

Objective:

This goal of this program was to show the proper use of sets, lists, dictionaries. This was accomplished by using several packages such as string, sys, requests, Image from PIL, pyplot from matplotlib. The program starts off with displaying a menu in which the user will enter an item that they would like to perform. The options would be display all states in the U.S. in alphabetical order with their state bird, flower, capital and population, choose which state they would like to see with their capital, flower, bird and population, to display the top 5 states with the highest population, to update a states population, and finally to exit the program.

Test Cases:

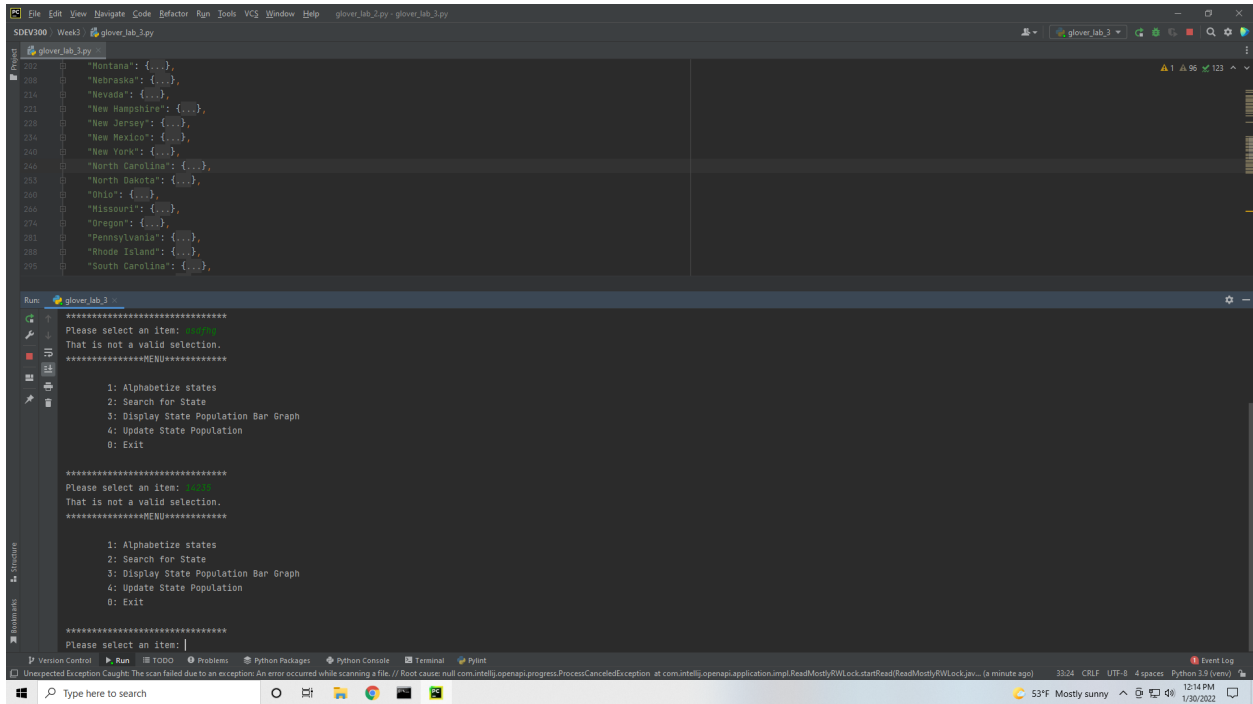
Below is the table used for test cases and the predicted vs actual output.

Test Cases:	User Input:	Predicted Output:	Actual Output:	Pass/Fail:
1	random key input	Invalid selection	Invalid selection	Pass, fig 1
2	14235	Invalid selection	Invalid selection	Pass, fig 1
3	1	Display all states with flower, bird, capital, population	Display all states with flower, bird, capital, population	pass, fig 2
4	2	Enter state you would like to search for:	Enter state you would like to search for:	pass, fig 3
5	newyork	Please verify input	Please verify input	pass, fig 3
6	NY	Please verify input	Please verify input	pass, fig 3
7	ny	Please verify input	Please verify input	pass, fig 3
8	2345	Please verify input	Please verify input	pass, fig 3
9	new York	Display state chosen, capital, bird, flower, population, new window with state flower	Display state chosen, capital, bird, flower, population, new window with state flower	pass, fig 3
10	3	Display 'Top 5 State Populations: followed by 5 states with the highest population value and their populations. Pop out graph of corresponding populations from highest to lowest	Display 'Top 5 State Populations: followed by 5 states with the highest population value and their populations. Pop out graph of corresponding populations from highest to lowest	pass, fig 4

