),
                                array('3.5+ - 5.0', 3.5)
                        ), 2.5
                ),
                'average'               => array(
                        array(
                                array('5.0+ - 6.5', 5.0)
                        ), 5.0
                ),
                'good'                  => array(
                        array(
                                array('6.5+ - 7.5', 6.5),
                                array('7.5+ - 8.5', 7.5)
                        ), 6.5
                ),
                'excellent'             => array(
                        array(
                                array('8.5+ - 9.5', 8.5),
                                array('9.5+ - 10.0', 9.5)
                        ), 8.5
                )
        )
),
$facetName_imdbVotes => array(
        'interval'                      => array(
                'very low vote'=> array(
                        array(
                                array('0 - 50', -1),
                                array('50+ - 250', 50)
                        ), -1
                ),
                'low vote'              => array(
                        array(
                                array('250+ - 500', 250),
                                array('500+ - 750', 500),
                                array('750+ - 1,000', 750)
                        ), 250
                ),
                'average vote' => array(
                        array(
                                array('1,000+ - 2,500', 1000),
                                array('2,500+ - 5,000', 2500),
                                array('5,000+ - 10,000', 5000)
                        ), 1000
                ),
                'high vote'             => array(
                        array(
                                array('10,000+ - 25,000', 10000),
                                array('25,000+ - 50,000', 25000),
                                array('50,000+ - 100,000', 50000)
                        ), 10000
                ),
                'very high vote'=> array(
                        array(
                                array('100,000+ - 500,000', 100000),
                                array('500,000+ - 1,000,000', 500000),
```

```
                                                array('1,000,000+', 1000000)
                                        ), 100000
                                )
                        )
                )
        );
?>
```

Functions.php

```
<?php
        /*
         * Function : sanitize_input
         * Usage    : clear whitespaces, slashes, and any special characters
         * Input    : string to be sanitized
         * Return   : sanitized value
         */
        function sanitize_input($user_input){
                $user_input = trim($user_input);
                $user_input = htmlspecialchars($user_input);
                $user_input = stripslashes($user_input);
                return $user_input;
        }

        /*
         * Function : inputNotEmpty
         * Usage    : check if input is empty
         *                      (to handle input with value of 0, as 0 is considered as empty)
         * Input    : input string to check
         * Return   : boolean true/false, true if not empty, false if empty and not 0
         */
        function inputNotEmpty($input){
                return ($input === "0" || $input);
        }

        /*
         * Function : section_facetName
         * Usage    : find facet name (or dbcolumn name) based on section name
         *                      that has facet name as the substring
         * Input    : section name
         * Return   : facet name (or dbcolumn name)
         */
        function section_facetName($section, $facetName_dbColumn = false)
        {       global $facetInfoArray;
                foreach ($facetInfoArray as $facetInfo)
                {       // return facetName or dbColumn name
                        if (strpos($section, $facetInfo[0]) != false){
                                if (!$facetName_dbColumn){ return $facetInfo[0]; }
                                else if ($facetName_dbColumn){ return $facetInfo[1]; }
                        }
                }
        }

        /*
         * Function : isDBColumn_commaDelimited
         * Usage    : check whether the current column is comma delimited (have multiples, e.g. genre)
         * Input    : DBColumn
         * Return   : facet name (or dbcolumn name)
         */
        function isDBColumn_commaDelimited($dbColumn)
        {       global $facetInfoArray;
                foreach ($facetInfoArray as $facetInfo)
                {       if ($dbColumn == $facetInfo[1])
                        {       // commaDelimited bool = true, for the dbColumn
                                if ($facetInfo[3])
                                { return true; }
                        }
```

```
		}
		return false;
	}

	/*
	 * Function : filter_classificationToInterval
	 * Usage    : get array of intervals for the classification/category from filterClassification array
	 * Input    : facet name and category
	 * Return   : array of intervals
	 */
	function filter_classificationToInterval($targetFacet, $targetCategory)
	{	global $filterClassification;

		$selectedIntervals = array();
		foreach($filterClassification[$targetFacet]['interval'][$targetCategory][0] as $interval_limit)
		{ $selectedIntervals[] = $interval_limit[0]; }

		return $selectedIntervals;
	}

	/*
	 * Function : filter_intervalToClassification
	 * Usage    : get name of facet, and interval to traverse filterClassification array to find classification
	 * Input    : facet name and interval
	 * Return   : classification
	 */
	function filter_intervalToClassification($targetFacet, $targetInterval)
	{	global $filterClassification;

		foreach (array_reverse($filterClassification[$targetFacet]['interval']) as $classification =>
$intervalGroup){
			foreach (array_reverse($intervalGroup[0]) as $interval){
				if ($interval[0] == $targetInterval)
				{ return $classification; }
			}
		}
		return "";
	}

	/*
	 * Function : filter_valueToInterval
	 * Usage    : get name of facet, and value to find interval in filterClassification array
	 * Input    : facet name and interval
	 * Return   : classification
	 */
	function filter_valueToInterval($targetFacet, $targetValue)
	{	global $filterClassification;

		foreach (array_reverse($filterClassification[$targetFacet]['interval']) as $classification =>
$intervalGroup){
			if ($targetValue > $intervalGroup[1]){
				foreach (array_reverse($intervalGroup[0]) as $interval){
					if ($targetValue > $interval[1])
					{ return $interval[0]; }
		}	}	}
		return "";
	}

	/*
	 * Function : db_updateIntervals
	 * Usage    : get the refID to update all interval values of a record
	 * Input    : imdbRefID
	 * Return   : none
	 */
	function db_updateIntervals($RefID = "")
	{	global $db_connect_var;
		global $dbName_searchRefID;
```

```
		global $valueAndIntervalArray;

		// Connect to database
		$DB_connect = @new mysqli ($db_connect_var['host'], $db_connect_var['username'],
$db_connect_var['password'], $db_connect_var['database']);
		// Database Connection Errors
		if ($DB_connect->connect_errno > 0){ die();	}

		if (inputNotEmpty($RefID)){
			$prefix = '';
			$query = "SELECT ";
			foreach ($valueAndIntervalArray as $valueIntervalPair){
				$query .= (empty($prefix))? ":$prefix;
				$query .= $valueIntervalPair[0];
				$prefix = ', ';
			}
			$query .= " FROM movie WHERE ".$dbName_searchRefID."='".$RefID."'";
			if ($pQuery = $DB_connect->prepare($query)){
				$pQuery->execute();

				// my_sqli result Object
				$result = $pQuery->get_result();
				// array of records
				$table = $result->fetch_all();

				// get intervals for the selected record
				$intervalsArray = array();
				$intervalsArray[] = filter_valueToInterval($valueAndIntervalArray[0][2],
explode('/', $table[0][0])[2]);
				for ($vip_i = 1; $vip_i < count($valueAndIntervalArray); $vip_i++)
				{	// pass facet name, record #1's value to function
					$intervalsArray[] =
filter_valueToInterval($valueAndIntervalArray[$vip_i][2],$table[0][$vip_i]);
				}

				// update the record's intervals
				$prefix = '';
				$queryString = "UPDATE movie SET ";
				for ($vip_j = 0; $vip_j < count($valueAndIntervalArray); $vip_j++){
					$queryString .= (empty($prefix))? ":$prefix;
					$queryString .=
$valueAndIntervalArray[$vip_j][1]."='".$intervalsArray[$vip_j]."'";
					$prefix = ', ';
				}
				$queryString .= " WHERE ".$dbName_searchRefID."='".$RefID."'";

				// Execute Update Query
				if ($DB_connect->query($queryString) === TRUE){ }
			}
		} // Default to update all records if no specific ID provided
		else {
			$prefix = '';
			$query = "SELECT ".$dbName_searchRefID.", ";
			foreach ($valueAndIntervalArray as $valueIntervalPair){
				$query .= (empty($prefix))? ":$prefix;
				$query .= $valueIntervalPair[0];
				$prefix = ', ';
			}
			$query .= " FROM movie";
			if ($pQuery = $DB_connect->prepare($query)){
				$pQuery->execute();

				// my_sqli result Object
				$result = $pQuery->get_result();
				// array of records
				$table = $result->fetch_all();
```

```
                                        foreach ($table as $record)
                                        {       // RefID
                                                $recordRefID = $record[0];

                                                // get intervals for the selected record
                                                $intervalsArray = array();
                                                $intervalsArray[] = filter_valueToInterval($valueAndIntervalArray[0][2],
explode('/', $record[1])[2]);
                                                for ($vip_i = 1; $vip_i < count($valueAndIntervalArray); $vip_i++)
                                                {       // pass facet name, record #1's value to function
                                                        $intervalsArray[] =
filter_valueToInterval($valueAndIntervalArray[$vip_i][2],$record[$vip_i + 1]);
                                                }

                                                // update the record's intervals
                                                $prefix = '';
                                                $queryString = "UPDATE movie SET ";
                                                for ($vip_j = 0; $vip_j < count($valueAndIntervalArray); $vip_j++){
                                                        $queryString .= (empty($prefix))? ":$prefix;
                                                        $queryString .=
$valueAndIntervalArray[$vip_j][1]."='".$intervalsArray[$vip_j]."'";
                                                        $prefix = ', ';
                                                }
                                                $queryString .= " WHERE
".$dbName_searchRefID."='".$recordRefID."'";

                                                // Execute Update Query
                                                if ($DB_connect->query($queryString) === TRUE){ }
                                        }
                                }
                        }

                        // Close database connection
                        $DB_connect->close();
                }

                /*
                 * Function : db_uniqueFacet_all
                 * Usage    : aggregate a facet to display
                 * Input    : facetInfo containing info of the facet, and sort order
                 * Return   : aggregated and sorted array of unique values of the facet
                 */
                function db_uniqueFacet_all($facetInfo, $sortOrderDefault = true)
                {       global $db_connect_var;

                        /* Connect to database */
                        $DB_connect = @new mysqli ($db_connect_var['host'], $db_connect_var['username'],
$db_connect_var['password'], $db_connect_var['database']);
                        /* Database Connection Errors */
                        if ($DB_connect->connect_errno > 0){ die();     }

                        /* Query : Get Unique Facets */
                        // if not comma delimited
                        if (!$facetInfo[3]){
                                $query = "SELECT ".$facetInfo[1].", COUNT(".$facetInfo[1].") total".
                                                " FROM movie".
                                                " GROUP BY ".$facetInfo[1];
                                if ($sortOrderDefault){ $query .= " ORDER BY ".$facetInfo[1]. " ASC"; }
                                else if (!$sortOrderDefault){ $query .= " ORDER BY total DESC"; }
                        } // if it is comma delimited
                        else if ($facetInfo[3]){
                                $query = "SELECT cd, COUNT(cd) total".
                                                " FROM (".
                                                " SELECT
SUBSTRING_INDEX(SUBSTRING_INDEX(".$facetInfo[1].", ',', movies.n), ',', -1) cd".
                                                " FROM (SELECT 1 n union all SELECT 2 union all SELECT 3 union
all SELECT 4 union all SELECT 5) movies".
```

```
                                                " INNER JOIN movie ON CHAR_LENGTH(".$facetInfo[1].")-
CHAR_LENGTH(REPLACE(".$facetInfo[1].", ',', ''))>=movies.n-1)".
                                                " AS x".
                                                " GROUP BY cd";
                                if ($sortOrderDefault){ $query .= " ORDER BY cd ASC"; }
                                else if (!$sortOrderDefault){ $query .= " ORDER BY total DESC"; }
                        }

