COMPUTER SCIENCE PROJECT

MULTI END GAME STORE

Name of candidate – Raman Gupta

Roll No. of Candidate-8747720

Class: XII B

15 YEARS OF EDUCATION EXCELLENCE

Global Indian International School

SINGAPORE • MALAYSIA • THAILAND • JAPAN • UAE • INDIA • VIETNAM

Certificate

Department of Information & Technology

Global Indian International School,

SMART Campus, Singapore

This is to certify that the project entitled *Multi End Game Store* is a genuine work of Raman Gupta Roll No. 8747720 undertaken as a part of fulfilment of Computer Science [083] practical syllabus for A.I.S.S.C.E. 2020-2021 to be conducted by C.B.S.E. and has been completed within stipulated time period under my guidance and supervision.

Signature

Ms. Radha Ganesh

Academic Coordinator
Computer Science Teacher

Date:

ACKNOWLEDGEMENT

I take this opportunity to express my acknowledgement and sincere gratitude to Mrs. Radha Ganesh for her valuable suggestions and able guidance required for this project. It is through her I have learnt efficient debugging skills which were helpful in completing the project on time.

Candidate Name – Raman Gupta

CONTENTS

PROJECT DOCUMENTATION

1. INTRODUCTION

The gaming industry has seen a huge upheaval in popularity. They are becoming an integral part of a common man's life. Game stores can be found almost anywhere, putting out games and consoles for people to experience and buy. With the emergence of online game stores, my team and I decided to come up with our own version of a game store, providing an interface not just for the game buyer, but also for the store manager or a game developer. Our aim was to create a code to simulate environments for all such needs and purposes.

Our code uses a variety of user defined functions to fulfil all the requirements for the game store simulator.

We have also implemented new concepts that we learnt in grade 12 like using MySQL to store data and accessing them with the use of the MySQL Connector module. Most of our data is stored in multiple tables and is presented to the user at the time of request in a clean, readable format.

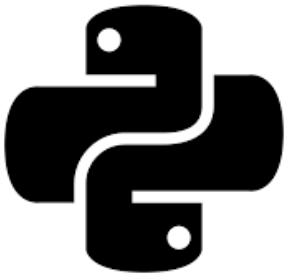
2. ABOUT THE PROGRAMMING TOOL USED

A - Python

Python is one of, if not the most popular programming language being used today. It is an interpreted, object-oriented, high-level programming language with dynamic semantics. It allows the extensive use of libraries and modules (most are open source) allowing the code to be shorter and more compact. Python data items and structures include string, integer, float, boolean, lists, tuples and so many more. Structures like lists also act as data items. These structures are dynamic and allow the user to store multiple items without the issue of unused space, as generally occurs in queues.

Python, being an interpreter based language, is very popular due the ease of debugging code written using it, and its easily readable syntax when compared to compiler based languages like C++. Error finding is easy and fast.Scope of variables is defined with indentation rather than curly brackets. Python supports multiple platforms like Mac,Windows,Linuz etc. It is very popular in multiple fields like Data Analytics and Machine Learning. Competitive programming with Python is also growing steadily, a field which is generally dominated by languages like C++ and Java.

Due to the extensive use of Python in various fields, it is of the utmost importance today that budding coders learn to utilize Python's full capabilities.

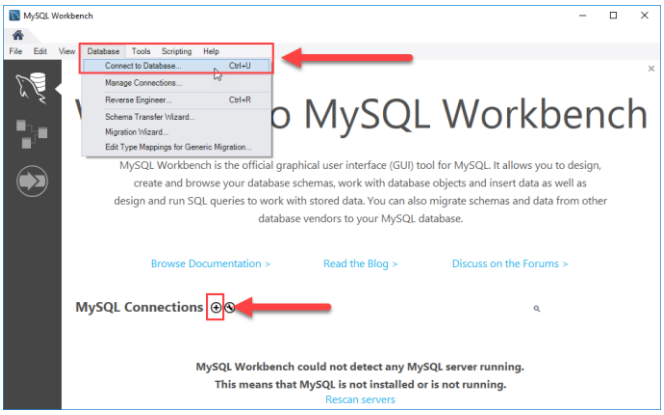


B - MySQL

MySQL is an open source client-server Relational Database Management System created by David Axmark and Michael Widenius. Written in C and C++, it has generally garnered positive reviews from users for its highly efficient performance. MySQL is supported on multiple platforms, including the highly popular ones like Microsoft Windows, MacOs, Linux and many more. It is also very easy to learn and use.

Data is stored in the form of tables. Each column is called an attribute and each row a tuple. MySQL allows users to set multiple constraints on each attribute. Data stored on MySQL is very secure. With the help of modules, it is also possible to import data from MySQL using programming languages like Python.

At the moment, MySQL is being developed by Oracle, which is very well known for its database software.

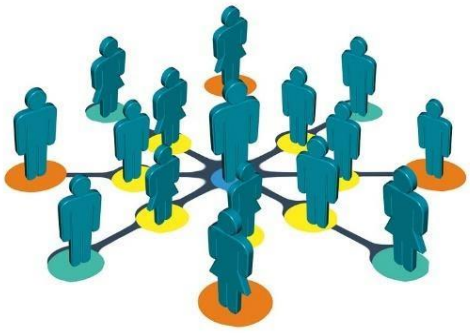


3. PROBLEM DEFINITION

Most game store programs are tailor made for their respective markets. Managers, Developers and consumers receive versions that each suit their role in the industry. They have different functions, different UIs and different environments. But this takes an enormous amount of money, time and effort. How do we produce a finished product that suits all needs and purposes and yet is still marketed as one version without complexities or issues?

This was what we aimed to accomplish when we decided to work on a game store simulator for our project. We believe the G-HUB software satisfies and presents a solution to all the problems pointed above. G-HUB encapsulates all participants in this gaming community - from the buyers to the sellers to the managers.

The user interface is similar for all participants but the extensive use of functions in our code separates consumers, developers, and managers from each other. Some functions are available to the manager, while some are universal. All data is stored in our private MySQL database rather than within the program itself. Thus storage is cost effective, data is well protected by encryptions and no data redundancy or inconsistency exists on our server.



4. DESIGN REQUIREMENTS

A Game Store Software is essentially a database management system which caters to the needs of the buying and selling games.

Main function of the code is to create and insert records(Developer), show selected records to the consumer, and to modify data in the table like games, stock, discounts, etc.

Usernames and Passwords are stored in text files to ensure that data is stored safely.

Redundancy is to be reduced by the use of tables and text files.

Each record in the tables are identified by a unique attribute i.e the Primary key. All primary keys are to be set up before and not during program execution.

4.1 Data Structures

We have made heavy use of a handful of Data Structures like Tuples, Lists and Dictionaries. The use of Stacks and Queues was avoided by the use of MySQL tables.

4.2 Files

A Text File *UserName.txt* is used to store usernames and passwords of the consumers
Developer.txt is used to store developer and item details.
In MySQL, 3 Tables - *Genre, Games, and Consoles* are used to store all data relating to games and consoles. *Genre* stores all game genres currently available to purchase.

Genre:

Field	Type	Null	Key	Default	Extra
Game_Genre	varchar(30)	YES		NULL	
Games	int(11)	YES		NULL	

Games:

Field	Type	Null	Key	Default	Extra
Developer	varchar(20)	YES		NULL	
ID	int(11)	NO	PRI	NULL	
Game_Name	varchar(30)	YES		NULL	
Store_Price	int(11)	YES		NULL	
Developer_Price	int(11)	YES		NULL	
Game_Genre	varchar(10)	YES		NULL	
Release_Year	int(11)	YES		NULL	
Stock_Overall	int(11)	YES		NULL	
Stock_PS	int(11)	YES		NULL	
Stock_XBOX	int(11)	YES		NULL	
Stock_Nintendo	int(11)	YES		NULL	
Sales_Overall	int(11)	YES		NULL	
Sales_PS	int(11)	YES		NULL	
Sales_XBOX	int(11)	YES		NULL	
Sales_Nintendo	int(11)	YES		NULL	
Revenue_Overall	int(11)	YES		NULL	
Revenue_PS	int(11)	YES		NULL	
Revenue_XBOX	int(11)	YES		NULL	
Revenue_Ninte…	int(11)	YES		NULL	
List_Status	tinyint(1)	YES		NULL	

Consoles:

Field	Type	Null	Key	Default	Extra
Developer	varchar(20)	YES		NULL	
ID	int(11)	NO	PRI	NULL	
Console_Name	varchar(30)	YES		NULL	
Store_Price	int(11)	YES		NULL	
Developer_Price	int(11)	YES		NULL	
Console_Type	varchar(10)	YES		NULL	
Release_Year	int(11)	YES		NULL	
Stock	int(11)	YES		NULL	
Sales	int(11)	YES		NULL	
Revenue	int(11)	YES		NULL	
List_Status	tinyint(1)	YES		NULL	

4.3 Environments and Modules

IDLE 3.8 or any similar Python IDE
MySQL 8.0 or any similar DBMS
Modules - mysql.connector, prettytable

4.4 Types of Users and Individual Requirements

User:
→ Sign In
→ Sign Up
→ View Games and Consoles
→ View and Edit Cart
→ Buy Items

Developer:
→ Sign in
→ Listing Games

Manager:
→ Add Games
→ Add Discounts
→ View Game and Console Details and Profits Made per Item. Approve Discounts, Special Messages, Game and Console Listings.

