

CSC510 Software Engineering

Project 2: Extend Code

Group 3

Anchal Kakadia - anchal-kakadia,

Aniruddha Kulkarni - akulka404,

Aranya Venugopal - aranya-v

CSC 510, SOFTWARE ENGINEERING

TEAM 3- Aniruddha Kulkarni, Anchal Kakadia, Aranya Venugopal

J-TRACKER



ALL YOUR JOB APPLICATIONS, ONE POWERFUL DASHBOARD

POWERED BY-  python™  React   

A POWERHOUSE OF A PLATFORM, COMBINING THE POWERS OF A.I. AND SOFTWARE DEVELOPMENT!

ABOUT

J-TRACKER is an AI-powered career management platform that streamlines the job application process while enhancing user experience with intelligent recommendations. The latest updates introduce a robust dashboard with a past analyses section, summarizing AI-driven resume-to-job comparisons, and an application tracker categorizing submissions as Applied, Rejected, Wish List, or Waiting for Referral.

New AI integration enhances job recommendations, curating listings based on user skills, preferences, and experience instead of generic searches. The platform also introduces resume parsing and comparison, where AI extracts key skills and experiences, aligns them with industry standards, and generates an ideal job description for benchmarking. Users can now store and revisit past analyses for continuous improvement. Additional enhancements include persistent skill tracking, ensuring user preferences remain intact upon re-login, and a database-backed application tracker providing clear insights into application status. By leveraging AI for deeper analysis and automation, J-TRACKER minimizes user effort, optimizing their career journey with precision and efficiency.

WHAT'S NEW W/ J-TRACKER?

• Enhanced Search & AI Matching

AI-powered job recommendations replace traditional search methods, aligning job listings with user skills, preferences, and experience. The platform now leverages user profiles and resumes to provide personalized job recommendations, helping users find relevant opportunities faster.

• Bug Fixes & Performance Enhancements

Fixed the random sign-out bug, ensuring a stable user session. Improved system stability, efficiency, and performance, leading to a smoother job application process.

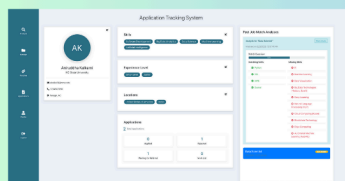
• Advanced Profile & Resume Management

Users can update and refine their professional profiles without losing previously added skills. Persistent skill tracking now ensures that user preferences remain saved after logging out, allowing for easy updates. Introduced AI-driven resume parsing and benchmarking, where AI extracts key skills, compares them against industry standards, and generates an ideal job description to help users evaluate their suitability for roles. User can upload a profile photo (code has been added but commented out), needs future update to view the profile photo.

• AI-Powered Insights & Career Recommendations

AI now suggests career paths based on users' skills, experience, and evolving job trends. A new past analyses section on the dashboard offers AI-driven resume-to-job comparisons, enabling users to store and revisit these insights for continuous improvement.

WHAT'S IN THE BOX? OUR ATTEMPT SUMMARIZED-



Snapshot 1- Application Management



Snapshots 2,3,4 - Showing an example of how a user can compare their experiences for a job using AI



THE FUTURE, VERSION I+1

WHAT'S NEXT FOR J-TRACKER?

- **Intelligent Job Application & Auto-Tracking** – AI suggests jobs based on profile strength and success rates. Auto-tracks applications via email parsing or LinkedIn.
- **Personalised Career Roadmap & Insights** – AI recommends skill-building paths, certifications, and real-time job market trends.
- **Smart Interview Prep** – AI-powered interview coach with personalized questions, mock simulations, and feedback.
- **Collaborative Job Search & Referrals** – Users connect for referrals, share job listings, and exchange interview tips.
- **AI Cover Letter Generator** – Auto-generates personalised cover letters with customisable tone and content.
- **Advanced Dashboard & Analytics** – Visualises job trends, career progress, and provides job alerts.
- **Profile Picture View** – Enable profile photo viewing (code exists but is commented out).

POWERING PRECISION: 70+ TEST CASES
FOR UNMATCHED RELIABILITY!



GitHub Repo



Video Demo





Repository URL: <https://github.com/ncsu-csc-510/Project2>

1. Fill Software Sustainability Evaluation self-assessment and Repository Rubric tables:





Software Sustainability Evaluation self-assessment table:




Category	Question	Yes	No	Evidence
Q1 - Software Overview				
Question 1.1	Does your website and documentation provide a clear, high-level overview of your software?	✓		Yes, README.md
Question 1.2	Does your website and documentation clearly describe the type of user who should use your software?	✓		Yes, README.md
Question 1.3	Do you publish case studies to show how your software has been used by yourself and others?	✓		Yes, we've uploaded pictures of the entire workflow in our README
Q2 - Identity				
Question 2.1	Is the name of your project/software unique?		✗	Many Projects are named this in our course





