Job-Applicator Plan

April 25, 2025

Contents

1 General

1.1 Project Overview

The base level premise of this project was to implement an autonomous one-stop shop for the entire recruitment cycle. The actual process personally for me took it into 3 major chunks all equally irritating and anxiety-inducing to every graduating college student looking for a job.

1. **Profile Manager:** A fully updated PR manager for you but one that is hyperactive in keeping all your information up to date across all platforms and aggregating information for individual experiences on resume. So it has an overall current updated understanding of each of your project/skills etc. and in case one of your platforms say a new resume version mentions some new details for an experience then update full knowledge base as well as update all platforms linkedin, indeed etc. everything with this new notch in your belt eliminating manual pain.

2. **Personalized Job Finder:** Just the very process of finding the applicable jobs to apply to like across various platforms but the caveat comes where there is a backed heuristic and whole lot of mambo jumbo optimizing each listing and ranking it hyperspecific to individual context.

3. **Applicator Slave:** No better name is accurate, this module is for the whole bruhaha. First creating the relevant account for whatever is the required job portal, personable to the company. Then retrieving the application questions. Another brings the company's basic information on values etc. Further check to answer application questions and optimize resume/CV or whatever as per the info of the job, the application questions, specific subset of my entire cohesive profile stored in Profile Manager finetuned keywords and documents as well as answers to fit the JD and company as much as possible. Then actually going through with applying

Dream: That applying for jobs is a thing of the past and as long as you are putting in the work for your field to become the best in your work then it won't be your job to reach people, network selling yourself, be subject to the 1000 rejections not optimized to understand you before getting that one sliver of hope somewhere. If you do your work with earnestness with the best possible effort the best job will automatically be found, applied for using your best representable skillset, and done without your knowledge. I mean you can see whats happening if you intend to and intervene but the idea is that one day you won't have to

1.2 Scope

The applications of this entire project are massive, eventually I hope this can be something that people from all professions can user. However since we have to start somewhere initially for the first MVP, we will be focusing on the entirety of this recruitment process for engineers graduating from undergraduate or graduate schools in USA. This subset of users are initially chosen since I belong to this category and can personally make sure if a designed module is serving my purpose directly, thereby ensuring it is useful.

1.3 Environment Setup

There is a single project folder where all the project files will reside and this is to be considered the root for the project called **job-applicator**. All files/folders regarding this entire implementation reside in this folder.

1.3.1 Local Development Environment

The following details are about the local development environment, setup to program this project

- **System:** Macbook Pro M1 Max
- **IDE:** Visual Studio Code
- **Environment:** Conda environment called *jobenv*
- **Python:** Version >3.11

2 Profile Manager

The Profile Manager is a key component of the job-applicator system responsible for unified management of the user's professional profile across multiple platforms.

NOTE: This is the section you are hand-waving the most like clearly layout the most base requirements of whatever you can think of in this section. Start with the actual information points you have to extract from the various sources

2.1 Information Points

These are diverse points of data from various professional sources and documents listed out below, that highlight very specific information about the user that is required for their professional

profile. There will be a lot of overlap in the information coming in from various sources about same descriptor and the challenge is keeping this up to date.

Each subsubsection represents a key area currently planned to become an individual part necessary to form a cohesive user profile. For each subsubsection, the specific extractable data fields are presented in the table format shown below. In these tables, required fields are indicated by an asterisk (*) after the `Variable Name`, and the entire row for that variable is displayed in bold.

Variable Name	Datatype	Description	Example

2.1.1 Education

Education encompasses the formal academic background of the user.

