

**TRIBHUVAN UNIVERSITY**

**INSTITUTE OF ENGINEERING**

**THAPATHALI CAMPUS**

**Proposal**

**On**

**Book Store Management System**

**Submitted by:**

Alok Sharma (THA081BEI004)

Anvil Shakya (THA081BEI006)

Aviyan Sharma Khabas (THA081BEI008)

Nishan Parajuli (THA081BEI023)

**Submitted to:**

Department of Electronics and Computer Engineering

Thapathali Campus

Kathmandu,Nepal

February, 2025

**ABSTRACT**

Our project is aimed to develop our ready-to-go Book Store Management System developed in the C programming language. The design of this project is to streamline the management of a book store, streamlining the management of products and transactions. The system focuses on the efficient and user-friendly interface for the store keeper to perform essential operations such as adding, updating, and deleting product details, tracking transactions, maintenance schedules and costs, The program utilizes file handling to store and retrieve data, ensuring consistent reliability. Key feature includes functionality to search for the available books, sales history and the dealer list. The project emphasizes on modular programming, with functions organized for clarity and reusability. By automating routine tasks with this Book Store Management System, a proper experience can be achieved through enhanced operational efficiency and reduced manual errors for the said store keeper. By applying the concepts of core C programming such as file handling and control flow, we can address the real-world challenges in handling a book store through this project.

*Keyword(s): Modular Programming, Book Store Management System,* [*catechisms*](https://en.wikipedia.org/wiki/Catechisms),[*almanacs*](https://en.wikipedia.org/wiki/Almanacs)*, Gospels, Missals*

**Table of Contents**

**ABSTRACT...................................................................................................................i**

**Table of Contents ........................................................................................................ ii**

**List of Figures..............................................................................................................iii**

**List of Tables…………………………………………………………………………iv**

**List of Abbreviations...................................................................................................v**

**1. INTRODUCTION....................................................................................................1**

1.1 Background Introduction…………………………………....................1

1.2 Motivation..............................................................................................1

1.3 Problem Definition.................................................................................1

1.4 Objectives...............................................................................................2

1.5 Applications…………………………………………………………………2

**2. LITERATURE REVIEW........................................................................................3**

3. **PROPOSED SYSTEM ARCHITECTURE............................................................5**

3.1 Block Diagram or System Architecture..................................................5

3.2 Parts of the Program…………………………........................................5

3.3 Tools and Environment...........................................................................6

**4. METHODOLOGY…...............................................................................................8**

4.1 Header Files and Namespaces................................................................8

4.2 Functions and Conditional Statements.……………………………… 8

**5. TIME ESTIMATION ..............................................................................................9**

**6. FEASIBILITY ANALYSIS...................................................................................10**

6.1 Usage…………………………………………………………………10

6.2 Cost Analysis…………………………………………………………10

6.3 Hardware Requirement……………………………………………….10

6.4 Software Requirement………………………………………………..10

**REFERENCES………………………………………………………………….......11**

**List of Figures**

Figure 1: Block Diagram of the Application…………………………………………...4

Figure 2: Store Activity Interface………………………………………………………5

**List of Tables**

Table 5-1: Time Estimation Gantt Chart……………………………………………….8

**List of abbreviations**

BSM Book Store Management System

**1. INTRODUCTION**

**1.1 Background**

Book stores often deal with volumes of books being brought in and traded. This requires intensive amount of time dedicated to record-keeping of each and every functionality that is happening in the store’s day to day activities. But in the modern world, even with the advancement of technologies, traditional methods of record-keeping, such as ledgers and spreadsheet are still common, consequently killing time due to errors and inefficiency. A need for automated systems that aid to the said issues have been a must in today’s world focusing on small-medium sized bookstore to enterprises as well. A BSMS is a software solution dedicated to address those very challenges by automating key store activities such as book cataloging, store maintenance record, dealership records and sales tracking. By digitizing those processes, a BSMS reduces manual effort alongside with better organization, faster access to information and overall improved experience. This project focuses on using the C programming language specifically using file handling, as C is efficient, has low-level control and portability making it an ideal choice for building a compelling system.

**1.2 Motivation**

As our first project on C programming language, we as a group decided to create this straightforward yet purposeful application. A real-world necessity was also observed in numerous situations where automation was deduced to be a viable factor to improve efficiency in these book stores. Alongside with learning and reinforcing our concepts of the programming language for a strong foundation, we looked out to conform to those ideas. Since the chance we were receiving to create for benefactory causes is the main idea, we believe that this system will prove to be a strong example to set up confidence for the upcoming challenges in the later years to come.

**1.3 Problem Definition**

The said project aims to create efficient method of keeping records which involve sales, maintenance, customers and overall book store functionality.

