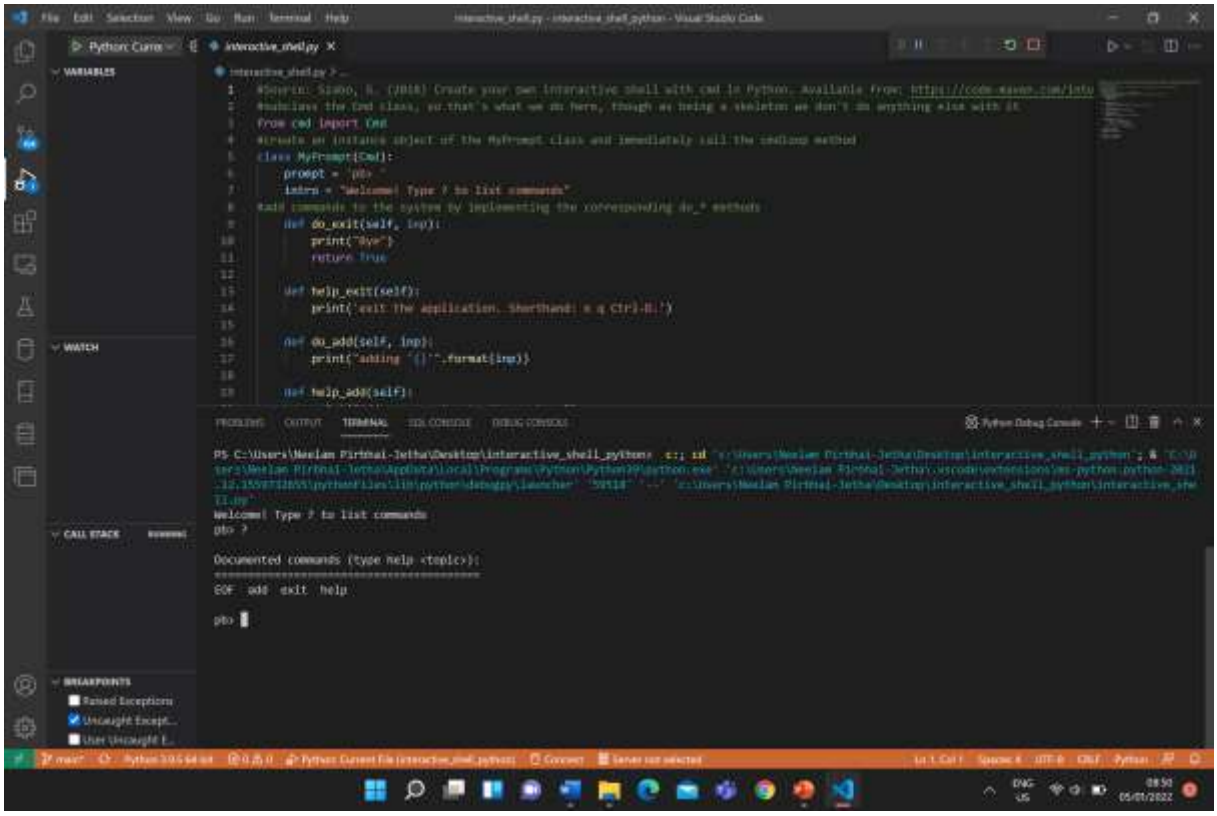
Exploring a simple Python shell

In this session, you will create a command shell in Python, and then run it and answer questions about it. You can use the **Jupyter Notebook space in Codio** for you work. I used VSCODE

