Find That Show Final Document

Prepared by Kahle Broadnax, Sarah Kercheval, Ben Yokoyama, Preston Fenimore, and Robin Purnama

Central Washington University

2/27/2022

CS 481

Table Of Contents

Robin Purnama:	26
Sarah Kercheval:	26
Razvan Andonie:	26

1. Introduction

The following document is meant to serve as full and comprehensive documentation of the capstone project created by team 5 in CS481, winter quarter at CWU. The focus of the project is providing an efficient search of television shows and movies across a range of all current streaming services via web scraping.

1.1 Purpose:

This document is a required submission and is intended to be turned in for a grade. Beyond the immediate purpose, it also serves as a means of documenting the project in its entirety. Instead of creating multiple documents such as software specifications requirements and design documents, this document will fulfill the purpose of those and more, which will be covered in the scope section next. This document is also intended to help with the creation of a presentation for this project since the scope of both is largely equivalent.

1.2 Scope:

This document will cover a range of topics, as laid out by the assignment requirements. This includes but may not be limited to:

- Problem Statement
- Requirements
- Low/High-Level design
- Software tools requirements (dependencies, packages, libraries, languages, containers, operating systems, etc.)
- Hardware requirements (CPU, memory, TCP/IP connection, etc.
- Software documentation (code organization, different modules, functions, parameters, algorithms, etc.)
- Installation
- User documentation (how the software can be operated once it is installed, screen captures, explanations, etc.)
- Conclusion (overview of the document, what the team learned, what we would have done differently)

1.3 Intended Audience:

The intended audience is Professor Razvan Andonie of Central Washington University, as he is the instructor for the class CS 481. Outside audiences could also view this document to evaluate the project or the members of the team.

1.4 Definitions and Acronyms:

Below is a running list of definitions and acronyms. More may be added as the document expands:

SRS: Software requirements specifications
CWU: Central Washington University
Q&A: Questions and answers
Web scraper: A program that parses websites for relevant information depending on the programmer's needs
VPN: Virtual Private Network. Allows for users to reassign their web-based location, essentially giving access to content outside of their region
Dub: A vocal track intended to replace the original dialogue of a show or movie. Typically, this is in a different language than the original to allow non-native speakers to understand a show or movie without resorting to subtitles.
Sub: Short for subtitles or subtitled, usually meant for English viewers to understand a show in a different language but can be in the same language as the vocals in a show or movie.

2. Problem Statement:

This section will outline the problem this software is attempting to solve and lay out some of the methods the software is/has employed to remedy this problem.

2.1 Ideal:

Ideally, streaming services are meant to allow users to easily and quickly access the shows they are interested in to watch them on a single site or application. This was true when Netflix was the main online streaming service available.

2.2 Reality:

In reality, there are dozens of mainstream streaming services available, and many more depending on how to explicitly define “streaming services”. Each of these sites also holds different licenses for shows and exclusive rights to some shows, and these licenses and rights can shift between platforms without warning.

2.3 Consequences:

- Users aren’t given an easy way to locate shows anymore, which makes some users predisposed against investing in any service
- Catalogs outside of the user's region can also differ greatly, but given the number of sites it’s hard to search them all efficiently
- The inconvenience of both of these factors may lead many to piracy as a means to avoid poor service from the lack of easily accessible information regarding what shows are available on each platform in every region

2.4 Proposal:

Developing software that acts as both a search engine and database for shows and movies across each platform should improve the user experience with streaming services

2.5 Overall Description of software

The software to be developed is a website that allows for the convenient search of any television show or movie across a range of current streaming services, both free and paid. The GUI frontend would allow for the selection of specific filters, such as by streaming service, by price, by region, or by genre. The search aspect of it is achieved via a backend web scrape search. The main backend feature that is desired to be implemented is regional searching, so if a show was available in the UK for instance, but the user was in the US, the user would be made aware of that availability.

3.1 Requirements:

This section will lay out the functional and non-functional requirements, as well as an analysis of them.

3.1.1 Requirement #1: The software will accept typed user input pertaining to what show they want to search for

Analysis: This is a functional requirement
Validity: This is congruent with the software description
Feasibility: Accepting user input is easy to implement
Consistency: This doesn't interfere with other tasks the software will perform
Prioritization: This is mandatory for general software functionally

3.1.2 Requirement #2: The software will search for a given show the user has inputted using web scraping

Analysis: This is a functional requirement
Validity: This is congruent with the software description
Feasibility: Web scraping is a well-documented functionally
Consistency: This doesn't interfere with other tasks the software will perform
Prioritization: This is mandatory for general software functionally

3.1.3 Requirement #3: The software will be designed for personal computers as a website

Analysis: This is a functional requirement
Validity: This is not incongruent with the software description
Feasibility: Designing websites is well-documented
Consistency: This doesn't interfere with other tasks the software will perform
Prioritization: This is mandatory for general software functionally, but the project could expand or shift to mobile development if more time was permitted or encouraged

3.1.4 Requirement #4: The software should utilize servers (if applicable)

Analysis: This is a functional requirement
Validity: This is not incongruent with the software description

Feasibility: Utilizing servers to host websites is well documented
Consistency: This doesn't interfere with other tasks the software will perform
Prioritization: This is not mandatory for the software to function, and should be worked on after a local website is shown to function. Since this software has no budget, it is not a priority at the moment

3.1.5 Requirement #5: Cookie use must be agreed to by the user

Analysis: This is a functional requirement
Validity: This is not incongruent with the software description
Feasibility: Cookies are well documented
Consistency: This doesn't interfere with other tasks the software will perform and is more or less a requirement for most websites
Prioritization: This is not mandatory for the software to function, but could improve the user experience by remembering preferences. Accounts could perform the same functionally, but those have less priority than cookies at the moment

3.1.6 Requirement #6: The software should have the ability to access external websites to direct the user to

Analysis: This is a functional requirement
Validity: This is not incongruent with the software description
Feasibility: Connecting websites via hyperlinks and links is well documented
Consistency: This doesn't interfere with other tasks the software will perform
Prioritization: This is not mandatory for the software to function, and could face some integration issues depending on how extensive the integration is (link directly to the web page the show was found on or just the website)

3.1.7 Requirement #7: Functional testing needs to be performed throughout the creation of the software

Analysis: This is a functional requirement
Validity: This is not incongruent with the software description
Feasibility: Unit tests are well documented, and required to assess the functionality of a software
Consistency: This doesn't interfere with other tasks the software will perform
Prioritization: This is not mandatory for the software to function, but is encouraged greatly

3.1.8 Requirement #8: The software will allow the user to search by genre or language instead of by show

Analysis: This is a functional requirement
Validity: This is congruent with the software description
Feasibility: Creating genre filters to input into the web scraper is documented
Consistency: This doesn't interfere with other tasks the software will perform
Prioritization: This is not mandatory for the software to function, but would increase the user experience significantly for users who want to search for shows they don't know the name of

3.1.9 Requirement #9: The software will have filter settings to enable and disable specific search options such as subscription price and service

Analysis: This is a functional requirement
Validity: This is congruent with the software description
Feasibility: Changing the parameters that go into a web scraper is well-documented
Consistency: This doesn't interfere with other tasks the software will perform, except for limiting search results
Prioritization: This is not mandatory for the software to function, but would increase the user experience significantly for users who want to limit or specify certain streaming services in case they don't own all of them