11	4	Enter state you would like to search for:	Enter state you would like to search for:	pass, fig 5
12	state, 12435, lousiana, LA	Please verify state input	Please verify state input	pass, fig 5
13	new york	display state entered and population followed by Enter new population:	display state entered and population followed by Enter new population:	pass, fig 5
14	1thousand, absd	Please verify new population. Enter state:	Please verify new population. Enter state:	pass, fig 5
15	new york	display state entered and population followed by Enter new population:	display state entered and population followed by Enter new population:	pass, fig 5
16	1	display state entered and new population entered	display state entered and new population entered	pass, fig 5
17	3	Display 'Top 5 State Populations: followed by 5 states with the highest population value and their populations. Pop out graph of corresponding populations from highest to lowest	Display 'Top 5 State Populations: followed by 5 states with the highest population value and their populations. Pop out graph of corresponding populations from highest to lowest	pass, fig 6
18	0	Thank you for using program, exit	Thank you for using program, exit	pass, fig 7

Test Case Screenshots:

Below are some screenshots of the file running and its corresponding output as referenced in the case study.



The screenshot shows a VS Code editor window with a file named 'glover_lab_3.py'. The code is a Python script that defines a dictionary of US states and their populations. It includes a menu-driven interface for the user to interact with the data. The Run console shows the output of the script, which prompts the user to select an item from the menu. The user has entered '1' and '2', and the script has responded accordingly.

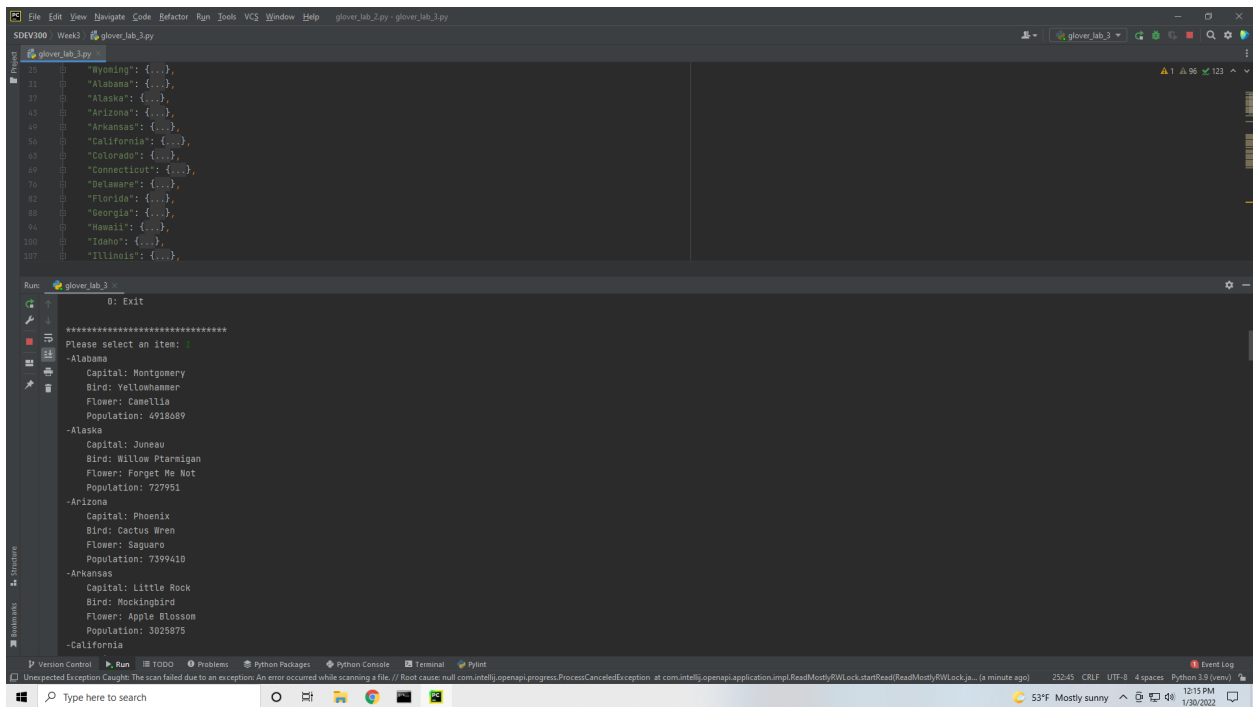
```
202 | "Montana": {...},
203 | "Nebraska": {...},
214 | "Nevada": {...},
221 | "New Hampshire": {...},
228 | "New Jersey": {...},
234 | "New Mexico": {...},
240 | "New York": {...},
246 | "North Carolina": {...},
253 | "North Dakota": {...},
260 | "Ohio": {...},
266 | "Missouri": {...},
274 | "Oregon": {...},
281 | "Pennsylvania": {...},
288 | "Rhode Island": {...},
295 | "South Carolina": {...},

Run: glover_lab_3
Please select an item: 1
That is not a valid selection.
*****MENU*****
1: Alphabetize states
2: Search for State
3: Display State Population Bar Graph
4: Update State Population
0: Exit

Please select an item: 2
That is not a valid selection.
*****MENU*****
1: Alphabetize states
2: Search for State
3: Display State Population Bar Graph
4: Update State Population
0: Exit

Please select an item: |
```

