**Pharmacy Drugs Inventory Management**

**Milestone: Python Application**

**Group 11**

**Sai Varun Kumar Namburi**

**FNU Meenal**

**857-265-1349 (Sai Varun)**

**848-667-4233 (Meenal)**

[namburi.sai@northeastern.edu](mailto:namburi.sai@northeastern.edu)

[lnu.meenal@northeastern.edu](mailto:lnu.meenal@northeastern.edu)

**Percentage of Effort Contributed by Student1: 50%**

**Percentage of Effort Contributed by Student2: 50%**

**Signature of Student 1: Sai Varun**

**Signature of Student 2: Meenal**

**Submission Date: November 5, 2022**

**Python Application:**

This milestone aims to connect the MySQL database with python and perform basic analysis.

I have used the “MySQLdb” library in python, which is used for connecting to the MySQL database

Here you can have a look at how I connected to my local database.

Text

Description automatically generated with medium confidence

Connecting to a database requires 4 attributes, username, password, dbname, and hostname.

After connecting to a database now we can start using python functionalities to query the database and retrieve the data.

**Basic Analysis:**

1. Checking all the orders which have a total amount of more than 25000 and are ordered by total\_amount in descending order

Query: **select \* from orders where total\_amount>25000 order by total\_amount desc;**

Output for the above query after running in the python, it returns as a data frame

Graphical user interface, text

Description automatically generated

1. Now I have added one more filter condition to check orders for the year 2021

Query: **select \* from orders where total\_amount>25000 and**

**order\_date<** **'2021-12-31' order by total\_amount desc;**

Output:

Graphical user interface

Description automatically generated with medium confidence

1. Now we can find the data which have a total amount of t less than 25000 and have an order date before Jan 2022

Query: **select \* from orders join order\_details on orders.order\_id = order\_details.order\_id where orders.total\_amount>25000 and orders.order\_date < '** **2021-12-31' order by orders.total\_amount desc;**

Output:

Text

Description automatically generated

1. Next, we are going to find the drugs which are having less than 200 stock in the store

Query: **select \* from stock\_details where stock\_left < 200;**

Output:

Graphical user interface, text

Description automatically generated

1. Finding the order details of those who have purchased the drugs with less quantity less than 30

Query:

select od.order\_id,od.drug\_id,d.drug\_name, od.quantity, o.order\_date, o.payment\_type, o.total\_amount

from orders o join order\_details od on o.order\_id=od.order\_id

join drugs d on d.drug\_id=od.drug\_id where od.quantity<30;

Output:

Graphical user interface, text

Description automatically generated with medium confidence

Now after doing the analysis with the data, you must close the DB connection which has created at the start of the application

Output:

