

Online Shopping Store

<https://classdb.it.mtu.edu/~hashamk/login>

Weihua Zhou

whzhou@mtu.edu

Assistant Professor, Health Informatics,
Applied Computing

Hasham Khan

hashamk@mtu.edu

Final Project Presentation

From Wish list to Cart – Shop Now, Thank Us Later!

Next Semester Goal(long term Goal) & Major Application

- **Launch it for all MTU user by having “@mtu.edu” condition**

The concept discuss by Professor in lecture about defining a pattern that matches the desired email addresses in the regex string(regular expressions)

- Completing all legal aspect of payments methods along with data and payment security
- Making it more user friendly with a lot of more feature (adding Chabot, integration of sms/email alerts, chat space for buyer and seller, conflict resolution mechanic and many more feature etc....)
- My timeline is to complete the whole project in the coming Spring and Summer. Launch it in the Fall 2025 start.

All MTU user will have their own website for showcasing their products, testing and even sale and purchasing with 100 percent MTU family.(Husky eats and closet first)

Challenges for Python Part only:

1. import MySQL. Connector

MySQL. Connector is used to connect to and interact with MySQL database to fetch data

2. import pandas as pd

Pandas is used to handle and manipulate data efficiently, converting the data from the database into DataFrame format for easy analysis and visualization.

3. import matplotlib.pyplot as plt

matplotlib.pyplot is used to create and display visualizations (like bar charts, pie charts) based on the data retrieved from the database.

Python code only

```
import mysql.connector
import pandas as pd
import matplotlib.pyplot as plt

# Connect to the MySQL database
db_connection = mysql.connector.connect(
    host="classdb.it.mtu.edu",
    user="hashamk",
    password="Huawei@786",
    database="hashamk"
)

# Function to execute a query and return data as a pandas DataFrame
def fetch_data(query):
    df = pd.read_sql(query, db_connection)
    return df

# 1. Product Price Distribution (Bar Chart)
def plot_product_prices():
    query = "SELECT name, price FROM Product"
    df = fetch_data(query)

    # Plotting product prices as a bar chart
    plt.figure(figsize=(10, 6))
    plt.bar(df['name'], df['price'], color='skyblue')
    plt.xlabel('Product Name')
    plt.ylabel('Price ($)')
    plt.title('Product Price Distribution')
    plt.xticks(rotation=90)
    plt.tight_layout()
    plt.show()

# 2. Product Stock Quantity Distribution (Bar Chart)
def plot_product_stock():
    query = "SELECT name, stock_quantity FROM Product"
    df = fetch_data(query)

    # Plotting stock quantities as a bar chart
    plt.figure(figsize=(10, 6))
    plt.bar(df['name'], df['stock_quantity'], color='lightgreen')
    plt.xlabel('Product Name')
    plt.ylabel('Stock Quantity')
    plt.title('Product Stock Quantity')
    plt.xticks(rotation=90)
    plt.tight_layout()
    plt.show()

# 3. Sales by Category (Pie Chart)
def plot_sales_by_category():
    query = """
    SELECT c.name AS category_name, SUM(oi.quantity * oi.price) AS total_sales
    FROM OrderItem oi
    JOIN Product p ON oi.product_id = p.product_id
    JOIN Category c ON p.category_id = c.category_id
    GROUP BY c.name
    """
    df = fetch_data(query)

    # Plotting sales by category as a pie chart
    plt.figure(figsize=(8, 8))
    plt.pie(df['total_sales'], labels=df['category_name'], autopct='%1.1f%%',
    plt.title('Sales Distribution by Category')
    plt.axis('equal')
    plt.show()

# 4. Total Orders per Customer (Bar Chart)
def plot_orders_per_customer():
    query = """
    SELECT cu.username, COUNT(o.order_id) AS total_orders
    FROM Orders o
    JOIN Customer cu ON o.customer_id = cu.customer_id
    GROUP BY cu.username
    ORDER BY total_orders DESC
    LIMIT 10
    """
    df = fetch_data(query)

    # Plotting total orders per customer as a bar chart
    plt.figure(figsize=(10, 6))
    plt.bar(df['username'], df['total_orders'], color='salmon')
    plt.xlabel('Customer Username')
    plt.ylabel('Total Orders')
    plt.title('Top 10 Customers by Order Count')
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.show()

# Example of running the functions to generate visualizations
plot_product_prices()      # Bar chart of product prices
plot_product_stock()       # Bar chart of product stock
plot_sales_by_category()   # Pie chart of sales by category
plot_orders_per_customer() # Bar chart of orders by customer
```