Fig. 1



The screenshot shows a VS Code editor window with a file named 'glover_lab_3.py'. The code is a Python script that defines a dictionary of US states and their populations. It includes a menu-driven interface for the user to interact with the data. The Run console shows the output of the script, which prompts the user to select an item from the menu. The user has entered '0' and '1', and the script has responded accordingly.

```
25 | "Wyoming": {...},
31 | "Alabama": {...},
37 | "Alaska": {...},
43 | "Arizona": {...},
49 | "Arkansas": {...},
55 | "California": {...},
61 | "Colorado": {...},
67 | "Connecticut": {...},
73 | "Delaware": {...},
79 | "Florida": {...},
85 | "Georgia": {...},
91 | "Hawaii": {...},
97 | "Idaho": {...},
103 | "Illinois": {...},

Run: glover_lab_3
0: Exit

Please select an item: 1
- Alabama
  Capital: Montgomery
  Bird: Yellowhammer
  Flower: Camellia
  Population: 4918689
- Alaska
  Capital: Juneau
  Bird: Willow Ptarmigan
  Flower: Forget Me Not
  Population: 727951
- Arizona
  Capital: Phoenix
  Bird: Cactus Wren
  Flower: Saguaro
  Population: 7399410
- Arkansas
  Capital: Little Rock
  Bird: Mockingbird
  Flower: Apple Blossom
  Population: 3025875
- California
```

