Introduction

The module will facilitate user end by bringing in the element of feasibility of shopping 24/7 with a bulk variety, easy navigation through the available stock, and at most a user-friendly interface. The portal operates two ends, which includes the admin (caters the inventory), and the customers end. The admin will support the user end, by restocking the inventory and managing the customer’s feedback. Hence, improvising customer satisfaction, as well as the overall functionality. We are keen to work on customer’s engagement, by collecting customer’s feedback, which will be later assessed by the team. This adds on, will generate greater traffic on our portal and will increase user satisfaction. Moreover, we aim to provide interactive user interface that helps in easy navigation, which is a major limitation in existing systems. We have also certain discount offers, to anticipate the user.

Main Features

* Admin Panel

1. Admin will repeatedly visit the inventory, update the stock, and assure user feasibility by collecting the feedback from the user end.
2. The saved feedback form can be controlled by the admin.
3. Monitor Customer’s details, maintaining the stock (add new record, delete stock, view stock), are the main features of the admin panel.
4. Secured login system.

* Customer Panel

1. Customers will advance by selecting the type, the size and the quantity of the attire.

2. The system will repeatedly ask the user if he wants to add any additional item in his cart.

3. The final billing receipt will display the user’s name, contact number and the

residential address.

4. Discounts are also offered on certain levels of purchasing.

5. Assure user feasibility by collecting the feedback from the user end. The

feedback form will gather customer response and navigate our areas of

improvement.

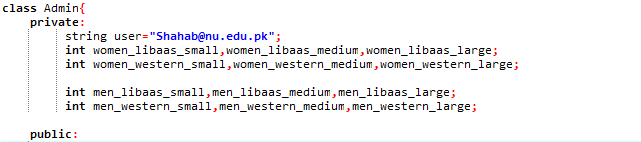
Concept of Abstraction

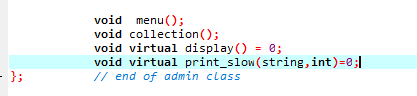
The concept of abstraction has been implemented in the base class (store), which has derived classes (Women Category and Men Category). The pure virtual function -showCollection () has no definition in the base class, however the child classes (Women Category and Men Category) has its implementation and definition-to display the respective eastern (Libaas Collection) or western collection (WesternBound Collection), as per the user choice. Therein, the pure virtual functions are introduced, and the concrete classes (Women Category and Men Category), provides the implementation to abstract class.



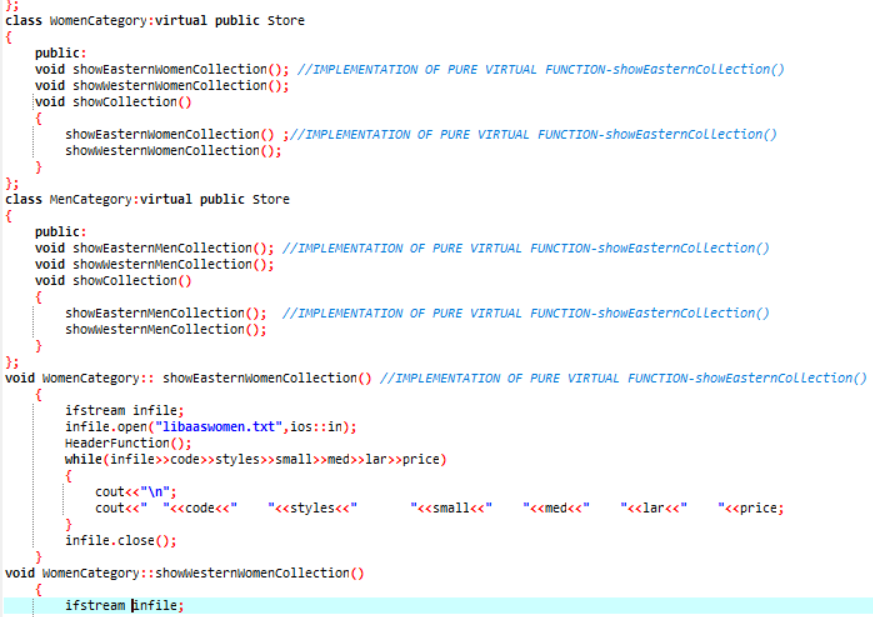
This Concept is also implemented in the Admin second Base class (ADMIN), which has a derived class – STOCK, having the feature to monitor record of the stocks. Function “print\_slow() and display() are declared as purely virtual function which are defined in the derived class STOCK.

**Admin Panel:**





**Customer Panel:**



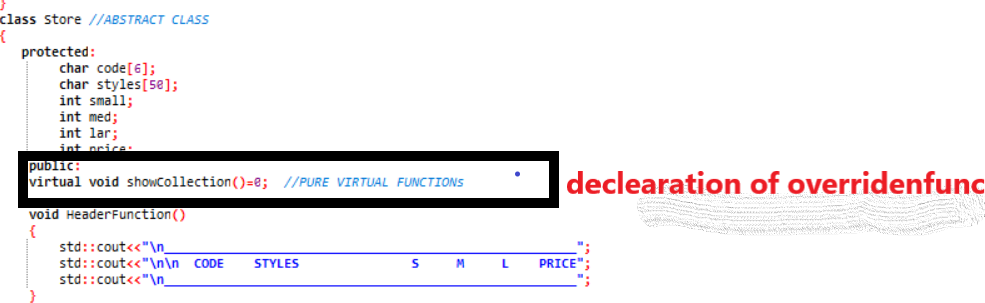
Concept of Polymorphism

* Operator Overloading

The stream extraction operator is overloaded to perform output of user defined type-object. It is overloaded with two parameters, cout and object of class Receipt. The operator is overloaded as a global function, and to access the private data members of the class Bill, such as total, billno, styles, price and required quantity; it is made friend function as well.



* Function Overriding: The base class-(Store) has showCollection () which is overridden in its derived classes Women Category and Men Category. These two classes have different implementation and definition of the base class (Store).





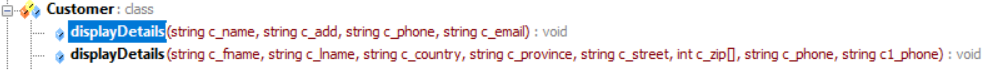
* Function Overloading

The function overloading concept is implemented through the function- displayDetails() in the class- Customer.

The first function takes in 4 parameters, which includes c\_name, c\_add ,c\_phone ,c\_email. The function will aim at displaying the billing details to the user- in the final receipt and in the BillingDetails file (BillingDetails.txt)-to store the details for future referencing for promotions and creating strong customer base.

The second function takes in 8 parameters, which includes c\_fname, c\_lname, c\_country,

c\_province, c\_street, c\_zip [], c\_phone, c1\_phone. This function is primarily meant to contain comprehensive details to ship the product and store the details in the ShippingDetails file (ShippingDetails.txt). This is more extensive than the billing details since these are details will be looked up to deliver the order.

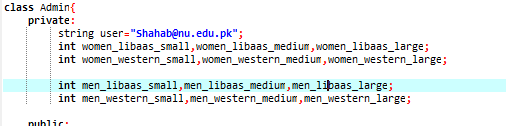




Concept of Encapsulation

Since encapsulation supports bundling of data within the capsule-to prevent it from external interference, we have implemented data hiding through user defined type-classes and access modifiers- private, public and protected.

The data members of Base class Admin are set private, in order to limit its scope.



 The data members of class Receipt, such as total, billno, req, code, styles and price are declared private, to limit its access within the class. The data members of class Store, such as code, small, med, lar and price are declared protected so that, they are accessible within the class and the derived classes-Women Category and Men Category.