<https://classdb.it.mtu.edu/~hashamk/login>

[Home](#) [Login](#) [Register](#)

e

From Wishlist to Cart – Shop Now, Thank Us Later!

Browse Our Categories



Electronics

Devices and gadgets

[View Products](#)

[Login to Add to Cart](#)



Clothing

Apparel and accessories

[View Products](#)

[Login to Add to Cart](#)



Home

Home goods and furniture

[View Products](#)

[Login to Add to Cart](#)



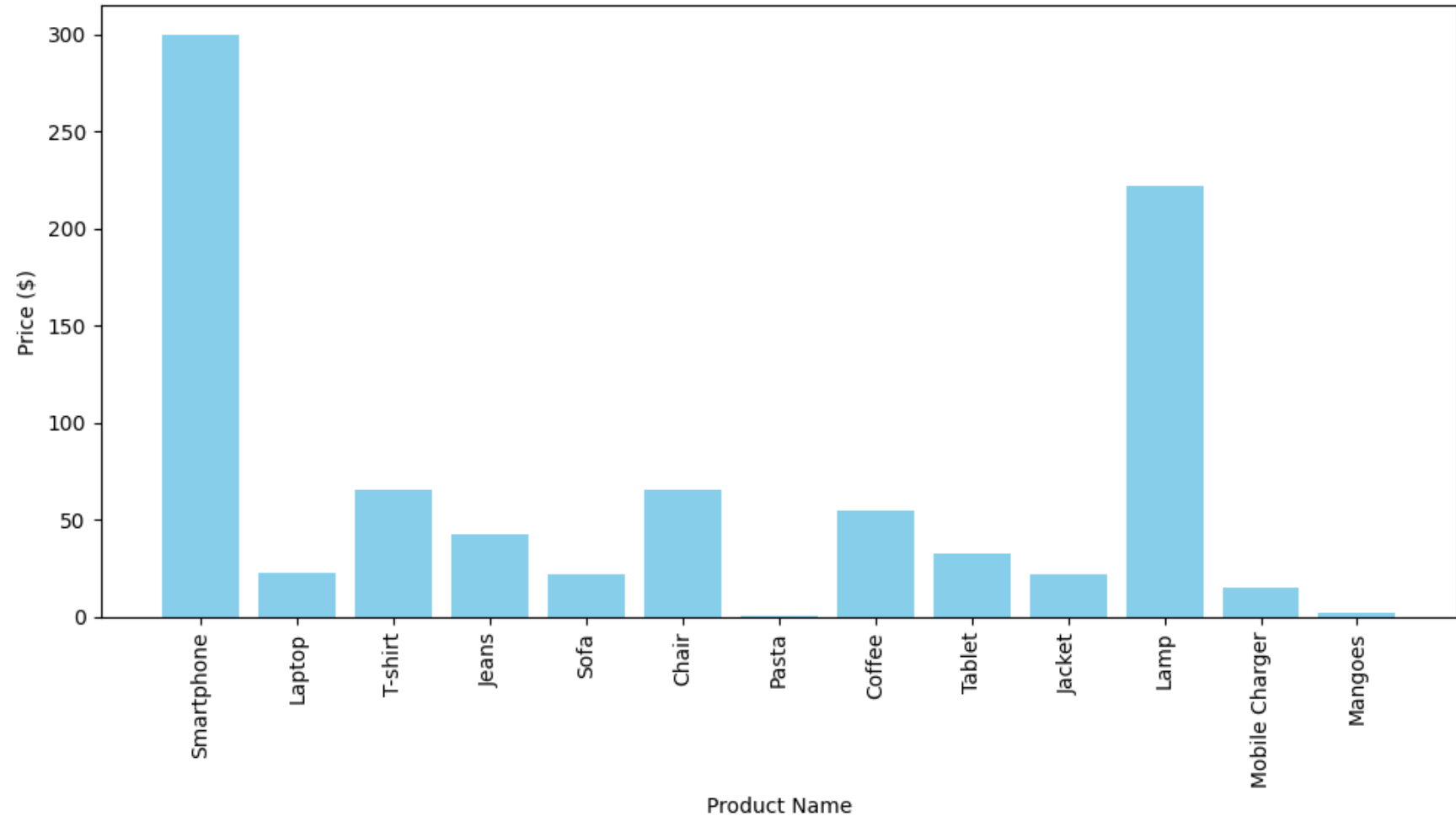
Food

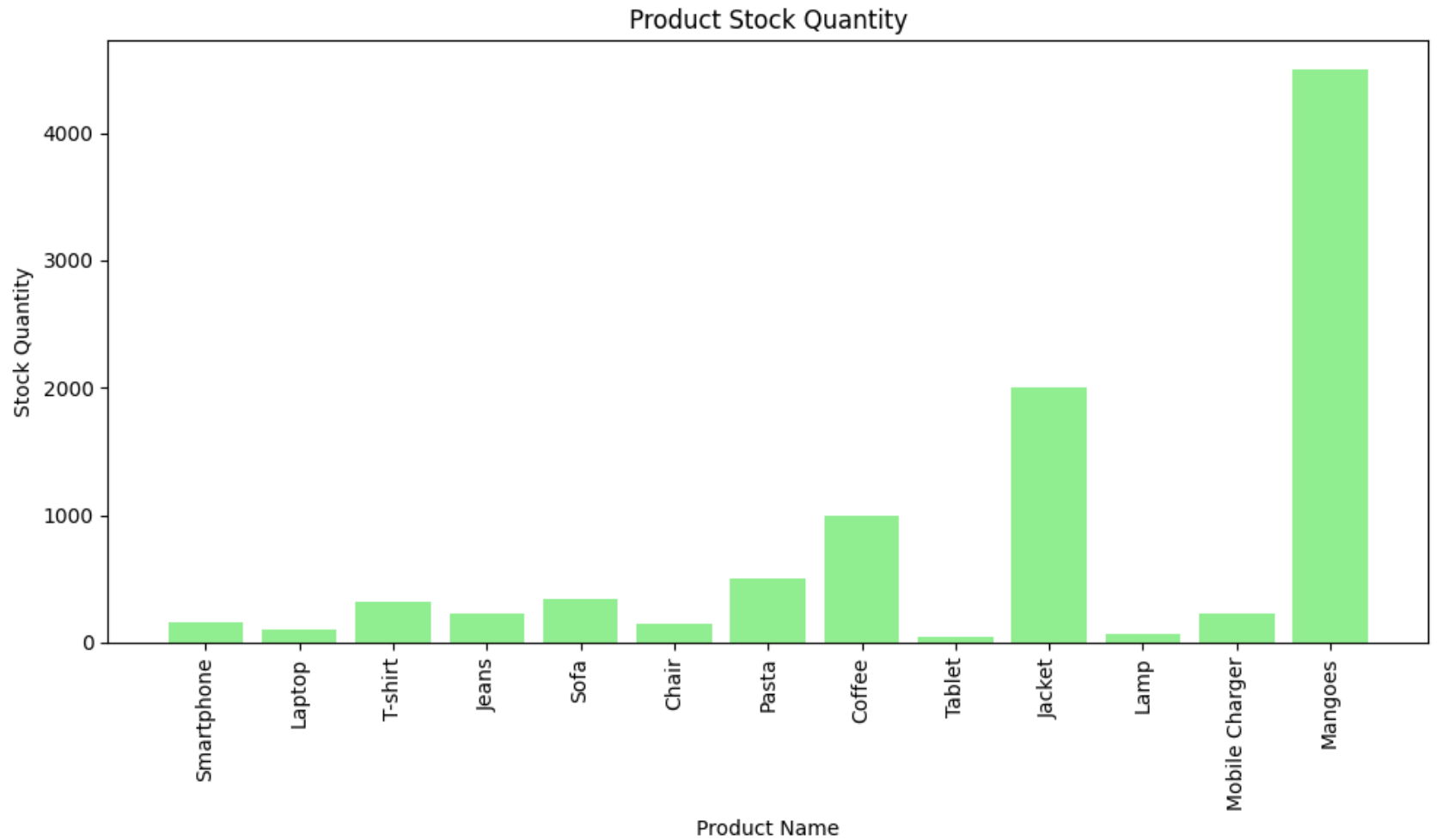
Groceries and consumables

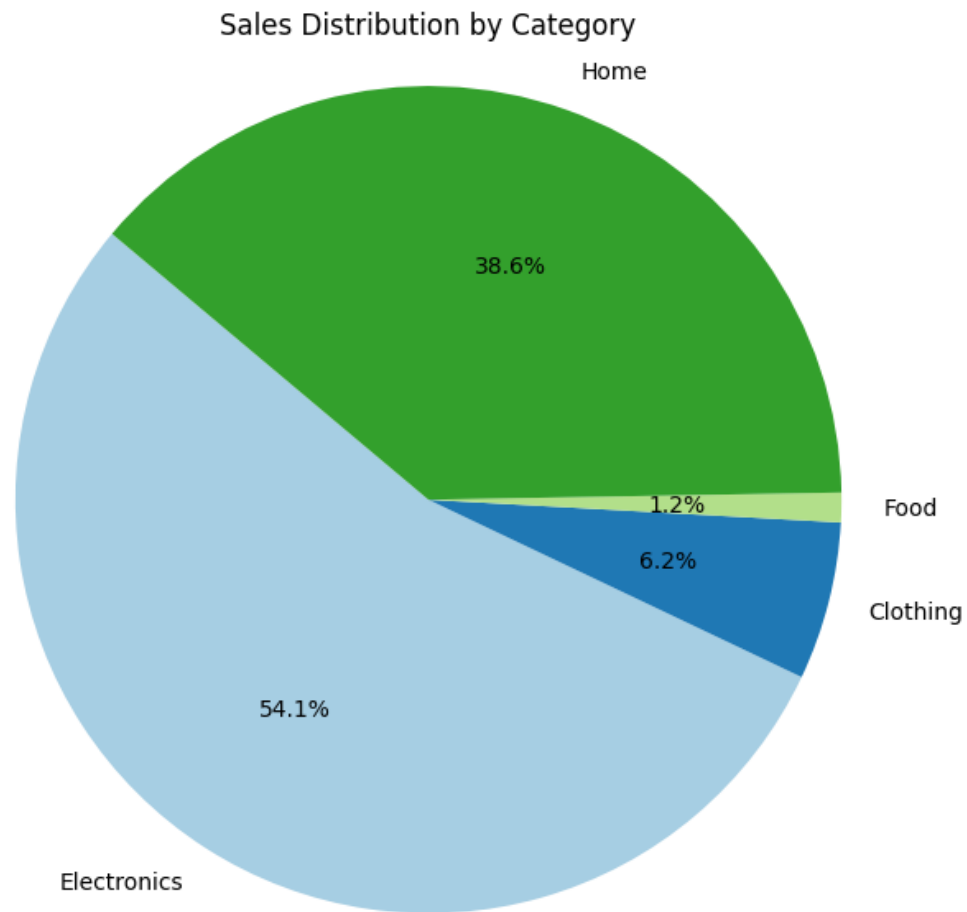
[View Products](#)

