CPSC 5340 – Information Retrieval and Search
Assignment 2

In this assignment you will build a web application that will use SOLR to deliver search results and product and review details. You will build on your SOLR implementation from Assignment 1 and the code we built in Flask Lab, but there will be some differences, explained below.

Your website will have five page types:

1. A gateway page – a very simple page providing some text and a link to the search page.
2. Review search: search for product reviews using keywords.
3. Review search results: summary information about reviews that match the review search query from the review search page. The review search results page will also involve some new SOLR features: pagination, faceting, and query suggestion.
4. Review detail: more detailed information about a single review
5. Product detail: more detailed information about a product

Compared to what we did in the Flask Lab in class

1. The gateway page is new, but simple.
2. The review search page is like the search form we built in lab, but without the dropdowns to filter on date and score
3. The search results page plays the same role as the results page in lab, but has new features like pagination and faceting
4. The review and product detail pages are similar to the review summary page we did in lab – just display information about a single review or product

Documents – the new Review Data Set

We will move away from scraping reviews from web pages, and instead read review and product information from data files. There is one data file for review information and one data file for product information. These two files are joined on a a field *asin.* There is more information about the data files and their format in the Jupyter Notebook for this assignment. The data files and notebook are all in the Github repository.

Search Types

The search types for Assignment 2 are simpler – for example, we won't do special phrase search on the product name, and won't do filtering on review score or review data. There will only be three search/retrieval use cases

1. Search for reviews matching keywords – you can use standard SOLR text analyzers and keyword matching to do this search, and you should search on words in the review summary and the review body. (These fields are defined in the Jupyter Notebook.)
2. Lookup by review ID to get review details
3. Lookup by product ID (asin) to get product details

Review Information versus Product Information

A significant difference between Assignments 1 and 2 is that the data set for Assignment 2 has more information about products – for example the name, a description, the price, the sales rank, and browse categories. You need to store some of these product attributes to render a product detail page. You are not allowed to put product information for a review in the document with the review itself – that would mean that the product information might be stored redundantly thousands of times, which is an unacceptable waste of space.

To avoid this redundant storage, you will build two different and separate SOLR collections: one to store reviews and one to store products. The trick will be that sometimes to render a page your code will have to make two calls to SOLR, one to get review information and the second to get product information. That's no problem, remember that your SOLR service can have two active collections, and when you do a search request to SOLR you specify the collection to search against.

The most common example that you will see below is that the review search results page displays the product name. To get search results you first query against the reviews collection and get a list of "review documents." The review document has the product ID (ASIN), but not the product name, and the product name must appear in the review search result line (see mockup below). So then you need to make a second SOLR call to the products collection, keyed by ASIN, so you can get the product name, which you then display on the review search results page along with data about the review. This is a really common practice in composing pages in a search application – when Amazon renders a product detail page for example, it makes calls out to *many* different services to pull together all the information that is on the page.

Steps to Take

The assignment divides into two main parts

1. Parsing and indexing the review and product data. This is much like Assignment 1 except you will be reading data from files. For this assignment you will write only the functions that parse data from the product and review data files and for each creates a dictionary that can be passed to SOLR for indexing.
2. The web app that takes search requests and renders information about search results and review and product details. The deliverable for this phase is a Flask application that calls SOLR to get information about reviews and products, then renders that information.

Web Page Types

This document contains more detail about each of the page types. Each includes a screen shot – these screen shots indicate content only – you do not have to imitate the page layouts, in fact you are encouraged to make them more stylish.

Also, the data set I used to generate the screen shots is not the same as your data set, so do not expect search results to look exactly the same!

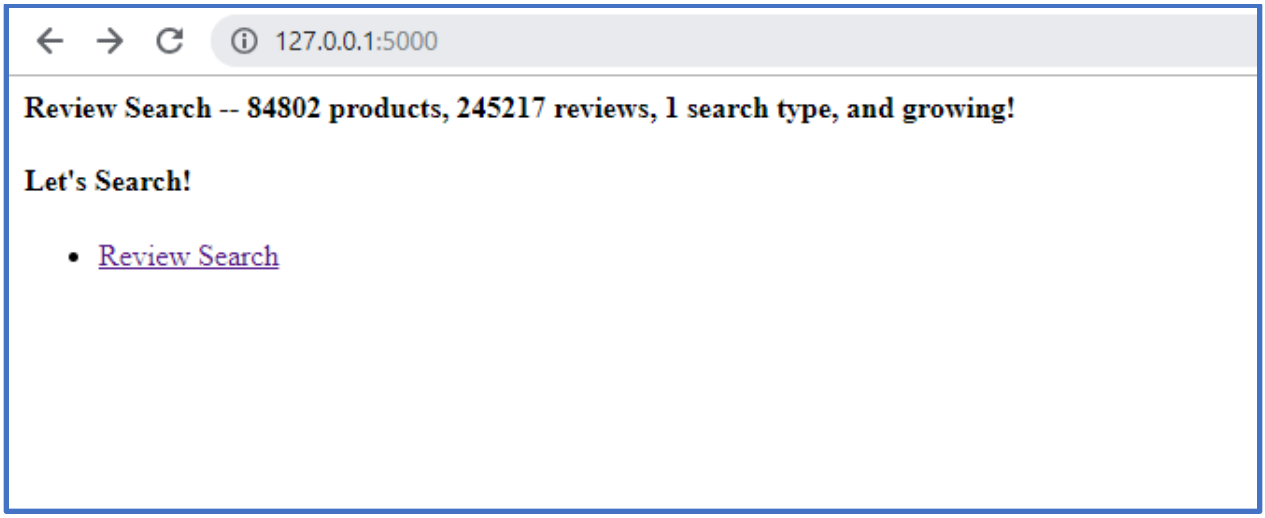
What to Hand In

You will hand in a Zip file containing these elements:

1. A Jupyter notebook containing the code that defines functions to parse the product and review data
2. Two folders containing configuration for your two SOLR collections. The names for these configuration directories and the SOLR collections are specified in the Jupyter Notebook.
3. A single folder named *reviewsite* containing your Flask application. Your application may assume that SOLR will be running on `localhost:8983` and that your document collections will have been created and indexed prior to starting Flask
4. A file retrospective.pdf, which must be a PDF file
    a. Your retrospective file must contain the following material
        i. Your name
        ii. Is your assignment fully working? If not, what pieces are working and what pieces are not
        iii. How much time did you spend on the assignment?
        iv. Overall how did you like the homework in terms of understanding and applying concepts covered in class

PAGE DETAILS

Gateway Page

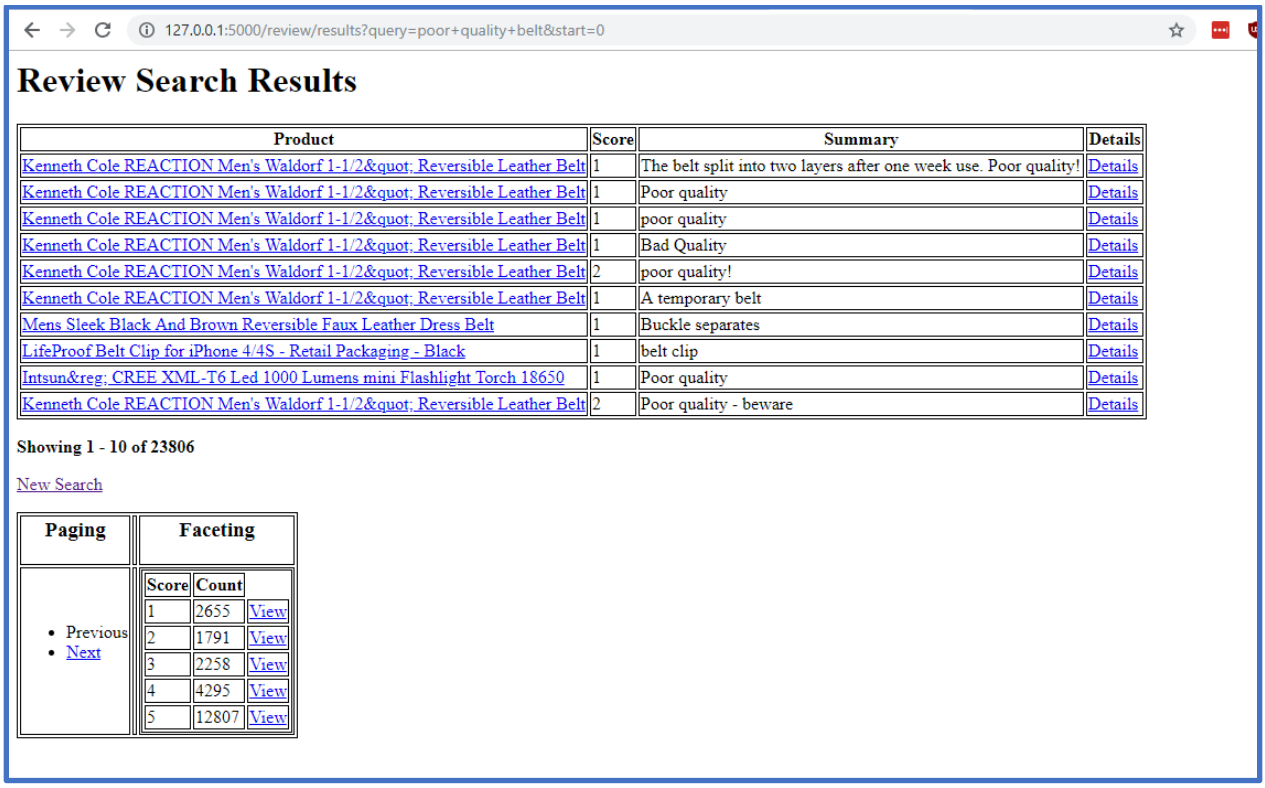


The counts on products and reviews are computed dynamically. The "1 search type" is hard-coded. The link is a link to the review search form.

Review Search Form

The form accepts keywords and tries to retrieve relevant review documents. Search is performed on words in the review's summary and body fields only. Indexing these fields as standard English language text and using the default SOLR query parser is fine.

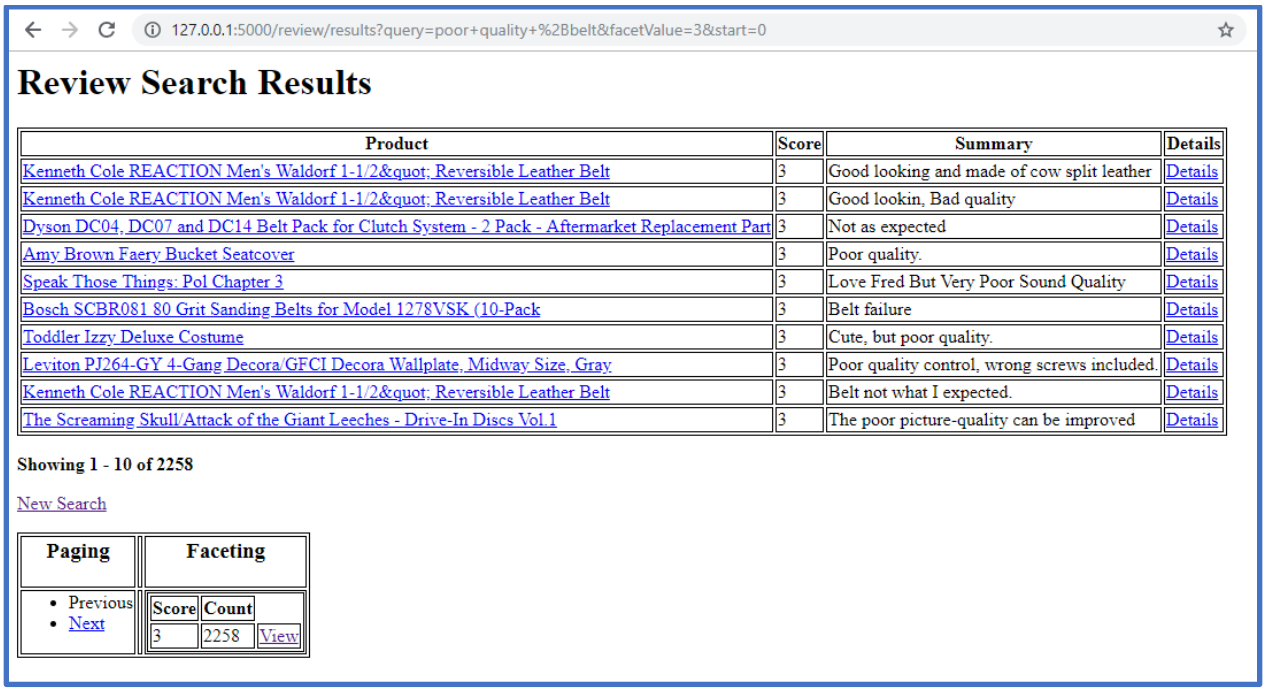
Search Results



This search was for "poor quality belt"

- Fields rendered are: the product's name, the review score, and the review summary
- The hyperlink on product name renders a product detail page for that product
- The hyperlink on Details renders a review detail page for that review
- Paging: notice that the "Previous" link is disabled because this page shows the first documents. Pressing the Next button shows results 11 – 20, and so on. When the last page of results is shown, the Next link is disabled. If there are fewer than 10 results, both Previous and Next links should be disabled
- Faceting: clicking through on the link shows only reviews for the same keywords with the specified review score (see next screen). The faceted results are paginated
- New Search returns to the search input form

Faceted Search Results



127.0.0.1:5000/review/results?query=poor+quality+%2Bbelt&facetValue=3&start=0

Review Search Results

Product	Score	Summary	Details
Kenneth Cole REACTION Men's Waldorf 1-1/2" Reversible Leather Belt	3	Good looking and made of cow split leather	Details
Kenneth Cole REACTION Men's Waldorf 1-1/2" Reversible Leather Belt	3	Good lookin, Bad quality	Details
Dyson DC04, DC07 and DC14 Belt Pack for Clutch System - 2 Pack - Aftermarket Replacement Part	3	Not as expected	Details
Amy Brown Faery Bucket Seatcover	3	Poor quality.	Details
Speak Those Things: Pol Chapter 3	3	Love Fred But Very Poor Sound Quality	Details
Bosch SCBR081 80 Grit Sanding Belts for Model 1278VSK (10-Pack	3	Belt failure	Details
Toddler Izzy Deluxe Costume	3	Cute, but poor quality.	Details
Leviton PJ264-GY 4-Gang Decora/GFCI Decora Wallplate, Midway Size, Gray	3	Poor quality control, wrong screws included.	Details
Kenneth Cole REACTION Men's Waldorf 1-1/2" Reversible Leather Belt	3	Belt not what I expected.	Details
The Screaming Skull/Attack of the Giant Leeches - Drive-In Discs Vol.1	3	The poor picture-quality can be improved	Details

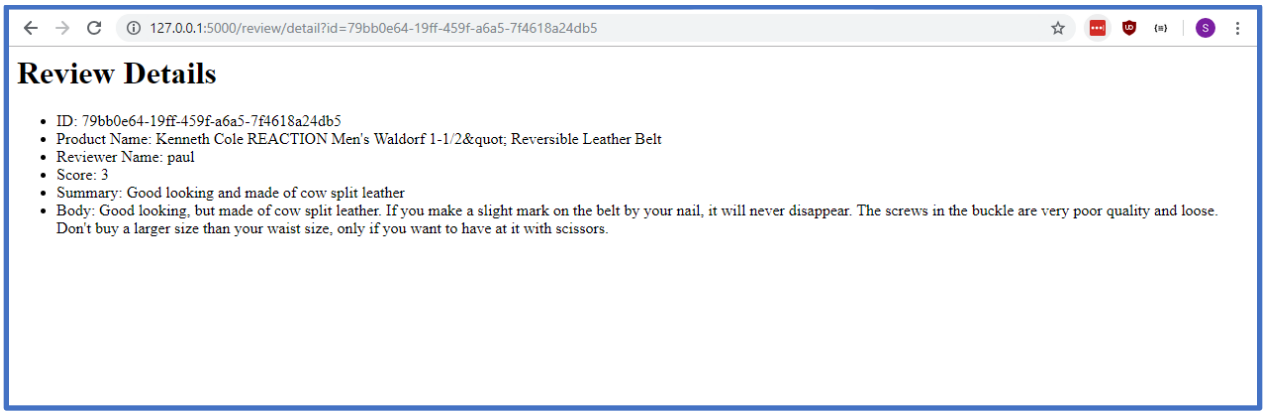
Showing 1 - 10 of 2258

New Search

Paging	Faceting		
• Previous • Next	Score	Count	
	3	2258	View

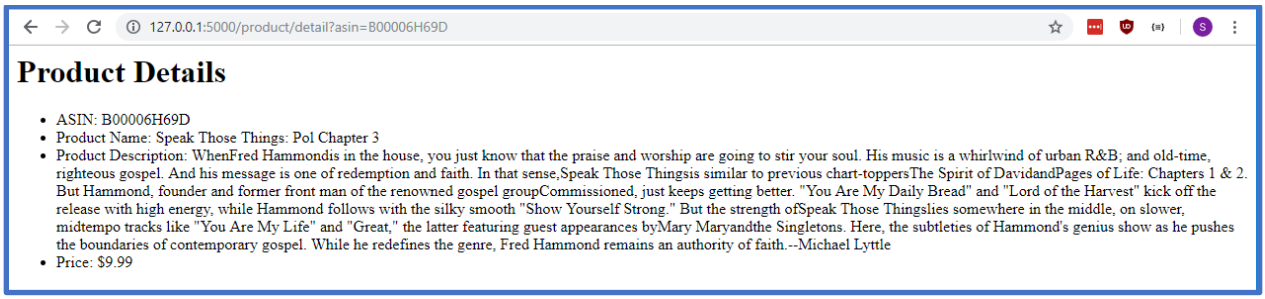
This is the result of clicking through on the facet for review score 3.

Review Details



This is the result of clicking the **Detail** link on the first search result.

Product Details



This is the result of clicking on the product name link *Speak Those Things: Pol Chapter 3* on the previous search results page.