**Sumit Parwal**

**5174593050**

**Website Used: Guardians.com**

**Report on Autocomplete, Spell Correct and Snippets:**

**Step:1 Spell Correct:**

The spell-correction functionality was implemented in two parts:

**1. Firstly I created a  of big.txt file** using the Guardians folder uploaded on drive and using Apache Tika in Java.

Using the Apache Tika jars I extracted the html content of all webpages and created

the big.txt using regular expression in Java. This list was used in Peter Norvig’s spell-

correction program

**2. Using Peter Norvig’s spell correction**

a. Here, we calculate all the possible words from an edit distance of 1 and 2. The

operations used to calculate the possible words are:

i. Insert

ii. Delete

iii. Transposition

iv. Substitution/Replace

b. From the words calculated, we return the most probable word from the corpus

For this we include the spell corrector program using “include SpellCorrctor.php” and then make a

call to the corrector() function using “echo SpellCorrector::correct('califonia');”(‘califonia is used as an

example here)

**Step 2:Autocomplete:**

* Autocomplete was implemented using multiple technologies. I used javascript and jquery & ajax’s autocomplete for this purpose.
* Autocomplete requires a source containing the terms that will be provided for suggestions in the drop-down menu
* For Autocomplete suggestions we made changes to the solrconfig.xml by telling solr to use the SuggestComponent. We then add a requestHandler to allowing to configure default parameters for suggestion requests. This was used to build the ‘source’.
* To ensure autocomplete works for n-term queries, we split the query terms based on space.(Assuming queries are separated by space) Autocomplete suggests for the latest term being entered returning an autocomplete for the last term tagged to all the initial query terms. These suggestions were finally stored in an array.
* The array was returned as a responseText to the javascript.
* The auto-complete results are clickable, will auto-complete the word when clicked and return focus to the text box.
* Once the user starts with a new word or a space separated word the ajax code is again activated.

**Step 3: Snippet**

• To enable this functionality, I have included simple\_html\_dom.php file in my ranking.php file. simple\_html\_dom.php is used to parse html pages of NBC\_News and generate the text out of it.

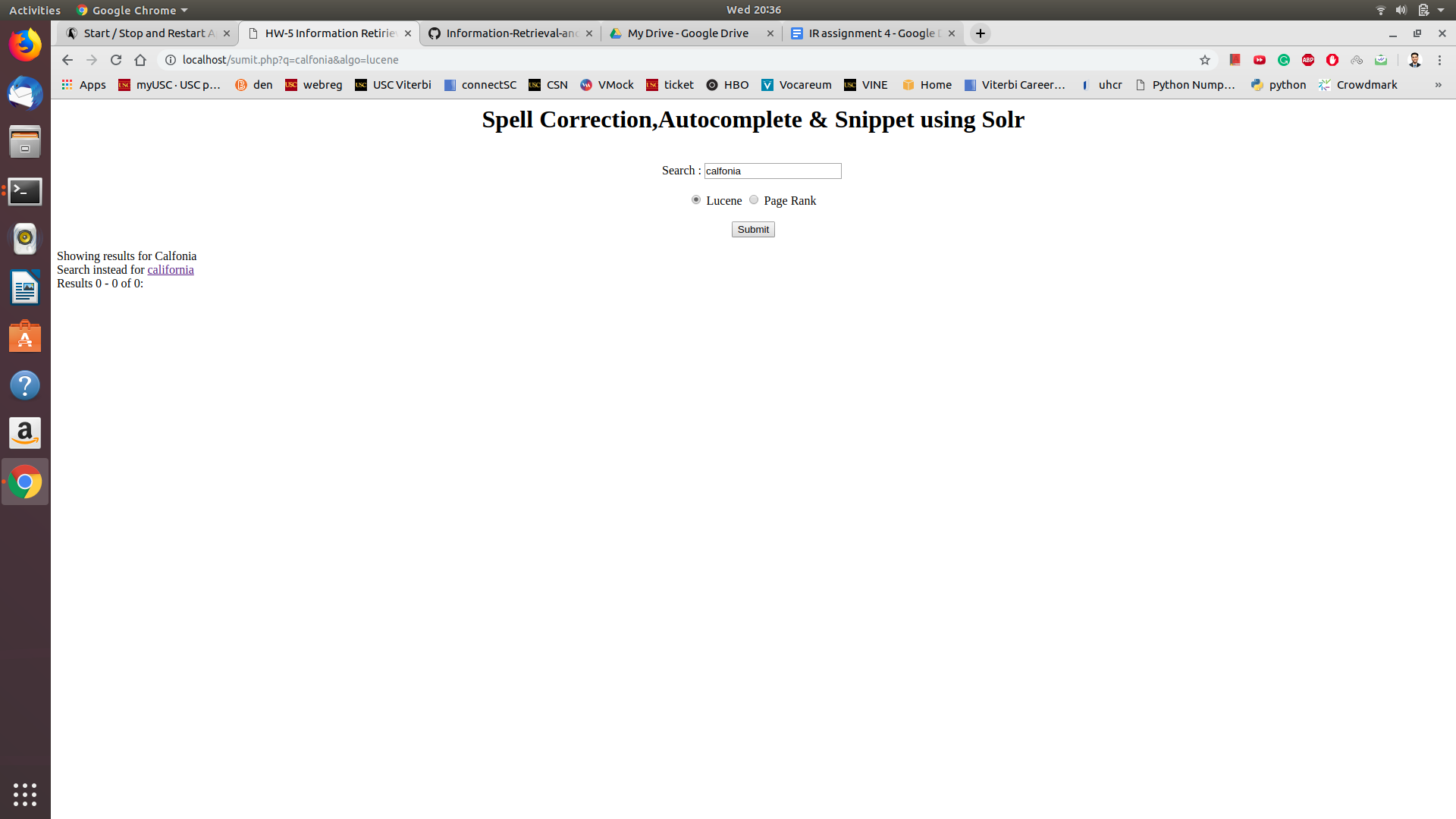
• I used file\_get\_contents function that extracted the contents of the given filename. Then I divided the content into sentences and words. Also removes the special characters so that they don’t affect in keywords matching.

