Amazon’t

A Computer Science project

[beta version]

# Index

|  |  |  |
| --- | --- | --- |
| *1.* | *Introduction .........................* | *1* |
| *2.* | *System requirements ............* | *2* |
| *3.* | *Feasibility study ...................* | *3* |
| *4.* | *Investigative analysis ...........* | *4* |
| *5.* | *Errors and its types ..............* | *5* |
| *6.* | *Source code ..........................* | *6* |
| *7.* | *Outputs .................................* | *19* |
| *8.* | *Conclusion ...........................* | *36* |
| *9.* | *Bibliography ........................* | *37* |

# 

# Introduction

Computer programming has become a necessity in recent times due to the various purposes it can be used for in day-to-day needs. There are many programming languages for these purposes

In this project, *Python* and *MySQL* are primarily used. Both are simple, open source, and easily available. Here, we connect both softwares using the ‘MySQL Connector’ package which is also open source and easily downloadable

We write the program’s source code, which is written in Python and MySQL queries, in the *Python interface*. Since both are connected, we only have to create the database using the ‘*MySQL Command Client’.* All the other queries (eg: creating tables, inserting values etc) are done within the python interface by mentioning the necessary queries in the required areas

In this project, we have used the latest version of both softwares. This project emphasizes on the fact that we are inputting the data through Python and storing it in the MySQL database

# System Requirements

## Hardware

* Processor
* Keyboard
* Minimum memory - 2 GB

## Software

* Operating system: Windows 10
* Python IDLE(3.10.2, 64 bit)
* MySQL (5.5)

# Feasibility Study

Feasibility study is a system proposal according to its work, ability, impact on the operations; ability to meet the needs of users, and efficient use of resources. It is an important outcome of preliminary investigations.

## Economic feasibility

Economic analysis is the most frequently used method for evaluating the effectiveness of the program, the benefits and savings that are expected from the system, and comparing them with cost.

This program code is not very expensive. So the user records can be maintained at a cheaper rate.

## Technical feasibility

Technical feasibility centers on existing computers and to what extent it can support the proposed task. This involves financial consideration to accommodate technical enhancement.

This is technically feasible because whatever technology is needed to develop the software is easily available.

# Investigative Analysis

**Amazon’t** :- It is an online store that offers the most random collection of items where users can purchase as much as they like without a care in the world or for their bank accounts.

The entire program consists of four files, one of which is used to display UI for the user. The second file will handle general functions (checking whether user is admin or not etc) and will redirect to the other files as necessary

Upon running the given source code, we are asked to log in where one of two things will happen:

1. **User mode**: It gives the perspective of a user who is casually shopping on the site and deleting all of their money
2. **Admin mode**: this time, we are an employee of Amazon’t and we deal with adding items for sale, confirming transactions, handing out deliveries etc.

Upon logging in as an admin, the following options are available:

1. **Alter items**: to add/remove items available for sale
2. **Check customers**: To verify customer details and to remove customers who have logged out and are now inactive
3. **Check transactions**: to ensure a transaction has been completed before it gets sent out for delivery
4. **Update delivery statuses**: Once the delivery has been completed, the admin sets the status to ‘True’ which will promptly allow the user to drown in whatever they bought

# Errors and its Types

An error, also called a bug, is some exception in the code that prevents the program from compiling or running as intended. They are broadly classified into three types:

## **Compile time errors**

An error that occurs during compilation of the program is called a compile time error. It is of two types as follows:

1. ***Syntax Error***: it refers to formal rules governing the construction of valid statements in a language
2. ***Semantic Error***: It refers to the set of rules which give the meaning of a statement

## **Runtime errors**

Errors that occur during the execution of the program are called runtime errors. These are harder to detect. Some runtime errors stop the execution of the program which would then be called “*Program Crashed*”

## **Logical errors**

Sometimes, no errors are encountered during compiling and runtime, but the program does not give the expected results. This is because of the programmer’s mistaken analysis of the problem he/she is trying to solve. Such errors are called logical errors

