11/9/2020

St. Anselm's Pink City Sr. Sec. School, Jaipur

ST. ANSELM'S PINK CITY SR. SEC. SCHOOL

# **Personal Computer Gaming Benchmark**

Made by Aaradhya Maharishi

# INDEX

# CERTIFICATE

This is to certify that the Informatics Practices project titled "**Personal Computer Gaming Benchmark**" has been successfully completed by _________________ of St. Anselm's Pink City Sr. Sec. School, Jaipur, of **Class XII-B,** board number **___________** under the guidance of Ms. Krishna Singh. This project is submitted as partial fulfillment for CBSE AISSCE (All India Senior School Certificate Examination), during the year 2020-2021.

Faculty Signature

(Ms. Krishna Singh)

# Acknowledgment

1. To my teachers for helping me to use the program and apply it practically.

2. To my Principal for providing us the atmosphere, computers and useful products on which we can work.

3. To my fellow batch mates for helping me out when in need or in problem regarding usage of programs.

4. To the staff who keeps our equipment's safe i.e. in working condition.

# EXISTING SYSTEM

***Till now if we have to buy a game we have to check that the game would run on our PC so to do this we have to go through many sites to see that configurations would match.***

# PROPOSED SYSTEM

***System requirement application to find games that can run on user's computer, And to provide user with Game Price, Price of the PC(Based upon the configuration provided by user + other important parts required to build the PC)***

## Hardware Requirement & Software Requirement

*Recommended System Requirements*

- ***Processors:Intel® Core™ i5 processor 4300M at 2.60 GHz or 2.59 GHz (1 socket, 2 cores, 2 threads per core), 8 GB of DRAMIntel® Xeon® processor E5-2698 v3 at 2.30 GHz (2 sockets, 16 cores each, 1 thread per core), 64 GB of DRAMIntel® Xeon Phi™ processor 7210 at 1.30 GHz (1 socket, 64 cores, 4 threads per core), 32 GB of DRAM, 16 GB of MCDRAM (flat mode enabled)***
- ***Disk space: 2 to 3 GB***
- ***Operating systems: Windows® 10, macOS*, and Linux****

*Minimum System Requirements*

- ***Processors: Intel Atom® processor or Intel® Core™ i3 processor***
- ***Disk space: 1 GB***
- ***Operating systems: Windows* 7 or later, macOS, and Linux***
- ***Python* versions: 2.7.X, 3.6.X***