4.5 Functions Used

→ MemberSignIn() - Allows users to try and sign in 3 times. After 3rd time returns to main menu
→ MemberSignUp() - Creates Account for new users
→ AddToCart() - Add Items to Cart
→ ViewCart() - View and Edit items in Cart
→ BuyProduct() - Make Changes to Stock and revenue fields in SQL table
→ DeveloperSignIn() - Allow game developers to enter and list games.
→ ManagerSignIn() - Allows Manager to add special messages, enter discounts, approve game listings, and view game, console, and overall sales, profits and revenue.

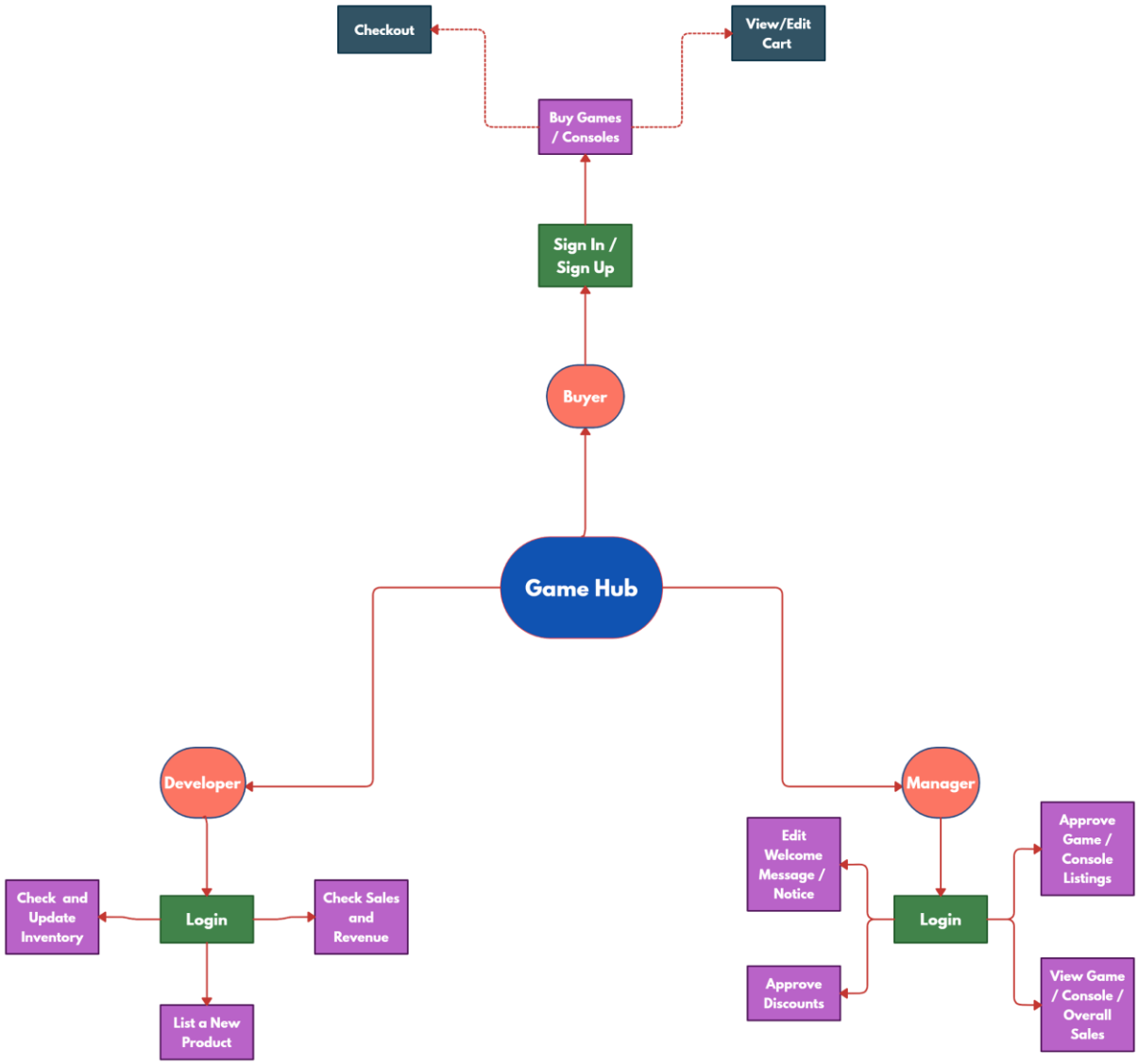
5. SYSTEM REQUIREMENTS

The G-HUB Gamestore will require the following:

1) Computer Minimum Requirements: Microsoft Windows 98 300 Mhz Processor 64MB RAM	2) Computer Operator Literacy: Basic Operations in Microsoft Windows Environment and MacOS Environment

