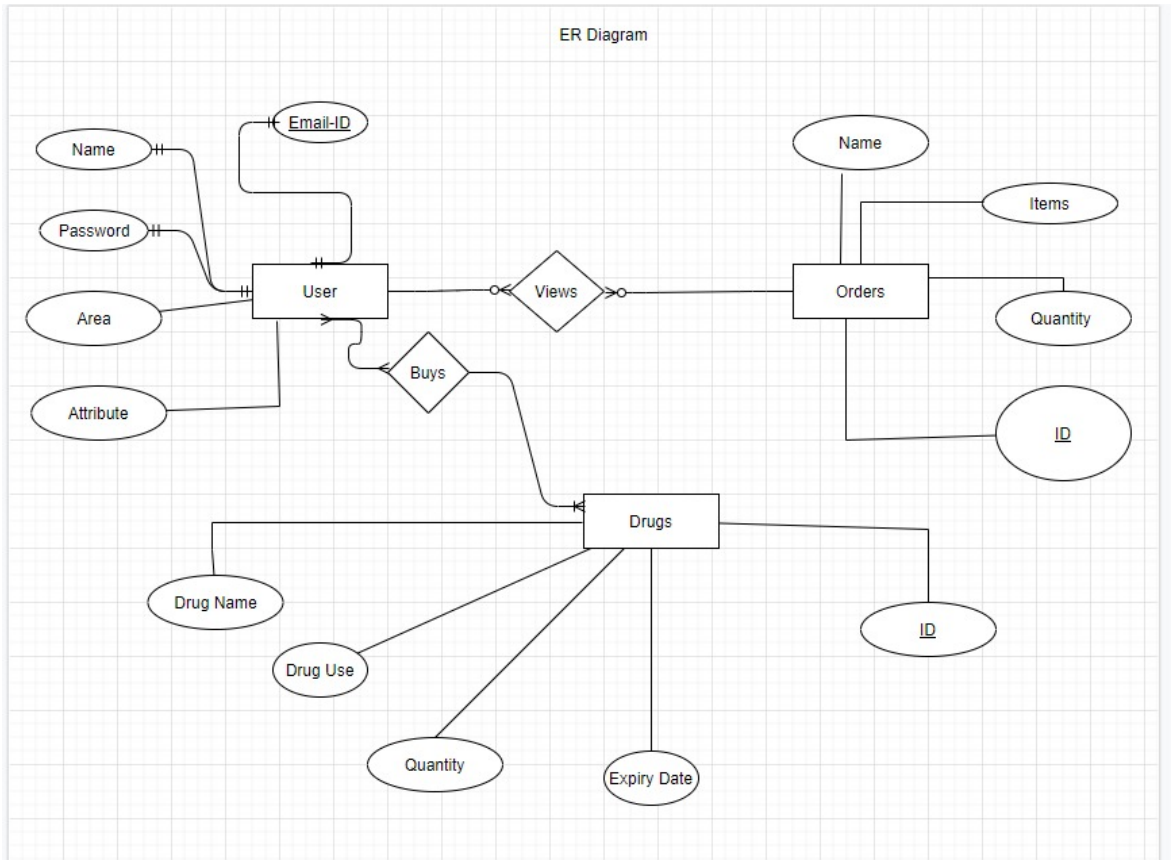
ECO-SQUAD | CSE-D |5TH SEM

PHARMACY MANAGEMENT SYSTEM

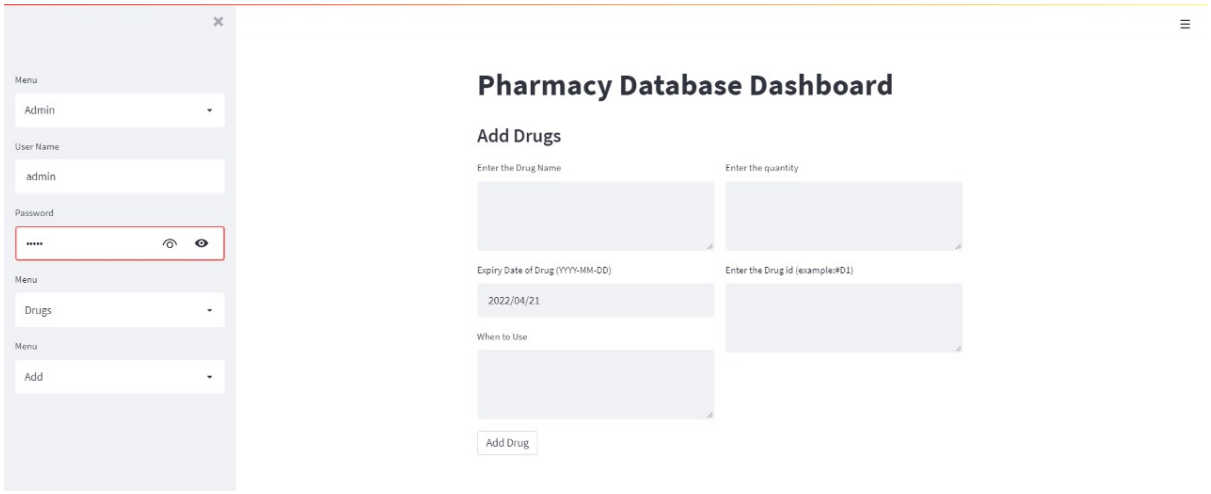
TEAM MEMBERS:-

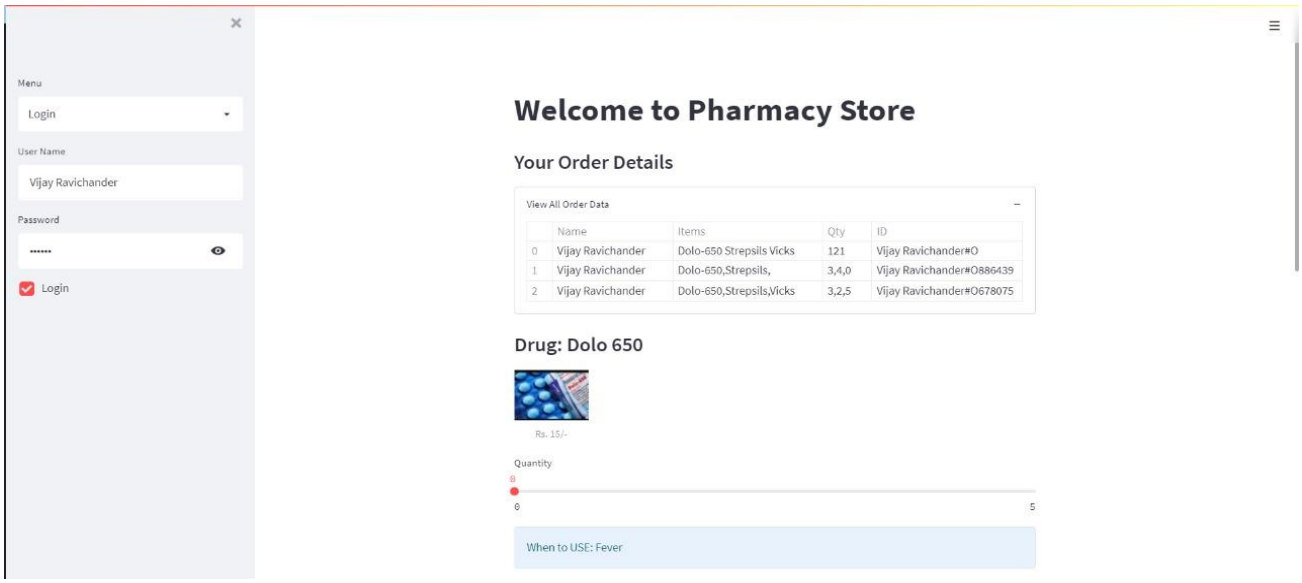
NISCHITH S	4NI22CS259
ROHAN P N	4NI22CS257
SUHAS B H	4NI23CS220
SUHAS B M	4NI22CS221
THANMAN Mahesh	4NI22CS236
VISWAS H T	4NI23CS248

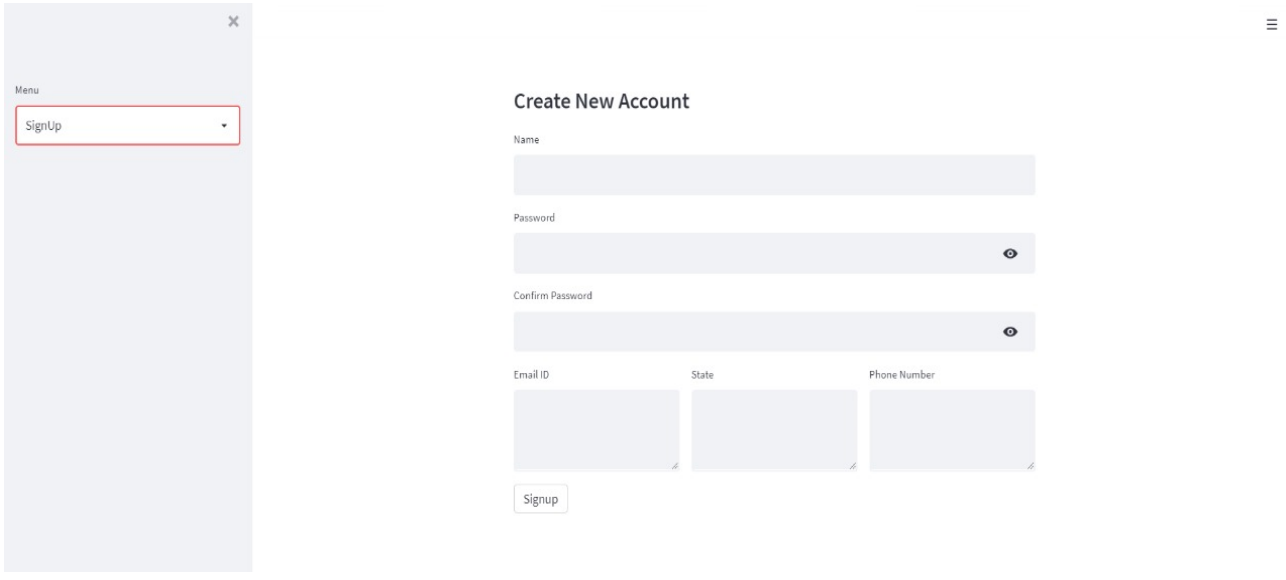
PHARMACY MANAGEMENT SYSTEM:-ER DIAGRAM



OUTPUT:




Menu
Login
User Name
Vijay Ravichander
Password
Login
Welcome to Pharmacy Store
Your Order Details
View All Order Data
Name
Items
Qty
ID
0 Vijay Ravichander Dolo-650 Strepsils Vicks 121 Vijay Ravichander#O
1 Vijay Ravichander Dolo-650,Strepsils, 3,4,0 Vijay Ravichander#O886439
2 Vijay Ravichander Dolo-650,Strepsils,Vicks 3,2,5 Vijay Ravichander#O678075
Drug: Dolo 650
Rs. 15/-
Quantity
0
0
5
When to USE: Fever


Menu
SignUp
Create New Account
Name
Password
Confirm Password
Email ID
State
Phone Number
Signup

PYTHON CODE:-

```
import streamlit as st
import pandas as pd
from PIL import Image
#from drug_db import *
import random

## SQL DATABASE CODE
import sqlite3

conn = sqlite3.connect("drug_data.db",check_same_thread=False)
c = conn.cursor()

def cust_create_table():
    c.execute('''CREATE TABLE IF NOT EXISTS Customers(
                C_Name VARCHAR(50) NOT NULL,
                C_Password VARCHAR(50) NOT NULL,
                C_Email VARCHAR(50) PRIMARY KEY NOT NULL,
                C_State VARCHAR(50) NOT NULL,
                C_Number VARCHAR(50) NOT NULL
                )''')
    print('Customer Table create Successfully')
```

```
def customer_add_data(Cname,Cpass, Cemail, Cstate,Cnumber):
    c.execute('''INSERT INTO Customers (C_Name,C_Password,C_Email, C_State, C_Number) VALUES(?,?,?,?,?)''', (Cname,Cpass,  Cemail, Cstate,Cnumber))
    conn.commit()

def customer_view_all_data():
    c.execute('SELECT * FROM Customers')
    customer_data = c.fetchall()
    return customer_data
def customer_update(Cemail,Cnumber):
    c.execute(''' UPDATE Customers SET C_Number = ? WHERE C_Email = ?''', (Cnumber,Cemail,))
    conn.commit()
    print("Updating")
def customer_delete(Cemail):
    c.execute(''' DELETE FROM Customers WHERE C_Email = ?''', (Cemail,))
    conn.commit()

def drug_update(Duse, Did):
    c.execute(''' UPDATE Drugs SET D_Use = ? WHERE D_id = ?''', (Duse,Did))
    conn.commit()
def drug_delete(Did):
    c.execute(''' DELETE FROM Drugs WHERE D_id = ?''', (Did,))
    conn.commit()

def drug_create_table():
    c.execute('''CREATE TABLE IF NOT EXISTS Drugs(
```

```
            D_Name VARCHAR(50) NOT NULL,
            D_ExpDate DATE NOT NULL,
            D_Use VARCHAR(50) NOT NULL,
            D_Qty INT NOT NULL,
            D_id INT PRIMARY KEY NOT NULL)
            ''')
    print('DRUG Table create Successfully')

def drug_add_data(Dname, Dexpdate, Duse, Dqty, Did):
    c.execute('''INSERT INTO Drugs (D_Name, D_Expdate, D_Use, D_Qty, D_id)
VALUES(?,?,?,?,?)''', (Dname, Dexpdate, Duse, Dqty, Did))
    conn.commit()

def drug_view_all_data():
    c.execute('SELECT * FROM Drugs')
    drug_data = c.fetchall()
    return drug_data

def order_create_table():
    c.execute('''
        CREATE TABLE IF NOT EXISTS Orders(
            O_Name VARCHAR(100) NOT NULL,
            O_Items VARCHAR(100) NOT NULL,
            O_Qty VARCHAR(100) NOT NULL,
            O_id VARCHAR(100) PRIMARY KEY NOT NULL)
    ''')
```

```
def order_delete(Oid):
    c.execute('DELETE FROM Orders WHERE O_id = ?', (Oid,))
    conn.commit()


def order_add_data(O_Name,O_Items,O_Qty,O_id):
    c.execute('''INSERT INTO Orders (O_Name, O_Items,O_Qty, O_id)
VALUES(?,?,?,?)''',
              (O_Name,O_Items,O_Qty,O_id))
    conn.commit()


def order_view_data(customername):
    c.execute('SELECT * FROM ORDERS Where O_Name == ?',(customername,))
    order_data = c.fetchall()
    return order_data


def order_view_all_data():
    c.execute('SELECT * FROM Orders')
    order_all_data = c.fetchall()
    return order_all_data




#_____
_____
```

```
def admin():

    st.title("Pharmacy Database Dashboard")
    menu = ["Drugs", "Customers", "Orders", "About"]
    choice = st.sidebar.selectbox("Menu", menu)


    ## DRUGS
    if choice == "Drugs":

        menu = ["Add", "View", "Update", "Delete"]
        choice = st.sidebar.selectbox("Menu", menu)
        if choice == "Add":

            st.subheader("Add Drugs")


            col1, col2 = st.columns(2)


            with col1:
                drug_name = st.text_area("Enter the Drug Name")
                drug_expiry = st.date_input("Expiry Date of Drug (YYYY-MM-DD)")
                drug_mainuse = st.text_area("When to Use")
```

```
        with col2:
            drug_quantity = st.text_area("Enter the quantity")
            drug_id = st.text_area("Enter the Drug id (example:#D1)")


        if st.button("Add Drug"):
            drug_add_data(drug_name,drug_expiry,drug_mainuse,drug_quantity,
drug_id)
            st.success("Successfully Added Data")
    if choice == "View":
        st.subheader("Drug Details")
        drug_result = drug_view_all_data()
        #st.write(drug_result)
        with st.expander("View All Drug Data"):
            drug_clean_df = pd.DataFrame(drug_result, columns=["Name",
"Expiry Date", "Use", "Quantity", "ID"])
            st.dataframe(drug_clean_df)
        with st.expander("View Drug Quantity"):
            drug_name_quantity_df = drug_clean_df[['Name','Quantity']]
            #drug_name_quantity_df = drug_name_quantity_df.reset_index()
            st.dataframe(drug_name_quantity_df)
    if choice == 'Update':
        st.subheader("Update Drug Details")
        d_id = st.text_area("Drug ID")
        d_use = st.text_area("Drug Use")
        if st.button(label='Update'):
            drug_update(d_use,d_id)
```

```
    if choice == 'Delete':
        st.subheader("Delete Drugs")
        did = st.text_area("Drug ID")
        if st.button(label="Delete"):
            drug_delete(did)


## CUSTOMERS
elif choice == "Customers":

    menu = ["View", "Update", "Delete"]
    choice = st.sidebar.selectbox("Menu", menu)
    if choice == "View":
        st.subheader("Customer Details")
        cust_result = customer_view_all_data()
        #st.write(cust_result)
        with st.expander("View All Customer Data"):
            cust_clean_df = pd.DataFrame(cust_result, columns=["Name",
"Password","Email-ID" ,"Area", "Number"])
            st.dataframe(cust_clean_df)


    if choice == 'Update':
        st.subheader("Update Customer Details")
        cust_email = st.text_area("Email")
        cust_number = st.text_area("Phone Number")
        if st.button(label='Update'):
            customer_update(cust_email,cust_number)
```

```
        if choice == 'Delete':
            st.subheader("Delete Customer")
            cust_email = st.text_area("Email")
            if st.button(label="Delete"):
                customer_delete(cust_email)



    elif choice == "Orders":
        order_menu = ["View", "Delete"]
        order_choice = st.sidebar.selectbox("Order Actions", order_menu)

        # Set session state flag for order deletion tracking
        if 'order_deleted' not in st.session_state:
            st.session_state.order_deleted = False

        if order_choice == "View":
            st.subheader("Order Details")
            # If an order was recently deleted, refresh data
            if st.session_state.order_deleted:
                order_result = order_view_all_data()  # Re-fetch the latest order data
                st.session_state.order_deleted = False  # Reset the deletion flag
            else:
                order_result = order_view_all_data()  # Load order data if no recent
deletion
```

```
        # Display the updated order data
        with st.expander("View All Order Data"):
            order_clean_df = pd.DataFrame(order_result, columns=["Name",
"Items", "Qty", "ID"])
            st.dataframe(order_clean_df)

    elif order_choice == "Delete":
        st.subheader("Delete an Order")
        order_id = st.text_input("Enter Order ID to delete")
        if st.button("Delete Order"):
            order_delete(order_id)
            st.success(f"Order with ID '{order_id}' has been deleted successfully.")

            # Set the deletion flag to true to refresh "View" on next load
            st.session_state.order_deleted = True

elif choice == "About":
    st.subheader("DBMS Mini Project")
    st.subheader("By")
    st.subheader("Nischith S (259)")
    st.subheader("Suhas B M (221)")
    st.subheader("Suhas B H (220)")
    st.subheader("Rohan P N (257)")
    st.subheader("Vishwas H T (248)")
    st.subheader("Thanman Mahesh (236)")
```

```
def getauthenicate(username, password):
    c.execute('SELECT C_Password FROM Customers WHERE C_Name = ?', (username,))
    cust_password = c.fetchall()

    # Check if any result is returned
    if cust_password and cust_password[0][0] == password:
        return True
    else:
        return False


#########################################################################

def customer(username, password):
    if getauthenicate(username, password):
        print("In Customer")
        st.title("Welcome to Pharmacy Store")

        st.subheader("Your Order Details")
        order_result = order_view_data(username)
        # st.write(cust_result)
        with st.expander("View All Order Data"):
            order_clean_df = pd.DataFrame(order_result, columns=["Name", "Items", "Qty", "ID"])
            st.dataframe(order_clean_df)
```

```
drug_result = drug_view_all_data()
print(drug_result)

st.subheader("Drug: "+drug_result[0][0])
img = Image.open('images/dolo650.jpg')
st.image(img, width=100, caption="Rs. 15/-")
dolo650 = st.slider(label="Quantity",min_value=0, max_value=5, key= 1)
st.info("When to USE: " + str(drug_result[0][2]))

st.subheader("Drug: " + drug_result[1][0])
img = Image.open('images/strepsils.JPG')
st.image(img, width=100 , caption="Rs. 10/-")
strepsils = st.slider(label="Quantity",min_value=0, max_value=5, key= 2)
st.info("When to USE: " + str(drug_result[1][2]))
   st.subheader("Drug: " + drug_result[2][0])
img = Image.open('images/vicks.JPG')
st.image(img, width=100, caption="Rs. 65/-")
vicks = st.slider(label="Quantity",min_value=0, max_value=5, key=3)
st.info("When to USE: " + str(drug_result[2][2]))


if st.button(label="Buy now"):
   O_items = ""

   if int(dolo650) > 0:
```

```
            O_items += "Dolo-650,"
        if int(strepsils) > 0:
            O_items += "Strepsils,"
        if int(vicks) > 0:
            O_items += "Vicks"
        O_Qty = str(dolo650)+str(',') + str(strepsils) + str(",") + str(vicks)

        O_id = username + "#O" + str(random.randint(0,1000000))
        #order_add_data(O_Name, O_Items,O_Qty, O_id):
        order_add_data(username, O_items, O_Qty, O_id)
if __name__ == '__main__':
    drug_create_table()
    cust_create_table()
    order_create_table()
    menu = ["Login", "SignUp","Admin"]
    choice = st.sidebar.selectbox("Menu", menu)
    if choice == "Login":
        username = st.sidebar.text_input("User Name")
        password = st.sidebar.text_input("Password", type='password')
        if st.sidebar.checkbox(label="Login"):
            customer(username, password)

    elif choice == "SignUp":
        st.subheader("Create New Account")
        cust_name = st.text_input("Name")
        cust_password = st.text_input("Password", type='password', key=1000)
```

```
        cust_password1 = st.text_input("Confirm Password", type='password',
key=1001)
        col1, col2, col3 = st.columns(3)
            with col1:
            cust_email = st.text_area("Email ID")
        with col2:
            cust_area = st.text_area("State")
        with col3:
            cust_number = st.text_area("Phone Number")


        if st.button("Signup"):
            if (cust_password == cust_password1):
                customer_add_data(cust_name,cust_password,cust_email, cust_area,
cust_number,)
                st.success("Account Created!")
                st.info("Go to Login Menu to login")
            else:
                st.warning('Password dont match')
    elif choice == "Admin":
        username = st.sidebar.text_input("User Name")
        password = st.sidebar.text_input("Password", type='password')
        # if st.sidebar.button("Login"):
        if username == 'admin' and password == 'admin':
admin()
```

