Develop mapreduce progrum to calculate the frequency of a given file.

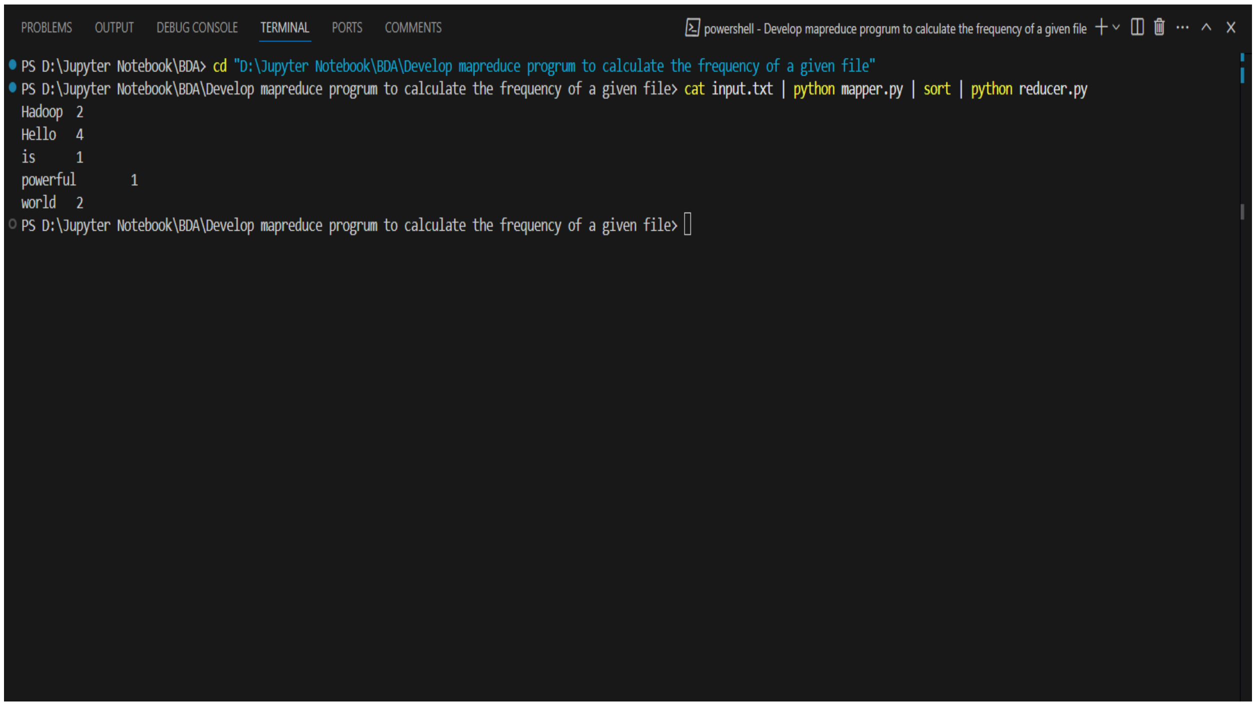
mapper.py

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


# Read input line by line
for line in sys.stdin:
    line = line.strip()  # Remove whitespace
    words = line.split()  # Split into words
    for word in words:
        print(f"{word}\t1")  # Emit (word, 1)
```

reducer.py

```
#!/usr/bin/env python3
import sys
from collections import defaultdict


word_count = defaultdict(int)


# Read input from standard input
for line in sys.stdin:
    word, count = line.strip().split("\t")
    word_count[word] += int(count)


# Print the final counts
for word, count in word_count.items():
    print(f"{word}\t{count}")
```

Output :-


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
COMMENTS
powershell - Develop mapreduce progrum to calculate the frequency of a given file

```
PS D:\Jupyter Notebook\BDA> cd "D:\Jupyter Notebook\BDA\Develop mapreduce progrum to calculate the frequency of a given file"
PS D:\Jupyter Notebook\BDA\Develop mapreduce progrum to calculate the frequency of a given file> cat input.txt | python mapper.py | sort | python reducer.py
Hadoop  2
Hello   4
is      1
powerful        1
world   2
PS D:\Jupyter Notebook\BDA\Develop mapreduce progrum to calculate the frequency of a given file>
```

Implement matrix multiplication using map-reduce.

matrix.txt