Fig. 2

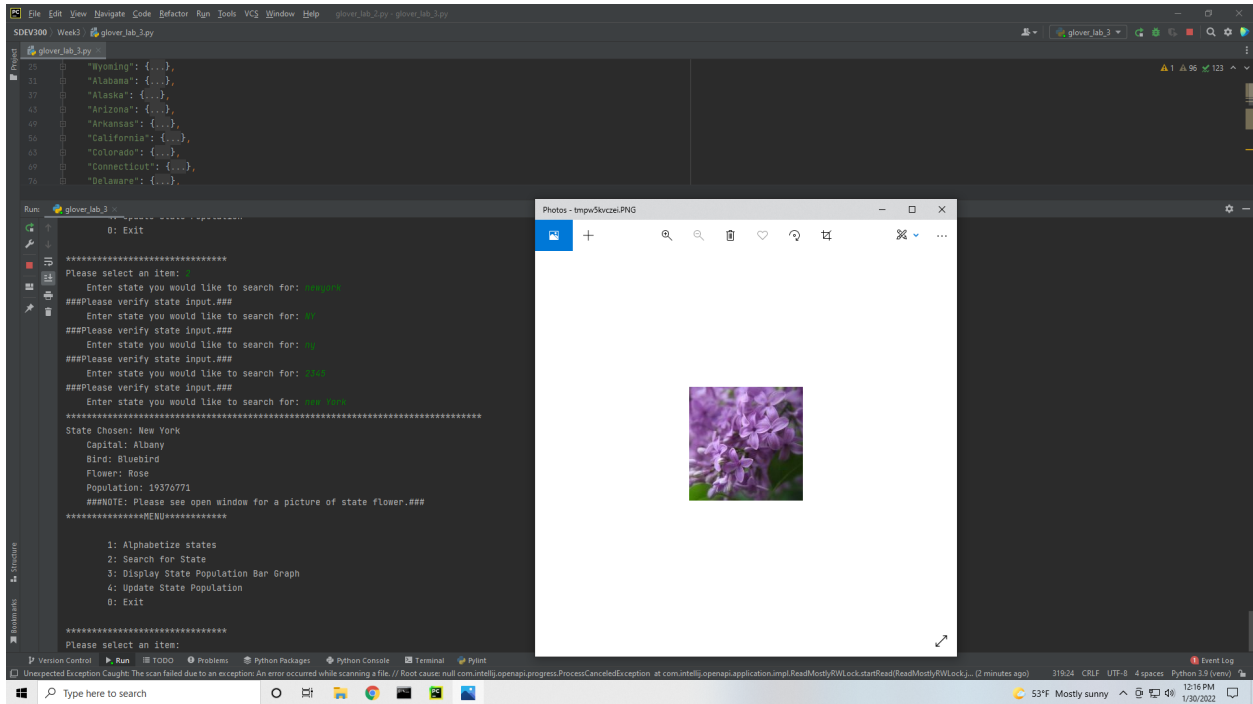


Fig. 3