# Source Code

## File 1: ‘*index.py’*

import mainFunctions as main

if main.connection():

print("\*"\*15,"CONNECTED","\*"\*15)

user = main.login() # 0 = user | 1 = admin

print("-+"\*10,"WELCOME TO AMAZON'T","+-"\*10,"\n\n")

if user == 1: # admin

while True:

print("-$"\*15,"ADMIN FUNCTIONS","$-"\*15)

print("1. View/Alter items \n2. Customers \n3. Transactions \n4. Delivery status \n0. Quit")

func = int(input("Enter function: "))

main.adminFunctions(func)

elif user == 0: # user

while True:

print("&="\*15,"USER FUNCTIONS","=&"\*15)

print("1. View items \n2. View Cart \n3. Delivery status \n0. Quit")

func = int(input("Enter function: "))

main.userFunctions(func)

## File 2: *‘mainFunctions.py’*

import mysql.connector as mys

import adminFunctions as af

import userFunctions as uf

def connection() -> bool:

try:

db = mys.connect(host="localhost", user='root', passwd="root", database="project")

c = db.cursor()

return True

except:

return False

def login() -> int:

user = input("Enter username: ")

passwd = input("Enter password: ")

if user == "admin" and passwd == "root":

print("Welcome admin!")

return 1

else:

print("Welcome user!")

return 0

def userFunctions(funcNum):

uf.Main.start()

if funcNum == 0:

quit()

elif funcNum == 1:

while True:

print("-="\*10,"ITEMS(USER)","=-"\*10)

print("[ITEM ID, NAME, QTY, PRICE]")

uf.Items.viewTable()

print("1. Place an order \n2. Search for an item \n0. Go back")

func1 = int(input("Enter function: "))

if func1 == 1:

uf.Items.order()

print("Item(s) added to cart")

elif func1 == 2:

uf.Items.search()

elif func1 == 0:

uf.Main.end()

break

elif funcNum == 2:

while True:

print("-="\*10,"CART","=-"\*10)

uf.Cart.viewTable()

print("1. Confirm Order \n0. Go back")

func2 = int(input("Enter function: "))

if func2 == 1:

uf.Cart.confirm()

print("Order confirmed!")

elif func2 == 0:

uf.Main.end()

break

elif funcNum == 3:

while True:

print("-="\*10,"DELIVERY","=-"\*10)

print("1. Check delivery status \n0. Go back")

func3 = int(input("Enter function: "))

if func3 == 1:

uf.deliveryStatus()

elif func3 == 0:

uf.Main.end()

break

def adminFunctions(funcNum):

af.Main.start()

if funcNum == 0:

quit()

elif funcNum == 1:

while True:

print("-="\*10,"ITEMS(ADMIN)","=-"\*10)

print("1. Add item \n2. Delete item \n3. Search item \n4. View table \n0. Go back")

func1 = int(input("Enter function: "))

if func1 == 1:

af.Items.insertItem()

print("Item added\n")

elif func1 == 2:

af.Items.deleteItem()

print("Item deleted\n")

elif func1 == 3:

af.Items.searchItem()

elif func1 == 4:

af.Items.viewTable()

elif func1 == 0:

af.Main.end()

break

else:

print("What are you doing here?")

elif funcNum == 2:

while True:

print("-="\*10,"CUSTOMERS(ADMIN)","=-"\*10)

print("1. View all customers \n2. Add customer\n3. Remove a customer \n4. Search for a customer \n0. Go back")

func2 = int(input("Enter function: "))

if func2 == 1:

af.Customers.viewTable()

elif func2 == 2:

af.Customers.addCustomer()

elif func2 == 3:

af.Customers.removeCustomer()

elif func2 == 4:

af.Customers.searchCustomer()

elif func2 == 0:

af.Main.end()

break

elif funcNum == 3:

while True:

print("-="\*10,"TRANSACTIONS(ADMIN)","=-"\*10)

