# Project Title : Inventory Management System using Streamlit

### i. Tools / Libraries Used

- **Python** – Core programming language
- **Streamlit** – Web-based UI framework
- **Pandas** – Data handling and processing
- **OpenPyXL** – Excel file handling
- **OS Module** – File existence checking

### ii. Project Description

The **Inventory Management System** is a Streamlit-based web application designed to manage product inventory efficiently using an Excel file as storage.

This system allows users to:

- Add new products
- Remove existing products
- Search for products
- Update stock quantity
- View complete inventory
- Calculate total inventory value

The application uses:

- **Persistent storage** via Excel file (inventory.xlsx)
- **Dynamic sidebar navigation**
- **Real-time data updates**
- **Error handling and validation**
- **Automatic total value calculation (Price × Stock)**

This project demonstrates:

- CRUD operations (Create, Read, Update, Delete)
- File handling
- Data persistence
- Conditional UI rendering
- Inventory valuation logic

**iii. Complete Code:**

```
import streamlit as st
import pandas as pd
import os

file_name = "inventory.xlsx"

# Initialize Excel File
def init_excel():
    if not os.path.exists(file_name):
        df = pd.DataFrame(columns=["Product", "Price", "Stock"])
        df.to_excel(file_name, index=False)

# Load Data
def load_inventory():
    return pd.read_excel(file_name)

# Save Data
def save_inventory(df):
    df.to_excel(file_name, index=False)

# Streamlit UI
st.set_page_config(page_title="Inventory System", layout="centered")
st.title("📦 Inventory Management System")

init_excel()
inventory_df = load_inventory()

menu = st.sidebar.radio(
    "Menu",
    [
        "Add Product",
```

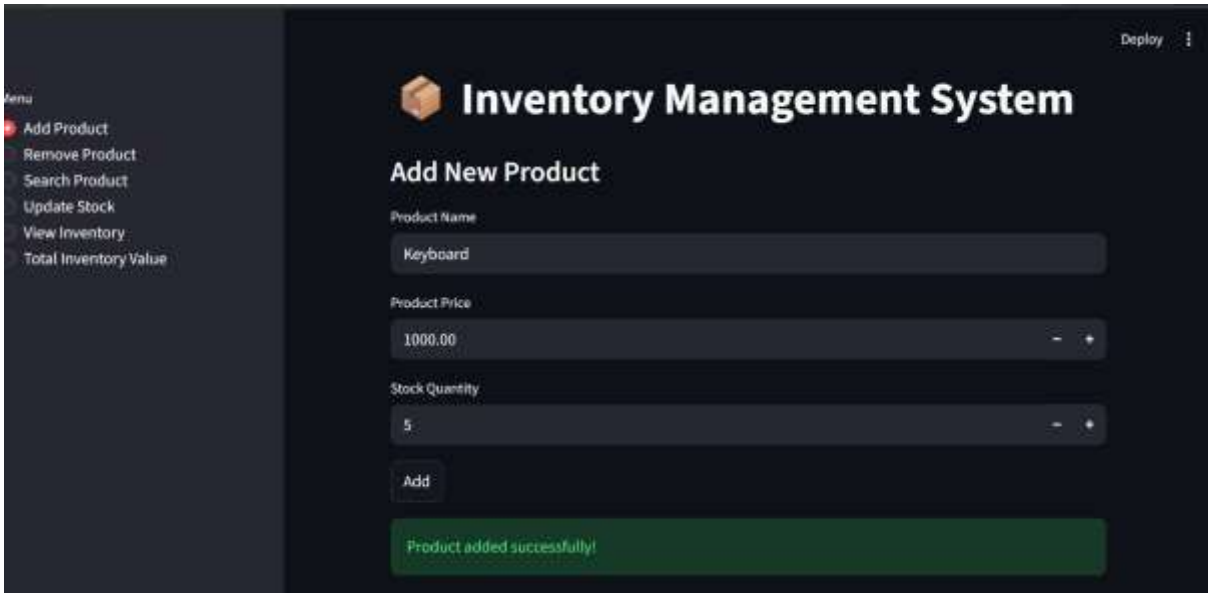
```
        "Remove Product",
        "Search Product",
        "Update Stock",
        "View Inventory",
        "Total Inventory Value",
    ],
)

# Add Product
if menu == "Add Product":
    st.subheader("Add New Product")

    product = st.text_input("Product Name")
    price = st.number_input("Product Price", min_value=0.0)
    stock = st.number_input("Stock Quantity", min_value=0)

    if st.button("Add"):
        if not product:
            st.warning("Please enter product name")
        elif product in inventory_df["Product"].values:
            st.error("Product already exists")
        else:
            new_row = pd.DataFrame(
                {"Product": [product], "Price": [price], "Stock": [stock]}
            )
            inventory_df = pd.concat([inventory_df, new_row], ignore_index=True)
            save_inventory(inventory_df)
            st.success("Product added successfully!")

# Remove Product
elif menu == "Remove Product":
    st.subheader("Remove Product")
```

