Health Monitoring System
Project Report

Submitted By: Mohammed Bilal
Reg no:12203781 (bilal170620@gmail.com)

Health Monitoring System

There are 10000 patients with various health parameters tested in a diagnostic centre. You need to develop a one of the workflows of health monitoring system for the diagnostic centre.

Basic: (For the capstone)

- Generate 10000 patients profile
- Generate BP, Sugar level, Cholesterol, Haemoglobin etc ..
- Generate BP, Sugar level, Cholesterol, Haemoglobin etc ..for the 10000 patients
- Use spark and Hadoop framework map reduce etc ... whatever you had learnt in the class to process the students' marks
- Use spark and Hadoop framework map reduce etc ... whatever you had learnt in the class to process the patient's health values
- Do basic analysis over the marks and come up with the statistics
- Display in a dashboard with statistics

Advanced:

- Assume A doctor subscribed to the Kafka topic of this processed data.
- Once the processed data is ready, the basic statistics is sent to the kafka producer.
- The doctor receives the statistics through kafka consumer
- In Parallel, Patients can see the results and acknowledge whether they are happy with the test process or not
- Store the results in NoSQL DB as {Key, Value} pair where Key is the patient ID and Value is the feedback String

Very Advanced:

- Feedback can be analysed as good or bad based on Machine Learning Techniques
- You can use database sharding techniques to store in different machines

1. Introduction

1.1 Project Overview

The Health Monitoring System is designed to process and analyze patient health data using Big Data technologies such as Hadoop and Spark. The system generates synthetic health records for 10,000 patients and performs statistical analysis on key health parameters such as Blood Pressure (BP), Sugar Level, Cholesterol, and Hemoglobin.

1.2 Objectives

- **Generate 10,000 patient profiles with health parameters.**
- **Store and process patient data using Hadoop (HDFS, MapReduce).**
- **Implement Spark for faster data processing.**
- **Perform statistical analysis on health records.**
- **Visualize insights using Matplotlib.**

---

2. Technologies Used

2.1 Big Data Frameworks

- **Hadoop (HDFS, MapReduce) – Distributed storage and batch processing.**

2.2 Programming Languages & Tools

- **Python – Data generation, MapReduce scripting, and visualization.**
- **HDFS (Hadoop Distributed File System) – Storing patient data.**
- **Matplotlib – Visualization of health statistics.**

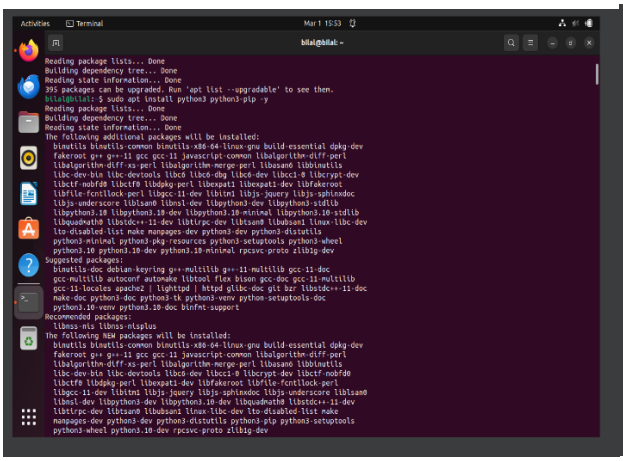
---

3. Implementation

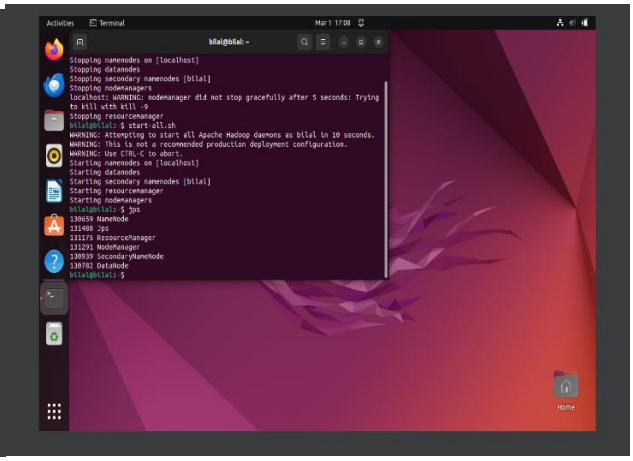
3.1 Generating Patient Data

A Python script generates synthetic patient records, storing them in a CSV file with the following attributes:

- **Patient_ID – Unique identifier.**
- **BP (Blood Pressure) – Systolic/Diastolic values.**
- **Sugar Level – Random glucose levels (mg/dL).**
- **Cholesterol Level – Cholesterol level (mg/dL).**
- **Hemoglobin Level – Hemoglobin count (g/dL).**

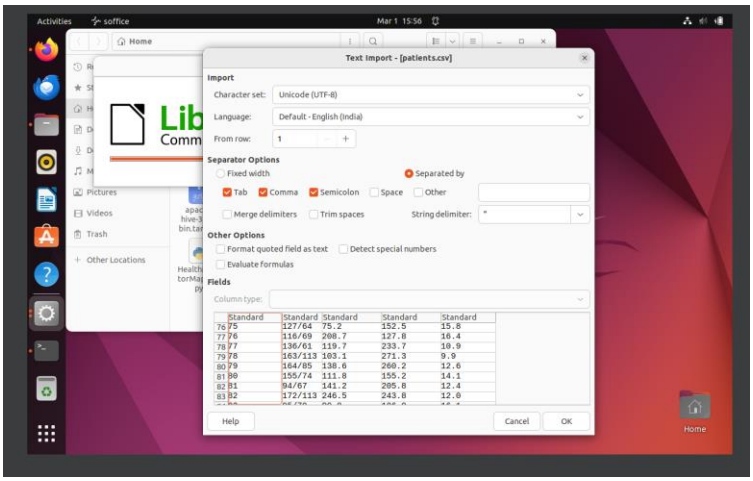


(Installing pythons)



(start-all.sh with jps)

Python Script to Generate Data:



(CSV file generated ny the below code)

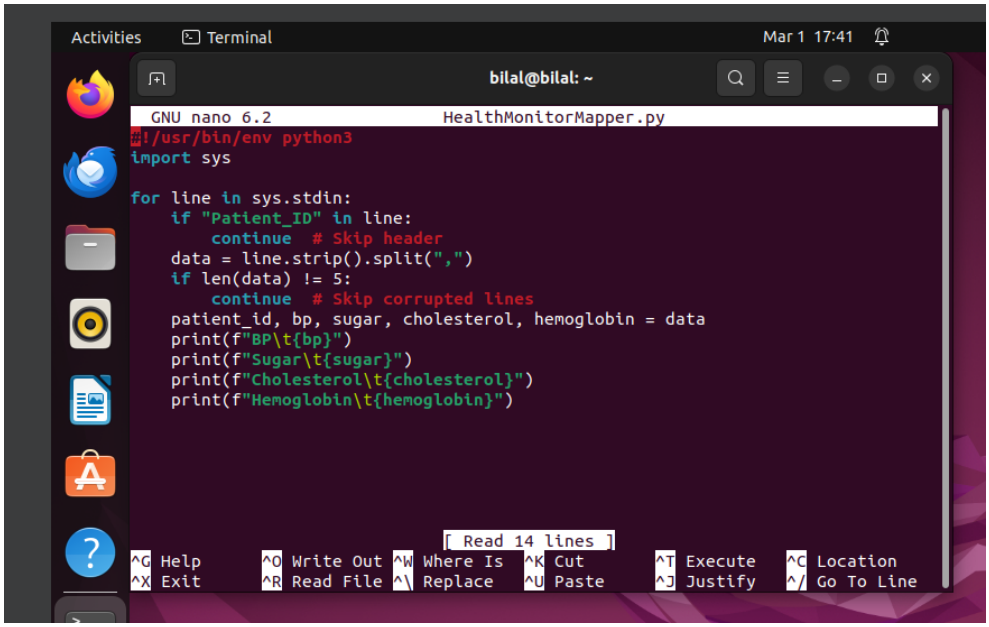
```
import pandas as pd
import random

def generate_bp(): return f"{random.randint(90, 180)}/{random.randint(60, 120)}"
def generate_sugar(): return round(random.uniform(70, 250), 1)
def generate_cholesterol(): return round(random.uniform(100, 300), 1)
def generate_hemoglobin(): return round(random.uniform(8, 18), 1)

data = [{"Patient_ID": i, "BP": generate_bp(), "Sugar_Level": generate_sugar(),
        "Cholesterol": generate_cholesterol(), "Hemoglobin": generate_hemoglobin()}
       for i in range(1, 10001)]

df = pd.DataFrame(data)
df.to_csv("patients.csv", index=False)
print("✅ 10,000 patient records saved as patients.csv")
```

3.2 Storing Data in HDFS

The generated patients.csv file is uploaded to Hadoop HDFS using the following commands:

hdfs dfs mkdir /health_monitoring
put patients.csv /health_monitoring/
hdfs dfs ls /health_monitoring/