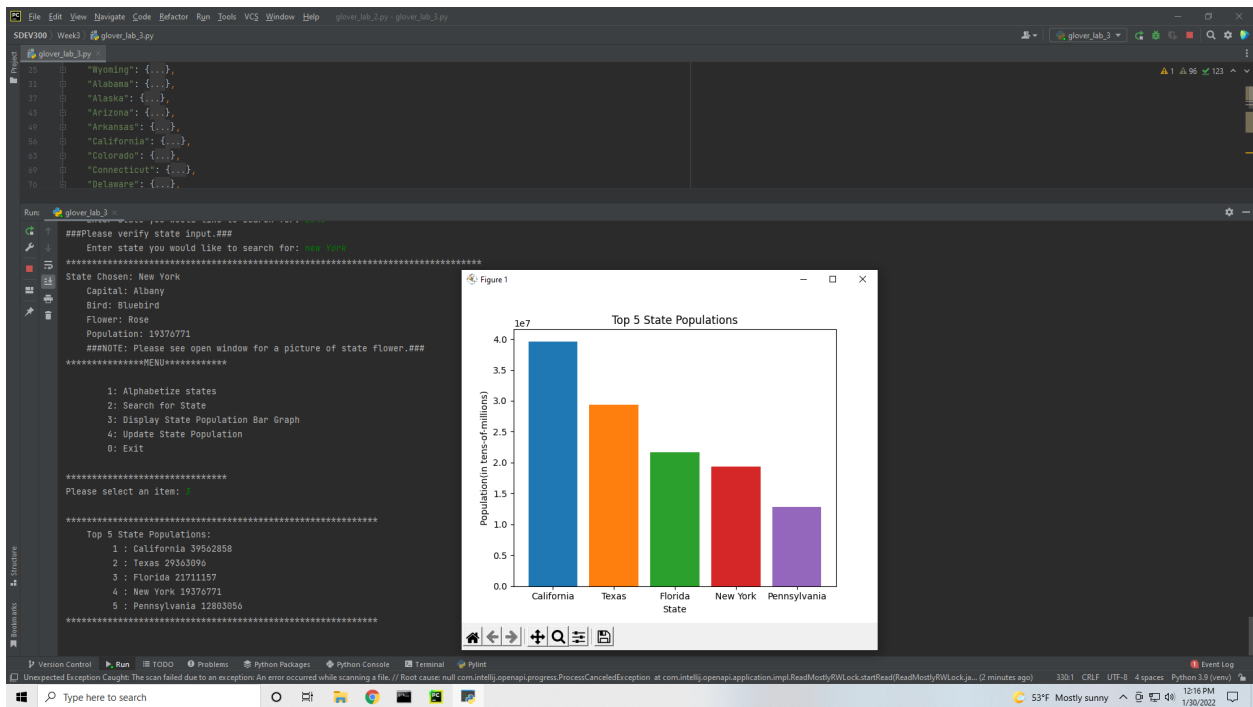
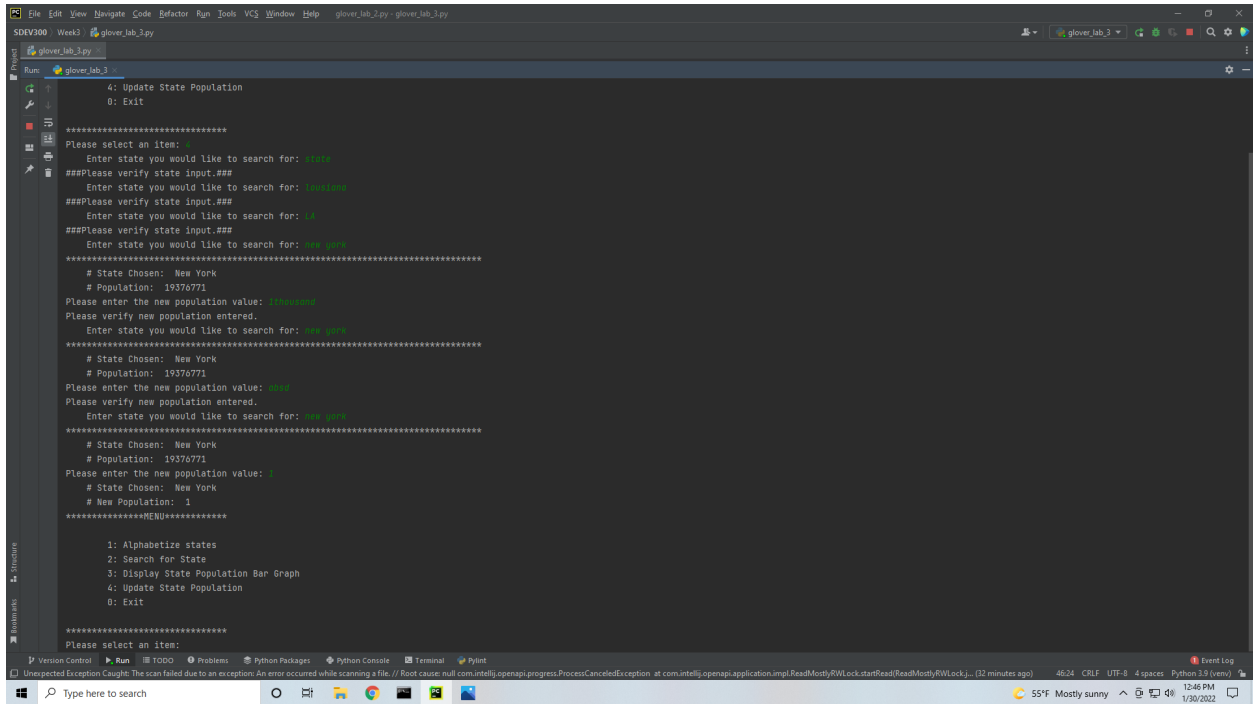


