Minor Project Report

On

"Medical Shop Management System"

Submitted in partial fulfillment for the award of degree of

BACHELOR OF TECNOLOGY
In
COMPUTER SCIENCE & ENGINEERING

Submitted to :
MR. Mahesh Joshi

Submitted by :
Vinod Vedwal(16EGJCS400)
Yash Suthar (16EGJCS301)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

GLOBAL INSTITUTE OF TECHNOLOGY, JAIPUR
(RAJASTHAN)-302022

CANDIDATE'S DECLARATION

I hereby declare that the work, which is being presented in the minor project, entitled "**Medical Shop Management System**" in partial fulfillment for the award of Degree of "Bachelor of Technology" in Department of Computer Science and Engineering submitted to the **Department of Computer Science & Engineering**, Global Institute of Technology, Rajasthan Technical University is a record of our own investigations carried under the Guidance of **Ms. Neel Kamal Choudhary** , Department of Computer Science and Engineering, Global Institute of Technology.

I have not submitted the matter presented in this project anywhere for the award of any other Degree.

Vinod Vedwal(16EGJCS400)

Yash Suthar(16EGJCS301)

Computer Science & Engineering

Global Institute of Technology, Jaipur

Counter Signed by

Mahesh Joshi

Assistant Professor

Department of Computer Science and Engineering

Global Institute of Technology, Jaipur

ACKNOWLEDGEMENT

I would like to articulate our deep gratitude to our project guide **Ms. Neel Kamal Choudhary** Ma'am for her guidance, advice and constant support in the project work. I would like to thank her specially for being our advisor here at Global Institute of Technology, Jaipur. I would like to thank all faculty members and staff of the Department of Computer Science and Engineering for their generous help in various ways for this project.

Last but not the least; I give my sincere thanks to all of mine friends who have patiently extended all sorts of help in this project.

ABSTRACT

The project titled is Medical shop management system for monitoring and controlling the system at medical shop. The project “Medical Shop Management System” is developed in python, which mainly focuses on basic operations in a shop like checking for the available stock, adding details of new stock and updating new information, deleting information. It features a familiar and well thought-out, an attractive user interface combined with strong searching insertion and reporting capabilities. The backend of the project designed using MySQL.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

Medical shop management requires user entry of container and product dimensions, together with information on weight and orientation constraints

It implements the storage and retrieval system of day-to-day activities which involves daily transaction report, monthly wise report on goods delivered, enquires which enables us to provide efficient and accurate methods of organizing and accessing different types of information.

In-built database facilities for up to 50 containers and 4000 products are provided so that input into these screens can be carried out with minimal effort. If information for a particular product code is already held in the 4000 product database then this is automatically entered into the appropriate fields.

The first option will attempt to pack as much of the medicines using any of the packing methods available to medical shop management. This may be a loading from the floor or from the end of the container. The second and third options are self-explanatory and the one most appropriate to the practical circumstances should be selected.

CHEPTER- 2

Introduction to Python and Libraries

2.1 Python

Python is one of those rare languages which can claim to be both simple and powerful. You will find yourself pleasantly surprised to see how easy it is to concentrate on the solution to the problem rather than the syntax and structure of the language you are programming in.

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

2.2 History

Python was conceived in the late 1980s by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC language (itself inspired by SETL), capable of exception handling and interfacing with the Amoeba operating system. Its implementation began in December 1989. Van Rossum shouldered sole responsibility for the project, as the lead developer, until July 12, 2018, when he announced his "permanent vacation" from his responsibilities as Python's *Benevolent Dictator For Life*, a title the Python community bestowed upon him to reflect his long-term commitment as the project's chief decision-maker. He now shares his leadership as a member of a five-person steering council. In January, 2019, active Python core developers elected Brett Cannon, Nick Coghlan, Barry Warsaw, Carol Willing and Van Rossum to a five-member "Steering Council" to lead the project.

Python 2.0 was released on 16 October 2000 with many major new features, including a cycle-detecting garbage collector and support for Unicode.

Python 3.0 was released on 3 December 2008. It was a major revision of the language that is not completely backward-compatible. Many of its major features were backported to Python 2.6.x and 2.7.x version series. Releases of Python 3 include the 2to3 utility, which automates (at least partially) the translation of Python 2 code to Python 3.

Python 2.7's end-of-life date was initially set at 2015 then postponed to 2020 out of concern that a large body of existing code could not easily be forward-ported to Python 3.

2.3 Features in Python

There are many features in Python, some of which are discussed below –

1. Easy to code:

Python is high level programming language. Python is very easy to learn language as compared to other language like c, c#, java script, java etc. It is very easy to code in python language and anybody can learn python basic in few hours or days. It is also developer-friendly language.

2. Free and Open Source:

Python language is freely available at official website and you can download it from the given download link below click on the **Download Python** keyword. Since, it is open-source; this means that source code is also available to the public. So you can download it as, use it as well as share it.

3. Object-Oriented Language:

One of the key features of python is Object-Oriented programming. Python supports object oriented language and concepts of classes, objects encapsulation etc.

4. GUI Programming Support:

Graphical Users interfaces can be made using a module such as PyQt5, PyQt4, wxPython or Tk in python. PyQt5 is the most popular option for creating graphical apps with Python.

5. High-Level Language:

Python is a high-level language. When we write programs in python, we do not need to remember the system architecture, nor do we need to manage the memory.

6. Extensible feature:

Python is a **Extensible** language. We can write our some python code into c or c++ language and also we can compile that code in c/c++ language.

7. Python is Portable language:

Python language is also a portable language. For example, if we have python code for windows and if we want to run this code on other platform such as Linux, Unix and Mac then we do not need to change it, we can run this code on any platform.

8. Python is integrated language:

Python is also an Integrated language because we can easily integrated python with other language like c, c++ etc.

9. Interpreted Language:

Python is an Interpreted Language. Because python code is executed line by line at a time. like other language c, c++, java etc there is no need to compile python code this makes it easier to debug our code. The source code of python is converted into an immediate form called **bytecode**.

10. Large Standard Library

Python has a large standard library which provides rich set of module and functions so you do not have to write your own code for every single thing.There are many libraries present in python for such as regular expressions, unit-testing, web browsers etc.

11. Dynamically Typed Language:

Python is dynamically-typed language. That means the type (for example- int, double, long etc) for a variable is decided at run time not in advance.because of this feature we don't need to specify the type of variable.

2.4 Reasons Why Python is Best-Suited for Machine Learning?

Python is currently the most popular programming language for research and development in Machine Learning. But you don't need to take my word for it! According to **Google Trends**, the interest in Python for Machine Learning has spiked to an all-new high with other ML languages such as R, Java, Scala, Julia, etc. lagging far behind.

So now that we have established that Python is by far the most popular programming language for Machine Learning, the **WHY** still remains. So let's now understand why Python is so popular and consequently why it is best-suited for ML. Some of these reasons for this are given as follows:

1. Python is Easy To Use

Nobody likes excessively complicated things and so the ease of using Python is one of the main reasons why it is so popular for Machine Learning. It is **simple** with an **easily readable syntax** and that makes it well-loved by both seasoned developers and experimental students. The simplicity of Python means that developers can focus on actually solving the Machine Learning problem rather than spend all their time (and energy!) understanding just the technical nuances of the language.

In addition to this, Python is also **supremely efficient**. It allows developers to complete more work using fewer lines of code. The Python code is also easily understandable by humans, which

makes it ideal for making Machine Learning models. With all these advantages, what's not to love?!!

2. Python has multiple Libraries and Frameworks

Python is already quite popular and consequently, it has hundreds of different libraries and frameworks that can be used by developers. These libraries and frameworks are really useful in saving time which in turn makes Python even more popular (That's a beneficial cycle!!!).

There are many Python libraries that are specifically useful for Artificial Intelligence and Machine Learning. Some of these are given below:

Keras is an open-source library that is particularly focused on experimentation with deep neural networks.

TensorFlow is a free software library that is used for many machine learning applications like neural networks. (They seem to be quite popular!)

Scikit-learn is a free software library for Machine Learning that various classification, regression and clustering algorithms related to this. Also, Scikit-learn can be used in conjugation with NumPy and SciPy.

3. Python has Community and Corporate Support

Python has been around since 1990 and that is ample time to create a **supportive community**. Because of this support, Python learners can easily improve their Machine Learning knowledge, which only leads to increasing popularity. And that's not all! There are many resources available online to promote ML in Python, ranging from Geeks for Geeks Machine Learning tutorials to YouTube tutorials that are a big help for learners.

Also, **Corporate support** is a very important part of the success of Python for ML. Many top companies such as Google, Facebook, Instagram, Netflix, Quora, etc use Python for their

products. In fact, Google is single-handedly responsible for creating many of the Python libraries for Machine Learning such as Keras, TensorFlow, etc.

4. Python is Portable and Extensible

This is an important reason why Python is so popular in Machine Learning. A lot of cross-language operations can be performed easily on Python because of its **portable and extensible nature**. There are many data scientists who prefer using Graphics Processing Units (GPUs) for training their ML models on their own machines and the portable nature of Python is well suited for this.

Also, many different platforms support Python such as Windows, Macintosh, Linux, Solaris, etc. In addition to this, Python can also be integrated with Java, .NET components or C/C++ libraries because of its extensible nature.

2.5 Libraries

Python's large standard library, commonly cited as one of its greatest strengths, provides tools suited to many tasks. For Internet-facing applications, many standard formats and protocols such as MIME and HTTP are supported. It includes modules for creating graphical user interfaces, connecting to relational databases, generating pseudorandom numbers, arithmetic with arbitrary-precision decimals, manipulating regular expressions, and unit testing.

Some parts of the standard library are covered by specifications (for example, the Web Server Gateway Interface (WSGI) implementation wsgiref follows PEP 333), but most modules are not. They are specified by their code, internal documentation, and test suites (if supplied). However, because most of the standard library is cross-platform Python code, only a few modules need altering or rewriting for variant implementations.

As of November 2019, the Python Package Index (PyPI), the official repository for third-party Python software, contains over 200,000 packages with a wide range of functionality, including:

- Graphical user interfaces
- Web frameworks

- Multimedia
- Databases
- Networking
- Test frameworks
- Automation
- Web scraping
- Documentation
- System administration
- Scientific computing
- Text processing
- Image processing
- Machine learning
- Data analytics

CHAPTER 3

SYSTEM ANALYSIS

Analysis may be defined as a process of dividing the problems into parts, identifying each part and establishing relationship in the parts. Analysis is a detailed study of the various operations performed by a system and the relationships within and outside of the system.

Analysis is a continuing activity at all stages of the project. It is the process of studying problems. To find the best solution to the problem, by which the existing system is learnt, existing problems are understood .Objectives and requirements are defined and the solutions are evaluated. Once analysis is completed, the analyst has a firm understanding of what is to be done.

Analysis consists of two sub phases: Planning and Requirements Definition. They include understanding the customer's problem, performing a feasibility study and developing a recommended solution strategy, determining the acceptance criteria, and planning the development product.

The products of planning a System Definition and Project Plan. The system definition is typically expressed in English or some other natural language, and may incorporate charts, figures, graphs, tables and equations of various kinds. The exact notation is used in the system definition are highly dependent on the problem area. Obviously, one uses different terminology to describe an accounting system than to describe a process control system. Analysis is defined as the procedure by which we break down at intellectual or substantial whole into parts or components.

3.1 Existing System

Existing system is very complicated to keep the track of all registers and handle them manually. As well as this work is time consuming & also expensive in this system report work may be not accurate and not fastest. To avoid all these limitations and make the working more accurately the system needs to be computerized

Drawbacks of the existing system

- Time consuming
- More expensive
- Searching problem
- Maintains problem of all registers
- Less accuracy
- Display multiple reports

3.2 Proposed System

In proposed system, the management needs not to keep any type of registers, which they use to keep in old one. They have work only with one computer. All the details are stored in computer files. The dual entries are done very quickly as entry in one file only , affects the other file where it has to record. In the way there is no need to record the computer operator.

The working through registers system is not only very laborious and time consuming , but there are number of errors found. In comparison to old system the proposed system of computer processes many advantages.

