# Inventory Management System

# 1. Introduction

## Title Of Project

Inventory Management System

## Problem Definition

> To design a comprehensive user-friendly Inventory Management System that efficiently handles stock details and staff information for a retail business. The system should provide functionalities to manage inventory, track stock levels, handle stock transactions, and maintain staff records. The system shall be tailored for customers, aiming to enhance their shopping experience and empower the customers with tools to seamlessly monitor their product stock, ensure availability, and make informed purchasing decisions. The system should be user-friendly and capable of generating useful reports to aid decision-making processes.

## Reason for choosing the topic

Inventory management is a vital necessity as:

- ➔ Monitoring Inventory Levels: It helps the company to monitor inventory levels which in turn helps it to avoid overstocking and minimize stock outs.
- ➔ Maximises Efficiency: It helps the company keep operations efficient by avoiding delays ensuring that operations run smoothly.
- ➔ Increases Profits And Precision: Inventory management also allows a company to maximize profits by minimizing carrying costs and maintaining precise inventory levels so that stock is not lost, misplaced, or stolen.
- ➔ Aids Company Legally: It keeps a company running as it assists businesses in meeting legal and regulatory requirements like tax purposes and compliance with safety laws.
- ➔ Customer Satisfaction: A company's greatest assets are its customers, it ensures customer satisfaction by stocking items at the proper time ensuring client loyalty and satisfaction.

Overall, inventory control management is critical for a company's performance.

## Objective

I. Viewing item details in a well-organized manner
II. Adding new item details
III. Updating/Modifying any item detail
IV. Displaying a particular item's details
V. Creating bills for customers and assisting in the purchase
VI. Analyzing data and generating useful graphs and statistics

## Hardware requirements

**Processor :** Intel® Core i7 2.40 @ GHz or more
**RAM:** 1 GB or More
**Hard disk:** 256 GB or more
**Monitor :** LCD monitor
**Keyboard:** Normal or Multimedia
**Mouse:** Compatible mouse

## Software requirements

I. Python IDLE / VS Code / Google Collab or any other IDE
II. Microsoft Excel

# 2. Modules Used

I. **CSV:** This module is used for reading and writing CSV files. It's used to manage user and staff credentials, as well as order details and staff information.

II. **NumPy:** While imported, it doesn't seem to be directly used in this code. It might have been intended for some numerical operations, but it's not used here.

III. **pandas:** This module is used for data manipulation and analysis. It reads and handles CSV files, primarily for stock data, user and staff credentials, and staff information.

IV. **matplotlib.pyplot:** This module is used for creating visualizations, such as charts and plots. It's used to generate bar charts for the total cost in each category and staff scores.

V. **string:** This module provides a collection of string constants that represent ASCII character classes. It's used to generate random 10-digit IDs for user accounts.

VI. **random:** This module provides functions for generating random numbers. It's used to create random 10-digit order IDs for user accounts while checking out.

VII. **mysql.connector:** This module enables Python programs to access and modify MySQL databases.

VIII. **time:** This module offers functions for accessing and converting time in various formats. We have majorly used it to create a time lag in updating databases.

# 3. Built-in functions

I. Input and Output Functions:
- ➔ input(): Used to take user input.
- ➔ print(): Used to display output to the user.

II. Time Functions:
- ➔ time.sleep(): Used to add delay in execution of program.

III. File Handling Functions:
- ➔ open(): Used to open files for reading or writing.
- ➔ csv.reader(): Used to read CSV files.
- ➔ csv.writer(): Used to write to CSV files.

IV. String Functions:
- ➔ str.lower(): Used to convert strings to lowercase.
- ➔ str.format(): Used for string formatting
- ➔ str.contains() - Used to check if a string contains a specified substring.
- ➔ join() - Used to concatenate characters randomly chosen from a set to generate random IDs.

V. List Functions:
- ➔ list.append(): Used to add an element to the end of a list.
- ➔ list.empty(): Used to check if a DataFrame is empty.

VI. Pandas Functions:
- ➔ pd.read_csv() - Used to read data from a CSV file into a Pandas DataFrame.
- ➔ pd.DataFrame.groupby(): Used to group data in a DataFrame.
- ➔ pd.DataFrame.loc[]: Used to access elements in a DataFrame by label.
- ➔ df.to_csv() - Used to write a Pandas DataFrame to a CSV file.
- ➔ DataFrame indexing and slicing - Used to access and manipulate data within DataFrames.

VII. NumPy Functions:
- ➔ np.random.randint(): Used to generate random integers.

VIII. Matplotlib Functions:
- ➔ plt.figure(): Used to create a new figure for a plot.
- ➔ plt.bar(): Used to create a bar chart.
- ➔ plt.xlabel(): Used to set the x-axis label.

- ➔ plt.ylabel(): Used to set the y-axis label.
- ➔ plt.title(): Used to set the title of the plot.
- ➔ plt.xticks(): Used to set the x-axis tick positions and labels.
- ➔ plt.tight_layout(): Used to improve the layout of the plot.
- ➔ plt.show(): Used to display the plot.

# 4. User-Defined Functions

I. print_section_header:
II. main_menu:

## ***#For USER***

III. create_user_account:
IV. user_sign_in:
V. view_stock:
VI. add_items_to_cart:
VII. check_out:
VIII. status_of_ordered_items:
IX. View_chart_of_stock
X. collect_review:
XI. User_sign_out:

## ***#For STAFF***

XII. staff_login:
XIII. add_new_item:
XIV. add_item_status:
XV. search_for_item:
XVI. remove_item:
XVII. show_all_items:
XVIII. show_all_item_status:
XIX. add_new_staff
XX. search_for_staff
XXI. remove_staff
XXII. show_staff_details
XXIII. staff_functionality
XXIV. sql_connectivity_backup

# 5. Data Types Used

- **Float:** Used for storing decimal values like costs, amounts, etc.
- **List:** Used for storing collections of items, like the shopping cart.
- **Pandas DataFrame:** A two-dimensional, size-mutable, and heterogeneous tabular data structure used for storing and manipulating data, like user credentials, stock details, staff information, and order details.
- **Boolean:** Used to represent True or False values, often in conditions and loops.

# 6. MYSQL Connectivity

The sql_connectivity_backup() function facilitates SQL connectivity within the provided code. It serves the purpose of backing up information into a MySQL database called 'Inventory_Management_System' by establishing a connection through the mysql.connector library.

This function performs several key tasks:

- **Establishing Connection:** It connects to the MySQL database using sql.connect() method, specifying parameters such as host, username, password, etc.
- **Creating Tables:** If absent, it generates tables (credentials_user, credentials_staff, stock, order1) within the database.
- **Importing Data from CSV Files:** It reads data from CSV files (credentials_user.csv, credentials_staff.csv, stock.csv, order.csv) and inserts it into the relevant tables within the MySQL database.
- **Extracting Data from Database:** After data insertion, it retrieves and prints data from each table within the database.

Primarily, this function serves as a backup mechanism by transferring data from CSV files into a MySQL database. It oversees table creation, data insertion, and retrieval within the database.

The SQL connectivity component involves establishing the connection (sql.connect()), creating tables (via CREATE TABLE queries), inserting data (using INSERT INTO queries), and retrieving data (via SELECT * FROM queries). This functionality enables Python code to interact with and manipulate data stored in a MySQL database.

