NC STATE UNIVERSITY

Master of Science, Department of Computer Science
North Carolina State University

Coursework Session
2024 - 2025

CSC 510-001
Software Engineering

Team Members	Unity ID
Meet Vora	mvora2
Raj Patel	rbpatel4
Vihar Shah	vshah23

PROJECT 2
NCSU CAMPUS JOB REVIEW SYSTEM 2.0

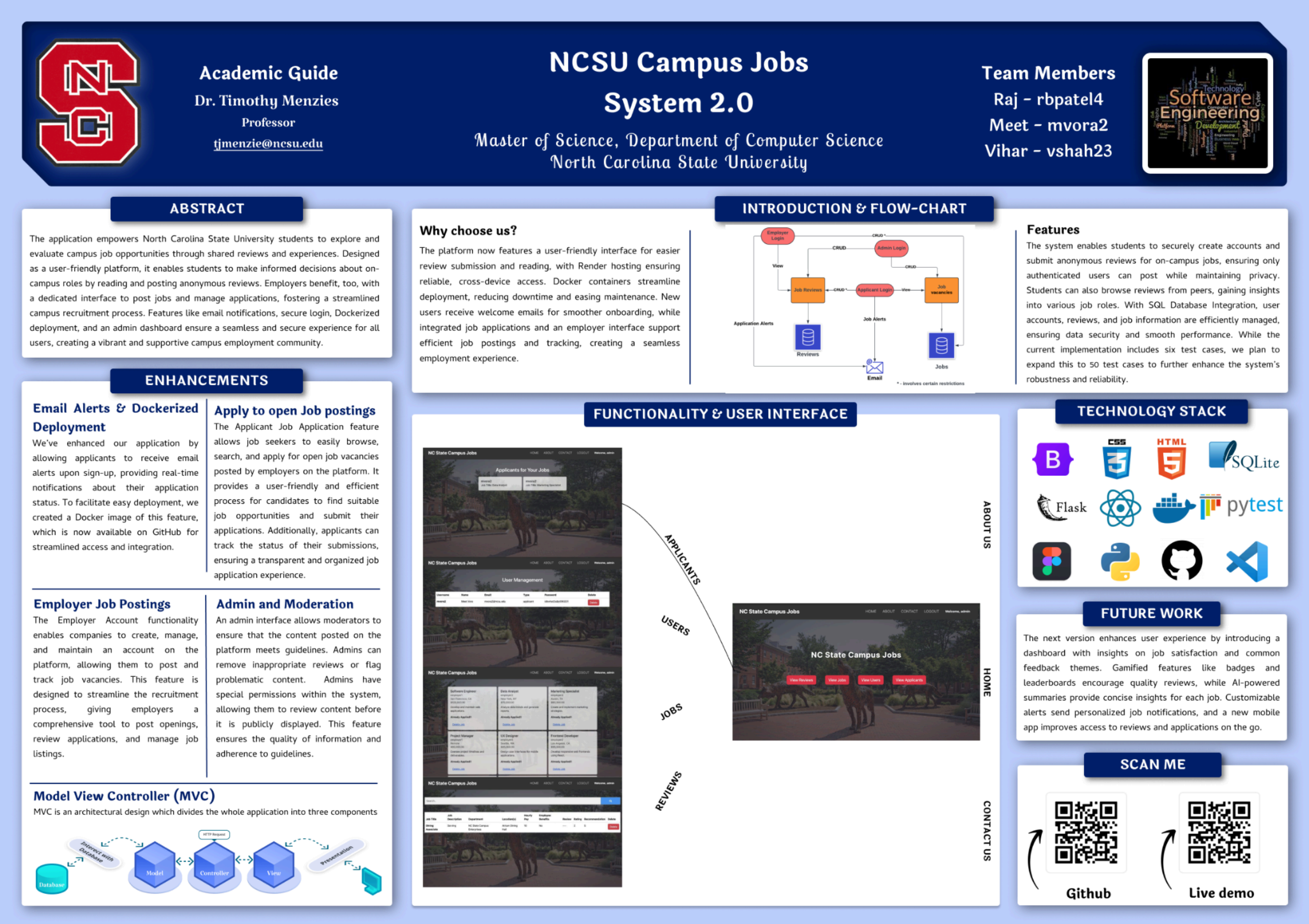
Declaration

We hereby declare that all work submitted for this coursework is our own, except where explicitly stated otherwise.

NCSU CAMPUS JOB REVIEW SYSTEM 2.0

Github Link: https://github.com/Rmv-se-510/NCSU_Campus_Jobs_Review_System_2.0

Documentation link:
https://rmv-se-510.github.io/NCSU_Campus_Jobs_Review_System_2.0/app/models.html
Link to website: https://ncsu-campus-jobs-review-system-2-0.onrender.com/
Link to Video:
https://drive.google.com/file/d/13QHYH2dspuiw1XZEtsmbRkKkH-YnE_Em/view?usp=sharing



NCSU Campus Jobs System 2.0

Master of Science, Department of Computer Science
North Carolina State University

Academic Guide
Dr. Timothy Menzies
Professor
tjmenzie@ncsu.edu

Team Members
Raj - rbpatel4
Meet - mvora2
Vihar - vshah23

ABSTRACT

The application empowers North Carolina State University students to explore and evaluate campus job opportunities through shared reviews and experiences. Designed as a user-friendly platform, it enables students to make informed decisions about on-campus roles by reading and posting anonymous reviews. Employers benefit, too, with a dedicated interface to post jobs and manage applications, fostering a streamlined campus recruitment process. Features like email notifications, secure login, Dockerized deployment, and an admin dashboard ensure a seamless and secure experience for all users, creating a vibrant and supportive campus employment community.

ENHANCEMENTS

Email Alerts & Dockerized Deployment

We've enhanced our application by allowing applicants to receive email alerts upon sign-up, providing real-time notifications about their application status. To facilitate easy deployment, we created a Docker image of this feature, which is now available on GitHub for streamlined access and integration.

