AUTOSEARCH FOR STACK OVERFLOW

MINOR PROJECT REPORT

By

PARVATHY V NAIR (RA2211003010295)

ANYESHA BISWAS (RA2211003010298)

ACHANTA BHAVYASREE (RA2211003010316)

Under the guidance of
Dr. Suresh Anand M
In partial fulfilment for the Course
of

21CSC201J – DATA STRUCTURES AND ALGORITHM

in CSE CORE

FACULTY OF ENGINEERING AND TECHNOLOGY

SCHOOL OF COMPUTING

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR

NOVEMBER 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this minor project report for the Course **21CSC201J – DATA STRUCTURES AND ALGORITHM** entitled in "**AUTOSEARCH FOR STACK OVERFLOW**" is the bonafide work of **Parvathy V Nair (RA2211003010295)** and **Anyesha Biswas (RA2211003010298)** and **Achanta Bhavyasree (RA2211003010316)** who carried out the work under my supervision.

SIGNATURE

Dr. Suresh Anand M	Dr. Pushpalatha M
Assistant Professor	**HoD Professor and Head ,**
CTech	**CTech**
SRM Institute of Science and Technology	SRM Institute of Science and
Kattankulathur	Technology Kattankulathur

ABSTRACT

"This graphical user interface (GUI) application is built using Python and the Tkinter module. Its primary function is to streamline the process of troubleshooting errors in Python code. Users can select a specific Python file, and the application automatically queries the Stack Exchange API, specifically Stack Overflow, to find potential solutions and answers related to any errors in the selected file. This tool enhances the efficiency of error resolution, making it a valuable resource for Python developers seeking quick and accurate solutions to their coding issues."

FEATURES:

1. Automatically searchs for error solution and opens tabs in browser related to the error solution on StackOverflow.

2. Saves time and increases productivity.

BUILT WITH:

1. Python

2. Tkinter Module

3. Stack Exchange API

ACKNOWLEDGEMENT

We express our heartfelt thanks to our honorable **Vice Chancellor Dr. C. MUTHAMIZHCHELVAN**, for being the beacon in all our endeavors.

We would like to express my warmth of gratitude to our **Registrar Dr. S. Ponnusamy,** for his encouragement.

We express our profound gratitude to our **Dean (College of Engineering and Technology) Dr. T. V. Gopal,** for bringing out novelty in all executions.

We would like to express my heartfelt thanks to Chairperson, School of Computing **Dr. Revathi Venkataraman,** for imparting confidence to complete my course project.

We wish to express my sincere thanks to **Course Audit Professors Dr. Vadivu. G , Professor, Department of Data Science and Business Systems and Dr. Sasikala. E Professor, Department of Data Science and Business Systems** and **Course Coordinators** for their constant encouragement and support.

We are highly thankful to our my Course project Faculty **Dr. Suresh Anand M, Assistant Professor, CTech ,** for his/her assistance, timely suggestion and guidance throughout the duration of this course project.

We extend my gratitude to our **HoD Dr. Pushpalatha M , Professor and Head , CTech** and my Departmental colleagues for their Support.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project. Above all, I thank the almighty for showering his blessings on me to complete my Course project.

TABLE OF CONTENTS

INTRODUCTION

As a developer, finding solutions to coding challenges or issues is an essential part of the job. One of the most popular platforms for developers to find answers to their coding queries is Stack Overflow. However, with thousands of questions and answers on the platform, it can be time-consuming to search for relevant solutions.

This project aims to create an autosearch feature for Stack Overflow that will help developers find solutions to their coding problems quickly. By leveraging natural language processing techniques and machine learning algorithms, the autosearch feature will suggest the most relevant answers based on the user's query.

With this feature, developers can save time and effort spent on manually searching through numerous Stack Overflow threads, allowing them to focus on other critical aspects of their work.

The auto search feature can also benefit developers by exposing them to new solutions and approaches that they may not have considered before, leading to more efficient and streamlined coding practices.

