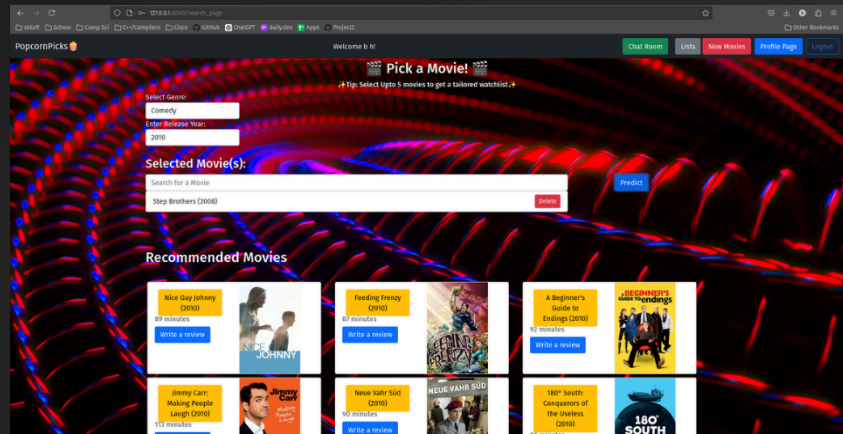


Popcorn Picks!



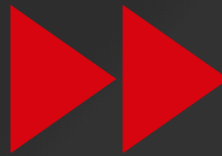
About

PopcornPicks is more than just a movie recommender system; it's a gateway to a world of cinematic adventures. With an ever-expanding library of films, a powerful recommendation algorithm, an all-new chat room, a new state-of-the-art account system, a newly added robust review system, and more, PopcornPicks is here to transform the way you discover, enjoy, and connect with movies.



Whats new?

- A flashy new grid-style UI layout
- Movies can now be individually viewed along with their reviews.
- Users can recommend movies by release year and genre now
- Lists feature: Users can create a list of movies visible to other users, a more friendly search bar
- The site is permanently deployed on Render at popcornpicks.onrender.com



Looking forward

- Review rating system to allow users to thumbs-up or thumbs-down movie reviews.
- The recommendation system is in dire need of a caching/performance improvement.
- A follower system allowing users to follow each other would add a great social-media element to the application.



Tests!

- Improved test on search bar: test misspell, test broader word matching options, test multi-word matching.
- 97 % Code coverage
- 41 total tests



Git Repo



Live Demo

CSC510 Fall 2024 Group 15:
Aditya Karthikeyan | Braden Helmer | Quoc Hung Le

Point	Self Assessment (Sum of Points)			Evidence
	Aditya (101)	Braden (101)	Quoc (101)	
Workload is spread over the whole team (one team member is often Xtimes more productive than the others...	3	3	2	Github Commits
but nevertheless, here is a track record that everyone is contributing a lot)	3	3	3	GH
Number of commits	3	3	3	GH
Number of commits: by different people	3	3	3	GH
Issues reports: there are many	3	3	3	https://github.com/NCSU-CS-C510-Group-BHAKQH/CSC-510-Project2-PopcornPicks/issues?q=is%3Aissue+is%3Aclosed
Issues are being closed	3	3	3	GH
Docs: doco generated, format not ugly	3	3	3	GH
Docs: what: point descriptions of each class/function (in isolation)	3	3	3	Code
Docs: how: for common use cases X,Y,Z mini-tutorials showing worked examples on how to do X,Y,Z	3	3	3	Gifs in README.md
Docs: why: docs tell a story, motivate the whole thing, deliver a punchline that makes you want to rush out and use the thing	3	3	3	README.md
Docs: short video, animated, hosted on your repo. That convinces people why they want to work on your code.	3	3	3	README.md