                        // Execute Query
                        $uniqueArray = array();
                        if ($pQuery = $DB_connect->prepare($query)){
                                $pQuery->execute();

                                // my_sqli result Object
                                $result = $pQuery->get_result();
                                // array of records
                                $table = $result->fetch_all();

                                // no categories
                                if (!$facetInfo[2]){
                                        foreach($table as $record)
                                        {       $uniqueArray[$record[0]] = $record[1];          }
                                }
                                // has categories
                                else if ($facetInfo[2]){
                                        foreach($table as $record)
                                        {       $uniqueArray[ filter_intervalToClassification($facetInfo[0], $record[0])
][$record[0]] = $record[1];     }

                                        // sort categories
                                        if ($sortOrderDefault){ ksort($uniqueArray); }
                                        else if (!$sortOrderDefault){
                                                $array_of_categoryCount = array();
                                                foreach ($uniqueArray as $category => $intervalGroup)
                                                {       // get total count for each category
                                                        $totalCategoryCount = 0;
                                                        foreach($intervalGroup as $interval => $count)
                                                        {       $totalCategoryCount += $count;          }
                                                        $array_of_categoryCount[$category] = $totalCategoryCount;
                                                }
                                                // sort category-totalcount array based on total count
                                                arsort($array_of_categoryCount);

                                                // pass the sorted intervalGroup array to array_of_categoryCount of the
same key
                                                foreach($array_of_categoryCount as $category => $totalCategoryCount)
                                                { $array_of_categoryCount[$category] = $uniqueArray[$category]; }
                                                // overwrite uniqueArray with sorted array of category-intervals
                                                $uniqueArray = $array_of_categoryCount;
                                        }
                                }

                                // Free Results
                                $pQuery->free_result();
                                $pQuery->close();
                        } else {    echo $DB_connect->error;    }

                        // Close database connection
                        $DB_connect->close();

                        return $uniqueArray;
                }

                /*
                 * Function : db_matchResults
                 * Usage    : match results from database based on user selections and search term
                 * Input    : user selections and search term
```

```
	 * Return   : array of matching records
	 */
	function db_matchResults($userSelections, $searchVal_RefID="")
	{		global $db_connect_var;
			global $queryMatch;
			global $dbName_searchTitle;
			global $dbName_searchRefID;
			$table_resultDisplay = array();

			// connect to database
			$DB_connect = @new mysqli ($db_connect_var['host'], $db_connect_var['username'],
$db_connect_var['password'], $db_connect_var['database']);
			if ($DB_connect->connect_errno > 0){ die(); }

			// Query : Get Records based on user selections
			$resultQuery = "SELECT * FROM movie WHERE ";
			if (!empty($userSelections)){
				foreach($userSelections as $section => $intervalGroup){
					$resultQuery .= empty($prefixAND)? "":$prefixAND;
					$resultQuery .= "(";

					// get each interval
					foreach ($intervalGroup as $interval)
					{		// check if column is comma delimited
						if (!(isDBColumn_commaDelimited($section)))
						{
							$resultQuery .= empty($prefixOR)? "":$prefixOR;
							$resultQuery .= $section ." = '". $interval ."'";

							// OR
							$prefixOR = ' OR ';
						} else if (isDBColumn_commaDelimited($section))
						{
							$resultQuery .= empty($prefixOR)? "":$prefixOR;
							$resultQuery .= $section ." LIKE '%". $interval ."%'";

							// OR (changed to AND for dbColumns with multiple values
(comma-delimited)
							$prefixOR = ' AND ';
						}
					}
					$resultQuery .= ")";

					// AND
					$prefixAND = " AND ";
					$prefixOR = "";
				}
				if (!empty($searchVal_RefID))
				{ $resultQuery .= " AND (".$dbName_searchTitle." LIKE '%".$searchVal_RefID."%')"; }
			}
			else if (empty($userSelections)){
				if (!empty($searchVal_RefID)){
					$resultQuery .= "(".$dbName_searchTitle." LIKE '%".$searchVal_RefID."%')";
				}
			}

			if (!empty($userSelections) || !empty($searchVal_RefID)){
				// Execute Query
				if ($pResultQuery = $DB_connect->prepare($resultQuery)){
					$pResultQuery->execute();

					// my_sqli result object
					$result_pResultQuery = $pResultQuery->get_result();
					// array of records
					$table_resultQuery = $result_pResultQuery->fetch_all();

					foreach ($table_resultQuery as $record)
```

```
                                        {
                                                $resultQueryRecord = array();
                                                $resultQueryRecord['movieIndex'] = $record[0];
                                                $resultQueryRecord['movieTitle'] = $record[1];
                                                $resultQueryRecord['imdbRefID'] = $record[2];
                                                $resultQueryRecord['releasedDate'] = $record[3];
                                                $resultQueryRecord['movieType'] = $record[5];
                                                $resultQueryRecord['contentRating'] = $record[6];
                                                $resultQueryRecord['movieGenre'] = array_map('trim',explode(',',
$record[7]));
                                                $resultQueryRecord['totalSeason'] = $record[8];
                                                $resultQueryRecord['movieDuration'] = $record[10];
                                                $resultQueryRecord['movieLanguage'] = array_map('trim',explode(',',
$record[12]));
                                                $resultQueryRecord['movieCountry'] = array_map('trim',explode(',',
$record[13]));
                                                $resultQueryRecord['metaScore'] = $record[14];
                                                $resultQueryRecord['imdbRating'] = $record[16];
                                                $resultQueryRecord['imdbVotes'] = $record[18];

                                                $table_resultDisplay[] = $resultQueryRecord;
                                        }
                                        $queryMatch = $resultQuery;

                                        // Free Results
                                        $pResultQuery->free_result();
                                        $pResultQuery->close();
                                } else {    echo $DB_connect->error;    }
                        }

                        // Close Connection
                        $DB_connect->close();

                        return $table_resultDisplay;
                }

                /*
                 * Function : db_matchResults_unique
                 * Usage    : aggregate a facet based on a filtered collection (query string) to display
                 * Input    : facet info for the facet, an sql query string, sort order
                 * Return   : aggregated and sorted array of unique values of the facet
                 */
                function db_matchResults_unique($facetInfo, $collectionQuery, $sortOrderDefault = true)
                {       global $db_connect_var;

                        // connect to database
                        $DB_connect = @new mysqli ($db_connect_var['host'], $db_connect_var['username'],
$db_connect_var['password'], $db_connect_var['database']);
                        if ($DB_connect->connect_errno > 0){ die(); }
                        // ...database connection success

                        /* Query : Get Unique Facets from collectionQuery */
                        // if not comma delimited
                        if (!$facetInfo[3]){
                                $query = "SELECT ".$facetInfo[1].", COUNT(".$facetInfo[1].") total".
                                                " FROM (".$collectionQuery.") AS fc".
                                                " GROUP BY ".$facetInfo[1];
                                if ($sortOrderDefault){ $query .= " ORDER BY ".$facetInfo[1]. " ASC"; }
                                else if (!$sortOrderDefault){ $query .= " ORDER BY total DESC"; }
                        }
                        // if it is comma delimited
                        else if ($facetInfo[3]){
                                $query = "SELECT cd, COUNT(cd) total".
                                                " FROM (".
                                                " SELECT
SUBSTRING_INDEX(SUBSTRING_INDEX(".$facetInfo[1].", ',', movies.n), ',', -1) cd".
```

```
                                                " FROM (SELECT 1 n union all SELECT 2 union all SELECT 3 union
all SELECT 4 union all SELECT 5) movies".
                                                " INNER JOIN (".$collectionQuery.") AS fc ON
CHAR_LENGTH(".$facetInfo[1].")-CHAR_LENGTH(REPLACE(".$facetInfo[1].", ',', ''))>=movies.n-1)".
                                                " AS x".
                                                " GROUP BY cd";
                                if ($sortOrderDefault){ $query .= " ORDER BY cd ASC"; }
                                else if (!$sortOrderDefault){ $query .= " ORDER BY total DESC"; }
                        }

                        // Execute Query
                        $uniqueArray = array();
                        if ($pQuery = $DB_connect->prepare($query)){
                                $pQuery->execute();

                                // my_sqli result Object
                                $result = $pQuery->get_result();
                                // array of records
                                $table = $result->fetch_all();

                                // no categories
                                if (!$facetInfo[2]){
                                        foreach($table as $record)
                                        {       $uniqueArray[$record[0]] = $record[1];          }
                                }
                                // has categories
                                else if ($facetInfo[2]){
                                        foreach($table as $record)
                                        {       $uniqueArray[ filter_intervalToClassification($facetInfo[0], $record[0])
][$record[0]] = $record[1];     }

                                        // sort categories
                                        if ($sortOrderDefault){ ksort($uniqueArray); }
                                        else if (!$sortOrderDefault){

                                                $array_of_categoryCount = array();
                                                foreach ($uniqueArray as $category => $intervalGroup)
                                                {       // get total count for each category
                                                        $totalCategoryCount = 0;
                                                        foreach($intervalGroup as $interval => $count)
                                                        {       $totalCategoryCount += $count;          }
                                                        $array_of_categoryCount[$category] = $totalCategoryCount;
                                                }
                                                // sort category-totalcount array based on total count
                                                arsort($array_of_categoryCount);

                                                // pass the sorted intervalGroup array to array_of_categoryCount of the
same key
                                                foreach($array_of_categoryCount as $category => $totalCategoryCount)
                                                { $array_of_categoryCount[$category] = $uniqueArray[$category]; }
                                                // overwrite uniqueArray with sorted array of category-intervals
                                                $uniqueArray = $array_of_categoryCount;
                                        }
                                }

                                // Free Results
                                $pQuery->free_result();
                                $pQuery->close();
                        } else {    echo $DB_connect->error;    }

                        // Close Connection
                        $DB_connect->close();

                        return $uniqueArray;
                }
?>
```

Collection.php

```
<?php
	/* Connect to database */
	$DB_connect = @new mysqli ($db_connect_var['host'], $db_connect_var['username'],
$db_connect_var['password'], $db_connect_var['database']);
	/* Database Connection Errors */
	if ($DB_connect->connect_errno > 0)
	{ die(); }

	/* Query : Get All Records */
	$query = "SELECT * FROM movie";

	if ($pQuery = $DB_connect->prepare($query)){
		$pQuery->execute();

		// my_sqli result Object
		$result = $pQuery->get_result();
		// array of records
		$table = $result->fetch_all();

		// convert array of records into an associative array
		$collection = array();
		foreach ($table as $record)
		{
			$collectionRecord = array();
			$collectionRecord['movieIndex'] = $record[0];
			$collectionRecord['movieTitle'] = $record[1];
			$collectionRecord['imdbRefID'] = $record[2];
			$collectionRecord['releasedDate'] = $record[3];
			$collectionRecord['movieType'] = $record[5];
			$collectionRecord['contentRating'] = $record[6];
			$collectionRecord['movieGenre'] = array_map('trim',explode(',', $record[7]));
			$collectionRecord['totalSeason'] = $record[8];
			$collectionRecord['movieDuration'] = $record[10];
			$collectionRecord['movieLanguage'] = array_map('trim',explode(',', $record[12]));
			$collectionRecord['movieCountry'] = array_map('trim',explode(',', $record[13]));
			$collectionRecord['metaScore'] = $record[14];
			$collectionRecord['imdbRating'] = $record[16];
			$collectionRecord['imdbVotes'] = $record[18];