`ed_lvl*`	**enum**	**Ed. Lvl: HS, Assoc, UG, PG, PhD, PD**	**UG**
`ed_org*`	**str**	**Name of educational institution**	**Drexel University**
`ed_degree*`	**str**	**Degree name**	**Bachelor of Science**
`ed_startdate*`	**date**	**MM/YY started**	**08/20**
`ed_enddate*`	**date**	**MM/YY ending**	**05/25**
`ed_status*`	**enum**	**Status: Complete, Ongoing, Drop**	**Complete**
`ed_majors*`	**list**	**List of majors, 1 required**	**Computer Science**
`ed_minors`	str	List of minors	Artificial Intelligence
`ed_location`	str	Parsable location	Philadelphia, PA
`ed_gpa`	float	Grade Point Average listed	3.85

Resumé usually has this with all the above requirements in it as required, different templates can obviously make this harder to distinguish and extract

Linkedin on the other hand might store this section in more detail making it so that we have to corroborate across sources

CV will store the bare minimum representation of the same

2.1.2 Work Experience

Work experience encompasses professional roles that contribute to a user's career progression.

`exp_org*`	**str**	**Company Name**	**Google Inc**
`exp_role*`	**str**	**Job role**	**CTO and Founder**
`exp_startdate*`	**date**	**MM/YY started**	**06/22**
`exp_enddate*`	**date**	**MM/YY ending or "Present"**	**08/23**
`exp_location*`	**str**	**Parsable location**	**New Delhi, India**
`exp_type*`	**enum**	**Type: full-time, intern, research, freelance**	**intern**
`exp_desc*`	**str**	**Full description typed for experience**	**Led ERP product development**
`exp_skills*`	**2.1.4 Skill**	**Skills associated with the experience**	**Skill object in 2.1.4**
`exp_tech`	list	Technologies used	Laravel, Angular, MySQL, AWS
`exp_action_words`	list	Cross-checked with available list to judge resume	Implemented, Enhanced
`exp_modality`	enum	Remote, IRL, Hybrid	Remote
`exp_industry`	str	Industry sector	SaaS, ERP Software

Resumé: Primary source for structured experience (Org, Role, Dates, Location, Responsibilities/Achievements). Detail varies significantly by format.
Linkedin: Often more comprehensive than resumés, with expanded descriptions, linked skills, and longer timelines. Useful for corroboration and details omitted for brevity elsewhere.
Cover Letter: Highlights relevant experiences, providing narrative context or elaborating on key achievements, but usually not a full history.

2.1.3 Key Projects

Key projects encompass significant technical or research endeavors that demonstrate the user's capabilities and expertise.

`proj_name*`	**str**	**Project title**	**Spreadsheet 2D Context Model**
`proj_startdate*`	**date**	**MM/YY started**	**08/24**
`proj_enddate*`	**date**	**MM/YY ending or "Present"**	**Present**
`proj_desc*`	**str**	**Full description of the project**	**Developed a Transformer-based model optimized for spreadsheet data analysis**
`proj_skills*`	**2.1.4 Skill**	**Skills associated with the project**	**Skill object in 2.1.4**
`proj_tech`	list	Technologies/tools used	PyTorch, Hugging Face, Pandas
`proj_metrics`	list	Quantifiable or qualatative achievements	F1 score: 0.99, Produced accurate summary
`proj_url`	str	Link to project repository or demo	github.com/username/project
`proj_team`	list	Team members involved	Nakul, Self
`proj_role`	str	Role in the project	Lead Researcher
`proj_type`	enum	Type: research, coursework, personal	research

Resumé: Often lists key projects with concise titles, descriptions, technologies used, and potentially a URL (`proj_url`).

Linkedin: Has a dedicated section to list projects, allowing names, descriptions, associated roles/education, team members, skills, and media/URLs.

Cover Letter: Selectively highlights specific projects, providing narrative context on their relevance or impact.

Github: Serves as the primary repository for code-based projects, providing detailed READMEs, code, commit history, and validation. Often the target of `proj_url` from other sources.

2.1.4 Skills and Knowledge Areas

Skills and knowledge areas represent the user's technical abilities, domain expertise, and specialized knowledge. Also contain soft skills

`skill_category*`	**str**	**General skill category**	**Artificial Intelligence & Machine Learning**
`skill_items*`	**list**	**Specific skills within the category**	**Supervised learning, Deep learning, Transformers**
`skill_type*`	**enum**	**Soft, Hard/Technical**	**Hard**
`skill_lvl`	enum	Internal algo	Some score
`skill_exp`	int	Ceiling yrs	4
`skill_spread`	Dict	Projects/roles demonstrating skill	Spreadsheet 2D Context Model

Resumé: Often lists hard skills in a dedicated section or implicitly within experience/project descriptions.
Linkedin: Features a 'Skills' section where connections can provide endorsements for validation. Skills (hard & soft) are also inferred from text content.
Cover Letter: Narratively highlights specific hard and soft skills tailored to the target application.
Github: Demonstrates technical skills via code (languages, libraries), repository topics, and READMEs. Contribution activity implies proficiency/collaboration.