```
A 0 0 1
A 0 1 2
A 1 0 3
A 1 1 4
B 0 0 5
B 0 1 6
B 1 0 7
B 1 1 8
```

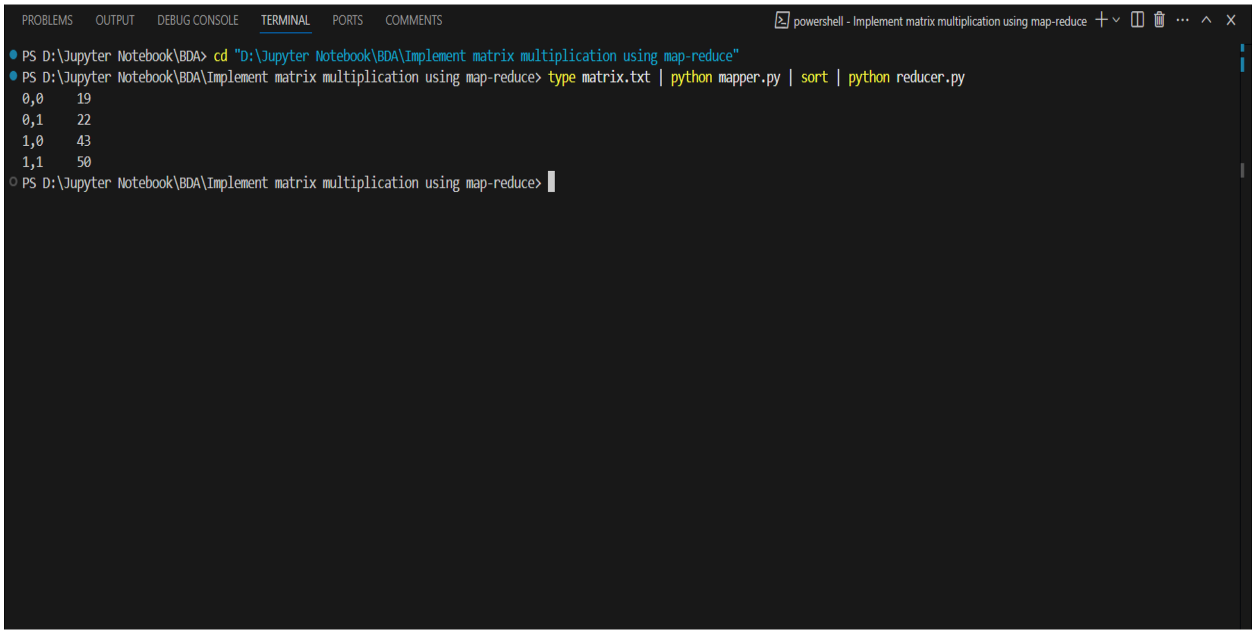
mapper.py

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

# Read input line by line
for line in sys.stdin:
    line = line.strip()
    matrix, row, col, value = line.split()
    row, col, value = int(row), int(col), int(value)

    if matrix == 'A':  # Emit values to be multiplied with B
        for k in range(2):  # Assuming B has 2 columns
            print(f"{row},{k}\tA,{col},{value}")

    elif matrix == 'B':  # Emit values to be multiplied with A
        for i in range(2):  # Assuming A has 2 rows
            print(f"{i},{col}\tB,{row},{value}")
```

