J - Tracker

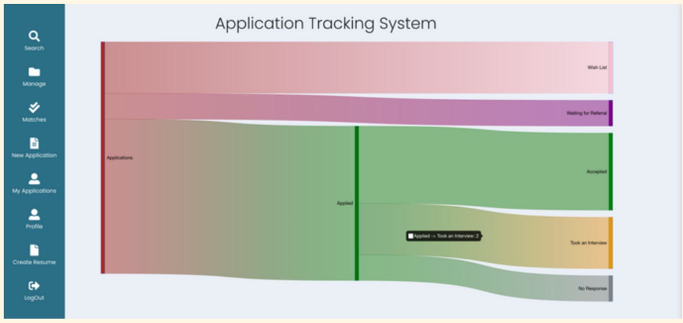
Your Application Tracking Partner

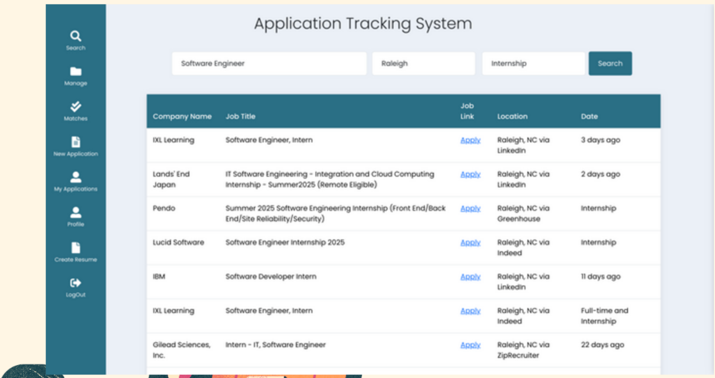
Our innovative Job Application Tracker— A seamless way to manage your job search journey

Features in J-Tracker?

- Search for the latest job openings for specific keywords and filter them based on Job Type and Preferred Location
- Automatic sankey graph generator
- Resume builder with multiple templates to choose from.
- Our application stores job application documents, such as resumes and cover letters
- Organize applications using a Kanban-style board with statuses like Wishlist, Waiting for Referral, Applied, and Rejected

Screen Snaps





Application Tracking System

Software Engineer | Raleigh | Internship | Search

Company Name	Job Title	Job Link	Location	Date
IXL Learning	Software Engineer, Intern	Apply	Raleigh, NC via LinkedIn	3 days ago
Lands' End Japan	IT Software Engineering - Integration and Cloud Computing Internship - Summer2025 (Remote Eligible)	Apply	Raleigh, NC via LinkedIn	2 days ago
Pendo	Summer 2025 Software Engineering Internship (Front End/Back End/Site Reliability/Security)	Apply	Raleigh, NC via Greenhouse	Internship
Lucid Software	Software Engineer Internship 2025	Apply	Raleigh, NC via Indeed	Internship
IBM	Software Developer Intern	Apply	Raleigh, NC via LinkedIn	11 days ago
IXL Learning	Software Engineer, Intern	Apply	Raleigh, NC via Indeed	Full-time and Internship
Gilead Sciences, Inc.	Intern - IT, Software Engineer	Apply	Raleigh, NC via ZipRecruiter	22 days ago

Demo Video **Live Demo** **Repository**

https://ats.cyril.pw

Credentials to try application

Username: se_demo
Password: fall2024

Future Enhancements

- LinkedIn integration for finding connections
- Resume sugggestions using AI
- Make application management easy by automatically filling the “Applied for Job” form.
- AI Agents for curating job postings and improving profile

Tools used

node JS
Flask
mongoDB
React

Group 55: Cyril Melvin Vincent, Bhushan Patil and Balaji Sankar
Repository Link: https://github.com/CSC-510-G55/project1-ats
Contact: cvincen@ncsu.edu, bpatil@ncsu.edu, bsankar2@ncsu.edu

Repository Link: https://github.com/CSC-510-G55/project1-ats

Refer GitHub Rubrics for evidence hyperlinks

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)	2	In GH
Number of commits	3	in GH
Number of commits: by different people	2	in GH
Issues reports: there are **many**	2	
Issues are being closed	3	evidence in GH
Docs: doco generated, format not ugly	3	in GH
Docs: what: point descriptions of each class/function (in isolation)	3	Docstrings
Docs: how: for common use cases X,Y,Z mini-tutorials showing worked examples on how to do X,Y,Z	2	doc page entries

Docs: why: docs tell a story, motivate the whole thing, deliver a punchline that makes you want to rush out and use the thing	1	
Docs: short video, animated, hosted on your repo. That convinces people why they want to work on your code.	3	Readme.md
Use of version control tools	3	GH
Test cases exist	3	test_app.py
Test cases are routinely executed	3	GH workflows
Issues are discussed before they are closed	3	GH
Chat channel: exists	3	GH Issues
Test cases: a large proportion of the issues related to handling failing cases.	3	
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)	2	
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	2	
Short release cycles	2	(hard to see in short projects) project members are committing often enough so that everyone can get your work
The file .gitignore lists what files should not be saved to the repo. See [examples]i(https://github.com/github/gitignore)	3	in GH
The file INSTALL.md lists how to install the code	3	in GH
The file LICENSE.md lists rules of usage for this repo	3	in GH

The file CODE-OF-CONDUCT.md lists rules of behavior for this repo; e.g. see example	3	in GH
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	2	in GH
The file README.md contains all the following	3	in GH
Video	3	Readme.md
DOI badge: exists. To get a Digitial Object Indentifier, regiser the project at Zenodo. DOI badges look like this:	3	Readme.md
Badges showing your style checkers	3	config files in GH showing your config, badges in README
Badges showing your code formatters.	3	config files in GH showing your this formatter's config, badges in README
Badges showing your syntax checkers.	3	config files iin GH showing this checker's config, badges in README

Badges showing your code coverage tools	3	config files in GH, badges in README
Badges showing any other Other automated analysis tools	3	config files in GH, badges in README
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	Demo
Question 1.3: Do you publish case studies to show how your software has been used by yourself and others?	No	
Question 2.1: Is the name of your project/software unique?	No	
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?	No	
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 likeGitHub (https://github.com), BitBucket (https://bitbucket.org),LaunchPad (https://launchpad.net) orSourceForge (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?	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?	Not applicable	
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?	No	
Question 5.4: Are e-mails to your support e-mail address received by more than one person?	Not applicable	
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?	Yes	
Question 7.2: Does your software allow communications using open communications protocols?	Yes	
Question 8.1: Is your software cross-platform compatible?	Yes	
Question 9.1: Does your software adhere to appropriate accessibility conventions or standards?	No	
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?	Yes	
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 package manager?	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?	No	
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?	Not applicable	
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?	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?	No	

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?	Not applicable	
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?	Not applicable	
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?	No	
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?	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)?	No	
Question 16.2: Does your website or documentation describe how your project is funded, and the period over which funding is guaranteed?	No	
Question 16.3: Do you make timely announcements of the deprecation of components, APIs, etc.?	No	