Technological Feasibility

Oct. 22, 2021

Team CareerNet

Team Sponsor:

Dr. Andy Wang,

José R. Díaz Aquino

Team Mentor:

Han Peng

Team Member:

Carter Taylor,

McKenzie Clark,

Carmen Montalvo,

Ran Li

Table of Contents

1.0 Introduction

Creating an environment where college students can achieve their goals after their undergraduate degree is a united objective for college deans. One way to achieve that goal is to encourage external learning outside of the curriculum. External learning could take the form of a job, internship or research related to a student's field of study. However, observing students' external learning as a collective can prove difficult.

This project aims to bridge the gap between the information on students' external learning to the college dean. To achieve this goal, information will be scraped from a website that student's regularly update about their current professional status. This information gathered will be stored, sorted and displayed on a website. The college dean can select how he wants to view or sort the collected data, in order to better understand the students' achieved milestones.

2.0 Technological Challenges

As with any project, there are challenges that will present themselves. This section is meant to highlight some of these challenges that may present themselves throughout the development of this web application.
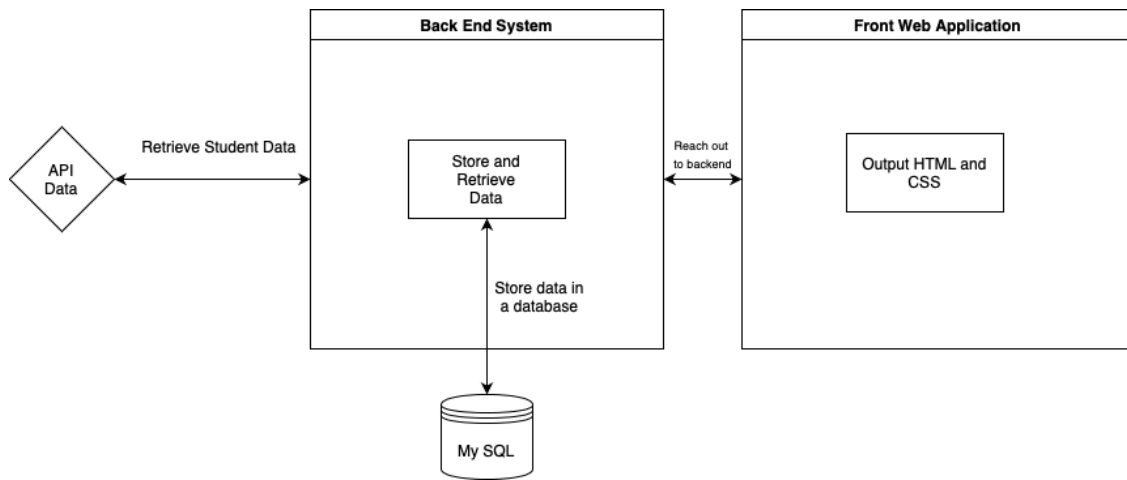

Figure 1: Details our broad goal for how it is expected for the system to work. As technical challenges are described more, this system will expand to include precisely how each chosen technology fits in.

2.1 Data Sources

The most important challenge is finding reliable data sources to fuel the project. This project has a reliance on the way that thousands of NAU students have handled their data, each one potentially being unique. Thus it is important to have many different sources to draw from, in order to make sure it covers all possible outlets of data.

2.2 Data Retrieval

Another important obstacle is finding the most optimal process to fetch the necessary data in order to develop the project. It requires a powerful language that can easily handle the calls to API sources, as well as a fast way of interpreting and normalizing duplicate information that may be received to store in our database.

2.3 Front End Web Framework

This program will require a front end system in order for the user to be able to access the data being retrieved. The framework will need to provide a straightforward user interface for the client, while also being able to communicate with the backend system effectively to display accurate data.

2.4 Database Management System

A database system is absolutely necessary for the product being developed, as it will need a well organized storage location to easily access student data. The database will need to smoothly communicate to the back end system to store and retrieve data.

2.5 Back End Framework

The back end system will act as an interface between the front end and the database. In order to be successful, it must be able to seamlessly respond to front requests, with accurate information retrieved from the database.

2.6 Data Visualization

To provide a quick "information at a glance" interface, the program will need a powerful data visualization framework. This framework must be able to transmit faultlessly with the back end system, in order to ensure our data represented in our visualizations is accurate. The system should have a small barrier of entry to create visually appealing graphs as well.

3.0 Technological Analysis

This section will address each of the technological challenges, to finalize how each challenge can be solved for this project to succeed.

3.1 Data Sources

3.1.1 Introduction

The main objective of this project is to gather data of current NAU students and it's alumni in order to track their career growth. It will not only need a source of data able to reflect the student as they attend college, but also be able to track them after graduation and future job opportunities they received. Through research, it is understood that multiple sources are needed in order to accomplish these main goals. For this reason, the sources selected to be investigated are Linkedin, Handshake, and self made student surveys.

3.1.2 Desired Characteristics

While searching for data sources,a few key characteristics have been deemed important to gain success:

- **Ease of Access:** How easy is it to gain access to the source data/ API?
- **Automation:** How can the data source be used to easily automate the storage of each piece of information?
- **Up To Date Data:** How valid and recent is the data being grabbed?