6. FLOWCHART AND CODE

A - FLOWCHART



B - CODE

```
#Importing Modules
import csv
import mysql.connector as sqltor
import sys
from prettytable import PrettyTable

#Store Initialisation
StorePIN = 1104
StoreDiscount = 0
SpecialMessage = "Happy Chinese New Year!!!"
SpecialMessage1="Discount of {} % on all products".format(StoreDiscount)
WelcomeMessage = "Welcome to G-Hub!"
WelcomeMessageUser = "One stop for all your gaming needs!"
ExitMessage="Thank You for Visiting G-Hub! Come Back Again! :))"
Genre=["Open-World", "Adventure", "Sports", "FPS", "Platformer"]

mycon=sqltor.connect(host="localhost", user="root", passwd="12345678",
database="ghub2")
gamecursor=mycon.cursor()
buycursor=mycon.cursor()
browsecursor=mycon.cursor()
consolecursor=mycon.cursor()
developercursor=mycon.cursor()
developercursor2=mycon.cursor()
managercursor=mycon.cursor()
approvecursor=mycon.cursor()
showcursor1=mycon.cursor()
showcursor2=mycon.cursor()
listcursor=mycon.cursor()


print(r''' _______     ___      .___  ___.  _______       __   __   __   __  ._______    __
__
 /  _____|   /   \    |   \/   ||   ____|     | |  | ||  | |  ||   _  \  | ||  |
| |  __     /  ^  \   |  \  /  ||  |__    _______| |__| ||  | |  ||  |_)  || ||  |
| | |_ |   /  /_\  \  |  |\/|  ||   __|  |_______|   __   ||  | |  ||   _ <  | ||  |
| |__| |  /  _____  \ |  |  |  ||  |____       |  |  |  ||  `--'  ||  |_)  ||__||__|
 \______| /__/     \__\|__|  |__||_______|      |__|  |__| \_______/ |_______/  (__) (__) '')
print()
print()
```

```
def MemberSignIn():
    UserName=input("Enter your UserName : ")
    UserPassword=input("Enter your Password : ")
    UD=open("UserData.txt","r+")
    for i in UD:
        if i.split()[0] == UserName:
            if i.split()[1] == UserPassword:
                print("Welcome {}!!".format(UserName))
                UD.close()
                return ("success")
            else:
                for y in (0,3):                                    #3 Tries for Password
                    print("Incorrect Password!! Try Again! Enter 0 to go back")
                    UserPassword=input("Enter your Password Again: ")
                    if UserPassword=="0":
                        UD.close()
                        return ("fail")                            #Take Back to Main Menu
                    elif i.split()[1] == UserPassword:
                        print("Welcome {}!!".format(UserName))
                        UD.close()
                        return ("success")
                    else:
                        print("Too many wrong attempts! Impostor Detected :<")  #Take Back
to Main Menu
                        UD.close()
                        sys.exit()
                break
    else:
        print("UserName does not exist... You will be taken back to Main Menu")
        UD.close()
        return ("fail")



def MemberSignUp():
    UserName=input("Enter a UserName: ")
    UD=open("UserData.txt","r+")
    c=1
    for i in UD:
        c+=1
        if i.split()[0] == UserName:
            print("UserName exists already! Please LogIn or choose another
UserName!!")
            return None
    else:
        UserPassword=input("UserName available, Enter a Password: ")
```

```
        UD.write("\n")
        UD.write("{} {}".format(UserName,UserPassword))
        UD.close()
        print("Member Account created!! Please LogIn again!")
        return None



def DeveloperSignIn():
    DeveloperName=input("Enter your DeveloperName : ")
    DeveloperPassword=input("Enter your Password : ")
    DD=open("DeveloperData.txt","r+")
    for i in DD:
        if i.split("-")[1] == DeveloperName:

            if (i.split("-")[2]) == (DeveloperPassword+"\n"):
                print("Welcome {}!!".format(DeveloperName))
                return (DeveloperName)
            else:
                for y in (0,3):                                    #3 Tries for Password
                    print("Incorrect Password!! Try Again! Enter 0 to go back")
                    DeveloperPassword=input("Enter your Password Again: ")
                    if int(DeveloperPassword)==0:
                        DD.close()
                        return ("fail") #Take Back to Main Menu
                        break

                    elif i.split("-")[2] == DeveloperPassword+"\n":
                        print("Welcome {}!!".format(DeveloperName))
                        DD.close()
                        return (DeveloperName)
                        break

                    else:
                        print("Too many wrong attempts! Impostor Detected :<")  #Take Back
to Main Menu
                        DD.close()
                        sys.exit()
                break
    else:
        print("DeveloperName does not exist... You will be taken back to Main Menu")
        DD.close()
        return ("fail")


def AddToCart(y):
```

```
    z=int(input("""Enter ID to Select Product
Enter 0 to Go Back
--->""" ))
    if z==0:
        return None

    elif z not in y.keys():
        print("Please Select Valid Choice!")
        AddToCart(y)
    else:
        if y[z][0]=="Console":
            z1=int(input("""You Have Selected {}
1. Buy Product
2. Add to Cart
0. Exit
--->""".format(y[z][2])))
            if z1==0:
                return None

            elif z1==1:
                BuyProduct(y[z])

            elif z1==2:
                cart.append(y[z])
                z2=int(input(("""{} successfully added! Cart has {} items.
0. Return and Shop
1. View Cart
--->""".format(y[z][2],len(cart)))))
                if z2==0:
                    return None
                elif z2==1:
                    GoToCart()
        elif y[z][0]=="Game":
            a1=int(input("""Select Console
1. PS
2. XBOX
3. NINTENDO
--->""" ))
            if a1==1:
                browsecursor.execute("select Stock_PS from Games where
ID={}".format(y[z][1]))
                s=browsecursor.fetchone()
                if s[0]==0:
                    print("NO STOCK AVAILABLE FOR PS\n")
                    return None
                else:
```

```
                y[z][0]="Game-PS"
                z1=int(input("""You Have Selected {}
1. Buy Product
2. Add to Cart
0. Exit
--->""".format(y[z][2])))
                if z1==0:
                    return None

                elif z1==1:
                    BuyProduct(y[z])

                elif z1==2:
                    cart.append(y[z])
                    z2=int(input(("""{} successfully added! Cart has {} items.
0. Return and Shop
1. View Cart
--->""".format(y[z][2],len(cart)))))
                    if z2==0:
                        return None
                    elif z2==1:
                        GoToCart()
            elif a1==2:
                browsecursor.execute("select Stock_XBOX from Games where
ID={}".format(y[z][1]))
                s=browsecursor.fetchone()
                if s[0]==0:
                    print("NO STOCK AVAILABLE FOR XBOX\n")
                    return None
                else:
                    y[z][0]="Game-XBOX"
                    z1=int(input("""You Have Selected {}
1. Buy Product
2. Add to Cart
0. Exit
--->""".format(y[z][2])))
                if z1==0:
                    return None

                elif z1==1:
                    BuyProduct(y[z])

                elif z1==2:
                    cart.append(y[z])
                    z2=int(input(("""{} successfully added! Cart has {} items.
0. Return and Shop
```

```
1. View Cart
--->""".format(y[z][2],len(cart)))))
                    if z2==0:
                        return None
                    elif z2==1:
                        GoToCart()
            elif a1==3:
                browsecursor.execute("select Stock_Nintendo from Games where
ID={}".format(y[z][1]))
                s=browsecursor.fetchone()
                if s[0]==0:
                    print("NO STOCK AVAILABLE FOR Nintendo\n")
                    return None
                else:
                    y[z][0]="Game-Nintendo"
                    z1=int(input("""You Have Selected {}
1. Buy Product
2. Add to Cart
0. Exit
--->""".format(y[z][2])))

                    if z1==0:
                        return None

                    elif z1==1:
                        BuyProduct(y[z])

                    elif z1==2:
                        cart.append(y[z])
                        z2=int(input(("""{} successfully added! Cart has {} items.
0. Return and Shop
1. View Cart
--->""".format(y[z][2],len(cart)))))
                    if z2==0:
                        return None
                    elif z2==1:
                        GoToCart()



def GoToCart():
    print("Your Cart --> ")
    cx=PrettyTable()
    cx.field_names=["No.", "Type", "ID", "Item", "Price"]
    z=1
    for i in cart:
```

```
        cx.add_row([z]+i)
        z+=1
    print(cx)


    while True:
        c1=int(input("""Actions:
1. Checkout Now
2. Delete One Item
3. Clear Cart
0. Go to User Main Menu
--->""""))

        if c1==0:
            print()
            return None
        elif c1==1:
            total=0
            for i in cart:
                total+=i[3]
            print("Total Amount is ${} for {} items".format(total*((100-
StoreDiscount)/100), len(cart)))
            for i in range(0,len(cart)):
                BuyProduct(cart[i])
            cart.clear()
            print("Thank you for visiting G-Hub!! :)")
            print("You will now be taken to your previous Menu")
            print()
            return None
        elif c1==2:
            dc=int(input("Enter Product No. to Delete -->"))
            if dc>len(cart):
                print("Please enter valid choice! :<")
                continue
            else:
                del cart[dc-1]
                print("Item Deleted Successfully!")
                print()
                continue
        elif c1==3:
            print("Are you sure you want to clear your cart?")
            c22=int(input("'1.Yes
2.No
--->'"))
            if c22==2:
                continue
```

```
        elif c22==1:
            cart.clear()
            print()
```

```
def BuyProduct(a):
    if a[0]=="Game-PS":
        price=a[3]*((100-StoreDiscount)/100)
        print("Stock is available!! Price is $",price)
        buycursor.execute("update Games set Stock_PS=Stock_PS-1,
Stock_Overall=Stock_Overall-1, Sales_PS=Sales_PS+1,
Sales_Overall=Sales_Overall+1, Revenue_PS=Revenue_PS+{},
Revenue_Overall=Revenue_Overall+{} where ID={}".format(price,price,a[1]))
        print("{} PURCHASE SUCCESSFUL!!".format(a[2]))
        mycon.commit()
        print()
        return None
    elif a[0]=="Game-XBOX":
        price=a[3]*((100-StoreDiscount)/100)
        print("Stock is available!! Price is $",price)
        buycursor.execute("update Games set Stock_XBOX=Stock_XBOX-1,
Stock_Overall=Stock_Overall-1, Sales_XBOX=Sales_XBOX+1,
Sales_Overall=Sales_Overall+1, Revenue_XBOX=Revenue_XBOX+{},
Revenue_Overall=Revenue_Overall+{} where ID={}".format(price,price,a[1]))
        print("{} PURCHASE SUCCESSFUL!!".format(a[2]))
        mycon.commit()
        print()
        return None
    elif a[0]=="Game-Nintendo":
        price=a[3]*((100-StoreDiscount)/100)
        print("Stock is available!! Price is $",price)
        buycursor.execute("update Games set Stock_Nintendo=Stock_Nintendo-1,
Stock_Overall=Stock_Overall-1, Sales_Nintendo=Sales_Nintendo+1,
Sales_Overall=Sales_Overall+1, Revenue_Nintendo=Revenue_Nintendo+{},
Revenue_Overall=Revenue_Overall+{} where ID={}".format(price,price,a[1]))
        print("{} PURCHASE SUCCESSFUL!!".format(a[2]))
        mycon.commit()
        print()
```

```
            return None
    elif a[0]=="Console":
        browsecursor.execute("select Stock from Consoles where ID = {}".format(a[1]))
        atemp=browsecursor.fetchall()
        if atemp==0:
            print("Sorry, No stock available!!")
            print()
        else:
            price=a[3]*((100-StoreDiscount)/100)
            print("Stock is available!! Price is $",price)
            buycursor.execute("update Consoles set Stock=Stock-1, Sales=Sales+1,
Revenue=Revenue+{} where ID={}".format(price,a[1]))
            print("{} PURCHASE SUCCESSFUL!!".format(a[2]))
            mycon.commit()
            print()
            return None
```

```
cart = []
GameProduct={}
ConsoleProduct={}

while True:                                          #Main Menu
    print("Welcome to G-Hub.....",SpecialMessage1,".", SpecialMessage)
    i=int(input("""1. User Login/Sign Up
2. Developer LogIn
3. Store Manager LogIn
0. Exit Store
----> """))


    if i==0:
        print("Thank You for Visiting G-Hub!")
        break
```

```
    elif i==1:
        i1=int(input("""Welcome!!
1. Sign In
2. Sign Up
0. Exit
-----> """))
        if i1 == 0:
            print("Thank You!!")
            continue
        elif i1 == 1:
            i1a=MemberSignIn()
            if i1a == "fail" :
                continue
            elif i1a == "success" :
                while True:
                    print()
                    i1b = int(input(""" Welcome !!
1. Buy Games
2. Buy Consoles
3. Check Cart
0. Log Out
-----> """))


                    if i1b==0:
                        print("Thanks for visiting G-Hub!\n")
                        break

                    elif i1b==1:
                        while True:
                            i1b1 = input("""1. List All Games
2. List By Genre
To sort by price, add P after choice. (1P)
To sort by release year, add Y after choice.  (1Y)
0. Exit
------> """).upper()

                            if "1" in i1b1:
                                if "P" in i1b1:
                                    gamecursor.execute("select ID, Game_Name, Store_Price,
Game_Genre, Release_Year, List_Status from Games order by Store_Price")
                                    data=gamecursor.fetchall()
                                    sx=PrettyTable()
                                    sx.field_names=["ID","Game","Price","Genre","Release
Year"]
                                    for i in data:
```

```
                    if i[5]==1:
                        sx.add_row(i[:5])

                        GameProduct[i[0]]=["Game",i[0],i[1],i[2]]
                    print(sx)
                    gxx=AddToCart(GameProduct)
                    if gxx==None:
                        continue


                elif "Y" in i1b1:
                    gamecursor.execute("select ID, Game_Name, Store_Price,
Game_Genre, Release_Year, List_Status from Games order by Release_Year desc")
                    data=gamecursor.fetchall()
                    sx=PrettyTable()
                    sx.field_names=["ID","Game","Price","Genre","Release
Year"]
                    for i in data:
                        if i[5]==1:
                            sx.add_row(i[:5])

                            GameProduct[i[0]]=["Game",i[0],i[1],i[2]]
                    print(sx)
                    gxx=AddToCart(GameProduct)
                    if gxx==None:
                        continue

                else:
                    gamecursor.execute("select ID, Game_Name, Store_Price,
Game_Genre, Release_Year, List_Status from Games")
                    data=gamecursor.fetchall()
                    sx=PrettyTable()
                    sx.field_names=["ID","Game","Price","Genre","Release
Year"]
                    for i in data:
                        if i[5]==1:
                            sx.add_row(i[:5])

                            GameProduct[i[0]]=["Game",i[0],i[1],i[2]]
                    print(sx)
                    gxx=AddToCart(GameProduct)
                    if gxx==None:
                        continue


            elif "2" in i1b1:
```

```
                    gen=int((input("""Enter Genre
1. Open-World
2. Adventure
3. Sports
4. FPS
5. Platformer
0. Exit
-------> """)))
                    if gen==0:
                        continue
                    else:

                        if "P" in i1b1:

                            gamecursor.execute("select ID, Game_Name, Store_Price,
Game_Genre, Release_Year, List_Status from Games where Game_Genre='{}' order
by Store_Price".format(Genre[gen-1]))
                            data=gamecursor.fetchall()
                            sx=PrettyTable()
                            sx.field_names=["ID","Game","Price","Genre","Release
Year"]
                            for i in data:
                                if i[5]==1:
                                    sx.add_row(i[:5])

                                    GameProduct[i[0]]=["Game",i[0],i[1],i[2]]
                            print(sx)
                            gxx=AddToCart(GameProduct)
                            if gxx==None:
                                continue

                        elif "Y" in i1b1:
                            gamecursor.execute("select ID, Game_Name, Store_Price,
Game_Genre, Release_Year, List_Status from Games where Game_Genre='{}' order
by Release_Year desc".format(Genre[gen-1]))
                            data=gamecursor.fetchall()
                            sx=PrettyTable()
                            sx.field_names=["ID","Game","Price","Genre","Release
Year"]
                            for i in data:
                                if i[5]==1:
                                    sx.add_row(i[:5])

                                    GameProduct[i[0]]=["Game",i[0],i[1],i[2]]
                            print(sx)
                            gxx=AddToCart(GameProduct)
```

```
                        if gxx==None:
                            continue

                    else:

                        gamecursor.execute("select ID, Game_Name, Store_Price,
Game_Genre, Release_Year, List_Status from Games where
Game_Genre='{}';".format(Genre[gen-1]))
                        data=gamecursor.fetchall()
                        sx=PrettyTable()
                        sx.field_names=["ID","Game","Price","Genre","Release
Year"]
                        for i in data:
                            if i[5]==1:
                                sx.add_row(i[:5])

                                GameProduct[i[0]]=["Game",i[0],i[1],i[2]]
                        print(sx)
                        gxx=AddToCart(GameProduct)
                        if gxx==None:
                            continue



            elif "0" in i1b1:
                print("You will be taken back.....")
                break

            else:
                print("Invalid Input, Please Try Again!")
                continue

        elif i1b==2:
            while True:

                i1c1=input("""1. List All Consoles
2. HandHeld Consoles
3. Home Consoles
0. Exit
To sort by price, add P after choice. (1P)
To sort by release year, add Y after choice.  (1Y)
----->""").upper()

                if "0" in i1c1:
                    print("You will be taken back....")
```

```
                    break

                elif "1" in i1c1:
                    if "P" in i1c1:
                        consolecursor.execute("select ID, Console_Name, Store_Price,
Console_Type, Release_Year, List_Status from Consoles order by Store_Price")
                        data=consolecursor.fetchall()
                        sx=PrettyTable()
                        sx.field_names=["ID","Console","Price","Genre","Release