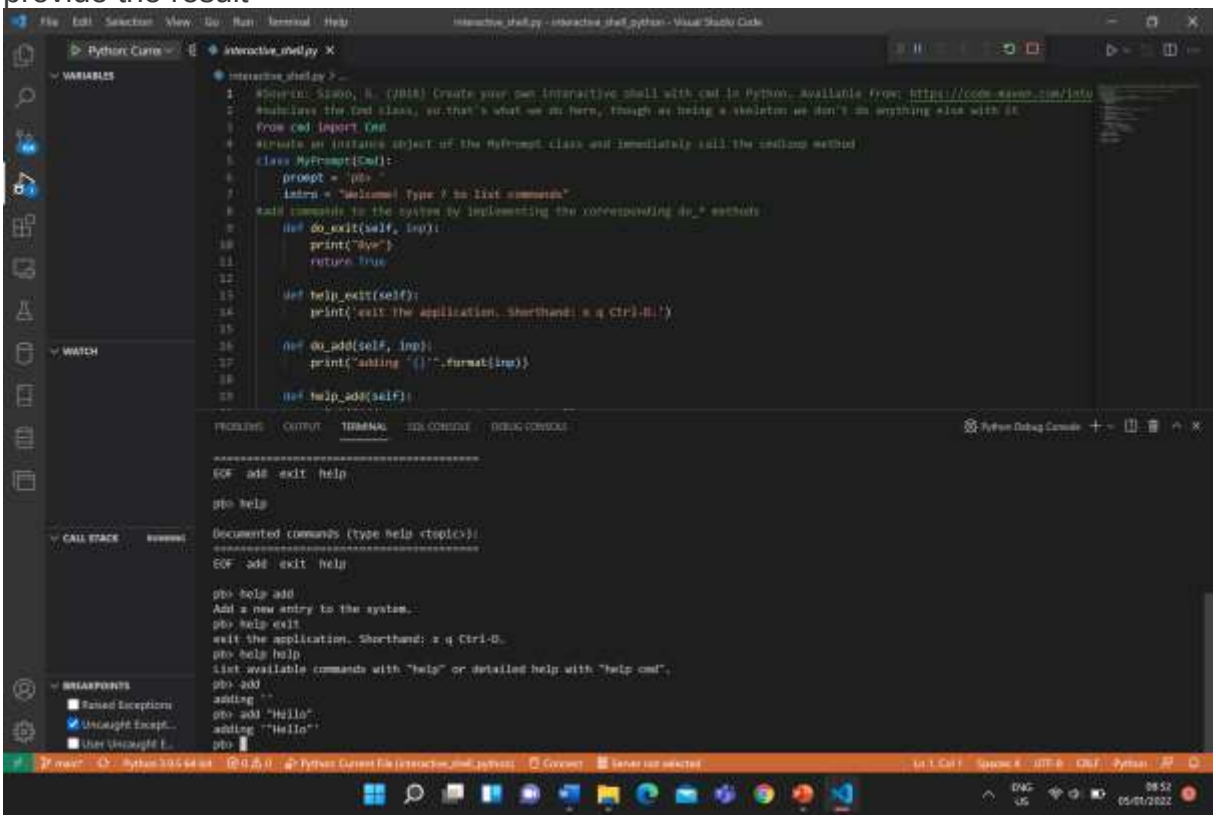
Review the blogs at Praka (2018) and Szabo (n.d.) and then create a CLI/ shell that implements the following:

```
#Source: Szabo, G. (2018) Create your own interactive shell with cmd in
Python. Available from: https://code-maven.com/interactive-shell-with-cmd-in-
python
#subclass the Cmd class, so that's what we do here, though as being a skeleton
we don't do anything else with it
from cmd import Cmd
#create an instance object of the MyPrompt class and immediately call the
cmdloop method
class MyPrompt(Cmd):
    prompt = 'pb> '
    intro = "Welcome! Type ? to list commands"
#add commands to the system by implementing the corresponding do_* methods
    def do_exit(self, inp):
        print("Bye")
        return True

    def help_exit(self):
        print('exit the application. Shorthand: x q Ctrl-D.')

    def do_add(self, inp):
        print("adding '{}'".format(inp))

    def help_add(self):
        print("Add a new entry to the system.")

    def default(self, inp):
        if inp == 'x' or inp == 'q':
            return self.do_exit(inp)

        print("Default: {}".format(inp))

    do_EOF = do_exit
    help_EOF = help_exit

if __name__ == '__main__':
    MyPrompt().cmdloop()
```