3.3 Objectives of the Proposed System

- Accuracy
- Security
- Time Saving
- Perform Repetitive Task Very Well
- Low Cost
- Easy Maintenance
- Easy Modification
- Attractive and Status

3.4 Feasibility Study

An important objective of conducting the System Analysis is the determination of the feasibility. Feasibility study is carried out to select the best system that meets the performance requirements. Feasibility study is necessary to evaluate the feasibility of a project at the earliest possible time. It is used to investigate the project and examines whether the designed system will be useful to the organization. Feasibility and risk analysis are related in many ways. If project risk is great, the feasibility of producing quality software is reduced. The different type of feasibility are:

- Technical Feasibility
- Operational Feasibility
- Economic Feasibility

Technical Feasibility

Technical Feasibility is used to find out whether the necessary technology, the proposed equipment have the capacity to hold the data, which is used in the project. It is a study of function, performance and constraints that may affect the ability to achieve an acceptance system.

The technical Feasibility issues usually raised during the feasibility stage of investigation includes these:

- The software which is running on windows XP operating system can be easily installed.
- The hardware required is Pentium based server.

Operational Feasibility

This feasibility test asks if the system will works when it is developed is installed.

Operational feasibility in this project are:

- The proposed system offers greater level of user-friendly lines.
- The proposed system produces best results and gives high performance.
- It can be implemented easily. So the project is operational feasible.

Economic Feasibility

Economic feasibility deals about the economic impact faced by the organization to implement a new system. Financial benefits must be equal or exceed the cost of the existing system. The cost for implementing the proposed system, including hardware and software cost should be evaluated.

Economic feasibility in this project are:

- There is no additional man power requirement.
- There is no additional cost involved maintaining the proposed system.

CHAPTER 4

SYSTEM SPECIFICATION

4.1 Hardware Specification

Processor	:	Intel core i3 processor
Processor speed	:	2.0GHZ
RAM	:	4 GB
Hard Disk Drive	:	1 TB

4.2 Software Specification

Operating System	:	Windows 10
Front end tool	:	PyCharm
Back end tool	:	Xampp

4.3 Language Tool

PyCharm

PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django as well as DataScience with Anaconda.PyCharmis crossplatform,with Windows, macOS and Lin

ux versions. The Community Edition is released under the Apache License, and there is also Professional Edition with extra features – released under a proprietary license.

Features

- Coding assistance and analysis, with code completion, syntax and error highlighting, linter integration, and quick fixes
- Project and code navigation: specialized project views, file structure views and quick jumping between files, classes, methods and usages
- Python refactoring: includes rename, extract method, introduce variable, introduce constant, pull up, push down and others

Support for web frameworks: Django, web2py and Flask

- Integrated Python debugger
- Integrated unit testing, with line-by-line code coverage
- Google App Engine Python development
- Version control integration: unified user interface
  for Mercurial, Git, Subversion, Perforce and CVS with change lists and merge

Oracle

Oracle database is capable of supporting over 10000 simultaneous users and a database size of up to 100 terabytes. It is preferred to the other pc based RDBMS packages because its client/server database qualities, failure handling, recovery management, administrative tools to manage users and the database, object-oriented capabilities, graphical user interface (GUI) tools, and web interface capabilities. It is widely used by corporation of all size to develop mission-critical application. It is also used as a teaching tool by educational institutions to teach object-oriented relational database technology, structured query language (SQL), PL/SQL (oracle's procedural language extension to SQL), and interfacing web and oracle database. Oracle has an educational initiative program to form partnerships with educational institutional to obtain oracle database software at a nominal membership.

Oracle software is installed to work in three different environments in a stand-alone environment, such as a laptop or desktop that is not an a network, oracle enterprise database software SQL*plus client software are installed on same machine in a client/server environment, a two-tier architecture with a server, oracle enterprise database software resides on the server side, and SQL*plus client software resides on the client machine. In three-tier architecture, the client communicates with oracle database server through a middle-tier iSQL*plus, an interface through a web browser.

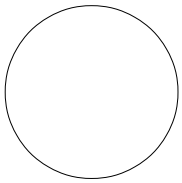
- The oracle environment provides utilities to work with database tables, developing forms, reports, and graphs, managing users and database, and interfacing the web and database.
- SQL*plus is oracle's proprietary environment to enter SQL queries
- SQL is a language with data retrieval, DML, DDL, DCL, and transaction control query statements.
- SQL* plus is an environment that provides user with editing, file, formatting, execution, interaction commands.

CHAPTER-5

5.1 Data Flow Diagram

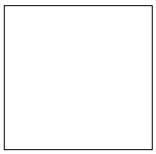
Data flow oriented techniques advocate that the major data items handled by a system must be first identified and then the processing required on these data items to produce the desired outputs should be determined. The DFD (also called as bubble chart) is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on these data, and the output generated by the system. It was introduced by De Macro (1978), Gane and Sarson (1979).The primitive symbols used for constructing DFD's are:

Symbols used in DFD

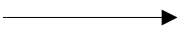


A circle represents a process.

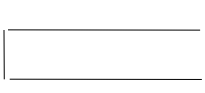
A rectangle represents external entity



A square defines a source or destination of the system data.



An arrow identifies dataflow.



Double line with one end closed indicate data store.

DFD:

It is also known as context diagram. It's designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as single bubble with input and output data indicated by incoming/outgoing arrows.

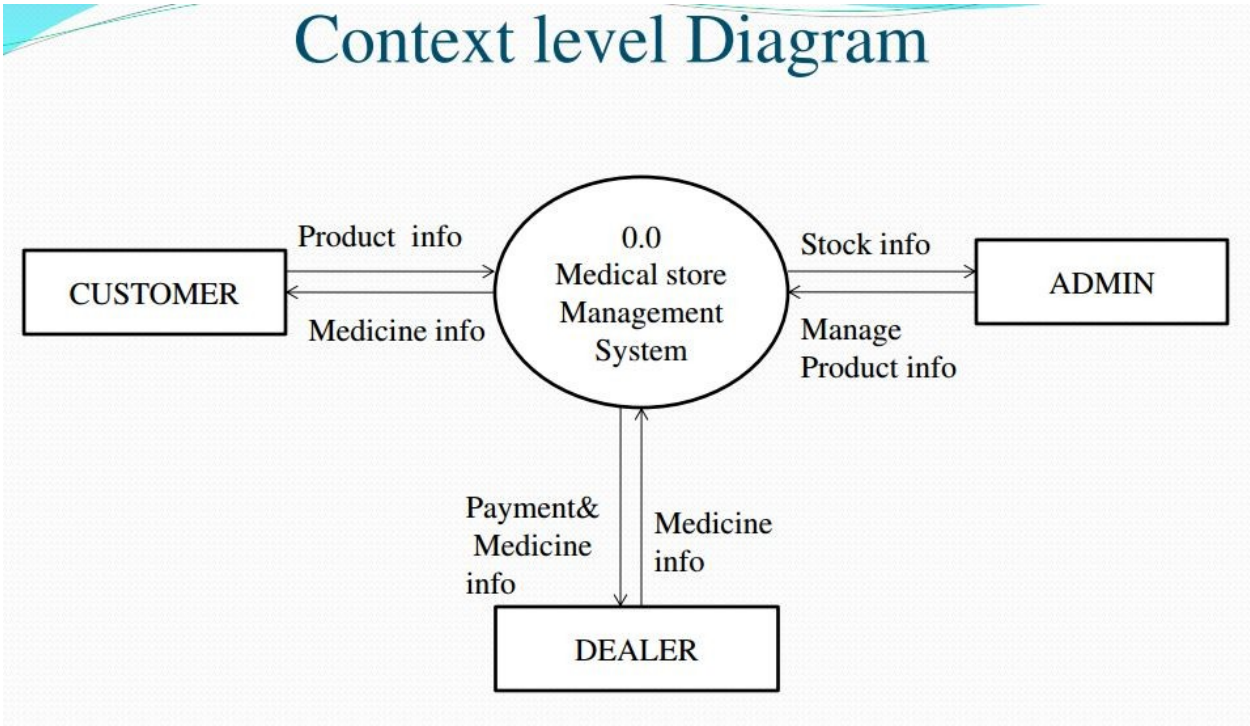


Figure 5.1 Data Flow Diagram

CHAPTER-6

SYSTEM DEVELOPMENT

Module Description

6.1 Add product details

The medicines been purchased by the medical shop, the details are recorded in the Purchase Item. Some of the details been recorded by the Medical shop agent is the agency from which the medicines are bought, item details such as item type, item name, quantity, manufacturing date, expiry date etc.

The operations that are been performed here:

To add new records into the database, to modify an already existing record in the database , to search a record and delete a record from the database.

6.2 Sales Item

The medicines been sold out by the medical shop, the details are recorded in the Sales Item. Some of the details been recorded by the Medical shop agent is the agency or the customer and their respective address to whom the medicines are sold, item details such as item type, item name, quantity, manufacturing date, expiry date etc.:

6.3 Check product

The medicine details such as name of the medicine, bill number, manufacturing date, expiry date, price and the quantity of the medicines etc are been recorded in the Item Details.

6.4 Stock Details

The Stock details deals with the details of the agency as well as the stock item such as the item type, item name, price and quantity.

The operations that are been performed here:

To display the sales details of the stock as recorded in the database ,to search a record in the database.

6.5 Add supplier details

In this module we can add supplier details like add supplier's name, supplier's address and phone number and this module is required for further contact when we need medicine stock

CHAPTER-7

SYSTEM TESTING AND IMPLEMENTATION

Testing is an activity to verify that a correct system is being build and if perform with the intention of results once gathered and evaluated provide a qualitative indication of software quality and reliability and serve as a basis for design modification. A project is said to be incomplete without proper testing.

7.1 System Testing

Testing Strategies

System testing is the stage of implementation that is aimed at ensuring that the system works accurately and efficiently before live operation commences. Testing is vital for the success of the system. System testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved.

A series of tests are performed for the proposed system is ready for User Acceptance Testing. The testing steps are

- Unit Testing
- Integration Testing
- Validation Testing
- Output Testing
- User Acceptance Testing

Unit Testing

Unit testing focuses verification efforts on the smallest unit of software design, the machine module. This is also known as "MODULE TESTING". The modules are tested separately is carried out during programming stage itself. In this step, each module is found to be working satisfactory. So the expected output from the module is arrived.

Integration Testing

Data can be lost across and interface, one module can have an adverse effect on others, and sub-functions when combined may not produce the desired major functions. Integration Testing is a Systematic Structure, while at the same time conducting to uncover errors associated within the interfaces. The objective is to take unit-tested modules and to combine them and test it as a whole. Here correction is difficult because the vast expenses of the entire program complicate the isolation of causes. This is the integration testing step. All the errors encountered will be carried over to the next testing step.

Validation Testing

Validation testing is performed in order to find whether there is any flow and deficiency in requirements of the system. Each and every module is tested independently. The input and output process are validated effectively. The input of the user for each field is validated to check if value entered is null (or) valid. After field level validation, the module level is made such that they are integrated and a validation is performed for the whole system.

Output Testing

After performing the validation, the next is output testing of the proposed system. Since no system could be useful. If it does not produce the required output in the special format asking the user about the requirement, test the output generated (or) displayed by the system. Hence the output format is considered in two ways. On is on of the screen and other is printed format.

User-Acceptance Testing

User-Acceptance of a system is the key factors for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the perspective system users at the time of developing and makes changes whenever required. This is done with the regard of the following points.

- Input Screen design
- Output Screen
- Menu driven system
- Format of crystal reports and other output.

7.2 System Implementation

System implementation is the stage of the project that the theoretical design is turned into a working system. If the implementation stage is not properly plane and control, it can be considered to be the most crucial stage in achieving a successful new system and in giving the user confidence that the new system will work and be effective.

Normally this stage involves setting up a coordinating committee, which will act as a sounding board for ideas; complaints and problem. The first task is implementation planning; i.e., deciding on the methods and time scale to be adopted. Apart from planning two major task of preparing for implementation are, education taken place much earlier in the project at the implementation stage the emphasis must be on training in new skills to gives staff confidence that they can use the system. Once staff has been trained the system can be tested.