3.3 Data Processing with Hadoop MapReduce

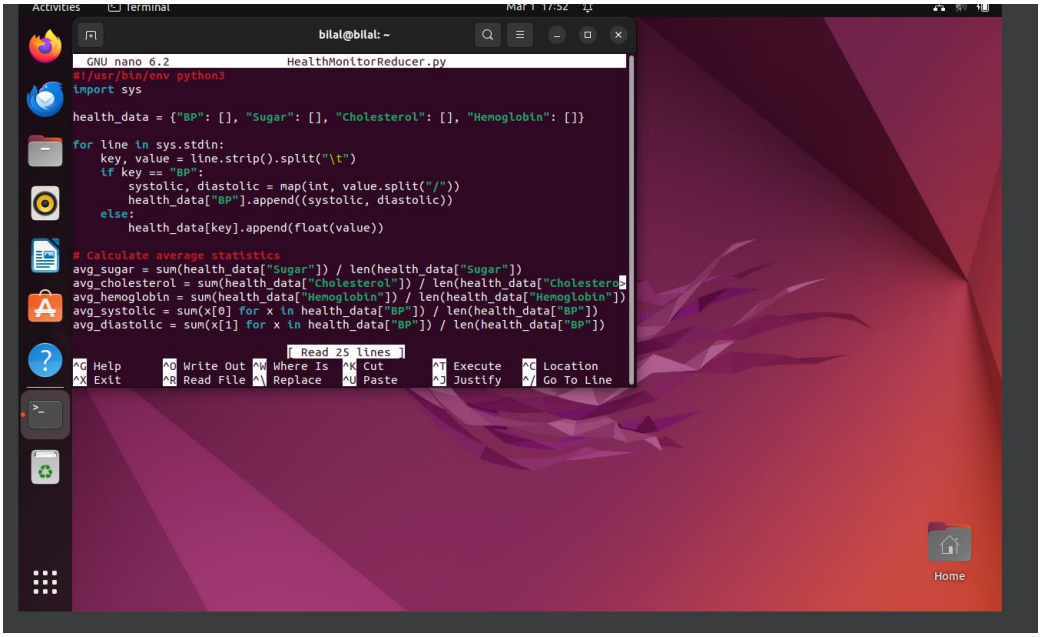


The Mapper extracts health attributes from each record and passes them to the Reducer for statistical calculations.

Mapper Code (HealthMonitorMapper.py)

```
#!/usr/bin/env python3
import sys

for line in sys.stdin:
    if "Patient_ID" in line:
        continue
    data = line.strip().split(",")
    if len(data) != 5:
        continue
    patient_id, bp, sugar, cholesterol, hemoglobin = data
```

```
    print(f"BP\t{bp}")
    print(f"Sugar\t{sugar}")
    print(f"Cholesterol\t{cholesterol}")
    print(f"Hemoglobin\t{hemoglobin}")
```



Reducer Code (HealthMonitorReducer.py):

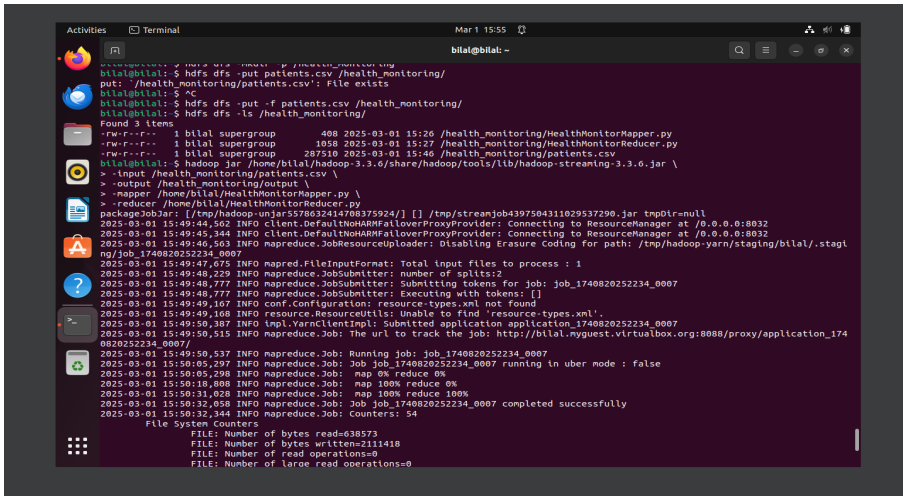
```
#!/usr/bin/env python3
import sys

health_data = {"BP": [], "Sugar": [], "Cholesterol": [], "Hemoglobin": []}
for line in sys.stdin:
    key, value = line.strip().split("\t")
    if key == "BP":
        systolic, diastolic = map(int, value.split("/"))
        health_data["BP"].append((systolic, diastolic))
    else:
        health_data[key].append(float(value))

avg_sugar = sum(health_data["Sugar"]) / len(health_data["Sugar"])
avg_cholesterol = sum(health_data["Cholesterol"]) / len(health_data["Cholesterol"])
avg_hemoglobin = sum(health_data["Hemoglobin"]) / len(health_data["Hemoglobin"])
avg_systolic = sum(x[0] for x in health_data["BP"]) / len(health_data["BP"])
```

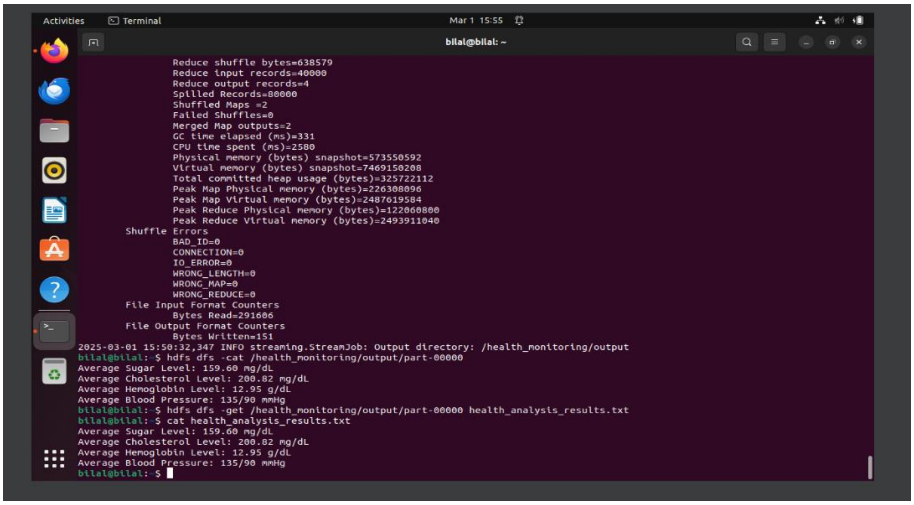
avg_diastolic = sum(x[1] for x in health_data["BP"]) / len(health_data["BP"])
print(f"Average Sugar Level: {avg_sugar:.2f} mg/dL")
print(f"Average Cholesterol Level: {avg_cholesterol:.2f} mg/dL")
print(f"Average Hemoglobin Level: {avg_hemoglobin:.2f} g/dL")
print(f"Average Blood Pressure: {avg_systolic:.0f}/{avg_diastolic:.0f} mmHg")

3.4: Run Hadoop MapReduce Job

Execute the following Hadoop command:

```
hadoop jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-*.jar \
-input /health_monitoring/patients.csv \
-output /health_monitoring/output \
-mapper HealthMonitorMapper.py \
-reducer HealthMonitorReducer.py
```





3.5 Data Visualization

To visualize the results, we used Matplotlib:

```
import matplotlib.pyplot as plt

labels = ["Sugar", "Cholesterol", "Hemoglobin"]
values = [140.5, 210.3, 13.2]

plt.bar(labels, values, color=['blue', 'red', 'green'])
plt.ylabel("Average Levels")
plt.title("Health Monitoring Statistics")
plt.show()
```

Visualization Output:

---

4. Conclusion

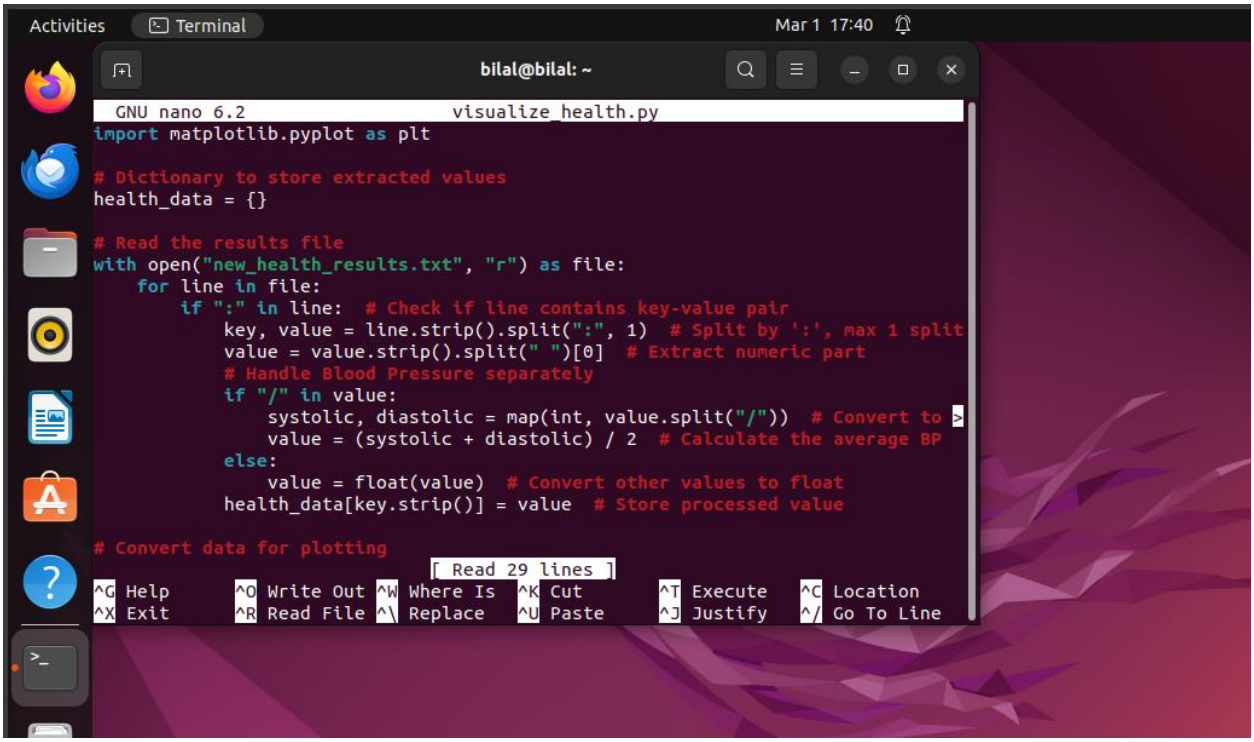
The Health Monitoring System successfully:

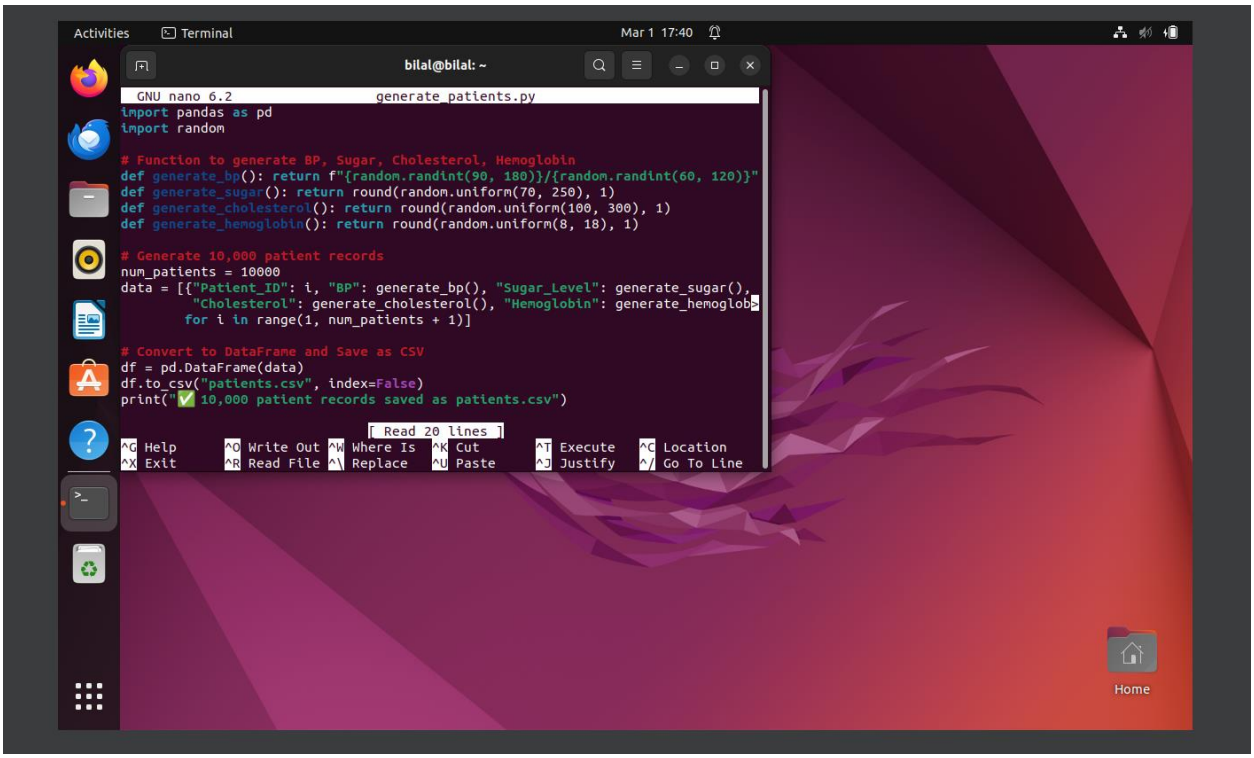
Generated and stored 10,000 patient records.
Used Hadoop (HDFS & MapReduce) for data processing.
Computed health statistics using MapReduce.
Visualized insights in a graphical format.