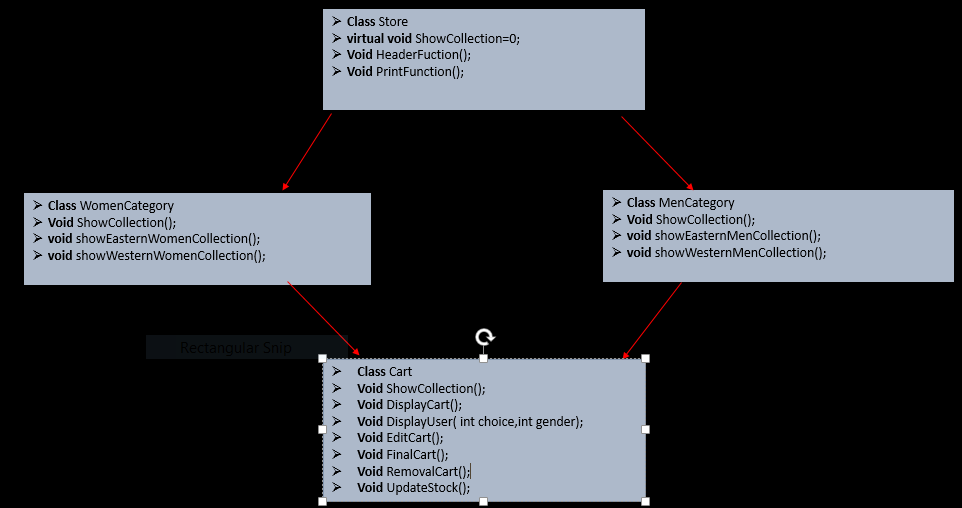
Concept of Inheritance

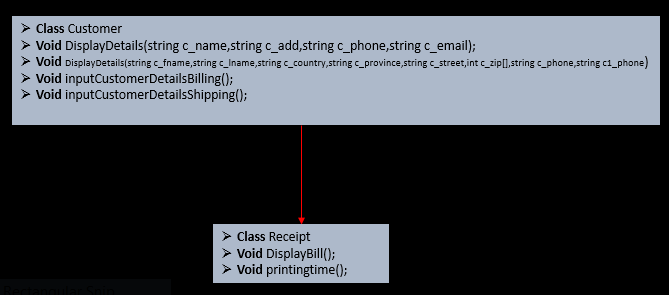
The project binds with diamond problem, or multiple inheritance. The ambiguity of function call is dealt with virtual inheritance. The data members of class Store, such as code, small, med, lar and price are declared protected so that, they are accessible within the class and the properties can be inherited in the derived classes-Women Category and Men Category. The concept of single inheritance is implemented when the class Receipt, inherits the functionalities of class Customer. The class Customer is recognized as Base class, whereas class Receipt is the derived class.

**Admin Panel:**



**Customer Panel:**



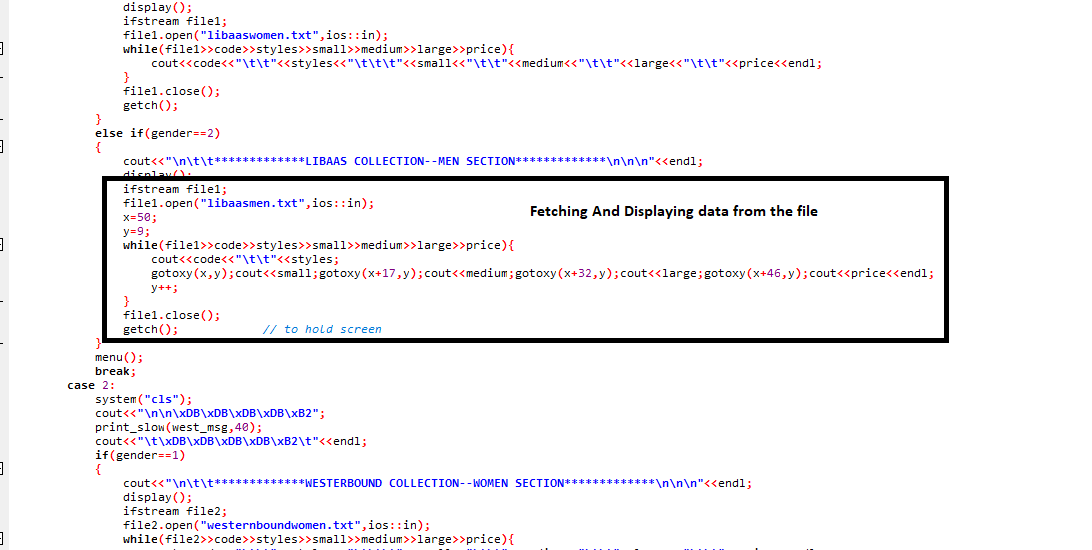


Concept of Templates

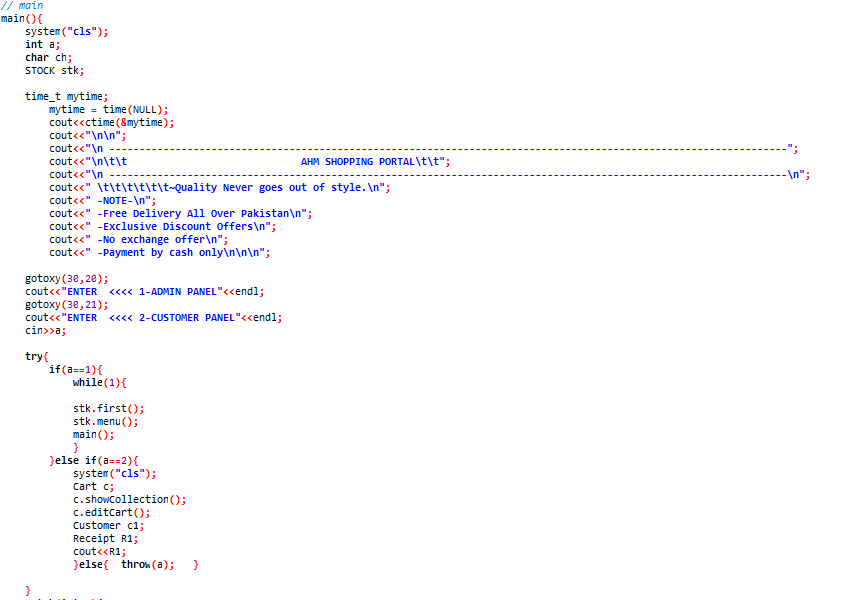
Concept of Exception and Filing

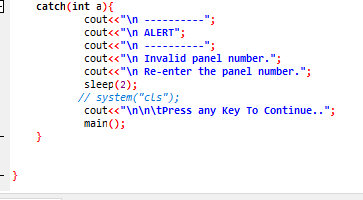
**Admin Panel:**

The main feature of Admin Panel is to monitor all the data. To view the stock stored in the files following logic is implemented which fetch the data from the file of required category and gender: 



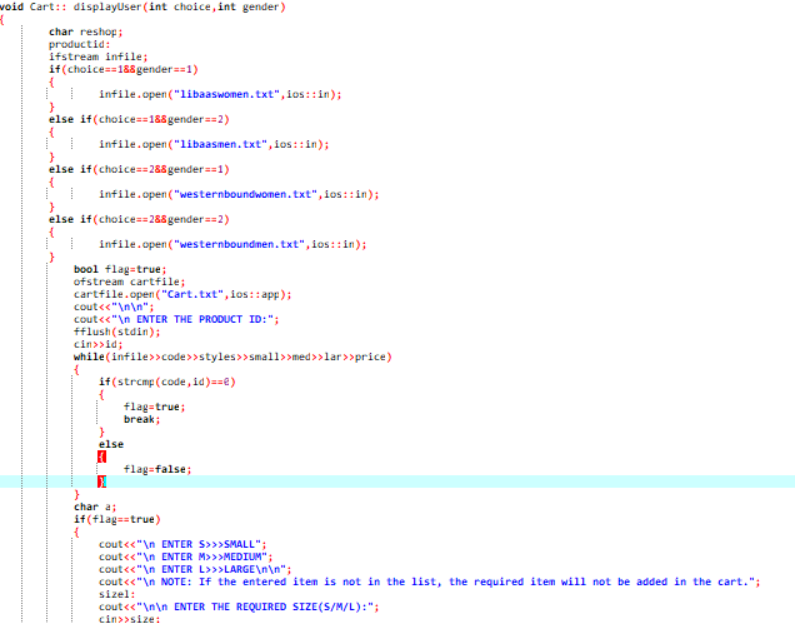
In order to control the abnormal behavior of the program, exceptional handling is implemented, for example in order to check whether the correct number is entered, first **try** the number and if its true program continues, else it **throws** the number and **catch** prompts the error message.





**Customer Panel:**

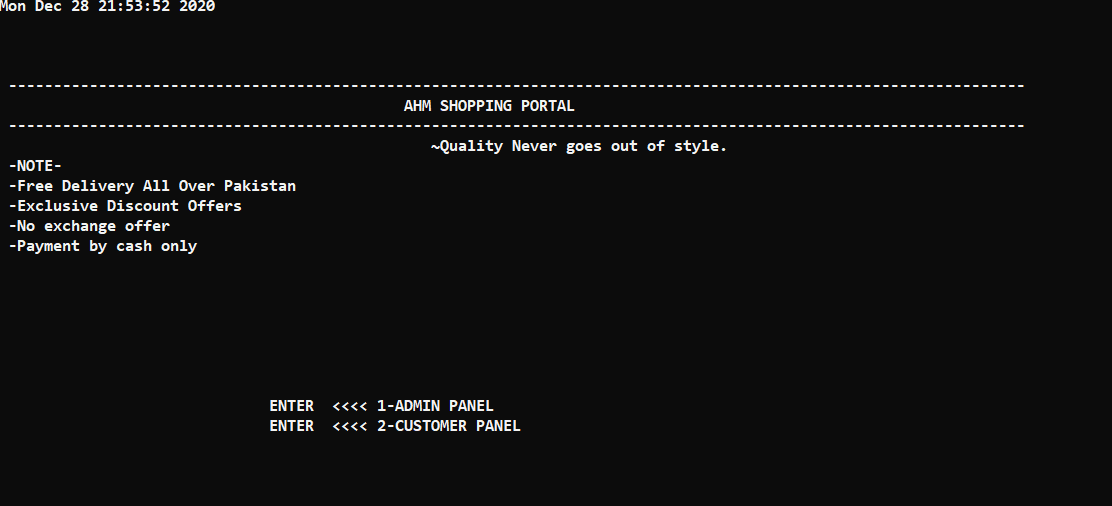
To store the stock, of the two categories-Women and Men and the apparent collection, WesternBound and Libaas, we have made use of filing. A small database is created within each file, to display to the customer, and for the admin to manage the inventory.