• I run two loops, one for sentences and other for words to find the query terms from the contents of the given filename and find the index of the terms in the string that is used to print that part string along with ellipses (…) either at the beginning or at the end of the snippet.

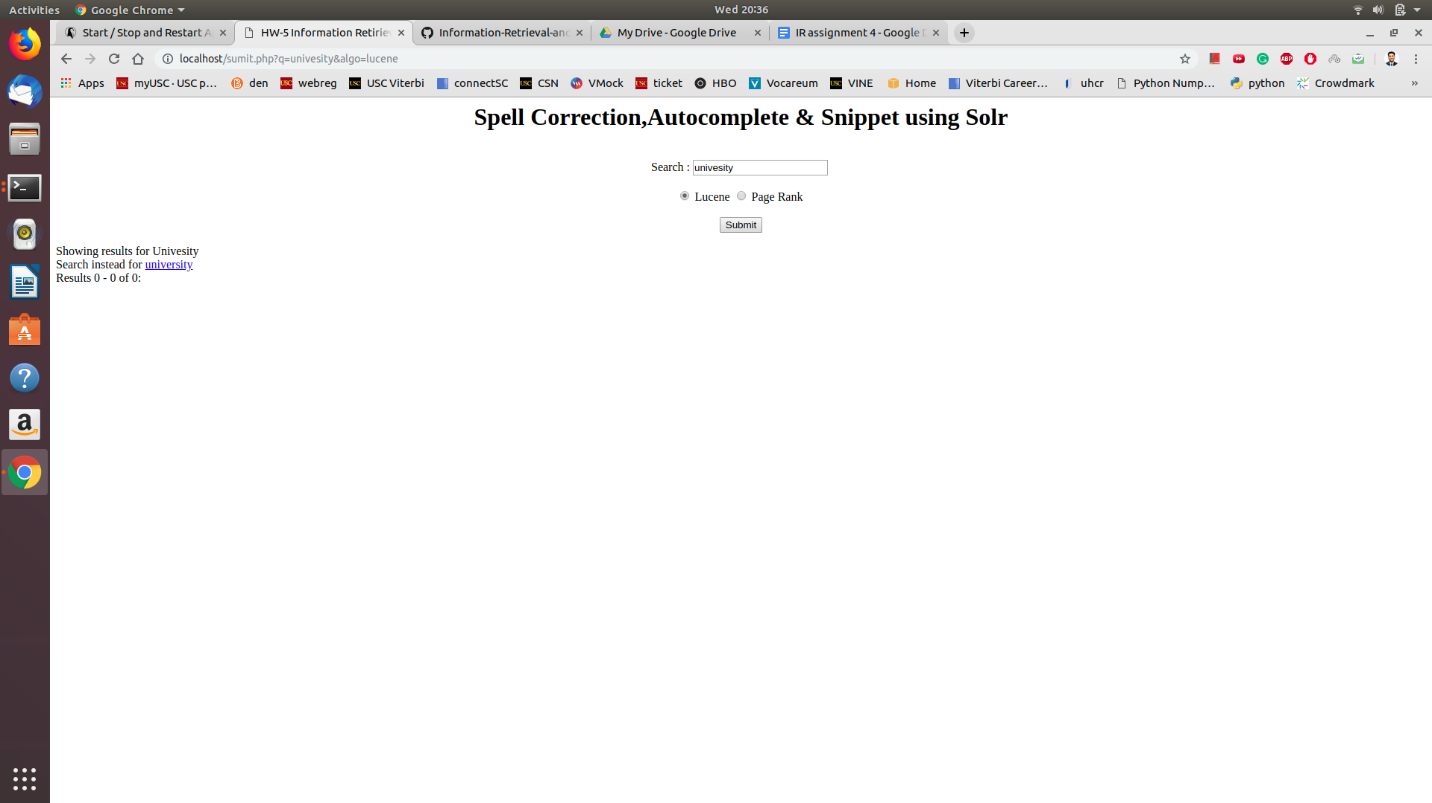
• If a snippet is found, then I displayed the snippet with bold query terms and if snippet is not found then I displayed N/A