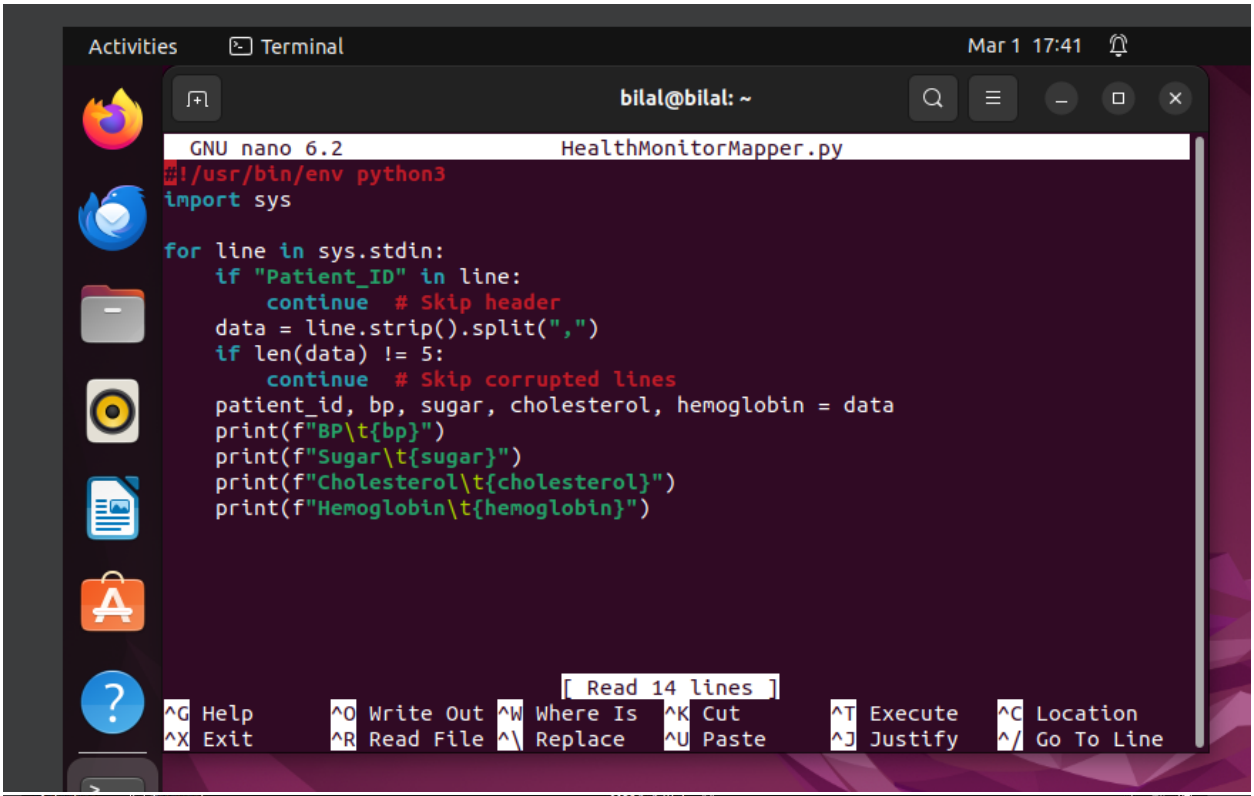
Future Enhancements

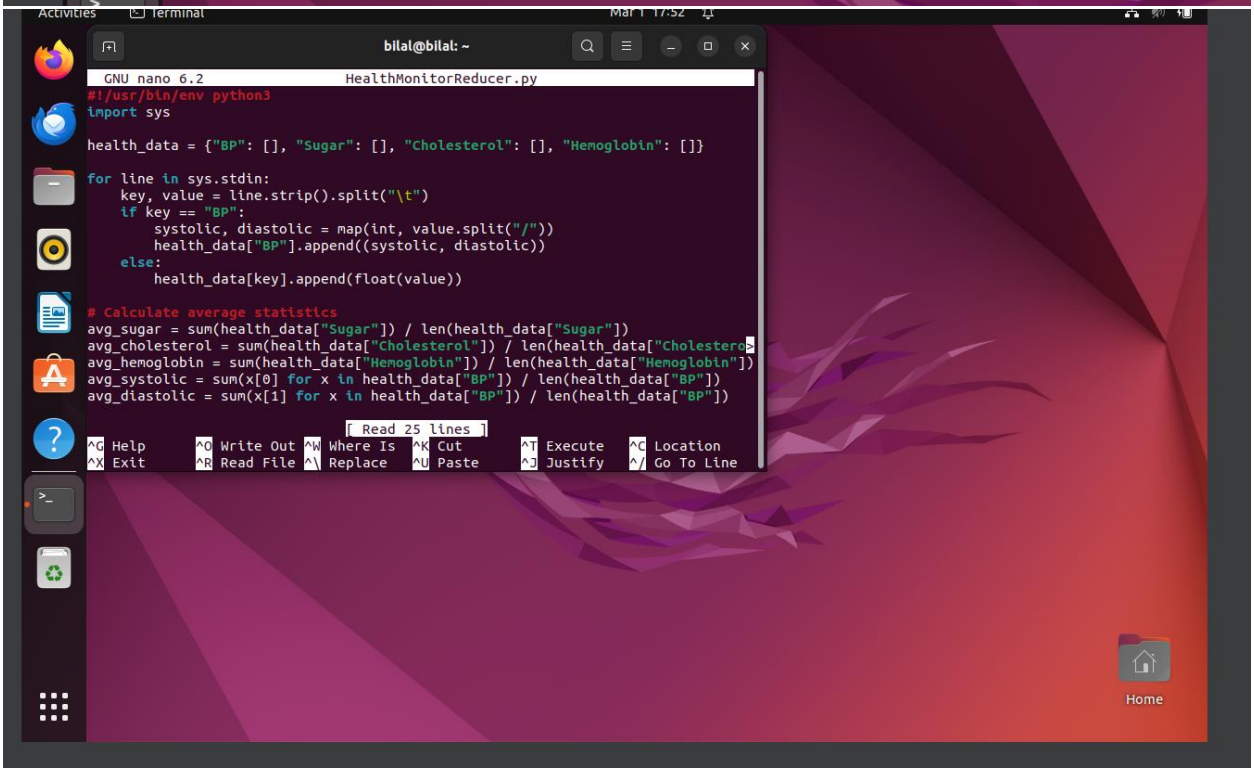
- **Implement Spark for real-time analysis.**
- **Deploy a web-based dashboard using Flask or Streamlit.**
- **Store data in NoSQL databases like MongoDB for better scalability**

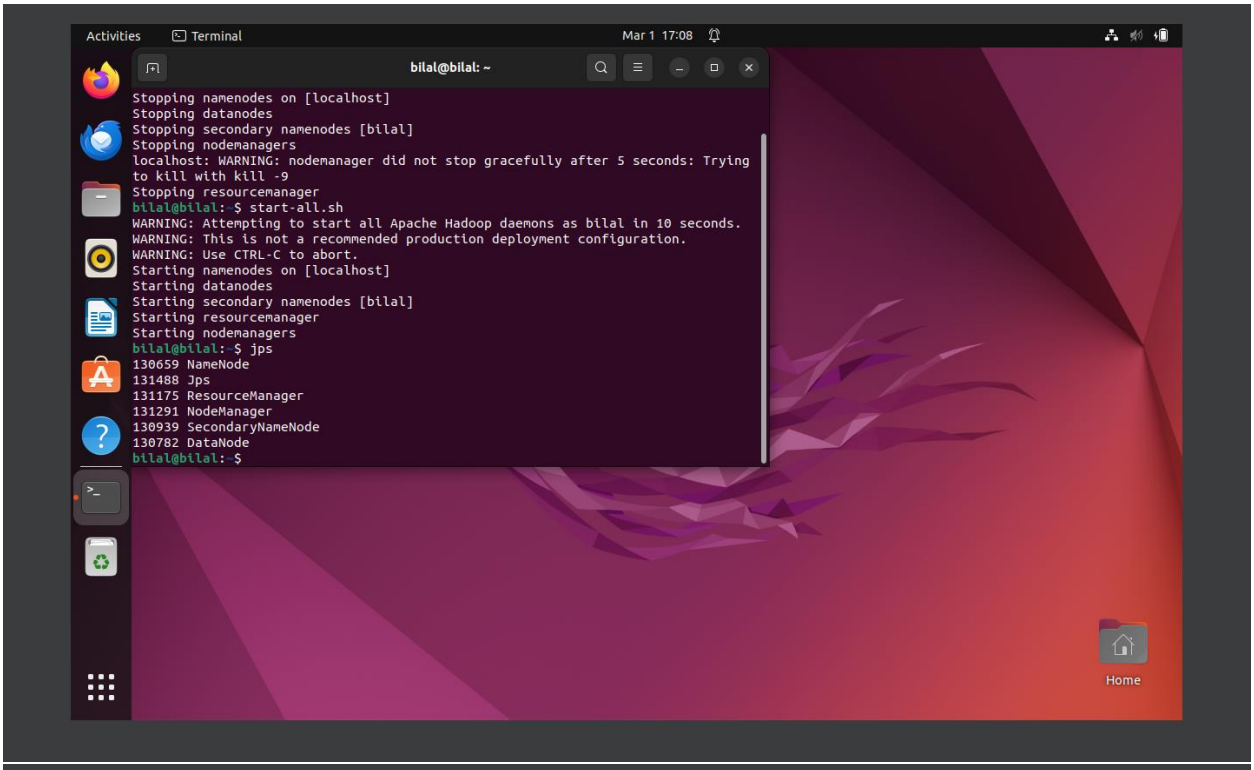
Proof that I did the project using Hadoop with screenshots (total -12):

