Appendix 1

Medical Inventory Management System

End Term Project

By

Jatin Agrawal, Neeraj Keerthi

Section: K19QW

RollNumbers: 51, 54

Department of Intelligent Systems,

School of Computer Science Engineering,

Lovely Professional University, Jalandhar

November, 2020

Appendix 2

Student Declaration

This is to declare that this report has been written by me/us. No part of the report is copied from other sources. All information included from other sources have been duly acknowledged. I/We aver that if any part of the report is found to be copied, I/we are shall take full responsibility for it.

Signature: Jatin, Neeraj
Name: Jatin Agrawal, Neeraj Keerthi
Roll Number: 51, 54

Place: Udaipur (Raj)

Date: 28 Oct 2020

Appendix 3

TABLE OF CONTENTS

Appendix 4

BONAFIDE CERTIFICATE

Certified that this project report “Medical Inventory Management System” is the bonafide work of “JATIN AGRAWAL, NEERAJ KEERTHI” who carried out the project work under my supervision.

<<Signature of the Supervisor>>(Due to Covid 19, signature is exempted)
Dr. Dhanpratap Singh
Assistance professor
24706
CSE

BACKGROUND AND OBJECTIVE OF PROJECT

objectives

The main motive is to develop the software that covers all
the aspects of management and operations of a medical inventory. It enables healthcare providers to improve operational effectiveness, reduce costs and reduce time consumption thus enhancing the quality of service

The medical inventory system in Python helps the user to maintain medicinal shop record. This project mainly focuses on CRUD with search functionality. The design is so simple that the database manager won't find any difficulties while working on it.

This is a command line interface

Background:

It focuses to make the life of a person managing a medical shop easier, he will be able to input all the data about the medical inventory in to the database and can manage the details of the inventory using the database.
He can manage the stock available in the inventory, see what all medicines are actually available to buy or to sell, how many suppliers are available and what medicines they supply to us and he can also keep track of the medicines bought from the suppliers and the medicines sold to the customers.
Before using this kind of a database, other alternatives could be to write it all in a register and keep a log of it, but when it grows larger in scale and there are more medicines in stock, more customers, more suppliers, then this becomes a hassle and very hard to tally. And thus using this project, where we can store all these details in a virtual environment. This makes searching for a particular item very easy. Here inputing new information, removing information and editing information is very easy and hence it is a much more efficient way to manage a medical store.

Outcome:

Medical stores would increase their effectiveness in delivering the requirements of the patients and be able to run the store without any interruptions.

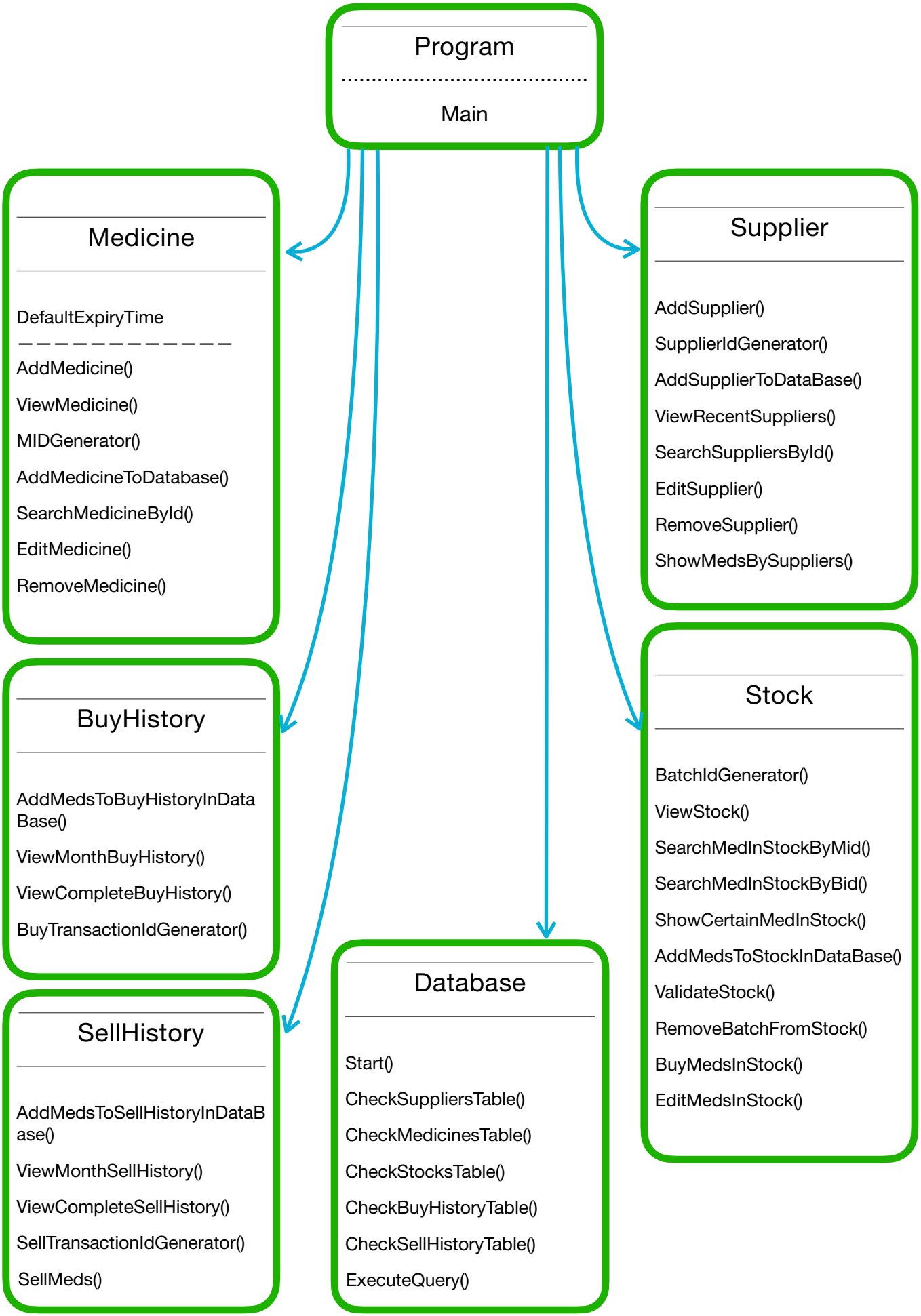
They will be ahead of time and can request stock supply from suppliers as per the stock in the inventory

whenever the stock is sold to the customers or stock is bought from the suppliers, sell history and buy history is automatically updated which helps them to keep track of the stock and the finances of the medical store

Goals:

The goals are to be able to store the details of the inventory like the stock available, the suppliers- which supplier supplies which medicine,
details of all the medicines,
keeping a track of buy history and sell history when the store buys medicines from the database and when they sell it to customers, the quantity bought or sold and the date when they were bought or sold.

We should be able to add medicines into the database, edit them or remove them,
we should be able to add corresponding suppliers of those medicines to the database and edit or remove them as well.
We should be able to store the details of all the stock of the medicines that are available in our inventory.
We should be able to sell medicines to customers and buy medicines from suppliers, and add this to sell history and buy history of our inventory and updating the stock as well.


Program
Main
Medicine
DefaultExpiryTime
AddMedicine()
ViewMedicine()
MIDGenerator()
AddMedicineToDatabase()
SearchMedicineById()
EditMedicine()
RemoveMedicine()
Supplier
AddSupplier()
SupplierIdGenerator()
AddSupplierToDataBase()
ViewRecentSuppliers()
SearchSuppliersById()
EditSupplier()
RemoveSupplier()
ShowMedsBySuppliers()
BuyHistory
AddMedsToBuyHistoryInDataBase()
ViewMonthBuyHistory()
ViewCompleteBuyHistory()
BuyTransactionIdGenerator()
Stock
BatchIdGenerator()
ViewStock()
SearchMedInStockByMid()
SearchMedInStockByBid()
ShowCertainMedInStock()
AddMedsToStockInDataBase()
ValidateStock()
RemoveBatchFromStock()
BuyMedsInStock()
EditMedsInStock()
SellHistory
AddMedsToSellHistoryInDataBase()
ViewMonthSellHistory()
ViewCompleteSellHistory()
SellTransactionIdGenerator()
SellMeds()
Database
Start()
CheckSuppliersTable()
CheckMedicinesTable()
CheckStocksTable()
CheckBuyHistoryTable()
CheckSellHistoryTable()
ExecuteQuery()