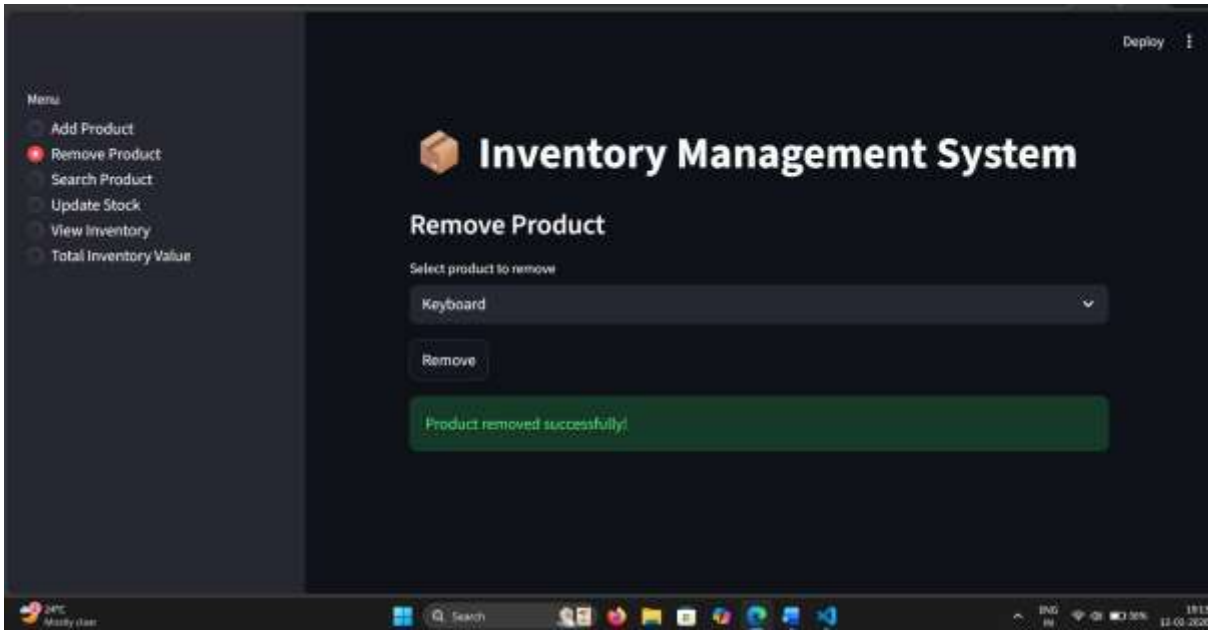
```
    product_list = inventory_df["Product"].tolist()

    if product_list:
        product = st.selectbox("Select product to remove", product_list)
        if st.button("Remove"):
            inventory_df = inventory_df[inventory_df["Product"] != product]
            save_inventory(inventory_df)
            st.success("Product removed successfully!")
    else:
        st.info("No products available")

# Search Product
elif menu == "Search Product":
    st.subheader("Search Product")

    search_term = st.text_input("Enter product name")

    if st.button("Search"):
        if search_term:
            result = inventory_df[
                inventory_df["Product"].str.contains(search_term, case=False, na=False)
            ]

            if not result.empty:
                st.dataframe(result)
            else:
                st.error("Product not found")
        else:
            st.warning("Please enter product name")

# Update Stock
```