2. REQUIREMENTS

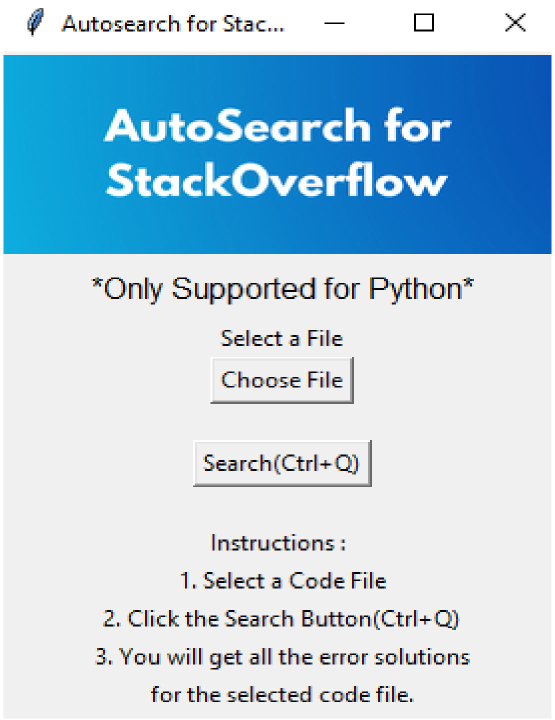
Requirement Analysis:
We will create a Python script that acts as a wrapper for testing other Python code. The primary function of this script is to perform the following steps:

1. Testing Python Code: The script will take a Python code file as input and execute it. It will check for any syntactic or runtime errors in the code. If no errors are found, it will print "No errors found."
2. Error Handling: In the event of encountering an error, the script will extract the error type and the error message. This information will be crucial for identifying and resolving the issue.
3. Stack Exchange API Call: Upon identifying an error, the script will make a REST API call to the Stack Exchange API, specifically Stack Overflow. This call will include the error type and message to search for relevant solutions and answers in the Stack Overflow community.
4. JSON Response Processing: The script will process the JSON response obtained from the API call. It will extract at most N (a user-defined number) links to Stack Overflow threads. These links will point to discussions and solutions related to the encountered error.
5. Opening Threads in a Web Browser: Finally, the script will open the extracted links in a web browser. This allows the user to access Stack Overflow threads with relevant information on how to resolve the specific error encountered in the code.

This wrapper script simplifies the error-checking process for Python developers by automating error identification, searching for solutions, and providing direct access to community-driven solutions on Stack Overflow. It enhances productivity and helps developers resolve issues efficiently.

3. ARCHITECTURE AND DESIGN

- **Network Architecture**



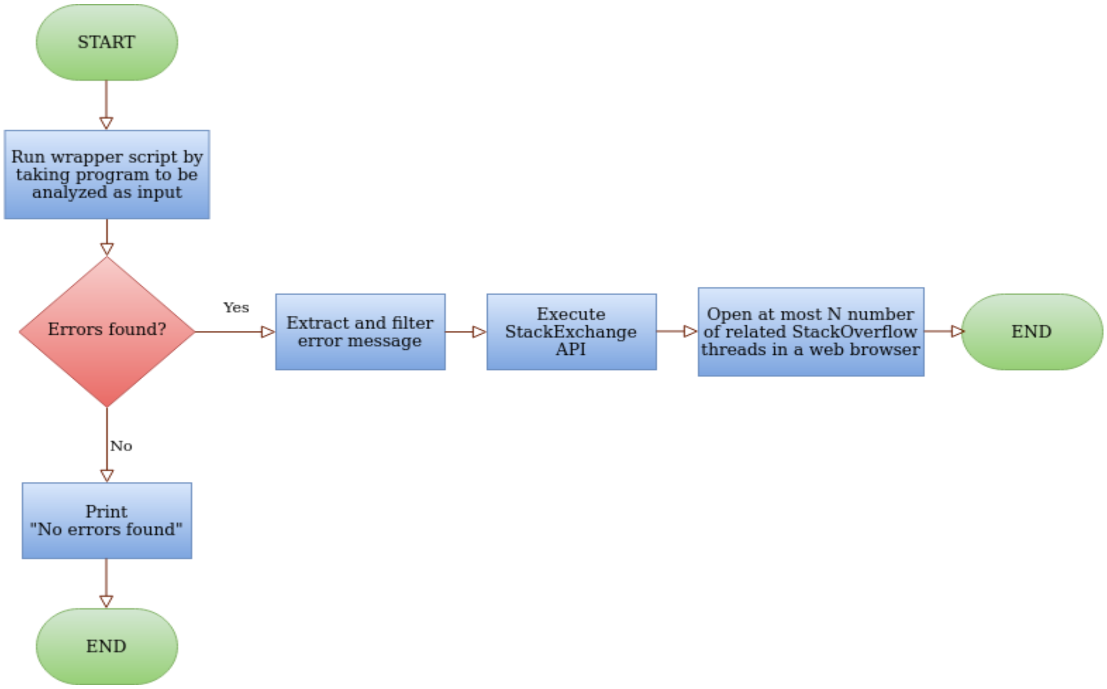
AutoSearch for StackOverflow

Only Supported for Python

Select a File

Instructions :

1. Select a Code File
2. Click the Search Button(Ctrl+Q)
3. You will get all the error solutions for the selected code file.