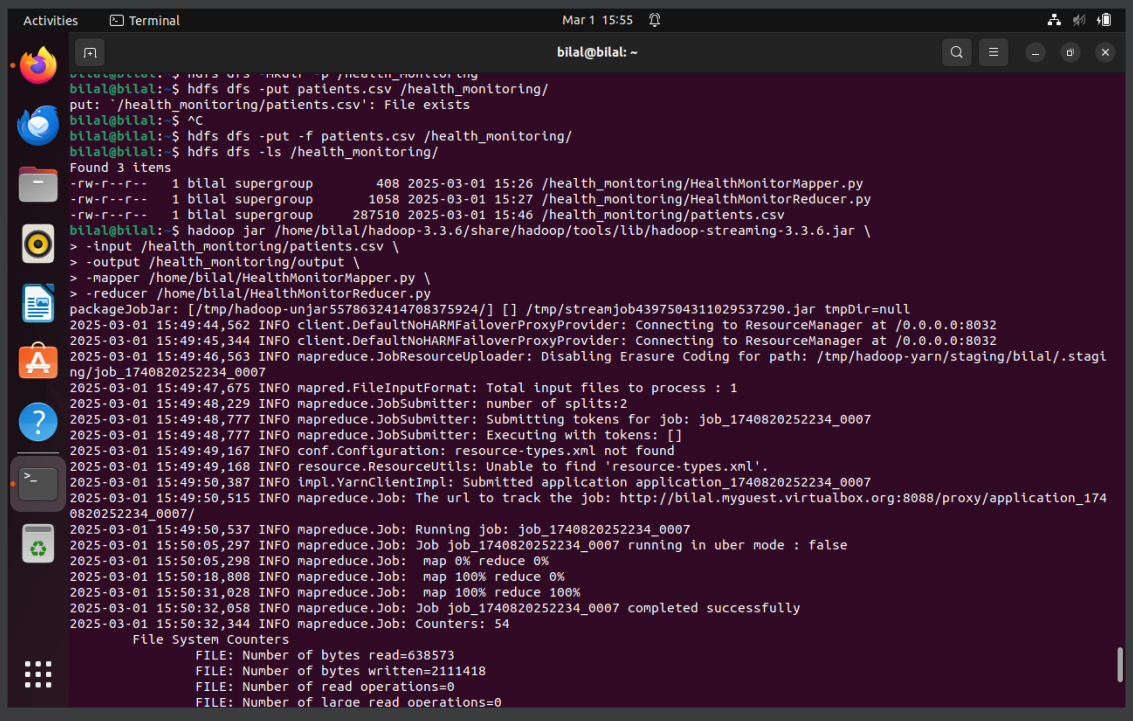


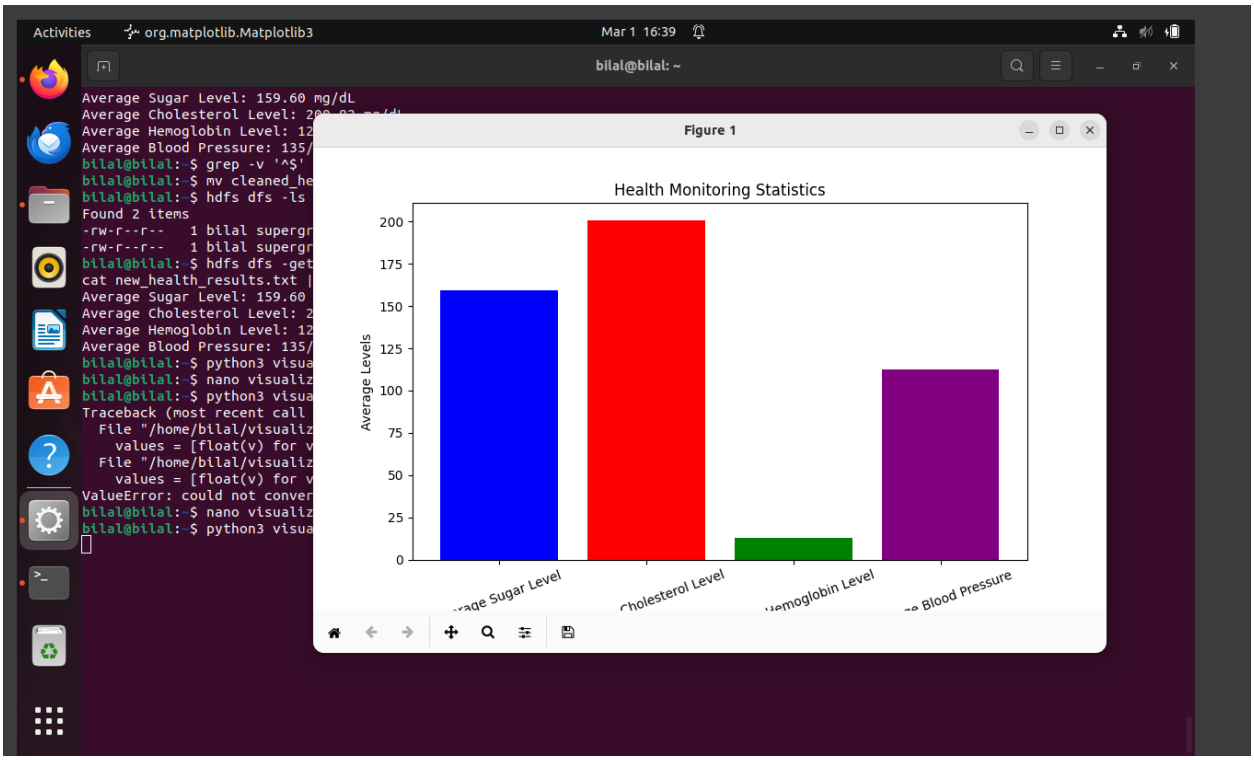

Activities
Terminal
Mar 1 17:40
bilal@bilal: ~
GNU nano 6.2 generate_patients.py
import pandas as pd
import random
# Function to generate BP, Sugar, Cholesterol, Hemoglobin
def generate_bp(): return f"{random.randint(90, 180)}/{random.randint(60, 120)}"
def generate_sugar(): return round(random.uniform(70, 250), 1)
def generate_cholesterol(): return round(random.uniform(100, 300), 1)
def generate_hemoglobin(): return round(random.uniform(8, 18), 1)
# Generate 10,000 patient records
num_patients = 10000
data = [{"Patient_ID": i, "BP": generate_bp(), "Sugar_Level": generate_sugar(),
"Cholesterol": generate_cholesterol(), "Hemoglobin": generate_hemoglob>
for i in range(1, num_patients + 1)]
# Convert to DataFrame and Save as CSV
df = pd.DataFrame(data)
df.to_csv("patients.csv", index=False)
print("✅ 10,000 patient records saved as patients.csv")
[Read 20 lines]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line
Home

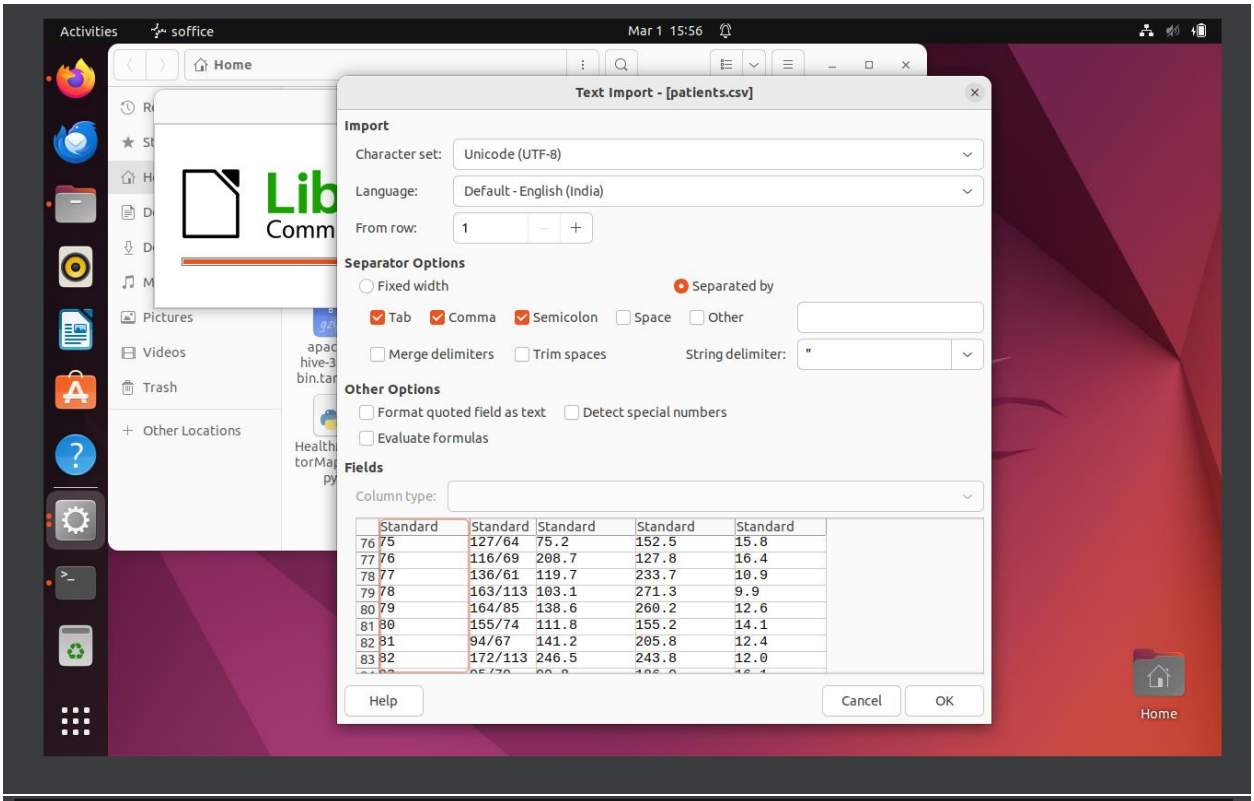

Activities
Terminal
Mar 1 17:41
bilal@bilal: ~
GNU nano 6.2 HealthMonitorMapper.py
#!/usr/bin/env python3
import sys
for line in sys.stdin:
if "Patient_ID" in line:
continue # Skip header
data = line.strip().split(",")
if len(data) != 5:
continue # Skip corrupted lines
patient_id, bp, sugar, cholesterol, hemoglobin = data
print(f"BP\t{bp}")
print(f"Sugar\t{sugar}")
print(f"Cholesterol\t{cholesterol}")
print(f"Hemoglobin\t{hemoglobin}")
[Read 14 lines]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line

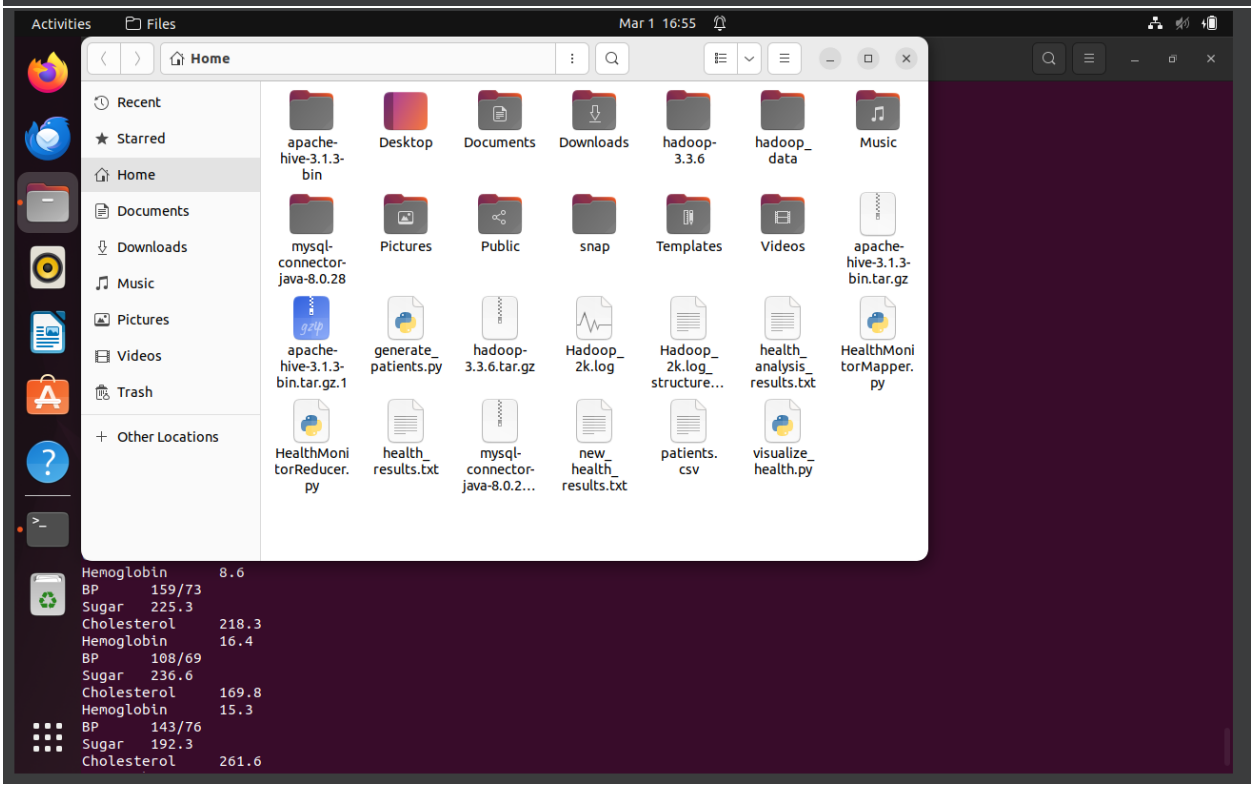

Activities
Terminal
Mar 1 17:52
bilal@bilal: ~
GNU nano 6.2 HealthMonitorReducer.py
#!/usr/bin/env python3
import sys
health_data = {"BP": [], "Sugar": [], "Cholesterol": [], "Hemoglobin": []}
for line in sys.stdin:
key, value = line.strip().split("\t")
if key == "BP":
systolic, diastolic = map(int, value.split("/"))
health_data["BP"].append((systolic, diastolic))
else:
health_data[key].append(float(value))
# Calculate average statistics
avg_sugar = sum(health_data["Sugar"]) / len(health_data["Sugar"])
avg_cholesterol = sum(health_data["Cholesterol"]) / len(health_data["Cholesterol"])
avg_hemoglobin = sum(health_data["Hemoglobin"]) / len(health_data["Hemoglobin"])
avg_systolic = sum(x[0] for x in health_data["BP"]) / len(health_data["BP"])
avg_diastolic = sum(x[1] for x in health_data["BP"]) / len(health_data["BP"])
[Read 25 lines]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line
Home

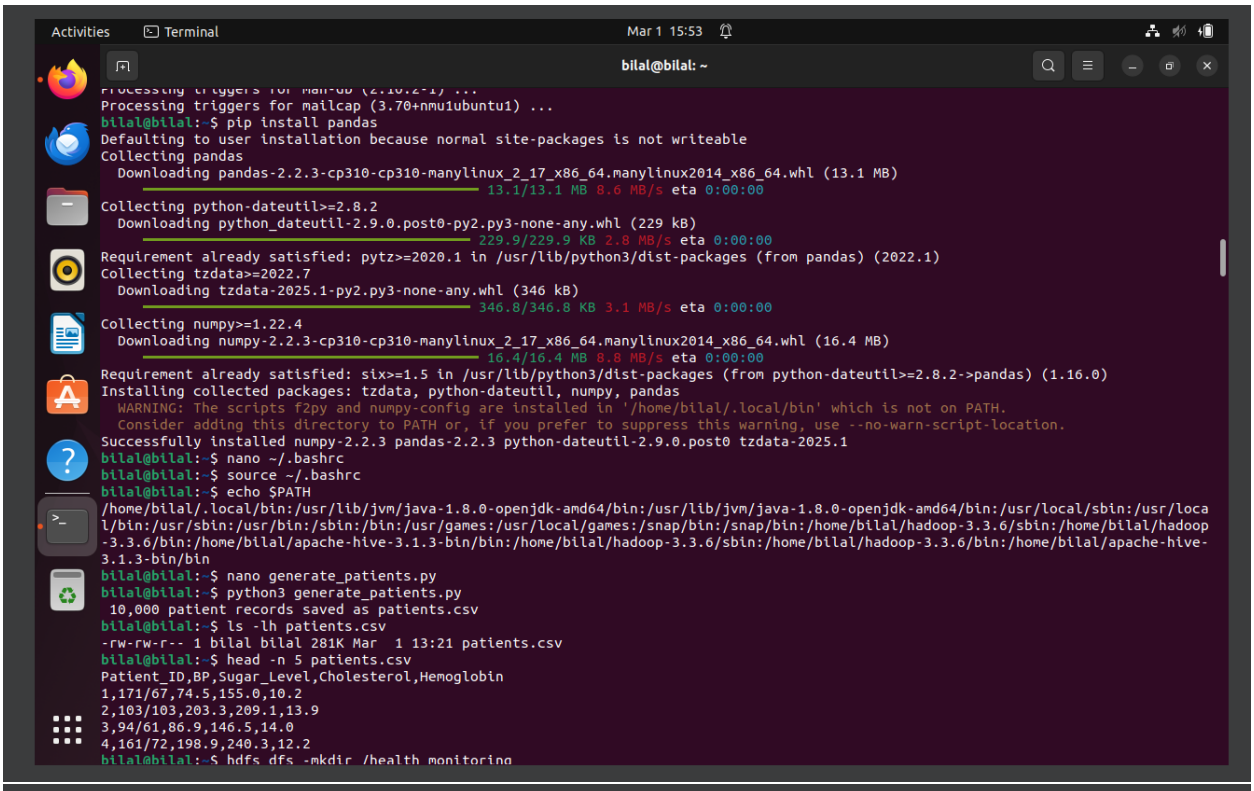

Activities
Terminal
Mar 1 17:08
bilal@bilal: ~
Stopping namenodes on [localhost]
Stopping datanodes
Stopping secondary namenodes [bilal]
Stopping nodemanagers
localhost: WARNING: nodemanager did not stop gracefully after 5 seconds: Trying to kill with kill -9
Stopping resourcemanager
bilal@bilal:~$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as bilal in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [bilal]
Starting resourcemanager
Starting nodemanagers
bilal@bilal:~$ jps
130659 NameNode
131488 Jps
131175 ResourceManager
131291 NodeManager
130939 SecondaryNameNode
130782 DataNode
bilal@bilal:~$
Home