Question 2.2	Is your project/software name free from trademark violations?	✓		Yes, it is hosted under the MIT License
Q3 - Availability				
Question 3.1	Is your software available as a package that can be deployed without building it?		✗	No, it is a docker image that needs to be built
Question 3.2	Is your software available for free?	✓		Yes, it is a public github repository
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, it is a public github repository
Question 3.4	Is your software hosted in an established, third-party repository like GitHub , BitBucket , LaunchPad , or SourceForge ?	✓		Yes, on GitHub
Q4 - Documentation				
Question 4.1	Is your documentation clearly available on your website or within your software?	✓		Documentation is available on our organization's






				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?			Yes, in the readme file
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, all deployment information can be found in the readme
Question 4.4	Do you provide a comprehensive guide to all your software's commands, functions and options?			Yes, Command and functions can be found on the repository
Question 4.5	Do you provide troubleshooting information that describes the symptoms and step-by-step solutions for problems and error messages?			We have exception handling statements in the code, so any mishaps will throw an error message

Question 4.6	If your software can be used as a library, package or service by other software, do you provide comprehensive API documentation?		✗	No we do not have comprehensive API documentation
Question 4.7	Do you store your documentation under revision control with your source code?		✗	No, we do not have documentation under revision control our source code
Question 4.8	Do you publish your release history e.g. release date, version numbers, key features of each release etc. on your web site or in your documentation?	✓		
Q5 - Support				
Question 5.1	Does your software describe how a user can get help with using your software?	✓		Yes, it does. It has various functions that a user can use for an advantage in their job search journey.

Question 5.2	Does your website and documentation describe what support, if any, you provide to users and developers?			Yes, our code has exception handling coded into it, which will notify users of what is going wrong if needed
Question 5.3	Does your project have an e-mail address or forum that is solely for supporting users?			Not yet, but we're planning on implementing that
Question 5.4	Are emails to your support e-mail address received by more than one person?			-
Question 5.5	Does your project have a ticketing system to manage bug reports and feature requests?			The software uses a system similar to a ticketing system while interacting with the database while adding skills and experience to it.





Question 5.6	Is your project's ticketing system publicly visible to your users, so they can view bug reports and feature requests?			Not yet. Since its being used in the database it is currently private
Q6 - Maintainability				
Question 6.1	Is your software's architecture and design modular?			Yes, the frontend and backend are further divided into modules for easier debugging
Question 6.2	Does your software use an accepted coding standard or convention?			Yes, we've tried our level best to keep the size of the functions as small as possible
Q7 - Open Standards				

Question 7.1	Does your software allow data to be imported and exported using open data formats?			Yes, the data can be found on MongoDB atlas cloud platform
Question 7.2	Does your software allow communications using open communications protocols?			No, our software does not allow communications using open communication protocols
Q8 - Portability				
Question 8.1	Is your software cross-platform compatible?			It's an application run through local network
Q9 - Accessibility				
Question 9.1	Does your software adhere to appropriate accessibility conventions or standards?			Frondend shows use of AJAX and Bootstrap which helps with

				accessibility in applications
Question 9.2	Does your documentation adhere to appropriate accessibility conventions or standards?			It's easy to read
Q10 - Source Code Management				
Question 10.1	Is your source code stored in a repository under revision control?			No we do not have revision control but it is stored under a repository
Question 10.2	Is each source code release a snapshot of the repository?			No, each source code release is not a snapshot of the repository
Question 10.3	Are releases tagged in the repository?			No releases are not tagged in the repository
Question 10.4	Is there a branch of the repository that is always stable? (i.e. tests			Yes, The main branch is

	always pass, code always builds successfully)			a stable repository where all the tests pass and the code builds successfully
Question 10.5	Do you back-up your repository?	✓		There are many forks for this project.
Q11 - Building & Installing				
Question 11.1	Do you provide publicly-available instructions for building your software from the source code?	✓		Yes, all instructions to run the software are mentioned in the README
Question 11.2	Can you build, or package, your software using an automated tool?		✗	Yes, IaC tools can be used to automate this process.
Question 11.3	Do you provide publicly-available instructions for deploying your software?		✗	No, we don't. Our current implement

				ation is a local one
Question 11.4	Does your documentation list all third-party dependencies?	✓		Yes, it does.
Question 11.5	Does your documentation list the version number for all third-party dependencies?	✓		Yes, our README file lists versions for all dependencies
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, all the links are mentioned as hyperlinks
Question 11.7	Can you download dependencies using a dependency management tool or package manager?	✓		Yes, all the required dependencies can be installed using a Python package installer.
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, the actions tab in our repository mentions whether the test cases

				have passed or not
Q12 - Testing				
Question 12.1	Do you have an automated test suite for your software?			Yes, we have automated test suite for your software
Question 12.2	Do you have a framework to periodically (e.g. nightly) run your tests on the latest version of the source code?			No we have not set up a framework to periodically run the tests on the latest version of the code
Question 12.3	Do you use continuous integration, automatically running tests whenever changes are made to your source code?			Yes, we have automated running test cases for any changes made.
Question 12.4	Are your test results publicly visible?			Yes, we had a badge on the readme showcasi

