# 

**PROJECT ON ELECTRONIC SHOP**

**MANAGEMENT SYSTEM**



**Done by**:

Name :R.Muhammad Shaariq , Keerthivasan.T, Manikandan.M

Class & Sec : 12A

Roll No :12A33,12A31,12A32

**CERTIFICATE**

This is to certify that  **\_\_\_\_\_\_,** student of class XII of Amrita Vidyalayam Senior Secondary School has successfully completed the research on the project**, “Technosparkz”**, under the guidance of Mr**s. M. REKHA**  (subject teacher) during the academic year 2024 – 2025.

**Date:**

**Teacher in charge Principal’s Signature**

**(Smt. M. Rekha) (Smt. VaniSrinivasan)**

**External Examiner: Internal Examiner:**

**School Seal:**

**Acknowledgements**

I would like to express my heartfelt gratitude to all those who have supported and guided me throughout the preparation of this Chemistry project.

Firstly, I would like to thank the **Management** and **Swamini Bhakthipriyamrita Prana, Correspondent Amrita Vidyalayam TN & Pondicherry** of the school for providing the necessary resources and a conducive environment to carry out my project. Their encouragement has been a source of motivation throughout this endeavor.

I extend my sincere thanks to our **Principal Ms. B. Vani Srinivasan**, whose leadership and vision inspire us to excel in every aspect of our academic journey. Her constant support and belief in my abilities have helped me stay focused on my goals.

A special thanks to my **Subject Teacher Ms. M. Rekha**, who has been instrumental in guiding me through the various stages of the project. Their invaluable advice, deep knowledge, and continuous encouragement have greatly enriched my understanding of the subject.

I am also grateful to the **Lab Attender Mr .Mari Muthu**, who ensured that the lab was well-equipped and provided assistance whenever needed. Their cooperation and support have been crucial for the successful completion of my project.

I would like to acknowledge my **friends**, whose camaraderie and ideas have played a significant role in making this project a reality. Their help and moral support were essential during the entire process.

Lastly, I wish to express my deepest gratitude to my **family** for their constant love, understanding, and encouragement. They have been a source of strength and inspiration, without which this project would not have been possible.

Thank you all for your contribution to my academic success.

Name:SKM

# TABLE OF CONTENTS

* 1. ABSTRACT
  2. SYSTEM CONFIGURATION
  3. INTRODUCTION
  4. OBJECTIVES OF THE PROJECT
  5. PROPOSED SYSTEM
  6. LIBRARIES AND FUNCTIONS USED
  7. MYSQL QUERIES USED IN THIS PROJECT
  8. SOURCE CODE
  9. OUTPUT [SCREENSHOT]
  10. CONCLUSION
  11. BIBILIOGRAPHY

# ABSTRACT

**“Electronic shop”** project is developed using Python and MySQL Database. the

development of an Electronic Shop Program, designed to streamline the management

and operations of an electronic retail store. The program features a user-friendly

interface that facilitates inventory management, sales tracking, and customer

relationship management. Key functionalities include real-time inventory updates, sales

analytics, and a secure payment processing system, enabling staff to efficiently manage

transactions and monitor stock levels. Additionally, the program incorporates customer

profiles to enhance personalized marketing efforts and improve customer service. By

integrating modern technologies, the Electronic Shop Program aims to optimize

operational efficiency, enhance the shopping experience, and support strategic

decision-making, ultimately contributing to the shop's growth and success in

acompetitive market.

# SYSTEM CONFIGURATION

# HARD SPECIFICATIONS:

* CPU: Intel ® Core™ i5-750 CPU @ 3.40GHZ

3.14 GHz

* Installed memory (RAM): 4.00 GN (3.86 GB usable)
* System type: 64-bit Operating system, x64-based processor
* Pen and Touch: No Pen or Touch Input is available for this Display

**SOFTWATE SPECIFICATIONS:**

* Operating System: Windows 10 or UP
* Platform: Python IDLE 3.12 64-bit
* Database: MySQL
* Language: Python
* Computer Name: DESKTOP=B2T77J6

# INTRODUCTION

**"Electronic shop"** In the digital age, the retail landscape is rapidly evolving, with

electronic shops playing a pivotal role in providing consumers with access to a wide

range of electronic products. The Electronic Shop Project is an innovative software

application designed to address the complexities of managing an electronic retail

business. This project aims to create a robust system that simplifies various aspects of

shop operations, including inventory management, sales tracking, customer

engagement, and reporting.

The project is built on modern software development principles and employs a user-

friendly interface to ensure accessibility for both staff and customers. By integrating

essential functionalities such as product catalog management, order processing, and

customer relationship management, the system not only enhances operational

efficiency but also improves the overall shopping experience.

## **OBJECTIVES OF THE PROJECT**

The objective of this project is to let the students apply the programming knowledge into a real- world situation/problem and exposed the students how programming skills helps in developing a good software.

1. Write programs utilizing modern software tools.
2. Apply objects oriented programming principles effectively when developing small to medium sized projects.
3. Write effective procedural code to solve small to medium sized problems.
4. Students will demonstrate a breadth of knowledge in computer science, as exemplified in the areas of systems, theory and software development.
5. Students will demonstrate ability to conduct a research or applied Computer Science project, requiring writing and presentation skills which exemplify scholarly style in computer science.