# CODING (Python)

```
def main():
    c = 'y'
    while c == 'y':
        loginwindow = input("TO Login as USER (Enter U)\nTO
Login as ADMIN (Enter A) \n: ").upper()
        print()
        if loginwindow == "A":
            adminwindow()
        elif loginwindow == "U":
            program()
        else:
            print("Invalid Input")
            exit()
    else:
        print('wrong input')


def program():
    cont = "y"
    if cont == "y":
        def pcgaming():
            import pandas as pd
            import mysql.connector
            pcgbm = mysql.connector.connect(host="localhost",
user="root", passwd='abc@123', database="pcgbm")
            print("Can I Run It ?")
            print(
                "System requirement application to find games
that can run on your computer,\nAnd to provide you with Game
Price, "
                "Price of the PC(Based upon the configuration
provided by you + other important parts required to build the
PC)\n"
                "What games your laptop/PC can run - from our
list of over 70 PC games. \n")
            print("Tell us about your PC specification from the
following option's\n ")
            graphics = (input("Choose from the folLowing
GRAPHICS CARD\n"
```

```
                             "   Geforce GTX 1050 Ti  (ENTER
g1)\n "
                             "  Geforce GTX 1060    (ENTER
g2)\n "
                             "  Geforce GTX 1070 Ti (ENTER
g3)\n "
                             " :  ")).lower()
           print()
           g1 = "Geforce GTX 1050 Ti"
           g2 = "Geforce GTX 1060"
           g3 = "Geforce GTX 1070 Ti"
           if graphics == "g1":
               g = g1
           elif graphics == "g2":
               g = g2
           elif graphics == "g3":
               g = g3
           else:
               return "ERROR"
           processor = (input("Choose from the folLowing
PROCESSOR\n"
                              "  Intel core i3 (ENTER p1)\n "
                              " Intel core i5 (ENTER p2)\n "
                              " Intel core i7 (ENTER p3)\n "
                              " :  ")).lower()
           print()
           p1 = "i3"
           p2 = "i5"
           p3 = "i7"
           if processor == "p1":
               p = p1
           elif processor == "p2":
               p = p2
           elif processor == "p3":
               p = p3
           else:
               return "ERROR"
           memory = (input("Choose from the folLowing
MEMORY\n"
                           " 4GB RAM (ENTER r1)\n"
                           " 8GB RAM (ENTER r2)\n"
                           "16GB RAM (ENTER r3)\n"
                           " :  ")).lower()
           r1 = "4GB"
```

```
            r2 = "8GB"
            r3 = "16GB"
            if memory == "r1":
                r = r1
            elif memory == "r2":
                r = r2
            elif memory == "r3":
                r = r3
            else:
                return "ERROR"
            cursor = pcgbm.cursor()
            cursor.execute(
                f'select
G_name,storage_GB_,price_$_,ReviewScore_outof100,PCprice_Rs
from pcgbm where processor="{p}" and graphics="{g}" and
memory="{r}"')
            myresult = cursor.fetchall()
            data = pd.DataFrame(myresult, columns=['Game Name',
'Storage(GB)', 'Price($)', 'ReviewScore(out of 100)',

'PCprice(Rs)'])
            pd.set_option('display.max_rows' and
'display.max_columns', None)
            for row in range(len(data)):
                print()
                print(data.loc[row])
                print()

            def graph():
                def again():
                    again1 = input("\nDO YOU WANT TO SEE MORE
GRAPHS (ENTER Y) : ").upper()
                    if again1 == 'Y':
                        graph()

                import matplotlib.pyplot as mat

                def mypieU():
                    label1 = ['Geforce GTX 1050 Ti', 'Geforce
GTX 1060', 'Geforce GTX 1070 Ti']
                    sizes = [23, 27, 21]
                    colors1 = ['yellowgreen', 'lightskyblue',
'magenta']
```

```
                   mat.title("Games in PCGBM for various
Graphics card")
                   mat.pie(sizes, explode=None, labels=label1,
colors=colors1, shadow=True, autopct='%1.1f%%',
                           startangle=140)
                   mat.axis('equal')
                   a = mat.show()
                   print(a)

                   label1 = ['i3', 'i5', 'i7']
                   sizes = [25, 22, 24]
                   colors1 = ['red', 'blue', 'magenta']
                   mat.title("Games in PCGBM for various
Processors")
                   mat.pie(sizes, explode=None, labels=label1,
colors=colors1, shadow=True, autopct='%1.1f%%',
                           startangle=140)
                   mat.axis('equal')
                   b = mat.show()
                   print(b)

                   label1 = ['4GB', '8GB', '16GB']
                   sizes = [24, 19, 28]
                   colors1 = ['lightskyblue', 'yellowgreen',
'brown']
                   mat.title("Games in PCGBM for various RAM")
                   mat.pie(sizes, explode=None, labels=label1,
colors=colors1, shadow=True, autopct='%1.1f%%',
                           startangle=140)
                   mat.axis('equal')
                   c = mat.show()
                   print(c)

                   label1 = ['Free', '1 to 20', '$21 to $40',
'$41 to $60', '$61 to $100']
                   sizes = [14, 22, 20, 12, 3]
                   colors1 = ['red', 'blue', 'magenta',
'yellowgreen', 'brown']
                   mat.title("Games in PCGBM for various
prices($)\n")
                   mat.pie(sizes, explode=None, labels=label1,
colors=colors1, shadow=True, autopct='%1.1f%%',
                           startangle=140)
                   mat.axis('equal')
```

```
          d = mat.show()
          print(d)

          label1 = ['40 to 60', '61 to 80', '81 to
100']
          sizes = [3, 19, 48]
          colors1 = ['red', 'blue', 'cyan', ]
          mat.title("Games in PCGBM for various
Review Score out of 100\n")
          mat.pie(sizes, explode=None, labels=label1,
colors=colors1, shadow=True, autopct='%1.1f%%',
              startangle=140)
          mat.axis('equal')
          e = mat.show()
          print(e)

        def barU():
          import numpy as np
          import matplotlib.pyplot as mat15
          objects = ('Geforce GTX 1050 Ti', 'Geforce
GTX 1060', 'Geforce GTX 1070 Ti')
          y_pos = np.arange(len(objects))
          types = (23, 27, 21)
          mat15.bar(y_pos, types, align='center',
color='yellowgreen')
          mat15.xticks(y_pos, objects)
          mat15.ylabel('Number Of Games')
          mat15.title('Number Of Games According to
Graphics Card')
          mat15.show()

          import numpy as np
          import matplotlib.pyplot as mat14
          objects = ('i3', 'i5', 'i7')
          y_pos = np.arange(len(objects))
          types = (25, 22, 24)
          mat14.bar(y_pos, types, align='center',
color='lightgreen')
          mat14.xticks(y_pos, objects)
          mat14.ylabel('Number Of Games')
          mat14.title('Number Of Games According to
processor')
          mat14.show()
```

```
                   import numpy as np
                   import matplotlib.pyplot as mat13
                   objects = ('4GB', '8GB', '16GB')
                   y_pos = np.arange(len(objects))
                   types = (24, 19, 28)
                   mat13.bar(y_pos, types, align='center',
color='lightskyblue')
                   mat13.xticks(y_pos, objects)
                   mat13.ylabel('Number Of Games')
                   mat13.title('Number Of Games According to
RAM')
                   mat13.show()

                   import numpy as np
                   import matplotlib.pyplot as mat12
                   objects = ('Free', '1 to 20', '$21 to $40',
'$41 to $60', '$61 to $100')
                   y_pos = np.arange(len(objects))
                   types = (14, 22, 20, 12, 3)
                   mat12.bar(y_pos, types, align='center',
color='brown')
                   mat12.xticks(y_pos, objects)
                   mat12.ylabel('Number Of Games')
                   mat12.title('Number Of Games According to
Various prices($)')
                   mat12.show()

                   import numpy as np
                   import matplotlib.pyplot as mat11
                   objects = ('40 to 60', '61 to 80', '81 to
100')
                   y_pos = np.arange(len(objects))
                   types = (3, 19, 48)
                   mat11.bar(y_pos, types, align='center',
color='magenta')
                   mat11.xticks(y_pos, objects)
                   mat11.ylabel('Number Of Games')
                   mat11.title('Number Of Games According to
Various review score(out of 100)')
                   mat11.show()

               def lineU():
                   label1 = ['Geforce GTX 1050 Ti', 'Geforce
GTX 1060', 'Geforce GTX 1070 Ti']
```

```
                   sizes = [23, 27, 21]
                   import matplotlib.pyplot as mat10
                   mat10.plot(label1, sizes)
                   mat10.title("Games in PCGBM for various
Graphics card")
                   mat10.xlabel('Graphics card')
                   mat10.ylabel('No. Of Games')
                   mat10.show()

                   label1 = ['i3', 'i5', 'i7']
                   sizes = [25, 22, 24]
                   import matplotlib.pyplot as mat9
                   mat9.plot(label1, sizes)
                   mat9.title("Games in PCGBM for various
Processors")
                   mat9.xlabel('Processors')
                   mat9.ylabel('No. Of Games')
                   mat9.show()

                   label1 = ['4GB', '8GB', '16GB']
                   sizes = [24, 19, 28]
                   import matplotlib.pyplot as mat8
                   mat8.plot(label1, sizes)
                   mat8.title("Games in PCGBM for various
RAM")
                   mat8.xlabel('Ram')
                   mat8.ylabel('No. Of Games')
                   mat8.show()

                   label1 = ['Free', '1 to 20', '$21 to $40',
'$41 to $60', '$61 to $100']
                   sizes = [14, 22, 20, 12, 3]
                   import matplotlib.pyplot as mat7
                   mat7.plot(label1, sizes)
                   mat7.title("Games in PCGBM for various
prices($)\n")
                   mat7.xlabel('Prices($)')
                   mat7.ylabel('No. Of Games')
                   mat7.show()

                   label1 = ['40 to 60', '61 to 80', '81 to
100']
                   sizes = [3, 19, 48]
                   import matplotlib.pyplot as mat6
                   mat6.plot(label1, sizes)
                   mat6.title("Games in PCGBM for various
Review Score out of 100\n")
                   mat6.xlabel('Review score')
                   mat6.ylabel('No. Of Games')
                   mat6.show()

               def scatterU():

                   import matplotlib.pyplot as mat5
                   label1 = ['Geforce GTX 1050 Ti', 'Geforce
GTX 1060', 'Geforce GTX 1070 Ti']
                   sizes = [23, 27, 21]
                   mat5.scatter(label1, sizes, c="blue")
                   mat5.show()

                   import matplotlib.pyplot as mat4
                   label1 = ['i3', 'i5', 'i7']
                   sizes = [25, 22, 24]
                   mat4.scatter(label1, sizes, c="green")
                   mat4.show()

                   import matplotlib.pyplot as mat3
                   label1 = ['4GB', '8GB', '16GB']
                   sizes = [24, 19, 28]
                   mat3.scatter(label1, sizes, c="green")
                   mat3.show()

                   import matplotlib.pyplot as mat2
                   label1 = ['Free', '1 to 20', '$21 to $40',
'$41 to $60', '$61 to $100']
                   sizes = [14, 22, 20, 12, 3]
                   mat2.scatter(label1, sizes, c="green")
                   mat2.show()

                   import matplotlib.pyplot as mat1
                   label1 = ['40 to 60', '61 to 80', '81 to
100']
                   sizes = [3, 19, 48]
                   mat1.scatter(label1, sizes, c="green")
                   mat1.show()

               print('\nSelect the graph you want to see')
               print('1.Pie chart')
               print('2.Bar graph')
               print('3.Line Graph')
               print('4.Scatter plot')
               choice = int(input('Enter choice of graphs :
'))
               if choice == 1:
                   mypieU()
                   again()
               elif choice == 2:
                   barU()
                   again()
               elif choice == 3:
                   lineU()
                   again()
               elif choice == 4:
                   scatterU()
                   again()
               else:
                   print("\nWRONG INPUT \nCHOOSE AGAIN")
                   graph()

           forgraphs1 = input("\nIF YOU WANT TO SEE GRAPHS
(ENTER Y) : ").lower()
           if forgraphs1 == "y":
               graph()

       print(pcgaming())
   rerun = input("\nDo you want run the program again \nIf yes
then type y \nElse enter anything\n: ").lower()
   print()
   if rerun == 'y':
       main()
       print()
   else:
       print("I wish you have a Good Day!!")
       exit()


def adminwindow():
   loginid = "pcgbm"
   password = "****"
   lid = input("Enter LoginID : ")
   passwd = input("Enter Password : ")
```

```
if lid == loginid and passwd == password:
    print("Access Granted!!\n")
    print("1. Add Record")
    print("2. Delete record")
    print("3. Show records")
    print("4. Update records")
    print("5. Graphs")
    print("6. Exit")
    print()
    choice = int(input("Enter choice : "))
    if choice == 1:
        adddata()
    elif choice == 2:
        deldata()
    elif choice == 3:
        fetchdata()
    elif choice == 4:
        print("What do you want to Update ?\n")
        print("1. Game name")
        print("2. Processor")
        print("3. Memory")
        print("4. Storage")
        print("5. Price")
        print("6. Review Score")
        print("7. Graphics")
        print("8. PC price")
        choice = int(input("Enter Choice : "))
        if choice == 1:
            updateG_name()
        elif choice == 2:
            updateprocessor()
        elif choice == 3:
            updatememory()
        elif choice == 4:
            updatestorage()
        elif choice == 5:
            updateprice()
        elif choice == 6:
            updateReviewScore()
        elif choice == 7:
            updategraphics()
        elif choice == 8:
            updatePCprice()
        else:
                print("wrong input")
        elif choice == 5:
            graphs()
            print()
        elif choice == 6:
            print("Exiting")
            exit()

        else:
            print("wrong input")
    else:
        print("Exiting")
        exit()


def adddata():
    import mysql.connector
    pcgbm = mysql.connector.connect(host="localhost", user="root", passwd='abc@123', database="pcgbm")
    g = str(input("game name :  "))
    p = str(input("processor from Intel core i3, Intel core i5 and Intel core i7 : "))
    m = str(input("memory from 4GB, 8GB and 16GB : "))
    gr = str(input("graphics from Geforce GTX 1050 Ti, Geforce GTX 1060 and Geforce GTX 1070 Ti : "))
    s = int(input("storage_GB_ : "))
    pr = int(input("price_$_ : "))
    pl = str(input("platform : ")).upper()
    r = str(input("ReviewScore_outof100 : "))
    pc = int(input("PCprice_Rs : "))
    cursor = pcgbm.cursor()
    cursor.execute(f'insert into pcgbm values("{g}","{p}","{m}","{gr}","{s}","{pr}","{pl}","{r}","{pc}");')
    pcgbm.commit()
    print("records added")

    r1 = input("Do you want to ADD more records : ")
    if r1 == "y":
        adddata()

    print()
    ad1 = input("Do you want to return to Admin Window : ")
    if ad1 == "y":
        adminwindow()
    else:
        exit()


def deldata():
    import mysql.connector
    pcgbm = mysql.connector.connect(host="localhost",
user="root", passwd='abc@123', database="pcgbm")
    g = str(input("game name :  "))
    cursor = pcgbm.cursor()
    cursor.execute(f'delete from pcgbm where G_name="{g}"')
    pcgbm.commit()
    print("records deleted")
    r2 = input("Do you want to DELETE more records : ")
    if r2 == "y":
        deldata()

    print()
    ad2 = input("Do you want to return to Admin Window : ")
    if ad2 == "y":
        adminwindow()
    else:
        exit()


def fetchdata():
    import mysql.connector
    from mysql.connector import Error

    try:
        pcgbm = mysql.connector.connect(host='localhost',
                                        database='pcgbm',
                                        user='root',
                                        password='abc@123')

        sql_select_query = "select * from pcgbm"
        cursor = pcgbm.cursor()
        cursor.execute(sql_select_query)
        records = cursor.fetchall()
        print("Total number of rows in database is : ",
cursor.rowcount)

        print("\nPrinting each record")
```

```
       for row in records:
           print("G_name = ", row[0], )
           print("processor = ", row[1])
           print("memory = ", row[2])
           print("graphics  = ", row[3])
           print("storage_GB_ = ", row[4])
           print("price_$_ = ", row[5])
           print("platform = ", row[6])
           print("ReviewScore_outof100 = ", row[7])
           print("PCprice_Rs = ", row[8], "\n")
   except Error as e:
       print("Error reading data from MySQL table", e)

   print()
   ad13 = input("Do you want to return to Admin Window : ")
   if ad13 == "y":
       adminwindow()
   else:
       exit()


def updateG_name():
   import mysql.connector
   pcgbm = mysql.connector.connect(host="localhost",
user="root", passwd='abc@123', database="pcgbm")
   g = str(input("Old game name :  "))
   g1 = str(input("New Game Name : "))
   cursor = pcgbm.cursor()
   cursor.execute(f'update pcgbm set G_name="{g1}" where
G_name="{g}"')
   pcgbm.commit()
   print("records updated")
   print()
   r3 = input("Do you want to UPDATE more Games name : ")
   if r3 == "y":
       updateG_name()

   print()
   ad23 = input("Do you want to return to Admin Window : ")
   if ad23 == "y":
       adminwindow()
   else:
       exit()
```

```
def updateprocessor():
    import mysql.connector
    pcgbm = mysql.connector.connect(host="localhost",
user="root", passwd='abc@123', database="pcgbm")
    g = str(input("game name :  "))
    p1 = str(input("New processor:"))
    cursor = pcgbm.cursor()
    cursor.execute(f'update pcgbm set processor="{p1}" where
G_name="{g}"')
    pcgbm.commit()
    print("records updated")

    r4 = input("Do you want to UPDATE processor of more games :
")
    if r4 == "y":
        updateprocessor()

    print()
    a1d2 = input("Do you want to return to Admin Window : ")
    if a1d2 == "y":
        adminwindow()
    else:
        exit()


def updategraphics():
    import mysql.connector
    pcgbm = mysql.connector.connect(host="localhost",
user="root", passwd='abc@123', database="pcgbm")
    g = str(input("game name :  "))
    gr1 = str(input("New graphics:"))
    cursor = pcgbm.cursor()
    cursor.execute(f'update pcgbm set graphics="{gr1}" where
G_name="{g}"')
    pcgbm.commit()
    print("records updated")

    r5 = input("Do you want to UPDATE Graphics of more Games :
")
    if r5 == "y":
        updategraphics()

    print()
    a2d2 = input("Do you want to return to Admin Window : ")
    if a2d2 == "y":
        adminwindow()
    else:
        exit()


def updatememory():
    import mysql.connector
    pcgbm = mysql.connector.connect(host="localhost",
user="root", passwd='abc@123', database="pcgbm")
    g = str(input("game name :  "))
    m1 = str(input("New memory:"))
    cursor = pcgbm.cursor()
    cursor.execute(f'update pcgbm set memory="{m1}" where
G_name="{g}"')
    pcgbm.commit()
    print("records updated")

    r6 = input("Do you want to UPDATE Memory of more Games : ")
    if r6 == "y":
        updatememory()

    print()
    a3d2 = input("Do you want to return to Admin Window : ")
    if a3d2 == "y":
        adminwindow()
    else:
        exit()


def updatestorage():
    import mysql.connector
    pcgbm = mysql.connector.connect(host="localhost",
user="root", passwd='abc@123', database="pcgbm")
    g = str(input("game name :  "))
    s1 = str(input("New storage:"))
    cursor = pcgbm.cursor()
    cursor.execute(f'update pcgbm set storage_GB_="{s1}" where
G_name="{g}"')
    pcgbm.commit()
    print("records updated")
    r7 = input("Do you want to UPDATE Storage of more Games :
")
    if r7 == "y":
        updatestorage()

    print()
    ad24 = input("Do you want to return to Admin Window : ")
    if ad24 == "y":
        adminwindow()
    else:
        exit()


def updateprice():
    import mysql.connector
    pcgbm = mysql.connector.connect(host="localhost",
user="root", passwd='abc@123', database="pcgbm")
    g = str(input("game name :  "))
    pr1 = str(input("New price:"))
    cursor = pcgbm.cursor()
    cursor.execute(f'update pcgbm set price_$_="{pr1}" where
G_name="{g}"')
    pcgbm.commit()
    print("records updated")

    r8 = input("Do you want to UPDATE Price of more Games : ")
    if r8 == "y":
        updateprice()

    print()
    ad5 = input("Do you want to return to Admin Window : ")
    if ad5 == "y":
        adminwindow()
    else:
        exit()


def updateReviewScore():
    import mysql.connector
    pcgbm = mysql.connector.connect(host="localhost",
user="root", passwd='abc@123', database="pcgbm")
    g = str(input("game name :  "))
    r1 = str(input("New ReviewScore:"))
    cursor = pcgbm.cursor()
    cursor.execute(f'update pcgbm set 
ReviewScore_outof100="{r1}" where G_name="{g}"')
    pcgbm.commit()
    print("records updated")

    r9 = input("Do you want to UPDATE ReviewScore of more Games
: ")
    if r9 == "y":
        updateReviewScore()

    print()
    ad6 = input("Do you want to return to Admin Window : ")
    if ad6 == "y":
        adminwindow()
    else:
        exit()


def updatePCprice():
    import mysql.connector
    pcgbm = mysql.connector.connect(host="localhost",
user="root", passwd='abc@123', database="pcgbm")
    g = str(input("game name :  "))
    pc1 = str(input("New PCprice:"))
    cursor = pcgbm.cursor()
    cursor.execute(f'update pcgbm set PCprice_Rs="{pc1}" where
G_name="{g}"')
    pcgbm.commit()
    print("records updated")

    r10 = input("Do you want to UPDATE PCprice of more Games :
")
    if r10 == "y":
        updatePCprice()

    print()
    ad7 = input("Do you want to return to Admin Window : ")
    if ad7 == "y":
        adminwindow()
    else:
        exit()


def graphs():
    def again10():
        again2 = input("\nDO YOU WANT TO SEE MORE GRAPHS (ENTER
Y) : ").upper()
        if again2 == 'Y':
            graphs()

        print()
        ad9 = input("Do you want to return to Admin Window : ")
        if ad9 == "y":
            adminwindow()
        else:
            exit()

    import matplotlib.pyplot as mat

    def mypie():
        label1 = ['Geforce GTX 1050 Ti', 'Geforce GTX 1060',
'Geforce GTX 1070 Ti']
        sizes = [23, 27, 21]
        colors1 = ['yellowgreen', 'lightskyblue', 'magenta']
        mat.title("Games in PCGBM for various Graphics card")
        mat.pie(sizes, explode=None, labels=label1,
colors=colors1, shadow=True, autopct='%1.1f%%',
               startangle=140)
        mat.axis('equal')
        a = mat.show()
        print(a)

        label1 = ['i3', 'i5', 'i7']
        sizes = [25, 22, 24]
        colors1 = ['red', 'blue', 'magenta']
        mat.title("Games in PCGBM for various Processors")
        mat.pie(sizes, explode=None, labels=label1,
colors=colors1, shadow=True, autopct='%1.1f%%',
               startangle=140)
        mat.axis('equal')
        b = mat.show()
        print(b)

        label1 = ['4GB', '8GB', '16GB']
        sizes = [24, 19, 28]
        colors1 = ['lightskyblue', 'yellowgreen', 'brown']
        mat.title("Games in PCGBM for various RAM")
        mat.pie(sizes, explode=None, labels=label1,
 colors=colors1, shadow=True, autopct='%1.1f%%',
                startangle=140)
        mat.axis('equal')
        c = mat.show()
        print(c)

        label1 = ['Free', '1 to 20', '$21 to $40', '$41 to
 $60', '$61 to $100']
        sizes = [14, 22, 20, 12, 3]
        colors1 = ['red', 'blue', 'magenta', 'yellowgreen',
 'brown']
        mat.title("Games in PCGBM for various prices($)\n")
        mat.pie(sizes, explode=None, labels=label1,
 colors=colors1, shadow=True, autopct='%1.1f%%',
                startangle=140)
        mat.axis('equal')
        d = mat.show()
        print(d)

        label1 = ['40 to 60', '61 to 80', '81 to 100']
        sizes = [3, 19, 48]
        colors1 = ['red', 'blue', 'cyan', ]
        mat.title("Games in PCGBM for various Review Score out
 of 100\n")
        mat.pie(sizes, explode=None, labels=label1,
 colors=colors1, shadow=True, autopct='%1.1f%%',
                startangle=140)
        mat.axis('equal')
        d = mat.show()
        print(d)

    def bar():
        import numpy as np
        import matplotlib.pyplot as mat15
        objects = ('Geforce GTX 1050 Ti', 'Geforce GTX 1060',
 'Geforce GTX 1070 Ti')
        y_pos = np.arange(len(objects))
        types = (23, 27, 21)
        mat15.bar(y_pos, types, align='center',
 color='yellowgreen')
        mat15.xticks(y_pos, objects)
        mat15.ylabel('Number Of Games')
        mat15.title('Number Of Games According to Graphics
 Card')
        mat15.show()

        import numpy as np
        import matplotlib.pyplot as mat14
        objects = ('i3', 'i5', 'i7')
        y_pos = np.arange(len(objects))
        types = (25, 22, 24)
        mat14.bar(y_pos, types, align='center',
 color='lightgreen')
        mat14.xticks(y_pos, objects)
        mat14.ylabel('Number Of Games')
        mat14.title('Number Of Games According to processor')
        mat14.show()

        import numpy as np
        import matplotlib.pyplot as mat13
        objects = ('4GB', '8GB', '16GB')
        y_pos = np.arange(len(objects))
        types = (24, 19, 28)
        mat13.bar(y_pos, types, align='center',
 color='lightskyblue')
        mat13.xticks(y_pos, objects)
        mat13.ylabel('Number Of Games')
        mat13.title('Number Of Games According to RAM')
        mat13.show()

        import numpy as np
        import matplotlib.pyplot as mat12
        objects = ('Free', '1 to 20', '$21 to $40', '$41 to
 $60', '$61 to $100')
        y_pos = np.arange(len(objects))
        types = (14, 22, 20, 12, 3)
        mat12.bar(y_pos, types, align='center', color='brown')
        mat12.xticks(y_pos, objects)
        mat12.ylabel('Number Of Games')
        mat12.title('Number Of Games According to Various
 prices($)')
        mat12.show()

        import numpy as np
        import matplotlib.pyplot as mat11
        objects = ('40 to 60', '61 to 80', '81 to 100')
```

```
        y_pos = np.arange(len(objects))
        types = (3, 19, 48)
        mat11.bar(y_pos, types, align='center',
color='magenta')
        mat11.xticks(y_pos, objects)
        mat11.ylabel('Number Of Games')
        mat11.title('Number Of Games According to Various
review score(out of 100)')
        mat11.show()

    def line():
        label1 = ['Geforce GTX 1050 Ti', 'Geforce GTX 1060',
'Geforce GTX 1070 Ti']
        sizes = [23, 27, 21]
        import matplotlib.pyplot as mat10
        mat10.plot(label1, sizes)
        mat10.title("Games in PCGBM for various Graphics card")
        mat10.xlabel('Graphics card')
        mat10.ylabel('No. Of Games')
        mat10.show()

        label1 = ['i3', 'i5', 'i7']
        sizes = [25, 22, 24]
        import matplotlib.pyplot as mat9
        mat9.plot(label1, sizes)
        mat9.title("Games in PCGBM for various Processors")
        mat9.xlabel('Processors')
        mat9.ylabel('No. Of Games')
        mat9.show()

        label1 = ['4GB', '8GB', '16GB']
        sizes = [24, 19, 28]
        import matplotlib.pyplot as mat8
        mat8.plot(label1, sizes)
        mat8.title("Games in PCGBM for various RAM")
        mat8.xlabel('Ram')
        mat8.ylabel('No. Of Games')
        mat8.show()

        label1 = ['Free', '1 to 20', '$21 to $40', '$41 to
$60', '$61 to $100']
        sizes = [14, 22, 20, 12, 3]
        import matplotlib.pyplot as mat7
        mat7.plot(label1, sizes)
        mat7.title("Games in PCGBM for various prices($)\n")
        mat7.xlabel('Prices($)')
        mat7.ylabel('No. Of Games')
        mat7.show()

        label1 = ['40 to 60', '61 to 80', '81 to 100']
        sizes = [3, 19, 48]
        import matplotlib.pyplot as mat6
        mat6.plot(label1, sizes)
        mat6.title("Games in PCGBM for various Review Score out
of 100\n")
        mat6.xlabel('Review score')
        mat6.ylabel('No. Of Games')
        mat6.show()

    def scatter():

        import matplotlib.pyplot as mat5
        label1 = ['Geforce GTX 1050 Ti', 'Geforce GTX 1060',
'Geforce GTX 1070 Ti']
        sizes = [23, 27, 21]
        mat5.scatter(label1, sizes, c="blue")
        mat5.show()

        import matplotlib.pyplot as mat4
        label1 = ['i3', 'i5', 'i7']
        sizes = [25, 22, 24]
        mat4.scatter(label1, sizes, c="green")
        mat4.show()

        import matplotlib.pyplot as mat3
        label1 = ['4GB', '8GB', '16GB']
        sizes = [24, 19, 28]
        mat3.scatter(label1, sizes, c="green")
        mat3.show()

        import matplotlib.pyplot as mat2
        label1 = ['Free', '1 to 20', '$21 to $40', '$41 to
$60', '$61 to $100']
        sizes = [14, 22, 20, 12, 3]
        mat2.scatter(label1, sizes, c="green")
        mat2.show()

        import matplotlib.pyplot as mat1
        label1 = ['40 to 60', '61 to 80', '81 to 100']
        sizes = [3, 19, 48]
        mat1.scatter(label1, sizes, c="green")
        mat1.show()

    print('\nSelect the graph you want to see')
    print('1.Pie chart')
    print('2.Bar graph')
    print('3.Line Graph')
    print('4.Scatter plot')
    choice = int(input('Enter choice of graphs : '))
    if choice == 1:
        mypie()
        again10()
    elif choice == 2:
        bar()
        again10()
    elif choice == 3:
        line()
        again10()
    elif choice == 4:
        scatter()
        again10()
    else:
        print("\nWRONG INPUT \nCHOOSE AGAIN")
        graphs()


main()
```

# CODING (MySQL)

```
create database pcgbm;
use pcgbm;
create table pcgbm(G_name varchar(50) primary key,processor
varchar(50),memory varchar(50),graphics
varchar(50),storage_GB_ int(5),price_$_ int(6),platform
varchar(50),ReviewScore_outof100 int(6),PCprice_Rs int);

insert into pcgbm values("Grand Theft Auto V (GTA 5)",
"i7","4GB","Geforce GTX
1060","72","15","WINDOWS","96","128000");
insert into pcgbm values("Destiny 2", "i7","16GB","Geforce GTX
1060","105","0","WINDOWS","83","141000");
insert into pcgbm values("Destiny 2: Beyond Light",
"i7","8GB","Geforce GTX
1060","105","70","WINDOWS","83","131000");
insert into pcgbm values("World of Warcraft", "i7","8GB","Geforce
GTX 1050 Ti","5","15","WINDOWS","93","127000");
insert into pcgbm values("Portal 2", "i7","4GB","Geforce GTX
1070 Ti","10","0","WINDOWS","95","144000");
insert into pcgbm values("Mount and Blade", "i7","8GB","Geforce
GTX 1070 Ti","16","0","WINDOWS","78","147000");
insert into pcgbm values("Monster Hunter:World",
"i7","8GB","Geforce GTX 1070
Ti","20","21","WINDOWS","88","147000");
insert into pcgbm values("Assassins Creed", "i7","8GB","Geforce
GTX 1060","50","13","WINDOWS","80","131000");
insert into pcgbm values('The red dead redemption 2', 'i7', '16GB',
'Geforce GTX 1060', '150', '30', 'WINDOWS',"94","141000");
```

```
insert into pcgbm values('League of Legends', 'i7', '16GB',
'Geforce GTX 1060', '12', '16', 'WINDOWS',"76","141000");
insert into pcgbm values('Fifa 20', 'i7', '8GB', 'Geforce GTX 1060',
'50', '21', 'WINDOWS',"20","131000");
insert into pcgbm values('CRUCIBLE', 'i7', '8GB', 'Geforce GTX
1060', '15', '16', 'WINDOWS',"60","131000");
insert into pcgbm values("STAR WARS Jedi: Fallen Order",
"i5","8GB","Geforce GTX 1070
Ti","55","35","WINDOWS","90","119000");
insert into pcgbm values('GTA IV', 'i5', '16GB', 'Geforce GTX 1050
Ti', '16', '11', 'WINDOWS',"90","109000");
insert into pcgbm values('NBA 2K20', 'i5', '8GB', 'Geforce GTX
1050 Ti', '80', '13', 'WINDOWS',"79","99000");
insert into pcgbm values("iRacing", "i5","8GB","Geforce GTX
1060","6","0","WINDOWS","79","103000");
insert into pcgbm values("DOOM Eternal", "i5","8GB","Geforce
GTX 1060","50","31","WINDOWS","90","103000");
insert into pcgbm values("The Elder Scrolls V: Skyrim",
"i5","4GB","Geforce GTX
1060","6","16","WINDOWS","94","100000");
insert into pcgbm values("Divinity", "i5","4GB","Geforce GTX
1060","16","0","WINDOWS","94","100000");
insert into pcgbm values('Valorant', 'i3', '8GB', 'Geforce GTX 1070
Ti', '120', '0', 'WINDOWS',"81","114000");
insert into pcgbm values('The Witcher 3', 'i3', '8GB', 'Geforce GTX
1070 Ti', '55', '40', 'WINDOWS',"93","114000");
insert into pcgbm values('MINECRAFT', 'i3', '8GB', 'Geforce GTX
1070 Ti', '4', '0', 'WINDOWS',"93","114000");
insert into pcgbm values('Pubg Lite', 'i3', '4GB', 'Geforce GTX
1050 Ti', '4', '0', 'WINDOWS',"86","91000");
```

