**ABSTRACT**

The project is designed for developing the medical store management system whose purpose is to reduce the complexities of record keeping and documentation in the inventory management, payments and list of suppliers. This work discusses the scope, objective and aim of the project along with the analysis of the risks and constraints in the project. The entire project plan is also discussed in this work with the Gantt chart and WBS diagram. The staffing required for completing is project is also mentioned in this work. The estimated cost has also been provided in this report.

This mini project in C Medical Store Management System is a console application without graphic. In this project, you can add, modify and delete records of customers, suppliers and medicines. You can also search for customer or supplier details and medicines added into stock. Search can be done by medicine id, customer id or supplier name. The source code for this project is large, complete and totally error-free. It is compiled in DEV C++ IDE.

**CHAPTER 1**

**INTRODUCTION**

In the present time, the medical stores are looking for the services that are accurate and reliable for providing services to the customers and workers. Every store is making efforts to computerize their activities for providing better services to the customers. The medical store management system is the system used for medicine stock inventory in the medical stores. This system enables the manager of the store to record and manage all activities of the medical shop.

This system also enables the workforce of the medical store to offer their services in a manner which is more efficient and systematic which also improve of the medical store. this also helps in analyzing the performance of the store. the medical store management software can organize the daily activities in the medical store such as information of tablets, billing, details of stock and others. The medical store management software enables to maintain the details of purchase stock by the store and the details of the selling stock by customer wise (Hull, 2012).

This software generates the automated bill for every sale and also generate the reports for the customer detail, sell and stock. The use of this system reduces the time and effort involved in managing inventory of the medical store. It also reduces the hard work involved in using paper for record keeping. The use of this system enables the managers to easily record the details of their suppliers and assess them whenever required.

The use of medical store management system improves the efficiency of staff of the store as it provides information about the medicines and its location in the store. The medical store management system can give ease to the medical stores in managing their paper work, payment details and inventory online along with retrieving them.

**CHAPTER 2**

**AIM, OBJECTIVE AND SCOPE OF THE PROJECT**

**2.1 Aim of the project**

The medical store management system is a generic software which can be used in an array of the outlets that is wholesalers and retailers for automating process of the manual record keeping mainly relevant to the cash flows and stocks. The aim of this project is to design and develop the software which provide ease ad improvement from the manual inventory system used by the medical stores.

**2.2 Objective of the project**

The objective of this project is to organize the medical store details like stocks, medicines, expense, sales and others. The project s mainly developed at the administrative end and therefore it only provides access to the administrator of the store. The medical store management system is developed as application program for reducing manual work managing medical store, it also helps in keeping track of all the details regarding the expense, stocks and sales.

**The Medical Store Management System Also Has Following Objectives:**

* To provide searching facilities on the basis of various factors like stocks and theirs.
* To manage the user details for analyzing sales and expenses.
* To keep track and manage all information.
* To show the description and information of the stocks and others.
* To enhance the efficiency of management in medical store.
* To monitor the information regarding the transaction of the sales.
* To improve the function of edit, add and update the records for the proper resource management.

**2.3 Scope of this project**

The medical store management system helps in gathering the perfect management in the details. The collection and storage of data with the help of this project will be sensible, obvious and simple. It also helps in understanding the management of the previous time vividly. It also helps in performing the various functions of medical store. The medical store management system is developed to provide ease in documentation, keeping records, maintaining inventory and payment in the medical store which requires proper record keeping and management (Mahatme,et al., 2012). The project also helps the manager or administrator of the medical store to assess the stored information in the time of need such as for contacting the suppliers.

**The Scope for Developing the Medical Store Management System Are as Follows:**

* The detail of medicine which are available in medical store is easily managed and organized by using this system.
* The medical store management system also helps in keeping track of the available stock of medicine and also update them on a regular manner.
* It also helps in analyzing the location of the specific medicine available in the medical store.
* This project also enables to store the detail of suppliers and the stocks supplied by them.
* The medical store management system also provides the facility of generated automatic bills for the customers.
* The database for the customers and employees can also be managed by using this project.
* The designing of medical store management system saves the time which one spent in keeping records and managing payments of the stocks.
* The process of sailing medicine is very crucial as the medical stores must keep eye on the expiry date of every stock and medicine. This project helps the medical stores to keep track on the expiry date of every medicine they have in the store.
* After the designing and developing the software of medical store management system, it can be used by the medical stores for conducting, organizing and managing the different activities that are involved in keeping inventory, managing payments and others in the medical stores.