**PROPOSED SYSTEM**

Today one cannot afford to rely on the fallible human beings of be really wants to stand against today’s merciless competition where not to wise saying “to err is human” no longer valid, it’s outdated to rationalize your mistake. So, to keep pace with time, to bring about the best result without malfunctioning and greater efficiency so to replace the unending heaps of flies with a muchsophisticated hard disk of the computer.

One has to use the data management software. Software has been an ascent in atomization various organisations. Many software products working are now in markets, which have helped making the organizations work easier and efficiently. Data management initially had to maintain a lot of ledgers and a lot of paperwork has to be done but now software product on this organization has made their work faster and easier. Now only this software has to be loaded on the computer and work can be done. This prevents a lot of time and money. The work becomes fully automated and any information regarding the organization can be obtained by clicking the button. Moreover, now it’s an age of computers of and automating such an organization gives the better look.

**LIBRARIES AND FUNCTIONS USED**

**mysqlconnector -** MySQL Connector/Python **enables Python programs to access MySQL databases** .It is written in pure Python and does not have any dependencies except for the Python Standard Library.

**Time-** The time module in Python is a standard module that provides functions for working with time and dates. It's built on top of the C runtime library.

# FUNCTIONS USED: WRITE IN DETAIL ABOUT PURPOSE OF THE FUNCTION.

* **cur.execute()** - To execute a given query.
* **con.commit()** – To store given data or record permanently.
* **cur.fetchall()** - The method fetches all rows of a query result set and returns a list of tuples.
* **time.sleep()** - The sleep() function suspends (delays) execution of the current thread for the given number of seconds.

# SOURCE CODE:

# FOR DATABASE CREATION:

import mysql.connector

def create\_database():

"""Create the database if it doesn't exist and return the connection and cursor."""

db\_connection = mysql.connector.connect(

host="localhost", # Change to your MySQL host

user="root", # Your MySQL username

password="urlakalangu" # Your MySQL password

)

cursor = db\_connection.cursor()

# Create the database if it doesn't exist

cursor.execute("CREATE DATABASE IF NOT EXISTS TechnoSparkz")

# Use the created database

cursor.execute("USE TechnoSparkz")

return db\_connection, cursor

def create\_tables(cursor, db\_connection):

"""Create tables in the database."""

# Create the admin table

cursor.execute("""

CREATE TABLE IF NOT EXISTS admin (

Sno INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(50),

Username VARCHAR(50),

Password VARCHAR(50)

)

""")

# Create the cart table

cursor.execute("""

CREATE TABLE IF NOT EXISTS cart (

Sno INT,

Brand VARCHAR(255),

Model VARCHAR(255),

Price INT

)

""")

# Create the devices table

cursor.execute("""

CREATE TABLE IF NOT EXISTS devices (

Sno INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(50)

)

""")

# Create the adapter table

cursor.execute("""

CREATE TABLE IF NOT EXISTS adapter (

Sno INT PRIMARY KEY AUTO\_INCREMENT,

Brand VARCHAR(50),

Model VARCHAR(100),

OutputPower VARCHAR(50),

PortModel VARCHAR(50),

Price INT

)

""")

# Create the monitor table

cursor.execute("""

CREATE TABLE IF NOT EXISTS monitor (

Sno INT PRIMARY KEY AUTO\_INCREMENT,

Brand VARCHAR(50),

Model VARCHAR(100),

Price INT

)

""")

# Create the keyboard table

cursor.execute("""

CREATE TABLE IF NOT EXISTS keyboard (

Sno INT PRIMARY KEY AUTO\_INCREMENT,

Brand VARCHAR(50),

Model VARCHAR(100),

Price INT

)

""")

# Create the mouse table

cursor.execute("""

CREATE TABLE IF NOT EXISTS mouse (

Sno INT PRIMARY KEY AUTO\_INCREMENT,

Brand VARCHAR(50),

Model VARCHAR(100),

Price INT

)

""")

# Create the headphone table

cursor.execute("""

CREATE TABLE IF NOT EXISTS headphone (

Sno INT PRIMARY KEY AUTO\_INCREMENT,

Brand VARCHAR(50),

Model VARCHAR(100),

Price INT

)

""")

# Commit the changes using the db\_connection object

db\_connection.commit()

def insert\_sample\_data(cursor, db\_connection):

"""Insert sample data into the tables."""

