

PM SHRI KENDRIYA VIDYALAYA
BERHAMPUR



SESSION 2024-25

A Project Report on
PET SHOP MANAGEMENT

For
CBSE 2025 Examination

[As a part of the Computer Science Course (083)]

SUBMITTED BY :

SUBMITTED TO:

YOUR NAME: **RAMESWAR BEHERA**

Mr. S. K. MISRA

CLASS: **XII A**

PGT (Computer Science)

ROLL NO: **27**



THIS IS TO CERTIFY THAT RAMESWAR BEHERA STUDYING IN CLASS XII A, HAS
SATISFACTORILY COMPLETED PROJECT WITH THE TITLE PET SHOP MANAGEMENT
UNDER THE GUIDANCE OF Mr. SAROJ KANTA MISRA, PGT (COMPUTER SCIENCE)
DURING THE ACADEMIC YEAR 2024-25 IN PARTIAL FULFILLMENT OF "COMPUTER
SCIENCE" PRACTICAL EXAMINATION OF CENTRAL BOARD OF SECONDARY
EXAMINATION (CBSE)

Internal Examiner

External Examiner

PRINCIPAL

ACKNOWLEDGEMENT

I warmly acknowledge the continuous encouragement and timely suggestions offered by our Principal **Mr. SHIVAPRIYA DASH**. I extend my hearty thanks for giving me the opportunity to make use of the facilities available in the campus to carry out the project successfully.

I am highly indebted to **Mr. SAROJ KANTA MISRA**, (PGT Computer Science), for the constant supervision, providing necessary information and supporting in completing the project. I would like to express my gratitude towards them for their kind cooperation and encouragement.

Finally, I extend my gratefulness to one and all who are directly or indirectly involved in the successful completion of this project work.

Name: **RAMESWAR BEHERA**

Sign:

Class XII-A

CONTENTS

1. Introduction

2. System Implementation

2.1 The Hardware used

2.2 The Software's used

3. System Design & Development

3.1 Python Coding

3.2 Database

3.3 Output Screen

4. References

INTRODUCTION

“PET SHOP MANAGEMENT”

The Pet Shop Management System is a versatile and user-friendly software solution crafted to address the diverse needs of a pet shop. This project integrates advanced features to streamline the daily operations of managing customers, pets, accessories, and orders, ensuring efficient business processes and customer satisfaction. Built using Python for its logic and MySQL as its database backbone, the system combines robust functionality with ease of use.

The software caters to two main user groups: customers and administrators. Customers can register, browse available pets and accessories, place orders, and manage their profiles, while administrators have access to powerful tools for managing inventory, monitoring shop performance, and overseeing user activities. The program includes features like secure user authentication, detailed search filters, inventory updates, and real-time data integration, ensuring that the system adapts to the dynamic nature of a pet shop's operations.

For pet shop owners, the application reduces manual workload by automating essential tasks, such as tracking available stock, updating customer orders, and managing revenues. Customers benefit from a seamless shopping experience, with the ability to search for specific pet breeds, compare prices, and choose from a variety of accessories.

The Pet Shop Management System also emphasizes data integrity and security by implementing primary keys, foreign keys, and constraints within the database, ensuring that critical business data is stored reliably. By incorporating modern programming practices, the system is not only a practical tool for managing day-to-day operations but also an example of how software can enhance the efficiency and professionalism of retail businesses.

This project demonstrates the application of programming and database management to solve real-world challenges, offering a solution that is both functional and scalable. It is a testament to the power of technology in transforming traditional businesses into efficient, data-driven enterprises.



- ❖ Python is a high-level language. It is a free and open-source language.
- ❖ It is an interpreted language, as Python programs are executed by an interpreter.
- ❖ Python programs are easy to understand as they have a clearly defined syntax and relatively simple structure.
- ❖ Python is case-sensitive. For example, NUMBER and number are not same in Python.
- ❖ Python is portable and platform independent, means it can run on various operating systems and hardware platforms.
- ❖ Python has a rich library of predefined functions.
- ❖ Python is also helpful in web development. Many popular web services and applications are built using Python.
- ❖ Python uses indentation for blocks and nested blocks.

System Implementation

Hardware used:

While developing the software, the used hardware's are: PC with Intel Core i3 processor having 4.00 GB RAM and other required devices.

Software used:

- Microsoft Windows® 10 as Operating System.
- Python IDLE as Front-end Development environment.
- MS-Word 2010 for documentation.

Python Coding:

```
import mysql.connector
```

```
from datetime import date
```

```
mydb=mysql.connector.connect(host="localhost",user="root",password="admin",database="pet_management")
```

```
def create_table():
```

```
    cb = mydb.cursor()
```

```
    cb.execute("CREATE TABLE IF NOT EXISTS customers(cust_ID INT AUTO_INCREMENT PRIMARY KEY, cust_name VARCHAR(100) NOT NULL, cust_Address VARCHAR(255), cust_number VARCHAR(15) NOT NULL, cust_username VARCHAR(50) UNIQUE NOT NULL, cust_password VARCHAR(100) NOT NULL)")
```

```
    cb.execute("CREATE TABLE IF NOT EXISTS pets(pet_ID INT AUTO_INCREMENT PRIMARY KEY, pet_type VARCHAR(50) NOT NULL, pet_breed VARCHAR(50), pet_age INT(3) CHECK (pet_age >= 3), pet_gender VARCHAR(20) CHECK (pet_gender IN ('Male', 'Female', 'Unknown')), pet_price DECIMAL(10, 2) NOT NULL, pet_description VARCHAR(250), pet_quantity INT(3) NOT NULL CHECK (pet_quantity >= 0))")
```

```
    cb.execute("CREATE TABLE IF NOT EXISTS pet_accessories(accessory_ID INT AUTO_INCREMENT PRIMARY KEY, accessory_name VARCHAR(100), accessory_type VARCHAR(100), accessory_price INT, accessory_description VARCHAR(100), accessory_quantity INT(3) CHECK (accessory_quantity >= 0))")
```

```
    cb.execute("CREATE TABLE IF NOT EXISTS user_orders(order_ID INT AUTO_INCREMENT PRIMARY KEY, cust_ID INT(3) NOT NULL, pet_ID INT(3), accessory_ID INT(3), total_pet_quantity INT(3) NOT NULL DEFAULT 0, total_accessory_quantity INT(3) NOT NULL DEFAULT 0, total_order_price DECIMAL(10,2) NOT NULL CHECK (total_order_price >= 0), order_date DATE NOT NULL , FOREIGN KEY (cust_ID) REFERENCES customers(cust_ID), FOREIGN KEY (pet_ID) REFERENCES pets(pet_ID), FOREIGN KEY (accessory_ID) REFERENCES pet_accessories(accessory_ID))")
```

```
    mydb.commit()
```

```
    cb.close()
```

```
create_table()
```

```
users={}
```

```
admin={"ADMIN":"12345"}
```

```
current_DATE = date.today()
```

```
def fetch_data_into_users():
```

```
    cb=mydb.cursor()
```

```
    data="SELECT  
cust_ID,cust_name,cust_Address,cust_number,cust_username,cust_password FROM  
customers"
```

```
    cb.execute(data)
```

```
    records=cb.fetchall()
```

```
    for record in records:
```

```
        cust_ID,name,address,number,username,password=record
```

```
        users[username]={'ID':cust_ID,'name':name,'address':address,'number':number  
, 'password':password}
```

```
        print('Data fetched and loaded into Users')
```

```
    mydb.commit()
```

```
    cb.close()
```

```
fetch_data_into_users()
```

```
def fetch_available_pets():
```

```
    cb = mydb.cursor()
```

```
    try:
```

```
        query = "SELECT pet_ID, pet_type, pet_breed, pet_age, pet_gender, pet_price,  
pet_description FROM pets WHERE pet_quantity > 0"
```

```
        cb.execute(query)
```

```
available_pets = cb.fetchall()

if available_pets:

    print("Available Pets:")

    for pet in available_pets:

        print(pet)

else:

    print("No pets are currently available.")

except Exception as e:

    print('Error occur: ',e)

finally:

    cb.close()
```

fetch_available_pets()

```
def fetch_available_pet_accessories():

    cb = mydb.cursor()

    try:

        query = "SELECT accessory_ID, accessory_name, accessory_type,
        accessory_price, accessory_description FROM pet_accessories WHERE
        accessory_quantity > 0"

        cb.execute(query)

        available_accessories = cb.fetchall()

        if available_accessories:

            print("Available Pet Accessories:")

            for accessory in available_accessories:

                print(accessory)

        else:

            print("No accessories are currently available.")
```

except Exception as e:

print("Error occurred: ", e)

finally:

cb.close()

fetch_available_pet_accessories()

def user_login():

print("\n" + "=" * 40)

print(" " * 10 + "USER LOGIN")

print("=" * 40)

username = input("Enter your username: ")

password = input("Enter your password: ")

if username in users and users[username]['password'] == password:

print("Login successful")

print("Enter 0 to logout")

print("Enter 1 to see user profile")

print("Enter 2 to see PET'S ")

print("Enter 3 to see PET'S Accessories")

print("Enter 4 to Place order")

try:

choice = int(input("Enter your choice: "))

if choice == 1:

user_profile(username)

elif choice == 2:

pet_table_for_user()

elif choice == 3:

pet_accessories_for_user()

```

    elif choice == 4:
        placing_order(username)
    elif choice == 0:
        signup_user()
    else:
        print("INVALID choice!!!")
except ValueError:
    print("Please enter a valid number.")
elif username not in users:
    print("Username Not Found")
    signup_user()
elif users[username]['password'] != password:
    print("Invalid Password. Try Again.")
    user_login()
else:
    print("Invalid username or password")
    user_login()
print("=" * 40)