**CHAPTER 3**

**REQIREMENT SPECIFICATION**

**3.1 System Requirements**

The basic requirements for the development of this mini project are as follows

**3.1.1 Hardware Configuration**

1. Processor: Intel core i3 or above
2. Ram:512 MB
3. Hard disk:20 GB

**3.1.2 Software Configuration**

1. Language: C & C++
2. IDE: DEV C++
3. Documentation tool: Microsoft office 2003 or above

**3.2 Development Environment**

**3.2.1 C++**

**C++** is a general-purpose programming language that was developed as an enhancement of the C language to include object-oriented paradigm. It is an imperative and a **compiled** language[7].

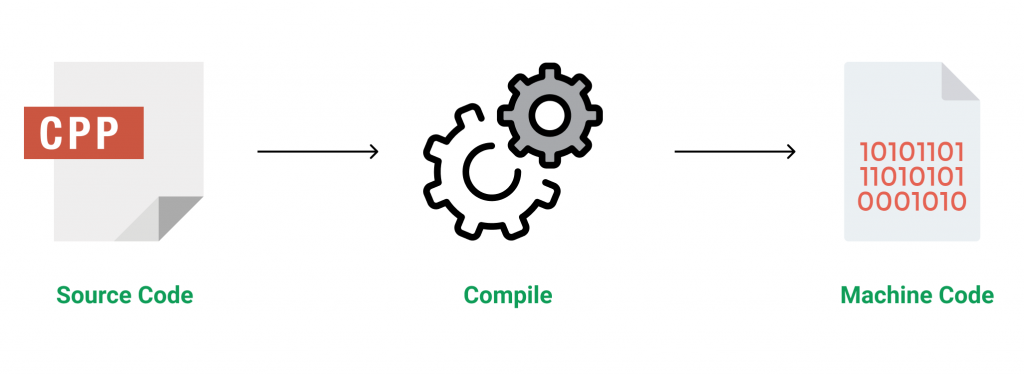


Figure 3.1 Source code compilation

C++ is a middle-level language rendering it the advantage of programming low-level (drivers, kernels) and even higher-level applications (games, GUI, desktop apps etc.). The basic syntax and code structure of both C and C++ are the same.

Some of the features & key-points to note about the programming language are as follows:

* Simple: It is a simple language in the sense that programs can be broken down into logical units and parts, has a rich library support and a variety of data-types.
* Machine Independent but Platform Dependent: A C++ executable is not platform-independent (compiled programs on Linux won’t run on Windows), however they are machine independent.
* Mid-level language: It is a mid-level language as we can do both systems-programming (drivers, kernels, networking etc.) and build large-scale user applications (Media Players, Photoshop, Game Engines etc.)
* Rich library support: Has a rich library support (Both standard ~ built-in data structures, algorithms etc.) as well 3rd party libraries (e.g. Boost libraries) for fast and rapid development.
* Speed of execution: C++ programs excel in execution speed. Since, it is a compiled language, and also hugely procedural. Newer languages have extra in-built default features such as garbage-collection, dynamic typing etc. which slow the execution of the program overall. Since there is no additional processing overhead like this in C++, it is blazing fast.
* Pointer and direct Memory-Access: C++ provides pointer support which aids users to directly manipulate storage address. This helps in doing low-level programming (where one might need to have explicit control on the storage of variables).
* Object-Oriented: One of the strongest points of the language which sets it apart from C. Object-Oriented support helps C++ to make maintainable and extensible programs. i.e. Large-scale applications can be built. Procedural code becomes difficult to maintain as code-size grows.
* Compiled Language: C++ is a compiled language, contributing to its speed.

**3.2.2 File Structure**

In computing, file system or filesystem (often abbreviated to fs) is a method and data structure that the operating system uses to control how data is stored and retrieved.[1] Without a file system, data placed in a storage medium would be one large body of data with no way to tell where one piece of data stopped and the next began, or where any piece of data was located when it was time to retrieve it [5]. By separating the data into pieces and giving each piece a name, the data is easily isolated and identified. Taking its name from the way a paper-based data management system is named, each group of data is called a "file”. The structure and logic rules used to manage the groups of data and their names is called a "file system”.