# Insert sample data into the admin table

cursor.executemany("""

INSERT INTO admin (Name, Username, Password)

VALUES (%s, %s, %s)

""", [

('Shaariq', 'MuhammadShaariq', 'urlakalangu'),

('Keerthi', 'Truly', 'BlackCape'),

('Mani Kandan', 'ex1', 'ps1')

])

# Insert categories into the devices table

cursor.executemany("""

INSERT INTO devices (name)

VALUES (%s)

""", [

('Adapter',),

('Monitor',),

('Keyboard',),

('Mouse',),

('Headphone',)

])

# Insert sample data into the adapter table

cursor.executemany("""

INSERT INTO adapter (Brand, Model, OutputPower, PortModel, Price)

VALUES (%s, %s, %s, %s, %s)

""", [

('Lenovo', 'U1000EA', '19W', 'USB Slim Tip', 1341),

('Dell', 'AC Adapter', '90W', 'Round', 2499),

('Dell', 'GaN Slim', '130W', 'USB-C', 11498),

('Lenovo', 'AC Adapter', '65W', 'USB Slim Port', 1021),

('Lenovo', 'AC Round Tip Adapter', '45W', 'Round Tip', 1041),

('HP', 'LC Power Adapter', '65W', 'USB-C', 2207),

('Acer', 'AC Power Adapter', '65W', 'USB-C', 2299),

('Acer', 'Small Pin Power Adapter', '65W', 'Small Pin', 1378)

])

# Insert sample data into the monitor table

cursor.executemany("""

INSERT INTO monitor (Brand, Model, Price)

VALUES (%s, %s, %s)

""", [

('Asus', 'ProArt Display PA278CFRV 100Hz QHD IPS', 31878),

('HP', 'Z Display Z22i 21.5-inch IPS LED Backlit Monitor', 14093),

('Acer', 'KA222Q E0 54.6 cm (21.5") Full HD IPS Panel', 5599),

('Dell', '27 Monitor - S2721HN', 12499),

('Zebronics', 'MT52-ZEB A19HD 46.9 cm (18.5 inch)', 5599),

('Samsung', 'LS24C366EAWXXL 24-inch Full HD VA Panel', 7499)

])

# Insert sample data into the keyboard table

cursor.executemany("""

INSERT INTO keyboard (Brand, Model, Price)

VALUES (%s, %s, %s)

""", [

('Zebronics', 'ZEBRONICS TRANSFORMER PRO Gaming Wireless Keyboard', 1800),

('Hp', 'HP 230 Wireless Black Keyboard', 1250),

('Lenovo', 'Lenovo 510 Wireless Keyboard', 1700),

('Asus', 'ASUS Membrane USB Gaming Keyboard for PC - TUF K1', 3400),

('Dell', 'Dell-KB700, Multi-Device Wireless Keyboard', 5260)

])

# Insert sample data into the mouse table

cursor.executemany("""

INSERT INTO mouse (Brand, Model, Price)

VALUES (%s, %s, %s)

""", [

('Logitech', 'Logitech G304 Lightspeed Wireless Gaming Mouse', 2995),

('Zebronics', 'ZEBRONICS-Transformer-M High-Performance USB Mouse', 400),

('HP', 'HP M260 RGB Backlighting USB Wired Gaming Mouse', 472),

('Dell', 'Dell WM118 Wireless Mouse', 580)

])

# Insert sample data into the headphone table

cursor.executemany("""

INSERT INTO headphone (Brand, Model, Price)

VALUES (%s, %s, %s)

""", [

('Boat', 'boAt Rockerz 450 Bluetooth On Ear Headphones with Mic', 1700),

('Zebronics', 'ZEBRONICS DUKE Wireless Headphone', 1400),

('Sony', 'Sony WH-CH520, Wireless On-Ear Bluetooth Headphones', 4500),

('Logitech', 'Logitech G733 Lightspeed Wireless Gaming Headset', 14900)

])

# Commit the changes using the db\_connection object

db\_connection.commit()

def setup\_database():

"""Setup the database, create tables, and insert sample data."""

db\_connection, cursor = create\_database()

create\_tables(cursor, db\_connection)

insert\_sample\_data(cursor, db\_connection)

# Close the connection after setting up the database

cursor.close()

db\_connection.close()

print("Database setup complete.")

if \_\_name\_\_ == "\_\_main\_\_":

try:

setup\_database()

except mysql.connector.Error as err:

print(f"Error: {err}")

else:

print("Database setup completed successfully.")

**SOURCE CODE**

**FOR TECHNOSPARKZ PROGRAM:**

import mysql.connector

# Connect to the database

mydb = mysql.connector.connect(

host='localhost',

user='root',

passwd='urlakalangu',

database='TechnoSparkz'

)

skm = mydb.cursor()

# Utility function to create the cart table

def create\_cart\_table():

skm.execute("CREATE TABLE IF NOT EXISTS cart (Sno INT, Brand VARCHAR(255), Model VARCHAR(255), Price INT);")

print("Cart table ready.")

# Function to add an item to the cart

def add\_to\_cart(sno, table\_name):

skm.execute(f"SELECT Sno, Brand, Model, Price FROM {table\_name} WHERE Sno = %s", (sno,))

item = skm.fetchone()

if item:

skm.execute("INSERT INTO cart (Sno, Brand, Model, Price) VALUES (%s, %s, %s, %s)", item)

mydb.commit()

print(f"Item '{item[2]}' added to cart.") # Item's Model

else:

print("Error: Item not found.")