START
Run wrapper script by taking program to be analyzed as input
Errors found?
Yes
Extract and filter error message
Execute StackExchange API
Open at most N number of related StackOverflow threads in a web browser
END
No
Print "No errors found"
END

IMPLEMENTATION:

LICENSE:

MIT License

Copyright (c) 2020 Dhawal Kamdar

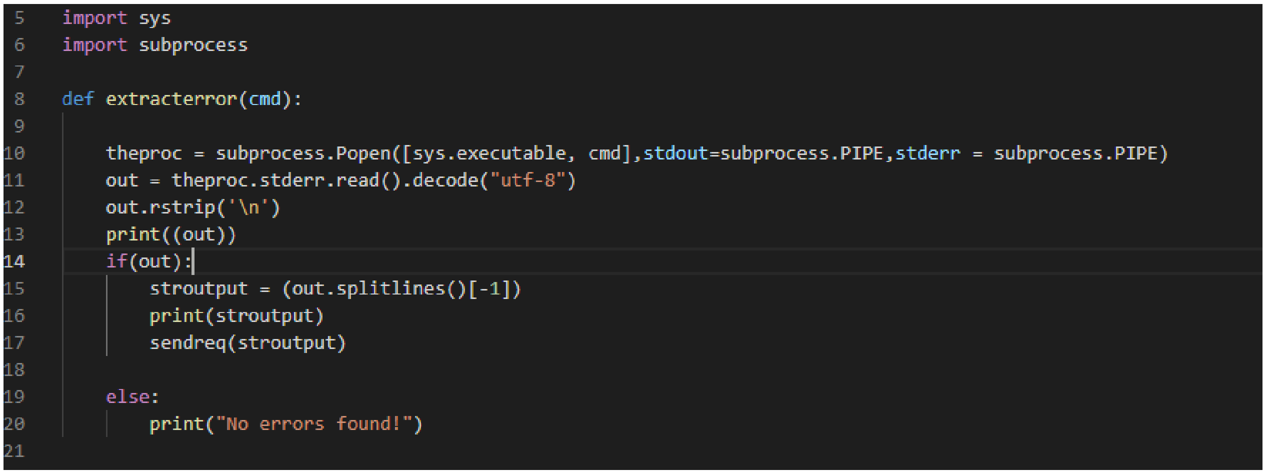
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

The software is provided "as is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and noninfringement. In no event shall the authors or copyright holders be liable for any claim, damages or other
liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the software.

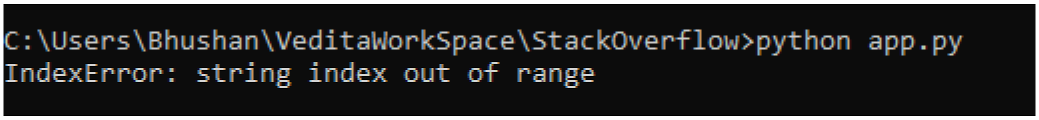
RESULTS AND DISCUSSION:

Code for extracting error:



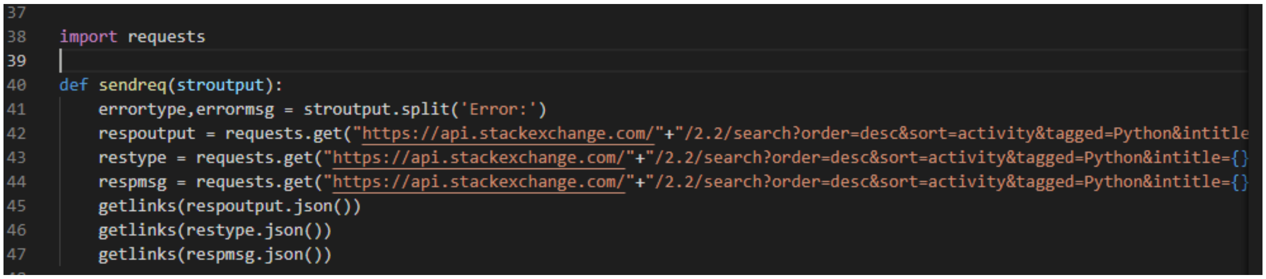
```
import sys
import subprocess

def extracterror(cmd):

    theproc = subprocess.Popen([sys.executable, cmd],stdout=subprocess.PIPE,stderr = subprocess.PIPE)
    out = theproc.stderr.read().decode("utf-8")
    out.rstrip('\n')
    print((out))
    if(out):
        stroutput = (out.splitlines()[-1])
        print(stroutput)
        sendreq(stroutput)

    else:
        print("No errors found!")