print("1. View transactions \n2. Update transaction \n3. Check transaction\n0. Go back")

func3 = int(input("Enter function: "))

if func3 == 1:

af.Transactions.viewTable()

elif func3 == 2:

af.Transactions.update()

elif func3 == 3:

af.Transactions.check()

elif func3 == 0:

af.Main.end()

break

elif funcNum == 4:

while True:

print("-="\*10,"DELIVERIES(ADMIN)","=-"\*10)

print("1. Check deliveries \n2. Search a delivery\n3. Update delivery status \n0. Go back")

func4 = int(input("Enter function: "))

if func4 == 1:

af.Deliveries.viewTable()

elif func4 == 2:

af.Deliveries.search()

elif func4 == 3:

af.Deliveries.updateStatus()

elif func4 == 0:

af.Main.end()

break

## File 3: *‘adminFunctions.py’*

import mysql.connector as mys

# Common for all admin functions

class Main:

def start():

global db ; global c;

db = mys.connect(host="localhost", user='root', passwd="root", database="project")

c = db.cursor()

def end():

c.close()

db.close()

# functions related to table 'ITEMS'

class Items:

def insertItem():

iD = int(input("Enter item iD: "))

name = input("Enter item name: ")

price = int(input("Enter price: "))

qty = int(input("Enter quantity: "))

c.execute(f"insert into items values ({iD},'{name}',{price},{qty})")

db.commit()

def deleteItem():

iD = int(input("Enter iD of item to be deleted: "))

c.execute(f"delete from items where ITEM\_ID={iD}")

db.commit()

def searchItem():

iD = int(input("Enter iD of item to be searched: "))

c.execute(f"select \* from items where ITEM\_ID={iD}")

record = c.fetchone()

print(record)

def viewTable():

c.execute("select \* from items")

table = c.fetchall()

for i in table:

print(i)

# functions related to table 'CUSTOMERS'

class Customers:

def viewTable():

c.execute("select \* from customers")

table = c.fetchall()

for i in table:

print(i)

def addCustomer():

ciD = int(input("Enter customer ID: "))

name = input("Enter name: ")

phno = int(input("Enter phone number: "))

email = input("Enter email address: ")

c.execute(f"insert into customers values ({ciD},'{name}',{phno},'{email}')")

db.commit()

def removeCustomer():

iD = input("Enter customer iD: ")

c.execute(f"delete from customers where C\_ID={iD}")

db.commit()

def searchCustomer():

iD = int(input("Enter iD of customer to be searched: "))

c.execute(f"select \* from customers where C\_ID={iD}")

record = c.fetchone()

print(record)

# functions related to table 'TRANSACTIONS'

class Transactions:

def viewTable():

c.execute("select \* from transactions")

table = c.fetchall()

for i in table:

print(i)

def update():

iD = int(input("Enter transaction ID: "))

value = int(input("Completed/Incomplete (0/1): "))

c.execute(f"update transactions set STATUS={value} where T\_ID={iD}")

db.commit()

print("Updated!")

def check():

iD = int(input("Enter transaction ID: "))

c.execute(f"select \* from transactions where T\_ID={iD}")

check = c.fetchone()

if check[-1] == 1:

print("Transaction completed")

else:

print("Transaction incomplete")

# functions related to table 'DELIVERY'

class Deliveries:

def viewTable():

c.execute("select \* from deliveries")

table = c.fetchall()

for i in table:

print(i)

def search():

iD = int(input("Enter transaction ID: "))

c.execute(f"select \* from deliveries where T\_ID={iD}")

record = c.fetchone()

print(record)

def updateStatus():

iD = int(input("Enter transaction ID: "))

value = int(input("Enter status (0/1): "))

c.execute(f"update deliveries set status={value} where T\_ID={iD}")

db.commit()

print("Status updated!")