To handle the abnormal execution of program, try and catch blocks are implemented at various spaces in the code. The example of it is asking the user to input any choice between these two- 1 for Libaas collection, and 2 for WesternBound collection. If the user enters any character, outside the scope of if- block, the exception will be thrown and caught in the catch block. Firstly, displaying the alert message, and prompting the user to re-enter.

Applications and Usability

Clothing store management system is specially designed for the purpose of adding clothing item’s detail. The system elaborates the basic concept for storing and generating item’s detail. In this system, staff can sign up as a system admin, He/she can have full access to the system for maintaining daily records. This project is a simple console-built application without graphic. In this project, you can manage a typical ‘fashion wear’ department store. You can edit goods, delete and display the goods and see the customer's feedback. File handling has been used to record the information (rate, quantity, styles and code) of the added goods. Switch cases has been used to display the records in categories. User can visit the web based online portal and view the stock. He can shop by adding goods to the cart. He also has an option to edit his cart. Discounts are also offered on certain levels of purchasing. A proper bill receipt will be printed after purchasing of items containing user detail and the item details. User can also provide his feedback of shopping from our clothing store. At various places, exceptions are handled on erroneous input from the user, prompting the user to reenter his choice. This project is feasible both for admin and the customer and is easy to understand.

Outputs:



**Admin Panel:**

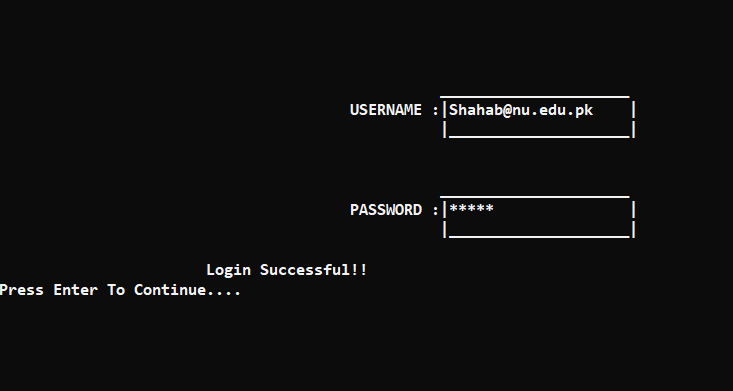


Figure 1

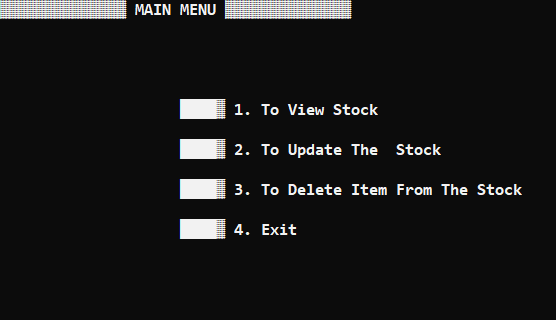
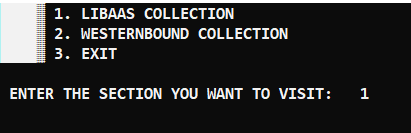


Figure 2



🡨 VIEW STOCK Section

Figure 3

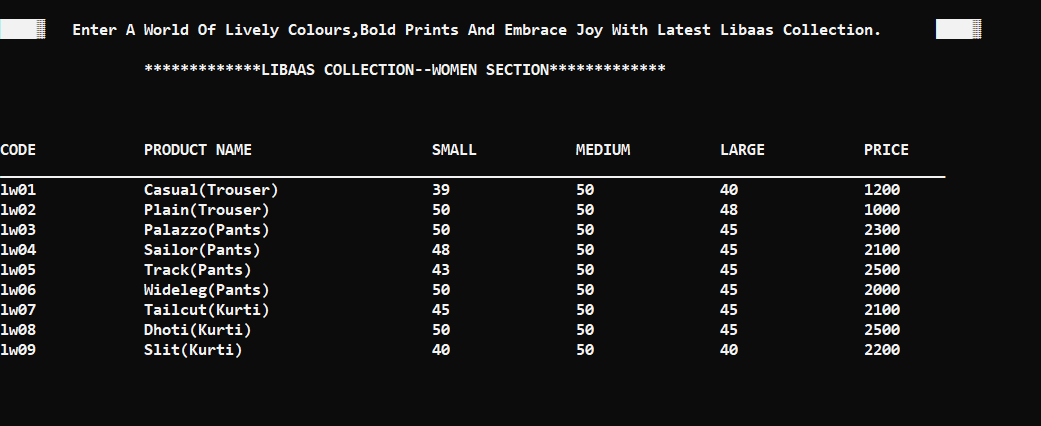


Figure 4

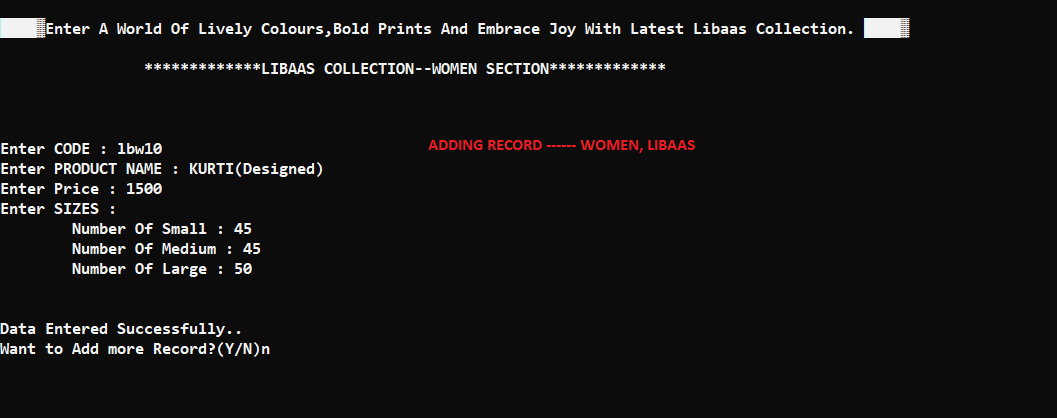
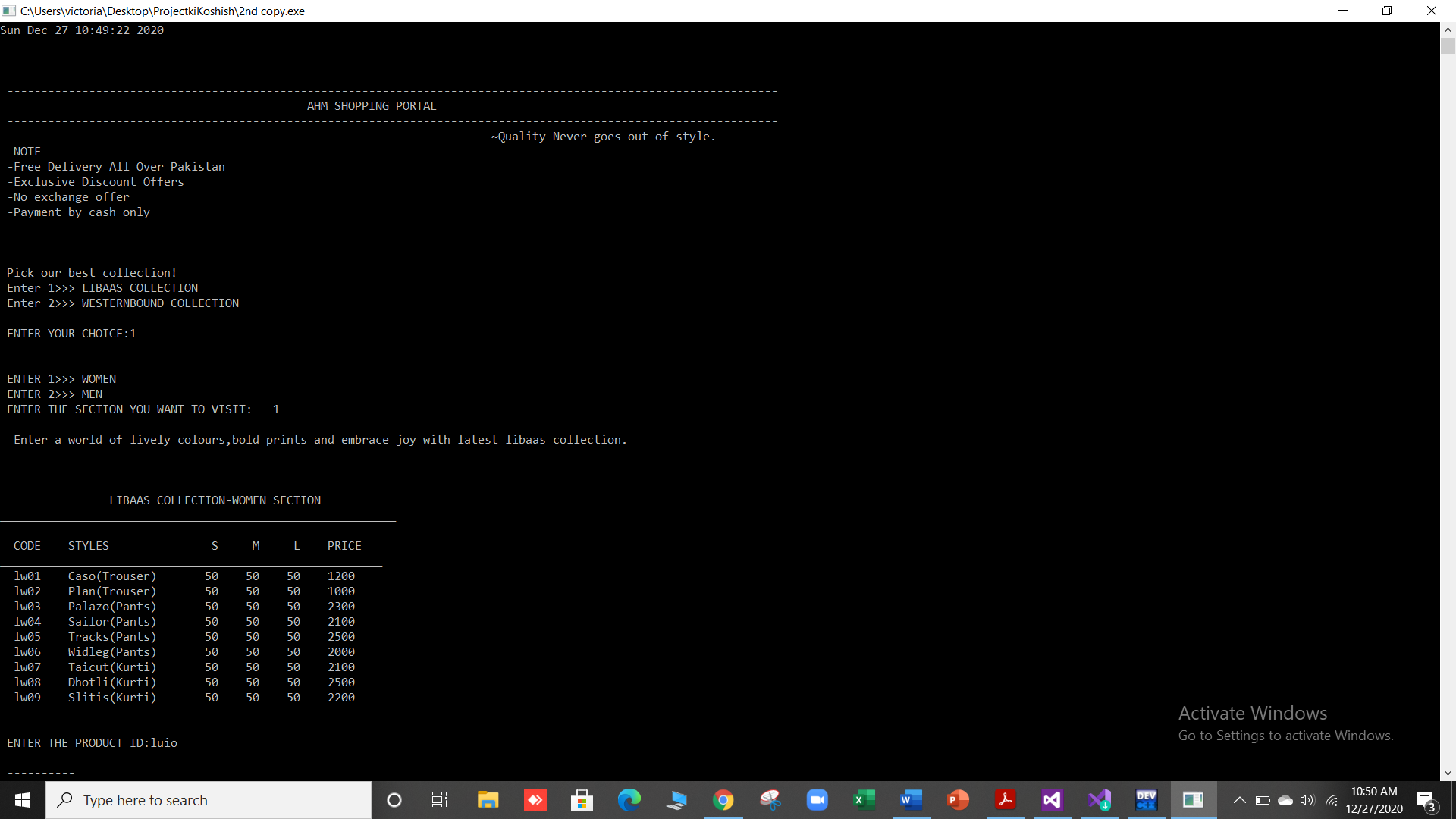