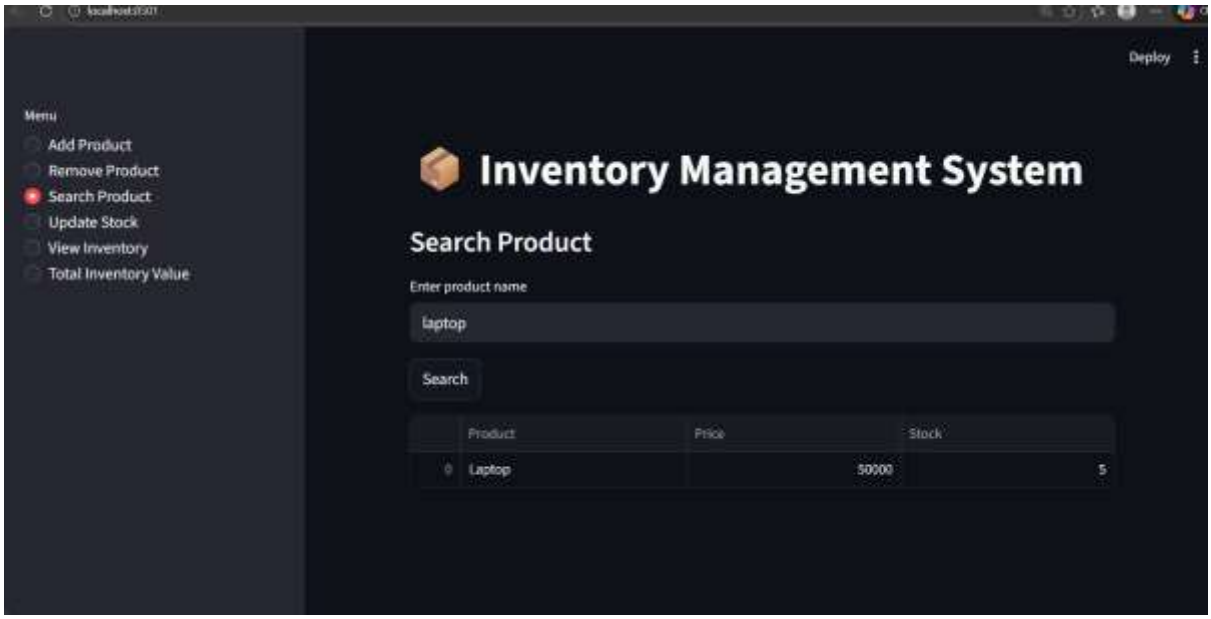
```
elif menu == "Update Stock":
    st.subheader("Update Stock")

    product_list = inventory_df["Product"].tolist()

    if product_list:
        product = st.selectbox("Select product", product_list)
        new_stock = st.number_input("New Stock Quantity", min_value=0)

        if st.button("Update"):
            inventory_df.loc[
                inventory_df["Product"] == product, "Stock"
            ] = new_stock
            save_inventory(inventory_df)
            st.success("Stock updated successfully!")
    else:
        st.info("No products available")

# View Inventory
elif menu == "View Inventory":
    st.subheader("Inventory List")

    if inventory_df.empty:
        st.info("Inventory is empty")
    else:
        st.dataframe(inventory_df)

# Total Inventory Value
elif menu == "Total Inventory Value":
    st.subheader("Total Inventory Value")

    if inventory_df.empty:
```