## File 4: *‘userFunctions.py’*

import mysql.connector as mys

from random import randint

from random import choice

addresses = [

"8939 High Point St.",

"233 New Saddle St.",

"9197 Sunbeam Ave.",

"7800 E. Lake Forest Lane",

"224 Lower River St.",

"8603 Bedford Street",

"94 St Margarets Dr.",

"688 Buckingham Dr.",

"378 Beaver Ridge Drive",

"82 W. Main Circle",

"423 Birchpond Dr.",

"636 Westport St.",

"444 Armstrong St.",

"50 Essex St.",

"8876 Hudson Court",

"19 Spruce Drive",

"630 Hudson Street",

"56 East Windsor Street",

"7799 Queen Street",

"89 Meadowbrook Rd.",

"33 Smith Dr.",

"8271 Fordham Ave.",

"522 Prince Ave.",

"5 East Lilac Street",

"553 Trenton St."

]

class Main:

def start():

global db ; global c;

db = mys.connect(host="localhost", user='root', passwd="root", database="project")

c = db.cursor()

def end():

c.close()

db.close()

class Items:

def viewTable():

c.execute("select \* from items;")

table = c.fetchall()

for i in table:

print(i)

def order():

global ciD

try:

ciD = int(input("Enter customer ID: "))

iD = int(input("Enter item ID: "))

qty = int(input("Enter quantity required: "))

c.execute(f"select \* from items where ITEM\_ID={iD}")

record = c.fetchone()

temp = list(record)

temp[-2] = qty

temp = [ciD]+temp

c.execute(f"insert into cart values {tuple(temp)}")

db.commit()

except:

print("ID not found. Try again")

def search():

iD = int(input("Enter item ID: "))

c.execute(f"select \* from items where ITEM\_ID={iD}")

record = c.fetchone()

if record is None:

print("Item not available")

else:

print(record)

class Cart:

def viewTable():

c.execute("select \* from cart")

table = c.fetchall()

for i in table:

print(i)

def confirm():

ciD = int(input("Enter customer ID: "))

moP = input("Enter mode of payment: (Cash/Credit/UPI): ")

t\_ID = randint(500, 599)

address = choice(addresses)

c.execute(f"insert into transactions values ({t\_ID},{ciD},'{moP}', 0)")

c.execute(f"insert into deliveries values ({t\_ID},'{address}', 0)")

c.execute("delete from cart")

db.commit()

print("Your transaction ID is:",t\_ID)

def deliveryStatus():

t\_ID = int(input("Enter transaction ID: "))

try:

c.execute(f"select \* from deliveries where T\_ID={t\_ID}")

record = c.fetchone()

if record[-1] == 0:

print("Order awaiting delivery")

else:

print("Order has been delivered")

except:

print("Transaction ID not found")

# Outputs

## Logged in as a user

### 1. Placing an order

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CONNECTED \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter username: bing

Enter password: chilling

Welcome user!

-+-+-+-+-+-+-+-+-+-+ WELCOME TO AMAZON'T +-+-+-+-+-+-+-+-+-+-

&=&=&=&=&=&=&=&=&=&=&=&=&=&=&= USER FUNCTIONS =&=&=&=&=&=&=&=&=&=&=&=&=&=&=&

1. View items

2. View Cart

3. Delivery status

0. Quit

Enter function: 1

-=-=-=-=-=-=-=-=-=-= ITEMS(USER) =-=-=-=-=-=-=-=-=-=-

[ITEM ID, NAME, QTY, PRICE]

(101, 'Dum', 10200, 4)

(102, 'b', 1200, 1)

1. Place an order

2. Search for an item

0. Go back

Enter function: 1

Enter customer ID: 201

Enter item ID: 101

Enter quantity required: 20

Item(s) added to cart

### 2. Searching for an item

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CONNECTED \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter username: bing

Enter password: chilling

Welcome user!