3.1.10 Requirement #10: The software will have options to search via geographic region to provide region-locked options

Analysis: This is a functional requirement
Validity: This is congruent with the software description
Feasibility: VPNs are well documented
Consistency: This doesn't interfere with other tasks the software will perform, except for limiting search results
Prioritization: This is not mandatory for the software to function, but would increase the user experience significantly for users who want to search outside of their region for shows they can view with their VPN

3.1.11 Requirement #11: The website should have a dark mode option

Analysis: This is a functional requirement
Validity: This is not incongruent with the software description

Feasibility: Dark mode is a common setting on most modern applications and software
Consistency: This doesn't interfere with other tasks the software will perform
Prioritization: This is not mandatory for the software to function, but would increase the user experience significantly since many users enjoy using dark mode

3.1.12 Requirement #12: The software should be easy to use

Analysis: This is a non-functional requirement
Validity: This is not incongruent with the software description
Feasibility: Making a user-friendly interface is well-documented
Consistency: This doesn't interfere with other tasks the software will perform
Prioritization: This is not mandatory for the software to function, but would increase the user experience significantly

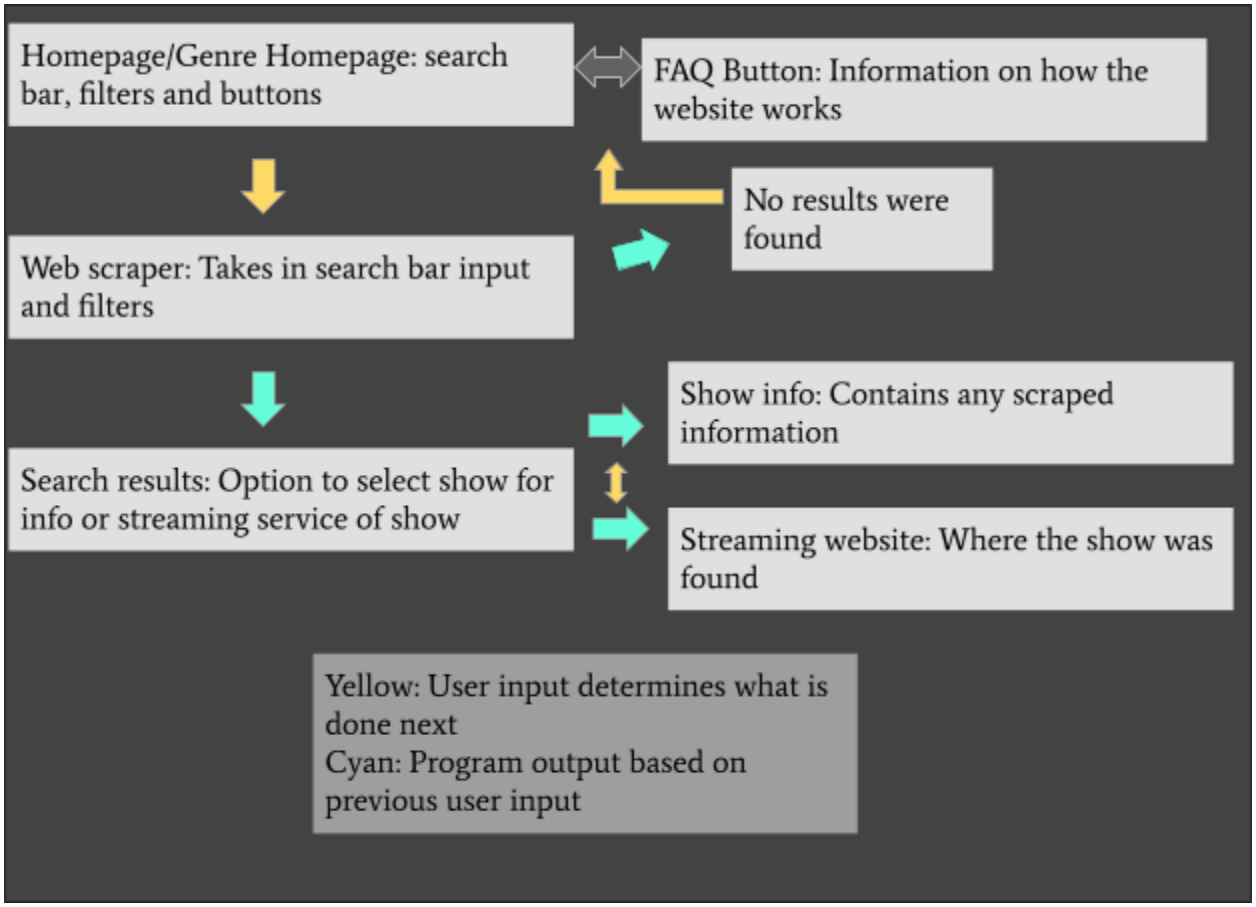
3.1.13 Requirement #13: Results should appear within 20 seconds

Analysis: This is a non-functional requirement
Validity: This is not incongruent with the software description
Feasibility: Tests can be run to determine how to best increase accuracy and speed
Consistency: This doesn't interfere with other tasks the software will perform
Prioritization: This is not mandatory for the software to function, but would increase the user experience significantly. The software should be accurate first, then be fast if one has to take priority over the other

4.1 High-Level Design/Low-Level Design:

This section will cover the initial high-level and low-level designs.

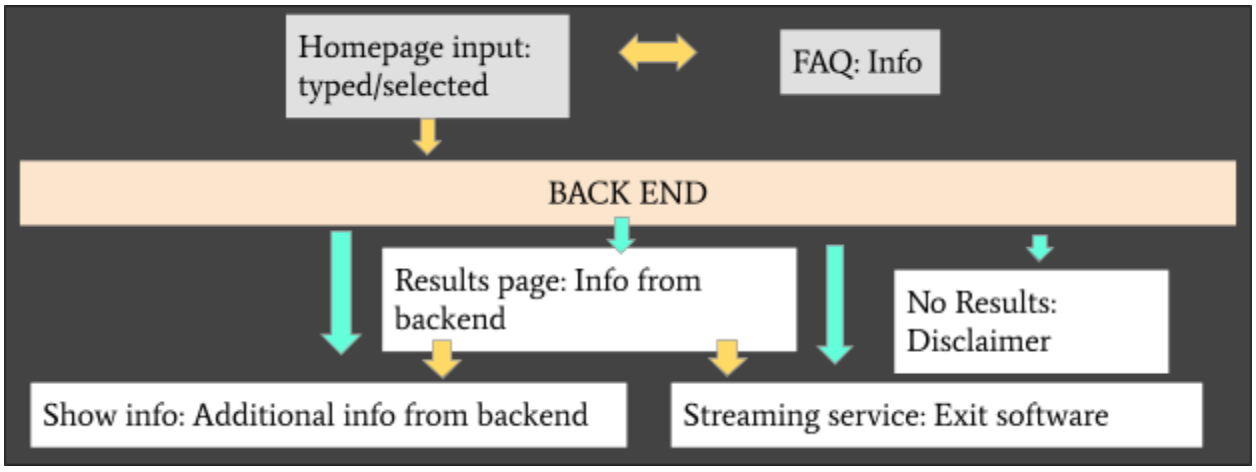
4.2 High-Level Design Diagram:



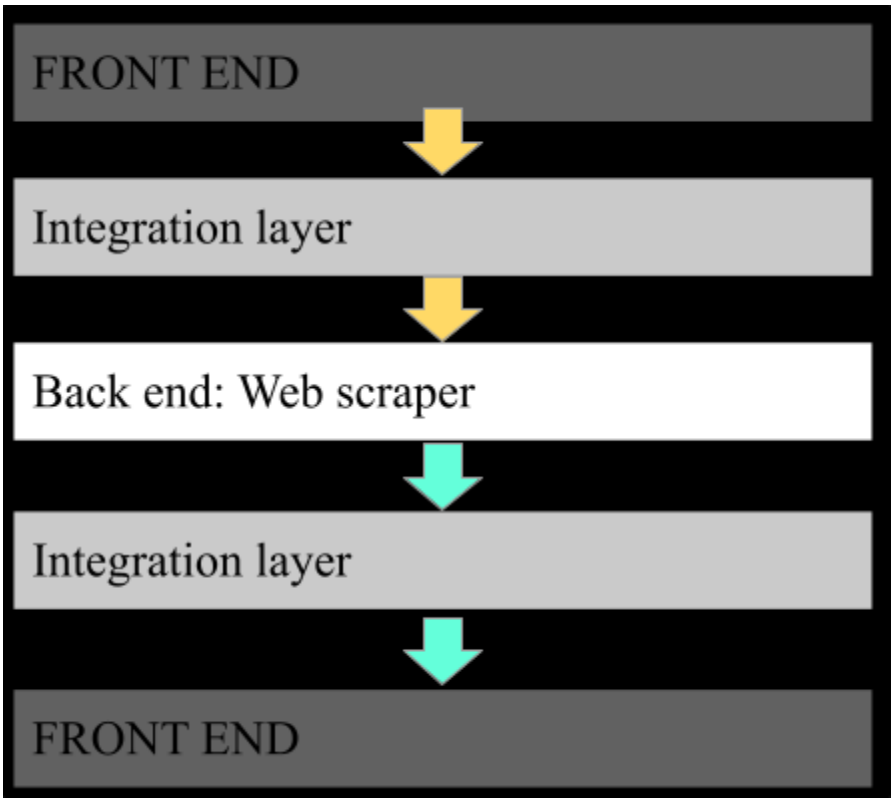
4.3 High-Level Design explanation:

Starting from the homepage, this is a react webpage and allows the user to select a variety of search filters and typed input. These affect what the web scraper will look for. The FAQ page is accessible on this page as well and will give information about the website and the project.
The web scraper will take in the search bar input and filters, and do a web scrape to find the information. If no movie/show is found, the website will display “No results were found”, and prompt the user into searching again.
The search results page will display results if the web scraper finds the searched show/movie, and presents the option to show information about it or an external link. The information contains relevant information like platforms, languages, age rating, while the streaming website is an external link

4.4 Low-Level Design Diagram (Front End):



4.5 Low-Level Design Diagram (Back End):



4.6 Low-Level Design explanation:

For the front end, every page is set up in react.
Homepage input: Sets up a react webpage. Takes in user input, both typed and disabled/enabled filters. Option to enter FAQ page here too.
FAQ: Display info on how to use this website, any disclaimers we have would be displayed too.
Results Page: From backend scrapers, the shows/movies found and what site they are available on would be presented.
Streaming service: Exit the software to enter a streaming site such as Hulu or HBOMax.
No Results found: If the searched show/movie cannot be found, display this page and give a disclaimer about why this showed up.

Backend: Coded in python, imports Beautiful soup, selenium, and requests. Scrapes the website depending on the user input, which includes the name of the show, genre, language filters, and the steaming service filters. If a filer section isn't checked, it looks through all available options (e.g. no language selected so all languages found are displayed).

5.1 Software tools requirements:

The following section will go over the software tools required for the project. The exact specifics for installation will be covered in the Installation section (see 8.1)

5.2 Dependencies:

5.2.1 Python dependencies:

Selenium
Requests
Beautiful Soup

5.2.2 React:

Node.js

5.2.3 Flask:

send_from_directory
CORS
cross_origin
IntEnum
json

5.3 Libraries:

5.3.1 Python

os
time
dataclasses
Typing

5.3.2 React

React-bootstrap
Mdb-react-ui-kit
React-icons
Antd
React-collapsible
react-router-dom

5.3.3 Flask

Flask
pathlib
os
sys
enum
flask_cors

5.4 Languages:

Python
HTML
CSS
JavaScript
Shell

5.5 Operating Systems:

Windows OS
Mac OS

6.1 Hardware Requirements:

This section briefly covers the hardware requirements to use the software. Of note is that full hardware requirements were not able to be calculated, but the software is most likely to be bottlenecked by a poor internet connection (given that it performs web scrapes).

Mouse
Keyboard
Display
Internet Connection

7.1 Software Documentation:

This section will cover the methods/functions created, and how they work.

7.1.1 Python Backend:

The following datatype is used to store show information.

```
@dataclass
class show:
    genres: List[str]
    title: str = "NULL"
    price: str = "NULL"
    url: str = "NULL"
    ageRating: str = "NULL"
```

createHuluTable.py
The program goes through the Hulu sitemap for shows and the Hulu sitemap for movies and obtains all the URLs. Then the program scrapes each URL for the age rating and genres. The price is hardcoded as the subscription price of the service. This would be a flaw if the scrapers were set to autorun to update the show database; however, our database does not auto-update. The result is an array of class show and each element is saved onto a line in a text file.

scrapeNetflix.py
The program uses a third-party site that lists genre URLs for the scraper to access. It accesses each genre URL and scrapes every show in the genre. When a show is scraped, the program checks if it is already in the dataclass array used to store show information and if it is, the genre information is appended to, else a new dataclass element will be added.

paramount shows.py
The program finds the name and URL for every show from the show list and puts that data into an array of the provided dataclass. An array containing all the links to the different genres is then made. Each show found on each genre page is accessed and when a show is found on that genre page, the same show stored in the dataclass array will have its genre list appended. The url for each show is then accessed and scraped to obtain the age rating for each show if it has one.

scrapeParamountMovies.py
This program follows the same steps as paramount shows.py but uses Selenium webdriver to scrape Paramount's movies. Selenium was required because not all movies can be loaded at once.

7.1.2 React Frontend:

Home.js

This file contains the source code for the home page. It displays the navigation bar, search bar, and label of the website title. The navigation bar is imported from Navigation.js inside components/NavBar folder. The search bar is imported from SearchBar.js in the same directory as Home.js.

Navigation.js
This file contains the navigation bar of our website. A third-party library from 'react-bootstrap' was installed to create this navigation bar.

Genre.js
This file contains the source code for the genre page. The page displays navigation bar at the top. Below the navigation bar, there are 6 different main genres that can be clicked to search for those specific genres directly. If users want to search for other genres, there is a search bar below those 6 genre images where users can input for specific genres along with filters after the search bar. Instead of table, divisions were used to organize the elements on the index.css file.

Movies.js
This file contains the source code for information on a movie or tv show searched by a user. Both divisions and table were used to organize the elements on this page. The information contains pictures, title, synopsis, rating on IMDb, casts, and streaming services that stream the movie.

faq.js
This file contains the source code for the questions and answers page of the website. Collapsible lists were used from 'react-collapsible' library to display the questions on the outside along with the answers on the inside.

SearchResult.js
This file displays the search results at the bottom of any page that it's imported into.

SearchResultList.jsx
This file takes in the data retrieved from the server backend and displays it through the SearchResult function defined in SearchResult.js. This file is still in development to finish implementing the filter checkboxes. It utilizes the map functionality to do so.

SearchBar.js
This file defines the UI of the homepage and implements the onChange and onSubmit functions for searching the titles in the backend and applying the filters. The onChange button functionality is still in development, it currently calls the onProviderChange function to filter the streaming services requested.

