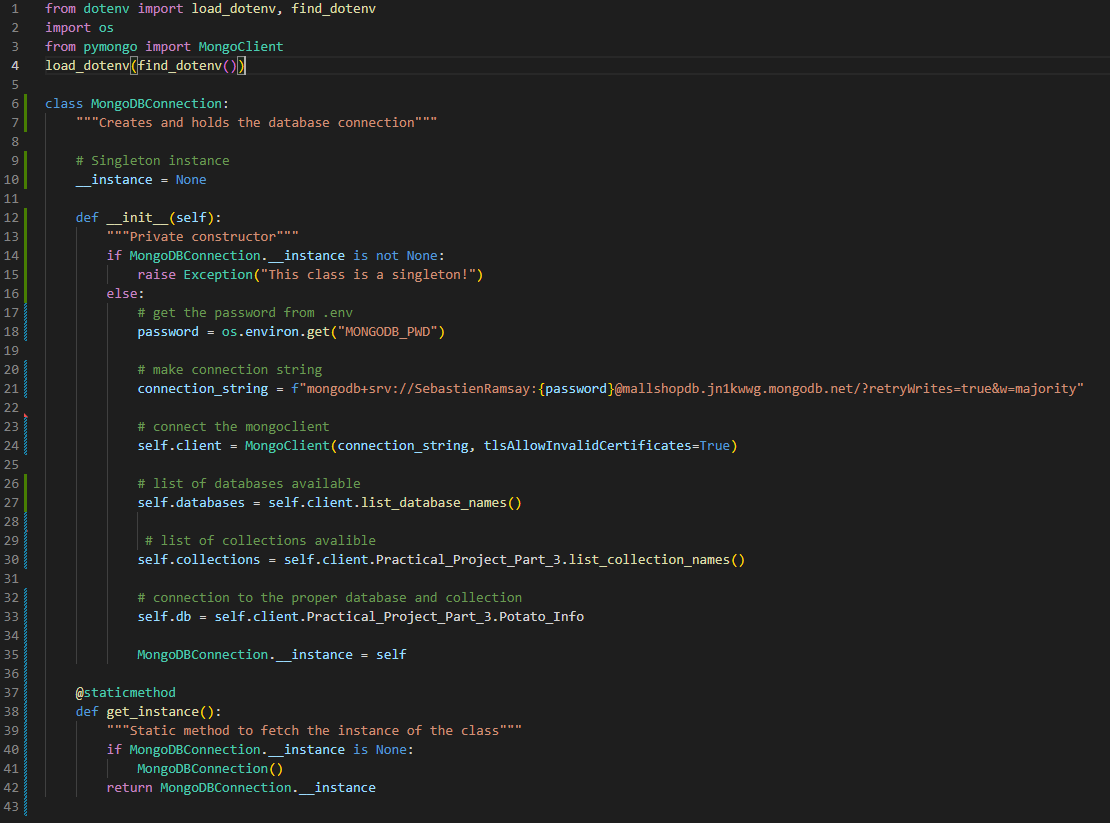
**CST8333 Project By Sebastien Ramsay**

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

**Evidence of Learning**

**MongoDB Connection:**

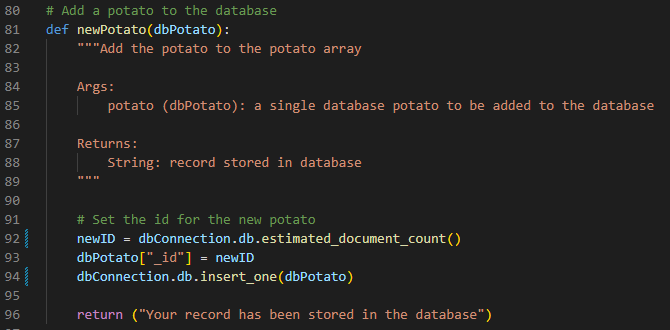


**The code above creates a connection to the MongoDB database and uses the client to connect to the correct database and collection.**

**The code also provides a variable that holds the available databases and another variable to hold the available collections.**

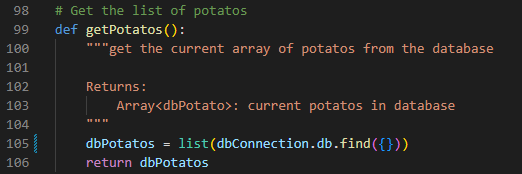
**Dotenv is used to store the password for the database connection.**

