Summary

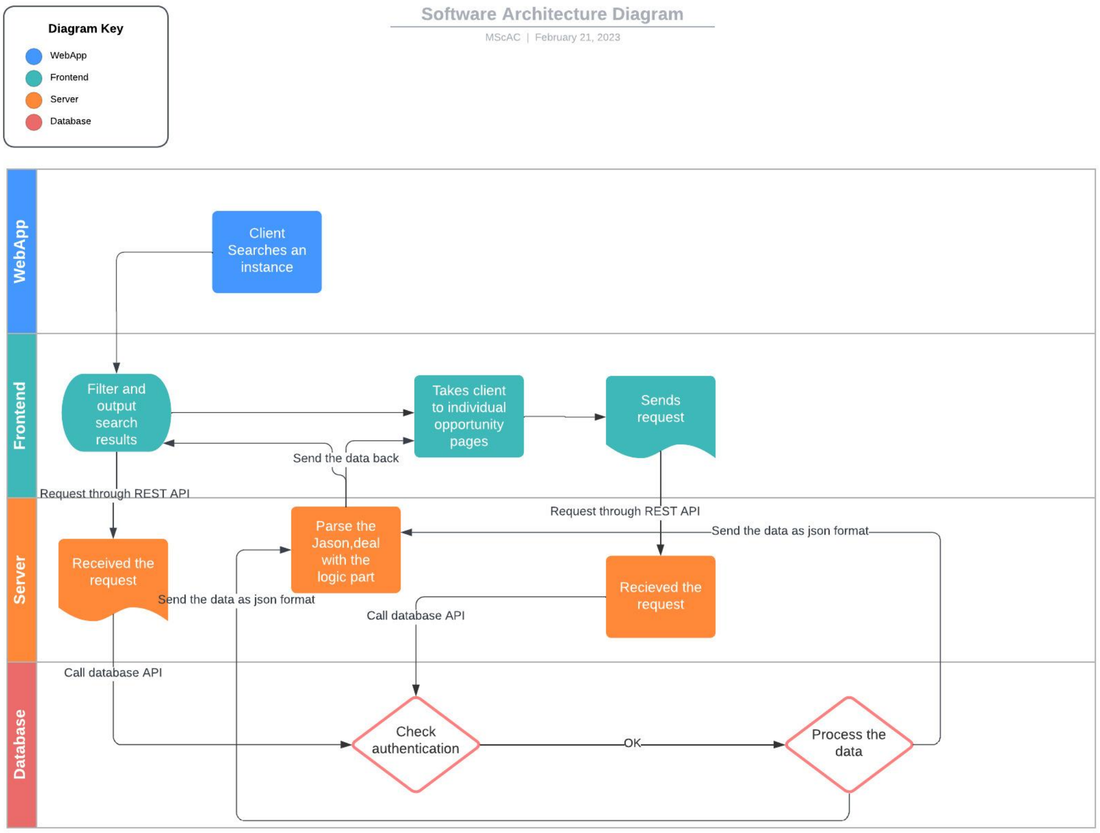
The project aims to be a web interface that allows students, industry partners, and researchers to easily browse, filter, and search through public abstracts of applied research internship projects carried out by MScAC students in previous years at the University of Toronto. These projects represent exceptional resources for attracting technical talent at the graduate level and promoting university research collaborations. This will be an interactive, searchable, and responsive website that works equally well on desktop and mobile devices.

We have partnered with Daniel Giovannini, the Associate Director for MScAC Partnerships, to ensure that the platform is designed to meet the needs of both students and industry partners. With his guidance, we are confident that we can create a website that is not only informative but also engaging and inspiring for all stakeholders.

Existing software/infrastructure

There is an air table available which stores information about each MScAC internship project, including project names, organization names and logos, links to organizations, academic supervisors, year posted, and the project abstractions.

Software Architecture Diagram



Project Division

We decided to divide the project into three parts: front-end development, back-end development, testing and deployment Infrastructure.