Fig. 4



```
4: Update State Population
0: Exit

*****
Please select an item:
Enter state you would like to search for: NY
***Please verify state input***
Enter state you would like to search for: NY
***Please verify state input***
Enter state you would like to search for: NY
***Please verify state input***
Enter state you would like to search for: New York
*****
# State Chosen: New York
# Population: 19376771
Please enter the new population value: 19376771
Please verify new population entered.
Enter state you would like to search for: New York
*****
# State Chosen: New York
# Population: 19376771
Please enter the new population value: New
Please verify new population entered.
Enter state you would like to search for: New York
*****
# State Chosen: New York
# Population: 19376771
Please enter the new population value:
# State Chosen: New York
# New Population: 1
*****MENU*****

1: Alphabetize states
2: Search for State
3: Display State Population Bar Graph
4: Update State Population
0: Exit

*****
Please select an item:
```

Fig. 5

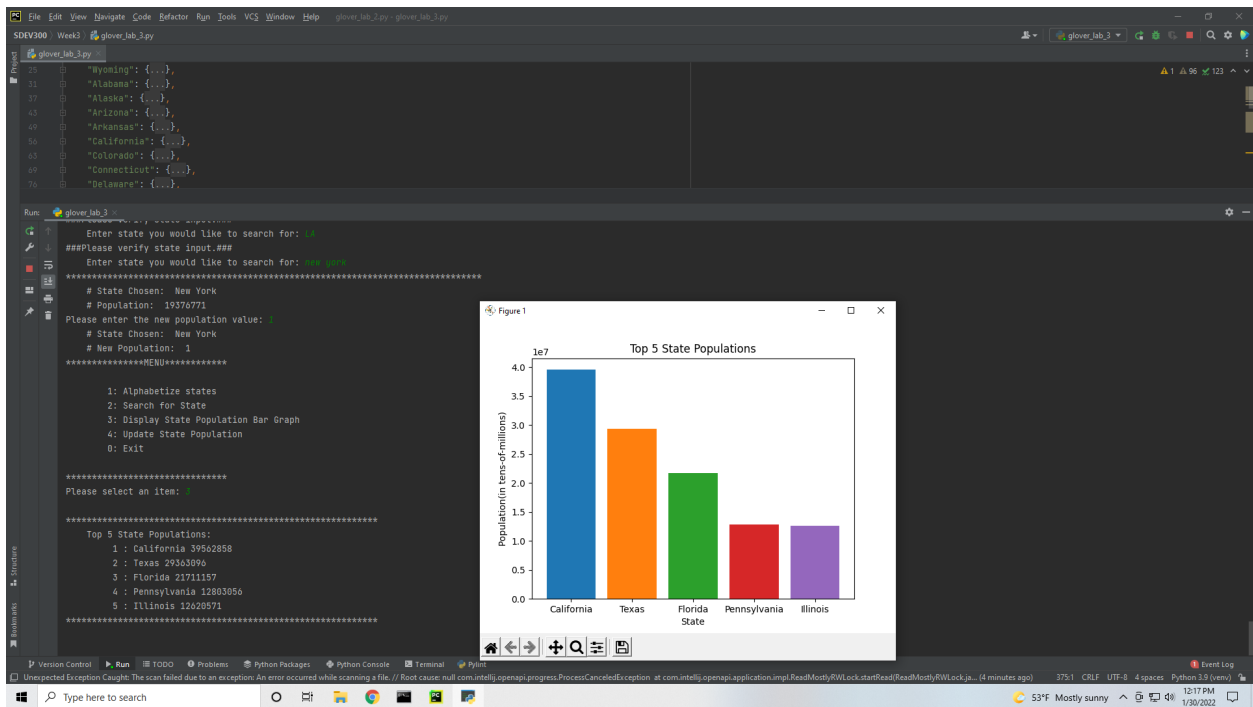


Fig. 6

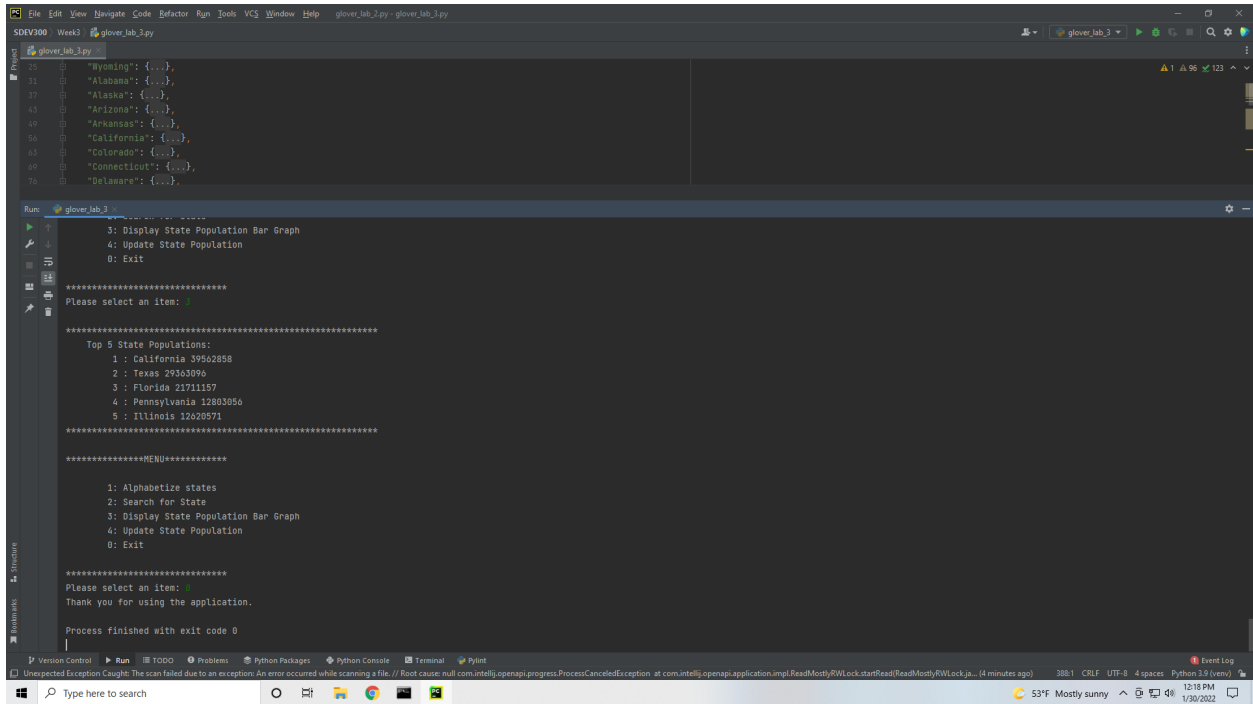
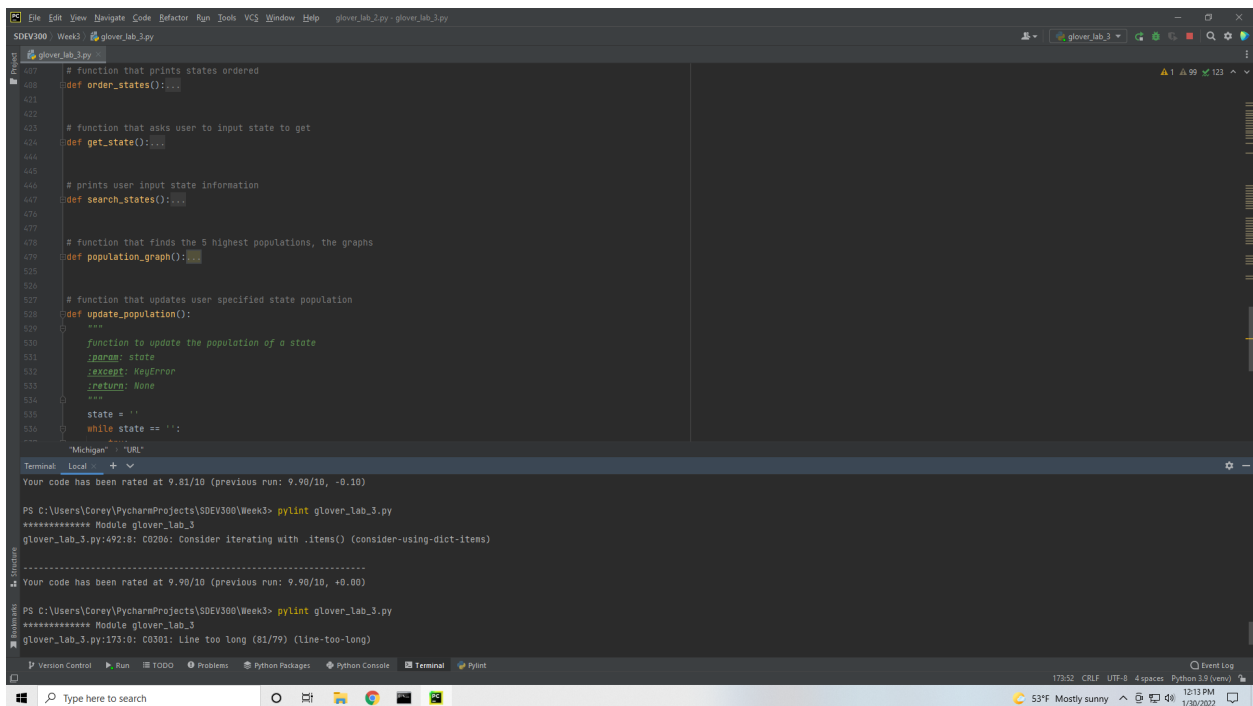