3.1.3 Alternatives

3.1.3.1 Linkedin API:

Linkedin is a popular business and employment service that has been around since 2002. Because it has been around for so long, it has been the long established reigning king for employees and employers to post their professional success and milestones they have achieved throughout the years. The information Linkedin has available is expansive, and with that information they provide developers an API that allows them to interface and retrieve the said data. This is what will be used in order to retrieve user information.

        Pros:
          ● Most updated information source for alumni students

        Cons:
          ● Very Strict API usage (gaining access and what is allowed to de done with said data)
          ● Documentation is vague, and doesn't provide clear examples
          ● Limited amount of requests per day.

3.1.3.2 Handshake API:

Handshake was founded in 2013 and is used to give all college students access to career opportunities. It has over nine million active student users and over 600 thousand employers. It has thousands of college and university partners, including NAU. It allows students to update their career information, submit a resume, and look for external learning opportunities. Handshake is also highly endorsed by faculty for students at NAU to use.

Pros:
- Direct partner with NAU
- Easy

Cons:
- Not the most updated information for alumni
- Currently lacking obvious documentation.

3.1.3.3 Student Surveys:

Every year, Northern Arizona University sends out a senior survey to graduating seniors to gage their overall satisfaction on their four years at school. But additionally, the survey also seeks to learn more about what the student will be doing after graduation professionally (job opportunities, graduate school, work location). This works well in gathering information for students who are getting a job right out of college, but doesn't work as well for students who receive a job offer after graduation, and completely leaves out tracking success of students who are currently attending college. So it would be optimal to pilot a new yearly survey for non-senior students that would ask them similar questions, to track the career success they are having already.

Pros:
- Direct Student Data
- High reliability

Cons:
- Inability to track students after graduation

3.1.4 Analysis

After comparing each of the sources, it can be seen that each of the sources balances one another out by covering the weaknesses of the others.

Student Surveys are an absolute necessity in order to get the information of existing underclassmen who might not use other programs to acknowledge their professional growth. Using google survey, a demo data retrieval has been made possible (see figure 3.1.1). Additionally, Handshake should allow the obtainment of other updated students' information as they attend NAU. This should be an API that is accessible because NAU is using Handshake as a service for the university.

Lastly Linkedin has proved to be the most difficult API to get access to. Currently the only retrieval of information is of one authenticated group member using Linkedin's *r_liteprofile* permissions. With this permission, the only knowledge able to be fetched are the users first and last name, and only after they approve the application to access their data (See figure 3.1.1). Unfortunately, this leaves no useful data able to be pulled from Linkedin. The next step is to leverage the connection with NAU in order to receive higher permissions to have access to more data. This is imperative to complete the goal of retrieving alumni career information. While waiting for official Linkedin API permissions to be approved, an alternate solution is using an unofficial Linkedin API. With less permissions needed, the source can still be tested until official access is granted.
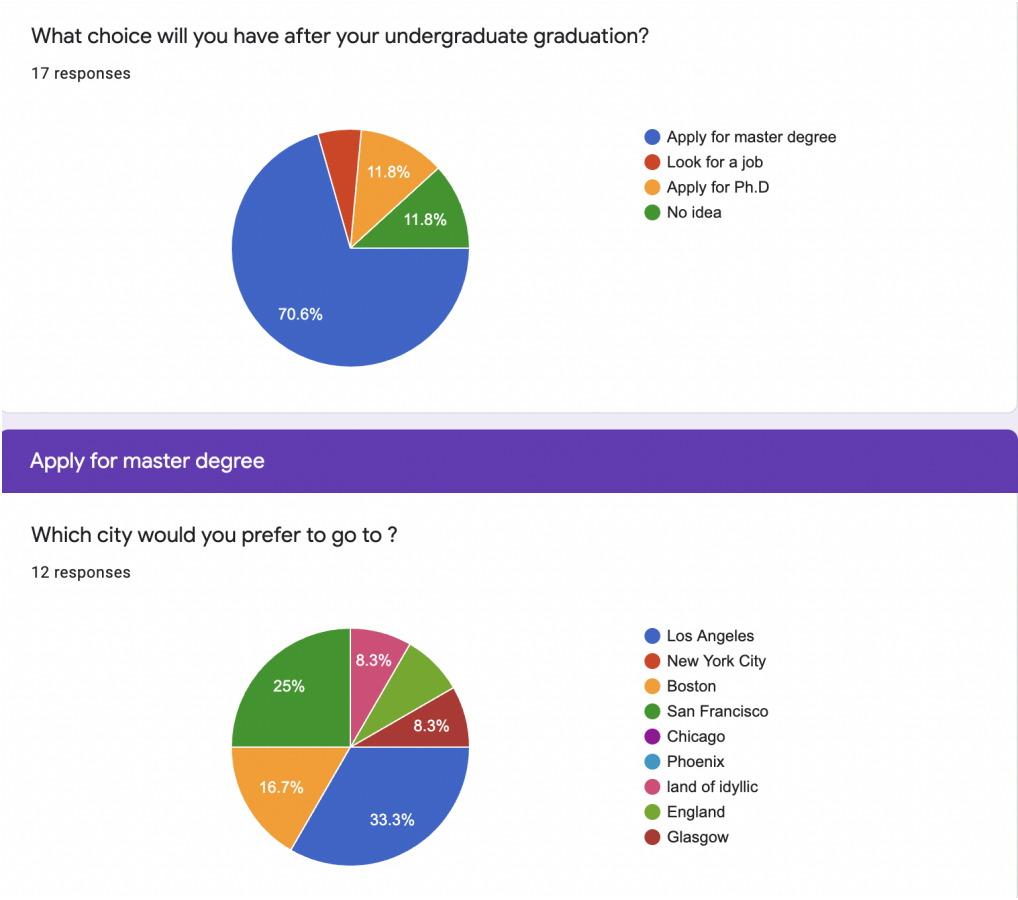
What choice will you have after your undergraduate graduation?