**Adding a new potato to the database:**



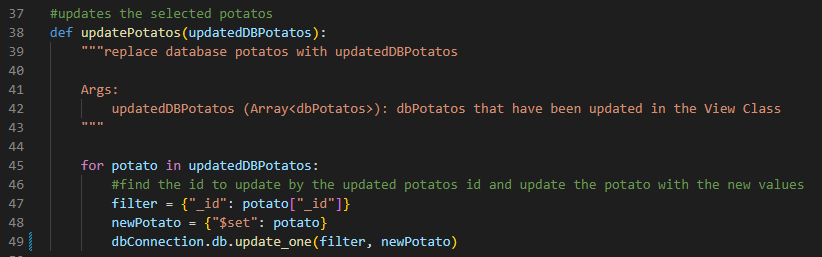
**newID is set to the amount of entries in the database so it will always have the next available id**

**Reading all potatos from the database:**



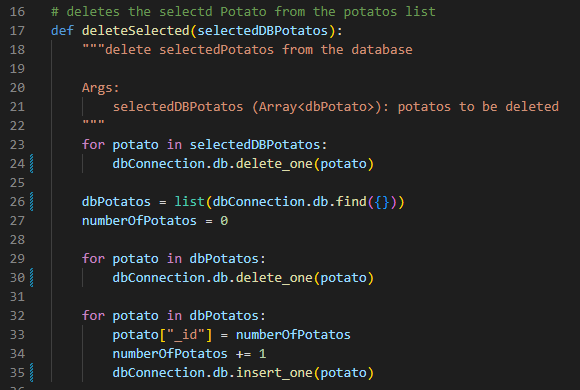
**Db.find({}) is empty to find all potatos in the database.**

**Updating items in the database:**



**The filter uses the potatoes ids to find the potatos to update and then the newPotato tells the db to set the values to the values in potato.**