Year"]
                        for i in data:
                            if i[5]==1:
                                sx.add_row(i[:5])

                                ConsoleProduct[i[0]]=["Console",i[0],i[1],i[2]]
                        print(sx)
                        gxx=AddToCart(ConsoleProduct)
                        if gxx==None:
                            continue

                    elif "Y" in i1c1:
                        consolecursor.execute("select ID, Console_Name, Store_Price,
Console_Type, Release_Year, List_Status from Consoles order by Release_Year
desc")
                        data=consolecursor.fetchall()
                        sx=PrettyTable()
                        sx.field_names=["ID","Console","Price","Genre","Release
Year"]
                        for i in data:
                            if i[5]==1:
                                sx.add_row(i[:5])

                                ConsoleProduct[i[0]]=["Console",i[0],i[1],i[2]]
                        print(sx)
                        gxx=AddToCart(ConsoleProduct)
                        if gxx==None:
                            continue

                    else:
                        consolecursor.execute("select ID, Console_Name, Store_Price,
Console_Type, Release_Year, List_Status from Consoles")
                        data=consolecursor.fetchall()
                        sx=PrettyTable()
                        sx.field_names=["ID","Console","Price","Genre","Release
Year"]
                        for i in data:
```

```
                        if i[5]==1:
                            sx.add_row(i[:5])

                            ConsoleProduct[i[0]]=["Console",i[0],i[1],i[2]]
                    print(sx)
                    gxx=AddToCart(ConsoleProduct)
                    if gxx==None:
                        continue

                if "2" in i1c1:

                    if "P" in i1c1:
                        consolecursor.execute("select ID, Console_Name, Store_Price,
Console_Type, Release_Year, List_Status from Consoles where Console_Type='{}'
order by Store_Price".format("Handheld"))
                        data=consolecursor.fetchall()
                        sx=PrettyTable()
                        sx.field_names=["ID","Console","Price","Genre","Release
Year"]
                        for i in data:
                            if i[5]==1:
                                sx.add_row(i[:5])

                                ConsoleProduct[i[0]]=["Console",i[0],i[1],i[2]]
                        print(sx)
                        gxx=AddToCart(ConsoleProduct)
                        if gxx==None:
                            continue
                    elif "Y" in i1c1:
                        consolecursor.execute("select ID, Console_Name, Store_Price,
Console_Type, Release_Year, List_Status from Consoles where Console_Type='{}'
order by Release_Year desc".format("Handheld"))
                        data=consolecursor.fetchall()
                        sx=PrettyTable()
                        sx.field_names=["ID","Console","Price","Genre","Release
Year"]
                        for i in data:
                            if i[5]==1:
                                sx.add_row(i[:5])

                                ConsoleProduct[i[0]]=["Console",i[0],i[1],i[2]]
                        print(sx)
                        gxx=AddToCart(ConsoleProduct)
                        if gxx==None:
                            continue
                    else:
```

```
                        consolecursor.execute("select ID, Console_Name, Store_Price,
Console_Type, Release_Year, List_Status from Consoles where
Console_Type='{}'".format("Handheld"))
                        data=consolecursor.fetchall()
                        sx=PrettyTable()
                        sx.field_names=["ID","Console","Price","Genre","Release
Year"]
                        for i in data:
                            if i[5]==1:
                                sx.add_row(i[:5])

                                ConsoleProduct[i[0]]=["Console",i[0],i[1],i[2]]
                        print(sx)
                        gxx=AddToCart(ConsoleProduct)
                        if gxx==None:
                            continue

                    if "3" in i1c1:
                        if "P" in i1c1:
                            consolecursor.execute("select ID, Console_Name, Store_Price,
Console_Type, Release_Year, List_Status from Consoles where Console_Type='{}'
order by Store_Price".format("Home"))
                            data=consolecursor.fetchall()
                            sx=PrettyTable()
                            sx.field_names=["ID","Console","Price","Genre","Release
Year"]
                            for i in data:
                                if i[5]==1:
                                    sx.add_row(i[:5])

                                    ConsoleProduct[i[0]]=["Console",i[0],i[1],i[2]]
                            print(sx)
                            gxx=AddToCart(ConsoleProduct)
                            if gxx==None:
                                continue
                        elif "Y" in i1c1:
                            consolecursor.execute("select ID, Console_Name, Store_Price,
Console_Type, Release_Year, List_Status from Consoles where Console_Type='{}'
order by Release_Year desc".format("Home"))
                            data=consolecursor.fetchall()
                            sx=PrettyTable()
                            sx.field_names=["ID","Console","Price","Genre","Release
Year"]
                            for i in data:
                                if i[5]==1:
                                    sx.add_row(i[:5])
```

```
                        ConsoleProduct[i[0]]=["Console",i[0],i[1],i[2]]
                    print(sx)
                    gxx=AddToCart(ConsoleProduct)
                    if gxx==None:
                        continue
                else:
                    consolecursor.execute("select ID, Console_Name, Store_Price,
Console_Type, Release_Year, List_Status from Consoles where
Console_Type='{}'".format("Home"))
                    data=consolecursor.fetchall()
                    sx=PrettyTable()
                    sx.field_names=["ID","Console","Price","Genre","Release
Year"]
                    for i in data:
                        if i[5]==1:
                            sx.add_row(i[:5])

                            ConsoleProduct[i[0]]=["Console",i[0],i[1],i[2]]
                    print(sx)
                    gxx=AddToCart(ConsoleProduct)
                    if gxx==None:
                        continue



            elif i1b==3:
                i1bg=GoToCart()
                if i1bg==None:
                    continue




        elif i1 == 2:
            MemberSignUp()
            continue

    elif i==2:
        while True:
            i2=int(input(""" Welcome Developer!!
1. Sign In
0. Exit
----> """))
            if i2 == 0:
                print("Thank You!!")
```

```
            break
        elif i2 == 1:

            i2a=DeveloperSignIn()
            if i2a != "fail" :
                while True:
                    ddd=[]
                    ddc=[]

                    i2b = int(input(""" Welcome !!
1. Check Sales and Revenue
2. Check and Update Inventory
3. List New Product
0. Log Out
-----> """))
                    if i2b==0:
                        print("Thank you for visiting G-Hub!")
                        break

                    elif i2b==3:
                        i2bl=int(input("""List New
1. Game
2. Console
0. Exit
------> """))
                        if i2bl==0:
                            print("Hope you have something new for G-Hub next time! :>")
                            break
                        elif i2bl==1:
                            listcursor.execute("select * from Games")
                            ran=listcursor.fetchall()

                            GD=i2a
                            GID=len(ran)+1
                            GName=input("Enter Game name -->")
                            GSPrice=0
                            GDPrice=int(input("Enter Developer Price -->"))
                            GGenre=input("Enter Game Genre -->")
                            GYear=int(input("Enter Release Year -->"))
                            while True:
                                GStock=eval(input("Enter Stock for Game as list in format
[Stock Overall, Stock PS, Stock XBOX, Stock Nintendo] -->"))
                                if GStock[0]!=GStock[1]+GStock[2]+GStock[3]:
                                    print("Please enter valid stock numbers, and ensure that
PS/Xbox/Nintendo Stock total comes to Overall!!")
                                    continue
```

```
                    else:
                        break

                    listcursor.execute("insert into Games values
('{}',{},'{}',{},{},'{}',{},{},{},{},{},{},{},{},{},{},{},{},{},{})".format(GD,GID,G
Name,GSPrice,GDPrice,GGenre,GYear,GStock[0],GStock[1],GStock[2],GStock[3],0
,0,0,0,0,0,0,0,0))
                    mycon.commit()
                    print("Game Listed Succesfully. Will Soon Be Added By Store
Manager")
                    print()

                elif i2bl==2:
                    listcursor.execute("select * from Consoles")
                    ran=listcursor.fetchall()

                    CD=i2a
                    CID=len(ran)+1
                    CName=input("Enter Console Name --> ")
                    CSPrice=0
                    CDPrice=int(input("Enter Developer Price --> "))
                    CType=input("Enter Console Type (Handheld/Home) --> ")
                    CYear=int(input("ENter Release Year -->"))
                    CStock=int(input("Enter Stock -->"))

                    listcursor.execute("insert into Consoles values
('{}',{},'{}',{},{},'{}',{},{},{},{},{})".format(CD,CID,CName,CSPrice,CDPrice,CTy
pe,CYear,CStock,0,0,0))
                    mycon.commit()
                    print("Console Listed Succesfully. Will Soon Be Added By Store
Manager")
                    print()



            elif i2b==1:
                developercursor2.execute("select ID, Game_Name,
Developer_Price, Sales_Overall, Sales_PS, Sales_XBOX, Sales_Nintendo from
Games where Developer='{}'".format(i2a))
                datag=developercursor2.fetchall()
                dgx=PrettyTable()
                dgx.field_names=["ID","Game","Dev. Price","Sales Overall","Sales
PS","Sales XBOX","Sales Nintendo","Game Revenue"]
                gdrevenue=0
                for i in datag:
```

```
                    dgx.add_row(list(i[:7])+[i[2]*i[3]])
                    gdrevenue+=i[2]*i[3]


                print(dgx)
                print("----------X----------")
                print()
                print("Total Revenue from Games --> $",gdrevenue)



                developercursor2.execute("Select ID, Console_Name,
Developer_Price, Sales from Consoles where Developer='{}'".format(i2a))
                datac=developercursor2.fetchall()
                dcx=PrettyTable()
                dcx.field_names=["ID","Console","Dev. Price","Sales","Console
Revenue"]
                cdrevenue=0
                for i in datac:
                    dcx.add_row(list(i[:5])+[i[2]*i[3]])
                    cdrevenue+=i[2]*i[3]

                print(dcx)
                print("----------X----------")
                print("Total Revenue from Consoles --> $",cdrevenue)
                print()
                print()
                print("----------------------")
                print("Total Revenue --> $",cdrevenue+gdrevenue)

                print()
                print()
                continue


            elif i2b==2:
                i2b1=int(input("""Inventory for.....
1.Games
2.Consoles
0. Go Back
-----> """))
                if i2b1==0:
                    break
                elif i2b1==1:
```

```
                developercursor.execute("select ID, Game_Name,
Developer_Price, Stock_Overall, Stock_PS, Stock_Xbox, Stock_Nintendo,
List_Status from Games where Developer='{}'".format(i2a))
                data=developercursor.fetchall()
                dsgx=PrettyTable()
                dsgx.field_names=["ID", "Game", "Dev. Price", "Stock Overall",
"Stock PS", "Stock XBOX", "Stock Nintendo"]
                for i in data:
                    if i[7]==1:
                        ddd.append(i[0])
                        dsgx.add_row(i[:7])
                print(dsgx)

                while True:
                    updinput=int(input("Enter 0 to Go Back, Select ID to update
stock-->"))
                    if updinput == 0:
                        break
                    elif updinput not in ddd:
                        print("Enter valid choice!!")
                        continue
                    else:
                        while True:
                            upd2=eval(input("Enter Stock to be added as a list of
format [Stock_Overall, Stock_PS, Stock_Xbox, Stock_Nintendo]"))
                            if upd2[0]!=upd2[1]+upd2[2]+upd2[3]:
                                print("Enter valid choice")
                                continue
                            else:
                                developercursor.execute("update Games set
Stock_Overall=Stock_Overall+{}, Stock_PS=Stock_PS+{},
Stock_XBOX=Stock_XBOX+{}, Stock_Nintendo=Stock_Nintendo+{} where
ID={}".format(upd2[0],upd2[1],upd2[2],upd2[3],updinput))
                                print("Stock updated successfully!!")
                                mycon.commit()
                                print()
                                break

            elif i2b1==2:
                developercursor.execute("select ID, Console_Name,
Developer_Price, Stock, List_Status from Consoles where
Developer='{}'".format(i2a))
                data=developercursor.fetchall()
                dscx=PrettyTable()
                dscx.field_names=["ID","Console","Dev. Price","Stock"]
                for i in data:
```

```
                        if i[4]==1:
                            ddc.append(i[0])
                            dscx.add_row(i[:4])
                    print(dscx)

                    while True:
                        updcinput=int(input("Enter 0 to Go Back, Select ID to update
stock-->"))
                        if updcinput == 0:
                            break
                        elif updcinput not in ddc:
                            print("Enter valid choice!!")
                            continue
                        else:
                            while True:
                                updc2=int(input("Enter Stock to be added -->"))
                                developercursor.execute("update Consoles set
Stock=Stock+{} where ID={}".format(updc2,updcinput))
                                print("Stock updated successfully!!")
                                mycon.commit()
                                print()
                                break
            else:
                break

    elif i==3:
        countpin=0
        while True:
            InputPIN=int(input("'Enter PIN To Continue
Enter 0 To Return To Main Menu
--->'"))

            if InputPIN==0:
                print()
                break
            elif InputPIN!=StorePIN:
                if countpin==2:
                    print("Too Many Wrong Entries.You Will Be Taken Back To The Main
Menu")
                    print()
                    break
                else:
                    print("Wrong PIN, Try Again!")
                    print()
                    countpin+=1
```

```
                continue
            else:
                while True:
                    print()
                    i4=int(input("""Hello Manager!!
1. Update Store Discount
2. Special Message/Notice
3. Approve Listings
4. Console Sales
5. Game Sales
6. Overall Sales
0. Exit
-----> """))

                    if i4 == 0:
                        countpin=0
                        print("Thank You!")
                        print()
                        break

                    elif i4 == 1:
                        print("{} % is the current storewide discount".format(StoreDiscount))
                        StoreDiscount = int(input("Enter New Store Discount -->"))
                        print("{} % is the new Store Discount!".format(StoreDiscount) )
                        SpecialMessage1="{} % is the new Store
Discount!".format(StoreDiscount)
                        print()
                        continue

                    elif i4 == 2:
                        SpecialMessage=input("Enter Special Message/Notice :")
                        print("Done!")
                        print()
                        continue

                    elif i4==3:
                        choice=int(input("'1. Games
2. Consoles
--->'"))
                        if choice==1:
                            approvecursor.execute("select ID, Game_Name, Developer_Price,
Stock_Overall from games where List_Status=0")
                            pending=approvecursor.fetchall()
                            print("ID---Game_Name---Developer_Price---Stock_Overall")
                            for i in pending:
                                for j in range(0,4):
```

```
                        print(i[j],end=" -- ")
                    StorePrice=int(input("Enter Store Price for Game : "))
                    approvecursor.execute("update Games set
Store_Price={},List_Status={} where ID={}".format(StorePrice,1,i[0]))
                    mycon.commit()
                  print("All Games Listed!!")
                  print()
                  continue
                elif choice==2:
                    approvecursor.execute("select ID, Console_Name, Developer_Price,
Stock from Consoles where List_Status=0")
                    pending=approvecursor.fetchall()
                    print("ID---Console_Name---Developer_Price---Stock_Overall")
                    for i in pending:
                      for j in range(0,4):
                        print(i[j],end=" -- ")
                      StorePrice=int(input("Enter Store Price for Game : "))
                      approvecursor.execute("update Consoles set
Store_Price={},List_Status={} where ID={}".format(StorePrice,1,i[0]))
                      mycon.commit()
                    print("All Consoles Listed!!")
                    print()
                    continue
              elif i4==4:
                showcursor1.execute("select Developer, ID, Console_Name,
Store_Price, Developer_Price, Sales, Revenue from Consoles")
                dis=showcursor1.fetchall()
                ssc=PrettyTable()
                ssc.field_names=["Developer","ID","Console","Price","Dev.
Price","Sales","Revenue","Profit"]

                crevenue=0
                cprofit=0
                for i in dis:
                  crevenue+=i[6]
                  ssc.add_row(list(i[:7])+[i[6]-i[4]*i[5]])
                  cprofit+=i[6]-(i[4]*i[5])
                print(ssc)
                print("Total Revenue from Consoles --> ", crevenue)
                print("Total Profit from Consoles --> ", cprofit)
                print()
                continue


              elif i4==5:
```