reducer.py

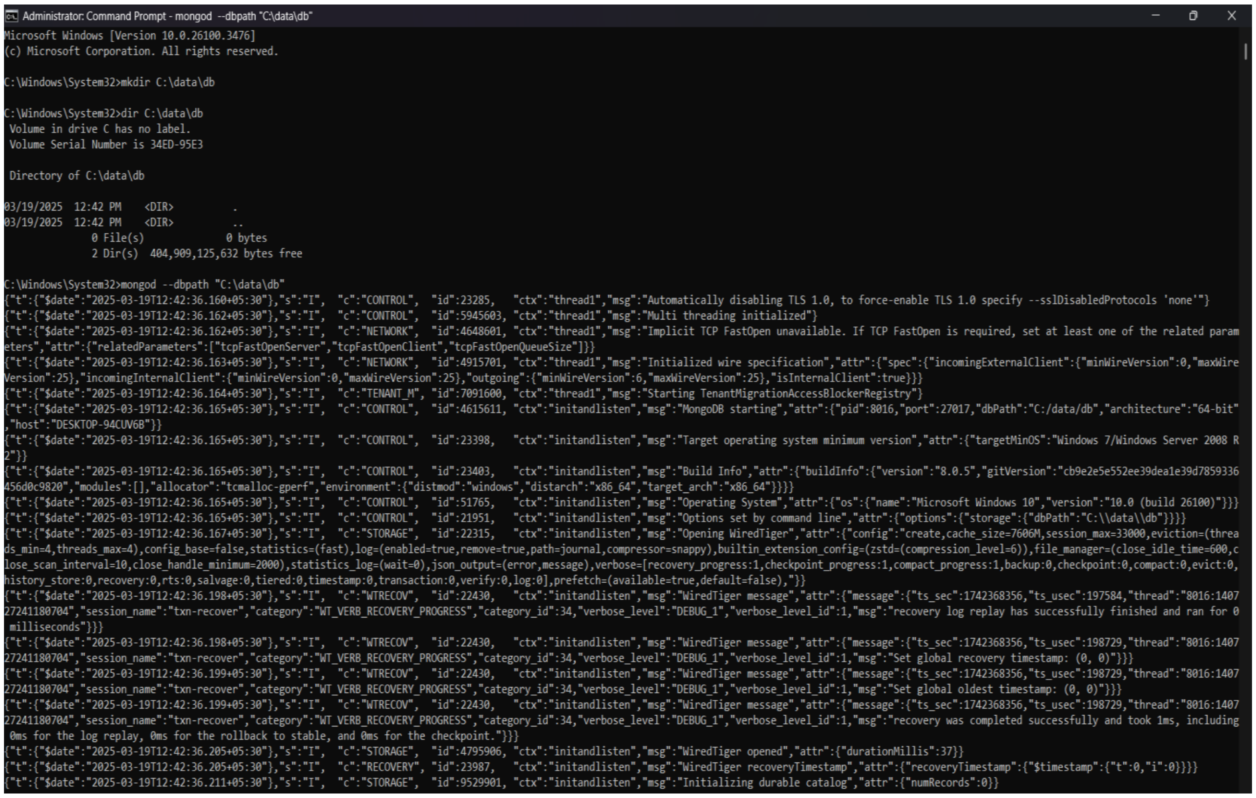
```
#!/usr/bin/env python3
import sys
from collections import defaultdict

# Store values by keys
product_terms = defaultdict(list)

# Read input from standard input
for line in sys.stdin:
    key, value = line.strip().split("\t")
    product_terms[key].append(value)

# Compute matrix multiplication result
result = defaultdict(int)
for key, values in product_terms.items():
    a_values = {int(v.split(",")[1]): int(v.split(",")[2]) for v in values if v.startswith("A")}
    b_values = {int(v.split(",")[1]): int(v.split(",")[2]) for v in values if v.startswith("B")}

    for k in a_values:
        if k in b_values:
            result[key] += a_values[k] * b_values[k]

# Print the final matrix product
for key, value in sorted(result.items()):
    print(f"{key}\t{value}")
```

Output:-


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
COMMENTS
powershell - Implement matrix multiplication using map-reduce