			$collection[] = $collectionRecord;
		}

		// Close Connection
		$pQuery->free_result();
		$pQuery->close();
	}

	// Close database connection
	$DB_connect->close();
?>
```

Detailed CSS Code

cssCommons.css

```
/* ******* *
 * All CSS *
 * ******* */

* {
	-webkit-box-sizing: border-box;
	  -moz-box-sizing: border-box;
	       box-sizing: border-box;

	margin: 0px;
}
body {
	font-family: "Arial", sans-serif;
}
.page_container {
}

/* ********** *
 * Header CSS *
 * ********** */

/* Header Container */
.header {
	background-color: grey;
	margin: 0px;
}
.header > div {
	padding: 0px;
}
/* Header Content */
.header_content {
	text-align: center;
	width: 65%;
	margin: 0 auto;
	padding-top: 25px;
	padding-bottom: 20px;
}
/* Header Title Div */
.header_title {
	margin-bottom: 10px;
}
/* Header Title Text */
.header_title a{
	color: white;
	font-weight: bold;
	font-size: 40px;

	vertical-align: middle;
	text-decoration: none;
}
/* Header Shortcuts Div */
.header_shortcuts {
}
.header_shortcuts p {
	color: white;
}
/* Header Shortcuts Text */
.header_shortcuts a,
.header_shortcuts a:hover,
.header_shortcuts a:focus {
	color: white;
	font-size: 12px;
	font-weight: bold;
```

```
            text-decoration: none;
            vertical-align: middle;

            margin-left: 10px;
            margin-right: 10px;
}

/* ********** *
 * Footer CSS *
 * ********** */

 /* Footer Container */
.footer {
            background-color : #595959;
            clear: both;
}
.footer > div {
            padding: 0px;
}
/* Footer Content */
.footer_content {
            text-align: center;
            width: 65%;

            margin: 0 auto;
            padding-top: 15px;
            padding-bottom: 25px;
}
/* Footer Links Div */
.footer_links {
            font-size: 32px;
            padding: 0px;
}
.footer_links a{
            color: white;
            text-decoration: none;
}
/* Footer Description Div */
.footer_description {
}
.footer_description p{
            color: white;
            font-size: 15px;
}
/* Footer Copyright Div */
.footer_copyright {
}
.footer_copyright p{
            color: white;
            font-size: 12px;
}

/* ********** *
 * Utility CSS *
 * ********** */

.vertical_center {
   display: inline-block;
   vertical-align: middle;
   float: none;
}
```

cssUserPage.css

```
/* ************ *
 * Content CSS *
 * ************ */

.content {
	width: 100%;
	margin: 0px;
	padding: 0px;
	display: flex;
}
/* Left Content */
.left_content {
	width: 400px;
	background-color: #f2f2f2;
	box-shadow: 1px 0px 1px lightgrey;

	float: left;
	overflow: hidden;
}
/* Right Content */
.right_content {
	width: 100%;
	overflow: hidden;
}
/* Right Content Title */
.right_content_title {
	background-color: #f2f2f2;
	border-left: 1px solid lightgrey;
	border-bottom: 1px solid lightgrey;
}
.right_content_title h1{
	width: 90%;
	text-align: left;
	font-size: 40px;
	font-weight: bold;
	color: #669999;
	margin: 0 auto;
}

 /* *************** *
 * Top Content Div *
 * *************** */

.top_content {
	width: 100%;
	background-color: #41d0f4;
}
#lockedSequenceDisplay {
	padding: 3px;
	display: block;
}
#lockedSequenceDisplay:empty {
	display: none;
}
/* Locked Sequence Sections/Elements */
#lockedSequenceDisplay > div{
	font-size: 14px;
	color: white;

	border: 1px solid grey;
	border-radius: 5px;
	margin: 2px 5px;
	padding: 3px 8px;
	display: inline-block;
}
```

```
#lockedSequenceDisplay > div:hover{
	color: #1e0a82;
	border: 3px solid #1e0a82;
}
#lockedSequenceDisplay > div:hover .hoverX {
	font-weight: bold;
	color: #1e0a82;
}
.lockedSequence_notLastSection {
	background-color: #3393d2;
}
.lockedSequence_lastSection {
	background-color: #d3345f;
}

/* ****************** *
 * Left Content Divs *
 * ****************** */

/* Left Content Divs */
.left_content > div{
	overflow: hidden;
	margin: 5px auto;
	padding: 2px;
}
/* Search Container */
.left_content_facetedSearch {
}
/* Sort Container */
.left_content_facetedSort {
}
/* Filter Container */
.left_content_facetedFilter {
}

/* ****************** *
 * Right Content Divs *
 * ****************** */

/* Right Content Divs */
.right_content > div{
	width: 100%;
}
/* Result Container */
.right_content_resultDisplay {
}

/* ************************** *
 * Left Content CSS (Search) *
 * ************************** */

.left_content_facetedSearch {
	width: 100%;

	white-space: nowrap;

	margin-top: 12px;
	margin-bottom: 16px;
}
.left_content_facetedSearch > div {
	width: 90%;
	margin: 0px auto;
	position: relative;
}
.left_content_facetedSearch input{
	width: 100%;
	font-size: 12px;
```

```
		font-weight: bold;
		text-align: center;
		color: #669999;

		display: inline-block;
		padding: 5px 0px;
}

/* ********************** *
 * Left Content CSS (Sort) *
 * ********************** */

.left_content_facetedSort {
		width: 100%;
		margin: 0px auto;
}
.left_content_facetedSort > div {
		width: 90%;
		text-align: center;
		font-size: 12px;

		background-color: lightgrey;
		border: 1px solid lightgrey;
		border-radius: 5px;

		overflow: hidden;
		white-space: nowrap;
		clear: both;
		margin: 0px auto;
		padding: 5px 0px;
}
.left_content_facetedSort label {
		margin: 3px 0px;
		display: inline-block;
}

/* ************************* *
 * Left Content CSS (Filter) *
 * ************************* */

/* Faceted Filter Form and Form Div */
#facetedFilterForm {
}
#facetedFilterForm_Content {
		clear: both;
		white-space: nowrap;
}
/* Form Sections */
.facetedFilterFormSection {
		max-height: 45px;

		border-bottom: 1px solid grey;
		margin: 0px 3px;
		overflow: hidden;
}
/* Section Title Div */
.facetedFilterFormSection_Label {
		width: 100%;
		margin: 0px auto;
		padding-top: 20px;
}
.facetedFilterFormSection_Label > div:first-child {
		width: 65%;
		text-align: center;

		padding: 0px 3px;
		float: left;
```

```
}
.facetedFilterFormSection_Label > div:nth-child(2) {
	width: 35%;
	padding: 0px 3px;
	float: right;
}
/* Section Title Div (Label) */
.facetedFilterFormSection_Label label{
	width: 100%;
	font-size: 16px;
	font-weight: bold;
	color: #4d4d4d;
}
/* Section Title Div (Span) - Lock Text Span for Progressive Filter */
.lockText {
	width: 100%;
	text-align: center;
	font-size: 12px;
	background-color: lightgrey;

	border: 1px solid grey;
	border-radius: 5px;
	display: inline-block;

	padding: 2px;
}
.lockText:hover {
	color: lightgrey;
	background-color: grey;
}

/* Section Input Div */
.facetedFilterFormSection_Input {
	width: 80%;
	text-align: left;

	margin: 0px auto;
	padding-top: 5px;
	padding-bottom: 10px;
	clear: both;
}
/* Section Input */
.facetedFilterFormSection_Input label{
	font-size: 14px;
	color: grey;
}
/* categories (super & sub) input */
.facetedFilterFormSection_inputSuper {
	margin-left: 5px;
	margin-right: 1px;
	margin-bottom: 8px;
}
.facetedFilterFormSection_inputSub {
	margin-left: 20px;
	margin-bottom: 8px;
}

/* ************************** *
 * Left Content CSS (Buttons) *
 * ************************** */

/* Form Content Submit Button & Reset Link */
.btns_container {
	width: 100%;
	margin: 0px auto;
	padding: 0px;
	padding-bottom: 40px;
```

```
}
.btns_container > div:first-child {
	width: 50%;
	padding: 0px 3px;
	float: left;
}
.btns_container > div:nth-child(2) {
	width: 50%;
	padding: 0px 3px;
	float: right;
}
.btns {
	width: 100%;

	text-align: center;
	font-size: 20px;
	font-weight: bold;
	color: #669999;

	background-color: lightgrey;
	border: 1px solid lightgrey;
	border-radius: 5px;
	display: inline-block;
	margin: 0px;
	padding: 5px;
	overflow: hidden;
}
.submitBtn {
}
.resetLink {
}
.resetLink,
.resetLink:link,
.resetLink:visited,
.resetLink:hover,
.resetLink:active {
	text-decoration: none;
}

/* ************************** *
 * Right Content CSS (Display) *
 * ************************** */

/* Result Content Div */
.right_content_resultDisplay_content {
	width: 90%;
	margin: 15px auto;
}
/* Result Content - Result Div */
.right_content_resultDisplay_content > div{
	overflow: hidden;
	word-wrap: break-word;

	border: 1px solid lightgrey;
	margin: 15px auto;
	position: relative;
}

.right_content_resultDisplay_content > div > div{
}
.right_content_resultDisplay_content > div > div div{
	padding: 5px;
}
/* First Row */
.right_content_resultDisplay_content > div > div:first-child {
	text-align: center;
	background-color: #669999;
```

```
        display: flex;
        overflow: hidden;
}
.right_content_resultDisplay_content > div > div:first-child > div:first-child{
        min-width: 15%;
        font-size: 18px;
        font-style: italic;
        line-height: 25px;
        vertical-align: bottom;

        background-color: lightgreen;
}
.right_content_resultDisplay_content > div > div:first-child > div:nth-child(2){
        width: 55%;
        font-size: 24px;
        font-weight: bold;
        line-height: 50px;
        vertical-align: bottom;
        text-align: center;
}
.right_content_resultDisplay_content > div > div:first-child > div:nth-child(3){
        min-width: 10%;
        background-color: red;
}
.right_content_resultDisplay_content > div > div:first-child > div:nth-child(4){
        min-width: 10%;
        background-color: lightgreen;
}
.right_content_resultDisplay_content > div > div:first-child > div:nth-child(5){
        min-width: 10%;
        background-color: cyan;
}
/* Second Row */
.right_content_resultDisplay_content > div > div:nth-child(2) {
        overflow: hidden;
        clear: both;
        position: relative;
}
.right_content_resultDisplay_content > div > div:nth-child(2) span {
        width: 15%;
        font-weight: bold;
        text-align: center;
        display: inline-block;
}
.right_content_resultDisplay_content > div > div:nth-child(2) a:link,
.right_content_resultDisplay_content > div > div:nth-child(2) a:visited,
.right_content_resultDisplay_content > div > div:nth-child(2) a:hover,
.right_content_resultDisplay_content > div > div:nth-child(2) a:active {
        text-decoration: none;
}
.ext_link_imdb {
        color: white;
        background-color: lightgrey;

        position: absolute;
        bottom: 0px;
        right: 0px;
}
/* Third Row */
.right_content_resultDisplay_content > div > div:nth-child(3) {
        overflow: hidden;
        display: flex;
        clear: both;
}
.right_content_resultDisplay_content > div > div:nth-child(3) > div{
        width: 33.3334%;
        text-align: center;
```