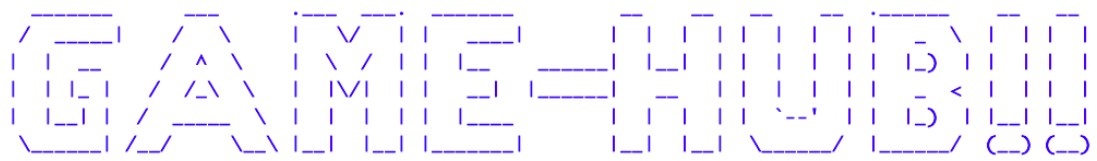
```
                showcursor2.execute("Select Developer, ID, Game_Name,
Store_Price, Developer_Price, Sales_Overall, Sales_PS, Sales_XBOX,
Sales_Nintendo, Revenue_Overall from Games")
                gis=showcursor2.fetchall()
                ssg=PrettyTable()
                ssg.field_names=["Developer","ID","Game","Price","Dev.
Price","Sales_Overall","Sales_PS","Sales_XBOX","Sales_Nintendo","Revenue","Pro
fit"]

                grevenue=0
                gprofit=0
                for i in gis:
                   grevenue+=i[9]
                   ssg.add_row(list(i[:10])+[i[9]-i[4]*i[5]])
                   gprofit+=i[9]-i[4]*i[5]
                print(ssg)
                print("Total Revenue from Games -->",grevenue)
                print("Total Profit from Games -->",gprofit)
                print()
                continue

            elif i4==6:

                showcursor1.execute("select Developer, ID, Console_Name,
Store_Price, Developer_Price, Sales, Revenue from Consoles")
                dis=showcursor1.fetchall()
                ssc=PrettyTable()
                ssc.field_names=["Developer","ID","Console","Price","Dev.
Price","Sales","Revenue","Profit"]

                crevenue=0
                cprofit=0
                for i in dis:
                   crevenue+=i[6]
                   ssc.add_row(list(i[:7])+[i[6]-i[4]*i[5]])
                   cprofit+=i[6]-(i[4]*i[5])
                print(ssc)
                print("Total Revenue from Consoles --> ", crevenue)
                print("Total Profit from Consoles --> ", cprofit)


                showcursor2.execute("Select Developer, ID, Game_Name,
Store_Price, Developer_Price, Sales_Overall, Sales_PS, Sales_XBOX,
Sales_Nintendo, Revenue_Overall from Games")
                gis=showcursor2.fetchall()
                ssg=PrettyTable()
```

```
                ssg.field_names=["Developer","ID","Game","Price","Dev.
Price","Sales_Overall","Sales_PS","Sales_XBOX","Sales_Nintendo","Revenue","Pro
fit"]

                grevenue=0
                gprofit=0
                for i in gis:
                    grevenue+=i[9]
                    ssg.add_row(list(i[:10])+[i[9]-i[4]*i[5]])
                    gprofit+=i[9]-i[4]*i[5]
                print(ssg)
                print("Total Revenue from Games -->",grevenue)
                print("Total Profit from Games -->", gprofit)


                print("---------------------------------------")
                print()
                print()
                print("Total Revenue --> $", grevenue+crevenue)
                print("Total Profit --> $", gprofit+cprofit)

                print()
                continue



        #COMPLETE
```