Description of methods

Supplier

ADDSUPPLIER()
Parameters:
None
Description:
Method asks user to input data and calls AddSupplierToDataBase to save data
Return:
null

SUPPLIERIDGENERATOR()
Parameters:
None
Description:
Will generate a new supplier id based on latest SID in data base.
Return: new integer supplier id

ADDSUPPLIERTODATABASE()
Parameters:
Sid: supplier id
Name: supplier name
Email: supplier email
Address: supplier address
Phone: supplier contact number
Description:
Will add the details of supplier to data base
Return:
null

VIEWRECENTSUPPLIERS()
Parameters:
Bool
False: view all
True: view recent 5
Description:
Method to get list of rows of suppliers data from database
Return:
List of rows

SEARCHSUPPLIERBYID()
Parameters:
Int SID : supplier id
Description:
Searches for supplier based on given id in database and returns details if found or empthy kist is not found
Return:
List of details

EDITSUPPLIER()
Parameters:
None
Description:
Method lets user edit details of supplier
Return:
null

REMOVESUPPLIER()
Parameters:
None
Description:
Method ask user for a SID and deletes the supplier with that SID from database
Return:
null

SHOWMEDSBYSUPPLIER()
Parameters:
None
Description:
Shows medicines supplied by all suppliers
Return:
none

Medicine

ADDMEDICINE()
Parameters:
None
Description:
Method asks user for details of medicine and calls AddMedicineToDataBase to store data
Return:
null

VIEWMEDICINE()
Parameters:
None
Description:
Retrieves rows of medicines data form database
Return:
List of rows of medicine data

MIDGENERATOR()
Parameters:
None
Description:
Method generates MID based on latest medicine id in database
Return:
Int New Medicine Id

ADDMEDICINETODATABASE()
Parameters:
Mid: medicine id
Name: medicine name
Price: medicine price
Minreq: medicine minimum requirements in stock
Sid: id of supplier supplying this medicine
Description:
Method adds the given medicine to database
Return:
null

SEARCHMEDICINEBYID()
Parameters:
Mid: medicine id
Description:
Retrieves the details of medicine by given mid from database
Return:
List of details of medicine

EDITMEDICINE()
Parameters:
None
Description:
Method lets user edit medicine's details
Return:
null

REMOVEMEDICINE()
Parameters:
None
Description:
Method ask user for an mid and deletes from database
Return:
null

Stock

BATCHIDGENERATOR()
Parameters:
None
Description:
Method generates a new BID based on latest stock BID
Return:
Int new batch id

VIEWSTOCK()
Parameters:
None
Description:
Method returns list of rows of batches in stock database
Return:
List of of batches

SEARCHMEDICINEINSTOCKBYID()
Parameters:
Mid: medicine id
Description:
Method retrieves details of medicine by mid from database
Return:
List of details of medicine

SEARCHMEDINSTOCKBYBID()
Parameters:
Bid: batch id
Description:
Method retrieves details of medicine by bid from database
Return:
List of details of medicine

ADDMEDSTOSTOCKINDATABASE()
Parameters:
Bid: batch id
Mid: medicine id
Amount: medicine amount
Description:
Method adds a new batch of medicine to stock
Return:
null

VALIDATESTOCK()
Parameters:
None
Description:
Method deletes expired and empty stocks form database
Return:
null

REMOVEBATCHFROMSTOCK()
Parameters:
None
Description:
Method asks user for bid and deletes that batch from database
Return:
null

BUYMEDSINSTOCK()
Parameters:
None
Description:
Method lets user choose from available known medicines and take details and add it to stock and buy history
Return:
null

EDITMEDSINSTOCK()
Parameters:
None
Description:
Method lets user edit details of stock by asking for bid
Return:
null

BuyHistory

ADDMEDSTOBUYHISTORYINDATABASE()
Parameters:
Bid: batch id
Mid: medicine id
Amount: amount of medicines
Description:
Method generates expiry date and total price and adds all details to database
Return:
null

VIEWMONTHBUYHISTORY()
Parameters:
None
Description:
Method retrieves rows of buy history for last 1 month
Return:
List of purchases this month