# 7. CODE

```
# Importing all required modules

import mysql.connector as sql
import csv
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import string
import random
import time

# Define constants for file paths
CREDENTIALS_USER_FILE = 'credentials_user.csv'
CREDENTIALS_STAFF_FILE = 'credentials_staff.csv'
STOCK_FILE = 'stock.csv'
ORDER_FILE = 'order.csv'

# Create User Defined Functions
current_user_name = None
cart = []
stock_data = pd.read_csv(STOCK_FILE)
order_data = pd.read_csv(ORDER_FILE)
staff_data = pd.read_csv(CREDENTIALS_STAFF_FILE)


def main_menu():
        print()
        print('='*50)
        print('Inventory Management System'.center(50))
        print('='*50)
        print("Welcome to IMS !")
        print()

def print_section_header(txt):
    print('-' * 50)
    width = 50
    print(f'{txt.upper():^{width}}')
    print('-' * 50)
    print()

    # For User
def create_user_account():
    print_section_header('Create Account')
    name = input("Enter your name: ")
    email = input("Enter your email: ")
    password = input("Enter your password: ")

    id_number = ''.join(random.choices(string.ascii_uppercase + string.digits, k=10))
```

```
    with open(CREDENTIALS_USER_FILE, 'a', newline='') as file:
        writer = csv.writer(file)
        writer.writerow([id_number, name, email, password])

    print(f'Your ID number is: {id_number}')
    print("Account created successfully!")

def user_sign_in():
    print_section_header('sign in')
    global current_user_name
    while True:
        email_or_id = input("Enter your email or ID: ")
        password = input("Enter your password: ")

        with open(CREDENTIALS_USER_FILE, 'r') as csvfile:
            csv_reader = csv.reader(csvfile)
            for row in csv_reader:
                if (row[2] == email_or_id or row[0] == email_or_id) and row[3] == password:
                    current_user_name = row[1]
                    print(f"Welcome Back {current_user_name}!")
                    return

        print("Incorrect email/ID or password. Please try again.")

def view_stock():
    print_section_header('View Stock')
    global stock_data
    stock_data = pd.read_csv(STOCK_FILE)
    print(stock_data)

def add_items_to_cart():
    print_section_header('Add items to cart')
    global cart, stock_data
    choice = 'yes'

    with open(r'cart.csv', 'w', newline='') as f:
        writer = csv.writer(f)
        writer.writerow(['item_number', 'item_name', 'item_brand', 'quantity', 'total_cost'])

        while choice.lower() == 'yes':
            item_number = input("Enter the item number to add to cart: ")
            quantity = int(input("Enter the quantity: "))
            item_info = stock_data[stock_data['item_no'] == int(item_number)]
            if item_info.empty:
                print("Item not found.")
                return
            available_quantity = item_info['quantity'].values[0]
            if quantity > available_quantity:
                print(f"Quantity entered exceeds available stock ({available_quantity} units). Please
enter a valid quantity.")
                return

            item_name = item_info['item_name'].values[0]
```

```
            item_brand = item_info['item_brand'].values[0]
            cost = item_info['cost'].values[0]
            total_cost = cost * quantity  # Calculate total cost
            # Add item to cart if quantity is available
            cart.append([item_number, item_name, item_brand, quantity, total_cost])
            writer.writerow([item_number, item_name, item_brand, quantity, total_cost])


            choice = input('Do you want to add more items: (yes/no) ')


def check_out():
    print_section_header('Check Out')
    global stock_data

    # Read cart items from cart.csv using pandas
    cart_df = pd.read_csv('cart.csv')

    if cart_df.empty:
        print("Your cart is empty. Please add items before checking out.")
        return

    # Loop through items in the cart to update stock quantity
    for index, row in cart_df.iterrows():
        item_number = row['item_number']
        quantity_purchased = row['quantity']

        # Find the item in the stock data
        item_index = stock_data.index[stock_data['item_no'] == item_number].tolist()

        if item_index:
            # Update stock quantity by subtracting purchased quantity
            stock_data.at[item_index[0], 'quantity'] -= quantity_purchased

    # Save the updated stock data back to the CSV file
    stock_data.to_csv(STOCK_FILE, index=False)

    # Calculate the total cost of items in the cart
    total_cost = cart_df['total_cost'].sum()

    # Calculate tax (VAT 5%) and cost after tax
    tax_rate = 0.05
    tax = total_cost * tax_rate
    cost_after_tax = total_cost + tax

    # Print the cart using pandas
    print("Your Cart:")
    print(cart_df)

    # Print the total cost, tax, and cost after tax
    print(f"Total Cost: ${total_cost:.2f}")
    print(f"Tax (VAT 5%): ${tax:.2f}")
    print(f"Cost After Tax: ${cost_after_tax:.2f}")
```

```
    # Generate a unique order ID
    order_id = np.random.randint(10**5, 10**6)

    # Check if the generated order ID already exists
    while order_id in order_data['order_id'].values:
        order_id = np.random.randint(10**5, 10**6)  # Regenerate if ID exists

    # Create a string representation of the order details
    order_details = "\n".join([f"{row['item_name']} (Quantity: {row['quantity']}) - Total Cost:
${row['total_cost']:.2f}" for index, row in cart_df.iterrows()])

    # Add the new order to the orders DataFrame
    with open(ORDER_FILE, 'a', newline='') as file:
        writer = csv.writer(file)
        writer.writerow([order_id, order_details, 'pending'])

    print(f"Order placed successfully! Your Order ID is {order_id}.")

    # Clear the cart by overwriting cart.csv
    with open('cart.csv', 'w', newline='') as file:
        writer = csv.writer(file)
        writer.writerow(['item_number', 'item_name', 'item_brand', 'quantity', 'cost'])

def status_of_ordered_items():
    print_section_header('Status of ordered items')
    order_id = int(input("Enter your Order ID: "))

    with open(ORDER_FILE, 'r') as file:
        reader = csv.reader(file)
        next(reader)
        for row in reader:
            if int(row[0]) == order_id:
                print("Order ID:", row[0])
                print("Order Details:", row[1])
                print("Status:", row[2])
                return
        else:
                print("Order not found")

def view_chart_of_stock(stock_data, staff_data):
    print_section_header('View Chart and Statistics')
    plt.figure(figsize=(12, 8))

    # Calculate the total cost for each category
    category_totals = stock_data.groupby('category')['cost'].sum()

    # Create a bar chart for category totals
    plt.bar(category_totals.index, category_totals.values)
    plt.xlabel('Category')
    plt.ylabel('Total Cost')
    plt.title('Total Cost in Each Category')
    plt.xticks(rotation=45)
    plt.show()
```

```
# Create a bar chart for staff scores
plt.figure(figsize=(8, 6))
plt.bar(staff_data['name'], staff_data['staff_score'])
plt.xlabel('Staff Name')
plt.ylabel('Staff Score')
plt.title('Staff Score')
plt.xticks(rotation=45)
plt.show()

# Generate brand vs. cost chart
plt.figure(figsize=(12, 8))
plt.bar(stock_data['item_brand'], stock_data['cost'])
plt.xlabel('Item Brand')
plt.ylabel('Average Cost')
plt.title('Average Cost by Item Brand')
plt.xticks(rotation=45)
plt.show()

# Generate item vs. cost chart
plt.figure(figsize=(12, 8))
plt.bar(stock_data['item_name'], stock_data['cost'])
plt.xlabel('Item Name')
plt.ylabel('Average Cost')
plt.title('Average Cost by Item Name')
plt.xticks(rotation=45)
plt.show()

# Display additional statistics
most_common_brand = stock_data['item_brand'].mode().values[0]
least_common_brand = stock_data['item_brand'].value_counts().idxmin()
top_expensive_items = stock_data.nlargest(5, 'cost')
top_cheap_items = stock_data.nsmallest(5, 'cost')
highest_quantity_item = stock_data.nlargest(1, 'quantity')
lowest_quantity_item = stock_data.nsmallest(1, 'quantity')
avg_price = stock_data['cost'].mean()
mode_price = stock_data['cost'].mode().values[0]
median_price = stock_data['cost'].median()
price_variance = stock_data['cost'].var()
price_std_dev = stock_data['cost'].std()

print("Additional Statistics:")
print(f"Most Common Brand: {most_common_brand}")
print(f"Least Common Brand: {least_common_brand}")
print("Top 5 Most Expensive Items:")
print(top_expensive_items[['item_name', 'cost']])
print("Top 5 Least Expensive Items:")
print(top_cheap_items[['item_name', 'cost']])
print("Highest Quantity Item:")
print(highest_quantity_item[['item_name', 'quantity']])
print("Lowest Quantity Item:")
print(lowest_quantity_item[['item_name', 'quantity']])
print(f"Average Price: {avg_price:.2f}")
print(f"Mode Price: {mode_price:.2f}")
print(f"Median Price: {median_price:.2f}")
```

```
    print(f"Price Variance: {price_variance:.2f}")
    print(f"Price Standard Deviation: {price_std_dev:.2f}")
    print("Quartiles:")
    print(f"- Q1 (lower half price median): {stock_data['cost'].quantile(0.25):.2f}")
    print(f"- Q2 (median): {stock_data['cost'].quantile(0.50):.2f}")
    print(f"- Q3 (upper half price median): {stock_data['cost'].quantile(0.75):.2f}")
    print(f"Interquartile Range (IQR): {stock_data['cost'].quantile(0.75) -
stock_data['cost'].quantile(0.25):.2f}")

def collect_review():
    print_section_header('User review')
    global current_user_name
    if current_user_name:
        review = input("Enter your review: ")

        with open('reviews.csv', 'a', newline='') as file:
            writer = csv.writer(file)
            writer.writerow([current_user_name, review])

        print("Thank you for your review!")
    else:
        print("No user signed in. Cannot collect review.")

def user_sign_out():
    print("Signing out...")
    time.sleep(1)
    global current_user_name
    current_user_name = None
    print("Signed out successfully.")

    # For Staff
def staff_login():
    print_section_header('Staff login')
    max_attempts = 3
    while max_attempts > 0:
        staff_id = input("Enter staff ID: ")
        password = input("Enter staff password: ")

        with open('credentials_staff.csv', 'r') as csvfile:
            csvreader = csv.reader(csvfile)
            for row in csvreader:
                if staff_id == row[1] and password == row[2]:
                    print(f"Welcome staff member: {row[0]}")
                    return True  # Return True if login is successful

        max_attempts -= 1
        if max_attempts > 0:
            print(f"Invalid credentials. You have {max_attempts} attempts left.")

    print("Access denied.")
    return False  # Return False if login is unsuccessful

def add_new_item(stock_data):
    print_section_header('Add new item')
```

```
    item_no = int(input("Enter item number: "))
    item_name = input("Enter item name: ")
    item_brand = input("Enter item brand: ")
    cost = float(input("Enter cost: "))
    category = input("Enter category: ")
    status = input("Enter status: ")
    quantity = int(input("Enter quantity: "))

    new_item = {
        'item_no': item_no,
        'item_name': item_name,
        'item_brand': item_brand,
        'cost': cost,
        'category': category,
        'status': status,
        'quantity': quantity
    }

    # Append the new item to the DataFrame
    stock_data = pd.concat([stock_data, pd.DataFrame([new_item])], ignore_index=True)
    stock_data.to_csv(STOCK_FILE, index=False)

    print("Item added successfully!")

def add_item_status():
    print_section_header('Add item status')
    item_no = input("Enter item number to update status: ")
    status = input("Enter status: ")

    # Read the existing data
    with open(STOCK_FILE, 'r', newline='') as csvfile:
        reader = csv.DictReader(csvfile)
        rows = list(reader)

    # Update the status for the item with the specified item number
    updated_rows = []
    found = False
    for row in rows:
        if row['item_no'] == item_no:
            row['status'] = status
            found = True
        updated_rows.append(row)

    # Write the updated data back to the CSV file
    with open(STOCK_FILE, 'w', newline='') as csvfile:
        fieldnames = ['item_no', 'item_name', 'item_brand', 'cost', 'category', 'status', 'quantity']
        writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
        writer.writeheader()
        writer.writerows(updated_rows)

    if found:
        print("Status updated successfully!")
    else:
        print("Item not found.")
```

```
def search_for_item(stock_data):
    print_section_header('Search item')
    item_name = input("Enter item name to search: ")

    # Convert the 'item_name' and 'item_brand' columns to string type
    stock_data['item_name'] = stock_data['item_name'].astype(str)
    stock_data['item_brand'] = stock_data['item_brand'].astype(str)

    search_result = stock_data[stock_data['item_name'].str.contains(item_name, case=False) |
stock_data['item_brand'].str.contains(item_name, case=False)]

    if not search_result.empty:
        print("Search Results:")
        print(search_result)
    else:
        print("Item not found.")

def remove_item(stock_data):
    print_section_header('Remove item')
    item_no = int(input("Enter item number to remove: "))

    stock_data = stock_data[stock_data['item_no'] != item_no]
    stock_data.to_csv('stock.csv', index=False)

    print("Item removed successfully!")

def show_all_items(stock_data):
    print_section_header('Show all items')
    print("All Items:")
    stock_data = pd.read_csv(STOCK_FILE)
    print(stock_data)

def show_all_item_status(stock_data):
    print_section_header('Show all item statuses')
    print("All Item Status:")
    stock_data = pd.read_csv(STOCK_FILE)
    print(stock_data[['item_no', 'item_name', 'item_brand', 'status']])

def add_new_staff(staff_data):
    print_section_header('Add new staff')
    name = input("Enter staff name: ")
    staff_id = input("Enter staff ID: ")
    password = input("Enter password: ")
    phone_number = input("Enter phone number: ")
    staff_score = int(input("Enter staff score: "))

    new_staff = {'name': name, 'staff_id': staff_id, 'password': password, 'phone_number':
phone_number, 'staff_score': staff_score}
    staff_data = staff_data._append(new_staff, ignore_index=True)
    staff_data.to_csv('credentials_staff.csv', index=False)

    print("New staff added successfully!")
```

```
def search_for_staff(staff_data):
    print_section_header('Search staff details')
    staff_name = input("Enter staff name to search: ")
    staff_data['name'] = staff_data['name'].astype(str)
    search_result = staff_data[staff_data['name'].str.contains(staff_name, case=False)]

    if not search_result.empty:
        print("Search Results:")
        print(search_result)
    else:
        print("Staff member not found.")

def remove_staff(staff_data):
    print_section_header('Remove staff')
    staff_id = input("Enter staff ID to remove: ")

    staff_data = staff_data[staff_data['staff_id'] != staff_id]
    staff_data.to_csv('credentials_staff.csv', index=False)

    print("Staff member removed successfully!")

def show_staff_details(staff_data):
    print_section_header('Show staff details')
    print("Staff Details:")
    print(staff_data)

def sql_connectivity_backup():
    print_section_header('BACKUP DATABASE')
    print("Backing up data...")
    time.sleep(1.5)

    object = sql.connect(host = 'localhost', user = 'root', password = 'Ihs@10987864')
    pointer = object.cursor()
    pointer.execute("CREATE database if not exists Inventory_Management_System")
    pointer.execute("Use Inventory_Management_System")

    #Creating Empty Tables
    pointer.execute("""CREATE TABLE IF NOT EXISTS credentials_user (
                id_number CHAR(10) PRIMARY KEY,
                name VARCHAR(20),
                email VARCHAR(50),
                password VARCHAR(25)
                )""")
    pointer.execute("""CREATE TABLE IF NOT EXISTS credentials_staff (
                name VARCHAR(20),
                staff_id INT PRIMARY KEY,
                password VARCHAR(15),
                phone_number BIGINT,
                staff_score INT
                )""")
    pointer.execute("""CREATE TABLE IF NOT EXISTS stock (
                item_no INT PRIMARY KEY,
                item_name VARCHAR(255),
                item_brand VARCHAR(255),
```

```
            cost INT,
            category VARCHAR(30),
            status VARCHAR(50),
            quantity INT
            )""")
    pointer.execute("""CREATE TABLE IF NOT EXISTS order1 (
            order_id INT PRIMARY KEY,
            order_details VARCHAR(255),
            status VARCHAR(30)
            )""")

    #Inserting values from csv file
    CREDENTIALS_USER_FILE = csv.reader(open('credentials_user.csv'))
    next(CREDENTIALS_USER_FILE)
    for row in CREDENTIALS_USER_FILE:
        query = ("INSERT INTO credentials_user(id_number, name, email, password) "
                "VALUES(%s, %s, %s, %s) "
                "ON DUPLICATE KEY UPDATE name = VALUES(name), email = VALUES(email), password =
VALUES(password)"
                )
        pointer.execute(query, row)
        object.commit()

    CREDENTIALS_STAFF_FILE = csv.reader(open('credentials_staff.csv'))
    next(CREDENTIALS_STAFF_FILE)
    for row in CREDENTIALS_STAFF_FILE:
        query = ("INSERT INTO credentials_staff(name, staff_id, password, phone_number, staff_score) "
                "VALUES(%s, %s, %s, %s, %s) "
                "ON DUPLICATE KEY UPDATE name = VALUES(name), staff_id = VALUES(staff_id), password =
VALUES(password),\
                phone_number = VALUES(phone_number), staff_score = VALUES(staff_score)"
                )
        pointer.execute(query,row)
        object.commit()

    STOCK_FILE = csv.reader(open('stock.csv'))
    next(STOCK_FILE)
    for row in STOCK_FILE:
        query = ("INSERT INTO stock(item_no, item_name, item_brand, cost, category, status, quantity) "
                "VALUES(%s, %s, %s, %s, %s, %s, %s) "
                "ON DUPLICATE KEY UPDATE item_no = VALUES(item_no), item_name = VALUES(item_name),
item_brand = VALUES(item_brand),\
                cost = VALUES(cost), category = VALUES(category), status = VALUES(status), quantity =
VALUES(quantity)"
                )
        pointer.execute(query,row)
        object.commit()

    ORDER_FILE = csv.reader(open('order.csv'))
    next(ORDER_FILE)
    for row in ORDER_FILE:
        query = ("INSERT INTO order1(order_id, order_details, status) "
                "VALUES(%s, %s, %s) "
```

```
            "ON DUPLICATE KEY UPDATE order_id = VALUES(order_id), order_details =
VALUES(order_details), status = VALUES(status)"
            )
        pointer.execute(query,row)
        object.commit()

    #Extracting data from database
    pointer.execute("Select * from credentials_user")
    print("User Credentials")
    a = pointer.fetchall()
    for i in a:
        print(i)
    print('\n')

    pointer.execute("Select * from credentials_staff")
    print("Staff Credentials")
    b = pointer.fetchall()
    for i in b:
        print(i)
    print('\n')

    pointer.execute("Select * from stock")
    print("Stock details")
    c = pointer.fetchall()
    for i in c:
        print(i)
    print('\n')

    pointer.execute("Select * from order1")
    print("Order details")
    d = pointer.fetchall()
    for i in d:
        print(i)
    print('\n')

    print("All data has been backed-up to your mysql database 'Inventory_Management_System'.")


def staff_functionality():
    global staff_data, stock_data
    while True:
        print("\nStaff Menu:")
        print("Press 1 - Add new item")
        print("Press 2 - Add item status")
        print("Press 3 - Search for an item")
        print("Press 4 - Remove an item")
        print("Press 5 - Show all items")
        print("Press 6 - Show all item statuses")
        print("Press 7 - Add new staff")
        print("Press 8 - Search for staff")
        print("Press 9 - Remove staff")
        print("Press 10 - Show details of staff")
        print("Press 11 - To view chart")
        print("Press 12 - To backup the database")
```

```
        print("Press 13 - To exit")

        staff_option = input("Enter option number: ")

        if staff_option == '1':
            add_new_item(stock_data)
        elif staff_option == '2':
            add_item_status()
        elif staff_option == '3':
            search_for_item(stock_data)
        elif staff_option == '4':
            remove_item(stock_data)
        elif staff_option == '5':
            show_all_items(stock_data)
        elif staff_option == '6':
            show_all_item_status(stock_data)
        elif staff_option == '7':
            add_new_staff(staff_data)
        elif staff_option == '8':
            search_for_staff(staff_data)
        elif staff_option == '9':
            remove_staff(staff_data)
        elif staff_option == '10':
            show_staff_details(staff_data)
        elif staff_option == '11':
            view_chart_of_stock(stock_data, staff_data)
        elif staff_option == '12':
            sql_connectivity_backup()
        elif staff_option == '13':
            print("Exiting staff functionality...")
            break
        else:
            print("Invalid staff option.")

# Main Code
main_menu()
user_type = input("Are you a user or staff? ").lower()

if user_type == 'user':
    print('''Choice:
          1. Create Account
          2. Sign In''')
    user_choice = input("Enter choice as (1/2) ")

    if user_choice == '1':
        create_user_account()
        user_sign_in()
    elif user_choice == '2':
        user_sign_in()
    else:
        print("Invalid choice.")

    # After successful sign-in, show the menu
    while True:
```

```
        print("\nMenu:")
        print("1. View Stock")
        print("2. Add Items to Cart")
        print("3. Check Out")
        print("4. Status of Ordered Items")
        print("5. View Chart of Stock")
        print("6. Exit")

        option = input("Enter option number: ")

        if option == '1':
            view_stock()
        elif option == '2':
            add_items_to_cart()
        elif option == '3':
            check_out()
        elif option == '4':
            status_of_ordered_items()
        elif option == '5':
            view_chart_of_stock(stock_data, staff_data)
        elif option == '6':
            collect_review()
            user_sign_out()
            cart = []
            print("Exiting...")
            break
        else:
            print("Invalid option.")

elif user_type == 'staff':
    staff_logged_in = staff_login()

    if staff_logged_in:
        staff_data = pd.read_csv(CREDENTIALS_STAFF_FILE)
        staff_functionality()
else:
    print("Invalid user type.")
```