8. USER DOCUMENTATION

A - Buyer

1) Member Sign Up and Sign In



```
Welcome to G-Hub..... Happy Chinese New Year!!! Discount of 0 % on all products
1. User Login/Sign Up
2. Developer LogIn
3. Store Manager LogIn
0. Exit Store
----> 1
Welcome!!
1. Sign In
2. Sign Up
0. Exit
-----> 2
Enter a UserName: Akhx123
UserName available, Enter a Password: 12345678
Member Account created!! Please LogIn again!

Member Account created!! Please LogIn again!
Welcome to G-Hub..... Happy Chinese New Year!!! Discount of 0 % on all products
1. User Login/Sign Up
2. Developer LogIn
3. Store Manager LogIn
0. Exit Store
----> 1
Welcome!!
1. Sign In
2. Sign Up
0. Exit
-----> 1
Enter your UserName : Akhx123
Enter your Password : 12345678
Welcome Akhx123!!
```

2) Buying Games and Consoles

```
 Welcome !!
1. Buy Games
2. Buy Consoles
3. Check Cart
0. Log Out
-----> 1
1. List All Games
2. List By Genre
To sort by price, add P after choice. (1P)
To sort by release year, add Y after choice.  (1Y)
0. Exit
------> 1P
+----+---------------------------+-------+------------+--------------+
| ID |            Game           | Price |   Genre    | Release Year |
+----+---------------------------+-------+------------+--------------+
| 1  |           GTA V           |   50  | Open-World |     2013     |
| 6  |         Mario Bros        |   50  | Platformer |     2018     |
| 3  |        Uncharted 4        |   60  | Adventure  |     2016     |
| 4  |          FIFA 21          |   70  |   Sports   |     2020     |
| 2  | Assassin Creed : Valhalla |   80  | Open-World |     2020     |
| 5  |       Halo Infinite       |   80  |    FPS     |     2021     |
+----+---------------------------+-------+------------+--------------+
Enter ID to Select Product
Enter 0 to Go Back
--->6
```

```
Select Console
1. PS
2. XBOX
3. NINTENDO
--->3
You Have Selected Mario Bros
1. Buy Product
2. Add to Cart
0. Exit
--->2
Mario Bros successfully added! Cart has 1 items.
0. Return and Shop
1. View Cart
--->0
1. List All Games
2. List By Genre
To sort by price, add P after choice. (1P)
To sort by release year, add Y after choice.  (1Y)
0. Exit
```

```
------> 2
Enter Genre
1. Open-World
2. Adventure
3. Sports
4. FPS
5. Platformer
0. Exit
-------> 1
+----+---------------------------+-------+------------+--------------+
| ID |            Game           | Price |   Genre    | Release Year |
+----+---------------------------+-------+------------+--------------+
| 1  |          GTA V            |   50  | Open-World |     2013     |
| 2  | Assassin Creed : Valhalla |   80  | Open-World |     2020     |
+----+---------------------------+-------+------------+--------------+
Enter ID to Select Product
Enter 0 to Go Back
--->2
Select Console
1. PS
2. XBOX
3. NINTENDO
--->2
You Have Selected Assassin Creed : Valhalla
1. Buy Product
2. Add to Cart
0. Exit
--->2
Assassin Creed : Valhalla successfully added! Cart has 2 items.
0. Return and Shop
1. View Cart
--->0
1. List All Games
2. List By Genre
To sort by price, add P after choice. (1P)
To sort by release year, add Y after choice.  (1Y)
0. Exit
------> 0
You will be taken back......
 Welcome !!
1. Buy Games
2. Buy Consoles
3. Check Cart
0. Log Out
-----> 2
```

```
1. List All Consoles
2. HandHeld Consoles
3. Home Consoles
0. Exit
To sort by price, add P after choice. (1P)
To sort by release year, add Y after choice.  (1Y)
----->1
+----+----------------+-------+----------+--------------+
| ID |     Console    | Price |  Genre   | Release Year |
+----+----------------+-------+----------+--------------+
| 1  |    XBOX ONE S  |  400  |   Home   |     2016     |
| 2  |  XBOX SERIES X |  700  |   Home   |     2020     |
| 3  |  XBOX SERIES S |  500  |   Home   |     2020     |
| 4  |     PS4 Pro    |  600  |   Home   |     2017     |
| 5  |       PS5      |  750  |   Home   |     2020     |
| 6  |       PSP      |  200  | Handheld |     2008     |
| 7  | Nintendo Switch |  400  |  Hybrid  |     2017     |
| 8  |   Nintendo 3DS |  250  | Handheld |     2012     |
+----+----------------+-------+----------+--------------+
Enter ID to Select Product
Enter 0 to Go Back
--->2
You Have Selected XBOX SERIES X
1. Buy Product
2. Add to Cart
0. Exit
--->2
XBOX SERIES X successfully added! Cart has 3 items.
0. Return and Shop
1. View Cart
--->0
1. List All Consoles
2. HandHeld Consoles
3. Home Consoles
0. Exit
To sort by price, add P after choice. (1P)
To sort by release year, add Y after choice.  (1Y)
----->0
You will be taken back....
```