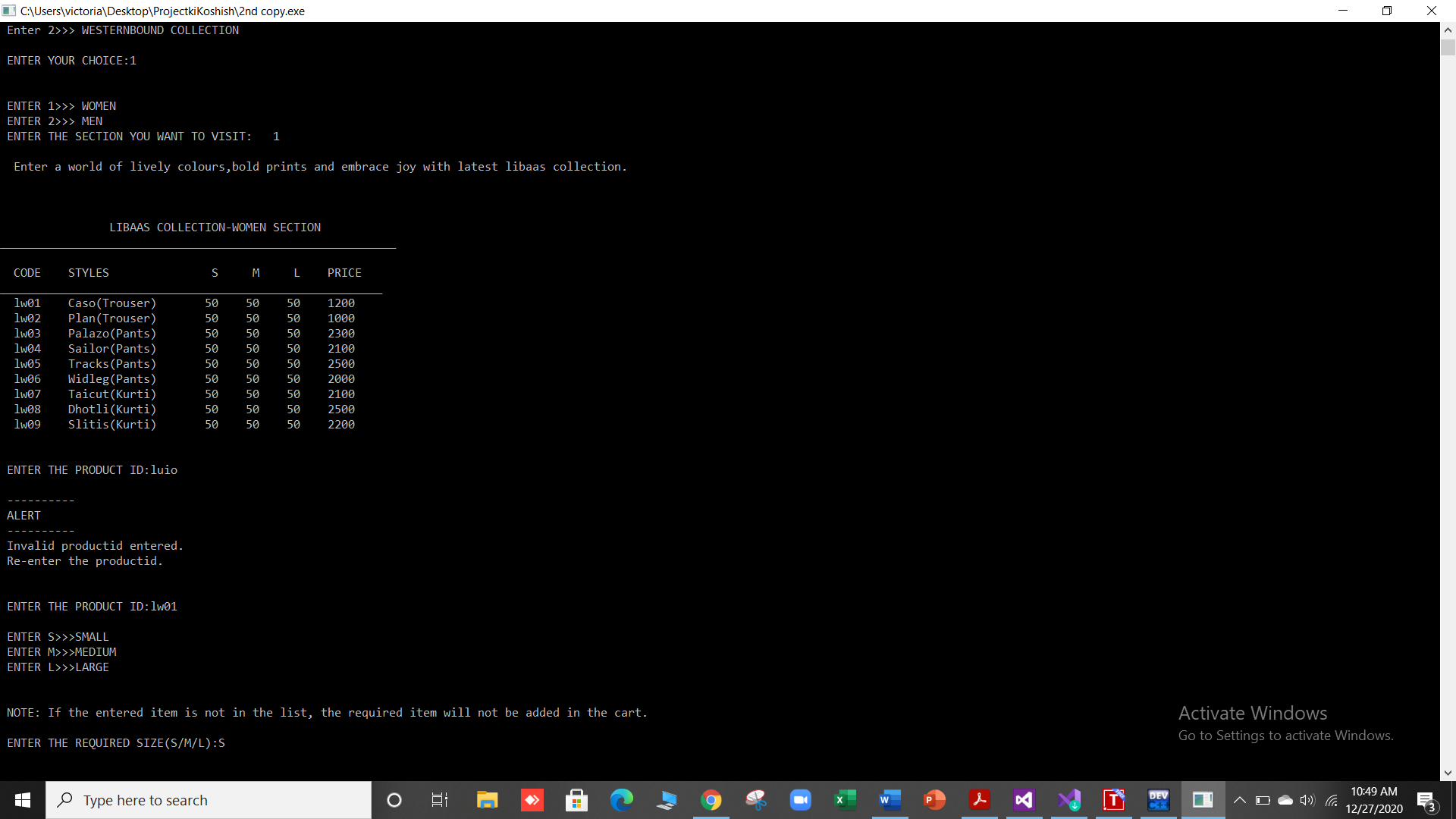
Figure 5

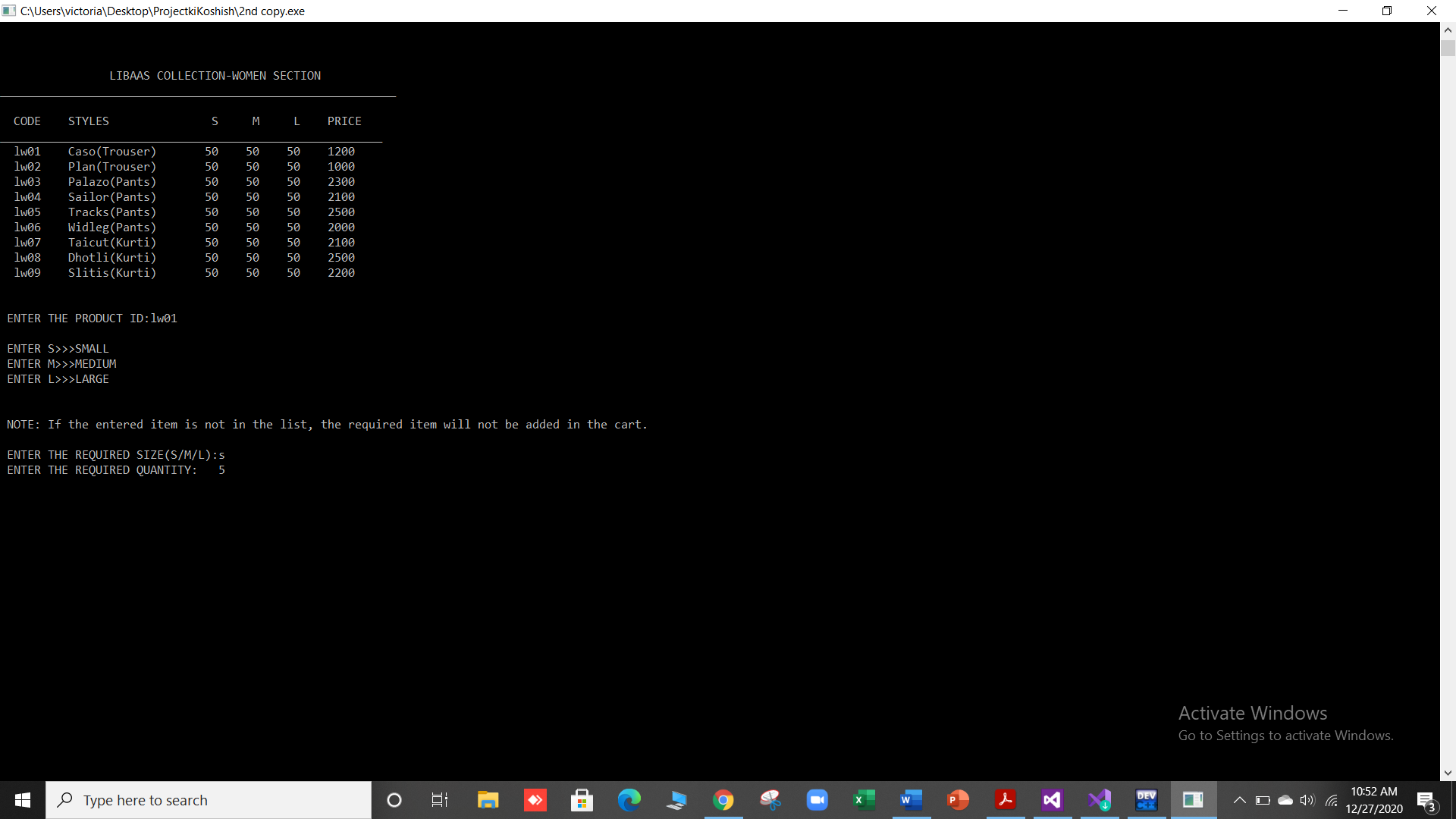


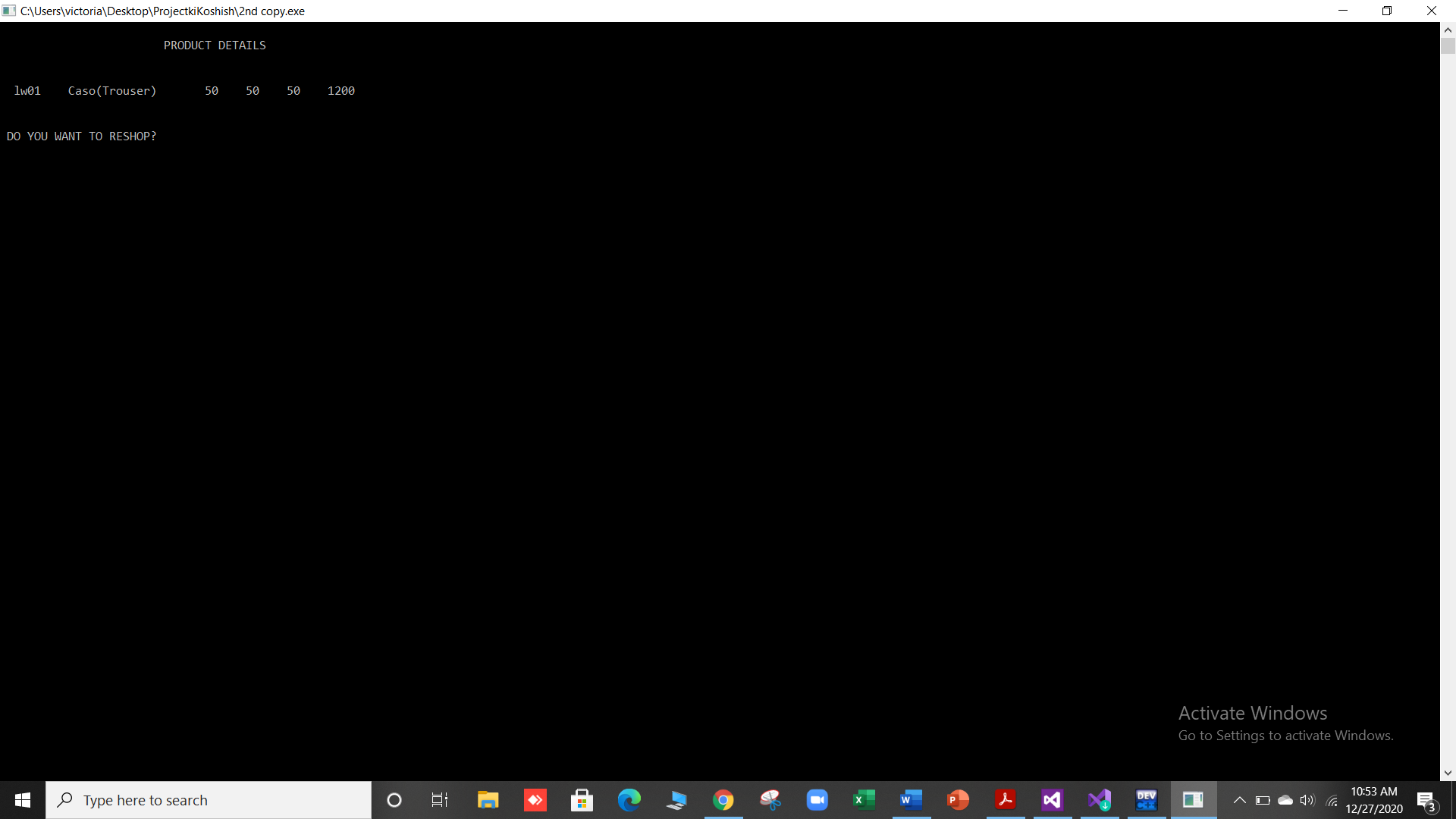
Figure 6

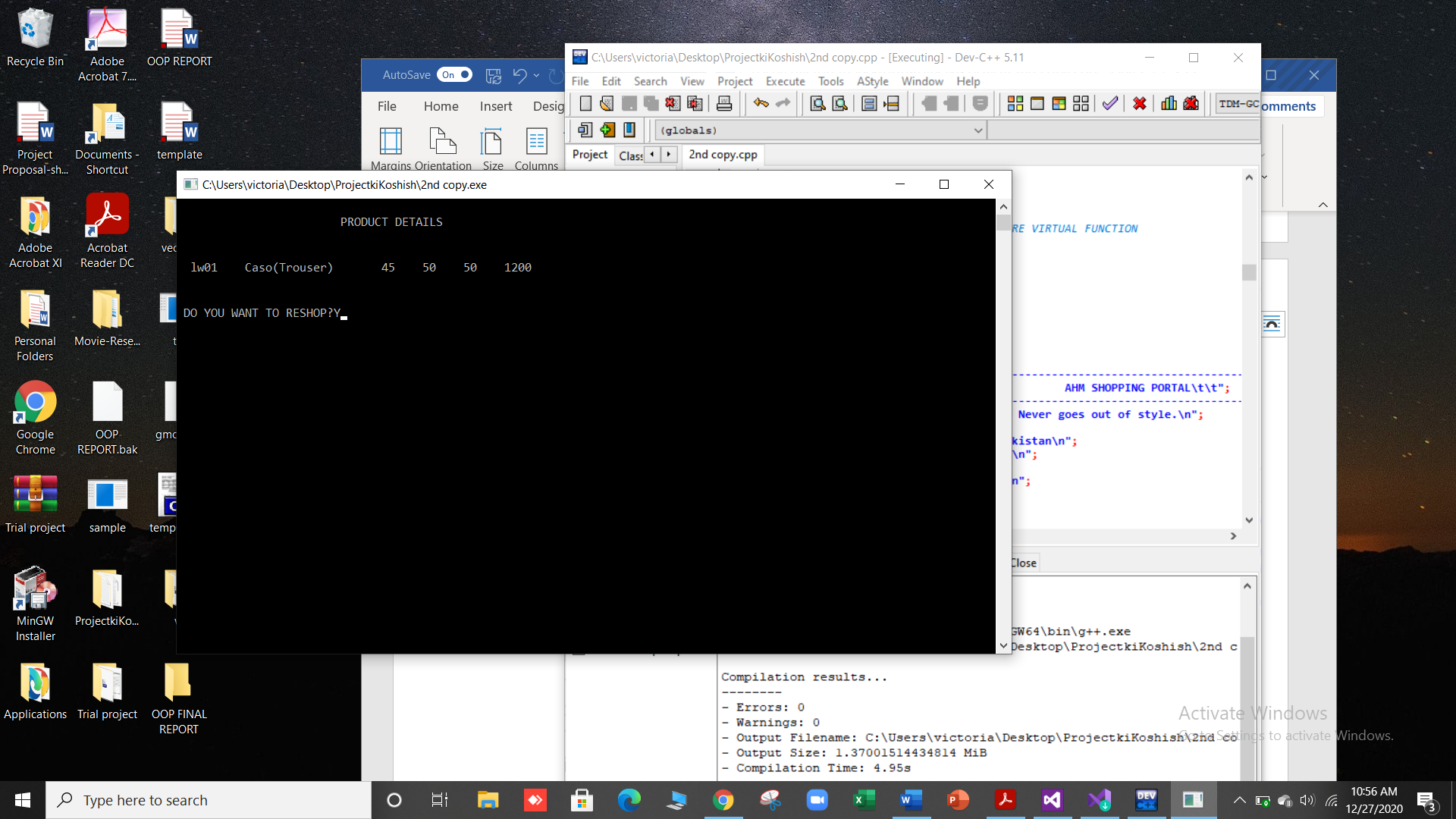
**CUSTOMER Panel:**