# 8. OUTPUT

## #USER

```
Menu:
1. View Stock
2. Add Items to Cart
3. Check Out
4. Status of Ordered Items
5. View Chart of Stock
6. Exit
Enter option number: 2
--------------------------------------------------
                ADD ITEMS TO CART
--------------------------------------------------

Enter the item number to add to cart: 1
Enter the quantity: 2
Do you want to add more items: (yes/no) yes
Enter the item number to add to cart: 2345
Enter the quantity: 3
Do you want to add more items: (yes/no) no
```

```
Menu:
1. View Stock
2. Add Items to Cart
3. Check Out
4. Status of Ordered Items
5. View Chart of Stock
6. Exit
Enter option number: 3
--------------------------------------------------
                    CHECK OUT
--------------------------------------------------

Your Cart:
   item_number      item_name item_brand  quantity  total_cost
0            1     kappa-shoe      kappa         2      1378.0
1         2345  boat-earphone       boat         3      4197.0
Total Cost: $5575.00
Tax (VAT 5%): $278.75
Cost After Tax: $5853.75
Order placed successfully! Your Order ID is 117328.
```

```
Menu:
1. View Stock
2. Add Items to Cart
3. Check Out
4. Status of Ordered Items
5. View Chart of Stock
6. Exit
Enter option number: 4
--------------------------------------------------
             STATUS OF ORDERED ITEMS
--------------------------------------------------

Enter your Order ID: 608517
Order ID: 608517
Order Details: kappa-shoe (Quantity: 2) - Total Cost: $1378.00
boat-earphone (Quantity: 1) - Total Cost: $1399.00
Status: Inventory
```

```
Menu:
1. View Stock
2. Add Items to Cart
3. Check Out
4. Status of Ordered Items
5. View Chart of Stock
6. Exit
Enter option number: 5
--------------------------------------------------
            VIEW CHART AND STATISTICS
--------------------------------------------------
```

