Building Scalable Software: A Work Trial at Paraform

CS4991 Capstone Report, 2025

Jayanth Peetla
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
ynv7jk@virginia.edu

ABSTRACT
Paraform, an early-stage startup based in the Bay Area, was struggling to identify contact information for company executives as part of their cold email outreach campaign. To accelerate their cold outreach, I created a web scraper that finds a company's top 4-7 executives along with their LinkedIn given the company domain as input. The web scraper relied on SERP API to extract relevant websites and AI LLM agents to semantically analyze webpage content. The tool was created using Node.js and deployed through Render. To enhance user interaction, I created a front-end through Next.js and launched using Vercel. Integrated into the top of a three-step waterfall, the final API I developed replaced Paraform's existing lead generation method, generating over 3,000 leads weekly. This exponentially increased the spread of Paraform and heavily contributed to its ongoing business growth. Further optimization would focus on eliminating false leads mistakenly collected by AI LLM agents and scaling the backend infrastructure to store more data.

1. INTRODUCTION

Recruiting new employees is often a daunting task for a company to take on. There are many factors to consider from a candidate's technical expertise to whether he or she would be a good fit on the team. To address this issue, most companies at scale hire a dedicated Human Resources (HR) team that focuses primarily on hiring and onboarding new employes. However, smaller companies and startups are left to waste precious time and resources on scouting for new employees.

Paraform is an innovative marketplace designed to streamline the recruitment process for small companies and startups that lack dedicated HR teams. The platform connects these businesses with freelance recruiters, allowing companies to post job openings while recruiters leverage their personal networks to identify potential candidates. When a successful match is made, recruiters earn a commission, making it a mutually beneficial ecosystem for all parties involved. During my time with Paraform, I was a member of the growth team, which was tasked with expanding the company's client base. This team focused on a cold outreach strategy, targeting a vast array of startups and small businesses by reaching out directly to their top executives via email.

Despite having access to public people search APIs such as Apollo and Crunchbase, the growth team faced significant challenges in automating the identification of a company's leaders. The available data was often incomplete or inaccurate, hampering the efficiency of their outreach efforts. To overcome this, prior to my arrival, the growth team was hiring part-time, remote workers in Indonesia and Singapore to manually discover these leads overnight, making use of the time

zone discrepancies. Recognizing the inefficiency, I developed a solution to streamline the process: a web scraper that could accurately extract the top executives' contact information and LinkedIn profiles. This innovative tool not only replaced the unreliable manual methods but also significantly enhanced our lead generation, directly contributing to Paraform's ongoing growth and success.

2. RELATED WORKS

The primary, initial source of inspiration for this project was an internal web scraper developed by a fellow Paraform employee. This script was originally designed to scan a company's website for available job postings. This tool demonstrated effective techniques for handling asynchronous operations in TypeScript, generating clear, user-friendly logging information to track the script's progress, and tips to producing professional, understandable code. By studying this implementation, I was able to adapt and extend its core principles to meet the challenges of our lead generation process. (Ram, 2024).

Another crucial source that significantly informed my work was the concept of Retrieval-Augmented Generation (RAG), as outlined by Lewis, et al. (2020). In scenarios where a website's content is far too extensive to serve directly as input for a LLM, the RAG approach provides an effective method for identifying relevant information. The core idea behind RAG is to retrieve only the most relevant text segments from a larger document and then use these concise passages as context for the LLM.

Using this technique, I implemented a keyword-driven mechanism in my script to isolate segments of text that contained keywords such as "CEO," "CTO," etc. By scanning the website's content for these keywords, my solution could extract the specific text chunks most likely to contain valuable lead information, which were then passed as context to the LLM (Lewis, et al., 2020).

3. PROJECT DESIGN

[Include a brief (one-sentence?) opening here to introduce the section and provide context for your readers.]

3.1 Process Design

The lead generation tool was designed as a modular system, with each component addressing a specific part of the overall workflow. At its core, the system is implemented in the crawler.ts module, where the primary function initiates the scraping process for a given company domain. This process orchestrates multiple steps that begin with issuing various SERP API queries to retrieve relevant URLs. SERP API is a service that programmatically retrieves search engine results, allowing the system to access curated data based on specific search queries. For example, a typical query might be "Microsoft C-suite," which helps identify pages likely to contain information about top executives.

