**Opus**: Helping Students find Work they Like

**Team members:**

|  |  |  |
| --- | --- | --- |
| Name | Email | GitHub |
| Shrivats Agrawal | shriv9@seas.upenn.edu | [ShrivatsAgrawal](https://github.com/ShrivatsAgrawal) |
| Abhinav Atrishi | abhinav1@seas.upenn.edu | [AbhinavAtrishi](https://github.com/AbhinavAtrishi) |
| Samuel Kamien | skamien@seas.upenn.edu | [skamien](https://github.com/skamien) |
| Bushra Kidwai | bkidwai@seas.upenn.edu | [bushrakidwai](https://github.com/bushrakidwai) |

1. Motivation:

As undergrad or grad students, we are in throes of finding a summer internship or a post-college job search. The process can be exhausting and time taking considering the research that goes behind finding the perfect role and more importantly the right company. Hence our application is aimed at making the ‘research’ simpler for people looking for jobs.

This application will enable the user to view a range of stats of a company and filter out companies that match their interests. The functionalities in the web app like viewing the latest news and sentiments will help user gain a holistic view about the reputation of the company. Eventually, the user can use it to find jobs at these companies and view recommendations of similar job opportunities.

1. Primary Features:

An outline of the primary functionalities we intend to include in the app are:

* **Company Search**: This will allow the user to search companies by applying multiple filters to refine the search from the broad range of companies to a narrower and more relevant search. We will use FMP dataset for retrieving these companies.
* **Job Search**: Once the user has a relevant search the web application will allow to search for jobs. The web app will use the dataset we are scrapping from Indeed to find job listings. The job search will use filters to list jobs from that company and provide recommendation for similar jobs.
* **News**: This will enable the user to view the latest news about the company as well as peers to get a sense of how the company is perceived. We will fetch news from FMP and will update the dataset dynamically to keep it consistent and up-to-date.
* **Sentiment**: This will provide an interactive visualization of the social media sentiments of that company against the sentiments of it’s peers. This will enable end user to explore and compare sentiments of varying social media data from a set of peers.

1. Future Scope:

In addition we plan to implement the following functionalities:

* Login Functionality: Helps user to create an account on the web application. This can enable user save previous search results.
* Daily Updates of the Dataset being scraped from Indeed to ensure consistent and up-to-date data.
* Additional filters like S&P 500 or Nasdaq 100 for filtering companies on the home page.
* Provision to filter by top gainers and losers and most active stocks for filtering companies on the home page.

1. Pages:

The pages we plan to implement are as follows:

* **Home Page:** This is the first page the user will end up on when they visit our web application. The home page has a search functionality which lets the user search for companies by applying filters like: Name of Company, Number of employees, market capitalization, sector etc. The search will list all the companies that match the description.
* **The Company Page:** Once the user selects a company from the search in the homepage they will be directed to the company page which will be dynamically generated. This page will hold general information about the company like description of the work it does, CEO, contact details like phone number, number of employees, website link etc.
* **The News Page:** From the company page the user can navigate to the news page. The news page will fetch the latest news from the FMP database and update the page. Another added functionality is we will also fetch news of similar companies and display for the user to get a sense how the company is performing compared to peers.
* **The Sentiment Page:** We will display the social media information from FMP. This page will have an interactive visualization where we fetch the sentiments of peers and give a visual representation of how the company is perceived in general.
* **The Job Search Page:** This will help user apply filters according to their preference and find relevant jobs. The application will also recommend similar jobs.

1. Relational Schema + ER Diagram:

**Relational Schema for Opus Database:**

**Companies**(id, symbol, name, price, exchange, exchangeShort, type)

**CompanyInformation**(id,symbol volAvg, marketCap, currency, phone, industry, website, description, ceo, sector, country, address, image, zip, state, city, fullTimeEmployees)

FOREIGN KEY (id) REFERENCES Companies(id)

**CompanyNews**(id,url, publishedDate, title, text, imageLink, website)

FOREIGN KEY (id) REFERENCES Companies(id)

**Sentiment**(companyID, date, absoluteIndex, relativeIndex, generalPerception, sentiment, redditPostMentions, redditPostSentiment, redditCommentMentions, redditCommentSentiment, tweetMentions, tweetSentiment, stocktwitsPostMentions, stocktwitsPostSentiment, yahooFinanceCommentMentions, yahooFinanceCommentSentiment)