3) Viewing And Editing Cart

```
 Welcome !!
1. Buy Games
2. Buy Consoles
3. Check Cart
0. Log Out
-----> 3
Your Cart -->
+-----+---------------+----+---------------------------+-------+
| No. |      Type     | ID |            Item           | Price |
+-----+---------------+----+---------------------------+-------+
|  1  | Game-Nintendo | 6  |         Mario Bros        |   50  |
|  2  |   Game-XBOX   | 2  | Assassin Creed : Valhalla |   80  |
|  3  |    Console    | 2  |       XBOX SERIES X       |  700  |
+-----+---------------+----+---------------------------+-------+
Actions:
1. Checkout Now
2. Delete One Item
3. Clear Cart
0. Go to Menu
--->2
Enter Product No. to Delete -->1
Item Deleted Successfully!
```

4) Checking Out

```
Actions:
1. Checkout Now
2. Delete One Item
3. Clear Cart
0. Go to Menu
--->1
Total Amount is $780.0 for 2 items
Stock is available!! Price is $ 80.0
Assassin Creed : Valhalla PURCHASE SUCCESSFUL!!

Stock is available!! Price is $ 700.0
XBOX SERIES X PURCHASE SUCCESSFUL!!

Thank you for visiting G-Hub!! :)
You will now be taken to your previous Menu

 Welcome !!
1. Buy Games
2. Buy Consoles
3. Check Cart
0. Log Out
-----> 0
Thanks for visiting G-Hub!
Welcome to G-Hub..... Happy Chinese New Year!!! Discount of 0 % on all products
1. User Login/Sign Up
2. Developer LogIn
3. Store Manager LogIn
0. Exit Store
----> 0
Thank You for Visiting G-Hub!
```

B - Developer

1) Developer Login

```
  _______     ___     .___  ___.  _______        __    __   __    __  .______    __  ___
 /  _____|   /   \    |   \/   | |   ____|      |  |  |  | |  |  |  | |   _  \  |  | |  |
|  |  __    /  ^  \   |  \  /  | |  |__   _____|  |__|  | |  |  |  | |  |_)  | |  | |  |
|  | |_ |  /  /_\  \  |  |\/|  | |   __| |______|   __   | |  |  |  | |   _ <  |  | |  |
|  |__| | /  _____  \ |  |  |  | |  |____        |  |  |  | |  `--'  | |  |_)  | |__| |__|
 \______|/__/     \__\|__|  |__| |_______|       |__|  |__|  \______/  |______/  (__) (__)

Welcome to G-Hub..... Happy Chinese New Year!!! Discount of 0 % on all products
1. User Login/Sign Up
2. Developer LogIn
3. Store Manager LogIn
0. Exit Store
----> 2
 Welcome Developer!!
1. Sign In
0. Exit
----> 1
Enter your DeveloperName : Sony
Enter your Password : GOD
Welcome Sony!!
```

2) Checking Sales And Revenue

```
Welcome Sony!!
 Welcome !!
1. Check Sales and Revenue
2. Check and Update Inventory
3. List New Product
0. Log Out
-----> 1
+----+--------------------+------------+---------------+----------+-----------+----------------+--------------+
| ID |        Game        | Dev. Price | Sales Overall | Sales PS | Sales XBOX | Sales Nintendo | Game Revenue |
+----+--------------------+------------+---------------+----------+-----------+----------------+--------------+
| 9  | Marvel's Spiderman |     30     |       0       |    0     |     0     |       0        |      0       |
+----+--------------------+------------+---------------+----------+-----------+----------------+--------------+
----------X-----------

Total Revenue from Games --> $ 0
+----+---------+------------+-------+-----------------+
| ID | Console | Dev. Price | Sales | Console Revenue |
+----+---------+------------+-------+-----------------+
| 4  | PS4 Pro |    400     |   0   |        0        |
| 5  |   PS5   |    550     |   0   |        0        |
| 6  |   PSP   |    100     |   0   |        0        |
+----+---------+------------+-------+-----------------+
----------X-----------
Total Revenue from Consoles --> $ 0


----------------------
Total Revenue --> $ 0
```

3) Checking Inventory And Updating Stock

```
 Welcome !!
1. Check Sales and Revenue
2. Check and Update Inventory
3. List New Product
0. Log Out
-----> 2
Inventory for.......
1.Games
2.Consoles
0. Go Back
-----> 1
+----+--------------------+------------+---------------+----------+-----------+----------------+
| ID |        Game        | Dev. Price | Stock Overall | Stock PS | Stock XBOX | Stock Nintendo |
+----+--------------------+------------+---------------+----------+-----------+----------------+
| 9  | Marvel's Spiderman |     30     |       20      |    20    |     0     |       0        |
+----+--------------------+------------+---------------+----------+-----------+----------------+
Enter 0 to Go Back, Select ID to update stock-->9
Enter Stock to be added as a list of format [Stock_Overall, Stock_PS, Stock_Xbox, Stock_Nintendo][10,10,0,0]
Stock updated successfully!!
Enter 0 to Go Back, Select ID to update stock-->0
```

4) Listing A New Product

```
 Welcome !!
1. Check Sales and Revenue
2. Check and Update Inventory
3. List New Product
0. Log Out
-----> 3
List New
1. Game
2. Console
0. Exit
------> 2
Enter Console Name --> PS Vita
Enter Developer Price --> 300
Enter Console Type (Handheld/Home) --> Handheld
ENter Release Year -->2012
Enter Stock -->20
Console Listed Succesfully. Will Soon Be Added By Store Manager
```

C - Manager

1) Manager Sign In

```
  _______      ___      .___  ___.  _______        __    __   __    __   .______     __   __
 /  _____|    /   \     |   \/   | |   ____|      |  |  |  | |  |  |  |  |   _  \   |  | |  |
|  |  __     /  ^  \    |  \  /  | |  |__    _____|  |__|  | |  |  |  |  |  |_)  |  |  | |  |
|  | |_ |   /  /_\  \   |  |\/|  | |   __|  |_____|   __   | |  |  |  |  |   _  <   |  | |  |
|  |__| |  /  _____  \  |  |  |  | |  |____        |  |  |  | |  `--'  |  |  |_)  |  |__| |__|
 \______| /__/     \__\ |__|  |__| |_______|       |__|  |__|  \______/   |______/  (__) (__)


Welcome to G-Hub..... Happy Chinese New Year!!! Discount of 0 % on all products
1. User Login/Sign Up
2. Developer LogIn
3. Store Manager LogIn
0. Exit Store
----> 3
Enter PIN To Continue
Enter 0 To Return To Main Menu
--->1104

Hello Manager!!
```

2) Approve Listings

a) Approve Games

```
1. Update Store Discount
2. Special Message/Notice
3. Approve Listings
4. Console Sales
5. Game Sales
6. Overall Sales
0. Exit
-----> 3
1. Games
2. Consoles
--->1
ID---Game_Name---Developer_Price---Stock_Overall
8 -- Max Payne III -- 20 -- 20 -- Enter Store Price for Game : 40
All Games Listed!!
```

---

b) Approve Consoles

```
Hello Manager!!
1. Update Store Discount
2. Special Message/Notice
3. Approve Listings
4. Console Sales
5. Game Sales
6. Overall Sales
0. Exit
-----> 3
1. Games
2. Consoles
--->2
ID---Console_Name---Developer_Price---Stock_Overall
10 -- PS Vita -- 150 -- 20 -- Enter Store Price for Game : 250
All Consoles Listed!!
```

---

3) Viewing Console Sales

```
Hello Manager!!
1. Update Store Discount
2. Special Message/Notice
3. Approve Listings
4. Console Sales
5. Game Sales
6. Overall Sales
0. Exit
-----> 4
+-----------+----+----------------------+-------+------------+-------+---------+--------+
| Developer | ID |       Console        | Price | Dev. Price | Sales | Revenue | Profit |
+-----------+----+----------------------+-------+------------+-------+---------+--------+
| Microsoft | 1  |      XBOX ONE S      |  400  |    200     |   0   |    0    |   0    |
| Microsoft | 2  |    XBOX SERIES X     |  700  |    550     |   1   |   700   |  150   |
| Microsoft | 3  |    XBOX SERIES S     |  500  |    300     |   0   |    0    |   0    |
|    Sony   | 4  |       PS4 Pro        |  600  |    400     |   0   |    0    |   0    |
|    Sony   | 5  |         PS5          |  750  |    550     |   0   |    0    |   0    |
|    Sony   | 6  |         PSP          |  200  |    100     |   0   |    0    |   0    |
|  Nintendo | 7  |   Nintendo Switch    |  400  |    250     |   0   |    0    |   0    |
|  Nintendo | 8  |     Nintendo 3DS     |  250  |    150     |   0   |    0    |   0    |
|  Nintendo | 9  | Nintendo Switch Lite |  300  |    200     |   0   |    0    |   0    |
|    Sony   | 10 |       PS Vita        |  250  |    150     |   0   |    0    |   0    |
+-----------+----+----------------------+-------+------------+-------+---------+--------+
Total Revenue from Consoles -->  700
Total Profit from Consoles -->  150
```