**Deleting from the database:**

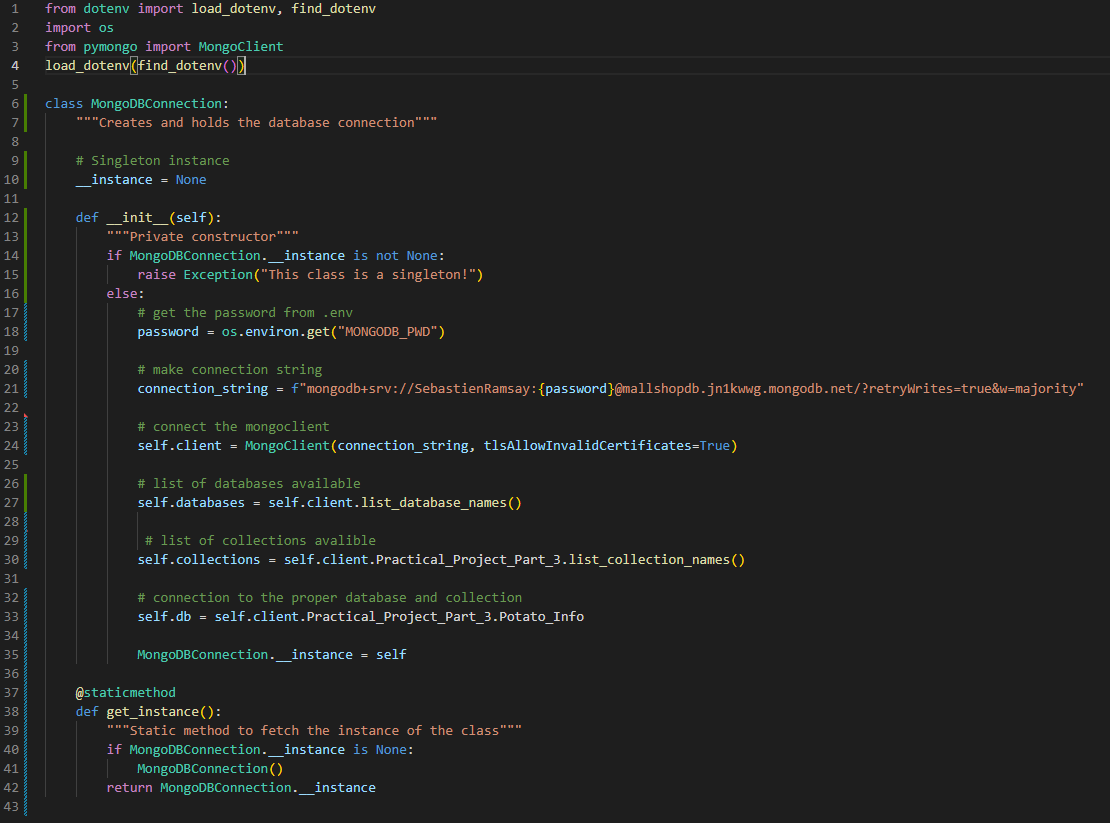


**The method takes in a list of potato’s selected by the user. the selected potato’s are deleted from the database. The dbPotatos is populated so the database can be cleared and the ids can be re-incremented.**

**Program Changes**

**it new class called MongoDBConnection in**

**\Project Files\MongoDBConnection**



**Crud operations added to controller script**

**PotatoModel is out of date and no longer used**

**Updated unit test to work with the database**

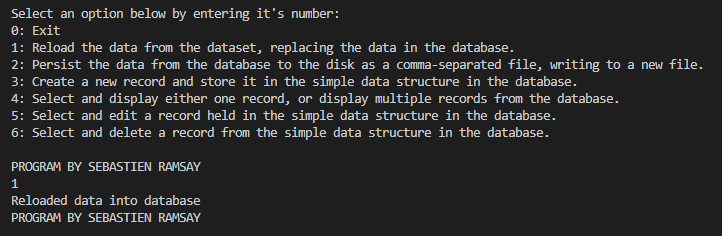
**Updated PracticalProjectPart#.py: better error handling**

**Updated PotatoView to work with dbPotatos and updated the selection system so items can be selected with ease**

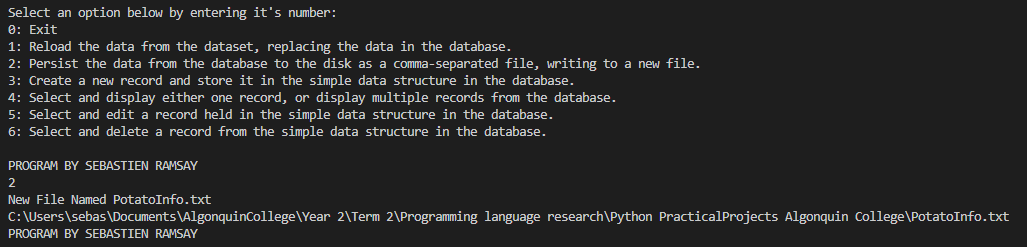
**Added a “Project Files” folder to Practical Project Part 2 and 3, uploaded to git repo in separate branches.**

**Program Demonstration Via Screen Shots**

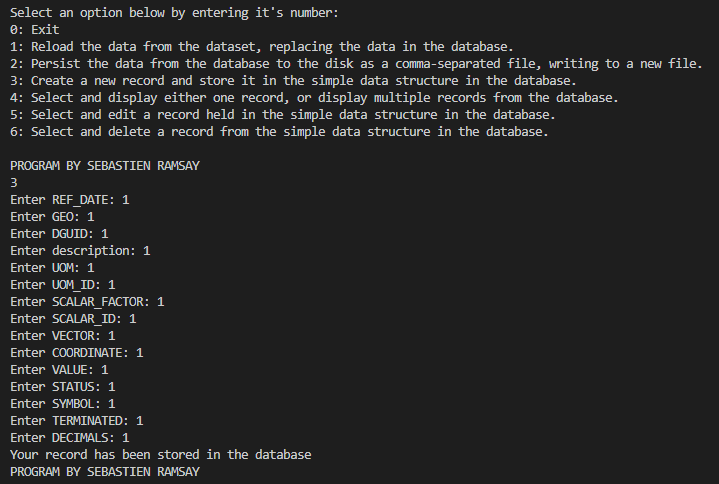
**Reload data into Database:**



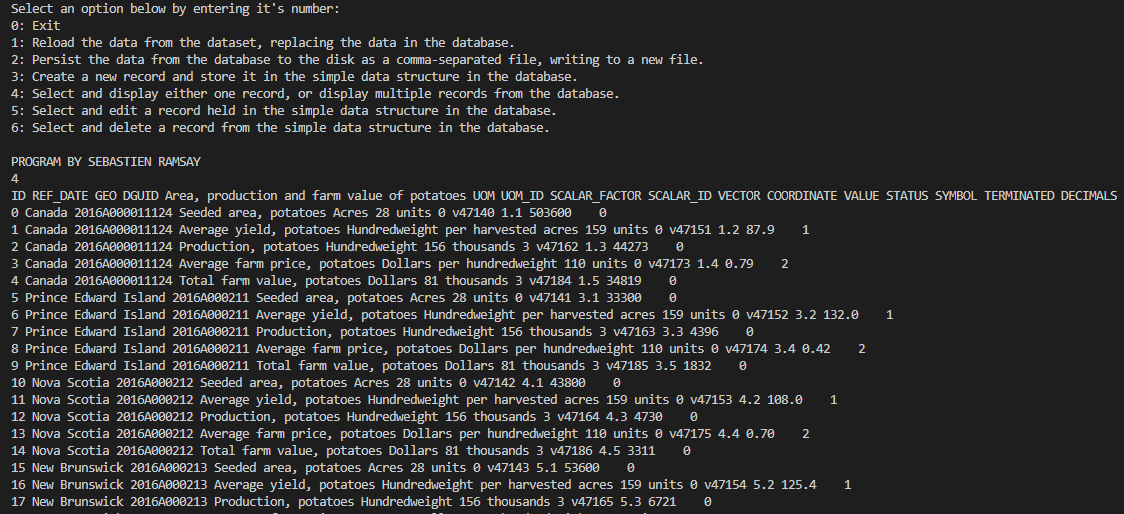
**Write to txt file:**



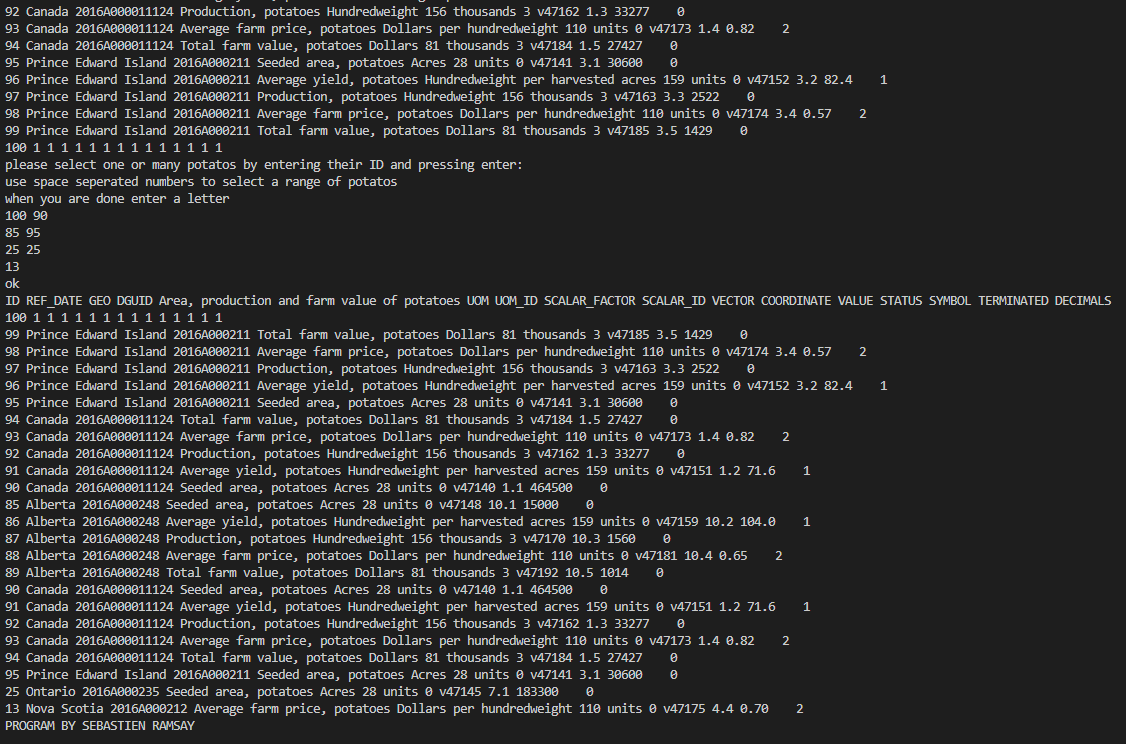
**Create new and add to database:**



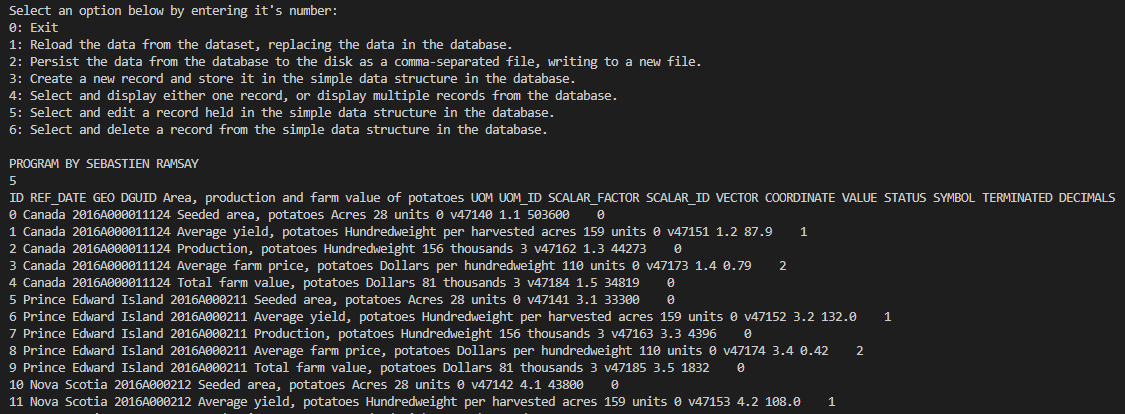
**Select and display in console:**



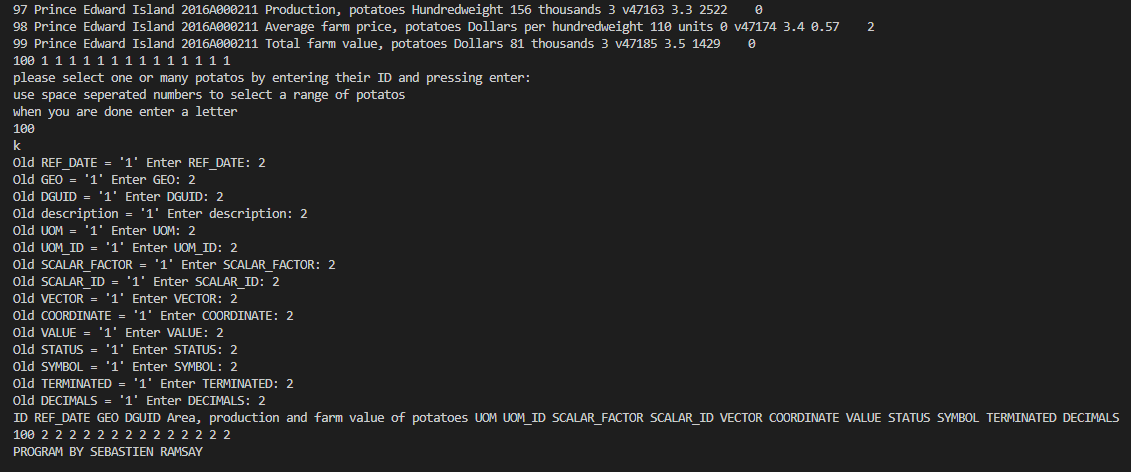
**…**



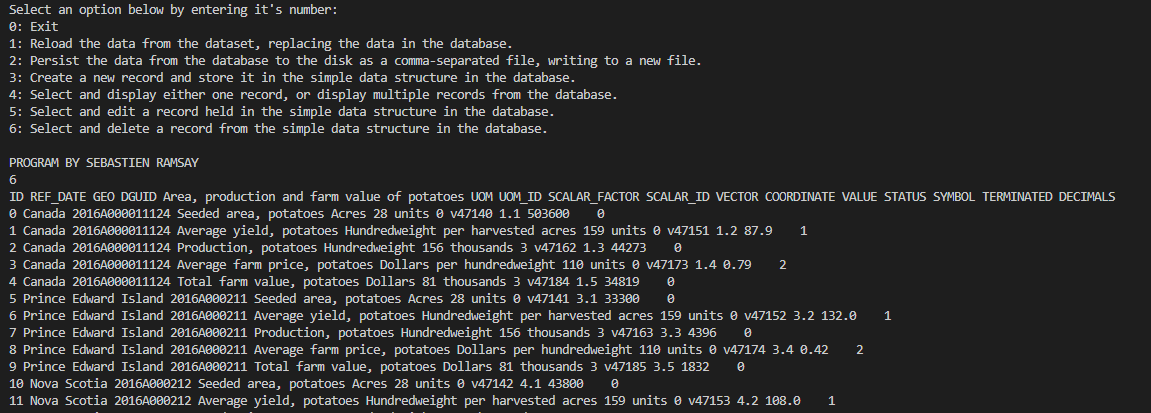
**Select and edit the database:**



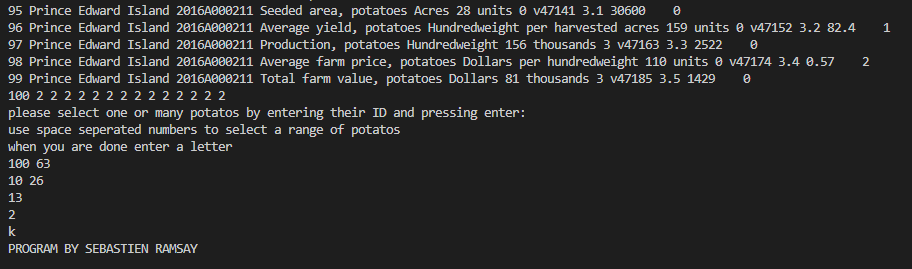
**…**



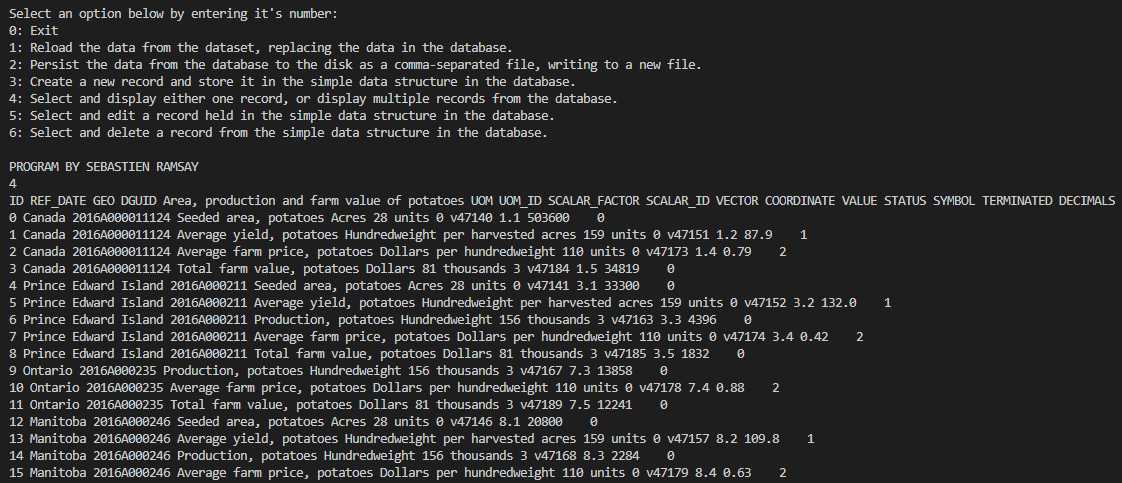