A file system consists of two or three layers. Sometimes the layers are explicitly separated, and sometimes the functions are combined. The logical file system is responsible for interaction with the user application. It provides the application program interface (API) for file operations — OPEN, CLOSE, READ, etc., and passes the requested operation to the layer below it for processing. The logical file system "manage[s] open file table entries and per-process file descriptors".[5] This layer provides "file access, directory operations, [and] security and protection".

The second optional layer is the virtual file system. "This interface allows support for multiple concurrent instances of physical file systems, each of which is called a file system implementation". The third layer is the physical file system. This layer is concerned with the physical operation of the storage device (e.g. disk). It processes physical blocks being read or written. It handles buffering and memory management and is responsible for the physical placement of blocks in specific locations on the storage medium. The physical file system interacts with the device drivers or with the channel to drive the storage device.

* + 1. **Files Operation**

A file is an abstract data type. To define a file properly, we need to consider the operations that can be performed on files. Six basic file operations. The OS can provide system calls to create, write, read, reposition, delete, and truncate files.

* Creating a file: Two steps are necessary to create a file.
  + Space in the file system must be found for the file.
  + An entry for the new file must be made in the directory
* Writing a file: To write a file, we make a system call specifying both the name of the file and the information to be written to the file. The system must keep a write pointer to the location in the file where the next write is to take place. The write pointer must be updated whenever a write occurs.
* Reading a file: To read from a file, we use a system call that specifies the name of the file and where (in memory) the next block of the file should be put. The system needs to keep a read pointer to the location in the file where the next read is to take place.
* Because a process is usually either reading from or writing to a file, the current operation location can be kept as a per-process current-file-position pointer.
* Both the read and write operations use this same pointer, saving space and reducing system complexity.
* Repositioning within a file: The directory is searched for the appropriate entry, and the current-file-position pointer is repositioned to a given value. Repositioning within a file need not involve any actual I/O. This file operation is also known as a file seek.
* Deleting a file: To delete a file, we search the directory for the named file. Having found the associated directory entry, we release all file space, so that it can be reused by other files, and erase the directory entry.
* Truncating a file: The user may want to erase the contents of a file but keep its attributes. Rather than forcing the user to delete the file and then recreate it, this function allows all attributes to remain unchanged (except for file length) but lets the file be reset to length zero and its file space released.

These six basic operations comprise the minimal set of required file operations. These primitive operations can then be combined to perform other file operations (i.e., copying). The OS keeps a small table, called the open-file table, containing information about all open files. When a file operation is requested, the file is specified via an index into this table, so no searching is required. When the file is no longer being actively used, it is closed by the process, and the OS removes its entry from the open-file table. Most systems require that the programmer open a file explicitly with the $open () system call before that file can be used. The $open () operation takes a file name and searches the directory, copying the directory entry into the open-file table.

This call can also accept access-mode information (create, read-only, read-write, append-only, and so on). This mode is checked against the file's permissions. If the request mode is allowed, the file is opened for the process. The $open () system call typically returns a pointer to the entry in the open-file table. This pointer, not the actual file name, is used in all I/O operations. The implementation of the $open () and $close () operations is more complicated in an environment where several processes may open the file at the same time. This may occur in a system where several different applications open the same file at the same time.

Typically, the OS uses two levels of internal tables:

* A per-process table. The per-process table tracks all files that a process has open. For instance, the current file pointer for each file is found here. Access rights to the file and accounting information can also be included.
* A system-wide table. Each entry in the per-process table in turn points to a system-wide open-file table. The system-wide table contains process-independent information, such as the location of the file on disk, access dates, and file size. Once a file has been opened by one process, the system-wide table includes an entry for the file.

Typically, the open-file table also has an open count associated with each file to indicate how many processes have e the file open. Each $close () decreases this open count, and when the open count reaches zero, the file is no longer in use, and the file's entry is removed from the open-file table. In summary, several pieces of information are associated with an open file.

