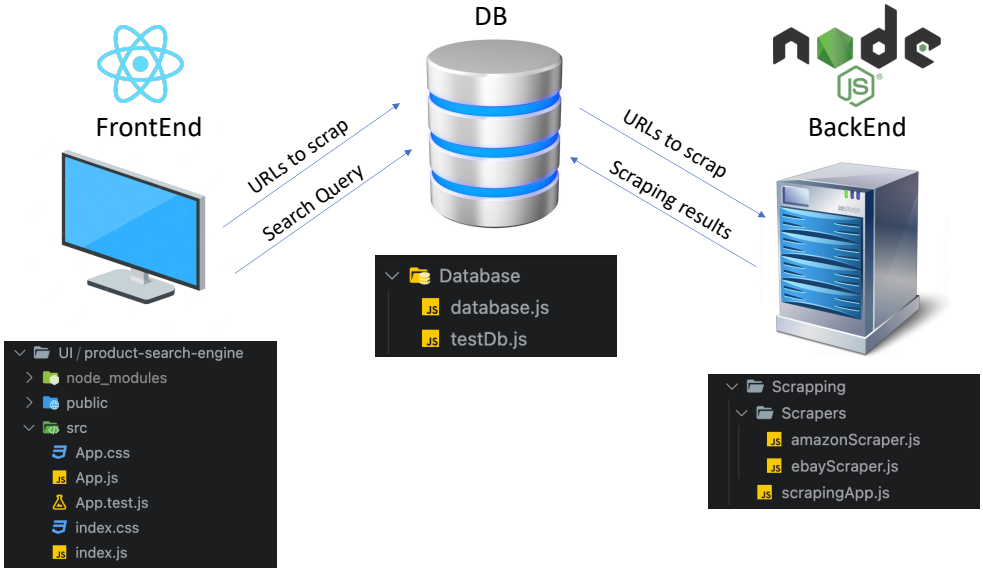
Skyhawk Security Home assignment (Full stack candidate)
Jonathan Matetzky

1. General Structure



   a. **Backend**
      - Implemented in Node.JS
      - Consist of; main scraping app, 2 designated scrapers (for amazon and eBay)
   b. **Database**
      - Implemented in JS
      - Simple mocked DB that supports several methods
      - Initiates a single-toned instance which is exported to be used by other apps in the project.
   c. **Frontend**
      - Implemented in React.JS
      - Uses DB's interface to insert URLs and submit queries.

2. **Engineering Aspects**
   - Considering the scope of the project and time constraints, the project was designed to be able to run in a local environment without setting up a server environment.
   - The DB initializes and exports a **single-toned instance**, the same instance imported by the frontend and the backend. Thanks to the methods exposed by the DB, both the frontend and the backend can update the list of URLs, save scrapping results, and perform a search among the scrapping results based on a given keyword.
   - In designing the project, I emphasized modularity to enable scalability in the future - a separate module for each domain.
   - Some tests are implemented.
   - Since relative imports outside of `src/` (in the Frontend) are not supported. I've tried moving it inside `src/`, and adding a **symlink to it from project's `node_modules/.**

3. **Application's flow**
    - The Frontend imports the single-toned instance of the Mocked DB.
    - The Frontend runs the Scrapping function it imported.
    - The Scrapping function runs in the background and listens for any changes in Mocked DB.
    - The UI is sped up and allows the user to send new URLs to the DB
    - The Scrapping module detects a change in the URLs database, performs Scrapping and saves the results in the Mocked DB
    - Now the user can perform a search within the Scrapping results saved in the DB.

4. **Points for further work/Meeting the objectives**
    - In sake of simplicity and to allow the project to be ran locally the Backend is ran locally by the Frontend.
    - Given more time and resources I would have create a **server-side module** or file within the React project that handles the backend functionality, such as handling API endpoints and interacting with the mocked database. Would have also implement the necessary API endpoints within the server-side module.
    - For example, could have defined an endpoint for receiving POST requests to /api/urls and handle the insertion of URLs into the database.
    - UI is very simple – **lack of styling**, **support for paging** and edge-cases handling.
    - The eBay Scrapper doesn't work.
    - I haven't limited the number of URLs that can be concurrently downloaded (configurable number).