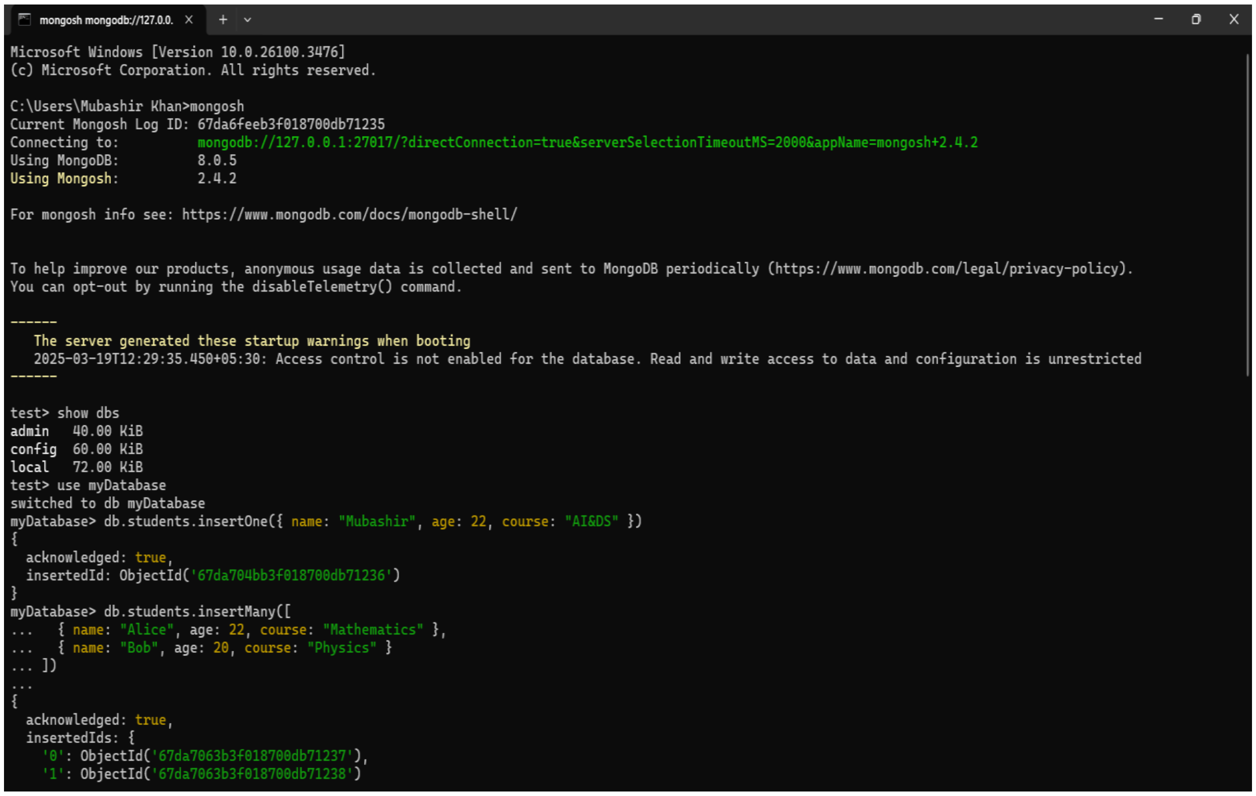
```
PS D:\Jupyter Notebook\BDA> cd "D:\Jupyter Notebook\BDA\Implement matrix multiplication using map-reduce"
PS D:\Jupyter Notebook\BDA\Implement matrix multiplication using map-reduce> type matrix.txt | python mapper.py | sort | python reducer.py
0,0     19
0,1     22
1,0     43
1,1     50
PS D:\Jupyter Notebook\BDA\Implement matrix multiplication using map-reduce>
```

Mongodb Installation 4 creation of database Collection Insert Query, update query & delete Documents.



