

Project 2

Group7:

Yinan Wu, Xuntao Lyu, Hua Yang

Link Repo:

<https://github.com/The-Powerpuff-Girls3/CSC-510-Movie-Mood/tree/master>

Video Link: video playlist:

https://studio.youtube.com/playlist/PLiz2sL3f73DgIWn2-vE_S02cNOEqLKk7Y/videos

Video1: Why choose our project for project3:

<https://www.youtube.com/watch?v=GT4u-Y58ngE>

Video2: Introduce our new features for project2:

<https://www.youtube.com/watch?v=VcdkWusnZE>

Evaluation Table:

<https://docs.google.com/forms/d/e/1FAIpQLSf0ccsVdN-nXJCHLluJ-hANZlp8rDKgprJa0oTYiLZSDxh3DA/viewform> Software sustainability Evaluation form

Group Member	Software sustainability Evaluation	Self Assessment	Support my claim
Yinan Wu	Waiting for the results score	3	https://github.com/The-Powerpuff-Girls3/CSC-510-Movie-Mood/tree/yinanwu/feature1
Xuntao Lyu	Waiting for the results score	3	https://github.com/The-Powerpuff-Girls3/CSC-510-Movie-Mood/tree/popular_people
Hua Yang	Waiting for the results score	3	https://github.com/The-Powerpuff-Girls3/CSC-510-Movie-Mood/tree/hyang45/private_chatroom

We will make an appointment with Liwen Du at 2pm on 11/04 Monday ,

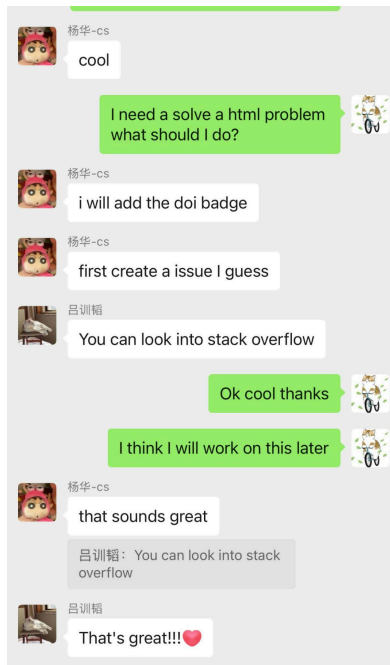


The-Powerpuff-Girls3 / CSC-510-Movie-Mood - Sustainability Evaluation

This software evaluation report is for your software: The-Powerpuff-Girls3 / CSC-510-Movie-Mood. It is a list of recommendations that are based on the survey questions to which you answered "no".

If no text appears below this paragraph, it means you must already be following all of the recommendations made in our evaluation. That's fantastic! We'd love to hear from you, because your project would make a perfect case study. Please get in touch (info@software.ac.uk)

Chat Channel Between group7 (right pic):
(Mainly use Wechat)



Self-assessment for rubric & evidence link:

Rubric	Score	Evidence
Workload is spread over the whole team (one team member is often X times more productive than the others...	3	link
a track record that everyone is contributing a lot	3	link
Number of commits	3	link
Number of commits: by different people	3	link
Issues reports: there are many and are being closed	3	link
Docs: doco generated, format not ugly	3	link
Docs: what: point descriptions of each class/function (in isolation)	3	link
Docs: how: for common use cases X,Y,Z mini-tutorials showing worked examples on how to do X,Y,Z	3	link
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	link

Docs: short video, animated, hosted on your repo. That convinces people why they want to work on your code.	3	link
Use of version control tools	3	link
Test cases exist	3	link
Test cases are routinely executed	3	link
Issues are discussed before they are closed	3	link
Chat channel: exists	3	
Test cases: a large proportion of the issues related to handling failing cases.	3	link
Evidence that the whole team is using the same tools: everyone can get to all tools and files	3	link
Evidence that the whole team is using the same tools (e.g. config files in the repo, updated by lots of different people)	3	link
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	link
Evidence that the members of the team are working across multiple places in the code base	3	link
Short release cycles	3	link
The file .gitignore lists what files should not be saved to the repo.	3	link
The file INSTALL.md lists how to install the code	3	link
The file LICENSE.md lists rules of usage for this repo	3	link
The file CODE-OF-CONDUCT.md lists rules of behavior for this repo	3	link
The file CONTRIBUTING.md lists coding standards and lots of tips on how to extend the system without screwing things up	3	link
The file README.md contains all the following: Video, DOI badge, Badges showing your style checkers, Badges showing your code formatters, Badges showing your syntax checkers, Badges showing your code coverage tools, Badges showing any other Other automated analysis tools	3	link

