



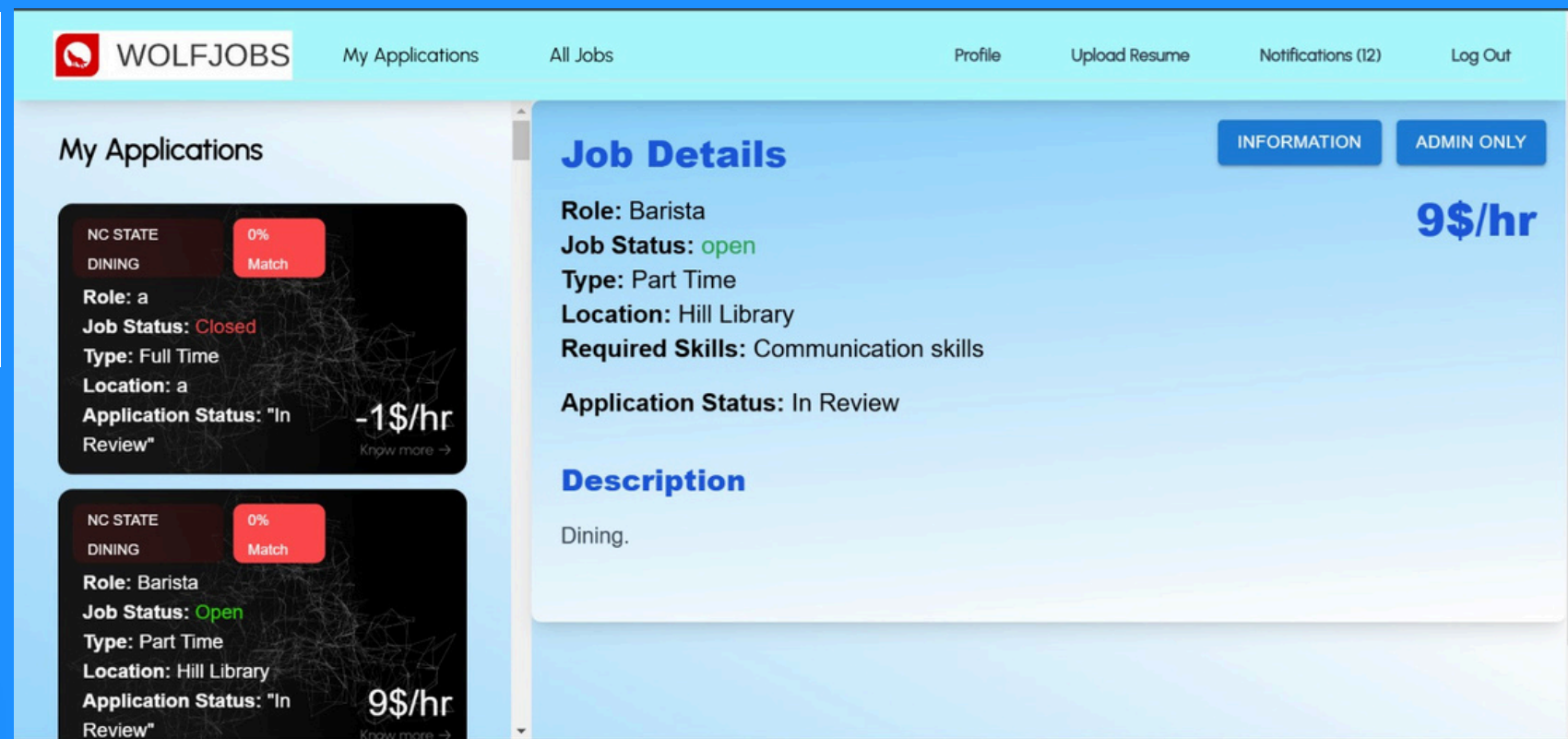
Group-67
Harsh Chaudhari
Yash Chaudhary
Yuvraj Singh Bhatia



Video

WolfJobs

Finding jobs made easy!



Introduction

- WolfJobs is a platform that connects students with part-time and full-time job opportunities.
- Offering personalized job matching, resume uploading, and easy application tracking, it simplifies the job search.
- With features like advanced job filtering and real-time notifications, WolfJobs ensures that students find roles aligned with their skills and career goals, helping them grow both personally and professionally.