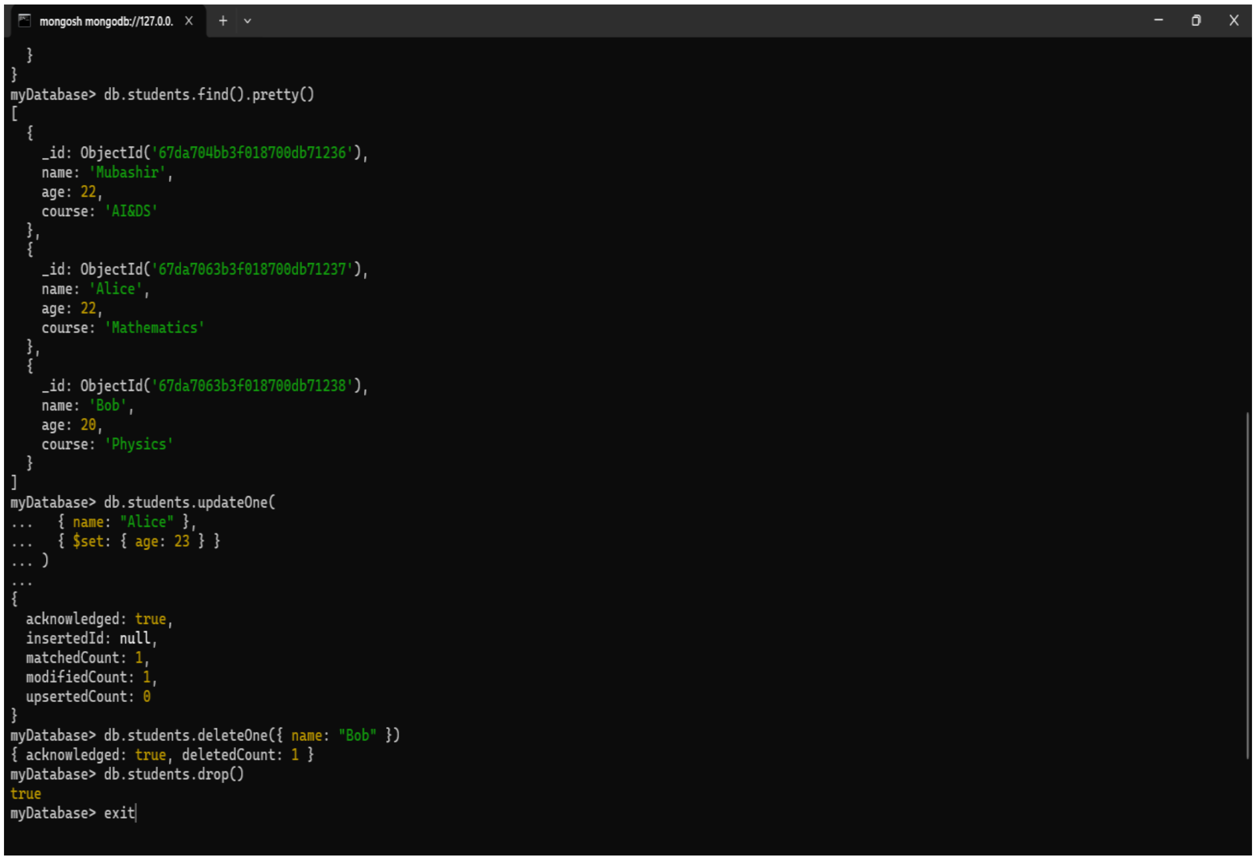
Administrator: Command Prompt - mongod --dbpath "C:\data\db"

```
Microsoft Windows [Version 10.0.26100.3476]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>mkdir C:\data\db

C:\Windows\System32>dir C:\data\db
 Volume in drive C has no label.
 Volume Serial Number is 34ED-95E3

 Directory of C:\data\db

03/19/2025  12:42 PM    <DIR>          .
03/19/2025  12:42 PM    <DIR>          ..
               0 File(s)              0 bytes
               2 Dir(s)  404,909,125,632 bytes free

C:\Windows\System32>mongod --dbpath "C:\data\db"
{"t":{"$date":"2025-03-19T12:42:36.160+05:30"},"s":"I",  "c":"CONTROL",  "id":23285,   "ctx":"thread1","msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2025-03-19T12:42:36.162+05:30"},"s":"I",  "c":"CONTROL",  "id":5945603, "ctx":"thread1","msg":"Multi threading initialized"}
{"t":{"$date":"2025-03-19T12:42:36.162+05:30"},"s":"I",  "c":"NETWORK",  "id":4648601, "ctx":"thread1","msg":"Implicit TCP FastOpen unavailable. If TCP FastOpen is required, set at least one of the related param
eters","attr":{"relatedParameters":["tcpFastOpenServer","tcpFastOpenClient","tcpFastOpenQueueSize"]}}
{"t":{"$date":"2025-03-19T12:42:36.163+05:30"},"s":"I",  "c":"NETWORK",  "id":4915701, "ctx":"thread1","msg":"Initialized wire specification","attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWire
Version":25},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":25},"outgoing":{"minWireVersion":6,"maxWireVersion":25},"isInternalClient":true}}}
{"t":{"$date":"2025-03-19T12:42:36.164+05:30"},"s":"I",  "c":"TENANT_M", "id":7091600, "ctx":"thread1","msg":"Starting TenantMigrationAccessBlockerRegistry"}
{"t":{"$date":"2025-03-19T12:42:36.165+05:30"},"s":"I",  "c":"CONTROL",  "id":4615611, "ctx":"initandlisten","msg":"MongoDB starting","attr":{"pid":8016,"port":27017,"dbPath":"C:/data/db","architecture":"64-bit"
,"host":"DESKTOP-94CUV6B"}}
{"t":{"$date":"2025-03-19T12:42:36.165+05:30"},"s":"I",  "c":"CONTROL",  "id":23398,   "ctx":"initandlisten","msg":"Target operating system minimum version","attr":{"targetMinOS":"Windows 7/Windows Server 2008 R
2"}}
{"t":{"$date":"2025-03-19T12:42:36.165+05:30"},"s":"I",  "c":"CONTROL",  "id":23403,   "ctx":"initandlisten","msg":"Build Info","attr":{"buildInfo":{"version":"8.0.5","gitVersion":"cb9e2e5e552ee39dea1e39d7859336
456d0c9820","modules":[],"allocator":"tcmalloc-gperf","environment":{"distmod":"windows","distarch":"x86_64","target_arch":"x86_64"}}}}
{"t":{"$date":"2025-03-19T12:42:36.165+05:30"},"s":"I",  "c":"CONTROL",  "id":51765,   "ctx":"initandlisten","msg":"Operating System","attr":{"os":{"name":"Microsoft Windows 10","version":"10.0 (build 26100)"}}}
{"t":{"$date":"2025-03-19T12:42:36.165+05:30"},"s":"I",  "c":"CONTROL",  "id":21951,   "ctx":"initandlisten","msg":"Options set by command line","attr":{"options":{"storage":{"dbPath":"C:\\data\\db"}}}}