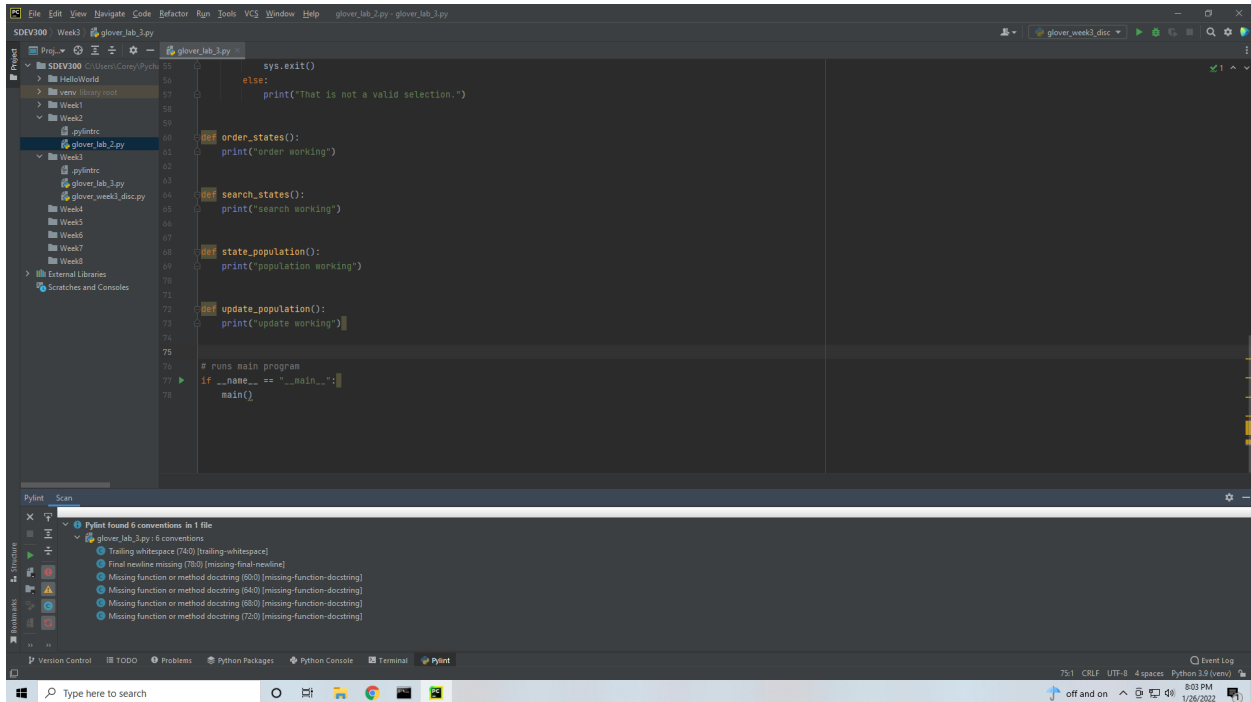
Fig. 7

Correction of Pylint with Screenshots:

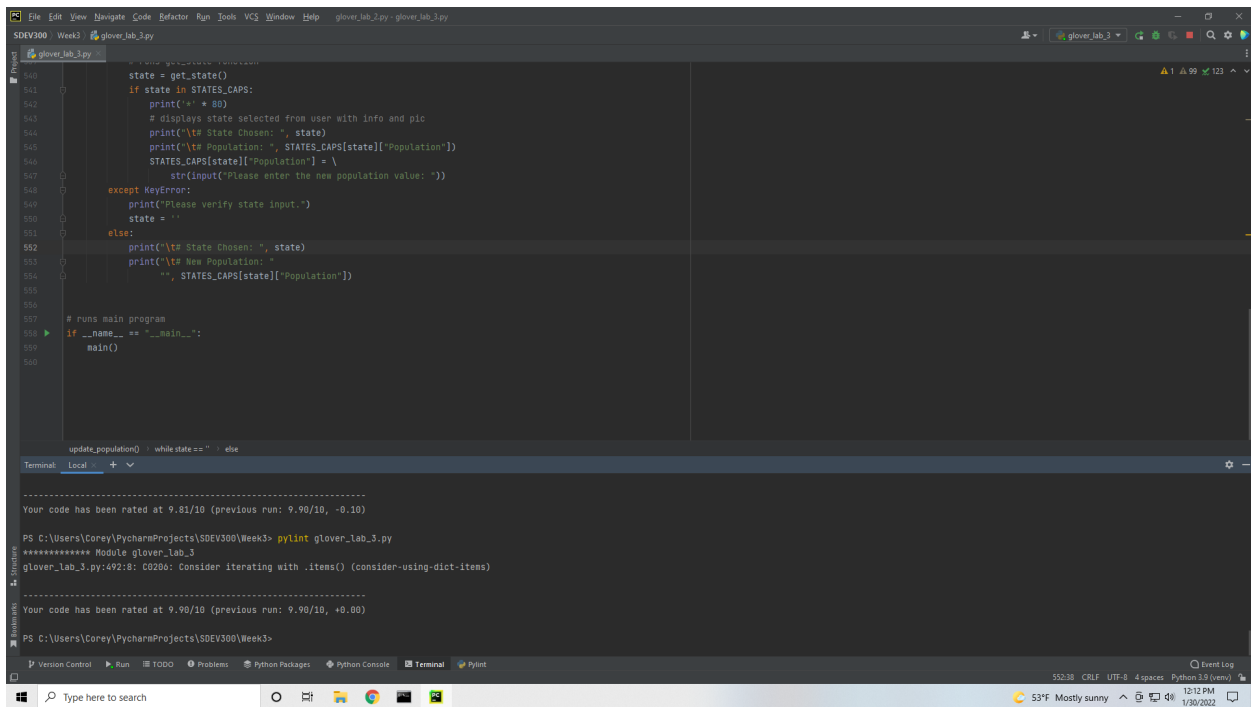
Below are screenshots of pylint with information on how they were handled/corrected.



A common issue was trailing whitespace. This was corrected by backspacing until the line was on the previous line then returning so pycharm would auto-format the line.



Missing final newline was corrected by adding new line after `main()` line.



This final issue from pylint suggested adding `.item()` to line. If added this causes multiple issues in running the program effectively so it was left as is.