INVENTORY MANAGEMENT SYSTEM

S.E. mini-project report submitted in partial

fulfilment of the requirements of the degree of

Information Technology

by

PRABAL NAIR EU1194018 (22)

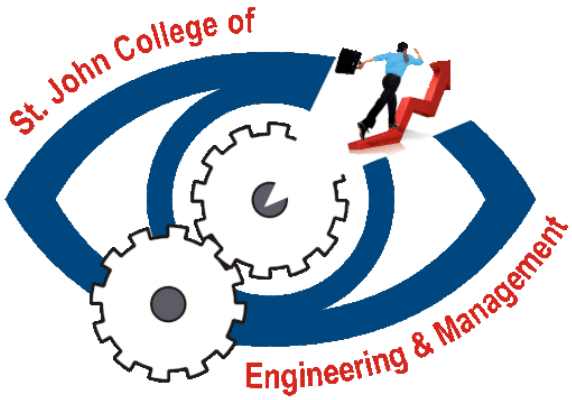
SHASHANK KARKERA EU1194013 (11)

PRITHVIRAJ PATIL EU1194011 (29)

NILESHSINGH MEHTA EU1194052 (15)

Under the guidance of

Ms. Shraddha More



Department of Information Technology

St. John College of Engineering and Management, Palghar

University of Mumbai

2020–2021

CERTIFICATE

This is to certify that the S.E. mini-project entitled **Inventory Management System** is a bonafide work of

Prabal Prabhakaran Nair PID No - EU1194018 (22), Shashank Sharath Karkera PID No - EU1194013 (11), Prithviraj Sanjay Patil PID No - EU1194011(29) and Nileshsingh Arjunsingh Mehta PID No – 1194052 (15) submitted to University of Mumbai in partial fulfilment of the requirement for the award of the degree of **"Information Technology Engineering"** during the academic year 2020-2021.

Ms. Shraddha More

Guide

Mrs. Anita Chaudhari

Head of Department

Dr. G.V. Mulgund

Principal

S.E. Mini-Project Report Approval

This mini-project synopsis entitled ***Inventory Management System*** by ***Prabal Nair, Shashank Karkera, Prithviraj Patil, Nileshsingh Mehta*** is approved for thedegreeof ***Information Technology Engineering*** from ***University of Mumbai***.

Examiners

1.----------

2.----------

Date: 29/05/2021

Place: Palghar

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken whenneeded.

Signature
Prabal Nair (EU1194018)

Signature
Shashank Karkera (EU1194013)

Signature
Prithviraj Patil (EU1194011)

Signature
Nileshsingh Mehta (EU1194052)

Date: 29/05/2021

Abstract

Inventory management system which is helpful for the business operators, where shopkeeper keep the records of purchase and sales. Inventory Management System is important to ensure quality control in businesses that handle transactions revolving around consumer goods. This inventory is eliminated paper work, human faults, manual delay and speed up process. Without proper inventory control, a large retail store may run out of stock on an important item. This inventory management system will have the ability to track sales and available inventory, tells a shopkeeper when it's time to reorder and how much to purchase. The objective is to develop an inventory management system to minimize the total inventory costs. The task of inventory management is to find the quantity of inventories that will fulfil the demand, avoiding overstocks. By implementing inventory management system to accelerate and smooth the physical flow of goods through a supply chain.

Table of Contents

Chapter 1

Introduction

Inventory management refers to the process of ordering, storing and using a company's inventory. This includes the management of raw materials, components and finished products, as well as warehousing and processing such items.An inventory system monitors the levels of inventory and determines the timeline and quantity of orders. Inventory Management is a technique through which stocked goods, inventories, and non-capitalized assets are kept in a proper manner according to their specific shape and placement.

1.1 Motivation

The main motivation behind building this system is to provide an efficient and user friendly application that improves product management and decreases time consumption. Transaction motive is the main objective of holding sufficient inventory that helps smooth operation of production and proper supply of goods to meet market demand. Inventory is hold according to the production capacity, demand, average lead time, ordering cost and carrying cost. A business firm should hold optimum level of inventory to ensure safety against unforeseen and unpredictable future situation such as strike, shutdown, shortage, delay on supply etc. So, sufficient volume of inventory provides safety against unpredictable events. To handle all these motives we are proposing this management application to make the work of user easier and much more efficient.

1.2 Problem Statement

- **Customer demand**:

  Customer demand is constantly shifting. Order strategies for core items, as well as technology to create and execute an inventory plan, can help compensate for changing demand.

- **Overstocking**:

  Keeping too much stock on hand can be as problematic as having too little. Overstock impacts business cash flow and leads to inventory-related problems, such as storage and loss.

- **Inventory Loss**:

  The loss of inventory due to spoilage due hacking or virus attack, damage or theft can be a supply chain problem. It requires identifying, tracking and measuring problem areas.

- **Increasing Competition**:

  Globalized supply chains are subject to unpredictable economic shifts and market forces that impact the competition for raw materials.

1.3 Objective

1. To ensure a continuous supply of materials and stock so that production should not suffer at the time of customers demand.
2. To avoid both overstocking and under-stocking of inventory.
3. To maintain the availability of materials whenever and wherever required in enough quantity.
4. To keep material cost under control as they contribute to reducing the cost of production.
5. To maintain a systematic record of inventory.

1.4 Scope

- **Manage Inventory**: Inventory management helps to manage the stock of the company. it provides proper details of the products what kind of raw material, what are the sizes we require and etc. to the purchasing department.
- **Less Storage**: When the inventory management provides proper information to management, they buy according to them which helps the company to store fewer products.
- **Improve Productivity:** Inventory management helps to improve the productivity of the machines and manpower. Employees are aware of stocks and the quantity that require to produce.
- **Increase Profits:** Inventory management helps to improve the profits of the company. it helps to provide proper information about stocks, that saves the unnecessary expenses on stocks.

Chapter 2

Review of Literature

Literature Survey of Inventory Management System

PAPER 1:-

Title- Inventory Management- A Review of Relevant Literature

Authors - V. Vijay Lakshmi, K. Ranganath

Description -

. This paper proposes an overview about the Inventory Managemment structure and how the system can be set up. This paper further deal with the problems associated with the Inventory Management and discusses how this problem can be solved.

Research Gaps –

1. High risk of malicious hacking
2. More time comsuming

PAPER 2:-

Title: Research paper on Inventory Management System

Authors - Punam Khobragade, Roshni Selokar, Rina Maraskolhe, Prof.Manjusha Talmale

Description –

This paper has discussed the technical aspects of an Inventory Management System lke the system requirements, features of the sytem, implementation of the system,