```
        border: 1px solid lightgrey;
}

/* ************ *
 * Utility CSS *
 * ************ */

.pushDiv {
        max-width: 3px;
        -webkit-transition: max-width 0.5s ease-out;
        transition: max-width 0.5s ease-out;
}
.pushDiv:hover {
        max-width: 1000px;
   -webkit-transition: max-width 0.5s ease-out;
        transition: max-width 0.5s ease-in;
}
.pullDiv{
}
.pullDiv:hover {
}

.shortenDiv {
        -webkit-transition: max-height 0.8s ease-out;
        transition: max-height 0.8s ease-out;
}
.shortenDiv:hover {
}
.extendDiv{
        max-height: 1000px;
   -webkit-transition: max-height 0.5s ease-out;
        transition: max-height 0.5s ease-in;
}
.extendDiv:hover {
}

.autocomplete_popup {
        width: 100%;
        font-size: 12px;
        text-align: center;

        background-color: #e6eaf2;
        display: block;
        position: absolute;
        top: 35px;
        left: 0px;
        z-index: 1;
}
.autocomplete_popup ul{
}
.autocomplete_popup li{
        list-style-type: none;
        padding: 8px 5px;
        overflow: hidden;
}
.autocomplete_popup li:hover{
        border: 1px solid grey;
        border-radius: 5px;
}
#autocomplete_s {
}
```

cssAdminPage.css

```
/* ************ *
 * Content CSS *
 * ************ */

.content {
	width: 100%;
	margin: 0px;
	padding: 0px;
	display: flex;
}
/* Left Content */
.left_content {
	max-width: 2px;

	background-color: #f2f2f2;
	box-shadow: 1px 0px 1px lightgrey;

	float: left;
	overflow: hidden;

	-webkit-transition: max-width 1s;
	transition: max-width 1s;
}
.left_content:hover {
	max-width: 325px;
	-webkit-transition: max-width 1s;
	transition: max-width 1s;
}
/* Left Content Div */
.left_content > div{
	margin: 0 auto;
}
/* Right Content */
.right_content {
	width: 100%;
	min-height: 500px;
	overflow: hidden;
}
/* Right Content Title */
.right_content_title {
	background-color: #f2f2f2;
	border-left: 1px solid lightgrey;
	border-bottom: 1px solid lightgrey;
}
.right_content_title h1{
	width: 90%;
	text-align: left;
	font-size: 40px;
	font-weight: bold;
	color: #669999;
	margin: 0 auto;
}

 /* *************** *
 * Top Content Div *
 * *************** */

.top_content {
	width: 100%;
	background-color: #41d0f4;
}
#top_content_msg {
	text-align: center;
	font-weight: bold;
	font-size: 24px;
```

```
		padding: 5px;
		display: block;
}
#top_content_msg:empty {
		display: none;
}
#top_content_msg > .success{
		color: green;
}
#top_content_msg > .fail{
		color: #ed2a34;
}

/* ********************** *
 * Left Content CSS (Tab) *
 * ********************** */

 /* Left Content Tab Nav */
.nav_devTabs {
		text-align: center;
}

/* Left Content Tabs */
.nav_devTabs button{
		width: 100%;
		font-size: 20px;
		background-color: #f2f2f2;

		border: none;
		padding: 15px 0px;
		white-space: nowrap;
}
.nav_devTabs button:hover{
		color: white;
		background-color: grey;
}
.nav_devTabs button:focus{
		outline: 0;
}

/* ******************* *
 * Right Content Divs *
 * ******************* */

/* Right Content Divs */
.right_content > div{
		width: 100%;
}
/* Summary Div */
#right_content_summary{
}
 /* Log Div */
#right_content_log{
		position: relative;
		display: none;
}
#right_content_form{
		display: none;
}
#right_content_api{
		display: none;
}

/* ************************** *
 * Right Content CSS (Summary) *
 * ************************** */
```

```
/* Summary Content */
#right_content_summary_content {
}
/* Summary Content Tables, th, tr, td */
#right_content_summary_content table{
		width: 90%;
		text-align: center;

		border-collapse: collapse;
		margin: 15px auto;
		padding: 20px;
}
#right_content_summary_content tr,
#right_content_summary_content th,
#right_content_summary_content td{
		background-color: #f0f5f5;
		border: 1px solid lightgrey;
		padding: 5px;
}
#right_content_summary_content th{
		font-size: 20px;
		color: white;
		background-color: #669999;
}
#right_content_summary_content td:first-child{
		width: 25%;
		font-weight: bold;
}
#right_content_summary_content td:nth-child(2){
		width: 50%;
}

/* ********************** *
 * Right Content CSS (Log) *
 * ********************** */

/* Log Download Form Div */
#right_content_download_log{
		text-align: center;

		position: absolute;
		top: 7px;
		right: 50px;
}
#right_content_download_log select{
		width: 150px;
		height: 35px;
		font-size: 18px;

		color: white;
   background-color: #24b2c2;
		border-radius: 5px;
		padding: 5px;
}
#right_content_download_log input{
		width: 100px;
		height: 35px;
		font-size: 18px;
		font-weight: bold;

		color: white;
   background-color: #209dac;
		border: none;
		padding: 5px;
}
/* Log Details Div */
```

```
#right_content_log_FileDetails {
	width: 97%;
	overflow: hidden;
	margin: 15px auto;
}
/* Log Details Tables, th, tr, td */
#right_content_log_FileDetails table{
	width: 100%;
	text-align: center;

	border-collapse: collapse;
	padding: 20px;
}
#right_content_log_FileDetails tr,
#right_content_log_FileDetails th,
#right_content_log_FileDetails td{
	background-color: #f0f5f5;
	border: 1px solid lightgrey;
	padding: 5px;
}
#right_content_log_FileDetails th{
	font-size: 18px;
	color: white;
	background-color: #669999;
}
#right_content_log_FileDetails td{
	font-size: 14px;
}

/* ************************* *
 * Right Content CSS (Form) *
 * ************************* */

/* Form Content Navigation Tabs */
.nav_formTabs {
	width: 90%;
	text-align: center;
	margin: 15px auto;
}
.nav_formTabs button{
	width: 31.5%;
	font-size: 20px;

	background-color: white;
	border: none;
	word-wrap: break-word;
	padding: 15px;
}
.nav_formTabs button:hover{
	background-color: #e6e6e6;
}
.nav_formTabs button:focus{
	outline: 0;
}
.nav_formTabs hr{
	width: 100%;
	height: 1px;
	border: 3px solid #f2f2f2;
	margin: 5px auto;
}
.formTabs_btn_active hr {
	border: 3px solid red;
}
/* Form Content Div (Forms) */
#right_content_forms {
	width: 90%;
	margin: 15px auto;
```

```
}
#right_content_forms > div{
	text-align: center;
}
#right_content_update_form,
#right_content_delete_form {
	display: none;
}
/* Form Content (Creation) */
#creation_form_content {
}
#creation_form_legend{
	font-size: 22px;
	font-style: italic;
	color: lightgrey;
}
/* Form Content Container & Section (Creation) */
#creation_form_inputs_container{
}
.creation_form_inputs_section {
	width: 65%;
	text-align: center;

	border: 1px solid lightgrey;
	padding: 15px;
	margin: 5px auto;
	word-wrap: break-word;
}
/* Form Content Section's Label, Input, and Error Span (Creation) */
.creation_form_inputs_section > label{
	width: 70%;
	font-size: 22px;
	font-weight: bold;
	color: grey;
	display: inline-block;
}
.creation_form_inputs_section > input{
	width: 70%;
	height: 35px;
	font-size: 20px;
	text-align: center;
	color: #669999;

	border: 1px solid lightgrey;
	border-radius: 5px;
	margin-top: 5px;
	padding: 5px;
}
.creation_form_inputs_section > input:focus{
	outline: #3dccdb solid 1px;
	border-radius: 5px;
}
.creation_form_inputs_section > span{
	color: red;
	font-weight: bold;
	display: block;
}
/* Form Content (Update) */
#update_form_content {
}
#update_form_legend{
	font-size: 22px;
	font-style: italic;
	color: lightgrey;
}
/* Form Content Search (Update) */
#update_form_search{
```

```
		width: 55%;
		margin: 0px auto;
		margin-top: 10px;
		margin-bottom: 30px;
		position: relative;
}
#update_form_search label{
		font-size: 32px;
		font-weight: bold;

		color: #669999;
		margin: 10px 0px;
		display: inline-block;
}
#update_form_search input{
		width: 100%;
		height: 36px;
		font-size: 22px;
		text-align: center;
		color: #669999;

		border: 1px solid lightgrey;
		border-radius: 5px;
		padding: 25px 0px;
}
#update_form_search button{
		width: 100%;
		height: 32px;
		font-size: 20px;
		font-weight: bold;
		color: #669999;
}
/* Form Content Container & Section (Update) */
#update_form_inputs_container{
		/*display: none;*/
}
.update_form_inputs_section {
		width: 65%;
		text-align: center;

		border: 1px solid lightgrey;
		padding: 15px;
		margin: 5px auto;
		word-wrap: break-word;
}
/* Form Content Section's Label, Input, and Error Span (Update) */
.update_form_inputs_section > label{
		width: 70%;
		font-size: 22px;
		font-weight: bold;
		color: grey;
		display: inline-block;
}
.update_form_inputs_section > input{
		width: 70%;
		height: 35px;
		font-size: 20px;
		text-align: center;
		color: #669999;

		border: 1px solid lightgrey;
		border-radius: 5px;
		margin-top: 5px;
		padding: 5px;
}
.update_form_inputs_section > input:focus{
		outline: #3dccdb solid 1px;
```

```
		border-radius: 5px;
}
.update_form_inputs_section > span{
		color: red;
		font-weight: bold;
		display: block;
}
/* Form Content (Delete) */
#delete_form_content {
}
#delete_form_legend {
		font-size: 22px;
		font-style: italic;
		color: lightgrey;
}
/* Form Content Search (Update) */
#delete_form_search{
		width: 55%;
		margin: 0px auto;
		margin-top: 10px;
		margin-bottom: 30px;
		position: relative;
}
#delete_form_search label{
		font-size: 32px;
		font-weight: bold;

		color: #669999;
		margin: 10px 0px;
		display: inline-block;
}
#delete_form_search input{
		width: 100%;
		height: 36px;
		font-size: 22px;
		text-align: center;
		color: #669999;

		border: 1px solid lightgrey;
		border-radius: 5px;
		padding: 25px 0px;
}
#delete_form_search button{
		width: 100%;
		height: 32px;
		font-size: 20px;
		font-weight: bold;
		color: #669999;
}
/* Form Content Container & Section (Delete) */
#delete_form_inputs_container {
}
.delete_form_inputs_section {
}
/* Form Content Section's Info Display Div (Delete) */
.delete_form_inputs_section > div{
}
/* Form Content Section's Info (Delete) */
.delete_form_inputs_section > div > div{
		width: 45%;

		font-size: 20px;
		word-wrap: break-word;

		color: #669999;
		border: 1px solid lightgrey;
		border-radius: 5px;
```

```
		padding: 0px;
		margin: 5px auto;
}
.delete_form_inputs_section > div > div > div:first-child{
		width: 50%;
		text-align: right;
		display: inline-block;
}
.delete_form_inputs_section > div > div > div:nth-child(2){
		width: 50%;
		text-align: left;
		display: inline-block;
		padding-left: 5px;
}
/* Form Content Submit Button & Reset Link */
.forms_btns_container {
		width: 65%;
		margin: 0px auto;
		margin-top: 30px;
		padding: 0px;
}
.forms_btns_container > div:first-child {
		width: 50%;
		padding: 0px 15px;
		float: left;
}
.forms_btns_container > div:nth-child(2) {
		width: 50%;
		padding: 0px 15px;
		float: right;
}
.forms_btns {
		width: 100%;

		text-align: center;
		font-size: 20px;
		font-weight: bold;
		color: #669999;

		background-color: lightgrey;
		border: 1px solid lightgrey;
		border-radius: 5px;
		display: inline-block;
		margin: 0px;
		padding: 5px;
		overflow: hidden;
}
.formSubmitBtn {
}
.resetLink {
}
.resetLink,
.resetLink:link,
.resetLink:visited,
.resetLink:hover,
.resetLink:active {
		text-decoration: none;
}

/* ************************ *
 * Right Content CSS (API) *
 * ************************ */

#right_content_api_form {
		width: 90%;
		margin: 15px auto;
}
```

```
#api_form_legend {
}
#api_form_inputs_container {
	padding: 5px;
}

/* ********** *
 * Utility CSS *
 * ********** */

.shortenDiv {
	max-height: 60px;
	-webkit-transition: max-height 0.8s ease-out;
	transition: max-height 0.8s ease-out;
}
.shortenDiv:hover {
	border: 3px solid black;
}
.extendDiv{
	max-height: 10000px;
   -webkit-transition: max-height 0.5s ease-out;
	transition: max-height 0.5s ease-in;
}
.extendDiv:hover {
	border: 3px solid black;
}

.autocomplete_popup {
	width: 100%;

	background-color: #e6eaf2;
	display: none;
	position: absolute;
	top: 107px;
	left: 0px;
	z-index: 1;
}
.autocomplete_popup li{
	list-style-type: none;
	padding: 8px 0px;
}
.autocomplete_popup li:hover{
	border: 1px solid grey;
	border-radius: 5px;
}
#autocomplete_u {
}
#autocomplete_d {
}
```

Detailed JavaScript Code

scriptCommons.js

```
// JavaScript : Function to disallow checkbox checking both super categories and sub categories
function toggleFacetCheck(facet, super_or_sub, classification)
{	// class names of super and sub checkboxes
	var toggleTrigger = facet + "_" + super_or_sub + "_" + classification;
	var toggleTarget;

	// if checkbox is super AND is checked
	if (super_or_sub == "super"){
		if (document.getElementsByClassName(toggleTrigger)[0].checked == true)
		{	// then uncheck the sub checkbox
			toggleTarget = facet + "_sub_" + classification;
			var toggleTargetElement = document.getElementsByClassName(toggleTarget)

			for (var i = 0; i < toggleTargetElement.length; i++)
			{ toggleTargetElement[i].checked = false; }
		}
	} // if checkbox is sub AND is checked
	else if (super_or_sub == "sub"){
		// as long as there's one sub checkbox that is checked, uncheck the super checkbox
		var toggleTriggerElement = document.getElementsByClassName(toggleTrigger);
		for (var i = 0; i < toggleTriggerElement.length; i++)
		{	if (toggleTriggerElement[i].checked == true)
			{	toggleTarget = facet + "_super_" + classification;
				var toggleTargetElement = document.getElementsByClassName(toggleTarget)
				toggleTargetElement[0].checked = false
				break;
			}
		}
	}
}

// JavaScript (UserPage) : Function to push/pull Div Sections for display
function setPushPull(elementID)
{	// Get Element
	var targetElement = document.getElementsByClassName(elementID)[0];
	// Push Div
	if(targetElement.classList.contains('pullDiv')){
		targetElement.classList.remove('pullDiv');
		targetElement.classList.add('pushDiv');
	}
}

// JavaScript (UserPage) : Function to extend/shorten Div Sections for display
function clickDisplayID(elementID)
{	// Get Element
	var targetElement = document.getElementById(elementID);

	// Extend Div
	if(targetElement.classList.contains('shortenDiv')){
		targetElement.classList.remove('shortenDiv');
		targetElement.classList.add('extendDiv');
	} // Shorten Div
	else if(targetElement.classList.contains('extendDiv')){
		targetElement.classList.remove('extendDiv');
		targetElement.classList.add('shortenDiv');
	}
}

// Javascript (DevPage) : Function to select tabs by displaying/hiding tab sections
function tabSelect(tabID, tabClass,
					tabButton = '', tabButtonClass = '',
					hasTabLine = false, tabLineClass = ''){
	var tabs = document.getElementsByClassName(tabClass);
```

```
	for (var i = 0; i < tabs.length; i++)
	{ tabs[i].style.display = "none"; }

	// if <hr/> line underneath is active
	if (hasTabLine) {
		var tab_btns = document.getElementsByClassName(tabButtonClass);
		for (var j = 0; j < tab_btns.length; j++)
		{ tab_btns[j].classList.remove(tabLineClass); }
	}

	document.getElementById(tabID).style.display = "block";
	if (hasTabLine) tabButton.classList.add(tabLineClass);
}

// JavaScript (DevPage) : Function to extend/shorten Div Sections for display
function toggleDisplayHorizontal(elementID)
{	// Get Element
	var targetElement = document.getElementById(elementID);

	// Extend Div
	if(targetElement.classList.contains('shortenDiv')){
		targetElement.classList.remove('shortenDiv');
		targetElement.classList.add('extendDiv');
	} // Shorten Div
	else if(targetElement.classList.contains('extendDiv')){
		targetElement.classList.remove('extendDiv');
		targetElement.classList.add('shortenDiv');
	}
}

// Javascript : Function to send AJAX call to OMDB Api to retrieve record information
function getData_OMDbAPI(){
	var ID = document.getElementById('i').value;

	var xhttpAPI = new XMLHttpRequest();
	xhttpAPI.onreadystatechange = function() {
		if (this.readyState == 4 && this.status == 200) {
			var api_response_decoded = JSON.parse(this.responseText);

			// if response is valid (valid record)
			if (api_response_decoded['Response'] == "True"){
				//console.log(api_response_decoded);
				updateDatabase_OMDbAPI(api_response_decoded);

			} // Debug
			else {
				console.log('invalid response from api');
				console.log(api_response_decoded);
			}
		}
	};
	xhttpAPI.open("GET", "http://www.omdbapi.com/?i="+ID, true);
	xhttpAPI.send();
}

// Javascript : Function to update database with record information obtained from OMDB Api
function updateDatabase_OMDbAPI(api_response_decoded){
	var xhttpUpdate = new XMLHttpRequest();
	xhttpUpdate.onreadystatechange = function() {
		if (this.readyState == 4 && this.status == 200) {
			var result = this.responseText;

			// if record successfully created
			if (result == "true"){
				console.log("record creation success");
			} else { console.log("record creation failed"); }
		}
```

```
		};

		// url to respond to AJAX call
		var url = "dev_recordCreate_OMDbAPI.php";

		// open POST request, and set header for POST request
		xhttpUpdate.open("POST", url, true);
		xhttpUpdate.setRequestHeader("Content-type", "application/x-www-form-urlencoded");

		// convert array to JSON string
		var api_response_decoded_jsonString = JSON.stringify(api_response_decoded);
		var var_send = "api_response="+encodeURIComponent(api_response_decoded_jsonString);
		// send AJAX POST request
		xhttpUpdate.send(var_send);
}

// Javascript : Function to set value to an element
function setToInput(word, targetElement){
		ID = document.getElementById(targetElement);
		ID.value = word.innerHTML;
}

// Javascript : Function to hide Popup
function popup_disappear(targetElement){
		document.getElementById(targetElement).style.display = "none";
}

// Javascript : Function to send AJAX call to get list of matching RefID
function getData_FormAutoComplete(wordsElement, suggestionElement){
		var ID = document.getElementById(wordsElement).value;

		var xhttpUpdate = new XMLHttpRequest();
		xhttpUpdate.onreadystatechange = function() {
			if (this.readyState == 4 && this.status == 200) {
				//console.log(this.responseText);
				if(this.responseText){
					var target = document.getElementById(suggestionElement);
					target.style.display = "block";

					var record_details = JSON.parse(this.responseText);

					target.innerHTML = "<ul>";
					for (var i=0; i<record_details.length; i++)
					{ target.innerHTML += "<li onclick='setToInput(this, \""+wordsElement+"\")'>"+
record_details[i] +"</li>"; }
					target.innerHTML += "</ul>";

					//console.log(record_details);
				} else { document.getElementById(suggestionElement).style.display = "none"; }
			}
		}

		// url to respond to AJAX call
		var url = "dev_recordRefIDList_Form.php";

		// open POST request, and set header for POST request
		xhttpUpdate.open("POST", url, true);
		xhttpUpdate.setRequestHeader("Content-type", "application/x-www-form-urlencoded");

		// convert array to JSON string
		var var_send = "RefID="+encodeURIComponent(ID);
		// send AJAX POST request
		xhttpUpdate.send(var_send);
}

// Javascript : Function to send AJAX call to get list of matching RefID based on user selections
function getData_FormAutoComplete_Filtered(wordsElement, suggestionElement){
```

```
	var ID = document.getElementById(wordsElement).value;
	var userSelectionChecked = {};

	/* Get All checked values */
	var filterForm_sections = document.getElementById('facetedFilterForm_Content').children;
	for (var ff_i = 0; ff_i < filterForm_sections.length; ff_i++)
	{	var filterForm_sections_Element = filterForm_sections[ff_i];
		var filterForm_sections_ElementID = filterForm_sections_Element.id;

		// look into each input element for the each current section
		for (var ffc_i = 0; ffc_i < filterForm_sections_Element.children[1].children.length; ffc_i += 2 )
		{	var current_input = filterForm_sections_Element.children[1].children[ffc_i].children[0];

			// if section is checked, add into user selections array
			if(current_input.checked == true){
				if (userSelectionChecked[filterForm_sections_ElementID] == undefined)
				{ userSelectionChecked[filterForm_sections_ElementID] = {}; }

				// create a category array to store intervals
				if ( userSelectionChecked[filterForm_sections_ElementID][current_input.name] ==
undefined )
				{ userSelectionChecked[filterForm_sections_ElementID][current_input.name] = [];
}

				// store intervals into category array

	userSelectionChecked[filterForm_sections_ElementID][current_input.name].push(current_input.value);
			}
		}
	}

	var xhttpUpdate = new XMLHttpRequest();
	xhttpUpdate.onreadystatechange = function() {
		if (this.readyState == 4 && this.status == 200) {
			if(this.responseText){
				var target = document.getElementById(suggestionElement);
				target.style.display = "block";

				var record_details = JSON.parse(this.responseText);

				target.innerHTML = "<ul>";
				for (var i=0; i<record_details.length; i++)
				{ target.innerHTML += "<li onclick='setToInput(this, \""+wordsElement+"\")'>"+
record_details[i] +"</li>"; }
				target.innerHTML += "</ul>";
			} else { document.getElementById(suggestionElement).style.display = "none"; }
		}
	}

	// url to respond to AJAX call
	var url = "dev_recordTitleList_Form.php";

	// open POST request, and set header for POST request
	xhttpUpdate.open("POST", url, true);
	xhttpUpdate.setRequestHeader("Content-type", "application/x-www-form-urlencoded");

	// convert array to JSON string
	var var_send = "RefID="+encodeURIComponent(ID);
	var userSelectionChecked_jsonString = JSON.stringify(userSelectionChecked);
	var_send += "&userSelections="+encodeURIComponent(userSelectionChecked_jsonString);
	// send AJAX POST request
	xhttpUpdate.send(var_send);
}