{"t":{"$date":"2025-03-19T12:42:36.167+05:30"},"s":"I",  "c":"STORAGE",  "id":22315,   "ctx":"initandlisten","msg":"Opening WiredTiger","attr":{"config":"create,cache_size=7606M,session_max=33000,eviction=(threa
ds_min=4,threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,remove=true,path=journal,compressor=snappy),builtin_extension_config=(zstd=(compression_level=6)),file_manager=(close_idle_time=600,c
lose_scan_interval=10,close_handle_minimum=2000),statistics_log=(wait=0),json_output=(error,message),verbose=[recovery_progress:1,checkpoint_progress:1,compact_progress:1,backup:0,checkpoint:0,compact:0,evict:0,
history_store:0,recovery:0,rts:0,salvage:0,tiered:0,timestamp:0,transaction:0,verify:0,log:0],prefetch=(available=true,default=false),"}}
{"t":{"$date":"2025-03-19T12:42:36.198+05:30"},"s":"I",  "c":"WTRECOV",  "id":22430,   "ctx":"initandlisten","msg":"WiredTiger message","attr":{"message":{"ts_sec":1742368356,"ts_usec":197584,"thread":"8016:1407
27241180704","session_name":"txn-recover","category":"WT_VERB_RECOVERY_PROGRESS","category_id":34,"verbose_level":"DEBUG_1","verbose_level_id":1,"msg":"recovery log replay has successfully finished and ran for 0
 milliseconds"}}}
{"t":{"$date":"2025-03-19T12:42:36.198+05:30"},"s":"I",  "c":"WTRECOV",  "id":22430,   "ctx":"initandlisten","msg":"WiredTiger message","attr":{"message":{"ts_sec":1742368356,"ts_usec":198729,"thread":"8016:1407
27241180704","session_name":"txn-recover","category":"WT_VERB_RECOVERY_PROGRESS","category_id":34,"verbose_level":"DEBUG_1","verbose_level_id":1,"msg":"Set global recovery timestamp: (0, 0)"}}}
{"t":{"$date":"2025-03-19T12:42:36.199+05:30"},"s":"I",  "c":"WTRECOV",  "id":22430,   "ctx":"initandlisten","msg":"WiredTiger message","attr":{"message":{"ts_sec":1742368356,"ts_usec":198729,"thread":"8016:1407
27241180704","session_name":"txn-recover","category":"WT_VERB_RECOVERY_PROGRESS","category_id":34,"verbose_level":"DEBUG_1","verbose_level_id":1,"msg":"Set global oldest timestamp: (0, 0)"}}}
{"t":{"$date":"2025-03-19T12:42:36.199+05:30"},"s":"I",  "c":"WTRECOV",  "id":22430,   "ctx":"initandlisten","msg":"WiredTiger message","attr":{"message":{"ts_sec":1742368356,"ts_usec":198729,"thread":"8016:1407
27241180704","session_name":"txn-recover","category":"WT_VERB_RECOVERY_PROGRESS","category_id":34,"verbose_level":"DEBUG_1","verbose_level_id":1,"msg":"recovery was completed successfully and took 1ms, including
 0ms for the log replay, 0ms for the rollback to stable, and 0ms for the checkpoint."}}}
{"t":{"$date":"2025-03-19T12:42:36.205+05:30"},"s":"I",  "c":"STORAGE",  "id":4795906, "ctx":"initandlisten","msg":"WiredTiger opened","attr":{"durationMillis":37}}
{"t":{"$date":"2025-03-19T12:42:36.205+05:30"},"s":"I",  "c":"RECOVERY", "id":23987,   "ctx":"initandlisten","msg":"WiredTiger recoveryTimestamp","attr":{"recoveryTimestamp":{"$timestamp":{"t":0,"i":0}}}}
{"t":{"$date":"2025-03-19T12:42:36.211+05:30"},"s":"I",  "c":"STORAGE",  "id":9529901, "ctx":"initandlisten","msg":"Initializing durable catalog","attr":{"numRecords":0}}
```



