Queries:

Query1:Query to retrieve usernames when a keyword is entered:

sparql = SPARQLWrapper("http://18.222.233.173:3030/db1/sparql")

sparql.setQuery(

"""PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX loc: <http://www.semanticweb.org/tarunsnehithkishorereddykarna/ontologies/2020/10/untitled-ontology-16#>

PREFIX use: <http://www.semanticweb.org/anushakolan/ontologies/2020/10/untitled-ontology-14#>

PREFIX det: <http://www.semanticweb.org/tarunsnehithkishorereddykarna/ontologies/2020/10/untitled-ontology-15#>

PREFIX owl: <http://www.w3.org/2002/07/owl#>

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?id ?name

WHERE {

?person rdf:type use:User;

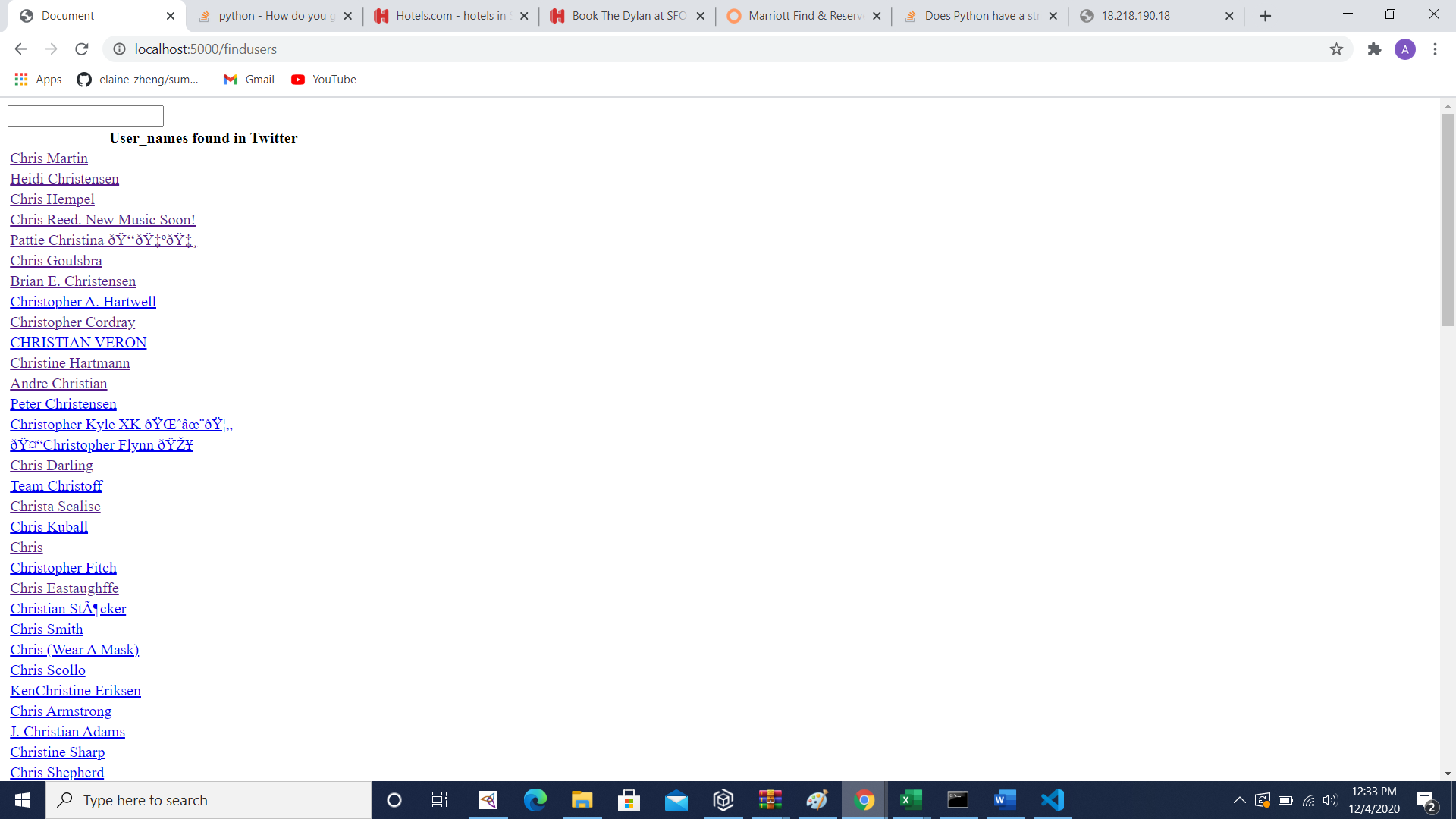
use:has\_user\_name ?name;

use:has\_screen\_name ?id.

FILTER( regex(?name, '%s', "i" ) )

}LIMIT 100""" % (user))

Output:



Query2:Query to retrieve the store details near the user

sparql = SPARQLWrapper("http://18.218.190.18:3030/db1/sparql")

sparql.setQuery(

"""PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX loc: <http://www.semanticweb.org/tarunsnehithkishorereddykarna/ontologies/2020/10/untitled-ontology-16#>

PREFIX use: <http://www.semanticweb.org/anushakolan/ontologies/2020/10/untitled-ontology-14#>

PREFIX det: <http://www.semanticweb.org/tarunsnehithkishorereddykarna/ontologies/2020/10/untitled-ontology-15#>

PREFIX owl: <http://www.w3.org/2002/07/owl#>

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?name ?rating ?street ?city ?state ?postal ?open ?cate

WHERE {

SERVICE <http://18.216.61.64:3030/db1/query>

{

SELECT ?store ?elat ?elong ?id

WHERE {

SERVICE <http://18.222.233.173:3030/db1/sparql>

{

SELECT ?name ?lat ?lon

WHERE {

?name rdf:type use:User;

use:has\_screen\_name '%s';

use:has\_latitude ?lat;

use:has\_longitude ?lon.

}

}

?store rdf:type loc:Store.

?store loc:has\_BusinessID ?id.

?store loc:has\_Location ?loc.

?loc loc:has\_Latitude ?elat.

?loc loc:has\_Longitude ?elong.

FILTER((abs(xsd:float(?lat) - xsd:float(?elat)) < 5) && (abs(xsd:float(?lon) - xsd:float(?elong)) < 5))

}

ORDER BY ((abs(xsd:float(?lat) - xsd:float(?elat))) && (abs(xsd:float(?lon) - xsd:float(?elong))))

LIMIT 100

}

?st rdf:type det:Store.

?st det:has\_Business\_ID ?id.

?st det:has\_name ?name.

?st det:has\_Rating ?rating.

?st det:has\_Address ?ad.

?ad det:has\_City ?city.

?ad det:has\_State ?state.

?ad det:has\_StreetNo ?street.

?ad det:has\_PostalCode ?postal.

?st det:has\_IsOpen ?open.

?st det:has\_Catogories ?cate.

}

#ORDER BY DESC (xsd:float(?rating))

LIMIT 20""" % (p))

Output:

