PROJECT REPORT

BY-RIYA GARG

17BIT0017

GDG 2 CREDIT COURSE

ABSTRACT

This project aims to create an API using Flask and scrape the GSOC website to fetch details of all the organizations in Google summer of codes and in the api send their name, link to their website, description (which comes on clicking learn more), The technologies they use and their contact email.

INTRODUCTION OF THE PROJECT

This project displays the details of the organizations in Google summer of codes using requests library of python and BeautifulSoup

The user should get a list of organizations and each member of the list should have the schema given below:

```
{
organization: 3DTK,
link: http://threedtk.de/,
description: The 3D Toolkit ....,
technologies: [c/c++, cmake, opencv, ros, boost], c contact:
johannes.schauer@uni-wuerzburg.de
}
```

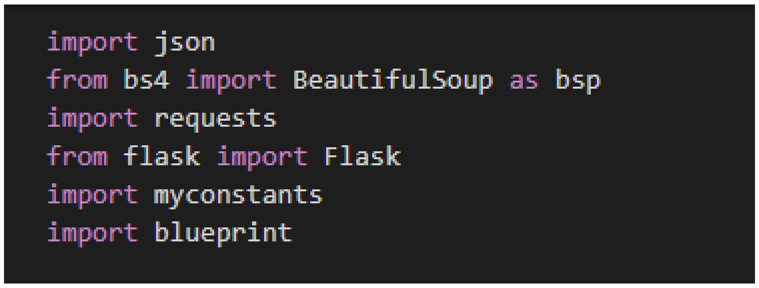
Website -https://summerofcode.withgoogle.com/organizations
FrameWorks used- Flask(Python webframework)
Library used- Beautiful Soup(Python library)
Text Editor- visual studio code

METHODOLOGY:

Libraries used:

Requests: Requests will allow you to send HTTP/1.1 requests using Python.
BeautifulSoup: for pulling data out of HTML and XML files.
JSON: used to work with JSON data.

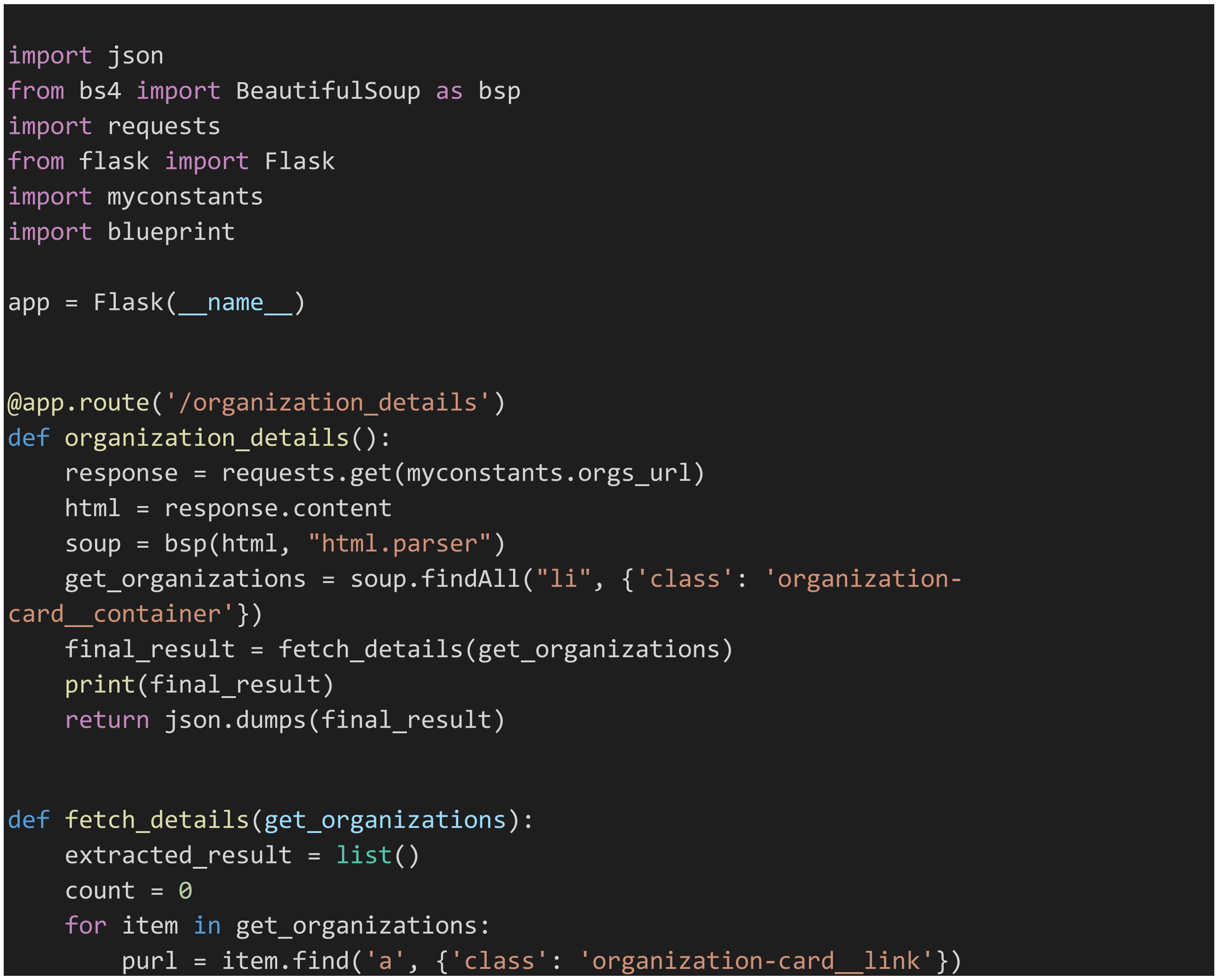
Framework used:
Flask:



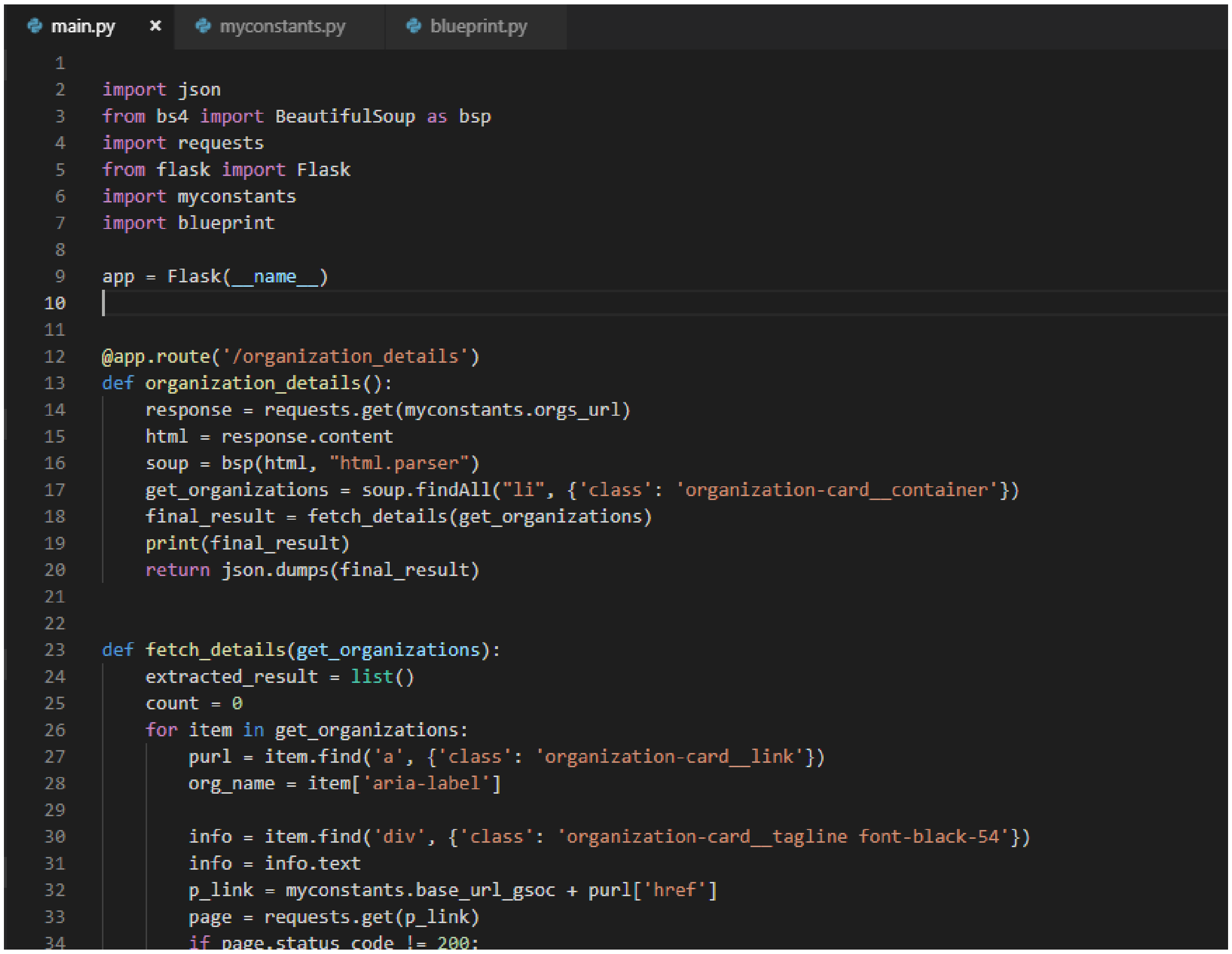
```python
import json
from bs4 import BeautifulSoup as bsp
import requests
from flask import Flask
import myconstants
import blueprint
```

CODE

MAIN.PY:(The main file which fetches and displays details)

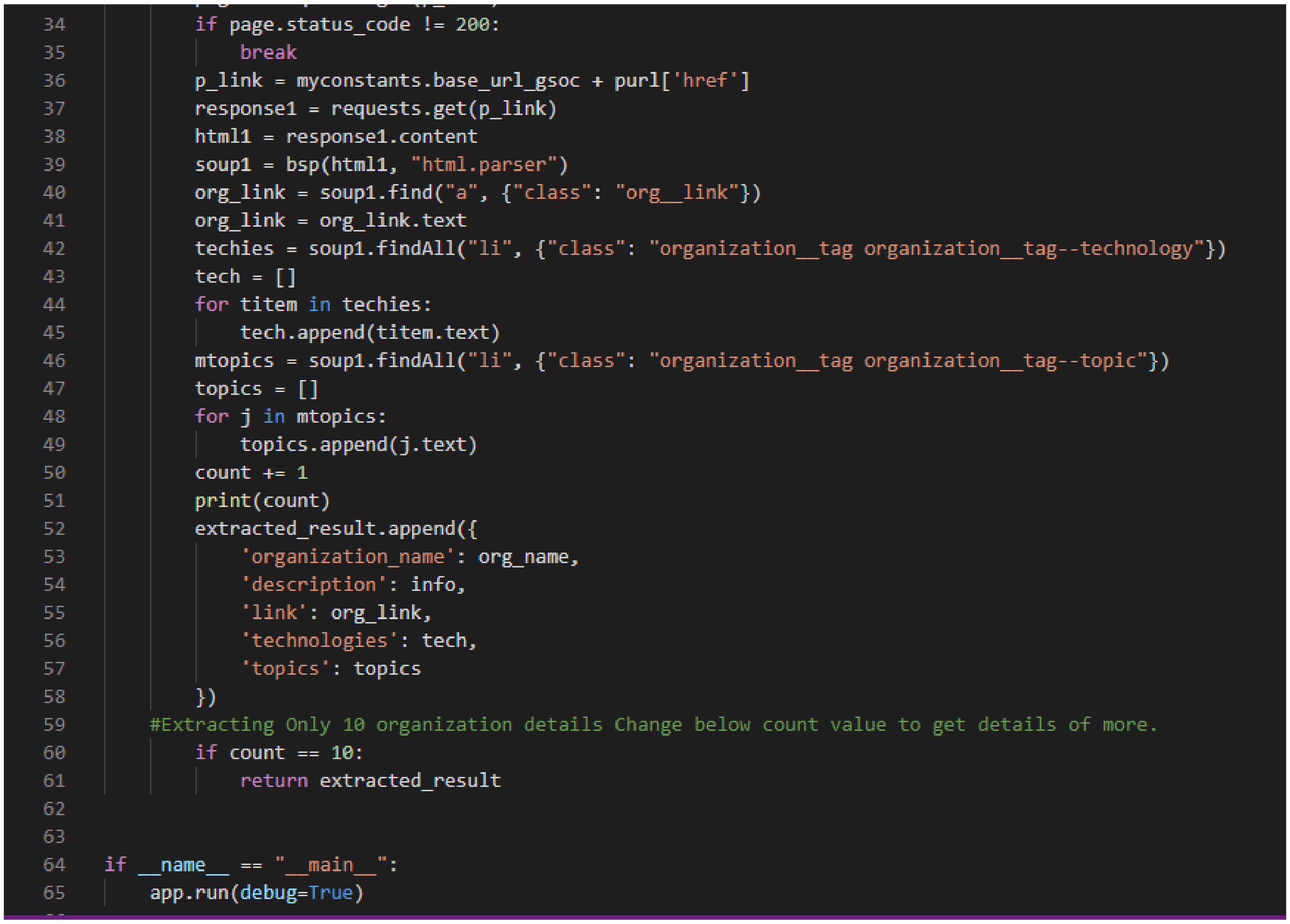


```python
import json
from bs4 import BeautifulSoup as bsp
import requests
from flask import Flask
import myconstants
import blueprint

app = Flask(__name__)


@app.route('/organization_details')
def organization_details():
    response = requests.get(myconstants.orgs_url)
    html = response.content
    soup = bsp(html, "html.parser")
    get_organizations = soup.findAll("li", {'class': 'organization-
card__container'})
    final_result = fetch_details(get_organizations)
    print(final_result)
    return json.dumps(final_result)


def fetch_details(get_organizations):
    extracted_result = list()
    count = 0
    for item in get_organizations:
        purl = item.find('a', {'class': 'organization-card__link'})
        org_name = item['aria-label']

        info = item.find('div', {'class': 'organization-card__tagline font-black-
54'})
        info = info.text
        p_link = myconstants.base_url_gsoc + purl['href']
        page = requests.get(p_link)
        if page.status_code != 200:
            break
        p_link = myconstants.base_url_gsoc + purl['href']
        response1 = requests.get(p_link)
        html1 = response1.content
        soup1 = bsp(html1, "html.parser")
        org_link = soup1.find("a", {"class": "org__link"})
        org_link = org_link.text
        techies = soup1.findAll("li", {"class": "organization__tag
organization__tag--technology"})
        tech = []
        for titem in techies:
            tech.append(titem.text)
        mtopics = soup1.findAll("li", {"class": "organization__tag
organization__tag--topic"})
        topics = []
        for j in mtopics:
            topics.append(j.text)
        count += 1
        print(count)
        extracted_result.append({
            'organization_name': org_name,
            'description': info,
            'link': org_link,
            'technologies': tech,
            'topics': topics
        })
    #Extracting Only 10 organization details Change below count value to get details
of more.
        if count == 10:
            return extracted_result


if __name__ == "__main__":
    app.run(debug=True)
```