-+-+-+-+-+-+-+-+-+-+ WELCOME TO AMAZON'T +-+-+-+-+-+-+-+-+-+-

&=&=&=&=&=&=&=&=&=&=&=&=&=&=&= USER FUNCTIONS =&=&=&=&=&=&=&=&=&=&=&=&=&=&=&

1. View items

2. View Cart

3. Delivery status

0. Quit

Enter function: 1

-=-=-=-=-=-=-=-=-=-= ITEMS(USER) =-=-=-=-=-=-=-=-=-=-

[ITEM ID, NAME, QTY, PRICE]

(101, 'Kothamangalam Subbu', 10200, 4)

(102, 'Dum', 1200, 1)

1. Place an order

2. Search for an item

0. Go back

Enter function: 2

Enter item ID: 102

(102, 'Dum', 1200, 1)

### 3. Confirming the order

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CONNECTED \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter username: bing

Enter password: chilling

Welcome user!

-+-+-+-+-+-+-+-+-+-+ WELCOME TO AMAZON'T +-+-+-+-+-+-+-+-+-+-

&=&=&=&=&=&=&=&=&=&=&=&=&=&=&= USER FUNCTIONS =&=&=&=&=&=&=&=&=&=&=&=&=&=&=&

1. View items

2. View Cart

3. Delivery status

0. Quit

Enter function: 2

-=-=-=-=-=-=-=-=-=-= CART =-=-=-=-=-=-=-=-=-=-

(201, 101, 'Dum', 20, 4)

1. Confirm Order

0. Go back

Enter function: 1

Enter customer ID: 201

Enter mode of payment: (Cash/Credit/UPI): cash

Your transaction ID is: 584

Order confirmed!

### 4. Checking delivery status

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CONNECTED \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter username: bing

Enter password: chilling

Welcome user!

-+-+-+-+-+-+-+-+-+-+ WELCOME TO AMAZON'T +-+-+-+-+-+-+-+-+-+-

&=&=&=&=&=&=&=&=&=&=&=&=&=&=&= USER FUNCTIONS =&=&=&=&=&=&=&=&=&=&=&=&=&=&=&

1. View items

2. View Cart

3. Delivery status

0. Quit

Enter function: 3

-=-=-=-=-=-=-=-=-=-= DELIVERY =-=-=-=-=-=-=-=-=-=-

1. Check delivery status

0. Go back

Enter function: 1

Enter transaction ID: 584

Order awaiting delivery

## Logged in as an admin

### 1. Adding an item to table “ITEMS”

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CONNECTED \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter username: admin

Enter password: root

Welcome admin!

-+-+-+-+-+-+-+-+-+-+ WELCOME TO AMAZON'T +-+-+-+-+-+-+-+-+-+-

-$-$-$-$-$-$-$-$-$-$-$-$-$-$-$ ADMIN FUNCTIONS $-$-$-$-$-$-$-$-$-$-$-$-$-$-$-

1. View/Alter items

2. Customers

3. Transactions

4. Delivery status

0. Quit

Enter function: 1

-=-=-=-=-=-=-=-=-=-= ITEMS(ADMIN) =-=-=-=-=-=-=-=-=-=-

1. Add item

2. Delete item

3. Search item

4. View table

0. Go back

Enter function: 1

Enter item iD: 103

Enter item name: er

Enter price: 1200

Enter quantity: 350

Item added

### 2. Deleting an item from “ITEMS”

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CONNECTED \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter username: admin

Enter password: root

Welcome admin!

-+-+-+-+-+-+-+-+-+-+ WELCOME TO AMAZON'T +-+-+-+-+-+-+-+-+-+-

-$-$-$-$-$-$-$-$-$-$-$-$-$-$-$ ADMIN FUNCTIONS $-$-$-$-$-$-$-$-$-$-$-$-$-$-$-

1. View/Alter items

2. Customers

3. Transactions

4. Delivery status

0. Quit

Enter function: 1

-=-=-=-=-=-=-=-=-=-= ITEMS(ADMIN) =-=-=-=-=-=-=-=-=-=-

1. Add item

2. Delete item

3. Search item

4. View table

0. Go back

Enter function: 2

Enter iD of item to be deleted: 103

Item deleted

### 3. Searching for an item in “ITEMS”

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CONNECTED \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter username: admin

Enter password: root

Welcome admin!