Self-evaluation:

Question	Answer
Question 1.1: Does your website and documentation provide a clear, high-level overview of your software?	Yes

Question 1.2: Does your website and documentation clearly describe the type of user who should use your software?	Yes
Question 1.3: Do you publish case studies to show how your software has been used by yourself and others?	Yes
Question 2.1: Is the name of your project/software unique?	Yes
Question 2.2: Is your project/software name free from trademark violations?	Yes
Question 3.1: Is your software available as a package that can be deployed without building it?	Yes
Question 3.2: Is your software available for free?	Yes
Question 3.3: Is your source code publicly available to download, either as a downloadable bundle or via access to a source code repository?	Yes
Question 3.4: Is your software hosted in an established, third-party repository like GitHub (https://github.com), BitBucket (https://bitbucket.org), LaunchPad (https://launchpad.net) or SourceForge (https://sourceforge.net)?	Yes
Question 4.1: Is your documentation clearly available on your website or within your software?	Yes
Question 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
Question 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
Question 4.4: Do you provide a comprehensive guide to all your software's commands, functions and options?	Yes
Question 4.5: Do you provide troubleshooting information that describes the symptoms and step-by- step solutions for problems and error messages?	Yes
Question 4.6: If your software can be used as a library, package or service by other software, do you provide comprehensive API documentation?	Yes
Question 4.7: Do you store your documentation under revision control with your source code?	Yes
Question 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?	Yes
Question 5.1: Does your software describe how a user can get help with using your software?	Yes
Question 5.2: Does your website and documentation describe what support, if any, you provide to users and developers?	Yes
Question 5.3: Does your project have an e-mail address or forum that is solely for supporting users?	Yes
Question 5.4: Are e-mails to your support e-mail address received by more than one person?	Yes
Question 5.5: Does your project have a ticketing system to manage bug reports and feature requests?	Yes
Question 5.6: Is your project's ticketing system publicly visible to your users, so they can view bug reports and feature requests?	Yes
Question 6.1: Is your software's architecture and design modular?	Yes
Question 6.2: Does your software use an accepted coding standard or convention?	Yes

Question 7.1: Does your software allow data to be imported and exported using open data formats? e.g. GIF, SVG, HTML, XML, tar, zip, CSV, JSON, NetCDF, or domain specific ones	Yes
Question 7.2: Does your software allow communications using open communications protocols? e.g. HTTP, FTP, XMPP, SOAP over HTTP, or domain-specific ones	Yes
Question 8.1: Is your software cross-platform compatible? e.g. does it run under two or more of Windows, Unix/Linux and Mac OS X, or can be used from within two or more of Internet Explorer, Chrome, Firefox and Safari?	Yes
Question 9.1: Does your software adhere to appropriate accessibility conventions or standards?	Yes
Question 9.2: Does your documentation adhere to appropriate accessibility conventions or standards?	Yes
Question 10.1: Is your source code stored in a repository under revision control?	Yes
Question 10.2: Is each source code release a snapshot of the repository?	Yes
Question 10.3: Are releases tagged in the repository?	Yes
Question 10.4: Is there a branch of the repository that is always stable? (i.e. tests always pass, code always builds successfully)	Yes
Question 10.5: Do you back-up your repository?	Yes
Question 11.1: Do you provide publicly-available instructions for building your software from the source code?	Yes
Question 11.2: Can you build, or package, your software using an automated tool? e.g. Make (https://www.gnu.org/software/make/), ANT (http://ant.apache.org/), Maven (https://maven.apache.org/), CMake (https://cmake.org/), Python setup- tools (https://pypi.python.org/pypi/setuptools), or R package tools (https://cran.r-project.org/doc/manuals/r-devel/R-exts.html)	No
Question 11.3: Do you provide publicly-available instructions for deploying your software?	Yes
Question 11.4: Does your documentation list all third-party dependencies?	Yes
Question 11.5: Does your documentation list the version number for all third-party dependencies?	Yes
Question 11.6: Does your software list the web address, and licences for all third-party dependencies and say whether the dependencies are mandatory or optional?	Yes
Question 11.7: Can you download dependencies using a dependency management tool or pack- age manager? e.g. Ivy (http://ant.apache.org/ivy/), Maven (https://maven.apache.org/), Python pip (https://pypi.python.org/pypi/pip) or setuptools (https://pypi.python.org/pypi/setuptools), PHP Composer (https://getcomposer.org/), Ruby gems (https://rubygems.org), or R PackRat (https://rstudio.github.io/packrat/)	Yes
Question 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
Question 12.1: Do you have an automated test suite for your software?	Yes
Question 12.2: Do you have a framework to periodically (e.g. nightly) run your tests on the latest version of the source code?	Yes