Apply to open Job postings

The Applicant Job Application feature allows job seekers to easily browse, search, and apply for open job vacancies posted by employers on the platform. It provides a user-friendly and efficient process for candidates to find suitable job opportunities and submit their applications. Additionally, applicants can track the status of their submissions, ensuring a transparent and organized job application experience.

Employer Job Postings

The Employer Account functionality enables companies to create, manage, and maintain an account on the platform, allowing them to post and track job vacancies. This feature is designed to streamline the recruitment process, giving employers a comprehensive tool to post openings, review applications, and manage job listings.

Admin and Moderation

An admin interface allows moderators to ensure that the content posted on the platform meets guidelines. Admins can remove inappropriate reviews or flag problematic content. Admins have special permissions within the system, allowing them to review content before it is publicly displayed. This feature ensures the quality of information and adherence to guidelines.

Model View Controller (MVC)

MVC is an architectural design which divides the whole application into three components

INTRODUCTION & FLOW-CHART

Why choose us?

The platform now features a user-friendly interface for easier review submission and reading, with Render hosting ensuring reliable, cross-device access. Docker containers streamline deployment, reducing downtime and easing maintenance. New users receive welcome emails for smoother onboarding, while integrated job applications and an employer interface support efficient job postings and tracking, creating a seamless employment experience.

Features

The system enables students to securely create accounts and submit anonymous reviews for on-campus jobs, ensuring only authenticated users can post while maintaining privacy. Students can also browse reviews from peers, gaining insights into various job roles. With SQL Database Integration, user accounts, reviews, and job information are efficiently managed, ensuring data security and smooth performance. While the current implementation includes six test cases, we plan to expand this to 50 test cases to further enhance the system's robustness and reliability.

FUNCTIONALITY & USER INTERFACE

TECHNOLOGY STACK

FUTURE WORK

The next version enhances user experience by introducing a dashboard with insights on job satisfaction and common feedback themes. Gamified features like badges and leaderboards encourage quality reviews, while AI-powered summaries provide concise insights for each job. Customizable alerts send personalized job notifications, and a new mobile app improves access to reviews and applications on the go.