**1.4 Project Objectives**

The objectives are as follows:

* To automate the manual processes of book store management,
* To maintain database of products specifically books and subsidiary stationary items,
* To facilitate the employees in their specified tasks,
* To generate reports on book availability, sales and maintenance.

**1.5 Project Applications**

Our proposed project put forward for following applications:

* Strengthening the foundation of programming,
* Developing flexible and easy to maintain application with reliability and efficiency,
* Maintaining the versatility of the program.

**2. LITERATURE REVIEW**

The selling of [books](https://en.wikipedia.org/wiki/Book) dates back to ancient times. The founding of [libraries](https://en.wikipedia.org/wiki/Libraries) in c.300 BC stimulated the energies of the [Athenian](https://en.wikipedia.org/wiki/Athenian) [booksellers](https://en.wikipedia.org/wiki/Booksellers). In [Rome](https://en.wikipedia.org/wiki/Ancient_Rome), toward the end of the [republic](https://en.wikipedia.org/wiki/Roman_Republic), it became the fashion to have a library, and Roman booksellers carried on a flourishing trade. In [Rome](https://en.wikipedia.org/wiki/Ancient_Rome), toward the end of the [republic](https://en.wikipedia.org/wiki/Roman_Republic), it became the fashion to have a library, and Roman booksellers carried on a flourishing trade. The spread of [Christianity](https://en.wikipedia.org/wiki/Christianity) naturally created a great demand for copies of the[*Gospels*](https://en.wikipedia.org/wiki/Gospel) and other sacred books, and later on for *missals* and other devotional volumes for both church and private use. The modern system of bookselling dates from soon after the introduction of [printing](https://en.wikipedia.org/wiki/Printing). Through the new mechanized process for printing, books became more affordable.[[11]](https://en.wikipedia.org/wiki/History_of_bookselling#cite_note-11) By the nineteenth century, the model of bookselling as we know it began to emerge. A professional group of booksellers in [Leipzig](https://en.wikipedia.org/wiki/Leipzig) decided to form their own association in 1824, and in 1825 the *[Börsenverein der Deutschen Buchhändler zu Leipzig](https://en.wikipedia.org/w/index.php?title=B%C3%B6rsenverein_der_Deutschen_Buchh%C3%A4ndler_zu_Leipzig&action=edit&redlink=1" \o "Börsenverein der Deutschen Buchhändler zu Leipzig (page does not exist))* [German] became the first group to publish outside of the printer's guilds, leading to more people joining the profession without needing to be attached to a guild. The earliest printers were also [editors](https://en.wikipedia.org/wiki/Editing) and booksellers; but being unable to sell every copy of the works they printed, they had agents at most of the seats of learning, such as [Anton Koberger](https://en.wikipedia.org/wiki/Anton_Koberger), who introduced the art of printing into [Nuremberg](https://en.wikipedia.org/wiki/Nuremberg) in 1470[1].The most common types of books printed in large quantities were able to be cheaply produced like [*catechisms*](https://en.wikipedia.org/wiki/Catechisms) and [*almanacs*](https://en.wikipedia.org/wiki/Almanacs) and often not bound at all. It became the foundation for modern book sales and policies such as *copyright infringement* were born. Bookstores often sell other printed matter besides books, such as [newspapers](https://en.wikipedia.org/wiki/Newspapers), [magazines](https://en.wikipedia.org/wiki/Magazines), and [maps](https://en.wikipedia.org/wiki/Map); additional product lines may vary enormously, particularly among independent bookstores. [Colleges](https://en.wikipedia.org/wiki/College) and [universities](https://en.wikipedia.org/wiki/University) often have bookstores on campus that focus on providing course [textbooks](https://en.wikipedia.org/wiki/Textbook) and scholarly books and also sell other supplies.[2]

Learning from the above exquisite history regarding how its beginning of flourishment, Modern book-selling and store-keeping has changed dramatically with the advent of the computers. The best way to maintain, organize, and handle countless books systematically is to implement a book store management system application to ease the process. A book store management system can prove to be viable for said activities. It tracks the records of the books that include its author, price, distributor, sales being produced and overall maintenance records as well. We can find books in an instant that we have in our store, look up activities happening in our stores by managing all the data efficiently and orderly using this system. The purpose of a book store management system is to provide instant and accurate data, thereby saving a lot of time and effort.

**3.PROPOSED SYSTEM ARCHITECTURE**

The different blocks of the system architecture are explained below:

**3.1 Block Diagram or System Architecture**

User

Menu UI

Fig 3-1: Block Diagram of the Application

**3.2 Parts of the Program**

**3.2.1 Menu UI**

The home page of the application has the title “Book Store Management System” and the menu which shows login credentials for user to complete their respective job on the computer.