**CHAPTER 4**

**SYSTEM IMPLEMENTATION**

This mini project is a very comprehensive one. File handling has been extensively and effectively used for almost major functions. The whole project is based on file handling as all medical records are stored in file. Data structure have been used to store and organize records. Overall, understanding this project will provide you valuable information on how to store, edit, search and delete data using file. Here, you can input many information like medicine Id, rack no., cabinet no., supplier’s name, unit cost, sales price, etc while adding a medicine into the store. You can also view information about report and billing. (Also check out customer billing system project.)

There are over 25 functions used in this mini project. I have divided those into the parent functions listed below to help you understand the project better.

1. Customer and supplier Id

int getcust\_id();

int getsupp\_id();

2. Welcome and main menu

void welcome();

void main\_menu();

3. All boxes

void main\_box();

void box1();

void wbox();

4. Bill slip

void bill();

5. About menu

void about();

6. Medicine menu

void medicine();

void medi\_sale();

void stock();

void update\_stock();

void medi\_entry();

void medi\_search();

void remainder();

7. Supplier menu

void supplier();

void supp\_entry();

void supp\_list();

void sup\_update();

void search();

void search\_id();

void search\_name();

8. Customer menu

void customer();

void cust\_search();

void search\_cid();

void search\_cname();

void cust\_entry();

void cust\_list();

void cust\_update();

9. Report menu

void report\_menu();

void report();

void sale\_rpt();

void sale\_rpt\_daily();

void profit\_rpt();

void pur\_rpt();

void pur\_rpt\_daily();

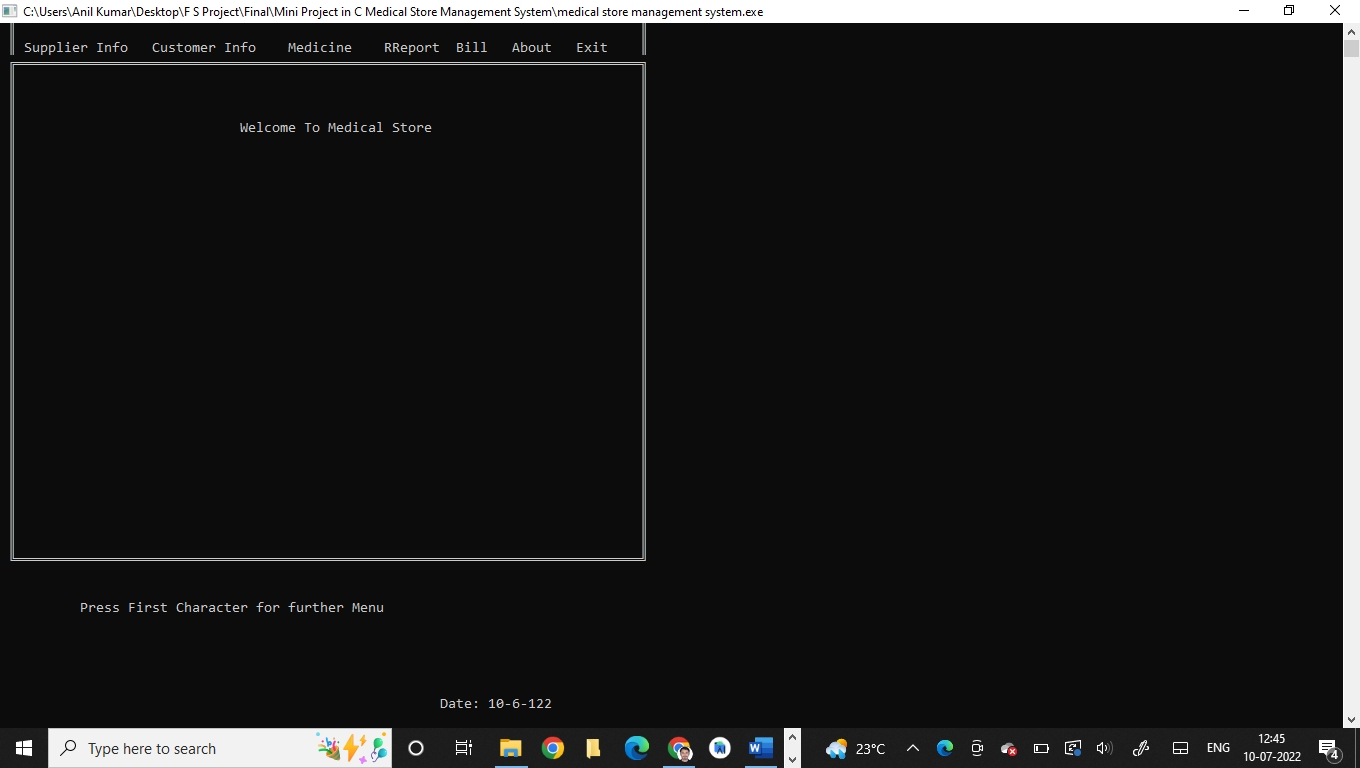
void gotoxy (int x, int y) – I have been describing this function in every C mini project published on this site. You need to understand this function as it is one of the most important one used in Medical Store Management System Project.