17 responses

- Apply for master degree
- Look for a job
- Apply for Ph.D
- No idea

11.8%

11.8%

70.6%

Apply for master degree

Which city would you prefer to go to ?

12 responses

- Los Angeles
- New York City
- Boston
- San Francisco
- Chicago
- Phoenix
- land of idyllic
- England
- Glasgow

8.3%

8.3%

25%

16.7%

33.3%

Figure 3.1.1: Shows the demo data that able to be fetch using google survey

```
{
    "firstName": {
        "localized": {
            "en_US": "Carter"
        },
        "preferredLocale": {
            "country": "US",
            "language": "en"
        }
    },
    "id": "EhpwUK2BMO",
    "lastName": {
        "localized": {
            "en_US": "Taylor"
        },
        "preferredLocale": {
            "country": "US",
            "language": "en"
        }
    },
    "localizedFirstName": "Carter",
    "localizedLastName": "Taylor",
    "profilePicture": {
        "displayImage": "urn:li:digitalmediaAsset:C4E03AQFuMjNFdnjPxw"
    }
}
```

Figure 3.1.2: Shows the data that can currently be pulled with the Linkedin API access

3.1.5 Chosen Technology

Because each of the sources balance each other out, each will be
necessary in order to build the best possible data storage. Student
surveys will be utilized as a way for clients to manually import data
through CSV files, with data students currently attending the college.
Handshake data can be accessed automatically on a scheduled basis for
as long as the application is live. This will allow the gathering of
information of current students and also cross the threshold of
students that do update their Handshake with information after they
graduate. Lastly Linkedin, with extended privileges, will allow the
fetching of information for alumni students, closing the final gap in
information.

Table 3.1: Data Sources Summary Table (ratings out of 5)

Technology	Ease Of Access	Automation	Up to Date Data	Total
Linkedin API	1	5	5	11
Handshake API	3	5	3	11
Student Surveys	4	3	4	11

3.1.6 Proving Feasibility

Providing feasibility for all sources each presents it's own unique challenge. Specifically having to wait for full Linkedin API access in order to fully test out the software. To ease the transition, while waiting for full access, testing of the API has been started with the permissions available now. Additionally, testing has begun on a 3rd party Linkedin API that can be used to gage how to possibly parse the data for storage, and begin to write test functions that should work very similarly with the official API. Testing out Handshake will be much simpler, since access to the API will be gained very soon. Once gaining access, test sample pulls with user data will be done to simulate the flow of information that might enter the database. Lastly, testing of the student surveys of a small group will be completed by simulating the way it exports to a CSV. By following the same schema that a survey would usually export as, functions can be written to correctly parse as if it were fetching real data. So although challenges have presented themselves, there is confidence that there are ways to prove how well the functionality will work.

3.2 Data Retrieval

3.2.1 Introduction

Once there are stable sources to draw from, it is also important to have a consistent way of retrieving the data from said sources.

3.2.2 Desired Characteristics

The language/framework should follow these guidelines:

- **Ease of Use:** How easy is it for a member to understand and contribute with the technology?
- **Efficient Parsing:** How efficient is the language in fetching from our sources as well as normalizing the data between our sources?
- **Effective Back-end Communication:** Can the language communicate effectively with the back end system and databases to store data?

3.2.3 Alternatives

3.2.3.1 Python:

Python is a popular programming language with support for thousands of additional open source libraries. Python is a versatile programming language that has been used often for data science.

Pros:
- Extensive libraries supporting CSV parsing, API data retrieval, and database storage.
- Immense documentation and resources to draw from.

Cons:
- Not built directly into the back end

3.2.3.2 Node JS:

Node JS is an open source back end Javascript library that runs outside a browser. It's main function is to provide a back end framework for an application to run on, but also supports additional add ons to compute back end processes.

Pros:
- Built directly into the back end

Cons:
- Not as extensive and powerful for data parsing.