**ANALYSIS OF RESULTS Examples of Spell Correction & Snippet**

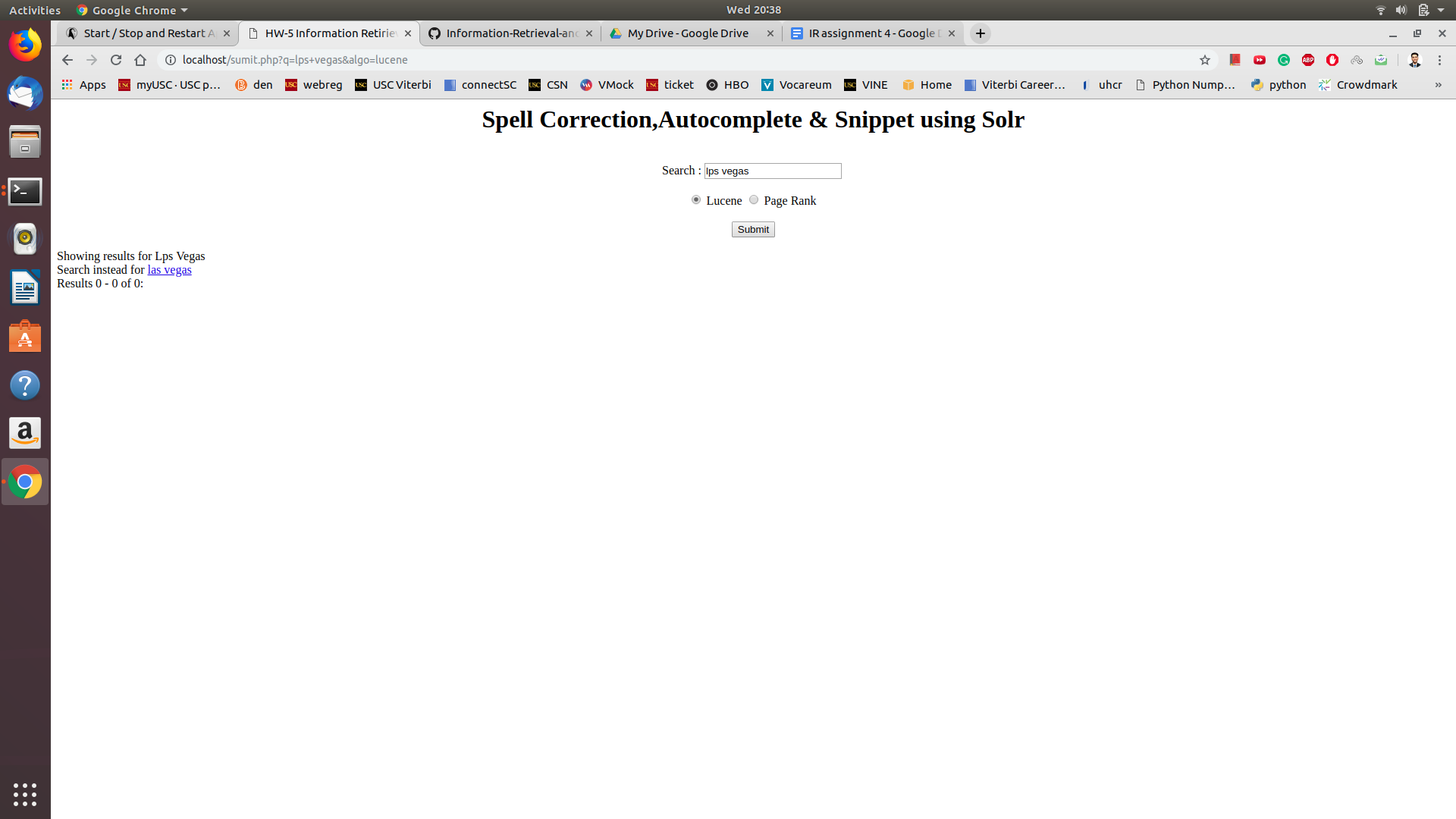