```

Extracted error output :



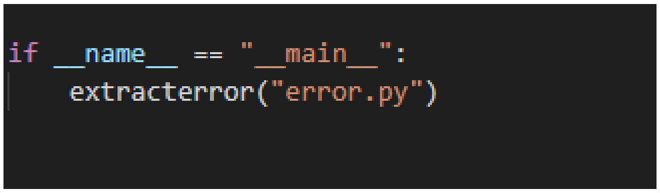
```
C:\Users\Bhushan\VeditaWorkSpace\StackOverflow>python app.py
IndexError: string index out of range
```

Code for making REST API calls :



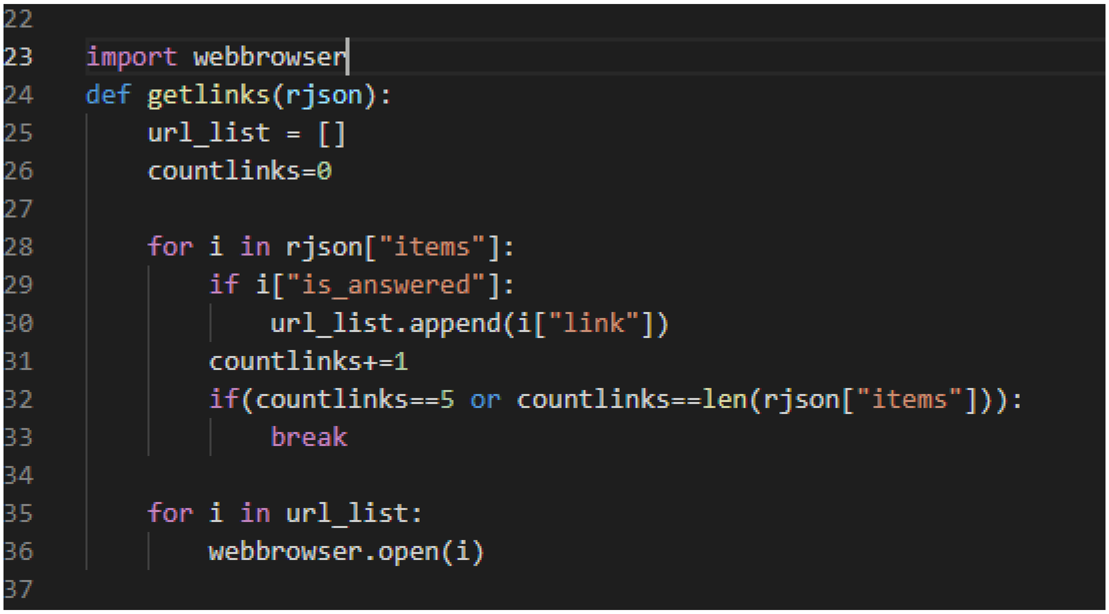
```

import requests

def sendreq(stroutput):
    errortype,errormsg = stroutput.split('Error:')
    respoutput = requests.get("https://api.stackexchange.com/"+"/2.2/search?order=desc&sort=activity&tagged=Python&intitle
    restype = requests.get("https://api.stackexchange.com/"+"/2.2/search?order=desc&sort=activity&tagged=Python&intitle={}
    respmsg = requests.get("https://api.stackexchange.com/"+"/2.2/search?order=desc&sort=activity&tagged=Python&intitle={}
    getlinks(respoutput.json())
    getlinks(restype.json())
    getlinks(respmsg.json())
```

Calling the extractor function from main :



