Solo Project (2):

General Requirements:

1. New repository on GitHub with a proper naming (Invoicing System on Console)
2. Commits should be with meaningful comments.
3. Commits at least 2 times every 1 hour.
4. Task duration: 2 full days.
5. Open book – you can use google / chatGPT or any book you want.
6. Please use your own code, code parts will be questioned and if any copy-pasting found, student will answer to a committee.

Build application to represent Invoicing system for a groceries shop.

Each invoice contains, customer full name, phone number, invoice date, number of items, total amount, paid amount and balance. On every item, each item contains item ID, item name, unit price, quantity and qty amount/price.

Application Main Menu:

1-Shop Settings

1. Load Data (Items and invoices)
2. Set Shop Name (data should be saved)
3. Set Invoice Header (Tel / Fax / Email / Website) (Data should be saved)
4. Go Back

2- Manage Shop Items

1. Add Items (Item should be saved/serialized)
2. Delete Items
3. Change Item Price
4. Report All Items
5. Go Back

3- Create New Invoice (Invoices should be save/serialized)

4- Report: Statistics (No Of Items, No of Invoices, Total Sales)

5- Report: All Invoices (Invoice No, Invoice Date, Customer Name, No of items, Total, Balance)

6- Search (1) Invoice (Search by Invoice No and Report All Invoice details with items)

7- Program Statistics (Print each Main Menu Item with how many number selected).

8- Exit (Are you sure you want to exit? If yes, program ends, if No , then main menu reprinted again)

Requirements:

1. **Menu should be handled by a Menu class with Show method to show all types of menus. One method to show menus should be accepted.**
2. **Use any suitable data structure (write your reasons why did you choose a particular one on paper) to represent menu items – Keep in mind that this menu itmes might change in order or in number)**
3. **Make Product and Invoice classes and build a relationship between them**
4. **Use Switch case to navigate between options**
5. **Use Exception Handlers to anticipate for any errors or none valid inputs from the user.**