VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

GDG – 2 CREDIT COURSE REPORT

Name – Anirudh Batra Reg No – 17BCE2145

Topic –

SCRAP THE GSOC WEBSITE

Abstract-

This project uses flask which is a python based microframework to create the API and we use beautiful soup4 to scrape the website and bring out the important information that is needed.

Inroduction-

Creating an API using flask framework and scraping the GSOC website to fetch details of all the organisations in Google summer of code and in the API send their name, link to their website, description, the technologies they use and their contact email.

Methodology-

```python
import requests
import json
from bs4 import BeautifulSoup as bsp
from flask import Flask

main_org_url = 'https://summerofcode.withgoogle.com/archive/2017/organizations/'
base_url = "https://summerofcode.withgoogle.com"
app = Flask(__name__)
```

In this part of the code we have imported all the libraries that were needed for this project. Also we have defined the URLs required.

```python
def org_info():
    fetch = requests.get(main_org_url)
    html = fetch.content
    b_soup1 = bsp(html, "html.parser")
    fetch_org = b_soup1.findAll("li", {'class': 'organization-card__container'})
    final_result = get_details(fetch_org)
    print(final_result)
    return json.dumps(final_result)

```

In this part we access the gsoc website and fetch the details by putting a request and storing it in fetch variable. Then we scrape it using beautifulsoup and save it in b_soup1(Name of the organisation) and we store the data i he final_result by calling the function get_details which will be defined in the next segment.

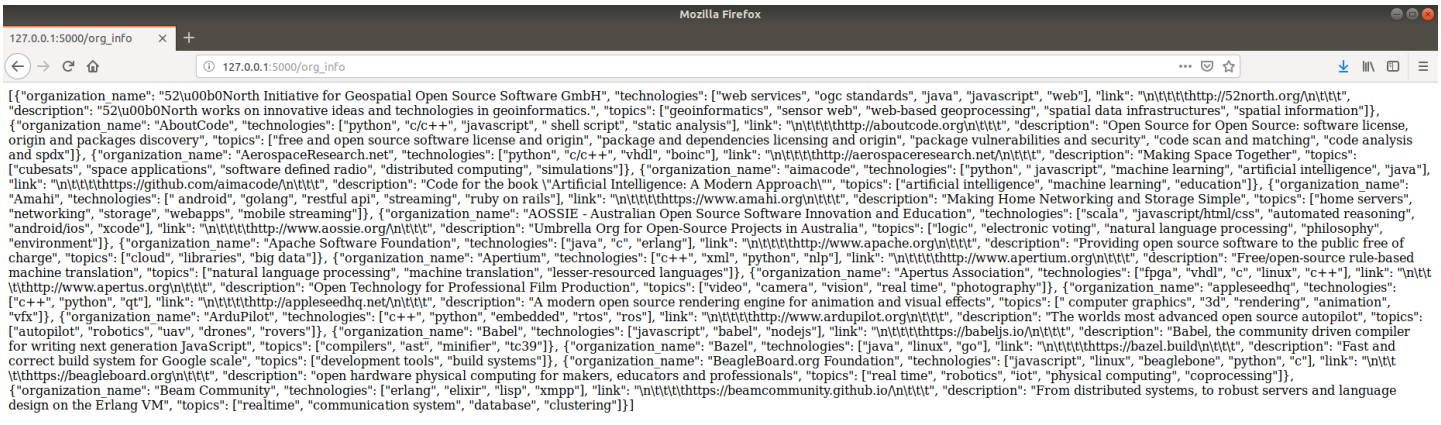
```python
def get_details(fetch_org):
    extracted_result = list()
    count = 0
    for item in fetch_org:
        purl = item.find('a', {'class': 'organization-card__link'})
        organisation_name = item['aria-label']

        information = item.find('div', {'class': 'organization-card__tagline font-black-54'})
        information = information.text
        p_link = base_url + purl['href']
        page = requests.get(p_link)
        if page.status_code != 200:
            break
        p_link = base_url + purl['href']
        response1 = requests.get(p_link)
        html1 = response1.content
        b_soup2 = bsp(html1, "html.parser")
        organisation_link = b_soup2.find("a", {"class": "org__link"})
        organisation_link = organisation_link.text
        tech_info = b_soup2.findAll("li", {"class": "organization__tag organization__tag--technology"})
        technology = []
        for t_tech in tech_info:
            technology.append(t_tech.text)
        t_topics = b_soup2.findAll("li", {"class": "organization__tag organization__tag--topic"})
        topics = []
        for i in t_topics:
            topics.append(i.text)
        count += 1
        print(count)
        extracted_result.append({
            'organization_name': organisation_name,
            'description': information,
            'link': organisation_link,
            'technologies': technology,
            'topics': topics
        })
    #Extracting Only 15 organization details Change below count value to get details of more.
        if count == 15:
            return extracted_result

