Robbie Thomas - rtthoma3@ncsu.edu
Zayaan Husain Siddiqui - Zsiddiq@ncsu.edu
Anuraag Jajoo - ajajoo3@ncsu.edu

Group 31 - Project 2

Project2 Repository:
https://github.com/SE-Fall24-RZA/Recipe_Recommender/tree/group31-fall24

Our work is on branch “group31-fall24”
Previous semester work is on branch “master” (we accidentally made one commit to this branch, but have since reverted it)

GitHub Usernames to Track Contributions:

Robbie Thomas = `robbiethomas1101` or `rtthoma3`
Zayaan Husain Siddiqui = `siddiquizayaan`
Anuraag Jajoo = `anurgj`

Table Sums

- Project 2 Rubric Sum = [Robbie: 99, Zayaan: 96, Anuraag: 97]
- Software Sustainability Evaluation Sum = 155
- Overall Sum = [Robbie: 254, Zayaan: 251, Anuraag: 252]

Item	Self-Score	Evidence
Workload is spread over the whole team (one team member is often Xtimes more productive than the others…	Robbie: 3 Zayaan: 3 Anuraag: 3	https://github.com/SE-Fall24-RZA/Recipe_Recommender/commits/group31-fall24/
but nevertheless, here is a track record that everyone is contributing a lot)	Robbie: 3 Zayaan: 2 Anuraag: 3	https://github.com/SE-Fall24-RZA/Recipe_Recommender/commits/group31-fall24/
Number of commits	3	https://github.com/SE-Fall24-RZA/Recipe_Recommender/commits/group31-fall24/
Number of commits: by different people	Robbie: 3 Zayaan: 2 Anuraag: 2	https://github.com/SE-Fall24-RZA/Recipe_Recommender/commits/group31-fall24/
Issues reports: there are **many**	3	https://github.com/SE-Fall24-RZA/Recipe_Recommender/issues

Issues are being closed	3	https://github.com/SE-Fall24-RZA/Recipe_Recommender/issues?q=is%3Aissue+is%3Aclosed
Docs: doco generated, format not ugly	3	Lots of in-line code comments that explain parts of the code
Docs: what: point descriptions of each class/function (in isolation)	3	Almost all functions have a comment block before them detailing what they do Ex) https://github.com/SE-Fall24-RZA/Recipe_Recommender/blob/group31-fall24/Code/backend/dao/recipesDAO.js
Docs: how: for common use cases X,Y,Z mini-tutorials showing worked examples on how to do X,Y,Z	2	
Docs: why: docs tell a story, motivate the whole thing, deliver a punchline that makes you want to rush out and use the thing	2	
Docs: short video, animated, hosted on your repo. That convinces people why they want to work on your code.	3	
Use of version control tools	3	Used Git
Test cases exist	3	Frontend system tests: https://github.com/SE-Fall24-RZA/Recipe_Recommender/tree/group31-fall24/Code/frontend/src/components/__tests__ Backend unit/integration tests: https://github.com/SE-Fall24-RZA/Recipe_Recommender/tree/group31-fall24/Code/back

		end/__tests__
Test cases are routinely executed	3	Workflows run tests on push and pull requests https://github.com/SE-Fall24-RZA/Recipe_Recommender/actions/workflows/backendTests.yml
Issues are discussed before they are closed	2	Issues are discussed in our group discord channel, we often let each other know when something has changed. Issues are also closed with comments on GitHub, letting everyone know how/if they are resolved.
Chat channel: exists	3	We have a discord channel for communication. Linked below is a folder containing some screenshots of our conversations: Project2_Chat_Screen…
Test cases: a large proportion of the issues related to handling failing cases.	2	https://github.com/SE-Fall24-RZA/Recipe_Recommender/issues?q=is%3Aissue+is%3Aclosed A decent portion of issues are about tests that were failing
Evidence that the whole team is using the same tools: everyone can get to all tools and files	Robbie: 3 Zayaan: 3 Anuraag: 3	Everyone has access to and worked on the GitHub repository which contains all needed files and instructions on how to install needed tools
Evidence that the whole team is using the same tools (e.g. config files in the repo, updated by lots of different people)	Robbie: 3 Zayaan: 3 Anuraag: 3	https://github.com/SE-Fall24-RZA/Recipe_Recommender/blob/group31-fall24/Code/backend/env.sample
Evidence that the whole team is using the same tools (e.g. tutor can ask anyone to share screen,	Robbie: 3 Zayaan: 3 Anuraag: 3	