Research Gaps –

- Complex functionalities
- High cost implementation

Chapter 3

Requirements analysis

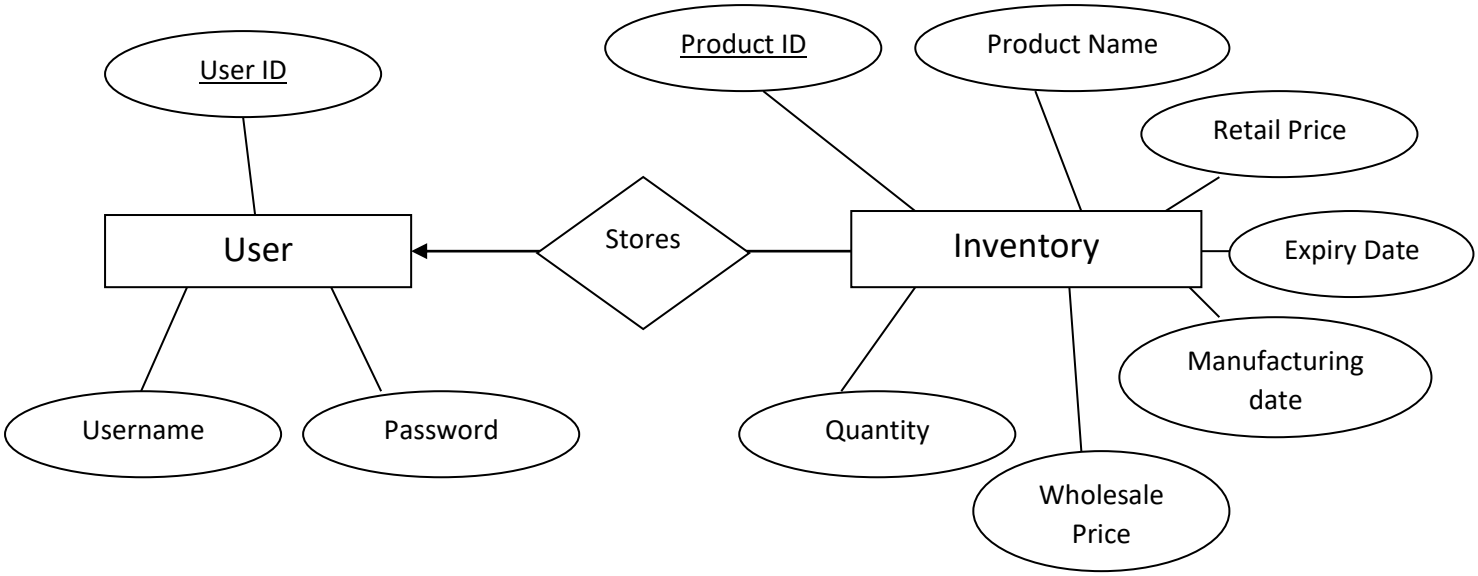
3.1 Hardware and Software Requirements

This system is developed using Python scripting language in PyCharm Community Edition 2020.3.5 IDE(Integrated Development Environment).

For creating database we have used MySQL Database.

In hardware, we have used Acer laptop with Windows 10, 64 Operating System, Intel core i3, 4GB RAM to run the softwares required.

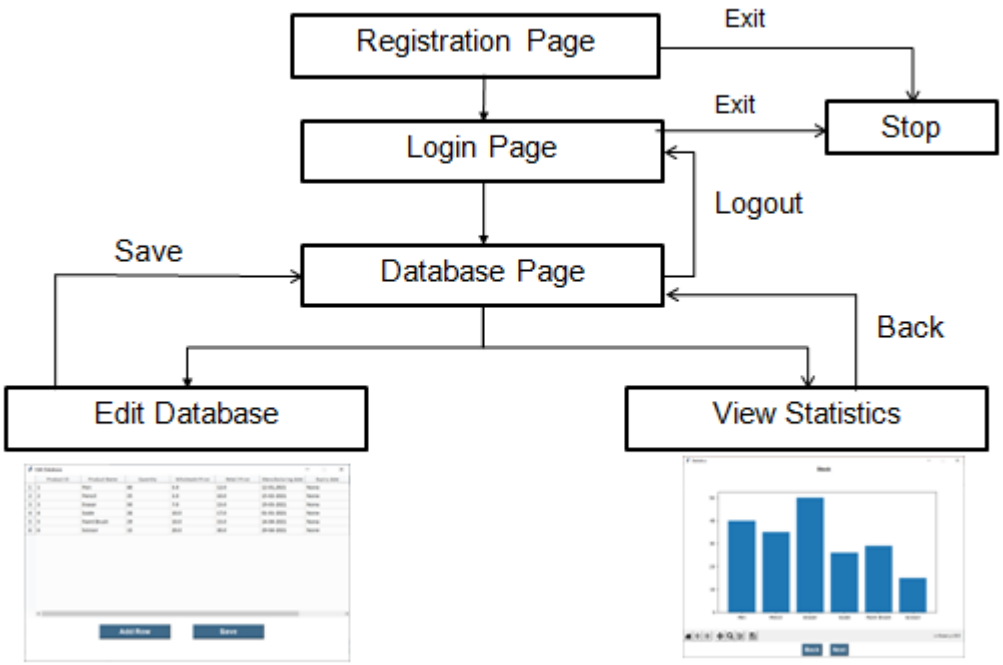
3.2 UML Diagram (ER digram)



Chapter 4

Report on Present Investigation

4.1 Proposed System



Registration Page: This is the page where users have set their username and password to access the system.

Login Page: In this page users have to enter their username and password to enter into the datanase

Database Page: This is the main page where the database will be displayed along with its data. This page will be having

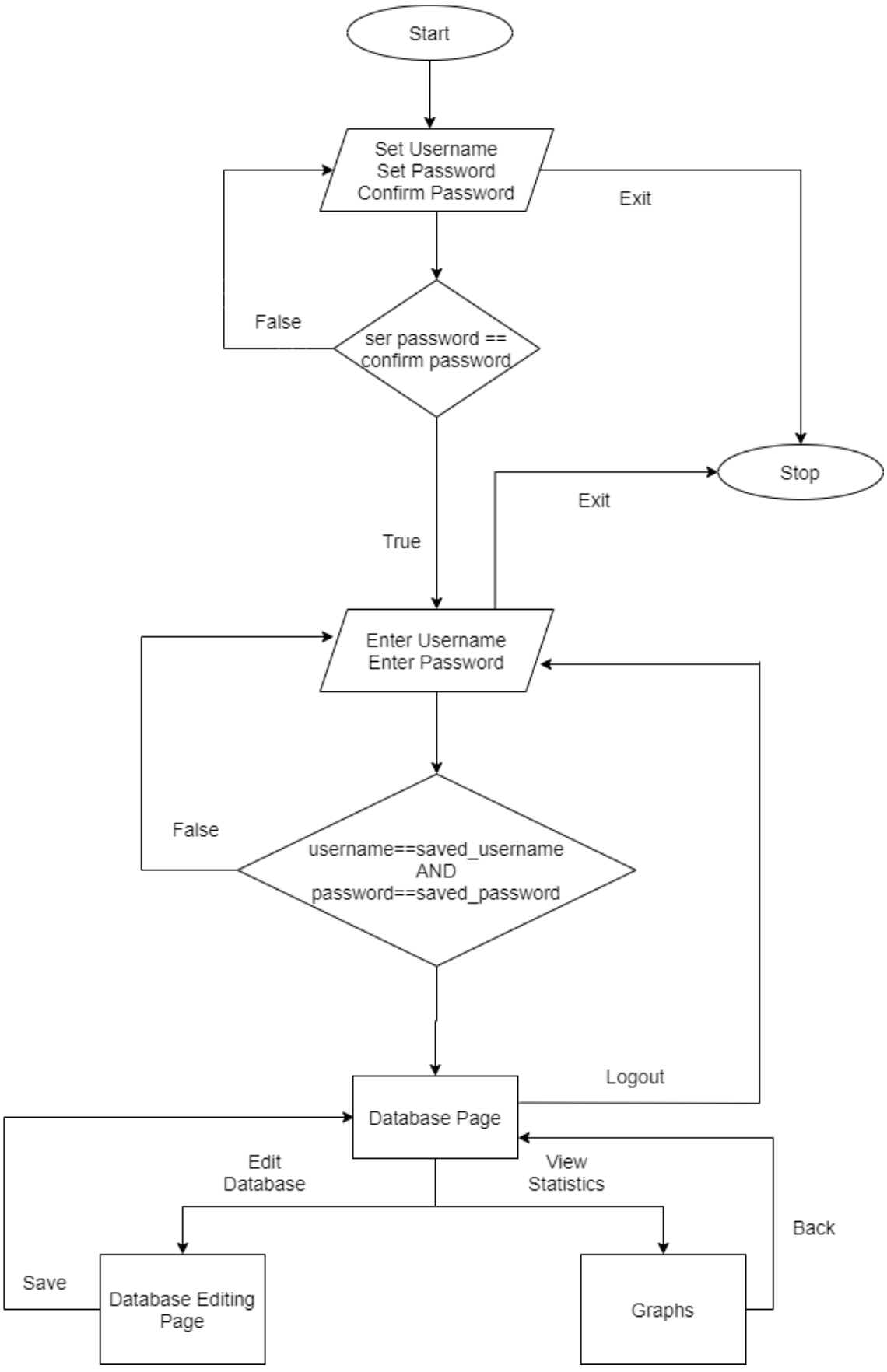
the options of search, edit database, view statistics and logout.