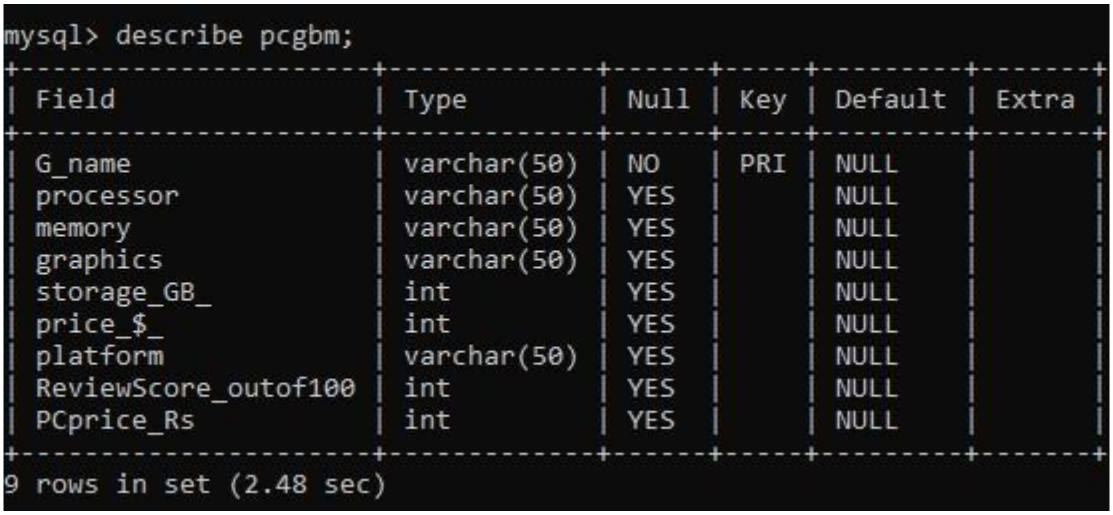
```
insert into pcgbm values('Counter Strike:Global Offensive', 'i3',
'4GB', 'Geforce GTX 1050 Ti', '15', '0', 'WINDOWS',"83","91000");
insert into pcgbm values("Call of Duty: Black Ops III",
"i3","4GB","Geforce GTX 1050
Ti","100","36","WINDOWS","73","91000");
insert into pcgbm values("Mass Effect 2", "i3","4GB","Geforce
GTX 1070 Ti","14","4","WINDOWS","94","111000");
insert into pcgbm values("Black Desert", "i3","4GB","Geforce GTX
1070 Ti","15","0","WINDOWS","73","111000");
insert into pcgbm values("FIFA 19", "i3","8GB","Geforce GTX
1050 Ti","50","40","WINDOWS","81","94000");
insert into pcgbm values("Fortnite", "i3","8GB","Geforce GTX
1050 Ti","15","0","WINDOWS","81","94000");
insert into pcgbm values('Age Of Empires II', 'i5', '4GB', 'Geforce
GTX 1050 Ti', '30', '15', 'WINDOWS',"81","96000");
insert into pcgbm values('The Borderlands 2', 'i5', '4GB', 'Geforce
GTX 1050 Ti', '20', '26', 'WINDOWS',"90","96000");
insert into pcgbm values('Divinity:Original', 'i5', '4GB', 'Geforce
GTX 1050 Ti', '10', '0', 'WINDOWS',"94","96000");
insert into pcgbm values("GTA:Vice City", "i5","4GB","Geforce
GTX 1050 Ti","10","3","WINDOWS","94","96000");
insert into pcgbm values("FINAL FANTASY XIV Online",
"i5","4GB","Geforce GTX 1050
Ti","60","15","WINDOWS","83","96000");
insert into pcgbm values("Football Manager 2020",
"i5","4GB","Geforce GTX 1050
Ti","50","30","WINDOWS","87","96000");
insert into pcgbm values('Apex', 'i7', '16GB', 'Geforce GTX 1060',
'30', '3', 'WINDOWS',"88","141000");
```

```
insert into pcgbm values('Warzone', 'i7', '16GB', 'Geforce GTX
1060', '197', '61', 'WINDOWS',"79","141000");
insert into pcgbm values('Star Citizen 2', 'i7', '16GB', 'Geforce
GTX 1060', '40', '10', 'WINDOWS',"42","141000");
insert into pcgbm values("Sid Meier’s Civilization VI",
"i7","4GB","Geforce GTX 1050
Ti","15","60","WINDOWS","88","124000");
insert into pcgbm values("Dota 2", "i7","4GB","Geforce GTX 1050
Ti","15","0","WINDOWS","90","124000");
insert into pcgbm values("Cities: Skylines", "i7","4GB","Geforce
GTX 1050 Ti","5","30","WINDOWS","85","124000");
insert into pcgbm values("Apex Legends", "i7","16GB","Geforce
GTX 1050 Ti","35","0","WINDOWS","88","137000");
insert into pcgbm values("PUBG PC", "i7","16GB","Geforce GTX
1050 Ti","35","31","WINDOWS","86","137000");
insert into pcgbm values("Call of Duty: Modern Warfare",
"i7","16GB","Geforce GTX 1050
Ti","55","40","WINDOWS","89","137000");
insert into pcgbm values("Star Citizen", "i3","16GB","Geforce GTX
1050 Ti","40","11","WINDOWS","42","104000");
insert into pcgbm values("The Sims 4", "i3","16GB","Geforce GTX
1050 Ti","20","41","WINDOWS","70","104000");
insert into pcgbm values("Tom Clancy's Rainbow Six Siege",
"i3","16GB","Geforce GTX 1050
Ti","61","20","WINDOWS","79","104000");
insert into pcgbm values("Assassin's Creed Odyssey",
"i5","16GB","Geforce GTX
1060","46","60","WINDOWS","98","113000");
```

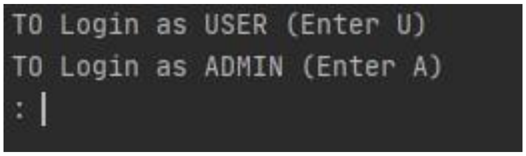
```
insert into pcgbm values("ARK: Survival Evolved",
"i5","16GB","Geforce GTX
1060","60","50","WINDOWS","71","113000");
insert into pcgbm values("Planet Zoo", "i5","16GB","Geforce GTX
1060","16","50","WINDOWS","82","113000");
insert into pcgbm values("Overwatch", "i3","4GB","Geforce GTX
1060","30","40","WINDOWS","91","95000");
insert into pcgbm values("Euro Truck Simulator 2",
"i3","4GB","Geforce GTX
1060","3","20","WINDOWS","79","95000");
insert into pcgbm values("Farming Simulator 19",
"i3","4GB","Geforce GTX
1060","23","25","WINDOWS","77","95000");
insert into pcgbm values("Anno 1800", "i3","8GB","Geforce GTX
1060","40","28","WINDOWS","81","98000");
insert into pcgbm values("Terraria", "i3","8GB","Geforce GTX
1060","2","10","WINDOWS","83","98000");
insert into pcgbm values("Age of Empires II: Definitive Edition",
"i3","8GB","Geforce GTX
1060","30","20","WINDOWS","90","98000");
insert into pcgbm values("Total War: WARHAMMER II",
"i3","16GB","Geforce GTX
1060","51","60","WINDOWS","87","108000");
insert into pcgbm values("Dead by Daylight", "i3","16GB","Geforce
GTX 1060","25","20","WINDOWS","71","108000");
insert into pcgbm values("Scrap Mechanic", "i3","16GB","Geforce
GTX 1060","15","20","WINDOWS","85","108000");
insert into pcgbm values("Rust", "i7","16GB","Geforce GTX 1070
Ti","20","40","WINDOWS","69","157000");
```

```
insert into pcgbm values("Battlefield 5", "i7","16GB","Geforce GTX
1070 Ti","50","50","WINDOWS","81","157000");
insert into pcgbm values("Metro Exodus", "i7","16GB","Geforce
GTX 1070 Ti","59","40","WINDOWS","95","157000");
insert into pcgbm values("Far Cry 3", "i5","4GB","Geforce GTX
1070 Ti","15","20","WINDOWS","88","116000");
insert into pcgbm values("Call of Duty: Black Ops II",
"i5","4GB","Geforce GTX 1070
Ti","16","60","WINDOWS","74","116000");
insert into pcgbm values("Black Desert 2", "i5","4GB","Geforce
GTX 1070 Ti","28","100","WINDOWS","73","116000");
insert into pcgbm values("MONSTER HUNTER: WORLD",
"i5","16GB","Geforce GTX 1070
Ti","20","30","WINDOWS","88","129000");
insert into pcgbm values("Assassin's Creed Unity",
"i5","16GB","Geforce GTX 1070
Ti","50","30","WINDOWS","77","129000");
insert into pcgbm values("Assassin's Creed Origins",
"i5","16GB","Geforce GTX 1070
Ti","42","60","WINDOWS","81","129000");
insert into pcgbm values("Far Cry 5", "i3","16GB","Geforce GTX
1070 Ti","40","60","WINDOWS","89","124000");
insert into pcgbm values("No Man's Sky", "i3","16GB","Geforce
GTX 1070 Ti","10","60","WINDOWS","61","124000");
insert into pcgbm values("Fallout 4", "i3","16GB","Geforce GTX
1070 Ti","30","55","WINDOWS","84","124000");
```

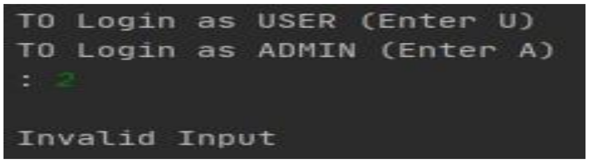
# Input and Output screen



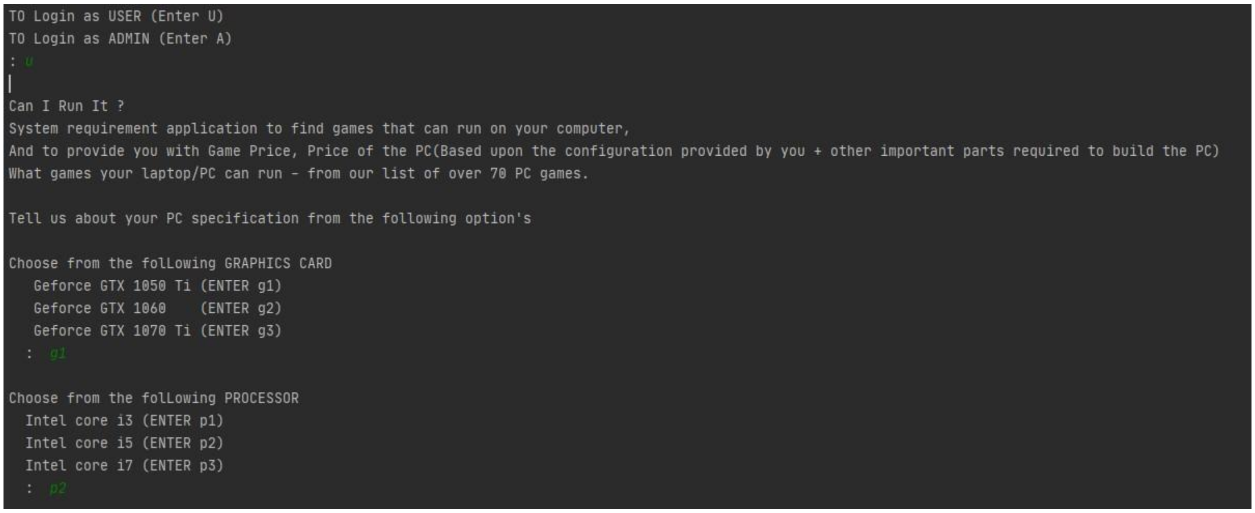
```
mysql> describe pcgbm;
+---------------------+-------------+------+-----+---------+-------+
| Field               | Type        | Null | Key | Default | Extra |
+---------------------+-------------+------+-----+---------+-------+
| G_name              | varchar(50) | NO   | PRI | NULL    |       |
| processor           | varchar(50) | YES  |     | NULL    |       |
| memory              | varchar(50) | YES  |     | NULL    |       |
| graphics            | varchar(50) | YES  |     | NULL    |       |
| storage_GB_         | int         | YES  |     | NULL    |       |
| price_$_            | int         | YES  |     | NULL    |       |
| platform            | varchar(50) | YES  |     | NULL    |       |
| ReviewScore_outof100 | int         | YES  |     | NULL    |       |
| PCprice_Rs          | int         | YES  |     | NULL    |       |
+---------------------+-------------+------+-----+---------+-------+
9 rows in set (2.48 sec)
```



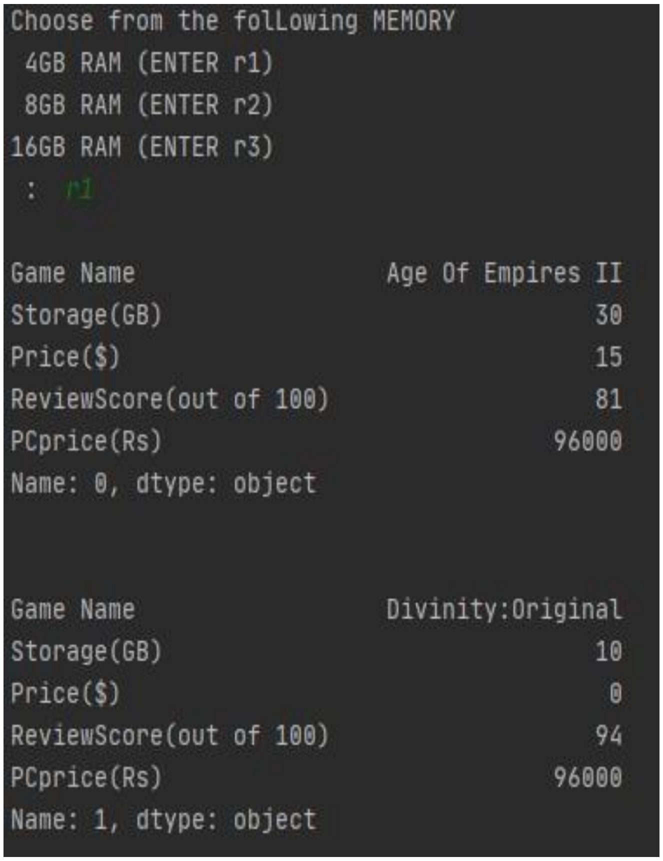
```
TO Login as USER (Enter U)
TO Login as ADMIN (Enter A)
: |
```



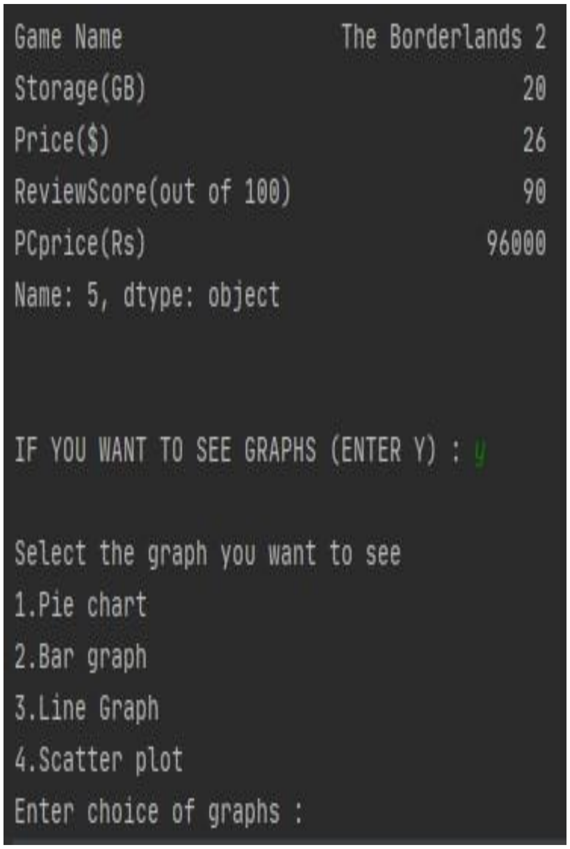
```
TO Login as USER (Enter U)
TO Login as ADMIN (Enter A)
: 2

Invalid Input
```



