Patrick Canny
Ellis Springe
Liam Ormiston

Landscrape Product Backlog

Web Scraper

The main feature of our application is the web-scraping functionality running in the background. Our web scraper will be iteratively improved in order to allow for support of multiple sites and comparison of results from those sites. We were able to successfully implement support for both Yelp and Foursquare via the BeautifulSoup library for web parsing. Down the line, we would like to find a way to increase the speed of the scraper using either a cloud-based or database solution.

- *V1.0 - Yelp API, Single Query Support*
- *V1.1 - BeautifulSoup Parsing, Multiquery Support*
- *V1.2 - Multiple Site Support Using Foursquare and Yelp, Tested for Verification*

User View Control

The main view includes a persistent page layout created with HTML templating and Bootstrap. All routing is done server-side in Python. We used a form client for Python/Flask called WTForms in order to generate and handle form fields. The most up-to-date solution includes form fields that allow the user to select a location to source search results from.

- *V1.0 - Navbar, Jumbotron, WTForms*
- *V1.1 - Includes Loader*
- *V1.2 - Includes location input control*

Server

The server is generated and run via a Python server file. As mentioned above, this file is where all routing information lives. This is also the primary point of contact between the search engine and the frontend. We deployed our app to Heroku with one small bug - testing suite does not work, but this feature would likely not even reach the production version, so it's pretty low on our priority list.

- *V1.0 - Server Functions Locally*
- *V1.2 - Application Deployed on Heroku Server, Testing Suite Doesn't Function on Heroku Side.*