7.1.3 Flask Integration:

The flask-server folder holds the flask integration. Inside of this folder is the configuration for the flask environment, the secret key (built-in flask functionality for security), and the app folder. The app folder contains server.py which defines the routes for the flask server and fetches the movie information from the python backend text files. CORS browser security dictated that we had to import and use flask_cors as a workaround. Using a DataFields enum we define the data fields in the backend text files, then we use the "load_files" function to populate a dictionary which gets sent to the frontend as requested. Our "get_if" function is for if the list is not the length expected and allows us to correctly populate the returning dictionary. There are two routes defined in server.py, the default path and the searchResult path. As the application is still in development, we have kept the flask server in debug mode.

8.1 Installation:

This section covers the installation steps. This software has not been tested on Linux yet, so full compatibility is unknown.

- Download source code zipped from the GitHub: SarahKercheval/Capstone-Winter22
- Unzip the file
- Install Node.js/npm (if not already installed)
- Install python (if not already installed)
- Ensure the latest versions of npm and python are installed, elsewise errors may appear.
- There may be issues with installing npm if there is not enough disk space available.
- The front-end may compile with errors if the react version is not updated.

8.1.1 Mac OS:

To begin running the server: Open a terminal and go into the main directory and run "pip3 install flask", then "pip3 install flask-cors". From here, run "cd flask-server/app" to go into the main flask server directory. Run the command "export FLASK_APP=server.py" to set your environment, then run "python3 -m flask run" to start the server.

To begin running the frontend: Open another terminal, go into the my-app directory and run "npm install". If another application is running on localHost:3000 it will ask if you want to run it

on a different port. Answer yes. After npm installs, run the command “npm run start”. This will open a tab on your browser on the localhost.

If you need to exit either the server or the react UI, “control-c” will exit out of them.

8.1.2 Windows OS:

To begin running the server: Open a powershell terminal and go into the main directory. Run “pip3 install flask” then “pip3 install flask-cors”. From here, run “cd flask-server/app” to go into the main flask server directory. Run the command “ $env:FLASK_APP=”server.py” ” to set your environment, then run “python -m flask run” to start the server. The server will start on 127.0.0.1:5000, in order to see the search results the URL will need to be 127.0.0.1:5000/search-result/movieTitle, where movieTitle is the searched title you want.

To begin running the frontend: Open a powershell terminal and go to the main directory. From here run “cd my-app” then run “npm install”. This installation will likely take a couple minutes. After npm has installed, run “cd my-app” which is the main react directory. Inside here run “npm run start”. This will start the development page on a web browser.

If you need to exit either the server or the react UI, “control-c” will exit out of them.

9.1 User Documentation:

The following section will lay out some user documentation regarding the software.

9.1.1 How to use this webpage:

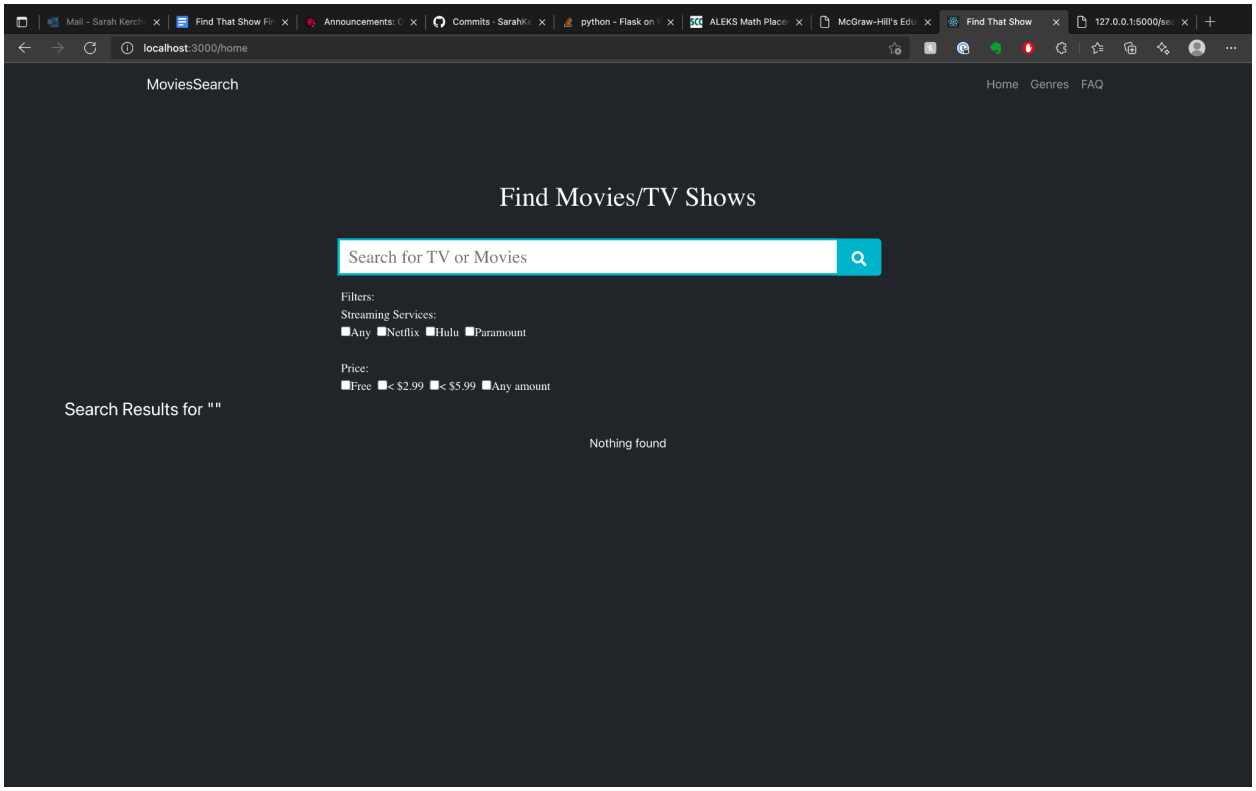
At the home page, type the show you are looking for in the search bar and click search. Be aware that if there’s a great number of search results it will take a few seconds for the results to appear. Sometimes the button doesn’t click! This is a bug that is still being fixed.
Currently there is a CORS policy issue that is being worked out. Though we can see the information being correctly received from the server, we cannot view it in the UI.
If the user would like to view the information being drawn from the server they can go to their localhost:5000/search-result/title where title is the title of the search.

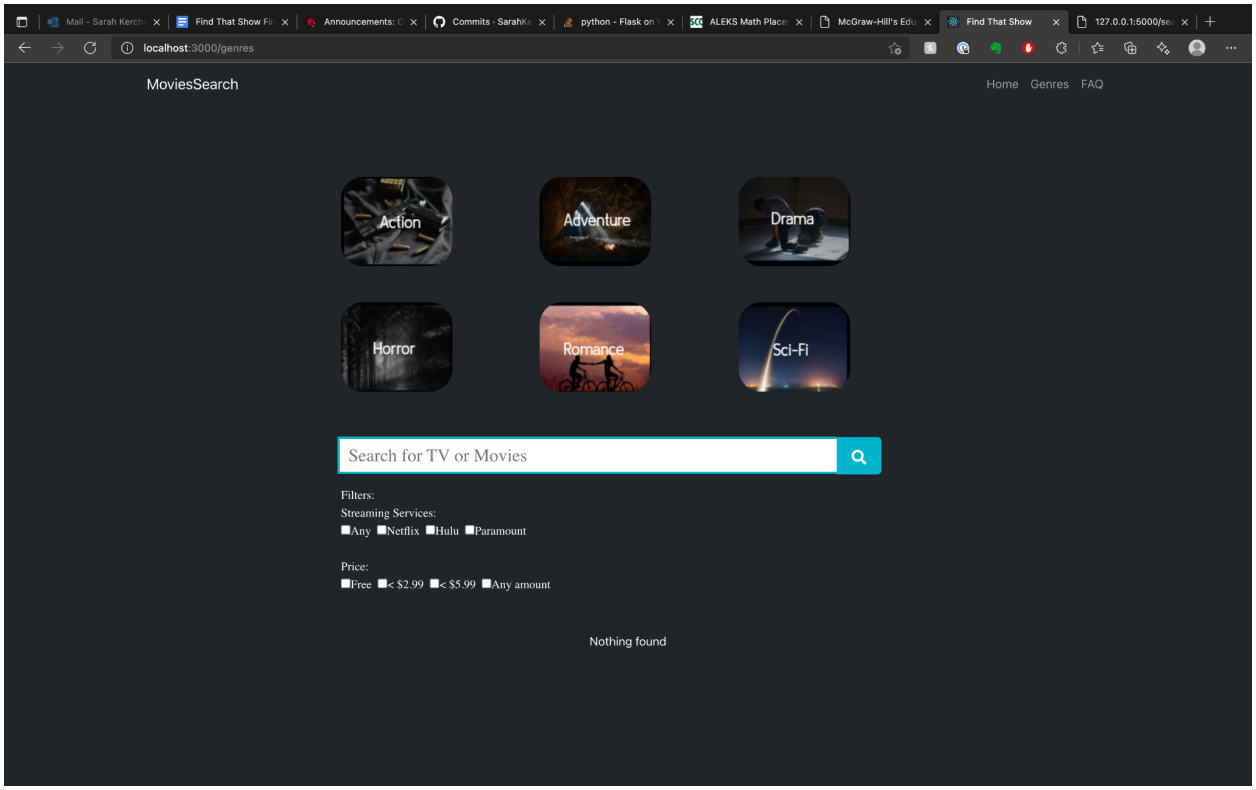
9.1.2 Known Bugs:

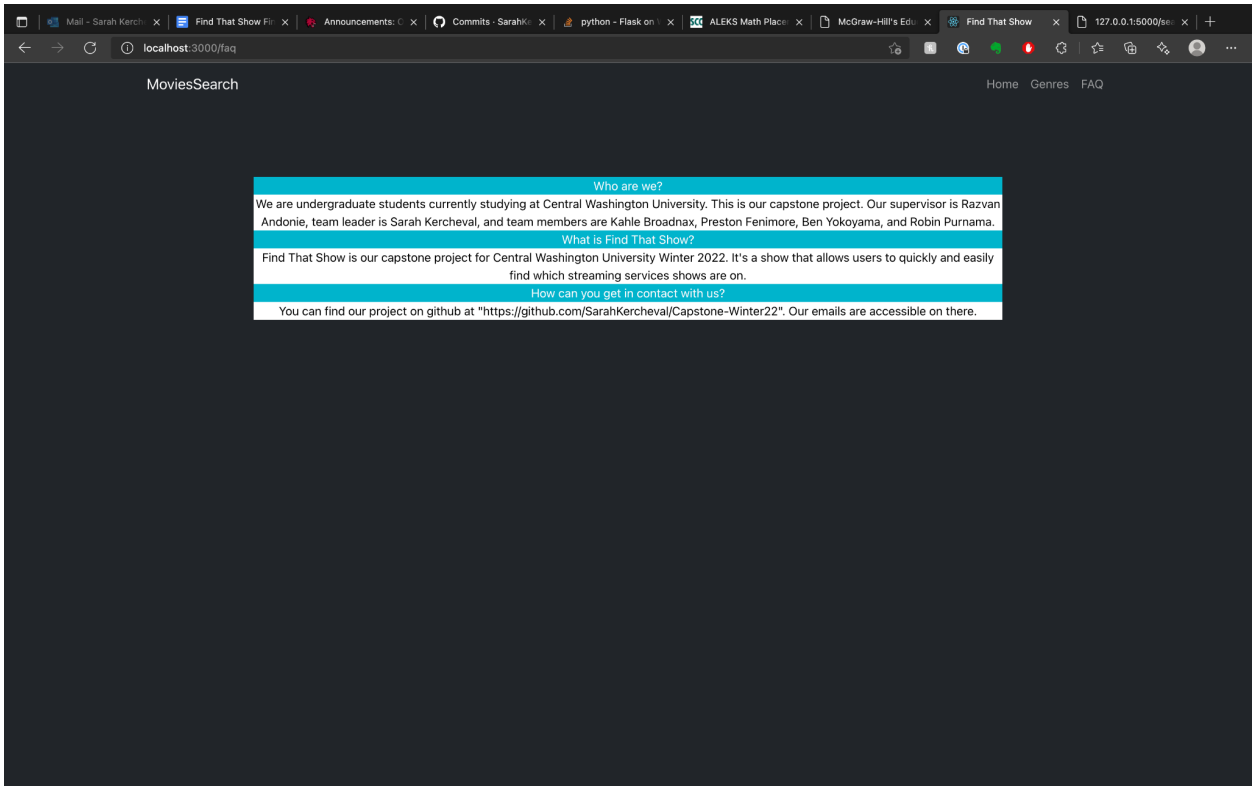
CORS policy is not allowing the frontend fetch to access the server data. This was mitigated for a while with flask-cors but occasionally will throw an error.

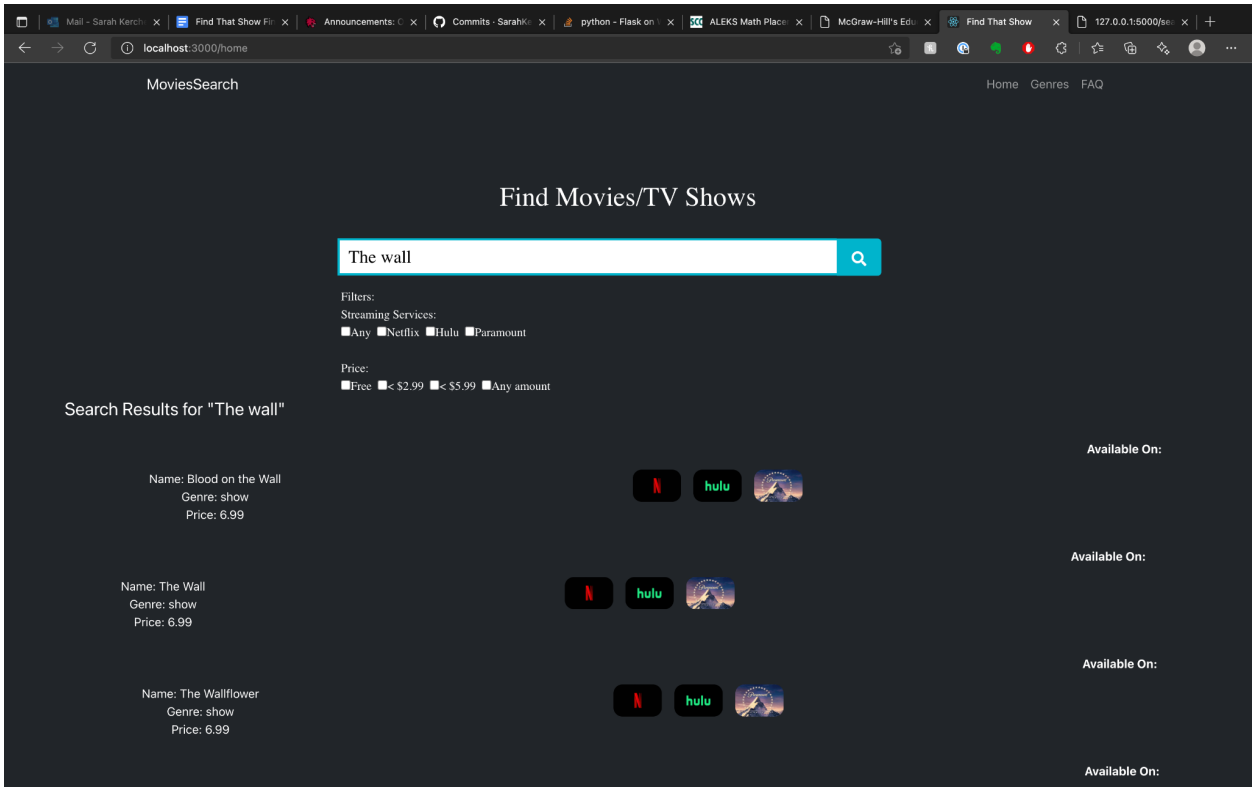
Filter automatically gets set to Paramount as the streaming service, so the filtering isn't fully working.