-+-+-+-+-+-+-+-+-+-+ WELCOME TO AMAZON'T +-+-+-+-+-+-+-+-+-+-

-$-$-$-$-$-$-$-$-$-$-$-$-$-$-$ ADMIN FUNCTIONS $-$-$-$-$-$-$-$-$-$-$-$-$-$-$-

1. View/Alter items

2. Customers

3. Transactions

4. Delivery status

0. Quit

Enter function: 1

-=-=-=-=-=-=-=-=-=-= ITEMS(ADMIN) =-=-=-=-=-=-=-=-=-=-

1. Add item

2. Delete item

3. Search item

4. View table

0. Go back

Enter function: 3

Enter iD of item to be searched: 102

(102, 'Dum', 1200, 1)

### 4. Viewing the table “ITEMS”

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CONNECTED \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter username: admin

Enter password: root

Welcome admin!

-+-+-+-+-+-+-+-+-+-+ WELCOME TO AMAZON'T +-+-+-+-+-+-+-+-+-+-

-$-$-$-$-$-$-$-$-$-$-$-$-$-$-$ ADMIN FUNCTIONS $-$-$-$-$-$-$-$-$-$-$-$-$-$-$-

1. View/Alter items

2. Customers

3. Transactions

4. Delivery status

0. Quit

Enter function: 1

-=-=-=-=-=-=-=-=-=-= ITEMS(ADMIN) =-=-=-=-=-=-=-=-=-=-

1. Add item

2. Delete item

3. Search item

4. View table

0. Go back

Enter function: 4

(101, 'Mud', 10200, 4)

(102, 'Dum', 1200, 1)

### 5. Viewing all customers in “CUSTOMERS” table

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CONNECTED \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter username: admin

Enter password: root

Welcome admin!

-+-+-+-+-+-+-+-+-+-+ WELCOME TO AMAZON'T +-+-+-+-+-+-+-+-+-+-

-$-$-$-$-$-$-$-$-$-$-$-$-$-$-$ ADMIN FUNCTIONS $-$-$-$-$-$-$-$-$-$-$-$-$-$-$-

1. View/Alter items

2. Customers

3. Transactions

4. Delivery status

0. Quit

Enter function: 2

-=-=-=-=-=-=-=-=-=-= CUSTOMERS(ADMIN) =-=-=-=-=-=-=-=-=-=-

1. View all customers

2. Add customer

3. Remove a customer

4. Search for a customer

0. Go back

Enter function: 1

(201, 'S Vasan', 1234567890, 'vasan@gmail.com')

(202, 'PP Ramani', 987654321, 'ppraman[@gmail.com](mailto:gooberish@gmail.com)')

### 6. Adding a customer to table “CUSTOMERS”

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CONNECTED \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter username: admin

Enter password: root

Welcome admin!

-+-+-+-+-+-+-+-+-+-+ WELCOME TO AMAZON'T +-+-+-+-+-+-+-+-+-+-

-$-$-$-$-$-$-$-$-$-$-$-$-$-$-$ ADMIN FUNCTIONS $-$-$-$-$-$-$-$-$-$-$-$-$-$-$-

1. View/Alter items

2. Customers

3. Transactions

4. Delivery status

0. Quit

Enter function: 2

-=-=-=-=-=-=-=-=-=-= CUSTOMERS(ADMIN) =-=-=-=-=-=-=-=-=-=-

1. View all customers

2. Add customer

3. Remove a customer

4. Search for a customer

0. Go back

Enter function: 2

Enter customer ID: 203

Enter name: Mahesh Dalle

Enter phone number: 987654321

Enter email address: [jupiterRain@gmail.com](mailto:jupiterRain@gmail.com)

### 7. Removing a customer from “CUSTOMERS”

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CONNECTED \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter username: admin

Enter password: root

Welcome admin!