1. **Califonia**



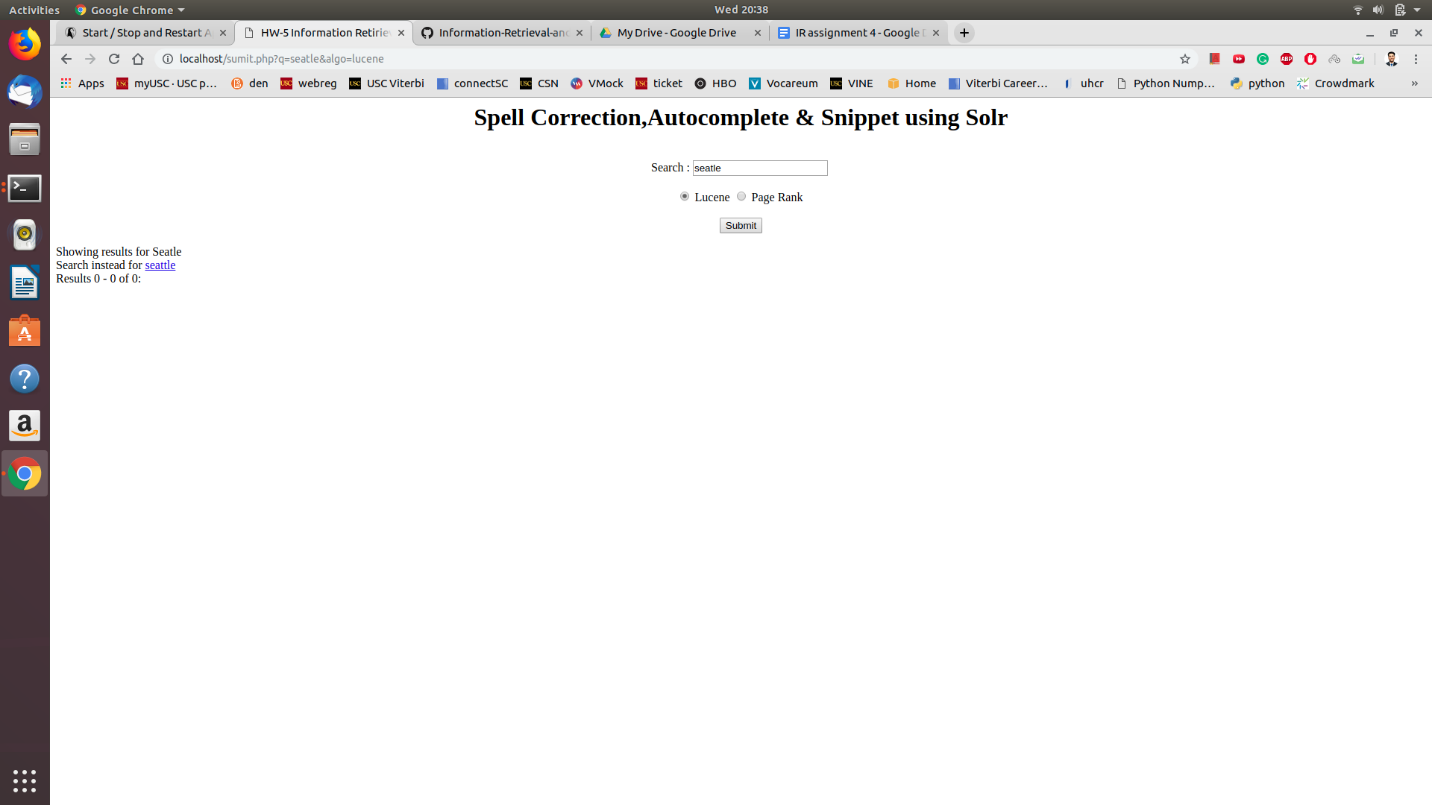
1. **Univesity**



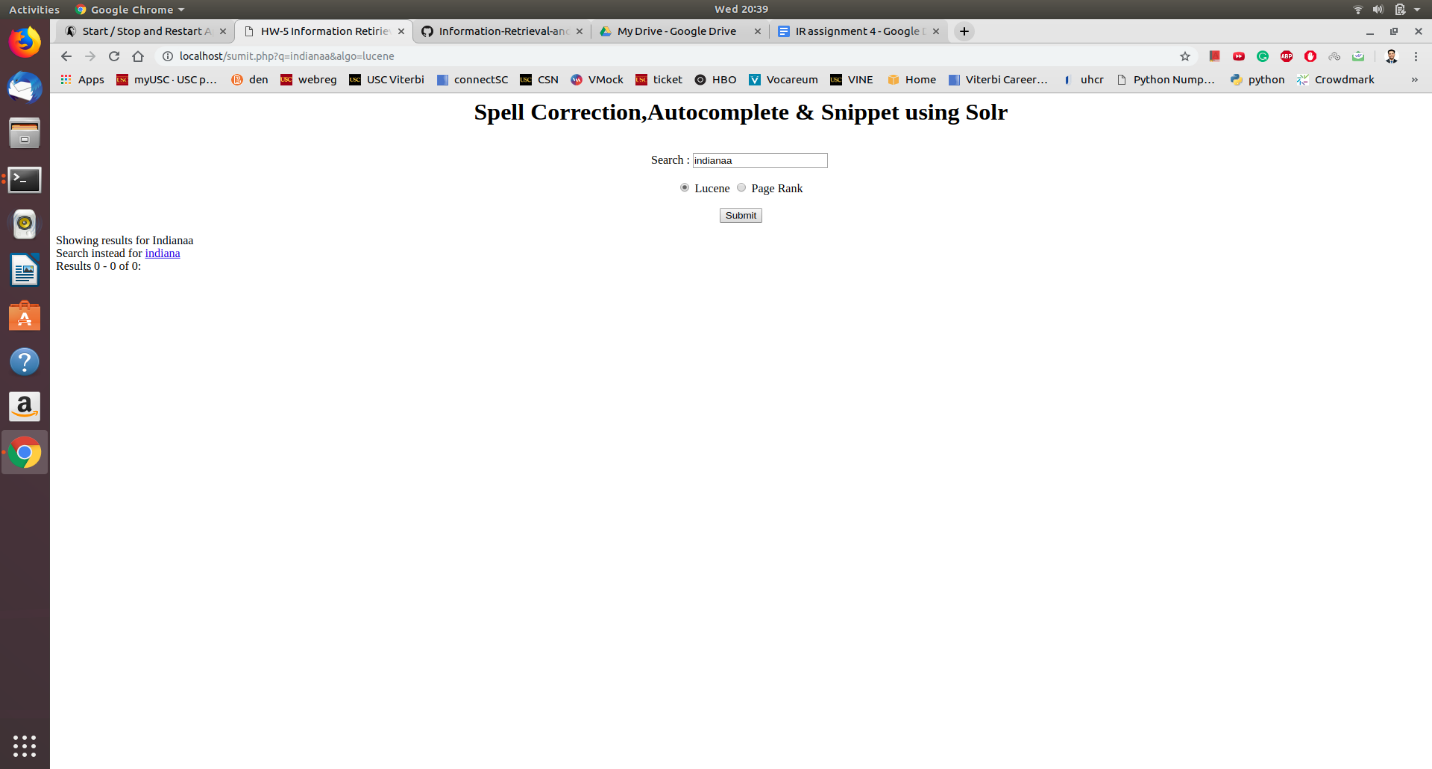
