Applied Cyber Security

Programming

Assignment 1

Module Code: IY1S408

Module Leader/Tutor: Joshua Richard

Mohammad Saeed Ahmed

Release Date: 10th October, 2022

Submission Date: 13th January 2023

Introduction

To create a to-list for personal tasks. Due to this development of a to-do list written in Python. Will be done to ensure tasks are being documented.

Aims

- [ ] 1.500 word Technical Report
- [ ] Python Application (To-Do Tasks App)
- [ ] Make the program secure

In order to ensure that the tasks will be catalogued, the program will have to design a python application which stores the data in an SQL database as it is a good way of sorting data like this. The database will also need to be password protected to strengthen the security.

Software Development Life Cycle

The life cycle goes as followed: Requirements –> System design –> Development –> Testing –> Deployment and Maintenance.

Software Life Cycle Models are categorised into two categories: heavyweight(Predictive) and Lightweight (Adaptive/Agile).

Predictive

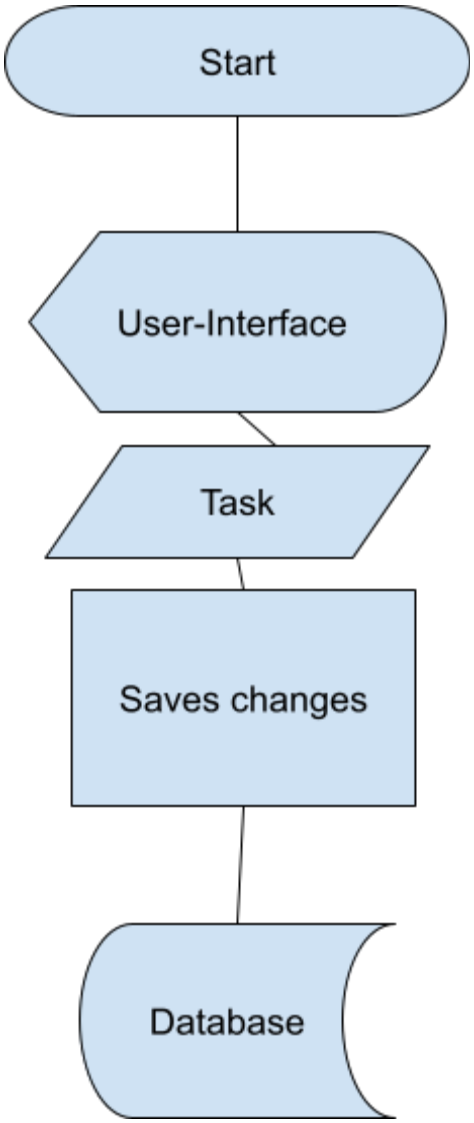
The predictive method is your traditional waterfall method where you complete each step 3 and then continue on. All requirements of each step need to be met before continuing on to the next step. The advantage of this is that you know that each step is fully done and therefore you can move on without worry however due to the nature of this it does result in a longer development cycle.

Agile

Agile is a forward thinking approach to development that is different from the waterfall approach. Rather than completing each stage fully, agile iterative development results in a barebone product, known as a Minimum Viable Product (MVP). This would be very much like a game coming out but lacking the multiplayer on day one and updates will come improving the controls and adding new features.

The goal of Agile is the simplest that people want to use or can fully utilise to their needs. Once the product has been released developers will continue applying the Agile methodology by fixing bugs, sending out updates and patches, making improvements and adding new features.

Flow Chart



Prototype Coding Stage

Pseudocode

#password protection

username is "1234"

password = "pasword"

user's input(*Enter username*)

Userpass is input(*Enter password*)

