

Congenial Broccolis: Wen Hao Dong (Jal Hordan), Austin Ngan (Gerald), Liesel Wong (King Hagrid), Rachel Xiao (Mooana)
SoftDev
P01 -- ArRESTed Development
2021-12-09

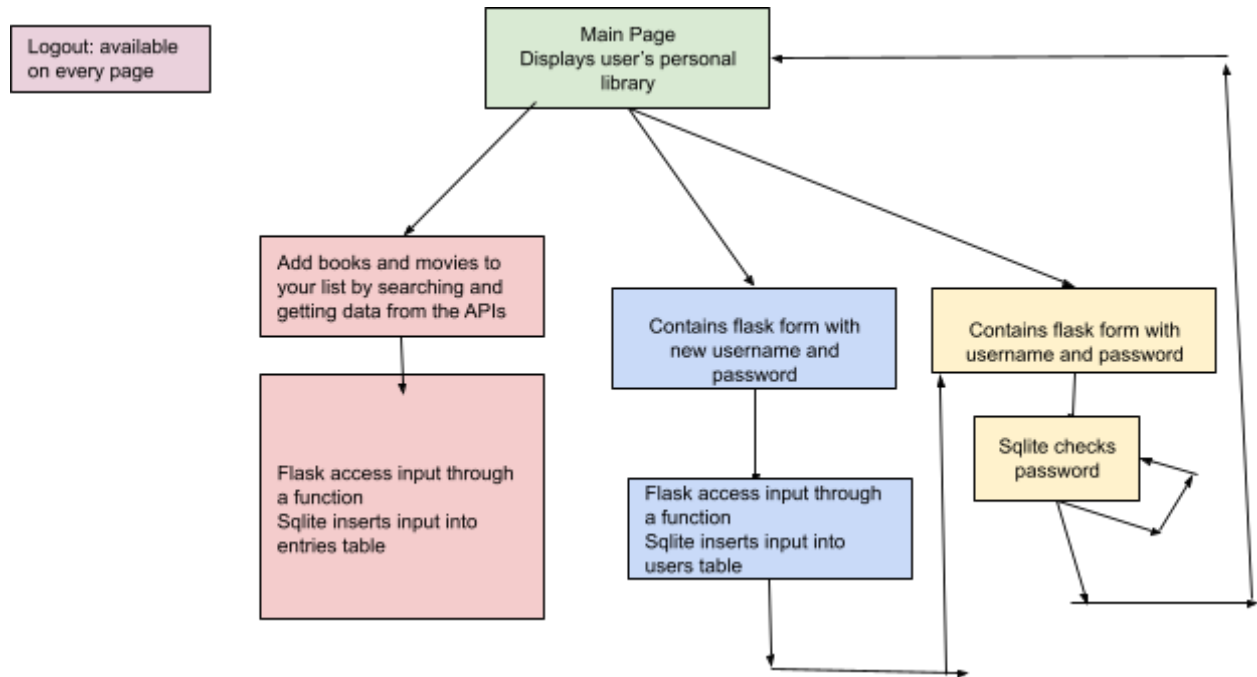
- ☒ ~~A list of program components with role of each specified. (e.g., a car engine is comprised of various components: carburetor, alternator, radiator, spark plugs, etc. Each must perform its role for the engine to do its overall job.)~~
- ☒ ~~Explanation of how each component relates to the others.~~
 - ☒ ~~Component map visualizing relationships between components.~~
- ☒ ~~Database Organization (tables? Relationships b/t tables? etc.)~~
- ☒ ~~Site map for front end~~
 - ☒ ~~Represent each page you envision for your site.~~
 - ☒ ~~Show linkages conveying all possible pathways for a user traversing site.~~
- ☒ ~~A breakdown of the different tasks required to complete this project~~
 - ☒ ~~Include assignments of each task to each group member~~
- ☒ ~~Clearly labeled section delineating APIs you will use.~~
- ☒ ~~Clearly labeled section noting which front-end framework you will use and why/how. (E.g., Which of its features will you make integral use of?)~~
- ☒ ~~Target "ship date."~~
- ☐ Amalgamate these components into a single PDF, store in designated location.

