

easily find the error and their replacement by running this extension.

For the tool to work the developer has to run the code initially. When the code is executed we collect the diagnostics using the vs code api which gives us all the information about the error. We then extract the message and the source language out of it and tweak it into a message to pass it to the stack overflow api.

This API returns a json object containing different queries and discussions related to the message. We format the json object into a user friendly form ,form cards out of the queries with HTML, CSS and JavaScript and show it to the user.

If the user finds a related code snippet and wants it in his editor he can copy and paste the code with a copy button provided in the interface.

3 FUNCTIONING OF THE TOOL

All the functioning of this tool is done with a single command. The command is as follows:

- Command: **StackOverflow View**
- This command should be run after the code is executed.
- When this command is executed a html view inside the editor shows up showing a list view of all the errors.
- Users can select an error and click it.
- When the user clicks a selected error a another html page shows up showing all the formatted json object queries related to that selected error.

4 USE CASE

- Suppose there is a user who is writing code or developing in the vs code editor. He tries to resolve his errors by finding them in google and stack overflow. It can become a hectic task for the developer to switch tabs and look into finding solutions for the errors raised. This is where our tool comes in handy where the user can stay in the editor itself and get the solutions required for his errors from the stack overflow.

```

1 def shortestSubString(A):
2
3     i = 0
4     j = 0
5     result = 0
6
7     n = len(A)
8
9     L = set()
10
11     while i < n and j < n:
12         if A[j] in L:
13             L.remove(A[i])
14             i += 1
15         else:
16             L.add(A[j])
17             j += 1
18             result = max(result, j-i)
19
20     return result
21
22
23
24 print(shortestSubString("dabbcbacd"))
25

```

Figure 2: Sample code with a error and extension activated

- The user just has to run his program and then run our command from the command palette.

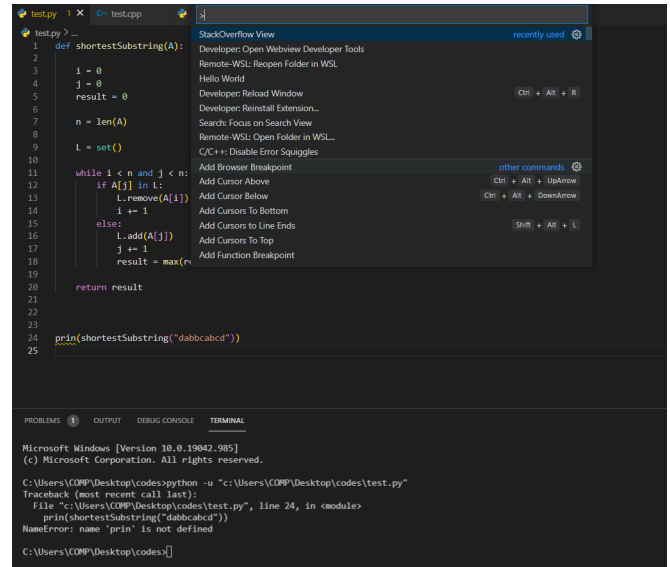


Figure 3: Showing the command to be run after running the program

- The user then can see different errors listed from the code executed.

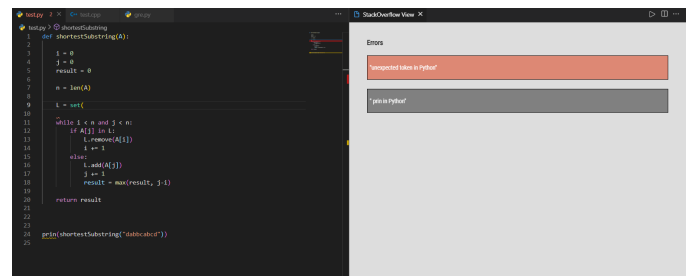


Figure 4: Showing the list view of errors from the diagnostic report of the program

- Now the user can select an error and click it to see the queries related to it from the stack overflow.
- If the user wants that piece of code from the queries shown, a copy button is provided on top of the snippet to copy the code into the clipboard so the user can paste it wherever and whenever required. In this way this tool saves a lot of time for the user and helps in improving his performance by just letting him stay in one tab and keeping focused.

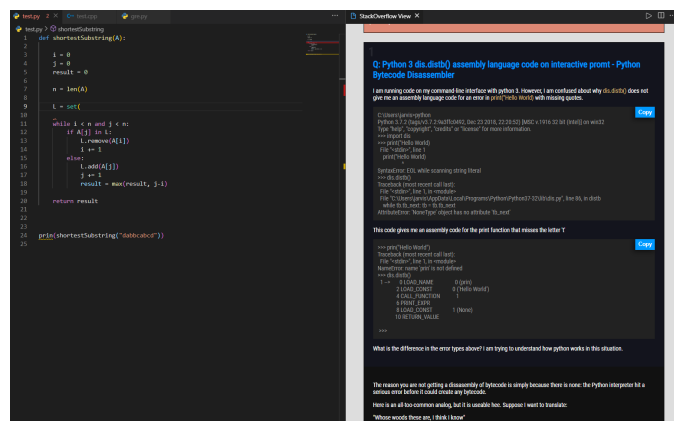


Figure 5: Showing the formatted result of the JSON object received from the stack overflow API

5 LIMITATIONS OF THE TOOL

There are some limitations to this tool. The API used does not guarantee to return accurate results every time and is prone to give useless queries which is a waste of time for the developer. We have limited ourselves to hard code the message generation from the diagnostic report but can be done in a much better way using NLP and DL techniques. This could improve the accuracy of the tool by almost 60 %. Better the queries obtained better is the tool.

6 RELATED WORK

- There is an extension in which all the errors will be shown in the search bar, if we click on any of the shown errors, it will redirect us to the stack overflow website to show us the relevant answer. But there is no panel where we can directly get suggestions in vs code itself and copy it directly to our code.
- There is an extension in which the stack overflow panel will be seen in the vs code itself, but we have to search for the error by typing it manually in the stack overflow search bar.

7 CONCLUSION AND FUTURE WORK

This paper is about the vs code extension tool which gives suggestions to the errors identified. This tool can be improved with applying NLP and DL techniques to improve the message formation. Other improvement can be with error identification whether it is a logical or syntactical error which can be derived by asking some questions about the program or by analysing the program what it does and then doing the error identification. Better query response is also a major aspect in improving the tool. These can be some methods to improve this tool.

8 ARTIFACTS

Github link for the tool : <https://github.com/SaiThanoj/AutoErrorCorrector-Tool/tree/master>

REFERENCES

- Practical syntactic error recovery - Susan L. Graham, Steven P. Rhodes
- Automatic Error Recovery - M.Dennis Mickunas, John A.Modry
- A Practical Approach for Detecting Logical Error - Ghassan Samara
- Common logical errors - Richard Bellairs