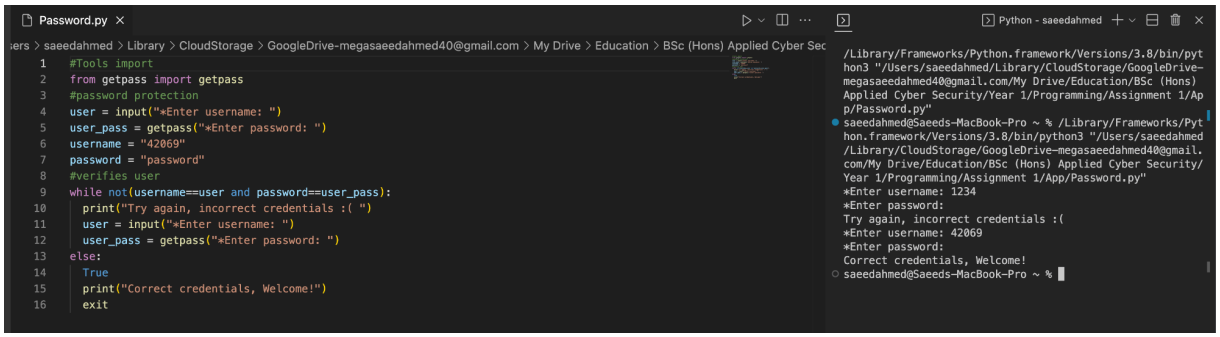
#verifies user

if not(*username equal to user and password equal to userpass*):

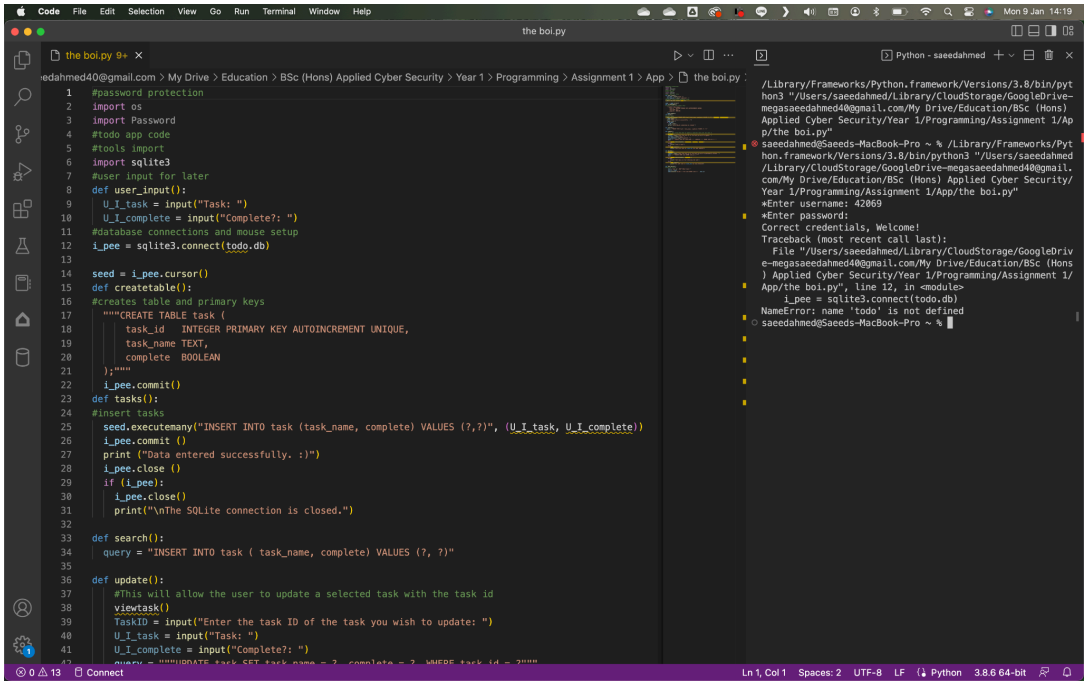
print("Try again, incorrect credentials :(")