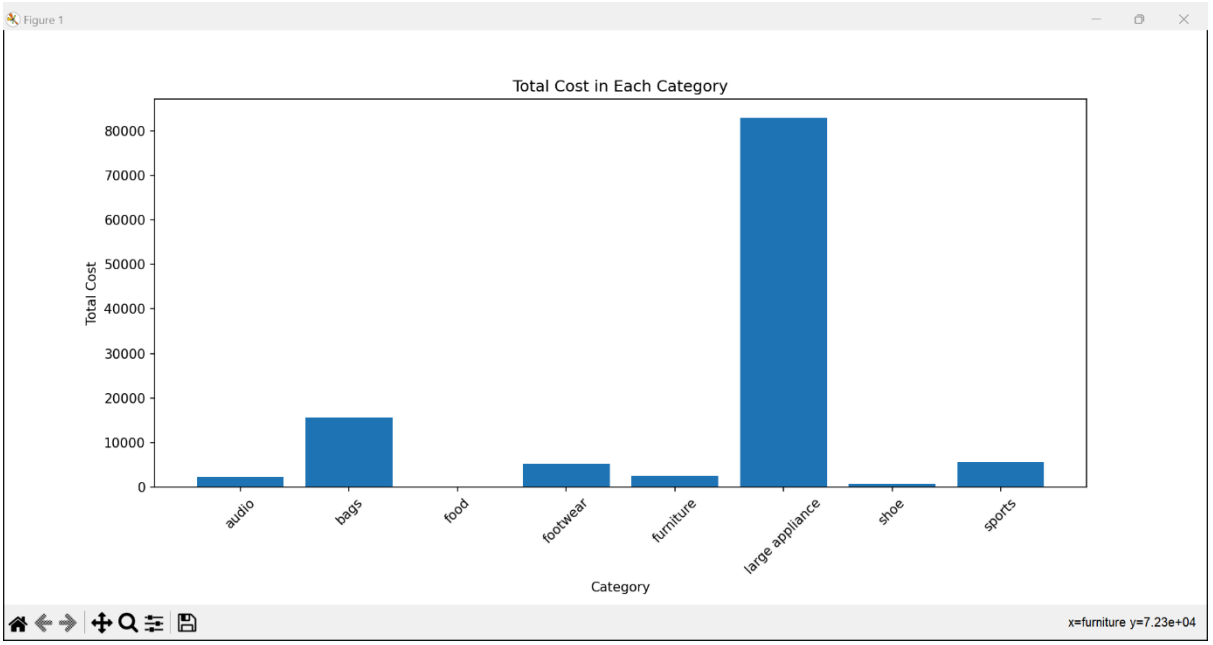


Figure 1
Total Cost in Each Category
Total Cost
80000
70000
60000
50000
40000
30000
20000
10000
0
audio
bags
food
footwear
furniture
large appliance
shoe
sports
Category
x=furniture y=7.23e+04

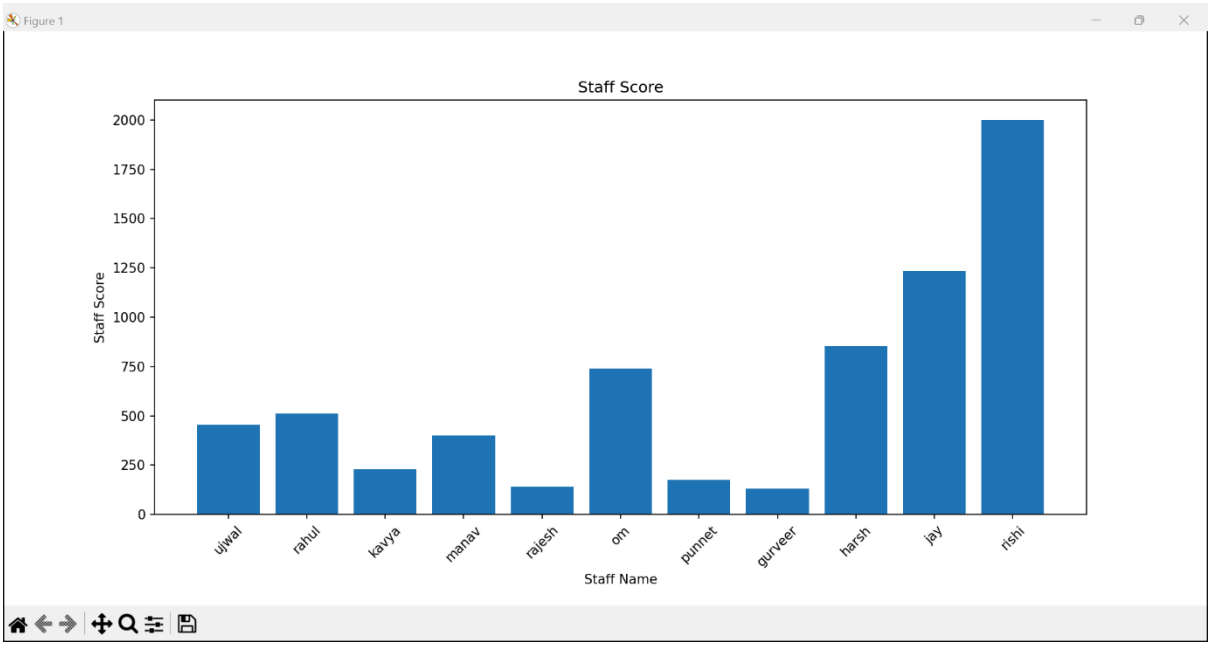
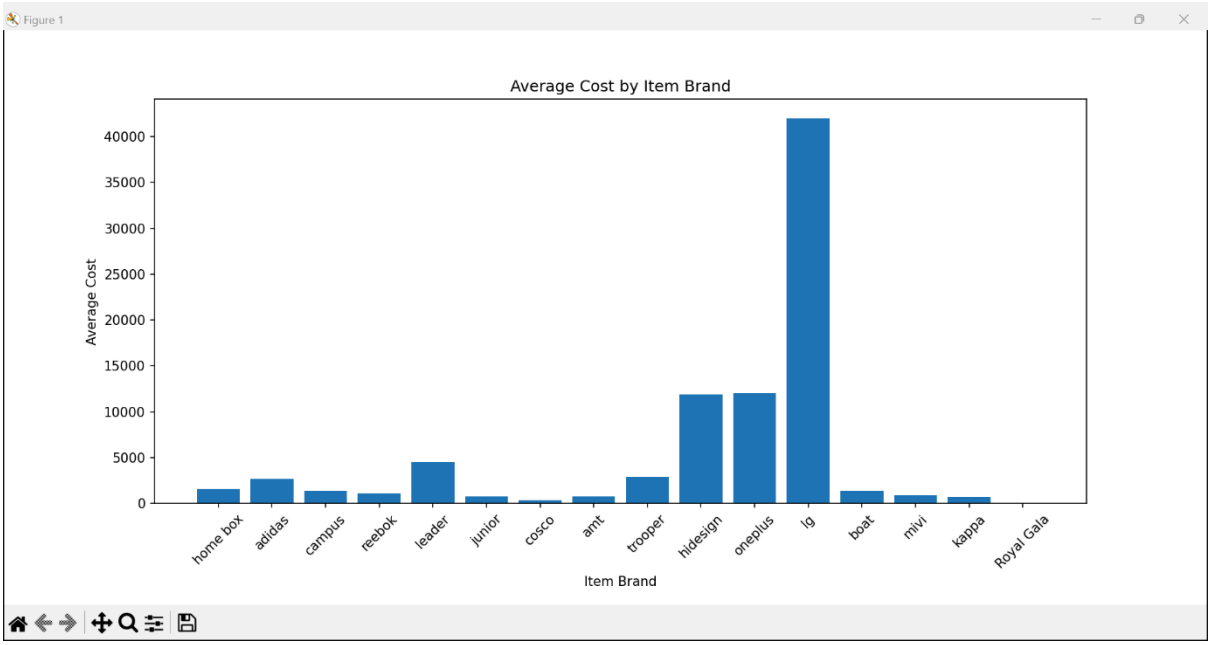
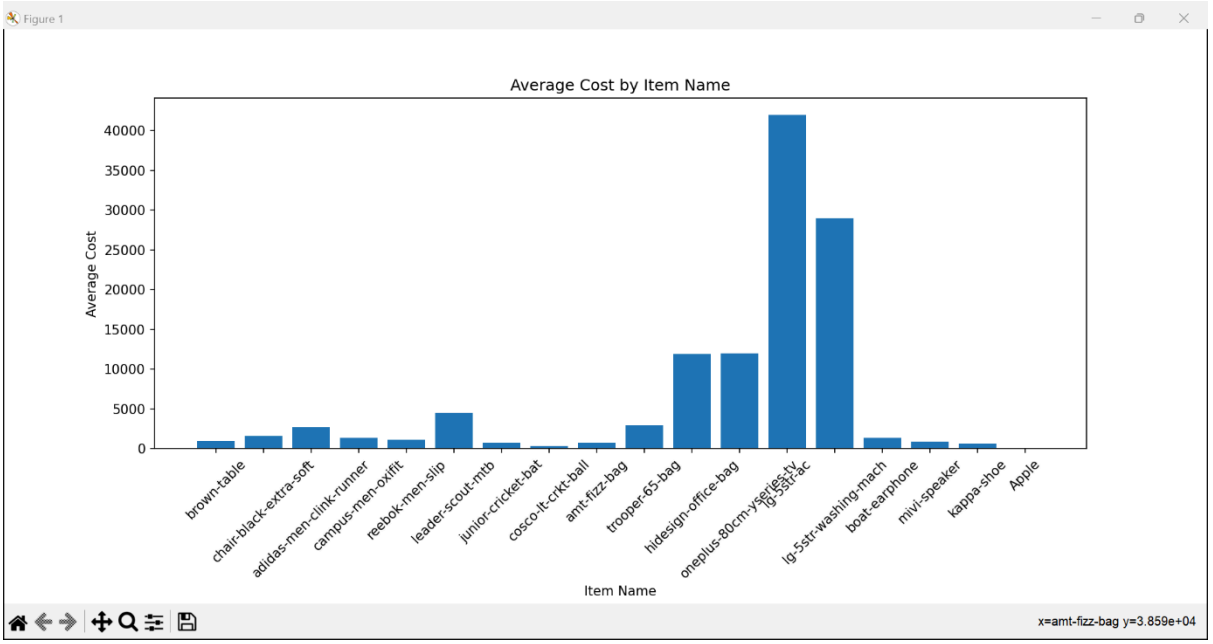


