Software Engineering CSC 510 Project 2 Report

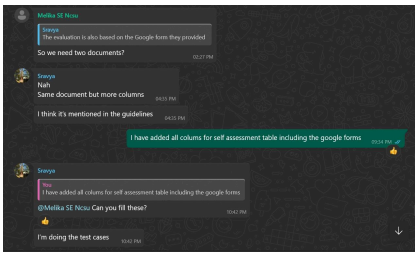
StreamR: [GitHub Repository Link]

Group 54: StreamR

Chirag Rajesh Hegde (chegde), Melika Ahmadi Ranjar (mahmadi), Sravya Yepuri (syepuri)

Evaluation	Self-Assessment (Same for all)	Description
Workload is spread over the whole team (one team member is often Xtimes more productive than the others...but nevertheless, here is a track record that everyone is contributing a lot)	3	We divided the improvement milestones between the three of us and worked on the issues together
Number of commits	3	60
Number of commits: by different people	3	[sravya: 19, Chirag: 17, melika: 24] The number of commits does not justify the lines of code and code quality.
Issues reports	3	The issues are reported as and when we come across them in the tests.
Issues are being closed	3	We close issues by giving comments and linking the commits.
Docs: doco generated, format not ugly	3	Generated well documented wiki.
Docs: what: point descriptions of each class/function (in isolation)	3	Had a document for describing features/functions
Docs: how: for common use cases X,Y,Z mini-tutorials showing worked examples on how to do X,Y,Z	3	There are sample videos linked for each functionailty
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	This is integrated with the original poster as well
Docs: short video, animated, hosted on	3	Although there were so many

your repo. That convinces people why they want to work on your code.		features at the beginning and our team implemented most of them, there is always room for improvement, which is discussed in the demo of our improved software
Use of version control tools	3	Git
Test cases exist	3	There are existing test cases, which completely pass
Test cases are routinely executed	3	As each team member was working on a specific feature, we also routinely executed test cases to make sure our code is working properly and this was confirmed by the whole team as well

Evaluation	Self-Assessment	Description
Issues are discussed before they are closed	3	All the issues were mentioned, discusses, and solved before being close in either meetings or via chatting
Chat channel: exists	3	Whatsapp group 
Test cases: a large proportion of the issues related to handling failing cases.	3	There are test cases that show failing and passing cases
Evidence that the whole team is using the same tools: everyone can get to all tools and files	3	The git repository will confirm this
Evidence that the whole team is using the same tools (e.g. config files in the repo, updated by lots of different people)	3	Each team member was using the latest version of the forked repo
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	Everyone is aware of all parts of the code that has been modified and maintained
Evidence that the members of the team are working across multiple places in the code base	3	Other than the features which were assigned to each person, all team members have been working on the main file as well
Short release cycles	3	The team tried to release the working features before moving to the next one, which made it a short release cycle
The file .gitignore lists what files should not be saved to the repo. See [examples]i(https://github.com/github/gitignore)	3	Added in github repo
The file INSTALL.md lists how to install the code	3	Added in github repo

The file LICENSE.md lists rules of usage for this repo	3	Added in github repo
The file CODE-OF-CONDUCT.md lists rules of behavior for this repo; e.g. see example	3	Added in github repo
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	Added in github repo
The file README.md contains all the following	3	Added in github repo

Evaluation	Self-Assessment	Description
Video	3	There is a YouTube video available to demonstrate the improved version of the software
DOI badge: exists. To get a Digitial Object Indentifier, regiser the project at Zenodo. DOI badges look like this: 	3	Yes it is added in the readme.
Badges showing your style checkers	3	Shown as Black
Badges showing your code formatters.	3	Autopep8
Badges showing your syntax checkers.	3	Pylance
Badges showing your code coverage tools	3	Code Coverage, CodeCov
Badges showing any other Other automated analysis tools	3	Maintained, Issues, ...
Question 1.1: Does your website and documentation provide a clear, high-level overview of your software?	3	README, Document
Question 1.2: Does your website and documentation clearly describe the type of user who should use your software?	3	Yes, it shows the user demographic.
Question 1.3: Do you publish case studies to show how your software has been used by yourself and others?	0	We will have to work on this
Question 2.1: Is the name of your project/software unique?*	3	The name for our software is 'StreamR'
Question 2.2: Is your project/software name free from trademark violations?*	3	Yes, we checked https://tmsearch.uspto.gov/.
Question 3.1: Is your software available as a package that can be deployed without building it?	2	It has the potential for that but it is not currently available

Evaluation	Self-Assessment	Description
Question 3.2: Is your software available for free?*	3	Yes, just need to clone and install from Git
Question 3.3: Is your source code publicly available to download, either as a downloadable bundle or via access to a source code repository?	3	Yes
Question 3.4: Is your software hosted in an established, third-party repositorylikeGitHub (https://github.com), BitBucket (https://bitbucket.org),LaunchPad (https://launchpad.net) orSourceForge (https://sourceforge.net)?	3	GitHub
Question 4.1: Is your documentation clearly available on your website or within your software?	3	README, Documentation is also available on the GitHub page
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?	3	The installation process and different steps are available on the GitHub repository
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?	3	Yes our wiki page provides all details.
Question 4.4: Do you provide a comprehensive guide to all your software's commands, functions and options?	3	Comments, Documents
Question 4.5: Do you provide troubleshooting information that describes the symptoms and step-by-step solutions for problems and error messages?	1	Not in case we have not seen it