# Function to display the cart

def view\_cart():

skm.execute("SELECT \* FROM cart;")

cart\_items = skm.fetchall()

if not cart\_items:

print("Your cart is empty.")

else:

print("\nYour Cart:")

print("Sno\tBrand\t\tModel\t\tPrice")

for item in cart\_items:

print(f"{item[0]}\t{item[1]}\t{item[2]}\t₹{item[3]}")

total = sum(item[3] for item in cart\_items)

print(f"Total: ₹{total}")

# Customer flow

def customer\_flow():

create\_cart\_table()

categories = {

1: "Adapter",

2: "Monitor",

3: "Keyboard",

4: "Mouse",

5: "Headphone"

}

while True:

print("\nAvailable Categories:")

for key, value in categories.items():

print(f"{key}. {value}")

category\_choice = int(input("Choose a category (1-5): "))

if category\_choice in categories:

table\_name = categories[category\_choice]

skm.execute(f"SELECT \* FROM {table\_name};")

items = skm.fetchall()

print("\nProducts Available:")

if table\_name == "Adapter":

print("Sno\tBrand\t\tModel\t\tOutput Power\tPort Model\tPrice")

for item in items:

print(f"{item[0]}\t{item[1]}\t{item[2]}\t{item[3]}\t{item[4]}\t₹{item[5]}")

else:

print("Sno\tBrand\t\tModel\t\tPrice")

for item in items:

print(f"{item[0]}\t{item[1]}\t{item[2]}\t₹{item[3]}")

sno = int(input("Enter the Serial Number (Sno) of the product to add to cart: "))

add\_to\_cart(sno, table\_name)

else:

print("Invalid category choice!")

another = input("Would you like to add more items? (yes/no): ").strip().lower()

if another != 'yes':

break

view\_cart()

checkout = input("Would you like to checkout? (yes/no): ").lower()

if checkout == "yes":

print("Thank you for shopping with us!")

skm.execute("DELETE FROM cart;") # Clear cart after checkout

mydb.commit()

# Admin flow

def admin\_flow():

username = input("Enter username: ")

password = input("Enter password: ")

skm.execute("SELECT \* FROM admin WHERE Username = %s AND Password = %s", (username, password))

admin = skm.fetchone()

if admin:

print(f"Welcome, {admin[1]}!")

while True:

print("\nAdmin Menu:")

print("1. View Stock")

print("2. Exit")

choice = int(input("Choose an option: "))

if choice == 1:

print("\nAvailable Stock Tables:")

stock\_tables = {

1: "Adapter",

2: "Monitor",

3: "Keyboard",

4: "Mouse",

5: "Headphone"

}

for key, value in stock\_tables.items():

print(f"{key}. {value}")

print("6. Exit")

table\_choice = int(input("Choose a stock table to view: "))

if table\_choice in stock\_tables:

table\_name = stock\_tables[table\_choice]

skm.execute(f"SELECT \* FROM {table\_name};")

items = skm.fetchall()

print(f"\n{table\_name} Stock:")

for item in items:

print(item)

elif table\_choice == 6:

break

else:

print("Invalid choice!")

elif choice == 2:

print("Goodbye!")

break

else:

print("Invalid choice!")

else:

print("Invalid username or password!")

# Main Program

if \_\_name\_\_ == "\_\_main\_\_":

print("\nWelcome to TechnoSparkz!")

print("Are you:")

print("1. Admin")

print("2. Customer")

role = int(input("Choose your role (1 or 2): "))

if role == 1:

admin\_flow()

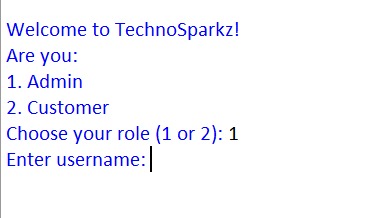
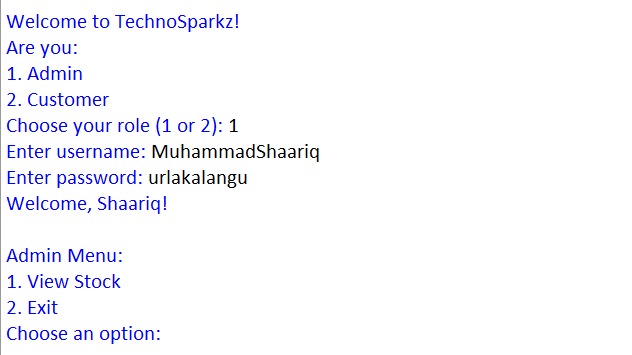
elif role == 2:

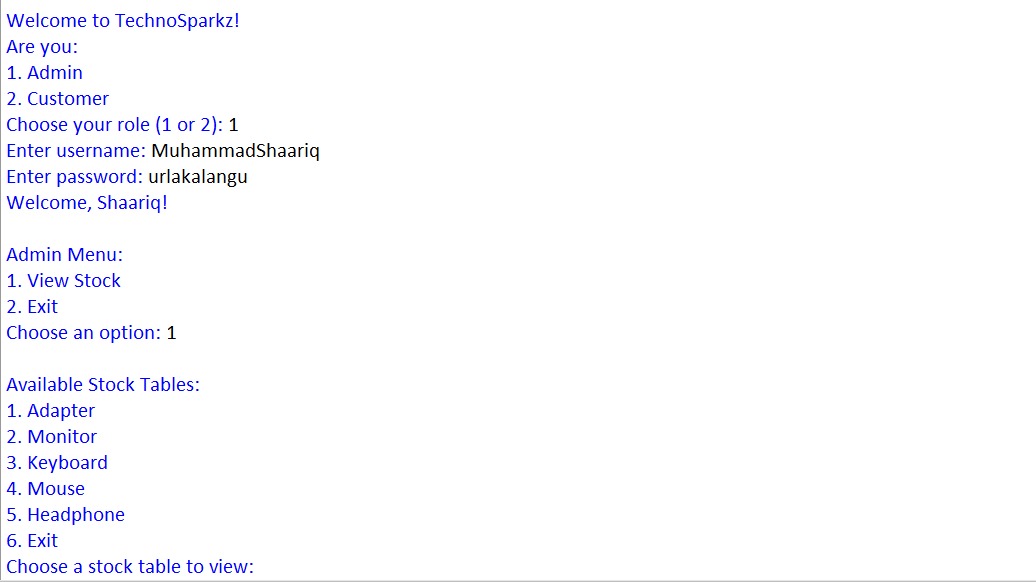
customer\_flow()