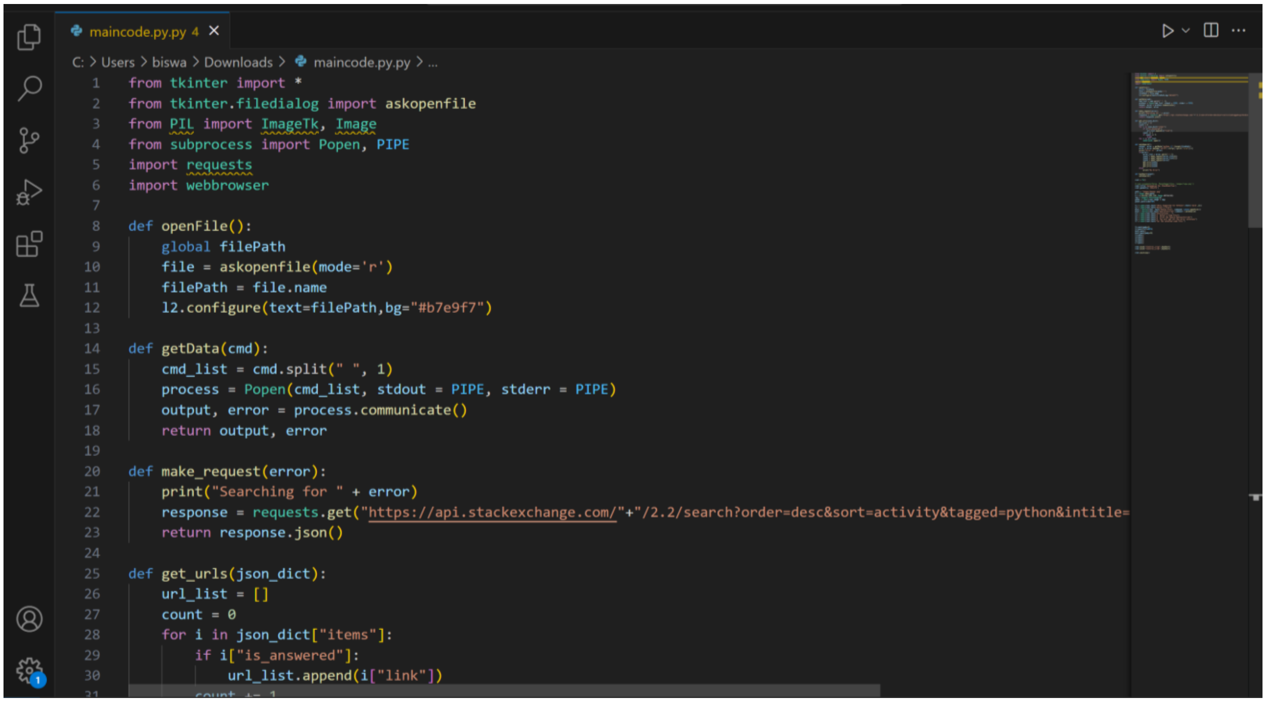
```
if __name__ == "__main__":
    extracterror("error.py")
```

Use the webbrowser module to open the extracted links in a web browser :