main.py × myconstants.py blueprint.py

```python

import json
from bs4 import BeautifulSoup as bsp
import requests
from flask import Flask
import myconstants
import blueprint

app = Flask(__name__)


@app.route('/organization_details')
def organization_details():
    response = requests.get(myconstants.orgs_url)
    html = response.content
    soup = bsp(html, "html.parser")
    get_organizations = soup.findAll("li", {'class': 'organization-card__container'})
    final_result = fetch_details(get_organizations)
    print(final_result)
    return json.dumps(final_result)


def fetch_details(get_organizations):
    extracted_result = list()
    count = 0
    for item in get_organizations:
        purl = item.find('a', {'class': 'organization-card__link'})
        org_name = item['aria-label']

        info = item.find('div', {'class': 'organization-card__tagline font-black-54'})
        info = info.text
        p_link = myconstants.base_url_gsoc + purl['href']
        page = requests.get(p_link)
        if page.status_code != 200:
```



```python
        if page.status_code != 200:
            break
        p_link = myconstants.base_url_gsoc + purl['href']
        response1 = requests.get(p_link)
        html1 = response1.content
        soup1 = bsp(html1, "html.parser")
        org_link = soup1.find("a", {"class": "org__link"})
        org_link = org_link.text
        techies = soup1.findAll("li", {"class": "organization__tag organization__tag--technology"})
        tech = []
        for titem in techies:
            tech.append(titem.text)
        mtopics = soup1.findAll("li", {"class": "organization__tag organization__tag--topic"})
        topics = []
        for j in mtopics:
            topics.append(j.text)
        count += 1
        print(count)
        extracted_result.append({
            'organization_name': org_name,
            'description': info,
            'link': org_link,
            'technologies': tech,
            'topics': topics
        })
    #Extracting Only 10 organization details Change below count value to get details of more.
        if count == 10:
            return extracted_result


if __name__ == "__main__":
    app.run(debug=True)
```

EXPLANATION:

Function fetch_details

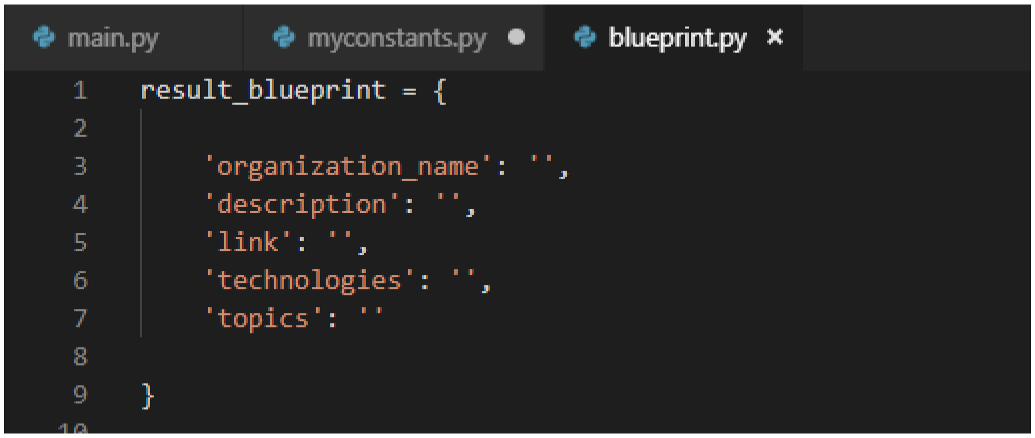
Fetches the organziation name, description , link,technologies, topics for the number of organizations specified(in this case -10) by looping through the result received from organization_details function. Stores the final result in extracted_result using the find method of beautifulSoup library.

function organization_details

Line 13 -fetches the details from the url specified in the myconstants.py file.
Line 14 -sends the get request and saves the response as a response object.
Line 16 -parses the page into beautifulSoup format
Line 17 -extracts the content in the list using the HTML class organization-card__container
Line 18 -passes on the collective information to the fetch_details function .
Finally returns the result in JSON format in line 20.

BLUEPRINT.PY(The format of the information fetched)

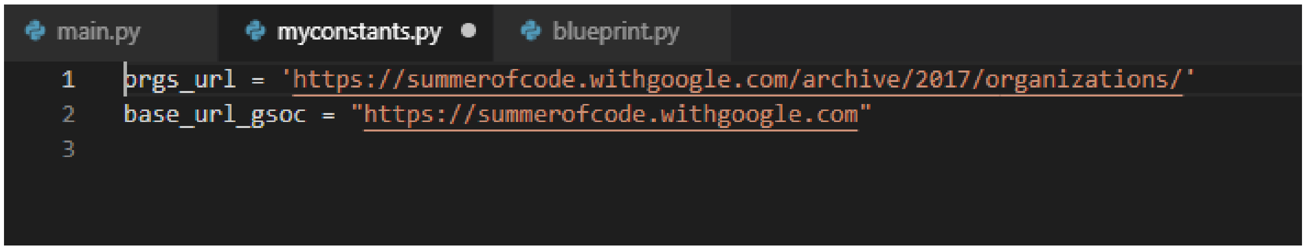
```
result_blueprint = {

    'organization_name': '',
    'description': '',
    'link': '',
    'technologies': '',
    'topics': ''

}
```



```
result_blueprint = {

    'organization_name': '',
    'description': '',
    'link': '',
    'technologies': '',
    'topics': ''

}
```

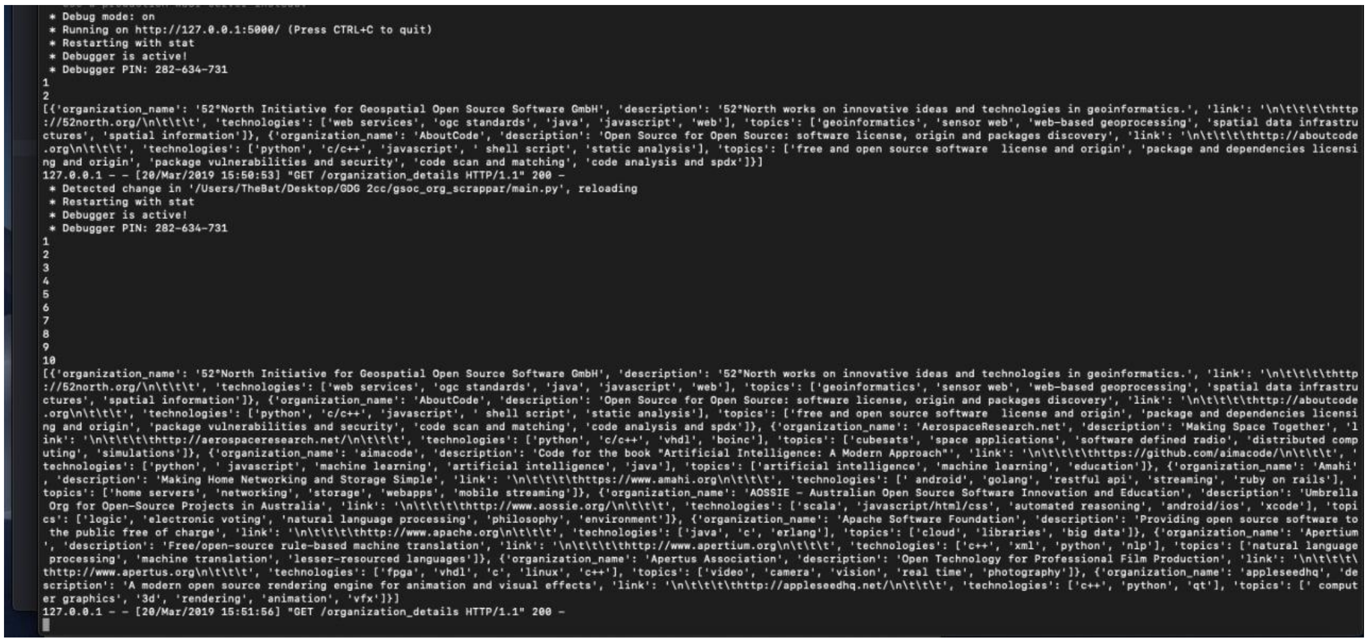
MYCONSTANTS.PY(contains constants used in scripts.)

```python
orgs_url = 'https://summerofcode.withgoogle.com/archive/2017/organizations/'
base_url_gsoc = "https://summerofcode.withgoogle.com"
```



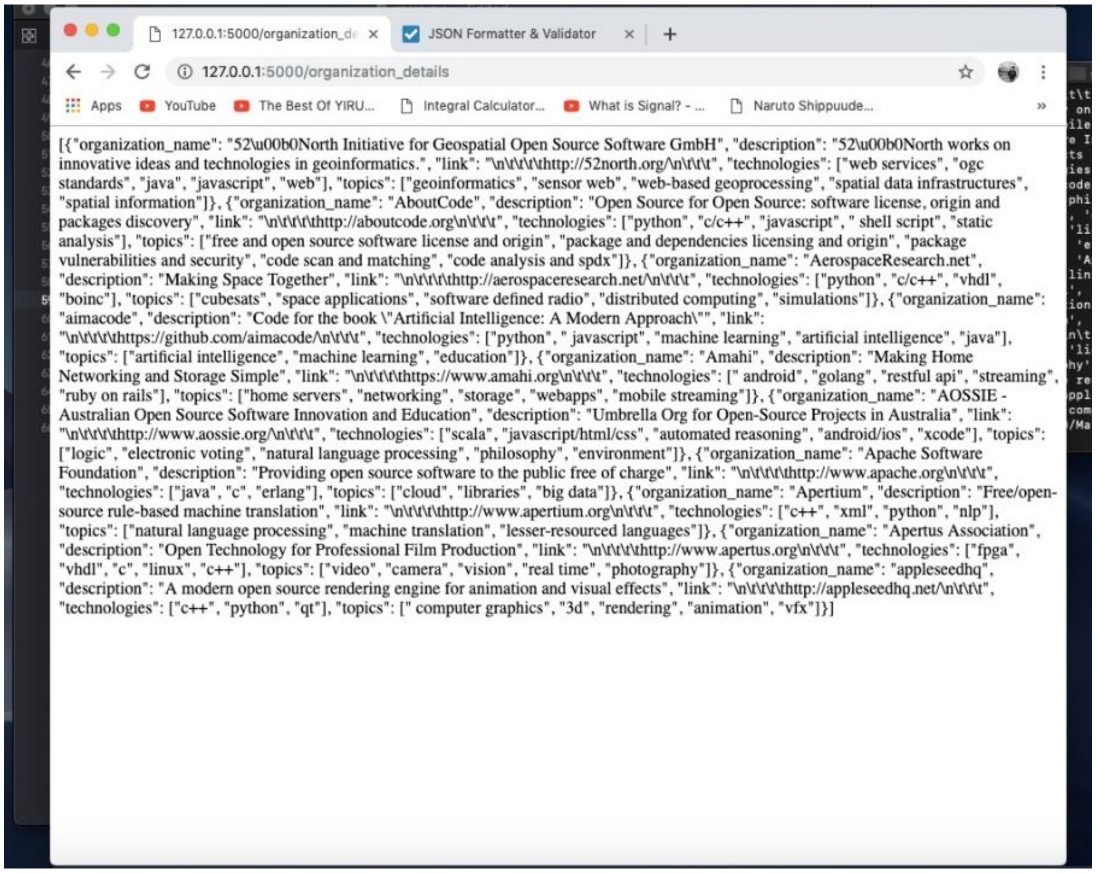
```python
orgs_url = 'https://summerofcode.withgoogle.com/archive/2017/organizations/'
base_url_gsoc = "https://summerofcode.withgoogle.com"
```

RESULT OF EXECUTION



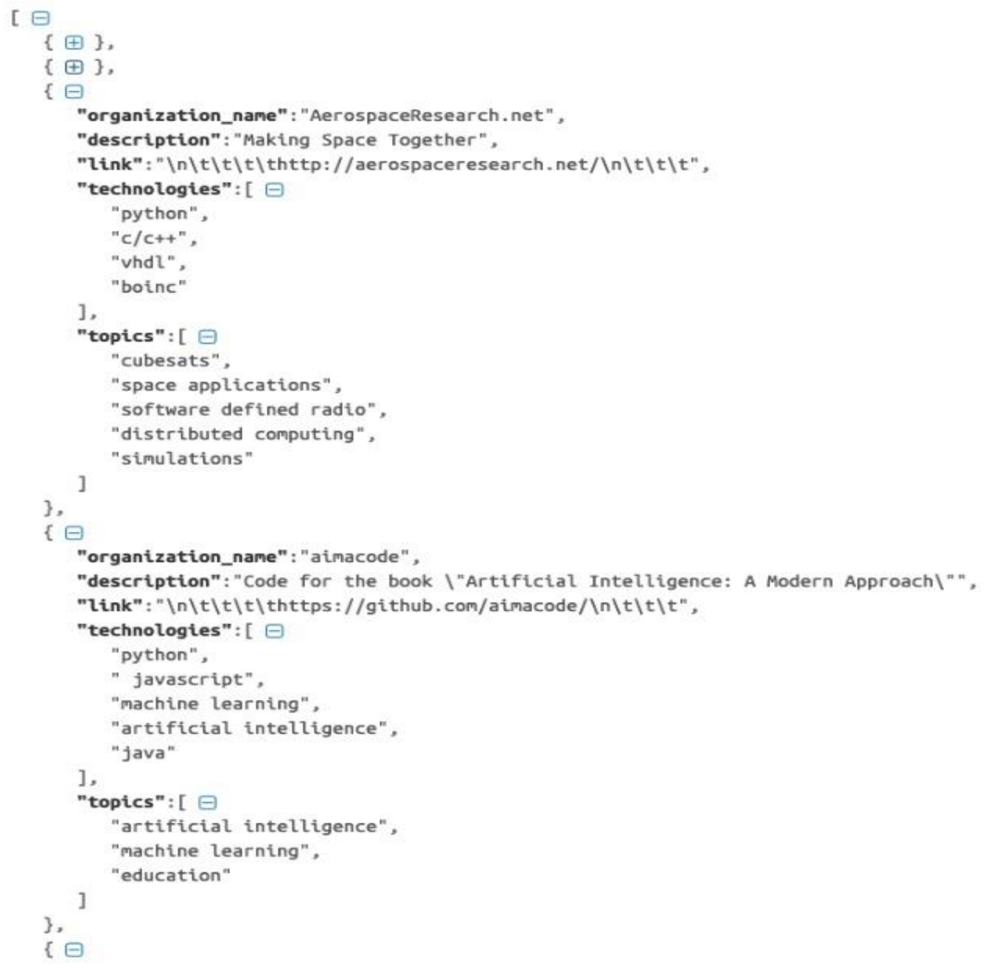
```
 * Debug mode: on
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 282-634-731
1
2
[{'organization_name': '52°North Initiative for Geospatial Open Source Software GmbH', 'description': '52°North works on innovative ideas and technologies in geoinformatics.', 'link': '\n\t\t\t\thttp
://52north.org/\n\t\t\t', 'technologies': ['web services', 'ogc standards', 'java', 'javascript', 'web'], 'topics': ['geoinformatics', 'sensor web', 'web-based geoprocessing', 'spatial data infrastru
ctures', 'spatial information']}, {'organization_name': 'AboutCode', 'description': 'Open Source for Open Source: software license, origin and packages discovery', 'link': '\n\t\t\t\thttp://aboutcode
.org\n\t\t\t', 'technologies': ['python', 'c/c++', 'javascript', ' shell script', 'static analysis'], 'topics': ['free and open source software  license and origin', 'package and dependencies licensi
ng and origin', 'package vulnerabilities and security', 'code scan and matching', 'code analysis and spdx']}]
127.0.0.1 - - [20/Mar/2019 15:50:53] "GET /organization_details HTTP/1.1" 200 -
 * Detected change in '/Users/TheBat/Desktop/GDG 2cc/gsoc_org_scrappar/main.py', reloading
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 282-634-731
1
2
3
4
5
6
7
8
9
10
[{'organization_name': '52°North Initiative for Geospatial Open Source Software GmbH', 'description': '52°North works on innovative ideas and technologies in geoinformatics.', 'link': '\n\t\t\t\thttp
://52north.org/\n\t\t\t', 'technologies': ['web services', 'ogc standards', 'java', 'javascript', 'web'], 'topics': ['geoinformatics', 'sensor web', 'web-based geoprocessing', 'spatial data infrastru
ctures', 'spatial information']}, {'organization_name': 'AboutCode', 'description': 'Open Source for Open Source: software license, origin and packages discovery', 'link': '\n\t\t\t\thttp://aboutcode
.org\n\t\t\t', 'technologies': ['python', 'c/c++', 'javascript', ' shell script', 'static analysis'], 'topics': ['free and open source software  license and origin', 'package and dependencies licensi
ng and origin', 'package vulnerabilities and security', 'code scan and matching', 'code analysis and spdx']}, {'organization_name': 'AerospaceResearch.net', 'description': 'Making Space Together', 'l
ink': '\n\t\t\t\thttp://aerospaceresearch.net/\n\t\t\t', 'technologies': ['python', 'c/c++', 'vhdl', 'boinc'], 'topics': ['cubesats', 'space applications', 'software defined radio', 'distributed comp
uting', 'simulations']}, {'organization_name': 'aimacode', 'description': 'Code for the book "Artificial Intelligence: A Modern Approach"', 'link': '\n\t\t\t\thttps://github.com/aimacode/\n\t\t\t', '
technologies': ['python', ' javascript', 'machine learning', 'artificial intelligence', 'java'], 'topics': ['artificial intelligence', 'machine learning', 'education']}, {'organization_name': 'Amahi'
, 'description': 'Making Home Networking and Storage Simple', 'link': '\n\t\t\t\thttps://www.amahi.org\n\t\t\t', 'technologies': [' android', 'golang', 'restful api', 'streaming', 'ruby on rails'], '
topics': ['home servers', 'networking', 'storage', 'webapps', 'mobile streaming']}, {'organization_name': 'AOSSIE - Australian Open Source Software Innovation and Education', 'description': 'Umbrella
 Org for Open-Source Projects in Australia', 'link': '\n\t\t\t\thttp://www.aossie.org/\n\t\t\t', 'technologies': ['scala', 'javascript/html/css', 'automated reasoning', 'android/ios', 'xcode'], 'topi
cs': ['logic', 'electronic voting', 'natural language processing', 'philosophy', 'environment']}, {'organization_name': 'Apache Software Foundation', 'description': 'Providing open source software to
 the public free of charge', 'link': '\n\t\t\t\thttp://www.apache.org\n\t\t\t', 'technologies': ['java', 'c', 'erlang'], 'topics': ['cloud', 'libraries', 'big data']}, {'organization_name': 'Apertium
', 'description': 'Free/open-source rule-based machine translation', 'link': '\n\t\t\t\thttp://www.apertium.org\n\t\t\t', 'technologies': ['c++', 'xml', 'python', 'nlp'], 'topics': ['natural language
 processing', 'machine translation', 'lesser-resourced languages']}, {'organization_name': 'Apertus Association', 'description': 'Open Technology for Professional Film Production', 'link': '\n\t\t\t\
thttp://www.apertus.org\n\t\t\t', 'technologies': ['fpga', 'vhdl', 'c', 'linux', 'c++'], 'topics': ['video', 'camera', 'vision', 'real time', 'photography']}, {'organization_name': 'appleseedhq', 'de
scription': 'A modern open source rendering engine for animation and visual effects', 'link': '\n\t\t\t\thttp://appleseedhq.net/\n\t\t\t', 'technologies': ['c++', 'python', 'qt'], 'topics': [' comput
er graphics', '3d', 'rendering', 'animation', 'vfx']}]
127.0.0.1 - - [20/Mar/2019 15:51:56] "GET /organization_details HTTP/1.1" 200 -
```

DETAILS OF 10 ORGANIZATIONS

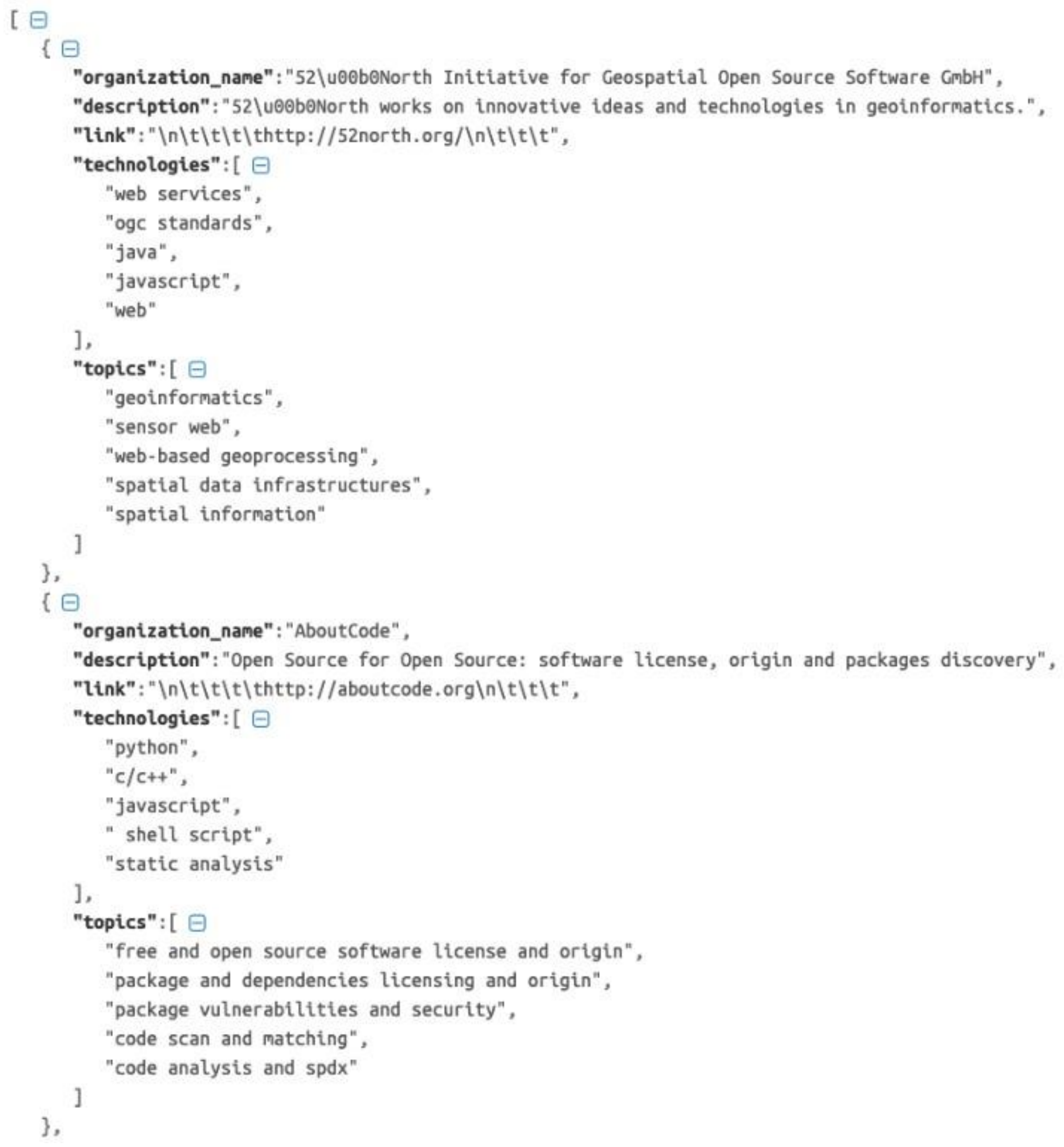


127.0.0.1:5000/organization_details

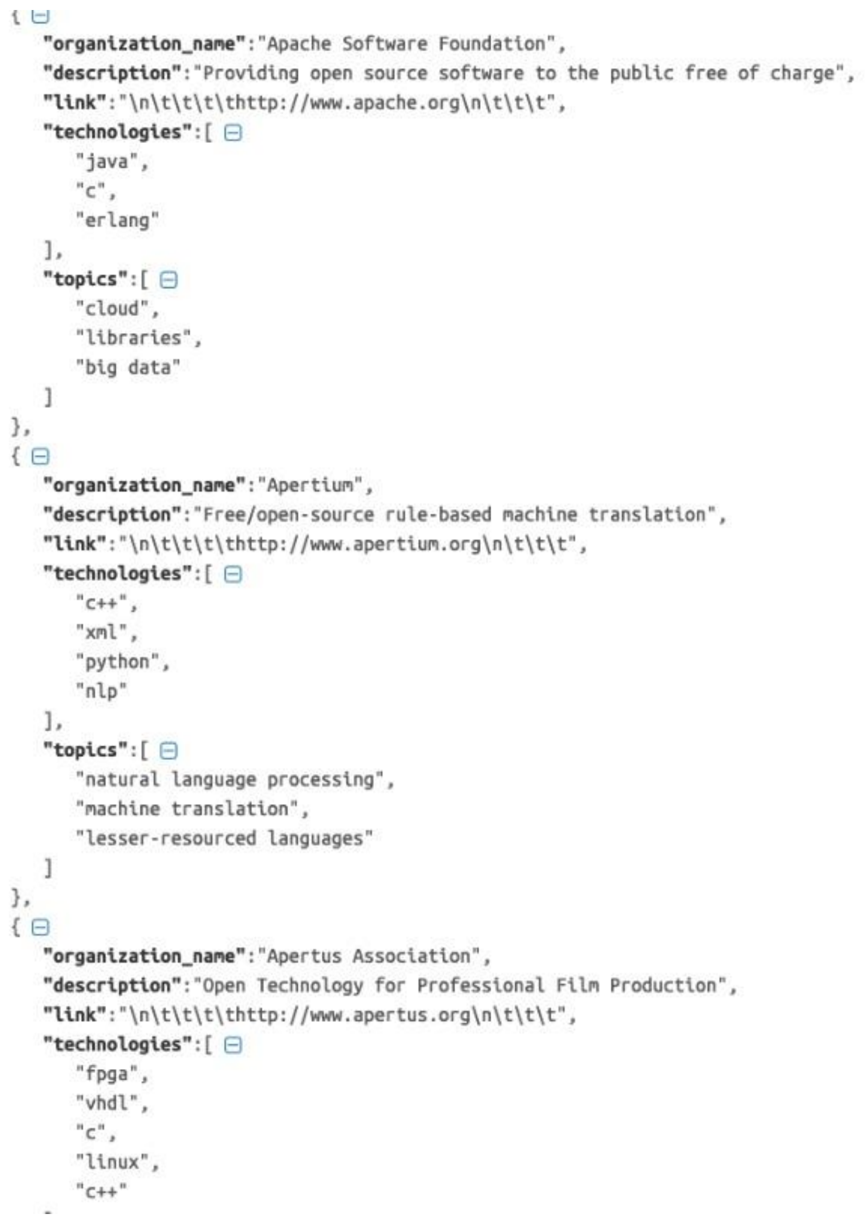
```
[{"organization_name": "52\u00b0North Initiative for Geospatial Open Source Software GmbH", "description": "52\u00b0North works on
innovative ideas and technologies in geoinformatics.", "link": "\n\t\t\t\thttp://52north.org/\n\t\t\t", "technologies": ["web services", "ogc
standards", "java", "javascript", "web"], "topics": ["geoinformatics", "sensor web", "web-based geoprocessing", "spatial data infrastructures",
"spatial information"]}, {"organization_name": "AboutCode", "description": "Open Source for Open Source: software license, origin and
packages discovery", "link": "\n\t\t\t\thttp://aboutcode.org\n\t\t\t", "technologies": ["python", "c/c++", "javascript", " shell script", "static
analysis"], "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",
"description": "Making Space Together", "link": "\n\t\t\t\thttp://aerospaceresearch.net/\n\t\t\t", "technologies": ["python", "c/c++", "vhdl",
"boinc"], "topics": ["cubesats", "space applications", "software defined radio", "distributed computing", "simulations"]}, {"organization_name":
"aimacode", "description": "Code for the book \"Artificial Intelligence: A Modern Approach\"", "link":
"\n\t\t\t\thttps://github.com/aimacode/\n\t\t\t", "technologies": ["python", " javascript", "machine learning", "artificial intelligence", "java"],
"topics": ["artificial intelligence", "machine learning", "education"]}, {"organization_name": "Amahi", "description": "Making Home
Networking and Storage Simple", "link": "\n\t\t\t\thttps://www.amahi.org\n\t\t\t", "technologies": [" android", "golang", "restful api", "streaming",
"ruby on rails"], "topics": ["home servers", "networking", "storage", "webapps", "mobile streaming"]}, {"organization_name": "AOSSIE -
Australian Open Source Software Innovation and Education", "description": "Umbrella Org for Open-Source Projects in Australia", "link":
"\n\t\t\t\thttp://www.aossie.org/\n\t\t\t", "technologies": ["scala", "javascript/html/css", "automated reasoning", "android/ios", "xcode"], "topics":
["logic", "electronic voting", "natural language processing", "philosophy", "environment"]}, {"organization_name": "Apache Software
Foundation", "description": "Providing open source software to the public free of charge", "link": "\n\t\t\t\thttp://www.apache.org\n\t\t\t",
"technologies": ["java", "c", "erlang"], "topics": ["cloud", "libraries", "big data"]}, {"organization_name": "Apertium", "description": "Free/open-
source rule-based machine translation", "link": "\n\t\t\t\thttp://www.apertium.org\n\t\t\t", "technologies": ["c++", "xml", "python", "nlp"],
"topics": ["natural language processing", "machine translation", "lesser-resourced languages"]}, {"organization_name": "Apertus Association",
"description": "Open Technology for Professional Film Production", "link": "\n\t\t\t\thttp://www.apertus.org\n\t\t\t", "technologies": ["fpga",
"vhdl", "c", "linux", "c++"], "topics": ["video", "camera", "vision", "real time", "photography"]}, {"organization_name": "appleseedhq",
"description": "A modern open source rendering engine for animation and visual effects", "link": "\n\t\t\t\thttp://appleseedhq.net/\n\t\t\t",
"technologies": ["c++", "python", "qt"], "topics": [" computer graphics", "3d", "rendering", "animation", "vfx"]}]
```