```

In the function get_details first of we have stored the result in list format to the variable extracted_result. Now we have started a loop for extracting the organisation details which include the technology, link to their home site and the description of the company. The result is in Json format we need to convert it to dictionary format.

Result-

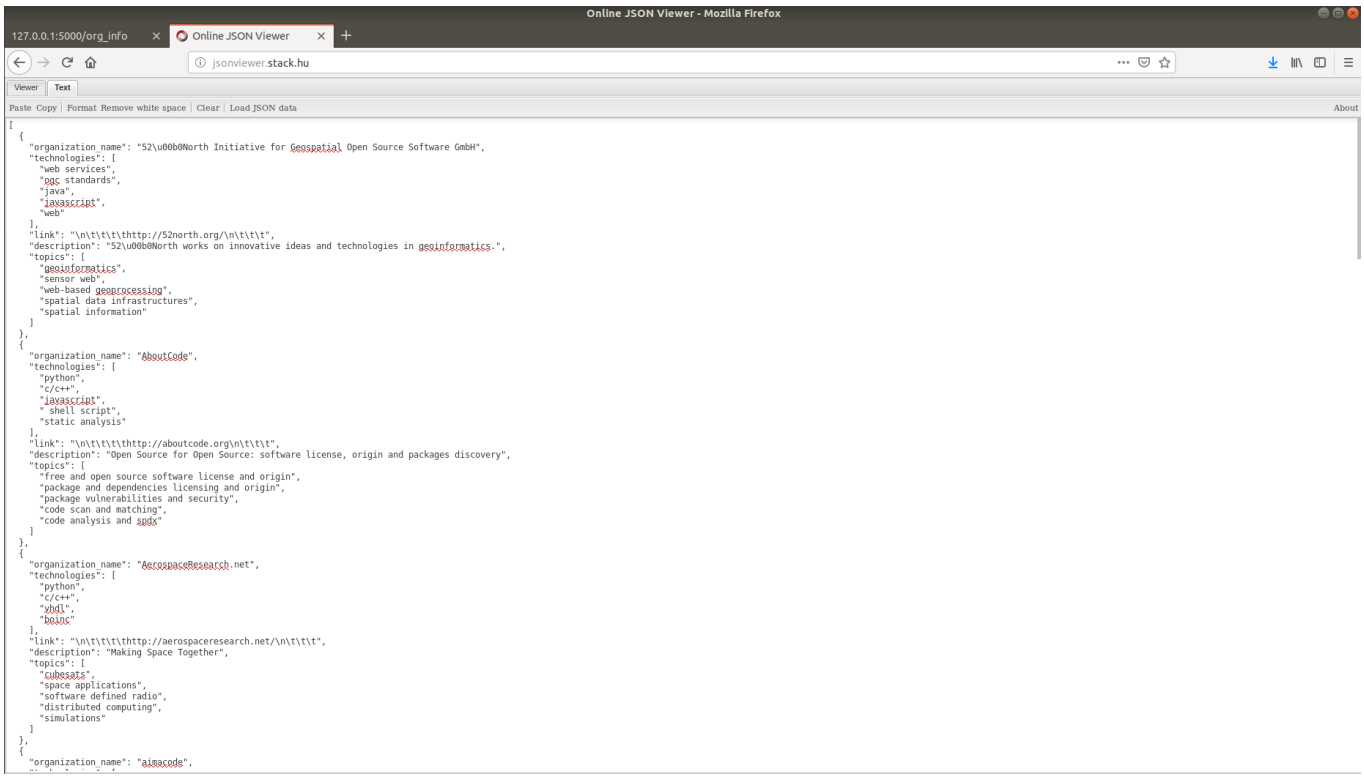
```
anirudh@the3d3n:~/Downloads/pypy$ python gdg_gsoc1.py
 * Serving Flask app "gdg_gsoc1" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 299-845-447
127.0.0.1 - - [24/Mar/2019 19:48:16] "GET / HTTP/1.1" 404 -
1
2
[{'organization_name': u'52\xb0North Initiative for Geospatial Open Source Software GmbH', 'technologies': [u'web services', u'ogc standards', u'java', u'javascript', u'web'], 'link':
 u'\n\t\t\t\thttp://52north.org/\n\t\t\t', 'description': u'52\xb0North works on innovative ideas and technologies in geoinformatics.', 'topics': [u'geoinformatics', u'sensor web', u'
web-based geoprocessing', u'spatial data infrastructures', u'spatial information']}, {'organization_name': u'AboutCode', 'technologies': [u'python', u'c/c++', u'javascript', u' shell
script', u'static analysis'], 'link': u'\n\t\t\t\thttp://aboutcode.org\n\t\t\t', 'description': u'Open Source for Open Source: software license, origin and packages discovery', 'topic