**Select and delete from database:**



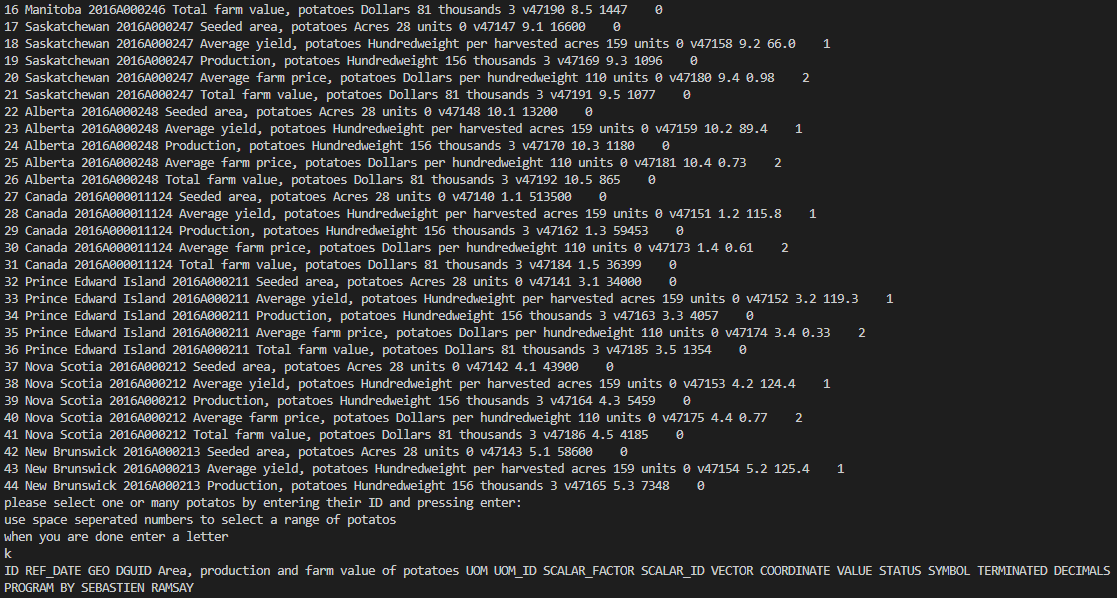
**…**



**…**

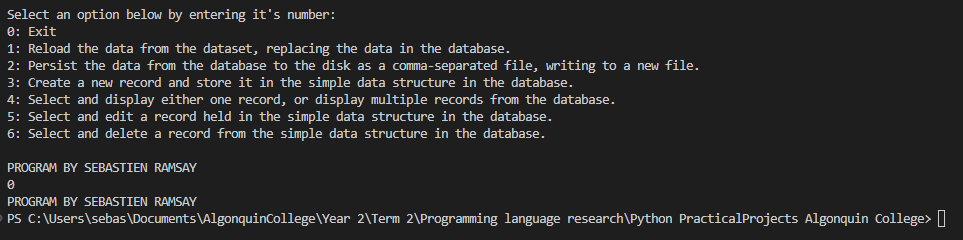


**…**

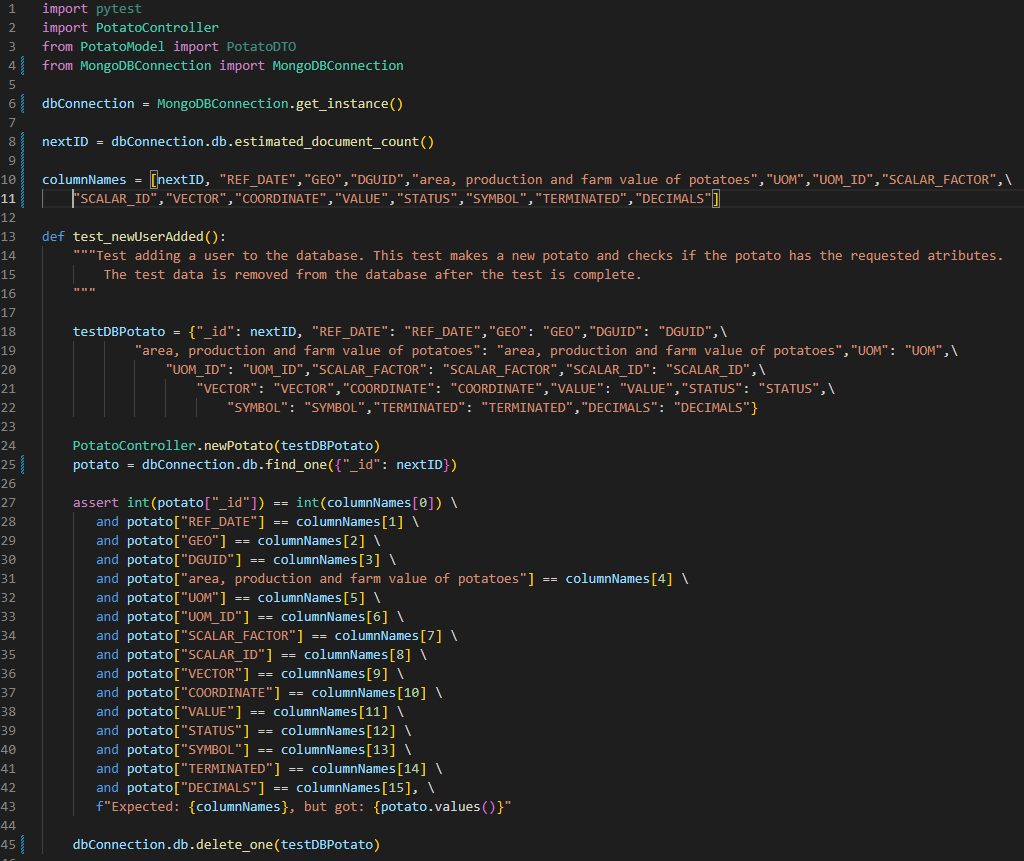


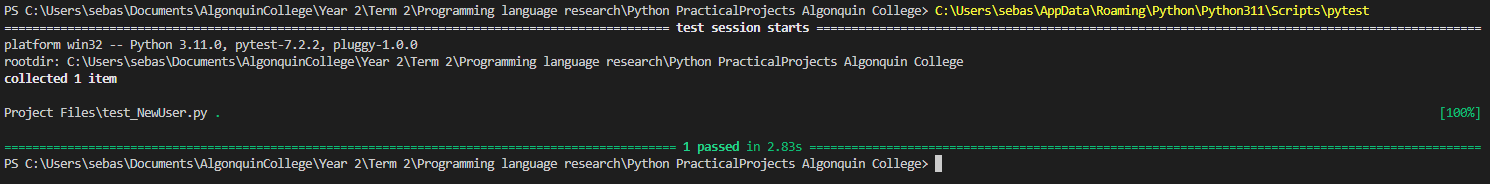
**The program re-increments the id’s after deleting to avoid errors and keep the id’s clean**

**Exit the program:**



**Unit Testing:**





**This is an example of my pytest running and passing. It makes a new dbPotato and inserts it into the db and checks if the object in the db holds what it is asking it to hold.