Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
(Religious Jain Minority)

A MINI PROJECT REPORT

On

SHOP MANAGEMENT SYSTEM

Submitted in partial fulfillment of the requirement of
University of Mumbai for the Course

In

COMPUTER ENGINEERING (IV SEM)

Submitted By
Sanket Nehe(Roll No:-26)
Pratik Patil (Roll No:- 42)
Gautam Pandey(Roll No:-32)

Subject Incharge
Prof. Merlin Priya Jacob

Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
(Religious Jain Minority)

CERTIFICATE

This is to certify that the requirements for the project report entitled '**SHOP MANAGEMENT SYSTEM**' have been successfully completed by the following students:

Name	Moodle Id
Gautam Pandey	21102117
Sanket Nehe	21102119
Pratik Patil	21102099

In partial fulfilment of the course Skill Base Lab Course: Python Programming (CSL405) in Sem: IV of Mumbai University in the Department of Computer Engineering during academic year 2022-2023.

Sub-in-Charge

Parshvanath Charitable Trust's

A. P. SHAH INSTITUTE OF TECHNOLOGY

(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)

(Religious Jain Minority)

PROJECT APPROVAL

The project entitled '**SHOP MANAGEMENT SYSTEM**' by **Gautam Pandey, Sanket Nehe and Pratik Patil** are approved for the course of Skill Base Lab Course: Python Programming (CSL405) in Sem: IV of Mumbai University in the Department of Computer Engineering.

Subject-in-Charge

Date:

Place: Thane

List Of Figures

Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
(Religious Jain Minority)

ABSTRACTION

A shop management system is an essential tool for any business that aims to keep track of its inventory, sales, and customers. With the use of modern technology, shop management systems can now be designed with a graphical user interface (GUI) to make it more user-friendly and efficient. The implementation of a shop management system using the tkinter library in Python provides a simple yet effective solution for small businesses. The system can allow the user to add, delete, search, and display items in the shop inventory through an intuitive interface. The user can easily input new items and their corresponding details, such as price, quantity, and description, and store them in a database. The system can also retrieve and display the items in a visually pleasing and organized way. The ability to delete items from the inventory through the GUI can save the user time and effort compared to manually editing the database. The search function can also help the user find specific items quickly and efficiently. Moreover, the system can provide the user with the total value of the inventory, which can aid in decision-making and restocking. Overall, the implementation of a shop management system using the tkinter library in Python is a practical solution for small businesses looking to organize their inventory and sales data. Its user-friendly GUI and efficient functions make it an effective tool for managing a shop's inventory.

Problem Definition:-

A shop management system is a software application designed to help manage various aspects of a retail business, such as inventory, sales, customer data, and employee management.
The problem definition for a shop management system should outline the specific needs and requirements of the business in order to create an efficient and effective solution. The following are some key elements that should be included in the problem definition:

Inventory management: The system should be able to track inventory levels, monitor stock movements, and generate alerts when stock levels are low.

Sales management: The system should be able to process sales transactions, track sales trends, and generate reports on sales performance.

Customer management: The system should be able to store and manage customer information, including contact details, purchase history, and loyalty program information.

Employee management: The system should be able to manage employee schedules, track time and attendance, and generate payroll reports.

Reporting and analytics: The system should be able to generate reports and provide insights into key business metrics, such as sales, inventory levels, and customer behavior.

Integration: The system should be able to integrate with other systems, such as accounting software, to streamline business processes and improve efficiency.

By defining the specific needs and requirements of the business, a shop management system can be developed to improve operational efficiency, increase sales, and enhance the overall customer experience.

Introduction:-

A shop management system is a software application designed to help retailers manage their businesses more effectively. With the rise of e-commerce and omnichannel retail, it has become increasingly important for businesses to have efficient and streamlined systems in place to manage their operations.

A shop management system can help businesses automate and optimize key processes such as inventory management, sales processing, customer management, and employee management. By doing so, it can help businesses reduce costs, increase productivity, and improve the overall customer experience.

The benefits of a shop management system are numerous. By automating manual processes, businesses can reduce the likelihood of errors and improve accuracy. By providing real-time dataand analytics, businesses can make better-informed decisions about inventory levels, sales trends,and customer behavior. And by streamlining operations, businesses can improve their bottom lineand provide better service to their customers.

Overall, a shop management system is a powerful tool for retailers looking to stay competitive in a rapidly evolving market. Whether you're a small business owner looking to streamline your operations or a larger enterprise seeking to optimize your processes, a shop management system can help you achieve your goals and drive your business forward.

Description of modules used:

The code uses the tkinter module which is a standard GUI library in Python. It provides classes to build graphical user interface applications. The following modules are used in the code:

Tk: The Tk class is the root window of the application. It is used to create a window to hold the widgets of the application.

Label: The Label class is used to create a static text widget that is used to display information.

Entry: The Entry class is used to create an input field where the user can enter text.

Button: The Button class is used to create a button widget that the user can click to trigger an action.

Listbox: The Listbox class is used to create a list widget that is used to display a list of items.

Toplevel: The “Toplevel” class is used to create a new top-level window.The code also defines a class ShopManager which has methods to add, delete, search, and display items in the shop inventory.

The “_init_” method sets up the main window, the labels, the entry fields, the buttons, and the list box. The “add_item” method adds a new item to the inventory and displays it in the list box. The “delete_item” method deletes the selected item from the list box. The “search_item” methodsearches for an item by name and selects it in the list box. The “display_inventory” method displays the inventory in a new window. Finally, the “root.mainloop()” method starts the GUI event loop to handle user input and update the GUI.

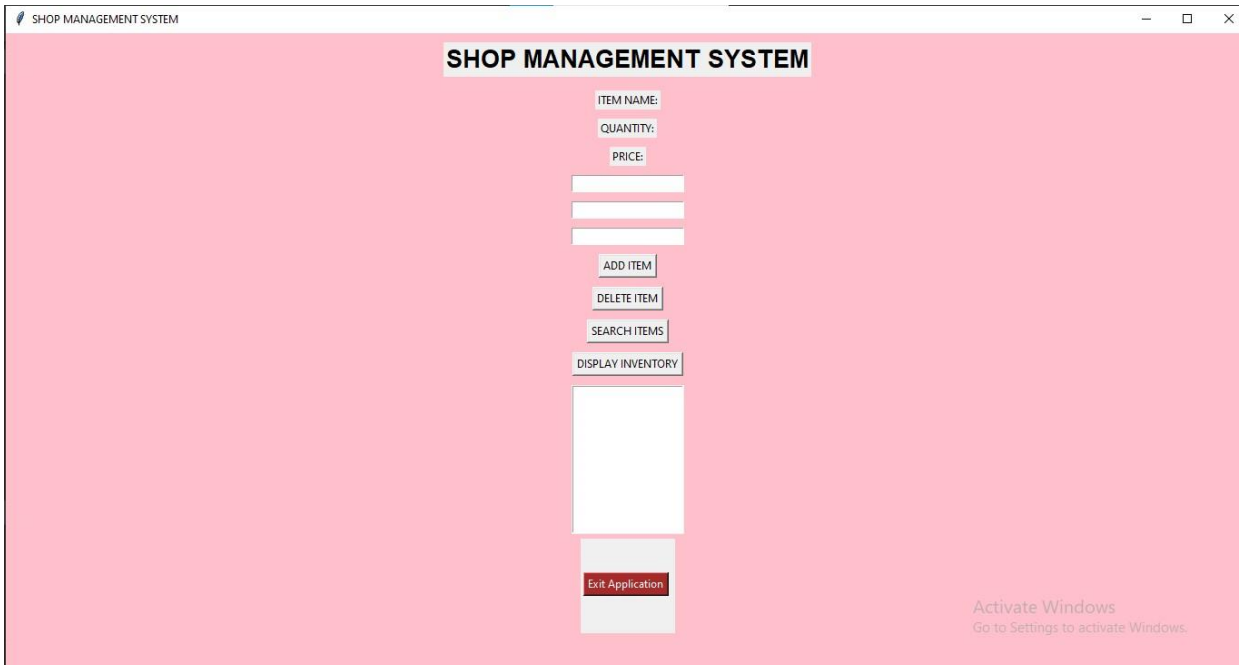
IMPLEMENTATION DETAILS:

```
from tkinter import *
import tkinter as tk
from tkinter import messagebox
class ShopManager:
    def __init__(self, master):
        self.master = master
        self.master.title("SHOP MANAGEMENT SYSTEM")
        self.master.geometry("1350x1000")

        # Create Labels
        self.lbl_title = Label(master, text="SHOP MANAGEMENT SYSTEM",
font=("Arial", 20, "bold"))
        self.lbl_title.pack(pady=10)

        self.lbl_name = Label(master, text="ITEM NAME:")
        self.lbl_name.pack(pady=5)

        self.lbl_qty = Label(master, text="QUANTITY:")
        self.lbl_qty.pack(pady=5)

        self.lbl_price = Label(master, text="PRICE:")
        self.lbl_price.pack(pady=5)

        # Create Entry Fields
        self.ent_name = Entry(master)
        self.ent_name.pack(pady=5)

        self.ent_qty = Entry(master)
        self.ent_qty.pack(pady=5)

        self.ent_price = Entry(master)
        self.ent_price.pack(pady=5)

        # Create Buttons
        self.btn_add = Button(master, text="ADD ITEM", command=self.add_item)
        self.btn_add.pack(pady=5)

        self.btn_delete = Button(master, text="DELETE ITEM",
command=self.delete_item)
        self.btn_delete.pack(pady=5)

        self.btn_search = Button(master, text="SEARCH ITEMS",
command=self.search_item)
```

```
        self.btn_search.pack(pady=5)

        self.btn_display = Button(master, text="DISPLAY INVENTORY",
command=self.display_inventory)
        self.btn_display.pack(pady=5)

        # Create Listbox
        self.lst_items = Listbox(master)
        self.lst_items.pack(pady=5)

    # Add Item Function
    def add_item(self):
        name = self.ent_name.get()
        qty = self.ent_qty.get()
        price = self.ent_price.get()

        # Add item to Listbox
        self.lst_items.insert(END, f"{name} ({qty}): ${price}")

        # Clear Entry Fields
        self.ent_name.delete(0, END)
        self.ent_qty.delete(0, END)
        self.ent_price.delete(0, END)

    # Delete Item Function
    def delete_item(self):
        selection = self.lst_items.curselection()
        if selection:
            self.lst_items.delete(selection)

    # Search Item Function
    def search_item(self):
        name = self.ent_name.get()

        for i in range(self.lst_items.size()):
            if name in self.lst_items.get(i):
                self.lst_items.selection_clear(0, END)
                self.lst_items.selection_set(i)
                self.lst_items.activate(i)
                break

    # Display Inventory Function
    def display_inventory(self):
        inventory = []
```