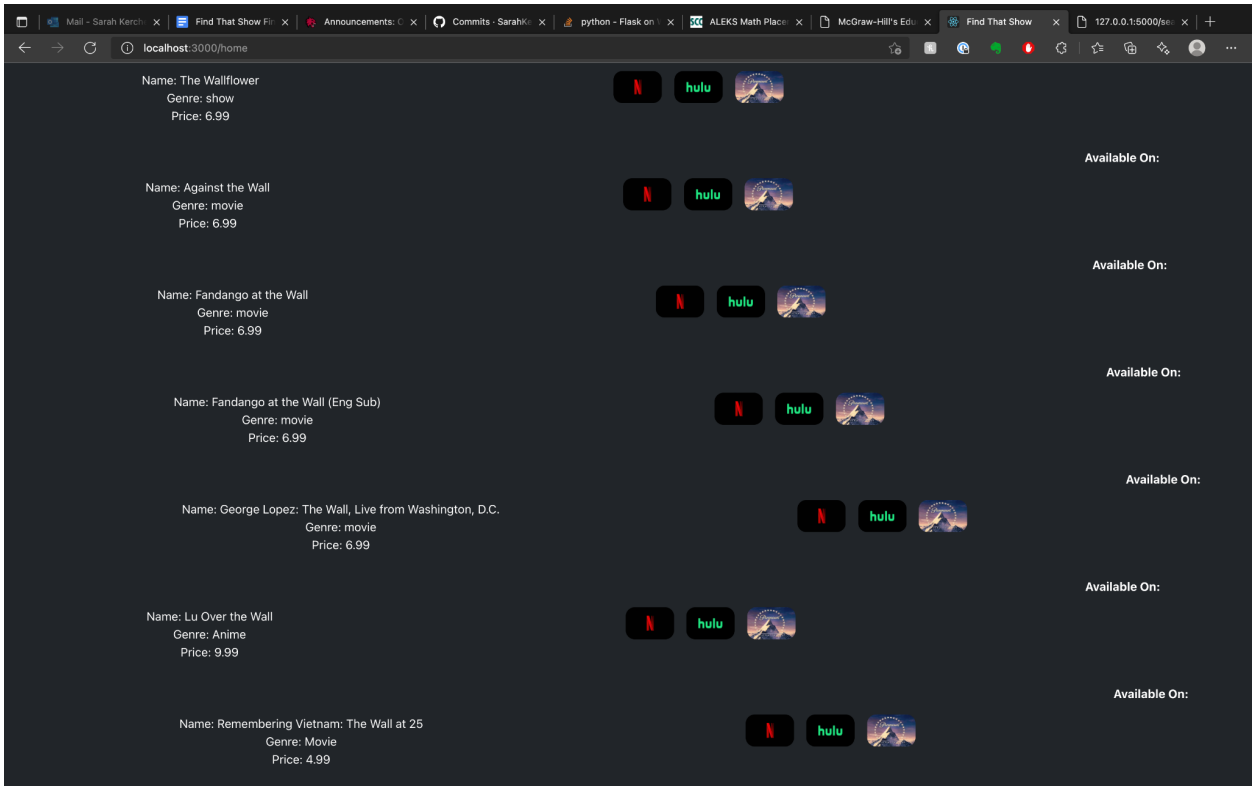
9.1.3 Screenshots:

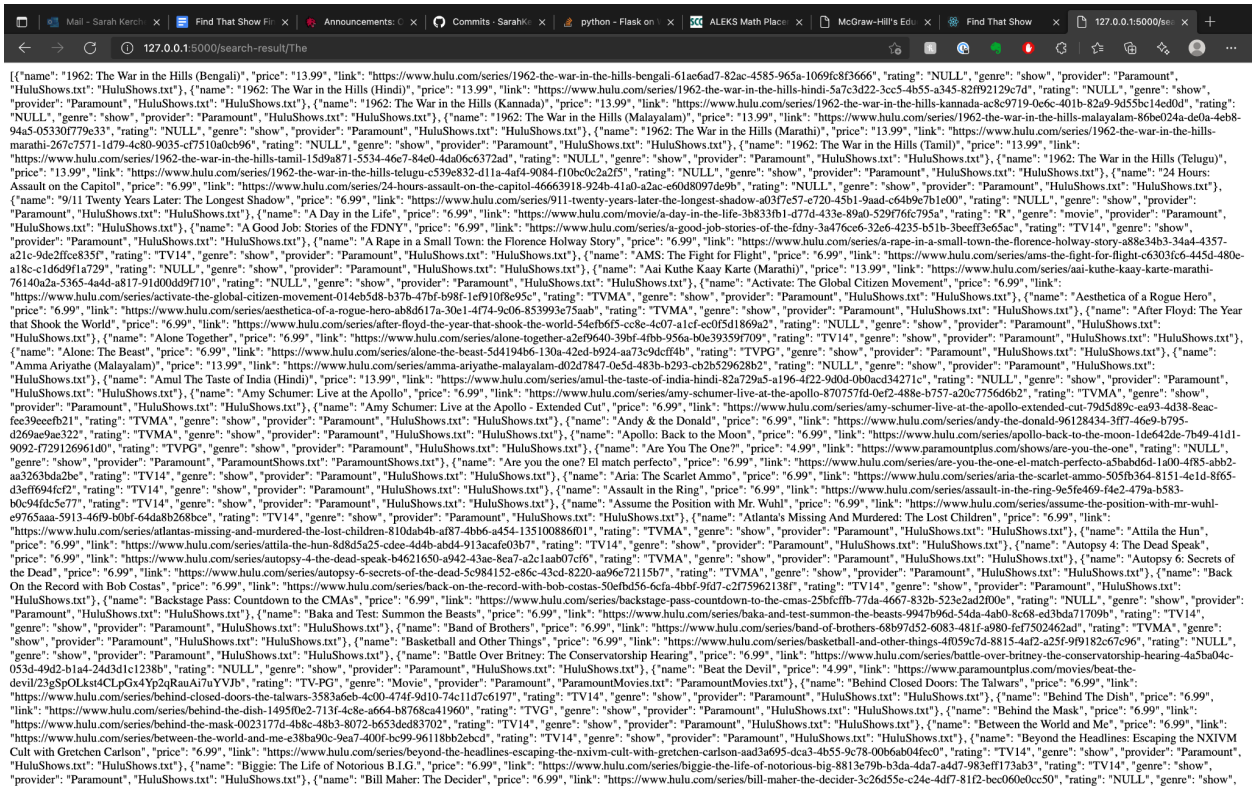



Find That Show
localhost:3000/genres
MoviesSearch
Home
Genres
FAQ
Action
Adventure
Drama
Horror
Romance
Sci-Fi
Search for TV or Movies
Filters:
Streaming Services:
Any
Netflix
Hulu
Paramount
Price:
Free
< $2.99
< $5.99
Any amount
Nothing found