Figure 1
Staff Score
Staff Score
2000
1750
1500
1250
1000
750
500
250
0
ujwal
rahul
kavya
manav
rajesh
om
punnet
gurveer
harsh
jay
rishi
Staff Name




```
Additional Statistics:
Most Common Brand: home box
Least Common Brand: adidas
Top 5 Most Expensive Items:
                item_name     cost
12              lg-5str-ac  41990.0
13     lg-5str-washing-mach  28990.0
11  oneplus-80cm-yseries-tv  11999.0
10      hidesign-office-bag  11895.0
5          leader-scout-mtb   4499.0
Top 5 Least Expensive Items:
            item_name   cost
7   cosco-lt-crkt-ball  365.0
16          kappa-shoe  689.0
6   junior-cricket-bat  749.0
8         amt-fizz-bag  749.0
15        mivi-speaker  899.0
Highest Quantity Item:
       item_name  quantity
15  mivi-speaker       237
Lowest Quantity Item:
          item_name  quantity
5  leader-scout-mtb         5
Average Price: 6439.78
Mode Price: 749.00
Median Price: 1399.00
Price Variance: 128865700.18
Price Standard Deviation: 11351.90
```

```
Quartiles:
- Q1 (lower half price median): 924.00
- Q2 (median): 1399.00
- Q3 (upper half price median): 4099.00
Interquartile Range (IQR): 3175.00
```

```
Menu:
1. View Stock
2. Add Items to Cart
3. Check Out
4. Status of Ordered Items
5. View Chart of Stock
6. Exit
Enter option number: 6
--------------------------------------------------
                  USER REVIEW
--------------------------------------------------

Enter your review: Really good
Thank you for your review!
Signing out...
```

# #STAFF

```
C:\WINDOWS\py.exe

==================================================
          Inventory Management System
==================================================
Welcome to IMS !

Are you a user or staff? staff
--------------------------------------------------
                  STAFF LOGIN
--------------------------------------------------

Enter staff ID: 2232
Enter staff password: worker
Welcome staff member: ujwal
```

```
Staff Menu:
Press 1 - Add new item
Press 2 - Add item status
Press 3 - Search for an item
Press 4 - Remove an item
Press 5 - Show all items
Press 6 - Show all item statuses
Press 7 - Add new staff
Press 8 - Search for staff
Press 9 - Remove staff
Press 10 - Show details of staff
Press 11 - To view chart
Press 12 - To backup the database
Press 13 - To exit
Enter option number: 1
--------------------------------------------------
                  ADD NEW ITEM
--------------------------------------------------

Enter item number: 89568
Enter item name: Airpods Gen 2
Enter item brand: Apple
Enter cost: 999
Enter category: Electronics
Enter status: Inventory
Enter quantity: 80
Item added successfully!
```

```
Staff Menu:
Press 1 - Add new item
Press 2 - Add item status
Press 3 - Search for an item
Press 4 - Remove an item
Press 5 - Show all items
Press 6 - Show all item statuses
Press 7 - Add new staff
Press 8 - Search for staff
Press 9 - Remove staff
Press 10 - Show details of staff
Press 11 - To view chart
Press 12 - To backup the database
Press 13 - To exit
```

```
Enter option number: 2
--------------------------------------------------
                ADD ITEM STATUS
--------------------------------------------------

Enter item number to update status: 12345
Enter status: in stock
Status updated successfully!
```

```
Staff Menu:
Press 1 - Add new item
Press 2 - Add item status
Press 3 - Search for an item
Press 4 - Remove an item
Press 5 - Show all items
Press 6 - Show all item statuses
Press 7 - Add new staff
Press 8 - Search for staff
Press 9 - Remove staff
Press 10 - Show details of staff
Press 11 - To view chart
Press 12 - To backup the database
Press 13 - To exit
Enter option number: 3
--------------------------------------------------
                  SEARCH ITEM
--------------------------------------------------

Enter item name to search: Airpods Gen 2
Search Results:
    item_no      item_name item_brand   cost     category     status  quantity
18    89568  Airpods Gen 2      Apple  999.0  Electronics  Inventory        80
```

```
Staff Menu:
Press 1 - Add new item
Press 2 - Add item status
Press 3 - Search for an item
Press 4 - Remove an item
Press 5 - Show all items
Press 6 - Show all item statuses
Press 7 - Add new staff
Press 8 - Search for staff
Press 9 - Remove staff
Press 10 - Show details of staff
Press 11 - To view chart
Press 12 - To backup the database
Press 13 - To exit
Enter option number: 4
--------------------------------------------------
                  REMOVE ITEM
--------------------------------------------------

Enter item number to remove: 12345
Item removed successfully!
```

```
Staff Menu:
Press 1 - Add new item
Press 2 - Add item status
Press 3 - Search for an item
Press 4 - Remove an item
Press 5 - Show all items
Press 6 - Show all item statuses
Press 7 - Add new staff
Press 8 - Search for staff
Press 9 - Remove staff
Press 10 - Show details of staff
Press 11 - To view chart
Press 12 - To backup the database
Press 13 - To exit
Enter option number: 5
```

```
--------------------------------------------------
                  SHOW ALL ITEMS
--------------------------------------------------

All Items:
    item_no               item_name item_brand     cost         category      status  quantity
0     12566             brown-table   home box    999.0        furniture     scanned        10
1     12903   chair-black-extra-soft   home box   1599.0        furniture     shipped        13
2     12677  adidas-men-clink-runner     adidas   2699.0         footwear     shipped        60
3    327798        campus-men-oxifit     campus   1399.0         footwear  inventory       150
4      3386          reebok-men-slip     reebok   1099.0         footwear  inventory        60
5     82476         leader-scout-mtb     leader   4499.0           sports  inventory         5
6     28456       junior-cricket-bat     junior    749.0           sports  inventory        10
7      9862       cosco-lt-crkt-ball      cosco    365.0           sports     scanned       100
8      7686             amt-fizz-bag        amt    749.0             bags     scanned        48
9     28385          trooper-65-bag    trooper   2899.0             bags     shipped        37
10     2869      hidesign-office-bag   hidesign  11895.0             bags     shipped         7
11     9286  oneplus-80cm-yseries-tv    oneplus  11999.0  large appliance  inventory        10
```

```
12      384              lg-5str-ac         lg  41990.0  large appliance     scanned         8
13     2535     lg-5str-washing-mach         lg  28990.0  large appliance     scanned         9
14     2345            boat-earphone       boat   1399.0            audio     shipped        36
15      267             mivi-speaker       mivi    899.0            audio     scanned       237
16        1               kappa-shoe      kappa    689.0             shoe     shipped        27
17    89568            Airpods Gen 2      Apple    999.0      Electronics   Inventory        80
```

```
Staff Menu:
Press 1 - Add new item
Press 2 - Add item status
Press 3 - Search for an item
Press 4 - Remove an item
Press 5 - Show all items
Press 6 - Show all item statuses
Press 7 - Add new staff
Press 8 - Search for staff
Press 9 - Remove staff
Press 10 - Show details of staff
Press 11 - To view chart
Press 12 - To backup the database
Press 13 - To exit
Enter option number: 6
```

```
------------------------------------------------
              SHOW ALL ITEM STATUSES
------------------------------------------------

All Item Status:
    item_no               item_name item_brand      status
0     12566             brown-table   home box     scanned
1     12903   chair-black-extra-soft   home box     shipped
2     12677  adidas-men-clink-runner     adidas     shipped
3    327798        campus-men-oxifit     campus   inventory
4      3386          reebok-men-slip     reebok   inventory
5     82476         leader-scout-mtb     leader   inventory
6     28456       junior-cricket-bat     junior   inventory
7      9862       cosco-lt-crkt-ball      cosco     scanned
8      7686             amt-fizz-bag        amt     scanned
9     28385          trooper-65-bag    trooper     shipped
10     2869      hidesign-office-bag   hidesign     shipped
11     9286  oneplus-80cm-yseries-tv    oneplus   inventory
12      384              lg-5str-ac         lg     scanned
13     2535     lg-5str-washing-mach         lg     scanned
14     2345            boat-earphone       boat     shipped
15      267             mivi-speaker       mivi     scanned
16        1               kappa-shoe      kappa     shipped
17    89568            Airpods Gen 2      Apple   Inventory
```

```
Staff Menu:
Press 1 - Add new item
Press 2 - Add item status
Press 3 - Search for an item
Press 4 - Remove an item
Press 5 - Show all items
Press 6 - Show all item statuses
Press 7 - Add new staff
Press 8 - Search for staff
Press 9 - Remove staff
Press 10 - Show details of staff
Press 11 - To view chart
Press 12 - To backup the database
Press 13 - To exit
Enter option number: 7
------------------------------------------------
                  ADD NEW STAFF
------------------------------------------------

Enter staff name: Evan
Enter staff ID: 30482
Enter password: Manager
Enter phone number: 0508094714
Enter staff score: 95
New staff added successfully!
```

```
Staff Menu:
Press 1 - Add new item
Press 2 - Add item status
Press 3 - Search for an item
Press 4 - Remove an item
Press 5 - Show all items
Press 6 - Show all item statuses
Press 7 - Add new staff
Press 8 - Search for staff
Press 9 - Remove staff
Press 10 - Show details of staff
Press 11 - To view chart
Press 12 - To backup the database
Press 13 - To exit
```

```
Enter option number: 8
--------------------------------------------------
               SEARCH STAFF DETAILS
--------------------------------------------------

Enter staff name to search: Evan
Search Results:
    name  staff_id password  phone_number  staff_score
11  Evan     30482  Manager     508094714           95
```