2.1.5 Professional Certifications

Professional certifications represent formal credentials and qualifications earned by the user.

`cert_name*`	**str**	**Full certification name**	**AWS Certified AI Practitioner**
`cert_issuer*`	**str**	**Issuing organization**	**Amazon Web Services**
`cert_date*`	**date**	**MM/YY issued**	**03/24**
`cert_id`	str	Certification ID or reference number	AIF-C01-12345
`cert_expiry`	date	MM/YY expires (if applicable)	03/27
`cert_url`	str	Verification link	verify.aws.amazon.com/12345
`cert_skills`	2.1.4 Skills	Skills type object validated by certification	AWS, AI, Machine Learning
`cert_lvl`	str	Level of certification (if applicable)	Associate

Resumé: Often includes a section listing certifications, potentially with issuing organization and date/ID.

Linkedin: Features a dedicated 'Licenses & Certifications' section for detailed entries including name, issuer, dates, and credential URLs/IDs.

Cover Letter: May highlight highly relevant certifications narratively to support qualifications for a specific role.

2.1.6 Related Coursework

Related coursework encompasses academic courses relevant to the user's professional field and expertise.

`course_name*`	**str**	**Course title**	**Natural Language Processing**
`course_org*`	**str**	**Institution offering the course**	**Drexel University**
`course_category*`	**str**	**The general category it falls under we use to classify skills etc.**	**NLP**
`course_code`	str	Course code or identifier	CS-584
`course_date`	date	MM/YY completed	12/23
`course_grade`	str	Grade received	A
`course_desc`	str	Brief description of course content	Advanced techniques in natural language understanding and generation
`course_skills`	2.1.4 Skills	Skills/knowledge acquired	Transformers, Tokenization, Embeddings
`course_projects`	2.1.3 Key Projects	Projects completed during course	Abstractive Summarizer with BART

Resumé: Frequently lists relevant course titles under education or a dedicated coursework section.
Linkedin: Courses can be added under the 'Accomplishments' section or associated with specific Education entries.
Cover Letter: Unlikely to list specific courses; might reference a general field of study relevant to the application.

2.1.7 Publications

Publications represent scholarly articles, research papers, or other formal written works authored by the user.

`pub_title*`	**str**	**Publication title**	**Adversarial Abstractive Summarization with BART**
`pub_date*`	**date**	**MM/YY published**	**06/24**
`pub_authors*`	**list**	**List of authors**	**Mirza, F., Williams, J.**
`pub_venue`	str	Journal/conference/publisher	International Conference on NLP
`pub_doi`	str	Digital Object Identifier	10.1109/ICNLP.2024.12345
`pub_url`	str	Link to publication	icnlp.org/proceedings/2024/12345
`pub_citations`	int	Number of citations	7
`pub_abstract`	str	Brief abstract of publication	Novel approach to abstractive summarization using adversarial learning
`pub_keywords`	list	Publication keywords	NLP, Summarization, GAN, BART
`pub_type`	enum	Type: Journal, Conference, Workshop, Preprint	Conference
`pub_status`	enum	Status: Published, In Press, Under Review, In Preparation	Published

Resumé: Lists publications (title, authors, venue, date), potentially with URL/DOI, in a dedicated section or linked from projects/education.

Linkedin: Has a dedicated 'Publications' section for detailed entries including title, publisher, date, authors, URL, and description.

Cover Letter: May highlight significant or highly relevant publications narratively to demonstrate expertise.

Github: Can host associated code, datasets, or supplementary materials, often linked via the publication's URL (`pub_url`) or a related project entry.