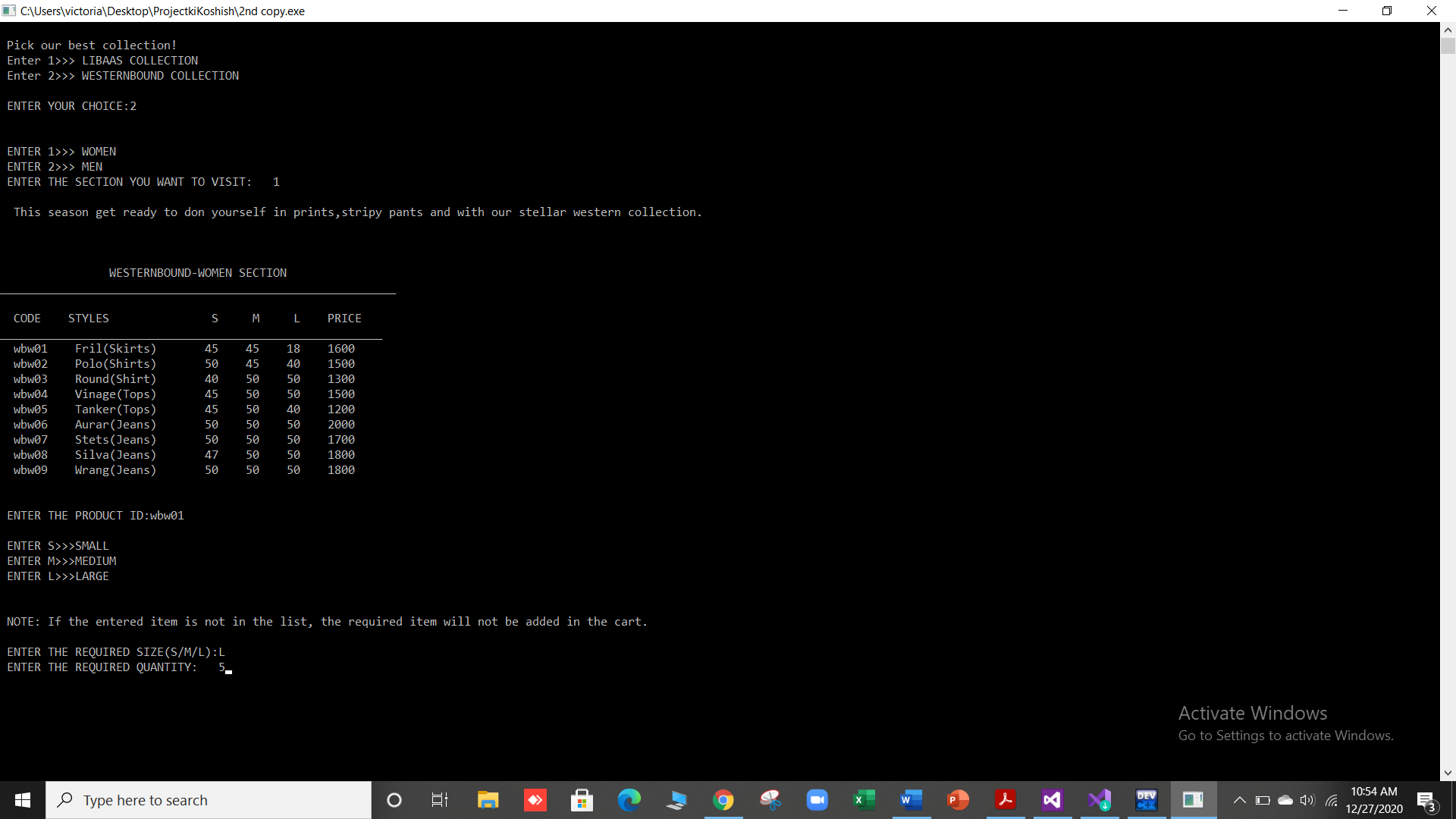


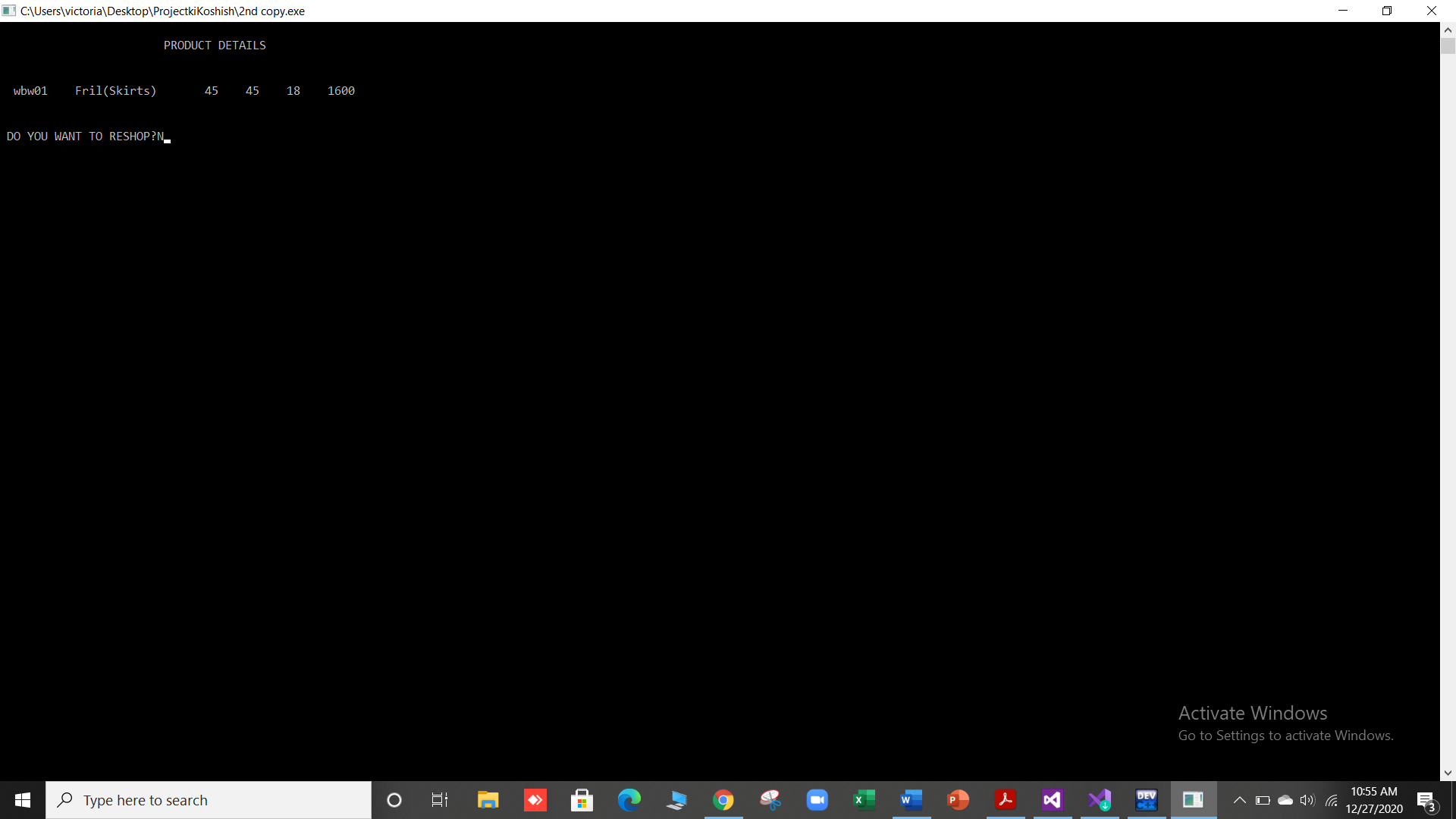


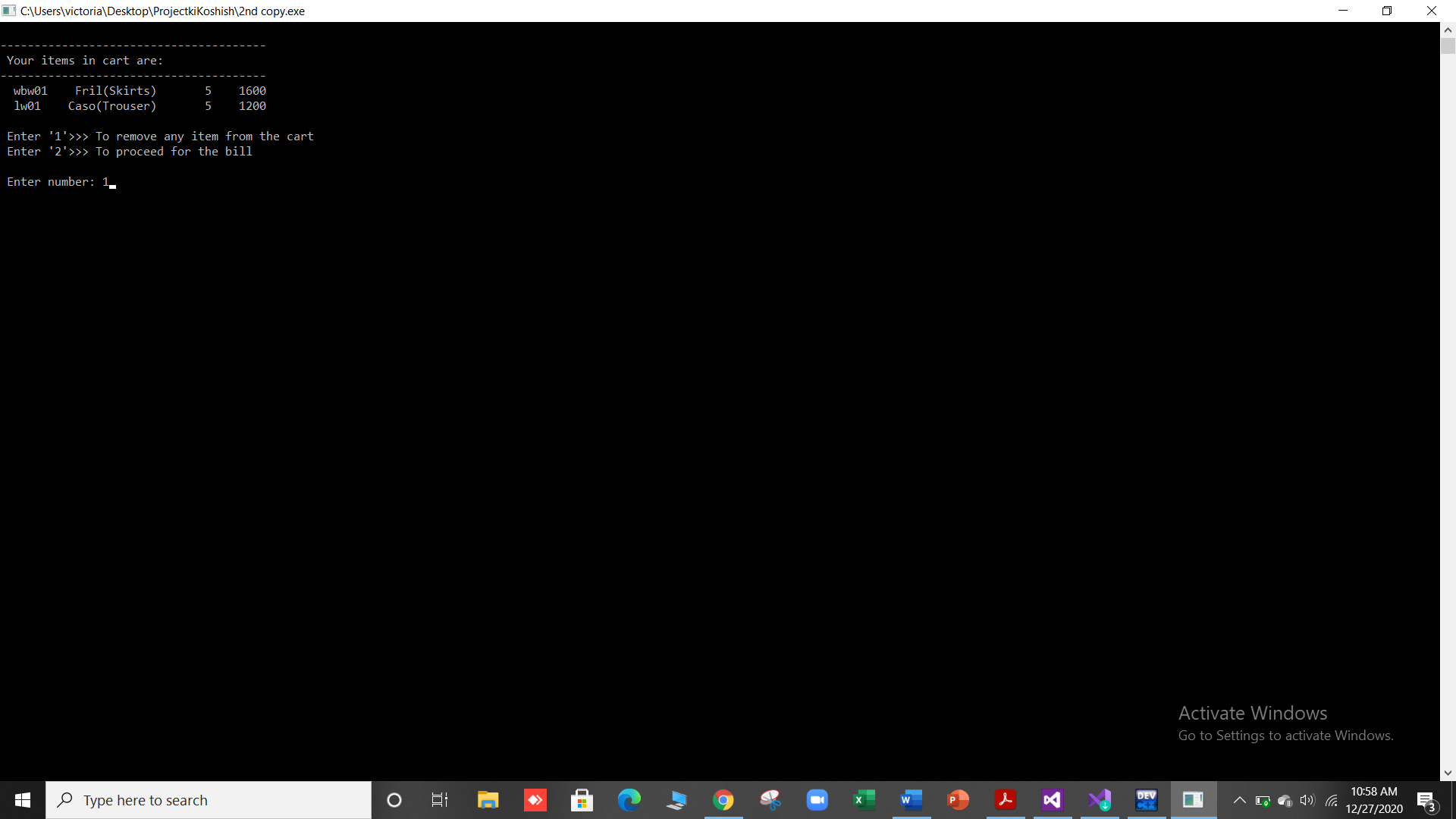


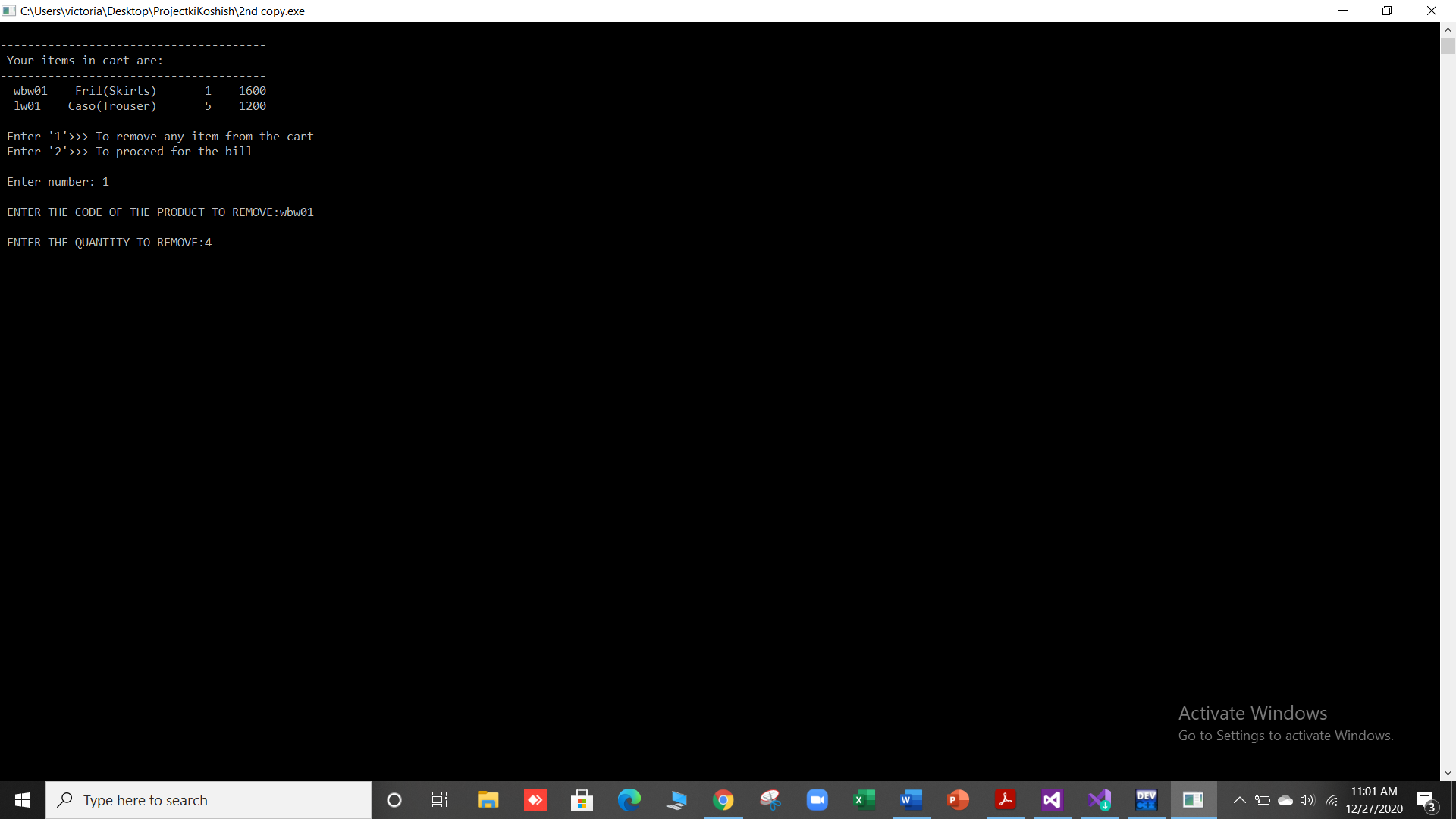


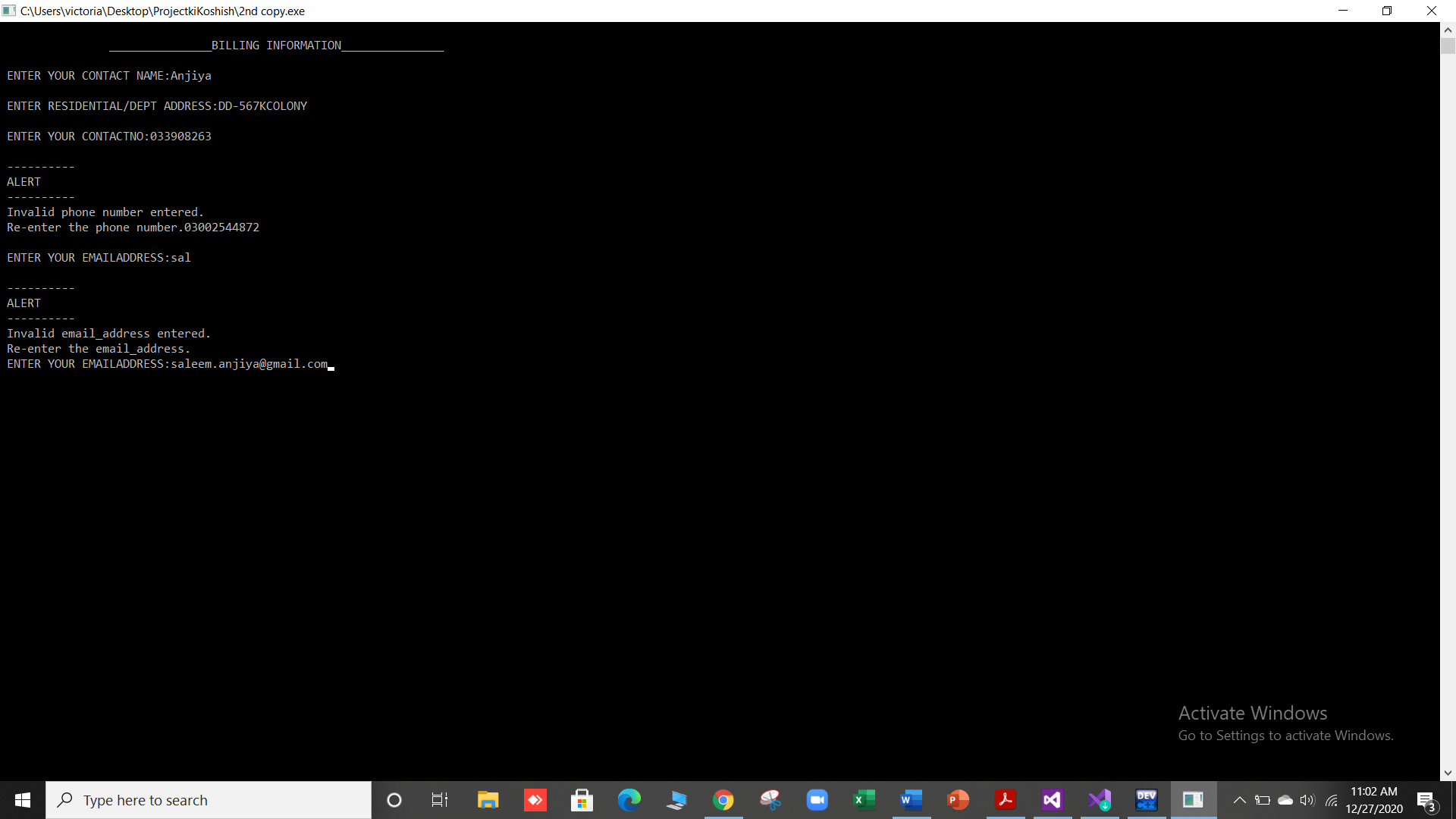