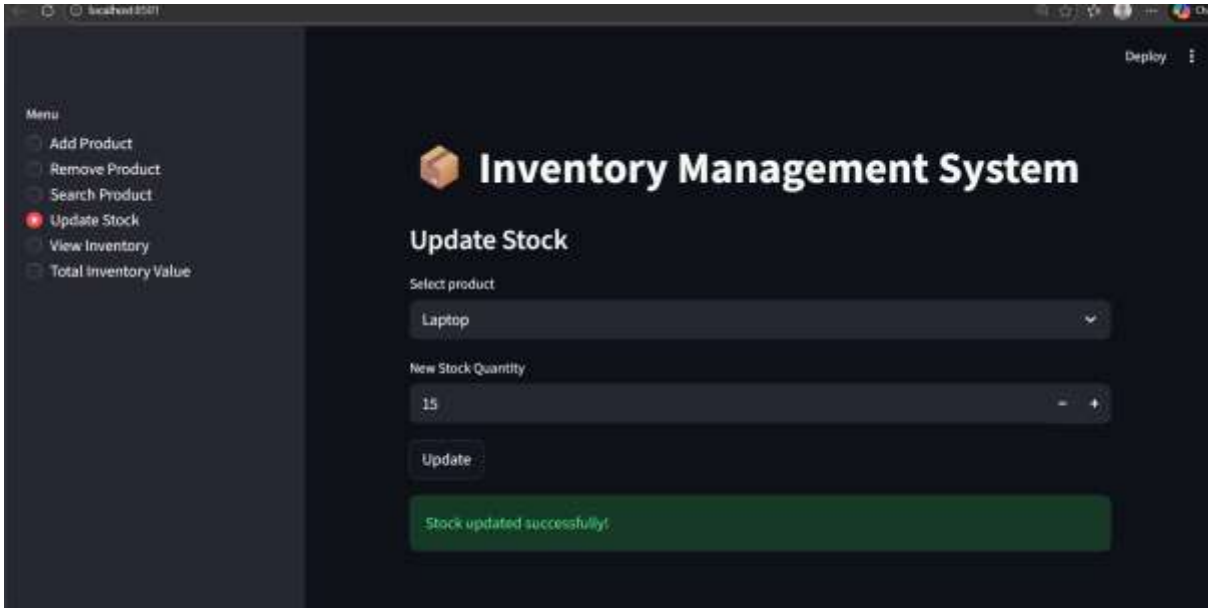
```
        st.info("Inventory is empty")
    else:
        inventory_df["Total Value"] = (
            inventory_df["Price"] * inventory_df["Stock"]
        )

        total_value = inventory_df["Total Value"].sum()

        st.dataframe(inventory_df)
        st.success(f"Total Inventory Value: ₹ {total_value:.2f}")
```