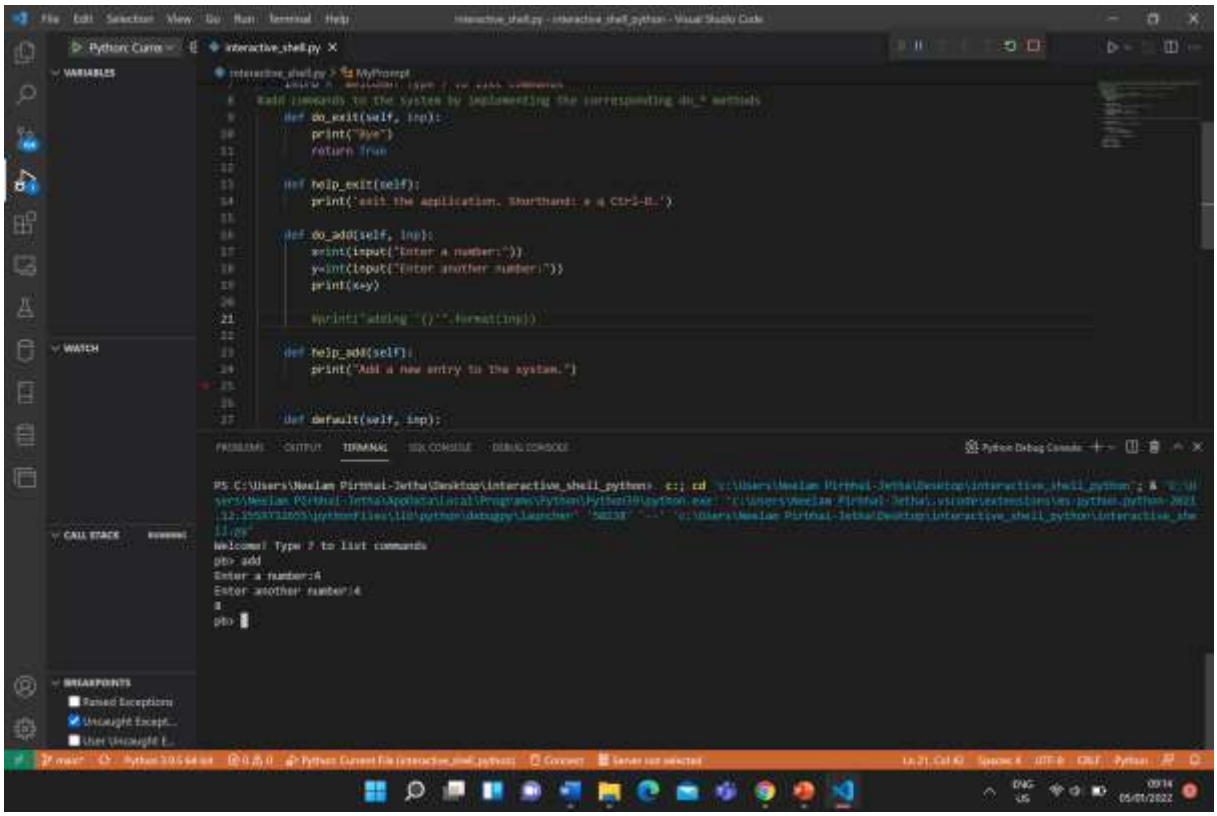
- When you enter the command LIST it lists the contents of the current directory