User Workflow

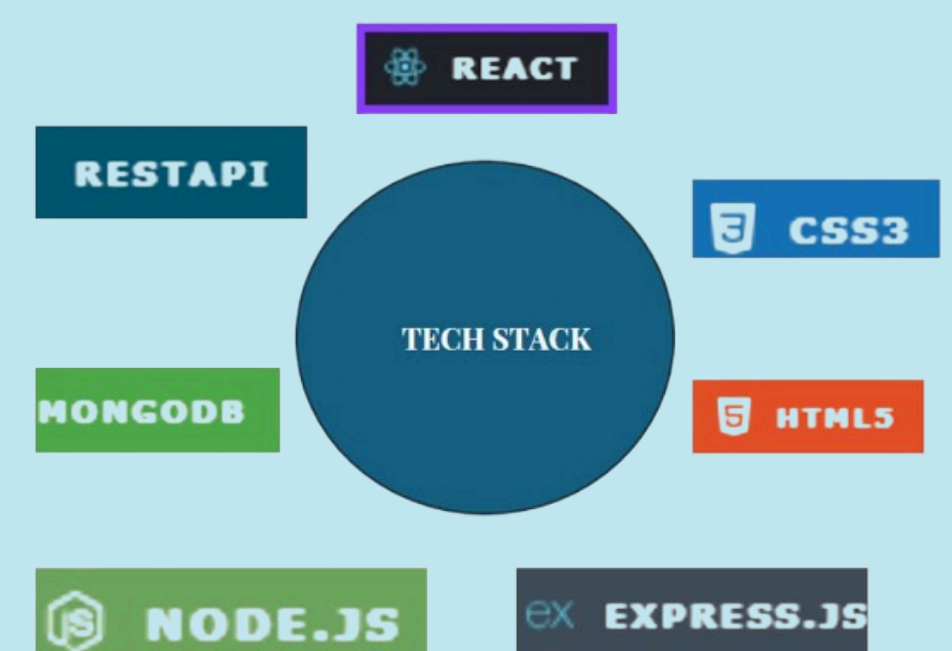
- Users can create an account or log in as a student or manager using their credentials.
- Students complete their profile by adding skills and uploading their resume.
- Managers create job postings and provide details like pay, city, and job type.
- Students browse job listings and apply to those that match their qualifications.
- Resumes are parsed and stored in MongoDB for manager review.
- Students track their application status in the portal and receive notifications for updates.
- Managers review, filter, and manage applications, accepting or rejecting candidates as needed.
- Both students and managers are notified of updates, such as application status changes or interview requests.

Future Enhancements

- Final Acceptance Email Notifications: Applicants will receive an email upon final acceptance, ensuring they are informed of their successful application status.
- Enhanced Admin Functionality: Expanded features for admin users will streamline management and improve oversight capabilities.
- UI Enhancements with Dark Mode: Introducing a dark mode option wherever possible and additional UI changes to enhance visual quality and user comfort.
- General Messaging Page: A new messaging feature will promote communication among users, fostering a more interactive community.
- System Stability and Speed Improvements: Enhancements will ensure faster loading times and greater stability, eliminating glitches for a smoother user experience.



Repository



Test case coverage

- There are total of 80 test cases and they're all passing perfectly.

What's new?

- Automated Email Notifications: Instant updates keep applicants informed of every status change in their application journey.
- Enhanced User Security: Passwords are securely hashed in MongoDB, ensuring stronger protection for user data.
- Job Match Percentage: Applicants can now see a "match percentage" that shows how well their skills align with job requirements.
- Visual and UI Enhancements: High-quality images and an intuitive, interactive design make the platform more engaging, professional, and user-friendly.
- University Admin Role: A new University Admin user type enables structured oversight and management of both students and hiring managers.
- WHY THIS PROJECT? "To simplify and enhance the job search experience for students, connecting them with meaningful career opportunities."

Project Rubric

Notes	Score	Evidence
SUM	90	
Video	3	https://drive.google.com/drive/u/2/folders/1Ys8NtxcmsuSlcsw4gpXddf
Workload is spread over the whole team (one team member is often Xtimes more productive than the others...	3	https://github.com/YHY-NCSU/wolfjobs_fall_24/pulse
but nevertheless, here is a track record that everyone is contributing a lot)	3	https://github.com/YHY-NCSU/wolfjobs_fall_24/pulse
Number of commits	3	https://github.com/YHY-NCSU/wolfjobs_fall_24/pulse
Number of commits: by different people	3	https://github.com/YHY-NCSU/wolfjobs_fall_24/pulse
Issues reports: there are many	3	https://github.com/YHY-NCSU/wolfjobs_fall_24/issues
Issues are being closed	3	https://github.com/YHY-NCSU/wolfjobs_fall_24/issues?q=is%3Aissue+is%3Aclosed
DOI badge: exists	3	https://github.com/YHY-NCSU/wolfjobs_fall_24/blob/feature/README
Docs: doco generated, format not ugly	3	https://github.com/YHY-NCSU/wolfjobs_fall_24/blob/feature/frontend