they demonstrate the system running on their computer)		
Evidence that the members of the team are working across multiple places in the code base	Robbie: 3 Zayaan: 2 Anuraag: 2	https://github.com/SE-Fall24-RZA/Recipe_Recommender/commits/group31-fall24/ Commits show where each of us were working in the codebase over time
Short release cycles	1	
The file .gitignore lists what files should not be saved to the repo. See [examples]i(https://github.com/github/gitignore)	3	https://github.com/SE-Fall24-RZA/Recipe_Recommender/blob/group31-fall24/.gitignore
The file INSTALL.md lists how to install the code	3	https://github.com/SE-Fall24-RZA/Recipe_Recommender/blob/group31-fall24/INSTALL.md
The file LICENSE.md lists rules of usage for this repo	3	https://github.com/SE-Fall24-RZA/Recipe_Recommender/blob/group31-fall24/LICENSE
The file CODE-OF-CONDUCT.md lists rules of behavior for this repo; e.g. see example	3	https://github.com/SE-Fall24-RZA/Recipe_Recommender/blob/group31-fall24/CODE_OF_CONDUCT.md
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	https://github.com/SE-Fall24-RZA/Recipe_Recommender/blob/group31-fall24/CONTRIBUTING.md
The file README.md contains all the following	3	https://github.com/SE-Fall24-RZA/Recipe_Recommender/blob/group31-fall24/README.md
Video	3	README (see above)

DOI badge: exists. To get a Digitial Object Indentifier, regiser the project at Zenodo.	3	README (see above)
Badges showing your style checkers	3	README (see above)
Badges showing your code formatters.	3	README (see above)
Badges showing your syntax checkers.	3	README (see above)
Badges showing your code coverage tools	3	README (see above)
Badges showing any other Other automated analysis tools	3	README (see above)
Question 1.1: Does your website and documentation provide a clear, high-level overview of your software?	3 (Yes)	
Question 1.2: Does your website and documentation clearly describe the type of user who should use your software?	3 (Yes)	
Question 1.3: Do you publish case studies to show how your software has been used by yourself and others?	2 (Yes)	We have created a video showing how to use the software
Question 2.1: Is the name of your project/software unique?	3 (Yes)	

Question 2.2: Is your project/software name free from trademark violations?	3 (Yes)	
Question 3.1: Is your software available as a package that can be deployed without building it?	1 (No)	
Question 3.2: Is your software available for free?	3 (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?	3 (Yes)	Project is in a public GitHub repository
Question 3.4: Is your software hosted in an established, third-party repository likeGitHub (https://github.com), BitBucket (https://bitbucket.org),LaunchPad (https://launchpad.net) orSourceForge (https://sourceforge.net)?	3 (Yes)	Project is in a public GitHub repository
Question 4.1: Is your documentation clearly available on your website or within your software?	3 (Yes)	Documentation can be found in the README
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 (Yes)	INSTALL.md explains how to set up the system. README explains how to use it
Question 4.3: If you provide	3 (Yes)	INSTALL.md explains how to

more extensive documentation, does this provide clear, step-by-step instructions on how to deploy and use your software?		set up the system. README explains how to use it
Question 4.4: Do you provide a comprehensive guide to all your software's commands, functions and options?	3 (Yes)	Located in README
Question 4.5: Do you provide troubleshooting information that describes the symptoms and step-by-step solutions for problems and error messages?	1 (No)	
Question 4.6: If your software can be used as a library, package or service by other software, do you provide comprehensive API documentation?	N/A	
Question 4.7: Do you store your documentation under revision control with your source code?	3 (Yes)	Documentation is found in GitHub repository with source code
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?	3 (Yes)	README contains a section about what was added in the most recent versions
Question 5.1: Does your software describe how a user can get help with using your software?	2 (Yes)	They can submit issues in GitHub

