Introduction

Background

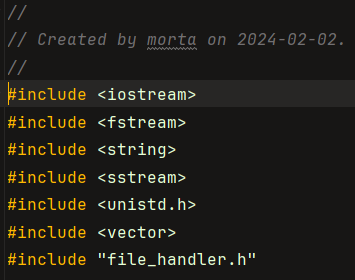
The system's sophisticated data structures and clever packaging techniques are designed to improve a user's overall online buying experience. It is developed to efficiently manage large inventories to maximize the performance of backend systems. Therefore, the program's primary goal is to provide students with the skills necessary to overcome real-world obstacles in data processing and algorithmic application development, particularly for e-commerce platforms.

Objective

Snapshots

Data Load

List of libraries used by the file\_handler class:



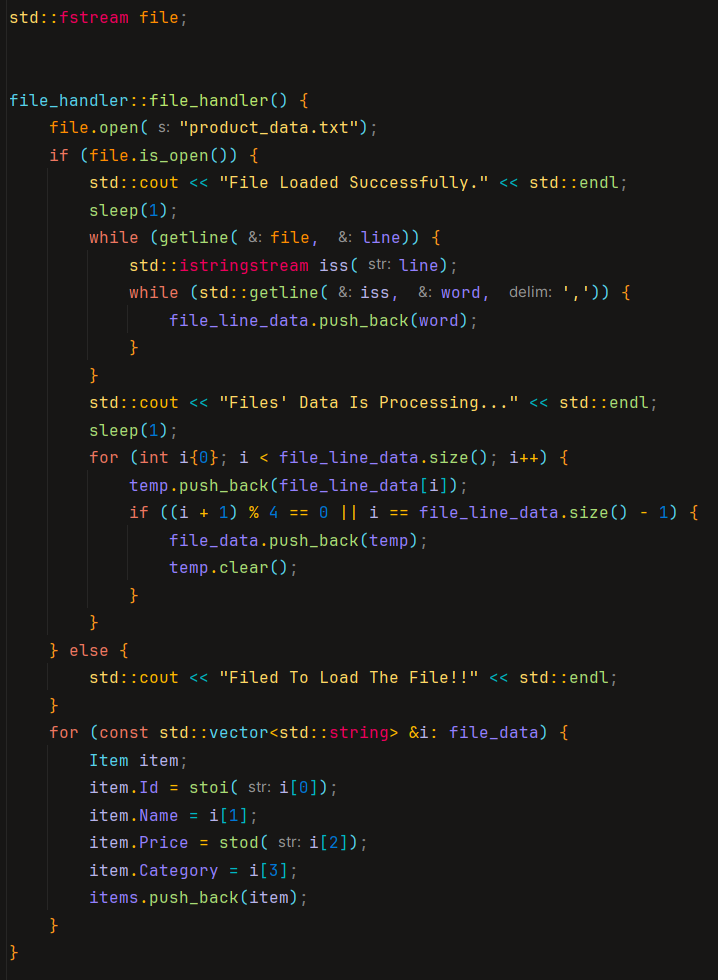
These are all the libraries used in the file\_handler class.

file\_handler.h:



This file contains all our declarations for the file\_handler class. Here we can see a declaration of constructor for the class file\_handler() and an object constructor for Item within the class, and all other functions and variables necessary for the classes function.

file\_handler.cpp:



Here file\_handler::file\_handler() defines the behavior of our constructor for the class. The file first makes an instance of fstream class called file for file handling purposes. In the constructor we use file.open() function to open the file containing our product list. After opening the file an if statement checks to see if the file was open successfully using the value returned by file.is\_open() function all part of the fstream class included at the top. After loading the file, the while loop will iterate trough each line and insert it word by word in the file\_line\_data vector. After the while is done with the file another for loop would go through the vector and make smaller vectors pf every five element in the file\_line\_data vector and insert the new vectors in the file\_data 2D vector using the temp vector that we clear using the clear() function to ensure memory use efficacy. This method puts all product attributes together in a single vector and all products as a whole in a bigger vector of vectors. After isolating the data, we now can use the last for loop to make an object using all the elements we extracted stored in vectors inside the file\_data vector.

Insert

A screen shot of a computer screen

Description automatically generated

A screen shot of a computer program

Description automatically generated

In here new\_item() method is our insert method. This method constructs a new Item object using the data passed on to it from the user. In our main.cpp we can see that function is\_id\_there() is called before the information is passed to the method, this method checks and makes sure there are no duplicate ids. After making sure id is not there and constructing the object then the new\_item() method will insert the object in the items object vector for storage.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generatedA screen shot of a computer

Description automatically generated

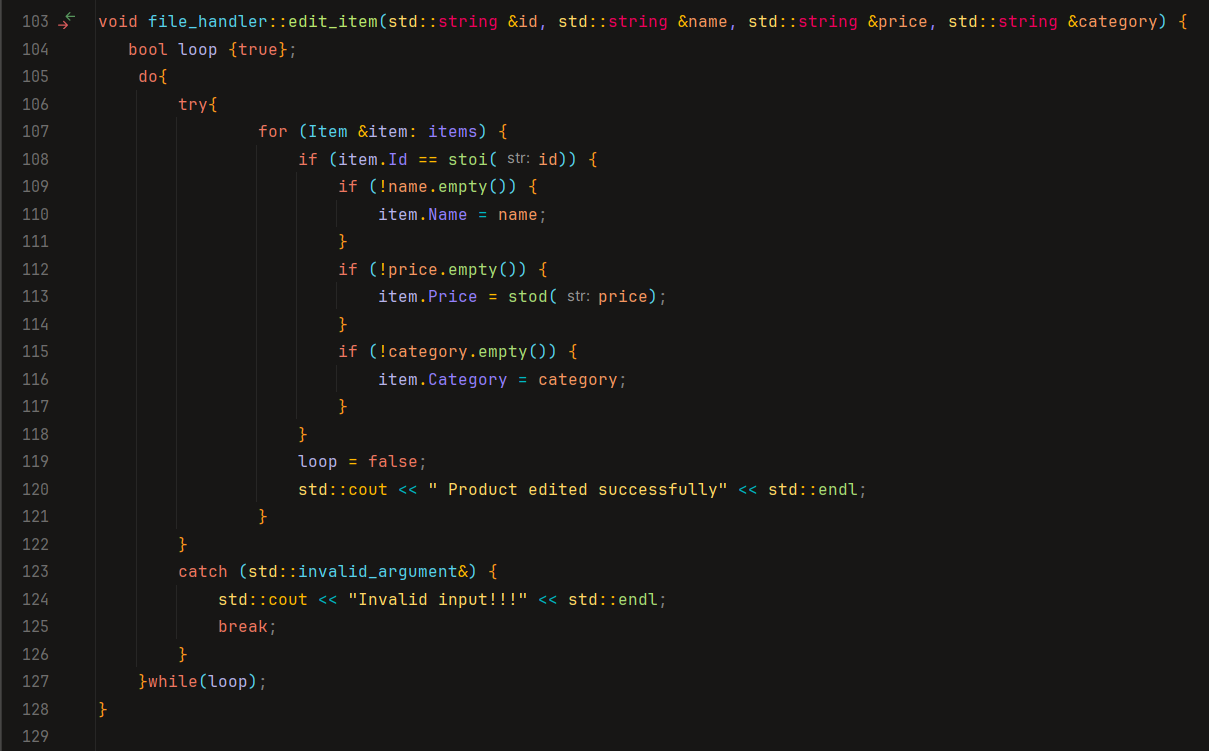
A screenshot of a computer

Description automatically generated

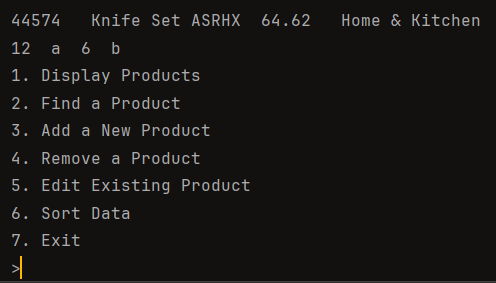
A screenshot of a computer

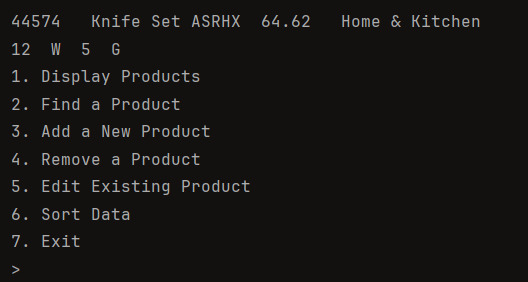
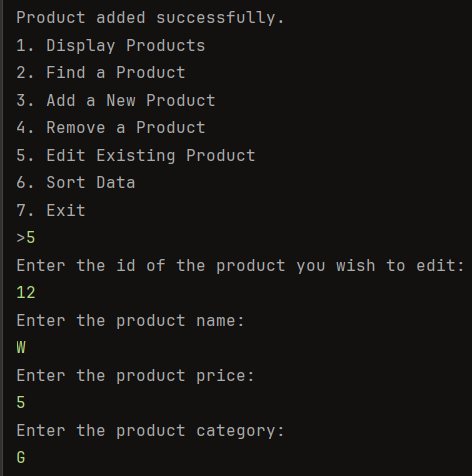
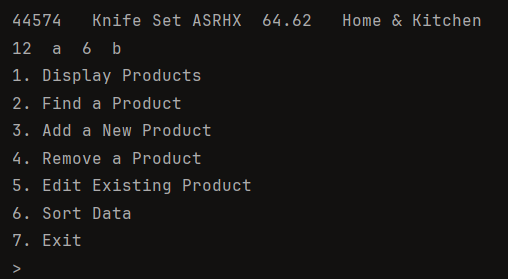
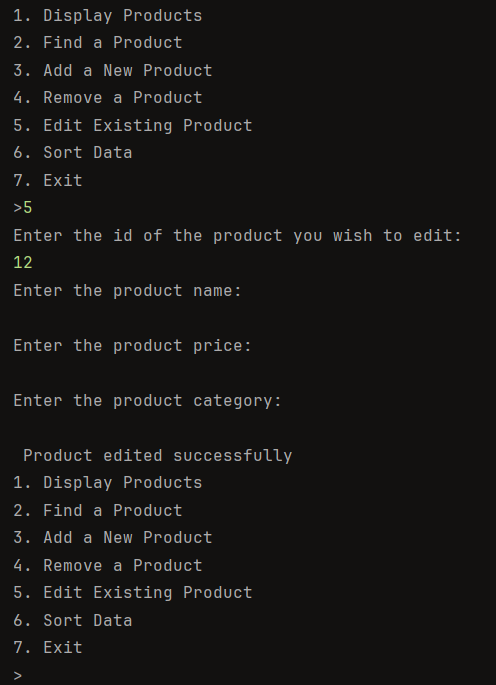
Description automatically generated

Update



Here we have the edit\_item() method. This method iterates trough the items vector and checks if the id inputted by the user matches the id of the item in the vector. Then it checks to see if any changes have been made. At the end it writes the new changes if there are any and keeps the original value if there are none.





