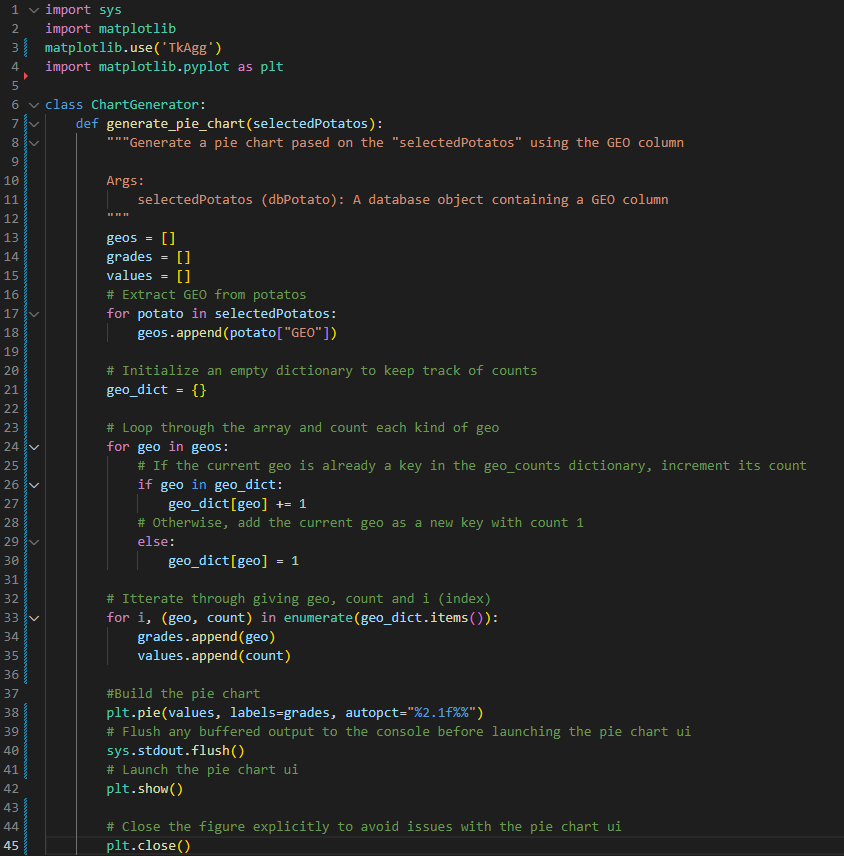
**CST8333 Project By Sebastien Ramsay**

[Rams0130@algonquinlive.com](mailto:Rams0130@algonquinlive.com)

**Evidence of Learning**

**PieChart.py**



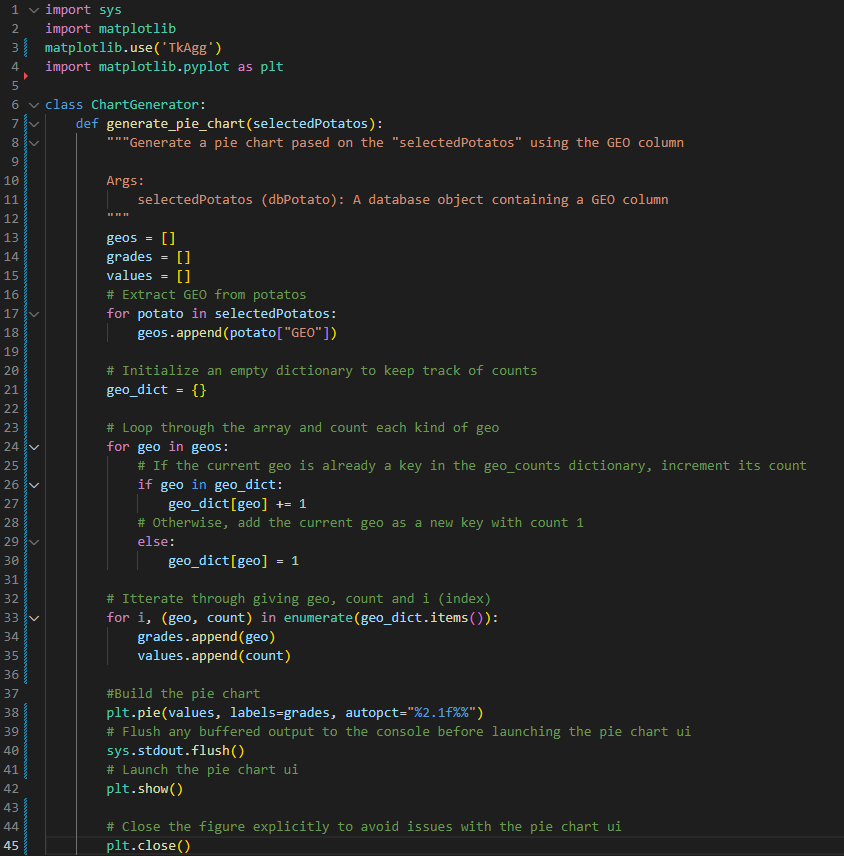
This is the **PieChart.py** file that I made to allow the program to make pie charts based on user selection. The **user makes a selection of “potatos”** to be run against the **generate\_pie\_chart method**