-+-+-+-+-+-+-+-+-+-+ WELCOME TO AMAZON'T +-+-+-+-+-+-+-+-+-+-

-$-$-$-$-$-$-$-$-$-$-$-$-$-$-$ ADMIN FUNCTIONS $-$-$-$-$-$-$-$-$-$-$-$-$-$-$-

1. View/Alter items

2. Customers

3. Transactions

4. Delivery status

0. Quit

Enter function: 2

-=-=-=-=-=-=-=-=-=-= CUSTOMERS(ADMIN) =-=-=-=-=-=-=-=-=-=-

1. View all customers

2. Add customer

3. Remove a customer

4. Search for a customer

0. Go back

Enter function: 3

Enter customer iD: 203

### 8. Searching for a customer from “CUSTOMERS”

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CONNECTED \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter username: admin

Enter password: root

Welcome admin!

-+-+-+-+-+-+-+-+-+-+ WELCOME TO AMAZON'T +-+-+-+-+-+-+-+-+-+-

-$-$-$-$-$-$-$-$-$-$-$-$-$-$-$ ADMIN FUNCTIONS $-$-$-$-$-$-$-$-$-$-$-$-$-$-$-

1. View/Alter items

2. Customers

3. Transactions

4. Delivery status

0. Quit

Enter function: 2

-=-=-=-=-=-=-=-=-=-= CUSTOMERS(ADMIN) =-=-=-=-=-=-=-=-=-=-

1. View all customers

2. Add customer

3. Remove a customer

4. Search for a customer

0. Go back

Enter function: 4

Enter iD of customer to be searched: 201

(201, 'S Vasan', 1234567890, 'vasan[@gmail.com](mailto:goober@gmail.com)')

### 9. Viewing all transactions in table “TRANSACTIONS”

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CONNECTED \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter username: admin

Enter password: root

Welcome admin!

-+-+-+-+-+-+-+-+-+-+ WELCOME TO AMAZON'T +-+-+-+-+-+-+-+-+-+-

-$-$-$-$-$-$-$-$-$-$-$-$-$-$-$ ADMIN FUNCTIONS $-$-$-$-$-$-$-$-$-$-$-$-$-$-$-

1. View/Alter items

2. Customers

3. Transactions

4. Delivery status

0. Quit

Enter function: 3

-=-=-=-=-=-=-=-=-=-= TRANSACTIONS(ADMIN) =-=-=-=-=-=-=-=-=-=-

1. View transactions

2. Update transaction

3. Check transaction

0. Go back

Enter function: 1

(507, 201, 'Cash', 0)

(508, 201, 'cash', 0)

(554, 201, 'Credit', 0)

(584, 201, 'cash', 0)

### 10. Updating a transaction from “TRANSACTIONS”

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CONNECTED \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter username: admin

Enter password: root

Welcome admin!

-+-+-+-+-+-+-+-+-+-+ WELCOME TO AMAZON'T +-+-+-+-+-+-+-+-+-+-

-$-$-$-$-$-$-$-$-$-$-$-$-$-$-$ ADMIN FUNCTIONS $-$-$-$-$-$-$-$-$-$-$-$-$-$-$-

1. View/Alter items

2. Customers

3. Transactions

4. Delivery status

0. Quit

Enter function: 3

-=-=-=-=-=-=-=-=-=-= TRANSACTIONS(ADMIN) =-=-=-=-=-=-=-=-=-=-

1. View transactions

2. Update transaction

3. Check transaction

0. Go back

Enter function: 2

Enter transaction ID: 507

Completed/Incomplete (0/1): 1

Updated!

### 11. Checking a transaction from “TRANSACTIONS”

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CONNECTED \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter username: admin

Enter password: root

Welcome admin!