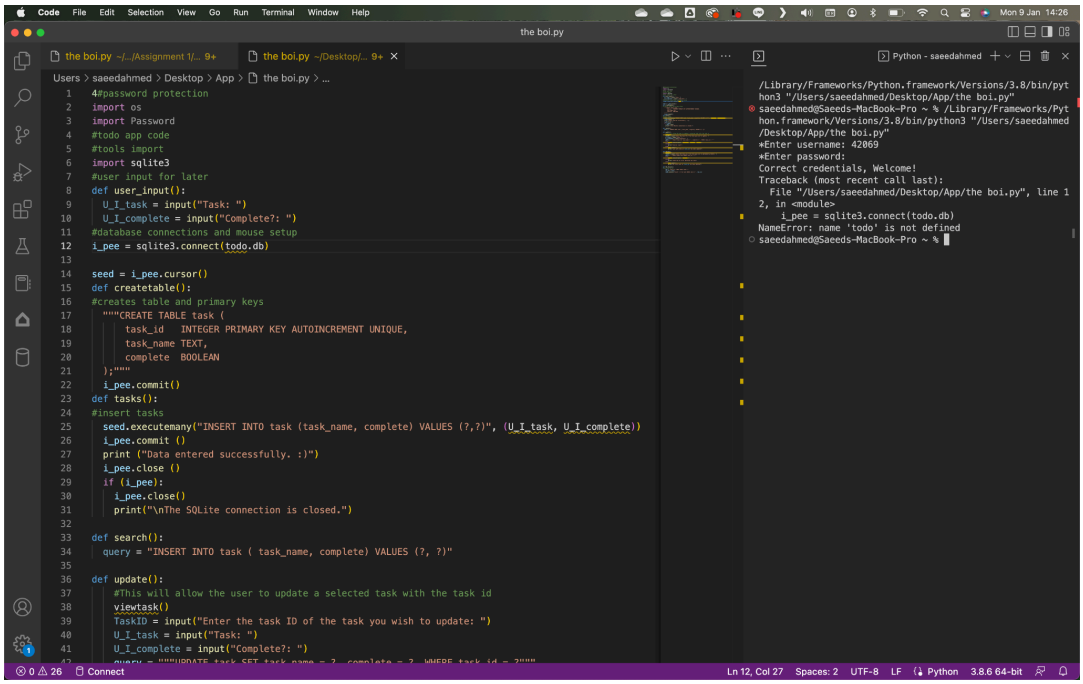
Exit

Coding

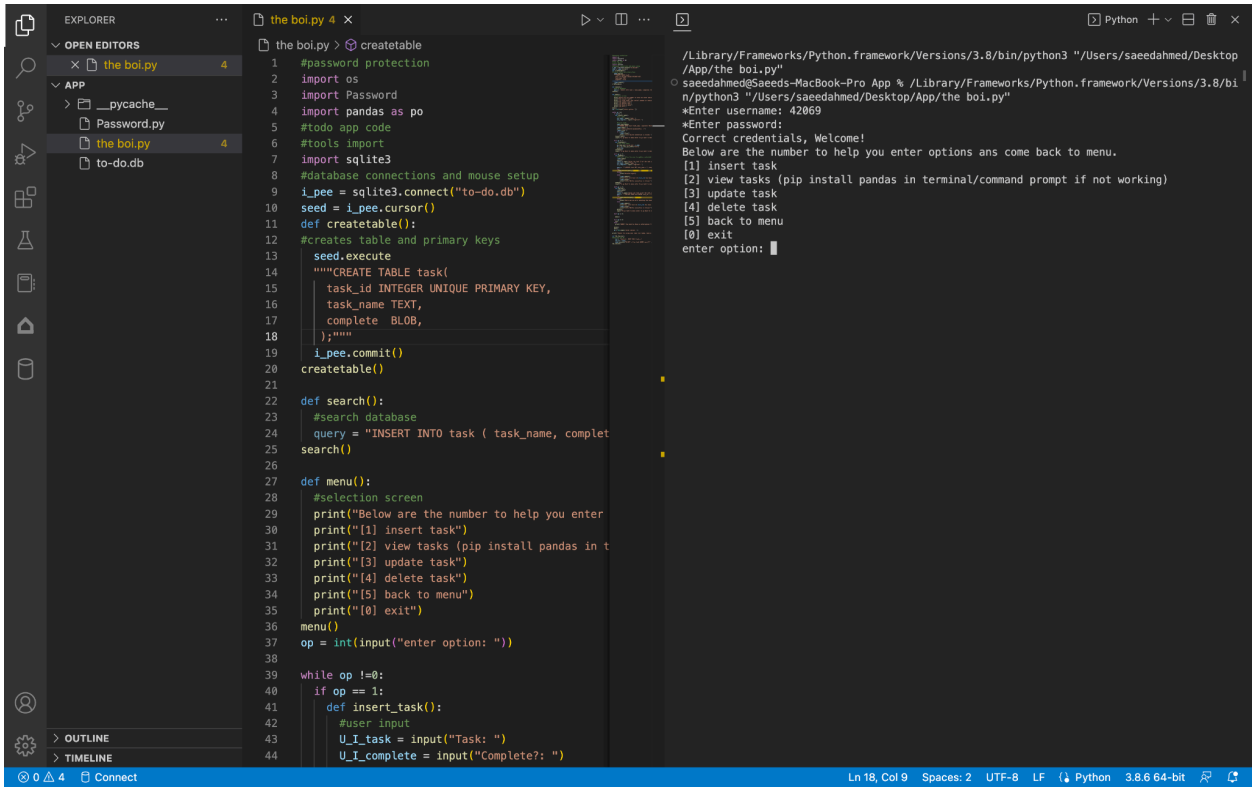


To ensure certain tasks which the user needs to do such as getting a person a present is kept secret, a password protect was implemented. In this the user enters their username and password. If those credentials are not matching then the user will get a message as seen in the image above. This is a separate program which gets imported in to ensure that no confusion is presented.