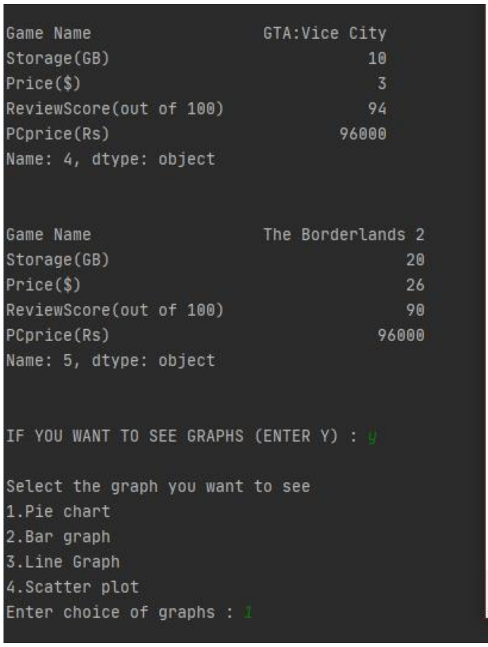
```
TO Login as USER (Enter U)
TO Login as ADMIN (Enter A)
: u
|
Can I Run It ?
System requirement application to find games that can run on your computer,
And to provide you with Game Price, Price of the PC(Based upon the configuration provided by you + other important parts required to build the PC)
What games your laptop/PC can run - from our list of over 70 PC games.

Tell us about your PC specification from the following option's

Choose from the folLowing GRAPHICS CARD
   Geforce GTX 1050 Ti (ENTER g1)
   Geforce GTX 1060    (ENTER g2)
   Geforce GTX 1070 Ti (ENTER g3)
  :  g1

Choose from the folLowing PROCESSOR
  Intel core i3 (ENTER p1)
  Intel core i5 (ENTER p2)
  Intel core i7 (ENTER p3)
  :  p2
```



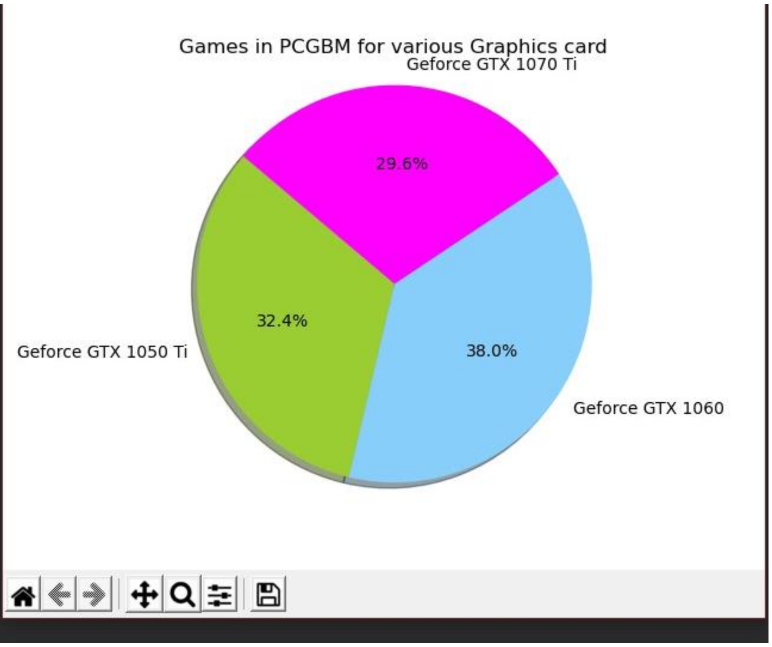
```
Choose from the folLowing MEMORY
 4GB RAM (ENTER r1)
 8GB RAM (ENTER r2)
16GB RAM (ENTER r3)
 :  r1

Game Name                 Age Of Empires II
Storage(GB)                              30
Price($)                                 15
ReviewScore(out of 100)                  81
PCprice(Rs)                           96000
Name: 0, dtype: object


Game Name                 Divinity:Original
Storage(GB)                              10
Price($)                                  0
ReviewScore(out of 100)                  94
PCprice(Rs)                           96000
Name: 1, dtype: object
```



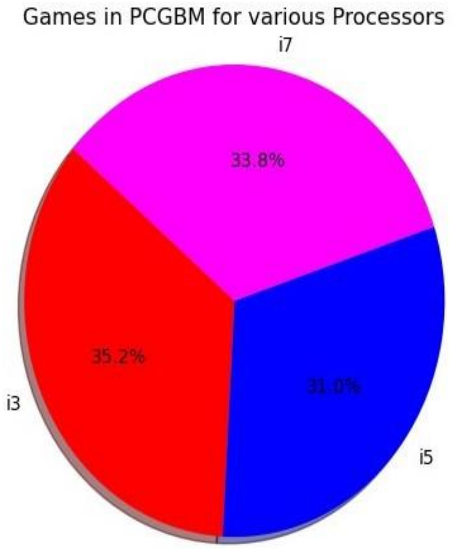
```
Game Name                 The Borderlands 2
Storage(GB)                              20
Price($)                                 26
ReviewScore(out of 100)                  90
PCprice(Rs)                           96000
Name: 5, dtype: object


IF YOU WANT TO SEE GRAPHS (ENTER Y) : y

Select the graph you want to see
1.Pie chart
2.Bar graph
3.Line Graph
4.Scatter plot
Enter choice of graphs :
```

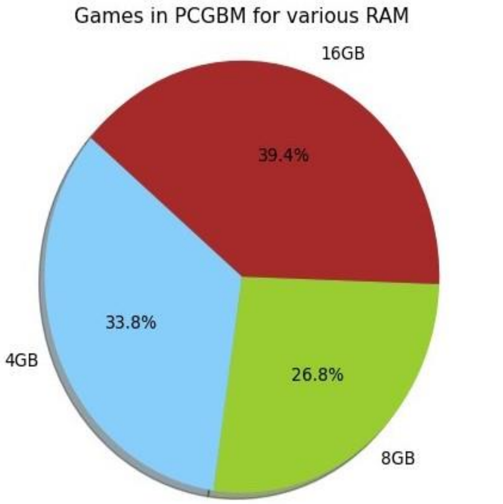


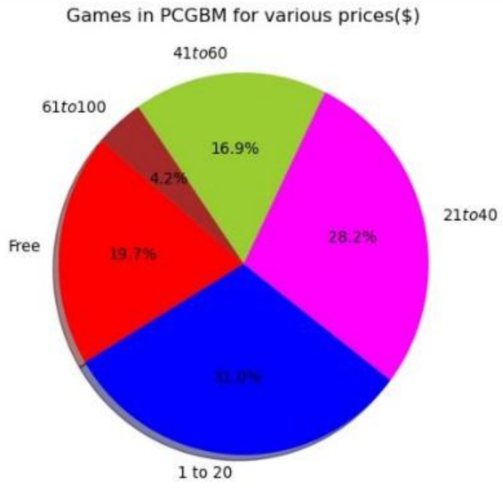
```
Game Name                      GTA:Vice City
Storage(GB)                               10
Price($)                                   3
ReviewScore(out of 100)                   94
PCprice(Rs)                            96000
Name: 4, dtype: object


Game Name                      The Borderlands 2
Storage(GB)                                   20
Price($)                                      26
ReviewScore(out of 100)                       90
PCprice(Rs)                                96000
Name: 5, dtype: object


IF YOU WANT TO SEE GRAPHS (ENTER Y) : y

Select the graph you want to see
1.Pie chart
2.Bar graph
3.Line Graph
4.Scatter plot
Enter choice of graphs : 1
```

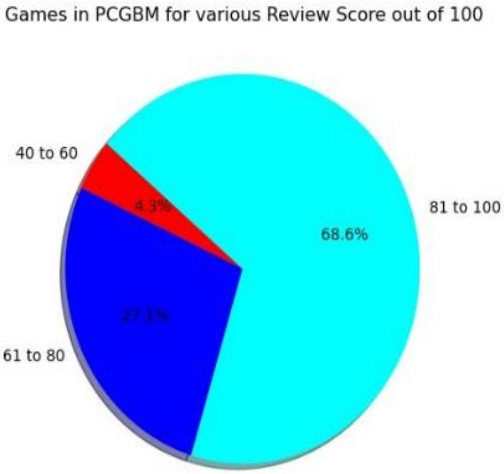


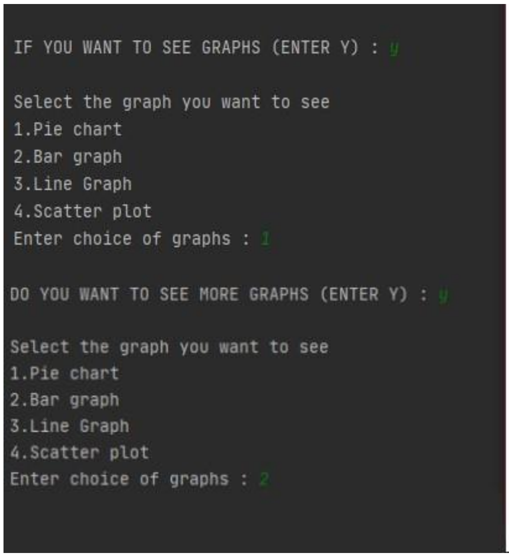
Games in PCGBM for various Graphics card



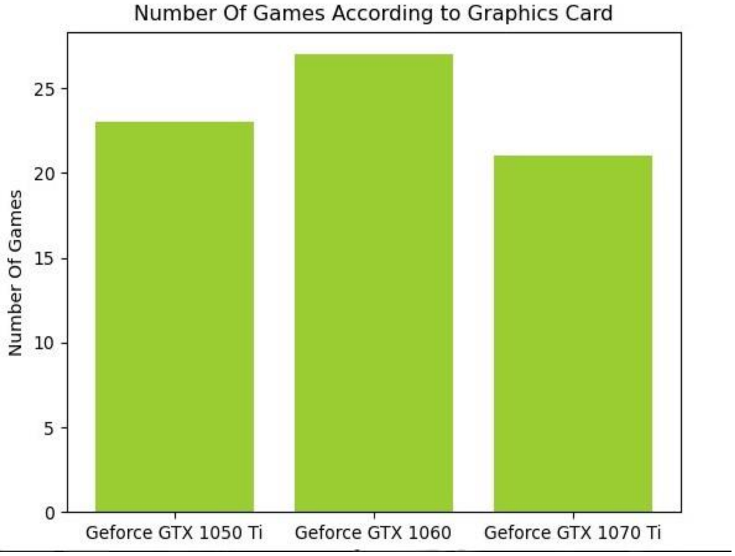


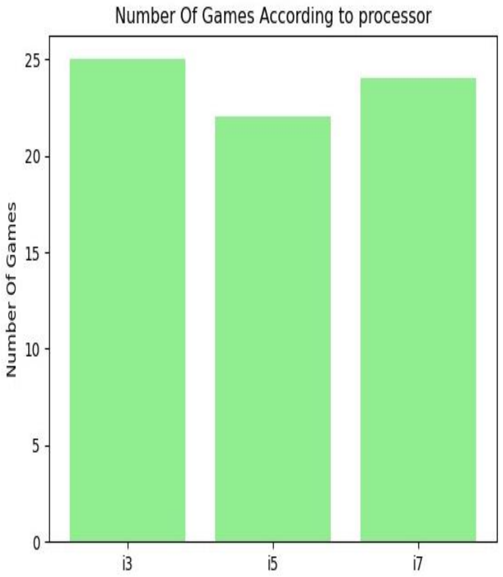


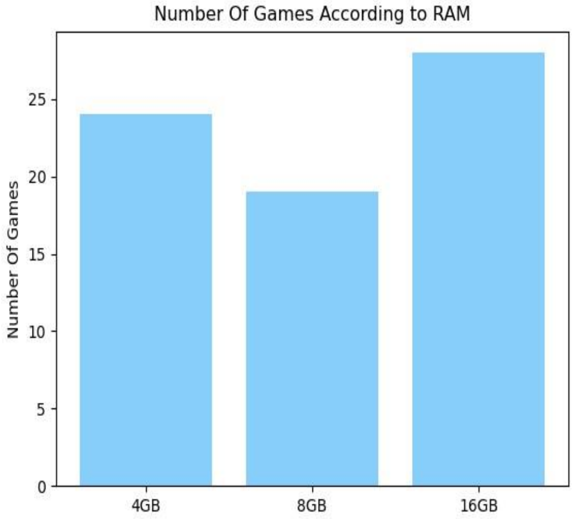


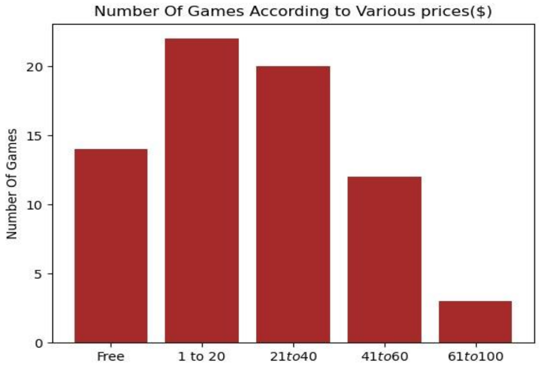


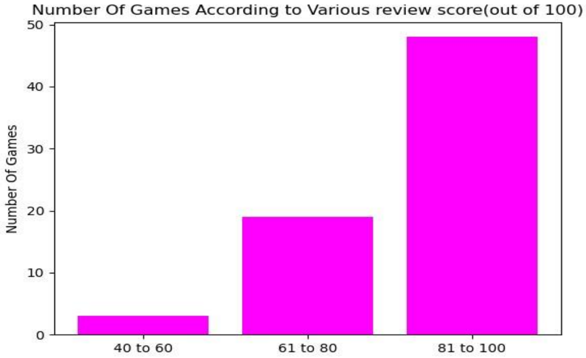
```
IF YOU WANT TO SEE GRAPHS (ENTER Y) : y

Select the graph you want to see
1.Pie chart
2.Bar graph
3.Line Graph
4.Scatter plot
Enter choice of graphs : 1

DO YOU WANT TO SEE MORE GRAPHS (ENTER Y) : y

Select the graph you want to see
1.Pie chart
2.Bar graph
3.Line Graph
4.Scatter plot
Enter choice of graphs : 2
```