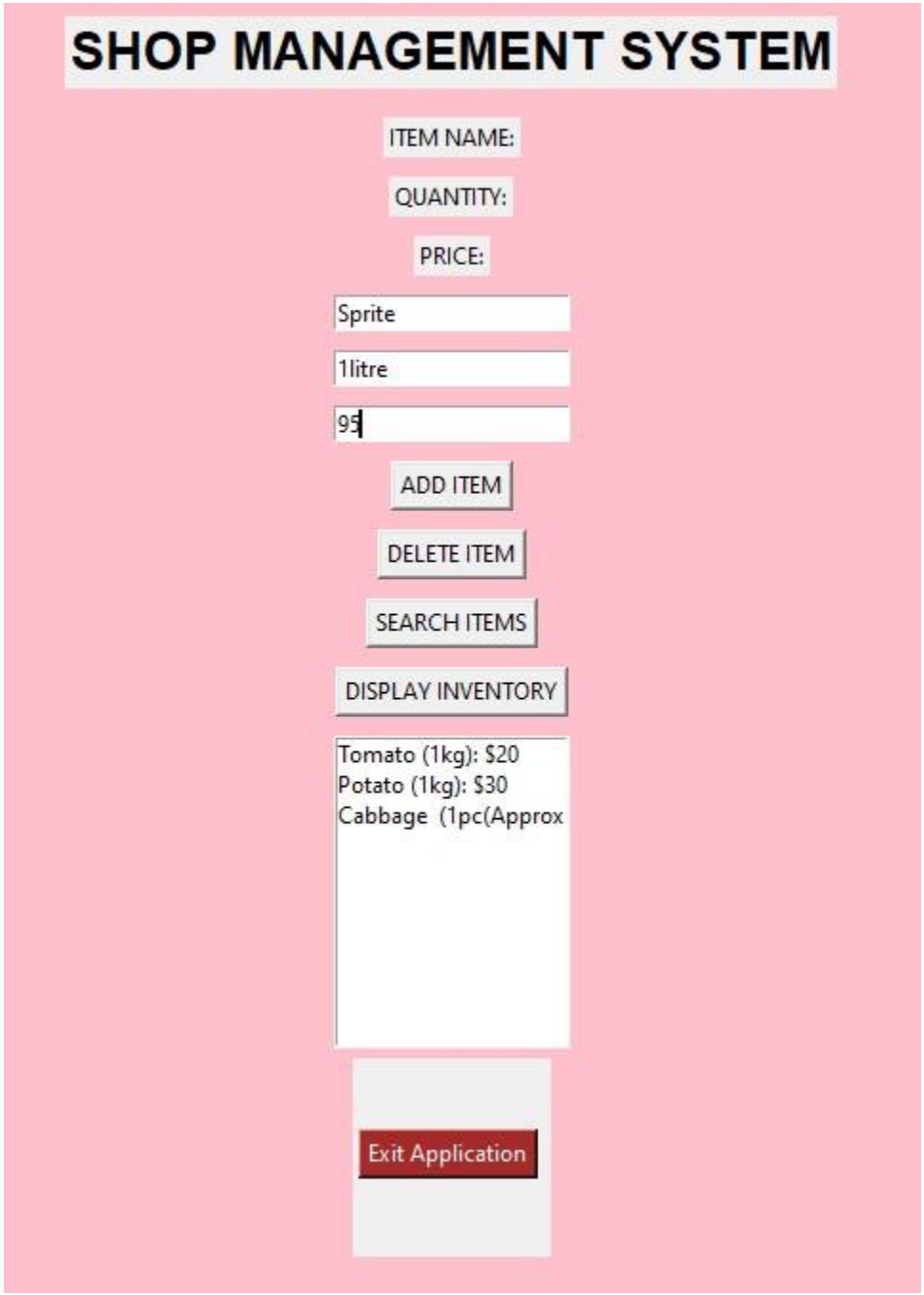
```
        for i in range(self.lst_items.size()):
            inventory.append(self.lst_items.get(i))

        # Show Inventory in a New Window
        new_window = Toplevel(self.master)
        new_window.title("SHOP INVENTORY")
        new_window.geometry("500x500")

        lbl_inventory = Label(new_window, text="\n".join(inventory),
font=("Arial", 12))
        lbl_inventory.pack(pady=10)

# Create Tkinter App
root = Tk()
app = ShopManager(root)
root.configure(bg='pink')

#exit
canvas1 = tk.Canvas(root, width=100, height=100)
canvas1.pack()
def exit_application():
    msg_box = tk.messagebox.askquestion('Exit Application', 'Are you sure you
want to exit the application?',icon='warning')
    if msg_box == 'yes':
        root.destroy()
    else:
        tk.messagebox.showinfo('Return', 'You will now return to the application
screen')

button1 = tk.Button(root, text='Exit Application', command=exit_application,
bg='brown', fg='white')
canvas1.create_window(50,50, window=button1)
root.mainloop()
```

