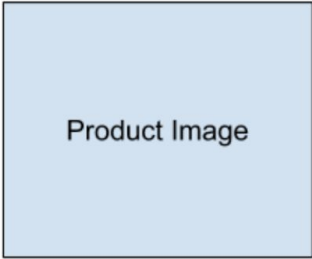
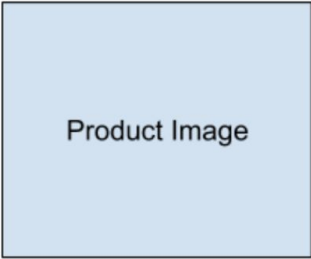
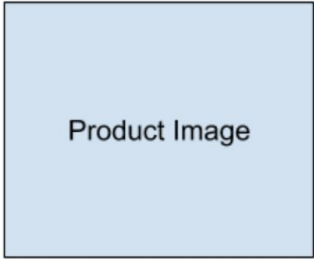


With our new push into helping our brands win on Walmart we want to give them the ability to analyze competitors to their products using a convenient UI:

Product ID:

1.		2.		3.	
	Title: iPhone 6s Avg. Organic Rank: 2.2		Title: Nokia 23 Avg. Organic Rank: 2.9		Title: Samsung S6 Avg. Organic Rank: 4.9

The goal of this task is to design and build an API that serves the necessary data for the frontend to display.

## API

This project will need to use the Walmart API which is documented here:

<https://developer.walmartlabs.com/docs>

In particular look at:

- <https://developer.walmartlabs.com/docs/read/Home>
- [https://developer.walmartlabs.com/docs/read/Search\\_API\\_IR\\_V2](https://developer.walmartlabs.com/docs/read/Search_API_IR_V2)

You can easily play around with the api here:

- <https://developer.walmartlabs.com/io-docs>

API Keys to use:

- ufrfx5tg4uc9cx6pjsq97b2
- 4xb5s6yeeuu385zunmqfunt4

Note that each API key allows for 5 requests/second and 5000 requests/day. The second key is there in case you accidentally run out of daily requests using the first one.

## Details

Build an API server that runs locally and serves a response for one “analyze” request given a productId (this product id is the one that Walmart recognizes)

The way we determine the biggest competitors is by seeing which other products show up when you search for the given product. The higher the “organic rank” (order that a product appears) of a product the bigger of a competitor that product is to the given product.

E.g. When searching for “laptop” the “Lenovo ideapad 330s 15.6” laptop” shows up 3rd in the results list, meaning it has an organic ranking of 3.

The ranking of competitors is determined by the “average organic rank” of that product showing up across all the tested search terms.

Therefore, rank #1 will have the lowest  $\text{sum}(\text{organic rank } 1, \text{organic rank } 2, \dots) / \text{num\_search\_terms}$  of all the competitor products.

If a product doesn’t show up on a search then it can be considered to have an organic ranking of 20 for the average calculation.

### **Base Case:**

The search terms consist of the product title split based on spaces or commas (up to a maximum of the first 10 terms)

E.g. “Apple iPod touch 32GB” -> Search for “Apple”, “iPod”, “touch”, “32GB”

### **Bonus (Extra):**

If you manage to finish the above task before time runs out and want to add some bonus points, consider either:

- Using the product description to more intelligently select useful search terms instead of simply breaking up the product title
- Add features which would improve the services scalability and performance
- Hosting your server somewhere so it can be accessed publicly