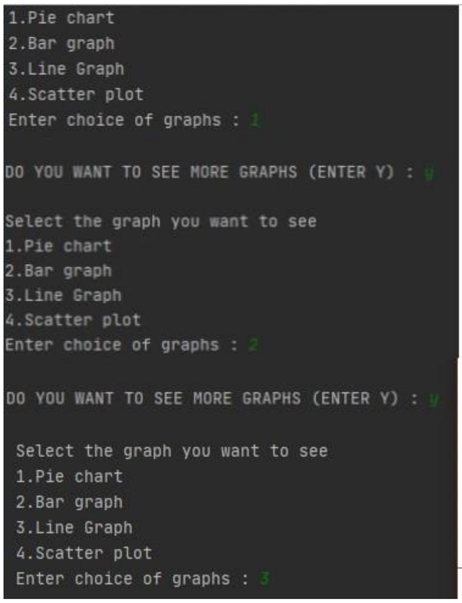




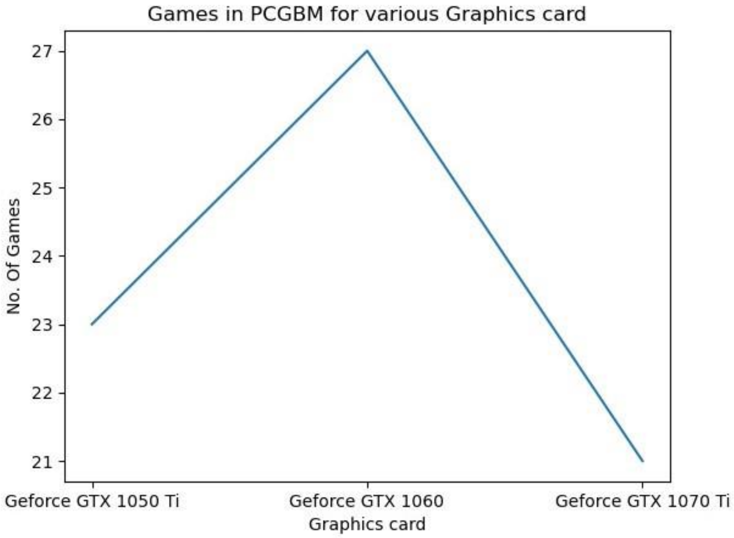


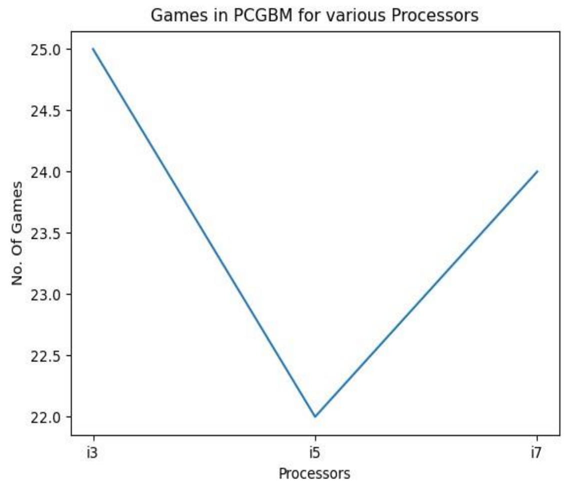


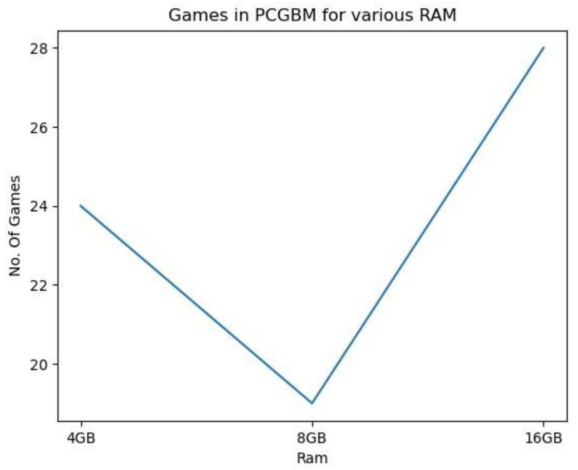


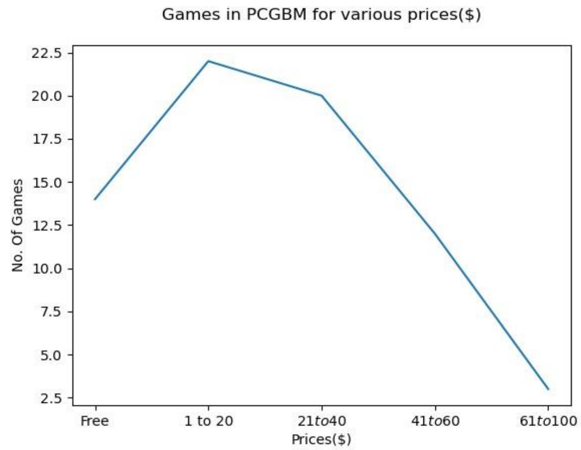


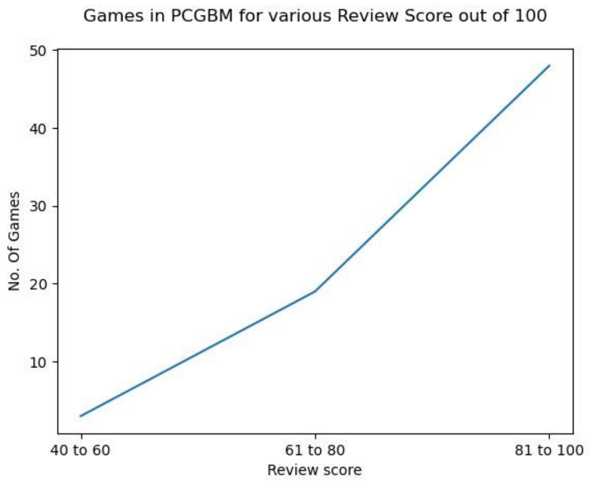
```
1.Pie chart
2.Bar graph
3.Line Graph
4.Scatter plot
Enter choice of graphs : 1

DO YOU WANT TO SEE MORE GRAPHS (ENTER Y) : y

Select the graph you want to see
1.Pie chart
2.Bar graph
3.Line Graph
4.Scatter plot
Enter choice of graphs : 2

DO YOU WANT TO SEE MORE GRAPHS (ENTER Y) : y

 Select the graph you want to see
 1.Pie chart
 2.Bar graph
 3.Line Graph
 4.Scatter plot
 Enter choice of graphs : 3
```

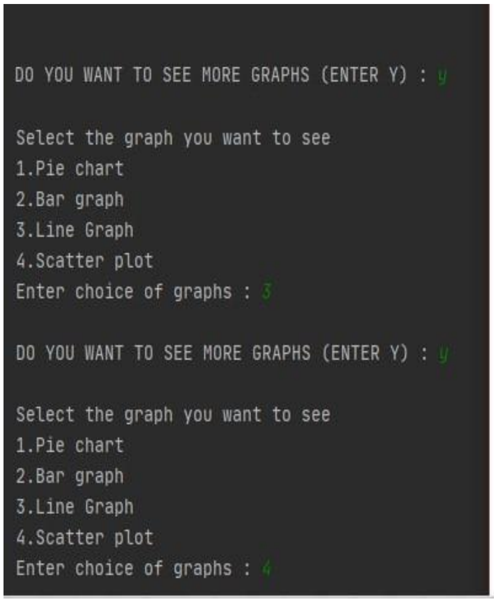




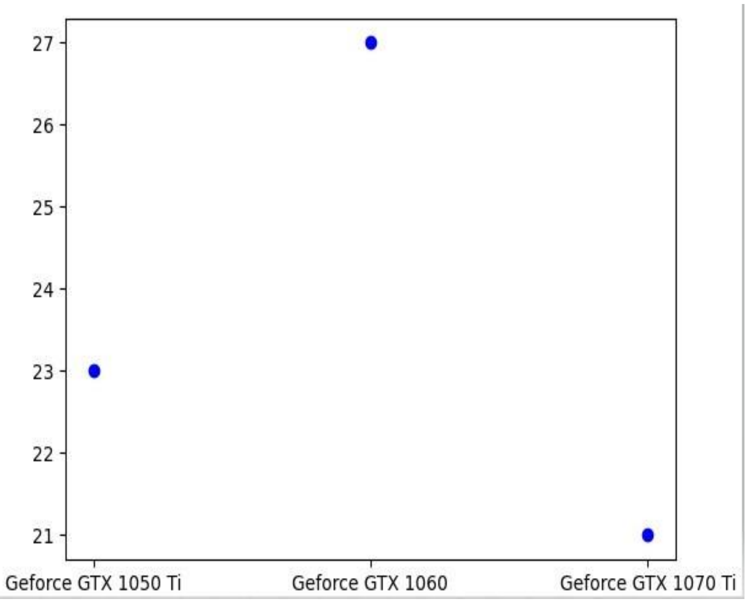


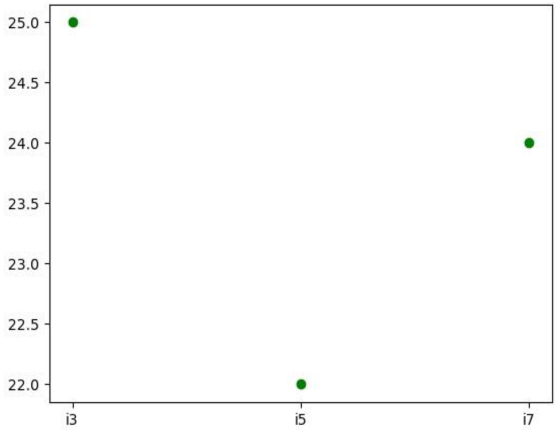


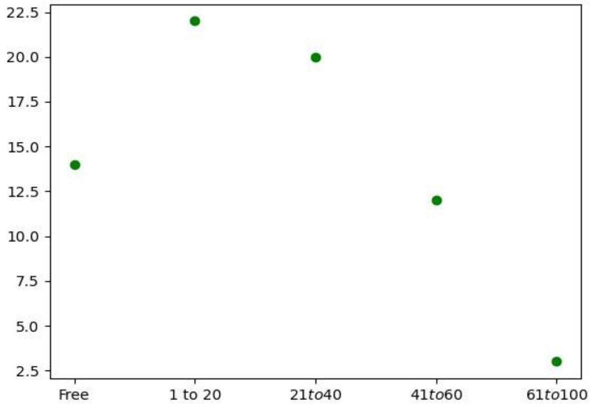


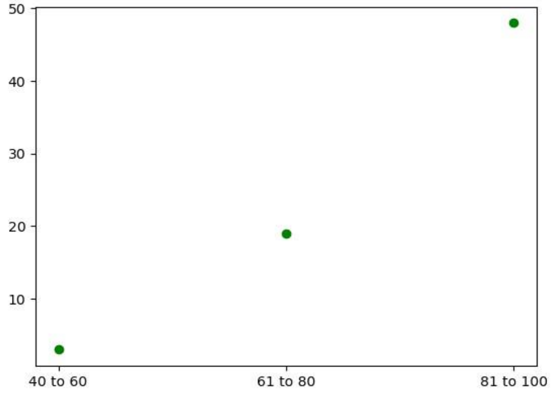


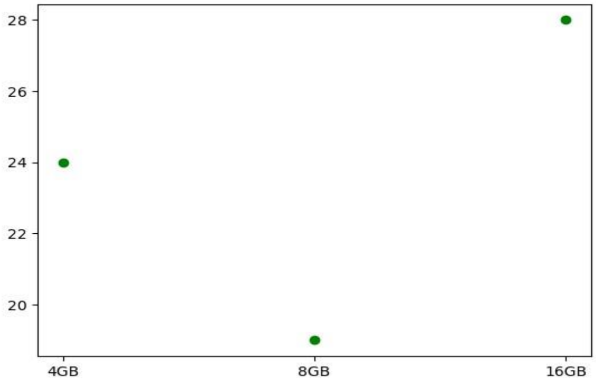
```
DO YOU WANT TO SEE MORE GRAPHS (ENTER Y) : y

Select the graph you want to see
1.Pie chart
2.Bar graph
3.Line Graph
4.Scatter plot
Enter choice of graphs : 3

DO YOU WANT TO SEE MORE GRAPHS (ENTER Y) : y

Select the graph you want to see
1.Pie chart
2.Bar graph
3.Line Graph
4.Scatter plot
Enter choice of graphs : 4
```

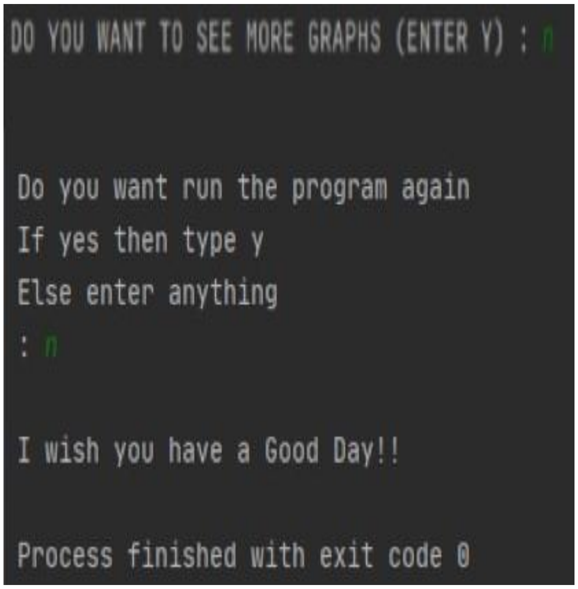




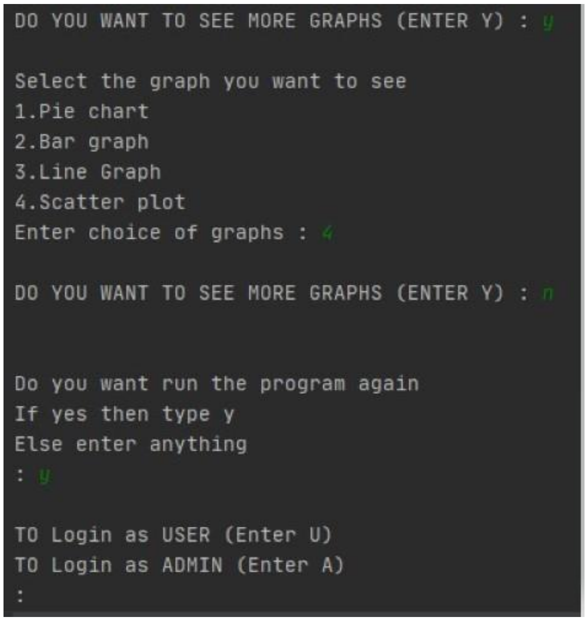




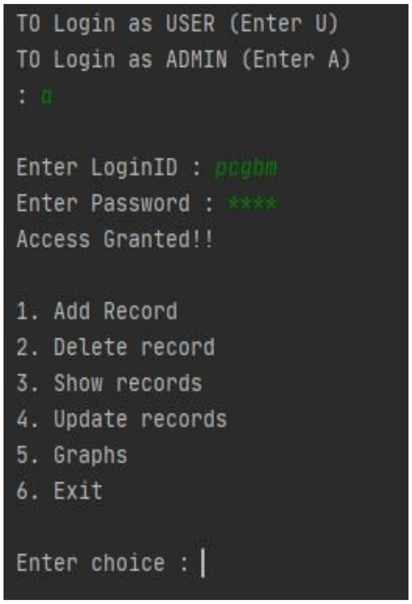




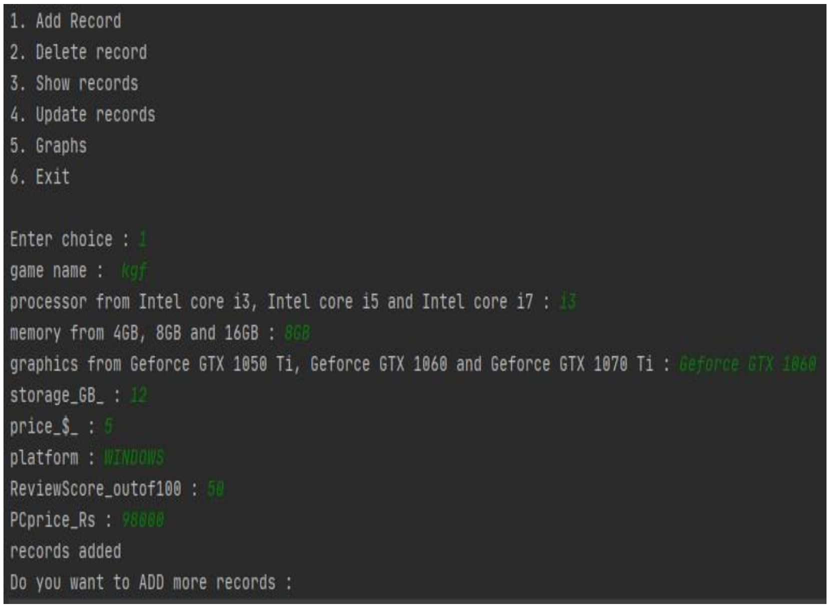
```
DO YOU WANT TO SEE MORE GRAPHS (ENTER Y) : n


Do you want run the program again
If yes then type y
Else enter anything
: n

I wish you have a Good Day!!

Process finished with exit code 0
```



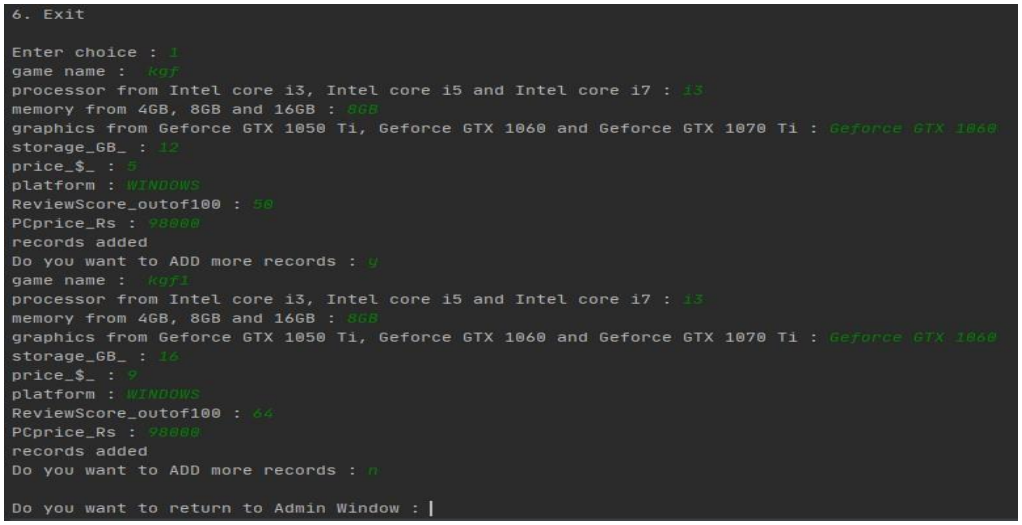
```
DO YOU WANT TO SEE MORE GRAPHS (ENTER Y) : y

Select the graph you want to see
1.Pie chart
2.Bar graph
3.Line Graph
4.Scatter plot
Enter choice of graphs : 4

DO YOU WANT TO SEE MORE GRAPHS (ENTER Y) : n


Do you want run the program again
If yes then type y
Else enter anything
: y

TO Login as USER (Enter U)
TO Login as ADMIN (Enter A)
:
```



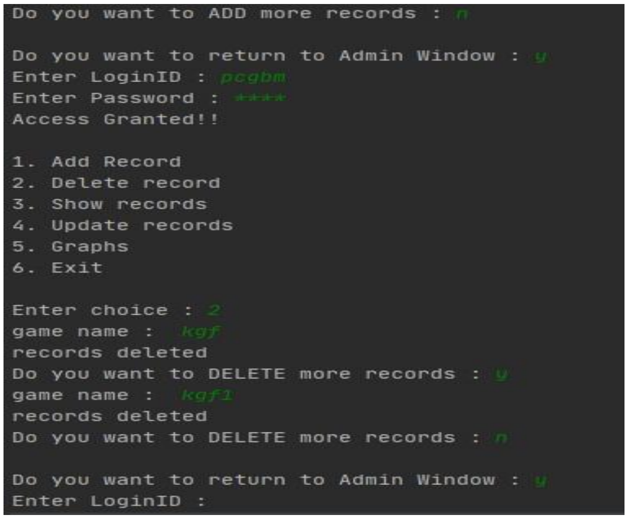
```
TO Login as USER (Enter U)
TO Login as ADMIN (Enter A)
: a

Enter LoginID : pcgbm
Enter Password : ****
Access Granted!!

1. Add Record
2. Delete record
3. Show records
4. Update records
5. Graphs
6. Exit

Enter choice : |
```



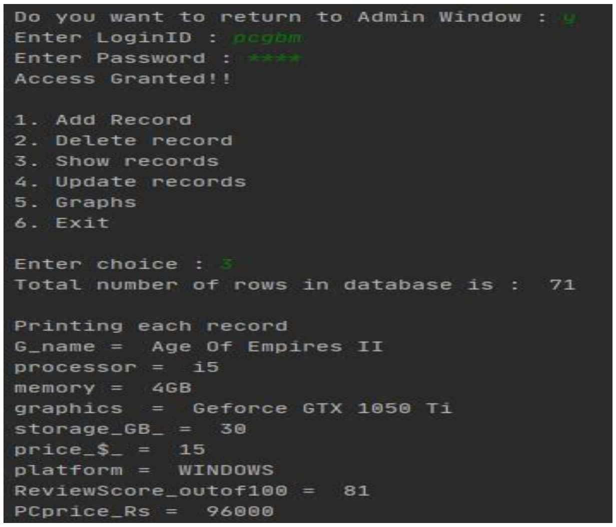
```
1. Add Record
2. Delete record
3. Show records
4. Update records
5. Graphs
6. Exit

Enter choice : 1
game name :  kgf
processor from Intel core i3, Intel core i5 and Intel core i7 : i3
memory from 4GB, 8GB and 16GB : 8GB
graphics from Geforce GTX 1050 Ti, Geforce GTX 1060 and Geforce GTX 1070 Ti : Geforce GTX 1060
storage_GB_ : 12
price_$_ : 5
platform : WINDOWS
ReviewScore_outof100 : 50
PCprice_Rs : 98000
records added
Do you want to ADD more records :
```



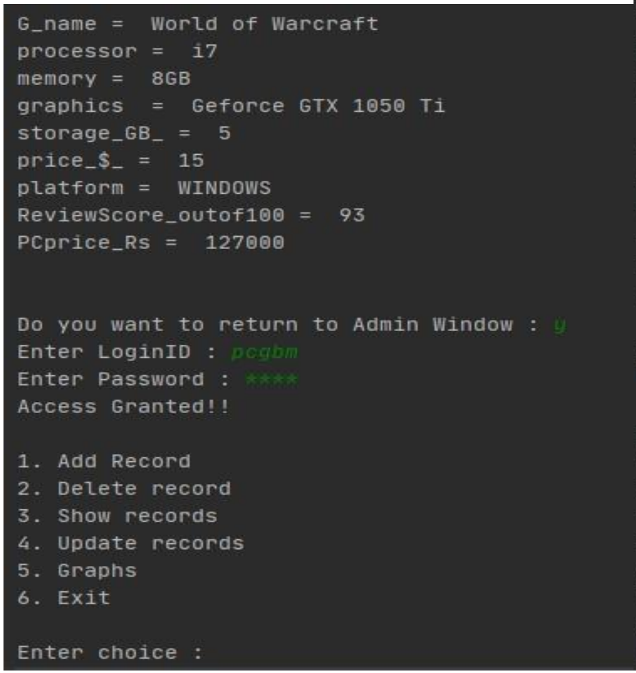
```
6. Exit

Enter choice : 1
game name :  kgf
processor from Intel core i3, Intel core i5 and Intel core i7 : i3
memory from 4GB, 8GB and 16GB : 8GB
graphics from Geforce GTX 1050 Ti, Geforce GTX 1060 and Geforce GTX 1070 Ti : Geforce GTX 1060
storage_GB_ : 12
price_$_ : 5
platform : WINDOWS
ReviewScore_outof100 : 50
PCprice_Rs : 98000
records added
Do you want to ADD more records : y
game name :  kgf1
processor from Intel core i3, Intel core i5 and Intel core i7 : i3
memory from 4GB, 8GB and 16GB : 8GB
graphics from Geforce GTX 1050 Ti, Geforce GTX 1060 and Geforce GTX 1070 Ti : Geforce GTX 1060
storage_GB_ : 16
price_$_ : 9
platform : WINDOWS
ReviewScore_outof100 : 64
PCprice_Rs : 98000
records added
Do you want to ADD more records : n

Do you want to return to Admin Window : |
```



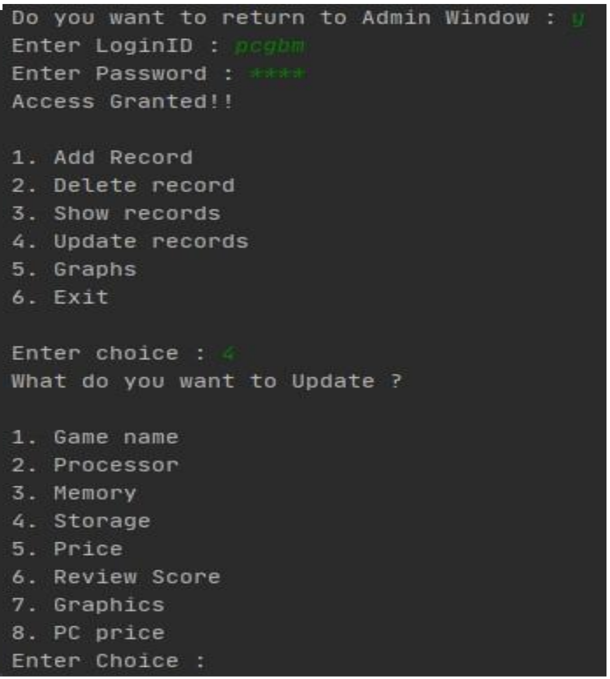
```
Do you want to ADD more records : n

Do you want to return to Admin Window : y
Enter LoginID : pcgbm
Enter Password : ****
Access Granted!!

1. Add Record
2. Delete record
3. Show records
4. Update records
5. Graphs
6. Exit

Enter choice : 2
game name :  kgf
records deleted
Do you want to DELETE more records : y
game name :  kgf1
records deleted
Do you want to DELETE more records : n

Do you want to return to Admin Window : y
Enter LoginID :
```



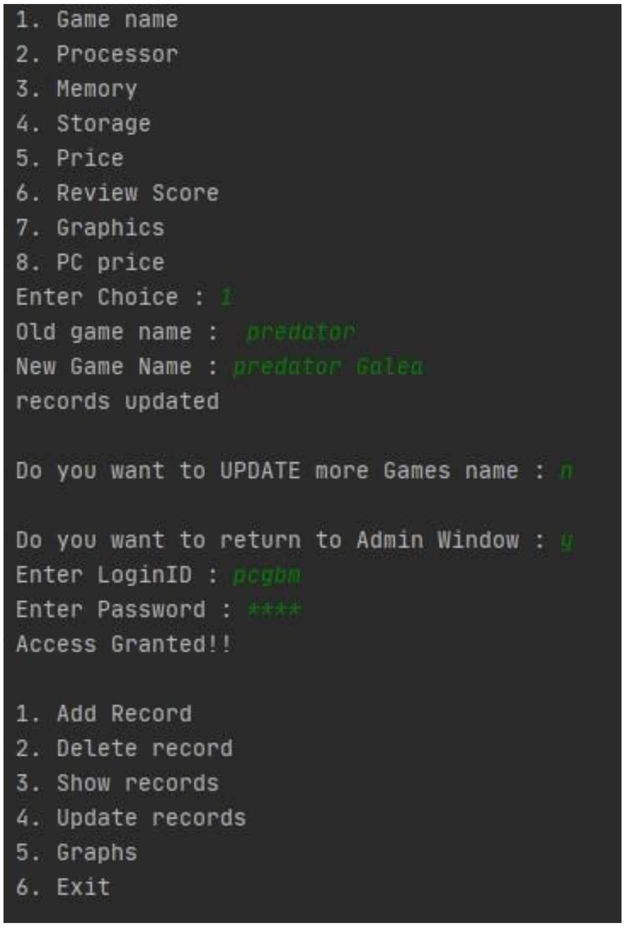
```
Do you want to return to Admin Window : y
Enter LoginID : pcgbm
Enter Password : ****
Access Granted!!

1. Add Record
2. Delete record
3. Show records
4. Update records
5. Graphs
6. Exit

Enter choice : 3
Total number of rows in database is :  71

Printing each record
G_name =  Age Of Empires II
processor =  i5
memory =  4GB
graphics  =  Geforce GTX 1050 Ti
storage_GB_ =  30
price_$_ =  15
platform =  WINDOWS
ReviewScore_outof100 =  81
PCprice_Rs =  96000
```



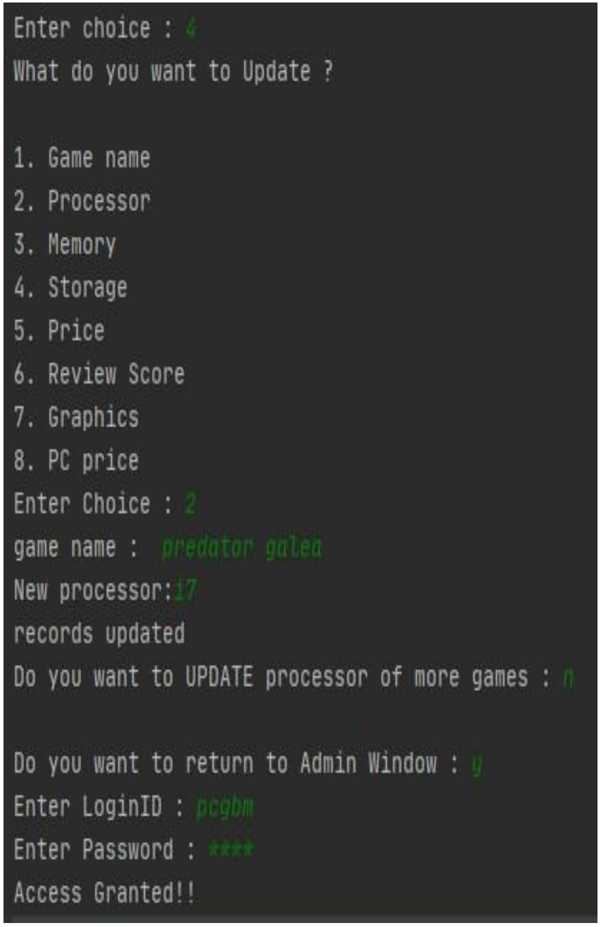
```
G_name =  World of Warcraft
processor =  i7
memory =  8GB
graphics  =  Geforce GTX 1050 Ti
storage_GB_ =  5
price_$_ =  15
platform =  WINDOWS
ReviewScore_outof100 =  93
PCprice_Rs =  127000


Do you want to return to Admin Window : y
Enter LoginID : pcgbm
Enter Password : ****
Access Granted!!

1. Add Record
2. Delete record
3. Show records
4. Update records
5. Graphs
6. Exit

Enter choice :
```



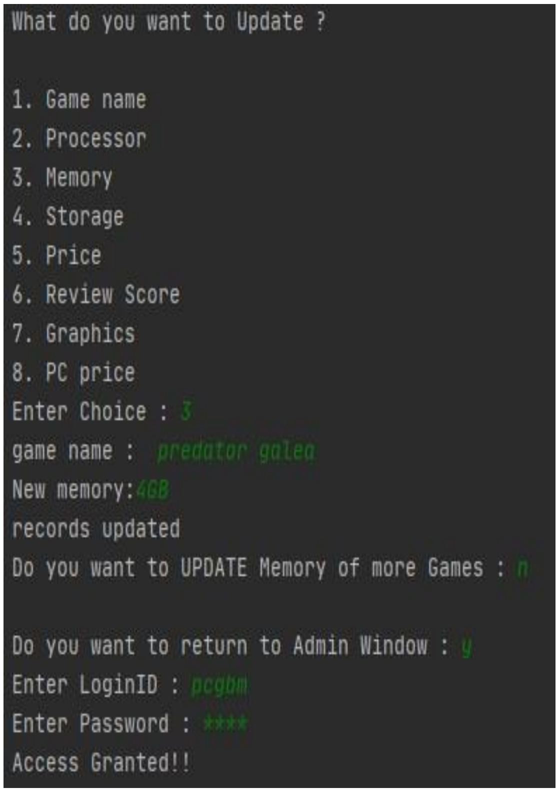
```
Do you want to return to Admin Window : y
Enter LoginID : pcgbm
Enter Password : ****
Access Granted!!

1. Add Record
2. Delete record
3. Show records
4. Update records
5. Graphs
6. Exit

Enter choice : 4
What do you want to Update ?

1. Game name
2. Processor
3. Memory
4. Storage
5. Price
6. Review Score
7. Graphics
8. PC price
Enter Choice :
```



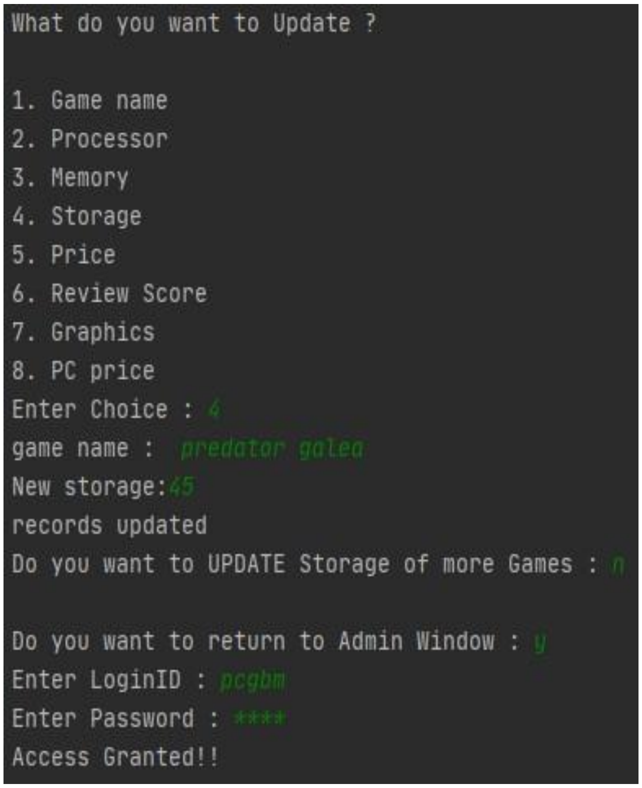
```
1. Game name
2. Processor
3. Memory
4. Storage
5. Price
6. Review Score
7. Graphics
8. PC price
Enter Choice : 1
Old game name :  predator
New Game Name : predator Galea
records updated

Do you want to UPDATE more Games name : n

Do you want to return to Admin Window : y
Enter LoginID : pcgbm
Enter Password : ****
Access Granted!!

1. Add Record
2. Delete record
3. Show records
4. Update records
5. Graphs
6. Exit
```



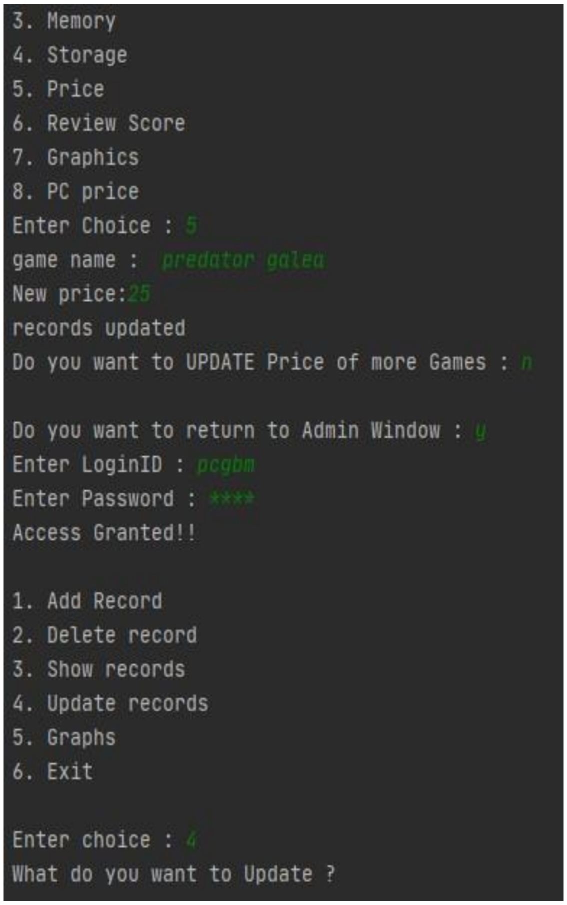
```
Enter choice : 4
What do you want to Update ?


1. Game name
2. Processor
3. Memory
4. Storage
5. Price
6. Review Score
7. Graphics
8. PC price
Enter Choice : 2
game name :  predator galea
New processor:i7
records updated
Do you want to UPDATE processor of more games : n

Do you want to return to Admin Window : y
Enter LoginID : pcgbm
Enter Password : ****
Access Granted!!
```



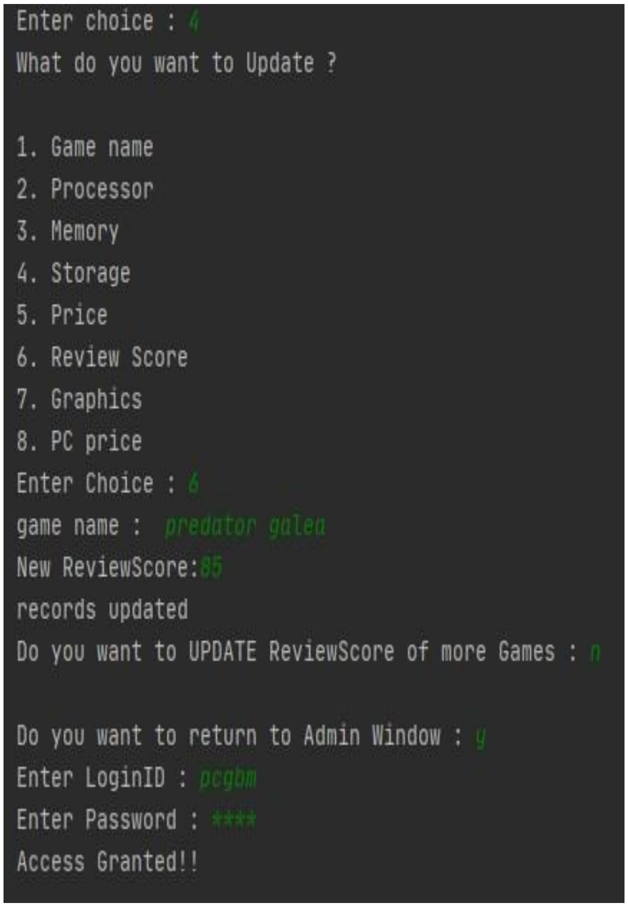
```
What do you want to Update ?

1. Game name
2. Processor
3. Memory
4. Storage
5. Price
6. Review Score
7. Graphics
8. PC price
Enter Choice : 3
game name :  predator galea
New memory:4GB
records updated
Do you want to UPDATE Memory of more Games : n

Do you want to return to Admin Window : y
Enter LoginID : pcgbm
Enter Password : ****
Access Granted!!
```



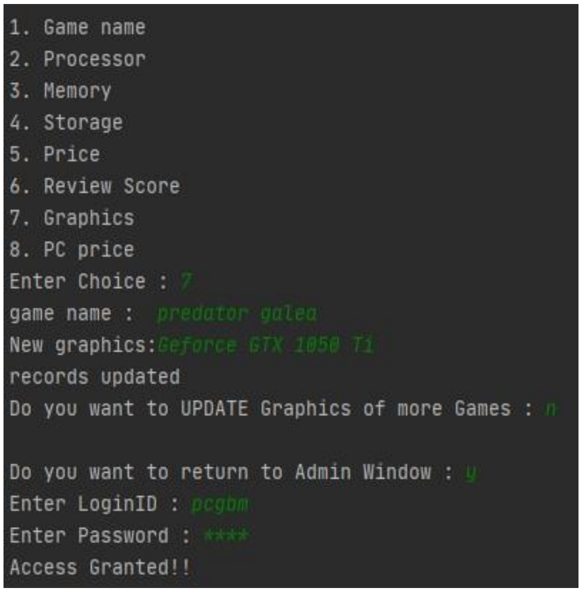
```
What do you want to Update ?

1. Game name
2. Processor
3. Memory
4. Storage
5. Price
6. Review Score
7. Graphics
8. PC price
Enter Choice : 4
game name :  predator galea
New storage:45
records updated
Do you want to UPDATE Storage of more Games : n

Do you want to return to Admin Window : y
Enter LoginID : pcgbm
Enter Password : ****
Access Granted!!
```



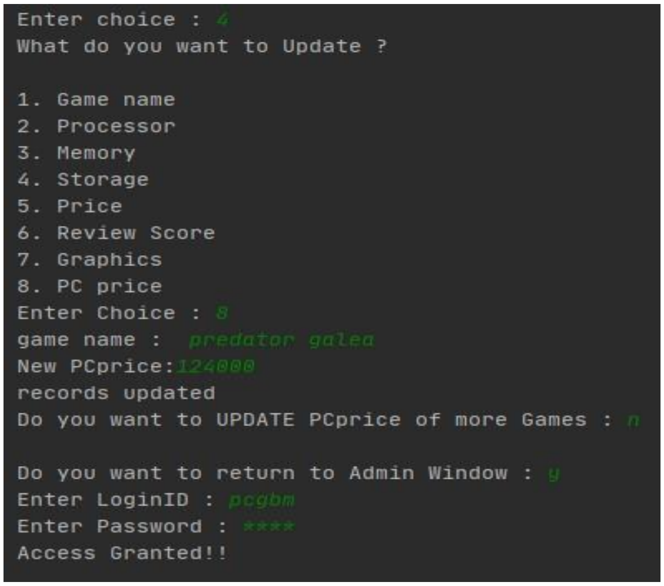
```
3. Memory
4. Storage
5. Price
6. Review Score
7. Graphics
8. PC price
Enter Choice : 5
game name :  predator galea
New price:25
records updated
Do you want to UPDATE Price of more Games : n

Do you want to return to Admin Window : y
Enter LoginID : pcgbm
Enter Password : ****
Access Granted!!

1. Add Record
2. Delete record
3. Show records
4. Update records
5. Graphs
6. Exit

Enter choice : 4
What do you want to Update ?
```



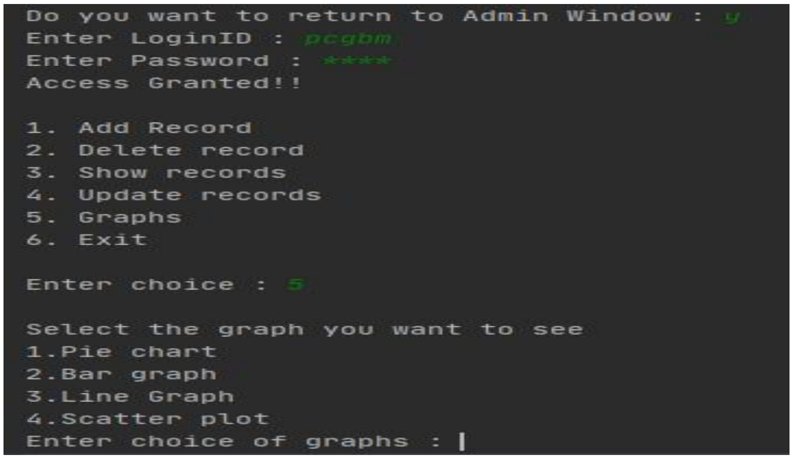
```
Enter choice : 4
What do you want to Update ?

1. Game name
2. Processor
3. Memory
4. Storage
5. Price
6. Review Score
7. Graphics
8. PC price
Enter Choice : 6
game name :  predator galea
New ReviewScore:85
records updated
Do you want to UPDATE ReviewScore of more Games : n

Do you want to return to Admin Window : y
Enter LoginID : pcgbm
Enter Password : ****
Access Granted!!
```



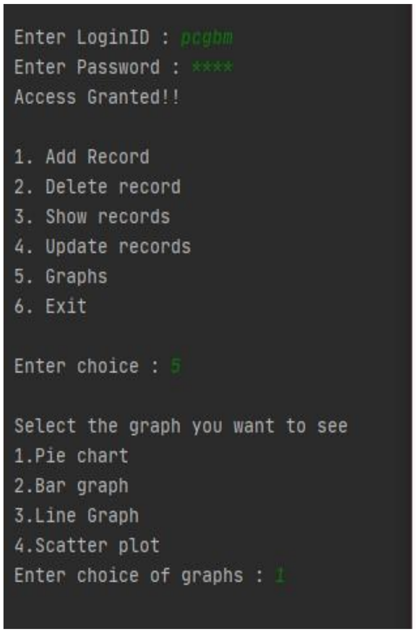
```
1. Game name
2. Processor
3. Memory
4. Storage
5. Price
6. Review Score
7. Graphics
8. PC price
Enter Choice : 7
game name :  predator galea
New graphics:Geforce GTX 1050 Ti
records updated
Do you want to UPDATE Graphics of more Games : n

Do you want to return to Admin Window : y
Enter LoginID : pcgbm
Enter Password : ****
Access Granted!!
```



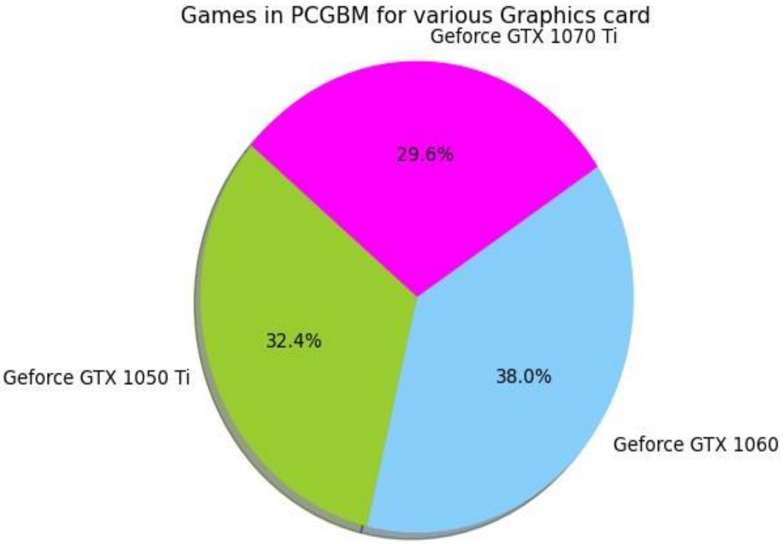
```
Enter choice : 4
What do you want to Update ?

1. Game name
2. Processor
3. Memory
4. Storage
5. Price
6. Review Score
7. Graphics
8. PC price
Enter Choice : 8
game name :  predator galea
New PCprice:124000
records updated
Do you want to UPDATE PCprice of more Games : n

Do you want to return to Admin Window : y
Enter LoginID : pcgbm
Enter Password : ****
Access Granted!!
```



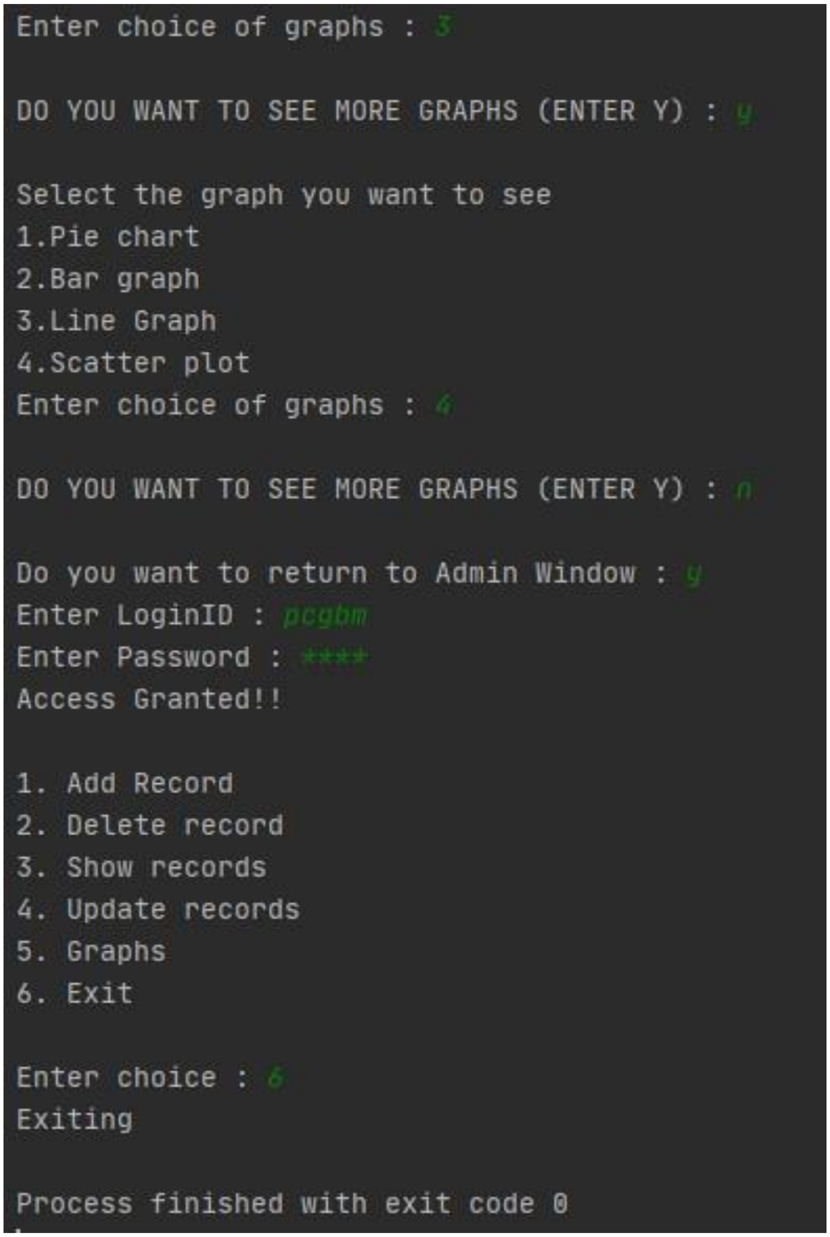
```
Do you want to return to Admin Window : y
Enter LoginID : pcgbm
Enter Password : ****
Access Granted!!

1. Add Record
2. Delete record
3. Show records
4. Update records
5. Graphs
6. Exit

Enter choice : 5

Select the graph you want to see
1.Pie chart
2.Bar graph
3.Line Graph
4.Scatter plot
Enter choice of graphs : |
```



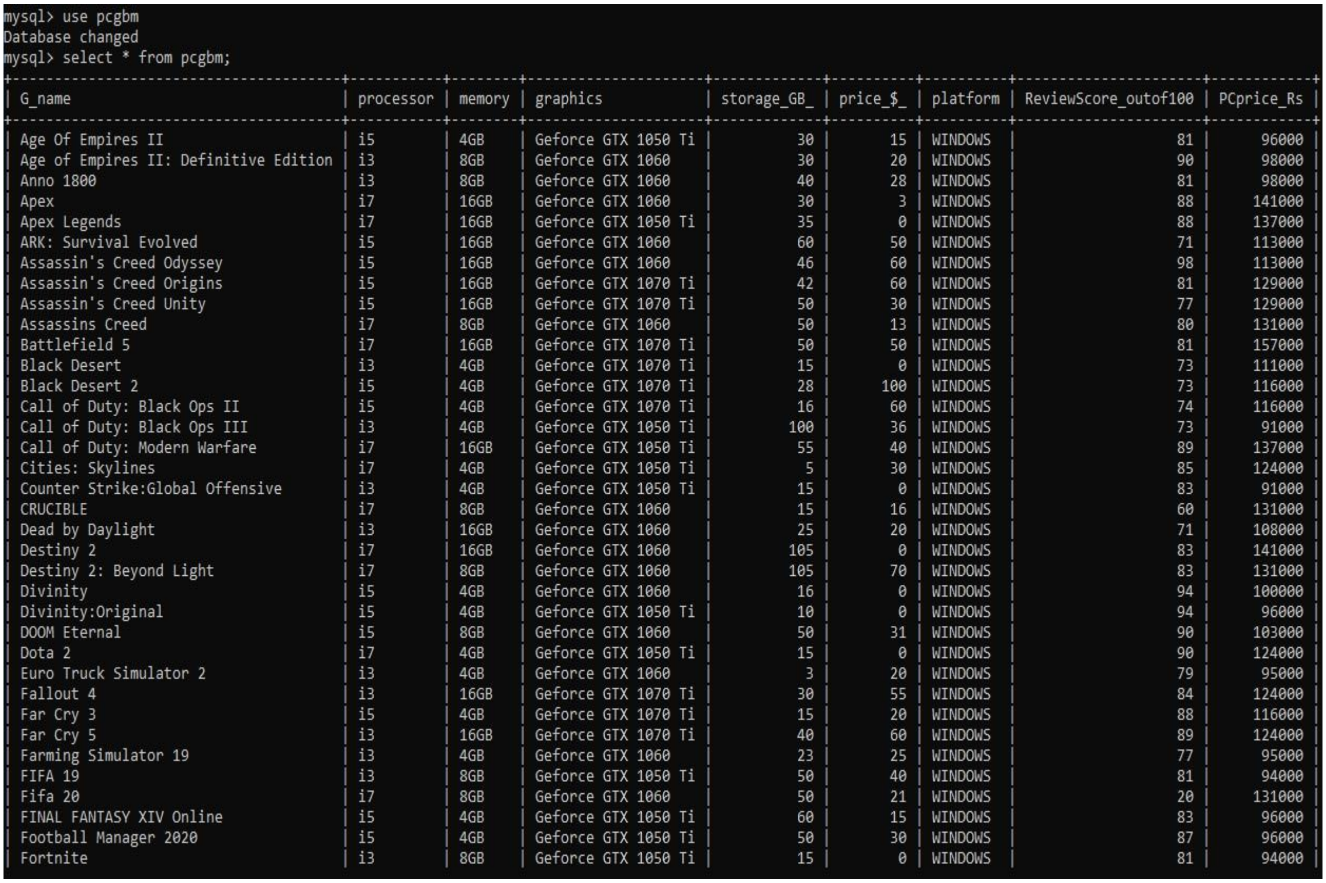
```
Enter LoginID : pcgbm
Enter Password : ****
Access Granted!!

1. Add Record
2. Delete record
3. Show records
4. Update records
5. Graphs
6. Exit

Enter choice : 5

Select the graph you want to see
1.Pie chart
2.Bar graph
3.Line Graph
4.Scatter plot
Enter choice of graphs : 1
```



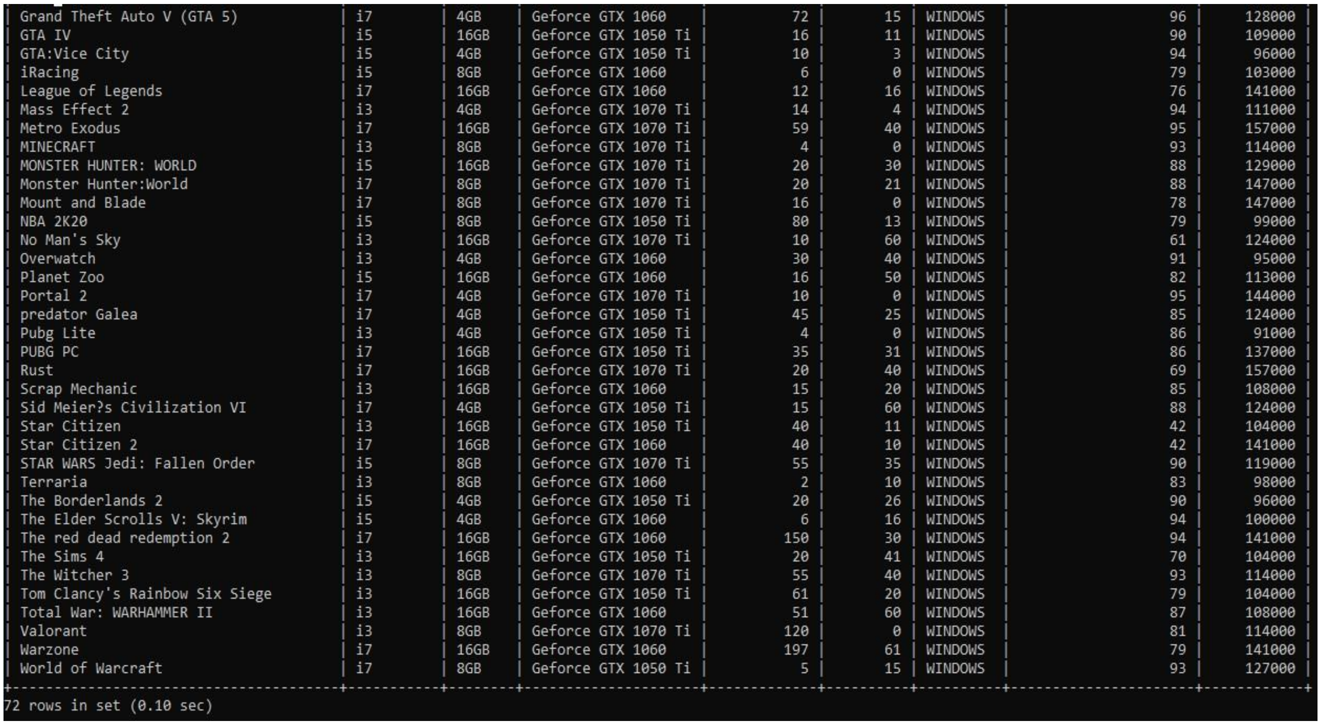
**Similarly, the program will show rest of the different types of graphs**



```
Enter choice of graphs : 3

DO YOU WANT TO SEE MORE GRAPHS (ENTER Y) : y

Select the graph you want to see
1.Pie chart
2.Bar graph
3.Line Graph
4.Scatter plot
Enter choice of graphs : 4

DO YOU WANT TO SEE MORE GRAPHS (ENTER Y) : n

Do you want to return to Admin Window : y
Enter LoginID : pcgbm
Enter Password : ****
Access Granted!!

1. Add Record
2. Delete record
3. Show records
4. Update records
5. Graphs
6. Exit

Enter choice : 6
Exiting

Process finished with exit code 0
```



```
mysql> use pcgbm
Database changed
mysql> select * from pcgbm;
+---------------------------------------+-----------+--------+---------------------+-------------+----------+----------+----------------------+------------+
| G_name                                | processor | memory | graphics            | storage_GB_ | price_$_ | platform | ReviewScore_outof100 | PCprice_Rs |
+---------------------------------------+-----------+--------+---------------------+-------------+----------+----------+----------------------+------------+
| Age Of Empires II                     | i5        | 4GB    | Geforce GTX 1050 Ti |          30 |       15 | WINDOWS  |                   81 |      96000 |
| Age of Empires II: Definitive Edition | i3        | 8GB    | Geforce GTX 1060    |          30 |       20 | WINDOWS  |                   90 |      98000 |
| Anno 1800                             | i3        | 8GB    | Geforce GTX 1060    |          40 |       28 | WINDOWS  |                   81 |      98000 |
| Apex                                  | i7        | 16GB   | Geforce GTX 1060    |          30 |        3 | WINDOWS  |                   88 |     141000 |
| Apex Legends                          | i7        | 16GB   | Geforce GTX 1050 Ti |          35 |        0 | WINDOWS  |                   88 |     137000 |
| ARK: Survival Evolved                 | i5        | 16GB   | Geforce GTX 1060    |          60 |       50 | WINDOWS  |                   71 |     113000 |
| Assassin's Creed Odyssey              | i5        | 16GB   | Geforce GTX 1060    |          46 |       60 | WINDOWS  |                   98 |     113000 |
| Assassin's Creed Origins              | i5        | 16GB   | Geforce GTX 1070 Ti |          42 |       60 | WINDOWS  |                   81 |     129000 |
| Assassin's Creed Unity                | i5        | 16GB   | Geforce GTX 1070 Ti |          50 |       30 | WINDOWS  |                   77 |     129000 |
| Assassins Creed                       | i7        | 8GB    | Geforce GTX 1060    |          50 |       13 | WINDOWS  |                   80 |     131000 |
| Battlefield 5                         | i7        | 16GB   | Geforce GTX 1070 Ti |          50 |       50 | WINDOWS  |                   81 |     157000 |
| Black Desert                          | i3        | 4GB    | Geforce GTX 1070 Ti |          15 |        0 | WINDOWS  |                   73 |     111000 |
| Black Desert 2                        | i5        | 4GB    | Geforce GTX 1070 Ti |          28 |      100 | WINDOWS  |                   73 |     116000 |
| Call of Duty: Black Ops II            | i5        | 4GB    | Geforce GTX 1070 Ti |          16 |       60 | WINDOWS  |                   74 |     116000 |
| Call of Duty: Black Ops III           | i3        | 4GB    | Geforce GTX 1050 Ti |         100 |       36 | WINDOWS  |                   73 |      91000 |
| Call of Duty: Modern Warfare          | i7        | 16GB   | Geforce GTX 1050 Ti |          55 |       40 | WINDOWS  |                   89 |     137000 |
| Cities: Skylines                      | i7        | 4GB    | Geforce GTX 1050 Ti |           5 |       30 | WINDOWS  |                   85 |     124000 |
| Counter Strike:Global Offensive       | i3        | 4GB    | Geforce GTX 1050 Ti |          15 |        0 | WINDOWS  |                   83 |      91000 |
| CRUCIBLE                              | i7        | 8GB    | Geforce GTX 1060    |          15 |       16 | WINDOWS  |                   60 |     131000 |
| Dead by Daylight                      | i3        | 16GB   | Geforce GTX 1060    |          25 |       20 | WINDOWS  |                   71 |     108000 |
| Destiny 2                             | i7        | 16GB   | Geforce GTX 1060    |         105 |        0 | WINDOWS  |                   83 |     141000 |
| Destiny 2: Beyond Light               | i7        | 8GB    | Geforce GTX 1060    |         105 |       70 | WINDOWS  |                   83 |     131000 |
| Divinity                              | i5        | 4GB    | Geforce GTX 1060    |          16 |        0 | WINDOWS  |                   94 |     100000 |
| Divinity:Original                     | i5        | 4GB    | Geforce GTX 1050 Ti |          10 |        0 | WINDOWS  |                   94 |      96000 |
| DOOM Eternal                          | i5        | 8GB    | Geforce GTX 1060    |          50 |       31 | WINDOWS  |                   90 |     103000 |
| Dota 2                                | i7        | 4GB    | Geforce GTX 1050 Ti |          15 |        0 | WINDOWS  |                   90 |     124000 |
| Euro Truck Simulator 2                | i3        | 4GB    | Geforce GTX 1060    |           3 |       20 | WINDOWS  |                   79 |      95000 |
| Fallout 4                             | i3        | 16GB   | Geforce GTX 1070 Ti |          30 |       55 | WINDOWS  |                   84 |     124000 |
| Far Cry 3                             | i5        | 4GB    | Geforce GTX 1070 Ti |          15 |       20 | WINDOWS  |                   88 |     116000 |
| Far Cry 5                             | i3        | 16GB   | Geforce GTX 1070 Ti |          40 |       60 | WINDOWS  |                   89 |     124000 |
| Farming Simulator 19                  | i3        | 4GB    | Geforce GTX 1060    |          23 |       25 | WINDOWS  |                   77 |      95000 |
| FIFA 19                               | i3        | 8GB    | Geforce GTX 1050 Ti |          50 |       40 | WINDOWS  |                   81 |      94000 |
| Fifa 20                               | i7        | 8GB    | Geforce GTX 1060    |          50 |       21 | WINDOWS  |                   20 |     131000 |
| FINAL FANTASY XIV Online              | i5        | 4GB    | Geforce GTX 1050 Ti |          60 |       15 | WINDOWS  |                   83 |      96000 |
| Football Manager 2020                 | i5        | 4GB    | Geforce GTX 1050 Ti |          50 |       30 | WINDOWS  |                   87 |      96000 |
| Fortnite                              | i3        | 8GB    | Geforce GTX 1050 Ti |          15 |        0 | WINDOWS  |                   81 |      94000 |
```



```
| Grand Theft Auto V (GTA 5)            | i7        | 4GB    | Geforce GTX 1060    |          72 |       15 | WINDOWS  |                   96 |     128000 |
| GTA IV                                | i5        | 16GB   | Geforce GTX 1050 Ti |          16 |       11 | WINDOWS  |                   90 |     109000 |
| GTA:Vice City                         | i5        | 4GB    | Geforce GTX 1050 Ti |          10 |        3 | WINDOWS  |                   94 |      96000 |
| iRacing                               | i5        | 8GB    | Geforce GTX 1060    |           6 |        0 | WINDOWS  |                   79 |     103000 |
| League of Legends                     | i7        | 16GB   | Geforce GTX 1060    |          12 |       16 | WINDOWS  |                   76 |     141000 |
| Mass Effect 2                         | i3        | 4GB    | Geforce GTX 1070 Ti |          14 |        4 | WINDOWS  |                   94 |     111000 |
| Metro Exodus                          | i7        | 16GB   | Geforce GTX 1070 Ti |          59 |       40 | WINDOWS  |                   95 |     157000 |
| MINECRAFT                             | i3        | 8GB    | Geforce GTX 1070 Ti |           4 |        0 | WINDOWS  |                   93 |     114000 |
| MONSTER HUNTER: WORLD                 | i5        | 16GB   | Geforce GTX 1070 Ti |          20 |       30 | WINDOWS  |                   88 |     129000 |
| Monster Hunter:World                  | i7        | 8GB    | Geforce GTX 1070 Ti |          20 |       21 | WINDOWS  |                   88 |     147000 |
| Mount and Blade                       | i7        | 8GB    | Geforce GTX 1070 Ti |          16 |        0 | WINDOWS  |                   78 |     147000 |
| NBA 2K20                              | i5        | 8GB    | Geforce GTX 1050 Ti |          80 |       13 | WINDOWS  |                   79 |      99000 |
| No Man's Sky                          | i3        | 16GB   | Geforce GTX 1070 Ti |          10 |       60 | WINDOWS  |                   61 |     124000 |
| Overwatch                             | i3        | 4GB    | Geforce GTX 1060    |          30 |       40 | WINDOWS  |                   91 |      95000 |
| Planet Zoo                            | i5        | 16GB   | Geforce GTX 1060    |          16 |       50 | WINDOWS  |                   82 |     113000 |
| Portal 2                              | i7        | 4GB    | Geforce GTX 1070 Ti |          10 |        0 | WINDOWS  |                   95 |     144000 |
| predator Galea                        | i7        | 4GB    | Geforce GTX 1050 Ti |          45 |       25 | WINDOWS  |                   85 |     124000 |
| Pubg Lite                             | i3        | 4GB    | Geforce GTX 1050 Ti |           4 |        0 | WINDOWS  |                   86 |      91000 |
| PUBG PC                               | i7        | 16GB   | Geforce GTX 1050 Ti |          35 |       31 | WINDOWS  |                   86 |     137000 |
| Rust                                  | i7        | 16GB   | Geforce GTX 1070 Ti |          20 |       40 | WINDOWS  |                   69 |     157000 |
| Scrap Mechanic                        | i3        | 16GB   | Geforce GTX 1060    |          15 |       20 | WINDOWS  |                   85 |     108000 |
| Sid Meier?s Civilization VI           | i7        | 4GB    | Geforce GTX 1050 Ti |          15 |       60 | WINDOWS  |                   88 |     124000 |
| Star Citizen                          | i3        | 16GB   | Geforce GTX 1050 Ti |          40 |       11 | WINDOWS  |                   42 |     104000 |
| Star Citizen 2                        | i7        | 16GB   | Geforce GTX 1060    |          40 |       10 | WINDOWS  |                   42 |     141000 |
| STAR WARS Jedi: Fallen Order          | i5        | 8GB    | Geforce GTX 1070 Ti |          55 |       35 | WINDOWS  |                   90 |     119000 |
| Terraria                              | i3        | 8GB    | Geforce GTX 1060    |           2 |       10 | WINDOWS  |                   83 |      98000 |
| The Borderlands 2                     | i5        | 4GB    | Geforce GTX 1050 Ti |          20 |       26 | WINDOWS  |                   90 |      96000 |
| The Elder Scrolls V: Skyrim           | i5        | 4GB    | Geforce GTX 1060    |           6 |       16 | WINDOWS  |                   94 |     100000 |
| The red dead redemption 2             | i7        | 16GB   | Geforce GTX 1060    |         150 |       30 | WINDOWS  |                   94 |     141000 |
| The Sims 4                            | i3        | 16GB   | Geforce GTX 1050 Ti |          20 |       41 | WINDOWS  |                   70 |     104000 |
| The Witcher 3                         | i3        | 8GB    | Geforce GTX 1070 Ti |          55 |       40 | WINDOWS  |                   93 |     114000 |
| Tom Clancy's Rainbow Six Siege        | i3        | 16GB   | Geforce GTX 1050 Ti |          61 |       20 | WINDOWS  |                   79 |     104000 |
| Total War: WARHAMMER II               | i3        | 16GB   | Geforce GTX 1060    |          51 |       60 | WINDOWS  |                   87 |     108000 |
| Valorant                              | i3        | 8GB    | Geforce GTX 1070 Ti |         120 |        0 | WINDOWS  |                   81 |     114000 |
| Warzone                               | i7        | 16GB   | Geforce GTX 1060    |         197 |       61 | WINDOWS  |                   79 |     141000 |
| World of Warcraft                     | i7        | 8GB    | Geforce GTX 1050 Ti |           5 |       15 | WINDOWS  |                   93 |     127000 |
+---------------------------------------+-----------+--------+---------------------+-------------+----------+----------+----------------------+------------+
72 rows in set (0.10 sec)
```

# Future Enhancements

1. Improving on presentation of the program with the help of Tkinter module and many more.

2. Adding more records to the database so the user has much better chances to solve his query.

3. Adding more graphs for visualizing purposes.

4. Giving user more graphics card, RAM and processor option so he has better chances to be updated to the latest products in the market and improve his chances for what he is looking for.

# Bibliography

1. https://buildmypc.net/list/ (for providing data for pc price)

2. https://www.pcgamebenchmark.com/ (for providing games configurations)

3. https://store.steampowered.com/ (for providing games price)

4. https://www.ea.com/ (for providing games price)

5. https://www.epicgames.com/store/en-US/ (for providing games price)

6. https://www.rockstargames.com/ (for providing games price)