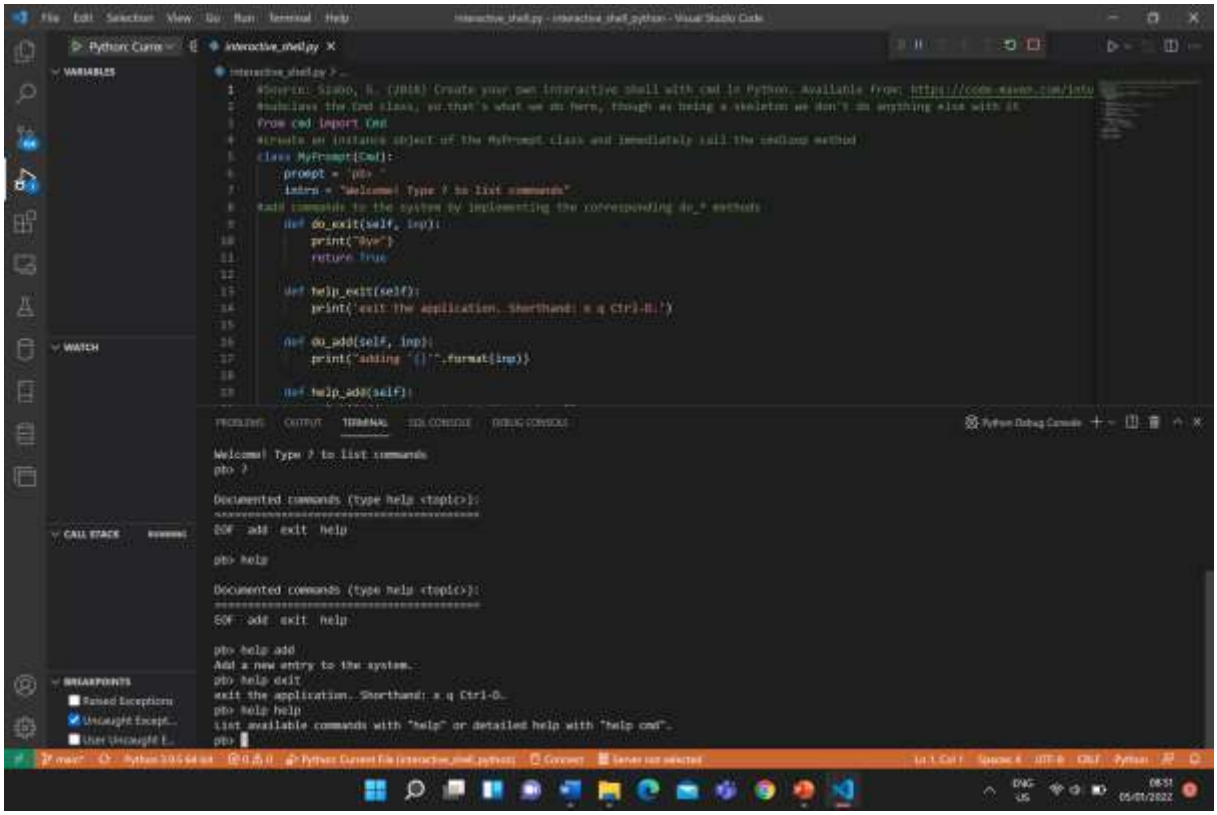
- The ADD command will add the following two numbers together and provide the result



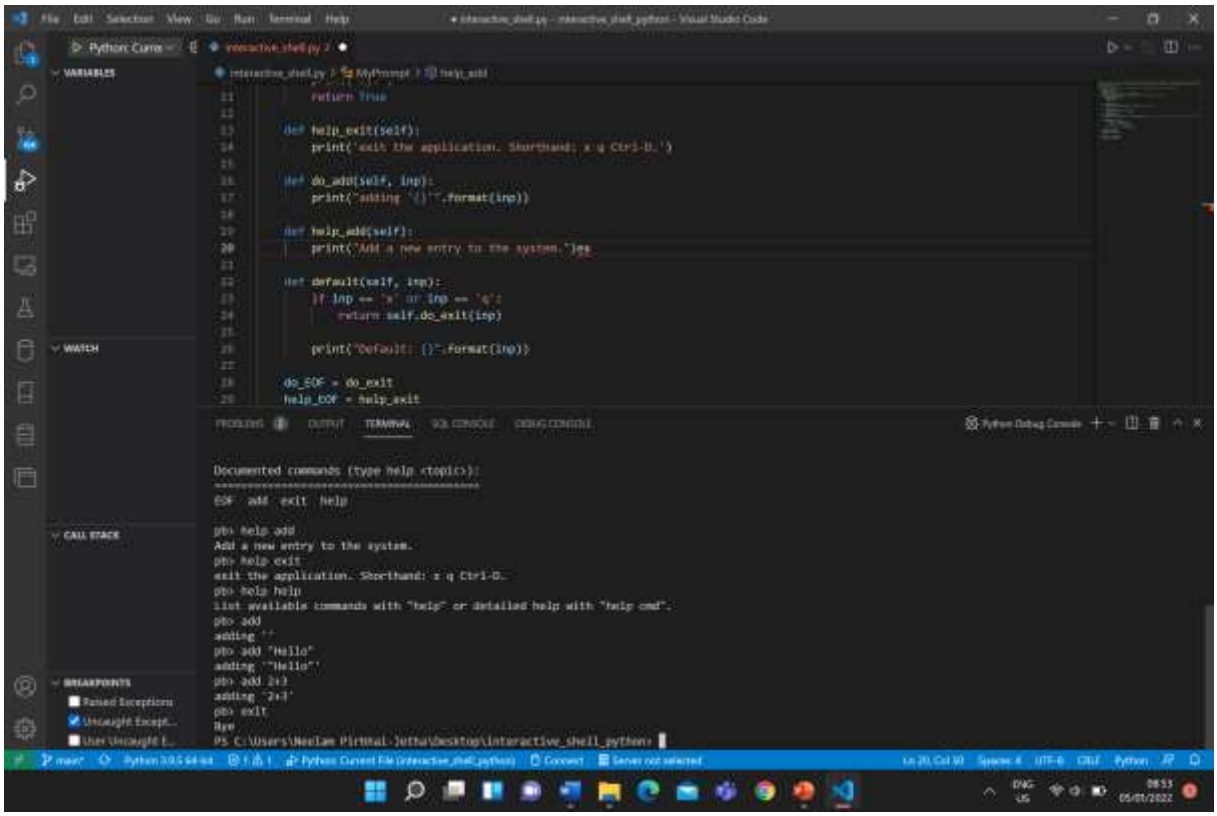
- Change in original codes: to calculate numbers