s': [u'free and open source software  license and origin', u'package and dependencies licensing and origin', u'package vulnerabilities and security', u'code scan and matching', u'code
 analysis and spdx']}]
127.0.0.1 - - [24/Mar/2019 19:48:29] "GET /org_info HTTP/1.1" 200 -
1
2
[{'organization_name': u'52\xb0North Initiative for Geospatial Open Source Software GmbH', 'technologies': [u'web services', u'ogc standards', u'java', u'javascript', u'web'], 'link':
 u'\n\t\t\t\thttp://52north.org/\n\t\t\t', 'description': u'52\xb0North works on innovative ideas and technologies in geoinformatics.', 'topics': [u'geoinformatics', u'sensor web', u'
web-based geoprocessing', u'spatial data infrastructures', u'spatial information']}, {'organization_name': u'AboutCode', 'technologies': [u'python', u'c/c++', u'javascript', u' shell
script', u'static analysis'], 'link': u'\n\t\t\t\thttp://aboutcode.org\n\t\t\t', 'description': u'Open Source for Open Source: software license, origin and packages discovery', 'topic
s': [u'free and open source software  license and origin', u'package and dependencies licensing and origin', u'package vulnerabilities and security', u'code scan and matching', u'code
 analysis and spdx']}]
127.0.0.1 - - [24/Mar/2019 19:48:50] "GET /org_info HTTP/1.1" 200 -
```

