

ITP 303 Final Project Summary

INSTRUCTIONS: Type out your answer directly below each question. If a question does not apply to your project, type in N/A. When finished, save this file as a PDF and upload it to the itpwebdev server via FileZilla. Then, add a link to this PDF in your student_page.html. Label it "Final Project Summary."

BASICS

Your name?

Abasiakan Udobong

What is the topic of your final project?

Biskan Lists, a web app that searches for movies and saves movie lists for people

Who is the intended audience?

Movie lovers like myself who often have a lot to do and can't keep track of all the movies they want to watch in the future, and also have a habit of forgetting if they've watched a movie or not

Provide a brief summary of the functionalities of your project.

KEY FEATURE: LOGIN

Biskan Lists features a personalized user experience where individual users can create and access their own lists and add and remove list items as they please. This is facilitated via a PHP server that communicates with a MySQL server in which data such as login information and lists are stored. The login page features user side verification for login and registration details to ensure privacy and avoid wasting server time.

KEY FEATURE: SEARCH

In order to effectively use the site, it is actually very intuitive. You may search movies as a guest but attempting to add items will result in a prompt to log in or sign up. User sessions are kept track of and can be ended via a logout. Searches display information such as synopsis and trailer videos for most search results.

KEY FEATURE: LISTS

To add a movie, a user can either click on the add button from the home screen (which shows movies currently in theatres) or from the search results screen (which enables a user to search any movie they please). To add a movie a user must first designate which list the movie is to be added to. As a default, upon registration a list called 'Favourites' is created for the user. Other lists can be accessed via the dropdown list on the Lists page. Users may add the same movie to multiple different lists.

KEY FEATURE: WATCHED

If a user has been able to watch their desired movie, they can toggle the 'watched' switch and they will be able to see whether they have watched a movie or not upon return to the sight.

FRONT-END

What is the total page count?

4 pages (Home page, List page, Login Page and Search Results page)

List the names of any external stylesheet used in this project below.

All styles were done via external stylesheet. Stylesheets that were used on all pages include the shared_styles.css and responsive.css, both located in the designated css folder.

List any CSS libraries/frameworks used in this project (e.g. Bootstrap) below.

Bootstrap (for the toggle switches on lists.php)

List any JavaScript libraries/frameworks used in this project below.

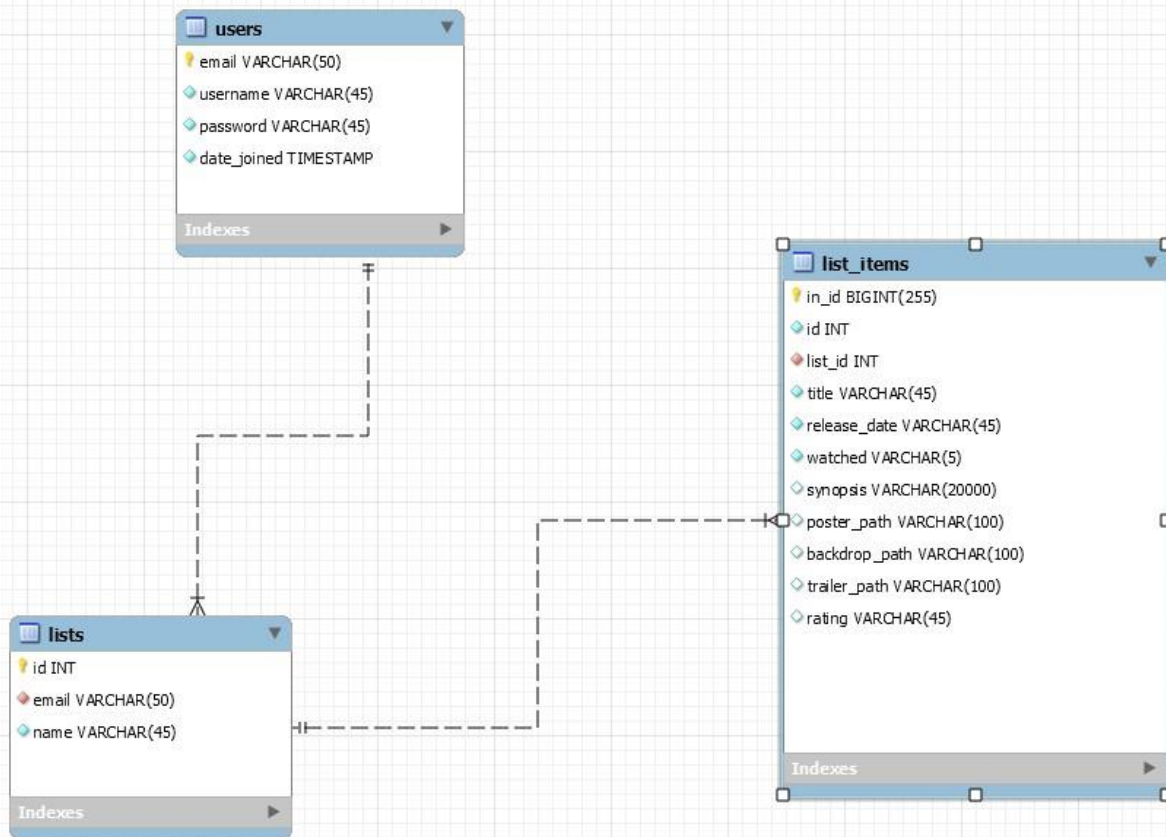
JQuery (for the toggle switches on lists.php)

How does your project meet the Interactivity requirement?

If you try to add a movie to a list, a pop up appears asking you to pick a list to add to. This is all done with event listeners. The form submissions are mostly done with event listeners. The switches on the lists page for watched also implement interactivity.

BACK-END

Attach a screenshot of the final database diagram (just the diagram, do not include any records) below.



Where in your project do you insert a record to the database (the 'C' in CRUD)?

Users, list items and the lists themselves are all dynamically created in the app.

Where in your project do you search and display record(s) from the database (the 'R' in CRUD)?

On the Lists page, all the records belonging to a user for specific lists are displayed in a list format.

Where in your project do you update and existing record(s) the database (the 'U' in CRUD)?

In order to change whether you have watched a movie or not, the database allows you update the 'watched' value to yes or no which saves it and allows you to come back later and see the appropriate status.

Where in your project do you delete existing record(s) from the database (the 'D' in CRUD)?

You can delete any movie item from any list at any time from the Lists page.

MISC

What two “extra” requirements did you implement and where can they be found?

I made use of PHP’s sessions features to allow users to login and maintain their session even after leaving the website (without logging out).

I used a JavaScript animation plugin to simulate a loading screen for the page. This can be seen on the home and search results pages right before the pages’ contents load.

If you used any APIs or JS plugins/frameworks as one of the extra requirements, list the name and a link to their documentation.

If your project requires any admin credentials (i.e. only admin users can access a certain page), list the credentials below.

Provide any other information that you think the grader/instructor should know when grading your final project below.

NOTE

My Chrome browser sometimes displays this error:

[Deprecation] Synchronous XMLHttpRequest on the main thread is deprecated because of its detrimental effects to the end user's experience. For more help, check <https://xhr.spec.whatwg.org/>.

which does not allow the API-sourced content on the home page to load. I have the same source code for a synchronous request on a different page (js/search_results.js) and it works just fine. I think this might be a local issue with my own browser though.