				ng passing test cases.
Question 12.5	Are all manually-run tests documented?		✗	No we do not have manually run test
Q13 - Community Engagement				
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?		✗	No regular updates
Question 13.2	Does your website state how many projects and users are associated with your project?		✗	No explicit information on the users associated with the project
Question 13.3	Do you provide success stories on your website?		✗	No success stories are provided on the website.
Question 13.4	Do you list your important partners and collaborators on your website?		✗	No, there is no list of important

				contributo rs.
Question 13.5	Do you list your project's publications on your website or link to a resource where these are available?		✗	No publicatio ns of the project are available.
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?	✓		Yes, resources to mongoDB are linked
Question 13.7	Can users subscribe to notifications to changes to your source code repository?	✓		It is possible to watch this repo to get notificatio ns
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, there is no governan ce model added
Q14 - Contributions				
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, the project we're working on will eventually be worked on by

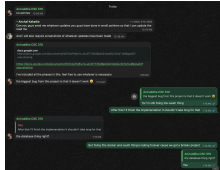
				other people
Question 14.2	Do you have a contributions policy?		✗	No, we don't. Not yet at least.
Question 14.3	Is your contributions' policy publicly available?	✓		Yes, all the contributors and their contributions are publicly available
Question 14.4	Do contributors keep the copyright/IP of their contributions?		✗	No, this is an open-source project being worked on by students.
Q15 - Licensing				
Question 15.1	Does your website and documentation clearly state the copyright owners of your software and documentation?		✗	No it is an open source project so anyone can make edits to it
Question 15.2	Does each of your source code files include a copyright statement?		✗	No there is copyrights

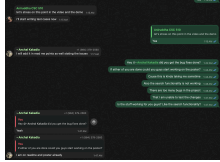
				in the project
Question 15.3	Does your website and documentation clearly state the licence of your software?	✓		Yes the MIT license is stated
Question 15.4	Is your software released under an open source licence?	✓		Yes it is released under the MIT license
Question 15.5	Is your software released under an OSI-approved open-source licence?	✓		MIT license used
Question 15.6	Does each of your source code files include a licence header?		✗	No there is no license header
Question 15.7	Do you have a recommended citation for your software?		✗	No citation for the software
Q16 - Future Plans				
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, it doesn't. The project poster however, contains suggestions that can be

				included in the next iteration of the project.
Question 16.2	Does your website or documentation describe how your project is funded, and the period over which funding is guaranteed?		✗	No, it doesn't.
Question 16.3	Do you make timely announcements of the deprecation of components, APIs, etc.?		✗	No, the project is passed onto the next team


Repository Rubric

Notes	Evidence	Score
Workload is spread over the whole team (one team member is often Xtimes more productive than the others... but nevertheless, here is track record that everyone is contributing a lot)	Evidence in GH	3
Number of commits	in GH	3
Number of commits: by different people	in GH	3
Issues reports: there are many	In GH	2
Issues are being closed	evidence in GH	3
Docs: doco generated, format not ugly	in GH	3

Docs: what: point descriptions of each class/function (in isolation)		3
Docs: how: for common use cases X,Y,Z mini-tutorials showing worked examples on how to do X,Y,Z	entries with images in the readme	3
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	In GH	3
Test cases exist	There are 70 test cases in the github page	3
Test cases are routinely executed	Automated test cases have been added which run with every push	3
Issues are discussed before they are closed	even if you discuss in slack, need a summary statement here	3
Chat channel: exists		3

		
Test cases: a large proportion of the issues related to handling failing cases.	There are a couple of failing test cases which can be worked on by future teams	2
Evidence that the whole team is using the same tools: everyone can get to all tools and files	Github is being used by the entire team along with the automated test cases. So the entire team is using the same tools and files for the code	3
Evidence that the whole team is using the same tools (e.g. config files in the repo, updated by lots of different people)	Can be seen in the GH commits	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)	Yes entire team is using the same tools, can be seen in github commits	3
Evidence that the members of the team are working across multiple places in the code base	Yes, different branches are present in github to showcase	3

	teammates working on different parts of the code	
Short release cycles	After every small implementation code is pushed for teammates to review	3
The file .gitignore lists what files should not be saved to the repo. See [examples]i(https://github.com/github/gitignore)	in GH	3
The file INSTALL.md lists how to install the code	in GH	3
The file LICENSE.md lists rules of usage for this repo	in GH	3
The file CODE-OF-CONDUCT.md lists rules of behavior for this repo; e.g. see example	in GH	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	in GH	3
The file README.md contains all the following		
Video	2min video of new functionality, showing a significant delta from prior.	3

DOI badge: exists. To get a Digital Object Identifier, register the project at Zenodo . DOI badges look like this: 	in GH	3
Badges showing your style checkers	config files in GH showing your config, badges in README	1
Badges showing your code formatters.	config files in GH showing your this formatter's config, badges in README	1
Badges showing your syntax checkers.	config files in GH showing this checker's config, badges in README	1
Badges showing your code coverage tools	config files in GH, badges in README	3
Badges showing any other Other automated analysis tools	config files in GH, badges in README	3
Total Score		90