Final Report

Abstract

The Aerospace Industry Association of Michigan (AIAM) represents more than 120 companies, universities, and other aerospace organizations. AIAM's members frequently have difficulty filling aerospace industry jobs in large part because positions are posted in disparate places. This GVSU capstone project includes custom web scrapes for more than 40 AIAM members, scrape management functionality to create and maintain additional scrapes, and enhancement to AIAM's website to aggregate scraped job data in one central location with search, sort, and filter functionality. The team developed the project using the Python programming language, the web crawling framework Scrapy, and AIAM's WordPress website.

Introduction

The Aerospace Industry Association of Michigan (AIAM) is a trade association representing more than 120 companies, universities, and other aerospace organizations. AIAM's members perpetually struggle to fill job vacancies with qualified applicants. One major reason for this challenge is that prospective aerospace industry employees must tediously search many different organizations' websites to find position vacancies. Thus, AIAM requested this capstone project implement a scraping engine and an extension to the AIAM website for its web development team to install and its marketing team to maintain. The completed project periodically scrapes and aggregates member sites' available aerospace jobs. It also supports a non-technical user adding AIAM members and updating scrape scripts as member websites change. The website extension displays the aggregated job data and supports keyword search and filtering. Overall, a job-hunter must be able to conveniently search for aerospace industry jobs in Michigan on the AIAM site with the ability to navigate to a member organization's site for more information or to apply.

Scrape Engine for Aerospace Industry Talent

Software Architecture

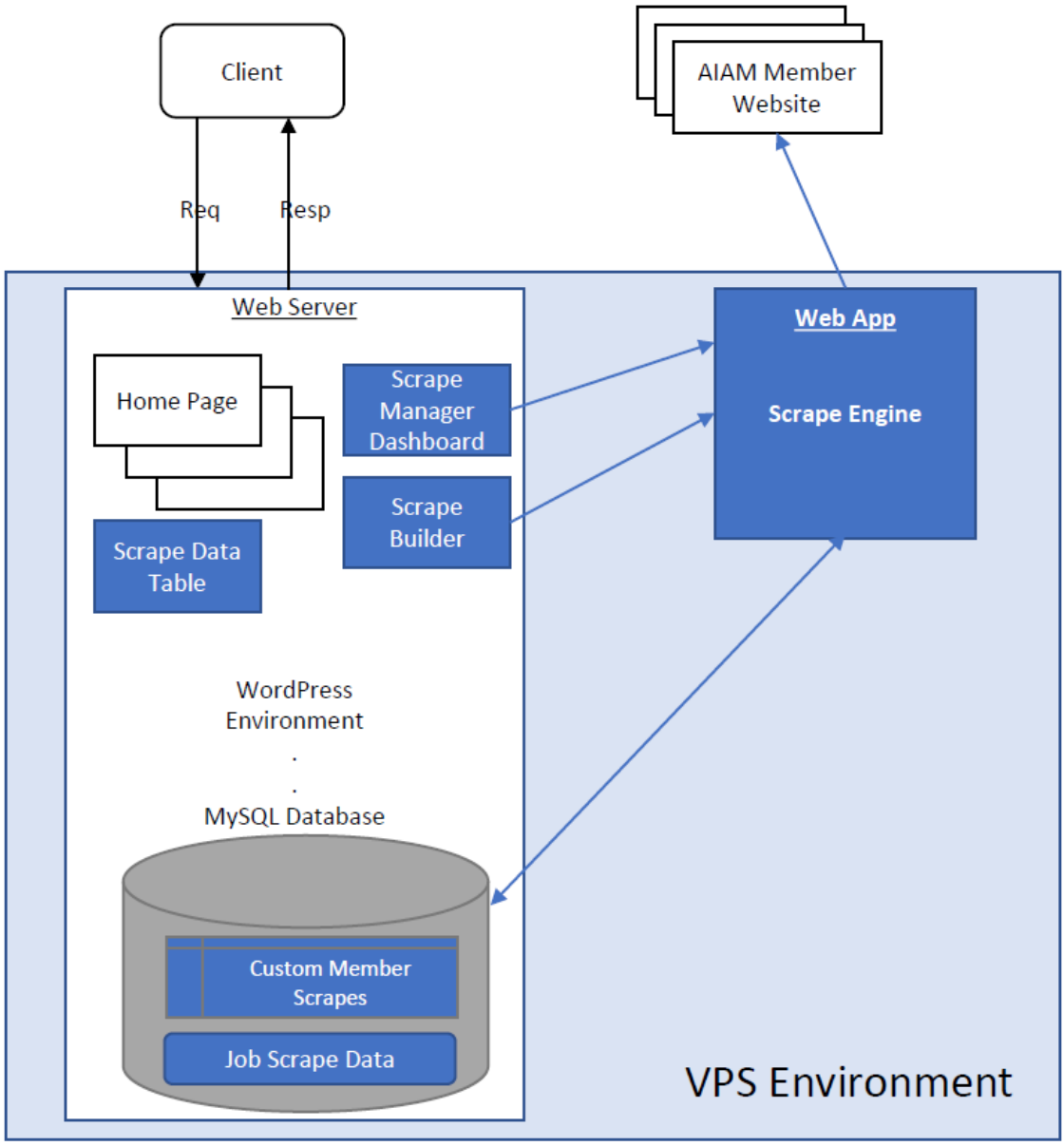
The top-level diagram below shows how the client begins by making a request on the AIAM website. The WordPress front controller and PHP file parser route the request.