1. **Lps Veagas**



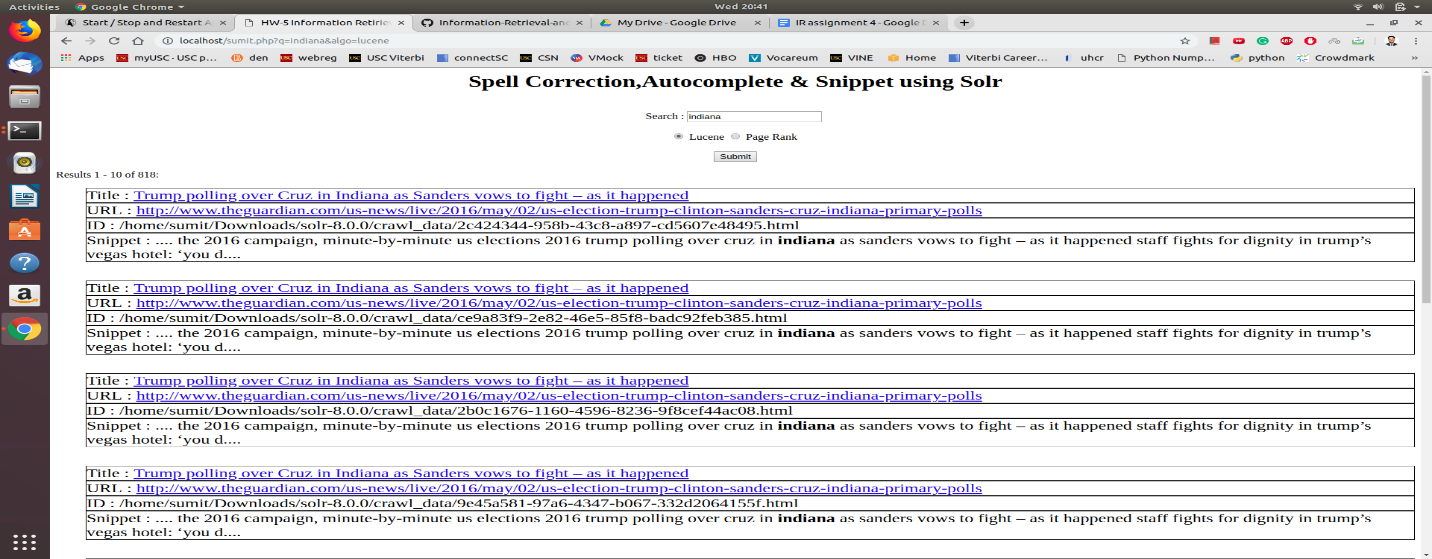
1. **Seatle**



1. **Indianaa**

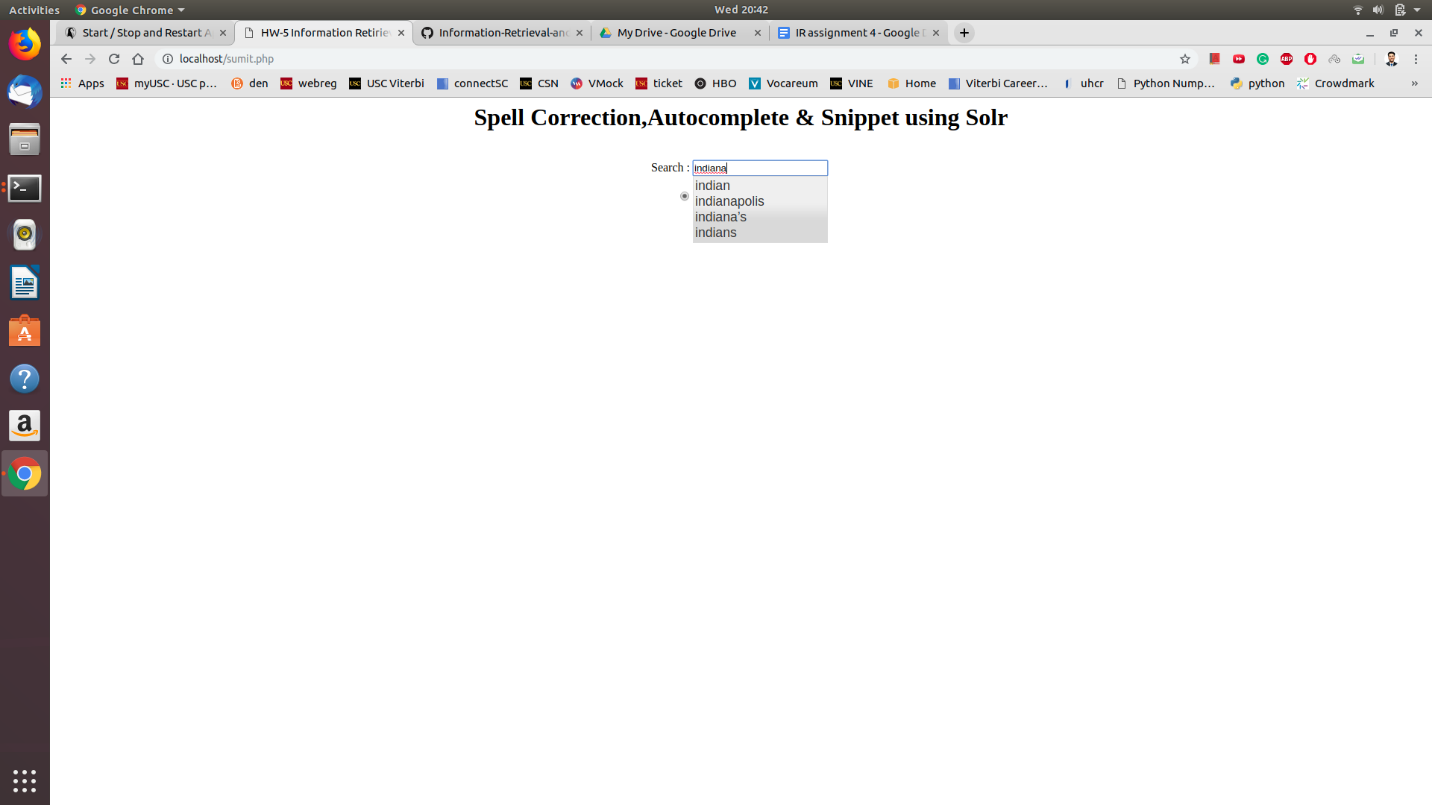


**Result when you click the Indiana Link**

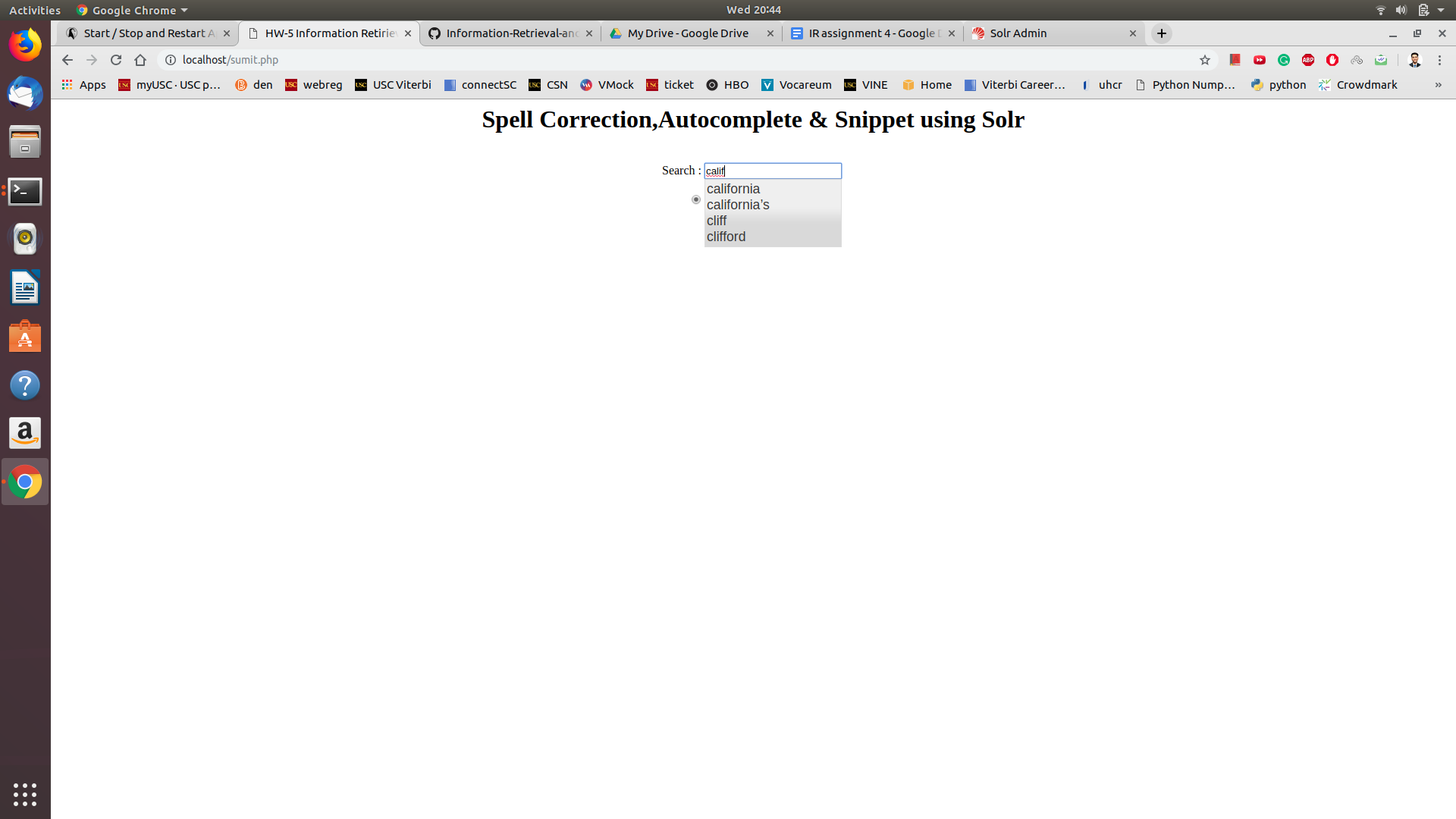