SQL CODE:-

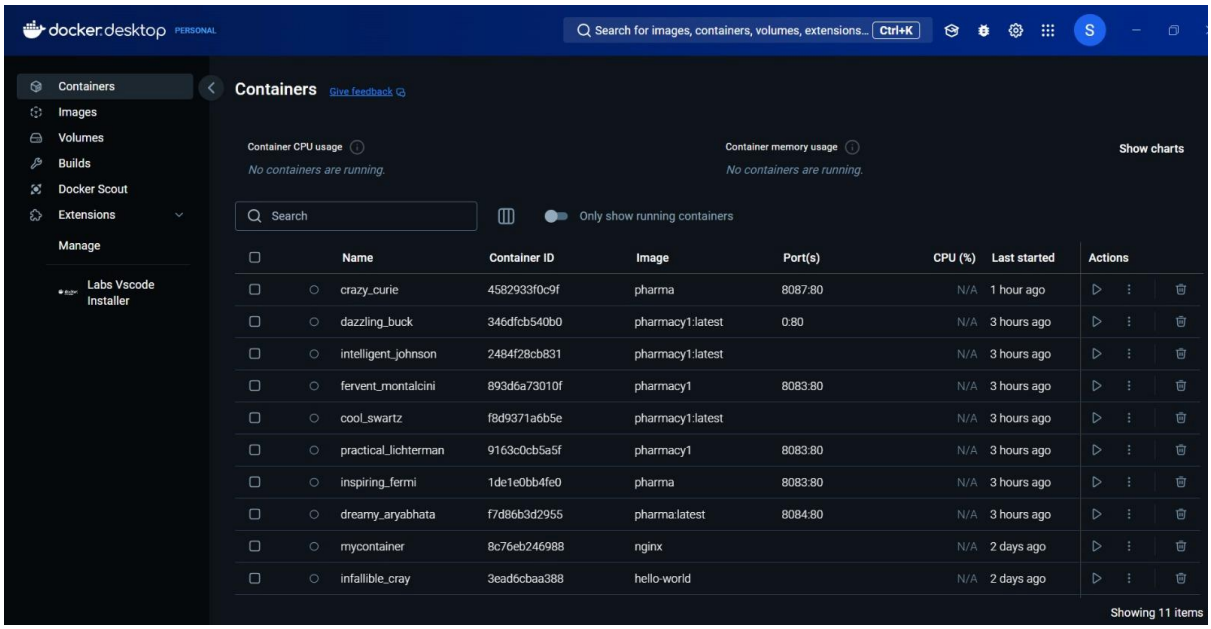
```
CREATE SCHEMA drugdatabase;


USE drugdatabase;


CREATE TABLE customer (
  uid varchar(20) NOT NULL,
  pass varchar(20) DEFAULT NULL,
  fname varchar(15) DEFAULT NULL,
  lname varchar(15) DEFAULT NULL,
  email varchar(30) DEFAULT NULL,
  address varchar(128) DEFAULT NULL,
  phno bigint DEFAULT NULL,
  PRIMARY KEY (uid)
);
CREATE TABLE seller (
  sid varchar(15) NOT NULL,
  sname varchar(20) DEFAULT NULL,
  pass varchar(20) DEFAULT NULL,
  address varchar(128) DEFAULT NULL,
  phno bigint DEFAULT NULL,
  PRIMARY KEY (sid)
);
CREATE TABLE product (
  pid varchar(15) NOT NULL,
  pname varchar(20) DEFAULT NULL,
```

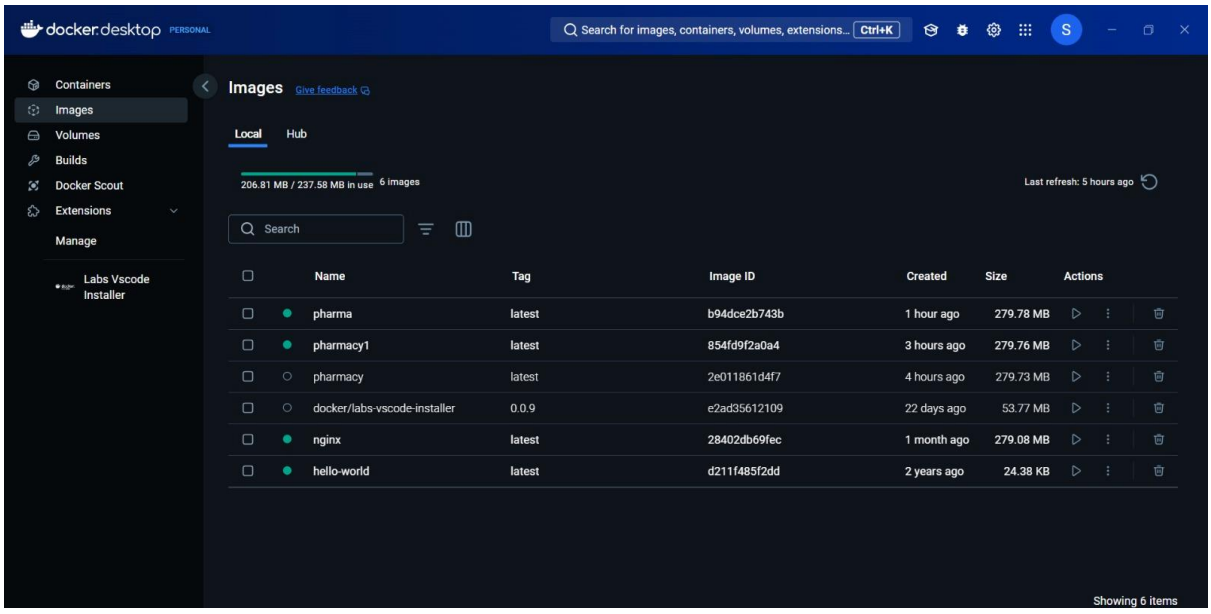
```
  manufacturer varchar(20) DEFAULT NULL,
  mfg date DEFAULT NULL,
  exp date DEFAULT NULL,
  price int DEFAULT NULL,
  PRIMARY KEY (pid),
  UNIQUE KEY pname (pname)
);
CREATE TABLE inventory (
  pid varchar(15) NOT NULL,
  pname varchar(20) DEFAULT NULL,
  quantity int unsigned DEFAULT NULL,
  sid varchar(15) NOT NULL,
  PRIMARY KEY (pid,sid),
  CONSTRAINT fk01 FOREIGN KEY (pid) REFERENCES product (pid) ON DELETE
CASCADE,
  CONSTRAINT fk02 FOREIGN KEY (pname) REFERENCES product (pname) ON
DELETE CASCADE,
  CONSTRAINT fk03 FOREIGN KEY (sid) REFERENCES seller (sid) ON DELETE
CASCADE
);
CREATE TABLE orders (
 oid int NOT NULL AUTO_INCREMENT,
 pid varchar(15) DEFAULT NULL,
 sid varchar(15) DEFAULT NULL,
 uid varchar(15) DEFAULT NULL,
 orderdatetime datetime DEFAULT NULL,
 quantity int unsigned DEFAULT NULL,
```

```
 price int unsigned DEFAULT NULL,
 PRIMARY KEY (oid),
 CONSTRAINT fk04 FOREIGN KEY (pid) REFERENCES product (pid) ON DELETE CASCADE,
 CONSTRAINT fk05 FOREIGN KEY (sid) REFERENCES seller (sid) ON DELETE CASCADE,
 CONSTRAINT fk06 FOREIGN KEY (uid) REFERENCES customer (uid) ON DELETE CASCADE
);
ALTER TABLE orders AUTO_INCREMENT=1000;
DELIMITER //
CREATE TRIGGER updatetime BEFORE INSERT ON orders FOR EACH ROW
BEGIN
   SET NEW.orderdatetime = NOW();
END//
DELIMITER ;
DELIMITER //
CREATE TRIGGER inventorytrigger AFTER INSERT ON orders
FOR EACH ROW
begin
DECLARE qnty int;
DECLARE productid varchar(20);
SELECT   pid INTO productid
FROM     orders
ORDER BY  oid DESC
LIMIT    1;
SELECT   quantity INTO qnty
```