Question 12.3: Do you use continuous integration, automatically running tests whenever changes are made to your source code?	Yes
Question 12.4: Are your test results publicly visible?	Yes
Question 12.5: Are all manually-run tests documented?	Yes
Question 13.1: Does your project have resources (e.g. blog, Twitter, RSS feed, Facebook page, wiki, mailing list) that are regularly updated with information about your software? e.g. release announcements, publications, workshops, conference presentations	Yes
Question 13.2: Does your website state how many projects and users are associated with your project?	No
Question 13.3: Do you provide success stories on your website?	Yes
Question 13.4: Do you list your important partners and collaborators on your website?	Yes
Question 13.5: Do you list your project's publications on your website or link to a resource where these are available?	Yes
Question 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
Question 13.7: Can users subscribe to notifications to changes to your source code repository?	Yes
Question 13.8: If your software is developed as an open source project (and, not just a project developing open source software), do you have a governance model?	No
Question 14.1: Do you accept contributions (e.g. bug fixes, enhancements, documentation updates, tutorials) from people who are not part of your project?	Yes
Question 14.2: Do you have a contributions policy?	Yes
Question 14.3: Is your contributions' policy publicly available?	Yes
Question 14.4: Do contributors keep the copyright/IP of their contributions?	Yes
Question 15.1: Does your website and documentation clearly state the copyright owners of your software and documentation?	Yes
Question 15.2: Does each of your source code files include a copyright statement?	Yes
Question 15.3: Does your website and documentation clearly state the licence of your software?	Yes
Question 15.4: Is your software released under an open source licence?	Yes
Question 15.5: Is your software released under an OSI-approved open-source licence?	Yes
Question 15.6: Does each of your source code files include a licence header?	No
Question 15.7: Do you have a recommended citation for your software?	Yes
Question 16.1: Does your website or documentation include a project roadmap (a list of project and development milestones for the next 3, 6 and 12 months)?	No
Question 16.2: Does your website or documentation describe how your project is funded, and the period over which funding is guaranteed?	Yes
Question 16.3: Do you make timely announcements of the deprecation of components, APIs, etc.?	Yes

Movie Mood

Movie Recommendations for your Personal Feeling

Movie Mood is an invitation to explore the boundless realms of cinema. With a constantly expanding library of films, an intuitive recommendation algorithm, a vibrant new chat room, a seamless account system, and a robust review platform, Movie Mood transforms the way you discover, enjoy, and engage with movies. It's a gateway to unforgettable cinematic adventures.

Why Movie Mood is Better

Private Sharing

Share your feelings and preferences with invited users.

Efficient Recommendations

Get quick, tailored recommendations for your interests.

Adaptable Suggestions

Recommendations adapt to your changing preferences.

Automatic Updates

Get the latest TV series & Celebrity news automatically.

Simple and Seamless

Built with CSS, HTML, JS, Python, Flask, and SQLite.

Key Features & Milestones

Private Chat Rooms

Exclusive, invite-only spaces for personalized movie and series discussions, creating a tailored experience.

Up-to-Date TV Recommendations

Get automatic recommendations for trending TV series, so you never miss new releases.