After implementation phase, the user staff is adjusted to the changes created by the candidate system, evaluation and maintenance to bring the new system to standard. There is an there types

- Implementation Planning
- Education Planning
- System Planning

CHAPTER-8

8.1 Database table

Database name: Project

Table Name : submit

Primary Key : id

Field name	Data type	Size	Description
id	Integer	5	Supplier Id
name	Varchar	40	Supplier's name
address	Varchar	60	Supplier's address
contact	Bigint	10	Supplier's contact

Database name: Project

Table Name : submit1

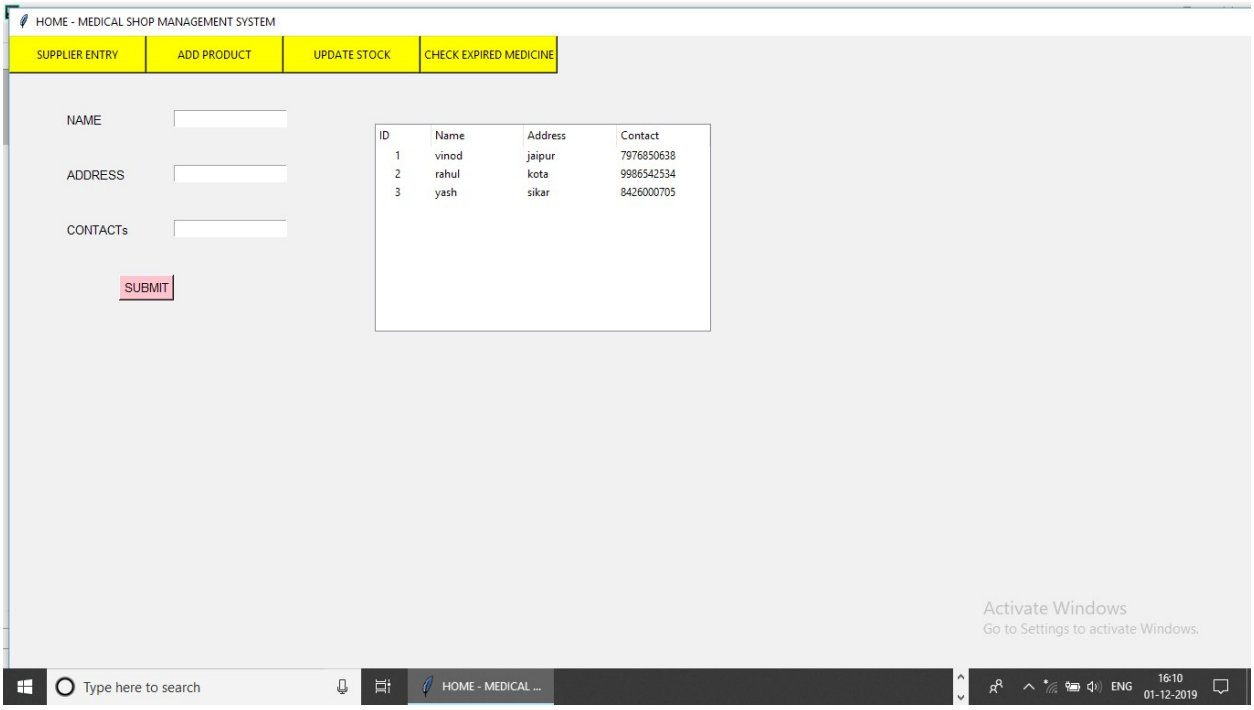
Primary Key : p_ id

Field name	Data type	Size	Description
P_id	Integer	5	Product Id
P_name	Varchar	30	Product's name
Company	Varchar	30	Product's company
quantity	double	-	Product's quantity
rateperitem	int	11	Rate per item
deliveryd	Varchar	30	Delivery date
manufacture	Varchar	30	Manufacture date
expirayd	Varchar	30	Expiry date
supplier	Varchar	30	Supplier name