Edit Database Page: In this page user will be able to edit i.e. add, delete, modify, etc. his/her data with help of given options.

View Statistics: In this page user will be able see the data in visual manner i.e. in graph format.

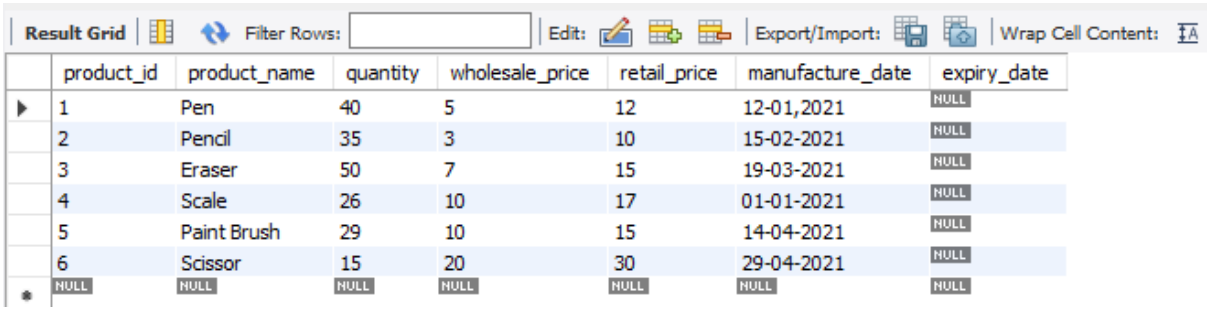
4.2 Implementation

4.2.1Algorithm / Flowchart



4.2.2 Dataset

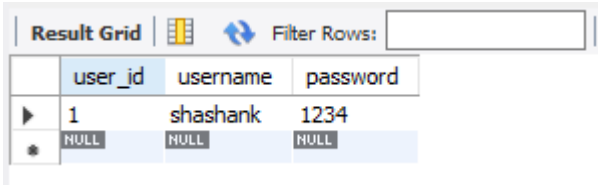
Inventory Data



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

product_id	product_name	quantity	wholesale_price	retail_price	manufacture_date	expiry_date
1	Pen	40	5	12	12-01,2021	NULL
2	Pencil	35	3	10	15-02-2021	NULL
3	Eraser	50	7	15	19-03-2021	NULL
4	Scale	26	10	17	01-01-2021	NULL
5	Paint Brush	29	10	15	14-04-2021	NULL
6	Scissor	15	20	30	29-04-2021	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL

User Data



Result Grid | Filter Rows:

user_id	username	password
1	shashank	1234
NULL	NULL	NULL

4.2.3 Pseudo Code

```
from tkinter import *
import mysql.connector
import tksheet
import numpy as np
from matplotlib.backends.backend_tkagg import (FigureCanvasTkAgg,
NavigationToolbar2Tk)
import matplotlib.figure as Figure
from tkinter import messagebox

my_db = mysql.connector.connect(host="localhost",
                                user="root",
                                password="root",
                                database="inventroy_management")
user_pass = ["a", "b"]
root_option = None
bt_place_x = 0
bt_place_y = 100
count = 0
root_db = None


# Registration Page
def registration_page():
    root_registration = Tk()
    root_registration.resizable(height=False, width=False)
    root_registration.title("Registration")
    root_registration.geometry("700x500")
    root_registration.config(bg="white")
    w_pos = int(root_registration.winfo_screenwidth() / 2 - 700 / 2)
    h_pos = int(root_registration.winfo_screenheight() / 2 - 500 /
2)
    root_registration.geometry("+{}+{}".format(w_pos, h_pos - 40))
    user_db = my_db.cursor()
    from tkinter import messagebox
    def getval(event):
        pass1 = set_pass_enrty.get()
        pass2 = confirm_pass_enrty.get()
        pass3 = username_enrty.get()
        if (pass1 == "" and pass2 == "") or (pass3 == ""):
            messagebox.showinfo("Error", "Kindly fill all the
fields")

        elif pass1 != pass2:
            messagebox.showinfo("Error", "Password did not match")

        else:
            print(user_pass)
            data = (pass3, pass2)
            user_db.execute("INSERT INTO user VALUES(1, %s, %s)",
data)
            my_db.commit()
            messagebox.showinfo("Success", "Successfully
Registered")
```

```
            root_registration.destroy()
            login_page()


    Frame(root_registration, bg="#3F6A8A", height=70,
width=700).place(x=0, y=0)
    Label(root_registration, text="Register", font=("arial 25
bold"), bg="#3F6A8A", fg="white").place(x=10, y=15)

    username = Label(root_registration, text="Username:",
font=("arial 13 bold"), bg="white")
    set_pass = Label(root_registration, text="Set Password:",
font=("arial 13 bold"), bg="white")
    confirm_pass = Label(root_registration, text="Confirm
Password:", font=("arial 13 bold"), bg="white")

    username.place(x=195, y=170)
    set_pass.place(x=165, y=220)
    confirm_pass.place(x=130, y=270)

    usernamevalue = StringVar
    set_passvalue = StringVar
    comfirm_passvalue = StringVar

    username_enrty = Entry(root_registration,
textvariable="usernamevalue", bd=2, font="15", width=15)
    set_pass_enrty = Entry(root_registration,
textvariable="set_passvalue", bd=2, font="15", width=15, show="*")
    confirm_pass_enrty = Entry(root_registration,
textvariable="comfirm_passvalue", bd=2, font="15", width=15,
show="*")

    username_enrty.place(x=290, y=170)
    set_pass_enrty.place(x=290, y=220)
    confirm_pass_enrty.place(x=290, y=270)

    Button(root_registration, text="Submit", font="arial 12 bold",
           command=lambda: [getval(None)], height=1,
           width=9, bg="lightblue").place(x=250, y=310)

    root_registration.bind('<Return>', getval)

    Label(root_registration, bg="#2C3E50", fg="white", height=50,
width=700).place(x=0, y=450)


    root_registration.mainloop()



# Login Page


def login_page():
    root_login = Tk()
    root_login.resizable(height=False, width=False)
    root_login.title("Login Page")
```

```
    root_login.geometry("700x500")
    root_login.config(bg="white")
    w_pos = int(root_login.winfo_screenwidth() / 2 - 700 / 2)
    h_pos = int(root_login.winfo_screenheight() / 2 - 500 / 2)
    root_login.geometry("+{}+{}".format(w_pos, h_pos))
    from tkinter import messagebox

    def ok(event):
        user_info = my_db.cursor()
        user_info.execute("SELECT * FROM user")
        data_user = user_info.fetchall()
        user = username_enrty.get()
        passw = password_enrty.get()
        for i in data_user:
            if user == "" and passw == "":
                messagebox.showinfo("Error", "Kindly fill all the
fields")

            elif user == i[1] and passw == i[2]:
                messagebox.showinfo("Success", "Logged in
Successful")
                root_login.destroy()
                database_page()
            else:
                messagebox.showinfo("Error", "Invalid details")

    Frame(root_login, bg="#3F6A8A", height=70, width=700).place(x=0,
y=0)
    Label(root_login, text="Login Page", font=("arial 25 bold"),
bg="#3F6A8A", fg="white").place(x=10, y=15)

    username = Label(root_login, text="Username:", font=("arial 15
bold"), bg="white")
    password = Label(root_login, text="Password:", font=("arial 15
bold"), bg="white")

    username.place(x=195, y=180)
    password.place(x=195, y=230)

    usernamevalue = StringVar
    passwordvalue = StringVar

    username_enrty = Entry(root_login, textvariable="usernamevalue",
font=15)
    password_enrty = Entry(root_login, textvariable="passwordvalue",
font=15)
    username_enrty.focus_set()

    username_enrty.place(x=310, y=185)
    password_enrty.place(x=310, y=235)

    password_enrty.config(show="*")

    Button(root_login, text="Login", font=("arial 12 bold"),
command=lambda:ok(None), bg="lightblue").place(x=280, y=270)
    root_login.bind('<Return>', ok)
```

```
    Label(root_login, bg="#2C3E50", fg="white", height=50,
width=700).place(x=0, y=450)

    root_login.mainloop()


# Database page code

def database_page():
    root_db = Tk()
    root_db.title("Database")
    root_db.geometry("800x600")
    root_db.config(bg="White")
    root_db.resizable(height=False, width=False)
    pos_wd = int(root_db.winfo_screenwidth()/2 - 700/2)
    pos_hg = int(root_db.winfo_screenheight()/2 - 500/2)
    root_db.geometry("+{}+{}".format(pos_wd, pos_hg-70))
    top_frame = Frame(root_db, bg="#3F6A8A", height=70, width=800)

    # Database Code starts from here
    table_sheet = tksheet.Sheet(root_db,
                                show_x_scrollbar=True,
                                show_y_scrollbar=False,
                                page_up_down_select_row=True,
                                startup_focus=True)
    table_sheet.enable_bindings('')
    headers = ("Product ID", "Product Name", "Quantity", "Wholesale
Price", "Retail Price", "Manufacturing date", "Expiry date")
    table_sheet.headers(headers)
    mydb = mysql.connector.connect(host="localhost",
                                   user="root",
                                   passwd="root",
                                   database="inventroy_management")

    my_cur = mydb.cursor()
    my_cur.execute("SELECT * FROM inventory")
    my_data = my_cur.fetchall()
    print(len(my_data))
    for i in my_data:
        table_sheet.set_sheet_data([[f"{i[c]}" for c in range(0,7)]
for r in range(len(my_data))])
    j=0
    for row in my_data:
        z=j
        print(z,j)
        table_sheet.set_row_data(z, values=row)
        j=j+1
    table_sheet.height_and_width(height=300, width=720)
    table_sheet.place(x=40, y=140)

    # Database code ends here

    def search(event):
        sr_val = search_entry.get()
        cure = my_db.cursor()
        cure.execute("SELECT * FROM inventory")
        sol = cure.fetchall()
```

```
        des = cure.description
        head_lst = []
        val_lst = []
        s = ""
        for j in des:
            for k in j[0]:
                if k=="_":
                    s=s+" "
                else:
                    s=s+k
            hd = s.capitalize()
            head_lst.append(hd)
            s = ""
        print(head_lst)
        pt = 0

        for i in sol:
            if i[1] == sr_val:
                pt=pt+1
                for m in i:
                    val_lst.append(m)

        if pt==0:
            messagebox.showinfo("Error", "No results")
        else:
            root_sr = Tk()
            root_sr.geometry("600x400")
            root_sr.config(bg="white")
            root_sr.title("Search Result")
            root_sr.resizable(width=False, height=False)
            sr_xpos = int(root_sr.winfo_screenwidth() / 2 - 600 / 2)
            sr_ypos = int(root_sr.winfo_screenheight() / 2 - 400 /
2)
            root_sr.geometry("+{}+{}".format(sr_xpos, sr_ypos))
            fr = Frame(root_sr, bg="#3F6A8A", width=600, height=50)
            fr2 = Frame(root_sr, bg="#3F6A8A", width=600, height=40)
            bt = Button(root_sr, text="Back", bg="#3F6A8A",
fg="white", font="arial 14 bold", width=15, command=root_sr.destroy)
            fr2.place(x=0, y=360)
            bt.place(x=200, y=300)
            fr.place(x=0, y=0)
            ct = 0
            dum_x = 50
            dum_y = 80
            for p in range(len(val_lst)):
                ct=ct+1
                print(len(val_lst))
                lb = Label(root_sr, text="{}:
{}".format(head_lst[p], val_lst[p]), bg="white", font="12")
                if(ct==1):
                    lb.place(x=dum_x, y=dum_y)
                elif(ct==5):
                    dum_y=80
                    lb.place(x=dum_x+200, y=dum_y)
                    dum_x=dum_x+200
                else:
                    lb.place(x=dum_x, y=dum_y+30)
```

```
                    dum_y=dum_y+30

            root_sr.mainloop()

    def edit_database():
        rooted_db = Tk()
        rooted_db.geometry("870x500")
        rooted_db.resizable(width=False, height=False)
        rooted_db.title("Edit Database")
        rooted_db.config(bg="white")
        table_sheet = tksheet.Sheet(rooted_db,
                                    show_x_scrollbar=True,
                                    show_y_scrollbar=False,
                                    page_up_down_select_row=True,
                                    startup_focus=True,
                                    table_selected_cells_bg="light
blue",
                                    )
        table_sheet.enable_bindings('single_select')
        table_sheet.enable_bindings('row_select')
        table_sheet.enable_bindings('edit_cell')
        table_sheet.enable_bindings('column_select')
        table_sheet.enable_bindings('rc_select')
        table_sheet.enable_bindings("right_click_popup_menu")
        table_sheet.enable_bindings("rc_delete_row")

        headers = ["Product ID", "Product Name", "Quantity",
"Wholesale Price", "Retail Price", "Manufacturing date",
                   "Expiry date"]
        table_sheet.headers(headers)
        mydb = mysql.connector.connect(host="localhost",
                                       user="root",
                                       passwd="root",

database="inventroy_management")
        my_db = mysql.connector.connect(host="localhost",
                                        user="root",
                                        passwd="root",

database="inventroy_management")

        my_cur = mydb.cursor()
        my_cur.execute("SELECT * FROM inventory")
        my_data = my_cur.fetchall()
        print(len(my_data))
        for i in my_data:
            table_sheet.set_sheet_data([[f"{i[c]}" for c in range(0,
7)] for r in range(len(my_data))])
        j = 0
        for row in my_data:
            z = j
            print(z, j)
            table_sheet.set_row_data(z, values=row)
            j = j + 1
        table_sheet.height_and_width(height=380, width=870)
        table_sheet.place(x=0, y=0)
```

```
        def get_data():
            d = table_sheet.get_sheet_data()
            for i in d:
                for k in i:
                    print(k)

        def add_row():
            table_sheet.insert_row(None, idx='end')

        def save():
            ins = mydb.cursor()
            lst = []
            d = table_sheet.get_sheet_data()
            '''
            ins.execute("TRUNCATE inventory")
            for i in d:
                for k in i:
                    lst.append(k)

                ins.execute("INSERT INTO inventory VALUES(%s, %s,
%s, %s, %s, %s, %s)", lst)
                mydb.commit()
                lst.clear()
        '''

        get_bt = Button(rooted_db, text="Save", bg="#3F6A8A",
fg="white", font="arial 14 bold", width=15,
command=lambda:[rooted_db.destroy(), database_page()])
        add_row = Button(rooted_db, text="Add Row", bg="#3F6A8A",
fg="white", font="arial 14 bold", width=15,
                         command=add_row)
        add_row.place(x=200, y=400)
        get_bt.place(x=450, y=400)
        rooted_db.mainloop()

    bottom_frame = Frame(root_db, bg="#2C3E50", height=50,
width=800)
    label_database = Label(top_frame, text="Database", bg="#3F6A8A",
font="arial 25 bold", fg="white")
    edit_button = Button(root_db, text="Edit Database",
bg="#3F6A8A", fg="white", font="arial 14 bold", width=15,
command=lambda:[root_db.destroy(), edit_database()])
    logout_button = Button(root_db, text="Logout", bg="#3F6A8A",
fg="white", font="arial 14 bold", width=15, command=lambda
:[root_db.destroy(),login_page()])
    view_button  = Button(root_db, text="View Statistics",
bg="#3F6A8A", fg="white", font="arial 14 bold", width=15,
command=stats)
    search_variable = StringVar()
    search_entry = Entry(root_db, width=25, bd=3, font="10",
textvariable=search_variable)
    sr_bt = Button(root_db, text="Search",
command=lambda:[search(None)])
    root_db.bind('<Return>', search)
    sr_bt.place(x=445, y=89)
    search_entry.place(x=500, y=90)
    view_button.place(x=300, y=480)
```

```
    logout_button.place(x=530, y=480)
    edit_button.place(x=70, y=480)

    label_database.place(x=10, y=15)
    bottom_frame.place(x=0, y=550)
    top_frame.place(x=0, y=0)
    root_db.mainloop()


# Button Functions Below


def stats():
    root_stats = Tk()
    root_stats.title("Statistics")
    root_stats.config(bg="white")
    root_stats.resizable(width=FALSE, height=False)
    st_w = int(root_stats.winfo_screenwidth()/2 -
root_stats.winfo_reqwidth()/2)
    st_h = int(root_stats.winfo_screenheight()/2 -
root_stats.winfo_reqheight()/2)
    root_stats.geometry("+{}+{}".format(st_w-250, st_h-250))
    fr = Frame(root_stats, bg="white", height=50,
width=root_stats.winfo_reqwidth())
    lb = Label(root_stats, text="Stock", font="arial 12 bold",
bg="white").pack()
    bt = Button(fr, text="Back", bg="#3F6A8A", font="arial 14 bold",
fg="white", command=lambda:[root_stats.destroy()]).place(x=30, y=5)
    bt2 = Button(fr, text="Next", bg="#3F6A8A", font="arial 14
bold", fg="white",
command=lambda:[root_stats.destroy(),stats2()]).place(x=120, y=5)
    fr.pack(side=BOTTOM)
    dbs = mysql.connector.connect(host="localhost",
                                  user="root",
                                  password="root",
                                  database="inventroy_management")
    curse = dbs.cursor()
    curse.execute("SELECT product_name, quantity FROM inventory")
    rs = curse.fetchall()
    lst1 = []
    lst2 = []
    for i in rs:
        for j in range(len(i)):
            if j==0:
                lst1.append(i[j])
            else:
                lst2.append(i[j])
    fg = Figure.Figure(figsize=(9,5), dpi=100)
    # y=[i**2 for i in range(101)]
    plt = fg.add_subplot(111)
    x = np.array([])
    y = np.array([])
    arr1 = np.append(x, lst1)
    arr2 = np.append(y, lst2)
    plt.bar(arr1,arr2)
    canvas = FigureCanvasTkAgg(fg, master=root_stats)
    canvas.draw()
```