Our AIAM web pages are dynamically generated from the native MySQL database instance. This database is the bridge between the Scrape Engine web application and the AIAM website. A newly-added web page, the Scrape Data Table page, conveniently displays the scraped data stored in the database in one central location with search, sort, and filter functionality.

The Scrape Engine supports two key interfaces: the Scrape Builder and the Scrape Manager Dashboard. These interfaces are implemented as new web pages on the AIAM website. The Scrape Builder web page allows for AIAM members to create and update the scrape script that scrapes their website. The Scrape Manager Dashboard is a simple GUI for a non-technical user to be able to validate, update, and delete the custom scrape scripts submitted by the AIAM members.

The Scrape Engine includes the logic to generate scrape scripts and store them in the MySQL database. It periodically (nightly) runs the “batch” of validated scrapes writing the scraped data into the database. Thereafter, the updated data is automatically incorporated into the Scrape Data Table web page.

Top-Level Diagram



In the above diagram, the components of the GVSU capstone project are shown in blue (and light-blue) while legacy components are shown in white. Go Lakers!

Non-functional Requirements

Performance

The project exhibits strong performance. Scraped data is stored in the WordPress instance so displaying the Scrape Data Table web page (and supporting search, sort,

and filter) is highly efficient. The Scrape Engine utilizes concurrency to scrape multiple member websites at the same time. Informally, scraping 40 member websites on a budget VPS took less than three minutes.

Security

Security is essential to any software project. The project implementation exposes a small interface to public users to support the Scrape Builder web page. Theoretically any interface could be an attack surface. For example, a bad actor could try to submit something akin to a SQL-inject on the Scrape Builder page. However, this is probably low-risk because that input data specifies X-Paths to attempt to scrape not commands to run. Moreover, the plan for fielding the project with Mind Utopia is to secure the page behind an AIAM member login portal. By limiting access to the page to members, we have developed a plan to minimize this threat.

Usability

Here we consider the usability of the project as a whole. With respect to the front-end, the Scrape Data Table page is reasonably simple and intuitive. We sought to prototype how the scraped data might be used, but most of our focus was on the actual Scrape Engine functionality. The team consistently recommended that AIAM consider employing Mind Utopia to have its UI/UX experts refine the front-end. This would further enhance the project's usability.

With respect to the back-end, usability is a function of how fluid it is to create a scrape and maintain existing scrapes. Importantly, the Scrape Builder and Scrape Manager Dashboard pages have simple layouts (though they are not especially pretty). Additionally, the team developed detailed instruction guides for the associated workflows. Taken together, the product is accessible even to non-technical users.