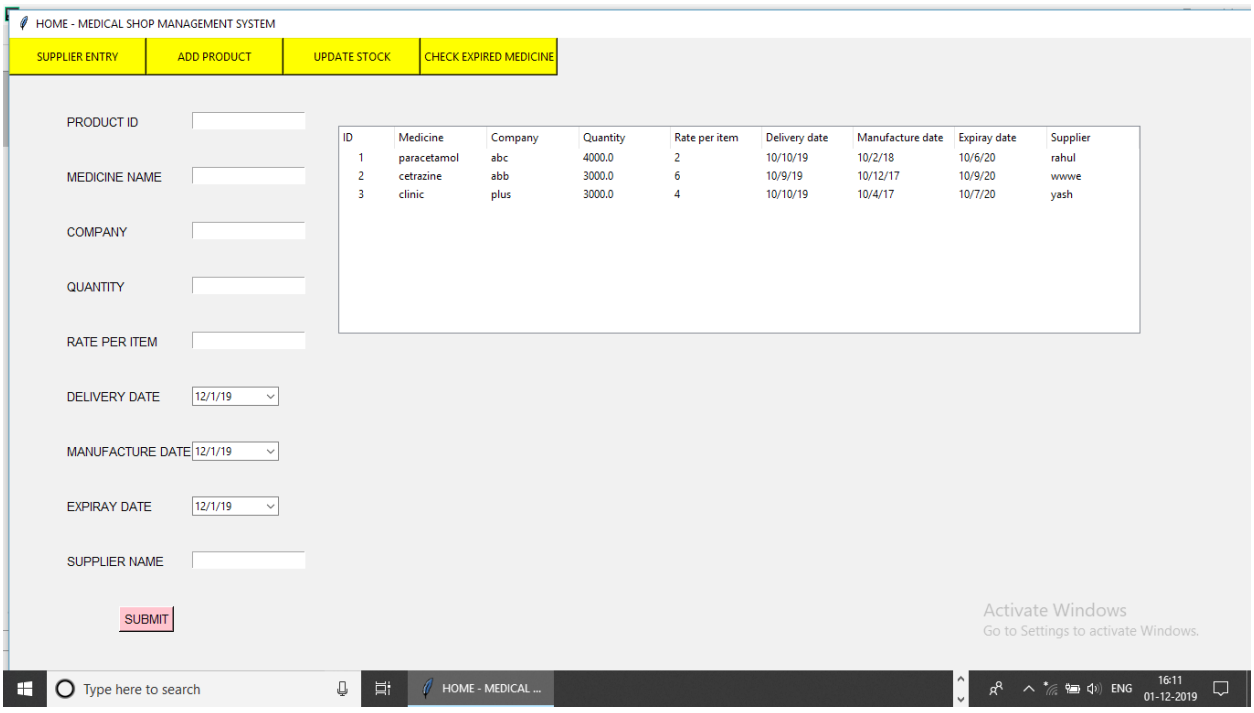
CHAPTER-9

9.1 Project's Screenshots

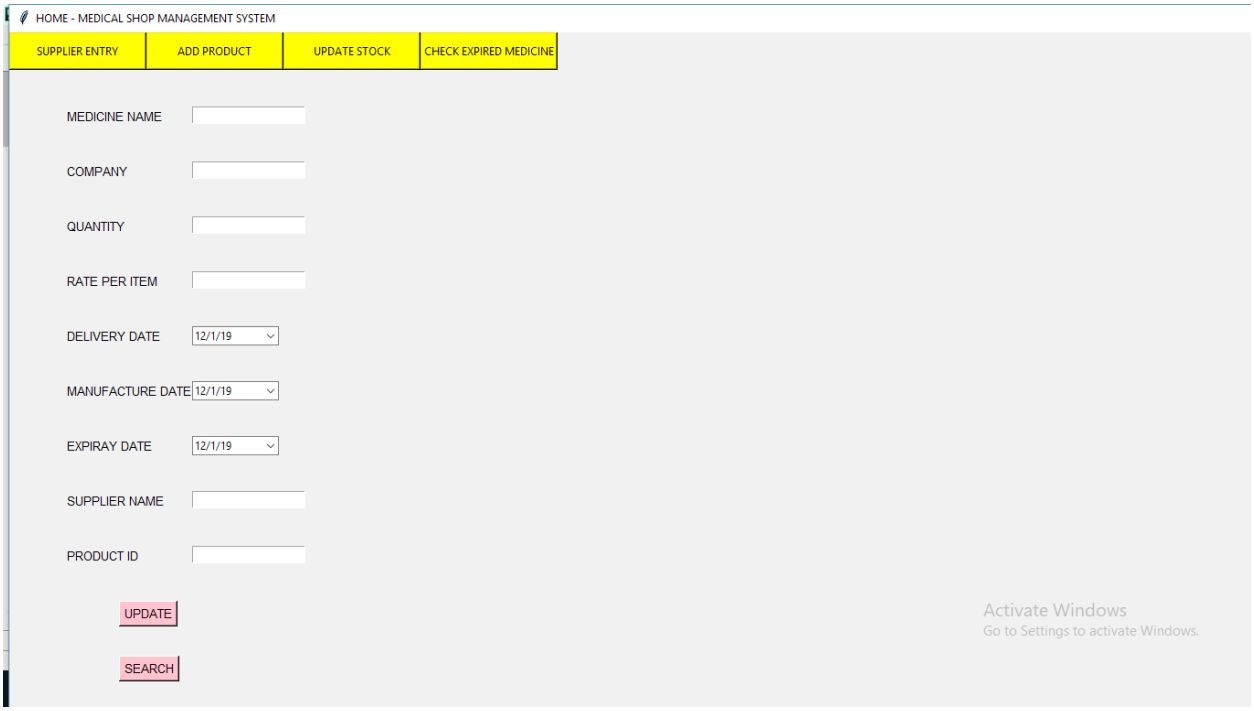
9.1.1 Supplier's entry



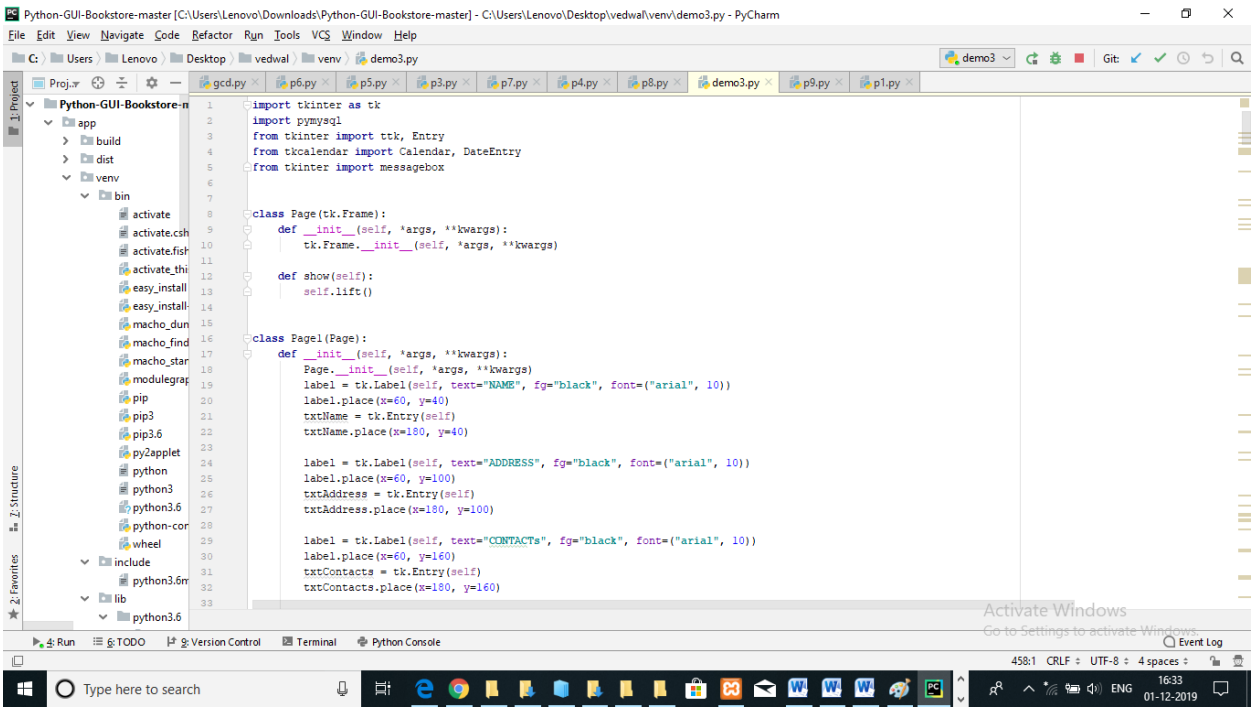
9.1.2 Add product details

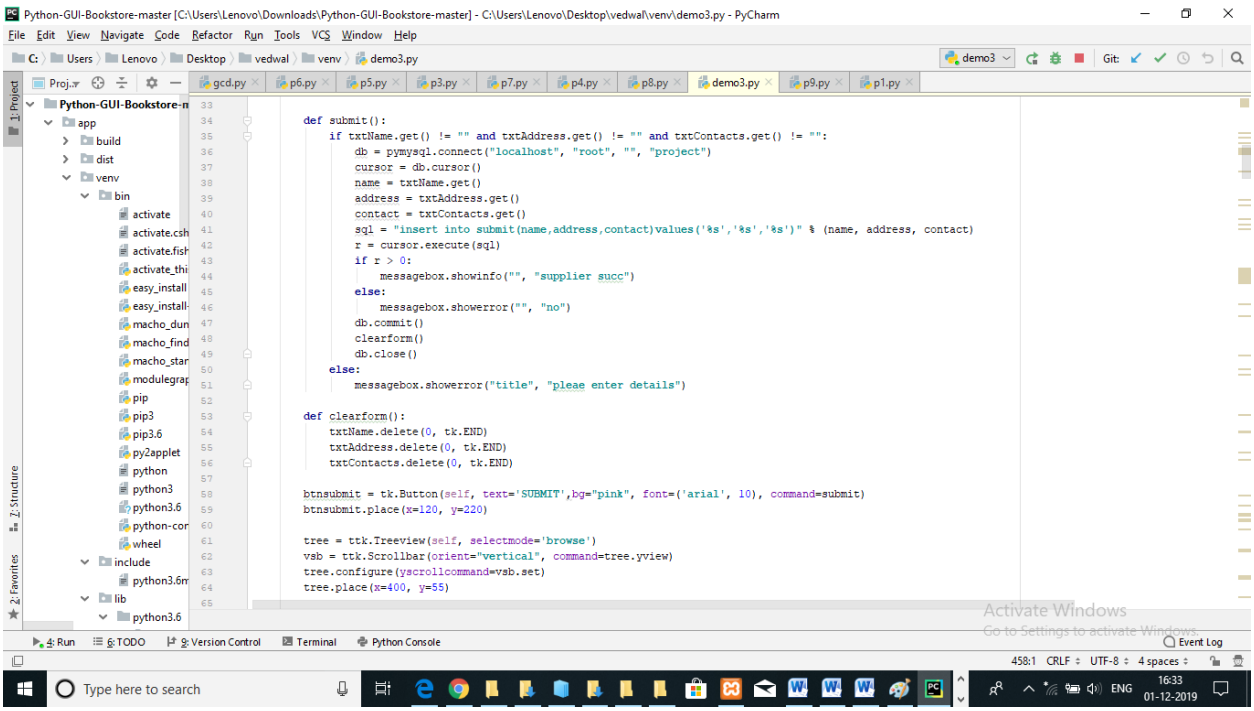


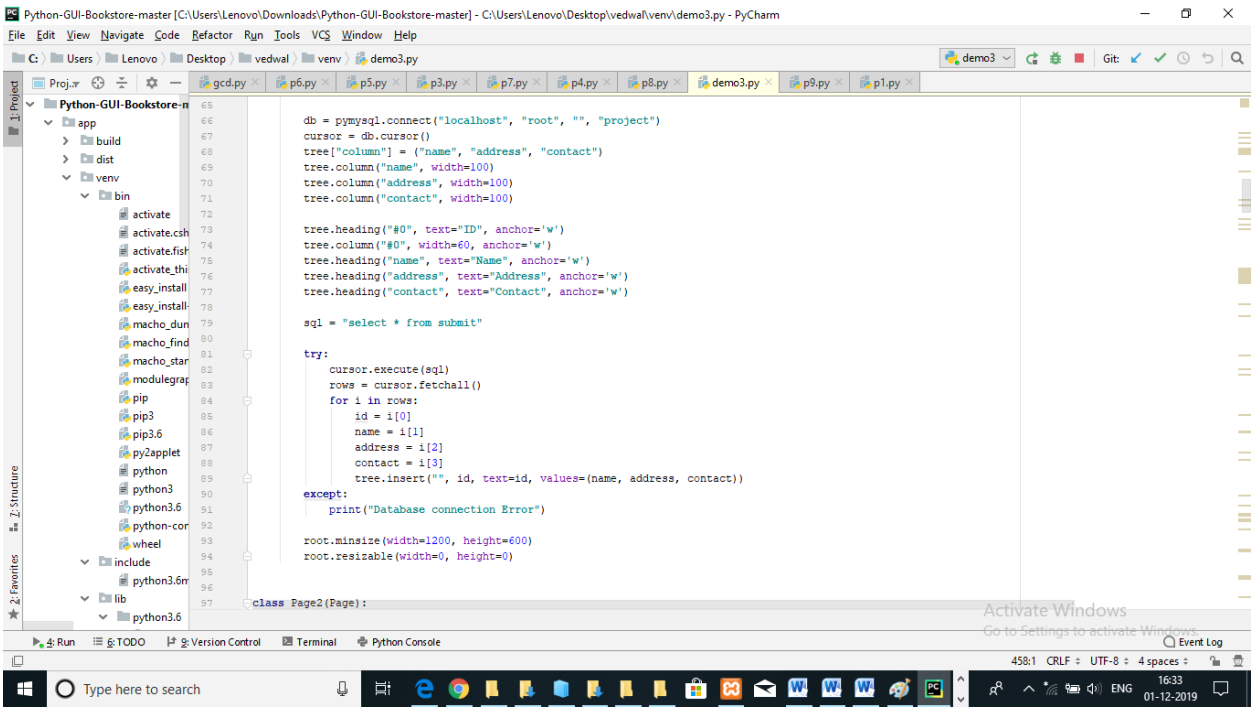
9.1.3 Update and Search product's details