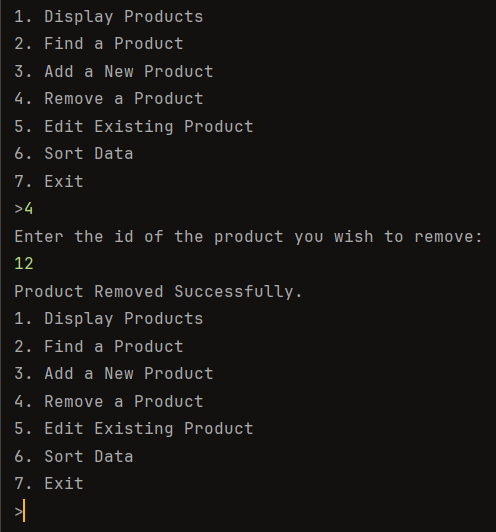
Delete

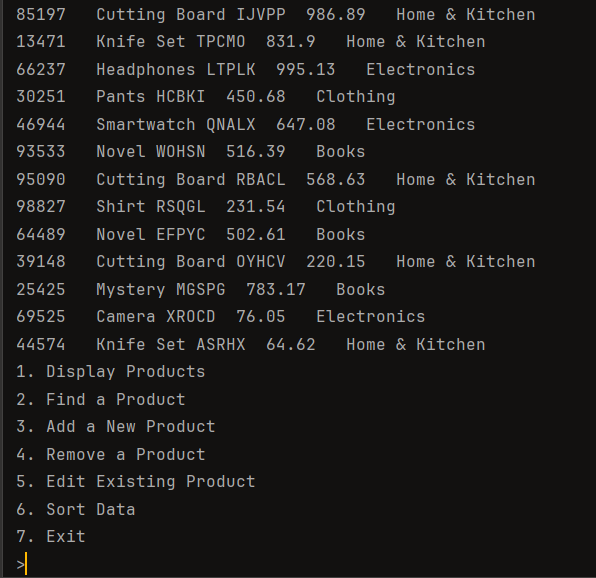
A screen shot of a computer code

Description automatically generated

This method goes through all the elements in the items vector keeping note of their index. Then it checks to see if the id requested to be removed matches the id of any of the items and if yes removes it from the items vector.



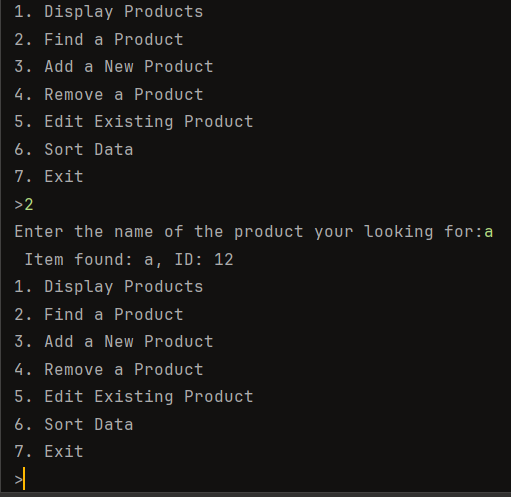


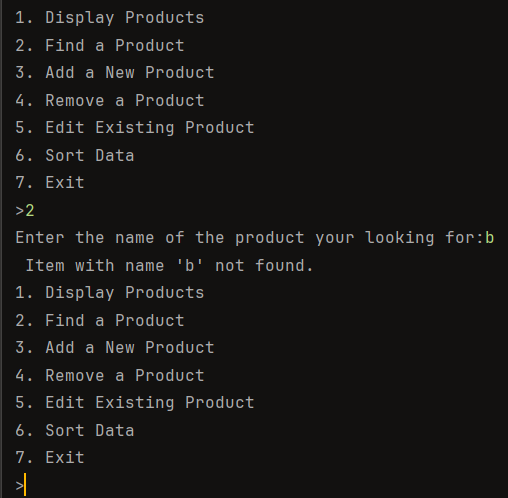


Search

A screen shot of a computer code

Description automatically generated

Checks the name of all the items in the items vector and see if the name matches the name we are searching for and if yes it returns the id for that item. 



Sort