Gotoxy function allows you to print text in any place of screen. Using this function in Code::Blocks requires coding, but it can be directly used in Turbo C. Here is a code for this function in Code::Blocks.

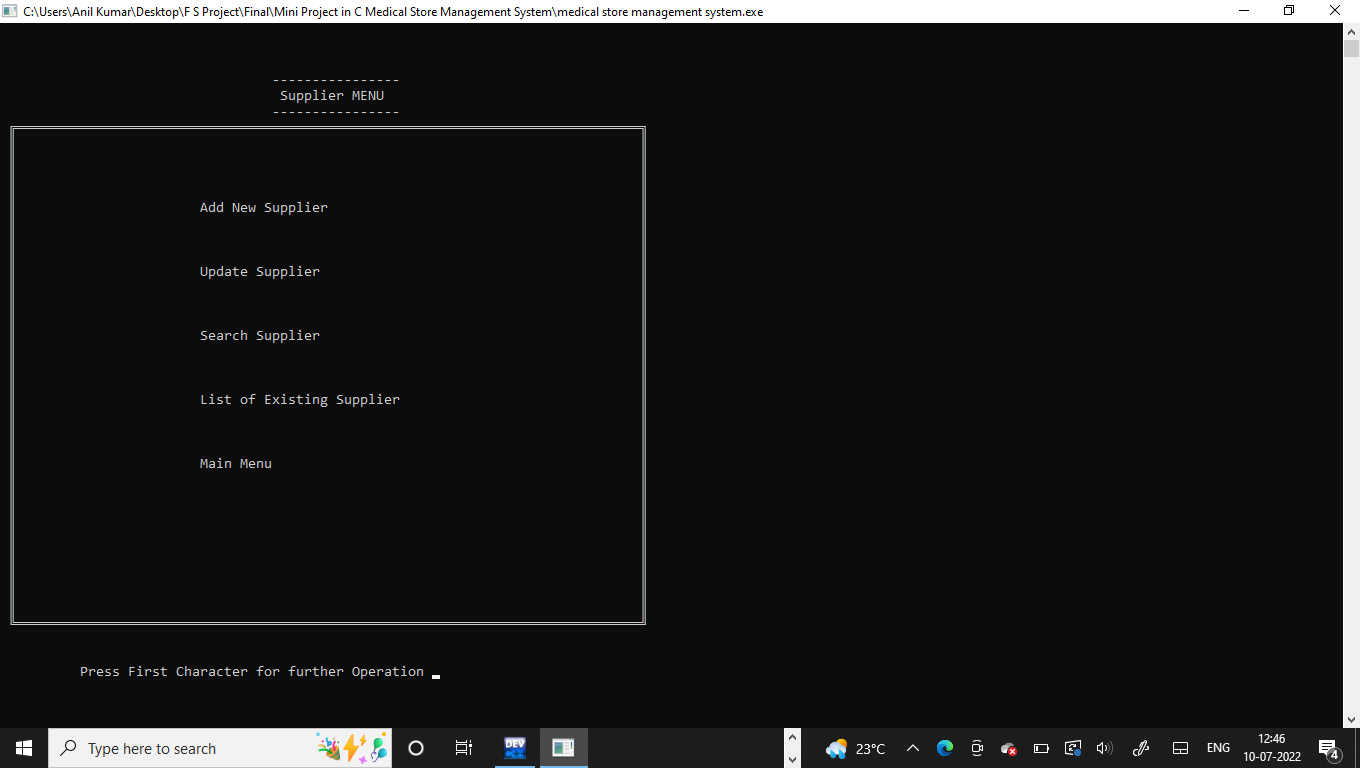
**CHAPTER 5**

**SAMPLE OUTPUT**

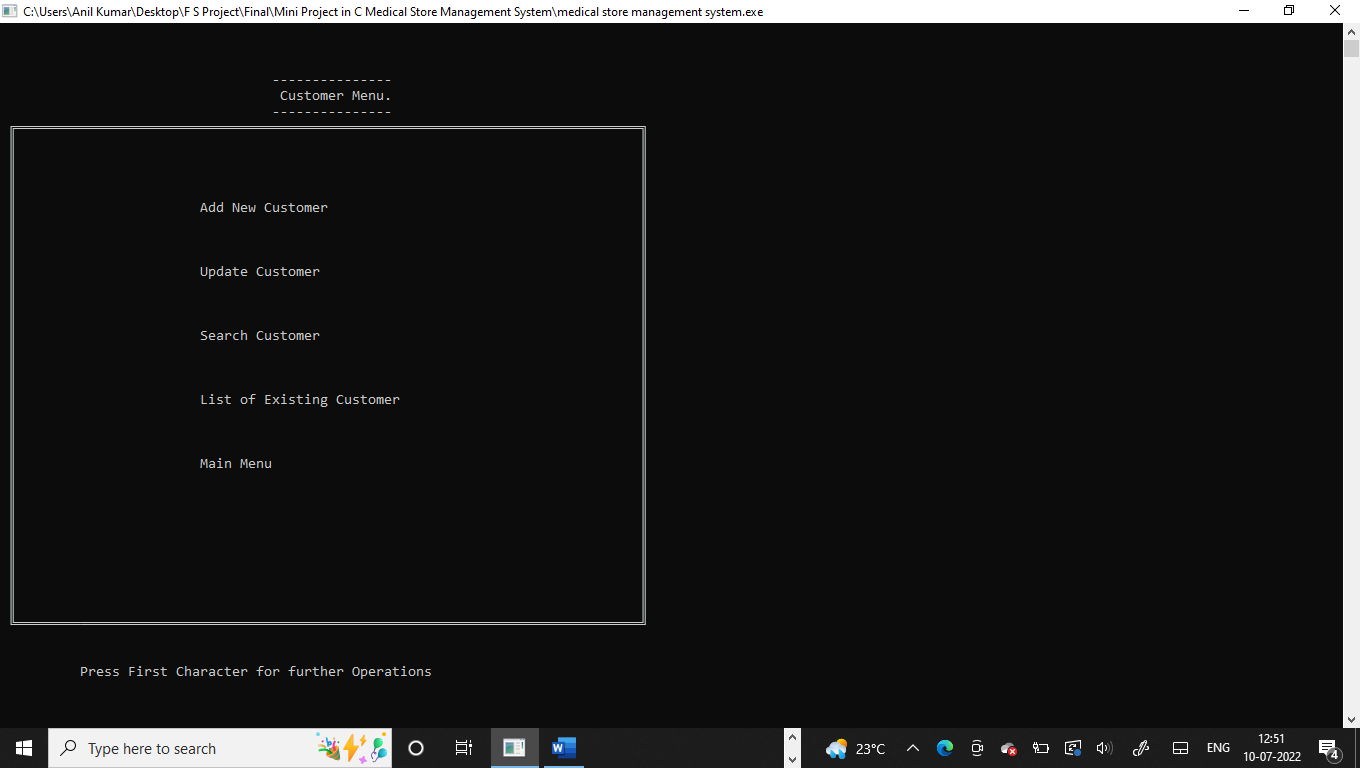
**5.1 Screenshots**



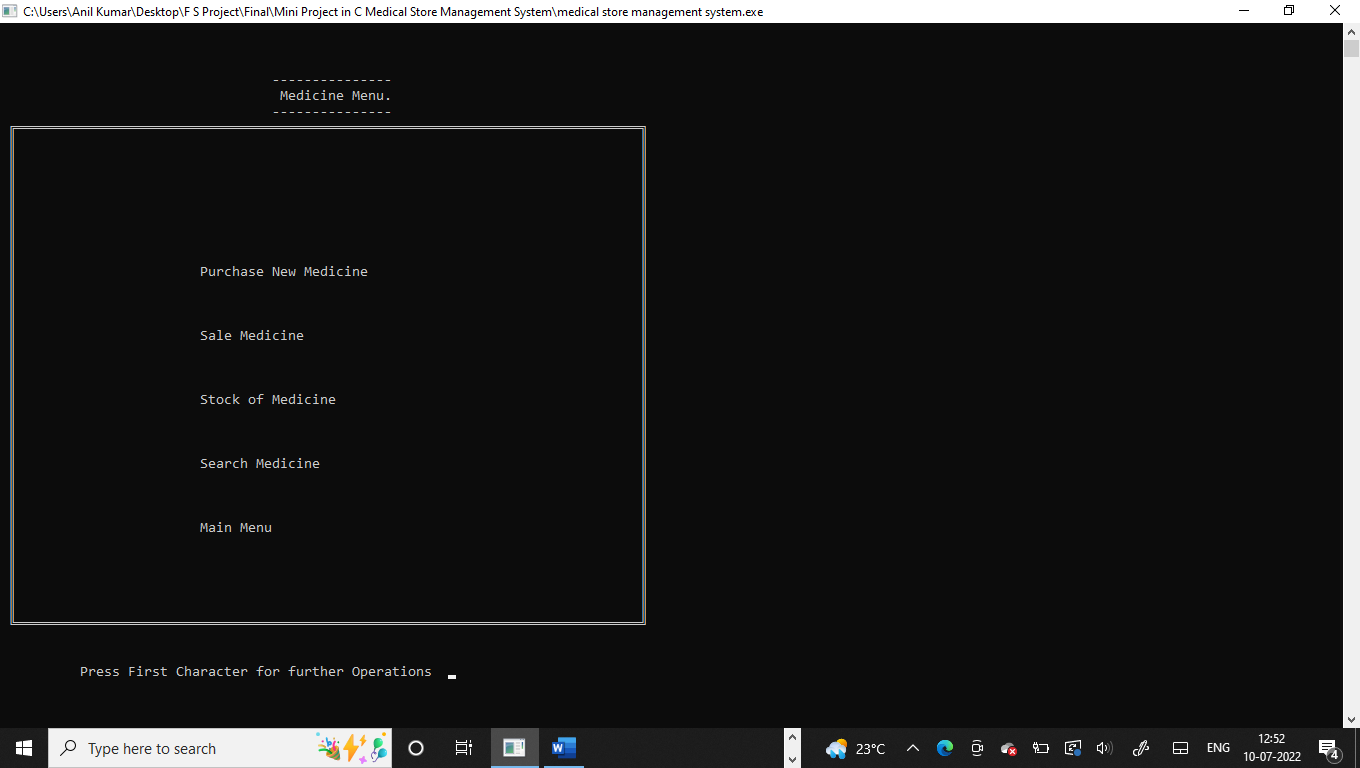
**Figure 5.1 Home Page**



**Figure 5.2 Supplier Menu**



**Figure 5.3 Customer Menu**



**Figure 5.4 Medicine Menu**

**Chapter 6**

**CONCLUSION**

To conclude the description about the project, this was developed using XAMPP Server with PHP & MySQL is based on the requirement specification of the users and analysis of the existing system, with flexibility for future enhancement. Newspaper Management System is very useful for agency managers. This software is designed for people who want to keep the record of the newspapers that are circulated in the city and it also helps the agency people to have an exact count of their paperboys and their distribution ratios. This also helps in calculating the price that must be paid by each customer.

**BIBLIOGRAPHY**

[1] https://www.codewithc.com/mini-project-in-c-medical-store-management-system/

[2] https://www.codewithc.com

[3] [www.youtube.com](http://www.youtube.com)

[4] [www.quora.com](http://www.quora.com)

[5] [www.wikipedia.org](http://www.wikipedia.org)

[6] <https://www.w3schools.com/cpp/cpp_files.asp>

[7] https://www.geeksforgeeks.org/introduction-to-c-programming-language/