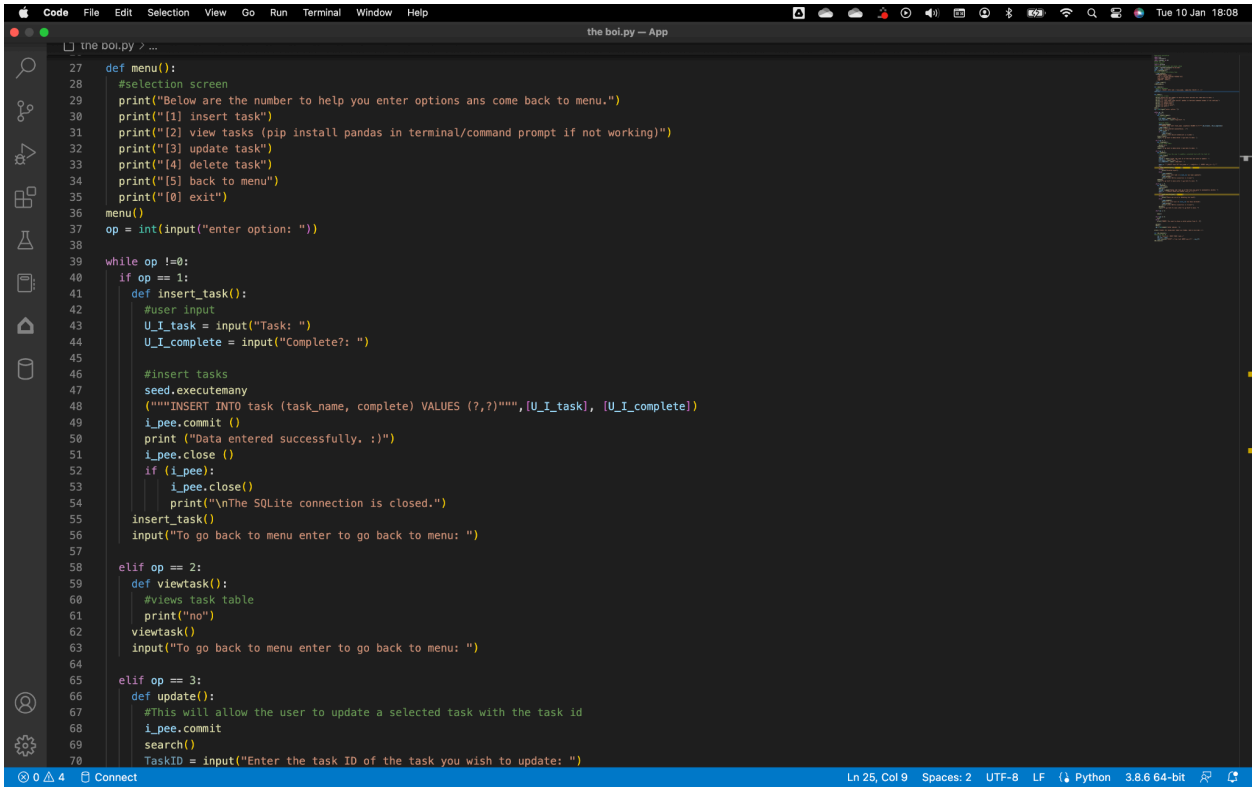




Due to an error of connecting the database file, this could have been due to the use of cloud storage. A second attempt was done with the application on local storage. The results of this showed that. The issue was not due to cloud storage.



After structure in the code into a more organised format, the missing part of the task inserts code with fixed. Thanks to the restructuring of the program a menu was implemented for the sake of ease of use. This vehicle allows the user to have an easier time to update or delete records within the to-do list.



The functionality of the menu system is as follows, application loads it will print the following;

“ Below are the numbers to help you enter options and come back to the menu.