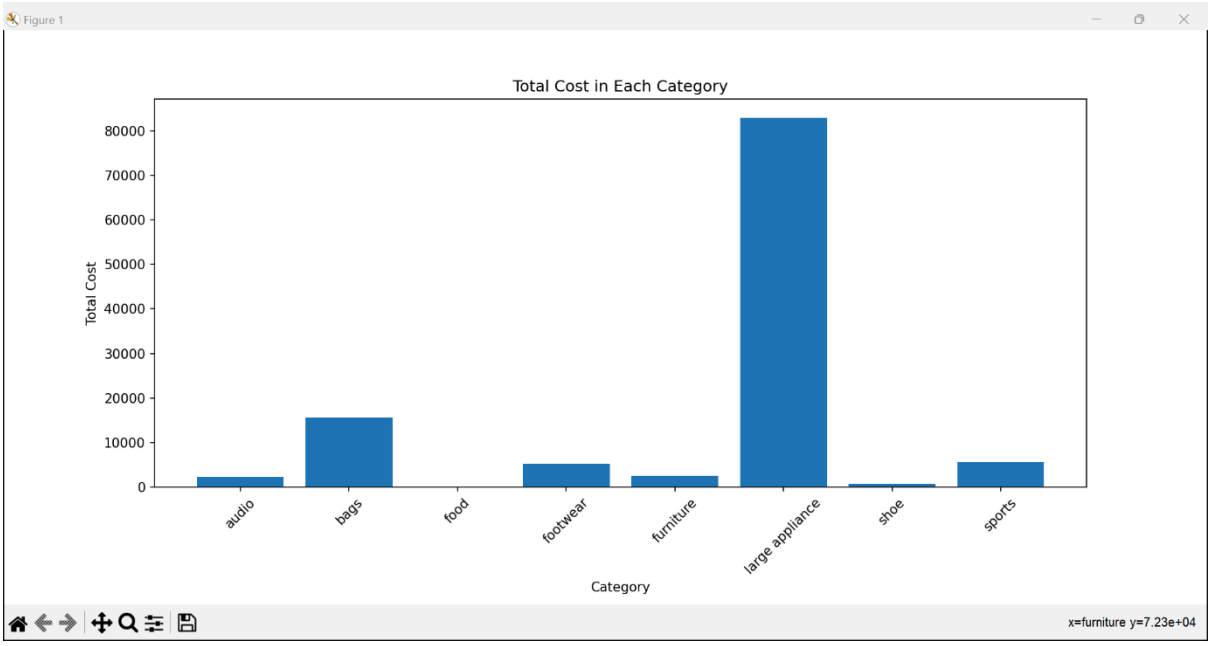
```
Staff Menu:
Press 1 - Add new item
Press 2 - Add item status
Press 3 - Search for an item
Press 4 - Remove an item
Press 5 - Show all items
Press 6 - Show all item statuses
Press 7 - Add new staff
Press 8 - Search for staff
Press 9 - Remove staff
Press 10 - Show details of staff
Press 11 - To view chart
Press 12 - To backup the database
Press 13 - To exit
Enter option number: 9
--------------------------------------------------
                  REMOVE STAFF
--------------------------------------------------

Enter staff ID to remove: 2234
Staff member removed successfully!
```

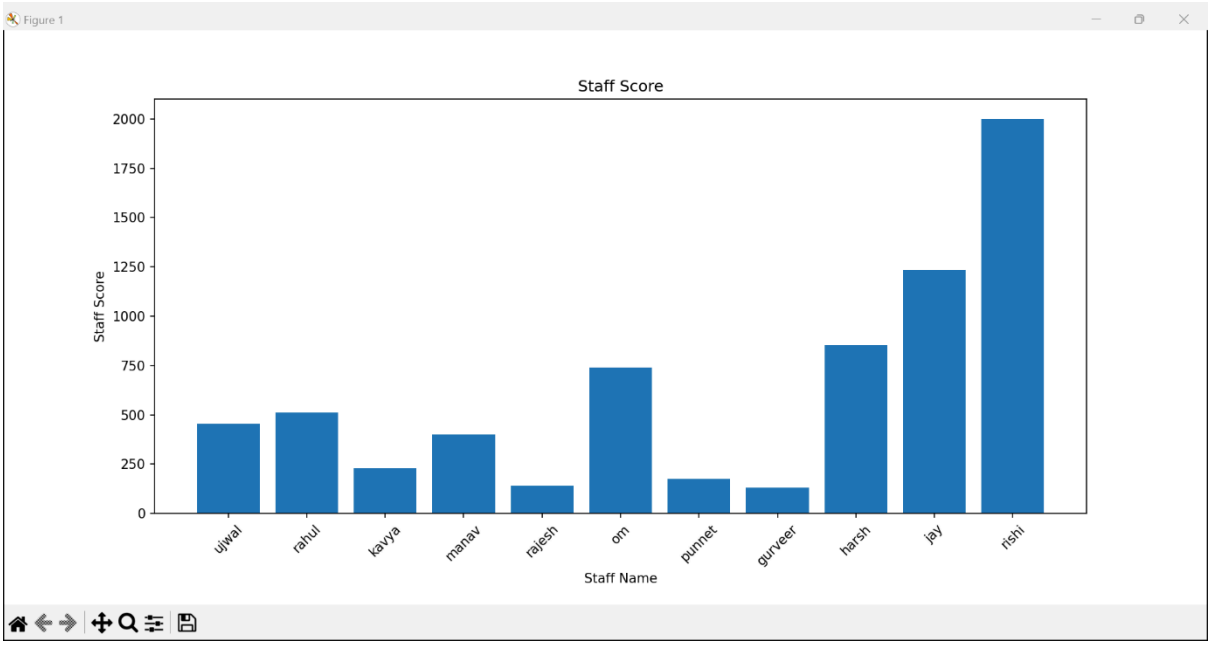
```
Staff Menu:
Press 1 - Add new item
Press 2 - Add item status
Press 3 - Search for an item
Press 4 - Remove an item
Press 5 - Show all items
Press 6 - Show all item statuses
Press 7 - Add new staff
Press 8 - Search for staff
Press 9 - Remove staff
Press 10 - Show details of staff
Press 11 - To view chart
Press 12 - To backup the database
Press 13 - To exit
Enter option number: 10
```

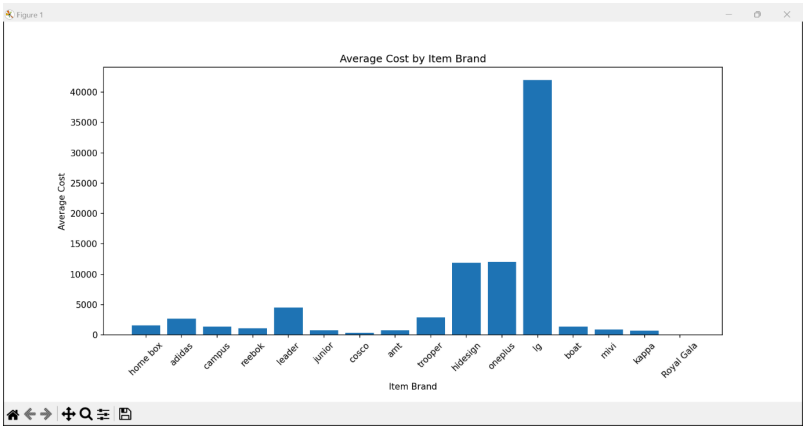
```
--------------------------------------------------
               SHOW STAFF DETAILS
--------------------------------------------------

Staff Details:
       name  staff_id        password  phone_number  staff_score
0     ujwal      2232          worker    7000453459          455
1     rahul      2234          worker    6574352134          510
2     kavya      2235          worker     765486223          230
3     manav      2236          worker    6893457891          400
4    rajesh      2237  crane-operator    8953462431          140
5        om      2238      accountant    7643981290          740
6    punnet      2239        security    4821983673          175
7   gurveer      2241         manager    5698345791          132
8     harsh      2242         manager    4572195672          853
9       jay      1234          worker      12345668         1234
10    rishi      2289             CEO       2898988         2000
11     Evan     30482         Manager     508094714           95

Staff Menu:
Press 1 - Add new item
Press 2 - Add item status
Press 3 - Search for an item
Press 4 - Remove an item
Press 5 - Show all items
```