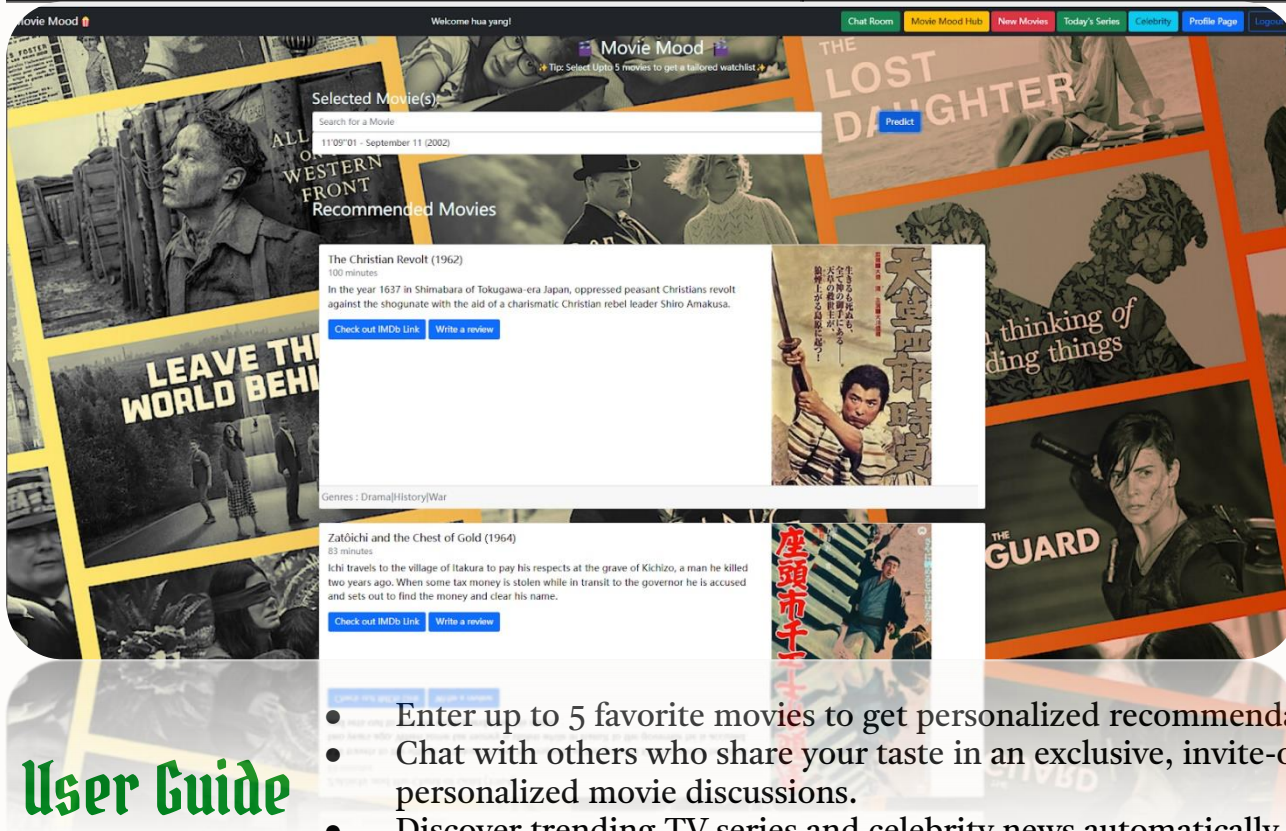
Celebrity News Feed

Stay updated with the latest celebrity news—a one-stop platform for entertainment and news fans.

Movie Mood Test Suite

The "Movie Mood" test suite includes 46 test cases, ensuring functionality and stability across key areas:

- **Private Chat:** tests for creating and closing invite-only chat rooms
- **Chat Room:** Tests for establishing/tearing down connections and broadcasting chats
- **Recommender System:** Tests for inputting comedy, cartoon, action, and multiple movies
- **Search Feature:** Tests for different keywords, including type, incorrect input, and year
- **Utility Functions:** Tests for storing watch history, creating colored tags, and sending notifications



User Guide

- Enter up to 5 favorite movies to get personalized recommendations.
- Chat with others who share your taste in an exclusive, invite-only space for personalized movie discussions.
- Discover trending TV series and celebrity news automatically.
- Create an account to save your favorite movies and track your viewing history.

Repo Link



Video Link



Contact Information For more intriguing ideas or problems, please contact Group 7 !!!