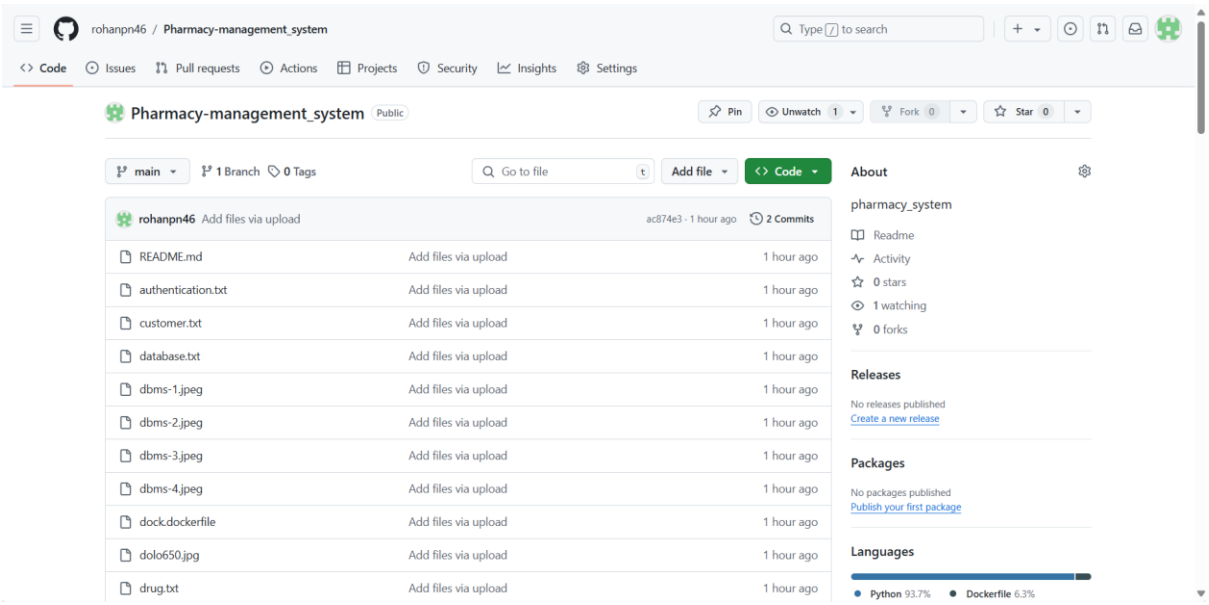
```
FROM      orders
ORDER BY  oid DESC
LIMIT     1;
UPDATE inventory
SET quantity=quantity-qnty
WHERE pid=productid;
END//
DELIMITER ;
DELIMITER //
CREATE PROCEDURE getsellerorders(IN param1 VARCHAR(20))
BEGIN
   SELECT *  FROM orders where sid=param1;
END //
DELIMITER ;
DELIMITER //
CREATE PROCEDURE getorders
(IN param1 VARCHAR(20))
BEGIN
  SELECT * FROM orders WHERE uid=param1;
END //
DELIMITER ;
```

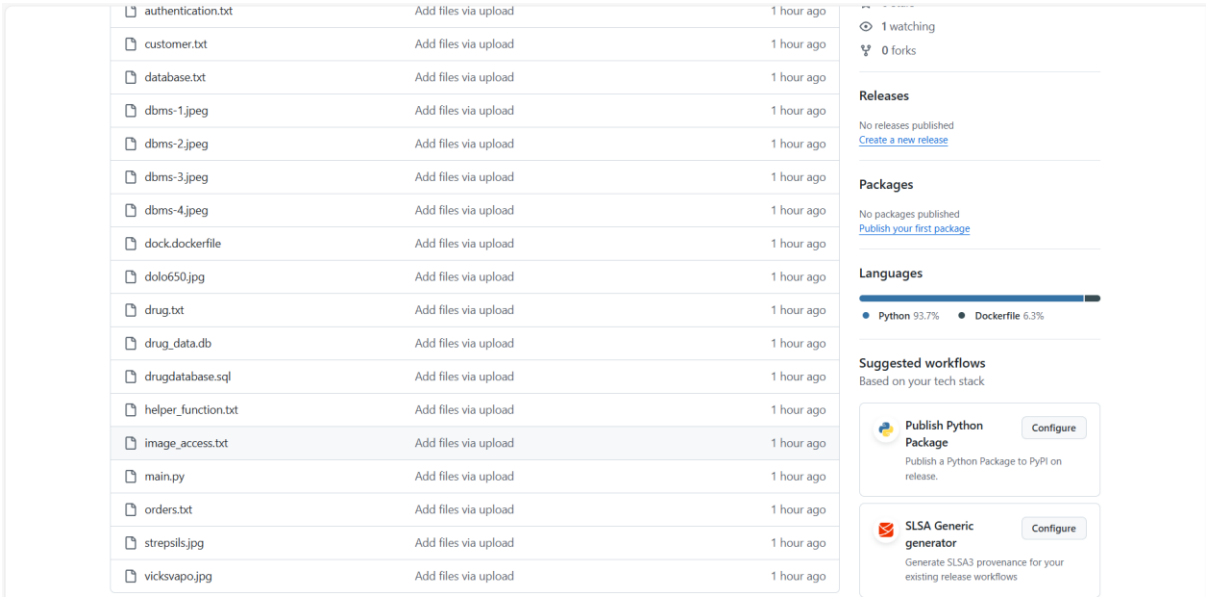
DOCKER :-


docker desktop PERSONAL
Search for images, containers, volumes, extensions... Ctrl+K
Containers
Images
Volumes
Builds
Docker Scout
Extensions
Manage
Labs Vscode Installer
Containers
Give feedback
Container CPU usage
No containers are running.
Container memory usage
No containers are running.
Show charts
Search
Only show running containers
Name Container ID Image Port(s) CPU (%) Last started Actions
crazy_curie 4582933f0c9f pharma 8087:80 N/A 1 hour ago
dazzling_buck 346dfcb540b0 pharmacy1:latest 0:80 N/A 3 hours ago
intelligent_johnson 2484f28cb831 pharmacy1:latest N/A 3 hours ago
fervent_montalcini 893d6a73010f pharmacy1 8083:80 N/A 3 hours ago
cool_swartz f8d9371a6b5e pharmacy1:latest N/A 3 hours ago
practical_lichterman 9163c0cb5a5f pharmacy1 8083:80 N/A 3 hours ago
inspiring_fermi 1de1e0bb4fe0 pharma 8083:80 N/A 3 hours ago
dreamy_aryabhata f7d86b3d2955 pharma:latest 8084:80 N/A 3 hours ago
mycontainer 8c76eb246988 nginx N/A 2 days ago
infallible_cray 3ead6cbaa388 hello-world N/A 2 days ago
Showing 11 items


docker desktop PERSONAL
Search for images, containers, volumes, extensions... Ctrl+K
Containers
Images
Volumes
Builds
Docker Scout
Extensions
Manage
Labs Vscode Installer
Images
Give feedback
Local
Hub
206.81 MB / 237.58 MB in use
6 images
Last refresh: 5 hours ago
Search
Name Tag Image ID Created Size Actions
pharma latest b94dce2b743b 1 hour ago 279.78 MB
pharmacy1 latest 854fd9f2a0a4 3 hours ago 279.76 MB
pharmacy latest 2e011861d4f7 4 hours ago 279.73 MB
docker/labs-vscode-installer 0.0.9 e2ad35612109 22 days ago 53.77 MB
nginx latest 28402db69fec 1 month ago 279.08 MB
hello-world latest d211f485f2dd 2 years ago 24.38 KB
Showing 6 items

GITHUB:-


rohanpn46 / Pharmacy-management_system
Type / to search
Code
Issues
Pull requests
Actions
Projects
Security
Insights
Settings
Pharmacy-management_system
Public
Pin
Unwatch 1
Fork 0
Star 0
main
1 Branch
0 Tags
Go to file
Add file
Code
About
pharmacy_system
Readme
Activity
0 stars
1 watching
0 forks
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package
Languages
Python 93.7%
Dockerfile 6.3%
rohanpn46 Add files via upload
ac874e3 · 1 hour ago
2 Commits
README.md
authentication.txt
customer.txt
database.txt
dbms-1.jpeg
dbms-2.jpeg
dbms-3.jpeg
dbms-4.jpeg
dock.dockerfile
dolo650.jpg
drug.txt
Add files via upload
1 hour ago


authentication.txt
customer.txt
database.txt
dbms-1.jpeg
dbms-2.jpeg
dbms-3.jpeg
dbms-4.jpeg
dock.dockerfile
dolo650.jpg
drug.txt
drug_data.db
drugdatabase.sql
helper_function.txt
image_access.txt
main.py
orders.txt
strepsils.jpg
vicksvapo.jpg
Add files via upload
1 hour ago
1 watching
0 forks
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package
Languages
Python 93.7%
Dockerfile 6.3%
Suggested workflows
Based on your tech stack
Publish Python Package
Configure
Publish a Python Package to PyPI on release.
SLSA Generic generator
Configure
Generate SLSA3 provenance for your existing release workflows

README

PROBLEM STATEMENT

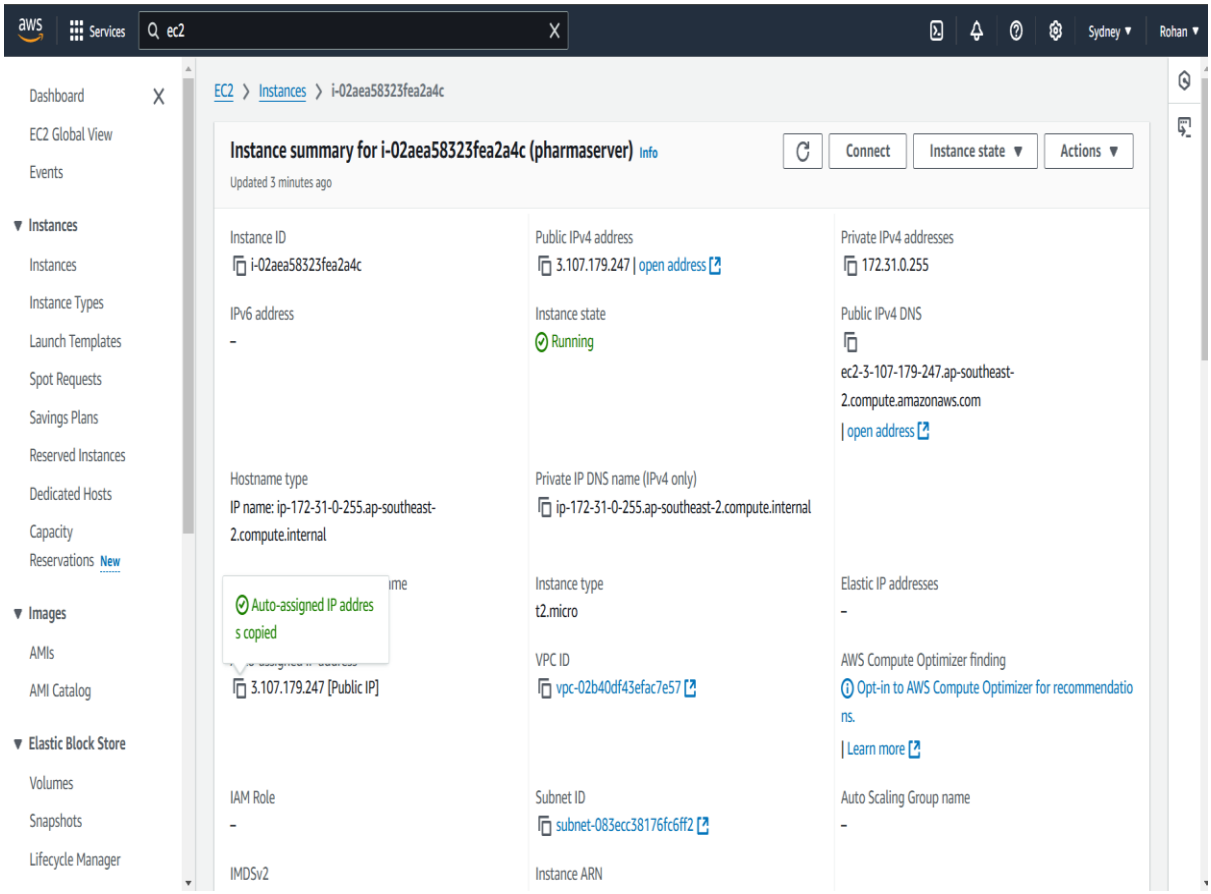
- In this era of growing demands of pharmaceutical and chemical products for health and medicine; every medical store, be it a small of medium sized businesses are running rampant towards efficient store management and rapid revenue computation.
- Instead of using the conventional style of paper-based records and documentation, electronic methods are preferred for stock maintenance, automatic updation of records while billing and reliable electronic documentation.
- The software would reduce daily effort wage and labour numbers. Crushing numbers can now be left to the system.

