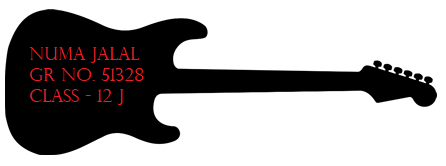
**THE INDIAN SCHOOL, KINGDOM OF BAHRAIN**

**COMPUTER SCIENCE PROJECT**

THE MUIC BARN





Your text here

***INDEX***

|  |  |
| --- | --- |
| Serial No. | Topic |
| 1 | Certificate |
| 2 | Acknowledgement |
| 3 | Introduction |
| 4 | About python |
| 5 | System analysis |
| 6 | System design |
| 8 | Sample outputs |
| 9 | Conclusion |

***CERTIFICATE***

*This is to certify that Numa Jalal of class XII-J has completed the Computer Science Project ‘Music Shop Program’ for the academic year 2020-2021 as per CBSE requirements.*

*Teacher’s Signature Examiner’s Signature*

*ACKNOWLEDGEMENT*

*I am extremely grateful to my Computer Science teacher, Mrs. Jyothi Abhilash, for giving me the knowledge, determination and creative freedom to come up with a project and work through all its errors and faults till I had achieved something I was proud of.*

*I would like to show my appreciation to my teammates, Mr. Brahmvir Singh, Ms. Richa D’mello and Ms. Anika Thirukkonda Karthikeyan without whose hard work and support I would not have been able to complete this project to the best of my abilities.*

*Lastly, I would like to thank the school authorities for providing me with the best facilities and infrastructure to carry out this project.*

***INTRODUCTION***

*The Music Barn is an application made using Python, an object-oriented programming language. It is a user-friendly program designed to both, help customers purchase music records from the wide selection stored in the back end of the program, and allow employees to manage different attributes of the database in the back.*

*It combines the use of a database in SQL with the ease and efficiency of the python interface. The program itself offers several functions to help users find music that would appeal to them and allows them to store their chosen records in a cart that erases with each new customer. The program also keeps track of the number of records sold for inventory purposes and provides statistics on bestsellers.*

***ABOUT PYTHON***

*Python is an interpreted, high-level, and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Python’s object-oriented approach aims to help programmers write clear, logical code for small and large-scale projects.*

*Python can serve as a scripting language for web applications. This means that it can automate specific series of tasks, making it more efficient.* *Python is used by Wikipedia, Google, Yahoo!, NASA, and among many other organizations. It is commonly used in Artificial Intelligence projects and Machine Learning projects with the help of libraries like TensorFlow, etc.*

***SYSTEM ANALYSIS***

*When the program is installed for the first time, the user is asked for their role, that is, if they are a customer or an employee. The user may also choose to exit the ‘MUSIC SHOP PROGRAM’.*

***MAIN MENU***

*Helps to handle and access various functions included in the program.*

1. ***CUSTOMER***

*The customer has a wide range of functions to choose from, like:*

*1)Display menu: This function displays all the songs in the database sorted in a number of ways like:*

1. *Record number*
2. *Artist*
3. *Date of recording and*
4. *Price*

*2)Add to Cart: In this function, the customer may add music of their choice to their cart.*

*3)Remove from Cart: The customer may remove music from his cart.*

*4)Display Cart: This function allows the customer to view their cart before checkout.*

*5)Proceed to Checkout: The customer can purchase their music.*

*6)Suggestions: Here, we suggest music to you based on your preferred genre or our Bestsellers.*

*7)Search Music: The customer can search for music based on The Track Name and by Artist’s Name.*

*8)Exit: The customer may exit from the ‘MUSIC SHOP PROGRAM’.*

***2.EMPLOYEE***

*The employee must first* ***sign in with their unique ID Code and password*** *to access the Employee Menu.*

*1)Insert Records: The Employee may insert New Music.*

*2) Edit Existing Records: This function is used to change any attribute of the existing data.*

*3)Display Records: It is used to view all the data we have.*

*4)Delete Records: The Employee may delete data.*

*5)Exit: It is used to exit from the ‘MUSIC SHOP PROGRAM’.*

***3.EXIT***

*The user may exit from the ‘MUSIC SHOP PROGRAM’.*

***SYSTEM DESIGN***

Login as employee

Login as customer

Exit

Customer menu

Enter password

Display records

Employee menu

Enter records

Add to cart

Edit records

Remove from cart

Display records

Display cart

Delete records

Checkout

Exit

Suggestions

Search for specific record

Exit

***PROJECT CODES***

*1.Music Barn program*

from Menu\_Functions import \*

print()

print('\t\t\t~~~~~~~~~~~~~~~~~~~~')

print('\t\t\t-----Welcome to-----')

print('\t\t\t\/\The Music Barn/\/\n')

print('\t\t\t~~~~~~~~~~~~~~~~~~~~')

print('\t\tHere to help with your music needs!\n')

print()

while True:

print('\t\tAre you signing in as \n\t\t1.a Customer\n\t\t2.an Employee\n\t\t3.Exit program')

print()

ch=input('\t\tEnter choice:')

print()

if ch=='1':

Customer()

break

elif ch=='2':

n=5

for k in range(n):

Pass=input('\t\tEnter password:')

print()

if Pass=='\*\*\*\*\*\*\*\*':

l=1

Employee()

break

else:

l=n-1-k

print('\t\tIncorrect password. You have',l,'tries left.')

j=input('\t\tPress any key to continue')

if l==0:

break

elif ch=='3':

print('\tThank you!')

break

else:

print('\tIncorrect option number, please try again')

ch=input('\tPress any key to continue')

*2. Menu\_Functions*

from tabulate import tabulate

from Cart\_Functions import \*

from employeefunctions import \*

import mysql.connector

mydb=mysql.connector.connect(host='localhost',user='root',database='music\_barn',passwd='limejuice')

mycursor=mydb.cursor()

#function for customer-The Music Barn

def Customer():

print('\t\t-------------------------------------------------')

print('\t\tHello! Welcome to The Music Barn! Happy Shopping!')

print('\t\t-------------------------------------------------\n')

while True:

print('\t\taming Menu:')

print('\t\t----------')

print('\t\t1.All The Songs We Have\n\t\t2.Add To Cart\n\t\t3.Remove from cart\n\t\t4.Display Cart\n\t\t5.Proceed To Checkout\n\t\t6.Suggestions\n\t\t7.Search for more music\n\t\t8.Exit\n')

ch=input('\tWhat would you like to do? ')

print()

if ch=='1':

Display\_menu()

elif ch=='2':

Add\_to\_cart()

elif ch=='3':

Remove\_from\_cart()

elif ch=='4':

Display\_cart()

elif ch=='5':

Checkout()

elif ch=='6':

Suggestions()

elif ch=='7':

Searchmusic()

elif ch=='8':

print('\tThank you for shopping at the Music Barn. Please come again!')

break

else:

print('\ERROR! Invalid Option! Please Try Again!')

key=input('\tPress any key to continue')

def Employee():

mydb=mysql.connector.connect(host='localhost',user='root',passwd='limejuice', database='music\_barn')

mycursor=mydb.cursor()

while True:

a=input('\tEnter employee ID:')

print()

try:

n=int(a)

except:

print('\tMake sure your employee ID has only number characters.')

continue

mycursor.execute("select EmpName from emp where EmpID={}".format(n))

res=mycursor.fetchall()

if res==[]:

print('\tInvalid Employee Number. Try again.')

ch=input('\tPress any key to continue')

else:

break

w=res[0]

print('\tWelcome back',w[0],'!\n')

while True:

print('\t\tMain Menu')

print('\t\t---------')

print('\t\t1.Insert new values \n\t\t2.Alter the existing values \n\t\t3.Display the records \n\t\t4.Delete a record \n\t\t5.Exit\n')

x=input('\tWhat are we doing today? ')

print()

if x=='1':

insert()

elif x=='2':

alter()

elif x=='3':

display()

elif x=='4':

delete()

elif x=='5':

print('\tGood Bye! Have a nice day!\n')

break

else:

print('\tInvalid option')

q=input('\tPress any key to continue')

*3 .Cart\_Functions*

import mysql.connector

mydb=mysql.connector.connect(host='localhost',user='root',passwd='limejuice', database='music\_barn')

mycursor=mydb.cursor()

from tabulate import tabulate

#Add to cart function

def Add\_to\_cart():

while True:

n=input('\tEnter Record Number of the track:')

try:

no=int(n)

except:

print('Invalid record number. Make sure it is only number character')

break

query1='create table if not exists Cart(rno int(5) primary key, title varchar(20),artist varchar(20), price decimal(5,2))'

mycursor.execute(query1)

mydb.commit()

try:

query2='select \* from playlist where rno={}'.format(no)

mycursor.execute(query2)

result=mycursor.fetchall()

for k in result:

title,artist,pri=k[1],k[2],k[6]

query3='insert into Cart values({},"{}","{}",{})'.format(no,title,artist,pri)

mycursor.execute(query3)

mydb.commit()

print('\tAdded to cart.')

query4='select count(\*) from cart'

mycursor.execute(query4)

count=mycursor.fetchall()

print('\tThere are',count[0][0],'items in your cart.')

except:

print('\tIncorrect record number, try again please.')

print()

def Display\_cart():

query='select \* from cart'

mycursor.execute(query)

result=mycursor.fetchall()

print()

if result==[]:

print('\tYour cart is empty.')

ch=input('\tPress any key to continue.')

print()

return

print(tabulate(result,headers=['Record Number','Track','Artist','Price'],tablefmt='psql'))

print()

def RecordNumber\_sort():

query='select \* from playlist order by Rno'

mycursor.execute(query)

result=mycursor.fetchall()

print(tabulate(result,headers=['Record Number','Track','Artist','Genre','Release Date','Records Sold','Price'],tablefmt='psql'))

def Artist\_sort():

query='select \* from playlist order by artist'

mycursor.execute(query)

result=mycursor.fetchall()

print(tabulate(result,headers=['Record Number','Track','Artist','Genre','Release Date','Records Sold','Price'],tablefmt='psql'))

def Dateofrec\_sort():

query='select \* from playlist order by DOE'

mycursor.execute(query)

result=mycursor.fetchall()

print(tabulate(result,headers=['Record Number','Track','Artist','Genre','Release Date','Records Sold','Price'],tablefmt='psql'))

def Price\_sort():

query='select \* from playlist order by price'

mycursor.execute(query)

result=mycursor.fetchall()

print(tabulate(result,headers=['Record Number','Track','Artist','Genre','Release Date','Records Sold','Price'],tablefmt='psql'))

#display menu function

def Display\_menu():

print('\t\tWould you like to see the music sorted by:')

print('\t\t------------------------------------------')

print('\t\t1.Record Number\n\t\t2.Artist\n\t\t3.Date of recording\n\t\t4.Price\n\t\t5.Return to Main Menu\n')

while True:

ch=input('\tTake Your Pick: ')

print()

if ch=='1':

RecordNumber\_sort()

elif ch=='2':

Artist\_sort()

elif ch=='3':

Dateofrec\_sort()

elif ch=='4':

Price\_sort()

elif ch=='5':

break

else:

print('\tInvalid Option. Please try Again')

print('\tPress 5 to return to Main Menu')

print()

def Remove\_from\_cart():

no=int(input('\tEnter record number to be removed:'))

try:

mycursor.execute('select count(\*) from cart')

c1=mycursor.fetchall()

query='Delete from cart where rno={}'.format(no)

mycursor.execute(query)

mydb.commit()

query4='select count(\*) from cart'

mycursor.execute(query4)

count=mycursor.fetchall()

if c1>count:

print('\tRecord removed.')

print('\tThere are',count[0][0],'items in your cart.\n')

except:

print('Incorrect record number. Please try again.')

print()

def Checkout():

query='select rno,price from cart'

mycursor.execute(query)

result=mycursor.fetchall()

if result==[]:

print('\tYour cart is empty. Add records you want to purchase to your cart and try again.')

ans=input('\tAre you sure you would like to proceed to checkout? (y or n):')

if ans=='n':

print()

return

elif ans=='y':

Price=0

for k in result:

Price+=k[1]

no=k[0]

query2='update playlist set No\_Purchased=No\_purchased+1 where rno={}'.format(no)

mycursor.execute(query2)

mydb.commit()

print('\tTotal amount=',Price,'/-')

print('\tPlease pay at the counter.')

ch=input('\tPress any key when amount paid')

query3='delete from cart'

mycursor.execute(query3)

mydb.commit()

print()

else:

print('\tInvalid answer. Answer "y" or "n"')

ch=input('\tPress any key to continue:')

print()

#search function-customer

def Searchmusic():

while True:

print('\t\tHow would you like to look for more music?')

print('\t\t------------------------------------------')

print('\t\t1.By Track\n\t\t2.By Artist\n\t\t3.Exit')

ch=input('\tWhich would you prefer: ')

try:

if ch=='1':

Track\_search()

elif ch=='2':

Artist\_search()

elif ch=='3':

break

except:

print('\tInvalid Option!Please Try Again!')

key=input('\tPress any key to continue')

print()

#To search for music by the name of the song

def Track\_search():

L=[]

print('\t\*Remember song titles are case sensitive.\*')

songname=input('\tEnter Track to be Searched:')

mycursor.execute('Select \* from playlist where Track="{}"'.format(songname))

result=mycursor.fetchall()

if result==[]:

print("\tTrack not found. We'll be sure to have it next time!")

return

for k in result:

if k[1]==songname:

L.append(k)

print(tabulate(L,headers=['Rec No','Track','Artist','Genre','DOE','No. of Rec','Price'],tablefmt='psql'))

print()

def Artist\_search():

L=[]

print('\t\* Remember Artist names are case sensitive.\*')

artistname=input('\tEnter name of Artist:')

mycursor.execute('select \* from playlist where Artist="{}"'.format(artistname))

result=mycursor.fetchall()

if result==[]:

print("\tArtist not found. We'll be sure to have it next time!")

return

for k in result:

if k[2]==artistname:

L.append(k)

print(tabulate(L,headers=['Rec No','Track','Artist','Genre','DOE','No. of Rec','Price'],tablefmt='psql'))

print()

#function for suggestions

def Suggestions():

while True:

print("\t\tChoose a Genre and we'll give you our suggestions!")

print('\t\t--------------------------------------------------')

print('\t\t1.Best sellers\n\t\t2.Bollywood\n\t\t3.Country\n\t\t4.K-pop\n\t\t5.Pop\n\t\t6.Rock\n\t\t7.Exit\n')

ch=input("\tSo what'll it be? " )

try:

if ch=='1':

Bestsellers()

elif ch=='2':

Bollywood()

elif ch=='3':

Country()

elif ch=='4':

Kpop()

elif ch=='5':

Pop()

elif ch=='6':

Rock()

elif ch=='7':

break

except:

print('\tInvalid Option! Please Try Again.')

l=input('\tPress any key to continue')

print()

#bestseller function

def Bestsellers():

L=[]

mycursor.execute('Select Rno,Track,Artist,Genre,No\_purchased,price from playlist')

resultg=mycursor.fetchall()

for k in resultg:

if k[4]>5:

L.append(k)

print(tabulate(L,headers=['Rec No','Track','Artist','Genre','Recs Sold','Price'],tablefmt='psql'))

def Bollywood():

L=[]

mycursor.execute('Select Rno,Track,Artist,Genre,No\_purchased,price from playlist')

resultg=mycursor.fetchall()

for k in resultg:

if k[3]=='Bollywood':

L.append(k)

print(tabulate(L,headers=['Rec No','Track','Artist','Recs Sold','Price'],tablefmt='psql'))

def Country():

L=[]

mycursor.execute('Select Rno,Track,Artist,Genre,No\_purchased,price from playlist')

resultg=mycursor.fetchall()

for k in resultg:

if k[3]=='Country':

L.append(k)

print(tabulate(L,headers=['Rec No','Track','Artist','Recs Sold','Price'],tablefmt='psql'))

def Kpop():

L=[]

mycursor.execute('Select Rno,Track,Artist,Genre,No\_purchased,price from playlist')

resultg=mycursor.fetchall()

for k in resultg:

if k[3] in'Kpop':

L.append(k)

print(tabulate(L,headers=['Rec No','Track','Artist','Recs Sold','Price'],tablefmt='psql'))

def Pop():

L=[]

mycursor.execute('Select Rno,Track,Artist,Genre,No\_purchased,price from playlist')

resultg=mycursor.fetchall()

for k in resultg:

if k[3]=='Pop':

L.append(k)

print(tabulate(L,headers=['Rec No','Track','Artist','Recs Sold','Price'],tablefmt='psql'))

def Rock():

L=[]

mycursor.execute('Select Rno,Track,Artist,Genre,No\_purchased,price from playlist')

resultg=mycursor.fetchall()

for k in resultg:

if k[3]=='Rock':

L.append(k)

print(tabulate(L,headers=['Rec No','Track','Artist','Recs Sold','Price'],tablefmt='psql'))

*4.Employee\_Function*

import mysql.connector

#alter function

def alter():

mydb=mysql.connector.connect(host='localhost',user='root',passwd='limejuice', database='music\_barn')

mycursor=mydb.cursor()

while True:

print("\n\t\t Update Menu")

print("\t\t------------")

print("\t\t1.Track")

print("\t\t2.Artist")

print("\t\t3.Genre")

print("\t\t4.Date of Entry")

print("\t\t5.Number of Records Purchased")

print("\t\t6.Price")

print("\t\t7.I would not like to alter anything\n")

x=input("\tWhat would you like to update? ")

print()

if x=="1":

a=int(input("\tChoose the record number to be updated:"))

mycursor.execute('select \* from playlist where Rno={}'.format(a))

c=mycursor.fetchall()

if c==[]:

print("\tRecord doesn't exist")

else:

b=input("\tEnter the altered track name:")

query="update playlist set Track='{}' where Rno={}".format(b,a)

mycursor.execute(query)

mydb.commit()

print("\tRecord updated!")

elif x=="2":

a=int(input("\tChoose the record number to be updated:"))

mycursor.execute('select \* from playlist where Rno={}'.format(a))

c=mycursor.fetchall()

if c==[]:

print("\tRecord doesn't exist")

else:

b=input("\tEnter the altered Artist name:")

query="update playlist set Artist='{}' where Rno={}".format(b,a)

mycursor.execute(query)

mydb.commit()

print("\tRecord updated!")

elif x=="3":

a=int(input("\tChoose the record number to be updated:"))

mycursor.execute('select \* from playlist where Rno={}'.format(a))

c=mycursor.fetchall()

if c==[]:

print("\tRecord doesn't exist")

else:

b=input("\tEnter the altered Genre:")

query="update playlist set Genre='{}' where Rno={}".format(b,a)

mycursor.execute(query)

mydb.commit()

print("\tRecord updated!")

elif x=="4":

a=int(input("\tChoose the record number to be updated:"))

mycursor.execute('select \* from playlist where Rno={}'.format(a))

c=mycursor.fetchall()

if c==[]:

print("\tRecord doesn't exist")

else:

b=input("\tEnter the altered Date of Entry:")

query="update playlist set DOE='{}' where Rno={}".format(b,a)

mycursor.execute(query)

mydb.commit()

print("\tRecord updated!")

elif x=="5":

a=int(input("\tChoose the record number to be updated:"))

mycursor.execute('select \* from playlist where Rno={}'.format(a))

c=mycursor.fetchall()

if c==[]:

print("\tRecord doesn't exist")

else:

b=int(input("\tEnter the altered Number of Records Purchased:"))

query="update playlist set No\_Purchased={} where Rno={}".format(b,a)

mycursor.execute(query)

mydb.commit()

print("\tRecord updated!")

elif x=="6":

a=int(input("\tChoose the record number to be updated:"))

mycursor.execute('select \* from playlist where Rno={}'.format(a))

c=mycursor.fetchall()

if c==[]:

print("\tRecord doesn't exist")

else:

b=float(input("\tEnter the altered Price:"))

query="update playlist set Price={} where Rno={}".format(b,a)

mycursor.execute(query)

mydb.commit()

print("\tRecord updated!")

elif x=="7":

break

else:

print("\tInvalid Option")

key=input("\tPress any key to continue.")

#insert function

def insert():

mydb=mysql.connector.connect(host='localhost',user='root',passwd='limejuice', database='music\_barn')

mycursor=mydb.cursor()

while True:

an=input('\tEnter the Record Number:')

try:

a=int(an)

except:

print('\tMake sure your record number contains numeric values only.\n')

continue

break

mycursor.execute('select \* from playlist where Rno={}'.format(a))

c=mycursor.fetchall()

if c!=[]:

print("\tRecord number already exists.\n")

else:

try:

b=input("\tEnter the track name:")

c=input('\tEnter Name of the Artist:')

d=input('\tEnter the Genre:')

e=input('\tEnter the date of entry (yyyy-mm-dd):')

f=float(input('\tEnter the price of the record:'))

query="insert into playlist values({},'{}','{}','{}','{}',0,{})".format(a,b,c,d,e,f)

mycursor.execute(query)

mydb.commit()

print('\tRecord added.\n')

except:

print('\tAn error occured, please try again.\n')

#display function

import mysql.connector

from tabulate import tabulate

def display():

mydb=mysql.connector.connect(host='localhost',user='root',passwd='limejuice', database='music\_barn')

mycursor=mydb.cursor()

mycursor.execute('\tselect \* from playlist')

result=mycursor.fetchall()

print(tabulate(result,headers=['Rno','Track','Artist','Genre','Date of Entry','No. of records purchased','Price'],tablefmt='psql'))

#delete function

def delete():

mydb=mysql.connector.connect(host='localhost',user='root',passwd='limejuice', database='music\_barn')

mycursor=mydb.cursor()

mycursor.execute('select count(\*) from playlist')

c=mycursor.fetchall()

a=int(input('\tEnter the record number to be deleted:'))

mycursor.execute('delete from playlist where Rno={}'.format(a))

mydb.commit()

mycursor.execute('select count(\*) from playlist')

c1=mycursor.fetchall()

if c1<c:

print('\tRecord deleted')

else:

print("\tRecord doesn't exist")

***SAMPLE OUTPUTS***

*Customer Output:*

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Table

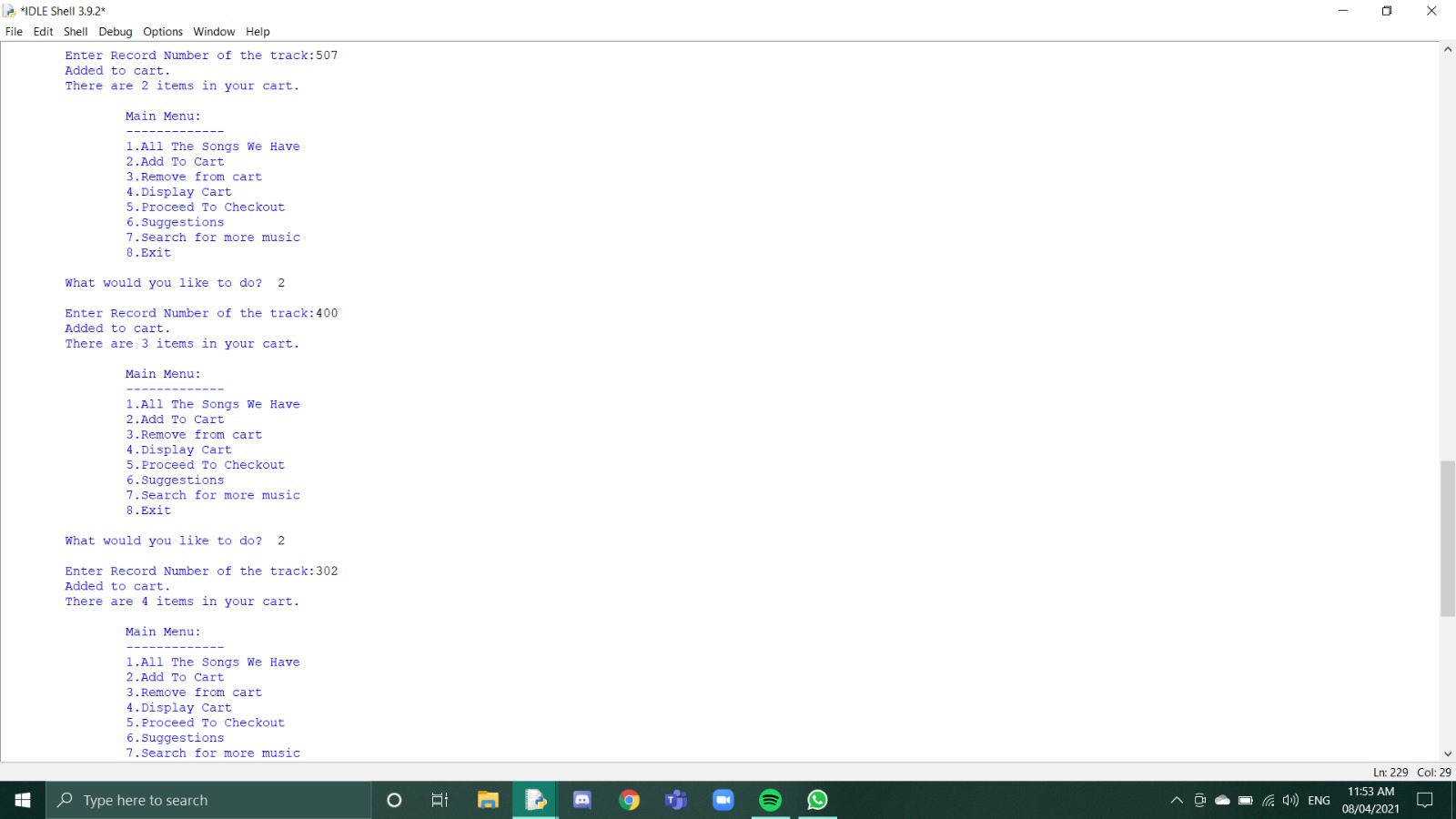
Description automatically generated

Table

Description automatically generated

Table

Description automatically generated



Text, table

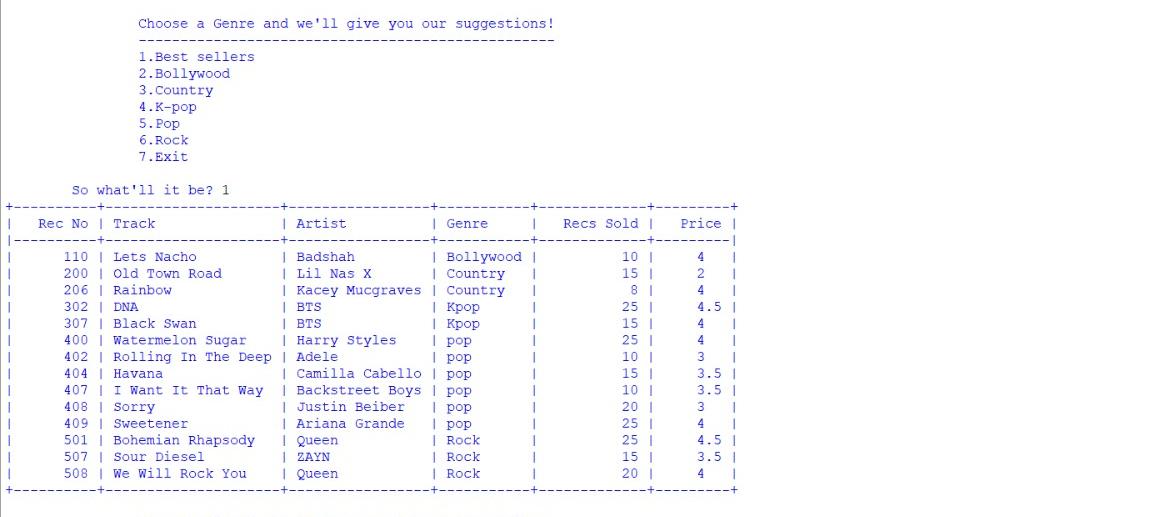
Description automatically generated with medium confidence

Text

Description automatically generated with medium confidence

Table

Description automatically generated

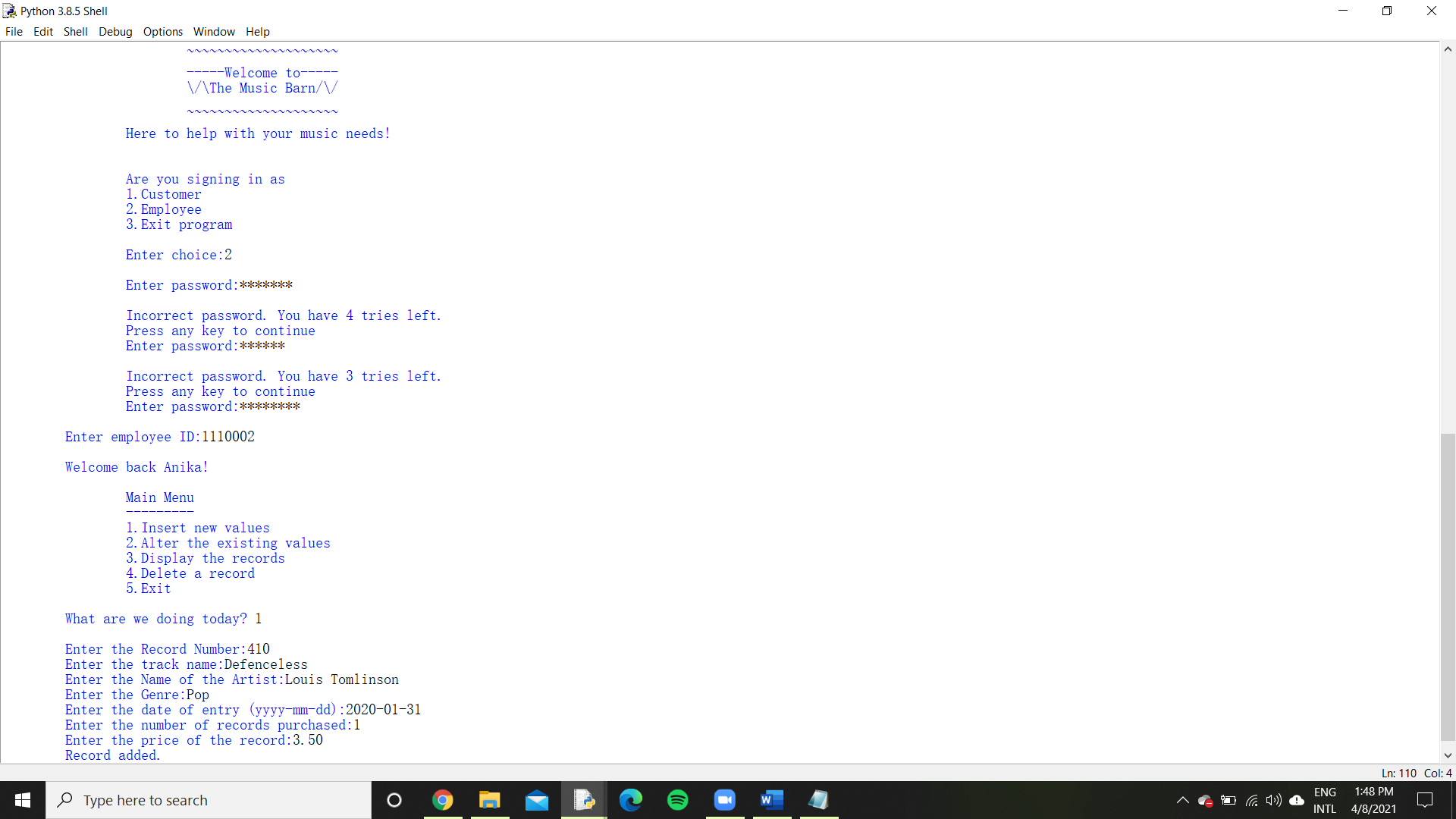
Text

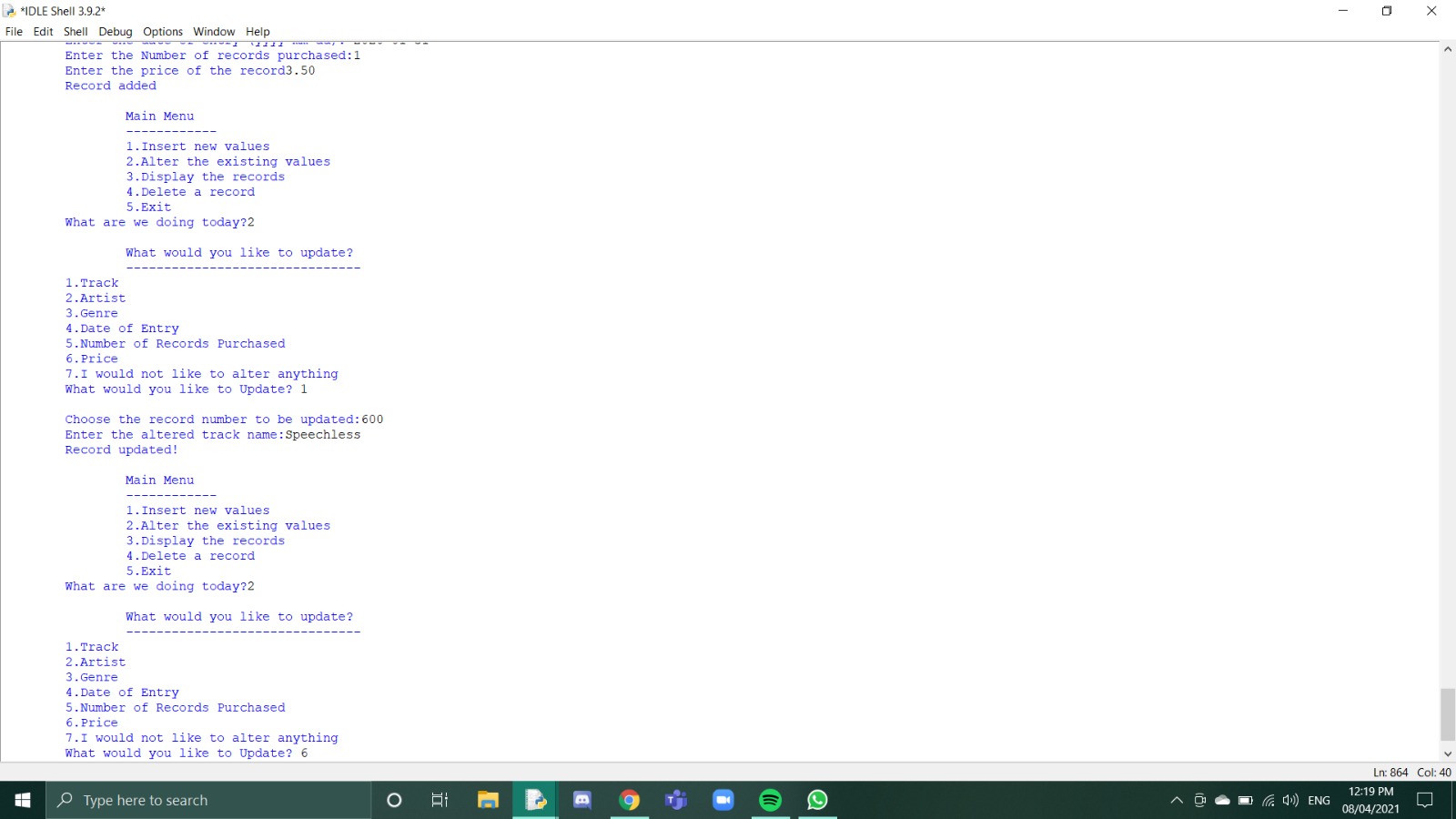
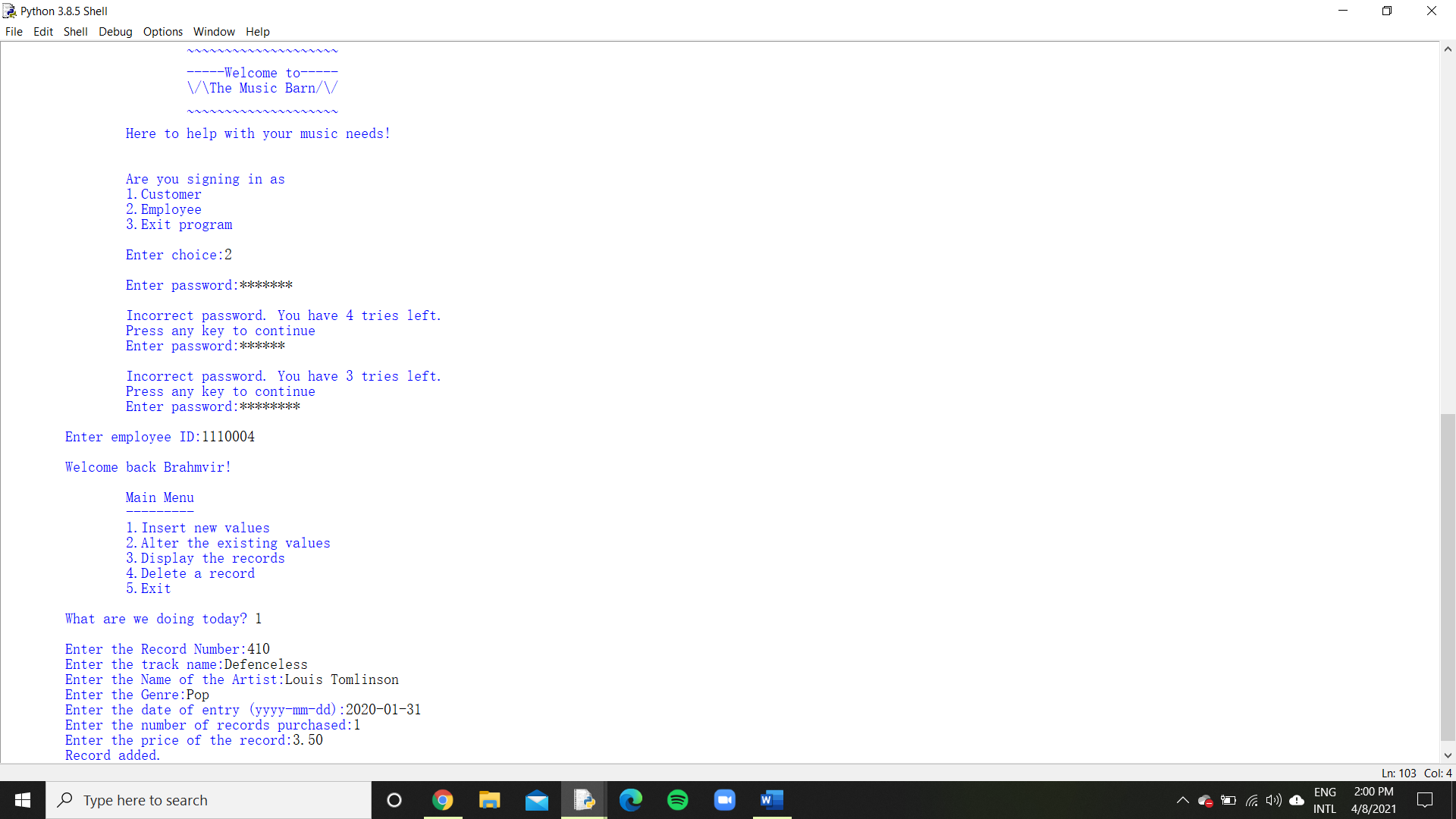
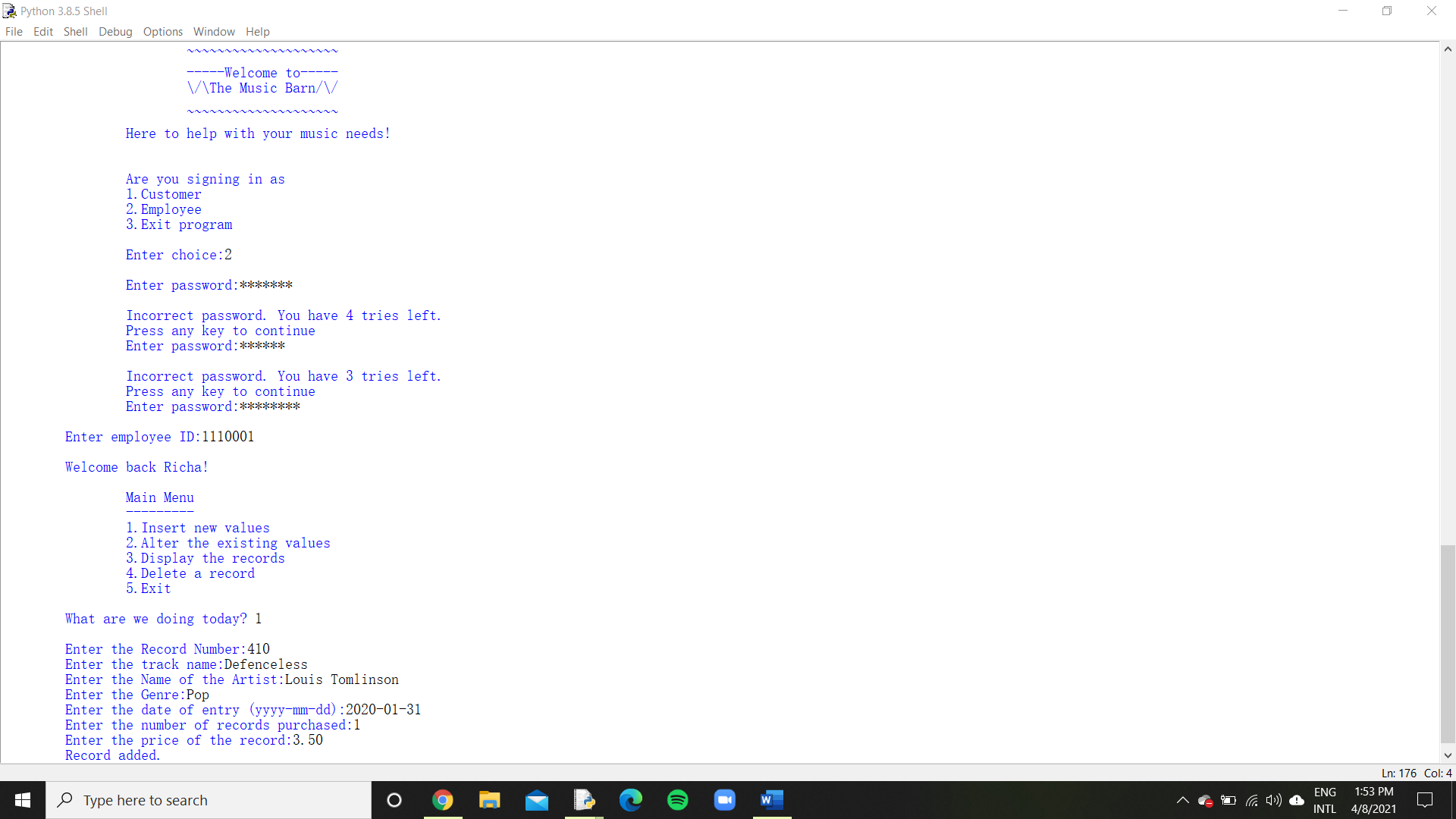
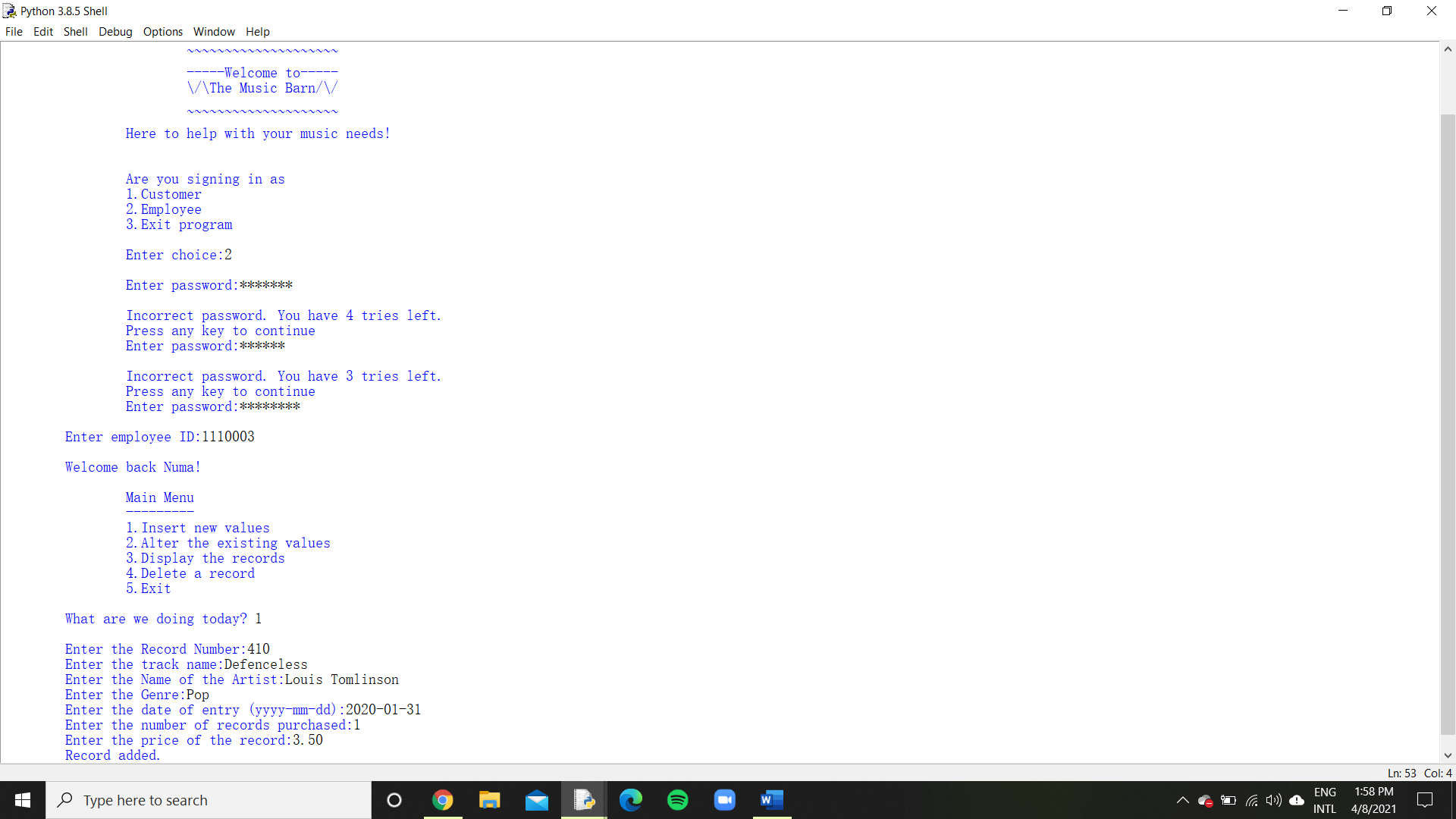
Description automatically generated with low confidence

Graphical user interface, text, application

Description automatically generated

*Employee Output:*





Table

Description automatically generated

Table

Description automatically generatedTable

Description automatically generated

Table

Description automatically generated

***Conclusion***

*The functioning of this project goes to prove that a complex and powerful computing language such as PYTHON can be used to create efficient and easy-to-use programs that are convenient for our everyday lives.*

*Hence, we can conclude that programming languages indeed make our lives easier to manage and provide an infinite amount of customization to fit in with our individual specifications.*