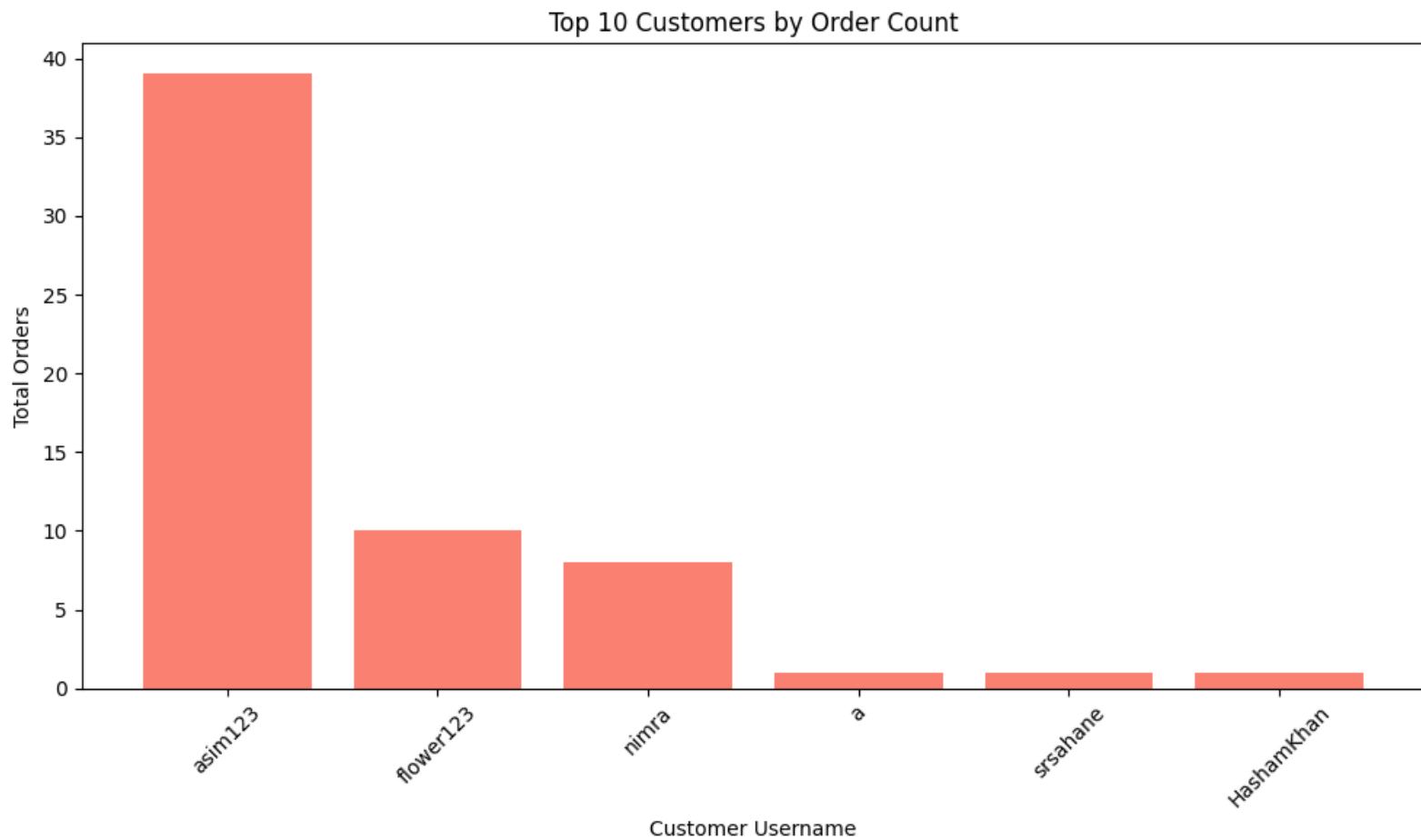
[Login to Add to Cart](#)

Product Price Distribution









"Thoughts, Questions, or Ideas?"

"Your feedback is key to our next steps!"

"Thank you all for your time and attention!"