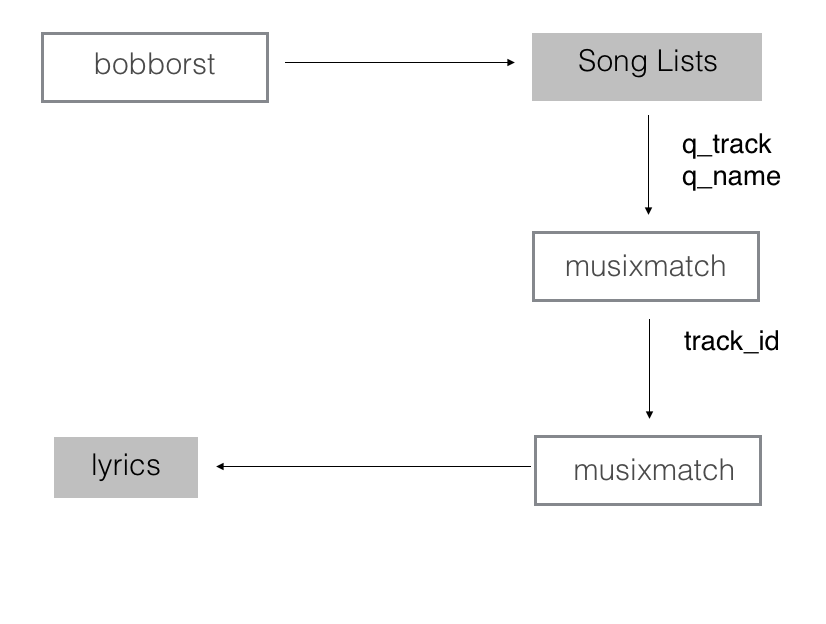
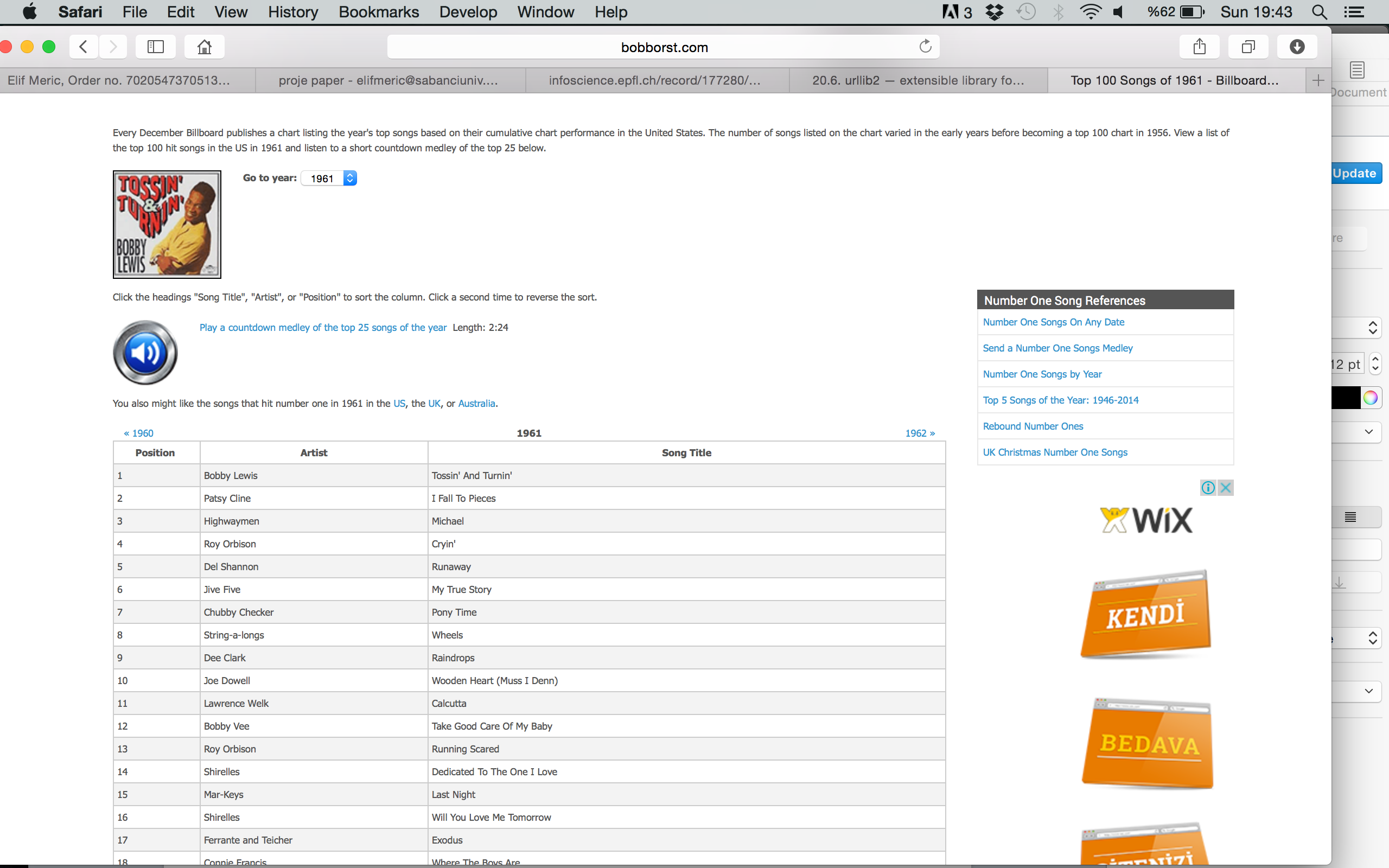
1. **Project Design and Implementation**