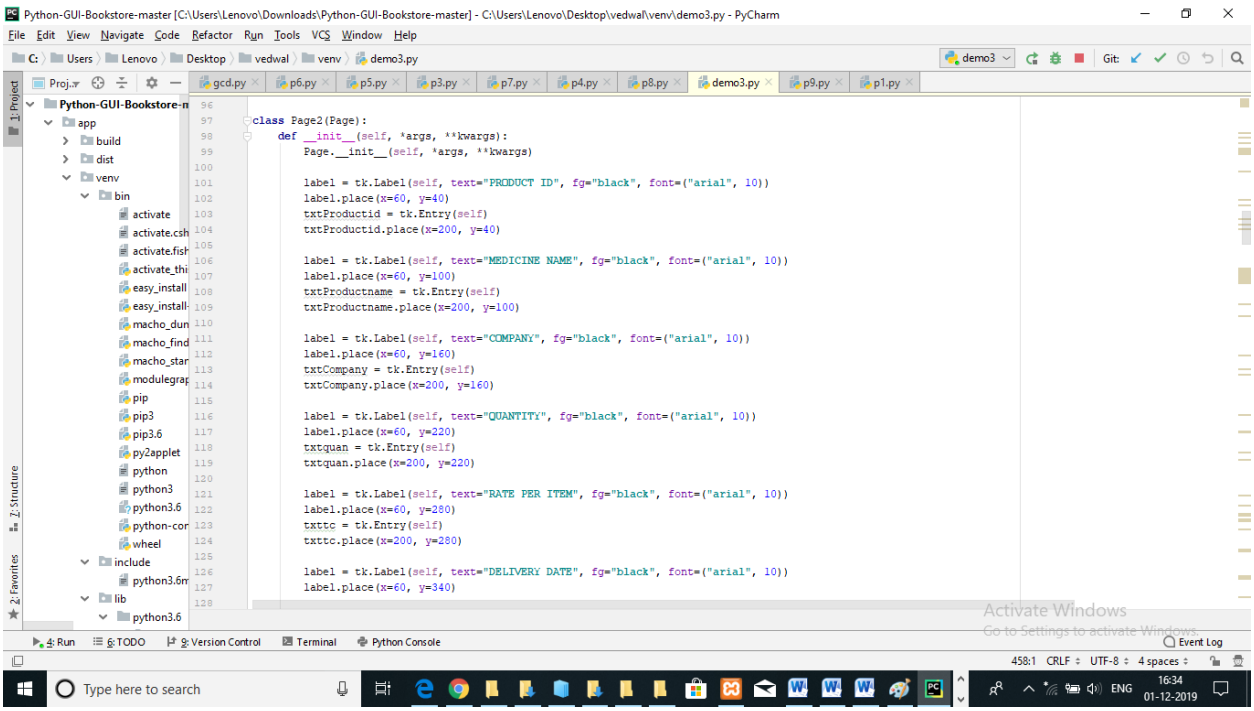


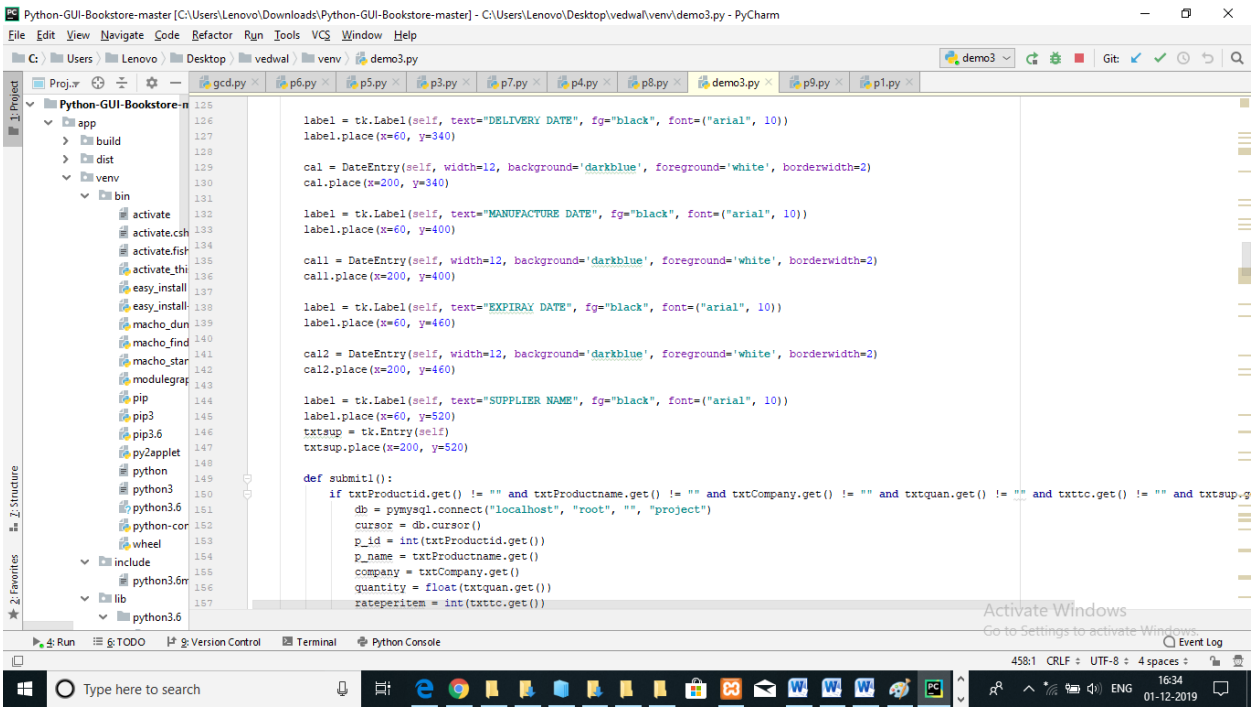
9.2 Screenshot of Code

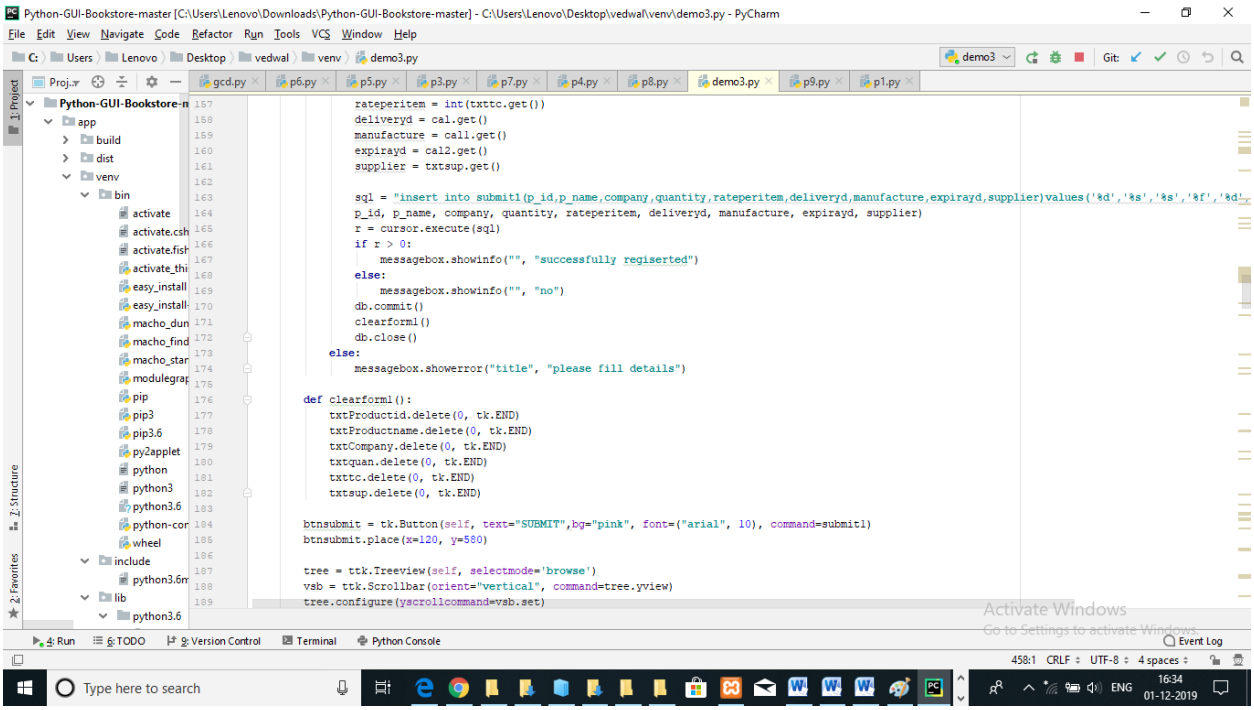

Python-GUI-Bookstore-master [C:\Users\Lenovo\Downloads\Python-GUI-Bookstore-master] - C:\Users\Lenovo\Desktop\vedwal\venv\demo3.py - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
demo3.py
import tkinter as tk
import pymysql
from tkinter import ttk, Entry
from tkcalendar import Calendar, DateEntry
from tkinter import messagebox
class Page(tk.Frame):
def __init__(self, *args, **kwargs):
tk.Frame.__init__(self, *args, **kwargs)
def show(self):
self.lift()
class Page1(Page):
def __init__(self, *args, **kwargs):
Page.__init__(self, *args, **kwargs)
label = tk.Label(self, text="NAME", fg="black", font=("arial", 10))
label.place(x=60, y=40)
txtName = tk.Entry(self)
txtName.place(x=180, y=40)
label = tk.Label(self, text="ADDRESS", fg="black", font=("arial", 10))
label.place(x=60, y=100)
txtAddress = tk.Entry(self)
txtAddress.place(x=180, y=100)
label = tk.Label(self, text="CONTACTs", fg="black", font=("arial", 10))
label.place(x=60, y=160)
txtContacts = tk.Entry(self)
txtContacts.place(x=180, y=160)
Activate Windows
Go to Settings to activate Windows.
4: Run 6: TODO 9: Version Control Terminal Python Console
Event Log
458:1 CRLF UTF-8 4 spaces
Type here to search
ENG 16:33 01-12-2019

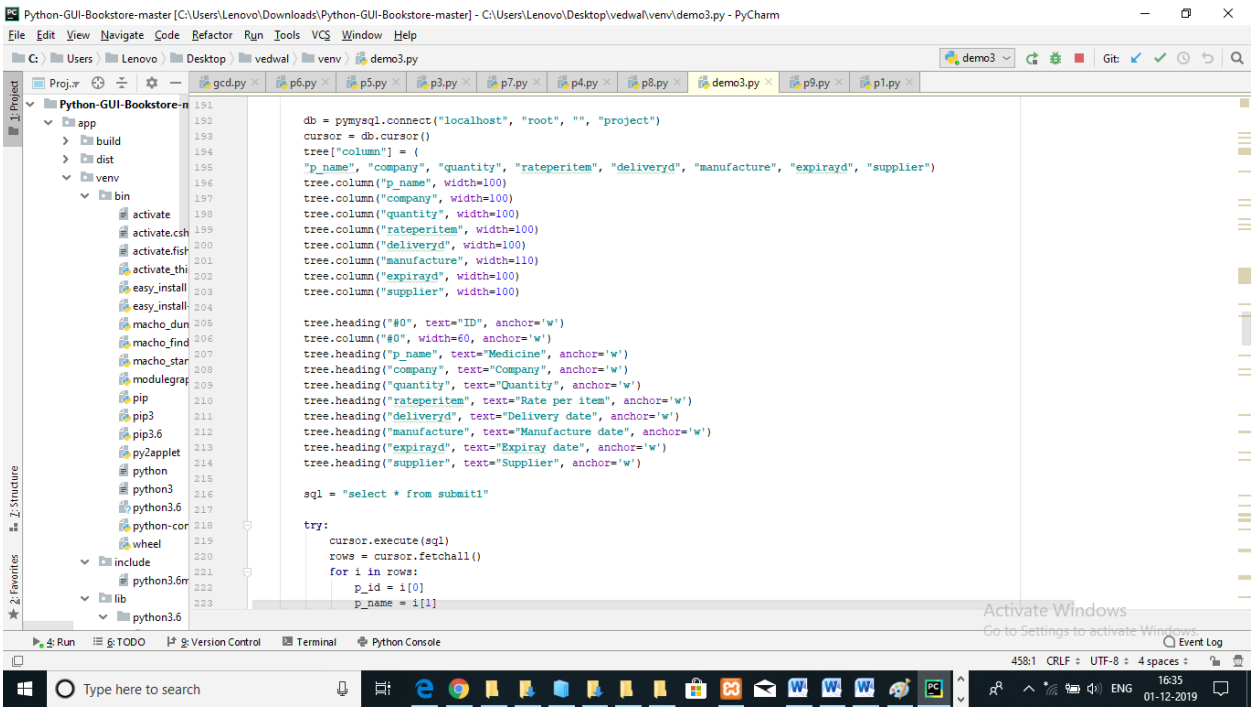

Python-GUI-Bookstore-master [C:\Users\Lenovo\Downloads\Python-GUI-Bookstore-master] - C:\Users\Lenovo\Desktop\vedwal\venv\demo3.py - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
demo3.py
def submit():
if txtName.get() != "" and txtAddress.get() != "" and txtContacts.get() != "":
db = pymysql.connect("localhost", "root", "", "project")
cursor = db.cursor()
name = txtName.get()
address = txtAddress.get()
contact = txtContacts.get()
sql = "insert into submit(name,address,contact)values('%s','%s','%s')" % (name, address, contact)
r = cursor.execute(sql)
if r > 0:
messagebox.showinfo("", "supplier succ")
else:
messagebox.showerror("", "no")
db.commit()
clearform()
db.close()
else:
messagebox.showerror("title", "pleae enter details")
def clearform():
txtName.delete(0, tk.END)
txtAddress.delete(0, tk.END)
txtContacts.delete(0, tk.END)
btnsubmit = tk.Button(self, text='SUBMIT',bg="pink", font=('arial', 10), command=submit)
btnsubmit.place(x=120, y=220)
tree = ttk.Treeview(self, selectmode='browse')
vsb = ttk.Scrollbar(orient="vertical", command=tree.yview)
tree.configure(yscrollcommand=vsb.set)
tree.place(x=400, y=55)
Activate Windows
Go to Settings to activate Windows.
4: Run 6: TODO 9: Version Control Terminal Python Console
Event Log
458:1 CRLF UTF-8 4 spaces
Type here to search
ENG 16:33 01-12-2019

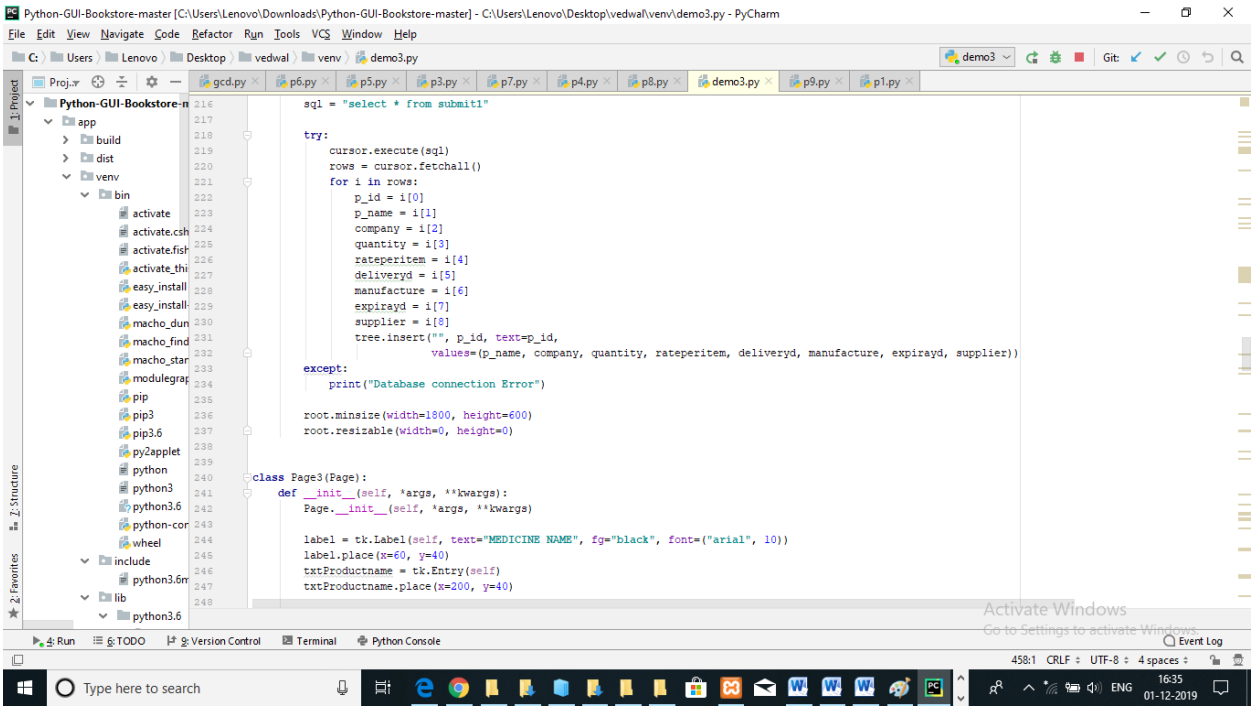

Python-GUI-Bookstore-master [C:\Users\Lenovo\Downloads\Python-GUI-Bookstore-master] - C:\Users\Lenovo\Desktop\vedwal\venv\demo3.py - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
gcd.py p6.py p5.py p3.py p7.py p4.py p8.py demo3.py p9.py p1.py
db = pymysql.connect("localhost", "root", "", "project")
cursor = db.cursor()
tree["column"] = ("name", "address", "contact")
tree.column("name", width=100)
tree.column("address", width=100)
tree.column("contact", width=100)
tree.heading("#0", text="ID", anchor='w')
tree.column("#0", width=60, anchor='w')
tree.heading("name", text="Name", anchor='w')
tree.heading("address", text="Address", anchor='w')
tree.heading("contact", text="Contact", anchor='w')
sql = "select * from submit"
try:
cursor.execute(sql)
rows = cursor.fetchall()
for i in rows:
id = i[0]
name = i[1]
address = i[2]
contact = i[3]
tree.insert("", id, text=id, values=(name, address, contact))
except:
print("Database connection Error")
root.minsize(width=1200, height=600)
root.resizable(width=0, height=0)
class Page2(Page):
Activate Windows
Go to Settings to activate Windows.
4: Run 6: TODO 9: Version Control Terminal Python Console
Event Log
458:1 CRLF UTF-8 4 spaces
Type here to search
ENG 16:33 01-12-2019