Components and Purpose:

- SQL database:
 - used to keep track of login information
- Bootstrap:
 - make site pretty
- Jinja2:
 - Used to create a layout for web pages the user will interact with
- Flask:
 - Uses Jinja2 and HTML templates to create the web pages that the user will see and interact with
 - Data is pulled from the database in order to populate each webpage
 - Will get username and passwords from the front end when a user logs in or registers
 - Will create sessions for each user when they are logged in
 - NOTE: login/logout is subject to removal if deemed impractical
- Main Page:

- User's personal library containing all the books/movies they have consumed
- Contains an add book and add movie button to add to their respective database
- Add Book/Movie:
 - Search button (searching through IMDB or OpenLibrary)
 - Returns entries that have been found through the API
 - User can then choose which entry they want to add to their personal library

Component Map:



Database Organization:

users:

id (integer primary key)	username (text)	password (text)
--------------------------	-----------------	-----------------

movies (for caching movie information):

id (integer primary key) the same "tt" id imdb uses	title (text)	director? (text)
--	--------------	------------------

books (for caching book information):

id (integer primary key) the isbn of the book	title (text)	author (text)
--	--------------	---------------

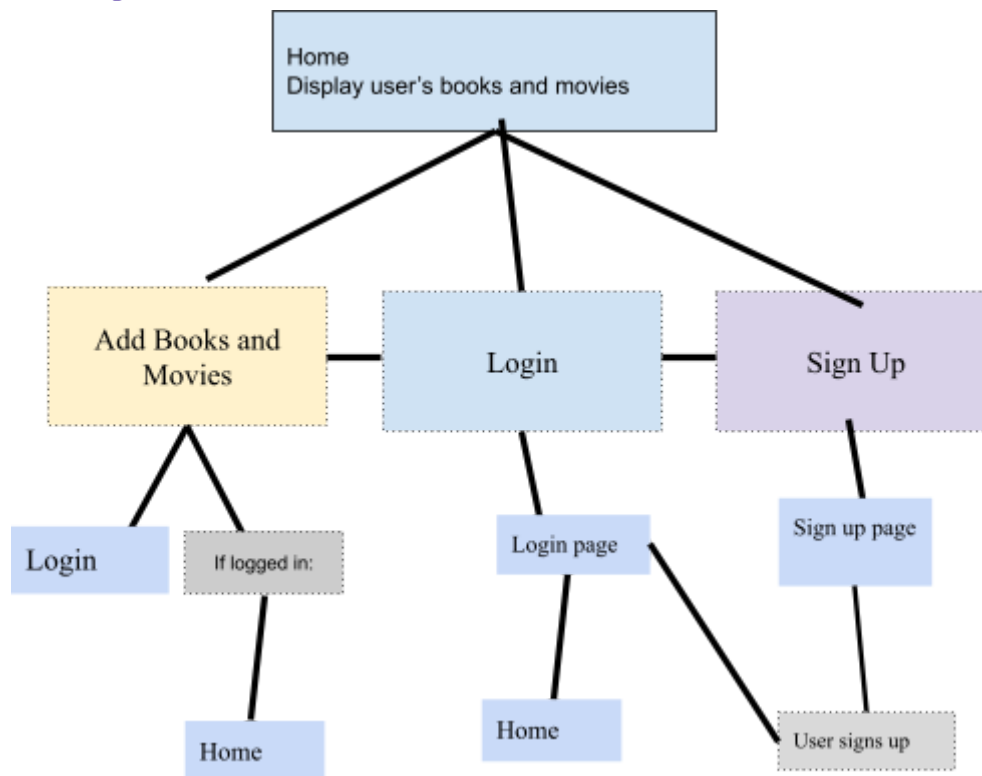
entries: whenever a user adds a book or movie to their library, a new entry is added with their user id and the id of the thing they added

id (integer primary key)	type (text) either "book" or "movie"	user_id (integer)	media_id (integer) the id of the book/movie	complete (boolean)
--------------------------	--------------------------------------	-------------------	---	--------------------

APIs used:

- IMDB API
 - Provides information (cast, ratings, trailers, posters) about movies with many ways of searching for them
- OpenLibrary API
 - Retrieves information about books using many book APIs
- NYT Movie Reviews and Books API
 - Provides subjective information about movies or books

Site Map:



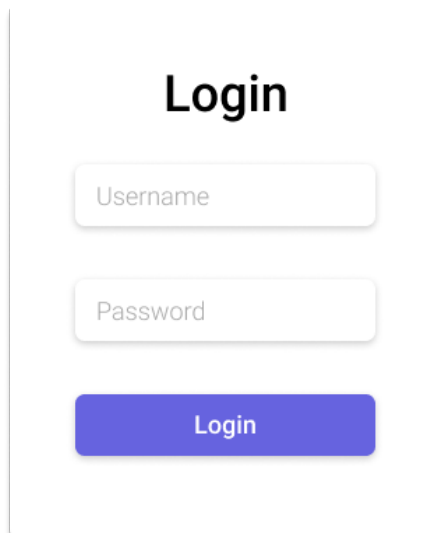
Front End Framework (Bootstrap):

We're using Bootstrap because we like the CSS styling they have. The default styling is cleaner and easier to implement, thus, easily giving our website more of a professional feel to it. We believe it has more features than Foundation. It also has a lot more documentation than Foundation because it is a very popular framework. Compared to Foundation, which for the most part requires a download, Bootstrap allows you to use the CSS through 1 to 2 external stylesheets. Lastly, it is more difficult to find ways to use Javascript in Foundation than in Bootstrap.

Template Representations:

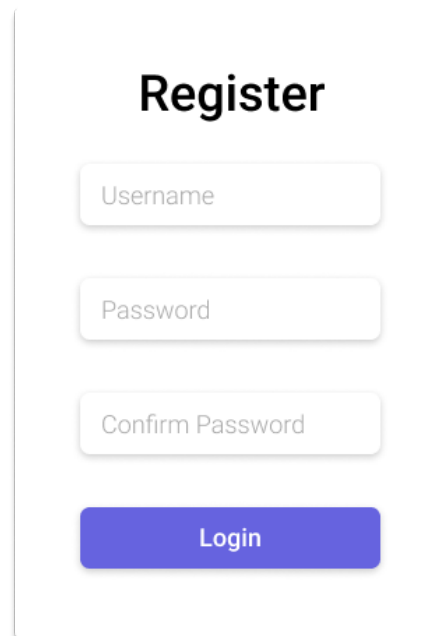
NOTE: design and color are subject to change (in particular the library "finished" switches, cause it may need JS to work properly)

Login:



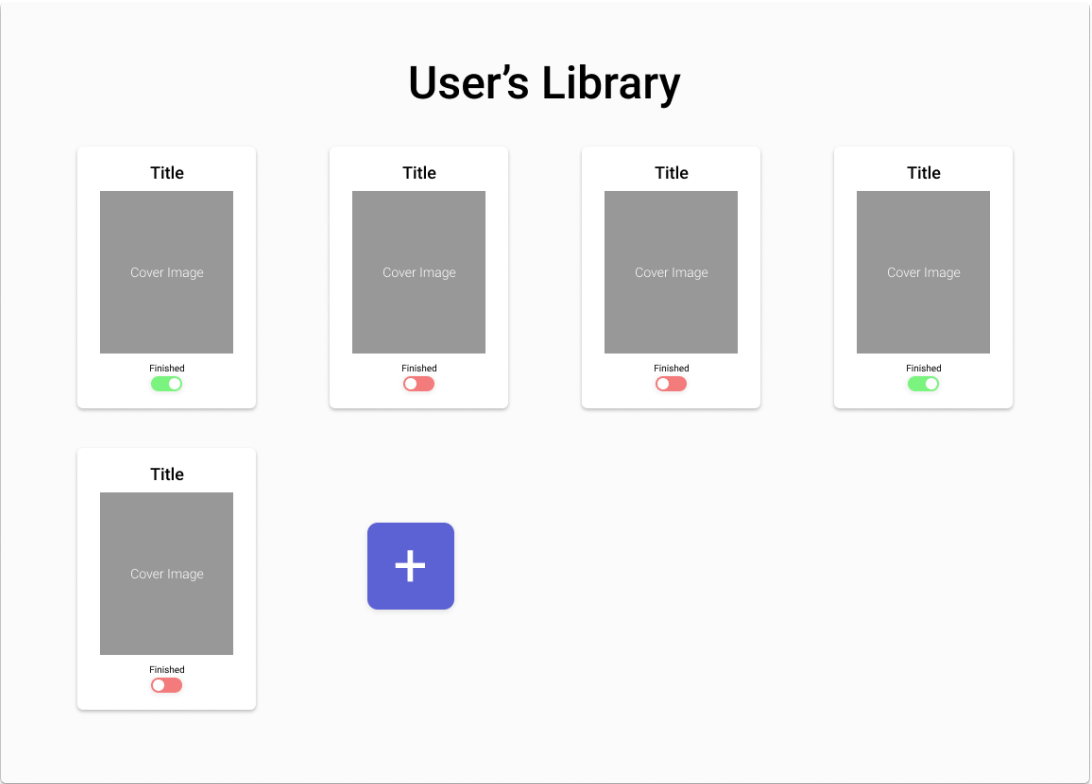
A login form template with a light gray background and a white border. It features a title "Login" in bold black text at the top. Below the title are two input fields: "Username" and "Password", both with light gray borders and rounded corners. At the bottom is a blue button with the text "Login" in white.