The generate\_pie\_chart method extracts the “geo” from each potato selected. Then it makes a dictionary keeping track of how many occurrences of each geo is in the geos array.

Using that information it then populates the values needed to create a pie chart. Finally the pie chart is displayed in it’s own UI. The UI needs to be closed to ensure no errors occur when the user closes the UI.

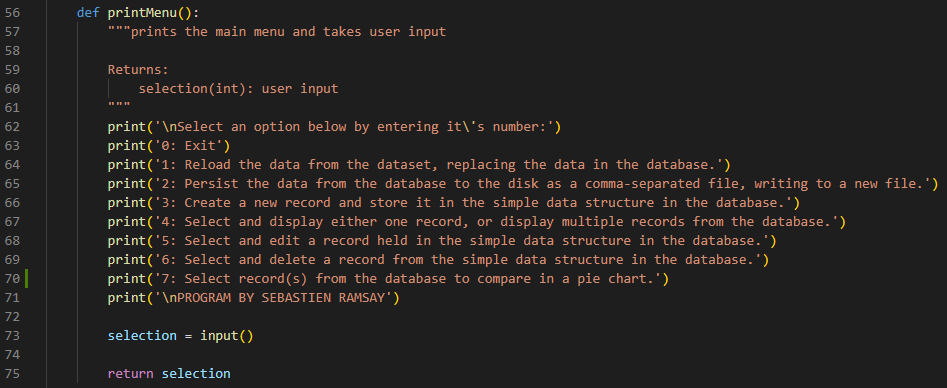
**Program Changes**

**PieChart.py**



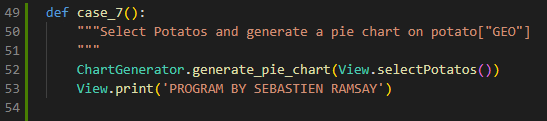
As described in Evidence Of Learning

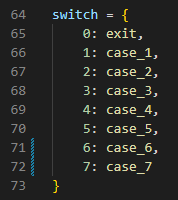
**PotatoView.py**



Line 70 added to show the user a new function is available.

**PracticalProjectPart3.py**





New case added. Potatos are selected using the View and then passed to the ChartGenerator to be shown in the pie chart UI.

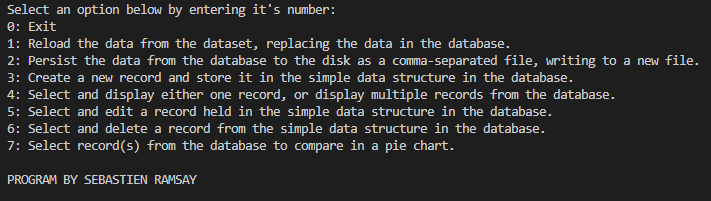
**PracticalProjectPart3.py**



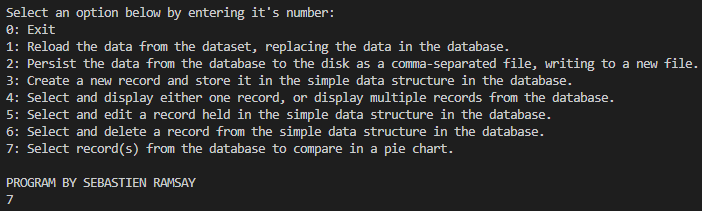
Added a catch for closing the program and sys.exit() to properly close the program.