// Javascript : Function to send AJAX call to retrieve database record by RefID
function getData_FormData(formTypeChar){
	if (formTypeChar == 'u') var targetID = 'UForm_search';
```

```
        else if (formTypeChar == 'd') var targetID = 'DForm_search';
        var ID = document.getElementById(targetID).value;

        var xhttpUpdate = new XMLHttpRequest();
        xhttpUpdate.onreadystatechange = function() {
                if (this.readyState == 4 && this.status == 200) {
                        if (formTypeChar == 'u'){
                                if(this.responseText){
                                        var record_details = JSON.parse(this.responseText);
                                        document.getElementById('UForm_movieTitle').value                =
record_details['movieTitle'];
                                        document.getElementById('UForm_imdbRefID').value                 =
record_details['imdbRefID'];
                                        document.getElementById('UForm_releasedDate').value     =
record_details['releasedDate'];
                                        document.getElementById('UForm_movieType').value                 =
record_details['movieType'];
                                        document.getElementById('UForm_contentRating').value   =
record_details['contentRating'];
                                        document.getElementById('UForm_movieGenre').value                =
record_details['movieGenre'];
                                        document.getElementById('UForm_totalSeason').value               =
record_details['totalSeason'];
                                        document.getElementById('UForm_movieDuration').value =
record_details['movieDuration'];
                                        document.getElementById('UForm_movieLanguage').value             =
record_details['movieLanguage'];
                                        document.getElementById('UForm_movieCountry').value  =
record_details['movieCountry'];
                                        document.getElementById('UForm_metaScore').value                 =
record_details['metaScore'];
                                        document.getElementById('UForm_imdbRating').value                =
record_details['imdbRating'];
                                        document.getElementById('UForm_imdbVotes').value                 =
record_details['imdbVotes'];
                                } else {
                                        document.getElementById('UForm_movieTitle').value                = "";
                                        document.getElementById('UForm_imdbRefID').value                 = "";
                                        document.getElementById('UForm_releasedDate').value     = "";
                                        document.getElementById('UForm_movieType').value                 = "";
                                        document.getElementById('UForm_contentRating').value   = "";
                                        document.getElementById('UForm_movieGenre').value                = "";
                                        document.getElementById('UForm_totalSeason').value               = "";
                                        document.getElementById('UForm_movieDuration').value = "";
                                        document.getElementById('UForm_movieLanguage').value             = "";
                                        document.getElementById('UForm_movieCountry').value  = "";
                                        document.getElementById('UForm_metaScore').value                 = "";
                                        document.getElementById('UForm_imdbRating').value                = "";
                                        document.getElementById('UForm_imdbVotes').value                 = "";
                                }
                        } else if (formTypeChar == 'd'){
                                if(this.responseText){
                                        var record_details = JSON.parse(this.responseText);
                                        document.getElementById('DForm_movieTitle').innerHTML
        =
                                                "<div><div>movieTitle : </div><div>"
        +record_details['movieTitle']           +"</div></div>";
                                        document.getElementById('DForm_imdbRefID').innerHTML
        =
                                                "<div><div>imdbRefID : </div><div>"
        +record_details['imdbRefID']            +"</div></div>";
                                        document.getElementById('DForm_releasedDate').innerHTML      =
                                                "<div><div>releasedDate : </div><div>"
        +record_details['releasedDate']                 +"</div></div>";
                                        document.getElementById('DForm_movieType').innerHTML
        =
```

```
                                        "<div><div>movieType : </div><div>"
+record_details['movieType']            +"</div></div>";
                                        document.getElementById('DForm_contentRating').innerHTML    =
                                        "<div><div>contentRating : </div><div>"
+record_details['contentRating']        +"</div></div>";
                                        document.getElementById('DForm_movieGenre').innerHTML
=
                                        "<div><div>movieGenre : </div><div>"
+record_details['movieGenre']           +"</div></div>";
                                        document.getElementById('DForm_totalSeason').innerHTML
=
                                        "<div><div>totalSeason : </div><div>"
+record_details['totalSeason']          +"</div></div>";
                                        document.getElementById('DForm_movieDuration').innerHTML    =
                                        "<div><div>movieDuration : </div><div>"
+record_details['movieDuration']        +"</div></div>";
                                        document.getElementById('DForm_movieLanguage').innerHTML    =
                                        "<div><div>movieLanguage : </div><div>"
+record_details['movieLanguage']        +"</div></div>";
                                        document.getElementById('DForm_movieCountry').innerHTML
=
                                        "<div><div>movieCountry : </div><div>"
+record_details['movieCountry']         +"</div></div>";
                                        document.getElementById('DForm_metaScore').innerHTML
=
                                        "<div><div>metaScore : </div><div>"
+record_details['metaScore']            +"</div></div>";
                                        document.getElementById('DForm_imdbRating').innerHTML
=
                                        "<div><div>imdbRating : </div><div>"
+record_details['imdbRating']           +"</div></div>";
                                        document.getElementById('DForm_imdbVotes').innerHTML
=
                                        "<div><div>imdbVotes : </div><div>"
+record_details['imdbVotes']            +"</div></div>";
                                } else {
                                        document.getElementById('DForm_movieTitle').innerHTML
= "";
                                        document.getElementById('DForm_imdbRefID').innerHTML
= "";
                                        document.getElementById('DForm_releasedDate').innerHTML     = "";
                                        document.getElementById('DForm_movieType').innerHTML
= "";
                                        document.getElementById('DForm_contentRating').innerHTML    = "";
                                        document.getElementById('DForm_movieGenre').innerHTML
= "";
                                        document.getElementById('DForm_totalSeason').innerHTML
= "";
                                        document.getElementById('DForm_movieDuration').innerHTML    = "";
                                        document.getElementById('DForm_movieLanguage').innerHTML    = "";
                                        document.getElementById('DForm_movieCountry').innerHTML
= "";
                                        document.getElementById('DForm_metaScore').innerHTML
= "";
                                        document.getElementById('DForm_imdbRating').innerHTML
= "";
                                        document.getElementById('DForm_imdbVotes').innerHTML
= "";
                                }
                        }
                }
};

// url to respond to AJAX call
var url = "dev_recordInfo_Form.php";

// open POST request, and set header for POST request
```

```
        xhttpUpdate.open("POST", url, true);
        xhttpUpdate.setRequestHeader("Content-type", "application/x-www-form-urlencoded");

        // convert array to JSON string
        var var_send = "RefID="+encodeURIComponent(ID);
        // send AJAX POST request
        xhttpUpdate.send(var_send);
}
```

script.js

```
var text_lock = "lock";
var text_toLock = "lock?";
var text_unlock = "unlock";
var text_toUnlock = "unlock?";

var userSelections_locked = {};
var sectionLocked_sequence = [];

// Javascript : Function to check whether section is currently locked or not
function isSectionLocked(facetSectionID)
{
	for (var i = 0; i < sectionLocked_sequence.length; i++)
	{	// section is currently locked
		if (sectionLocked_sequence[i].includes(facetSectionID)){ return true; }
	} return false;
}

// JavaScript : Function to	handle progressive filter from locking a section
function ajaxFacetFilter(facetSectionID)
{
	// facet section, label section, lock element, and the input section
	var facetSection = document.getElementById(facetSectionID);
	var facetSectionLabel = facetSection.children[0];
	var facetSectionLockElement = facetSection.children[0].children[1];
	var facetSectionInput = facetSection.children[1];

	// check if locked yet or not,
	// and find the current to-be-unlocked section's index in sectionLocked_sequence array
	var isLocked = false;
	var lockedIndex;
	for (var i = 0; i < sectionLocked_sequence.length; i++)
	{	// section is currently locked
		if (sectionLocked_sequence[i].includes(facetSectionID))
		{	isLocked = true;
			// locked index for the section recorded
			lockedIndex = i;
			break;
		}
	}

	var canAJAX = true;
	var isUserSelectionEmpty = true;
	// if current section is not in locked_sequence (section not yet locked)
	if (!isLocked)
	{
		// (LOCKS the section) add the section into the lock sequence
		sectionLocked_sequence.push(facetSectionID);
		// create user selections OBJECT for the SPECIFIC facet
		userSelections_locked[facetSectionID] = {};

		// look through each input checkbox element,
		// and add user selections for the section into userSelections_locked object
		for (var i = 0; i < facetSectionInput.children.length; i+=2)
		{	// get categories/intervals that are checked
			// note: children[i] was refering to the label, and then children[0] refers to the input
			if (facetSectionInput.children[i].children[0].checked == true)
			{	// check whether it is an empty selection
				isUserSelectionEmpty = false;

				// create a category array to store intervals
				if (
userSelections_locked[facetSectionID][facetSectionInput.children[i].children[0].name] == undefined )
				{
userSelections_locked[facetSectionID][facetSectionInput.children[i].children[0].name] = []; }
```

```
                                        // store intervals into category array

        userSelections_locked[facetSectionID][facetSectionInput.children[i].children[0].name].push(facetSectionInput.chi
ldren[i].children[0].value);
                                }
                        }

                        // if user selection is not empty, proceed to disabling the checkboxes, and then finally an AJAX call
                        if (!isUserSelectionEmpty)
                        {       // disable each checkboxes in the most recent locked section
                                for (var i = 0; i < facetSectionInput.children.length; i+=2)
                                {
                                        // disable checkbox
                                        facetSectionInput.children[i].children[0].disabled = true;
                                        // add a hidden input with same name and value to ensure checkbox value
submission (checked checkboxes only)
                                        if (facetSectionInput.children[i].children[0].checked == true)
                                        {
                                                var hiddenCheckboxNode = document.createElement("input");

                                                hiddenCheckboxNode.type = "hidden";
                                                hiddenCheckboxNode.name =
facetSectionInput.children[i].children[0].name;
                                                hiddenCheckboxNode.value =
facetSectionInput.children[i].children[0].value;

                                                // append to inside label (under checkbox)
                                                facetSectionInput.children[i].appendChild(hiddenCheckboxNode);
                                        }
                                }

                                // set lock status to lock
                                var lockElementOutput = "<span class='lockText'
onclick='ajaxFacetFilter(\""+facetSectionID+"\")'>";
                                lockElementOutput += text_lock;
                                lockElementOutput += "</span>";
                                facetSectionLockElement.innerHTML = lockElementOutput;
                        }
                        // however, if user selection is indeed empty
                        else if (isUserSelectionEmpty)
                        {
                                // (rollback) remove the latest section from the locked_sequence
                                if (sectionLocked_sequence.length != 0)
                                {       // rollback the addition of the newest (latest) section
                                        sectionLocked_sequence.splice((sectionLocked_sequence.length - 1), 1);
                                }

                                // as well as removing it from the userSelections_locked object
                                delete userSelections_locked[facetSectionID];

                                // change text back to text_unlock ("unlock")
                                // as user should not be able to lock an empty selection
                                var lockElementOutput = "<span class='lockText'
onclick='ajaxFacetFilter(\""+facetSectionID+"\")'>";
                                lockElementOutput += text_unlock;
                                lockElementOutput += "</span>";
                                facetSectionLockElement.innerHTML = lockElementOutput;

                                // do nothing (no AJAX call) as user selection is empty
                                canAJAX = false;
                                alert('cannot lock an empty selection');
                        }
                }
                // if locked already, then unlock it
                else if (isLocked)
                {       // check if section being unlocked is the latest section that was locked
                        if (lockedIndex == (sectionLocked_sequence.length - 1))
```