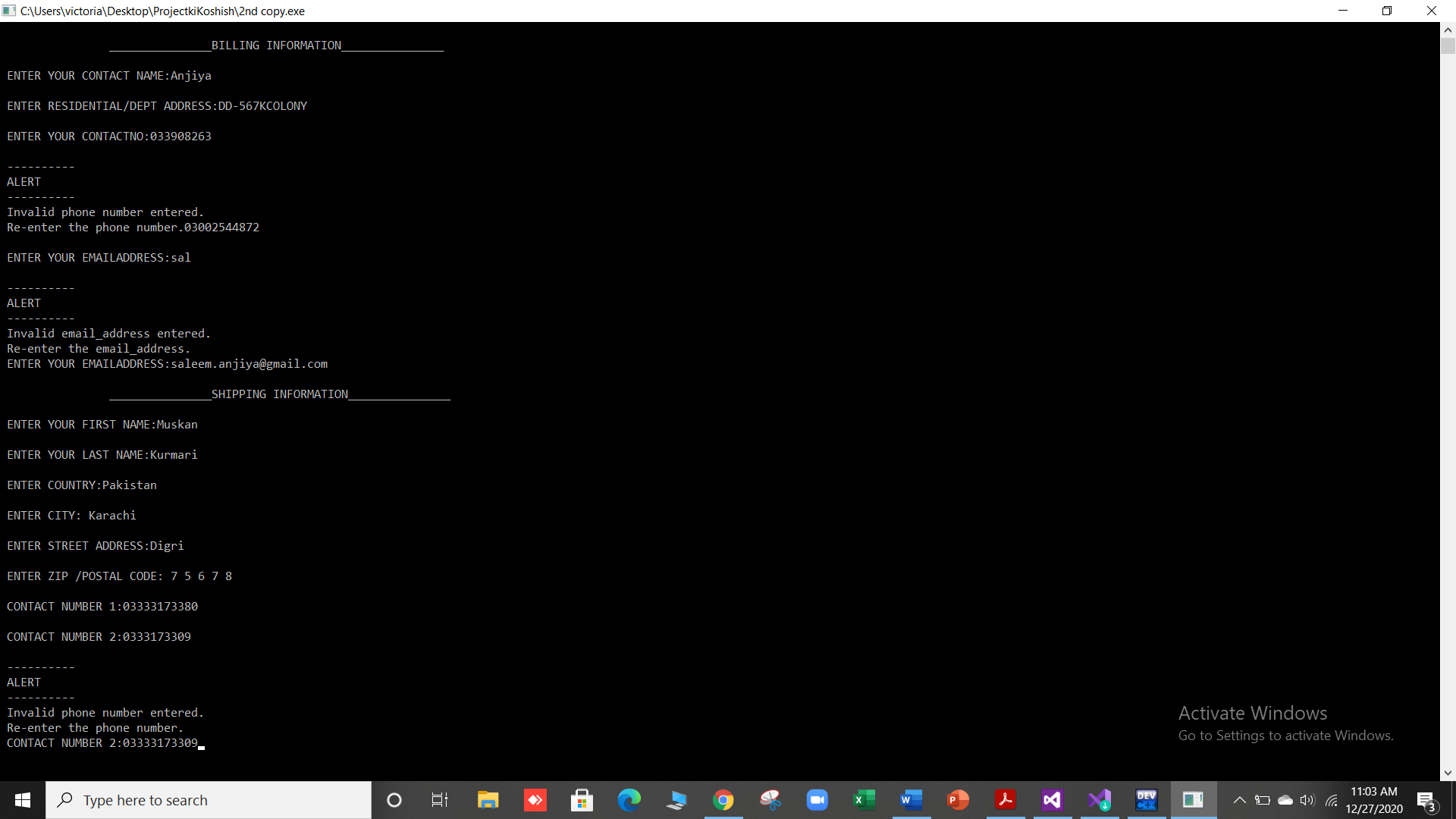


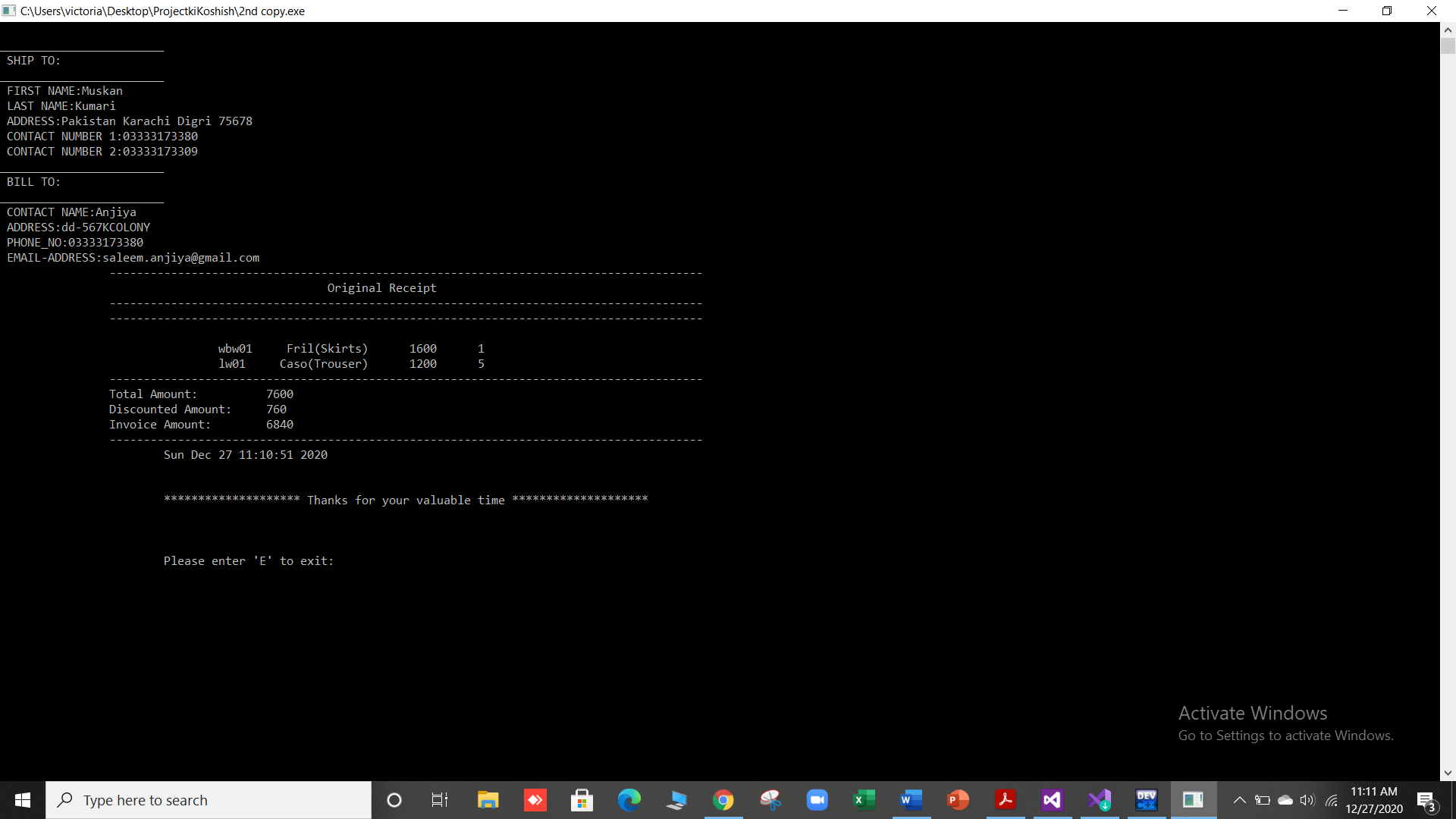














XXX------------XXX---------XXX