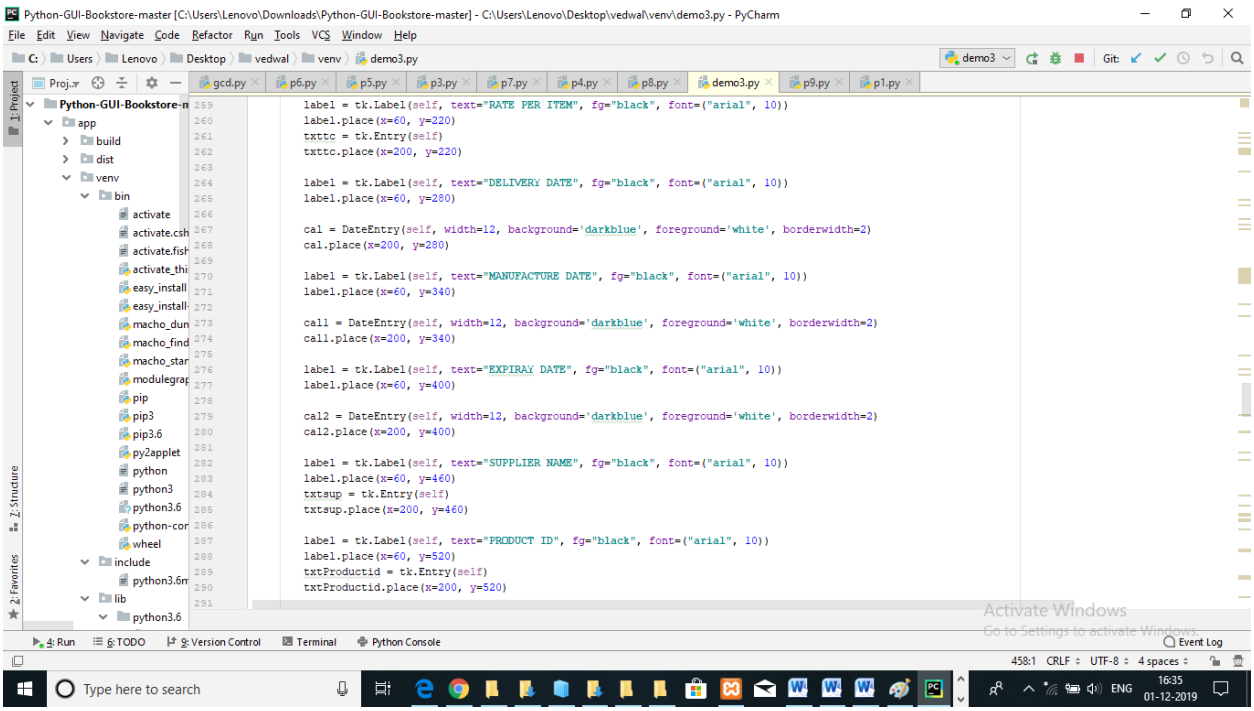

Python-GUI-Bookstore-master [C:\Users\Lenovo\Downloads\Python-GUI-Bookstore-master] - C:\Users\Lenovo\Desktop\vedwal\venv\demo3.py - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
gcd.py p6.py p5.py p3.py p7.py p4.py p8.py demo3.py p9.py p1.py
class Page2(Page):
def __init__(self, *args, **kwargs):
Page.__init__(self, *args, **kwargs)
label = tk.Label(self, text="PRODUCT ID", fg="black", font=("arial", 10))
label.place(x=60, y=40)
txtProductid = tk.Entry(self)
txtProductid.place(x=200, y=40)
label = tk.Label(self, text="MEDICINE NAME", fg="black", font=("arial", 10))
label.place(x=60, y=100)
txtProductname = tk.Entry(self)
txtProductname.place(x=200, y=100)
label = tk.Label(self, text="COMPANY", fg="black", font=("arial", 10))
label.place(x=60, y=160)
txtCompany = tk.Entry(self)
txtCompany.place(x=200, y=160)
label = tk.Label(self, text="QUANTITY", fg="black", font=("arial", 10))
label.place(x=60, y=220)
txtquan = tk.Entry(self)
txtquan.place(x=200, y=220)
label = tk.Label(self, text="RATE PER ITEM", fg="black", font=("arial", 10))
label.place(x=60, y=280)
txttc = tk.Entry(self)
txttc.place(x=200, y=280)
label = tk.Label(self, text="DELIVERY DATE", fg="black", font=("arial", 10))
label.place(x=60, y=340)
Activate Windows
Go to Settings to activate Windows.
4: Run 6: TODO 9: Version Control Terminal Python Console
Event Log
458:1 CRLF UTF-8 4 spaces
Type here to search
ENG 16:34 01-12-2019

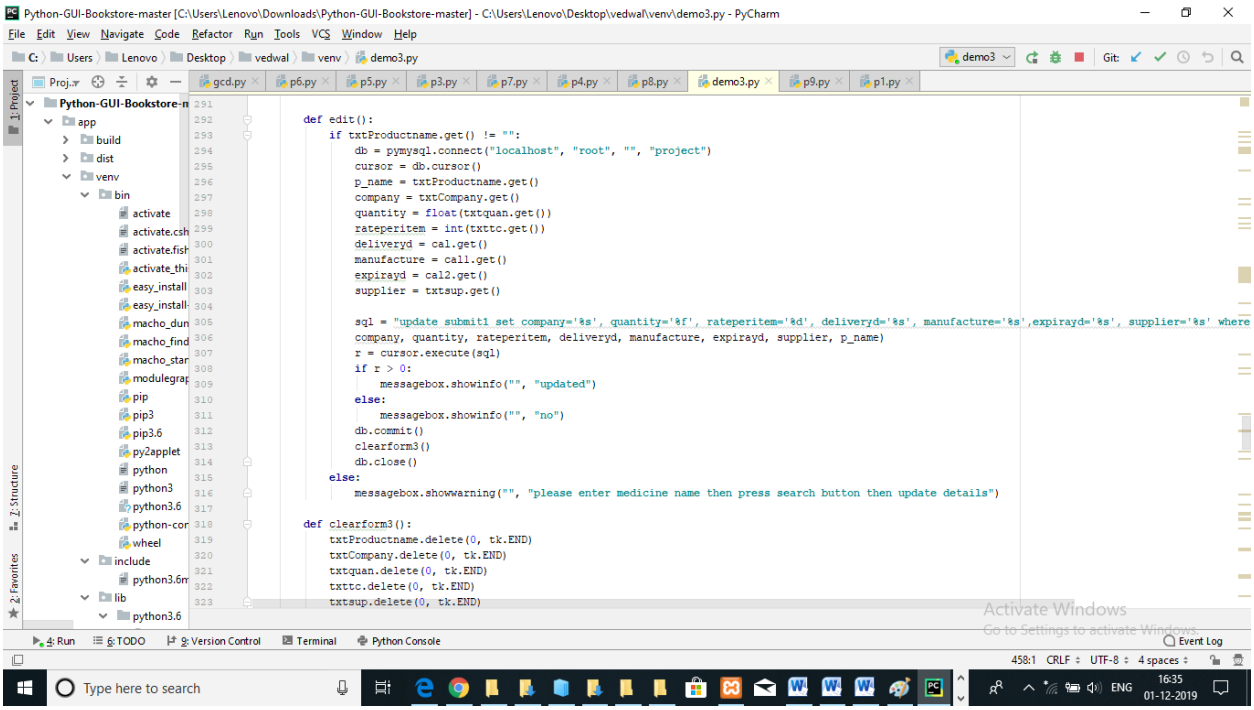

Python-GUI-Bookstore-master [C:\Users\Lenovo\Downloads\Python-GUI-Bookstore-master] - C:\Users\Lenovo\Desktop\vedwal\venv\demo3.py - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
label = tk.Label(self, text="DELIVERY DATE", fg="black", font=("arial", 10))
label.place(x=60, y=340)
cal = DateEntry(self, width=12, background='darkblue', foreground='white', borderwidth=2)
cal.place(x=200, y=340)
label = tk.Label(self, text="MANUFACTURE DATE", fg="black", font=("arial", 10))
label.place(x=60, y=400)
cal1 = DateEntry(self, width=12, background='darkblue', foreground='white', borderwidth=2)
cal1.place(x=200, y=400)
label = tk.Label(self, text="EXPIRAY DATE", fg="black", font=("arial", 10))
label.place(x=60, y=460)
cal2 = DateEntry(self, width=12, background='darkblue', foreground='white', borderwidth=2)
cal2.place(x=200, y=460)
label = tk.Label(self, text="SUPPLIER NAME", fg="black", font=("arial", 10))
label.place(x=60, y=520)
txtsup = tk.Entry(self)
txtsup.place(x=200, y=520)
def submit1():
if txtProductid.get() != "" and txtProductname.get() != "" and txtCompany.get() != "" and txtquan.get() != "" and txttc.get() != "" and txtsup.g
db = pymysql.connect("localhost", "root", "", "project")
cursor = db.cursor()
p_id = int(txtProductid.get())
p_name = txtProductname.get()
company = txtCompany.get()
quantity = float(txtquan.get())
rateperitem = int(txttc.get())
Activate Windows
Go to Settings to activate Windows.
4: Run 6: TODO 9: Version Control Terminal Python Console Event Log
458:1 CRLF UTF-8 4 spaces
Type here to search
ENG 16:34 01-12-2019

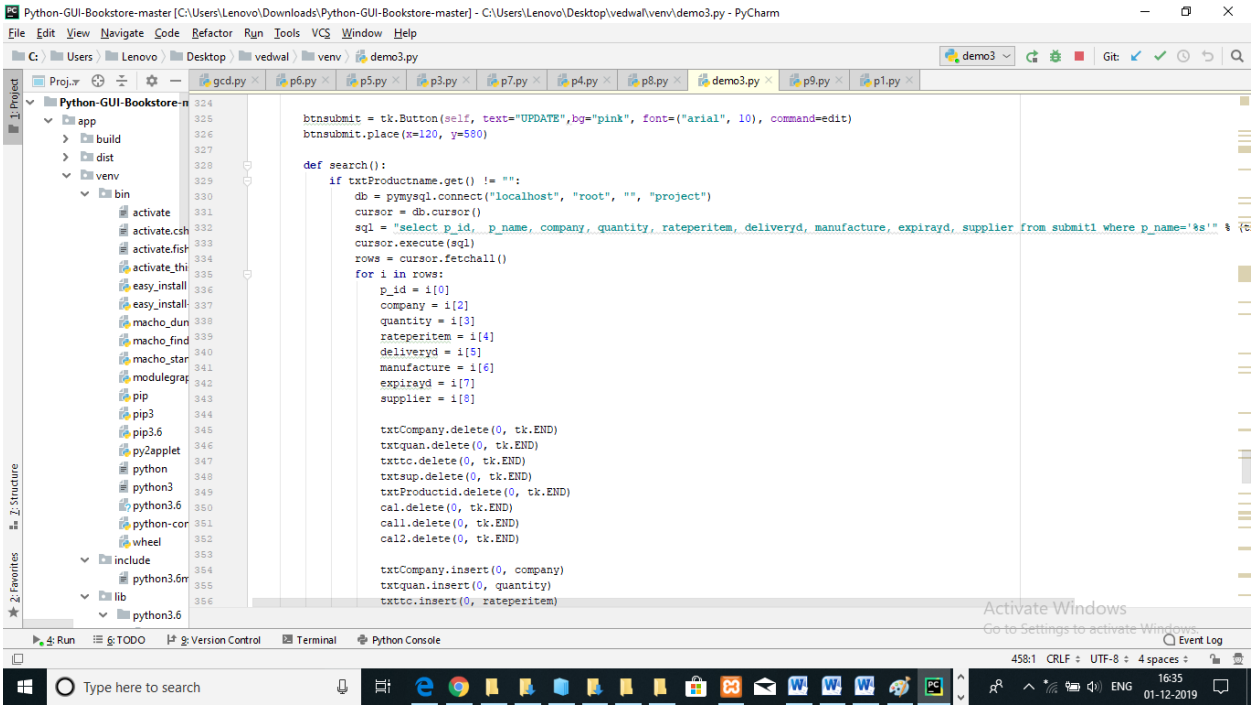

Python-GUI-Bookstore-master [C:\Users\Lenovo\Downloads\Python-GUI-Bookstore-master] - C:\Users\Lenovo\Desktop\vedwal\venv\demo3.py - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
rateperitem = int(txttc.get())
deliveryd = cal.get()
manufacture = cal1.get()
expirayd = cal2.get()
supplier = txtsup.get()
sql = "insert into submit1(p_id,p_name,company,quantity,rateperitem,deliveryd,manufacture,expirayd,supplier)values('%d','%s','%s','%f','%d',
p_id, p_name, company, quantity, rateperitem, deliveryd, manufacture, expirayd, supplier)
r = cursor.execute(sql)
if r > 0:
messagebox.showinfo("", "successfully regiserted")
else:
messagebox.showinfo("", "no")
db.commit()
clearform1()
db.close()
else:
messagebox.showerror("title", "please fill details")
def clearform1():
txtProductid.delete(0, tk.END)
txtProductname.delete(0, tk.END)
txtCompany.delete(0, tk.END)
txtquan.delete(0, tk.END)
txttc.delete(0, tk.END)
txtsup.delete(0, tk.END)
btnsubmit = tk.Button(self, text="SUBMIT",bg="pink", font=("arial", 10), command=submit1)
btnsubmit.place(x=120, y=580)
tree = ttk.Treeview(self, selectmode='browse')
vsb = ttk.Scrollbar(orient="vertical", command=tree.yview)
tree.configure(yscrollcommand=vsb.set)
Activate Windows
Go to Settings to activate Windows.
4: Run 6: TODO 9: Version Control Terminal Python Console Event Log
458:1 CRLF UTF-8 4 spaces
Type here to search
ENG 16:34 01-12-2019