VIEWCOMPLETEBUYHISTORY()
Parameters:
None
Description:
Method retrieves rows of complete buy history
Return:
List of purchases

BUYTRANSACTIONIDGENERATOR()
Parameters:
None
Description:
Method generates a new transaction id based on latest transaction id in database
Return:
Int new transaction id

SellHistory

ADDMEDSTOSELLHISTORYINDATABASE()
Parameters:
Mid: medicine id
Amount: medium sold amount
CPhone: customer's phone number
CName: custormer's name
Description:
Adds details of transaction in database with transaction id and date
Return:
null

VIEWMONTHSELLHISTORY()
Parameters:
None
Description:
Method Retrieves rows of sell history data form database for last 1 month
Return:
List of sell history data for last month

VIEWCOMPLETESELLHISTORY()
Parameters:
None
Description:
Method retrieves rows of complete sell history data
Return:
List of sell history data

SELLTRANSACTIONIDGENERATOR()
Parameters:
None
Description:
Method generates new transaction id based on latest transaction id in database
Return:
Int new transaction id

SELLMEDS()
Parameters:
None
Description:
Method asks user for details and reduces sold meds from stock and saves transaction
Return:
null

Databases

START()
Parameters:
None
Description:
Initialises the database
Return:
null

CHECKSUPPLIERSTABLE()
Parameters:
None
Description:
Method checks if table is present if not it creates new one
Return:
null

CHECKMEDICINESTABLE()
Parameters:
None
Description:
Method checks if table is present if not it creates new one
Return:
null

CHECKSTOCKSTABLE()
Parameters:
None
Description:
Method checks if table is present if not it creates new one
Return:
null

CHECKBUYHISTORYTABLE()
Parameters:
None
Description:
Method checks if table is present if not it creates new one
Return:
null

CHECKSELLHISTORYTABLE()
Parameters:
None
Description:
Method checks if table is present if not it creates new one
Return:
null

EXECUTEQUERY()
Parameters:
Query: sql query to be executed
Description:
Method connects to database and executes the query
Return:
Returns data from query (ex: from SELECT)

DESCRIPTION OF WORK DEVISION AMONG STUDENTS

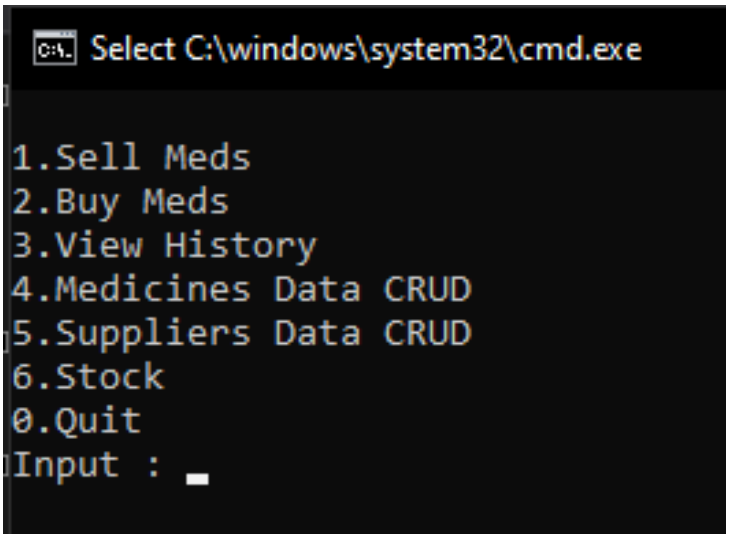
Member 1: Jatin Agrawal (Roll No. 51)

- Worked on initial planning and flow of project
- Worked on Suppliers, Stocks, BuyHistory, SellHistory, Databases
- Modified Program file for cleanliness.
- Worked on Final testing and changes.
- Uploaded Project to Github.
- Worked on Project file.