3.2.4 Analysis

By Looking at the benefits of each of the technologies, it is seen that Python would prove to be the best option for handling information needed. It provides a robust amount of features with all of its additional libraries, and pre-existing documentation for data parsing. Testing of some example API pulls using Python's requests library has been already proven to be possible (See figure 3.2.1 and 3.2.2). While NodeJS has the benefit of being directly built into the backend system, the use of Python is the better alternative since the back end framework (ExpressJS) supports Python programming calls to obtain data as needed.

```python
# Authenticate using any Linkedin account credentials
api = Linkedin('websitemessagecontact@gmail.com', 'Jimmy123!')

# GET a profile
profile = api.get_profile('carter-taylor')

# GET a profiles contact info
contact_info = api.get_profile_contact_info('carter-taylor')

# GET 1st degree connections of a given profile
connections = api.get_profile_connections('1234asc12304')

# Print data from profile
print(json.dumps(profile["education"], indent=4, sort_keys=True))
```

Figure 3.2.1: example Python code for pulling API information

```
"education": [
    {
        "activities": "Association for Computing Machinery (ACM)",
        "degreeName": "Bachelor of Science — BAS",
        "entityUrn": "urn:li:fs_education:(ACoAACrSlS8BuQtq1nGUURRm9GoFvmJhhWTDQTk,581049681)",
        "fieldOfStudy": "Computer Science",
        "fieldOfStudyUrn": "urn:li:fs_fieldOfStudy:100189",
        "grade": "Senior",
        "projects": [
            "urn:li:fs_project:(ACoAACrSlS8BuQtq1nGUURRm9GoFvmJhhWTDQTk,1184299946)",
            "urn:li:fs_project:(ACoAACrSlS8BuQtq1nGUURRm9GoFvmJhhWTDQTk,833229341)"
        ],
        "school": {
            "active": true,
            "entityUrn": "urn:li:fs_miniSchool:17776",
            "logoUrl": "https://media-exp1.licdn.com/dms/image/C560BAQHJP4PaFNhQLg/company-logo_",
            "objectUrn": "urn:li:school:17776",
            "schoolName": "Northern Arizona University",
            "trackingId": "22E+g68PQVmWhGcvKmBXTQ=="
        },
        "schoolName": "Northern Arizona University",
        "schoolUrn": "urn:li:fs_miniSchool:17776",
        "timePeriod": {
            "endDate": {
                "year": 2022
            },
            "startDate": {
                "year": 2018
            }
```

Figure 3.2.2: Example information able to be parsed from the API pull

3.2.5 Chosen Technology

Table 3.2: Data Retrieval Summary Table (ratings out of 5)

Technology	Ease Of Use	Efficient Parsing	Effective Back-end Communication	Total
Python	3	5	3	11
NodeJS	2	3	4	9

Python. Between the two, each one had a similar ease of use. But Python was put slightly higher since there is more familiarity with the language. In addition to this, Python accomplishes data parsing better since it contains many well documented libraries to assist with

fetching data from web APIs and CSVs which will be the main use case for this project. Lastly, in the topic of effective back-end communication, although Python isn't the best communicator with the back end, it is still very capable in doing so. In addition, the most important factor was how efficient it would be in handling data computation, thus Python is the best, making it the final decision.

3.2.6 Proving Feasibility

It should be fairly easy to plan feasibility for Python. Testing has already started by fetching data from CSVs from custom surveys. This will allow for a demo of how the functions will work to parse data and store it in the database. Additionally, tests pulling data for the unofficial Linkedin API have been done using Python's *request* library. Success has been achieved in correctly receiving data, the next step would just be to connect the code to the back end code which is covered more in the "Back End Framework" section of this document. With the success already had in testing out the Python code, confidence and success will only continue to grow as familiarity with other systems expands.

3.3 Front End Web Framework

3.3.1 Introduction

This project is based on scraping information and displaying it accurately to the user. A front-end library or framework is needed to retrieve data from the database and present it in a user-friendly environment. In this context the main functionality of a front-end library is to facilitate student information gathered to the user. A

front-end library provides a user interface that is not only efficient, but an overall pleasant experience to the user.

3.3.2 Desired Characteristics

The front end web framework should follow the following:

- **Ease of Use:** Is the framework design simple enough to execute in a short period of time? Does this framework have existing syntax documentation or resources?
- **Compatibility:** Is the framework compatible with our choice of database system and back-end framework? Is it compatible with popular browsers?
- **Included Tools:** Are the tools included with the framework beneficial to the website interface?

3.3.3 Alternatives

3.3.3.1 *React*:
React is an open-source JavaScript library that specializes in building user interface components. It is a popular tool used to create dynamic and appealing web applications. Because it is commonly used, there are a lot of resources and documentation found in the library. One of React's features include a virtual DOM or Document Object Model. Web applications created with a virtual DOM update and run much faster than applications created with a real DOM. React also includes specialized tools for debugging. Allowing developers to create more efficient code in a smaller amount of time. Because React is a library, it includes many third party packages that can be utilized for specific needs in the user environment. This includes packages that are compatible with NodeJS, a programming language used to execute code server-side.