```

```

def user_profile(username):
    print("\n" + "=" * 40)
    print(" " * 10 + "USER PROFILE")
    print("=" * 40)
    cb = mydb.cursor()
    try:
        print("Enter 0 to Previous Menu")
        print("Enter 1 for updating profile")
        print("Enter 2 for viewing profile")
    
```

```
print("Enter 3 for My Order Details")
choice = int(input("Enter Your Choice: "))
if choice == 1:
    user_update_profile(username)
elif choice == 2:
    user_viewing_profile(username)
elif choice == 3:
    user_order_details(username)
elif choice == 0:
    print("Going back to the previous menu...")
    user_login()
else:
    print("INVALID choice!!! Please try again.")
except ValueError:
    print("Please enter a valid number.")
except Exception as e:
    print("An error occurred: ", e)
finally:
    cb.close()
    print("=" * 40)
```

```
def pet_table_for_user():
    print("\n" + "=" * 40)
    print(" " * 10 + "PET MANAGEMENT OPTIONS")
    print("=" * 40)
    print("Enter 0 to Previous Menu")
    print("Enter 1 for Search pets")
    print("Enter 2 for To show all pets")
```

```

cb = mydb.cursor()

try:
    choice = int(input("Enter your choice: "))

    if choice == 1:
        search_pets()

    elif choice == 2:
        try:
            query = "SELECT pet_ID, pet_type, pet_breed, pet_age, pet_gender,
pet_price, pet_description, pet_quantity FROM pets"
            cb.execute(query)
            pets = cb.fetchall()

            if pets:
                print("\n" + "=" * 40)
                print(" " * 10 + "AVAILABLE PETS")
                print("=" * 40)

                for pet in pets:
                    print("ID: {}, Type: {}, Breed: {}, Age: {}, Gender: {}, Price: ${},
Description: {}, Quantity Available: {}".format(pet[0], pet[1], pet[2], pet[3], pet[4],
pet[5], pet[6], pet[7]))

            else:
                print("No pets available.")
                user_login()

        except Exception as e:
            print("Error occurred: ", e)

        finally:
            cb.close()

    elif choice == 0:
        print("Going back to the previous menu...")

    return

```

else:

```
print("INVALID choice!!! Please try again.")
```

```
pet_table_for_user()
```

except ValueError:

```
print("Please enter a valid number.")
```

```
pet_table_for_user()
```

finally:

```
print("=" * 40)
```

```
def pet_accessories_for_user():
```

```
    print("\n" + "=" * 40)
```

```
    print(" " * 10 + "SHOWING PET ACCESSORIES")
```

```
    print("=" * 40)
```

```
    print("Enter 0 to Previous menu")
```

```
    print("Enter 1 for Search accessories")
```

```
    print("Enter 2 for To show all accessories")
```

try:

```
    choice = int(input("Enter your choice: "))
```

if choice == 1:

```
    search_pet_accessories()
```

elif choice == 2:

```
    cb = mydb.cursor()
```

try:

```
        query = "SELECT accessory_ID, accessory_name, accessory_type,  
accessory_price, accessory_description, accessory_quantity FROM pet_accessories"
```

```
        cb.execute(query)
```

```
        accessories = cb.fetchall()
```

if accessories:

```

        print("\n" + "=" * 40)

        print(" " * 10 + "AVAILABLE PET ACCESSORIES")

        print("=" * 40)

        for accessory in accessories:

            print("ID: {}, Name: {}, Type: {}, Price: ${}, Description: {}, Quantity
Available: {}".format( accessory[0], accessory[1], accessory[2], accessory[3],
accessory[4], accessory[5]))

        else:

            print("No accessories available.")

    except Exception as e:

        print("Error occurred: ", e)

    finally:

        cb.close()

    elif choice == 0:

        print("Going back to the previous menu...")

        user_login()

    else:

        print("INVALID choice!!! Please try again.")

        pet_accessories_for_admin()

except ValueError:

    print("Please enter a valid number.")

    pet_accessories_for_admin()

finally:

    print("=" * 40)


def placing_order(username):

    print("\n" + "=" * 40)

    print(" " * 10 + "PLACE AN ORDER")

```

```

print("=" * 40)

cb = mydb.cursor()

cb.execute("SELECT pet_ID, pet_type, pet_price, pet_quantity FROM pets
WHERE pet_quantity > 0")

available_pets = cb.fetchall()

print("Available Pets:")

for pet in available_pets:

    print("ID: {}, Type: {}, Price: ${}, Quantity Available: {}".format(pet[0], pet[1],
pet[2], pet[3]))

    pet_id = int(input("Enter the Pet ID you want to order: "))

    pet_quantity = int(input("Enter the quantity of the pet you want to order: "))

    cb.execute("SELECT pet_price, pet_quantity FROM pets WHERE pet_ID = %s",
(pet_id,))

    pet_details = cb.fetchone()

    if pet_details and pet_details[1] >= pet_quantity:

        total_pet_price = pet_details[0] * pet_quantity

    else:

        print("Selected pet is not available in the requested quantity.")

        user_login()

        return

    cb.execute("SELECT accessory_ID, accessory_name, accessory_price,
accessory_quantity FROM pet_accessories WHERE accessory_quantity > 0")

    available_accessories = cb.fetchall()

    print("Available Accessories:")

    for accessory in available_accessories:

        print("ID: {}, Name: {}, Price: ${}, Quantity Available: {}".format(accessory[0],
accessory[1], accessory[2], accessory[3]))

        accessory_id = int(input("Enter the Accessory ID you want to order (or enter 0 if
none): "))

        total_accessory_price = 0

```

```

accessory_quantity = 0

if accessory_id != 0:

    accessory_quantity = int(input("Enter the quantity of the accessory you want to
order: "))

    cb.execute("SELECT accessory_price, accessory_quantity FROM
pet_accessories WHERE accessory_ID = %s", (accessory_id,))

    accessory_details = cb.fetchone()

    if accessory_details and accessory_details[1] >= accessory_quantity:

        total_accessory_price = accessory_details[0] * accessory_quantity
    else:

        print("Selected accessory is not available in the requested quantity.")

        user_login()

        return

total_order_price = total_pet_price + total_accessory_price

order_date = date.today()

cust_id = users[username]['ID']

insert_order_query = "INSERT INTO user_orders (cust_ID, pet_ID,
accessory_ID, total_pet_quantity, total_accessory_quantity, total_order_price,
order_date)VALUES (%s, %s, %s, %s, %s, %s, %s)"

cb.execute(insert_order_query, (cust_id, pet_id, accessory_id if accessory_id != 0
else None, pet_quantity, accessory_quantity, total_order_price, order_date))

cb.execute("UPDATE pets SET pet_quantity = pet_quantity - %s WHERE pet_ID =
%s", (pet_quantity, pet_id))

if accessory_id != 0:

    cb.execute("UPDATE pet_accessories SET accessory_quantity =
accessory_quantity - %s WHERE accessory_ID = %s", (accessory_quantity,
accessory_id))

mydb.commit()

print("Order placed successfully!")

```

```
print("Total Order Price: ${}".format(total_order_price))
```

```
user_login()
```

```
cb.close()
```

```
print("=" * 40)
```

```
def user_update_profile(username):
```

```
    print("\n" + "=" * 40)
```

```
    print(" " * 10 + "MY PROFILE UPDATE")
```

```
    print("=" * 40)
```

```
    cb = mydb.cursor()
```

```
    if username in users:
```

```
        print("Enter 0 to Previous Menu")
```

```
        print("Enter 1 to update Name")
```

```
        print("Enter 2 to update Address")
```

```
        print("Enter 3 to update Phone Number")
```

```
        print("Enter 4 to update Password")
```

```
        try:
```

```
            choice = int(input("Enter your choice: "))
```

```
            if choice == 1:
```

```
                new_name = input("Enter new name: ")
```

```
                users[username]['name'] = new_name
```

```
                update_query = "UPDATE customers SET cust_name = %s WHERE  
cust_username = %s"
```

```
                cb.execute(update_query, (new_name, username))
```

```
                print("Name updated successfully for ", username)
```

```
            elif choice == 2:
```

```
                new_address = input("Enter new address: ")
```

```
                users[username]['address'] = new_address
```

```

        update_query = "UPDATE customers SET cust_Address = %s WHERE
cust_username = %s"

        cb.execute(update_query, (new_address, username))

        print("Address updated successfully for ", username)

    elif choice == 3:

        new_number = input("Enter new phone number: ")

        users[username]['number'] = new_number

        update_query = "UPDATE customers SET cust_number = %s WHERE
cust_username = %s"

        cb.execute(update_query, (new_number, username))

        print("Phone number updated successfully for ", username)

    elif choice == 4:

        new_password = input("Enter new password: ")

        users[username]['password'] = new_password

        update_query = "UPDATE customers SET cust_password = %s WHERE
cust_username = %s"

        cb.execute(update_query, (new_password, username))

        print("Password updated successfully for ", username)

    elif choice == 0:

        print("Going back to user profile...")

        user_login()

    else:

        print("INVALID choice!!!")

    mydb.commit()

    print("Profile updated successfully for the User", username)

    user_update_profile()

except Exception as e:

    print("An unexpected error occurred: ", e)

    user_update_profile()