else:

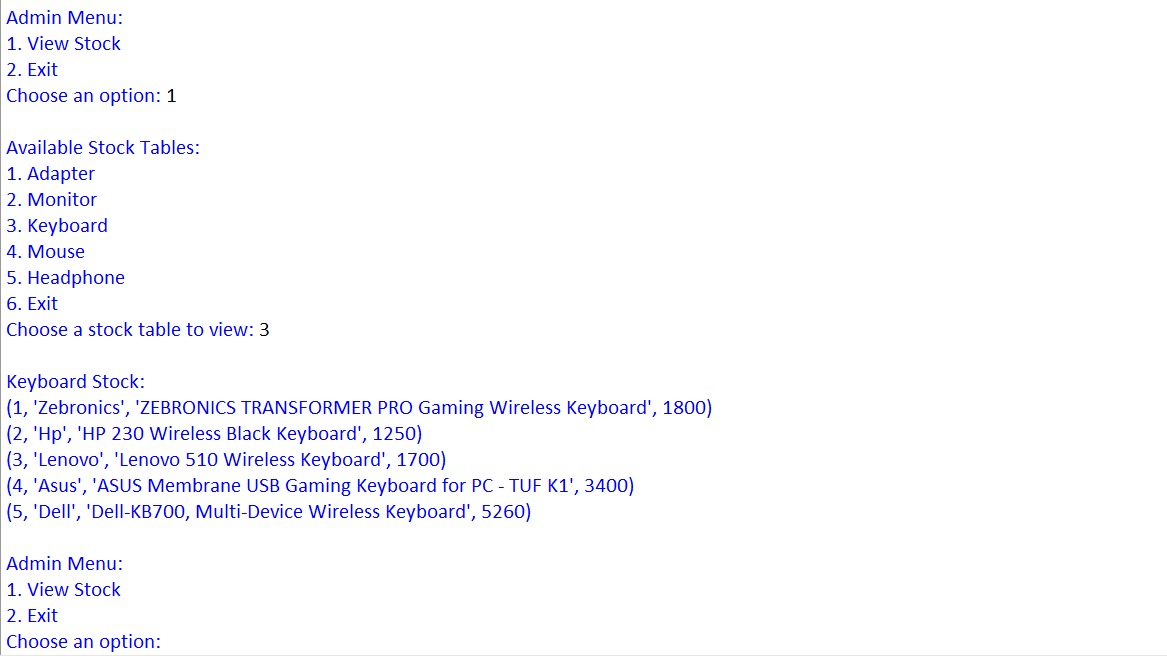
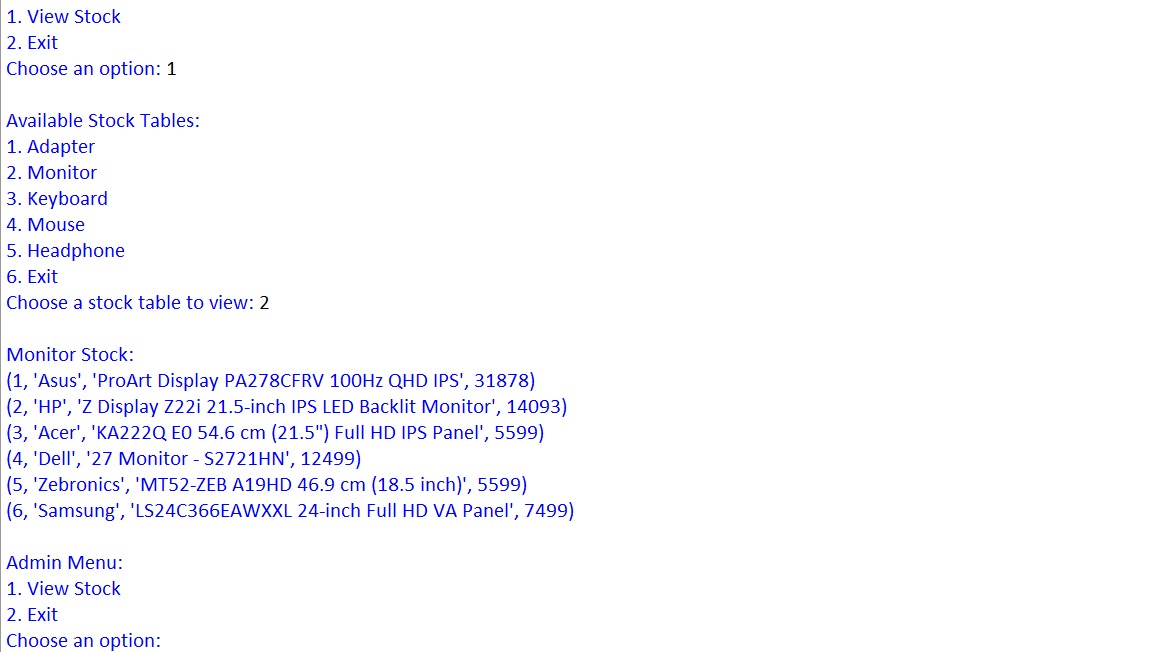
print("Invalid role selected!")

