NexdMovie

Final Presentation

Parker Billinger

Inspiration

-I love watching movies

-In age of streaming services there are TOO MANY options

-Spend more time picking a movie then actually watching it

-The site Letterboxd is great to track movies, but it does not give recommendations =(

My Project

-Integrate Letterboxd into a website that uses personal ratings to give a “Nexd Movie” to watch.

-Allow user to insert Letterboxd username, and if it is valid will use their ratings to find which movies are similar to the ones they have rated highly.

Letterboxd

NexdMovie

Background / Related Work

CIS 400: Full stack software development experience

CIS 501: General software structure / design

Internship work: Did some work with React

Side projects: Have worked with python to scrape data for a personal baseball stats project

NexdMovie

Requirements / MVP

-Gather information from letterboxd.com

-Use algorithm to suggest movies based on data

-Display movie suggestions in some type of readable context

NexdMovie

Version 1.0 Features

-Have usable website that is simple to use with clean UI

-Display movie posters and additional movie information

-Have links that help coordinate use between Letterboxd and NexdMovie

NexdMovie

Tech Stack

-Using React JS for the front end

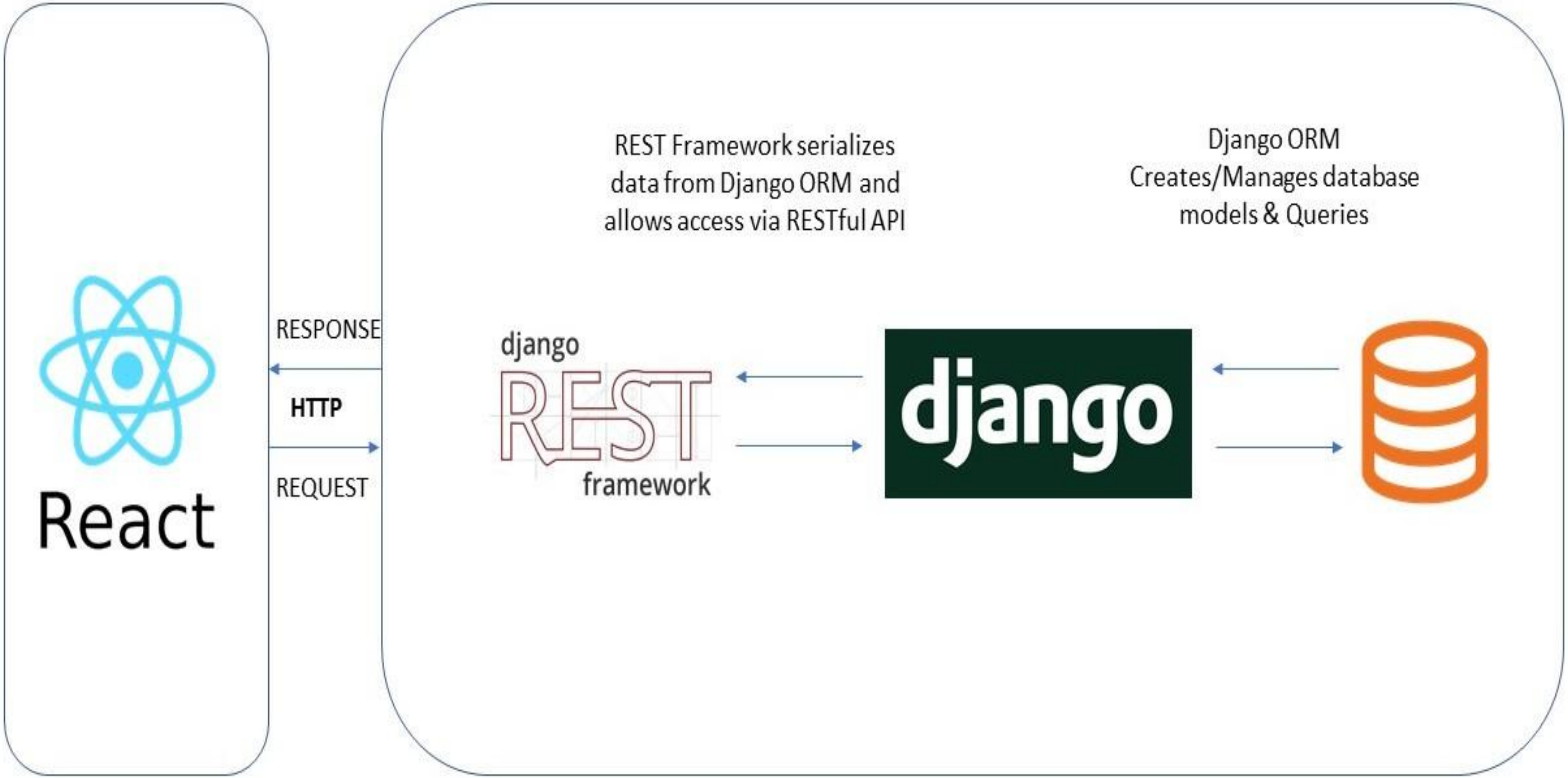
-Using Django for backend

-SQLite for any necessary database operations

-Using TMDB API to gather movie information and recommendations

Overall Structure

NexdMovie

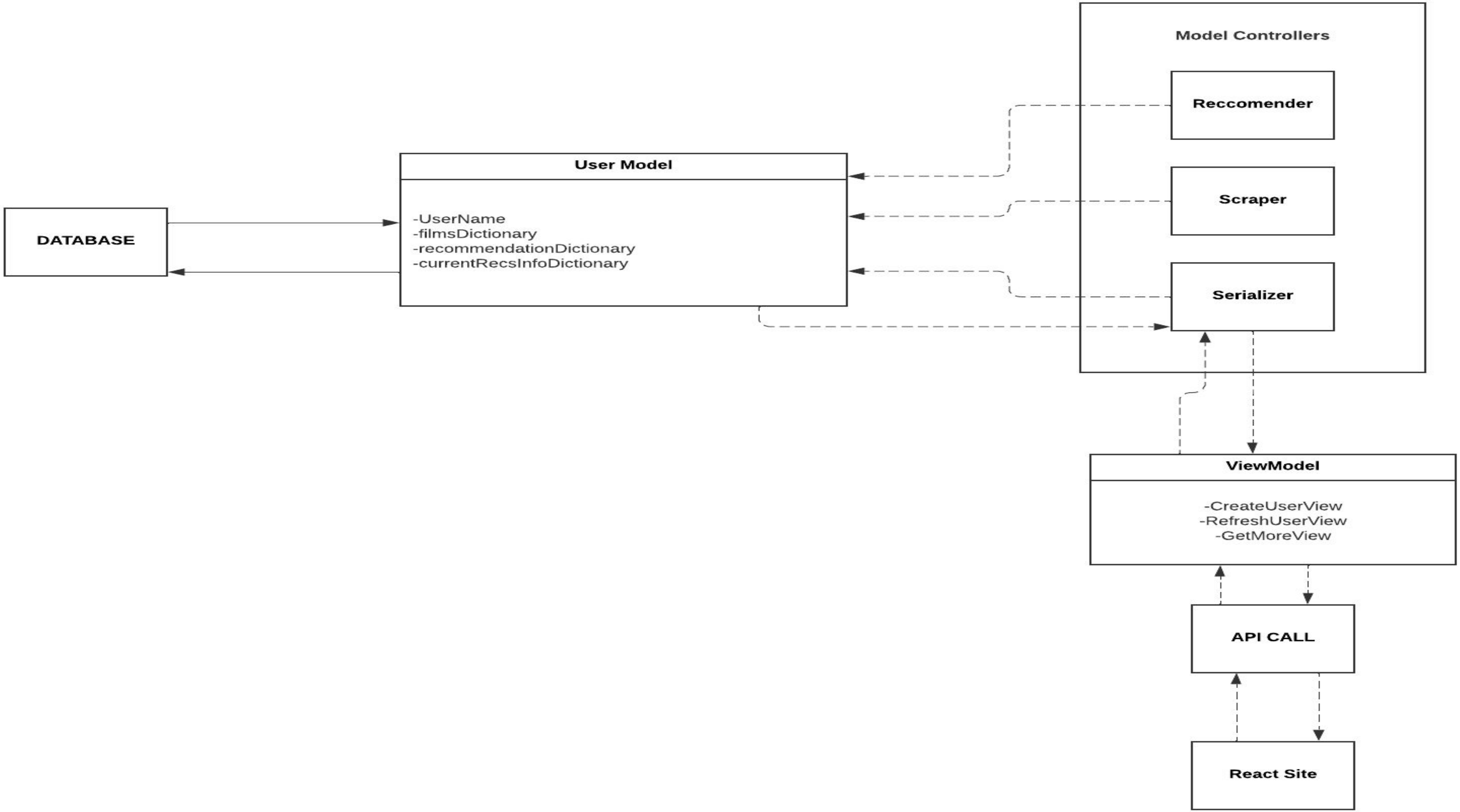


REST Framework serializes data from Django ORM and allows access via RESTful API

Django ORM Creates/Manages database models & Queries

General UML

NexdMovie


Model Controllers
Reccomender
Scraper
Serializer
User Model
-UserName
-filmsDictionary
-recommendationDictionary
-currentRecsInfoDictionary
DATABASE
ViewModel
-CreateUserView
-RefreshUserView
-GetMoreView
API CALL
React Site

Django REST API calls

- **create-user**: post call that gets the recommendations for the letterboxd user with the given username

- **refresh-user**: updates user with given username to show updated recommendations if letterboxd account is updated, or resets the recommendations to original most relevant recommendations

- **get-more**: Gets the movie posters and information for the next 5 most relevant recommendations and updates the users current recommendations to display the change.