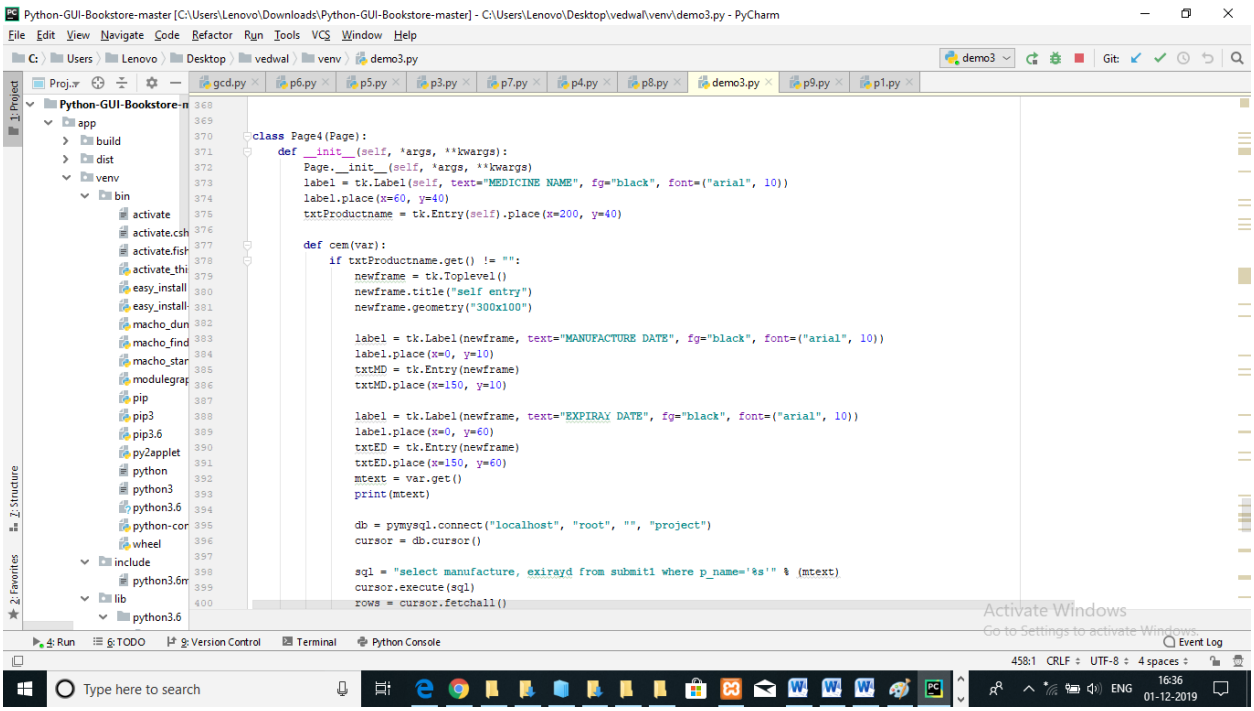

Python-GUI-Bookstore-master [C:\Users\Lenovo\Downloads\Python-GUI-Bookstore-master] - C:\Users\Lenovo\Desktop\vedwal\venv\demo3.py - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
gcd.py p6.py p5.py p3.py p7.py p4.py p8.py demo3.py p9.py p1.py
db = pymysql.connect("localhost", "root", "", "project")
cursor = db.cursor()
tree["column"] = (
"p_name", "company", "quantity", "rateperitem", "deliveryd", "manufacture", "expirayd", "supplier")
tree.column("p_name", width=100)
tree.column("company", width=100)
tree.column("quantity", width=100)
tree.column("rateperitem", width=100)
tree.column("deliveryd", width=100)
tree.column("manufacture", width=110)
tree.column("expirayd", width=100)
tree.column("supplier", width=100)
tree.heading("#0", text="ID", anchor='w')
tree.column("#0", width=60, anchor='w')
tree.heading("p_name", text="Medicine", anchor='w')
tree.heading("company", text="Company", anchor='w')
tree.heading("quantity", text="Quantity", anchor='w')
tree.heading("rateperitem", text="Rate per item", anchor='w')
tree.heading("deliveryd", text="Delivery date", anchor='w')
tree.heading("manufacture", text="Manufacture date", anchor='w')
tree.heading("expirayd", text="Expiray date", anchor='w')
tree.heading("supplier", text="Supplier", anchor='w')
sql = "select * from submit1"
try:
cursor.execute(sql)
rows = cursor.fetchall()
for i in rows:
p_id = i[0]
p_name = i[1]
Activate Windows
Go to Settings to activate Windows.
4: Run 6: TODO 9: Version Control Terminal Python Console Event Log
458:1 CRLF UTF-8 4 spaces
Type here to search
ENG 16:35 01-12-2019

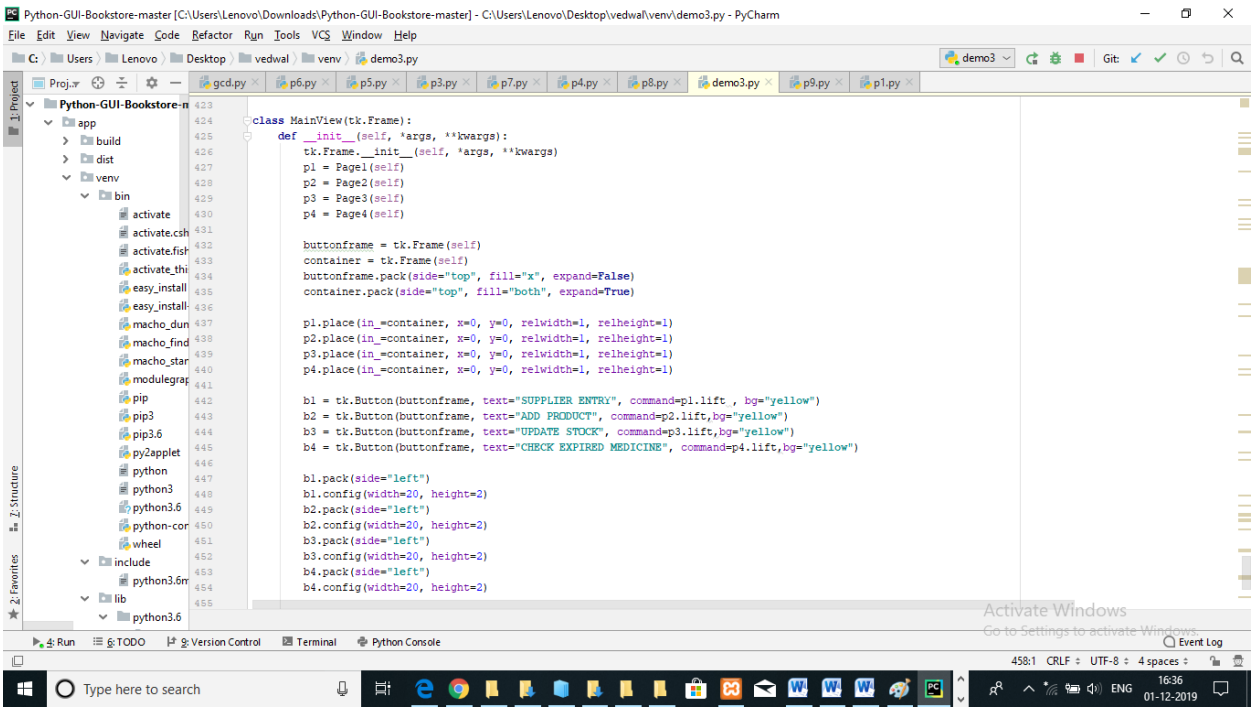

Python-GUI-Bookstore-master [C:\Users\Lenovo\Downloads\Python-GUI-Bookstore-master] - C:\Users\Lenovo\Desktop\vedwal\venv\demo3.py - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
gcd.py p6.py p5.py p3.py p7.py p4.py p8.py demo3.py p9.py p1.py
sql = "select * from submit1"
try:
cursor.execute(sql)
rows = cursor.fetchall()
for i in rows:
p_id = i[0]
p_name = i[1]
company = i[2]
quantity = i[3]
rateperitem = i[4]
deliveryd = i[5]
manufacture = i[6]
expirayd = i[7]
supplier = i[8]
tree.insert("", p_id, text=p_id,
values=(p_name, company, quantity, rateperitem, deliveryd, manufacture, expirayd, supplier))
except:
print("Database connection Error")
root.minsize(width=1800, height=600)
root.resizable(width=0, height=0)
class Page3(Page):
def __init__(self, *args, **kwargs):
Page.__init__(self, *args, **kwargs)
label = tk.Label(self, text="MEDICINE NAME", fg="black", font=("arial", 10))
label.place(x=60, y=40)
txtProductname = tk.Entry(self)
txtProductname.place(x=200, y=40)
Activate Windows
Go to Settings to activate Windows.
4: Run 6: TODO 9: Version Control Terminal Python Console Event Log
458:1 CRLF UTF-8 4 spaces
Type here to search
ENG 16:35 01-12-2019