Use of version control tools	3	3	3	Github history
Test cases exist	3	3	3	https://github.com/NCSU-CS-C510-Group-BHAKQH/CSC-510-Project2-PopcornPicks/tree/master/app/test
Test cases are routinely executed	3	3	3	https://github.com/NCSU-CS-C510-Group-BHAKQH/CSC-510-Project2-PopcornPicks/blob/master/.github/workflows/unittest.yml
Issues are discussed before they are closed	3	3	3	Most of this discussion was done in person. However, initial issue statements in closed issues can be seen here: https://github.com/NCSU-CS-C510-Group-BHAKQH/CSC-510-Project2-PopcornPicks/issues?q=is%3Aissue+is%3Aclosed
Chat channel: exists	3	3	3	Discord
Test cases: a large proportion of the issues related to handling failing cases.	2	2	3	
Evidence that the whole team is using the same tools: everyone can get to all tools and files	3	3	3	Will be shown in TA demo
Evidence that the whole team is using the same tools (e.g. config files in the repo, updated by lots of different people)	3	3	3	Will be shown in TA demo
Evidence that the whole team is using the same tools (e.g. tutor can ask anyone to share screen, they demonstrate the system running on their computer)	3	3	3	Will be shown in TA demo
Evidence that the members of the team are working across multiple places in the code	3	3	3	Will be shown in TA demo

base				
Short release cycles	3	3	3	Can be seen on github.
The file .gitignore lists what files should not be saved to the repo. See [examples](https://github.com/github/gitignore)	3	3	3	GH
The file INSTALL.md lists how to install the code	3	3	3	GH
The file LICENSE.md lists rules of usage for this repo	3	3	3	GH
The file CODE-OF-CONDUCT.md lists rules of behavior for this repo; e.g. see example	3	3	3	GH
The file CONTRIBUTING.md lists coding standards and lots of tips on how to extend the system without screwing things up; e.g. see example	3	3	3	GH
The file README.md contains all the following:				GH
Video	3	3	3	
DOI badge: exists.	3	3	3	GH
Badges showing your style checkers	3	3	3	GH
Badges showing your code formatters.	3	3	3	GH
Badges showing your syntax checkers.	3	3	3	GH
Badges showing your code coverage tools	3	3	3	GH
Badges showing any other Other automated analysis tools	3	3	3	GH

Sustainability Evaluation:

<https://docs.google.com/open?id=1GZEbcS8VbzaC1Y-CYhEIMD2MG-UVjOrCmTKPfkKh-vAI>

Link to the Github Repository:

<https://github.com/NCSU-CSC510-Group-BHAKQH/CSC-510-Project2-PopcornPicks>

Notes	Self-assessment	Evidence
Q1 - What your software does		
“1.1) Does your website and documentation provide a clear, high-level overview of your software? ”	YES	
1.2) Does your website and documentation clearly describe the type of user who should use your software?	YES	
1.3) Do you publish case studies to show how your software has been used by yourself and others?	NO	
Q2 - Your project's and software's identity		
2.1) Is the name of your project/software unique?	YES	
2.2) Is your project/software name free from trademark violations?	YES	
Q3 - Availability of your software		
3.1) Is your software available as a package that can be deployed without building it?	YES	
3.2) Is your software available for free?	YES	

Notes	Self-assessment	Evidence
“3.3) Is your source code publicly available to download either as a downloadable bundle or via access to a source code repository? ”	YES	
“3.4) Is your software hosted in an established third-party repository ? ”	YES	
Q4 - Your software's documentation		
4.1) Is your documentation clearly available on your website or within your software?	YES	
“ 4.2) Does your documentation include a “quick start” guide that provides a short overview of how to use your software with some basic examples of use? ”	YES	
“ 4.3) If you provide more extensive documentation does this provide clear step-by-step instructions on how to deploy and use your software? ”	YES	
“4.4) Do you provide a comprehensive guide to all your software's commands functions and options? ”	YES	
4.5) Do you provide troubleshooting information that describes the symptoms and step-by-step solutions for problems and error messages?	NO	
“4.6) If your software can be used as a library package or service by	YES	

Notes	Self-assessment	Evidence
other software do you provide comprehensive API documentation? ”		
4.7) Do you store your documentation under revision control with your source code?	YES	
“4.8) Do you publish your release history e.g. release data version numbers key features of each release etc. on your web site or in your documentation? ”	NO	
Q5 - How you support your software		
5.1) Does your software describe how a user can get help with using your software?	NO	
“5.2) Does your website and documentation describe what support if any you provide to users and developers? ”	NO	
5.3) Does your project have an e-mail address or forum that is solely for supporting users?	NO	
5.4) Are e-mails to your support e-mail address received by more than one person?	NO	
5.5) Does your project have a ticketing system to manage bug reports and feature requests?	YES	
“5.6) Is your project’s ticketing system publicly visible to your users so they can view bug reports and feature requests? ”	YES	