This is the terminal output of the code



[{"organization_name": "52\u00b0North Initiative for Geospatial Open Source Software GmbH", "technologies": ["web services", "ogc standards", "java", "javascript", "web"], "link": "\n\t\t\t\thttp://52north.org/\n\t\t\t", "description": "52\u00b0North works on innovative ideas and technologies in geoinformatics.", "topics": ["geoinformatics", "sensor web", "web-based geoprocessing", "spatial data infrastructures", "spatial information"]}, {"organization_name": "AboutCode", "technologies": ["python", "c/c++", "javascript", " shell script", "static analysis"], "link": "\n\t\t\t\thttp://aboutcode.org\n\t\t\t", "description": "Open Source for Open Source: software license, origin and packages discovery", "topics": ["free and open source software license and origin", "package and dependencies licensing and origin", "package vulnerabilities and security", "code scan and matching", "code analysis and spdx"]}, {"organization_name": "AerospaceResearch.net", "technologies": ["python", "c/c++", "vhdl", "boinc"], "link": "\n\t\t\t\thttp://aerospaceresearch.net/\n\t\t\t", "description": "Making Space Together", "topics": ["cubesats", "space applications", "software defined radio", "distributed computing", "simulations"]}, {"organization_name": "aimacode", "technologies": ["python", "javascript", "machine learning", "artificial intelligence", "java"], "link": "\n\t\t\t\thttps://github.com/aimacode/\n\t\t\t", "description": "Code for the book \"Artificial Intelligence: A Modern Approach\"", "topics": ["artificial intelligence", "machine learning", "education"]}, {"organization_name": "Amahi", "technologies": [" android", "golang", "restful api", "streaming", "ruby on rails"], "link": "\n\t\t\t\thttps://www.amahi.org\n\t\t\t", "description": "Making Home Networking and Storage Simple", "topics": ["home servers", "networking", "storage", "webapps", "mobile streaming"]}, {"organization_name": "AOSSIE - Australian Open Source Software Innovation and Education", "technologies": ["scala", "javascript/html/css", "automated reasoning", "android/ios", "xcode"], "link": "\n\t\t\t\thttp://www.aossie.org/\n\t\t\t", "description": "Umbrella Org for Open-Source Projects in Australia", "topics": ["logic", "electronic voting", "natural language processing", "philosophy", "environment"]}, {"organization_name": "Apache Software Foundation", "technologies": ["java", "c", "erlang"], "link": "\n\t\t\t\thttp://www.apache.org\n\t\t\t", "description": "Providing open source software to the public free of charge", "topics": ["cloud", "libraries", "big data"]}, {"organization_name": "Apertium", "technologies": ["c++", "xml", "python", "nlp"], "link": "\n\t\t\t\thttp://www.apertium.org\n\t\t\t", "description": "Free/open-source rule-based machine translation", "topics": ["natural language processing", "machine translation", "lesser-resourced languages"]}, {"organization_name": "Apertus Association", "technologies": ["fpga", "vhdl", "c", "linux", "c++"], "link": "\n\t\t\t\thttp://www.apertus.org\n\t\t\t", "description": "Open Technology for Professional Film Production", "topics": ["video", "camera", "vision", "real time", "photography"]}, {"organization_name": "appleseedhq", "technologies": ["c++", "python", "qt"], "link": "\n\t\t\t\thttp://appleseedhq.net/\n\t\t\t", "description": "A modern open source rendering engine for animation and visual effects", "topics": [" computer graphics", "3d", "rendering", "animation", "vfx"]}, {"organization_name": "ArduPilot", "technologies": ["c++", "python", "embedded", "rtos", "ros"], "link": "\n\t\t\t\thttp://www.ardupilot.org\n\t\t\t", "description": "The worlds most advanced open source autopilot", "topics": ["autopilot", "robotics", "uav", "drones", "rovers"]}, {"organization_name": "Babel", "technologies": ["javascript", "babel", "nodejs"], "link": "\n\t\t\t\thttps://babeljs.io/\n\t\t\t", "description": "Babel, the community driven compiler for writing next generation JavaScript", "topics": ["compilers", "ast", "minifier", "tc39"]}, {"organization_name": "Bazel", "technologies": ["java", "linux", "go"], "link": "\n\t\t\t\thttps://bazel.build\n\t\t\t", "description": "Fast and correct build system for Google scale", "topics": ["development tools", "build systems"]}, {"organization_name": "BeagleBoard.org Foundation", "technologies": ["javascript", "linux", "beaglebone", "python", "c"], "link": "\n\t\t\t\thttps://beagleboard.org\n\t\t\t", "description": "open hardware physical computing for makers, educators and professionals", "topics": ["real time", "robotics", "iot", "physical computing", "coprocessing"]}, {"organization_name": "Beam Community", "technologies": ["erlang", "elixir", "lisp", "xmpp"], "link": "\n\t\t\t\thttps://beamcommunity.github.io/\n\t\t\t", "description": "From distributed systems, to robust servers and language design on the Erlang VM", "topics": ["realtime", "communication system", "database", "clustering"]}]

This is the Json format output of all the 15 organisation we requested for.



This is the result that we get when we convert it to Dictionary readable format and this is the required output.

Difficulties-

Working on flask and beautifulsoup4 for the first time hence had some issues in implementing.

Conclusion-

The Aim of the project is achieved as I am able to complete the implementation of scraping the website using Beautifulsoup and creating an API using flask.

References-

https://summerofcode.withgoogle.com/organisations
https://www.crummy.com/software/BeautifulSoup/bs4/doc/
http://flask.pocoo.org/docs/1.0/