NexdMovie

TMDB API Calls

/search

-This call gathers movie information based on search by title

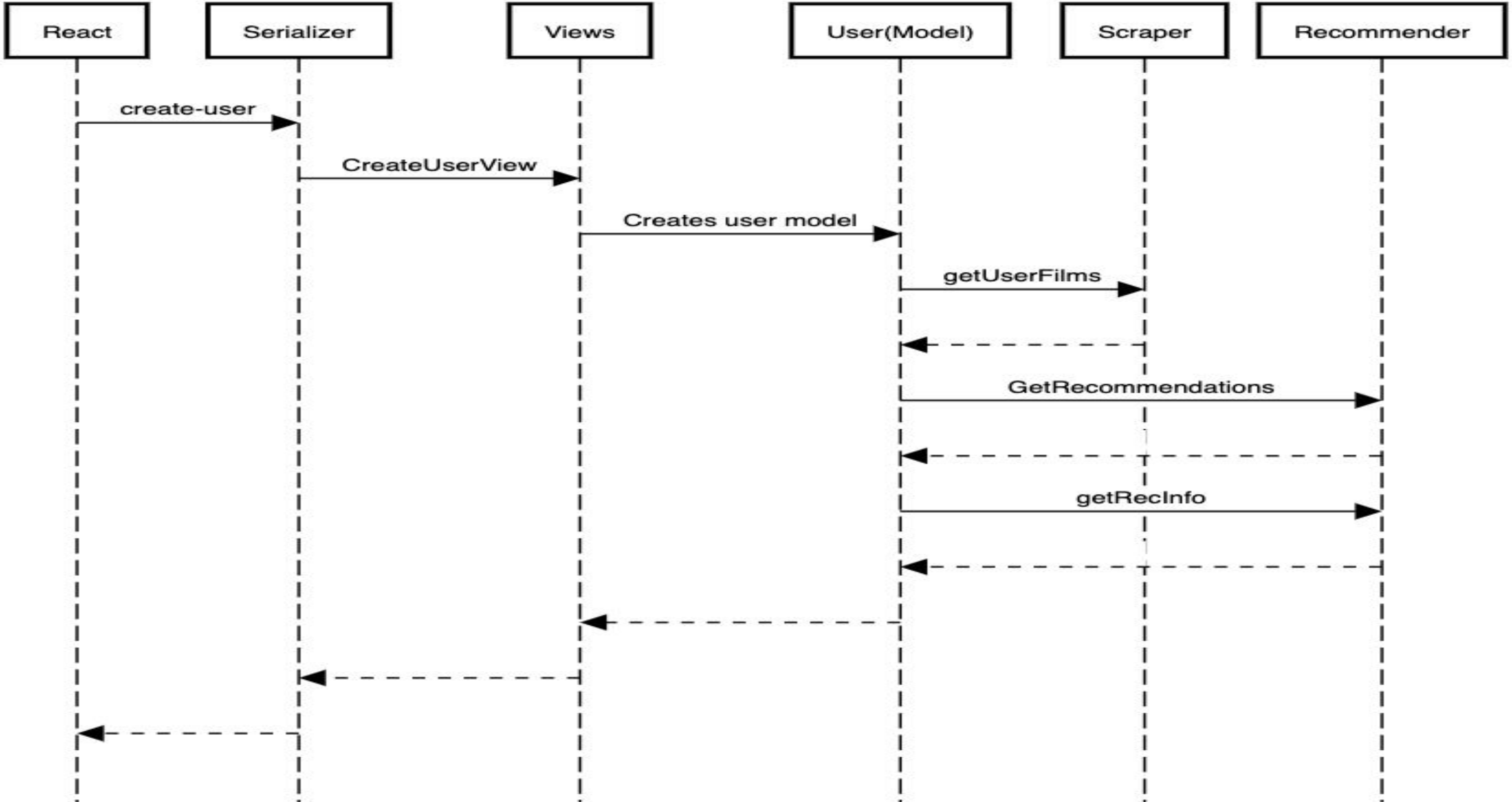
/recommendations

-This call gives several recommendations for one specific movie

CreateUser Sequence Diagram

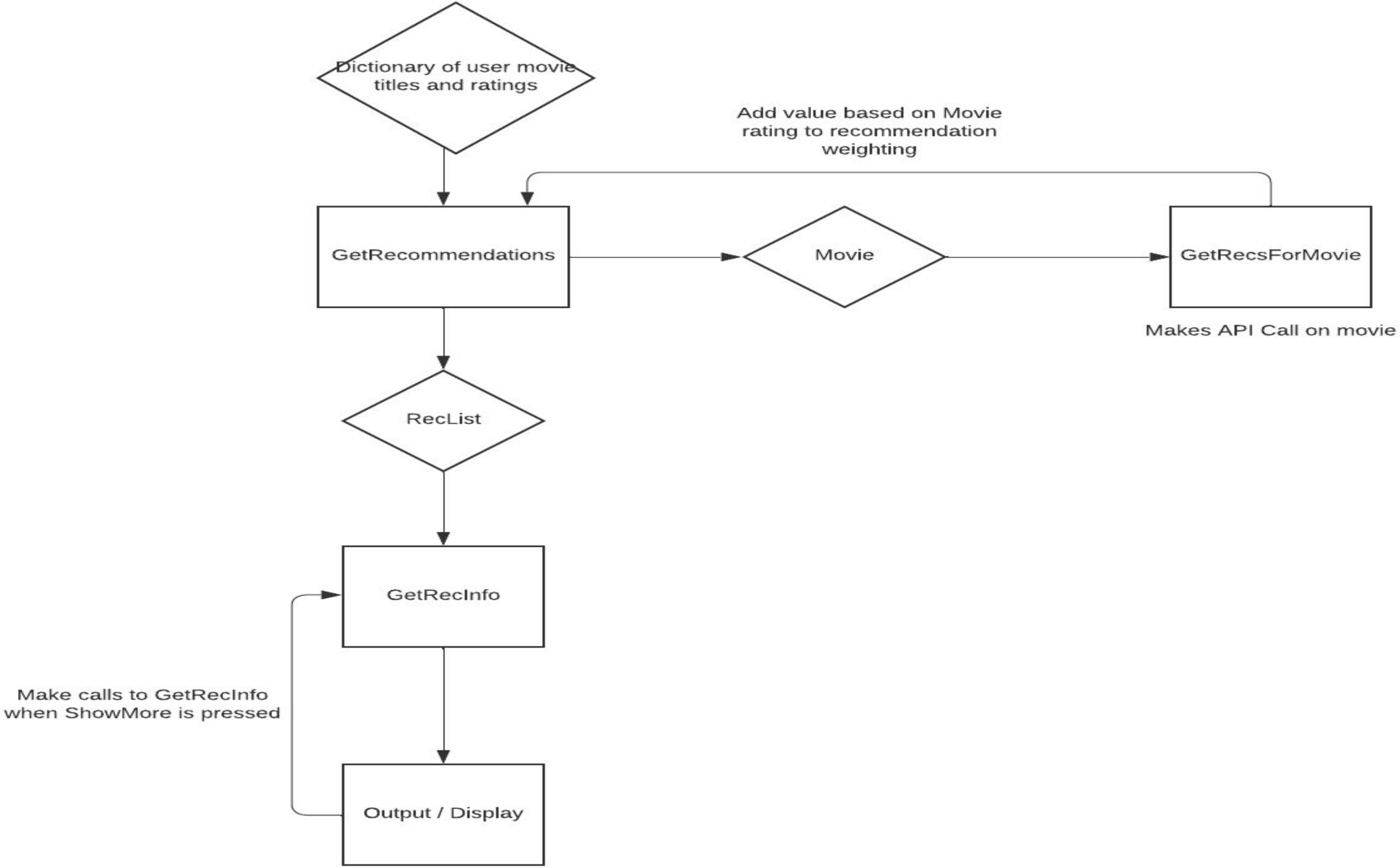
NexdMovie

Create User Sequence Diagram



Recommendation System

NexdMovie


Dictionary of user movie titles and ratings
Add value based on Movie rating to recommendation weighting
GetRecommendations
Movie
GetRecsForMovie
Makes API Call on movie
RecList
GetRecInfo
Make calls to GetRecInfo when ShowMore is pressed
Output / Display

Testing

-Have list of most popular letterboxd accounts I manually tested with

-Also made several dummy letterboxd accounts to manually test a different variety of edge cases

-Eventually want to make site public and send link to friends to test if they use letterboxd

Letterboxd

Demo

NexdMovie

Future Work / Version 2.0 Goals

-Implement more advanced machine learning recommendation engine

-Deploy working version of site

-Have a more direct connection to Letterboxd website

NexdMovie

Conclusion

-Made a satisfactory working product

-Implemented at a base level all the goals I set out for in my v1.0.0

-While not implementing the machine learning based recommendation system, was able to find an intuitive and streamlined way of finding recommendations instead

NexdMovie

Quick Overview

-Movie recommendation system using Letterboxd.com

-Using React JS for the front end

-Using Django for backend

-Using TMDB API to gather movie information

Questions?

NexdMovie