**ANALYSIS OF RESULTS Examples of Autocomplete**

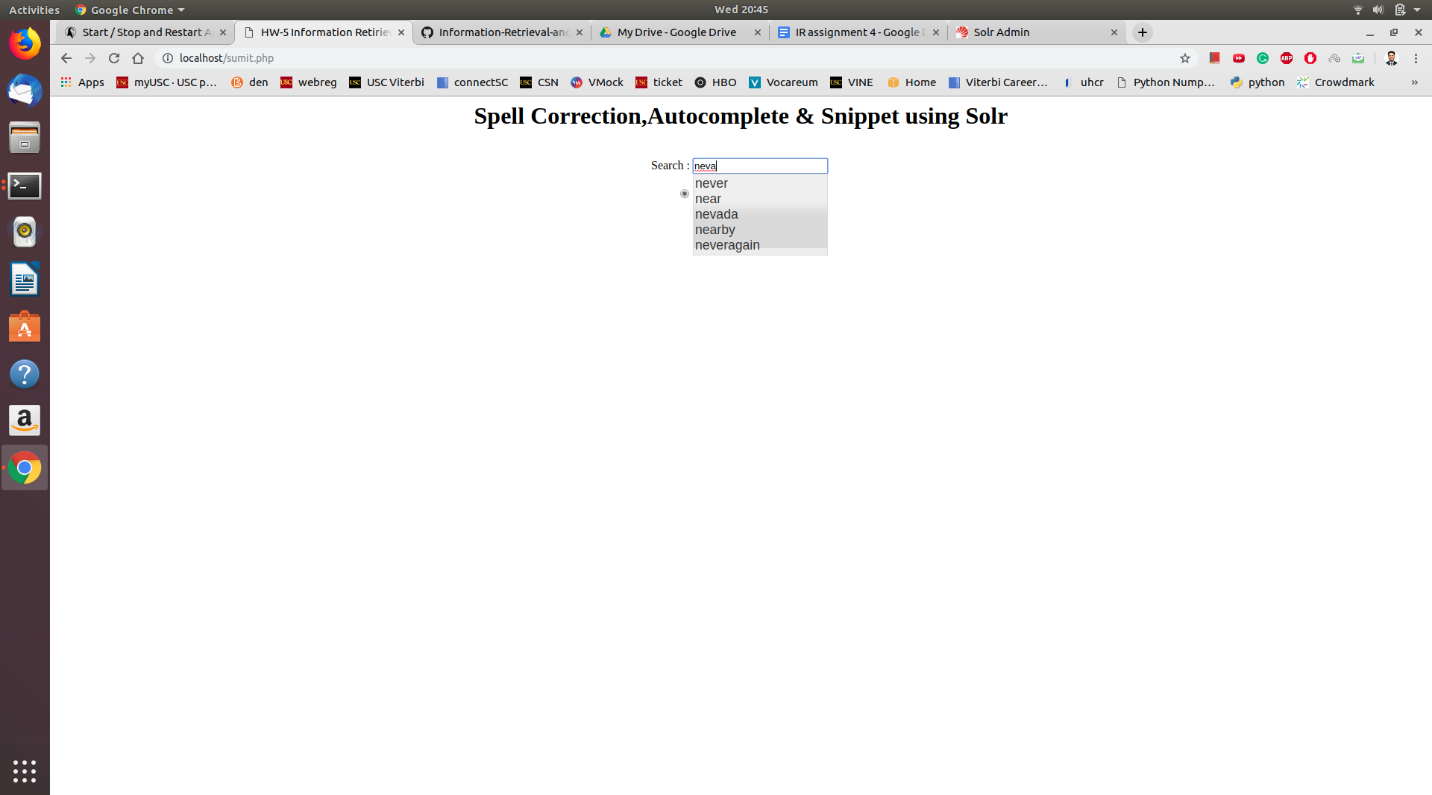