```

finally:

```
cb.close()
```

```
print("=" * 40)
```

else:

```
print("User not found.")
```

```
print("=" * 40)
```

```
def user_viewing_profile(username):
```

```
    """Displays the user's profile information."""
```

```
    print("\n" + "=" * 40)
```

```
    print(" " * 10 + "MY PROFILE VIEW")
```

```
    print("=" * 40)
```

```
    cb = mydb.cursor()
```

try:

```
        query = "SELECT cust_username, cust_name, cust_Address, cust_number FROM  
customers WHERE cust_username = %s"
```

```
        cb.execute(query, (username,))
```

```
        profile = cb.fetchone()
```

if profile:

```
    print("\n" + "=" * 40)
```

```
    print(" " * 10 + "USER PROFILE DETAILS")
```

```
    print("=" * 40)
```

```
    print("Username: ", profile[0])
```

```
    print("Name: ", profile[1])
```

```
    print("Address: ", profile[2])
```

```
    print("Phone Number: ", profile[3])
```

```
        order_query = "SELECT order_ID, total_order_price, order_date FROM  
user_orders WHERE cust_ID = (SELECT cust_ID FROM customers WHERE  
cust_username = %s)"
```

```
cb.execute(order_query, (username,))
orders = cb.fetchall()
if orders:
    print("\n" + "=" * 40)
    print(" " * 10 + "ORDER DETAILS")
    print("=" * 40)
    for order in orders:
        print("Order ID: ", order[0])
        print("Total Price: ${}".format(order[1]))
        print("Order Date: ", order[2])
        print("-----")
else:
    print("No orders found for this user.")
```

```
else:
    print("No profile found for this user.")
    user_login()
```

```
except Exception as e:
    print("An unexpected error occurred: ", e)
    user_login()
```

```
finally:
    cb.close()
    print("=" * 40)
```

```
def user_order_details(username):
    print("\n" + "=" * 40)
    print(" " * 10 + "MY ORDERS DETAILS")
    print("=" * 40)
    cb = mydb.cursor()
```

try:

```
query = "SELECT order_ID, pet_ID, accessory_ID, total_pet_quantity,  
total_accessory_quantity, total_order_price, order_date FROM user_orders WHERE  
cust_ID = (SELECT cust_ID FROM customers WHERE cust_username = %s)"
```

```
cb.execute(query, (username,))
```

```
order_details = cb.fetchall()
```

```
if order_details:
```

```
orders = []
```

```
for order in order_details:
```

```
    order_dict = {'order_id': order[0], 'pet_id': order[1], 'accessory_id':  
order[2], 'pet_quantity': order[3], 'accessory_quantity': order[4], 'total_price':  
order[5], 'order_date': order[6]}
```

```
    orders.append(order_dict)
```

```
print("\n" + "=" * 40)
```

```
print(" " * 10 + "ORDER DETAILS")
```

```
print("=" * 40)
```

```
for order in orders:
```

```
    cb.execute("SELECT pet_type, pet_price FROM pets WHERE pet_ID =  
%s", (order['pet_id'],))
```

```
    pet_details = cb.fetchone()
```

```
    cb.execute("SELECT accessory_type, accessory_price FROM  
pet_accessories WHERE accessory_ID = %s", (order['accessory_id'],))
```

```
    accessory_details = cb.fetchone()
```

```
    print("Order ID: ", order['order_id'])
```

```
if pet_details:
```

```
    print("Pet Type: ", pet_details[0])
```

```
    print("Pet Price: ${}".format(pet_details[1]))
```

```
else:
```

```
    print("No pet details found for this order.")
```

```
if accessory_details:
```

```

        print("Accessory Type: ", accessory_details[0])
        print("Accessory Price: ${}".format(accessory_details[1]))
    else:
        print("No accessory details found for this order.")
        print("Pet Quantity: ", order['pet_quantity'])
        print("Accessory Quantity: ", order['accessory_quantity'])
        print("Total Price: ${}".format(order['total_price']))
        print("Order Date: ", order['order_date'])
        print("-----")
        user_login()
    else:
        print("No orders found for this user.")
        user_login()
except Exception as e:
    print("Error occurred: ", e)
    user_login()
finally:
    cb.close()
    print("=" * 40)

```

```

def search_pets():
    print("\n" + "=" * 40)
    print(" " * 10 + "SEARCH FOR PETS")
    print("=" * 40)
    print("0. Previous Menu")
    print("1. Pet Type")
    print("2. Pet Breed")
    print("3. Pet Age")

```

```
print("4. Pet Gender")
print("5. Pet Price")
cb = mydb.cursor()
try:
    choice = int(input("Enter your choice: "))
    if choice == 1:
        pet_type = input("Enter pet type (e.g., Dog, Cat): ")
        query = "SELECT * FROM pets WHERE pet_type = %s"
        cb.execute(query, (pet_type,))
    elif choice == 2:
        breed = input("Enter pet breed: ")
        query = "SELECT * FROM pets WHERE pet_breed = %s"
        cb.execute(query, (breed,))
    elif choice == 3:
        age = int(input("Enter pet age: "))
        query = "SELECT * FROM pets WHERE pet_age = %s"
        cb.execute(query, (age,))
    elif choice == 4:
        gender = input("Enter pet gender (e.g., Male, Female): ")
        query = "SELECT * FROM pets WHERE pet_gender = %s"
        cb.execute(query, (gender,))
    elif choice == 5:
        price = float(input("Enter maximum price: "))
        query = "SELECT * FROM pets WHERE pet_price <= %s"
        cb.execute(query, (price,))
    elif choice == 0:
        pet_table_for_user()
else:
```

```

        print("Invalid choice. Please try again.")
        search_pets()
    pets = cb.fetchall()
    if pets:
        print("\n" + "=" * 40)
        print(" " * 10 + "SEARCH RESULTS")
        print("=" * 40)
        for pet in pets:
            print("Pet ID: {}, Type: {}, Breed: {}, Age: {}, Gender: {}, Price:
${}".format(pet[0], pet[1], pet[2], pet[3], pet[4], pet[5]))
        else:
            print("No pets found matching your criteria.")
            search_pets()
    except Exception as e:
        print("An error occurred: ", e)
        search_pets()
    finally:
        cb.close()
        print("=" * 40)

def search_pet_accessories():
    print("\n" + "=" * 40)
    print(" " * 10 + "SEARCH FOR PET ACCESSORIES")
    print("=" * 40)
    print("0. Previous Menu")
    print("1. Accessory Type")
    print("2. Accessory Name")
    print("3. Accessory Price")

```

```

cb = mydb.cursor()

try:
    choice = int(input("Enter your choice: "))

    if choice == 1:
        accessory_type = input("Enter accessory type (e.g., Collar, Toy): ")
        query = "SELECT * FROM pet_accessories WHERE accessory_type = %s"
        cb.execute(query, (accessory_type,))

    elif choice == 2:
        accessory_name = input("Enter accessory name: ")
        query = "SELECT * FROM pet_accessories WHERE accessory_name LIKE %s"
        cb.execute(query, ('%' + accessory_name + '%',))

    elif choice == 3:
        price = float(input("Enter maximum price: "))
        query = "SELECT * FROM pet_accessories WHERE accessory_price <= %s"
        cb.execute(query, (price,))

    elif choice == 0:
        pet_accessories_for_user()

    else:
        print("Invalid choice. Please try again.")
        search_pet_accessories()

accessories = cb.fetchall()

if accessories:
    print("\n" + "=" * 40)
    print(" " * 10 + "SEARCH RESULTS")
    print("=" * 40)

    for accessory in accessories:

```

```
print("Accessory ID: {}, Name: {}, Type: {}, Price: ${}, Description: {},  
Quantity Available: {}".format(accessory[0], accessory[1], accessory[2], accessory[3],  
accessory[4], accessory[5]))
```

```
else:
```

```
print("No accessories found matching your criteria.")
```

```
search_pet_accessories()
```

```
except Exception as e:
```

```
print("An error occurred: ", e)
```

```
search_pet_accessories()
```

```
finally:
```

```
cb.close()
```

```
print("=" * 40)
```

```
def admin_login():
```

```
print("\n" + "=" * 40)
```

```
print(" " * 10 + "ADMIN LOGIN")
```

```
print("=" * 40)
```

```
admin_username = input("Enter your admin username: ")
```

```
admin_password = input("Enter your admin password: ")
```

```
if admin_username in admin and admin_password == admin[admin_username]:
```

```
print("Login successful")
```

```
print("Enter 0 to logout")
```

```
print("Enter 1 to view My profile Page")
```

```
print("Enter 2 to view User Page")
```

```
print("Enter 3 to view Pet Page")
```

```
try:
```

```
choice = int(input("Enter your choice: "))
```

```
if choice == 1:
```

```

        admin_profile()
    elif choice == 2:
        user_profile_details()
    elif choice == 3:
        pet_details()
    elif choice == 0:
        signup_admin()
    else:
        print("INVALID choice!!!")
        admin_login()
except ValueError:
    print("Enter Number Please!!!")
    admin_login()
else:
    print("Invalid admin username or password")
    admin_login()
print("=" * 40)

```

```

def new_user_login():
    """Handles new user login """
    cb=mydb.cursor()
    print("\n" + "=" * 40)
    print(" " * 10 + "NEW USER LOGIN")
    print("=" * 40)
    username = input("Enter your username: ")
    password = input("Enter your password: ")
    name = input("Enter your name: ")
    address = input("Enter your address: ")

```

```

number = input("Enter your number: ")

if username in users:
    print("Username already exists. Choose a different one.")
    new_user_login()
else:
    cust_details = "INSERT INTO customers(cust_name, cust_Address,
cust_number, cust_username, cust_password) VALUES (%s, %s, %s, %s, %s)"
    cb.execute(cust_details, (name, address, number, username, password))
    fetch_data_into_users()
    mydb.commit()
    print("User ", username, "registered successfully!")
    user_login()
print("=" * 40)

def signup_user():
    """Handles user signup """
    print("\n" + "=" * 40)
    print(" " * 10 + "USER SIGNUP")
    print("=" * 40)
    print("Enter 0 to Go back")
    print("Enter 1 for user login")
    print("Enter 2 for new user registration")
    try:
        x = int(input("Enter your choice: "))
        if x == 1:
            user_login()
        elif x == 2:
            new_user_login()