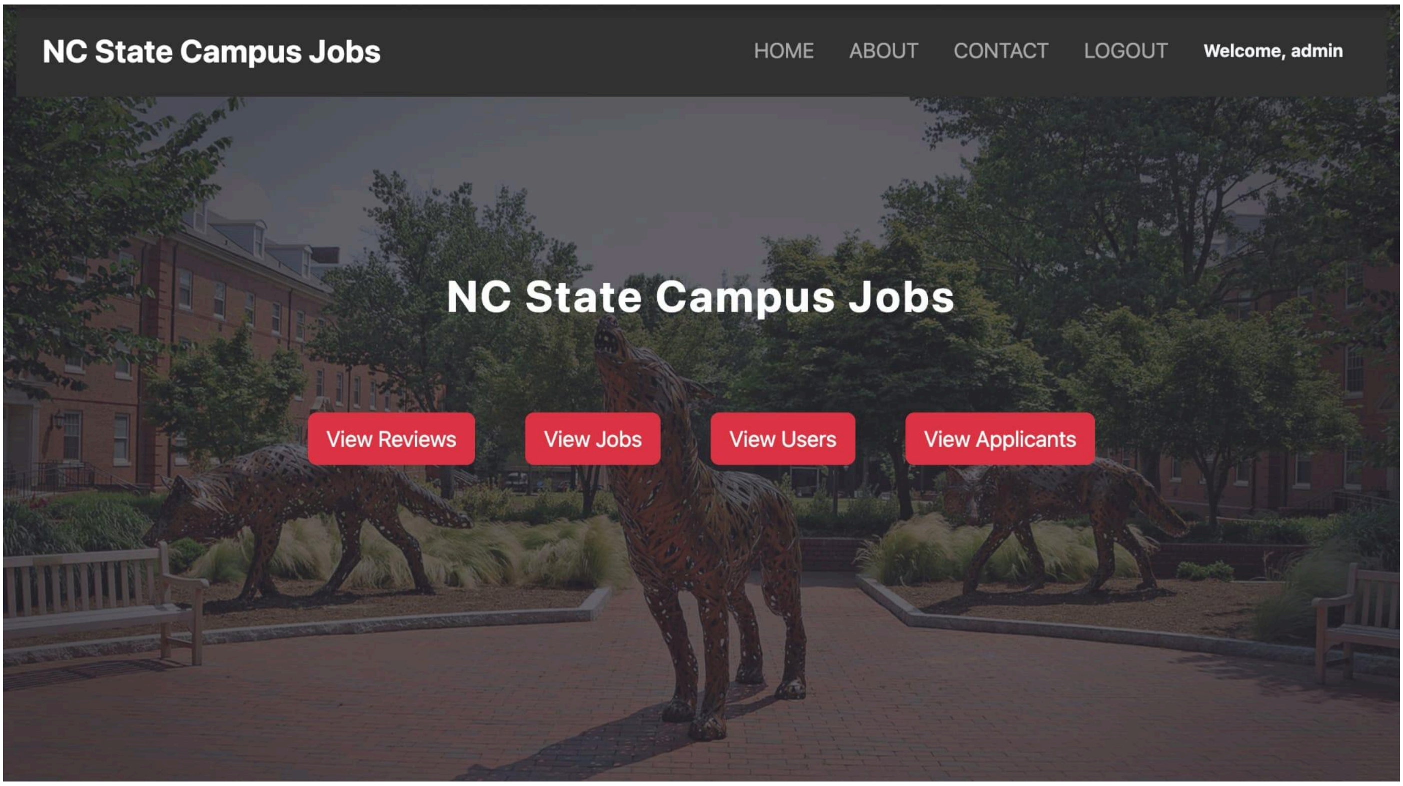
SCAN ME

Github

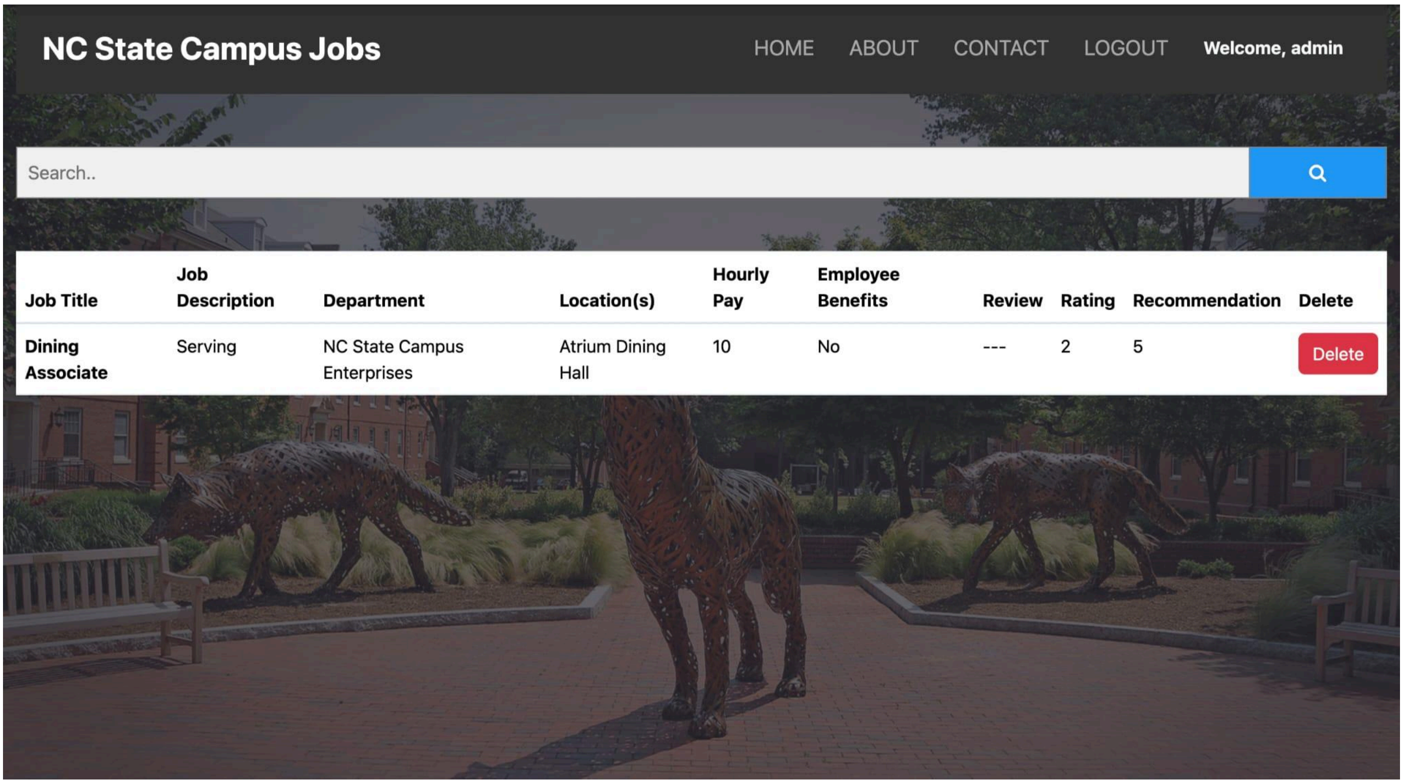
Live demo

Screenshots of Enhanced Functionalities

Home Page

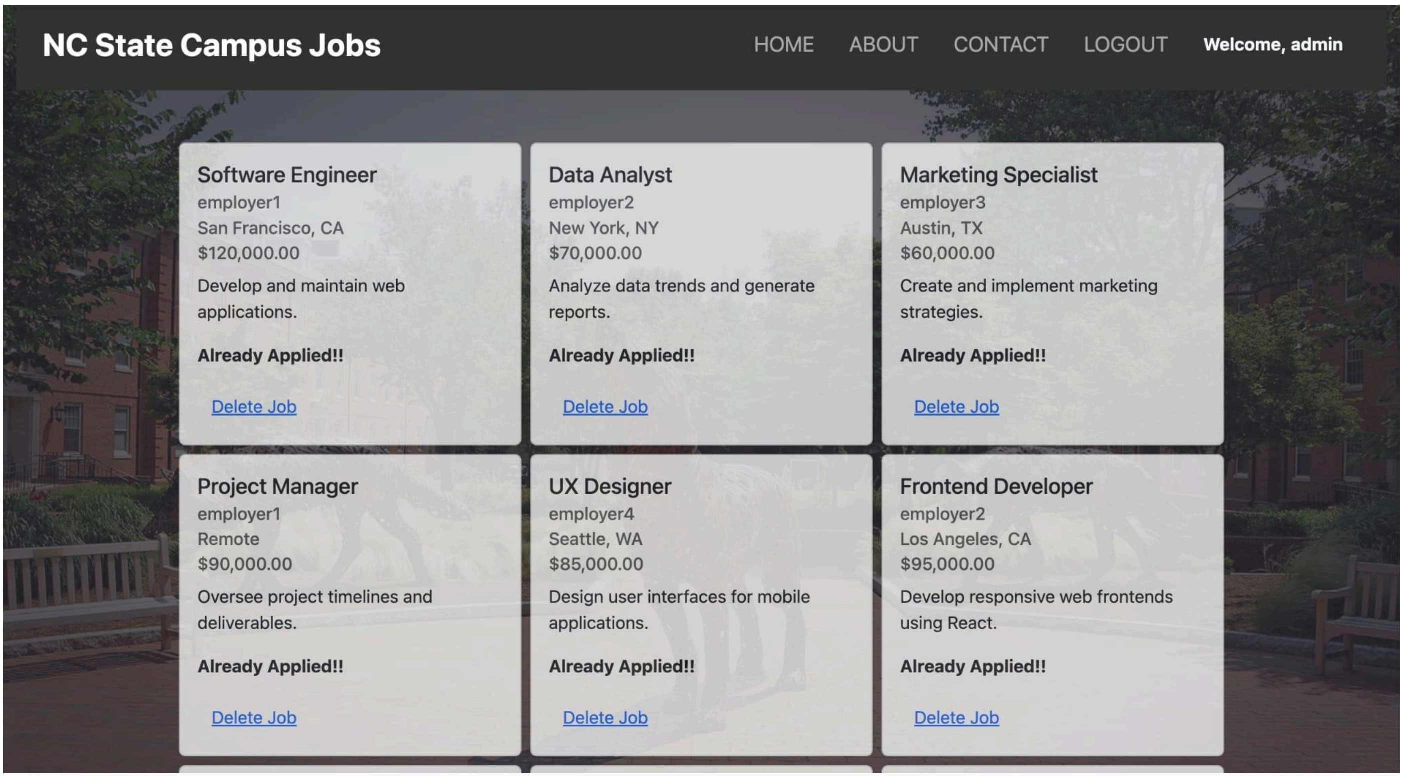


Review page



Job Title	Job Description	Department	Location(s)	Hourly Pay	Employee Benefits	Review	Rating	Recommendation	Delete
Dining Associate	Serving	NC State Campus Enterprises	Atrium Dining Hall	10	No	---	2	5	Delete

Jobs pages



Software Engineer
employer1
San Francisco, CA
$120,000.00
Develop and maintain web applications.

Already Applied!!

Data Analyst
employer2