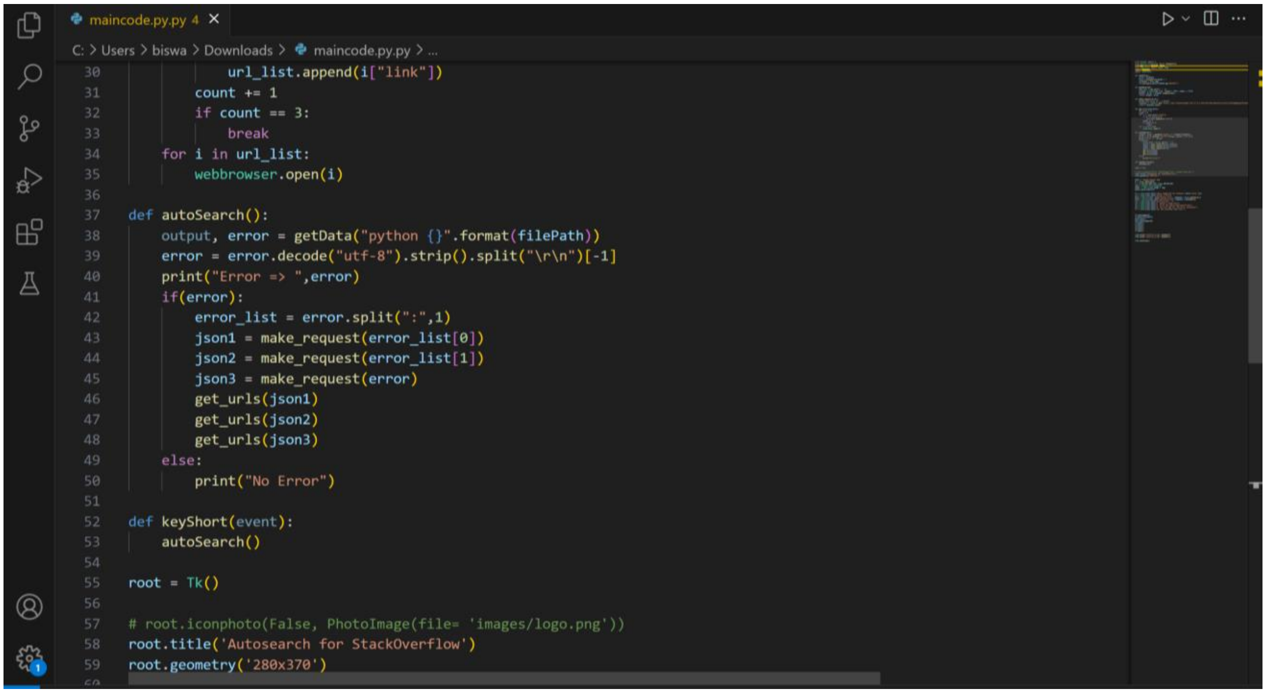


```

import webbrowser
def getlinks(rjson):
    url_list = []
    countlinks=0

    for i in rjson["items"]:
        if i["is_answered"]:
            url_list.append(i["link"])
        countlinks+=1
        if(countlinks==5 or countlinks==len(rjson["items"])):
            break

    for i in url_list:
        webbrowser.open(i)

