CS410 Project Progress Report: Food Recipe Search Engine

Team name: F20_TIS_DEV_TEAM

Team Members:

Jonathan LaFlamme(Captain)	**jml11@illinois.edu**
Pradeep Sakhamoori	**ps44@illinois.edu**
Rahul Sharma	**rahul9@illinois.edu**
Rohan Khatu	**khatu2@illinois.edu**

Project Summary:

Our original project proposal described a vertical search engine, specializing in Indian Cuisine that supports both queries and a recommender system. Our project scope has since narrowed due to time constraints. Our current project expectation is a working search engine that supports full text search with auto-suggestion as well some advanced filtering/query options. The application is run on Elasticsearch and containerized on Docker to run locally with minimal installation steps.

Progress made thus far:

- Identified our food recipe dataset
- Developed search engine application using Elasticsearch and Kibana (for index file generation)
- Dockerizing (container) to launch Elasticsearch server in the background and tunneling food recipe application to host machine on 127.0.0.1:5000 port using Flask
- Validating and debugging container on Ubuntu and MacOS
- Additional configurations and installations required for Mac users to get the tunneling of webserver onto localhost
- Python notebook to injest data
- Preprocessing and inverted index configuration tested on Kibana
- Limited autosuggest and autocomplete features are deployed
- Github project repo with build and deploy instructions

Remaining Tasks:

- Formatting search results page (html)
    - Organizing recipe links with single click URL
    - Displaying results on the same page as the query
- Improving and enhancing the autocomplete and autosuggest features
- Python script automating the preprocessing and inverted index generation
- Optional inclusion of images with search engine, if possible

Challenges and Issues:

- Bug fixing in terms of adding missing python packages for deployment of automation pipeline with container
- Running elasticsearch in background with Flask
- Identifying search and autosuggest algorithm (eg. n-gram, character n-gram, stemming, lemmitization)
- Team members learning and deploying Elasticsearch and Kibana toolkits.
- Enabling Docker on Ubuntu Linux, specifically identifying the correct docker run options for routing Docker network traffic to localhost with '--network host' option.
- Enabling Docker on Mac OS, specifically since docker runs as a virtual machine on Mac, additional installs and procedures are required for docker container to have access to mac host as a localhost
- Integrating python Flask app with Elasticsearch and front-end
- Identifying correct method for building Elasticsearch index