Question 5.2: Does your website and documentation describe what support, if any, you provide to users and developers?	2 (Yes)	CONTRIBUTING.md explains what to do if you want to add to the project or find a problem that needs fixing
Question 5.3: Does your project have an e-mail address or forum that is solely for supporting users?	1 (No)	
Question 5.4: Are e-mails to your support e-mail address received by more than one person?	N/A	
Question 5.5: Does your project have a ticketing system to manage bug reports and feature requests?	2 (Yes)	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?	2 (Yes)	GitHub Issues
Question 6.1: Is your software's architecture and design modular?	3 (Yes)	Separated frontend and backend components
Question 6.2: Does your software use an accepted coding standard or convention?	3 (Yes)	
Question 7.1: Does your software allow data to be imported and exported using open data formats?	0 (No)	
Question 7.2: Does your software allow	3 (Yes)	API calls use HTTP requests

communications using open communications protocols?		
Question 8.1: Is your software cross-platform compatible?	3 (Yes)	Project can be installed on multiple platforms (both Mac and Windows tested)
Question 9.1: Does your software adhere to appropriate accessibility conventions or standards?	3 (Yes)	
Question 9.2: Does your documentation adhere to appropriate accessibility conventions or standards?	3 (Yes)	
Question 10.1: Is your source code stored in a repository under revision control?	3 (Yes)	Project is in a GitHub repository
Question 10.2: Is each source code release a snapshot of the repository?	3 (Yes)	
Question 10.3: Are releases tagged in the repository?	1 (No)	
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)	“master” branch of old code is always stable. “Group31-fall24” branch is our team’s main branch, only stable code should be pushed here. Any failures were quickly fixed.
Question 10.5: Do you back-up your repository?	3 (Yes)	
Question 11.1: Do you provide publicly-available instructions for building your software from the source code?	3 (Yes)	INSTALL.md

Question 11.2: Can you build, or package, your software using an automated tool?	1 (No)	
Question 11.3: Do you provide publicly-available instructions for deploying your software?	3 (Yes)	INSTALL.md
Question 11.4: Does your documentation list all third-party dependencies?	3 (Yes)	README lists other tools/technologies used
Question 11.5: Does your documentation list the version number for all third-party dependencies?	2 (Yes)	Can be found in package.json files
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?	1 (No)	
Question 11.7: Can you download dependencies using a dependency management tool or package manager?	3 (Yes)	NPM can be used to install 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?	3 (Yes)	Automated tests run in GitHub actions after pushing
Question 12.1: Do you have an automated test suite for your software?	3 (Yes)	Tests are run using jest. These tests are automatically run by GitHub actions on push

Question 12.2: Do you have a framework to periodically (e.g. nightly) run your tests on the latest version of the source code?	3 (Yes)	Tests are run every time there is a push
Question 12.3: Do you use continuous integration, automatically running tests whenever changes are made to your source code?	3 (Yes)	Tests are run every time there is a push to any branch
Question 12.4: Are your test results publicly visible?	3 (Yes)	README has a badge for passing tests and code coverage. Details can be found in the individual workflows.
Question 12.5: Are all manually-run tests documented?	3 (Yes)	Frontend system tests are documented
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?	1 (No)	
Question 13.2: Does your website state how many projects and users are associated with your project?	3 (Yes)	Number of contributors noted in README
Question 13.3: Do you provide success stories on your website?	0 (No)	
Question 13.4: Do you list your important partners and collaborators on your website?	2 (Yes)	Contributors are listed in README

Question 13.5: Do you list your project's publications on your website or link to a resource where these are available?	0 (No)	
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 (No)	
Question 13.7: Can users subscribe to notifications to changes to your source code repository?	0 (No)	
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 (Yes)	CONTRIBUTING.md
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)	The repository is able to be freely forked by anyone. If they want to contribute, they only need to follow the guidelines in CONTRIBUTING.md
Question 14.2: Do you have a contributions policy?	3 (Yes)	CONTRIBUTING.md
Question 14.3: Is your contributions' policy publicly available?	3 (Yes)	CONTRIBUTING.md
Question 14.4: Do contributors keep the copyright/IP of their contributions?	N/A	

Question 15.1: Does your website and documentation clearly state the copyright owners of your software and documentation?	3 (Yes)	MIT License explains this LICENSE.md
Question 15.2: Does each of your source code files include a copyright statement?	1 (No)	
Question 15.3: Does your website and documentation clearly state the licence of your software?	3 (Yes)	MIT License explains this LICENSE.md
Question 15.4: Is your software released under an open source licence?	3 (Yes)	MIT License explains this LICENSE.md
Question 15.5: Is your software released under an OSI-approved open-source licence?	3 (Yes)	LICENSE.md
Question 15.6: Does each of your source code files include a licence header?	1 (No)	
Question 15.7: Do you have a recommended citation for your software?	0 (No)	
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 (Yes)	Future scope listed in README
Question 16.2: Does your website or documentation describe how your project is funded, and the period over	0 (No)	

which funding is guaranteed?		
Question 16.3: Do you make timely announcements of the deprecation of components, APIs, etc.?	3 (Yes)	No components have become deprecated