```

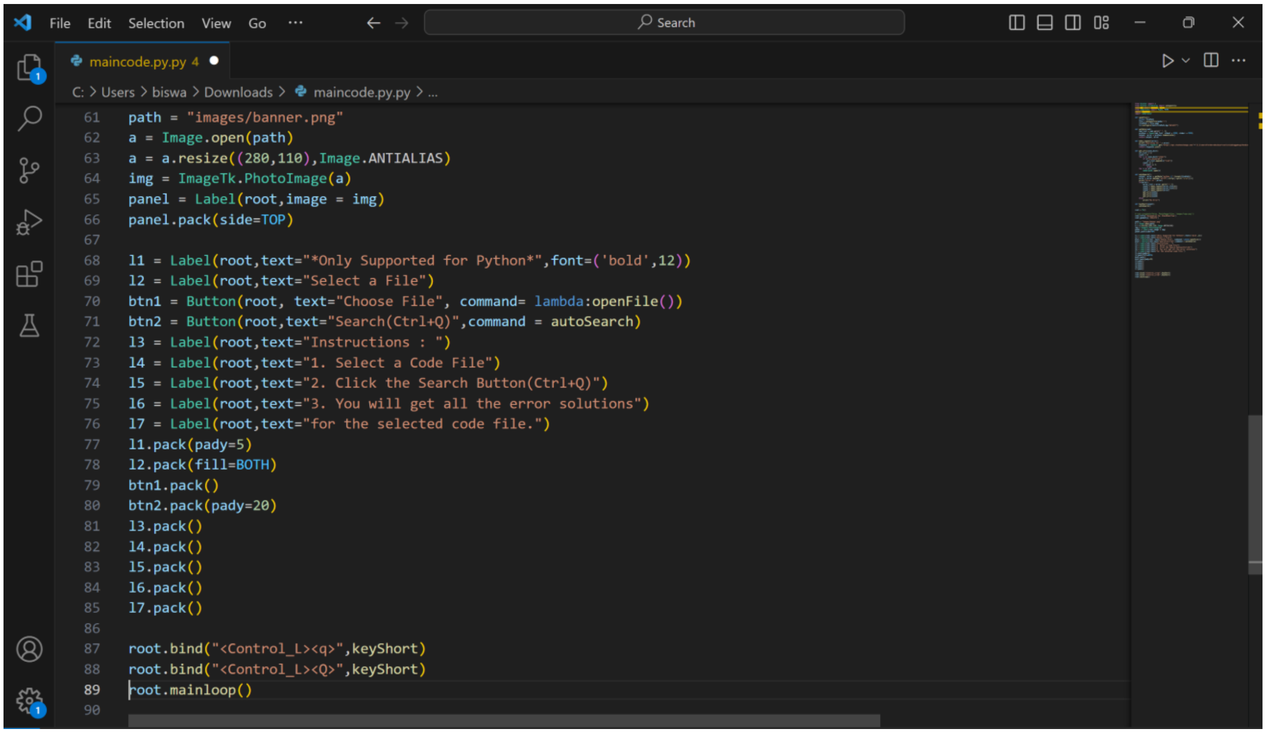
CODE:



```python
from tkinter import *
from tkinter.filedialog import askopenfile
from PIL import ImageTk, Image
from subprocess import Popen, PIPE
import requests
import webbrowser

def openFile():
    global filePath
    file = askopenfile(mode='r')
    filePath = file.name
    l2.configure(text=filePath,bg="#b7e9f7")

def getData(cmd):
    cmd_list = cmd.split(" ", 1)
    process = Popen(cmd_list, stdout = PIPE, stderr = PIPE)
    output, error = process.communicate()
    return output, error

def make_request(error):
    print("Searching for " + error)
    response = requests.get("https://api.stackexchange.com/"+"/2.2/search?order=desc&sort=activity&tagged=python&intitle=
    return response.json()

def get_urls(json_dict):
    url_list = []
    count = 0
    for i in json_dict["items"]:
        if i["is_answered"]:
            url_list.append(i["link"])
        count += 1
```



```python
            url_list.append(i["link"])
        count += 1
        if count == 3:
            break
    for i in url_list:
        webbrowser.open(i)

def autoSearch():
    output, error = getData("python {}".format(filePath))
    error = error.decode("utf-8").strip().split("\r\n")[-1]
    print("Error => ",error)
    if(error):
        error_list = error.split(":",1)
        json1 = make_request(error_list[0])
        json2 = make_request(error_list[1])
        json3 = make_request(error)
        get_urls(json1)
        get_urls(json2)
        get_urls(json3)
    else:
        print("No Error")

def keyShort(event):
    autoSearch()

root = Tk()

# root.iconphoto(False, PhotoImage(file= 'images/logo.png'))
root.title('Autosearch for StackOverflow')
root.geometry('280x370')
```



```python
path = "images/banner.png"
a = Image.open(path)
a = a.resize((280,110),Image.ANTIALIAS)
img = ImageTk.PhotoImage(a)
panel = Label(root,image = img)
panel.pack(side=TOP)

l1 = Label(root,text="*Only Supported for Python*",font=('bold',12))
l2 = Label(root,text="Select a File")
btn1 = Button(root, text="Choose File", command= lambda:openFile())
btn2 = Button(root,text="Search(Ctrl+Q)",command = autoSearch)
l3 = Label(root,text="Instructions : ")
l4 = Label(root,text="1. Select a Code File")
l5 = Label(root,text="2. Click the Search Button(Ctrl+Q)")
l6 = Label(root,text="3. You will get all the error solutions")
l7 = Label(root,text="for the selected code file.")
l1.pack(pady=5)
l2.pack(fill=BOTH)
btn1.pack()
btn2.pack(pady=20)
l3.pack()
l4.pack()
l5.pack()
l6.pack()
l7.pack()

root.bind("<Control_L><q>",keyShort)
root.bind("<Control_L><Q>",keyShort)
root.mainloop()

```

CONCLUSION

In conclusion, the "Autosearch for Stack Overflow" project has demonstrated the potential to significantly enhance developers' productivity and efficiency in finding solutions to their programming queries. Through the implementation of advanced algorithms and seamless integration with Stack Overflow's extensive knowledge base, the system effectively streamlines the search process and provides relevant, contextually-driven results.

The project not only showcases the power of data structures and algorithms in information retrieval but also highlights the importance of leveraging existing community-driven resources like Stack Overflow. By automating and optimizing the search process, developers can now spend more time focusing on actual problem-solving rather than sifting through numerous search results.

Moving forward, there is room for further improvement and expansion. This could include refining the search algorithm, incorporating additional features such as user preferences, and exploring integration with other developer platforms. Additionally, ensuring the system's scalability and robustness will be crucial for handling a growing user base and evolving programming landscapes.

Overall, the "Autosearch for Stack Overflow" project represents a valuable tool for developers seeking quick and reliable answers to their programming questions. It has the potential to become an indispensable asset in the toolkit of programmers across various skill levels and domains, ultimately contributing to a more efficient and collaborative developer community.

REFERENCES: