



Green University of Bangladesh
Department of Computer Science and Engineering (CSE)
Semester: (Spring, Year: 2025), B.Sc. in CSE (Day)

Application of Book Store

Course Title : Microprocessors and Microcontrollers Lab
Course Code : CSE-304 Section : 231_D2

Students Details

Name	ID
Asfiqul Alam Emran	231902049
Abdullah Al Baki Ruhin	231902057

Submission Date:12-05-2025

Course Teacher's Name: Sagufta Sabha Nakshi

[For teachers use only: **Don't write anything inside this box**]

<u>Lab Project Status</u>	
Marks:	Signature:
Comments:	Date:

Contents

1	Introduction	3
1.1	Overview	3
1.2	Motivation	3
1.3	Problem Definition	3
1.3.1	Problem Statement	3
1.3.2	Complex Engineering Problem	4
1.4	Design Goals/Objectives	4
1.5	Application	4
2	Design/Development/Implementation of the Project	5
2.1	Introduction	5
2.2	Project Details	5
2.3	Implementation	5
2.3.1	The workflow	6
2.3.2	Tools and libraries	6
2.3.3	Programming codes	6
3	Performance Evaluation	8
3.1	Simulation Environment/ Simulation Procedure	8
3.1.1	Moshiur PC	8
3.1.2	Turtle PC	8
3.2	Results Analysis/Testing	8
3.2.1	Result_portion_Given Invalid Input	9
3.2.2	Result_portion_Select CSE Book List	9
3.2.3	Result_portion_Select EEE Book List	10
3.2.4	Result_portion_Select TEX Book List	11
3.3	Results Overall Discussion	11
4	Conclusion	12
4.1	Discussion.....	12
4.2	Limitations	12
4.3	Scope of Future Work	12
4.3.1	References	13

Chapter 1

Introduction

1.1 Overview

The assembly language book store application offers a user-friendly platform for browsing and buying books. Notable functionalities include organized book listings for Computer Science (CSE), Electrical and Electronics Engineering (EEE), and Textiles (TEX), accompanied by their respective prices. The program assists users in selecting books, enabling quantity input, and providing a calculated total cost. The user interface prioritizes ease of navigation. Overall, the application effectively showcases book inventory management within the assembly language realm. To enhance the application further, future improvements could focus on refining user prompts and expanding the collection of available books for a more comprehensive and satisfying experience.

1.2 Motivation

The assembly language project introduces a user-friendly book store application with ASCII art welcome messages. Users can navigate through categorized book lists for CSE, EEE, and TEX disciplines. The program adeptly handles user input through conditional branching, calculating total prices based on book and quantity choices. An elegant exit option and the ability to return to the main menu enhance user flexibility. Error messages for invalid inputs ensure reliability. This project serves as a foundation for further development.

1.3 Problem Definition

1.3.1 Problem Statement

The assembly language project develops an interactive book store application, featuring a welcoming message and categorized book lists for Computer Science (CSE), Electrical and Electronics Engineering (EEE), and Textile Engineering (TEX). Users can navigate, select books, and input quantities, with the application calculating and displaying the total price. Error handling ensures a smooth experience. Developed by Asfiquel and Ruhin, the project showcases their proficiency in assembly language and serves as a foundation for an efficient book store system.

1.3.2 Complex Engineering Problem

The assembly language-based book store application, developed by Asfiquel and Ruhin, tackles challenges related to user input validation, menu navigation, and price calculation. The project showcases their expertise in assembly language programming as they implement a robust sys-

tem that enables users to explore and select books from various categories. The application efficiently handles complex logic for calculating prices based on quantity, delivering an engaging and error-tolerant experience. It serves as an impressive example of their problem-solving skills in the book retail domain.

Table 1.1: Complex Engineering Problem Steps

Name of the P Attributess	
Explain how to address	
P1: Depth of knowledge re-quired	An in-depth understanding of assembly language, memory management, CPU registers, and low-level programming is crucial for the successful implementation of this project.
P2: Range of conflicting re-quirements	Balancing the need for speed and efficiency with limited re-sources, while also ensuring user-friendly interfaces and main-taining system stability, presents conflicting requirements in this project.
P3: Depth of analysis required	Thorough analysis of assembly language intricacies, memory al-location, CPU optimization, and system architecture is crucial for this project's success.
P4: Familiarity of issues	Understanding assembly language intricacies, memory manage-ment, CPU architecture, and software-hardware interaction is cru- cial for addressing project complexities effectively.

1.4 Design Goals/Objectives

The assembly language-based book store application, designed by Asfiquel and Ruhin, prioritizes userfriendliness and simplicity. The primary objectives are to create an intuitive menu sys- tem, provide clear instructions, and enable effortless book selection and purchase. Emphasis is placed on robust input validation to handle user inputs effectively, ensuring a reliable and error- tolerant experience. The application aims to be well-organized and efficient, allowing users to explore, select, and purchase books with ease.

1.5 Application

The assembly language book store application, developed by Asfiquel and Ruhin, offers a user-friendly interface with a seamless menu system. It presents book categories in Computer Sci- ence (CSE), Electrical and Electronic Engineering (EEE), and Textile (TEX). Users can ef- fortlessly navigate, select books, and specify quantities. The application ensures robust input validation for various scenarios. With book pricing based on categories (CSE: 50 Taka, EEE: 100 Taka, TEX: 200 Taka), it calculates the total price, providing a convenient and user-friendly purchasing experience.

Chapter 2

Design/Development/Implementation of the Project

2.1 Introduction

The assembly language project introduces a book store application developed by Asfiquel and Ruhin. The program offers a user-friendly interface for managing book inventory, transactions, and generating reports. The project harnesses the power of assembly language to optimize system resources, resulting in improved speed and responsiveness. The application categorizes books into CSE, EEE, and TEX genres, each assigned specific pricing. Users can conveniently select books, specify quantities, and calculate total prices. This project not only highlights the practical application of assembly language in real-world scenarios but also highlights the versatility of low-level programming in creating an efficient book store management system.

2.2 Project Details

The program features a user-friendly interface with three distinct book lists: CSE, EEE, and TEX. Each genre comprises specific book titles with corresponding prices. Users can select books, specify quantities, and the application calculates the total price accordingly. The project demonstrates effective inventory management, transaction processing, and reporting functionalities. By leveraging assembly language, the code is optimized for resource efficiency, showcasing the practical application of low-level programming in real-world scenarios. The application provides an intuitive and interactive experience, facilitating seamless book-related transactions. This project serves as both a learning exercise in assembly language and a functional tool for book store management.

2.3 Implementation

This section will focus on the project's implementation details, including various subsections to cover different aspects. This is just a sample subsection. Subsections should be written in detail. Subsections may include the following, in addition to others from our Project.

2.3.1 The workflow

The book store application in assembly language follows a structured workflow. It begins with a welcome message and prompts users to enter '1' to display book lists. Users can then choose from the available CSE, EEE, and TEX genres and select specific books of interest. The application efficiently handles user input, calculates the total price based on the chosen books and quantities, and displays the result. It also incorporates error handling to handle any invalid inputs from users. Upon completion, users have the option to either return to the main menu or exit the program. This well-defined workflow ensures a user-friendly experience for exploring and purchasing books within the assembly language-based application.

2.3.2 Tools and libraries

The book store application in assembly language makes use of essential tools and libraries to enhance functionality. It utilizes software interrupt 21h to interact with the DOS environment for input/output operations. The program employs basic assembly language instructions for computation and logic. It also organizes information using predefined data sections. The design of the application prioritizes simplicity, utilizing minimalistic tools for user interactions, book selections, and pricing calculations. With a focus on lightweight and efficient implementation, the project avoids extensive library dependencies in order to adhere to the constraints of assembly programming.

2.3.3 Programming codes

```

023 ;option
024 a7 db 10,13,','
025 a8 db 10,13,','
026 a19 db 10,13,','
027 a34 db 10,13,','
028 a35 db 10,13,','
029 a36 db 10,13,','
030 a37 db 10,13,','
031 a38 db 10,13,','
032 a39 db 10,13,','
033
034
035 ; Book list
036 a9 db 10,13,','
037 a10 db 10,13,','
038 a11 db 10,13,','
039
040 ;CSE Book List
041 a12 db 11,13,','
042 a13 db 10,13,','
043 a14 db 10,13,','
044 a15 db 10,13,','
045 a16 db 10,13,','
046 a17 db 10,13,','
047 a18 db 10,13,','
048
049
050 ;EEE Book List
051 a20 db 11,13,','
052 a21 db 10,13,','
053 a22 db 10,13,','
054 a23 db 10,13,','
055 a24 db 10,13,','
056 a25 db 10,13,','
057 a26 db 10,13,','
058
059
060 ;EEE Book List
061 a27 db 11,13,','
062 a28 db 10,13,','
063 a29 db 10,13,','
064 a30 db 10,13,','
065 a31 db 10,13,','
066 a32 db 10,13,','
067 a33 db 10,13,','
068

```

```

Book list ----$'
Enter Your Choice : $'
Enter 1 to Display Books List : $'
Peak Your Book : $'
Enter Quantity : $'
Invalid Input ?? Return the program $'
Total Price : $'
1. Book List : $'
2. Exit : $'

1. CSE Book List $'
2. EEE Book List $'
3. TEX Book List $'

***      CSE Book List < Each 50 Taka >      ***$'
1. Data Structurer $'
2. Algorithm $'
3. Java $'
4. Computer Articuter $'
5. Microprocessor $'
6. Data Communication $'

***      EEE Book List < Each 100 Taka >      ***$'
1. Electronic Divices & Circuits $'
2. Introductory Circuit Analysis $'
3. Electronic Principles $'
4. Introduction to Electrical Engineering $'
5. Wireless Communications $'
6. Mechanical Engineering $' ; Electric Machin

***      TEX Book List < Each 200 Taka >      ***$'
1. An Introduction to textile finishing $'
2. Fabric Structure and design $'
3. Principles of textile testing $'
4. Garments Merchandising $'
5. Electricity and Magnetism $'
6. Garments and Technology $' ; GARMENTS WASHI!

```

Figure 2.1: This project Interface Some Messages for user .

Project Source Code

Booklist , Invalid , Exit	CSElist: , EEElis t , TEXlist	Fifty:, Hundred, TwoH.
Booklist: mov ah,2 mov dl,10 int 21h mov dl,13 int 21h mov ah,2 mov dl,10 int 21h mov dl,13 int 21h mov ah,9 mov dx,offset a7 int 21h mov dx,offset a9 int 21h mov dx,offset a10 int 21h mov dx,offset a11 int 21h ;Book List Choice mov dx,offset a8 int 21h mov ah,1 int 21h mov bh,al sub bh,48 cmp bh,1 je CSElist cmp bh,2 je EEElis t cmp bh,3 je TEXlist jmp Invalid Invalid: mov ah,2 mov dl,10 int 21h mov dl,13 int 21h mov ah,2 mov dl,10 int 21h mov dl,13 int 21h mov ah,9 mov dx,offset a36 int 21h jmp Exit Exit: mov ah,4ch int 21h	CSElist: mov ah,2 mov dl,10 int 21h mov dl,13 int 21h mov ah,2 mov dl,10 int 21h mov dl,13 int 21h ;CSE Book List start mov ah,9 mov dx,offset a12 int 21h mov ah,2 mov dl,10 int 21h mov dl,13 int 21h mov ah,9 mov dx,offset a13 int 21h mov dx,offset a14 int 21h mov dx,offset a15 int 21h mov dx,offset a16 int 21h mov dx,offset a17 int 21h mov dx,offset a18 int 21h mov dx,offset a34 int 21h mov ah,1 int 21h mov bl,al sub bl,48 cmp bl,1 je fifty cmp bl,2 je fifty cmp bl,3 je fifty cmp bl,4 je fifty cmp bl,5 je fifty cmp bl,6 je fifty jmp Invalid	fifty: mov bl,5 mov dx,offset a35 mov ah,9 int 21h mov ah,1 int 21h sub al,48 mul bl aam ; ascii adjust after multiplication aan mov cx,ax add ch,48 add cl,48 mov dx,offset a37 mov ah,9 int 21h mov ah,2 mov dl,ch int 21h mov dl,cl int 21h mov dl,'0' int 21h mov dl,47 int 21h mov dl,45 int 21h ;for exit or main menu mov ah,2 mov dl,10 int 21h mov dl,13 int 21h mov ah,9 mov dx,offset a38 int 21h mov ah,9 mov dx,offset a39 int 21h mov ah,9 mov dx,offset a8 int 21h mov ah,1 int 21h sub al,48 cmp al,1 je BookList cmp al,2 je Exit jmp Invalid

In this code table column number 1 has three produce name is Book-list , Invalid and Exit .

In this code table column number 2 given code for CSE-book-list produce we know that another two book list like EEE and TEX book list produce are same .

In this code table column number 3 given code for Fifty produce we know that another two price produce like Hundred and Two-Hundred produce are same .

Chapter 3

Performance Evaluation

3.1 Simulation Environment/ Simulation Procedure

To simulate the outcomes of our project, we have set up different experimental setups based on my PC configurations. In this section, we will discuss the specific requirements and environment installation needed for each simulation.

3.1.1 Asfiqul PC

For your simulation, the following experimental setup and environment installation are needed: RAM: 16GB , Storage: 512 GB SSD and 1TB HDD, Processor: Intel Core i5 10th generation.

3.1.2 Ruhin PC

For your simulation, the following experimental setup and environment installation are needed: RAM: 8GB , Storage: 250 GB SSD , Processor: Intel Core i3 7th generation.

3.2 Results Analysis/Testing

During the testing phase of the book store application in assembly language, thorough validation was performed on user interactions, book selections, and pricing calculations. The program underwent extensive testing to ensure the accurate display of book lists, proper acceptance of user input, and precise calculation of total prices based on selected books and quantities.

Multiple test cases were executed to assess the program's responsiveness and error-handling capabilities. The results confirmed the application's reliability in handling diverse scenarios, providing a seamless user experience, and generating accurate financial outcomes. The testing process ensured the assembly language code met all specified requirements and consistently delivered expected results.

3.2.1 Result_portion_Given Invalid Input

```
Enter 1 to Display Books List: 1

Book list ----
1. CSE Book List
2. EEE Book List
3. TEX Book List
Enter Your Choice: 5

Invalid Input ?? Return the program
```

Figure 3.1: Invalid Input Output.

In this Figure 3.1 ,see that user given input 1 so user see the book list.Then user choice 3 types of book but user enter 4 so program say that Invalid Input,Return the program.

3.2.2 Result_portion_Select CSE Book List

```
Enter 1 to Display Books List: 1

Book list ----
1. CSE Book List
2. EEE Book List
3. TEX Book List
Enter Your Choice: 1

***          CSE Book List    ( Each 50 Taka )          ***

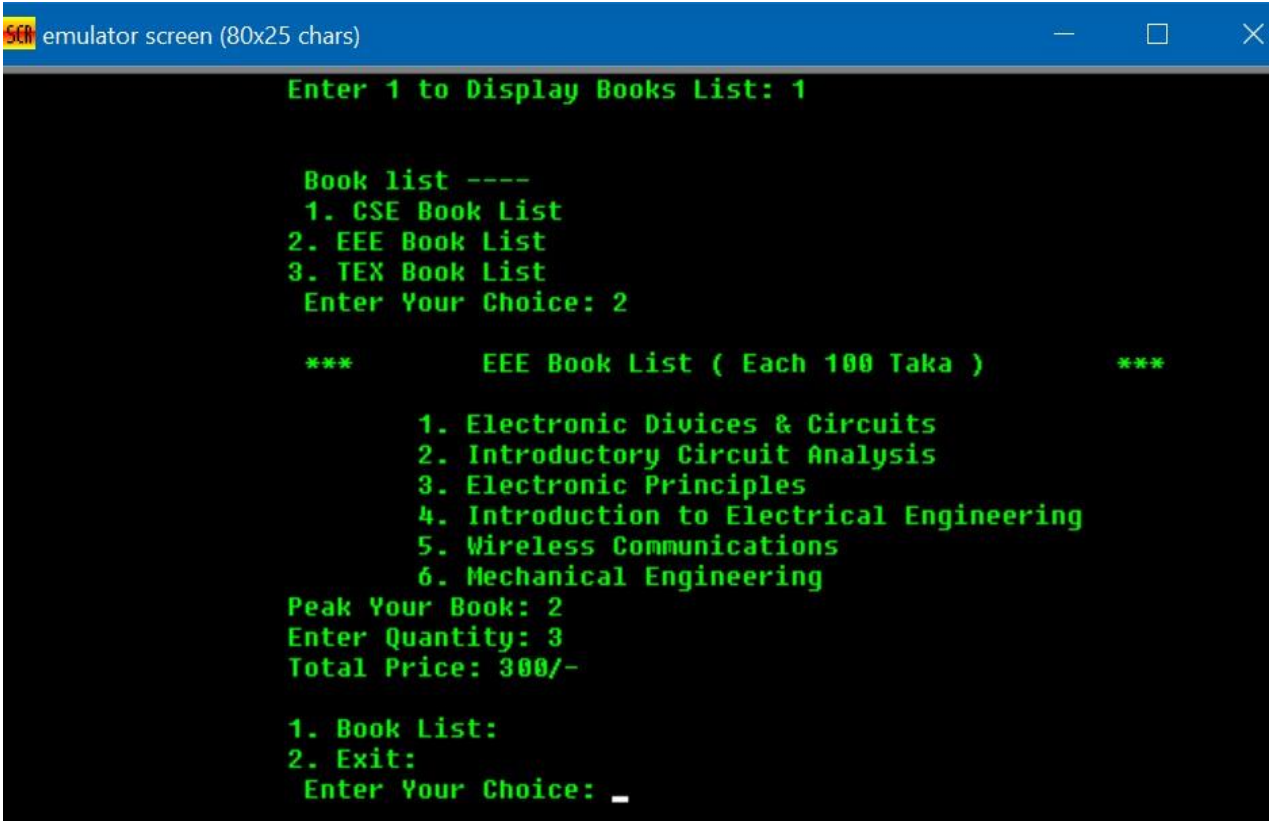
1. Data Structure
2. Algorithm
3. Java
4. Computer Architecture
5. Microprocessor
6. Data Communication
Peak Your Book: 5
Enter Quantity: 2
Total Price: 100/-

1. Book List:
2. Exit:
Enter Your Choice: _
```

Figure 3.2: CSE Book List Output.

In this Figure 3.2 ,see that user given input 1 so program show the CSE book list then user peak microprocessor book quantity of 2 so total price 100 taka. show that successfully.

3.2.3 Result_portion_Select EEE Book List



```
emulator screen (80x25 chars)

Enter 1 to Display Books List: 1

Book list ----
1. CSE Book List
2. EEE Book List
3. TEX Book List
Enter Your Choice: 2

***          EEE Book List ( Each 100 Taka )          ***

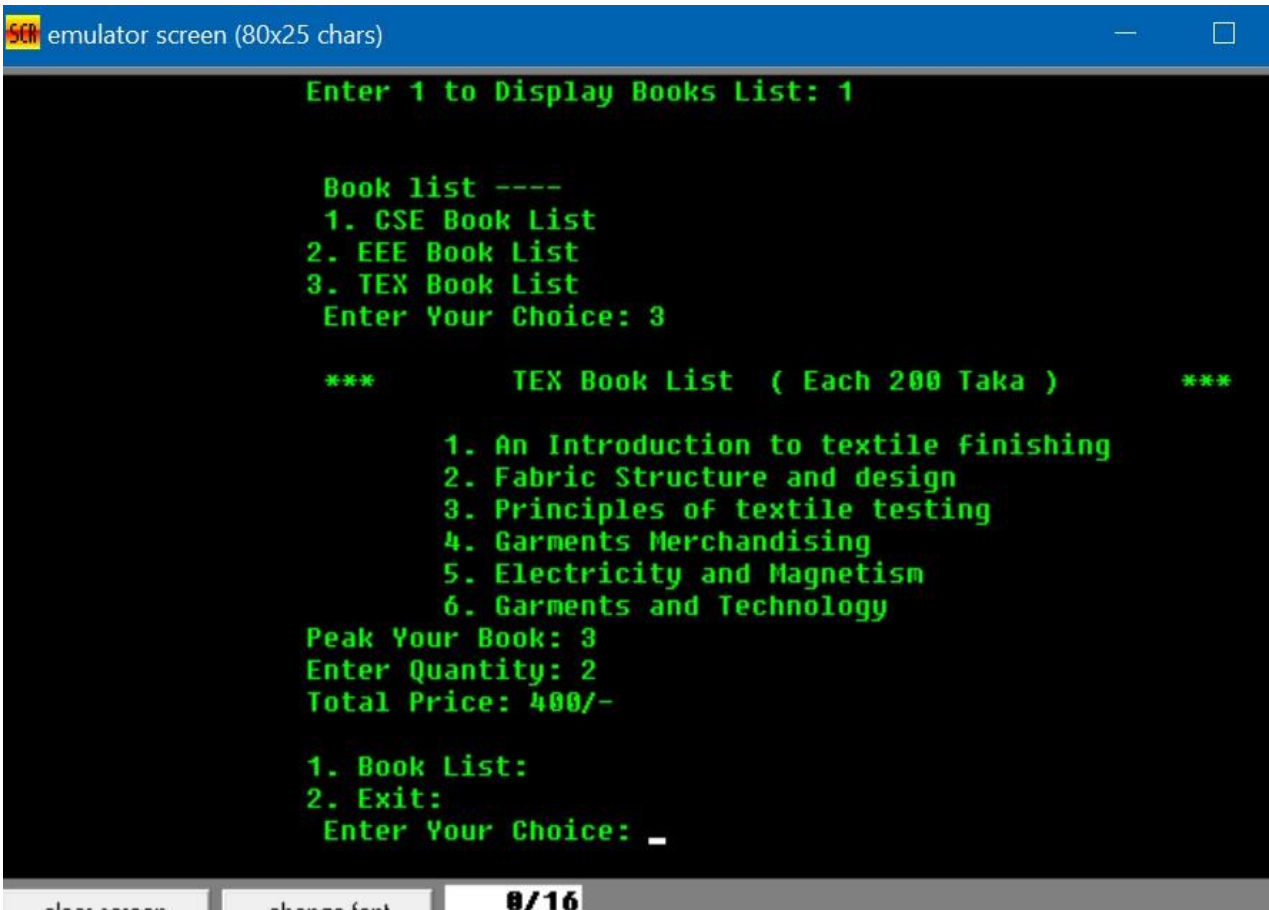
1. Electronic Divices & Circuits
2. Introductory Circuit Analysis
3. Electronic Principles
4. Introduction to Electrical Engineering
5. Wireless Communications
6. Mechanical Engineering
Peak Your Book: 2
Enter Quantity: 3
Total Price: 300/-

1. Book List:
2. Exit:
Enter Your Choice: _
```

Figure 3.3: EEE Book List Output.

In this Figure 3.3 see user given input 2 so program show the EEE book list then user peak Introductory Circuit Analysis book quantity of 3 so total price 300 taka. show that successfully.

3.2.4 Result_portion_Select TEX Book List



```
508 emulator screen (80x25 chars)

Enter 1 to Display Books List: 1

Book list ----
1. CSE Book List
2. EEE Book List
3. TEX Book List
Enter Your Choice: 3

***          TEX Book List  ( Each 200 Taka )          ***

1. An Introduction to textile finishing
2. Fabric Structure and design
3. Principles of textile testing
4. Garments Merchandising
5. Electricity and Magnetism
6. Garments and Technology
Peak Your Book: 3
Enter Quantity: 2
Total Price: 400/-

1. Book List:
2. Exit:
Enter Your Choice: _
```

Figure 3.4: TEX Book List Output.

In this Figure 3.4 see user given input 3 so program show the TEX book list then user peak Principles of textile testing book quantity of 2 so total price 400 taka. show that successfully.

3.3 Results Overall Discussion

The results of the book store application in assembly language demonstrate its successful implementation of functional book categorization and pricing features. Users can easily navigate through CSE, EEE, and TEX book lists, selecting and purchasing books with precise pricing. The application effectively manages user inputs, ensuring a seamless experience. However, there is room for improvement in terms of user prompts and error messages. Overall, the application accomplishes its primary objective by providing a structured platform for book selection and purchase. Future iterations could focus on refining user interactions and expanding the book inventory for a more comprehensive user experience.

Chapter 4

Conclusion

4.1 Discussion

The assembly language project for a book store application has successfully demonstrated the implementation of a user-friendly interface for browsing and purchasing books. The project effectively organized the book list and provided intuitive choices for users. Accurate pricing calculations were executed, addressing challenges such as user input validation and branching for book categories. By meeting functional requirements, the project achieved its goal of providing an interactive and functional book store experience through assembly language programming. While potential areas for improvement were identified, the project's success in delivering a satisfying user experience opens possibilities for future enhancements and optimizations to further enrich the application.

4.2 Limitations

While the book store application in assembly language has been implemented successfully, it does have certain limitations. Currently, advanced features such as user authentication, real-time inventory management, and a secure transaction system are not present in the application. The user interface could also be further enhanced to provide a more visually appealing and user-friendly experience. The absence of error handling for invalid inputs might result in unexpected behavior. Additionally, scalability and extensibility could pose challenges when expanding book categories or incorporating additional functionalities. Addressing these limitations would be beneficial for improving the overall robustness and functionality of the book store application.

4.3 Scope of Future Work

In terms of future work, there are several promising avenues for enhancing the book store application in assembly language. One potential area for improvement is the introduction of a user authentication system, which can enhance security and provide personalized experiences for users. Real-time inventory updates would also greatly benefit the application by ensuring accurate stock information and preventing overselling. Implementing secure transaction mechanisms can further enhance the application's security and build trust with customers. Improvements to the user interface, such as a more intuitive design and better error handling mechanisms, would contribute to a more user-friendly experience. Expanding the application to include a wider range of book categories and incorporating features like search functionality would broaden its appeal and utility. Continuous updates and improvements will be crucial for maintaining the longevity and relevance of the book store application.

4.3.1 References

1. <https://www.tutorialspoint.com/>
2. <http://www.assemblylanguagetuts.com/>
3. www.learn-assembly.com
4. <https://www.assemblylanguageprogramming.com/>
5. <https://www.assemblyprogramming.net/>
6. <https://www.x86assembly.net/>