Languages, Libraries, and Frameworks

Web Application

Python

The scrape engine was written in Python. It is a back-end service that supports creating, retrieving, updating, and deleting custom scrape scripts. It also periodically runs all of the validated scrapes passing the scrape results to the item pipeline that cleans, validates, and stores the scraped job data in the MySQL database

Scrapy

The scrape engine relies heavily on an open-source Python application framework called Scrapy. The framework provides spider classes that support custom behavior for crawling and parsing data on different sites. This is how our scrape scripts request web pages, traverse them, select relevant data, and pass the data through our item pipeline for processing.

Web Server

WordPress Plugins

The legacy AIAM website was built using WordPress. We cloned the original site with a freely available Duplicator plugin. During iterative development of the Scrape Data Tables webpage we used several free data table plugins. However, these plugins did not fully support the project features so we transitioned to using custom HTML and CSS for both the Scrape Data Tables web page and the Scrape Manager Dashboard.

MySQL

The scrape engine's data is added to AIAM's WordPress MySQL database via the Scrapy item pipeline. Essentially, this is the bridge between the Scrape Engine and the website. We also rely on an Object-Relational Mapper (ORM) framework, SQLAlchemy, to convert Python objects to MySQL data tables. Specifically, we store job data in tables with the following schemas:

COMPANY

CoID	**Company**	**Company URL**	**Careers URL**	**Job X**	**Location X**	**Next Page X**	**UseDriver**	**Default Location**

JOB

JobID	**Job**	**Location**	**JobURL**	**Company**

Notably, the attributes with an "X" are shorthand for a data field representing an Extensible Markup Language (XML) X-Path element to be scraped. Additionally, there is a duplicate database table named "TEMPORARYCOMPANY" that is the exact same as the COMPANY table. When a member submits a draft scrape for approval, the attributes are temporarily stored in the TEMPORARYCOMPANY table. Once the scrape is approved, the record is moved from the TEMPORARYCOMPANY table to the COMPANY table where it joins the set of validated scrapes that are run periodically.

Technical Growth

Derek

Going into this project, I had no knowledge of web scraping and had never heard of PHP. At the beginning of the project, I mostly worked on stuff I was comfortable with, such as managing ZenHub's Kanban board, creating wireframes, and working on some Python code. As the project progressed, I started working with Max extensively on the hosted website, where I was able to learn more about secure shells, VPS, and using PHP to run Python functions from a web page.

I was also able to further my knowledge of Python, SQLAlchemy, and JavaScript through assisting with the user interfaces for the profile builder, control panel, and job results pages. My knowledge of web scraping was greatly enhanced from my experiences while writing the user manual and troubleshooting guides for the project.

Overall, I think that reviewing code and assisting in some capacity with every part of the project helped to really broaden my knowledge and understanding in every framework and language used in the project.

Chandler

When I started the project I thought I would be working more with database management and the set up of the database. I did not get into the database details besides checking the data a couple of times. I was able to grow in the field of programming by working with Max and Derek to edit various parts of the spiders. I was then able to use some of my web programming skills from CIS 371 to check if proper connections were being made and verifying errors were minimal. I learned the concept of scraping data from a website using X-Paths to narrow down the elements of choice. I was in charge of going through the websites and making sure that the scraped data was valid.

At the start of the project, I was using Max's interface to check the validity of the 120 member websites for AIAM and the results that were returned from the scrape. Each website was unique and produced its own set of problems. Throughout the project I worked with Max and Derek to make sure we narrowed down the results to be correct. In this process I gained extra knowledge of Python, HTML, and JavaScript. Towards the end I was able to get some more experience with UI design while working with Max and Derek to create the different scraping, profile, and integration steps for AIAM's duplicated website. A couple of different static UI pages were created for the initial profile setup that uses the xpaths, a testing page for the current results gathered, and a final concept demonstration page for what the results could look like on AIAM's page.