    Pros:
- Use of Virtual Dom creates faster application run time
- Resources are easily accessible

- Compatibility with NodeJS
- Debugging Tool
- Many third party packages for specialized needs

Cons:
- Not a framework
- Javascript heavy

3.3.3.2 *Bootstrap*:

Bootstrap is a popular framework used to create an appealing user interface design. It is a collection of code that wraps around the front-end structure of a web application to simplify the design aspect of developing the user environment. Bootstrap produces a website that is compatible with all popular browsers and due to its popularity there are many resources available.

Pros:
- Compatibility with popular browsers
- Easy to use styling components
- Many resources
- It is a framework

Cons:
- Compatibility issues accessing back-end frameworks
- Possible learning curve

3.3.3.3 *Angular*:

Angular is an open-source Javascript framework that is used to develop the front-end of web applications. Angular is written in TypeScript and contains a real DOM or document object model. It is a popular framework so support and resources are available and it is compatible with all popular browsers. Angular features a two-way binding structure which allows the created application to listen for events and update values at the same time. The focus of Angular is for large commercial applications, instead of smaller projects.

Pros:
- Two-way binding structure

- Compatibility on popular browsers
- Easy to find resources

Cons:
- Not meant for smaller projects
- The use of Typescript creates a learning curve

3.3.4 Analysis

The purpose of a front-end framework is to create a dynamic user interface that is easy and useful to the developers creating the web application. This purpose can be achieved within each of the selected alternatives, Angular, React and Bootstrap. Each alternative has a specialized area that is useful to the project development in comparison to the others. For instance, Angular's two-way binding is useful for a faster response time in retrieving student information from the back-end while also listening to the user. Bootstrap's capabilities as a framework make it easier for developers to create a clean user interface that allows student information to be displayed in a clear and efficient manner. Although React is a library, it achieves faster application response times through its virtual DOM. React also has specialized packages that can be added on by developers, allowing compatibility with programming languages.

3.3.5 Chosen Technology

Table 3.3: Front-end Framework Summary Table (ratings out of 5)

Technology	Ease of Use	Compatibility	Tools	Total
React	4	5	5	14
Bootstrap	5	3	4	12
Angular	3	4	4	11

React with Bootstrap. The combination of features from the React library and the Bootstrap framework make this combination of front-end tools optimal for this project. The choice to include both stemmed from React's compatibility and faster run time, as well as Bootstrap's ease of use for developers.

3.3.6 Proving Feasibility

To prove feasibility of the chosen front-end framework and library, it is necessary to establish a successful connection between the compatibility of the chosen back-end framework and chosen front-end frameworks. After a connection is established, testing the compatibility between React and Bootstrap and the effectiveness of combining them is needed. Creating a simple web page that accesses student information from the database could be an effective way of determining the compatibility between the back-end and front, as well as React's compatibility with Bootstrap.

3.4 Database Management System

3.4.1 Introduction

For the development of this product, the database system is absolutely necessary, and even more necessary is the management of that system. The database needs to communicate smoothly with the back end system to store and retrieve data. The database management system conducts unified management and control of the database to ensure the security and integrity of the database. It provides a variety of functions, allowing multiple applications and users to use different methods in order to create, modify, and query databases at the same time or even at different times. It enables users to easily define and manipulate data, maintain data security and integrity, and perform concurrent control under multiple users and restore databases. A well-organized

storage location is needed to easily access student data to track students' career success. If a large amount of data is not well managed, errors may often occur, which will have a great impact on the product.

3.4.2 Desired Characteristics

The database framework should follow these guidelines:

- **Compatibility:** How well does the database interact with the back end system (Express JS) and the data retrieval language (Python)?
- **Ease Of Use:** How easy is it to retrieve data, and group different data sources together? Is there existing documentation for the language?
- **Performance:** How quickly is the database able to retrieve data without hindering performance of the server?

3.4.3 Alternatives

3.4.3.1 MySql:

MySQL is a relational database management system. Relational databases store data in different tables instead of putting all data in a large warehouse, which increases speed and flexibility. The SQL language used by MySQL is the most commonly used standardized language for accessing databases. MySQL software adopts a dual authorization policy, divided into a community version and a commercial version. Due to its small size, high speed, low total cost of ownership, especially the characteristics of open source, the development of small and medium-sized websites generally chooses MySQL as the website database.

phpMyAdmin is a MySQL database system management program written in PHP, allowing administrators to manage MySQL databases with a web interface. With this web interface, it can be a better way to enter complex SQL syntax in a simple way, especially for the import and export of large amounts of data. One of the greater advantages is that phpMyAdmin runs on the web server like other PHP programs, but can use HTML pages generated by these programs anywhere, that is, to remotely manage the MySQL database. Using phpMyAdmin, the creation, modification, and deletion of databases and data tables can easily be done.

Pros:
- small size, high speed, low total cost of ownership.
- Fast speed and high flexibility.
- Use phpMyAdmin to remotely manage the MySQL database, so that it can easily create, modify, and delete databases and data tables.