```

```
elif x == 0:
```

```
    main_menu()
```

```
else:
```

```
    print("Invalid choice. Please try again.")
```

```
    signup_user()
```

```
except ValueError:
```

```
    print("Invalid input. Please enter a number.")
```

```
    signup_user()
```

```
def admin_profile():
```

```
    print("\n" + "=" * 40)
```

```
    print(" " * 10 + "ADMIN PROFILE")
```

```
    print("=" * 40)
```

```
    print("Enter 0 to Previous Page")
```

```
    print("Enter 1 to view details")
```

```
    print("Enter 2 to Edit profile")
```

```
    print("Enter 3 to View Shop Status")
```

```
    try:
```

```
        choice = int(input("Enter your choice: "))
```

```
        if choice == 1:
```

```
            admin_view_profile()
```

```
        elif choice == 2:
```

```
            admin_update_profile()
```

```
        elif choice == 3:
```

```
            shop_status()
```

```
        elif choice == 0:
```

```
            admin_login()
```

```
    else:
```

```
print("INVALID choice!!! Please try again.")
```

```
admin_profile()
```

```
except ValueError:
```

```
print("Please enter a valid number.")
```

```
admin_profile()
```

```
finally:
```

```
print("=" * 40)
```

```
def admin_view_profile():
```

```
print("\n" + "=" * 40)
```

```
print(" " * 10 + "ADMIN PROFILE VIEW")
```

```
print("=" * 40)
```

```
admin_username = input("Enter your admin username: ")
```

```
if admin_username in admin:
```

```
print("\n" + "=" * 40)
```

```
print(" " * 10 + "ADMIN PROFILE DETAILS")
```

```
print("=" * 40)
```

```
print("Username: ", admin_username)
```

```
print("Password: ", admin[admin_username])
```

```
admin_profile()
```

```
else:
```

```
print("Admin username not found.")
```

```
admin_view_profile()
```

```
print("=" * 40)
```

```
def admin_update_profile():
```

```
print("\n" + "=" * 40)
```

```
print(" " * 10 + "ADMIN PROFILE UPDATE")
```

```
print("=" * 40)
print("Enter 0 to Go back")
print("Enter 1 to update Username")
print("Enter 2 to update Password")
username = input("Enter your old username: ")
if username not in admin:
    print("Admin username not found.")
    print("=" * 40)
    admin_update_profile()
try:
    choice = int(input("Enter your choice: "))
    if choice == 1:
        new_username = input("Enter new username: ")
        if new_username in admin:
            print("Username already exists. Choose a different one.")
        else:
            admin[new_username] = admin[username]
            admin.pop(username)
            print("Username updated successfully.")
    elif choice == 2:
        new_password = input("Enter new password: ")
        admin[username] = new_password
        print("Password updated successfully.")
    elif choice == 0:
        print("Going back to admin profile...")
        admin_profile()
    else:
        print("INVALID choice!!! Please try again.")
```

```
admin_update_profile()
```

```
except ValueError:
```

```
    print("Please enter a valid number.")
```

```
    admin_update_profile()
```

```
finally:
```

```
    print("=" * 40)
```

```
def shop_status():
```

```
    print("\n" + "=" * 40)
```

```
    print(" " * 10 + "SHOP STATUS")
```

```
    print("=" * 40)
```

```
    cb = mydb.cursor()
```

```
    try:
```

```
        cb.execute("SELECT COUNT(*) FROM pets")
```

```
        total_pets = cb.fetchone()[0]
```

```
        cb.execute("SELECT COUNT(*) FROM customers")
```

```
        total_customers = cb.fetchone()[0]
```

```
        cb.execute("SELECT COUNT(*) FROM pet_accessories")
```

```
        total_accessories = cb.fetchone()[0]
```

```
        cb.execute("SELECT SUM(pet_price) FROM pets")
```

```
        total_revenue_pets = cb.fetchone()[0] or 0
```

```
        cb.execute("SELECT SUM(accessory_price) FROM pet_accessories")
```

```
        total_revenue_accessories = cb.fetchone()[0] or 0
```

```
        cb.execute("SELECT COUNT(*) FROM user_orders")
```

```
        total_orders = cb.fetchone()[0]
```

```
    print("\n" + "=" * 40)
```

```
    print(" " * 10 + "SHOP STATISTICS")
```

```
    print("=" * 40)
```

```
print("Total Customers: {}".format(total_customers))
print("Total Pets Available: {}".format(total_pets))
print("Total Pet Accessories Available: {}".format(total_accessories))
print("Total Revenue from Pets: ${}".format(total_revenue_pets))
print("Total Revenue from Pet Accessories:
${}".format(total_revenue_accessories))

print("Total Orders Placed: {}".format(total_orders))

admin_profile()

print("=" * 40)

except Exception as e:

    print("An error occurred while checking shop status:", e)

    admin_profile()

finally:

    cb.close()

    print("=" * 40)
```

```
def user_profile_details():

    print("\n" + "=" * 40)

    print(" " * 10 + "USER PROFILE DETAILS")

    print("=" * 40)

    print("Enter 0 to Previous Menu")

    print("Enter 1 to add user")

    print("Enter 2 to delete user")

    print("Enter 3 to view user details")

    print("Enter 4 to update user details")

    try:

        choice = int(input("Enter your choice: "))

        if choice == 1:
```

```
        add_user()
    elif choice == 2:
        delete_user()
    elif choice == 3:
        view_user_details()
    elif choice == 4:
        update_user()
    elif choice == 0:
        admin_login()
    else:
        print("INVALID choice!!! Please try again.")
        user_profile_details()
except ValueError:
    print("Please enter a valid number.")
    user_profile_details()
finally:
    print("=" * 40)
```

```
def pet_details():
    print("\n" + "=" * 40)
    print(" " * 10 + "PET DETAILS")
    print("=" * 40)
    print("Enter 0 to Go back")
    print("Enter 1 for PET")
    print("Enter 2 for Pet Accessories")
    try:
        choice = int(input("Enter your choice: "))
        if choice == 1:
```

```
        pet_table_for_admin()
    elif choice == 2:
        pet_accessories_for_admin()
    elif choice == 0:
        admin_login()
    else:
        print("INVALID choice!!! Please try again.")
        pet_details()
except ValueError:
    print("Please enter a valid number.")
    pet_details()
finally:
    print("=" * 40)
```

```
def want_to_add_more_user():
    print("Do you want to add more user? (yes/no): ")
    choice = input().lower()
    if choice == 'yes':
        add_user()
    elif choice == 'no':
        user_profile_details()
```

```
def add_user():
    print("\n" + "=" * 40)
    print(" " * 10 + "ADDING NEW USER")
    print("=" * 40)
    cb = mydb.cursor()
    try:
```

```

username = input("Enter new username: ")

if username in users:

    print("Username already exists. Choose a different one.")

    return

name = input("Enter user's name: ")
address = input("Enter user's address: ")
number = input("Enter user's phone number: ")
password = input("Enter user's password: ")

insert_user_query = "INSERT INTO customers (cust_name, cust_Address,
cust_number, cust_username, cust_password) VALUES (%s, %s, %s, %s, %s)"

cb.execute(insert_user_query, (name, address, number, username, password))

mydb.commit()

users[username] = {'ID': len(users) + 1, 'name': name, 'address': address,
'number': number, 'password': password}

print("=" * 40)

print(" " * 10 + "USER ADDED SUCCESSFULLY!")

print(" " * 10 + "Username: {}".format(username))

print("=" * 40)

want_to_add_more_user()

except Exception as e:

    print("Error occurred:", e)

    add_user()

finally:

    cb.close()

    print("=" * 40)

def want_to_delete_more_user():

    choice = input("Do you want to delete more user? (yes/no): ").lower()

```

```

if choice == 'yes':
    delete_user()
elif choice == 'no':
    user_profile_details()

def delete_user():
    """Deletes a user from the database."""
    print("\n" + "=" * 40)
    print(" " * 10 + "DELETE USER")
    print("=" * 40)
    cb = mydb.cursor()
    try:
        username = input("Enter the username of the user you want to delete: ")
        if username not in users:
            print("User not found.")
            return
        confirm = input("Are you sure you want to delete the user '{}'? (yes/no): ".format(username)).lower()
        if confirm == 'yes':
            delete_user_query = "DELETE FROM customers WHERE cust_username = %s"
            cb.execute(delete_user_query, (username,))
            mydb.commit()
            del users[username]
            print("User '{}' has been successfully deleted.".format(username))
            want_to_delete_more_user()
        else:
            print("Deletion cancelled.")
            want_to_delete_more_user()
    except:
        print("An error occurred.")
        want_to_delete_more_user()

```

except Exception as e:

print("Error occurred: ", e)

delete_user()

finally:

cb.close()

print("=" * 40)

def view_user_details():

print("\n" + "=" * 40)

print(" " * 10 + "VIEWING USER DETAILS")

print("=" * 40)

cb = mydb.cursor()

try:

print("Enter 1 to view all users")

print("Enter 2 to search for a user by username")

choice = int(input("Enter your choice: "))

if choice == 1:

query = "SELECT * FROM customers"

cb.execute(query)

users_list = cb.fetchall()

if users_list:

print("\n" + "=" * 40)

print(" " * 10 + "USER DETAILS")

print("=" * 40)

for user *in* users_list:

print("ID: {}, Name: {}, Address: {}, Phone: {}, Username:
{}, Address: {}".format(user[0], user[1], user[2], user[3], user[4], user[5]))

else:

```

        print("No users found.")

    elif choice == 2:

        username = input("Enter the username of the user you want to view: ")

        if username in users:

            user_details = users[username]

            print("\n" + "=" * 40)

            print(" " * 10 + "USER DETAILS FOR {}".format(username.upper()))

            print("=" * 40)

            print("ID: {}, Name: {}, Address: {}, Phone: {}".format(user_details['ID'],
user_details['name'], user_details['address'], user_details['number']))

        else:

            print("User not found.")

            user_profile_details()

    else:

        print("Invalid choice! Please try again.")

        view_user_details()

except Exception as e:

    print("Error occurred: ", e)

    user_profile_details()

finally:

    cb.close()

    print("=" * 40)

def want_to_update_more_user():

    choice = input("Do you want to update more user? (yes/no): ").lower()

    if choice == 'yes':

        update_user()

    else:

```

```
admin_login()
```

```
def update_user():
```

```
    """Updates user information."""
```

```
    print("\n" + "=" * 40)
```

```
    print(" " * 10 + "UPDATING USER")
```

```
    print("=" * 40)
```

```
    cb = mydb.cursor()
```

```
    try:
```

```
        username = input("Enter the username of the user you want to update: ")
```

```
        if username not in users:
```

```
            print("User not found.")
```

```
            want_to_update_more_user()
```

```
        print("\n" + "=" * 40)
```

```
        print(" " * 10 + "CURRENT USER DETAILS")
```

```
        print("=" * 40)
```

```
        user_details = users[username]
```

```
        print("ID: {}, Name: {}, Address: {}, Phone: {}".format(user_details['ID'],  
user_details['name'], user_details['address'], user_details['number']))
```

```
        print("=" * 40)
```

```
        print("Enter the details you want to update:")
```

```
        print("1. Update Name")
```

```
        print("2. Update Address")
```

```
        print("3. Update Phone Number")
```

```
        print("4. Update Password")
```

```
        print("0. Go Back")
```

```
        choice = int(input("Enter your choice: "))
```

```
        if choice == 1:
```

```

new_name = input("Enter new name: ")
users[username]['name'] = new_name

update_query = "UPDATE customers SET cust_name = %s WHERE
cust_username = %s"

cb.execute(update_query, (new_name, username))

print("Name updated successfully for", username)

elif choice == 2:

    new_address = input("Enter new address: ")
    users[username]['address'] = new_address

    update_query = "UPDATE customers SET cust_Address = %s WHERE
cust_username = %s"

    cb.execute(update_query, (new_address, username))

    print("Address updated successfully for", username)

elif choice == 3:

    new_number = input("Enter new phone number: ")
    users[username]['number'] = new_number

    update_query = "UPDATE customers SET cust_number = %s WHERE
cust_username = %s"

    cb.execute(update_query, (new_number, username))

    print("Phone number updated successfully for", username)

elif choice == 4:

    new_password = input("Enter new password: ")
    users[username]['password'] = new_password

    update_query = "UPDATE customers SET cust_password = %s WHERE
cust_username = %s"

    cb.execute(update_query, (new_password, username))

    print("Password updated successfully for", username)

elif choice == 0:

    print("Going back to user profile...")

```

```
user_profile_details()
```

```
else:
```

```
    print("INVALID choice!!!")
```

```
    update_user()
```

```
mydb.commit()
```

```
print("Profile updated successfully for the User", username)
```

```
want_to_delete_more_user()
```

```
except Exception as e:
```

```
    print("Error occurred: ", e)
```

```
    want_to_update_more_user()
```

```
finally:
```

```
    cb.close()
```

```
    print("=" * 40)
```

```
def pet_table_for_admin():
```

```
    """Displays options for managing pets with improved design."""
```

```
    print("\n" + "=" * 40)
```

```
    print(" " * 10 + "PET MANAGEMENT OPTIONS")
```

```
    print("=" * 40)
```

```
    print("Enter 0 to Go back")
```

```
    print("Enter 1 to Add Pet")
```

```
    print("Enter 2 to Update Pet")
```

```
    print("Enter 3 to Delete Pet")
```

```
    print("Enter 4 to View Pet")
```

```
    print("=" * 40)
```

```
    try:
```

```
        choice = int(input("Enter your choice: "))
```

```
        if choice == 1:
```

```
        add_pet()
    elif choice == 2:
        update_pet()
    elif choice == 3:
        delete_pet()
    elif choice == 4:
        view_pet()
    elif choice == 0:
        admin_login()
    else:
        print("INVALID choice!!! Please try again.")
        pet_table_for_admin()
except ValueError:
    print("Please enter a valid number.")
    pet_table_for_admin()
finally:
    print("=" * 40)
```

```
def want_to_add_more():
    print("Want to add more pet? (y/n)")
    choice = input("Enter your choice: ")
    if choice.lower() == "y":
        add_pet()
    else:
        pet_table_for_admin()
```

```
def add_pet():
    """Adds a new pet to the database"""
```

```

print("\n" + "=" * 40)

print(" " * 10 + "ADDING NEW PET")

print("=" * 40)

cb = mydb.cursor()

try:

    pet_type = input("Enter pet type (e.g., Dog, Cat, Bird): ")
    pet_breed = input("Enter pet breed: ")
    pet_age = int(input("Enter pet age (in years) more than 3yrs: "))
    pet_gender = input("Enter pet gender (Male/Female/Unknown): ")
    pet_price = float(input("Enter pet price (in decimal): "))
    pet_description = input("Enter pet description: ")
    pet_quantity = int(input("Enter pet quantity: "))

    if pet_age < 3:

        print("Enter age more than 3yrs")

        add_pet()

    else:

        insert_pet_query = "INSERT INTO pets (pet_type, pet_breed, pet_age,
pet_gender, pet_price, pet_description, pet_quantity) VALUES (%s, %s, %s, %s, %s,
%s, %s)"

        cb.execute(insert_pet_query, (pet_type, pet_breed, pet_age, pet_gender,
pet_price, pet_description, pet_quantity))

        mydb.commit()

        print("=" * 40)

        print(" " * 10 + "NEW PET ADDED SUCCESSFULLY!")

        print("=" * 40)

        want_to_add_more()

except Exception as e:

    print("Error occurred: ", e)

    add_pet()

```

finally:

```
cb.close()
```

```
print("=" * 40)
```

```
def want_to_update_more():
```

```
    print("Want to update more pet? (y/n)")
```

```
    choice = input("Enter your choice: ")
```

```
    if choice.lower() == "y":
```

```
        update_pet()
```

```
    else:
```

```
        pet_table_for_admin()
```

```
def update_pet():
```

```
    """Updates pet information """
```

```
    print("\n" + "=" * 40)
```

```
    print(" " * 10 + "UPDATE PET")
```

```
    print("=" * 40)
```

```
    cb = mydb.cursor()
```

```
    try:
```

```
        pet_ID = int(input("Enter the Pet ID of the pet you want to update: "))
```

```
        check_query = "SELECT * FROM pets WHERE pet_ID = %s"
```

```
        cb.execute(check_query, (pet_ID,))
```

```
        pet = cb.fetchone()
```

```
        if pet:
```

```
            print("\n" + "=" * 40)
```

```
            print(" " * 10 + "CURRENT PET DETAILS")
```

```
            print("=" * 40)
```

```
print("ID: {}, Type: {}, Breed: {}, Age: {}, Gender: {}, Price: ${}, Description: {},  
Quantity Available: {}".format(pet[0], pet[1], pet[2], pet[3], pet[4], pet[5], pet[6],  
pet[7]))
```

```
print("=" * 40)
```

```
print("Enter the details you want to update:")
```

```
print("1. Update Type")
```

```
print("2. Update Breed")
```

```
print("3. Update Age")
```

```
print("4. Update Gender")
```

```
print("5. Update Price")
```

```
print("6. Update Description")
```

```
print("7. Update Quantity")
```

```
choice = int(input("Enter your choice: "))
```

```
if choice == 1:
```

```
    new_type = input("Enter new pet type (e.g., Dog, Cat, Bird): ")
```

```
    update_query = "UPDATE pets SET pet_type = %s WHERE pet_ID = %s"
```

```
    cb.execute(update_query, (new_type, pet_ID))
```

```
    print("Pet type updated successfully.")
```

```
elif choice == 2:
```

```
    new_breed = input("Enter new pet breed: ")
```

```
    update_query = "UPDATE pets SET pet_breed = %s WHERE pet_ID = %s"
```

```
    cb.execute(update_query, (new_breed, pet_ID))
```

```
    print("Pet breed updated successfully.")
```

```
elif choice == 3:
```

```
    new_age = int(input("Enter new pet age (in years): "))
```

```
    update_query = "UPDATE pets SET pet_age = %s WHERE pet_ID = %s"
```

```
    cb.execute(update_query, (new_age, pet_ID))
```

```
    print("Pet age updated successfully.")
```

elif choice == 4:

new_gender = input("Enter new pet gender (Male/Female/Unknown): ")

update_query = "UPDATE pets SET pet_gender = %s WHERE pet_ID = %s"

cb.execute(update_query, (new_gender, pet_ID))

print("Pet gender updated successfully.")

elif choice == 5:

new_price = float(input("Enter new pet price: "))

update_query = "UPDATE pets SET pet_price = %s WHERE pet_ID = %s"

cb.execute(update_query, (new_price, pet_ID))

print("Pet price updated successfully.")

elif choice == 6:

new_description = input("Enter new pet description: ")

update_query = "UPDATE pets SET pet_description = %s WHERE pet_ID = %s"

cb.execute(update_query, (new_description, pet_ID))

print("Pet description updated successfully.")

elif choice == 7:

new_quantity = int(input("Enter new pet quantity: "))

update_query = "UPDATE pets SET pet_quantity = %s WHERE pet_ID = %s"

cb.execute(update_query, (new_quantity, pet_ID))

print("Pet quantity updated successfully.")

else:

print("Invalid choice! Please try again.")

update_pet()

want_to_update_more()

mydb.commit()

else:

```

        print("No pet found with ID {}".format(pet_ID))
        update_pet()
    except Exception as e:
        print("Error occurred: ", e)
        update_pet()
    finally:
        cb.close()
        print("=" * 40)

def want_to_delete_more():
    choice = input("Do you want to delete more pets? (yes/no): ")
    if choice.lower() == "yes":
        delete_pet()
    else:
        pet_table_for_admin()

def delete_pet():
    """Deletes a pet from the database."""
    print("\n" + "=" * 40)
    print(" " * 10 + "DELETE PET")
    print("=" * 40)
    cb = mydb.cursor()
    try:
        pet_ID = int(input("Enter the Pet ID of the pet you want to delete: "))
        check_query = "SELECT * FROM pets WHERE pet_ID = %s"
        cb.execute(check_query, (pet_ID,))
        pet = cb.fetchone()
        if pet:

```

```

        confirm = input("Are you sure you want to delete the pet with ID {}? (yes/no):
".format(pet_ID)).lower()

        if confirm == "yes":

            delete_query = "DELETE FROM pets WHERE pet_ID = %s"

            cb.execute(delete_query, (pet_ID,))

            mydb.commit()

            print("Pet with ID {} has been successfully deleted.".format(pet_ID))

            want_to_delete_more()

        else:

            print("Deletion cancelled.")

            want_to_delete_more()

    else:

        print("No pet found with ID {}".format(pet_ID))

        delete_pet()

except Exception as e:

    print("Error occurred: ", e)

    delete_pet()

finally:

    cb.close()

    print("=" * 40)

def view_pet():

    """Viewing pets."""

    print("\n" + "=" * 40)

    print(" " * 10 + "VIEWING PETS")

    print("=" * 40)

    print("Enter 1 to view all pets")

    print("Enter 2 to search for a pet by ID")

```

try:

```
choice = int(input("Enter your choice: "))
```

```
cb = mydb.cursor()
```

```
if choice == 1:
```

```
    query = "SELECT * FROM pets"
```

```
    cb.execute(query)
```

```
    pets = cb.fetchall()
```

```
    if pets:
```

```
        print("\n" + "=" * 40)
```

```
        print(" " * 10 + "AVAILABLE PETS")
```

```
        print("=" * 40)
```

```
        for pet in pets:
```

```
            print("ID: {}, Type: {}, Breed: {}, Age: {}, Gender: {}, Price: ${},  
Description: {}, Quantity Available: {}".format(pet[0], pet[1], pet[2], pet[3], pet[4],  
pet[5], pet[6], pet[7]))
```

```
    else:
```

```
        print("No pets available.")
```

```
        pet_table_for_admin()
```

```
elif choice == 2:
```

```
    pet_ID = int(input("Enter the Pet ID: "))
```

```
    query = "SELECT * FROM pets WHERE pet_ID = %s"
```

```
    cb.execute(query, (pet_ID,))
```

```
    pet = cb.fetchone()
```

```
    if pet:
```

```
        print("\n" + "=" * 40)
```

```
        print(" " * 10 + "PET DETAILS")
```

```
        print("=" * 40)
```

```
print("ID: {}, Type: {}, Breed: {}, Age: {}, Gender: {}, Price: ${}, Description: {}, Quantity Available: {}".format(pet[0], pet[1], pet[2], pet[3], pet[4], pet[5], pet[6], pet[7]))
```

```
else:
```

```
print("No pet found with ID {}".format(pet_ID))
```

```
pet_table_for_admin()
```

```
else:
```

```
print("Invalid choice! Please try again.")
```

```
pet_table_for_admin()
```

```
except Exception as e:
```

```
print("Error occurred: ", e)
```

```
view_pet()
```

```
finally:
```

```
cb.close()
```

```
print("=" * 40)
```

```
def pet_accessories_for_admin():
```

```
    """Displays options for managing pet accessories"""
```

```
    print("\n" + "=" * 40)
```

```
    print(" " * 10 + "PET ACCESSORIES MANAGEMENT")
```

```
    print("=" * 40)
```

```
    print("Enter 0 to Previous Menu")
```

```
    print("Enter 1 to Add Pet Accessories")
```

```
    print("Enter 2 to Update Pet Accessories")
```

```
    print("Enter 3 to Delete Pet Accessories")
```

```
    print("Enter 4 to View Pet Accessories")
```

```
    try:
```

```
        choice = int(input("Enter your choice: "))
```

```
if choice == 1:
    add_pet_accessories()
elif choice == 2:
    update_pet_accessories()
elif choice == 3:
    delete_pet_accessories()
elif choice == 4:
    view_pet_accessories()
elif choice == 0:
    admin_login()
else:
    print("INVALID choice!!! Please try again.")
    pet_accessories_for_admin()
```

```
except ValueError:
    print("Please enter a valid number.")
    pet_accessories_for_admin()
```

```
def want_to_add_more_accesory():
    choice = input("Do you want to add more accessories? (yes/no): ").lower()
    if choice == "yes":
        add_pet_accessories()
    else:
        pet_accessories_for_admin()
```

```
def add_pet_accessories():
    """Adds a new pet accessory to the database"""
    print("\n" + "=" * 40)
    print(" " * 10 + "ADDING NEW PET ACCESSORY")
```

```

print("=" * 40)

cb = mydb.cursor()

try:

    accessory_name = input("Enter accessory name: ")
    accessory_type = input("Enter accessory type (e.g., Toys, Food, Bedding): ")
    check_query = "SELECT * FROM pet_accessories WHERE accessory_name = %s
AND accessory_type = %s"

    cb.execute(check_query, (accessory_name, accessory_type))
    existing_accessory = cb.fetchone()

    if existing_accessory:
        print("Accessory already exists in the database.")
        want_to_add_more_accessory()

    accessory_price = float(input("Enter accessory price: "))
    accessory_description = input("Enter accessory description: ")
    accessory_quantity = int(input("Enter accessory quantity: "))

    insert_accessory_query = "INSERT INTO pet_accessories (accessory_name,
accessory_type, accessory_price, accessory_description, accessory_quantity)
VALUES (%s, %s, %s, %s, %s)"

    cb.execute(insert_accessory_query, (accessory_name, accessory_type,
accessory_price, accessory_description, accessory_quantity))

    mydb.commit()

    print("New pet accessory added successfully!")
    want_to_add_more_accessory()

except Exception as e:

    print("Error occurred: ", e)
    want_to_add_more_accessory()

finally:

    cb.close()

print("=" * 40)

```

```

def want_to_update_more_accesory():
    choice = input("Do you want to update more accessories? (yes/no): ").lower()
    if choice == "yes":
        update_pet_accessories()
    else:
        pet_accessories_for_admin()

def update_pet_accessories():
    """Updates pet accessory information"""
    print("\n" + "=" * 40)
    print(" " * 10 + "UPDATE PET ACCESSORY")
    print("=" * 40)
    cb = mydb.cursor()
    try:
        accessory_ID = int(input("Enter the Accessory ID of the accessory you want to
update: "))

        check_query = "SELECT * FROM pet_accessories WHERE accessory_ID = %s"
        cb.execute(check_query, (accessory_ID,))
        accessory = cb.fetchone()
        if accessory:
            print("\n" + "=" * 40)
            print(" " * 10 + "CURRENT ACCESSORY DETAILS")
            print("=" * 40)
            print("ID: {}, Name: {}, Type: {}, Price: ${}, Description: {}, Quantity Available:
{}".format(accessory[0], accessory[1], accessory[2], accessory[3], accessory[4],
accessory[5]))
            print("=" * 40)
            print("Enter the details you want to update:")

```

```
print("1. Update Name")
print("2. Update Type")
print("3. Update Price")
print("4. Update Description")
print("5. Update Quantity")
choice = int(input("Enter your choice: "))
if choice == 1:
    new_name = input("Enter new accessory name: ")
    update_query = "UPDATE pet_accessories SET accessory_name = %s
WHERE accessory_ID = %s"
    cb.execute(update_query, (new_name, accessory_ID))
    print("Accessory name updated successfully.")
elif choice == 2:
    new_type = input("Enter new accessory type: ")
    update_query = "UPDATE pet_accessories SET accessory_type = %s
WHERE accessory_ID = %s"
    cb.execute(update_query, (new_type, accessory_ID))
    print("Accessory type updated successfully.")
elif choice == 3:
    new_price = float(input("Enter new accessory price: "))
    update_query = "UPDATE pet_accessories SET accessory_price = %s
WHERE accessory_ID = %s"
    cb.execute(update_query, (new_price, accessory_ID))
    print("Accessory price updated successfully.")
elif choice == 4:
    new_description = input("Enter new accessory description: ")
    update_query = "UPDATE pet_accessories SET accessory_description = %s
WHERE accessory_ID = %s"
    cb.execute(update_query, (new_description, accessory_ID))
```

```

        print("Accessory description updated successfully.")
    elif choice == 5:
        new_quantity = int(input("Enter new accessory quantity: "))
        update_query = "UPDATE pet_accessories SET accessory_quantity = %s
WHERE accessory_ID = %s"
        cb.execute(update_query, (new_quantity, accessory_ID))
        print("Accessory quantity updated successfully.")
    else:
        print("Invalid choice! Please try again.")
        update_pet_accessories()
        want_to_update_more_accessory()
        mydb.commit()
    else:
        print("No accessory found with ID {}".format(accessory_ID))
        want_to_update_more_accessory()
except Exception as e:
    print("Error occurred: ", e)
    want_to_update_more_accessory()
finally:
    cb.close()
    print("=" * 40)

def want_to_delete_more_accessory():
    choice = input("Do you want to delete more accessories? (yes/no): ").lower()
    if choice == "yes":
        delete_pet_accessories()
    else:
        pet_accessories_for_admin()

```

```

def delete_pet_accessories():
    """Deletes a pet accessory from the database """
    print("\n" + "=" * 40)
    print(" " * 10 + "DELETE PET ACCESSORY")
    print("=" * 40)
    cb = mydb.cursor()

    try:
        accessory_ID = int(input("Enter the Accessory ID of the accessory you want to
delete: "))

        check_query = "SELECT * FROM pet_accessories WHERE accessory_ID = %s"
        cb.execute(check_query, (accessory_ID,))
        accessory = cb.fetchone()

        if accessory:
            confirm = input("Are you sure you want to delete the accessory with ID {}?
(yes/no): ".format(accessory_ID)).lower()

            if confirm == "yes":
                delete_query = "DELETE FROM pet_accessories WHERE accessory_ID =
%s"
                cb.execute(delete_query, (accessory_ID,))
                mydb.commit()
                print("Accessory with ID {} has been successfully
deleted.".format(accessory_ID))
                want_to_delete_more_accessory()
            else:
                print("Deletion cancelled.")
        else:
            print("No accessory found with ID {}".format(accessory_ID))
            delete_pet_accessories()