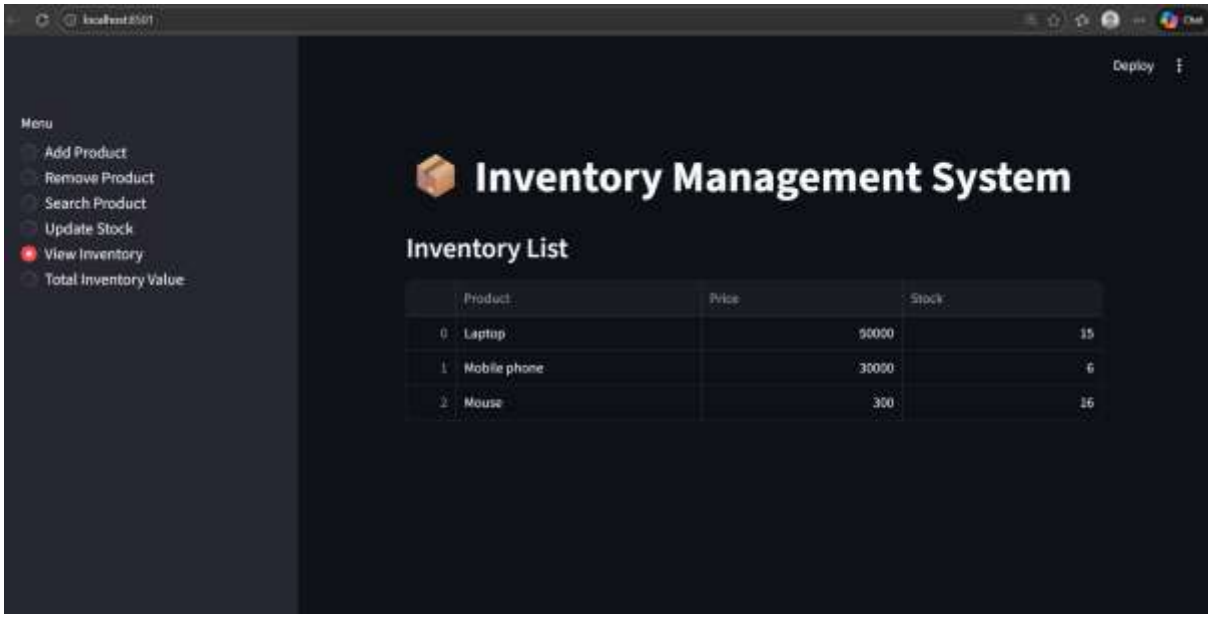
**iv. Output:**

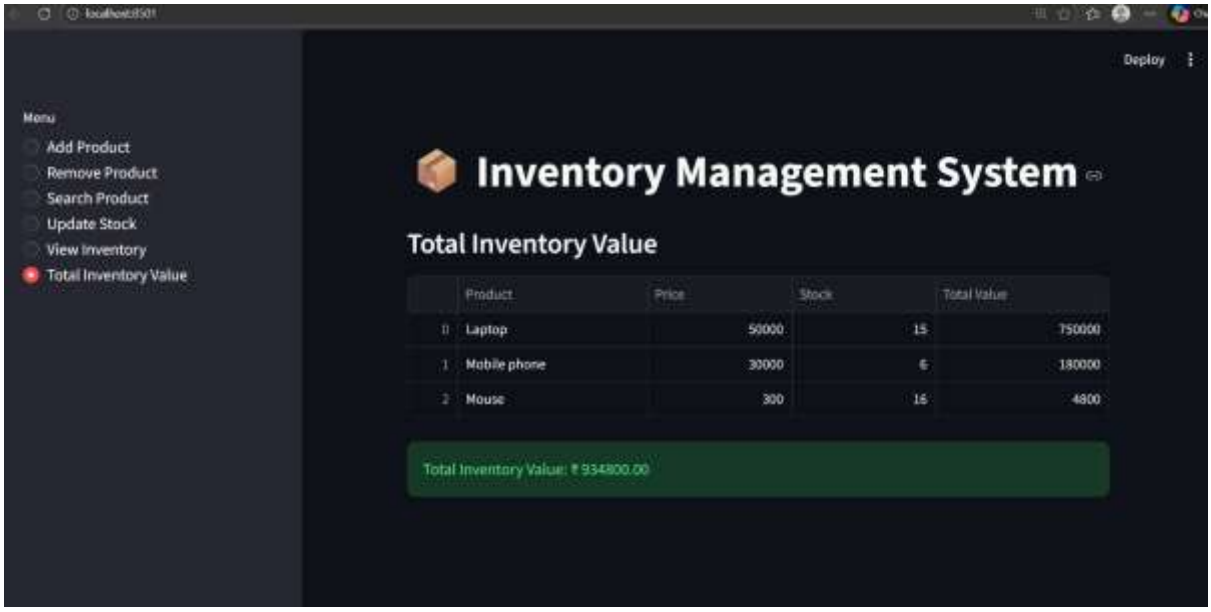



Deploy
Menu
Add Product
Remove Product
Search Product
Update Stock
View Inventory
Total Inventory Value
Inventory Management System
Remove Product
Select product to remove
Keyboard
Remove
Product removed successfully!


Deploy
Menu
Add Product
Remove Product
Search Product
Update Stock
View Inventory
Total Inventory Value
Inventory Management System
Search Product
Enter product name
laptop
Search
Product
Price
Stock
0
Laptop
50000
5


Deploy
Menu
Add Product
Remove Product
Search Product
Update Stock
View Inventory
Total Inventory Value
Inventory Management System
Update Stock
Select product
Laptop
New Stock Quantity
15
Update
Stock updated successfully!


Deploy
Menu
Add Product
Remove Product
Search Product
Update Stock
View Inventory
Total Inventory Value
Inventory Management System
Inventory List
Product
Price
Stock
0 Laptop 50000 15
1 Mobile phone 30000 6
2 Mouse 300 16


Deploy
Menu
Add Product
Remove Product
Search Product
Update Stock
View Inventory
Total Inventory Value
Inventory Management System
Total Inventory Value
Product
Price
Stock
Total Value
0 Laptop 50000 15 750000
1 Mobile phone 30000 6 180000
2 Mouse 300 16 4800
Total Inventory Value: ₹ 934800.00