**3.1 Project Design Data Collection**

**3.1.1 Musixmatch API**

*Figure 1 : Schematic of Data Collection Process*

Since our objective to get lyrics between 1947-2014, first we found a website that contains a lists of top 100 : http://www.bobborst.com (Figure 2). This website provides us same top 100 list as official Billboard website but more flexible. This website is more easy to parse and get the datas.

Figure 2 : Bobborst Top 100 / 1961

After list’s collected, we need to find these songs’ lyrics . Web services create APIs to external applications can collect them on websites. There are many useful websites for finding lyrics that provides API format like <https://developer.musixmatch.com/>. Musixmatch provides interface to get wanted data. These API methods provided by Musixmatch that we have used :

1. TRACK.SEARCH : we have used to search every song in our database.   
Needed Parameters : *q\_track* : track name, *q\_artist* : artist name

track.search?q\_track=*songname*&q\_artist=*artistname*&f\_has\_lyrics=1

Since we’ve got list of song names and artist names from [bobborst.com](http://bobborst.com) , we can search to find out if there is available lyrics. If response has available lyrics, we parsed the JSON response to get track\_id in order to get lyrics.

2. TRACK.LYRICS.GET: We’ve used to get the lyrics of the tracks on our lists.   
Needed Parameters : *track\_id*

track.lyrics.get?track\_id=*trackid*

Since we’ve got list of track ids for each year’s top 100 songs, we can finally get the lyrics from [musixmatch.com](http://musixmatch.com) . We stored lyrics year by year to search listed words.

After we get all the lyrics we parsed it by word count. So we can see word count year by year in descending order.

**3.1.2 Million Song Dataset**

The Million Song Dataset is a free collection of metadata for a million music tracks. They provide a subset which has 10.000 songs and it’s track\_id based on The Echo Nest API, mxm\_id based on Musixmatch API, words and other features that we may want to use. Also they provide an input file which includes 515.576 tracks of the dataset’s year information.

Once we get the SQLite database which contains song’s metadata and the year prediction input file, we have chance to observe and analyse words year by year and song by song. We have three tables in our database : lyrics, words and year\_song which we created using the input text file. We turned year information input file into CSV format and import to SQLite so we can manipulate and use the data. Table *lyrics* has 5 columns , we used 4 of 5 : *track\_id* ( provided by the Echo Nest API) , *mxm\_tid* ( provided by Musixmatch API) , *word* , *count*. The table *words* has only one column and it has primary key which named *word*. We used this primary key to get word count for all. The table *year\_song* has 4 columns : *year*, *track\_id* which is the primary key and we used this by joining *lyrics* table, *artist* which stands for artist name and *song* which is represents song name. We join these tables and get the word count by grouping by year.

**3.2 Tools Used**

**Canopy :** Enthought Canopy is Python development and analysis environment that provides easy scientific analysis and visualisation.

**Beautiful Soup :** Beautiful Soup is a Python library that designed for parsing HTMLs.

**URLLIB2 :** Library that defines functions and classes that make easier to open URLs.

**RazorSQL** : is an SQL query tool, database browser, SQL editor, and database administration tool.

**GitHub :** Website that where people build software and share. GitHub provides revision control system , that gives opportunity to control web-based projects development process and its history.

( ENS492 Project : [**https://github.com/elifmeric/ENS492**](https://github.com/elifmeric/ENS492) )

References

1.Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere.

The Million Song Dataset. In Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011), 2011.