[1] insert task

[2] view tasks (pip install pandas in terminal/command prompt if not working)

[3] update task

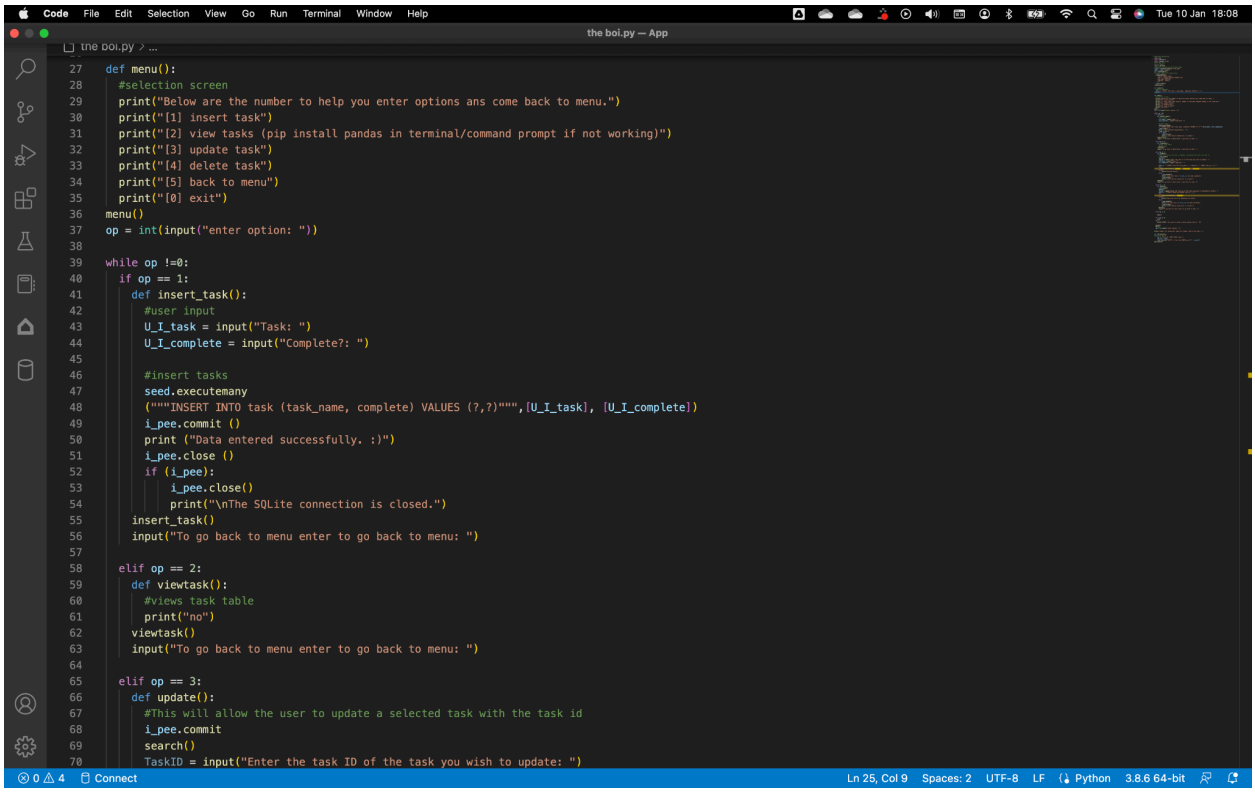
[4] delete task

[5] back to menu

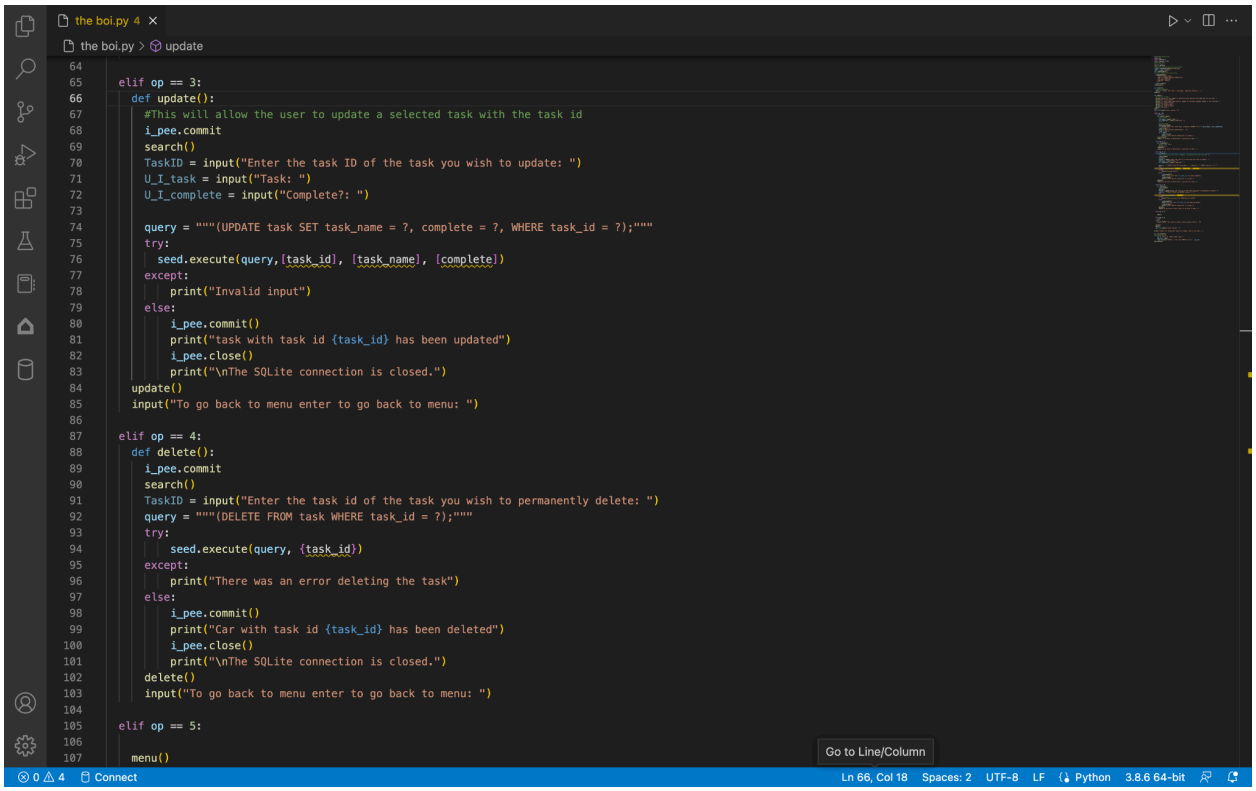
[0] exit

enter option: “

The user would therefore enter one of the numbers listed above for example if they wish to enter a task you would enter one and then it will load into the questions such as what name is a task and is it complete. Once completed with your entry they would press enter on the keyboard and be sent back to the menu where they will select another option.



The insert task code runs when the user enters 1 and insert task uses variables to connect to the columns within the database for example U_I_task which stands for user input task would be BA task that you would enter which will go in the task name column the task table. U_I_complete is the equivalent for the column complete within the table. The complete column within the table uses boolean but because of sqlite lack of support of boolean there was a work around implemented. Blob was used in the place of boolean and is a data type used in SQLite which saves the data as exactly as inputted. Once the data entry is complete, the program will then close the database.