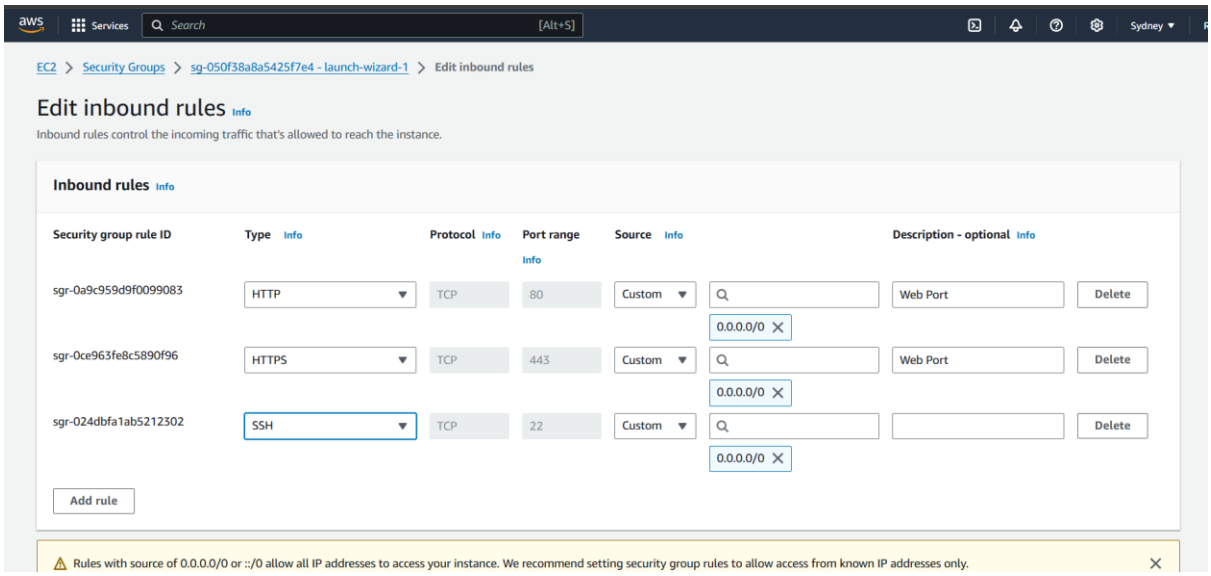
OBJECTIVE

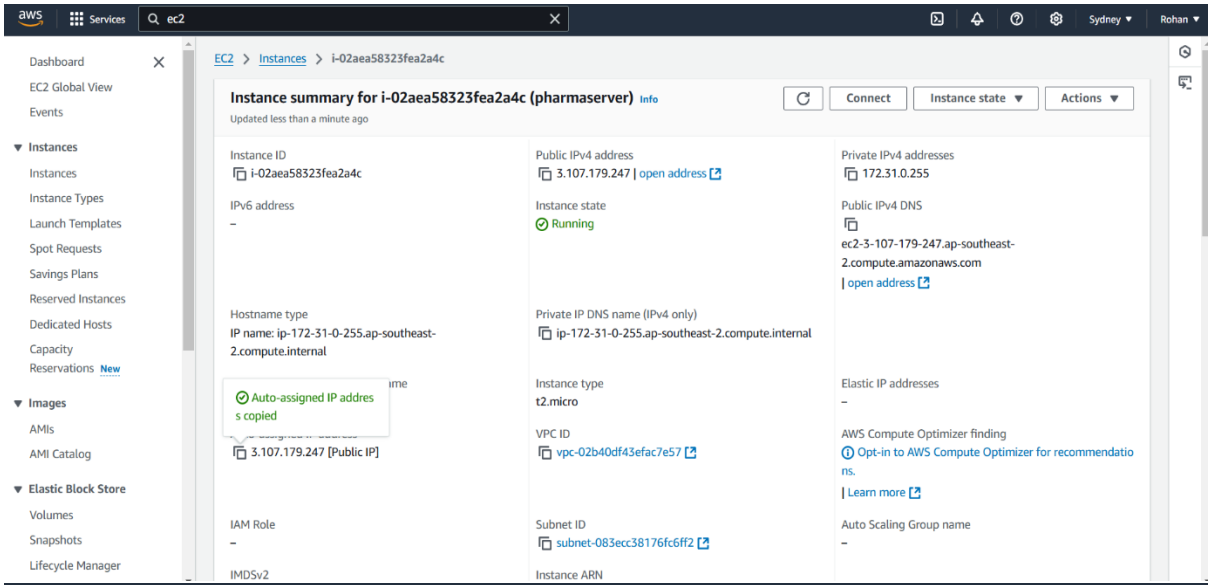
- The aim of this project is to develop a software for the effective management of the database of a pharmaceutical store.
- It would improve the accuracy and enhance safety and efficiency in the pharmaceutical store.
- This system will help the pharmacist to maintain the records of the medicines, handle user and admin details, check and renew the validity of the medicines and orders placed by the customer.