Once the URLs are collected, the system proceeds to extract and process web content using a dedicated content extraction module. In this stage, the extracted text is cleaned and refined by removing stop words and other irrelevant elements. A RAG-inspired strategy is then applied to pinpoint the most contextually relevant text segments—especially those containing keywords like "CEO" or "CTO." By isolating these key segments, the system ensures that only the most useful content is passed on for semantic analysis, ultimately leading to accurate identification of executive leads.

After isolating the key segments of webpage content, the most relevant information is passed to the ChatGPT large language model

using advanced prompt engineering. This carefully crafted prompt instructs the model to identify and extract details about individuals holding executive roles—specifically targeting keywords such as "CEO," "CTO," and other C-suite titles. In scenarios where this advanced semantic analysis does not yield sufficient or accurate results, the system is designed to fall back on additional resources, including the CRUST API and other people search APIs. This ensures that all input company domains have at least some outputs for the growth team to rely on.

In addition to the backend functionality, I developed a user interface specifically designed for the growth team, who did not have a technical background. The front end accepts an input CSV file containing a column of target company domains, enabling users to easily initiate the lead generation process. Once the process is complete, the tool outputs a CSV file that consolidates each domain with the corresponding executives and their LinkedIn profiles, making it straightforward for non-technical users to access and use the data.

Deploying the tool presented unique challenges, particularly due to the platform constraints of Vercel, which the team was already using for their dashboard. While the web scraper was deployed on Render, attempting to run the entire process directly on Vercel was not feasible due to its 1-minute maximum script runtime—whereas the complete scraping process could take between 4 to 7 minutes for a few hundred companies. To overcome this, I implemented a two-pronged deployment strategy: a Vercel API was created to accept the input CSV file and orchestrate the process by triggering individual script calls to Render for each company. Additionally, a retriever Vercel API was developed to collect the outputs from Render and consolidate them into a single CSV file, which is then returned to the user. This hybrid approach effectively leverages the strengths of both platforms while accommodating the runtime limitations of Vercel.

4. RESULTS

The implementation of this process design has led to a significant enhancement in lead generation capabilities for Paraform. By integrating multiple sources of data and employing advanced text extraction and processing techniques, the tool has successfully replaced older, less reliable methods. In practice, the system generates over 3,000 executive leads per week, demonstrating a substantial improvement in both the quantity and quality of leads compared to previous approaches. This surge in reliable data has enabled the growth team to conduct more effective cold outreach, thereby directly contributing to the company's overall business expansion.

5. CONCLUSION

The project directly demonstrates the transformative impact that tailored solutions can have on business development. By addressing critical challenges in identifying executive leads, the webscraper not only streamlined the cold outreach process for Paraform’s growth team but also significantly increased the quantity and quality of leads generated. It allowed a tedious, manual process to become automated and yield exponentially better results.

In addition to enhancing lead generation, the project provided valuable insights into overcoming deployment challenges in a hybrid cloud environment. The integration of user-friendly interfaces with robust backend processing has shown that even non-technical users can leverage complex systems to drive business growth. Personally, I learned how business functional requirements should drive

the decision-making process in a system design.

6. FUTURE WORK

Future work on this project will focus on refining the accuracy of executive identification and reducing false positives in the data collection process. Enhancements in the text extraction and keyword matching algorithms can be implemented, including further development of the RAG approach. Additionally, improvements in prompt engineering for the ChatGPT LLM component will be a priority. User feedback will be critical in guiding these iterative enhancements, ensuring that the tool meets evolving business requirements.

REFERENCES

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., … & Riedel, S. (2020). *Retrieval-augmented generation for knowledge-intensive NLP tasks*. Advances in Neural Information Processing Systems, 33, 9459–9474. Retrieved from https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html

Ram, R. (2022). *Internal web scraper for job postings* [Internal document]. Paraform.