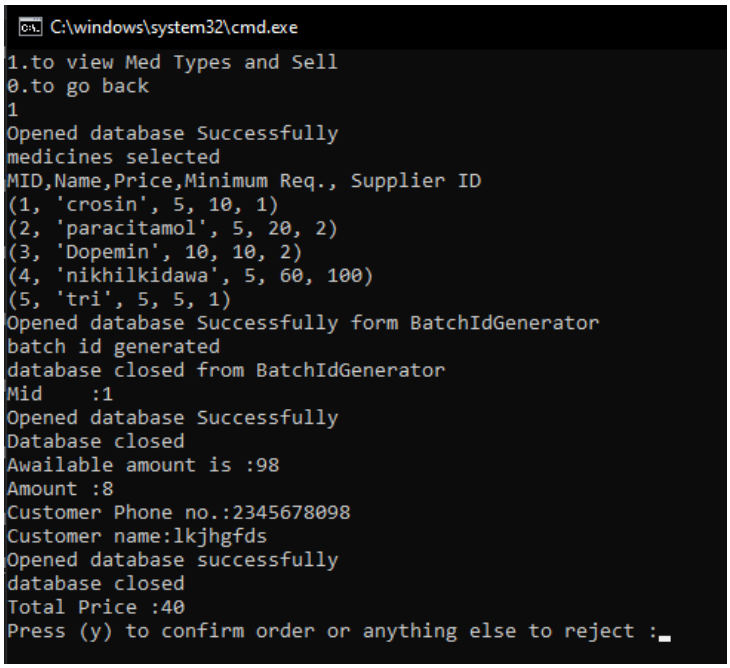
Member 2: Neeraj Keerthi (Roll No. 54)

- Helped in initial planning of project.
- Worked on Medicines and Program class.
- Made Input validation method (Later removed because of certain issues).
- Worked on Background and Objectives of project.
- Completed SWOT Analysis of Project.