3.4.3.2 MongoDB:

MongoDB is a product between relational and non-relational databases. It is the most versatile and most like relational database among non-relational databases. The data structure it supports is very loose, it is a bson format similar to json, so it can store more complex data types. The biggest feature of Mongo is that the query language it supports is very powerful. Its syntax is a bit similar to an object-oriented query language. It can almost achieve most of the functions similar to single-table queries in relational databases, and it also supports indexing of data. MongoDB is a document-oriented database based on distributed file storage. It can be easily expanded geographically and horizontally for better performance. The design

goal of MongoDB is high performance, scalability, easy deployment,
easy use, and very convenient to store data.

Pros:
- MongoDB is oriented to collection storage, and it is easy to
  store object type data.
- MongoDB has a free mode and uses a modeless structure for
  storage.
- MongoDB supports full indexing and can be indexed on any
  attribute, including internal objects.
- MongoDB supports queries. MongoDB supports rich query
  operations, and MongoDB almost supports most queries in SQL.
- MongoDB supports Perl, PHP, Java, C#, JavaScript, Ruby, C
  and C++ language drivers.

Cons:
- MongoDB does not support JOIN like relational databases.
- There is data redundancy in MongoDB, which takes up too much
  space. This can lead to unnecessary memory usage.
- MongoDB limits the document size to a maximum of 16MB.
- MongoDB has uneven data distribution in cluster shards.
- MongoDB does not have as mature maintenance tools as MySQL.

3.4.4 Analysis

The database management system is a data operation software that can
provide data entry, modification, and query. It has functions such as
data definition, data operation, data storage and management, data
maintenance, and communication, and can allow multiple users to use
it. In the csv file data collected in the following figure, it uses

the python CSV method to process the data and build a MySQL database, using DML (Data Manipulation Language), for users to implement operations such as adding, deleting, updating, and querying data.

Timestamp	Email Address	What choice will you hav	Which city would you pre	Which field are you willin	Which industry would yo	What is your estimated a	Which city would you pre	Which industry are
10/18/2021 20:53:25	cwt48@nau.edu	Look for a job					Phoenix	Technology
10/18/2021 20:53:55	rl759@nau.edu	Apply for master degree	San Francisco	STEM	Technology	$100,000 +		
10/18/2021 20:54:24	zl274@nau.edu	No idea						
10/18/2021 20:56:54	sl2532@nau.edu	Apply for Ph.D						
10/18/2021 20:59:56	mz366@nau.edu	Apply for master degree	Los Angeles	Business	Finance	$10,000 ~ $30,000		
10/18/2021 21:00:44	hp263@nau.edu	Apply for Ph.D						
10/18/2021 21:02:01	yy254@nau.edu	Apply for master degree	Boston	Business	Finance	$30,000 ~ $70,000		
10/18/2021 21:03:03	yc537@nau.edu	Apply for master degree	San Francisco	Business	Finance	$70,000 ~ $100,000		
10/18/2021 21:04:14	ym465@nau.edu	No idea						
10/18/2021 21:06:21	xc93@nau.edu	Apply for master degree	Boston	Business	Finance	$10,000 ~ $30,000		
10/18/2021 21:07:49	lw683@nau.edu	Apply for master degree	land of idyllic	Education	Education	$100,000 +		
10/18/2021 21:14:24	yc469@nau.edu	Apply for master degree	Glasgow	Psychology	Consulting	$10,000 ~ $30,000		
10/18/2021 21:15:09	qj27@nau.edu	Apply for master degree	San Francisco	Business	Consulting	$10,000 ~ $30,000		
10/18/2021 21:22:59	bl776@nau.edu	Apply for master degree	England	Business	finance or law	$30,000 ~ $70,000		
10/18/2021 21:30:14	rd775@nau.edu	Apply for master degree	Los Angeles	Healthcare and social se	Technology	$10,000 ~ $30,000		
10/18/2021 21:33:37	zg86@nau.edu	Apply for master degree	Los Angeles	Business	Finance	$70,000 ~ $100,000		
10/18/2021 21:57:35	yz482@nau.edu	Apply for master degree	Los Angeles	Business	Finance	$30,000 ~ $70,000		

Pros:

- Control data redundancy
- Ensure data consistency
- Improve data sharing

3.4.5 Chosen Technology

Table 3.4: Database Management System Summary Table (ratings out of 5)

Technology	Compatibility	Ease Of use	Performance	Total
MySQL	5	4	5	14
MongoDB	2	3	5	10

MySQL. Some important features revolve around DML (Data Manipulation Language) which allows us to insert, update, and retrieve data. This data manipulation method is flexible, simple to use for users, and provides excellent performance, which can provide data for data entry,

modification, and query. Choosing this high-performance data manipulation method will be the best decision for this web application.

3.4.6 Proving Feasibility

A major function of this product is the ability to design a reliable database to track the success of student careers. Collection of a large amount of students' data will occur.In order to improve the security of the data encryption system, improve the security of the data encryption system, improve data backup and recovery, and enhance the management of multimedia, these issues must be reflected in the database management system. Python will be used to extract all the data needed from the CSV file and the use of DML will be to implement operations such as appending, deleting, updating, and querying the data.

3.5 Back End Framework

3.5.1 Introduction

In order to have a way to serve data to the applications users,a strong back end framework is necessary. The back end framework will act as an interface between the front end and the database. The back end framework will need to be able to handle multiple connections to the server and balance multiple routes that will respond to users requests.