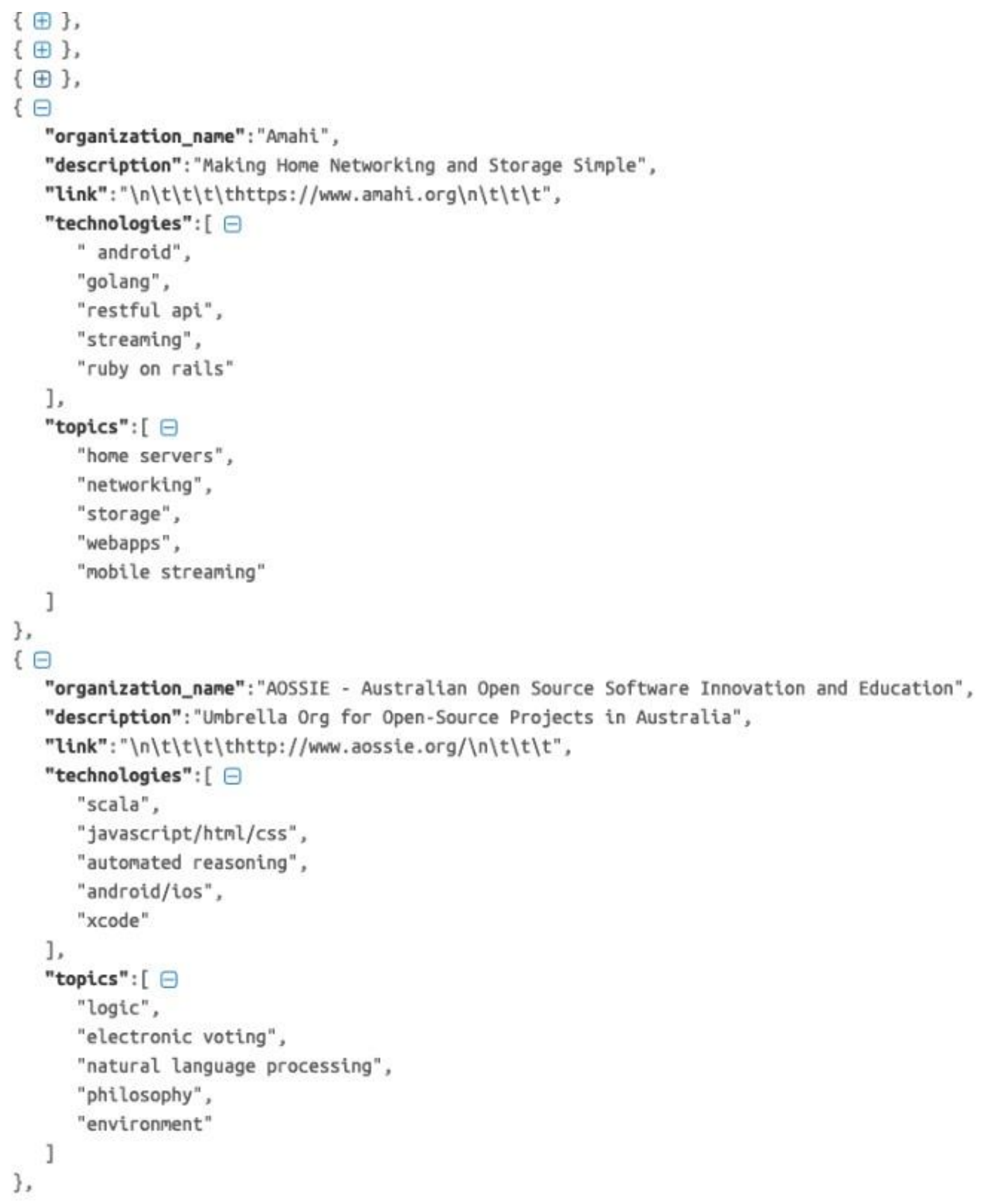
```
[ ⊟
   { ⊞ },
   { ⊞ },
   { ⊟
      "organization_name":"AerospaceResearch.net",
      "description":"Making Space Together",
      "link":"\n\t\t\t\thttp://aerospaceresearch.net/\n\t\t\t",
      "technologies":[ ⊟
         "python",
         "c/c++",
         "vhdl",
         "boinc"
      ],
      "topics":[ ⊟
         "cubesats",
         "space applications",
         "software defined radio",
         "distributed computing",
         "simulations"
      ]
   },
   { ⊟
      "organization_name":"aimacode",
      "description":"Code for the book \"Artificial Intelligence: A Modern Approach\"",
      "link":"\n\t\t\t\thttps://github.com/aimacode/\n\t\t\t",
      "technologies":[ ⊟
         "python",
         " javascript",
         "machine learning",
         "artificial intelligence",
         "java"
      ],
      "topics":[ ⊟
         "artificial intelligence",
         "machine learning",
         "education"
      ]
   },
   { ⊟
```