This is Actual Implementation's Output of Our Program.

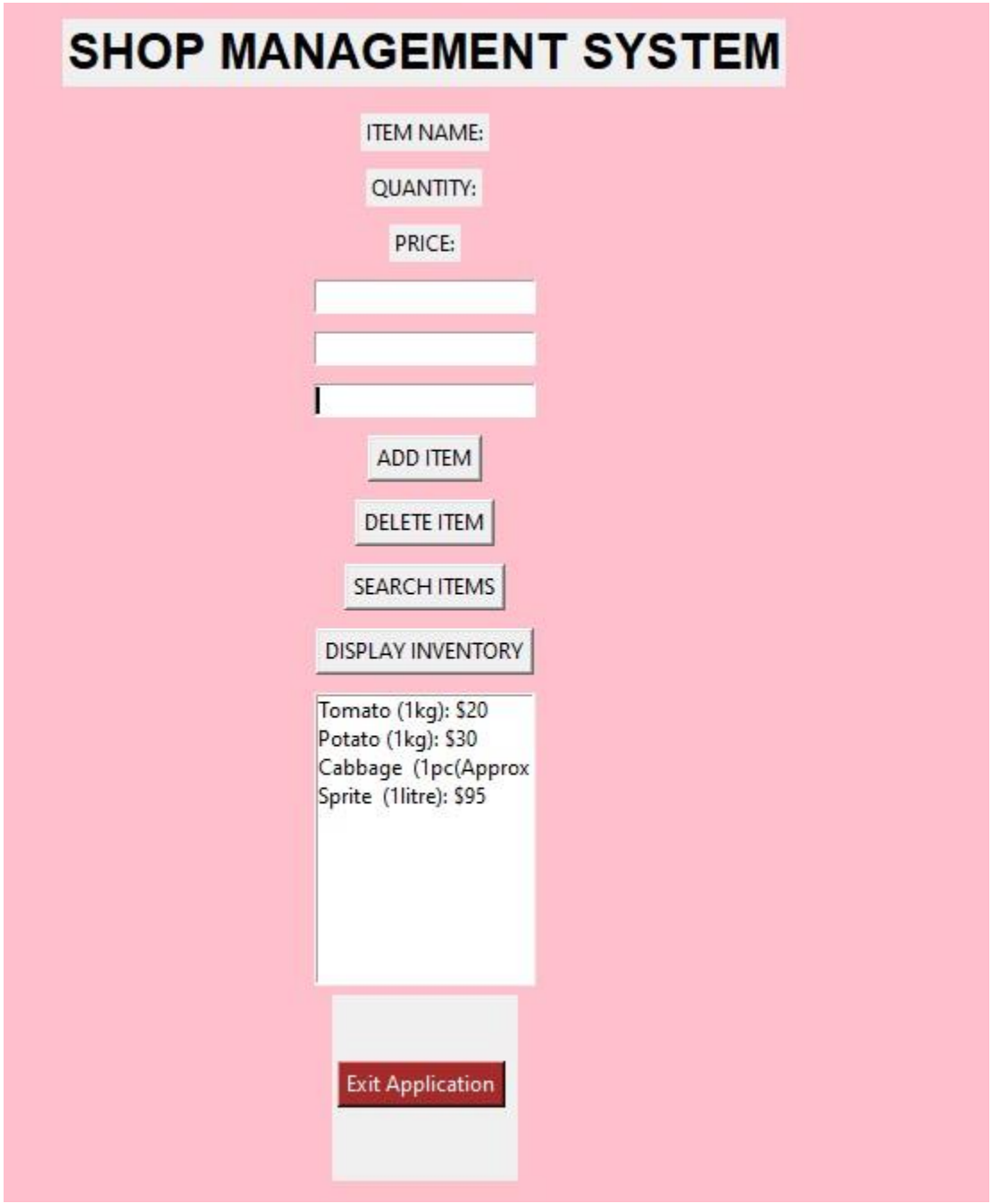


ADD_ITEM:

Adding Item name, Quantity, Price by Using ADD ITEM.