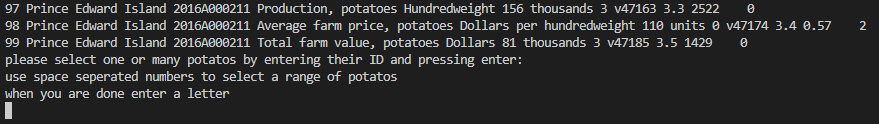
**Program Demonstration Via Screen Shots**

**Launch the Program:**

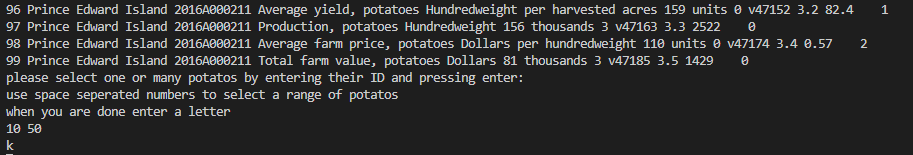


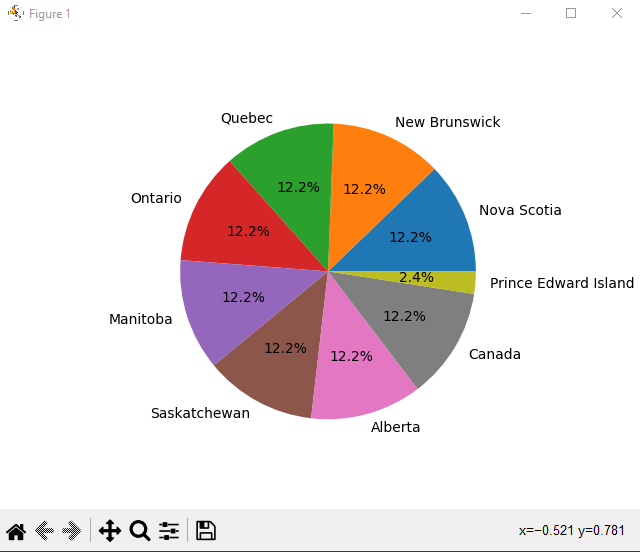
**Program is waiting for input, Input 7**



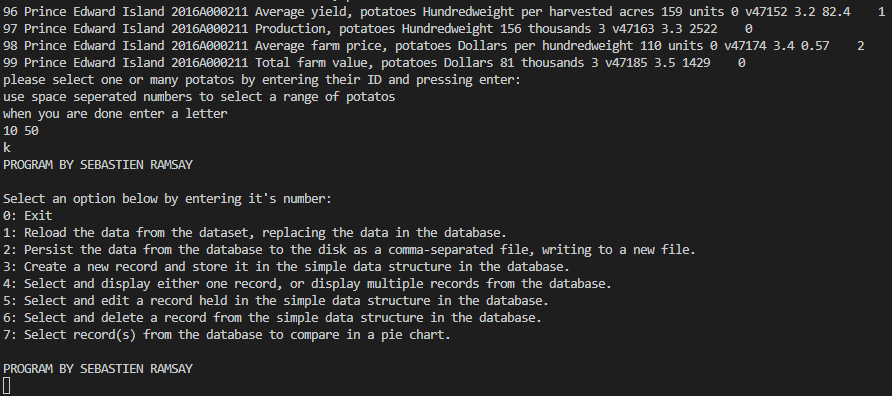


**Program requesting to select potatos, select 10 through 50**





**Pie chart UI has opened with the corresponding data**



**Pie chart UI has closed with no errors**

**Source Code Commenting Example:**

**import sys**

**import matplotlib**

**matplotlib.use('TkAgg')**

**import matplotlib.pyplot as plt**

**class ChartGenerator:**

**def generate\_pie\_chart(selectedPotatos):**

**"""Generate a pie chart pased on the "selectedPotatos" using the GEO column**

**Args:**

**selectedPotatos (dbPotato): A database object containing a GEO column**

**"""**

**geos = []**

**grades = []**

**values = []**

**# Extract GEO from potatos**

**for potato in selectedPotatos:**

**geos.append(potato["GEO"])**

**# Initialize an empty dictionary to keep track of counts**

**geo\_dict = {}**

**# Loop through the array and count each kind of geo**

**for geo in geos:**

**# If the current geo is already a key in the geo\_counts dictionary, increment its count**

**if geo in geo\_dict:**

**geo\_dict[geo] += 1**

**# Otherwise, add the current geo as a new key with count 1**

**else:**

**geo\_dict[geo] = 1**

**# Itterate through giving geo, count and i (index)**

**for i, (geo, count) in enumerate(geo\_dict.items()):**

**grades.append(geo)**

**values.append(count)**

**#Build the pie chart**

**plt.pie(values, labels=grades, autopct="%2.1f%%")**

**# Flush any buffered output to the console before launching the pie chart ui**

**sys.stdout.flush()**

**# Launch the pie chart ui**

**plt.show()**

**# Close the figure explicitly to avoid issues with the pie chart ui**

**plt.close()**