New York, NY
$70,000.00
Analyze data trends and generate reports.

Already Applied!!

Marketing Specialist
employer3
Austin, TX
$60,000.00
Create and implement marketing strategies.

Already Applied!!

Project Manager
employer1
Remote
$90,000.00
Oversee project timelines and deliverables.

Already Applied!!

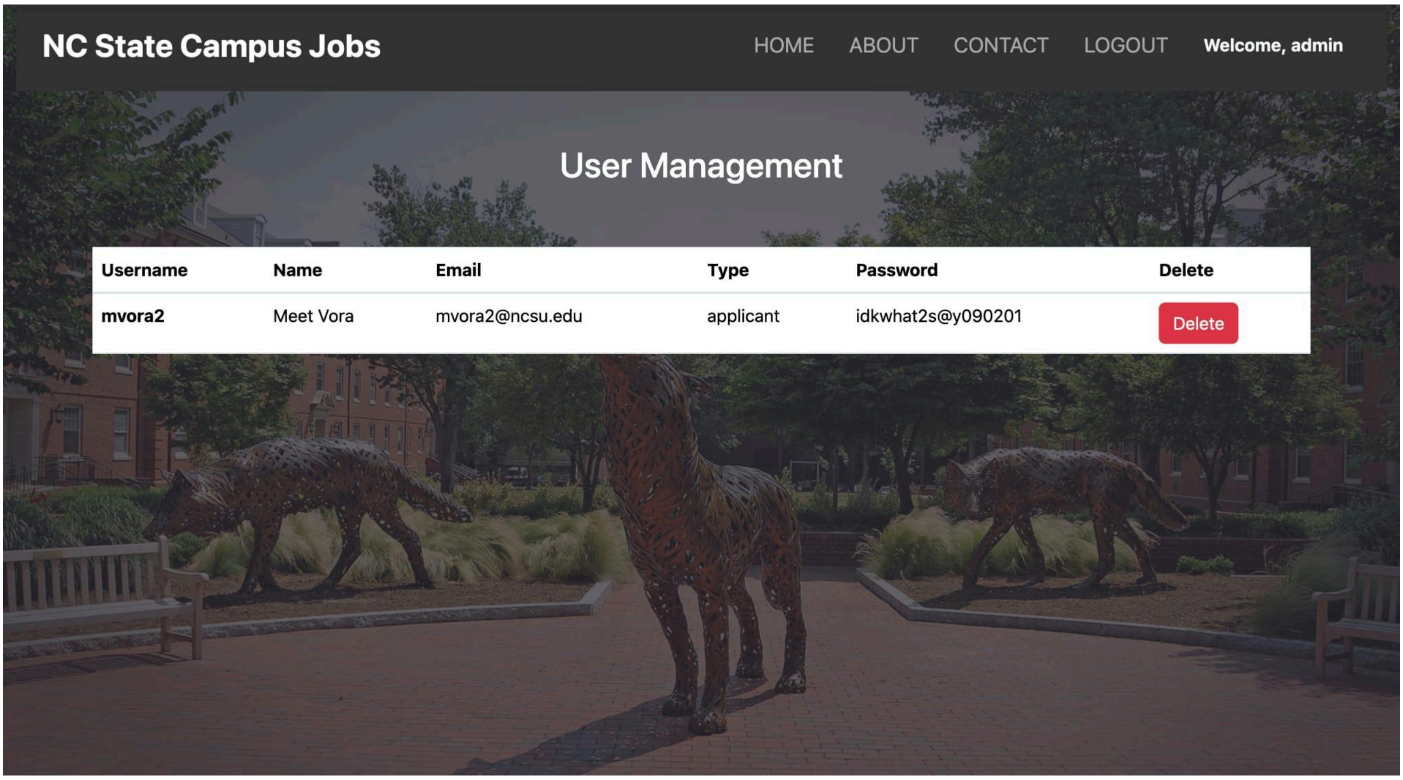
UX Designer
employer4
Seattle, WA
$85,000.00
Design user interfaces for mobile applications.

Already Applied!!

Frontend Developer
employer2
Los Angeles, CA
$95,000.00
Develop responsive web frontends using React.

Already Applied!!