FOREIGN KEY (companyID) REFERENCES Companies(id)

**Peers(**companyID, peerID)

FOREIGN KEY (companyID) REFERENCES Companies(id)

**IndeedJobs(**indeedJobLink, jobTitle, indeedCompanyName, rating, jobLocation, jobType, shortDescription, payPerHour, companyID, companyName)

FOREIGN KEY (companyID) REFERENCES Companies(id)

FOREIGN KEY (companyName) REFERENCES Companies(name)

Graphical user interface, application

Description automatically generated**ER Diagram:**

1. SQL DDL:

|  |
| --- |
| CREATE TABLE Companies( Id VARCHAR(256),  symbol VARCHAR(20),  name VARCHAR(100),  price FLOAT(3), exchange VARCHAR(50), exchangeShort VARCHAR(50),  type VARCHAR(100), PRIMARY KEY(Id)); |

|  |
| --- |
| CREATE TABLE CompanyInformation( Id VARCHAR(256), symbol VARCHAR(20), volAvg FLOAT(3),  marketCap FLOAT(3),  currency VARCHAR(5),  phone VARCHAR(20),  industry VARCHAR(50),  website VARCHAR(200),  description VARCHAR(MAX), ceo VARCHAR(100), sector VARCHAR(50), country VARCHAR(50), address VARCHAR(MAX), image VARCHAR(MAX),  zip VARCHAR(10),  state VARCHAR(50), city VARCHAR(100),  fullTimeEmployees INT, PRIMARY KEY(Id, symbol), FOREIGN KEY(Id) REFERENCES Companies(Id) ); |

|  |
| --- |
| CREATE TABLE CompanyNews( Id VARCHAR(256), url VARCHAR(MAX),  publishedDate DATETIME,  title VARCHAR(500),  text VARCHAR(MAX),  imageLink VARCHAR(MAX), website VARCHAR(MAX), PRIMARY KEY(Id, url), FOREIGN KEY(Id) REFERENCES Companies(Id) ); |

|  |
| --- |
| CREATE TABLE Peers( companyID VARCHAR(256),  peerID VARCHAR(256), PRIMARY KEY(companyID, peerID), FOREIGN KEY(companyID) REFERENCES Companies(Id), FOREIGN KEY(peerID) REFERENCES Companies(Id), ); |
|  |

|  |
| --- |
| CREATE TABLE Sentiment( companyID VARCHAR(256),  date DATETIME,  absoluteIndex FLOAT(5),  relativeIndex FLOAT(5),  generalPerception FLOAT(5),  sentiment FLOAT(5),  redditPostMentions INT,  redditPostSentiment FLOAT(5),  redditCommentMentions INT,  redditCommentSentiment FLOAT(5),  tweetMentions INT,  tweetSentiment FLOAT(5),  stocktwitsPostMentions INT,  stocktwitsPostSentiment FLOAT(5),  yahooFinanceCommentMentions INT,  yahooFinanceCommentSentiment FLOAT(5). PRIMARY KEY(companyID, date), FOREIGN KEY(companyID) REFERENCES Companies(Id), ); |

|  |
| --- |
| CREATE TABLE IndeedJobs( indeedJobLink VARCHAR(MAX),  jobTitle VARCHAR(200),  indeedCompanyName VARCHAR(100),  rating FLOAT(2),  jobLocation VARCHAR(200),  jobType ENUM('Intern', 'FT'), shortDescription VARCHAR(MAX),  payPerHour FLOAT(2),  companyID VARCHAR(256),  companyName VARCHAR(100), PRIMARY KEY(indeedJobLink), FOREIGN KEY(companyID) REFERENCES Companies(Id), FOREIGN KEY(companyName) REFERENCES Companies(name), ); |

1. Pre-Processing:

Data will be pre-processed using python to remove missing rows, apply default values to rows with missing key values, reject rows with unusable formatting, and correct errors from the scraper.

1. Technologies:

Technologies used will include Python’s requests library, Selenium, Beautiful Soup, MySQL, node.js, and react.js. Requests and Beautiful Soup will be used to build the database, MySQL will be used to handle queries, Node will be used to launch the backend, React will be used to create the UI, and Selenium will be used for testing.

1. Division of Responsibilities:

The database will be built by Shrivats and Abhinav, Bushra and Sam will start on the backend, and the frontend will be completed as a joint effort between everyone.