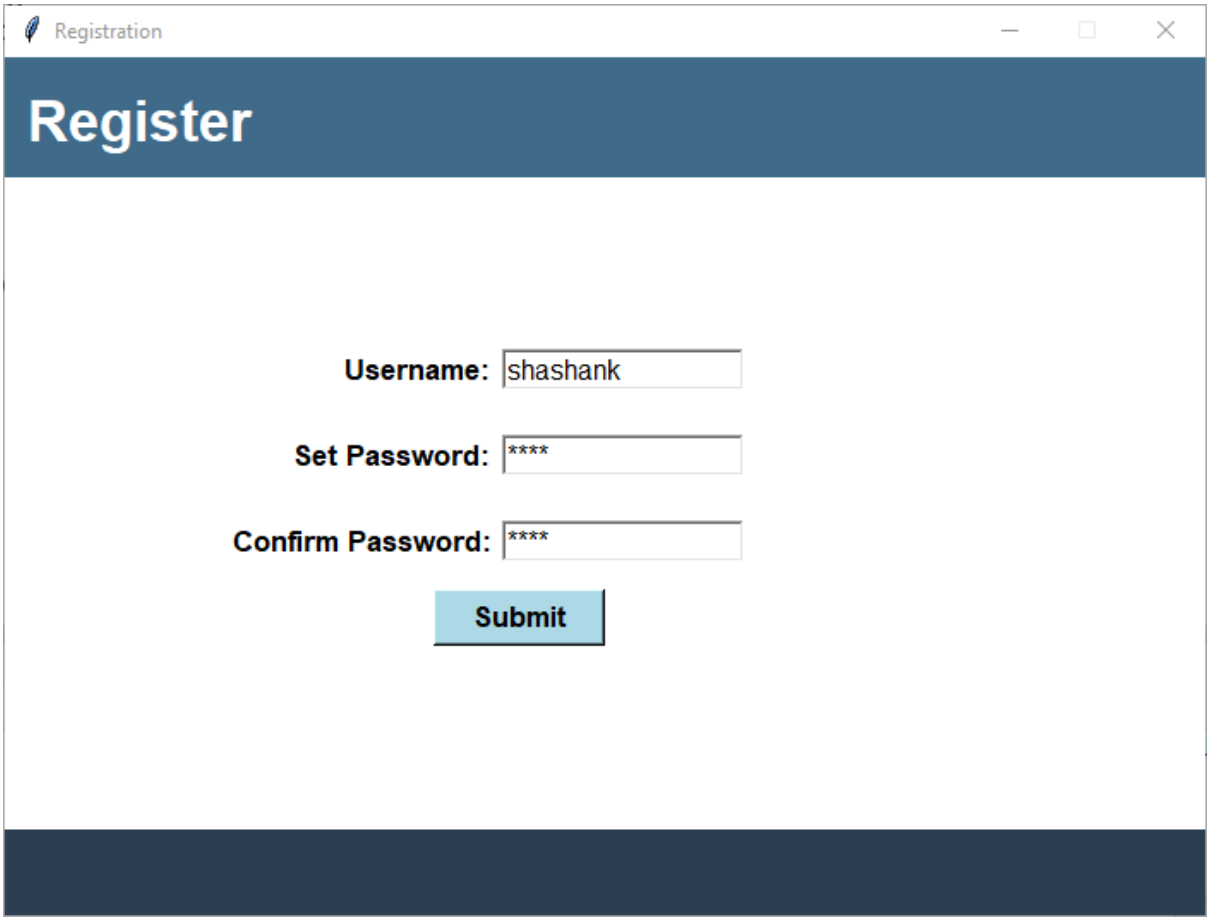
```
    canvas.get_tk_widget().pack()
    toolbar = NavigationToolbar2Tk(canvas, root_stats)
    toolbar.update()
    canvas.get_tk_widget().pack()
    root_stats.mainloop()



def stats2():
    root_st2 = Tk()
    root_st2.title("Statistics")
    root_st2.config(bg="white")
    root_st2.resizable(width=FALSE, height=False)
    st_w = int(root_st2.winfo_screenwidth() / 2 -
root_st2.winfo_reqwidth() / 2)
    st_h = int(root_st2.winfo_screenheight() / 2 -
root_st2.winfo_reqheight() / 2)
    root_st2.geometry("+{}+{}".format(st_w-250, st_h-250))
    lb = Label(root_st2, text="Profit per Product", font="arial 12
bold", bg="white").pack()
    bt = Button(root_st2, text="Back", bg="#3F6A8A", font="arial 14
bold", fg="white", command=lambda:[root_st2.destroy(),
stats()]).pack(side=BOTTOM)
    dbs = mysql.connector.connect(host="localhost",
                                  user="root",
                                  password="root",
                                  database="inventroy_management")
    curse = dbs.cursor()
    curse.execute("SELECT product_name, wholesale_price,
retail_price FROM inventory")
    rs = curse.fetchall()
    lst1 = []
    lst2 = []
    lst3 = []
    for i in rs:
        for j in range(len(i)):
            if j == 0:
                lst1.append(i[j])
            elif j==1:
                lst2.append(i[j])
            else:
                lst3.append(i[j])

    fin_lst = []
    zip_obj = zip(lst3, lst2)
    for a, b in zip_obj:
        fin_lst.append(a-b)
    print(fin_lst)
    fg = Figure.Figure(figsize=(10, 5), dpi=100)
    y = [i ** 2 for i in range(101)]
    plt = fg.add_subplot(111)
    x = np.array([])
    y = np.array([])
    arr1 = np.append(x, lst1)
    arr2 = np.append(y, fin_lst)
    plt.plot(lst1, fin_lst, marker='o')
    plt.grid()
```

```
    canvas = FigureCanvasTkAgg(fg, master=root_st2)
    canvas.draw()
    canvas.get_tk_widget().pack()
    toolbar = NavigationToolbar2Tk(canvas, root_st2)
    toolbar.update()
    canvas.get_tk_widget().pack()
    root_st2.mainloop()


my_cu = my_db.cursor()
my_cu.execute("SELECT * FROM user")
my_rs = my_cu.fetchall()
'''
for i in my_rs:
    print(i)
'''

if len(my_rs)==0:
    registration_page()
else:
    login_page()
```