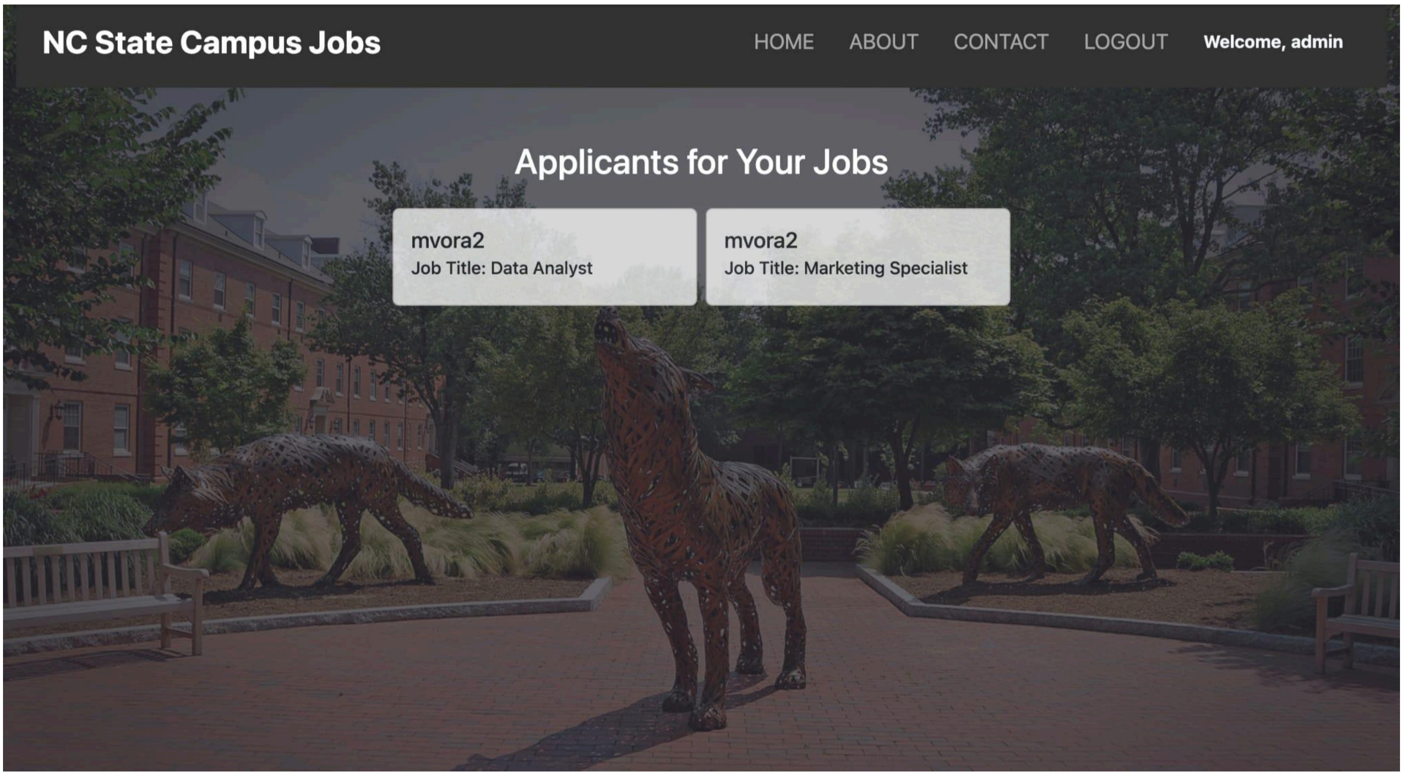
User Management Page



User Management

Username	Name	Email	Type	Password	Delete
mvora2	Meet Vora	mvora2@ncsu.edu	applicant	idkwhat2s@y090201	Delete