**OUTPUT:**

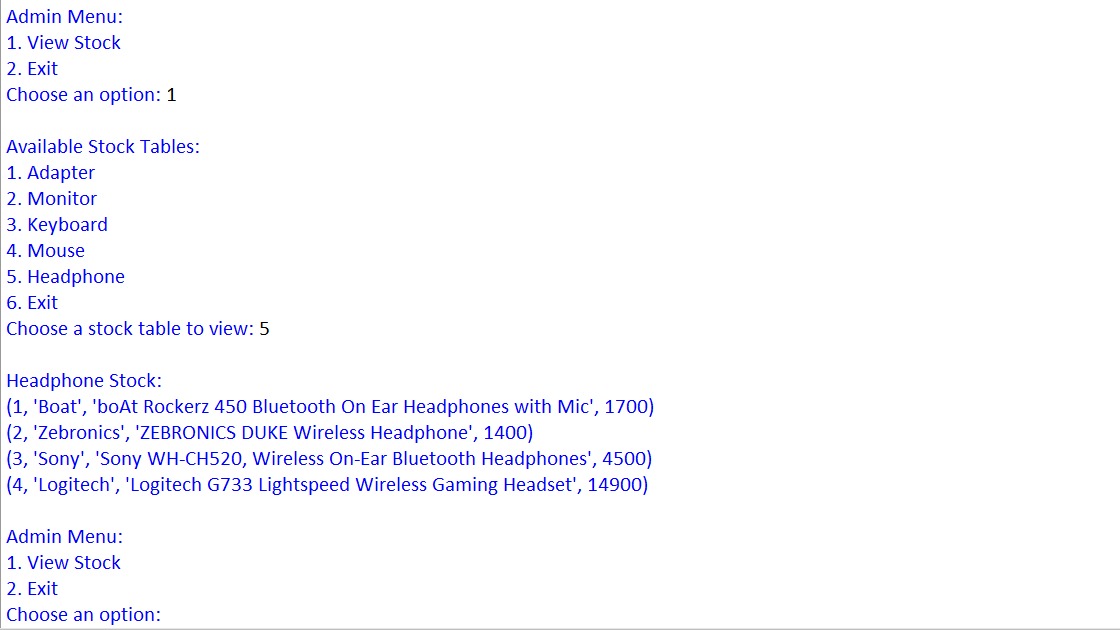
****

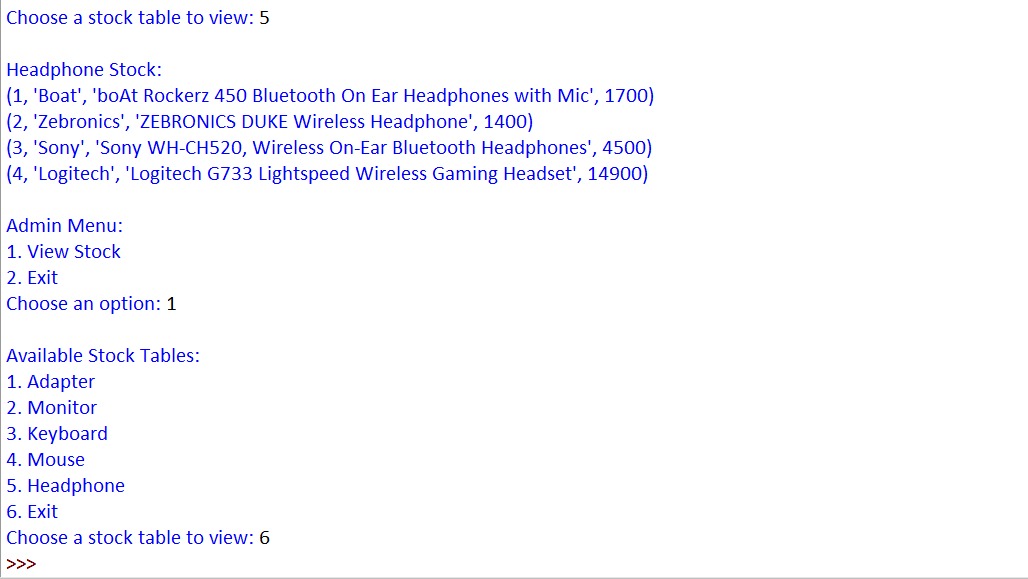
****

****

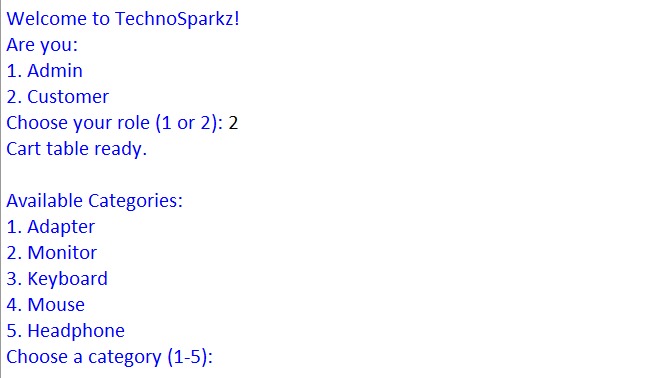
****

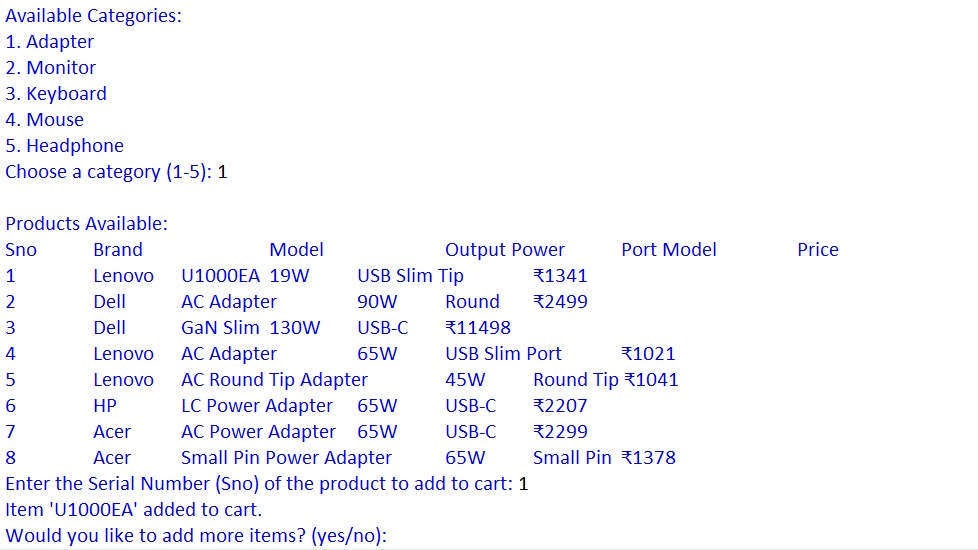
****

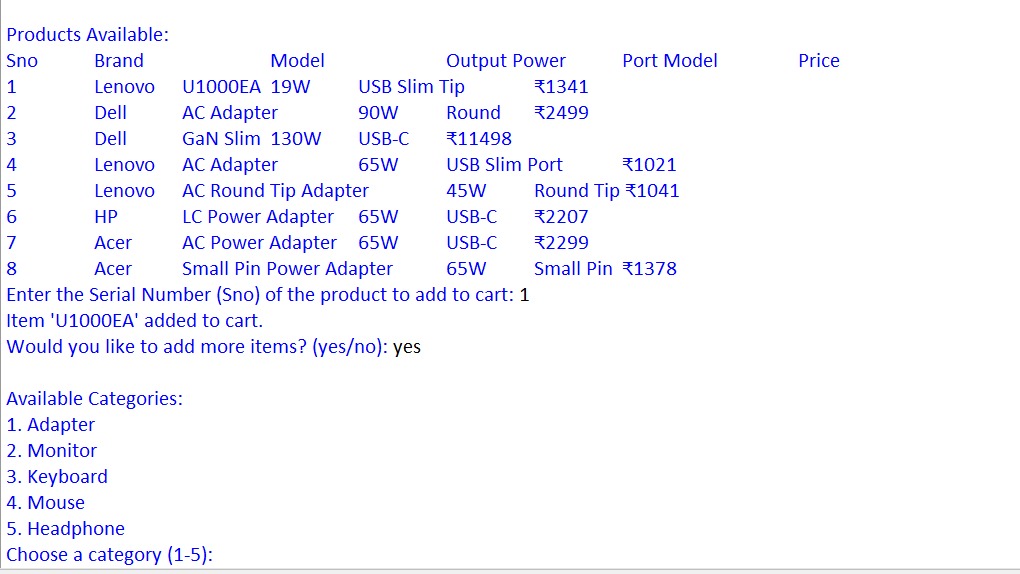
****

****

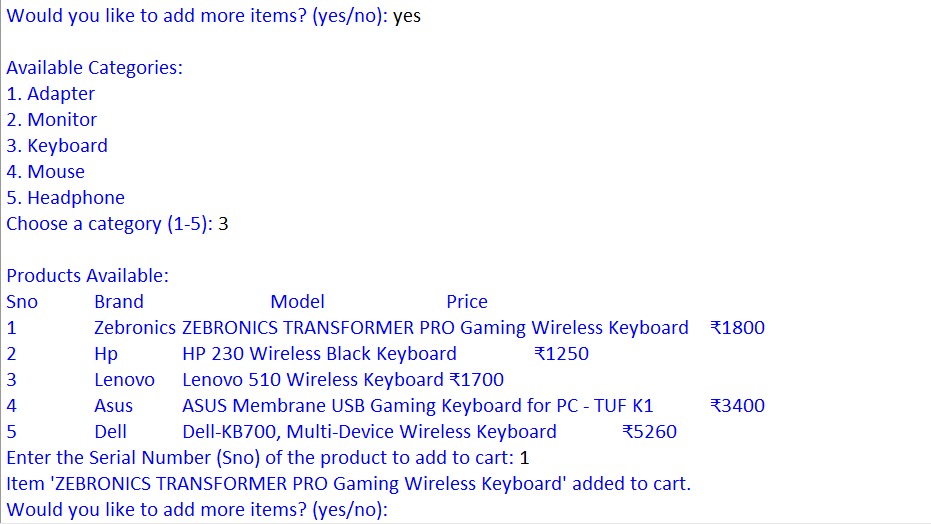
**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

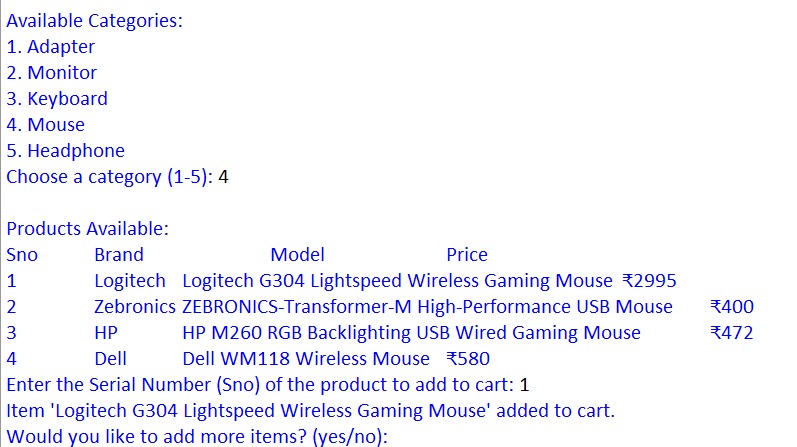