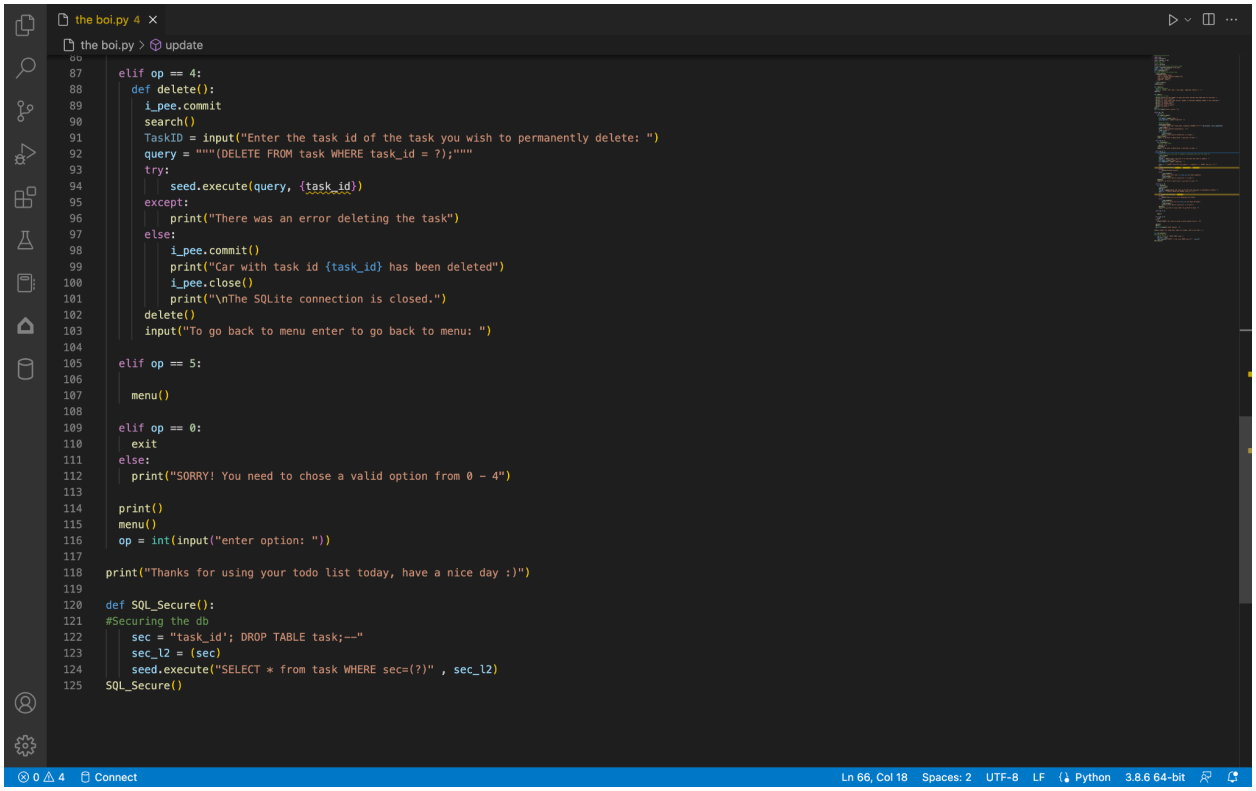
```
elif op == 3:
  def update():
    #This will allow the user to update a selected task with the task id
    i_pee.commit
    search()
    TaskID = input("Enter the task ID of the task you wish to update: ")
    U_I_task = input("Task: ")
    U_I_complete = input("Complete?: ")

    query = """(UPDATE task SET task_name = ?, complete = ?, WHERE task_id = ?);"""
    try:
      seed.execute(query,[task_id], [task_name], [complete])
    except:
        print("Invalid input")
    else:
        i_pee.commit()
        print("task with task id {task_id} has been updated")
        i_pee.close()
        print("\nThe SQLite connection is closed.")
  update()
  input("To go back to menu enter to go back to menu: ")

elif op == 4:
  def delete():
    i_pee.commit
    search()
    TaskID = input("Enter the task id of the task you wish to permanently delete: ")
    query = """(DELETE FROM task WHERE task_id = ?);"""
    try:
        seed.execute(query, {task_id})
    except:
        print("There was an error deleting the task")
    else:
        i_pee.commit()
        print("Car with task id {task_id} has been deleted")
        i_pee.close()
        print("\nThe SQLite connection is closed.")
  delete()
  input("To go back to menu enter to go back to menu: ")

elif op == 5:

  menu()
```

The update task code runs when the user enters 3. Once the code runs, the user will then be required to the task ID of the task they wish to update. For example:

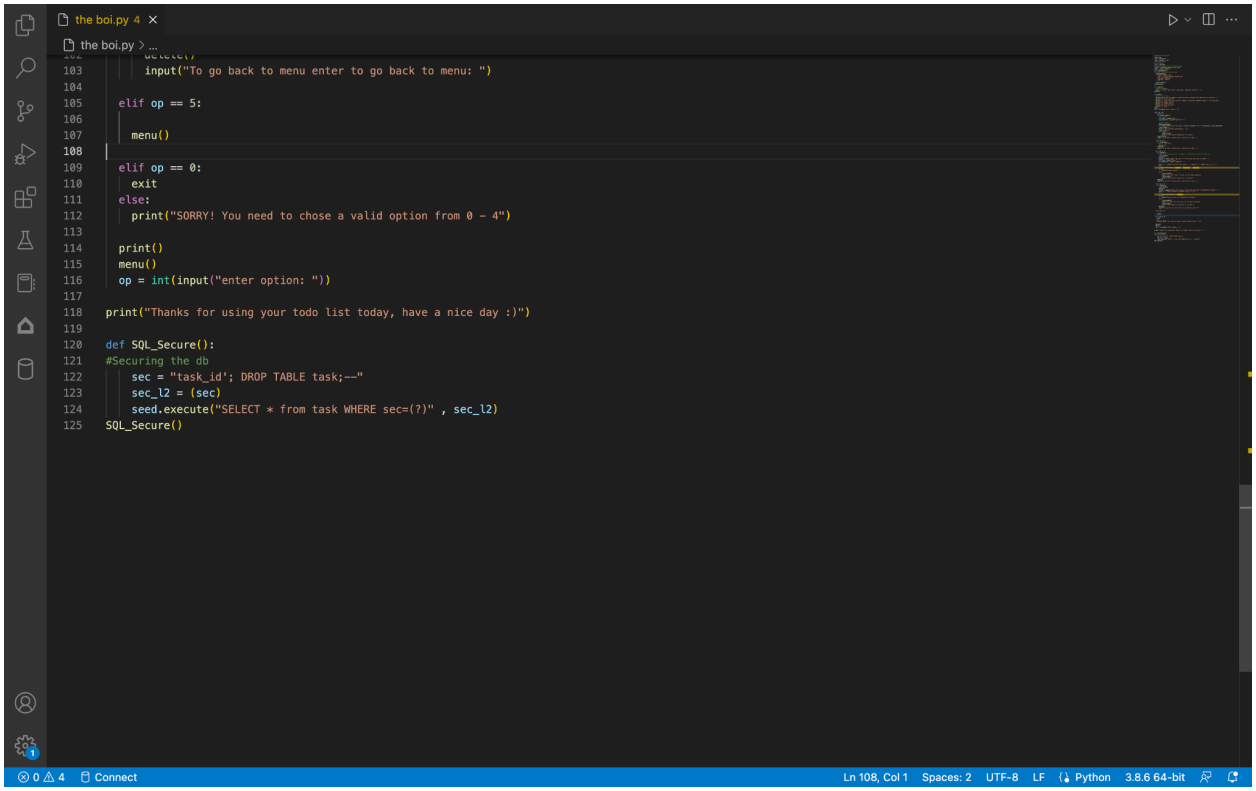
Task ID	Task Name	Complete
1	Play sonic	☐

The user wants to change the task above, so they will be asked by the program. “Enter the task ID of the task you wish to update:” They will then enter 1 in this example. The program will respond with “Task:” and now the user will now enter the new task to overwrite the pre existing task, this plays out similar to when they first entered the original task. Once completed, the new task within the table will look like this.

Task ID	Task Name	Complete
1	Drink coffee	☑



Option 4 in the menu is delete. The function of these works very similar to update as it runs research using the task ID in by the user. When it is found, the operation of the code will then delete the record within the database. to ensure that the deletion is correct the primary key that is the task ID was utilised.



Option 0 is a very simple operation within the program as it simply just exits to use it from the application. option 5 is an option which is not actually listed on the menu as it primarily to allow users to exit from option 1-4 the primary menu of the program however simply just inputting the enter button returns the user back to the menu making this redundant. But In the case where pressing enter does not work, using will be then sent back. to ensure that users are only entering lifted anything else other than option 0,1, 2, 3, 4 and 5 the user will be presented with an error message.

References

Dolinski, A. (2020) *Making a menu in Python*. YouTube. Available at: https://www.youtube.com/watch?v=63nw00JqHo0&t=20s (Accessed: January 10, 2023).

Drumond, C. (2011) *What is agile?*, *Atlassian*. Available at: https://www.atlassian.com/agile#:~:text=Agile%20is%20an%20iterative%20approach,small%2C%20but%20consumable%2C%20increments (Accessed: Accessed: November 1, 2022).

Galli, K. (2018) *Complete python pandas data science tutorial! (reading CSV/Excel files, sorting, filtering, Groupby)*, *YouTube*. YouTube. Available at: https://www.youtube.com/watch?v=vmEHCJofslg&t=742s (Accessed: December 13, 2022).

IT, P. (2022) *How to create password protected file in Python (8 lines only!) | mr. Perfect it*, *YouTube*. YouTube. Available at: https://www.youtube.com/watch?v=BsXSl66zaZk&t=113 (Accessed: January 10, 2023).

unknown, unknown (2022) *1. Datatypes in SQLite*, *Datatypes In SQLite*. Available at: https://www.sqlite.org/datatype3.html (Accessed: January 10, 2023).

Yurevich, D. (2020) *Adaptive vs. Predictive Software Development*. Syberry. Available at: https://www.syberry.com/blog/predictive-vs-adaptive-development (Accessed: November 1, 2022).

Appendix

1. See……Test Plan
2. See……the boi.py