**Alexander Yee**

For the final project, I am doing:

- Currently working on retrieving HTML files and parsing them to retrieve and organize relevant data such as the web title, body and source url
- Also focusing on outputting the data retrieved into a json format that can be used with ElasticSearch
- Worked with Rogelio to understand how to configure Elasticsearch on virtual cluster
- Helped with bulk-loading web crawled documents into Elasticsearch Lucene index
- Helped with issuing multi-match queries and rendering highlights

**Web crawling( Shiyao Feng )**

- Using multiple threading to get URL link without duplicates BY concurrent.futures
- Using multiple threading to download the HTML file to a local folder
- In the data folder, the application creates a different level folder to hold the files.
- The application can input the number of pages to crawl and the number of levels
- Default: if the application gets the file over 1GB in level K, the application will stop when it finished web crawling in level K.
- If the user wants to input the limit by page or level, the memory doesn't have the limit by 1 GB.
- When web crawling reaches Max level, the application will stop until the job at Max level.

**Brett McCausland**

- Configuration of the web server connecting to Elasticsearch
- The Querying of documents in Elasticsearch Lucene index through web interface
- Part of the communication between the Elasticsearch server and local website
- Issuing a query through a RESTful API over HTTP
- Returning the search results to the website
- Build Web application, with express framework and bootstrap

**Rogelio Macedo**

For the final project, I am doing:

- Configuration of the virtual cluster
- Configuration of running Elasticsearch on virtual cluster
- Whiteboarding of converting each html doc into loadable json object format
- Configuring the mapping of the data loaded into Elasticsearch
- The bulk-loading of the web-crawled documents into Elasticsearch Lucene index
- Part of the communication between the Elasticsearch cluster and local website

- Issuing requests (GET, HEAD, POST/PUT) through a RESTful API over HTTP to cluster/server
- Partially contributing towards returning the search results to the website
- Implementation of undefined exception handling of rendering null-defined variables
- Guided the implementation of multi-match queries of the fields: title, body
- Guided the implementation of highlighting for both title match and body match