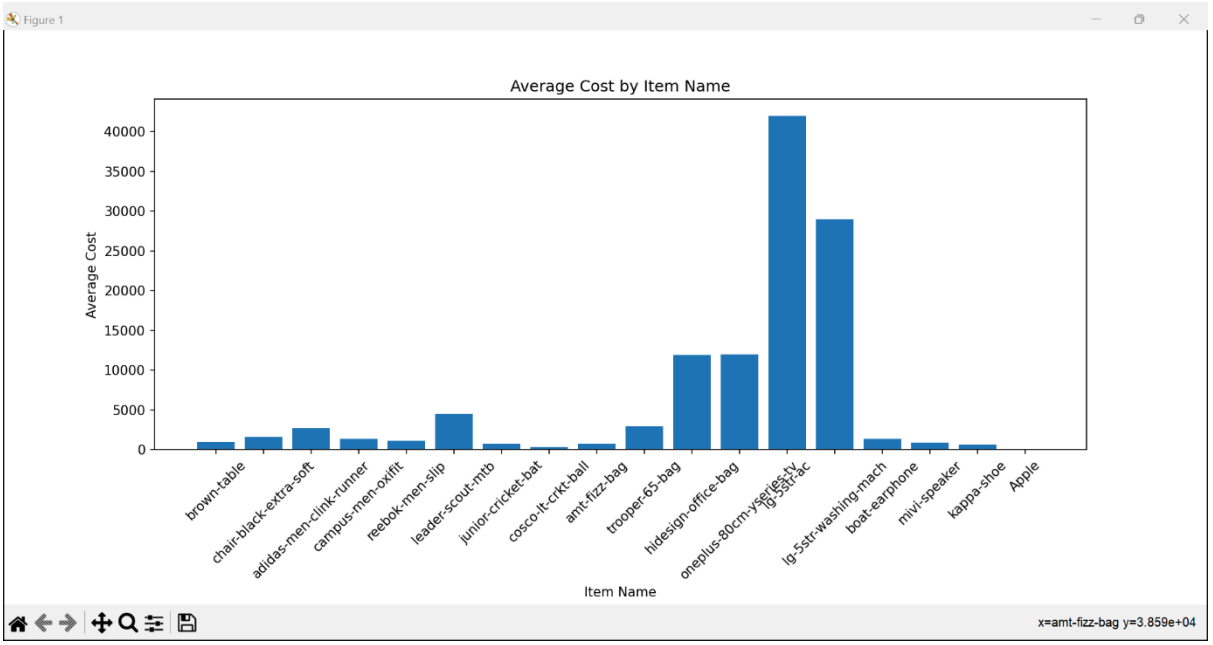
```
Staff Menu:
Press 1 - Add new item
Press 2 - Add item status
Press 3 - Search for an item
Press 4 - Remove an item
Press 5 - Show all items
Press 6 - Show all item statuses
Press 7 - Add new staff
Press 8 - Search for staff
Press 9 - Remove staff
Press 10 - Show details of staff
Press 11 - To view chart
Press 12 - To backup the database
Press 13 - To exit
```

```
Enter option number: 11
------------------------------------------------
            VIEW CHART AND STATISTICS
------------------------------------------------
```






```
Additional Statistics:
Most Common Brand: home box
Least Common Brand: adidas
Top 5 Most Expensive Items:
                item_name     cost
12              lg-5str-ac  41990.0
13    lg-5str-washing-mach  28990.0
11  oneplus-80cm-yseries-tv  11999.0
10     hidesign-office-bag  11895.0
5         leader-scout-mtb   4499.0
Top 5 Least Expensive Items:
           item_name   cost
7   cosco-lt-crkt-ball  365.0
16         kappa-shoe  689.0
6   junior-cricket-bat  749.0
8        amt-fizz-bag  749.0
15       mivi-speaker  899.0
Highest Quantity Item:
      item_name  quantity
15  mivi-speaker       237
Lowest Quantity Item:
        item_name  quantity
5  leader-scout-mtb         5
Average Price: 6439.78
Mode Price: 749.00
Median Price: 1399.00
Price Variance: 128865700.18
Price Standard Deviation: 11351.90
Quartiles:
- Q1 (lower half price median): 924.00
- Q2 (median): 1399.00
- Q3 (upper half price median): 4099.00
Interquartile Range (IQR): 3175.00
```

```
Staff Menu:
Press 1 - Add new item
Press 2 - Add item status
Press 3 - Search for an item
Press 4 - Remove an item
Press 5 - Show all items
Press 6 - Show all item statuses
Press 7 - Add new staff
Press 8 - Search for staff
Press 9 - Remove staff
Press 10 - Show details of staff
Press 11 - To view chart
Press 12 - To backup the database
Press 13 - To exit
Enter option number: 12
```

```
------------------------------------------------
                BACKUP DATABASE
------------------------------------------------

Backing up data...
User Credentials
('4AWJA5QHVR', 'Hemanth', 'hemanthsai137@gmail.com', 'HemanthSai')
('YAF6PBXKF9', 'Rishi Shah', 'rishishah@gmail.com', 'rishi_123')
('Z9L5SKOALE', 'hemanth', 'hemanth@gmail.com', 'hemanth')
```

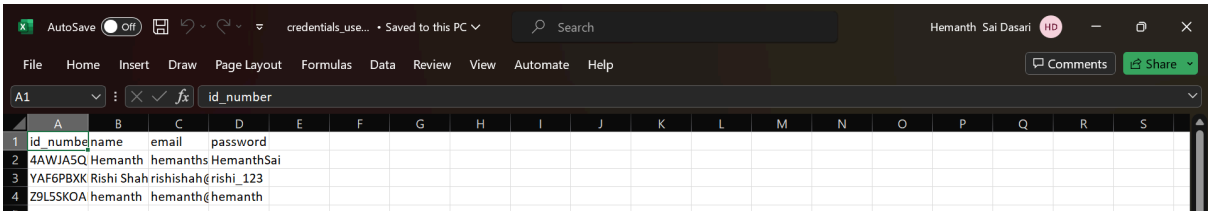
```
Staff Credentials
('jay', 1234, 'worker', 12345668, 1234)
('ujwal', 2232, 'worker', 7000453459, 455)
('rahul', 2234, 'worker', 6574352134, 510)
('kavya', 2235, 'worker', 765486223, 230)
('manav', 2236, 'worker', 6893457891, 400)
('rajesh', 2237, 'crane-operator', 8953462431, 140)
('om', 2238, 'accountant', 7643981290, 740)
('punnet', 2239, 'security', 4821983673, 175)
('gurveer', 2241, 'manager', 5698345791, 132)
('harsh', 2242, 'manager', 4572195672, 853)
('rishi', 2289, 'CEO', 2898988, 2000)
('Evan', 30482, 'Manager', 508094714, 95)
```

```
Stock details
(1, 'kappa-shoe', 'kappa', 689, 'shoe', 'shipped', 27)
(267, 'mivi-speaker', 'mivi', 899, 'audio', 'scanned', 237)
(384, 'lg-5str-ac', 'lg', 41990, 'large appliance', 'scanned', 8)
(2345, 'boat-earphone', 'boat', 1399, 'audio', 'shipped', 36)
(2535, 'lg-5str-washing-mach', 'lg', 28990, 'large appliance', 'scanned', 9)
(2869, 'hidesign-office-bag', 'hidesign', 11895, 'bags', 'shipped', 7)
(3386, 'reebok-men-slip', 'reebok', 1099, 'footwear', 'inventory ', 60)
(7686, 'amt-fizz-bag', 'amt', 749, 'bags', 'scanned', 48)
(9286, 'oneplus-80cm-yseries-tv', 'oneplus', 11999, 'large appliance', 'inventory ', 10)
(9862, 'cosco-lt-crkt-ball', 'cosco', 365, 'sports', 'scanned', 100)
(12345, 'Apple', 'Royal Gala', 30, 'food', 'out of stock', 2000)
(12566, 'brown-table', 'home box', 999, 'furniture', 'scanned', 10)
(12677, 'adidas-men-clink-runner', 'adidas', 2699, 'footwear', 'shipped', 60)
(12903, 'chair-black-extra-soft', 'home box', 1599, 'furniture', 'shipped', 13)
(28385, 'trooper-65-bag', 'trooper', 2899, 'bags', 'shipped', 37)
(28456, 'junior-cricket-bat', 'junior', 749, 'sports', 'inventory ', 10)
(82476, 'leader-scout-mtb', 'leader', 4499, 'sports', 'inventory ', 5)
(89568, 'Airpods Gen 2', 'Apple', 999, 'Electronics', 'Inventory', 80)
(327798, 'campus-men-oxifit', 'campus', 1399, 'footwear', 'inventory ', 150)
```