3.5.2 Desired Characteristics

To be as successful framework, these are the necessary features it needs:

- **Front End Compatibility:** Is there native support for the front end framework chosen and the back end system? How easy do both systems work together?
- **Ease of Use:** How easy is it to jump in and use the framework? Is there sufficient documentation and resources online to reference?

3.5.3 Alternatives

3.5.3.1 NodeJS / ExpressJS

Node JS is a back end runtime environment that uses JavaScript to deploy web applications. Because it is open source, it has access to a wide range of third party libraries. One of these is called ExpressJS, which is an additional back end framework that provides an added utility to quickly deploy web applications and create routes that users would access.

Pros:
- Expansive resources and documentation.
- Compatibility with React, the front end framework.
- Ease of use in quickly deploying web applications, and creating additional services.
- Pre-existing knowledge using NodeJS + ExpressJS.

Cons:
- Not as extensive and powerful for data parsing.

3.5.4 Analysis

NodeJS + ExpressJS provides many great features to help streamline the application. The main pro being that Express has direct support with the front end library: ReactJS from the research done. In addition, it is very simple to jump right into an ExpressJS application (see figure 3.5.1). This provides a simple way to jump into ExpressJS, and learn more advanced features later. Finally, being one of the most popular open source libraries, there are an abundant amount of sources holding information to help ease the learning process of the more complicated features.

```
1   const express = require('express'  4.17.1  )
2   const app = express()
3   const port = 3000
4
5   app.get('/', (req, res) => {
6     res.send('Hello World!')
7   })
8
9   app.listen(port, () => {
10    console.log(`Example app listening at http://localhost:${port}`)
11  })
```

Figure 3.5.1: Demonstration of the compact code needed to create an ExpressJS web application.

3.5.5 Chosen Technology

NodeJS + ExpressJS. Confidence with NodeJS + ExpressJS was evident from the beginning due to existing knowledge of the framework, but also because it works well with the front end framework, ReactJS. For this reason, NodeJS and ExpressJS will be the framework being used for the ECN project's back end system.

3.5.6 Proving Feasibility

Because Express allows for a minimal way of setting up an application, providing feasibility was acheived by quickly testing code on a local host server. The code created locally will translate almost 100% to our production code, with the only major change being the port that will run the application. This will allow testing of everything including user authentication, post routes, and database retrieval before moving the code live. Less time will have to be spent making sure this web application functions correctly when moving live, which in turn will free up manpower to work on other portions of the project. With NodeJS, we can continue to grow the application locally with absolute confidence the features will also function on live.

3.6 Data Visualization

3.6.1 Introduction

A main point of this project is data collection. Along with this, the web application should be able to display the data to the user. For this reason, it will be implementing a data visualization tool in order to provide users with an accessible way to see and understand any trends and patterns within the gathered data set.

3.6.2 Desired Characteristics

The Data Visualization tool should meet the following:

- **Compatibility:** How well will the library work with the front end and back end frameworks?

- **Ease of Use:** How steep is the learning curve? Are there any existing references?
- **Included Features:** Are the tools useful to the web application? How well will it visualize the information?

3.6.3 Alternatives

3.6.3.1 *Recharts*:
Recharts is a data visualization library that creates appealing charts. It allows for customization of charts such as the types, elements and shapes by just changing and passing custom components. This tool offers several ways in which the user can add interactions to the charts.

Pros:
- Shows documentation for each example of charting
- Works well with React
- Simple-to-use-API
- Easy to learn

Cons:
- Multiple animated charts on one page can cause lag
- Resource for issues are not reliable for answers

3.6.3.2 *D3*:
D3 is short for Data-Driven Documents and is a data visualization tool that uses HTML, SVG, and CSS. It aids in bringing data to life by allowing data manipulation within the provided documents. In doing this, D3 is extraordinarily flexible and shows the capabilities of the web standards. This tool uses an API holding a collection of modules that can be used independently or together to build a desirable chart.

Pros:
- Clean API
- Extremely fast
- Supports large datasets

Con:
- Outdated and confusing Documentation

- Steep learning curve
- Clashes with React

3.6.4 Analysis

The main purpose of a data visualization library is to provide tools that can manipulate data in a way in which it can be visualized by the user of the web application so that they can see and clearly understand all of the patterns, trends, and outliers within the dataset. Both Recharts and D3 can do this in an efficient and creative way. These libraries do this by providing documentation with data that can be manipulated to fit the desired charting type for the information. Either of these libraries will be useful in providing the charting desired with each having their pros and cons.

3.6.5 Chosen Technology

Table 3.5: Data Visualization Tools Summary Table (ratings out of 5)

Technology	Ease of Use	Compatibility	Features	Total
Rechart	4	5	5	14
D3	2	2	4	8

Recharts. Recharts is a great library to use in this web application. It is easy to learn and provides a simple-to-use-API with a variety of different types of charts that can be manipulated to hold the dataset that will be collected from API's and surveys.This tool also works well with the front end framework (React) that will be used in this application as well as the back end framework (NodeJS/ExpressJS). Recharts gives an endless number of possibilities of ways in which to show data so it can be easily accessible and understood by the users.