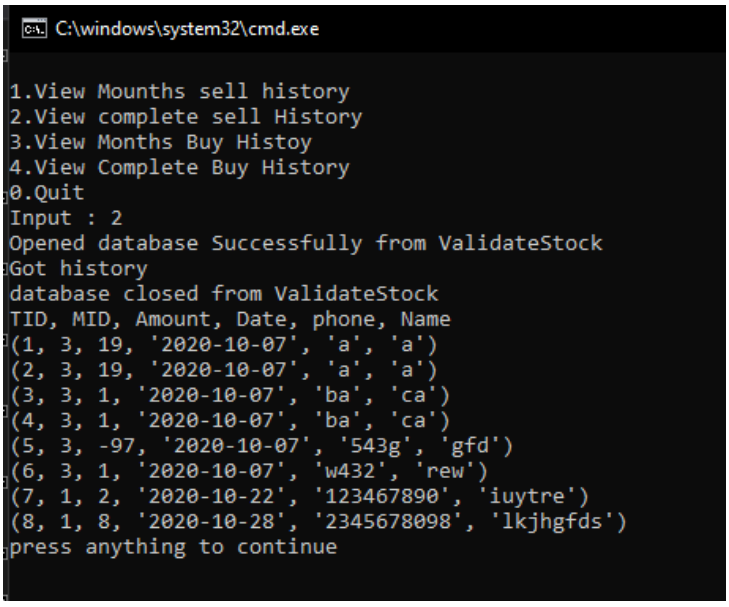
Screenshots



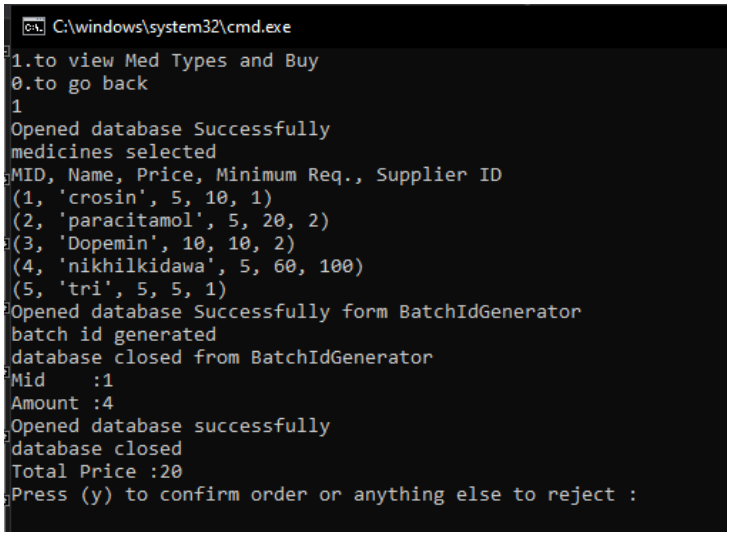
Front page



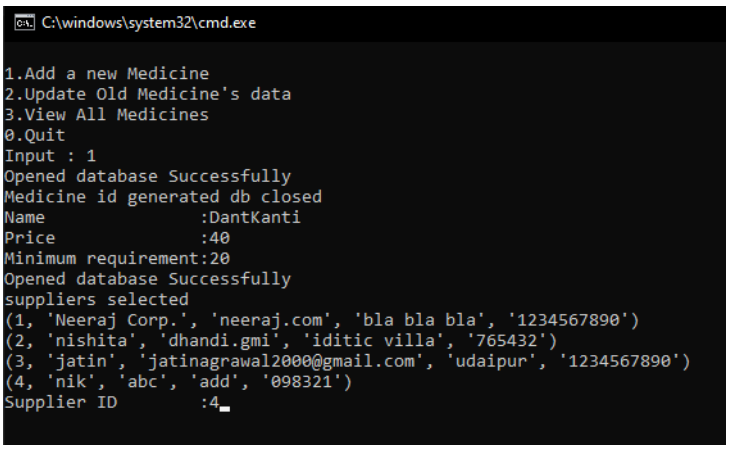
Sell Meds



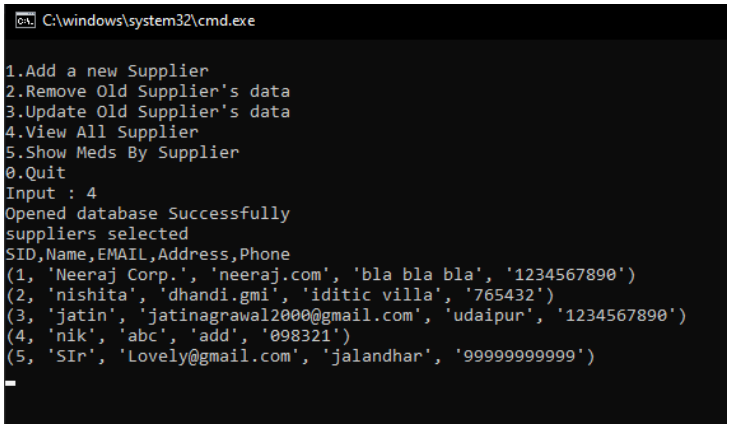
View History



Buy Meds



Medicine CRUD



Supplier CRUD

Technologies And Framework Used

Python:

The platforms which were used by us in creating the project were, Python3 which was used to manage the database and create the command line user interface between the user and the database.

Python is an interpreted, high-level and general-purpose programming language. Created by Guido Van Rossum and first released in 1991, Python's design philosophy emphasises code readability with its notable use of significant whitespace.

Sqlite3:

We used sqlite3 to create our database and use the required properties of the database.

SQLite3 is a relational database management system contained in a Python library. In contrast to many other database management systems, SQLite3 is not a client–server database engine. Rather, it is embedded into the end program.

Visual Studio Code:

we used vscode IDE as our primary code editor where we wrote all the code required for the project.

Visual Studio Code is a free source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.

sqliteonline.com

we also used a website https://sqliteonline.com/ where we verified all the code to use the database efficiently.

“The lectures given to us by our teachers also helped us a lot in completing the project.“

SWOT Analysis of the project

Strengths:

This project allows Medical Shop to have a cleaner medical Inventory Data Management System.

It allows them to Add medicines to dictionary.

It allows them to set up suppliers.

It allows them to log their purchases and sales.

It allows them to manage expired medicines with ease.

It allows them to have a look at what is in inventory with just a few clicks.

If database is deleted it can create a new one rather than breaking the whole application.

Weaknesses:

Its cross-platform efficiency decreases, it is most efficient in windows but in others it is not so efficient.

Opportunities:

we can make this project even more attractive by adding GUI to the userinterface instead of command based interface.

we can add even more functional capabilities so that the program itself will be able to notify the suppliers when the stock drops below a certain limit.

Threats:

This project is not immune to sql injection and sql injection can harm database.

PROJECT GITHUB LINK

https://github.com/jtagrawal2000/MedicalInventoryManagementSystem