Register:

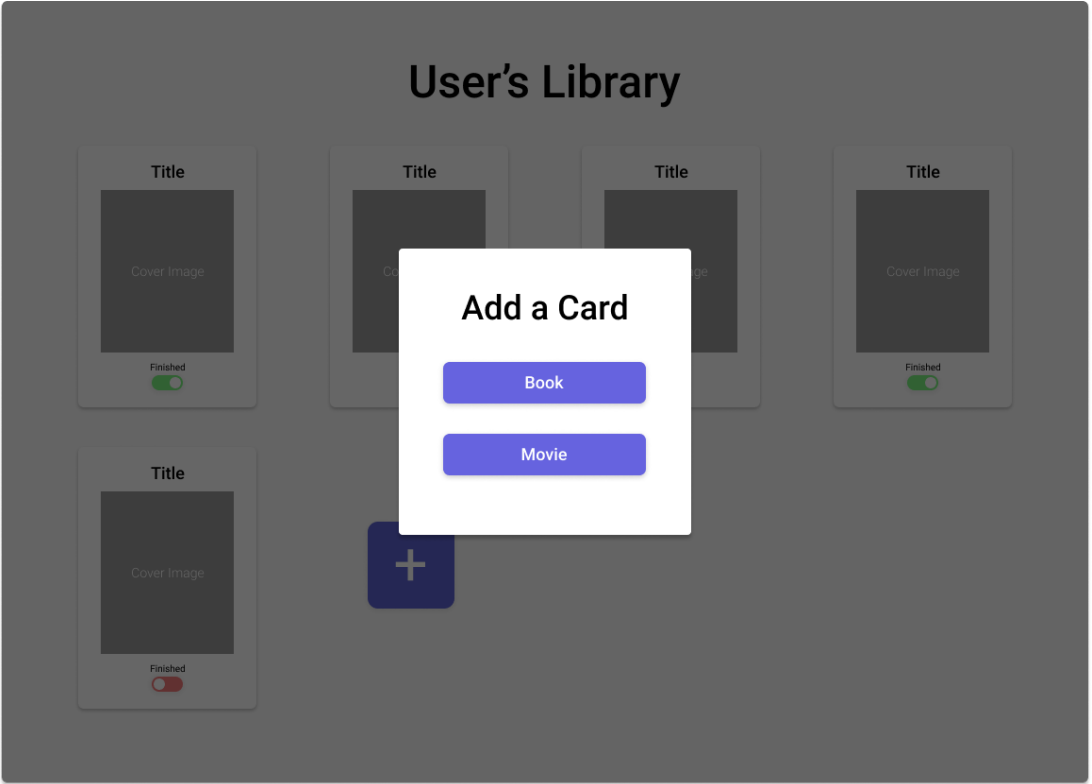


A register form template with a light gray background and a white border. It features a title "Register" in bold black text at the top. Below the title are three input fields: "Username", "Password", and "Confirm Password", all with light gray borders and rounded corners. At the bottom is a blue button with the text "Login" in white.

Library:



Select card type modal (pop up)



Search:

Search for Book

Search for Movie

Results:

Results for "Title"

Title

Cover Image

Title

Cover Image

Title

Cover Image

Title

Cover Image

Title

Cover Image

Title

Cover Image

Title

Cover Image

Division of Tasks:

Wen Hao: Project Manager and Flask

Austin: Flask

Liesel: Bootstrap / Templates

Rachel: Database

Target 'ship date': 12/23/21