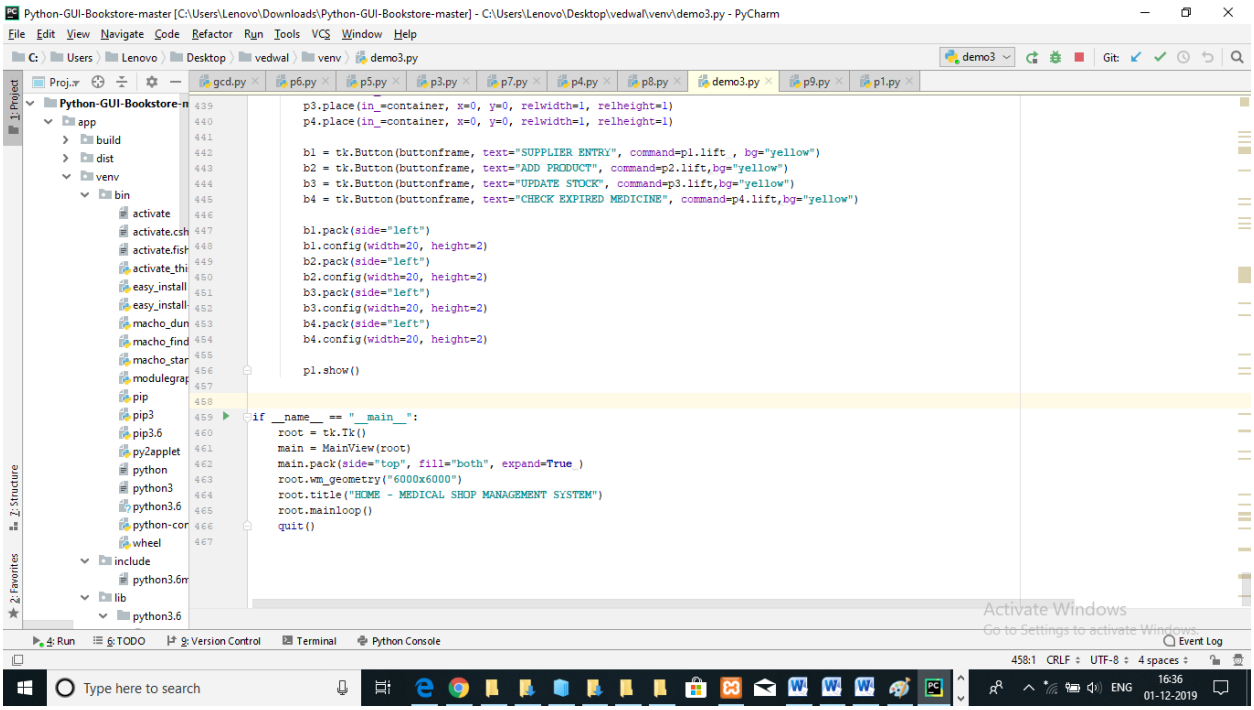

Python-GUI-Bookstore-master [C:\Users\Lenovo\Downloads\Python-GUI-Bookstore-master] - C:\Users\Lenovo\Desktop\vedwal\venv\demo3.py - PyCharm
label = tk.Label(self, text="RATE PER ITEM", fg="black", font=("arial", 10))
label.place(x=60, y=220)
txttc = tk.Entry(self)
txttc.place(x=200, y=220)
label = tk.Label(self, text="DELIVERY DATE", fg="black", font=("arial", 10))
label.place(x=60, y=280)
cal = DateEntry(self, width=12, background='darkblue', foreground='white', borderwidth=2)
cal.place(x=200, y=280)
label = tk.Label(self, text="MANUFACTURE DATE", fg="black", font=("arial", 10))
label.place(x=60, y=340)
cal1 = DateEntry(self, width=12, background='darkblue', foreground='white', borderwidth=2)
cal1.place(x=200, y=340)
label = tk.Label(self, text="EXPIRAY DATE", fg="black", font=("arial", 10))
label.place(x=60, y=400)
cal2 = DateEntry(self, width=12, background='darkblue', foreground='white', borderwidth=2)
cal2.place(x=200, y=400)
label = tk.Label(self, text="SUPPLIER NAME", fg="black", font=("arial", 10))
label.place(x=60, y=460)
txtsup = tk.Entry(self)
txtsup.place(x=200, y=460)
label = tk.Label(self, text="PRODUCT ID", fg="black", font=("arial", 10))
label.place(x=60, y=520)
txtProductid = tk.Entry(self)
txtProductid.place(x=200, y=520)
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG
16:35
01-12-2019


Python-GUI-Bookstore-master [C:\Users\Lenovo\Downloads\Python-GUI-Bookstore-master] - C:\Users\Lenovo\Desktop\vedwal\venv\demo3.py - PyCharm
def edit():
if txtProductname.get() != "":
db = pymysql.connect("localhost", "root", "", "project")
cursor = db.cursor()
p_name = txtProductname.get()
company = txtCompany.get()
quantity = float(txtquan.get())
rateperitem = int(txttc.get())
deliveryd = cal.get()
manufacture = cal1.get()
expirayd = cal2.get()
supplier = txtsup.get()
sql = "update submit1 set company='%s', quantity='%f', rateperitem='%d', deliveryd='%s', manufacture='%s',expirayd='%s', supplier='%s' where
company, quantity, rateperitem, deliveryd, manufacture, expirayd, supplier, p_name)
r = cursor.execute(sql)
if r > 0:
messagebox.showinfo("", "updated")
else:
messagebox.showinfo("", "no")
db.commit()
clearform3()
db.close()
else:
messagebox.showwarning("", "please enter medicine name then press search button then update details")
def clearform3():
txtProductname.delete(0, tk.END)
txtCompany.delete(0, tk.END)
txtquan.delete(0, tk.END)
txttc.delete(0, tk.END)
txtsup.delete(0, tk.END)
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG
16:35
01-12-2019


Python-GUI-Bookstore-master [C:\Users\Lenovo\Downloads\Python-GUI-Bookstore-master] - C:\Users\Lenovo\Desktop\vedwal\venv\demo3.py - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
C: Users Lenovo Desktop vedwal venv demo3.py
gcd.py p6.py p5.py p3.py p7.py p4.py p8.py demo3.py p9.py p1.py
Python-GUI-Bookstore-n
app
build
dist
venv
bin
activate
activate.csh
activate.fish
activate_thi
easy_install
easy_install-
macho_dun
macho_find
macho_star
modulegrap
pip
pip3
pip3.6
py2applet
python
python3
python3.6
python-cor
wheel
include
python3.6m
lib
python3.6
btnsubmit = tk.Button(self, text="UPDATE",bg="pink", font=("arial", 10), command=edit)
btnsubmit.place(x=120, y=580)
def search():
if txtProductname.get() != "":
db = pymysql.connect("localhost", "root", "", "project")
cursor = db.cursor()
sql = "select p_id, p_name, company, quantity, rateperitem, deliveryd, manufacture, expirayd, supplier from submit1 where p_name='%s'" %
cursor.execute(sql)
rows = cursor.fetchall()
for i in rows:
p_id = i[0]
company = i[2]
quantity = i[3]
rateperitem = i[4]
deliveryd = i[5]
manufacture = i[6]
expirayd = i[7]
supplier = i[8]
txtCompany.delete(0, tk.END)
txtquan.delete(0, tk.END)
txttc.delete(0, tk.END)
txtsup.delete(0, tk.END)
txtProductid.delete(0, tk.END)
cal.delete(0, tk.END)
cal1.delete(0, tk.END)
cal2.delete(0, tk.END)
txtCompany.insert(0, company)
txtquan.insert(0, quantity)
txttc.insert(0, rateperitem)
Activate Windows
Go to Settings to activate Windows.
4: Run 6: TODO 9: Version Control Terminal Python Console Event Log
458:1 CRLF UTF-8 4 spaces
Type here to search
ENG 16:35 01-12-2019


Python-GUI-Bookstore-master [C:\Users\Lenovo\Downloads\Python-GUI-Bookstore-master] - C:\Users\Lenovo\Desktop\vedwal\venv\demo3.py - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
C: Users Lenovo Desktop vedwal venv demo3.py
gcd.py p6.py p5.py p3.py p7.py p4.py p8.py demo3.py p9.py p1.py
Python-GUI-Bookstore-n
app
build
dist
venv
bin
activate
activate.csh
activate.fish
activate_thi
easy_install
easy_install-
macho_dun
macho_find
macho_star
modulegrap
pip
pip3
pip3.6
py2applet
python
python3
python3.6
python-cor
wheel
include
python3.6m
lib
python3.6
class Page4(Page):
def __init__(self, *args, **kwargs):
Page.__init__(self, *args, **kwargs)
label = tk.Label(self, text="MEDICINE NAME", fg="black", font=("arial", 10))
label.place(x=60, y=40)
txtProductname = tk.Entry(self).place(x=200, y=40)
def cem(var):
if txtProductname.get() != "":
newframe = tk.Toplevel()
newframe.title("self entry")
newframe.geometry("300x100")
label = tk.Label(newframe, text="MANUFACTURE DATE", fg="black", font=("arial", 10))
label.place(x=0, y=10)
txtMD = tk.Entry(newframe)
txtMD.place(x=150, y=10)
label = tk.Label(newframe, text="EXPIRAY DATE", fg="black", font=("arial", 10))
label.place(x=0, y=60)
txtED = tk.Entry(newframe)
txtED.place(x=150, y=60)
mtext = var.get()
print(mtext)
db = pymysql.connect("localhost", "root", "", "project")
cursor = db.cursor()
sql = "select manufacture, exirayd from submit1 where p_name='%s'" % (mtext)
cursor.execute(sql)
rows = cursor.fetchall()
Activate Windows
Go to Settings to activate Windows.
4: Run 6: TODO 9: Version Control Terminal Python Console Event Log
458:1 CRLF UTF-8 4 spaces
Type here to search
ENG 16:36 01-12-2019


Python-GUI-Bookstore-master [C:\Users\Lenovo\Downloads\Python-GUI-Bookstore-master] - C:\Users\Lenovo\Desktop\vedwal\venv\demo3.py - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
gcd.py p6.py p5.py p3.py p7.py p4.py p8.py demo3.py p9.py p1.py
class MainView(tk.Frame):
def __init__(self, *args, **kwargs):
tk.Frame.__init__(self, *args, **kwargs)
p1 = Page1(self)
p2 = Page2(self)
p3 = Page3(self)
p4 = Page4(self)
buttonframe = tk.Frame(self)
container = tk.Frame(self)
buttonframe.pack(side="top", fill="x", expand=False)
container.pack(side="top", fill="both", expand=True)
p1.place(in_=container, x=0, y=0, relwidth=1, relheight=1)
p2.place(in_=container, x=0, y=0, relwidth=1, relheight=1)
p3.place(in_=container, x=0, y=0, relwidth=1, relheight=1)
p4.place(in_=container, x=0, y=0, relwidth=1, relheight=1)
b1 = tk.Button(buttonframe, text="SUPPLIER ENTRY", command=p1.lift , bg="yellow")
b2 = tk.Button(buttonframe, text="ADD PRODUCT", command=p2.lift,bg="yellow")
b3 = tk.Button(buttonframe, text="UPDATE STOCK", command=p3.lift,bg="yellow")
b4 = tk.Button(buttonframe, text="CHECK EXPIRED MEDICINE", command=p4.lift,bg="yellow")
b1.pack(side="left")
b1.config(width=20, height=2)
b2.pack(side="left")
b2.config(width=20, height=2)
b3.pack(side="left")
b3.config(width=20, height=2)
b4.pack(side="left")
b4.config(width=20, height=2)
Activate Windows
Go to Settings to activate Windows.
4: Run 6: TODO 9: Version Control Terminal Python Console
Event Log
458:1 CRLF UTF-8 4 spaces
Type here to search
ENG 16:36 01-12-2019


Python-GUI-Bookstore-master [C:\Users\Lenovo\Downloads\Python-GUI-Bookstore-master] - C:\Users\Lenovo\Desktop\vedwal\venv\demo3.py - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
gcd.py p6.py p5.py p3.py p7.py p4.py p8.py demo3.py p9.py p1.py
p3.place(in_=container, x=0, y=0, relwidth=1, relheight=1)
p4.place(in_=container, x=0, y=0, relwidth=1, relheight=1)
b1 = tk.Button(buttonframe, text="SUPPLIER ENTRY", command=p1.lift , bg="yellow")
b2 = tk.Button(buttonframe, text="ADD PRODUCT", command=p2.lift,bg="yellow")
b3 = tk.Button(buttonframe, text="UPDATE STOCK", command=p3.lift,bg="yellow")
b4 = tk.Button(buttonframe, text="CHECK EXPIRED MEDICINE", command=p4.lift,bg="yellow")
b1.pack(side="left")
b1.config(width=20, height=2)
b2.pack(side="left")
b2.config(width=20, height=2)
b3.pack(side="left")
b3.config(width=20, height=2)
b4.pack(side="left")
b4.config(width=20, height=2)
p1.show()
if __name__ == "__main__":
root = tk.Tk()
main = MainView(root)
main.pack(side="top", fill="both", expand=True)
root.wm_geometry("6000x6000")
root.title("HOME - MEDICAL SHOP MANAGEMENT SYSTEM")
root.mainloop()
quit()
Activate Windows
Go to Settings to activate Windows.
4: Run 6: TODO 9: Version Control Terminal Python Console
Event Log
458:1 CRLF UTF-8 4 spaces
Type here to search
ENG 16:36 01-12-2019

CHAPTER-10

SCOPE FOR FUTURE ENHANCEMENT

This Application is designed to be generic as we develop our site we should take advantage of several areas in which you can improve and customize business .Automatic notification of newly placed orders to the company's shipping or processing department. Different kinds of products will be introduced in future in order to satisfy the customer needs. Many more facilities can be introduced.

CHAPTER-11

CONCLUSION

The software “Medical Shop Management System” has been developed in windows 10 environment using visual basic as front end and oracle as back end. Time consumptions reduced to a great extent and user as less complexity in handling this database.

The project is fully fledged and user friendly, End users will be lightened in using this software because it is easy to have bills and reports and mostly all contents to be entered are to selected from combo box. This reduces the calculating efforts to be carried out by the users.

BIBLIOGRAPHY

Referenced Book

1. Python programming fundamentals- a beginner's Handbook by Nischay kumar hegde (author)
2. Python and Tkinter Programming book by John E. Grayson
3. ‘Data base systems using oracle’, Nilesh shah, Phi publications
4. ‘Software Engineering’, Richard Fairley
5. ‘Software Analysis and Design’, Elnasri

Referenced Website

- www.jetbrains.com
- www.python.org
- https://www.w3schools.com/python/