localhost:3000/faq
MoviesSearch
Home Genres FAQ
Who are we?
We are undergraduate students currently studying at Central Washington University. This is our capstone project. Our supervisor is Razvan Andonie, team leader is Sarah Kercheval, and team members are Kahle Broadnax, Preston Fenimore, Ben Yokoyama, and Robin Purnama.
What is Find That Show?
Find That Show is our capstone project for Central Washington University Winter 2022. It's a show that allows users to quickly and easily find which streaming services shows are on.
How can you get in contact with us?
You can find our project on github at "https://github.com/SarahKercheval/Capstone-Winter22". Our emails are accessible on there.


localhost:3000/home
MoviesSearch
Home Genres FAQ
Find Movies/TV Shows
The wall
Filters:
Streaming Services:
Any Netflix Hulu Paramount
Price:
Free < $2.99 < $5.99 Any amount
Search Results for "The wall"
Available On:
Name: Blood on the Wall
Genre: show
Price: 6.99
Available On:
Name: The Wall
Genre: show
Price: 6.99
Available On:
Name: The Wallflower
Genre: show
Price: 6.99
Available On:





[{"name": "1962: The War in the Hills (Bengali)", "price": "13.99", "link": "https://www.hulu.com/series/1962-the-war-in-the-hills-bengali-61ac6ad7-82ac-4585-965a-1069fc8f3666", "rating": "NULL", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "1962: The War in the Hills (Hindi)", "price": "13.99", "link": "https://www.hulu.com/series/1962-the-war-in-the-hills-hindi-5a7c3d22-3cc5-4b55-a345-82ff92129c7d", "rating": "NULL", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "1962: The War in the Hills (Kannada)", "price": "13.99", "link": "https://www.hulu.com/series/1962-the-war-in-the-hills-kannada-ac8c9719-0e6c-401b-82a9-9d55bc14ed0d", "rating": "NULL", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "1962: The War in the Hills (Malayalam)", "price": "13.99", "link": "https://www.hulu.com/series/1962-the-war-in-the-hills-malayalam-86be024a-de0a-4eb8-94a5-05330f779e33", "rating": "NULL", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "1962: The War in the Hills (Marathi)", "price": "13.99", "link": "https://www.hulu.com/series/1962-the-war-in-the-hills-marathi-267c7571-1d79-4c80-9035-cf7510a0cb96", "rating": "NULL", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "1962: The War in the Hills (Tamil)", "price": "13.99", "link": "https://www.hulu.com/series/1962-the-war-in-the-hills-tamil-15d9a871-5534-46e7-84e0-4da06c6372ad", "rating": "NULL", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "1962: The War in the Hills (Telugu)", "price": "13.99", "link": "https://www.hulu.com/series/1962-the-war-in-the-hills-telugu-c539e832-d11a-4af4-9084-f10bc0c2a2f5", "rating": "NULL", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "24 Hours: Assault on the Capitol", "price": "6.99", "link": "https://www.hulu.com/series/24-hours-assault-on-the-capitol-46663918-924b-41a0-a2ac-e60d8097de9b", "rating": "NULL", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "9/11 Twenty Years Later: The Longest Shadow", "price": "6.99", "link": "https://www.hulu.com/series/911-twenty-years-later-the-longest-shadow-a03f7c57-e720-45b1-9aad-c64b9e7b1e00", "rating": "NULL", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "A Day in the Life", "price": "6.99", "link": "https://www.hulu.com/movie/a-day-in-the-life-3b833fb1-d77d-433e-89a0-529f76fc795a", "rating": "R", "genre": "movie", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "A Good Job: Stories of the FDNY", "price": "6.99", "link": "https://www.hulu.com/series/a-good-job-stories-of-the-fdny-3a476ce6-32e6-4235-b51b-3beeff3e65ac", "rating": "TV14", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "A Rape in a Small Town: the Florence Holway Story", "price": "6.99", "link": "https://www.hulu.com/series/a-rape-in-a-small-town-the-florence-holway-story-a88e34b3-34a4-4357-a21c-9de2ffce835f", "rating": "TV14", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "AMS: The Fight for Flight", "price": "6.99", "link": "https://www.hulu.com/series/ams-the-fight-for-flight-c6303fc6-445d-480e-a18c-c1d6d9f1a729", "rating": "NULL", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "Aai Kuthe Kaay Karte (Marathi)", "price": "13.99", "link": "https://www.hulu.com/series/aai-kuthe-kaay-karte-marathi-76140a2a-5365-4a4d-a817-91d00dd9f710", "rating": "NULL", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "Activate: The Global Citizen Movement", "price": "6.99", "link": "https://www.hulu.com/series/activate-the-global-citizen-movement-014eb5d8-b37b-47bf-b98f-1ef910f8e95e", "rating": "TVMA", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "Aesthetica of a Rogue Hero", "price": "6.99", "link": "https://www.hulu.com/series/aesthetica-of-a-rogue-hero-ab8d617a-30e1-4f74-9c06-853993e75aab", "rating": "TVMA", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "After Floyd: The Year that Shook the World", "price": "6.99", "link": "https://www.hulu.com/series/after-floyd-the-year-that-shook-the-world-54efb6f5-cc8e-4c07-a1cf-ec0f5d1869a2", "rating": "NULL", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "Alone Together", "price": "6.99", "link": "https://www.hulu.com/series/alone-together-a2ef9640-39bf-4fbb-956a-b0e39359f709", "rating": "TV14", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "Alone: The Beast", "price": "6.99", "link": "https://www.hulu.com/series/alone-the-beast-5d4194b6-130a-42ed-b924-aa73c9dcff4b", "rating": "TVPG", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "Amma Ariyathe (Malayalam)", "price": "13.99", "link": "https://www.hulu.com/series/amma-ariyathe-malayalam-d02d7847-0e5d-483b-b293-cb2b529628b2", "rating": "NULL", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "Amul The Taste of India (Hindi)", "price": "13.99", "link": "https://www.hulu.com/series/amul-the-taste-of-india-hindi-82a729a5-a196-4f22-9d0d-0b0acd34271c", "rating": "NULL", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "Amy Schumer: Live at the Apollo", "price": "6.99", "link": "https://www.hulu.com/series/amy-schumer-live-at-the-apollo-870757fd-0ef2-488e-b757-a20c7756d6b2", "rating": "TVMA", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "Amy Schumer: Live at the Apollo - Extended Cut", "price": "6.99", "link": "https://www.hulu.com/series/amy-schumer-live-at-the-apollo-extended-cut-79d5d89c-ea93-4d38-8eac-fee39eeefb21", "rating": "TVMA", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "Andy & the Donald", "price": "6.99", "link": "https://www.hulu.com/series/andy-the-donald-96128434-3ff7-46e9-b795-d269ae9ae322", "rating": "TVMA", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "Apollo: Back to the Moon", "price": "6.99", "link": "https://www.hulu.com/series/apollo-back-to-the-moon-1de642de-7b49-41d1-9092-f729126961d0", "rating": "TVPG", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "Are You The One?", "price": "4.99", "link": "https://www.paramountplus.com/shows/are-you-the-one", "rating": "NULL", "genre": "show", "provider": "Paramount", "ParamountShows.txt": "ParamountShows.txt"}, {"name": "Are you the one? El match perfecto", "price": "6.99", "link": "https://www.hulu.com/series/are-you-the-one-el-match-perfecto-a5babd6d-1a00-4f85-abb2-aa3263bda2be", "rating": "TV14", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "Aria: The Scarlet Ammo", "price": "6.99", "link": "https://www.hulu.com/series/aria-the-scarlet-ammo-505fb364-8151-4e1d-8f65-d3eff694fcf2", "rating": "TV14", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "Assault in the Ring", "price": "6.99", "link": "https://www.hulu.com/series/assault-in-the-ring-9e5fe469-f4e2-479a-b583-b0c94fdc5e77", "rating": "TV14", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "Assume the Position with Mr. Wuhl", "price": "6.99", "link": "https://www.hulu.com/series/assume-the-position-with-mr-wuhl-e9765aaa-5913-46f9-b0bf-64da8b268bce", "rating": "TV14", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "Atlanta's Missing And Murdered: The Lost Children", "price": "6.99", "link": "https://www.hulu.com/series/atlantas-missing-and-murdered-the-lost-children-810dab4b-af87-4bb6-a454-135100886f01", "rating": "TVMA", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "Attila the Hun", "price": "6.99", "link": "https://www.hulu.com/series/attila-the-hun-8d8d5a25-cdee-4d4b-abd4-913acafe03b7", "rating": "TV14", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "Autopsy 4: The Dead Speak", "price": "6.99", "link": "https://www.hulu.com/series/autopsy-4-the-dead-speak-b4621650-a942-43ae-8ea7-a2c1aab07cf6", "rating": "TVMA", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "Autopsy 6: Secrets of the Dead", "price": "6.99", "link": "https://www.hulu.com/series/autopsy-6-secrets-of-the-dead-5c984152-e86c-43cd-8220-aa96e72115b7", "rating": "TVMA", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "Back On the Record with Bob Costas", "price": "6.99", "link": "https://www.hulu.com/series/back-on-the-record-with-bob-costas-50efbd56-6cfa-4bbf-9fd7-c2f75962138f", "rating": "TV14", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "Backstage Pass: Countdown to the CMAs", "price": "6.99", "link": "https://www.hulu.com/series/backstage-pass-countdown-to-the-cmas-25bfcffb-77da-4667-832b-523e2ad2f00e", "rating": "NULL", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "Baka and Test: Summon the Beasts", "price": "6.99", "link": "https://www.hulu.com/series/baka-and-test-summon-the-beasts-9947b96d-54da-4ab0-8c68-ed3bda71709b", "rating": "TV14", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "Band of Brothers", "price": "6.99", "link": "https://www.hulu.com/series/band-of-brothers-68b97d52-6083-481f-a980-fef7502462ad", "rating": "TVMA", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "Basketball and Other Things", "price": "6.99", "link": "https://www.hulu.com/series/basketball-and-other-things-4f059c7d-8815-4af2-a25f-9f9182c67c96", "rating": "NULL", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "Battle Over Britney: The Conservatorship Hearing", "price": "6.99", "link": "https://www.hulu.com/series/battle-over-britney-the-conservatorship-hearing-4a5ba04c-053d-49d2-b1a4-24d3d1c1238b", "rating": "NULL", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "Beat the Devil", "price": "4.99", "link": "https://www.paramountplus.com/movies/beat-the-devil/23gSpOLkst4CLpGx4Yp2qRauAi7uYVJb", "rating": "TV-PG", "genre": "Movie", "provider": "Paramount", "ParamountMovies.txt": "ParamountMovies.txt"}, {"name": "Behind Closed Doors: The Talwars", "price": "6.99", "link": "https://www.hulu.com/series/behind-closed-doors-the-talwars-3583a6eb-4c00-474f-9d10-74c11d7c6197", "rating": "TV14", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "Behind The Dish", "price": "6.99", "link": "https://www.hulu.com/series/behind-the-dish-1495f0e2-713f-4c8e-a664-b8768ca41960", "rating": "TVG", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "Behind the Mask", "price": "6.99", "link": "https://www.hulu.com/series/behind-the-mask-0023177d-4b8c-48b3-8072-b653ded83702", "rating": "TV14", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "Between the World and Me", "price": "6.99", "link": "https://www.hulu.com/series/between-the-world-and-me-e38ba90c-9ea7-400f-bc99-96118bb2ebcd", "rating": "TV14", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "Beyond the Headlines: Escaping the NXIVM Cult with Gretchen Carlson", "price": "6.99", "link": "https://www.hulu.com/series/beyond-the-headlines-escaping-the-nxivm-cult-with-gretchen-carlson-aad3a695-dca3-4b55-9c78-00b6ab04fec0", "rating": "TV14", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "Biggie: The Life of Notorious B.I.G.", "price": "6.99", "link": "https://www.hulu.com/series/biggie-the-life-of-notorious-big-8813e79b-b3da-4da7-a4d7-983eff173ab3", "rating": "TV14", "genre": "show", "provider": "Paramount", "HuluShows.txt": "HuluShows.txt"}, {"name": "Bill Maher: The Decider", "price": "6.99", "link": "https://www.hulu.com/series/bill-maher-the-decider-3c26d55e-c24e-4df7-81f2-bec060e0cc50", "rating": "NULL", "genre": "show",

The above image is of the servers' search results hosted on localhost:5000

10.1 Conclusion:

10.1.1 Final Document Summary:

This document covered the proposed idea of a movie/show search using a web scraper backend. Several requirements were created to narrow the focus of this software, and following that, the high-level and low-level designs were created, which saw the project being split into three main components, the front-end, back-end, and integration layer. Our main software tools were React, Python, and Flask, for the front-end, back-end, and integration layer respectfully.
The front-end (React) allowed for user input, both typed and filter selected options. These inputs would be sent to the integration layer (Flask), which would parse for the relevant information from the back-end. The back-end has multiple scrapers and calls them all to identify all the possible locations of a show. It then compiles them into .txt files, which then gets sent to the front-end through Flask. The front-end will display the searched results.
For the installation, you need the source code from GitHub and have Node.js and python installed if they aren't already (look to the installation section for full details). The final section contains user documentation, including screenshots and how to use the software.

10.1.2 What we've learned:

We learned how projects can evolve in complexity as you attempt to solve software problems. This requires early identification of bottlenecks to keep the project flowing. We also learned the software tools we employed more deeply, given this was our first major project with those tools.

10.1.3 What we would do differently:

Decide integration and languages fully before starting the project and start the integration between front and back end earlier in the project. We would re-evaluate tasks and consider how long they will take more consistently.

10.2 Who did what?

We worked on the final document together in group calls, along with the midterm and final presentation. Specific slides for the presentation have the names of who presented and who created the slides.

Preston Fenimore:

Team member, python scrapers

Kahle Broadnax:

Team member, react frontend, user testing

Ben Yokoyama:

Team member, python scrapers

Robin Purnama:

Team member, react frontend

Sarah Kercheval:

Team leader, Flask integration, react frontend, Repository management

Razvan Andonie:

Team supervisor