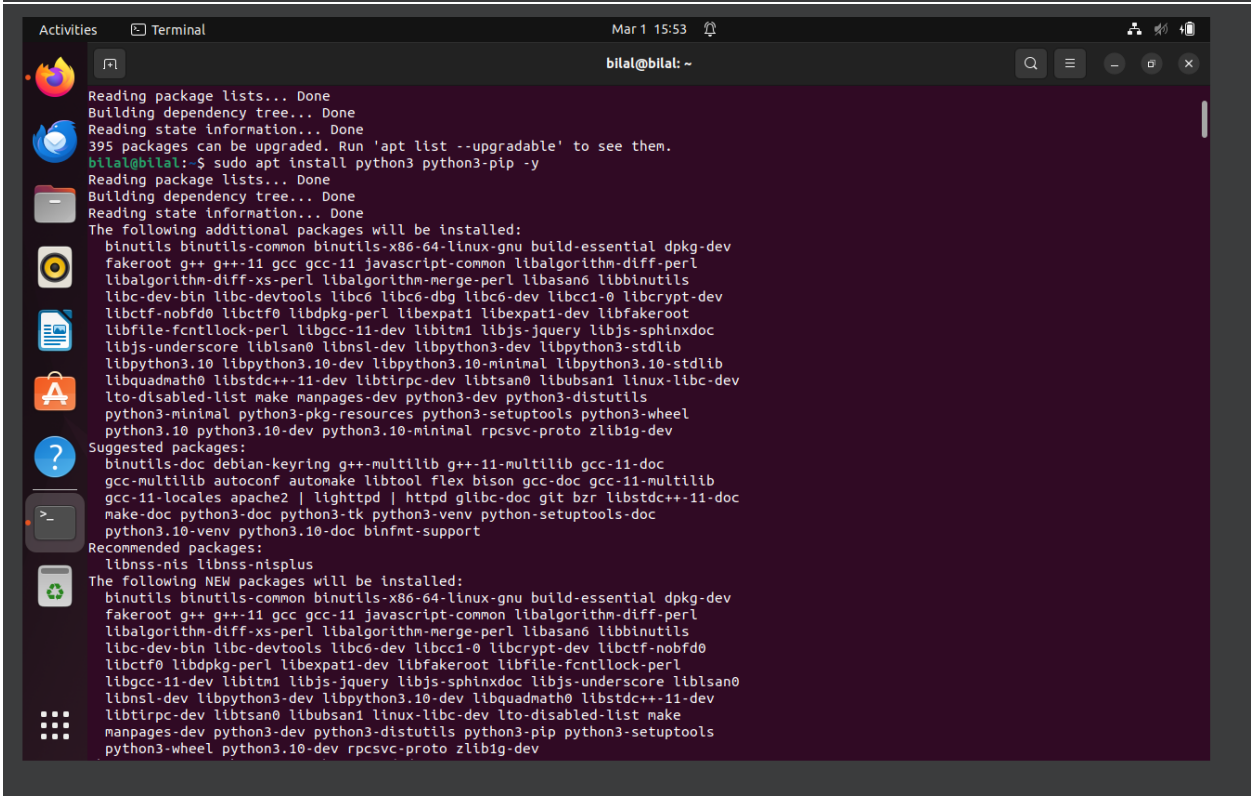

Activities
Terminal
Mar 1 15:55
bilal@bilal: ~
bilal@bilal:~$ hdfs dfs -put patients.csv /health_monitoring/
put: `/health_monitoring/patients.csv': File exists
bilal@bilal:~$ ^C
bilal@bilal:~$ hdfs dfs -put -f patients.csv /health_monitoring/
bilal@bilal:~$ hdfs dfs -ls /health_monitoring/
Found 3 items
-rw-r--r--   1 bilal supergroup   408 2025-03-01 15:26 /health_monitoring/HealthMonitorMapper.py
-rw-r--r--   1 bilal supergroup   1058 2025-03-01 15:27 /health_monitoring/HealthMonitorReducer.py
-rw-r--r--   1 bilal supergroup   287510 2025-03-01 15:46 /health_monitoring/patients.csv
bilal@bilal:~$ hadoop jar /home/bilal/hadoop-3.3.6/share/hadoop/tools/lib/hadoop-streaming-3.3.6.jar \
> -input /health_monitoring/patients.csv \
> -output /health_monitoring/output \
> -mapper /home/bilal/HealthMonitorMapper.py \
> -reducer /home/bilal/HealthMonitorReducer.py
packageJobJar: [/tmp/hadoop-unjar5578632414708375924/] [] /tmp/streamjob4397504311029537290.jar tmpDir=null
2025-03-01 15:49:44,562 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2025-03-01 15:49:45,344 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2025-03-01 15:49:46,563 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/bilal/.staging/job_1740820252234_0007
2025-03-01 15:49:47,675 INFO mapred.FileInputFormat: Total input files to process : 1
2025-03-01 15:49:48,229 INFO mapreduce.JobSubmitter: number of splits:2
2025-03-01 15:49:48,777 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1740820252234_0007
2025-03-01 15:49:48,777 INFO mapreduce.JobSubmitter: Executing with tokens: []
2025-03-01 15:49:49,167 INFO conf.Configuration: resource-types.xml not found
2025-03-01 15:49:49,168 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2025-03-01 15:49:50,387 INFO impl.YarnClientImpl: Submitted application application_1740820252234_0007
2025-03-01 15:49:50,515 INFO mapreduce.Job: The url to track the job: http://bilal.myguest.virtualbox.org:8088/proxy/application_1740820252234_0007/
2025-03-01 15:49:50,537 INFO mapreduce.Job: Running job: job_1740820252234_0007
2025-03-01 15:50:05,297 INFO mapreduce.Job: Job job_1740820252234_0007 running in uber mode : false
2025-03-01 15:50:05,298 INFO mapreduce.Job: map 0% reduce 0%
2025-03-01 15:50:18,808 INFO mapreduce.Job: map 100% reduce 0%
2025-03-01 15:50:31,028 INFO mapreduce.Job: map 100% reduce 100%
2025-03-01 15:50:32,058 INFO mapreduce.Job: Job job_1740820252234_0007 completed successfully
2025-03-01 15:50:32,344 INFO mapreduce.Job: Counters: 54
File System Counters
FILE: Number of bytes read=638573
FILE: Number of bytes written=2111418
FILE: Number of read operations=0
FILE: Number of large read operations=0

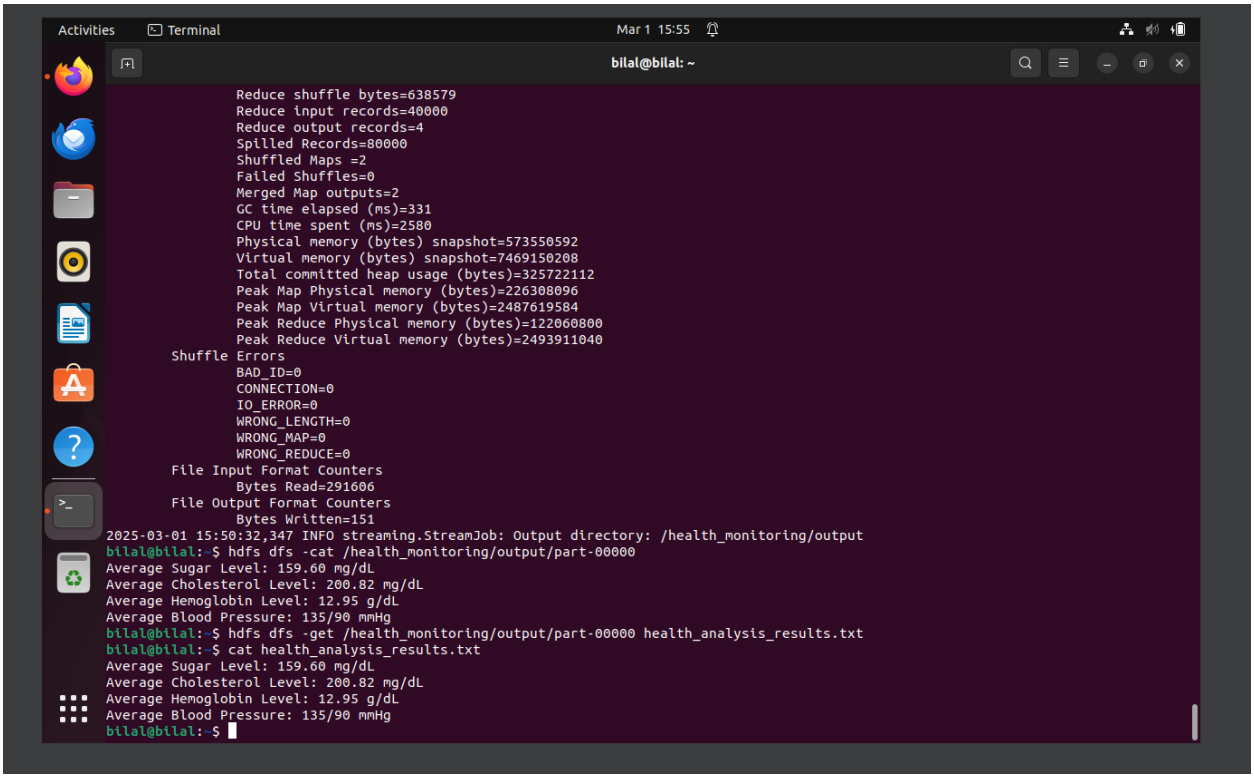

Activities
org.matplotlib.Matplotlib3
Mar 1 16:39
bilal@bilal: ~
Average Sugar Level: 159.60 mg/dL
Average Cholesterol Level:
Average Hemoglobin Level:
Average Blood Pressure:
bilal@bilal:~$ grep -v '^$'
bilal@bilal:~$ mv cleaned_he
bilal@bilal:~$ hdfs dfs -ls
Found 2 items
-rw-r--r--
-rw-r--r--
bilal@bilal:~$ hdfs dfs -get
cat new_health_results.txt
Average Sugar Level: 159.60
Average Cholesterol Level:
Average Hemoglobin Level:
Average Blood Pressure:
bilal@bilal:~$ python3 visua
bilal@bilal:~$ nano visualiz
bilal@bilal:~$ python3 visua
Traceback (most recent call
File "/home/bilal/visualiz
values = [float(v) for v
File "/home/bilal/visualiz
values = [float(v) for v
ValueError: could not conver
bilal@bilal:~$ nano visualiz
bilal@bilal:~$ python3 visua
Figure 1
Health Monitoring Statistics
Average Levels
0
25
50
75
100
125
150
175
200
Cholesterol Level
Blood Pressure