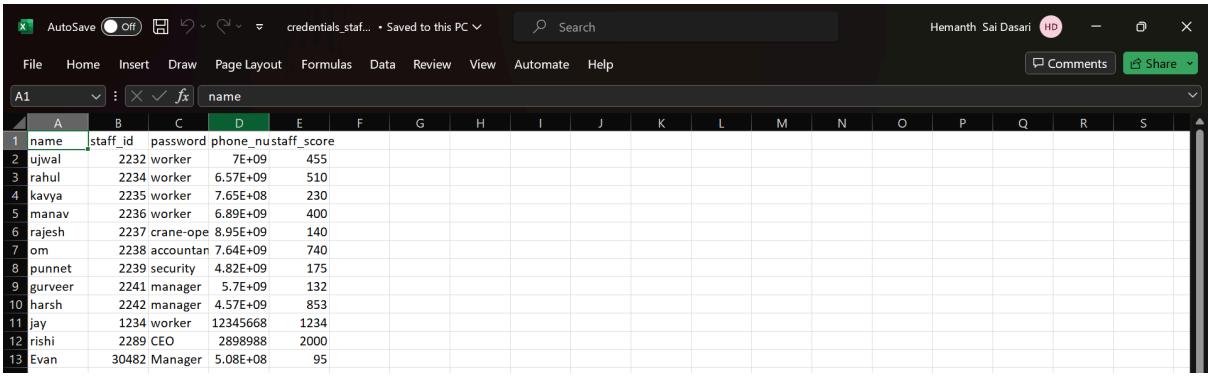
```
Order details
(117328, 'kappa-shoe (Quantity: 2) - Total Cost: $1378.00\nboat-earphone (Quantity: 3) - Total Cost: $4197.00', 'pending
')
(204086, '', 'pending')
(233064, 'boat-earphone (Quantity: 10) - Total Cost: $13990.00', 'shipped')
(246194, 'brown-table (Quantity: 3) - Total Cost: $nan\nchair-black-extra-soft (Quantity: 5) - Total Cost: $nan', 'pendi
ng')
(316299, 'brown-table (Quantity: 3) - Total Cost: $2997.00\nchair-black-extra-soft (Quantity: 3) - Total Cost: $4797.00'
, 'pending')
(463755, 'brown-table x2', 'pending')
(557943, 'brown-table x3', 'pending')
(608517, 'kappa-shoe (Quantity: 2) - Total Cost: $1378.00\nboat-earphone (Quantity: 1) - Total Cost: $1399.00', 'Invento
ry')
(628643, 'brown-table x2', 'pending')
(755612, 'brown-table x3', 'pending')
(756872, 'brown-table x3', 'pending')
(777249, 'brown-table x3', 'shipped')
(854459, 's x1, brown-table x39, chair-black-extra-soft x5', 'pending')


All data has been backed-up to your mysql database 'Inventory_Management_System'.
```

```
Staff Menu:
Press 1 - Add new item
Press 2 - Add item status
Press 3 - Search for an item
Press 4 - Remove an item
Press 5 - Show all items
Press 6 - Show all item statuses
Press 7 - Add new staff
Press 8 - Search for staff
Press 9 - Remove staff
Press 10 - Show details of staff
Press 11 - To view chart
Press 12 - To backup the database
Press 13 - To exit
Enter option number: 13
Exiting staff functionality...
```

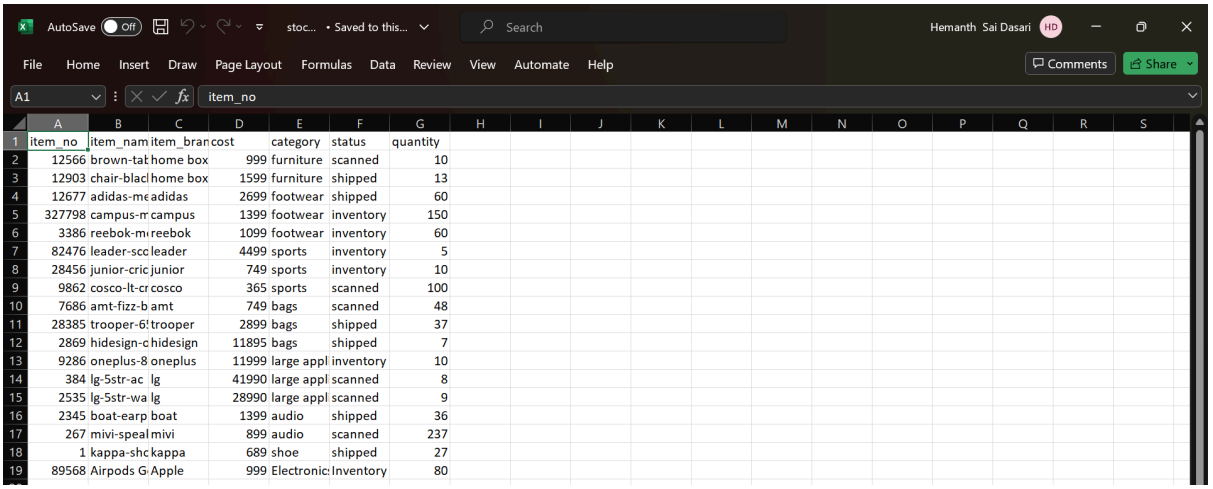
# # CSV FILES



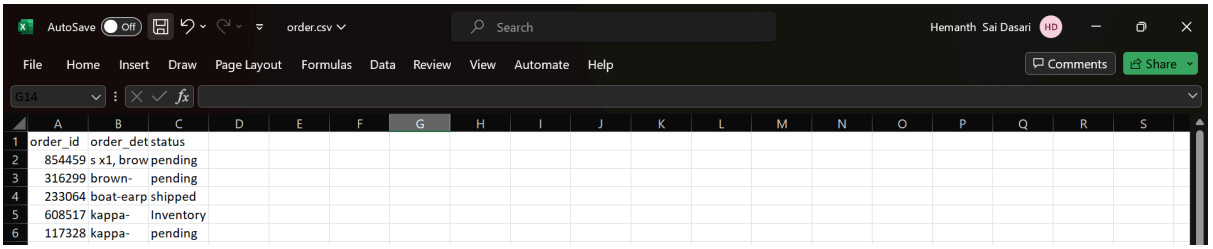
| id_number | name | email | password |
|---|---|---|---|
| 4AWJA5Q | Hemanth | hemanths | HemanthSai |
| YAF6PBXK | Rishi Shah | rishishah@ | rishi_123 |
| Z9L5SKOA | hemanth | hemanth@ | hemanth |



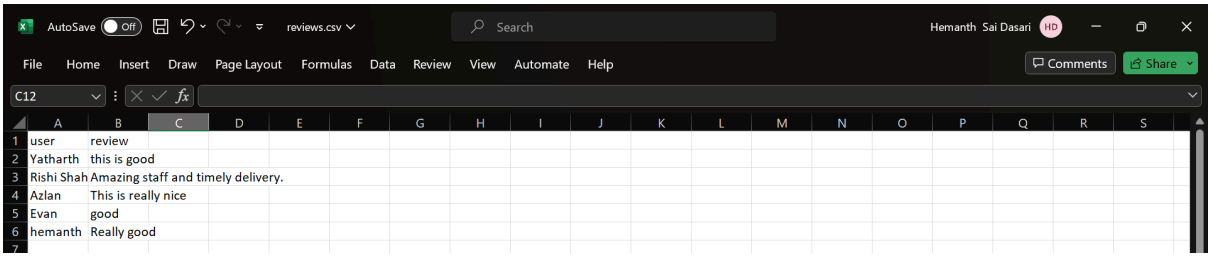
| name | staff_id | password | phone_nu | staff_score |
|---|---|---|---|---|
| ujwal | 2232 | worker | 7E+09 | 455 |
| rahul | 2234 | worker | 6.57E+09 | 510 |
| kavya | 2235 | worker | 7.65E+08 | 230 |
| manav | 2236 | worker | 6.89E+09 | 400 |
| rajesh | 2237 | crane-ope | 8.95E+09 | 140 |
| om | 2238 | accountan | 7.64E+09 | 740 |
| punnet | 2239 | security | 4.82E+09 | 175 |
| gurveer | 2241 | manager | 5.7E+09 | 132 |
| harsh | 2242 | manager | 4.57E+09 | 853 |
| jay | 1234 | worker | 12345668 | 1234 |
| rishi | 2289 | CEO | 2898988 | 2000 |
| Evan | 30482 | Manager | 5.08E+08 | 95 |



| item_no | item_nam | item_bran | cost | category | status | quantity |
|---|---|---|---|---|---|---|
| 12566 | brown-tat | home box | 999 | furniture | scanned | 10 |
| 12903 | chair-blac | home box | 1599 | furniture | shipped | 13 |
| 12677 | adidas-me | adidas | 2699 | footwear | shipped | 60 |
| 327798 | campus-m | campus | 1399 | footwear | inventory | 150 |
| 3386 | reebok-m | reebok | 1099 | footwear | inventory | 60 |
| 82476 | leader-scc | leader | 4499 | sports | inventory | 5 |
| 28456 | junior-cric | junior | 749 | sports | inventory | 10 |
| 9862 | cosco-lt-cr | cosco | 365 | sports | scanned | 100 |
| 7686 | amt-fizz-b | amt | 749 | bags | scanned | 48 |
| 28385 | trooper-6! | trooper | 2899 | bags | shipped | 37 |
| 2869 | hidesign-c | hidesign | 11895 | bags | shipped | 7 |
| 9286 | oneplus-8 | oneplus | 11999 | large appl | inventory | 10 |
| 384 | lg-5str-ac | lg | 41990 | large appl | scanned | 8 |
| 2535 | lg-5str-wa | lg | 28990 | large appl | scanned | 9 |
| 2345 | boat-earp | boat | 1399 | audio | shipped | 36 |
| 267 | mivi-speal | mivi | 899 | audio | scanned | 237 |
| 1 | kappa-shc | kappa | 689 | shoe | shipped | 27 |
| 89568 | Airpods G | Apple | 999 | Electronic | Inventory | 80 |



| order_id | order_det | status |
|---|---|---|
| 854459 | s x1, brow | pending |
| 316299 | brown- | pending |
| 233064 | boat-earp | shipped |
| 608517 | kappa- | Inventory |
| 117328 | kappa- | pending |



| user | review |
|---|---|
| Yatharth | this is good |
| Rishi Shah | Amazing staff and timely delivery. |
| Azlan | This is really nice |
| Evan | good |
| hemanth | Really good |

# 4. LIMITATIONS

I. **Time Efficiency**

May not work efficiently with the huge datasets of large corporations. Best suited for small company needs.

II. **Complexity**

Due to the technical nature of the software used in the inventory system, it can be difficult to thoroughly educate members of staff and customers on how to use the system effectively.

III. **Human Error**

An employee might enter data incorrectly, introducing inaccurate information to the system that can compromise decision-making. There is no system to check for this error.

# 5. BIBLIOGRAPHY

https://www.netsuite.com/portal/resource/articles/inventory-management/inventory-management-challenges.shtml

https://www.techwithtim.net/

https://stackoverflow.com/

**Code Reference**: Sumita Arora, NCERT and w3schools.