---

4) Viewing Game Sales

```
Hello Manager!!
1. Update Store Discount
2. Special Message/Notice
3. Approve Listings
4. Console Sales
5. Game Sales
6. Overall Sales
0. Exit
-----> 5
+----------------+----+--------------------------+-------+-----------+--------------+----------+-----------+---------------+---------+--------+
|    Developer   | ID |           Game           | Price | Dev. Price | Sales_Overall | Sales_PS | Sales_XBOX | Sales_Nintendo | Revenue | Profit |
+----------------+----+--------------------------+-------+-----------+--------------+----------+-----------+---------------+---------+--------+
|    Rockstar    | 1  |          GTA V           |  50   |    20     |      0       |    0     |     0     |       0       |    0    |   0    |
|    Ubisoft     | 2  | Assassin Creed : Valhalla |  80   |    50     |      1       |    0     |     1     |       0       |   80    |   30   |
|  Naughty Dog   | 3  |       Uncharted 4        |  60   |    40     |      0       |    0     |     0     |       0       |    0    |   0    |
| Electronic Arts | 4  |         FIFA 21          |  70   |    45     |      0       |    0     |     0     |       0       |    0    |   0    |
|   Microsoft    | 5  |       Halo Infinite      |  80   |    60     |      0       |    0     |     0     |       0       |    0    |   0    |
|    Nintendo    | 6  |        Mario Bros        |  50   |    30     |      0       |    0     |     0     |       0       |    0    |   0    |
|    Rockstar    | 7  |  Red Dead Redemption II  |  60   |    40     |      0       |    0     |     0     |       0       |    0    |   0    |
|    Rockstar    | 8  |      Max Payne III       |  40   |    20     |      0       |    0     |     0     |       0       |    0    |   0    |
|      Sony      | 9  |    Marvel's Spiderman    |  60   |    30     |      0       |    0     |     0     |       0       |    0    |   0    |
+----------------+----+--------------------------+-------+-----------+--------------+----------+-----------+---------------+---------+--------+
Total Revenue from Games --> 80
Total Profit from Games --> 30
```

---

5) Viewing Overall Sales

```
Hello Manager!!
1. Update Store Discount
2. Special Message/Notice
3. Approve Listings
4. Console Sales
5. Game Sales
6. Overall Sales
0. Exit
-----> 6
+-----------+----+----------------------+-------+------------+-------+---------+--------+
| Developer | ID |       Console        | Price | Dev. Price | Sales | Revenue | Profit |
+-----------+----+----------------------+-------+------------+-------+---------+--------+
| Microsoft | 1  |      XBOX ONE S      |  400  |    200     |   0   |    0    |   0    |
| Microsoft | 2  |    XBOX SERIES X     |  700  |    550     |   1   |   700   |  150   |
| Microsoft | 3  |    XBOX SERIES S     |  500  |    300     |   0   |    0    |   0    |
|    Sony   | 4  |       PS4 Pro        |  600  |    400     |   0   |    0    |   0    |
|    Sony   | 5  |         PS5          |  750  |    550     |   0   |    0    |   0    |
|    Sony   | 6  |         PSP          |  200  |    100     |   0   |    0    |   0    |
|  Nintendo | 7  |   Nintendo Switch    |  400  |    250     |   0   |    0    |   0    |
|  Nintendo | 8  |     Nintendo 3DS     |  250  |    150     |   0   |    0    |   0    |
|  Nintendo | 9  | Nintendo Switch Lite |  300  |    200     |   0   |    0    |   0    |
|    Sony   | 10 |       PS Vita        |  250  |    150     |   0   |    0    |   0    |
+-----------+----+----------------------+-------+------------+-------+---------+--------+
Total Revenue from Consoles -->  700
Total Profit from Consoles -->  150
+---------------+----+-------------------------+-------+------------+---------------+----------+-----------+---------------+---------+--------+
|   Developer   | ID |          Game           | Price | Dev. Price | Sales_Overall | Sales_PS | Sales_XBOX | Sales_Nintendo | Revenue | Profit |
+---------------+----+-------------------------+-------+------------+---------------+----------+-----------+---------------+---------+--------+
|    Rockstar   | 1  |         GTA V           |  50   |     20     |       0       |    0     |     0     |       0       |    0    |   0    |
|    Ubisoft    | 2  | Assassin Creed : Valhalla |  80   |     50     |       1       |    0     |     1     |       0       |   80    |   30   |
|  Naughty Dog  | 3  |       Uncharted 4       |  60   |     40     |       0       |    0     |     0     |       0       |    0    |   0    |
| Electronic Arts | 4  |         FIFA 21         |  70   |     45     |       0       |    0     |     0     |       0       |    0    |   0    |
|   Microsoft   | 5  |       Halo Infinite     |  80   |     60     |       0       |    0     |     0     |       0       |    0    |   0    |
|    Nintendo   | 6  |        Mario Bros       |  50   |     30     |       0       |    0     |     0     |       0       |    0    |   0    |
|    Rockstar   | 7  |  Red Dead Redemption II |  60   |     40     |       0       |    0     |     0     |       0       |    0    |   0    |
|    Rockstar   | 8  |      Max Payne III      |  40   |     20     |       0       |    0     |     0     |       0       |    0    |   0    |
|      Sony     | 9  |    Marvel's Spiderman   |  60   |     30     |       0       |    0     |     0     |       0       |    0    |   0    |
+---------------+----+-------------------------+-------+------------+---------------+----------+-----------+---------------+---------+--------+
Total Revenue from Games --> 80
Total Profit from Games --> 30
-------------------------------------


Total Revenue --> $ 780
Total Profit --> $ 180
```

6) Special Message/Notice

```
Hello Manager!!
1. Update Store Discount
2. Special Message/Notice
3. Approve Listings
4. Console Sales
5. Game Sales
6. Overall Sales
0. Exit
-----> 2
Enter Special Message/Notice :Wishing You A Happy Valentine's Day
Done!
```

```
Welcome to G-Hub..... Discount of 0 % on all products . Wishing You A Happy Valentine's Day
1. User Login/Sign Up
2. Developer LogIn
3. Store Manager LogIn
0. Exit Store
```

7) Approving Discounts

```
Hello Manager!!
1. Update Store Discount
2. Special Message/Notice
3. Approve Listings
4. Console Sales
5. Game Sales
6. Overall Sales
0. Exit
-----> 1
0 % is the current storewide discount
Enter New Store Discount -->10
10 % is the new Store Discount!
```

```
 Welcome !!
1. Buy Games
2. Buy Consoles
3. Check Cart
0. Log Out
-----> 1
1. List All Games
2. List By Genre
To sort by price, add P after choice. (1P)
To sort by release year, add Y after choice.  (1Y)
0. Exit
------> 1
+----+--------------------------+-------+------------+--------------+
| ID |           Game           | Price |   Genre    | Release Year |
+----+--------------------------+-------+------------+--------------+
| 1  |          GTA V           |   50  | Open-World |     2013     |
| 2  | Assassin Creed : Valhalla |   80  | Open-World |     2020     |
| 3  |       Uncharted 4        |   60  | Adventure  |     2016     |
| 4  |         FIFA 21          |   70  |   Sports   |     2020     |
| 5  |      Halo Infinite       |   80  |    FPS     |     2021     |
| 6  |        Mario Bros        |   50  | Platformer |     2018     |
| 7  |  Red Dead Redemption II  |   60  | Open-World |     2018     |
| 8  |      Max Payne III       |   40  | Adventure  |     2010     |
| 9  |    Marvel's Spiderman    |   60  | Open-World |     2018     |
+----+--------------------------+-------+------------+--------------+
Enter ID to Select Product
Enter 0 to Go Back
--->2
Select Console
1. PS
2. XBOX
3. NINTENDO
--->2
You Have Selected Assassin Creed : Valhalla
1. Buy Product
2. Add to Cart
0. Exit
```

```
--->1
Stock is available!! Price is $ 72.0
Assassin Creed : Valhalla PURCHASE SUCCESSFUL!!
```

(Price in Table is $80. Price Paid is $72)

9. BIBLIOGRAPHY

1) https://www.python.org/doc/essays/blurb/

2) Computer Science Python - Sumita Arora

3) https://zetcode.com/python/prettytable/#:~:text=PrettyTable%20deleting%20rows,that%20indexing%20start%20from%20zero.&text=In%20the%20example%2C%20we%20delete%20last%20four%20rows.

4) https://en.wikipedia.org/wiki/MySQL