Our main focus as a group was to put the concept UI's together for presenting but MindUtopia, AIAM's website designers, will probably focus on integrating our functionality.

Max

I began the project with the assumption that complex parsing techniques would be the bulk of the work. While getting the Scrape Engine working and iterating on it to improve the scrape results was definitely half of the work, I greatly underestimated how tough getting a web environment running would be. Specifically getting Apache working, MySQL instances running, and troubleshooting file permissions created cascading challenges to implementing complex back end functionality. This was definitely my first time needing to have an active awareness of the permissions that each process needed. I had never even touched WordPress before so having to configure settings from MySQL was time consuming to understand.

While I have no desire to jump into building another hosted web environment I have no doubt that I could do so efficiently. Also I learned a lot about xpaths although I'm still curious about other applications of xpaths, what other use they have aside from web scraping. Additionally I became adept, alongside Derek, at php which turns out is a great skill to have! Having lurked around developer forums online I was under the impression that javascript and php served similar services though have now learned php works best as a backend controller in conjunction with javascript as a frontend controller.

Andrew

At the beginning of the project I anticipated opportunities for technical growth in UI design and development, web scraping frameworks, Platform-as-a-Service deployment, and Python programming. In practice, I did not spend much time doing UI design but did some UI development for a preliminary version of the Scrape Data Table. The project focus was on the back-end Scrape Engine for the first three sprints. Since I wanted to prioritize delivering the minimum viable product to AIAM I did not have much time to explore UI designs.

In contrast, I did learn a lot about web scraping frameworks in general and Scrapy in particular. I developed a preliminary, parameterized, general spider that could tailor its scrape to custom parameters for the target site. Also, I configured the scrape item pipeline to perform all of the database operations.

Next, I did persuade our project sponsor to fund a “live” development environment. This let me learn how to configure a shared web host and later how to provision and configure a VPS. We had difficulty configuring our VPS to run both the web server and the Scrape Engine so out of necessity I researched and practiced establishing a VPS with several cloud providers.

Finally, I spent a fair amount of time learning about Python programming. However most of my work was utilizing the Scrapy framework or SQLAlchemy rather than pure Python.

Software Engineering Code of Ethics and Professional Practice

Principle 2: CLIENT AND EMPLOYER

Software engineers shall act in a manner that is in the best interests of their client and employer, consistent with the public interest. In particular, software engineers shall, as appropriate:

2.01. Provide service in their areas of competence, being honest and forthright about any limitations of their experience and education.

AIAM understood from the outset that the team was composed of students with limited experience in the relevant technologies. During the bi-weekly client engagements the team candidly shared its challenges and plans to overcome them.

Principle 3: PRODUCT

Software engineers shall ensure that their products and related modifications meet the highest professional standards possible. In particular, software engineers shall, as appropriate:

3.08. Ensure that specifications for software on which they work have been well documented, satisfy the users’ requirements and have the appropriate approvals.

A major focus of the final sprint was the project transition where team members updated code comments, refinined user guides, and sought AIAM approval of a “near-final” product.

3.10. Ensure adequate testing, debugging, and review of software and related documents on which they work.

While our debugging and review efforts were robust, testing custom web scrapes is challenging and the team chose to prioritize other responsibilities.

Principle 5: MANAGEMENT

Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance . In particular, those managing or leading software engineers shall, as appropriate:

5.01 Ensure good management for any project on which they work, including effective procedures for promotion of quality and reduction of risk.

Derek volunteered to serve as the team's project manager during our first class meeting. The team unanimously agreed to accept this arrangement. His target was approximately half project management and half development work. One-half of one person on a four-person project amounts to 12.5% of administrative overhead for the project as a whole. This seems like a reasonable investment to support coordination, communication, etc.

5.04. Assign work only after taking into account appropriate contributions of education and experience tempered with a desire to further that education and experience.

The team exhibited strong individual ownership. Everyone accepted responsibilities they knew how to perform or could feasibly learn.