Applicants Page

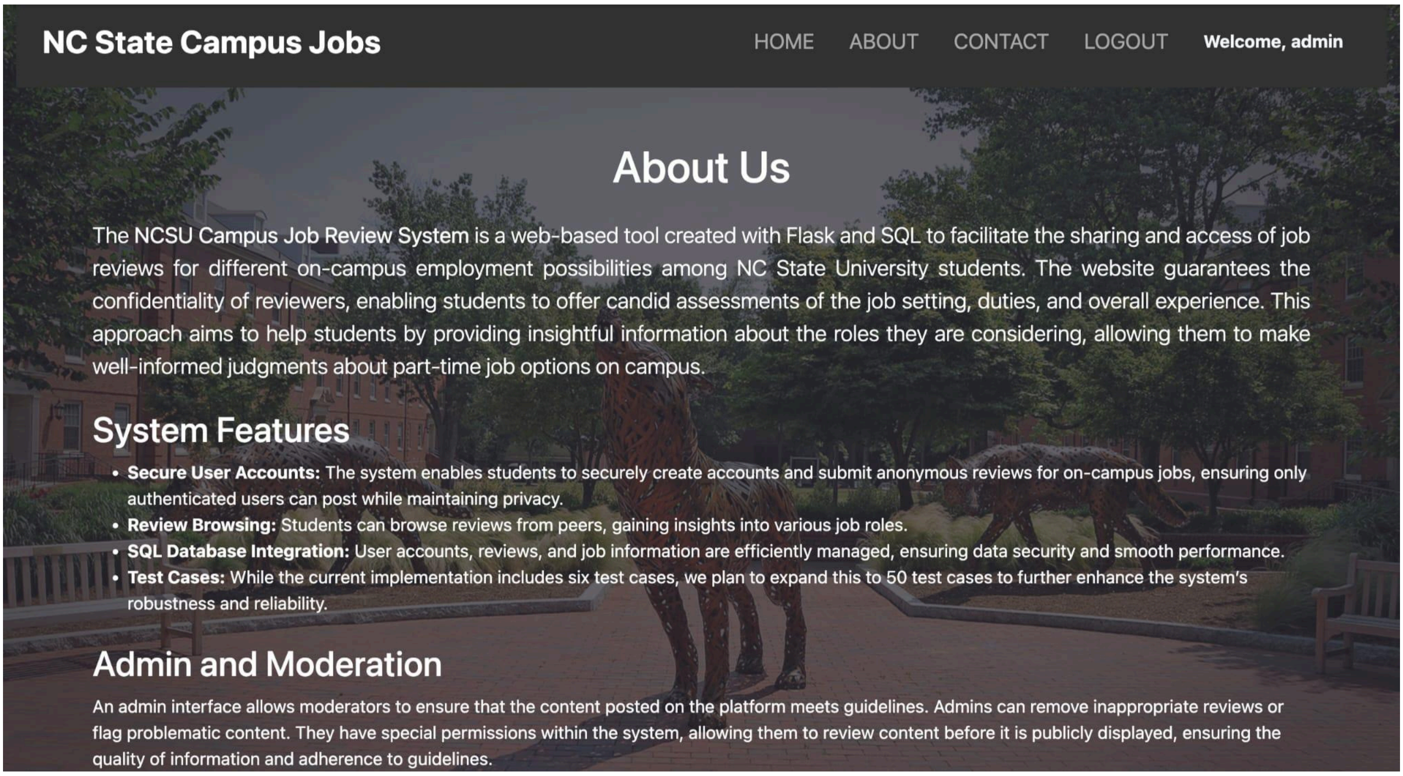


Applicants for Your Jobs

mvora2
Job Title: Data Analyst

mvora2
Job Title: Marketing Specialist

About us



About Us

The NCSU Campus Job Review System is a web-based tool created with Flask and SQL to facilitate the sharing and access of job reviews for different on-campus employment possibilities among NC State University students. The website guarantees the confidentiality of reviewers, enabling students to offer candid assessments of the job setting, duties, and overall experience. This approach aims to help students by providing insightful information about the roles they are considering, allowing them to make well-informed judgments about part-time job options on campus.

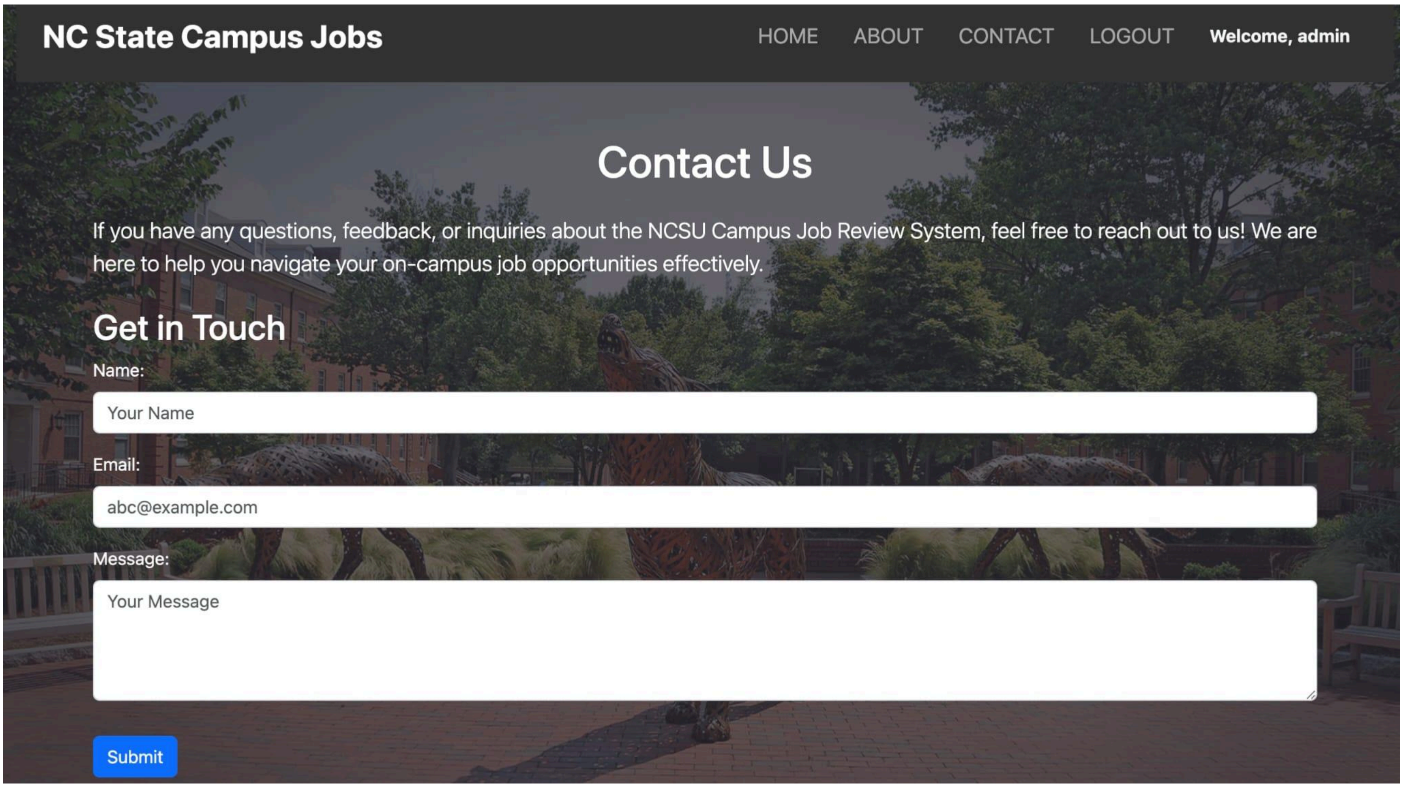
System Features

- **Secure User Accounts:** The system enables students to securely create accounts and submit anonymous reviews for on-campus jobs, ensuring only authenticated users can post while maintaining privacy.
- **Review Browsing:** Students can browse reviews from peers, gaining insights into various job roles.
- **SQL Database Integration:** User accounts, reviews, and job information are efficiently managed, ensuring data security and smooth performance.
- **Test Cases:** While the current implementation includes six test cases, we plan to expand this to 50 test cases to further enhance the system's robustness and reliability.

Admin and Moderation

An admin interface allows moderators to ensure that the content posted on the platform meets guidelines. Admins can remove inappropriate reviews or flag problematic content. They have special permissions within the system, allowing them to review content before it is publicly displayed, ensuring the quality of information and adherence to guidelines.

Contact us



Contact Us

If you have any questions, feedback, or inquiries about the NCSU Campus Job Review System, feel free to reach out to us! We are here to help you navigate your on-campus job opportunities effectively.