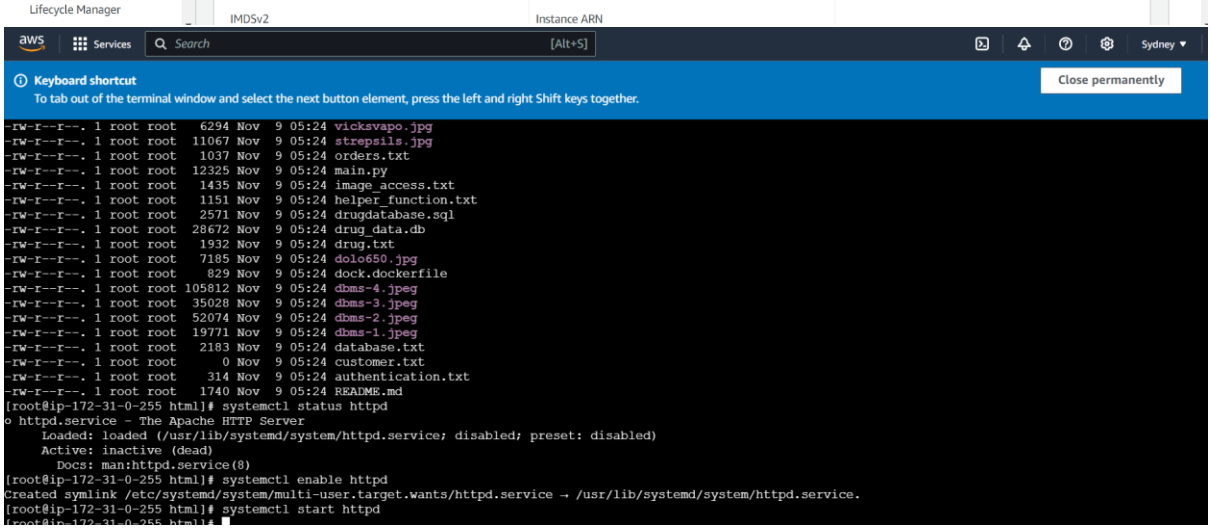
ER DIAGRAM

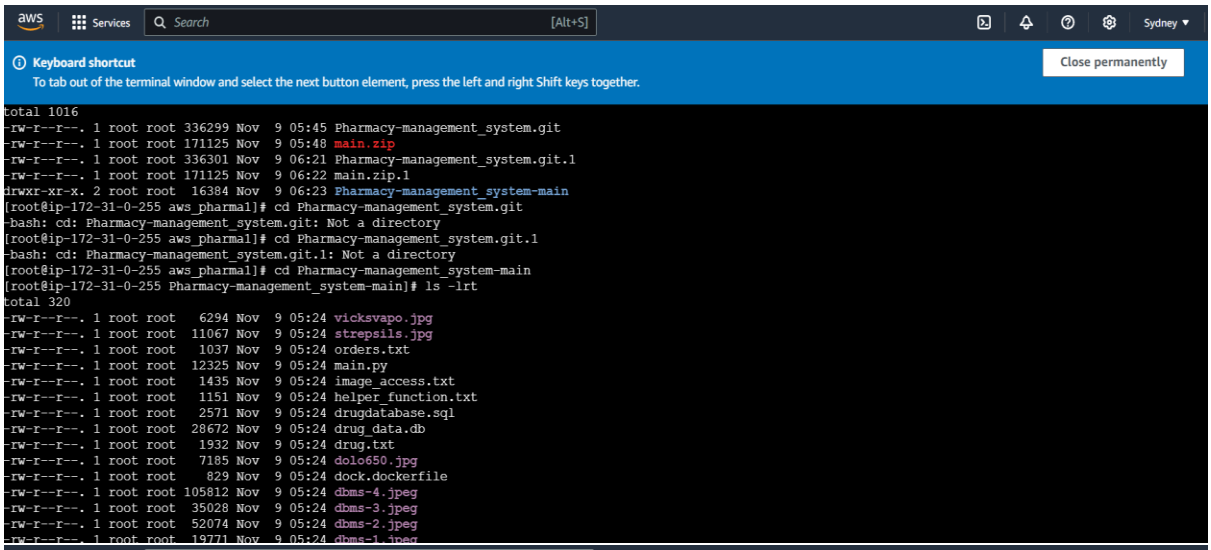
AWS:-

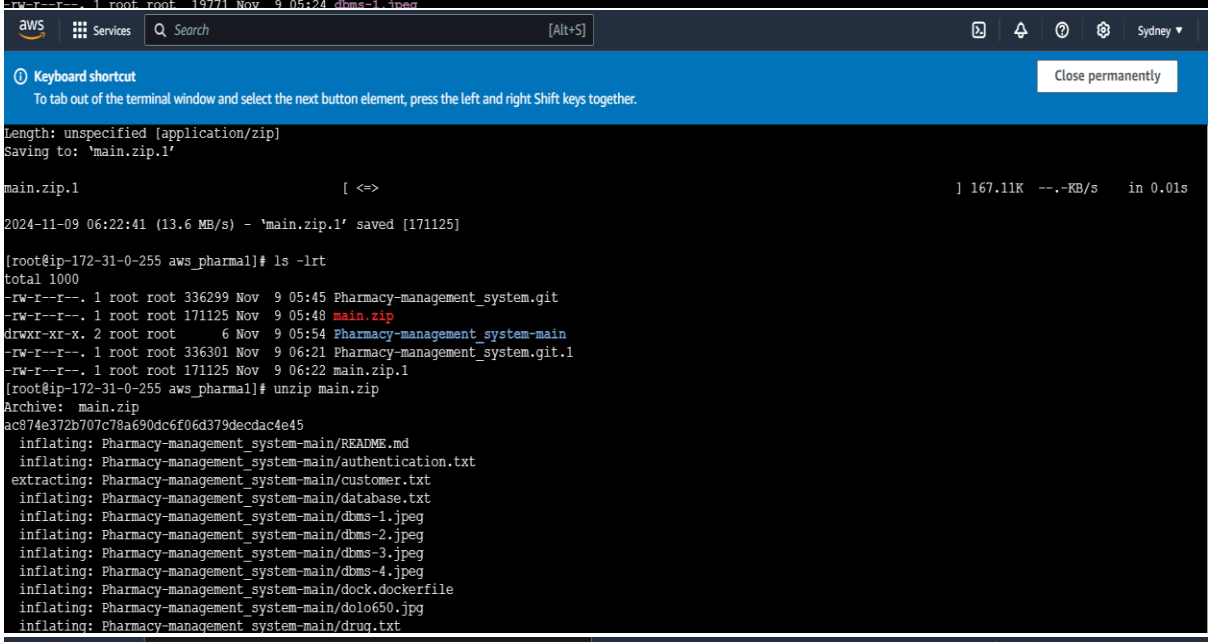


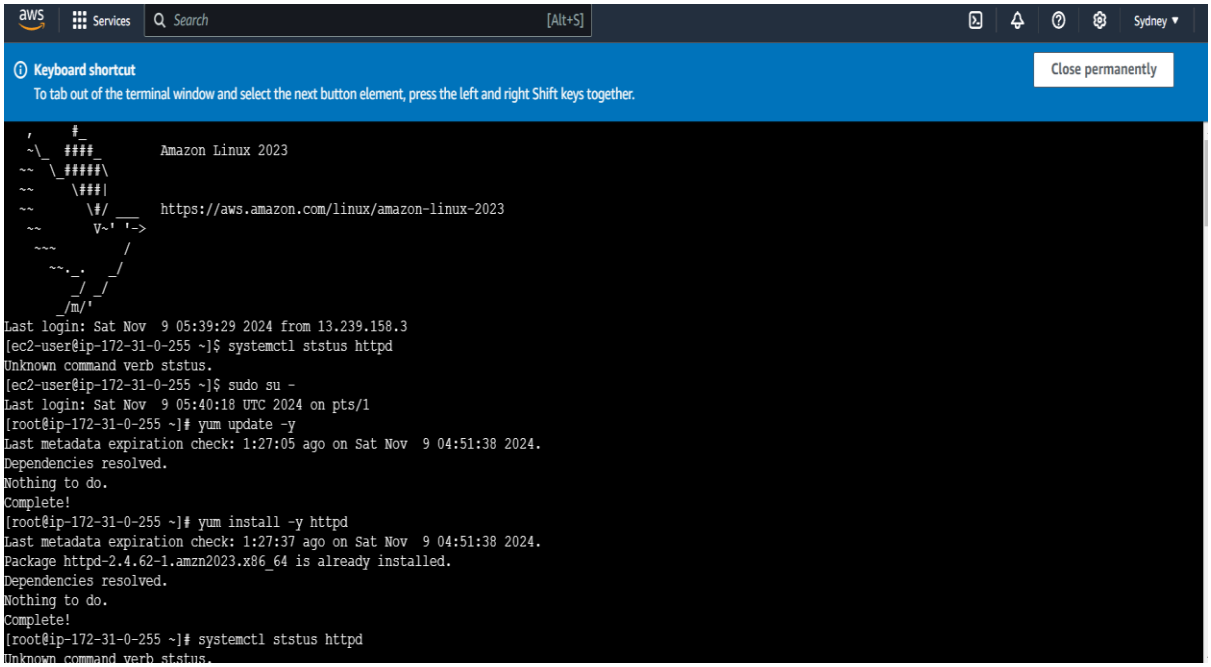

aws
Services
Search
[Alt+S]
Sydney
EC2 > Security Groups > sg-050f38a8a5425f7e4 - launch-wizard-1 > Edit inbound rules
Edit inbound rules Info
Inbound rules control the incoming traffic that's allowed to reach the instance.
Inbound rules Info
Security group rule ID
Type Info
Protocol Info
Port range Info
Source Info
Description - optional Info
sgr-0a9c959d9f0099083
HTTP
TCP
80
Custom
0.0.0.0/0
Web Port
Delete
sgr-0ce963fe8c5890f96
HTTPS
TCP
443
Custom
0.0.0.0/0
Web Port
Delete
sgr-024dbfa1ab5212302
SSH
TCP
22
Custom
0.0.0.0/0
Delete
Add rule
Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

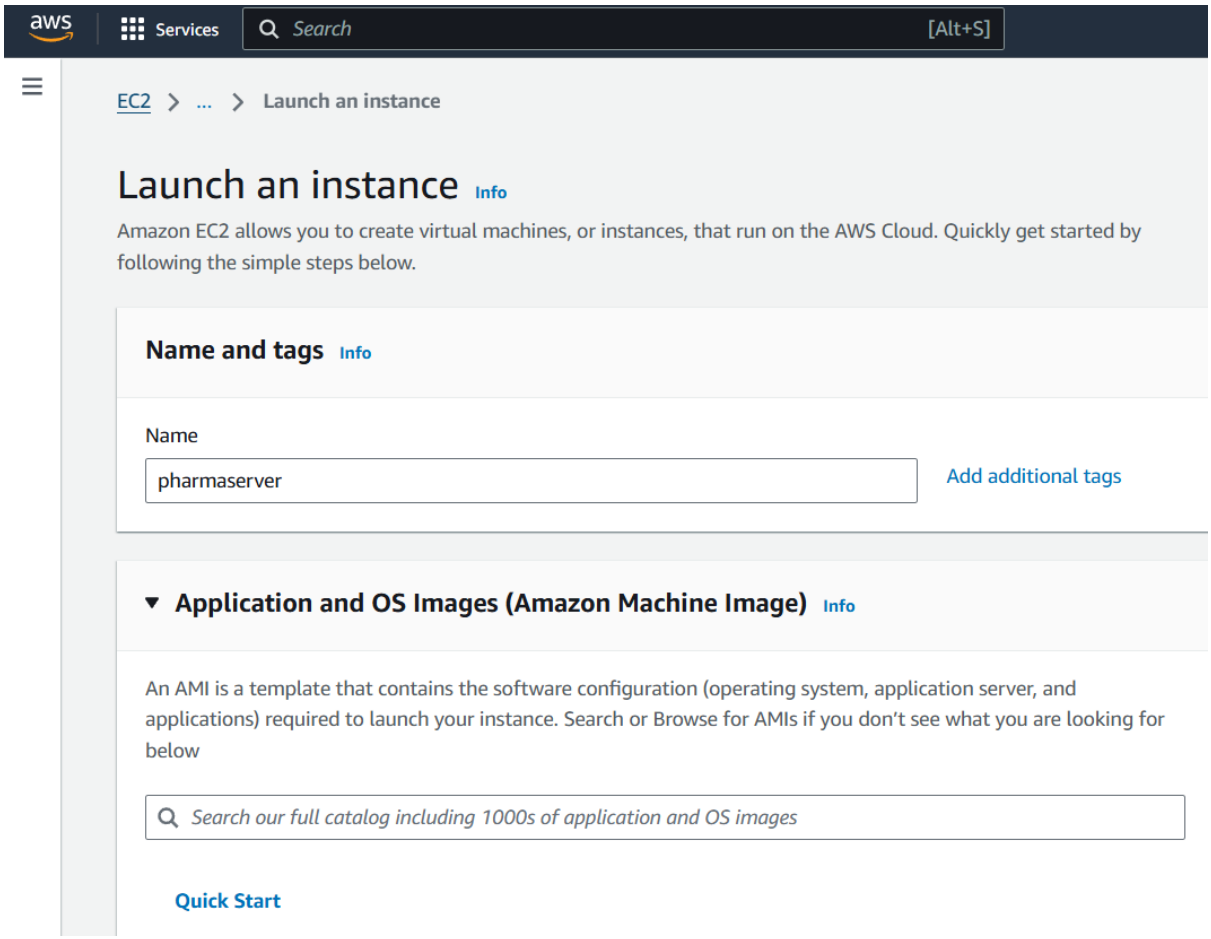

aws
Services
ec2
Sydney
Rohan
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations New
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
EC2 > Instances > i-02aea58323fea2a4c
Instance summary for i-02aea58323fea2a4c (pharmaserver) Info
Updated less than a minute ago
Connect
Instance state
Actions
Instance ID
i-02aea58323fea2a4c
Public IPv4 address
3.107.179.247 | open address
Private IPv4 addresses
172.31.0.255
IPv6 address
–
Instance state
Running
Public IPv4 DNS
ec2-3-107-179-247.ap-southeast-2.compute.amazonaws.com | open address
Hostname type
IP name: ip-172-31-0-255.ap-southeast-2.compute.internal
Private IP DNS name (IPv4 only)
ip-172-31-0-255.ap-southeast-2.compute.internal
Auto-assigned IP address copied
3.107.179.247 [Public IP]
Instance type
t2.micro
Elastic IP addresses
–
VPC ID
vpc-02b40df43efac7e57
AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role
–
Subnet ID
subnet-083ecc38176fc6ff2
Auto Scaling Group name
–
IMDSv2
Instance ARN