Front end development

For our project, we decide to use React, which is a widely used JavaScript library for building user interfaces for websites and web applications.

Ease of development: React's component-based architecture allows its ease of development. As shown in this blog, React has a shallower learning curve compared to Angular, another popularly used frontend technology, due to its building of reusable UI components, which can be easily combined to form complex user interfaces, speeds up the development process, and allows easy maintenance and update.

Availability of libraries & Maturity of the technology: The React ecosystem is home to a vast array of open-source libraries and tools, which can be used to simplify common development tasks such as routing, state management, and form validation, as mentioned in this blog. Many of these libraries have been developed and maintained by the React community for several years and have reached a high degree of maturity and stability. For instance, React Bootstrap is a widely used library that builds React components from scratch, without the need on dependency to jQuery. It has been around since 2011 and is now considered a mature and stable library but keeps evolving at the same time.

Domains: React is also widely used in various domains, including e-commerce, social networking, media, and healthcare. It has been adopted by many companies such as Facebook, Airbnb, and Netflix, who have built large-scale web applications using React (Olga Melnyk, 2022). For instance, Airbnb has a large-scale React application that handles millions of bookings per year.

Performance: React's virtual DOM and its efficient rendering mechanism help to optimize the performance of web applications built using React (Olga Melnyk, 2022). The virtual DOM is a lightweight representation of the actual DOM, which allows React to quickly compare changes to the UI and update only the necessary parts. React also uses a technique called reconciliation, which optimizes the way components are rendered by minimizing the number of DOM updates required. These performance optimizations can significantly improve the performance of web development using React.

Cooperation with Backend: Another main reason for choosing React is that it is compatible with Django, which is the backend technology that the other sub-team of our project team would choose. By integrating React into the Django environment, we can create a single-page application that uses React for front-end rendering and Django for back-end API services. This strategy can result in a more interactive and smooth user experience while also facilitating better code maintenance.

Back end development

For backend development, there are lots of options when deciding which programming language to use, but we have decided to use **python** for the following reasons:
Ease of use: Python is a general purpose high-level language that is considerably the easiest to learn and use. According to statistics, python will be known by 48.07% of programmers by the end of 2022, which is much higher than other languages like Java(33.27%) and C++(22.55%) (Vailshery). Therefore, using python makes the project easier for others to develop and maintain.
Maturity: With no doubt, Versatile and Mature Programming Language
Domain covered by python

- Machine learning / Artificial intelligence
- Desktop GUI
- Data analytics and data visualization
- Web development
- Game development
- Mobile app development
- Embedded systems

https://somenplus.blogspot.com/2020/11/top-7-domains-to-expertise-after.html

Abundance of libraries: Since our project is a search engine, the backend logic would involve some complicated concepts, such as spell-correcting and content-searching, which are difficult to be hand-coded from scratch. Python has a vast collection of libraries that can be used for various purposes, which are really helpful in backend development.

Wide support for frameworks: Python supports various modern web development frameworks, such as Flask and Django. In addition, thanks to the large developer community of python, there are third-party packages to support frameworks and platforms that originally don't support python(e.g. airtable).

Originally, we decided to use Django since we learned Django in CSC309. However since our partner has already implemented the entire database using airtable, we did some research, decided to try - **Flask** for the following reasons:

Database has been implemented by our partner: We don’t need to create models, the database is already implemented, Django is great for Model View Controller(MVC) and build in ORM which we don’t really need

Ease to use: Flask is well-integrated with python and requires minimum boilerplate code, which allows us to focus more on the searching logic, instead of struggling with the code templates and syntax issues. And if someone wants to take over our project in the future it’s easier and faster to understand the code

Minimum complexity: It’s easy for deployment and testing. First stage’s API testing using postman is sufficient.

Testing and deployment Infrastructure

Our team are responsible for testing codes. We separated our jobs into two different parts: front-end testing and back-end testing since the other two sub-teams are working on them.