Notes	Self-assessment	Evidence
Q6 - Your software's maintainability		
6.1) Is your software's architecture and design modular?	YES	
6.2) Does your software use an accepted coding standard or convention?	YES	
Q7 - Open standards and your software		
"7.1) Does your software allow data to be imported and exported using open data formats? "	YES	
"7.2) Does your software allow communications using open communications protocols? "	YES	
Q8 - Your software's portability		
"8.1) Is your software cross-platform compatible? "	YES	
Q9 - Your software and accessibility		
9.1) Does your software adhere to appropriate accessibility conventions or standards?	YES	
9.2) Does your documentation adhere to appropriate accessibility conventions or standards?	YES	
Q10 - How you manage your source code		
10.1) Is your source code stored in a repository under revision control?	YES	

Notes	Self-assessment	Evidence
10.2) Is each source code release a snapshot of the repository?	YES	
10.3) Are releases tagged in the repository?	YES	
“10.4) Is there a branch of the repository that is always stable? (i.e. tests always pass code always builds successfully) ”	YES	
10.5) Do you back-up your repository?	YES	
Q11 - Building and installing your software		
11.1) Do you provide publicly-available instructions for building your software from the source code?	YES	
“11.2) Can you build or package your software using an automated tool? ”	NO	
11.3) Do you provide publicly available instructions for deploying your software?	YES	
11.4) Does your documentation list all third-party dependencies?	NO	
11.5) Does your documentation list the version number for all third-party dependencies?	NO	
“11.6) Does your software list the web address and licences for all third-party dependencies and say whether the dependencies are mandatory or not ”	NO	
“11.7) Can you download dependencies using a	NO	

Notes	Self-assessment	Evidence
dependency management tool or package manager? ”		
11.8) Do you have tests that can be run after your software has been built or deployed to show whether the build or deployment has been successful?	YES	
Q12 - How you test your software		
12.1) Do you have an automated test suite for your software?	YES	
12.2) Do you have a framework to periodically (e.g. nightly) run your tests on the latest version of the source code?	YES	
“12.3) Do you use continuous integration automatically running tests whenever changes are made to your source code? ”	YES	
12.4) Are your test results publicly visible?	YES	
12.5) Are all manually run tests documented?	YES	
Q13 - How you engage with your community		
“13.1) Does your project have resources that are regularly updated with information about your software? ”	NO	
13.2) Does your website state how many projects and users are associated with your project?	YES	

Notes	Self-assessment	Evidence
13.3) Do you provide success stories on your website?	NO	
13.4) Do you list your important partners and collaborators on your website?	NO	
13.5) Do you list your project's publications on your website or link to a resource where these are available?	NO	
13.6) Do you list third-party publications that refer to your software on your website or link to a resource where these are available?	NO	
13.7) Can users subscribe to notifications to changes to your source code repository?	NO	
"13.8) If your software is developed as an open source project do you have a governance model? "	YES	
Q14 - How you manage contributions		
"14.1) Do you accept contributions (e.g. bug fixes, enhancements, documentation, updates, tutorials) from people who are not part of your project? "	YES	
14.2) Do you have a contributions policy?	YES	
14.3) Is your contributions' policy publicly available?	YES	

Notes	Self-assessment	Evidence
14.4) Do contributors keep the copyright/IP of their contributions?	YES	
Q15 - Your software's copyright and licensing		
15.1) Does your website and documentation clearly state the copyright owners of your software and documentation?	YES	
15.2) Does each of your source code files include a copyright statement?	YES	
15.3) Does your website and documentation clearly state the licence of your software?	YES	
15.4) Is your software released under an open source licence?	YES	
15.5) Is your software released under an OSI-approved open-source licence?	YES	
15.6) Does each of your source code files include a licence header?	YES	
15.7) Do you have a recommended citation for your software?	NO	
Q16 - Your plans for the future		
"16.1) Does your website or documentation include a project roadmap ? "	NO	
"16.2) Does your website or documentation describe how your project is funded and the period over which funding is guaranteed? "	NO	

Notes	Self-assessment	Evidence
“16.3) Do you make timely announcements of the deprecation of components APIs etc.? ”	NO	