**3.2.1.1 Store Keeper Login**

The store keeper login menu requires the login credentials which should be filled up by the respective admin to do the store keeping activities. The store activities are to be decided according to the admin’s wish and necessities.

Admin

Product details( book’s name, author’s name and the price)

Exit

Figure 3-2 Store Activity Interface

**3.2.1.2 Exit**

It exits the program after the command from the user.

**3.3 Tools and Environment**

During the making of the whole project, many tools were used as listed below:

**3.3.1 Development Tools Used**

* **GitHub:** It is a [proprietary](https://en.wikipedia.org/wiki/Proprietary_software) developer platform that allows developers to create, store, manage, and share their code. It uses [Git](https://en.wikipedia.org/wiki/Git) to provide [distributed version control](https://en.wikipedia.org/wiki/Distributed_version_control) and GitHub itself provides [access control](https://en.wikipedia.org/wiki/Access_control), [bug tracking](https://en.wikipedia.org/wiki/Bug_tracking_system), [software feature](https://en.wikipedia.org/wiki/Software_feature) requests, [task management](https://en.wikipedia.org/wiki/Task_management), [continuous integration](https://en.wikipedia.org/wiki/Continuous_integration), and [wikis](https://en.wikipedia.org/wiki/Wiki) for every project.
* **Visual Studio Code**: It is an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) developed by [Microsoft](https://en.wikipedia.org/wiki/Microsoft) for [Windows](https://en.wikipedia.org/wiki/Windows), [Linux](https://en.wikipedia.org/wiki/Linux), [macOS](https://en.wikipedia.org/wiki/MacOS) and [web browsers](https://en.wikipedia.org/wiki/Web_browser).[[9]](https://en.wikipedia.org/wiki/Visual_Studio_Code#cite_note-TechCrunch-9)[[10]](https://en.wikipedia.org/wiki/Visual_Studio_Code#cite_note-10) Features include support for [debugging](https://en.wikipedia.org/wiki/Debugging), [syntax highlighting](https://en.wikipedia.org/wiki/Syntax_highlighting), [intelligent code completion](https://en.wikipedia.org/wiki/Intelligent_code_completion), [snippets](https://en.wikipedia.org/wiki/Snippet_(programming)), [code refactoring](https://en.wikipedia.org/wiki/Code_refactoring), and embedded [version control](https://en.wikipedia.org/wiki/Version_control) with [Git](https://en.wikipedia.org/wiki/Git). Users can change the [theme](https://en.wikipedia.org/wiki/Theme_(computing)), [keyboard shortcuts](https://en.wikipedia.org/wiki/Keyboard_shortcut), preferences, and install [extensions](https://en.wikipedia.org/wiki/Plug-in_(computing)) that add functionality.

**4. METHODOLOGY**

In our project we used various ways to gather our data and analyze the relevant topics covering the area of file handling, header files and arrays.

**4.1 Header Files and Namespaces**

Header files contains set of predefined library functions. In this project, we have used several header files as per convenience that are listed below:

1. string.h
2. stdio.h
3. stdlib.h
4. ctype.h
5. books.h

**4.2 Functions and Conditional Statements**

Functions are used in integral part of this project development. Most of the activity that are presented in the software heavily relies on functions and the inputs that are given from the user using this particular application.

Likewise, conditional statements are also profoundly used in this application. The required data which is an absolute must and conform to the title that is supposed to be stored is how we can store the necessary data. The application becomes efficient and correct as we can filter what actual data is presented in the input with the help of conditional statements.

1. **TIME ESTIMATION**

Table 5-1: Time Estimation Gantt Chart

**6. FEASIBILITY ANALYSIS**

The main purpose of our application is to provide efficiency in a low cost. Below are the aspects of the feasibility analysis.

**6.1 Usage**

We expect our project to be capable of following usages:

* Automate the store activities to ease the process resulting in increased efficiency, productivity regarding data storage, user experience and better resource management,
* Develop sustainable environment without the need of papers, fostering to clean environment,
* To help in maintaining records of all transactions, useful for auditing and accountability.

**6.2 Cost**

In terms of expenditure, due to compilers like GCC being free and opensource, most of

cost goes towards developing. But if complex system was to be integrated, adding those

features will definitely need much more investment.

**6.3 Hardware Requirement**

* Intel i3 2.8 Ghz Processor and above
* 4GB DDR3 RAM and above
* 125GB HDD hard disk and above

**6.4 Software Requirement**

* Windows OS(7 or higher)
* Intel HD graphics driver

**REFERENCES**

[1] <https://en.wikipedia.org/wiki/History_of_bookselling>

[2] <https://en.wikipedia.org/wiki/Bookselling>

[3] The C Programming Language Book by Brian Kernighan and Dennis Ritchie