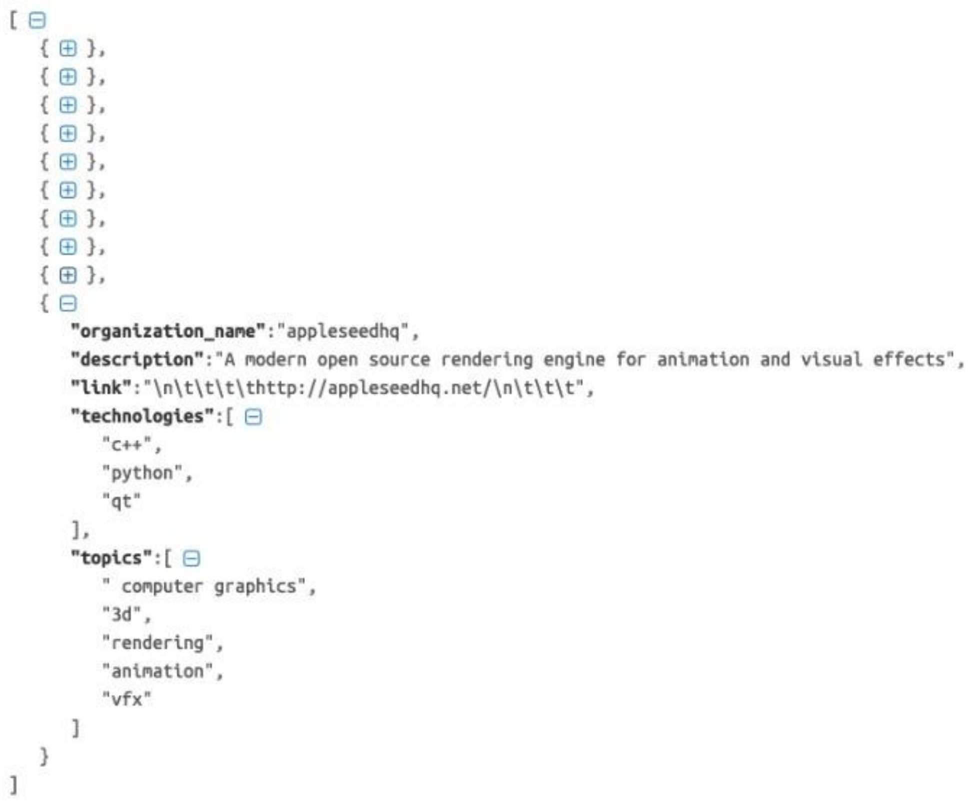
```
[
  {
    "organization_name":"52\u00b0North Initiative for Geospatial Open Source Software GmbH",
    "description":"52\u00b0North works on innovative ideas and technologies in geoinformatics.",
    "link":"\n\t\t\t\thttp://52north.org/\n\t\t\t",
    "technologies":[
      "web services",
      "ogc standards",
      "java",
      "javascript",
      "web"
    ],
    "topics":[
      "geoinformatics",
      "sensor web",
      "web-based geoprocessing",
      "spatial data infrastructures",
      "spatial information"
    ]
  },
  {
    "organization_name":"AboutCode",
    "description":"Open Source for Open Source: software license, origin and packages discovery",
    "link":"\n\t\t\t\thttp://aboutcode.org\n\t\t\t",
    "technologies":[
      "python",
      "c/c++",
      "javascript",
      " shell script",
      "static analysis"
    ],
    "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":"Apache Software Foundation",
   "description":"Providing open source software to the public free of charge",
   "link":"\n\t\t\t\thttp://www.apache.org\n\t\t\t",
   "technologies":[
      "java",
      "c",
      "erlang"
   ],
   "topics":[
      "cloud",
      "libraries",
      "big data"
   ]
},
{
   "organization_name":"Apertium",
   "description":"Free/open-source rule-based machine translation",
   "link":"\n\t\t\t\thttp://www.apertium.org\n\t\t\t",
   "technologies":[
      "c++",
      "xml",
      "python",
      "nlp"
   ],
   "topics":[
      "natural language processing",
      "machine translation",
      "lesser-resourced languages"
   ]
},
{
   "organization_name":"Apertus Association",
   "description":"Open Technology for Professional Film Production",
   "link":"\n\t\t\t\thttp://www.apertus.org\n\t\t\t",
   "technologies":[
      "fpga",
      "vhdl",
      "c",
      "linux",
      "c++"
```



```
{ ⊞ },
{ ⊞ },
{ ⊞ },
{ ⊟
   "organization_name":"Amahi",
   "description":"Making Home Networking and Storage Simple",
   "link":"\n\t\t\t\thttps://www.amahi.org\n\t\t\t",
   "technologies":[ ⊟
      " android",
      "golang",
      "restful api",
      "streaming",
      "ruby on rails"
   ],
   "topics":[ ⊟
      "home servers",
      "networking",
      "storage",
      "webapps",
      "mobile streaming"
   ]
},
{ ⊟
   "organization_name":"AOSSIE - Australian Open Source Software Innovation and Education",
   "description":"Umbrella Org for Open-Source Projects in Australia",
   "link":"\n\t\t\t\thttp://www.aossie.org/\n\t\t\t",
   "technologies":[ ⊟
      "scala",
      "javascript/html/css",
      "automated reasoning",
      "android/ios",
      "xcode"
   ],
   "topics":[ ⊟
      "logic",
      "electronic voting",
      "natural language processing",
      "philosophy",
      "environment"
   ]
},
```



```
[ ⊟
  { ⊞ },
  { ⊞ },
  { ⊞ },
  { ⊞ },
  { ⊞ },
  { ⊞ },
  { ⊞ },
  { ⊞ },
  { ⊞ },
  { ⊟
    "organization_name":"appleseedhq",
    "description":"A modern open source rendering engine for animation and visual effects",
    "link":"\n\t\t\t\thttp://appleseedhq.net/\n\t\t\t",
    "technologies":[ ⊟
      "c++",
      "python",
      "qt"
    ],
    "topics":[ ⊟
      " computer graphics",
      "3d",
      "rendering",
      "animation",
      "vfx"
    ]
  }
]
```

DIFFICULTIES FACED:

- Working with Python and learning about various libraries used.
- Extracting information from the website using various HTML tags and identifying which tag corresponds to which information .

CONCLUSION

The aim of the project has been successfully achieved. I am able to get the Details of the various organizations in google summer of codes. The code has been implemented in python as per the requirements.