Get in Touch

Name:

Your Name

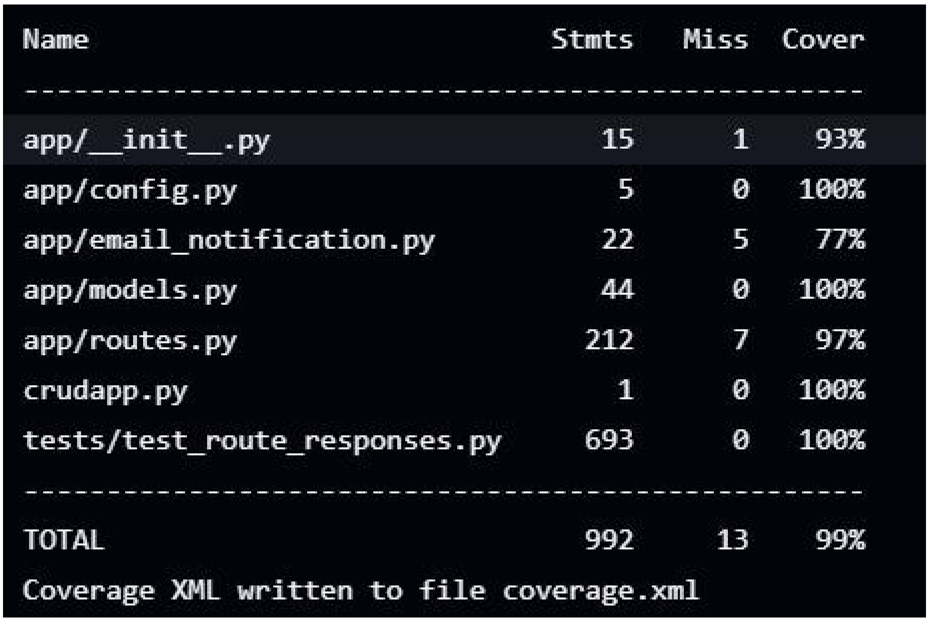
Email:

abc@example.com

Message:

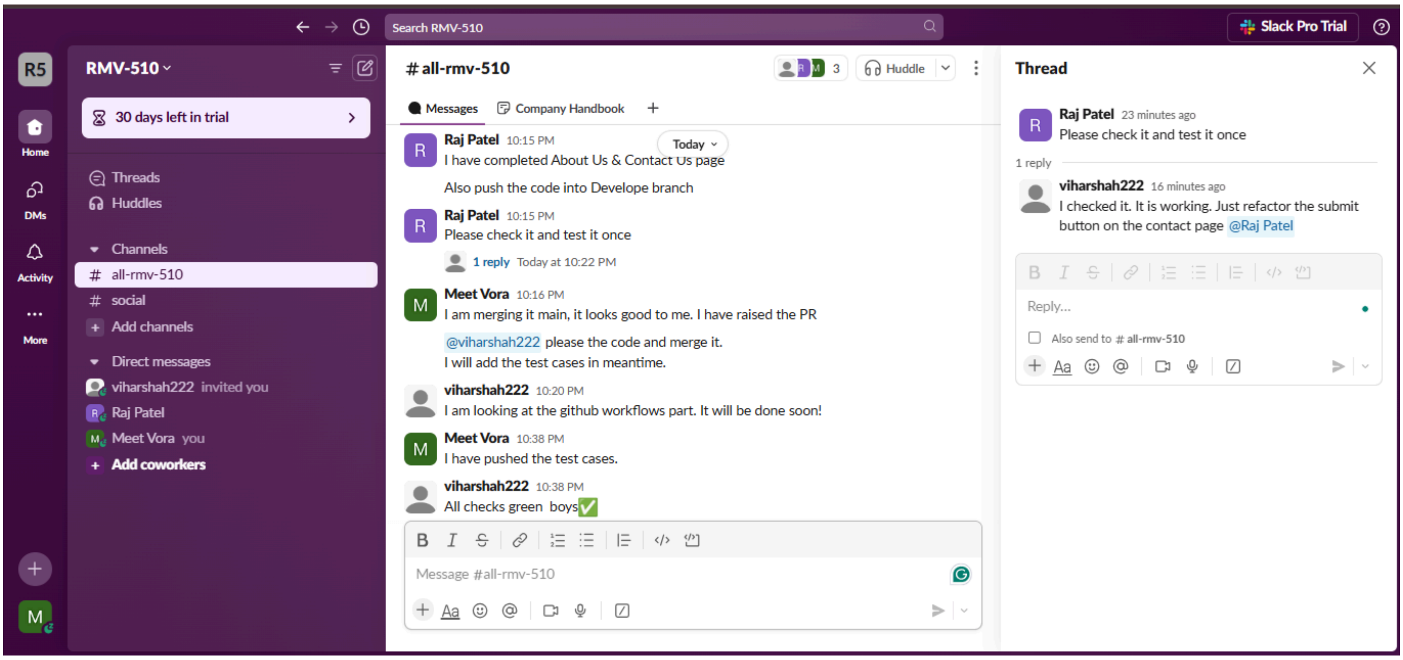
Your Message

code coverage snapshot



```
Name                             Stmts   Miss  Cover
----------------------------------------------------
app/__init__.py                     15      1    93%
app/config.py                        5      0   100%
app/email_notification.py           22      5    77%
app/models.py                       44      0   100%
app/routes.py                      212      7    97%
crudapp.py                           1      0   100%
tests/test_route_responses.py      693      0   100%
----------------------------------------------------
TOTAL                              992     13    99%
Coverage XML written to file coverage.xml
```

Slack communication channel



Raj Patel 10:15 PM
I have completed About Us & Contact Us page

Also push the code into Develope branch

Raj Patel 10:15 PM
Please check it and test it once
1 reply Today at 10:22 PM

Meet Vora 10:16 PM
I am merging it main, it looks good to me. I have raised the PR

@viharshah222 please the code and merge it.
I will add the test cases in meantime.

viharshah222 10:20 PM
I am looking at the github workflows part. It will be done soon!