- The HELP command provides a list of commands available



- The EXIT command exits the shell



Add suitable comments to your code and add the program to your e-portfolio. **Be prepared to demonstrate it in the seminar session next week.**

Run the shell you have created, try a few commands and then answer the questions below. Be prepared to discuss your answers in the seminar.

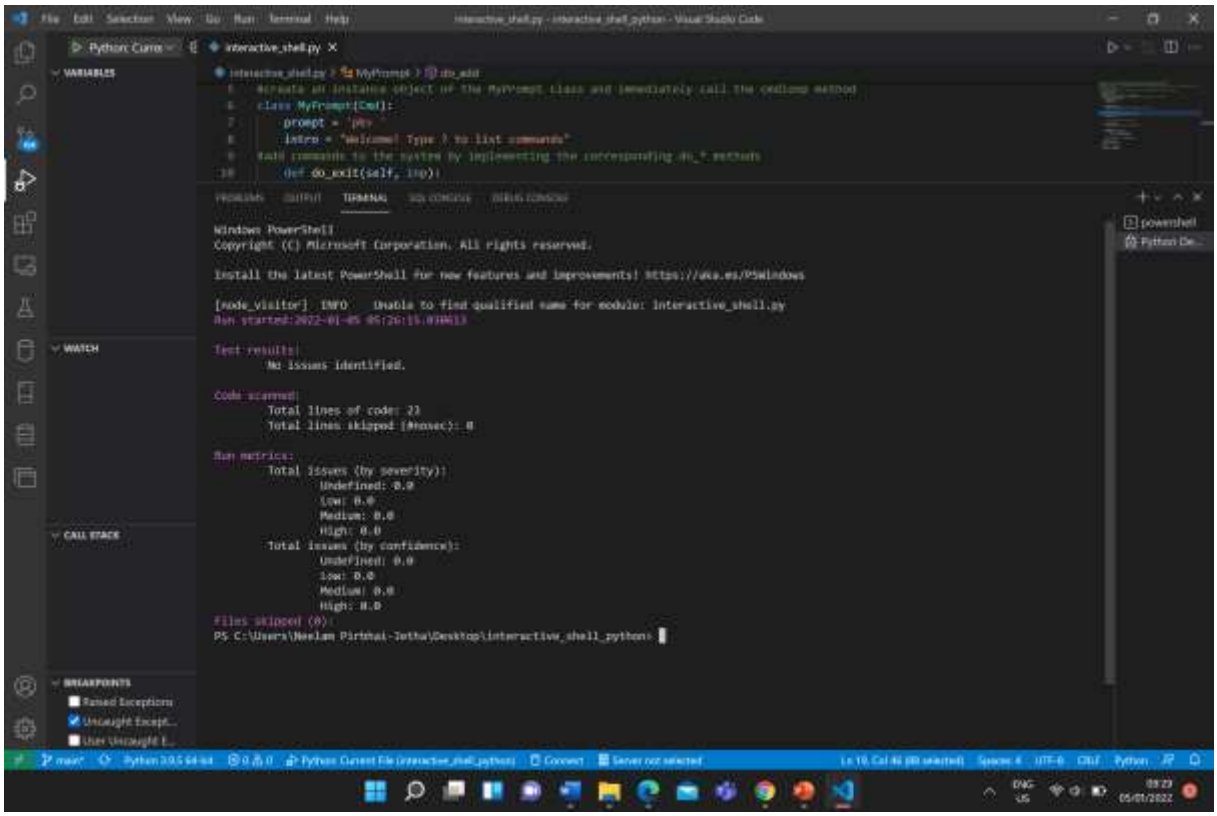
- What are the two main security vulnerabilities with your shell?