**1.Indiana**



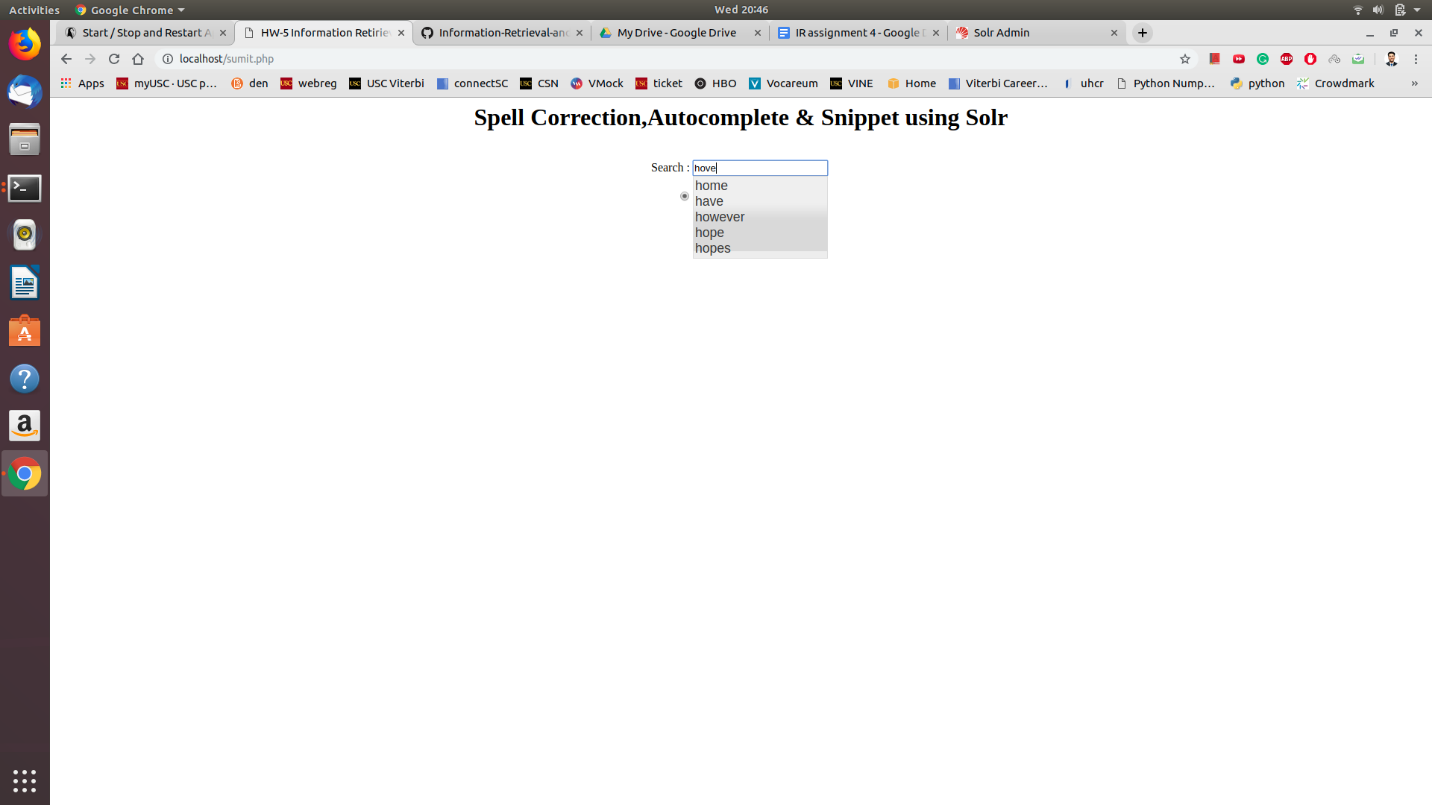
**2. Calif**



**3. Neva**



**4. Hove**



**5.** **Sect**