-+-+-+-+-+-+-+-+-+-+ WELCOME TO AMAZON'T +-+-+-+-+-+-+-+-+-+-

-$-$-$-$-$-$-$-$-$-$-$-$-$-$-$ ADMIN FUNCTIONS $-$-$-$-$-$-$-$-$-$-$-$-$-$-$-

1. View/Alter items

2. Customers

3. Transactions

4. Delivery status

0. Quit

Enter function: 3

-=-=-=-=-=-=-=-=-=-= TRANSACTIONS(ADMIN) =-=-=-=-=-=-=-=-=-=-

1. View transactions

2. Update transaction

3. Check transaction

0. Go back

Enter function: 3

Enter transaction ID: 507

Transaction completed

### 12. Checking deliveries from table “DELIVERIES”

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CONNECTED \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter username: admin

Enter password: root

Welcome admin!

-+-+-+-+-+-+-+-+-+-+ WELCOME TO AMAZON'T +-+-+-+-+-+-+-+-+-+-

-$-$-$-$-$-$-$-$-$-$-$-$-$-$-$ ADMIN FUNCTIONS $-$-$-$-$-$-$-$-$-$-$-$-$-$-$-

1. View/Alter items

2. Customers

3. Transactions

4. Delivery status

0. Quit

Enter function: 4

-=-=-=-=-=-=-=-=-=-= DELIVERIES(ADMIN) =-=-=-=-=-=-=-=-=-=-

1. Check deliveries

2. Search a delivery

3. Update delivery status

0. Go back

Enter function: 1

(507, '8603 Bedford Street', 0)

(508, '8876 Hudson Court', 0)

(524, '636 Westport St.', 1)

(540, '50 Essex St.', 0)

(554, '553 Trenton St.', 1)

(584, '56 East Windsor Street', 0)

### 13. Searching a delivery

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CONNECTED \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter username: admin

Enter password: root

Welcome admin!

-+-+-+-+-+-+-+-+-+-+ WELCOME TO AMAZON'T +-+-+-+-+-+-+-+-+-+-

-$-$-$-$-$-$-$-$-$-$-$-$-$-$-$ ADMIN FUNCTIONS $-$-$-$-$-$-$-$-$-$-$-$-$-$-$-

1. View/Alter items

2. Customers

3. Transactions

4. Delivery status

0. Quit

Enter function: 4

-=-=-=-=-=-=-=-=-=-= DELIVERIES(ADMIN) =-=-=-=-=-=-=-=-=-=-

1. Check deliveries

2. Search a delivery

3. Update delivery status

0. Go back

Enter function: 2

Enter transaction ID: 507

(507, '8603 Bedford Street', 0)

### 14. Updating delivery status

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CONNECTED \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter username: admin

Enter password: root

Welcome admin!

-+-+-+-+-+-+-+-+-+-+ WELCOME TO AMAZON'T +-+-+-+-+-+-+-+-+-+-

-$-$-$-$-$-$-$-$-$-$-$-$-$-$-$ ADMIN FUNCTIONS $-$-$-$-$-$-$-$-$-$-$-$-$-$-$-

1. View/Alter items

2. Customers

3. Transactions

4. Delivery status

0. Quit

Enter function: 4

-=-=-=-=-=-=-=-=-=-= DELIVERIES(ADMIN) =-=-=-=-=-=-=-=-=-=-

1. Check deliveries

2. Search a delivery

3. Update delivery status

0. Go back

Enter function: 3

Enter transaction ID: 507

Enter status (0/1): 1

Status updated!

# Conclusion

The program has been executed successfully and no errors have been found during execution

Thus, the program is economically and technically feasible with absence of compile-time, runtime, and logical errors. Ready for use

The source code can be further modified with more features according to the usability with the help of Python and MySQL

# Bibliography

* [www.python.org](http://www.python.org)
* “*Computer Science with python Class XII*” by Sumita Arora
* <https://github.com/>
* <https://www.geeksforgeeks.org/>
* [www.programiz.com](http://www.programiz.com)