2.2 Functionality

2.2.1 Information Corroboration

The main challenge as mentioned before is that the same piece of information distinguishable to a human can be represented differently across the various data sources. For example a certain education entry can have its GPA missing from the Resume but listed on Linkedin therefore it is important that one comprehensive and correct description of the same information point is created from the multiple **2.1 Info Points**.

3 Job Finder

The Job Finder component is the core job discovery engine of the job-applicator system, responsible for identifying relevant job opportunities across various platforms and filtering them according to user preferences.

3.1 Planned Portals

Since jobs can be found using a myriad of ways, it was important to first focus on certain areas specifically, to achieve rudimentary success in getting results from them before moving forward. There are 2 different types of job searching plaforms. The first type is the initial category we will begin with where we will find jobs via popular pre-determined platforms that remain largely static in their UI and can therefore be accessed in a more hard-coded manner. Essentially accessing and operating on these types of platforms should be fairly straightforward where the UI/UX of the website itself doesnt change therefore the agents have no surprises and we know what workflow to expect when searching through these platforms. The portals planned for this first category currently are:

1. **Linkedin:** Most popular professional platform customized to each user, this would entail logging in to a user's Linkedin and then updating search preferences to look for jobs in its individual sections it shows inside the jobs section too
2. **Handshake:** Setup for most university level students, this platform was recommended by Drexel therefore I am assuming it is a popular choice for college level students
3. **Google Search:** Generic search using various optimized queries to directly lead to available jobs on the basic google search

The second type is dynamic. The idea here is that the user would be able to provide individual links of various job search platforms specific to their usecase. For example the official company website for Microsoft Careers. The planned functionality then expected is:

1. The job searching agent will then dynamically be created to visit this website
2. It should apply the filters specific to the user for job filtering
3. It should be able to understand the produced list of jobs
4. Extract jobs specific to ones relevant to user

This is challenging since we cannot control what URLs the user will provide to serve as websites to browse for jobs therefore determining if they are valid job search webpages then adapting to such diverse UIs and navigating through them is a complicated task

4 Job Applicator

The Job Applicator component is responsible for evaluating job opportunities and automating the application process, functioning as the execution arm of the system that interacts with job application interfaces.
It will essentially be responsible for retreiving and filling all the application requirements for a single job, then also fill and submit the application after user approval or automatically

5 Failed Approaches

Throughout the development of the job-applicator system, several approaches were explored, implemented, and ultimately abandoned or significantly refactored. An analysis of the repository dynamics and commit history provides clear insights into these evolutionary paths and architectural pivots.

6 Linear Progress

The job-applicator project has evolved through several clearly defined development phases, as evidenced by the repository's commit history, issue tracking, and branch management. This section outlines the chronological progression of the project from concept to current implementation, informed by actual development activities tracked in the GitHub repository.

7 Rough

7.1 Inkling of Requirement

7.1.1 General Storage Pool / Categories

This can be anytype of architecture, the idea is something like encountering skills and constantly updating overall system keywords there needs to be a general pool of where the application has access to all this learned data.

Now we are seeing categories like 'AI and ML' emerge for clubbing skills together as per 2.1.4 but also clubbing 2.1.6 coursework and 2.1.5 also same can be used

7.2 Haywire Functionalities

7.2.1 Auto-Linkedin Skill bolster (2.1.4)

From the listed skills check what projects/tools verify it 100% to maintain our cohesive profile but also we can bolster it by perhaps having an advance bot that gets this **Linkedin verified**? Or some other way to passively bolster them on profile

7.2.2 Advanced Analytics on Required 'Skills' scorer (2.1.4)

Related to search more than profile right now, keep comprehensive info of the most popular skills we are hitting on the job profiles we search. On the flip side start aggregating a general set of skills we are encountering the most and an advanced analysis of what we sorely lack. This might be also that we might have that skill to show but just not on our profile and this will streamline that.

7.2.3 Skill Proficiency Algorithm

We can design based on heurstic on how much user has used here and there and what the level of complexity it is we are seeing of usage, plus maybe some fun quiz a smart agent auto generates to calculate proficiency for skill_lvl

8 Appendices

8.1 Professional Documents

8.1.1 Resumes