Principle 7: COLLEAGUES

Software engineers shall be fair to and supportive of their colleagues. In particular, software engineers shall, as appropriate:

7.04. Review the work of others in an objective, candid, and properly-documented way.

Every major work-product (pull-requests, scrapes, reports) were reviewed by another member of the team. There were several points of friction, but the discourse was always respectful and constructive.

Principle 8: SELF

Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession. In particular, software engineers shall continually endeavor to:

8.01. Further their knowledge of developments in the analysis, specification, design, development, maintenance and testing of software and related documents, together with the management of the development process.

This capstone project was an extraordinary learning opportunity to apply many of the major components of the GVSU Computer Science curriculum: Software Engineering (architecture design, project management), Computer Networking (SSL certificate troubleshooting), and Operating Systems (VPS provisioning and configuration).

8.02. Improve their ability to create safe, reliable, and useful quality software at reasonable cost and within a reasonable time.

The team researched and developed a feasible project roadmap and made reasonable adjustments as the project evolved. Good planning plus accountable execution equals a successful project.

Teamwork Reflection

This capstone project has been a success because every team member consistently demonstrated a strong sense of ownership. Everyone was working on the project part-time while balancing other competing responsibilities. Additionally, COVID-19 made trivial things like meetings more difficult. Nonetheless, everyone independently sought work and communicated their progress and challenges. For example, Derek facilitated every client engagement, Chandler developed almost all of the scrapes, Max built the sophisticated Scrape Engine, and Andrew wrote the majority of the feasibility study. The team worked well because everyone was self-motivated while committed to a shared vision.

Conclusions

Lessons Learned

Students in a Computer Science degree program must learn nuanced ideas about technology. These ideas can be difficult to master in their own right, but this capstone has shown that *implementing* even a simple solution to a practical problem can be challenging. We consider two specific examples in the next section. In general, the team found pair-programming and group brainstorm meetings helpful to solving its most difficult problems.

Challenges

Two major unforeseen challenges were the heterogeneity of AIAM's members' websites and the burden of configuring a VPS. First, while we anticipated differences in members' website structures, we thought that all of the companies would use reasonable job tables and lists. In fact, many of the sites were unorthodox or altogether inconsistent. This made scraping especially time-consuming and several members simply could not be included.

Second, provisioning and configuring a VPS to run both a web server and the Scrape Engine was easier said than done. Andrew spent many hours trying to run both services concurrently. He tried several cloud providers, several import/export utilities, and various configurations. Ultimately, Max realized the problem and manually modified several configuration files to get the VPS fully operational.

Reflections and Reconsiderations

While the project has been largely successful, we wish we could have emphasized our testing strategy more than we did. Regrettably, we overlooked researching and selecting a testing framework in the feasibility study. Thereafter, we were concerned about ensuring we could deliver a complete, maintainable product so we prioritized the Scrape Engine and reasonable maintenance features (the Scrape Manager Dashboard and Scrape Builder pages).

Opportunities for Extension

There are two main areas of prospective extension of the existing project: MindUtopia enhancements and a "Talent Feature" component. The MindUtopia enhancements include refining our draft Scrape Data Table web page based on UI and usability best practices, a security review of the potential vulnerabilities of the Scrape Builder web page, and a live fielding of the updated AIAM website (with access to the Scrape

Engine). Additionally, now that AIAM has a working Scrape Engine for aerospace talent *demand* it is considering marketing featured aerospace employees or *supply* as a separate module called “Talent Feature”.

Keys to Success

Our team’s success was a function of many factors. We started early by meeting with our sponsor during the first week of class. Then we did careful due diligence in the feasibility study as a “project roadmap” which decreased schedule risk. We requested a small budget to make the project 2x as fun (and 3x as much work). Finally, as we explained in detail in the Teamwork Reflection section, every member of the team exhibited strong ownership over the project’s results.

Despite the project’s success, we wish we would have had more practice working on larger projects (rather than developing an application and throwing it away). In general, internship and work experience was the best preparation for a complex, integrated capstone so the more development experience the better.