**

**Source Code Commenting Example:**

**from dotenv import load\_dotenv, find\_dotenv**

**import os**

**from pymongo import MongoClient**

**load\_dotenv(find\_dotenv())**

**class MongoDBConnection:**

**"""Creates and holds the database connection"""**

**# Singleton instance**

**\_\_instance = None**

**def \_\_init\_\_(self):**

**"""Private constructor"""**

**if MongoDBConnection.\_\_instance is not None:**

**raise Exception("This class is a singleton!")**

**else:**

**# get the password from .env**

**password = os.environ.get("MONGODB\_PWD")**

**# make connection string**

**connection\_string = f"mongodb+srv://SebastienRamsay:{password}@mallshopdb.jn1kwwg.mongodb.net/?retryWrites=true&w=majority"**

**# connect the mongoclient**

**self.client = MongoClient(connection\_string, tlsAllowInvalidCertificates=True)**

**# list of databases available**

**self.databases = self.client.list\_database\_names()**

**# list of collections avalible**

**self.collections = self.client.Practical\_Project\_Part\_3.list\_collection\_names()**

**# connection to the proper database and collection**

**self.db = self.client.Practical\_Project\_Part\_3.Potato\_Info**

**MongoDBConnection.\_\_instance = self**

**@staticmethod**

**def get\_instance():**

**"""Static method to fetch the instance of the class"""**

**if MongoDBConnection.\_\_instance is None:**

**MongoDBConnection()**

**return MongoDBConnection.\_\_instance**

**Work Cited (IEEE)**

“Tutorial¶,” *Tutorial - PyMongo 4.3.3 documentation*, 2008. [Online]. Available: https://pymongo.readthedocs.io/en/stable/tutorial.html. [Accessed: 17-Mar-2023].