Front-end testing(By Yuyang Wang, utorid(wang1423)); Back-end testing(By Jiaguan Tang, utorid(tangjiag))

We have considered two technologies: AVA and Pytest. We have selected to use Pytest in both back-end and front-end testing.

1.Ease of development: Both Pytest and AVA are easy to set up, use and import useful packages. Pytest has a simple syntax and provides many built-in fixtures that simplify test development. However, we are more familiar with Pytest than AVA, since Pytest is used in a lot of assignments in other computer science courses.

2.Maturity of the technology: Pytest has been around for more than a decade and is considered a mature technology. It has a large and active community that provides support, documentation, and plugins. AVA is a relatively new testing framework from 2009, and its ecosystem is not as mature as Pytest.

3.Domains covered: Pytest is a testing framework for Python, so it's ideal for testing Python-based backend applications. AVA is a testing framework for JavaScript, so it's ideal for testing Node.js-based backend applications. Our backend is built on the Flask framework, a Python-based backend application. Therefore,Pytest is the ideal testing framework for our application.

For our front-end html files, we imported “BeautifulSoup” package as a useful tool to examine the elements in the file. The package provides many useful functions that help us locate elements and find their relationships instantly.

4.Popularity:

Pytest is one of the most popular testing frameworks in the Python ecosystem, with a large and active community of users and contributors. AVA is popular among Node.js developers, but it is not as widely used as Pytest. Based on the information from Raygun, AVA only has 7% of users in JavaScript unit testing frameworks.

Sub-team Reports

Sub-team 10.1 report link:
https://github.com/csc301-2023-winter/assignment-2-10-1-chenx608-xuray2-zhan7289/blob/main/sub-team%2010.1%20assignment-2.pdf

Sub-team 10.2 report link:
https://github.com/csc301-2023-winter/assignment-2-10-2-dupeng2-zhuangfu/blob/main/Subteam%2010-2%20Report.pdf

Sub-team 10.3 report link:
https://github.com/csc301-2023-winter/assignment-2-10-3-tangjiag-wang1423/blob/main/Assignment_group_10.3_Summary.pdf

References

1. Angular vs React: Liu, E. (2021). Angular vs React: What Should I Learn in 2023? [Blog post]. CodersID. https://codersid.com/angular-vs-react-what-should-i-learn-in-2023/

2. React documentation: React. (n.d.). Getting Started [Documentation]. https://reactjs.org/docs/getting-started.html
3. React Router documentation: React Router. (n.d.). Documentation [Website]. https://reactrouter.com/
4. Top 10 companies that use React: Career Karma. (n.d.). Top 10 Companies That Use React [Blog post]. https://careerkarma.com/blog/companies-that-use-react/
5. State of JS 2021 survey: Sacha Greif et al. (2021). State of JS 2021. [Website]. https://2021.stateofjs.com/en-US/libraries/front-end-frameworks/
6. Testim. “What Is the Best Unit Testing Framework for JavaScript?” AI-driven E2E automation with code-like flexibility for your most resilient tests. AI-driven E2E automation with code-like flexibility for your most resilient tests, February 18, 2021. https://www.testim.io/blog/best-unit-testing-framework-for-javascript/.
7.“Unit Testing Frameworks in Python.” Zenesys. Accessed February 17, 2023. https://www.zenesys.com/unit-testing-frameworks-in-python.
8. o2task. “>Strategic Innovation – the AVAC Framework (Activities, Value, Appropriability, and Change).” O2task's Blog, May 10, 2011. https://o2task.wordpress.com/2010/08/03/strategic-innovation-the-avac-framework-activities-value-appropriability-and-change/.
9.“Top 6 Best Python Testing Frameworks [Updated 2023 List].” Software Testing Help, January 14, 2023. https://www.softwaretestinghelp.com/python-testing-frameworks/.