Notes	Score	Evidence
Docs: what: point descriptions of each class/function (in isolation)	3	https://github.com/YHY-NCSU/wolfjobs_fall_24/blob/feature/frontend
Docs: how: for common use cases X,Y,Z mini-tutorials showing worked examples on how to do X,Y,Z	3	https://github.com/YHY-NCSU/wolfjobs_fall_24/blob/feature/README
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	https://github.com/YHY-NCSU/wolfjobs_fall_24/blob/feature/README
Docs: short video, animated, hosted on your repo. That convinces people why they want to work on your code.	3	https://github.com/YHY-NCSU/wolfjobs_fall_24/blob/feature/README
Use of version control tools	3	https://github.com/YHY-NCSU/wolfjobs_fall_24/blob/feature/README
Use of style checkers	3	https://github.com/YHY-NCSU/wolfjobs_fall_24/blob/feature/README
Use of code formatters.	3	https://github.com/YHY-NCSU/wolfjobs_fall_24/blob/feature/README
Use of syntax checkers.	3	https://github.com/YHY-NCSU/wolfjobs_fall_24/blob/feature/README
Use of code coverage	3	https://github.com/YHY-NCSU/wolfjobs_fall_24/blob/feature/README

Notes	Score	Evidence
Other automated analysis tools	3	https://github.com/YHY-NCSU/wolfjobs_fall_24/blob/feature/README
Test cases exist	3	https://github.com/YHY-NCSU/wolfjobs_fall_24/tree/feature/frontend/
Test cases are routinely executed	3	https://github.com/YHY-NCSU/wolfjobs_fall_24/actions
The files CONTRIBUTING.md lists coding standards and lots of tips on how to extend the system without screwing things up	3	https://github.com/YHY-NCSU/wolfjobs_fall_24/blob/feature/CONTRIE
Issues are discussed before they are closed	3	https://github.com/YHY-NCSU/wolfjobs_fall_24/issues?q=is%3Aissue+is%3Aclosed
Chat channel: exists	3	Whatsapp, Teams
Test cases: a large proportion of the issues related to handling failing cases.	3	https://github.com/YHY-NCSU/wolfjobs_fall_24/issues?q=is%3Aissue+is%3Aclosed
Evidence that the whole team is using the same tools: everyone can get to all tools and files	3	https://github.com/YHY-NCSU/wolfjobs_fall_24/blob/feature/frontend/package.json
Evidence that the whole team is using the same tools (e.g. config files in the repo,	3	https://github.com/YHY-NCSU/wolfjobs_fall_24/blob/feature/.prettierr

Notes	Score	Evidence
updated by lots of different people)		
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	https://github.com/YHY-NCSU/wolfjobs_fall_24/blob/feature/.prettierr
Evidence that the members of the team are working across multiple places in the code base	3	

What your software does	
a) Does your website and documentation provide a clear, high-level overview of your software?	Yes
b) Does your website and documentation clearly describe the type of user who should use your software?	Yes
c) Do you publish case studies to show how your software has been used by yourself and others?	Yes
Your project's and software's identity	
a) Is the name of your project/software unique?	No
b) Is your project/software name free from trademark violations?	Yes
Availability of your software	
a) Is your software available as a package that can be deployed without building it?	Yes

What your software does	
b) Is your software available for free?	Yes
c) Is your source code publicly available to download, either as a downloadable bundle or via access to a source code repository?	Yes
d) Is your software hosted in an established, third-party repository like GitHub (https://github.com), BitBucket (https://bitbucket.org), LaunchPad (https://launchpad.net) or SourceForge (https://sourceforge.net)?	Yes
Your software's documentation	
a) Is your documentation clearly available on your website or within your software?	Yes
b) 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
c) If you provide more extensive documentation, does this provide clear, step-by-step instructions on how to deploy and use your software?	Yes
d) Do you provide a comprehensive guide to all your software's commands, functions and options?	Yes
e) Do you provide troubleshooting information that describes the symptoms and step-by-step solutions for problems and error messages?	Yes
f) If your software can be used as a library, package or service by other software, do you provide comprehensive API documentation?	Yes
g) Do you store your documentation under revision control with your source code?	Yes
h) 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
How you support your software	
a) Does your software describe how a user can get help with using your software?	Yes
b) Does your website and documentation describe what support, if any, you provide to users and developers?	Yes
c) Does your project have an e-mail address or forum that is solely for supporting users?	Yes
d) Are e-mails to your support e-mail address received by more than one person?	No
e) Does your project have a ticketing system to manage bug reports and feature requests?	Yes

