****

**COMPUTER ORGANIZATION AND ASSEMBLY LANGUAGE PROJECT REPORT**

**GROCERY STORE**

**Instructor: Sir Usman Khalil**

**Group Members**

**Marium Haris (15084)**

**M.Hammad Shaikh (14983)**

**Class ID: 114731**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **SNO** | **CONTENTS** | **PAGE NO** |
| 1 | Acknowledgments | 2 |
| 2 | Introduction | 3 |
| 3 | System requirements | 3 |
| 4 | System design | 3 |
| 5 | Data Flow | 4 |
| 6 | Implementation | 4 |
| 7 | Key Features | 4 |
| 8 | Testing and validations | 5 |
| 9 | Code Working with Screenshots | 5 |
| 10 | Future Enhancements | 8 |
| 11 | Conclusion | 8 |

**ACKNOWLEDGMENTS**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

We would like to express our sincere gratitude to our instructor, Sir Usman Khalil, for his invaluable guidance, support, and encouragement throughout the development of this project. His expertise and insights have been instrumental in the successful completion of the Grocery Store using Irvine MASM for x86. We appreciate his dedication and the time he invested in reviewing our work and providing constructive feedback.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**INTRODUCTION**

The Grocery Store project is an assembly language program that simulates a simple grocery store where users can view a list of items, input the item number and quantity they wish to purchase, and receive the total cost. The project is developed using Irvine MASM for x86 architecture and runs on a system with Irvine32 library support.

**SYSTEM REQUIREMENTS**

**Operating System:** Windows

**Assembler:** MASM (Microsoft Macro Assembler)

**Library:** Irvine32

**Processor:** x86

**SYSTEM DESIGN**

**ARCHITECTURE**

The system follows a modular design where different procedures handle distinct functionalities such as displaying the menu, taking item inputs, calculating prices, and displaying the total cost. The main components include:

**Display Module:** Shows the list of items and their prices.

**Input Module:** Takes user inputs for item selection and quantity.

**Calculation Module:** Computes the price based on item and quantity.

**Summary Module:** Displays the total cost and number of items purchased.

**DATA FLOW**

The data flow involves the following steps:

**Display Items:** The system displays a list of items and their respective prices.

**Take Input:** The user inputs the item number and quantity.

**Calculate Price:** The system calculates the total price for the selected item and quantity.

**Repeat or Exit:** The user decides to buy another item or exit.

**Display Summary:** The system displays the total items bought and the total price.

**IMPLEMENTATION**

**TECHNOLOGIES USED**

**Assembly Language:** The core programming language for the project.

**Irvine32 Library:** Provides useful macros and functions for input/output operations in assembly language.

**MASM:** Assembler used for compiling the assembly code.

**KEY FEATURES**

**Item Listing:** Displays a list of grocery items with their prices.

**Input Handling:** Allows users to select items and enter the quantity.

**Price Calculation:** Computes the total price for selected items.

**Loop for Multiple Items:** Users can buy multiple items in a single session.

**Summary Display:** Shows the total items bought and the total amount to be paid.

**TESTING AND VALIDATION**

**TEST CASES**

**Item Selection Validity:** Ensure that only valid item numbers (1-6) are accepted.

**Quantity Input:** Verify that the quantity entered is correctly read and processed.

**Price Calculation:** Check the accuracy of price calculation based on item price and quantity.

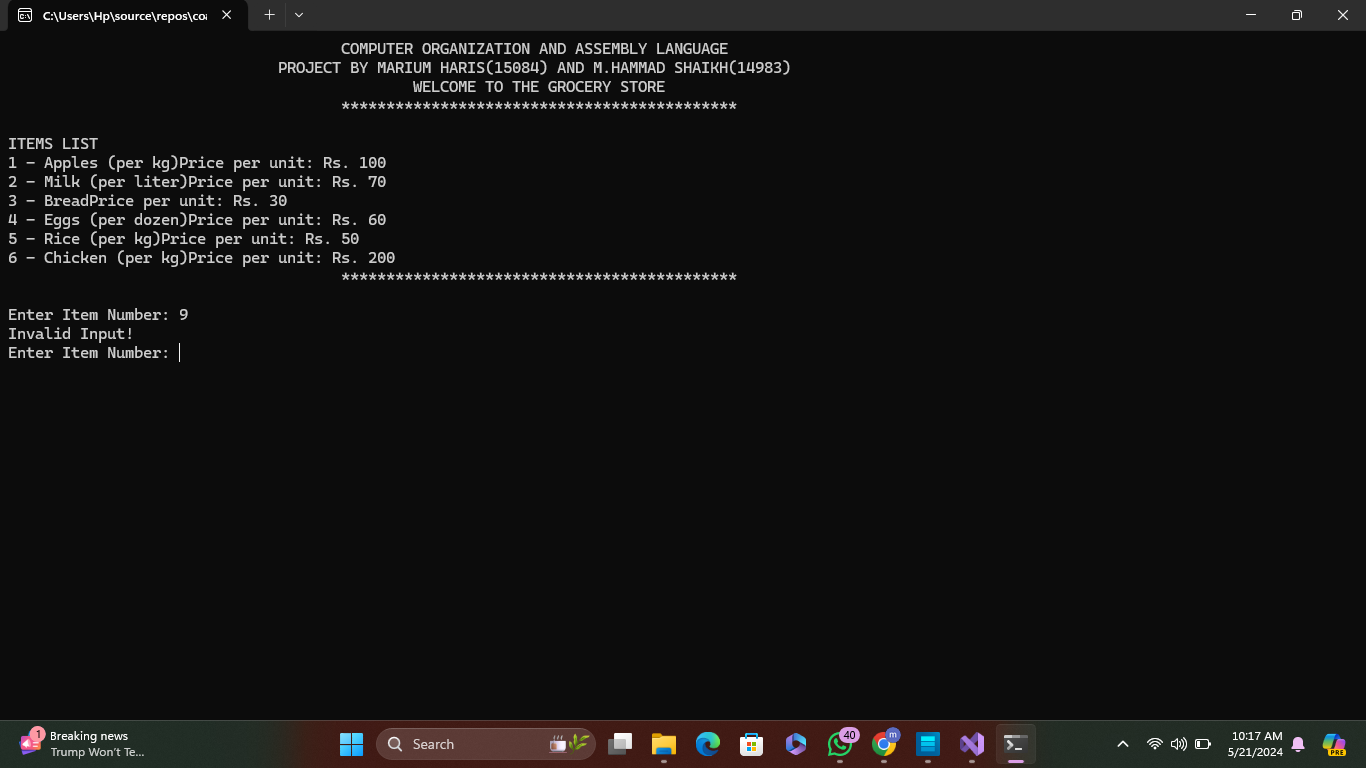
**Loop Functionality:** Test the loop for continuous item purchasing until the user decides to exit.

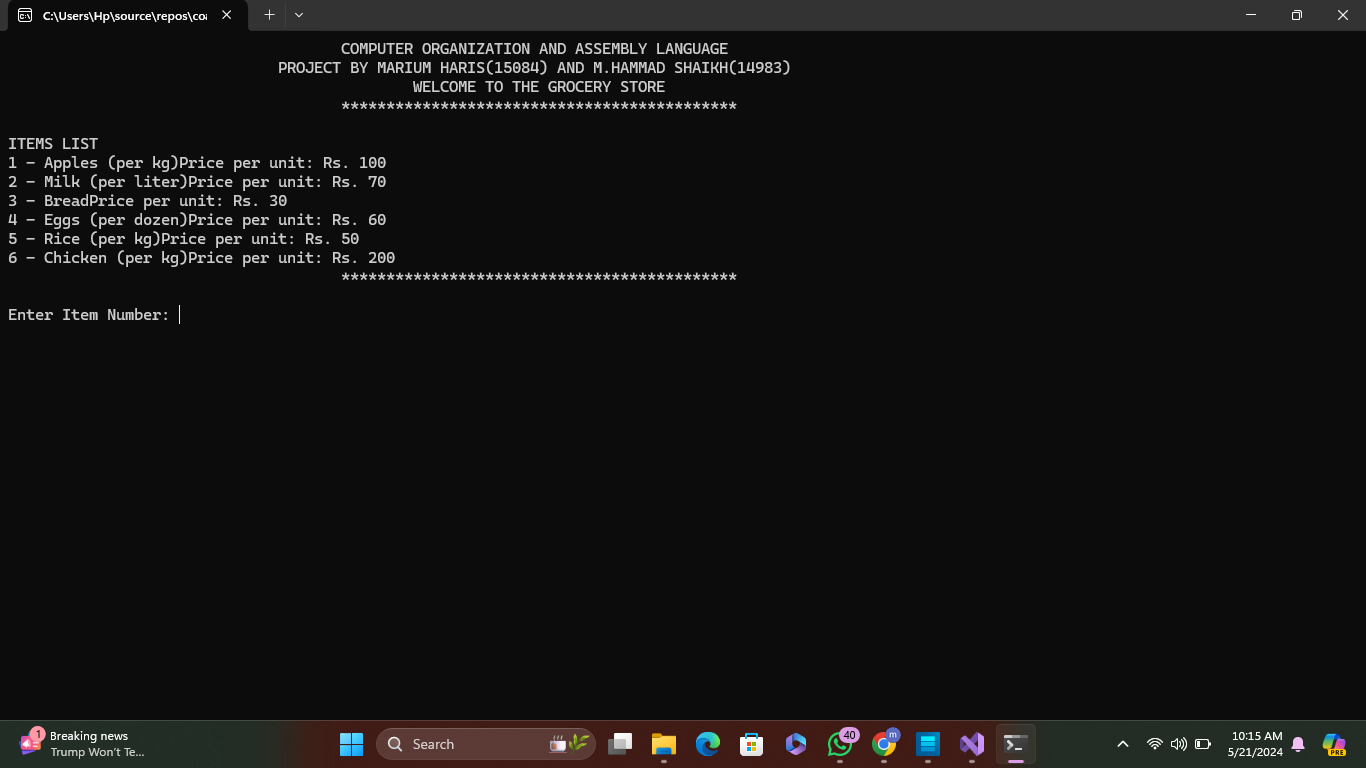
**Summary Display:** Validate the display of the total number of items and the total price.

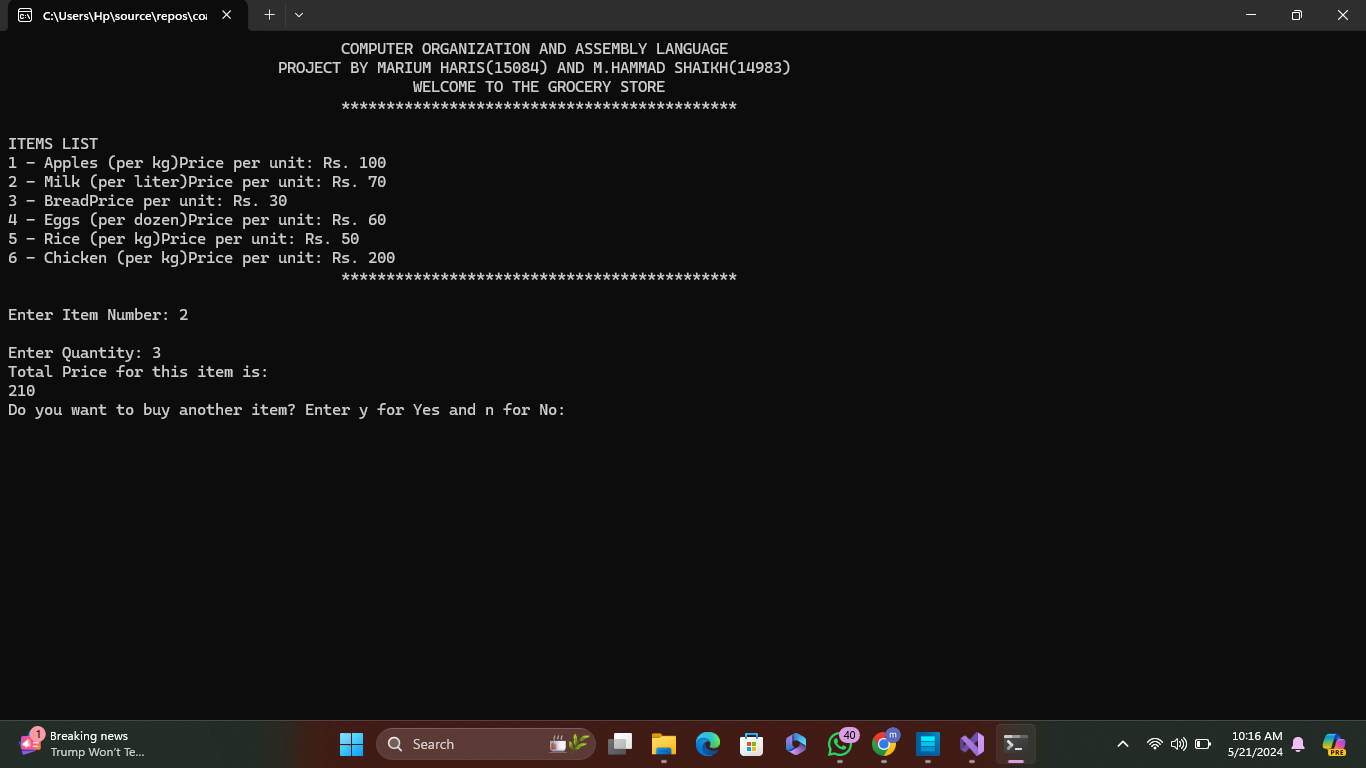
**TEST RESULTS**

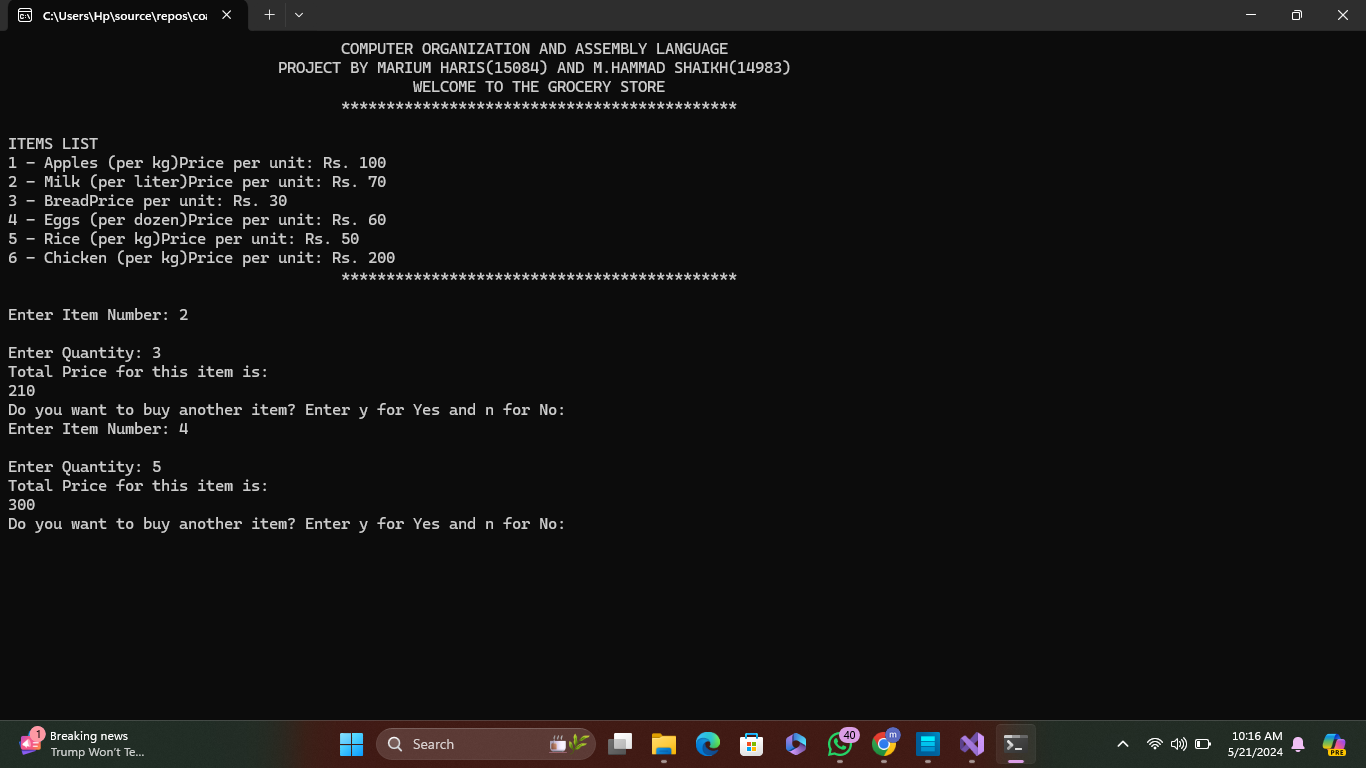
The system was tested with various inputs to ensure its reliability and correctness. All test cases passed successfully, confirming that the program meets the specified requirements and performs the intended functionalities correctly.

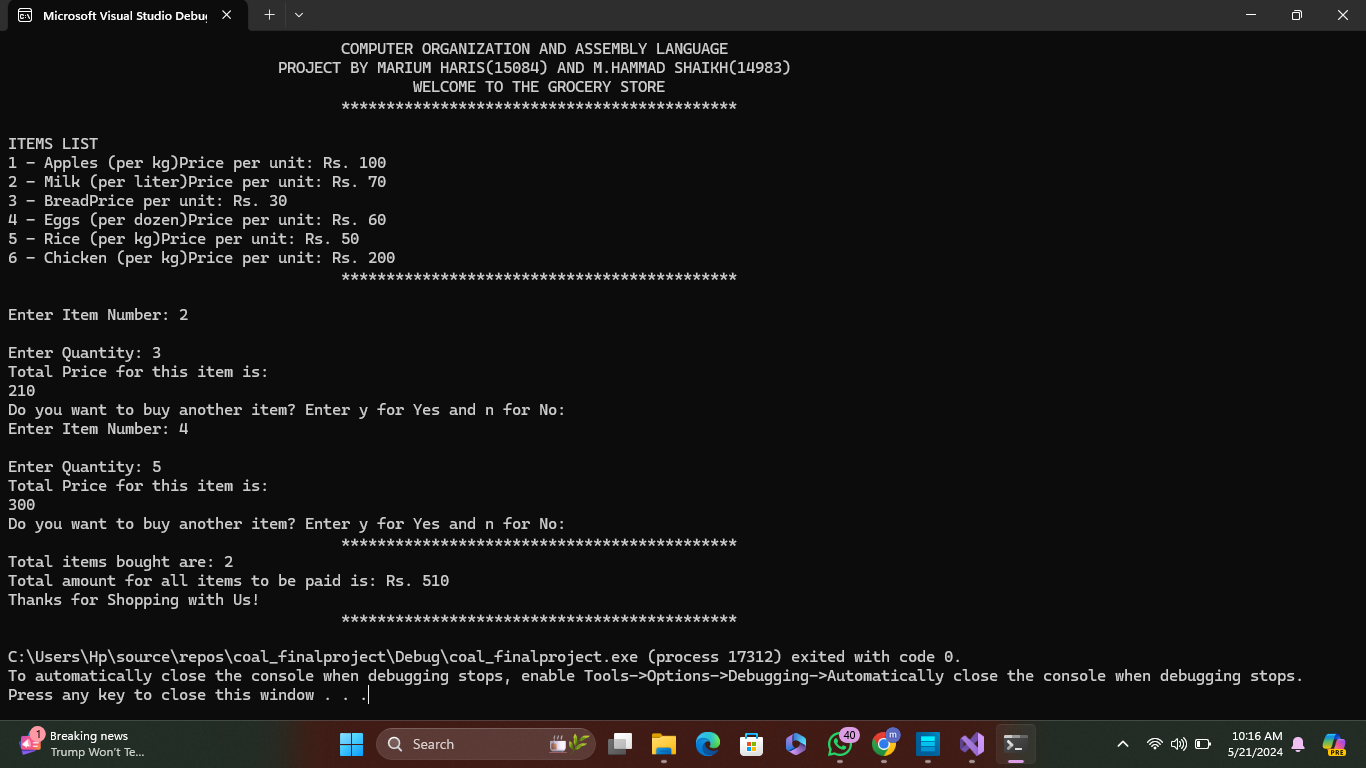
**CODE WORKING WITH SCREENSHOTS**

****

****

****

****

****

**FUTURE ENHANCEMENTS**

Future enhancements may include:

**Enhanced User Interface:** Adding graphical elements to improve user interaction.

**Discounts and Offers:** Implementing discounts and special offers for bulk purchases.

**Inventory Management:** Tracking the inventory of items and updating quantities in real-time.

**Receipt Generation:** Providing a printable receipt for the user after purchase.

**CONCLUSION**

The Grocery Store Simulation project demonstrates the practical application of assembly language programming to simulate a simple grocery store scenario. The project includes key functionalities such as item listing, input handling, price calculation, and summary display. Through rigorous testing, we ensured the system's reliability and correctness. This project provided valuable insights into low-level programming and the use of the Irvine32 library.