Meet Vora 10:38 PM
I have pushed the test cases.

viharshah222 10:38 PM
All checks green boys✅

Thread

Raj Patel 23 minutes ago
Please check it and test it once

1 reply

viharshah222 16 minutes ago
I checked it. It is working. Just refactor the submit button on the contact page @Raj Patel

SOFTWARE EVALUATION METRICS

Questions	**SA**
	Total Score = 163
What is the name of your software?	**NCSU Campus Jobs Review System 2.0**
Q1 - What your software does	
Does your website and documentation provide a clear, high-level overview of your software?	**3**
Does your website and documentation clearly describe the type of user who should use your software?	**3**
Do you publish case studies to show how your software has been used by yourself and others?	**2**
Q2 - Your project's and software's identity	
Is the name of your project/software unique?	**2**
Is your project/software name free from trademark violations?	**3**
Q3 - Availability of your software	
Is your software available as a package that can be deployed without building it?	**3**

Is your software available for free?	**3**
Is your source code publicly available to download, either as a downloadable bundle or via access to a source code repository?	**3**
Is your software hosted in an established, third-party repository likeGitHub (https://github.com)	**3**
Q4 - Your software's documentation	
Is your documentation clearly available on your website or within your software?	**3**
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?	**2**
If you provide more extensive documentation, does this provide clear, step-by-step instructions on how to deploy and use your software?	**3**
Do you provide a comprehensive guide to all your software's commands, functions and options?	**2**
If your software can be used as a library, package or service by other software, do you provide comprehensive API documentation?	**1**
Do you store your documentation under revision control with your source code?	**2**
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**
Q5 - How you support your software	
Does your software describe how a user can get help with using your software?	**3**
Does your website and documentation describe what support, if any, you provide to users and developers?	**2**
Does your project have an e-mail address or forum that is solely for supporting users?	**3**
Are e-mails to your support e-mail address received by more than one person?	**3**
Does your project have a ticketing system to manage bug reports and feature requests?	**3**
Is your project's ticketing system publicly visible to your users, so they can view bug reports and feature requests?	**3**
Q6 - Your software's maintainability	
Is your software's architecture and design modular?	**3**
Does your software use an accepted coding standard or convention?	**2**
Q7 - Open standards and your software	
Does your software allow data to be imported and exported using open data formats?	**0**
Does your software allow communications using open communications protocols?	**3**

Q8 - Your software's portability	
Is your software cross-platform compatible?	**3**
Q9 - Your software and accessibility	
Does your software adhere to appropriate accessibility conventions or standards?	**3**
Does your documentation adhere to appropriate accessibility conventions or standards?	**3**
Q10 - How you manage your source code	
Is your source code stored in a repository under revision control?	**2**
Is each source code release a snapshot of the repository?	**3**
Are releases tagged in the repository?	**3**
Is there a branch of the repository that is always stable? (i.e. tests always pass, code always builds successfully)	**3**
Do you back-up your repository?	**2**
Q11 - Building and installing your software	
Do you provide publicly-available instructions for building your software from the source code?	**3**
Can you build, or package, your software using an automated tool?	**2**
Do you provide publicly-available instructions for deploying your software?	**3**
Does your documentation list all third-party dependencies?	**3**
Does your documentation list the version number for all third-party dependencies?	**3**
Does your software list the web address, and licences for all third-party dependencies and say whether the dependencies are mandatory or optional?	**1**
Can you download dependencies using a dependency management tool or package manager?	**3**
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**
Q12 - How you test your software	
Do you have an automated test suite for your software?	**3**
Do you have a framework to periodically (e.g. nightly) run your tests on the latest version of the source code?	**0**
Do you use continuous integration, automatically running tests whenever changes are made to your source code?	**3**
Are your test results publicly visible?	**1**
Are all manually-run tests documented?	**0**
Q13 - How you engage with your community	

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?	**0**
Does your website state how many projects and users are associated with your project?	**3**
Do you provide success stories on your website?	**0**
Do you list your important partners and collaborators on your website?	**3**
Do you list your project's publications on your website or link to a resource where these are available?	**3**
Do you list third-party publications that refer to your software on your website or link to a resource where these are available?	**3**
Can users subscribe to notifications to changes to your source code repository?	**3**
If your software is developed as an open source project (and, 0t just a project developing open source software), do you have a governance model?	**3**
Q14 - How you manage contributions	
Do you accept contributions (e.g. bug fixes, enhancements, documentation updates, tutorials) from people who are part of your project?	**3**
Do you have a contributions policy?	**3**
Is your contributions' policy publicly available?	**3**
Do contributors keep the copyright/IP of their contributions?	**2**
Q15 - Your software's copyright and licensing	
Does your website and documentation clearly state the copyright owners of your software and documentation?	**3**
Does each of your source code files include a copyright statement?	**3**
Does your website and documentation clearly state the licence of your software?	**3**
Is your software released under an open source licence?	**3**
Is your software released under an OSI-approved open-source licence?	**3**
Does each of your source code files include a licence header?	**0**
Do you have a recommended citation for your software?	**0**
Q16 - Your plans for the future	
Does your website or documentation include a project roadmap (a list of project and development milestones for the next 3, 6 and 12 months)?	**1**
Does your website or documentation describe how your project is funded, and the period over which funding is guaranteed?	**0**
Do you make timely annuncements of the deprecation of components, APIs, etc.?	**2**

RUBRICS	**SA** Total = 90
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
Number of commits: by different people	3
Issues reports: there are **many**	0
Issues are being closed	3
Docs: doco generated, format not ugly	3
Docs: what: point descriptions of each class/function (in isolation)	2
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	3
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
Test cases exist	3
Test cases are routinely executed	3
Issues are discussed before they are closed	3
Chat channel: exists	3
Test cases: a large proportion of the issues related to handling failing cases.	2
Evidence that the whole team is using the same tools: everyone can get to all tools and files	3
Evidence that the whole team is using the same tools (e.g. config files in the repo, updated by lots of different people)	3
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
Evidence that the members of the team are working across multiple places in the code base	3
Short release cycles	2
The file .gitignre lists what files should be be saved to the repo. See	3
The file INSTALL.md lists how to install the code	3
The file LICENSE.md lists rules of usage for this repo	3
The file CODE-OF-CONDUCT.md lists rules of behavior for this repo; e.g. see example	3
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
The file README.md contains all the following	3
Video	3
DOI badge: exists. To get a Digitial Object Indentifier, regiser the project at Zenodo.	3
Badges showing your style checkers	3
Badges showing your code formatters.	3
Badges showing your syntax checkers.	3
Badges showing your code coverage tools	3
Badges showing any other automated analysis tools	1