Chapter 5

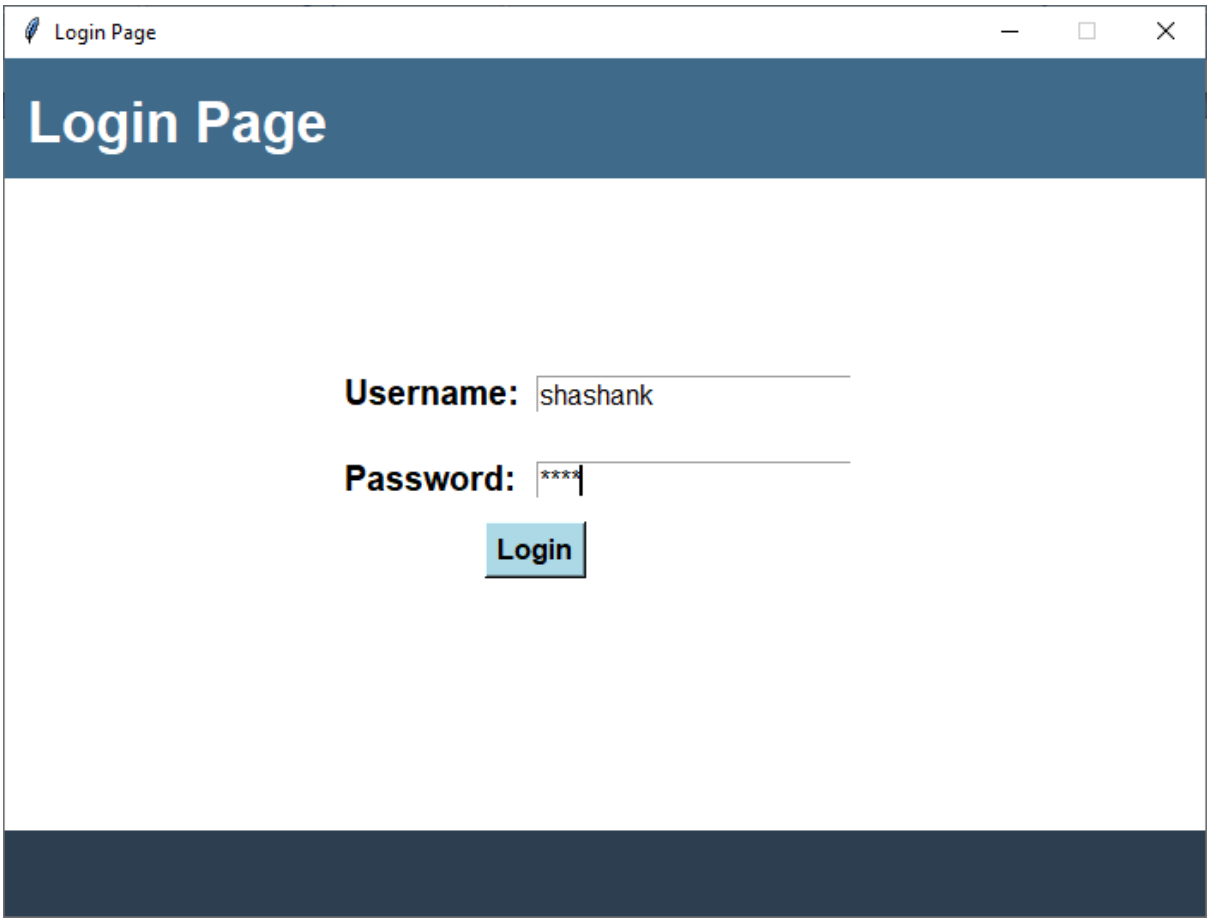
Results and Discussion

Registration Page



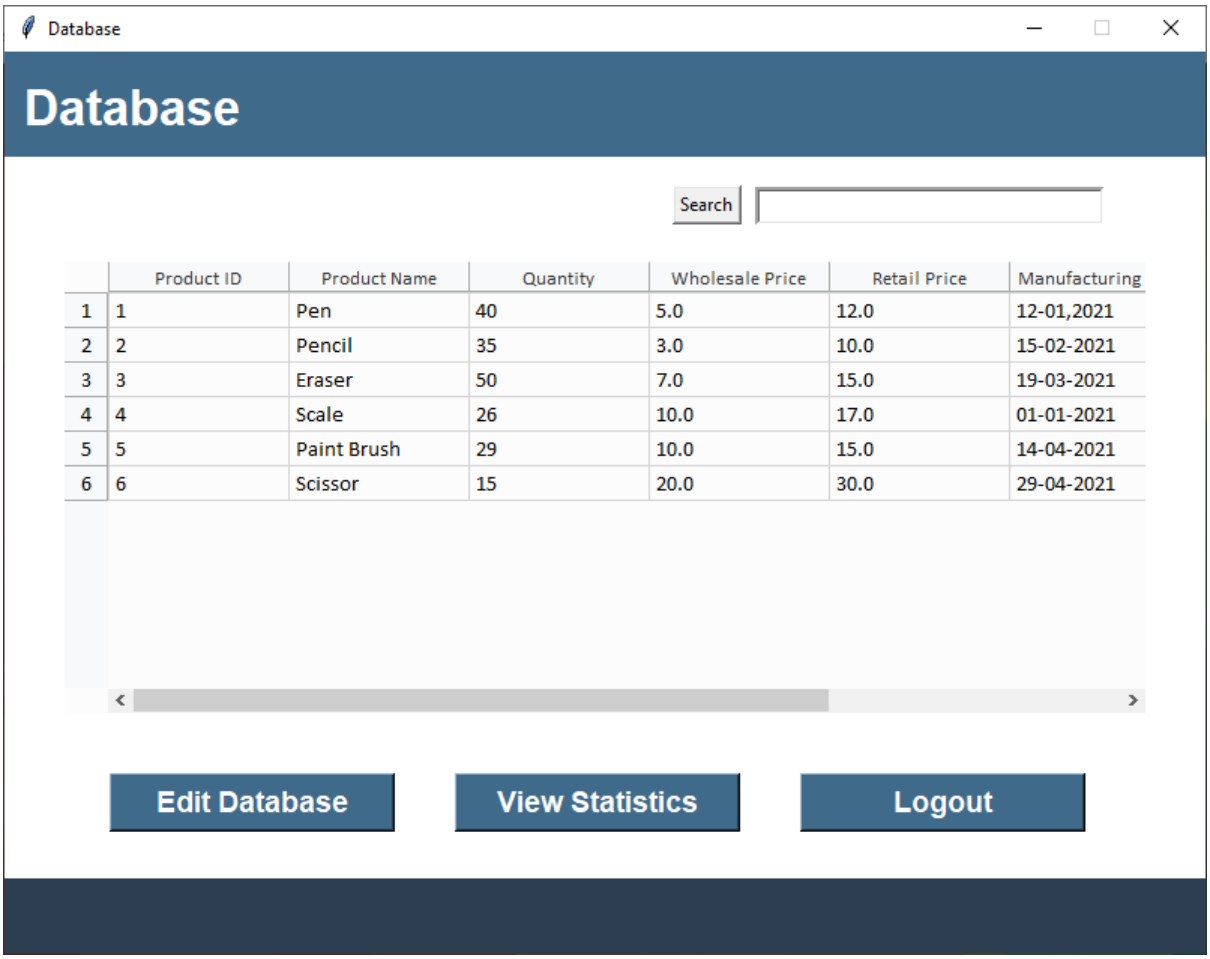
The first page is the registration page. In this page the users have to set their username and password which they later use to access the database. This page will only appear once for the purpose of registration. Later on, this won't be shown again after starting the application.

Login Page



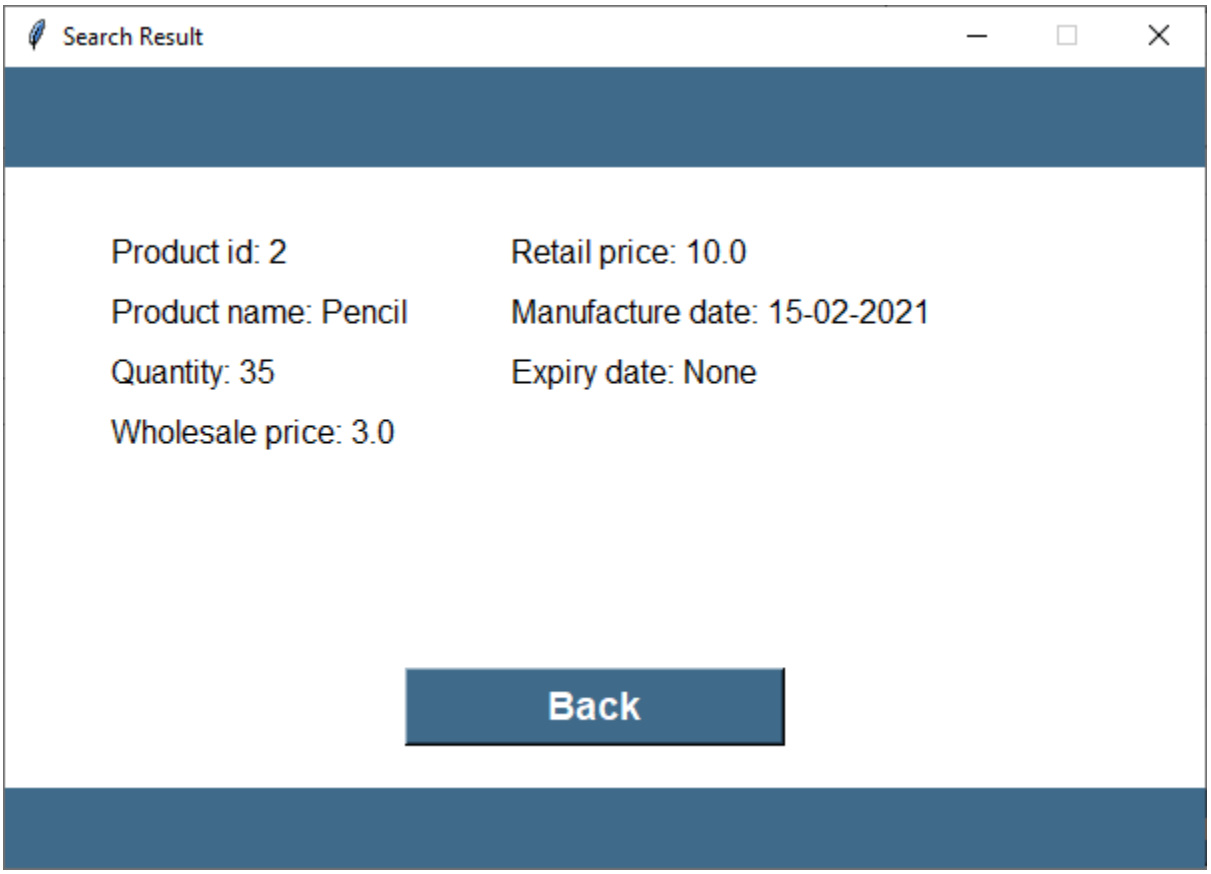
In the login page, the user has to enter the previously saved username and password into the given fields and press login. If the username or the password is incorrect than a message box with error will be shown.