3.6.6 Proving Feasibility

One of the key aspects of this project is data visualization. The data being collected through APIs form LinkedIn and Handshake along with

the student surveys will be manipulated using Recharts to show a graph of the different kinds of external learning students have done. This will be done by using the examples of charting in the documentation provided and changing them to fit our information and fitting them into the front and back end frameworks.

4.0 Technological Integration

The major technologies that will be used for the project have been shown to be effective in each of their respective use cases. The final major challenge being faced is how to make sure all of the technologies can communicate in an effective manner. In this section, how each of the technologies will work in unison in order to build the envisioned product will be discussed.

4.1 Envisioned System

The main action point from our system is triggered from the front end ReactJS pages. From the front end, a user is able to interact with the page, and certain actions will send a post request to the ExpressJS back end system. Depending on the specific action that the user makes, it could either be to send data, or retrieve data from the database, thereafter a response being sent back to the front end. ExpressJS will handle the request acting as an interface between the front end and database. The way ExpressJS handles a request will be contingent on the parameters sent from the front end. A request for data will have ExpressJS reach out to the existing database, retrieve the necessary data based on the parameters, and return the data back to the front end system. Anticipating the response from the server, ReactJS will serve the user the data delivered from the response.
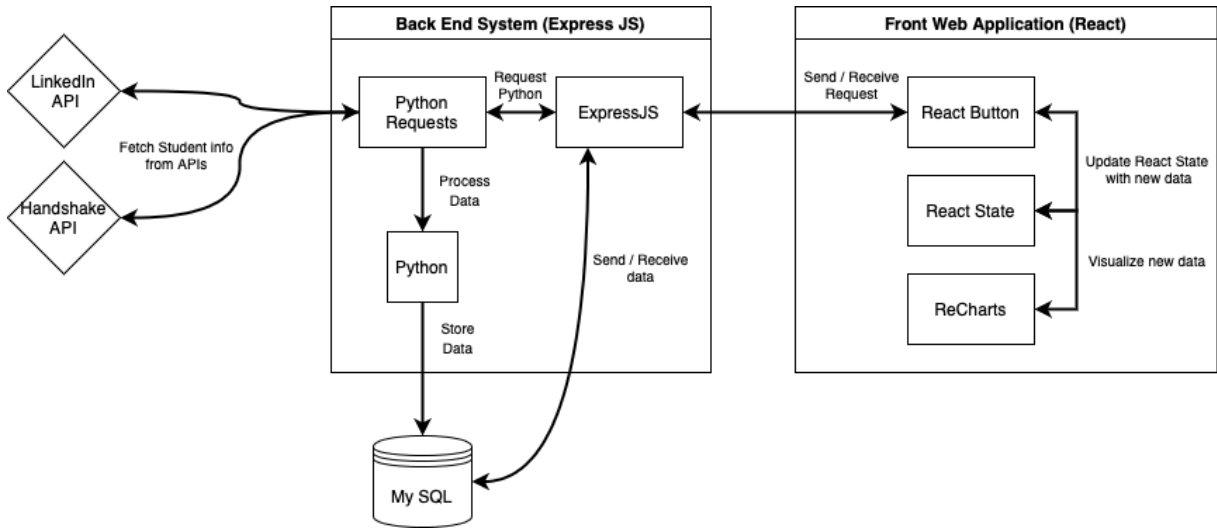
Figure 4.1: Describes the flow of action the product system will go through, including how each technology interacts with one another. The two largest boxes represent the front and back end system. Each individual technology is wrapped in a box and placed in the layer of the project where it will exist (ex: Python in the back end). Arrows represent the action that takes place in order to connect two technologies.

5.0 Conclusion

College deans aim to create an environment in which students have all
the help and resources they need in order to achieve their career
goals, not only while they are earning their undergraduate degree but
also after graduation. They do this by encouraging students to get
some form of external learning outside of the classroom, whether that
be a job, internship or research within their field of study. However
there still is a gap between students pursuing an external learning
opportunity and the dean's knowledge of them doing so.

The ECN project strives to bridge this gap by creating a web
application that will scrape information on the students' current
career status from applications such as LinkedIn and Handshake along
with other student surveys using Python. This data will then be stored
onto a database and sorted using MySQL due to its compatibility with
Python and Nodejs. Which brings up the topic of the back end system
using NodeJA/ExpressJS. This back end system will quickly deploy web
applications, routes users would access and is greatly suitable to
work with the front end system, which uses React with Bootstrap. Using
React with Bootstrap will give the optimal tool needed for this
program to run smoothly with easily accessible resources and easy to
use styling components. All of this will be presented to the user as
both a data set and charts. These charts will be made using Recharts
which is a data visualization library providing an endless number of
possibilities of ways to show data that can be easily accessible and
understood.

All of these components will come together to create an efficient yet appealing web application for users, like the dean, to use in order to gain information on the students' accomplishments in achieving the milestones needed to obtain career success.