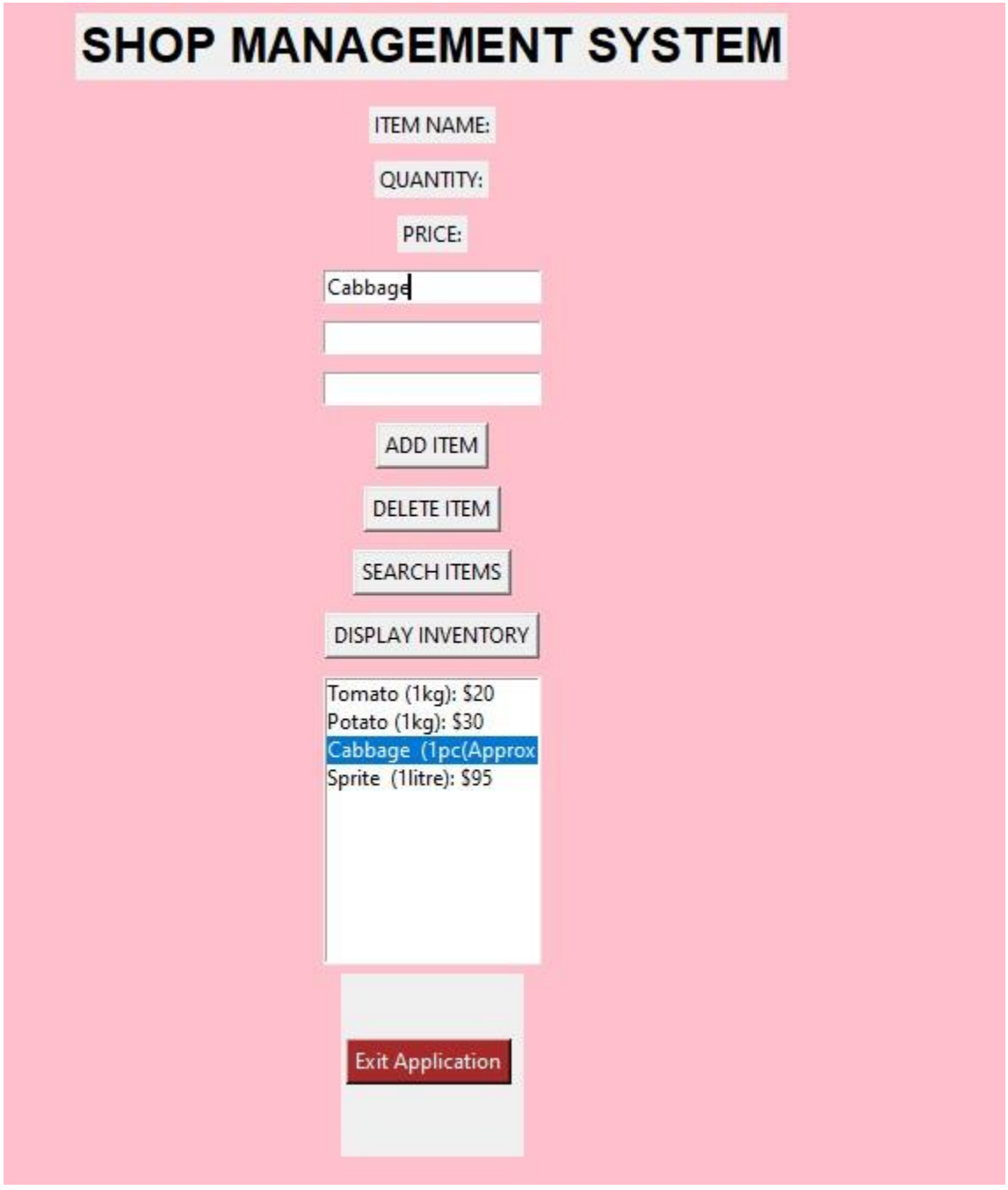


After adding Item name As Sprite we got output as below in which Sprite got added into the inventory list.



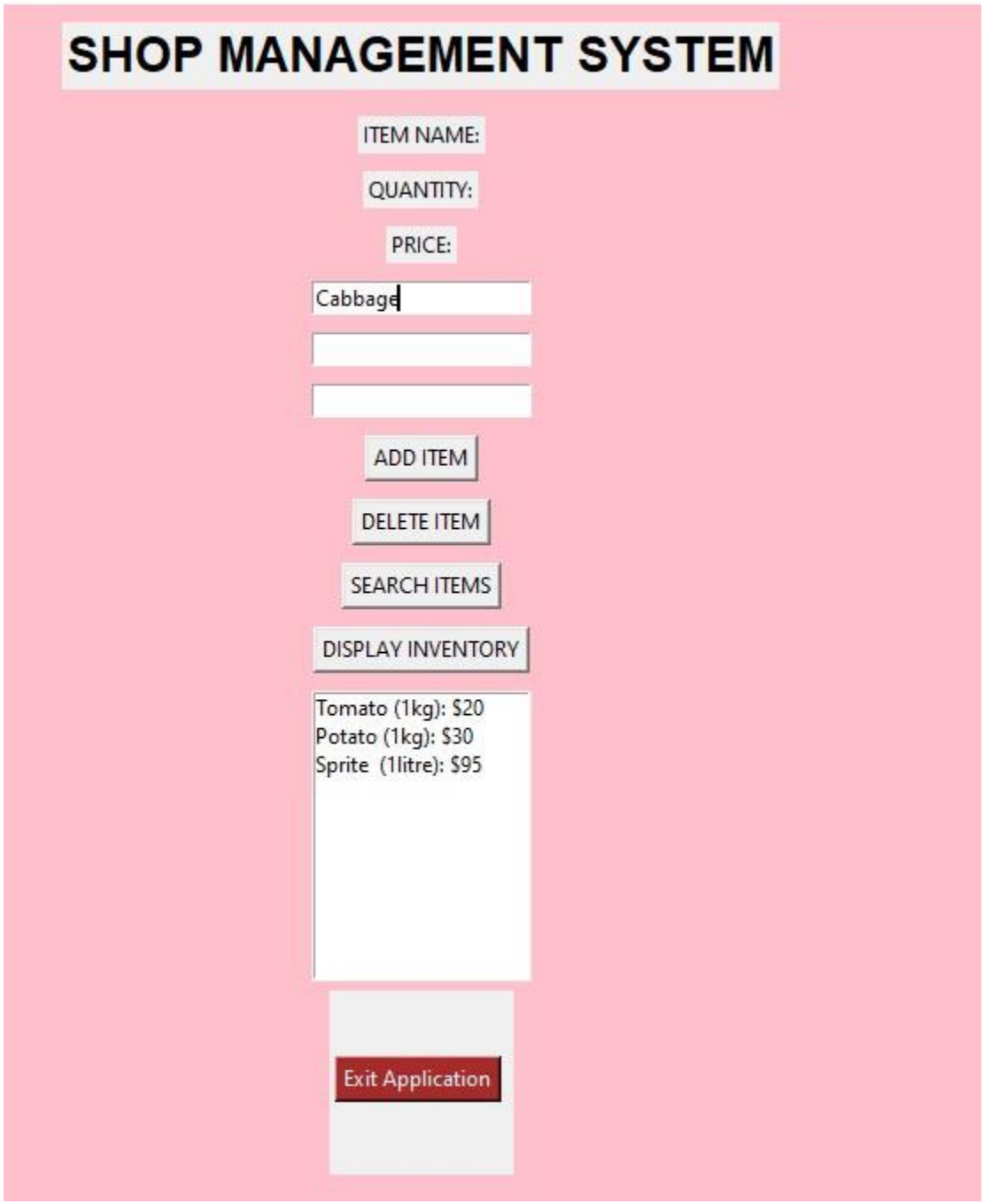
SEARCH_ITEMS:

We can also search items from our inventory, here we have Cabbage in our inventory so we entered Cabbage in Item Name and then we clicked on SEARCH ITEMS then with the help of search function the system will scan the whole inventory and finds for Cabbage. As here we have also searched for Cabbage so we Get Cabbage highlighted in blue colour as our Output.



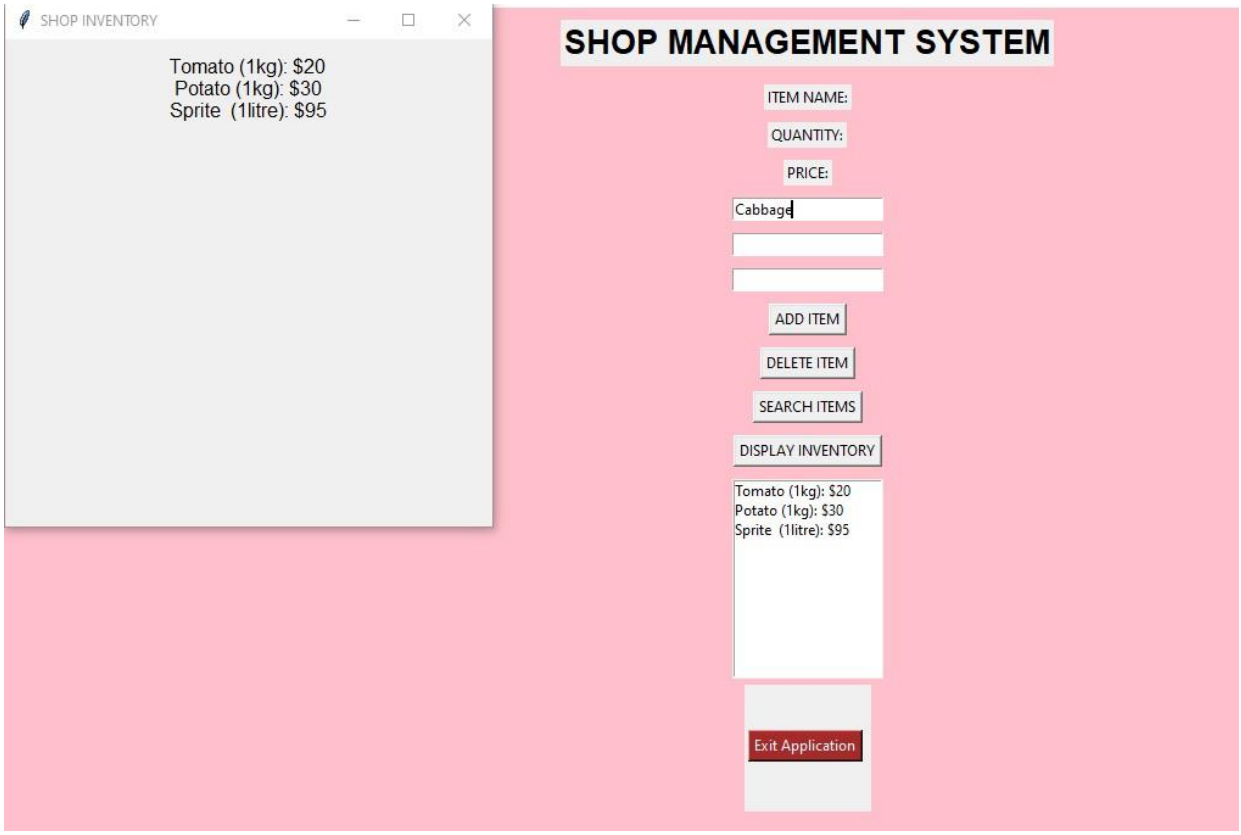
DELETE_ITEM:

Here it's also involves DELETE ITEM which helps to delete the particular items from the inventory. So for doing this we need to put give any input in ITEM NAME and then by clicking on DELETE ITEM, that entered item will be get deleted from Inventory. Here we have entered Cabbage so the system removed the Cabbage from Inventory.



DISPLAY_INVENTORY:

This how our Shop Inventory looks, It's like Database, all the things which we are adding, deleting or searching will be done with the help of this Shop Inventory.



EXIT():

This is the exit function button, once we click on the exit application button then one messagebox will be pop up which confirms where we want to exit the application or not by asking yes or no.

Conclusion and future scope:

In conclusion, a shop management system is a valuable tool for businesses to streamline their operations, improve customer satisfaction, and increase profitability. It can help automate tasks such as inventory management, sales tracking, and customer relationship management, freeing up time for business owners to focus on other aspects of their operations. Furthermore, by providing insights into sales trends and customer behavior, a shop management system can help businesses make informed decisions about pricing, promotions, and inventory management.

Looking towards the future, there are several trends that are likely to shape the development of shop management systems. Firstly, we can expect to see greater integration with other business software and tools, such as accounting software and marketing automation platforms. This will enable businesses to access a more comprehensive view of their operations and make data-driven decisions across all aspects of their business.

Another trend is the increasing use of artificial intelligence and machine learning in shop management systems. These technologies can help businesses to automate even more tasks and provide more accurate insights into customer behavior and sales trends. For example, AI-powered chatbots can be used to provide customer support and answer frequently asked questions, freeing up staff time and improving customer satisfaction.

Finally, we can expect to see continued innovation in the user experience of shop management systems. As businesses become more reliant on these tools, it will become increasingly important for them to be intuitive and easy to use, with features and functionality that are tailored to the specific needs of different industries and business types.

Overall, the future for shop management systems looks bright, with continued innovation and integration with other business tools set to improve the efficiency and profitability of businesses of all sizes.

References :

1. GeeksforGeeks. "Shop Management System." GeeksforGeeks. [Online]. Available: https://www.geeksforgeeks.org/shop-management-system/.

2. ShopVue. "ShopVue - Shop Floor Management System." ShopVue. [Online]. Available: https://www.shopvue.com/.

3. J. R. Patel and S. R. Patel, "Shop Management System." Proceedings of the 2022 International Conference on Advances in Computing, Communication and Control, Pune, India, pp. 227-232, Jan. 2022.

Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
(Religious Jain Minority)

Acknowledgement:

We have a great pleasure in presenting the mini project report on "**SHOP MANAGEMENT SYSTEM**" . We take this opportunity to express our sincere thanks towards our guide **Prof Merlin Priya Jacob**, Department of Computer Engineering, APSIT,Thane for providing the technical guidelines and suggestions regarding line of work. We work like to express our gratitude towards his constant encouragement, support and guidance through the development of project.

We also thank the entire staff of APSIT for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of APSIT for their encouragement.

Student 1: Gautam Pandey (21102117)
Student 2: Pratik Patil (21102099)
Student 3: Sanket Nehe (21102119)