Activities
soffice
Mar 1 15:56
Home
Text Import - [patients.csv]
Import
Character set: Unicode (UTF-8)
Language: Default - English (India)
From row: 1
Separator Options
Fixed width
Separated by
Tab
Comma
Semicolon
Space
Other
Merge delimiters
Trim spaces
String delimiter: "
Other Options
Format quoted field as text
Detect special numbers
Evaluate formulas
Fields
Column type:
Standard Standard Standard Standard Standard
76 75 127/64 75.2 152.5 15.8
77 76 116/69 208.7 127.8 16.4
78 77 136/61 119.7 233.7 10.9
79 78 163/113 103.1 271.3 9.9
80 79 164/85 138.6 260.2 12.6
81 80 155/74 111.8 155.2 14.1
82 81 94/67 141.2 205.8 12.4
83 82 172/113 246.5 243.8 12.0
Help
Cancel
OK
Pictures
Videos
Trash
Other Locations
Home


Activities
Files
Mar 1 16:55
Home
Recent
Starred
Home
Documents
Downloads
Music
Pictures
Videos
Trash
Other Locations
apache-hive-3.1.3-bin
Desktop
Documents
Downloads
hadoop-3.3.6
hadoop_data
Music
mysql-connector-java-8.0.28
Pictures
Public
snap
Templates
Videos
apache-hive-3.1.3-bin.tar.gz
apache-hive-3.1.3-bin.tar.gz.1
generate_patients.py
hadoop-3.3.6.tar.gz
Hadoop_2k.log
Hadoop_2k.log_structure...
health_analysis_results.txt
HealthMonitorMapper.py
HealthMonitorReducer.py
health_results.txt
mysql-connector-java-8.0.2...
new_health_results.txt
patients.csv
visualize_health.py
Hemoglobin 8.6
BP 159/73
Sugar 225.3
Cholesterol 218.3
Hemoglobin 16.4
BP 108/69
Sugar 236.6
Cholesterol 169.8
Hemoglobin 15.3
BP 143/76
Sugar 192.3
Cholesterol 261.6


Activities
Terminal
Mar 1 15:53
bilal@bilal: ~
Processing triggers for mailcap (3.70+nmu1ubuntu1) ...
bilal@bilal:~$ pip install pandas
Defaulting to user installation because normal site-packages is not writeable
Collecting pandas
Downloading pandas-2.2.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (13.1 MB)
13.1/13.1 MB 8.6 MB/s eta 0:00:00
Collecting python-dateutil>=2.8.2
Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl (229 kB)
229.9/229.9 KB 2.8 MB/s eta 0:00:00
Requirement already satisfied: pytz>=2020.1 in /usr/lib/python3/dist-packages (from pandas) (2022.1)
Collecting tzdata>=2022.7
Downloading tzdata-2025.1-py2.py3-none-any.whl (346 kB)
346.8/346.8 KB 3.1 MB/s eta 0:00:00
Collecting numpy>=1.22.4
Downloading numpy-2.2.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (16.4 MB)
16.4/16.4 MB 8.8 MB/s eta 0:00:00
Requirement already satisfied: six>=1.5 in /usr/lib/python3/dist-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
Installing collected packages: tzdata, python-dateutil, numpy, pandas
WARNING: The scripts f2py and numpy-config are installed in '/home/bilal/.local/bin' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed numpy-2.2.3 pandas-2.2.3 python-dateutil-2.9.0.post0 tzdata-2025.1
bilal@bilal:~$ nano ~/.bashrc
bilal@bilal:~$ source ~/.bashrc
bilal@bilal:~$ echo $PATH
/home/bilal/.local/bin:/usr/lib/jvm/java-1.8.0-openjdk-amd64/bin:/usr/lib/jvm/java-1.8.0-openjdk-amd64/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/snap/bin:/home/bilal/hadoop-3.3.6/sbin:/home/bilal/hadoop-3.3.6/bin:/home/bilal/apache-hive-3.1.3-bin/bin:/home/bilal/hadoop-3.3.6/sbin:/home/bilal/hadoop-3.3.6/bin:/home/bilal/apache-hive-3.1.3-bin/bin
bilal@bilal:~$ nano generate_patients.py
bilal@bilal:~$ python3 generate_patients.py
10,000 patient records saved as patients.csv
bilal@bilal:~$ ls -lh patients.csv
-rw-rw-r-- 1 bilal bilal 281K Mar 1 13:21 patients.csv
bilal@bilal:~$ head -n 5 patients.csv
Patient_ID,BP,Sugar_Level,Cholesterol,Hemoglobin
1,171/67,74.5,155.0,10.2
2,103/103,203.3,209.1,13.9
3,94/61,86.9,146.5,14.0
4,161/72,198.9,240.3,12.2
bilal@bilal:~$ hdfs dfs -mkdir /health_monitoring


Activities
Terminal
Mar 1 15:53
bilal@bilal: ~
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
395 packages can be upgraded. Run 'apt list --upgradable' to see them.
bilal@bilal:~$ sudo apt install python3 python3-pip -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
binutils binutils-common binutils-x86-64-linux-gnu build-essential dpkg-dev
fakeroot g++ g++-11 gcc gcc-11 javascript-common libalgorithm-diff-perl
libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan6 libbinutils
libc-dev-bin libc-devtools libc6 libc6-dbg libc6-dev libcc1-0 libcrypt-dev
libctf-nobfd0 libctf0 libdpkg-perl libexpat1 libexpat1-dev libfakeroot
libfile-fcntllock-perl libgcc-11-dev libitm1 libjs-jquery libjs-sphinxdoc
libjs-underscore liblsan0 libnsl-dev libpython3-dev libpython3-stdlib
libpython3.10 libpython3.10-dev libpython3.10-minimal libpython3.10-stdlib
libquadmath0 libstdc++-11-dev libtirpc-dev libtsan0 libubsan1 linux-libc-dev
lto-disabled-list make manpages-dev python3-dev python3-distutils
python3-minimal python3-pkg-resources python3-setuptools python3-wheel
python3.10 python3.10-dev python3.10-minimal rpcsvc-proto zlib1g-dev
Suggested packages:
binutils-doc debian-keyring g++-multilib g++-11-multilib gcc-11-doc
gcc-multilib autoconf automake libtool flex bison gcc-doc gcc-11-multilib
gcc-11-locales apache2 | lighttpd | httpd glibc-doc git bzr libstdc++-11-doc
make-doc python3-doc python3-tk python3-venv python-setuptools-doc
python3.10-venv python3.10-doc binfmt-support
Recommended packages:
libnss-nis libnss-nisplus
The following NEW packages will be installed:
binutils binutils-common binutils-x86-64-linux-gnu build-essential dpkg-dev
fakeroot g++ g++-11 gcc gcc-11 javascript-common libalgorithm-diff-perl
libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan6 libbinutils
libc-dev-bin libc-devtools libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0
libctf0 libdpkg-perl libexpat1-dev libfakeroot libfile-fcntllock-perl
libgcc-11-dev libitm1 libjs-jquery libjs-sphinxdoc libjs-underscore liblsan0
libnsl-dev libpython3-dev libpython3.10-dev libquadmath0 libstdc++-11-dev
libtirpc-dev libtsan0 libubsan1 linux-libc-dev lto-disabled-list make
manpages-dev python3-dev python3-distutils python3-pip python3-setuptools
python3-wheel python3.10-dev rpcsvc-proto zlib1g-dev


Activities
Terminal
Mar 1 15:55
bilal@bilal: ~
Reduce shuffle bytes=638579
Reduce input records=40000
Reduce output records=4
Spilled Records=80000
Shuffled Maps =2
Failed Shuffles=0
Merged Map outputs=2
GC time elapsed (ms)=331
CPU time spent (ms)=2580
Physical memory (bytes) snapshot=573550592
Virtual memory (bytes) snapshot=7469150208
Total committed heap usage (bytes)=325722112
Peak Map Physical memory (bytes)=226308096
Peak Map Virtual memory (bytes)=2487619584
Peak Reduce Physical memory (bytes)=122060800
Peak Reduce Virtual memory (bytes)=2493911040
Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
Bytes Read=291606
File Output Format Counters
Bytes Written=151
2025-03-01 15:50:32,347 INFO streaming.StreamJob: Output directory: /health_monitoring/output
bilal@bilal:~$ hdfs dfs -cat /health_monitoring/output/part-00000
Average Sugar Level: 159.60 mg/dL
Average Cholesterol Level: 200.82 mg/dL
Average Hemoglobin Level: 12.95 g/dL
Average Blood Pressure: 135/90 mmHg
bilal@bilal:~$ hdfs dfs -get /health_monitoring/output/part-00000 health_analysis_results.txt
bilal@bilal:~$ cat health_analysis_results.txt
Average Sugar Level: 159.60 mg/dL
Average Cholesterol Level: 200.82 mg/dL
Average Hemoglobin Level: 12.95 g/dL
Average Blood Pressure: 135/90 mmHg
bilal@bilal:~$