****

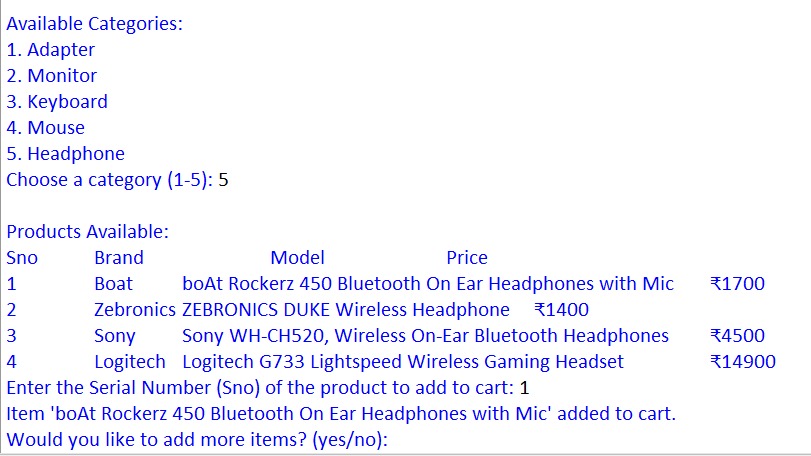
****

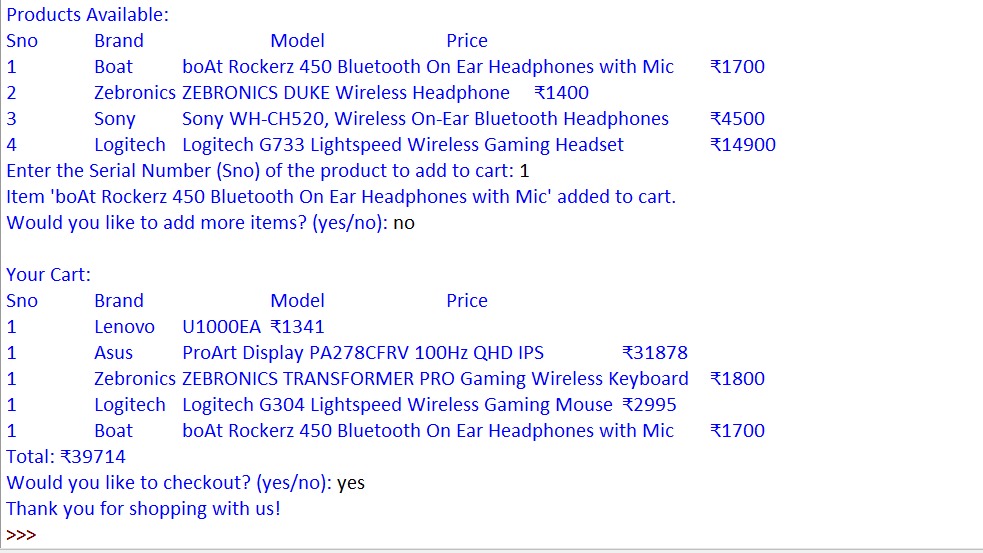
****

****

****

****

****

****

**CONCLUSION**

# BIBILIOGRAPHY