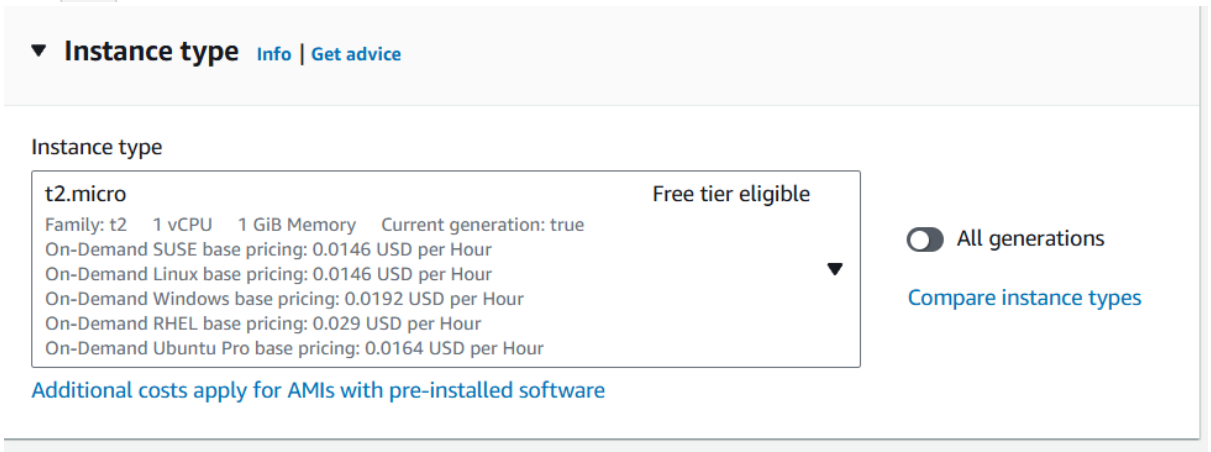

aws
Services
Search
[Alt+S]
Sydney
Keyboard shortcut
To tab out of the terminal window and select the next button element, press the left and right Shift keys together.
Close permanently
-rw-r--r--. 1 root root 6294 Nov 9 05:24 vicksvapo.jpg
-rw-r--r--. 1 root root 11067 Nov 9 05:24 strepsils.jpg
-rw-r--r--. 1 root root 1037 Nov 9 05:24 orders.txt
-rw-r--r--. 1 root root 12325 Nov 9 05:24 main.py
-rw-r--r--. 1 root root 1435 Nov 9 05:24 image_access.txt
-rw-r--r--. 1 root root 1151 Nov 9 05:24 helper_function.txt
-rw-r--r--. 1 root root 2571 Nov 9 05:24 drugdatabase.sql
-rw-r--r--. 1 root root 28672 Nov 9 05:24 drug_data.db
-rw-r--r--. 1 root root 1932 Nov 9 05:24 drug.txt
-rw-r--r--. 1 root root 7185 Nov 9 05:24 dolo650.jpg
-rw-r--r--. 1 root root 829 Nov 9 05:24 dock.dockerfile
-rw-r--r--. 1 root root 105812 Nov 9 05:24 dbms-4.jpeg
-rw-r--r--. 1 root root 35028 Nov 9 05:24 dbms-3.jpeg
-rw-r--r--. 1 root root 52074 Nov 9 05:24 dbms-2.jpeg
-rw-r--r--. 1 root root 19771 Nov 9 05:24 dbms-1.jpeg
-rw-r--r--. 1 root root 2183 Nov 9 05:24 database.txt
-rw-r--r--. 1 root root 0 Nov 9 05:24 customer.txt
-rw-r--r--. 1 root root 314 Nov 9 05:24 authentication.txt
-rw-r--r--. 1 root root 1740 Nov 9 05:24 README.md
[root@ip-172-31-0-255 html]# systemctl status httpd
o httpd.service - The Apache HTTP Server
Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
Active: inactive (dead)
Docs: man:httpd.service(8)
[root@ip-172-31-0-255 html]# systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[root@ip-172-31-0-255 html]# systemctl start httpd

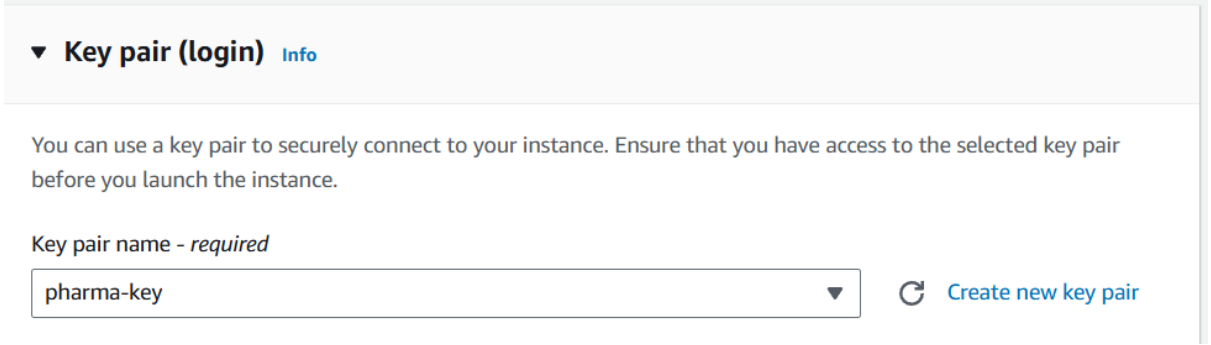

aws
Services
Search
[Alt+S]
Sydney
Keyboard shortcut
To tab out of the terminal window and select the next button element, press the left and right Shift keys together.
Close permanently
total 1016
-rw-r--r--. 1 root root 336299 Nov 9 05:45 Pharmacy-management_system.git
-rw-r--r--. 1 root root 171125 Nov 9 05:48 main.zip
-rw-r--r--. 1 root root 336301 Nov 9 06:21 Pharmacy-management_system.git.1
-rw-r--r--. 1 root root 171125 Nov 9 06:22 main.zip.1
drwxr-xr-x. 2 root root 16384 Nov 9 06:23 Pharmacy-management_system-main
[root@ip-172-31-0-255 aws_pharma1]# cd Pharmacy-management_system.git
-bash: cd: Pharmacy-management_system.git: Not a directory
[root@ip-172-31-0-255 aws_pharma1]# cd Pharmacy-management_system.git.1
-bash: cd: Pharmacy-management_system.git.1: Not a directory
[root@ip-172-31-0-255 aws_pharma1]# cd Pharmacy-management_system-main
[root@ip-172-31-0-255 Pharmacy-management_system-main]# ls -lrt
total 320
-rw-r--r--. 1 root root 6294 Nov 9 05:24 vicksvapo.jpg
-rw-r--r--. 1 root root 11067 Nov 9 05:24 strepsils.jpg
-rw-r--r--. 1 root root 1037 Nov 9 05:24 orders.txt
-rw-r--r--. 1 root root 12325 Nov 9 05:24 main.py
-rw-r--r--. 1 root root 1435 Nov 9 05:24 image_access.txt
-rw-r--r--. 1 root root 1151 Nov 9 05:24 helper_function.txt
-rw-r--r--. 1 root root 2571 Nov 9 05:24 drugdatabase.sql
-rw-r--r--. 1 root root 28672 Nov 9 05:24 drug_data.db
-rw-r--r--. 1 root root 1932 Nov 9 05:24 drug.txt
-rw-r--r--. 1 root root 7185 Nov 9 05:24 dolo650.jpg
-rw-r--r--. 1 root root 829 Nov 9 05:24 dock.dockerfile
-rw-r--r--. 1 root root 105812 Nov 9 05:24 dbms-4.jpeg
-rw-r--r--. 1 root root 35028 Nov 9 05:24 dbms-3.jpeg
-rw-r--r--. 1 root root 52074 Nov 9 05:24 dbms-2.jpeg
-rw-r--r--. 1 root root 19771 Nov 9 05:24 dbms-1.jpeg

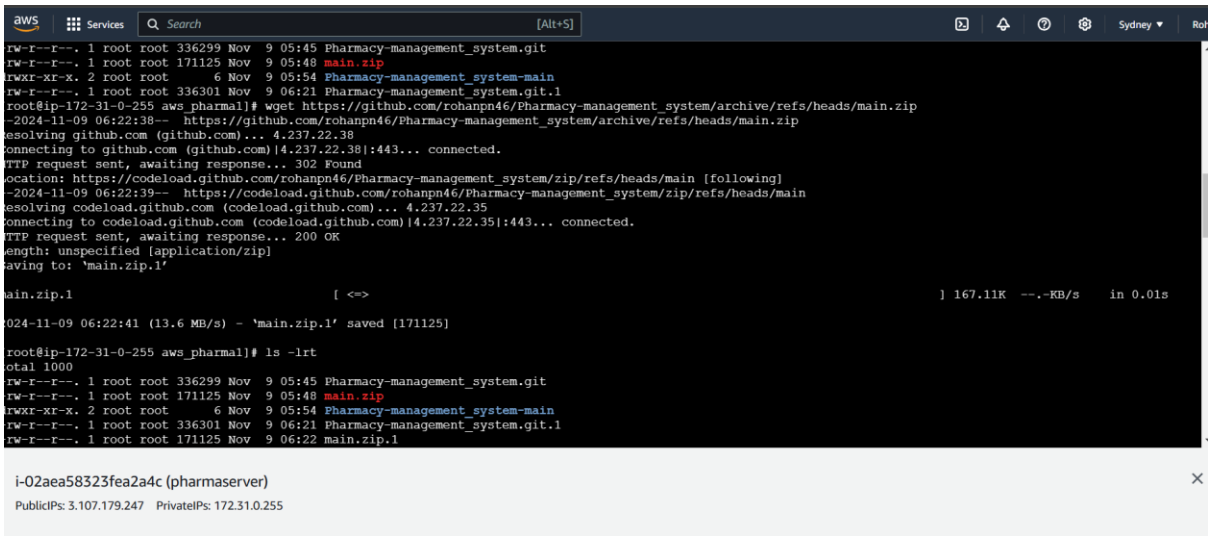

aws
Services
Search
[Alt+S]
Sydney
Keyboard shortcut
To tab out of the terminal window and select the next button element, press the left and right Shift keys together.
Close permanently
Length: unspecified [application/zip]
Saving to: 'main.zip.1'
main.zip.1 [<=>] 167.11K --.-KB/s in 0.01s
2024-11-09 06:22:41 (13.6 MB/s) - 'main.zip.1' saved [171125]
[root@ip-172-31-0-255 aws_pharma1]# ls -lrt
total 1000
-rw-r--r--. 1 root root 336299 Nov 9 05:45 Pharmacy-management_system.git
-rw-r--r--. 1 root root 171125 Nov 9 05:48 main.zip
drwxr-xr-x. 2 root root 6 Nov 9 05:54 Pharmacy-management_system-main
-rw-r--r--. 1 root root 336301 Nov 9 06:21 Pharmacy-management_system.git.1
-rw-r--r--. 1 root root 171125 Nov 9 06:22 main.zip.1
[root@ip-172-31-0-255 aws_pharma1]# unzip main.zip
Archive: main.zip
ac874e372b707c78a690dc6f06d379decdac4e45
inflating: Pharmacy-management_system-main/README.md
inflating: Pharmacy-management_system-main/authentication.txt
extracting: Pharmacy-management_system-main/customer.txt
inflating: Pharmacy-management_system-main/database.txt
inflating: Pharmacy-management_system-main/dbms-1.jpeg
inflating: Pharmacy-management_system-main/dbms-2.jpeg
inflating: Pharmacy-management_system-main/dbms-3.jpeg
inflating: Pharmacy-management_system-main/dbms-4.jpeg
inflating: Pharmacy-management_system-main/dock.dockerfile
inflating: Pharmacy-management_system-main/dolo650.jpg
inflating: Pharmacy-management_system-main/drug.txt


aws
Services
Search
[Alt+S]
Sydney
Keyboard shortcut
To tab out of the terminal window and select the next button element, press the left and right Shift keys together.
Close permanently
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023
Last login: Sat Nov 9 05:39:29 2024 from 13.239.158.3
[ec2-user@ip-172-31-0-255 ~]$ systemctl ststus httpd
Unknown command verb ststus.
[ec2-user@ip-172-31-0-255 ~]$ sudo su -
Last login: Sat Nov 9 05:40:18 UTC 2024 on pts/1
[root@ip-172-31-0-255 ~]# yum update -y
Last metadata expiration check: 1:27:05 ago on Sat Nov 9 04:51:38 2024.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-0-255 ~]# yum install -y httpd
Last metadata expiration check: 1:27:37 ago on Sat Nov 9 04:51:38 2024.
Package httpd-2.4.62-1.amzn2023.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-0-255 ~]# systemctl ststus httpd
Unknown command verb ststus.

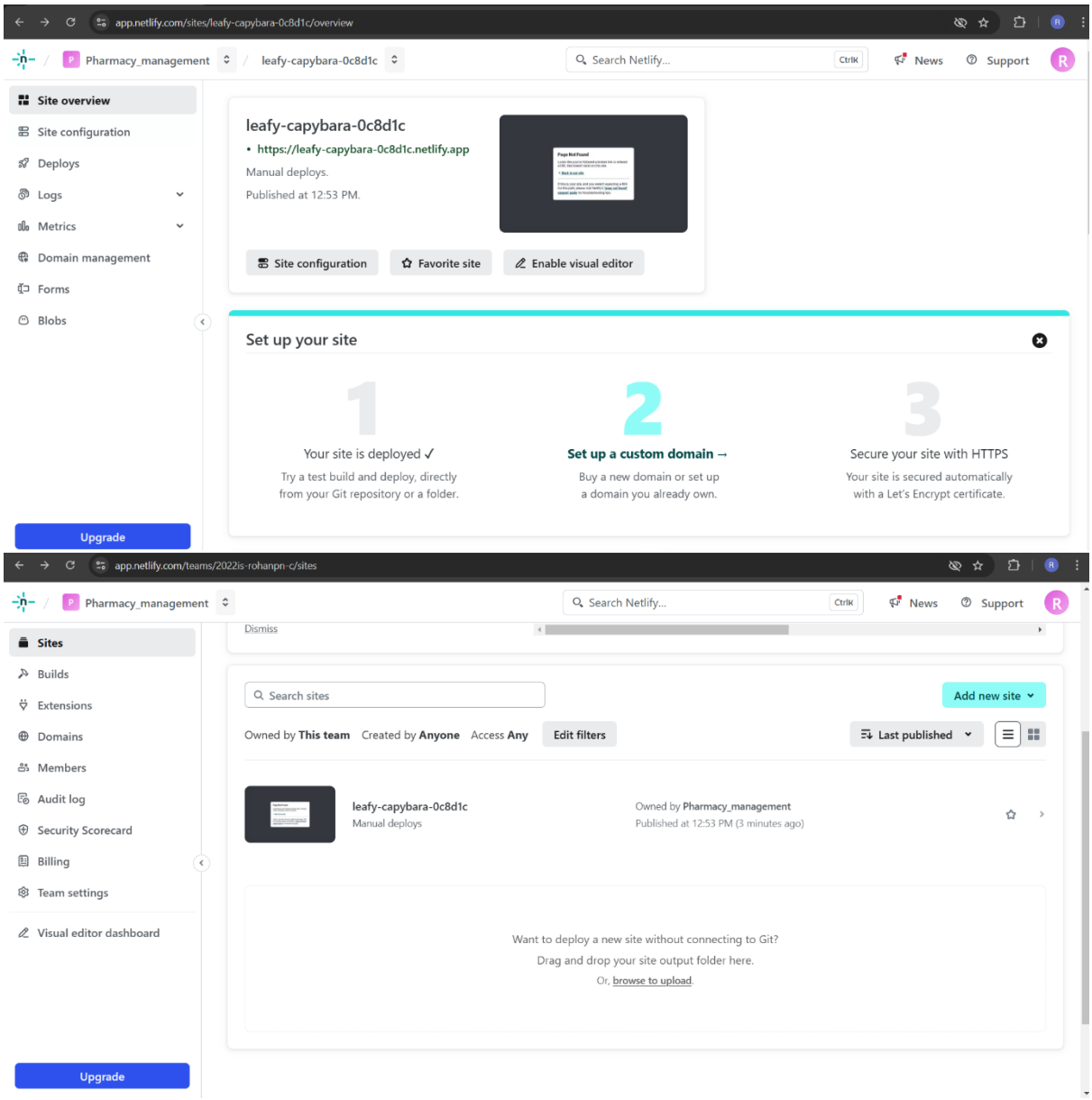

aws
Services
Search
[Alt+S]
EC2 > ... > Launch an instance
Launch an instance Info
Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.
Name and tags Info
Name
pharmaserver
Add additional tags
Application and OS Images (Amazon Machine Image) Info
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below
Search our full catalog including 1000s of application and OS images
Quick Start


Instance type Info | Get advice
Instance type
t2.micro
Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand SUSE base pricing: 0.0146 USD per Hour
On-Demand Linux base pricing: 0.0146 USD per Hour
On-Demand Windows base pricing: 0.0192 USD per Hour
On-Demand RHEL base pricing: 0.029 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0164 USD per Hour
All generations
Compare instance types
Additional costs apply for AMIs with pre-installed software


Key pair (login) Info
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.
Key pair name - required
pharma-key
Create new key pair


aws
Services
Search
[Alt+S]
Sydney
rw-r--r--. 1 root root 336299 Nov 9 05:45 Pharmacy-management_system.git
rw-r--r--. 1 root root 171125 Nov 9 05:48 main.zip
rwxr-xr-x. 2 root root 6 Nov 9 05:54 Pharmacy-management_system-main
rw-r--r--. 1 root root 336301 Nov 9 06:21 Pharmacy-management_system.git.1
root@ip-172-31-0-255 aws_pharma1]# wget https://github.com/rohanpn46/Pharmacy-management_system/archive/refs/heads/main.zip
-2024-11-09 06:22:38-- https://github.com/rohanpn46/Pharmacy-management_system/archive/refs/heads/main.zip
esolving github.com (github.com)... 4.237.22.38
onnecting to github.com (github.com)|4.237.22.38|:443... connected.
TTP request sent, awaiting response... 302 Found
ocation: https://codeload.github.com/rohanpn46/Pharmacy-management_system/zip/refs/heads/main [following]
-2024-11-09 06:22:39-- https://codeload.github.com/rohanpn46/Pharmacy-management_system/zip/refs/heads/main
esolving codeload.github.com (codeload.github.com)... 4.237.22.35
onnecting to codeload.github.com (codeload.github.com)|4.237.22.35|:443... connected.
TTP request sent, awaiting response... 200 OK
ength: unspecified [application/zip]
aving to: 'main.zip.1'
ain.zip.1 [<=>] 167.11K --.-KB/s in 0.01s
024-11-09 06:22:41 (13.6 MB/s) - 'main.zip.1' saved [171125]
root@ip-172-31-0-255 aws_pharma1]# ls -lrt
otal 1000
rw-r--r--. 1 root root 336299 Nov 9 05:45 Pharmacy-management_system.git
rw-r--r--. 1 root root 171125 Nov 9 05:48 main.zip
rwxr-xr-x. 2 root root 6 Nov 9 05:54 Pharmacy-management_system-main
rw-r--r--. 1 root root 336301 Nov 9 06:21 Pharmacy-management_system.git.1
rw-r--r--. 1 root root 171125 Nov 9 06:22 main.zip.1
i-02aea58323fea2a4c (pharmaserver)
PublicIPs: 3.107.179.247 PrivateIPs: 172.31.0.255

NETLIFY:-


app.netlify.com/sites/leafy-capybara-0c8d1c/overview
Pharmacy_management
leafy-capybara-0c8d1c
Search Netlify...
Ctrlk
News
Support
Site overview
Site configuration
Deploys
Logs
Metrics
Domain management
Forms
Blobs
Upgrade
leafy-capybara-0c8d1c
https://leafy-capybara-0c8d1c.netlify.app
Manual deploys.
Published at 12:53 PM.
Site configuration
Favorite site
Enable visual editor
Set up your site
1
Your site is deployed ✓
Try a test build and deploy, directly from your Git repository or a folder.
2
Set up a custom domain →
Buy a new domain or set up a domain you already own.
3
Secure your site with HTTPS
Your site is secured automatically with a Let's Encrypt certificate.
app.netlify.com/teams/2022is-rohanpn-c/sites
Pharmacy_management
Search Netlify...
Ctrlk
News
Support
Dismiss
Sites
Builds
Extensions
Domains
Members
Audit log
Security Scorecard
Billing
Team settings
Visual editor dashboard
Upgrade
Search sites
Add new site
Owned by This team
Created by Anyone
Access Any
Edit filters
Last published
leafy-capybara-0c8d1c
Manual deploys
Owned by Pharmacy_management
Published at 12:53 PM (3 minutes ago)
Want to deploy a new site without connecting to Git?
Drag and drop your site output folder here.
Or, browse to upload.