```
		{		// if it is the latest section that was locked,
				// go ahead and remove the section from the locked_sequence,
				if (sectionLocked_sequence.length != 0)
				{		// remove last element
						sectionLocked_sequence.splice((sectionLocked_sequence.length - 1), 1);
				}

				// as well as removing it from the userSelections_locked object
				delete userSelections_locked[facetSectionID];

				// set lock status to unlock
				var lockElementOutput = "<span class='lockText'
onclick='ajaxFacetFilter(\""+facetSectionID+"\")'>";
				lockElementOutput += text_unlock;
				lockElementOutput += "</span>";
				facetSectionLockElement.innerHTML = lockElementOutput;
		} else {
				// maintain lock text in element as text_lock ("locked")
				var lockElementOutput = "<span class='lockText'
onclick='ajaxFacetFilter(\""+facetSectionID+"\")'>";
				lockElementOutput += text_lock;
				lockElementOutput += "</span>";
				facetSectionLockElement.innerHTML = lockElementOutput;

				// do nothing (no AJAX call) if can't unlock
				canAJAX = false;
				alert('cannot unlock section');
		}
	}

	var lockedSequenceDiv = document.getElementById('lockedSequenceDisplay');
	var output = "";
	for (var i = 0; i < sectionLocked_sequence.length; i++)
	{
		// get facet name from section id (..._facetName_...)
		var start = sectionLocked_sequence[i].indexOf('_');
		var end = sectionLocked_sequence[i].lastIndexOf('_');
		var diff = end - start;

		// display locked sequence
		output += "<div ";
		if (i == sectionLocked_sequence.length - 1)
		{		output += "class='lockedSequence_lastSection'"; }
		else
		{		output += "class='lockedSequence_notLastSection'"; }
		output += "onclick='ajaxFacetFilter(\""+sectionLocked_sequence[i]+"\")'>";
				output += sectionLocked_sequence[i].substr(start+1, diff -1) + "<span class='hoverX'> X
</span>";
		output += "</div>";
	}
	lockedSequenceDiv.innerHTML = output;

	// get elements for selecting sort order
	var element_sortUnique = document.getElementById('btn_sortUnique');
	var element_sortCount = document.getElementById('btn_sortCount');
	// set sort unique order based on element checked status
	if (element_sortUnique.checked == true) { var sortUnique = true; }
	else if (element_sortCount.checked == true) { var sortUnique = false; }
	else { var sortUnique = true; }

	/************************* AJAX Call *************************/
	if (canAJAX)
	{		// XMLHttp object for AJAX call
		var xmlhttp = new XMLHttpRequest();

		xmlhttp.onreadystatechange = function(){
			if (this.readyState == 4 && this.status == 200){
```

```
                                        if (this.responseText != "")
                                        {       response_decoded = JSON.parse(this.responseText)[0];
                                                response_resultDisplay = JSON.parse(this.responseText)[1];

                                                // -FORMAT for response_decoded indexed array-
                                                // "section -- category~ interval|count||interval|count  |||  category~
interval|count"
                                                // "section -- value|count   |||   value|count"

                                                for (var rd_i = 0; rd_i < response_decoded.length; rd_i++)
                                                {       // full string
                                                        var string_facet = response_decoded[rd_i];
                                                        // section
                                                        var array_stringFacet = string_facet.split("--");
                                                        var string_section = array_stringFacet[0];

                                                        // check if section to display is locked or not
                                                        var toDisplay = true;
                                                        for (var l = 0; l < sectionLocked_sequence.length; l++)
                                                        {       // don't display for the sections that have been locked
                                                                // (toDisplay = false, if section is found inside
sectionLocked)
                                                                if
(sectionLocked_sequence[l].includes(string_section)){
                                                                        toDisplay = false;
                                                                        break;
                                                                }
                                                        }

                                                        // display new unique value-count for each section that's not
locked
                                                        if (toDisplay)
                                                        {
                                                                var targetSectionID =
'facetedFilterForm_'+string_section+'_Section';
                                                                var targetSectionElement =
document.getElementById(targetSectionID).children[1];
                                                                var output = "";

                                                                // if has values
                                                                if (array_stringFacet[1]){
                                                                        // category or value
                                                                        var array_stringCategoryOrValueGroup =
array_stringFacet[1].split("|||");
                                                                        // has category
                                                                        if
(array_stringCategoryOrValueGroup[0].split("~").length == 2)
                                                                        {
                                                                                // Set text strings
                                                                                var facetSection_inputClass_super
        = 'facetedFilterFormSection_inputSuper';
                                                                                var facetSection_inputClass_sub
        = 'facetedFilterFormSection_inputSub';
                                                                                var facetSection_inputName_super
        = string_section+'Select_super[]';
                                                                                var facetSection_inputName_sub
                = string_section+'Select_sub[]';

                                                                                // loop through each category-
intervalGroup pair
                                                                                for (var i = 0; i <
array_stringCategoryOrValueGroup.length; i++)
                                                                                {
                                                                                        var
categoryIntervalGroupPair = array_stringCategoryOrValueGroup[i].split("~");
                                                                                        // unique category
```

```
var uniqueCategory =
categoryIntervalGroupPair[0];

// Additional text strings
for classification
var
facetSection_inputClass_super_classification = string_section+'_super_'+uniqueCategory;
var
facetSection_inputClass_sub_classification = string_section+'_sub_'+uniqueCategory;

// Get total number of
unique_count (category count) from all intervals in each interval group
var categoryCount = 0;
var
array_stringIntervalGroup = categoryIntervalGroupPair[1].split("||");
for (var j = 0; j <
array_stringIntervalGroup.length; j++)
{       var
intervalCountGroupPair = array_stringIntervalGroup[j].split('|');
        // unique count
        var
uniqueCount = intervalCountGroupPair[1];
        categoryCount
+= +uniqueCount;
}

// Display current
category/classification
output += "<label>";
        output +=
"<input type='checkbox' ";
        output +=
        "class='"+facetSection_inputClass_super+" "+facetSection_inputClass_super_classification+"' ";
        output +=
        "name='"+facetSection_inputName_super+"' ";
        output +=
        "value='"+uniqueCategory+"' "
        output +=
        "onclick='toggleFacetCheck(\""+string_section+"\", \"super\", \""+uniqueCategory+"\")' ";
        // if previously
checked
        for (var c_i=0;
c_i < targetSectionElement.children.length; c_i+=2)
        {       if
(targetSectionElement.children[c_i].children[0].value == uniqueCategory){
        if (targetSectionElement.children[c_i].children[0].checked == true)
        { output += "checked='checked' "; }
                }
        }
        output += "/>";
        output +=
uniqueCategory+" ("+categoryCount+") ";
output +=
"</label><br/>";

// Display all intervals
for current category/classification
for (var k = 0; k <
array_stringIntervalGroup.length; k++)
{       var
intervalCountGroupPair = array_stringIntervalGroup[k].split('|');
        // unique
interval
        var
uniqueInterval = intervalCountGroupPair[0];
```

```
// unique count
var uniqueCount = intervalCountGroupPair[1];

output += "<label>";
	output += "<input type='checkbox' ";
	output += 	"class='"+facetSection_inputClass_sub+" "+facetSection_inputClass_sub_classification+"' ";
	output += 	"name='"+facetSection_inputName_sub+"' ";
	output += 	"value='"+uniqueInterval+"' "
	output += 	"onclick='toggleFacetCheck(\""+string_section+"\", \"sub\", \""+uniqueCategory+"\")' ";
// if previously checked
for (var c_i=0; c_i < targetSectionElement.children.length; c_i+=2)
{
	if (targetSectionElement.children[c_i].children[0].value == uniqueInterval){
		if (targetSectionElement.children[c_i].children[0].checked == true)
		{ output += "checked='checked' "; }
	}
}
	output += "/>";
	output += uniqueInterval+" ("+uniqueCount+") ";
output += "</label><br/>";
			}
		}
	}
	// not category
	else if(array_stringCategoryOrValueGroup[0].split("~").length == 1)
	{	// set text strings
		var facetSection_inputClass_select
	= 'facetedFilterFormSection_inputSuper';
		var facetSection_inputName_select	= string_section+'Select[]';

		// loop through each uniqueValue-count pair
		for (var i = 0; i < array_stringCategoryOrValueGroup.length; i++)
		{	var valueCountPair = array_stringCategoryOrValueGroup[i].split("|");
			// unique value
			var uniqueValue = valueCountPair[0];
			// unique count
			var uniqueCount = valueCountPair[1];

			output += "<label>";
				output += "<input type='checkbox' ";
				output += 	"class='"+facetSection_inputClass_select+"' ";
				output += 	"name='"+facetSection_inputName_select+"' ";
```

```
                                                                                output +=
            "value="'+uniqueValue+'" ";
                                                                                // if previously
checked
                                                                                for (var c_i=0;
c_i < targetSectionElement.children.length; c_i+=2)
                                                                                {           if
(targetSectionElement.children[c_i].children[0].value == uniqueValue){
            if (targetSectionElement.children[c_i].children[0].checked == true)
            { output += "checked='checked' "; }
                                                                                        }
                                                                                }
                                                                                output += "/>";
                                                                                output +=
uniqueValue+" ("+uniqueCount+") ";
                                                                                        output +=
"</label><br/>";
                                                                                    }
                                                                                }
                                                                            } // no values
                                                                            else { var output = "No Matching Filters"; }

                                                                            // add filtered facet input to the element
                                                                            targetSectionElement.innerHTML = output;
                                                                    }
                                                            }

                                                            // Display results
                                                            results_section_element = document.getElementById('results_section');
                                                            if (typeof response_resultDisplay !== 'undefined' &&
response_resultDisplay.length > 0)
                                                            {           var results_output = "";
                                                                        for (var rc_i = 0; rc_i < response_resultDisplay.length; rc_i++){
                                                                                    results_output += "<div>";
                                                                                            results_output += "<div>";
                                                                                                    results_output += "<div><span>"

                    + response_resultDisplay[rc_i]['movieType']

                    + "<br/>Rating: "+response_resultDisplay[rc_i]['contentRating']

            + "</span></div>";
                                                                                                    results_output +=
"<div><span>"+response_resultDisplay[rc_i]['movieTitle']+"</span></div>";
                                                                                                    results_output += "<div>meta
score<br/>"+response_resultDisplay[rc_i]['metaScore']+"/100</div>";
                                                                                                    results_output +=
"<div>rating<br/>"+response_resultDisplay[rc_i]['imdbRating']+"/10.0</div>";
                                                                                                    results_output +=
"<div>votes<br/>"+response_resultDisplay[rc_i]['imdbVotes']+"</div>";
                                                                                            results_output += "</div>";

                                                                                            results_output += "<div>";
                                                                                                    results_output +=
"<br/><div><span>Genre: </span>";
                                                                                                    var prefix = "";
                                                                                                    for (var i = 0; i <
response_resultDisplay[rc_i]['movieGenre'].length; i++)
                                                                                                    {
                                                                                                            results_output +=
!(prefix)?"":prefix;
                                                                                                            results_output +=
response_resultDisplay[rc_i]['movieGenre'][i];
                                                                                                            prefix = ", ";
                                                                                                    }
```

```
                                        results_output += "</div>";