```

except Exception as e:

print("Error occurred: ", e)

delete_pet_accessories()

finally:

cb.close()

print("=" * 40)

def view_pet_accessories():

"""Viewing pet accessories """

print("\n" + "=" * 40)

print(" " * 10 + "VIEWING PET ACCESSORIES")

print("=" * 40)

print("Enter 1 to view all accessories: ")

print("Enter 2 to search for an accessory by ID: ")

cb = mydb.cursor()

try:

choice = int(input("Enter your choice: "))

if choice == 1:

query = "SELECT * FROM pet_accessories"

cb.execute(query)

accessories = cb.fetchall()

if accessories:

print("\n" + "=" * 40)

print(" " * 10 + "AVAILABLE PET ACCESSORIES")

print("=" * 40)

for accessory *in* accessories:

```
print("ID: {}, Name: {}, Type: {}, Price: ${}, Description: {}, Quantity  
Available: {}".format(accessory[0], accessory[1], accessory[2], accessory[3],  
accessory[4], accessory[5]))
```

```
else:
```

```
print("No accessories available.")
```

```
pet_accessories_for_admin()
```

```
print("=" * 40)
```

```
elif choice == 2:
```

```
accessory_ID = int(input("Enter the Accessory ID: "))
```

```
query = "SELECT * FROM pet_accessories WHERE accessory_ID = %s"
```

```
cb.execute(query, (accessory_ID,))
```

```
accessory = cb.fetchone()
```

```
if accessory:
```

```
print("\n" + "=" * 40)
```

```
print(" " * 10 + "ACCESSORY DETAILS")
```

```
print("=" * 40)
```

```
print("ID: {}, Name: {}, Type: {}, Price: ${}, Description: {}, Quantity  
Available: {}".format(accessory[0], accessory[1], accessory[2], accessory[3],  
accessory[4], accessory[5]))
```

```
else:
```

```
print("No accessory found with ID {}".format(accessory_ID))
```

```
pet_accessories_for_admin()
```

```
print("=" * 40)
```

```
else:
```

```
print("Invalid choice! Please try again.")
```

```
pet_accessories_for_admin()
```

```
except Exception as e:
```

```
print("Error occurred: ", e)
```

```
delete_pet_accessories()
```

finally:

```
cb.close()
```

```
print("=" * 40)
```

```
def new_admin_login():
```

```
    """Handles new admin login"""
```

```
    print("\n" + "=" * 40)
```

```
    print(" " * 10 + "NEW ADMIN LOGIN")
```

```
    print("=" * 40)
```

```
    admin_username = input("Enter your username: ")
```

```
    admin_password = input("Enter your password: ")
```

```
    if admin_username in admin:
```

```
        print("Username already exists. Choose a different one.")
```

```
        new_admin_login()
```

```
    else:
```

```
        admin[admin_username] = admin_password
```

```
        print("Admin", admin_username, "registered successfully!")
```

```
        admin_login()
```

```
    print("=" * 40)
```

```
def signup_admin():
```

```
    """Handles admin signup"""
```

```
    print("\n" + "=" * 40)
```

```
    print(" " * 10 + "ADMIN SIGNUP")
```

```
    print("=" * 40)
```

```
    print("Enter 0 to Go back")
```

```
    print("Enter 1 for admin login")
```

```
    print("Enter 2 for new admin registration")
```

try:

`x = int(input("Enter your choice: "))`

if `x == 1:`

`admin_login()`

elif `x == 2:`

`new_admin_login()`

elif `x == 0:`

`main_menu()`

else:

`print("Invalid choice. Please try again.")`

`signup_admin()`

except `ValueError:`

`print("Invalid input. Please enter a number.")`

`signup_admin()`

def `main_menu():`

"""Displays the main menu for user and admin options."""

while `True:`

`print("\n" + "=" * 40)`

`print(" " * 10 + "WELCOME TO PET MANAGEMENT SYSTEM")`

`print("=" * 40)`

`print("Choose an option:")`

`print("1. Sign up as user")`

`print("2. Sign up as admin")`

`print("3. Exit")`

`print("=" * 40)`

try:

`choice = int(input("Enter your choice: "))`

```
if choice == 1:
    signup_user()
elif choice == 2:
    signup_admin()
elif choice == 3:
    print("Thank you for using the Pet Management System. Goodbye!")
    break
else:
    print("Invalid choice. Please try again.")
    main_menu()
except ValueError:
    print("Invalid input. Please enter a number.")
    main_menu()
```

```
main_menu()
```

Database:

CUSTOMER TABLE

```
mysql> use pet_management;  
Database changed  
mysql> select * from customers;
```

cust_ID	cust_name	cust_Address	cust_number	cust_username	cust_password
1	Rameswar Behera	Brahmapur	1234567890	rameswar@123	12345
2	Ayush Kumar	Ambapua	9999900000	ayushkumar@69	12345
3	Aahana Patro	First Gate	1234512345	aahanapatro@00	12345

3 rows in set (0.00 sec)

PET TABLE

```
mysql> select * from pets;
```

pet_ID	pet_type	pet_breed	pet_age	pet_gender	pet_price	pet_description	pet_quantity
1	Dog	Golden Retriever	4	Male	1200.00	Friendly and intelligent family dog.	5
2	Cat	Siamese	3	Female	800.00	Elegant and vocal cat with striking blue eyes.	7
3	Bird	Parrot	5	Unknown	300.00	Colorful and intelligent bird that can mimic sounds.	10
4	Rabbit	Holland Lop	3	Male	150.00	Small and friendly rabbit with floppy ears.	8
5	Fish	Goldfish	3	Unknown	10.00	Popular and easy-to-care-for freshwater fish.	20
6	Dog	German Shepherd	6	Male	1500.00	Loyal and protective working dog.	3
7	Cat	Persian	4	Female	1000.00	Long-haired and calm cat with a sweet temperament.	6
8	Bird	Canary	3	Male	50.00	Small and cheerful bird known for its singing.	15
9	Rabbit	Mini Rex	3	Female	200.00	Soft and velvety rabbit with a gentle nature.	5
10	Fish	Betta	3	Male	15.00	Vibrant and low-maintenance fish with flowing fins.	25

10 rows in set (0.00 sec)

PET ACCESSORIES TABLE

```
mysql> select * from pet_accessories;
```

accessory_ID	accessory_name	accessory_type	accessory_price	accessory_description	accessory_quantity
1	Dog Collar	Collars	15	Durable and adjustable collar for dogs.	50
2	Cat Toy	Toys	10	Interactive toy to keep your cat entertained.	100
3	Bird Cage	Cages	80	Spacious and sturdy cage for birds.	20
4	Rabbit Hutch	Housing	120	Comfortable and secure hutch for rabbits.	15
5	Fish Tank	Aquariums	200	Glass aquarium with filtration system.	10
6	Dog Leash	Leashes	20	Strong and reliable leash for dogs.	40
7	Cat Scratching Post	Furniture	50	Tall scratching post to protect your furniture.	30
8	Bird Feeder	Feeders	25	Hanging feeder for bird food and seeds.	60
9	Rabbit Food Bowl	Bowls	12	Sturdy and easy-to-clean food bowl for rabbits.	25
10	Fish Food	Food	8	Nutritious food for freshwater fish.	200

```
10 rows in set (0.00 sec)
```

USER ORDER TABLE

```
mysql> select * from user_orders;
```

order_ID	cust_ID	pet_ID	accessory_ID	total_pet_quantity	total_accessory_quantity	total_order_price	order_date
1	1	1	1	1	1	1215.00	2022-01-01
2	2	2	2	2	2	820.00	2022-01-05
3	3	3	3	1	1	380.00	2022-01-10
4	1	4	4	1	1	270.00	2022-01-15
5	2	5	5	1	1	210.00	2022-01-20
6	3	6	6	1	1	1500.00	2022-01-25
7	1	7	7	1	1	1000.00	2022-01-30
8	2	8	8	1	1	50.00	2022-02-01
9	3	9	9	1	1	200.00	2022-02-05
10	1	10	10	1	1	8.00	2022-02-10

```
10 rows in set (0.00 sec)
```

Outputs:

LOGIN INTERFACE

=====

WELCOME TO PET MANAGEMENT SYSTEM

=====

Choose an option:

1. Sign up as user
2. Sign up as admin
3. Exit

=====

Enter your choice:

INTERFACE FOR CUSTOMER

=====

USER SIGNUP

=====

Enter 0 to Go back

Enter 1 for user login

Enter 2 for new user registration

Enter your choice: 1

=====

USER LOGIN

=====

Enter your username: rameswar@123

Enter your password: 12345

Login successful

Enter 0 to logout

Enter 1 to see user profile

Enter 2 to see PET'S

Enter 3 to see PET'S Accessories

Enter 4 to Place order

Enter your choice: 1

=====

USER PROFILE

=====

Enter 0 to Previous Menu

Enter 1 for updating profile

Enter 2 for viewing profile

Enter 3 for My Order Details

Enter Your Choice: 1

=====

MY PROFILE UPDATE

=====

Enter 0 to Previous Menu

Enter 1 to update Name

Enter 2 to update Address

Enter 3 to update Phone Number

Enter 4 to update Password

Enter your choice:

Enter Your Choice: 2

=====

MY PROFILE VIEW

=====

=====

USER PROFILE DETAILS

=====

Username: rameswar@123

Name: Rameswar Behera

Address: Brahmapur

Phone Number: 1234567890

=====

ORDER DETAILS

=====

Order ID: 1

Total Price: \$1215.00

Order Date: 2022-01-01

Order ID: 4

Total Price: \$270.00

Order Date: 2022-01-15

Order ID: 7

Total Price: \$1000.00

Order Date: 2022-01-30

Order ID: 10

Total Price: \$8.00

Order Date: 2022-02-10

=====

Enter Your Choice: 3

=====

MY ORDERS DETAILS

=====

=====

ORDER DETAILS

=====

Order ID: 1

Pet Type: Dog

Pet Price: \$1200.00

Accessory Type: Collars

Accessory Price: \$15

Pet Quantity: 1

Accessory Quantity: 1

Total Price: \$1215.00

Order Date: 2022-01-01

=====

Enter your choice: 2

=====

PET MANAGEMENT OPTIONS

=====

Enter 0 to Previous Menu

Enter 1 for Search pets

Enter 2 for To show all pets

Enter your choice: 1

=====

SEARCH FOR PETS

=====

0. Previous Menu

1. Pet Type

2. Pet Breed

3. Pet Age

4. Pet Gender

5. Pet Price

Enter your choice:

Enter your choice: 2

=====

AVAILABLE PETS

=====

ID: 1, Type: Dog, Breed: Golden Retriever, Age: 4, Gender: Male, Price: \$1200.00, Description: Friendly and intelligent family dog., Quantity Available: 5

ID: 2, Type: Cat, Breed: Siamese, Age: 3, Gender: Female, Price: \$800.00, Description: Elegant and vocal cat with striking blue eyes., Quantity Available: 7

ID: 3, Type: Bird, Breed: Parrot, Age: 5, Gender: Unknown, Price: \$300.00, Description: Colorful and intelligent bird that can mimic sounds., Quantity Available: 10

ID: 4, Type: Rabbit, Breed: Holland Lop, Age: 3, Gender: Male, Price: \$150.00, Description: Small and friendly rabbit with floppy ears., Quantity Available: 8

ID: 5, Type: Fish, Breed: Goldfish, Age: 3, Gender: Unknown, Price: \$10.00, Description: Popular and easy-to-care-for freshwater fish., Quantity Available: 20

ID: 6, Type: Dog, Breed: German Shepherd, Age: 6, Gender: Male, Price: \$1500.00, Description: Loyal and protective working dog., Quantity Available: 3

ID: 7, Type: Cat, Breed: Persian, Age: 4, Gender: Female, Price: \$1000.00, Description: Long-haired and calm cat with a sweet temperament., Quantity Available: 6

ID: 8, Type: Bird, Breed: Canary, Age: 3, Gender: Male, Price: \$50.00, Description: Small and cheerful bird known for its singing., Quantity Available: 15

ID: 9, Type: Rabbit, Breed: Mini Rex, Age: 3, Gender: Female, Price: \$200.00, Description: Soft and velvety rabbit with a gentle nature., Quantity Available: 5

ID: 10, Type: Fish, Breed: Betta, Age: 3, Gender: Male, Price: \$15.00, Description: Vibrant and low-maintenance fish with flowing fins., Quantity Available: 25

=====

Enter your choice: 3

=====

SHOWING PET ACCESSORIES

=====

Enter 0 to Previous menu

Enter 1 for Search accessories

Enter 2 for To show all accessories

Enter your choice: 1

=====

SEARCH FOR PET ACCESSORIES

=====

0. Previous Menu

1. Accessory Type

2. Accessory Name

3. Accessory Price

Enter your choice:

Enter your choice: 2

=====

AVAILABLE PET ACCESSORIES

=====

ID: 1, Name: Dog Collar, Type: Collars, Price: \$15, Description: Durable and adjustable collar for dogs., Quantity Available: 50

ID: 2, Name: Cat Toy, Type: Toys, Price: \$10, Description: Interactive toy to keep your cat entertained., Quantity Available: 100

ID: 3, Name: Bird Cage, Type: Cages, Price: \$80, Description: Spacious and sturdy cage for birds., Quantity Available: 20

ID: 4, Name: Rabbit Hutch, Type: Housing, Price: \$120, Description: Comfortable and secure hutch for rabbits., Quantity Available: 15

ID: 5, Name: Fish Tank, Type: Aquariums, Price: \$200, Description: Glass aquarium with filtration system., Quantity Available: 10

ID: 6, Name: Dog Leash, Type: Leashes, Price: \$20, Description: Strong and reliable leash for dogs., Quantity Available: 40

ID: 7, Name: Cat Scratching Post, Type: Furniture, Price: \$50, Description: Tall scratching post to protect your furniture., Quantity Available: 30

ID: 8, Name: Bird Feeder, Type: Feeders, Price: \$25, Description: Hanging feeder for bird food and seeds., Quantity Available: 60

ID: 9, Name: Rabbit Food Bowl, Type: Bowls, Price: \$12, Description: Sturdy and easy-to-clean food bowl for rabbits., Quantity Available: 25

ID: 10, Name: Fish Food, Type: Food, Price: \$8, Description: Nutritious food for freshwater fish., Quantity Available: 200

=====

Enter your choice: 4

=====

PLACE AN ORDER

=====

Available Pets:

ID: 1, Type: Dog, Price: \$1200.00, Quantity Available: 5

ID: 2, Type: Cat, Price: \$800.00, Quantity Available: 7

ID: 3, Type: Bird, Price: \$300.00, Quantity Available: 10

ID: 4, Type: Rabbit, Price: \$150.00, Quantity Available: 8

ID: 5, Type: Fish, Price: \$10.00, Quantity Available: 20

ID: 6, Type: Dog, Price: \$1500.00, Quantity Available: 3

ID: 7, Type: Cat, Price: \$1000.00, Quantity Available: 6

ID: 8, Type: Bird, Price: \$50.00, Quantity Available: 15

ID: 9, Type: Rabbit, Price: \$200.00, Quantity Available: 5

ID: 10, Type: Fish, Price: \$15.00, Quantity Available: 25

Enter the Pet ID you want to order: 1

Enter the quantity of the pet you want to order: 1

Available Accessories:

ID: 1, Name: Dog Collar, Price: \$15, Quantity Available: 50

ID: 2, Name: Cat Toy, Price: \$10, Quantity Available: 100

ID: 3, Name: Bird Cage, Price: \$80, Quantity Available: 20

ID: 4, Name: Rabbit Hutch, Price: \$120, Quantity Available: 15

ID: 5, Name: Fish Tank, Price: \$200, Quantity Available: 10

ID: 6, Name: Dog Leash, Price: \$20, Quantity Available: 40

ID: 7, Name: Cat Scratching Post, Price: \$50, Quantity Available: 30

ID: 8, Name: Bird Feeder, Price: \$25, Quantity Available: 60

ID: 9, Name: Rabbit Food Bowl, Price: \$12, Quantity Available: 25

ID: 10, Name: Fish Food, Price: \$8, Quantity Available: 200

Enter the Accessory ID you want to order (or enter 0 if none): 2

Enter the quantity of the accessory you want to order: 1

Order placed successfully!

Total Order Price: \$1210.00

Enter your choice: 2

=====

NEW USER LOGIN

=====

Enter your username: ankit@11

Enter your password: 12345

Enter your name: Ankit Panda

Enter your address: Brahmapur

Enter your number: 5555500000

User ankit@11 registered successfully!

INTERFACE FOR ADMIN

ADMIN SIGNUP

Enter 0 to Go back

Enter 1 for admin login

Enter 2 for new admin registration

Enter your choice: 1

ADMIN LOGIN

Enter your admin username: ADMIN

Enter your admin password: 12345

Login successful

Enter 0 to logout

Enter 1 to view My profile Page

Enter 2 to view User Page

Enter 3 to view Pet Page

Enter your choice: 1

ADMIN PROFILE

Enter 0 to Previous Page

Enter 1 to view details

Enter 2 to Edit profile

Enter 3 to View Shop Status

Enter your choice: 1

ADMIN PROFILE VIEW

Enter your admin username: ADMIN

=====

ADMIN PROFILE DETAILS

=====

Username: ADMIN

Password: 12345

=====

ADMIN PROFILE

=====

Enter 0 to Previous Page

Enter 1 to view details

Enter 2 to Edit profile

Enter 3 to View Shop Status

Enter your choice: 2

=====

ADMIN PROFILE UPDATE

=====

Enter 0 to Go back

Enter 1 to update Username

Enter 2 to update Password

Enter your old username: ADMIN

Enter your choice: 0

Going back to admin profile...

=====

ADMIN PROFILE

=====

Enter 0 to Previous Page

Enter 1 to view details

Enter 2 to Edit profile

Enter 3 to View Shop Status

Enter your choice: 3

=====

SHOP STATUS

=====

=====

SHOP STATISTICS

=====

Total Customers: 4

Total Pets Available: 10

Total Pet Accessories Available: 10

Total Revenue from Pets: \$5225.00

Total Revenue from Pet Accessories: \$540

Total Orders Placed: 11

Enter your choice: 2

=====

USER PROFILE DETAILS

=====

Enter 0 to Previous Menu

Enter 1 to add user

Enter 2 to delete user

Enter 3 to view user details

Enter 4 to update user details

Enter your choice:

Enter your choice: 3

=====

PET DETAILS

=====

Enter 0 to Go back

Enter 1 for PET

Enter 2 for Pet Accessories

Enter your choice: 1

=====

PET MANAGEMENT OPTIONS

=====

Enter 0 to Go back

Enter 1 to Add Pet

Enter 2 to Update Pet

Enter 3 to Delete Pet

Enter 4 to View Pet

=====

Enter your choice:

Enter your choice: 2

=====

PET ACCESSORIES MANAGEMENT

=====

Enter 0 to Previous Menu

Enter 1 to Add Pet Accessories

Enter 2 to Update Pet Accessories

Enter 3 to Delete Pet Accessories

Enter 4 to View Pet Accessories

Enter your choice:

WE CAN ADD, DELETE, UPDATE, VIEW DETAILS OF USER, PETS, PET
ACCESSORIES.....THANK YOU

References

In order to work on this project titled -PET SHOP MANAGEMENT the following books and websites are referred by me during the various phases of development of the project.

- 1) COMPUTER SCIENCE (NCERT Class XI and XII)
- 2) Together with COMPUTER SCIENCE
- 3) www.youtube.com
- 4) www.python.com

Other than the above-mentioned books, the suggestions and supervision of my teacher and my class experience also helped me to develop this software project.