Database Page



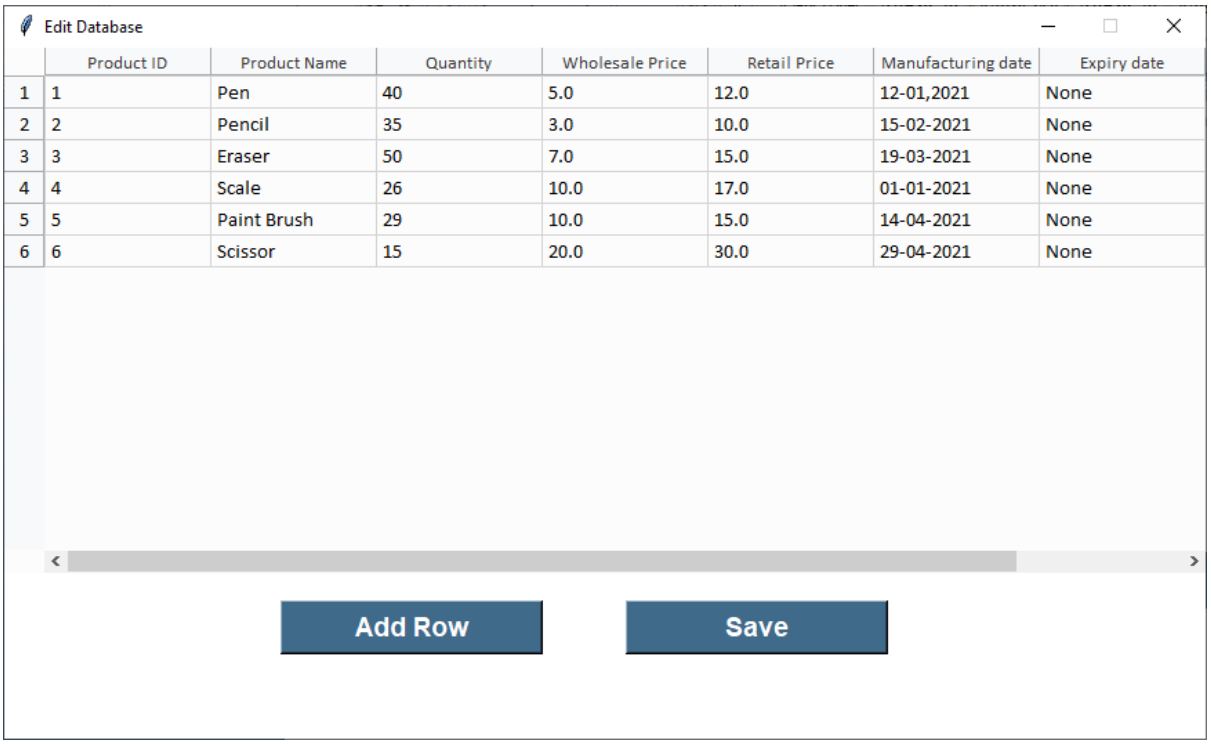
This is the main page where the database will be displayed. Firsttly, the user will be able to see the data in a tabular format at the centre of the screen. Next, the user will see a search bar where he/she will be able to search for any product from the table just by typing its name in the given field and press ‘Search’. To exit the database page the user has to click logout button at the bottom and he/she will be directed to login page.

Search Result Page



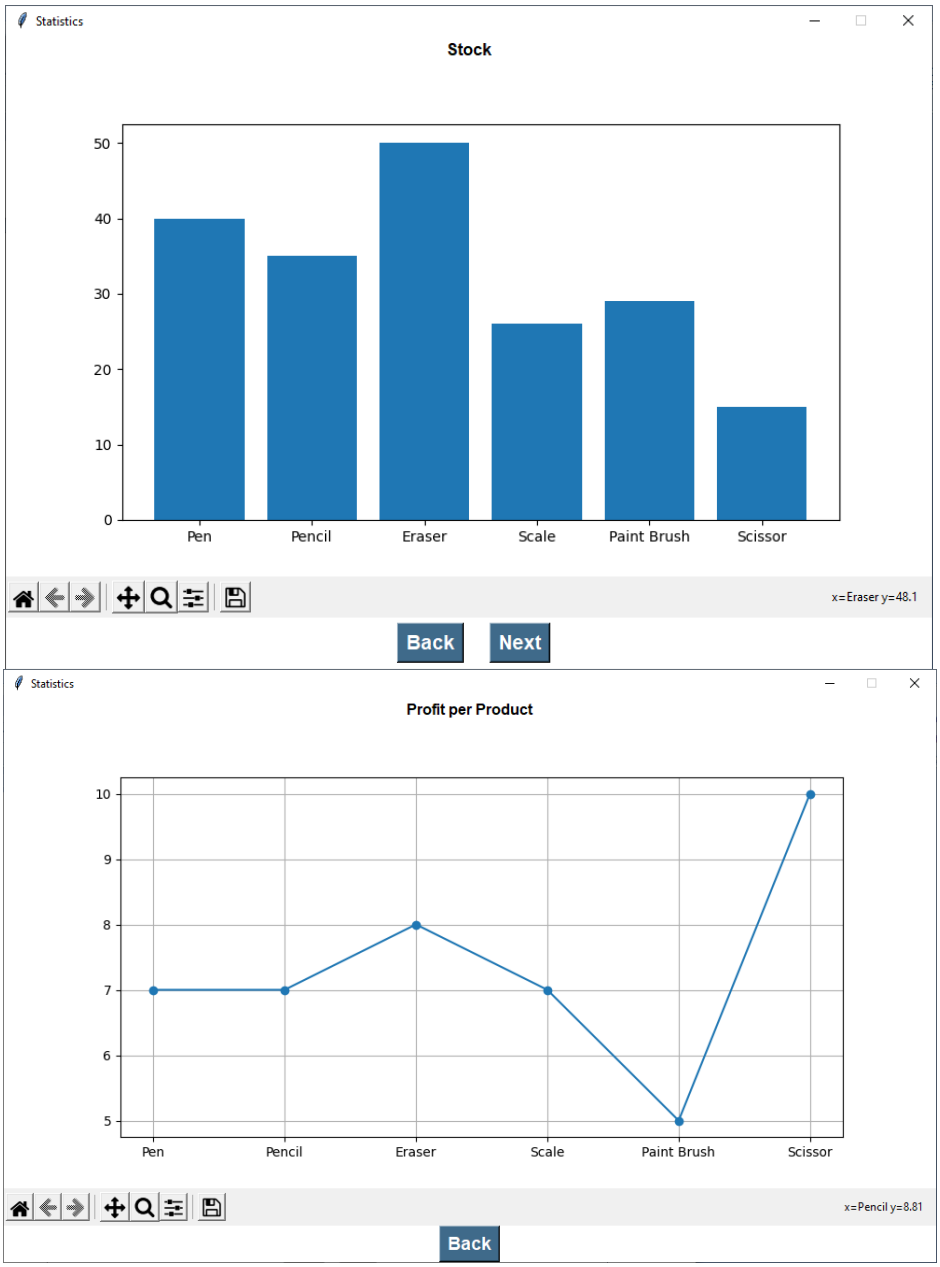
This page will show the you the details of a single product of which you entered the name in the search box at the database page.

Edit Database Page



In this page, the user will be able to edit the data directly on the table shown on the screen. He/She can even add rows to add new data into the table. To delete some data they can just right vlivk on the that particular row. Then by clicking save you edited database will be saved and you directed to edit database page

View Statistics Page



In this page, the user will be able to see the data in a visual manner i.e. in the form of graps. For example, in the above pages the user is able to see the stock of his products in bar graph form. In the next graph the user is able to profit per product statistics.

Chapter 6

Conclusion

Inventory Management System is a very useful and efficient tool that can be used to save time and money in many shop and industries**.** It will help in providing an easy way to manage inventories and help conserve a lot of time and manpower.

References

1. Inventory Management- A Review of Relevant Literature
   By, V. Vijay Lakshmi, K. Ranganath

2. Research paper on Inventory Management System
   By, Punam Khobragade, Roshni Selokar, Rina Maraskolhe, Prof.Manjusha Talmale

Websites:

https://www.w3schools.com/python/python_mysql_getstarted.asp

https://github.com/ragardner/tksheet/blob/master/DOCUMENTATION.md

https://www.geeksforgeeks.org/python-gui-tkinter/

https://www.activestate.com/resources/quick-reads/how-to-display-data-in-a-table-using-tkinter/

Acknowledgment

I would like to express my sincere gratitude to my teacher Ms. Shraddha More for giving proper direction to me and my team while working on our project. She inspired and motivated us. We would also like to thank our colleagues who helped us throughout the making of the project.