                                        results_output +=
"<br/><div><span>Language: </span>";
                                        prefix = "";
                                        for (var i = 0; i <
response_resultDisplay[rc_i]['movieLanguage'].length; i++)
                                        {
                                            results_output +=
!(prefix)?"":prefix;
                                            results_output +=
response_resultDisplay[rc_i]['movieLanguage'][i];
                                            prefix = ", ";
                                        }
                                        results_output += "</div>";

                                        results_output +=
"<br/><div><span>Country: </span>";
                                        prefix = "";
                                        for (var i = 0; i <
response_resultDisplay[rc_i]['movieCountry'].length; i++)
                                        {
                                            results_output +=
!(prefix)?"":prefix;
                                            results_output +=
response_resultDisplay[rc_i]['movieCountry'][i];
                                            prefix = ", ";
                                        }
                                        results_output += "</div><br/>";

                                        results_output += "<a
target='_blank' href='http://www.imdb.com/title/"
    + response_resultDisplay[rc_i]['imdbRefID']
    + "'><div class='ext_link_imdb'>";
                                            results_output += "To
IMDb Page";
                                        results_output += "</div></a>";
                                    results_output += "</div>";

                                    results_output += "<div>";
                                        results_output += "<div>Seasons:
";
                                        results_output +=
(response_resultDisplay[rc_i]['totalSeason'] == 0)?'No':response_resultDisplay[rc_i]['totalSeason']
                                        results_output += " season</div>";
                                        results_output += "<div>Duration:
"+response_resultDisplay[rc_i]['movieDuration']+" mins</div>";
                                        results_output += "<div>Released
Since: "+response_resultDisplay[rc_i]['releasedDate']+"</div>";
                                    results_output +=  "</div>";

                                results_output += "</div>";
                            }
                            results_section_element.innerHTML = results_output;
                        } else { results_section_element.innerHTML = "<div>No Matching Movie
Content</div>"; }
                    } //else {  console.log("empty response");             }
                }
            };

            // url to respond to AJAX call for progressive facet search
            var url = "user_progressiveFacetSearch.php";

            // open POST request, and set header for POST request
            xmlhttp.open("POST", url, true);
```

```
		xmlhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");

		// convert array to JSON string
		var userSelections_locked_jsonString = JSON.stringify(userSelections_locked);
		var var_send = "userSelections="+encodeURIComponent(userSelections_locked_jsonString);
		var_send += "&sortOrder="+sortUnique;
		// send AJAX POST request
		xmlhttp.send(var_send);
	} // finish AJAX call

}

// JavaScript : Function to sort by unique values alphabetically, or by counts highest-first
function sortUniqueOrCount(sortUnique = true)
{	//var userSelectionChecked = {};

	/* Get All checked values *
	var filterForm_sections = document.getElementById('facetedFilterForm_Content').children;
	for (var ff_i = 0; ff_i < filterForm_sections.length; ff_i++)
	{
		var filterForm_sections_Element = filterForm_sections[ff_i];
		var filterForm_sections_ElementID = filterForm_sections_Element.id;

		// look into each input element for the each current section
		for (var ffc_i = 0; ffc_i < filterForm_sections_Element.children[1].children.length; ffc_i += 2 )
		{	var current_input = filterForm_sections_Element.children[1].children[ffc_i].children[0];

			// if section is checked, add into user selections array
			if(current_input.checked == true){
				if (userSelectionChecked[filterForm_sections_ElementID] == undefined)
				{ userSelectionChecked[filterForm_sections_ElementID] = {}; }

				// create a category array to store intervals
				if ( userSelectionChecked[filterForm_sections_ElementID][current_input.name] ==
undefined )
				{ userSelectionChecked[filterForm_sections_ElementID][current_input.name] = [];
}

				// store intervals into category array

	userSelectionChecked[filterForm_sections_ElementID][current_input.name].push(current_input.value);
			}
		}
	}*/

	// XMLHttp object for AJAX call
	var xmlhttp = new XMLHttpRequest();

	/*
	 * Function to execute after receiving response
	 */
	xmlhttp.onreadystatechange = function(){
		if (this.readyState == 4 && this.status == 200){
			if (this.responseText != "")
			{	response_decoded = JSON.parse(this.responseText)[0];

				// -FORMAT for response_decoded indexed array-
				// "section : category~ interval|count||interval|count  |||  category~ interval|count"
				// "section : value|count   |||   value|count"

				for (var rd_i = 0; rd_i < response_decoded.length; rd_i++)
				{	// full string
					var string_facet = response_decoded[rd_i];
					// section
					var array_stringFacet = string_facet.split("--");
					var string_section = array_stringFacet[0];
```

```
// check if section to display is locked or not
var toDisplay = true;
for (var l = 0; l < sectionLocked_sequence.length; l++)
{	// don't display for the sections that have been locked
	// (toDisplay = false, if section is found inside sectionLocked)
	if (sectionLocked_sequence[l].includes(string_section)){
		toDisplay = false;
		break;
	}
}

// display new unique value-count for each section that's not locked
if (toDisplay)
{	var targetSectionID =
'facetedFilterForm_'+string_section+'_Section';
	var targetSectionElement =
document.getElementById(targetSectionID).children[1];
	var output = "";

	// if has values
	if (array_stringFacet[1]){
		// category or value
		var array_stringCategoryOrValueGroup =
array_stringFacet[1].split("|||");

		// has category
		if
(array_stringCategoryOrValueGroup[0].split("~").length == 2)
		{	// Set text strings
			var facetSection_inputClass_super	=
'facetedFilterFormSection_inputSuper';
			var facetSection_inputClass_sub		=
'facetedFilterFormSection_inputSub';
			var facetSection_inputName_super	=
string_section+'Select_super[]';
			var facetSection_inputName_sub
	= string_section+'Select_sub[]';

			// loop through each category-intervalGroup
pair
			for (var i = 0; i <
array_stringCategoryOrValueGroup.length; i++)
			{	var categoryIntervalGroupPair =
array_stringCategoryOrValueGroup[i].split("~");

				// unique category
				var uniqueCategory =
categoryIntervalGroupPair[0];

				// Additional text strings for
classification
				var
facetSection_inputClass_super_classification = string_section+'_super_'+uniqueCategory;
				var
facetSection_inputClass_sub_classification = string_section+'_sub_'+uniqueCategory;

				// Get total number of
unique_count (category count) from all intervals in each interval group
				var categoryCount = 0;
				var array_stringIntervalGroup =
categoryIntervalGroupPair[1].split("||");
				for (var j = 0; j <
array_stringIntervalGroup.length; j++)
				{
					var
intervalCountGroupPair = array_stringIntervalGroup[j].split('|');

					// unique count
					var uniqueCount =
intervalCountGroupPair[1];
```

```
                                                                        categoryCount +=
+uniqueCount;
                                                                }

                                                                // Display current
category/classification
                                                                output += "<label>";
                                                                        output += "<input
type='checkbox' ";
                                                                        output +=
        "class='"+facetSection_inputClass_super+" "+facetSection_inputClass_super_classification+"' ";
                                                                        output +=
        "name='"+facetSection_inputName_super+"' ";
                                                                        output +=
        "value='"+uniqueCategory+"' "
                                                                        output +=
        "onclick='toggleFacetCheck(\""+string_section+"\", \"super\", \""+uniqueCategory+"\")' ";
                                                                        // if previously checked
                                                                        for (var c_i=0; c_i <
targetSectionElement.children.length; c_i+=2)
                                                                        {       if
(targetSectionElement.children[c_i].children[0].value == uniqueCategory){
                                                                                        if
(targetSectionElement.children[c_i].children[0].checked == true)
                                                                                        {
output += "checked='checked' "; }
                                                                                }
                                                                        }
                                                                        output += "/>";
                                                                        output +=
uniqueCategory+" ("+categoryCount+") ";
                                                                output += "</label><br/>";

                                                                // Display all intervals for current
category/classification
                                                                for (var k = 0; k <
array_stringIntervalGroup.length; k++)
                                                                {       var
intervalCountGroupPair = array_stringIntervalGroup[k].split('|');
                                                                        // unique interval
                                                                        var uniqueInterval =
intervalCountGroupPair[0];
                                                                        // unique count
                                                                        var uniqueCount =
intervalCountGroupPair[1];

                                                                        output += "<label>";
                                                                                output +=
"<input type='checkbox' ";
                                                                                output +=
        "class='"+facetSection_inputClass_sub+" "+facetSection_inputClass_sub_classification+"' ";
                                                                                output +=
        "name='"+facetSection_inputName_sub+"' ";
                                                                                output +=
        "value='"+uniqueInterval+"' "
                                                                                output +=
        "onclick='toggleFacetCheck(\""+string_section+"\", \"sub\", \""+uniqueCategory+"\")' ";
                                                                                // if previously
checked
                                                                                for (var c_i=0;
c_i < targetSectionElement.children.length; c_i+=2)
                                                                                {       if
(targetSectionElement.children[c_i].children[0].value == uniqueInterval){
        if (targetSectionElement.children[c_i].children[0].checked == true)
        { output += "checked='checked' "; }
```

```
                                                }
                                            }
                                            output += "/>";
                                            output +=
uniqueInterval+" ("+uniqueCount+") ";
                                        output +=
"</label><br/>";
                                    }
                                }
                            }
                            // not category
                            else
if(array_stringCategoryOrValueGroup[0].split("~").length == 1)
                            {       // set text strings
                                var facetSection_inputClass_select      =
'facetedFilterFormSection_inputSuper';
                                var facetSection_inputName_select       =
string_section+'Select[]';

                                // loop through each uniqueValue-count pair
                                for (var i = 0; i <
array_stringCategoryOrValueGroup.length; i++)
                                {       var valueCountPair =
array_stringCategoryOrValueGroup[i].split("|");

                                    // unique value
                                    var uniqueValue =
valueCountPair[0];

                                    // unique count
                                    var uniqueCount =
valueCountPair[1];


                                    output += "<label>";
                                        output += "<input
type='checkbox' ";
                                        output +=
        "class='"+facetSection_inputClass_select+"' ";
                                        output +=
"name='"+facetSection_inputName_select+"' ";
                                        output +=
"value='"+uniqueValue+"' ";

                                        // if previously checked
                                        for (var c_i=0; c_i <
targetSectionElement.children.length; c_i+=2)
                                        {       if
(targetSectionElement.children[c_i].children[0].value == uniqueValue){
                                                        if
(targetSectionElement.children[c_i].children[0].checked == true)
                                                        {
output += "checked='checked'"; }

                                            }
                                        }
                                        output += "/>";
                                        output +=
uniqueValue+" ("+uniqueCount+") ";
                                    output += "</label><br/>";
                                }
                            }
                        } else { var output = "No Matching Filters"; }

                        // add filtered facet input to the element
                        targetSectionElement.innerHTML = output;
                    }
                }
            } //else {  console.log("empty response");         }
        }
    }
```

```
        // url to respond to AJAX call for progressive facet search
        var url = "user_progressiveFacetSearch.php";

        // open POST request, and set header for POST request
        xmlhttp.open("POST", url, true);
        xmlhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");

        // convert array to JSON string
        var userSelections_locked_jsonString = JSON.stringify(userSelections_locked);
        var var_send = "userSelections="+encodeURIComponent(userSelections_locked_jsonString);
        /*var userSelectionChecked_jsonString = JSON.stringify(userSelectionChecked);
        var var_send = "userSelections="+encodeURIComponent(userSelectionChecked_jsonString);*/
        var_send += "&sortOrder="+sortUnique;
        // send AJAX POST request
        xmlhttp.send(var_send);

        // finish AJAX call
}
```