To test the vulnerabilities, type:

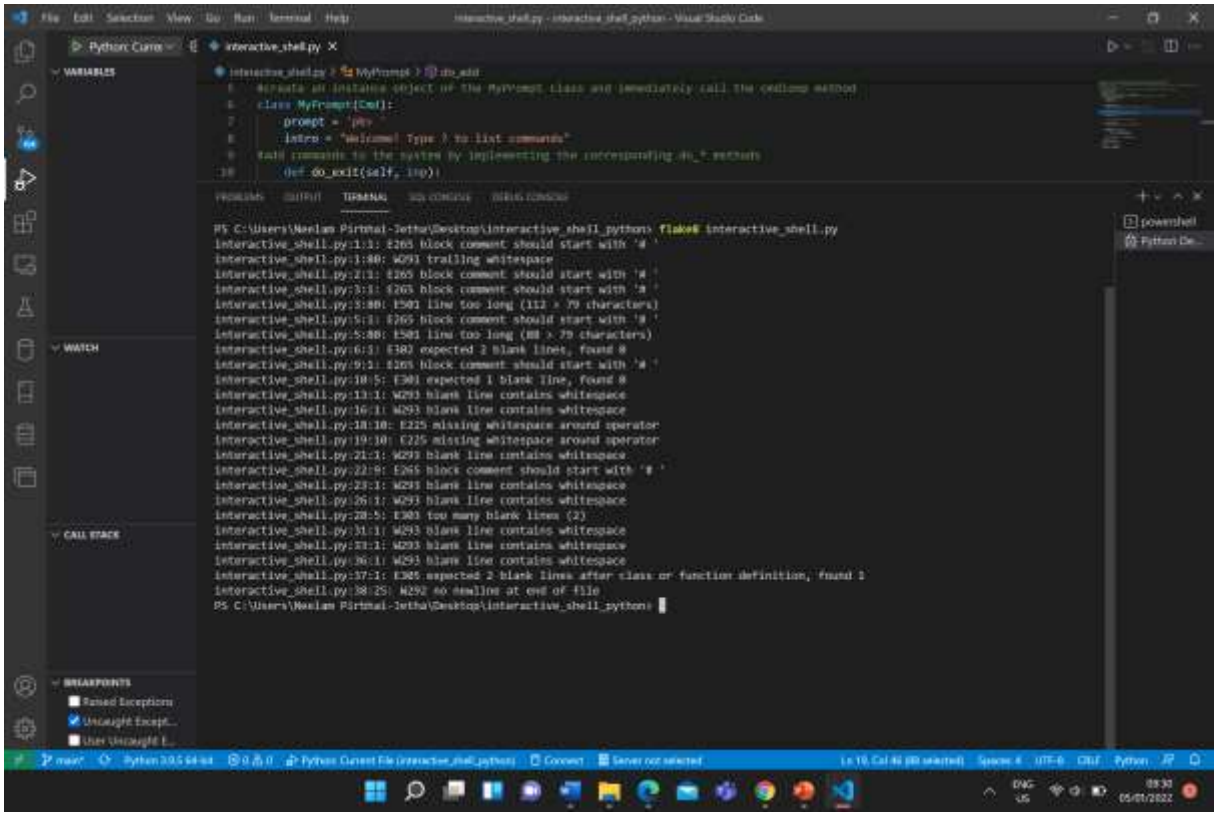
bandit file_name.py

Here, it is

bandit interactive_shell.py



Use flake8 to check



- What is one recommendation you would make to increase the security of the shell?

Review the spaces etc.

But the code taken online seems ok.

- Add a section to your e-portfolio that provides a (pseudo)code example of changes you would make to the shell to improve its security.