Evaluation	Self-Assessment	Description
Question 4.6: If your software can be used as a library, package or service by other software, do you provide comprehensive API documentation?	3	There is API documentation in the wiki page.
Question 4.7: Do you store your documentation under revision control with your source code?	3	Yes, its is stored under revision control with source code ensuring version tracking and syncronizatiojnn.
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?	2	For this submission the improvements are mentioned
Question 5.1: Does your software describe how a user can get help with using your software?	3	Yes , it does in the wiki page.
Question 5.2: Does your website and documentation describe what support, if any, you provide to users and developers?	3	Yes via email.
Question 5.3: Does your project have an e-mail address or forum that is solely for supporting users?	3	Developers emails are mentioned
Question 5.4: Are e-mails to your support e-mail address received by more than one person?	3	Common Emails
Question 5.5: Does your project have a ticketing system to manage bug reports and feature requests?	3	Github Issues
Question 5.6: Is your project's ticketing system publicly visible to your users, so they can view bug reports and feature requests?	3	Yes

Evaluation	Self-Assessment	Description
Question 6.1: Is your software's architecture and design modular?	3	Yes it is modular.
Question 6.2: Does your software use an accepted coding standard or convention?	3	Autopep8
Question 7.1: Does your software allow data to be imported and exported using open data formats?	3	Yes,The supports importing and exporting data in open formats
Question 7.2: Does your software allow communications using open communications protocols?	3	Yes, we use HTTP for our project.
Question 8.1: Is your software cross-platform compatible?	3	Yes, our project can be run on different platforms.
Question 9.1: Does your software adhere to appropriate accessibility conventions or standards?	3	Yes, we adhere to these standards.
Question 9.2: Does your documentation adhere to appropriate accessibility conventions or standards?	3	Yes, we adhere to these standards.
Question 10.1: Is your source code stored in a repository under revision control?	3	Yes, it is in git revision control, release 1.
Question 10.2: Is each source code release a snapshot of the repository?	2	Current;y, we are in release 1 of our project.
Question 10.3: Are releases tagged in the repository?	3	Present in the README
Question 10.4: Is there a branch of the repository that is always stable? (i.e. tests always pass, code always builds successfully)	3	Yes, the master branch is always stable.
Question 10.5: Do you back-up your repository?	3	Local for each team member
Question 11.1: Do you provide publicly-available instructions for building your software from the source code?	3	The setup and installation guide is mentioned on the wiki page.

Evaluation	Self-Assessment	Description
Question 11.2: Can you build, or package, your software using an automated tool?	3	We can build it using setup.py
Question 11.3: Do you provide publicly-available instructions for deploying your software?	3	This link helps users to help deploy the software.
Question 11.4: Does your documentation list all third-party dependencies?	3	They are listed on the dependencies section in our documentation.
Question 11.5: Does your documentation list the version number for all third-party dependencies?	3	Yes, they are included in the requirements.
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?	3	They are shown in the wiki page.
Question 11.7: Can you download dependencies using a dependency management tool or package manager?	3	We can use the requirements file to install the dependencies.
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?	2	Integration test
Question 12.1: Do you have an automated test suite for your software?	3	We have unittest framework to run tests
Question 12.2: Do you have a framework to periodically (e.g. nightly) run your tests on the latest version of the source code?	2	We have unittest framework to run tests
Question 12.3: Do you use continuous integration, automatically running tests whenever changes are made to your source code?	1	This test runs on search and some common functionalities
Question 12.4: Are your test results publicly visible?	3	Yes, they can be visible when run by the user.

Evaluation	Self-Assessment	Description
Question 12.5: Are all manually-run tests documented?	3	They are documented in the testing section in our documentation.
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?	3	There is a wiki page that has information about our project.
Question 13.2: Does your website state how many projects and users are associated with your project?	0	It doesn't show number of users as we do not require it to be shown in the movie recommender.
Question 13.3: Do you provide success stories on your website?	0	We are yet to implement this.
Question 13.4: Do you list your important partners and collaborators on your website?	2	The websites that their APIs were used
Question 13.5: Do you list your project's publications on your website or link to a resource where these are available?	2	We link the website sample API demo for resources as a reference.
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?	0	There are no third part publications that link our website.
Question 13.7: Can users subscribe to notifications to changes to your source code repository?	3	Yes, they can subscribe using watch button on our repo.
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?	3	We do have a governance model in our contributions wiki.
Question 14.1: Do you accept contributions (e.g. bug fixes, enhancements, documentation updates, tutorials) from people who are not part of your project?	3	Yes, we accept them and support these issues.

Evaluation	Self-Assessment	Description
Question 14.2: Do you have a contributions policy?	3	It is present in our documentation.
Question 14.3: Is your contributions' policy publicly available?	3	It is present in our documentation for public.
Question 14.4: Do contributors keep the copyright/IP of their contributions?	3	Yes, there is a copyright.
Question 15.1: Does your website and documentation clearly state the copyright owners of your software and documentation?	3	Yes , The license does.
Question 15.2: Does each of your source code files include a copyright statement?	3	Yes, Each files has copyright statement.
Question 15.3: Does your website and documentation clearly state the licence of your software?	3	In the GitHub repository
Question 15.4: Is your software released under an open source licence?	3	MIT Licence
Question 15.5: Is your software released under an OSI-approved open-source licence?	3	MIT Licence
Question 15.6: Does each of your source code files include a licence header?	3	Yes , each of the files has licence header.
Question 15.7: Do you have a recommended citation for your software?	3	Yes , added citation detailss to readme.
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)?	3	The README shows the future enhancement plans for our project.

Evaluation	Self-Assessment	Description
Question 16.2: Does your website or documentation describe how your project is funded, and the period over which funding is guaranteed?	3	We are self funded.
Question 16.3: Do you make timely announcements of the deprecation of components, APIs, etc.?	3	As and when there are changes, they will be announced in our wiki page.