What your software does	
f) Is your project's ticketing system publicly visible to your users, so they can view bug reports and feature requests?	Yes
Your software's maintainability	
a) Is your software's architecture and design modular?	Yes
b) Does your software use an accepted coding standard or convention?	Yes
Open standards and your software	
a) Does your software allow data to be imported and exported using open data formats? e.g. GIF, SVG, HTML, XML, tar, zip, CSV, JSON, NetCDF, or domain specific ones	Yes
b) Does your software allow communications using open communications protocols? e.g. HTTP, FTP, XMPP, SOAP over HTTP, or domain-specific ones	Yes
Your software's portability	
a) Is your software cross-platform compatible? e.g. does it run under two or more of Windows, Unix/Linux and Mac OS X, or can be used from within two or more of Internet Explorer, Chrome, Firefox and Safari?	Yes
Your software and accessibility	
a) Does your software adhere to appropriate accessibility conventions or standards?	Yes
b) Does your documentation adhere to appropriate accessibility conventions or standards?	Yes
How you manage your source code	
a) Is your source code stored in a repository under revision control?	Yes
b) Is each source code release a snapshot of the repository?	Yes
c) Are releases tagged in the repository?	Yes
d) Is there a branch of the repository that is always stable? (i.e. tests always pass, code always builds successfully)	Yes
e) Do you back-up your repository?	Yes
Building and installing your software	

What your software does	
a) Do you provide publicly-available instructions for building your software from the source code?	Yes
b) Can you build, or package, your software using an automated tool? e.g. Make (https://www.gnu.org/software/make/), ANT (http://ant.apache.org/), Maven (https://maven.apache.org/), CMake (https://cmake.org/), Python setuptools (https://pypi.python.org/pypi/setuptools), or R package tools (https://cran.r-project.org/doc/manuals/r-devel/R-exts.html)	Yes
c) Do you provide publicly-available instructions for deploying your software?	Yes
d) Does your documentation list all third-party dependencies?	Yes
e) Does your documentation list the version number for all third-party dependencies?	Yes
f) Does your software list the web address, and licences for all third-party dependencies and say whether the dependencies are mandatory or optional?	Yes
g) Can you download dependencies using a dependency management tool or package manager? e.g. Ivy (http://ant.apache.org/ivy/), Maven (https://maven.apache.org/), Python pip (https://pypi.python.org/pypi/pip) or setuptools (https://pypi.python.org/pypi/setuptools), PHP Composer (https://getcomposer.org/), Ruby gems (https://rubygems.org), or R PackRat (https://rstudio.github.io/packrat/)	Yes
h) 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
How you test your software	
a) Do you have an automated test suite for your software?	Yes
b) Do you have a framework to periodically (e.g. nightly) run your tests on the latest version of the source code?	Yes
c) Do you use continuous integration, automatically running tests whenever changes are made to your source code?	Yes
d) Are your test results publicly visible?	Yes
e) Are all manually-run tests documented?	Yes
How you engage with your community	

What your software does	
a) 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? e.g. release announcements, publications, workshops, conference presentations	No
b) Does your website state how many projects and users are associated with your project?	Yes
c) Do you provide success stories on your website?	Yes
d) Do you list your important partners and collaborators on your website?	Yes
e) Do you list your project's publications on your website or link to a resource where these are available?	No
f) Do you list third-party publications that refer to your software on your website or link to a resource where these are available?	
g) Can users subscribe to notifications to changes to your source code repository?	Yes
h) 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?	Yes
How you manage contributions	
a) Do you accept contributions (e.g. bug fixes, enhancements, documentation updates, tutorials) from people who are not part of your project?	Yes
b) Do you have a contributions policy?	Yes
c) Is your contributions' policy publicly available?	Yes
d) Do contributors keep the copyright/IP of their contributions?	Yes
Your software's copyright and licensing	
a) Does your website and documentation clearly state the copyright owners of your software and documentation?	Yes
b) Does each of your source code files include a copyright statement?	Yes
c) Does your website and documentation clearly state the licence of your software?	Yes
d) Is your software released under an open source licence?	Yes
e) Is your software released under an OSI-approved open-source licence?	Yes
f) Does each of your source code files include a licence header?	Yes

What your software does	
g) Do you have a recommended citation for your software?	Yes
Your plans for the future	
a) Does your website or documentation include a project roadmap (a list of project and development milestones for the next 3, 6 and 12 months)?	Yes
b) Does your website or documentation describe how your project is funded, and the period over which funding is guaranteed?	No
c) Do you make timely announcements of the deprecation of components, APIs, etc.?	Yes