mongosh mongodb://127.0.0.

```
Microsoft Windows [Version 10.0.26100.3476]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Mubashir Khan>mongosh
Current Mongosh Log ID: 67da6feeb3f018700db71235
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.4.2
Using MongoDB:          8.0.5
Using Mongosh:          2.4.2

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/


To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

------
   The server generated these startup warnings when booting
   2025-03-19T12:29:35.450+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
------

test> show dbs
admin   40.00 KiB
config  60.00 KiB
local   72.00 KiB
test> use myDatabase
switched to db myDatabase
myDatabase> db.students.insertOne({ name: "Mubashir", age: 22, course: "AI&DS" })
{
  acknowledged: true,
  insertedId: ObjectId('67da704bb3f018700db71236')
}
myDatabase> db.students.insertMany([
...   { name: "Alice", age: 22, course: "Mathematics" },
...   { name: "Bob", age: 20, course: "Physics" }
... ])
...
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('67da7063b3f018700db71237'),
    '1': ObjectId('67da7063b3f018700db71238')
```



mongosh mongodb://127.0.0.

```
  }
}
myDatabase> db.students.find().pretty()
[
  {
    _id: ObjectId('67da704bb3f018700db71236'),
    name: 'Mubashir',
    age: 22,
    course: 'AI&DS'
  },
  {
    _id: ObjectId('67da7063b3f018700db71237'),
    name: 'Alice',
    age: 22,
    course: 'Mathematics'
  },
  {
    _id: ObjectId('67da7063b3f018700db71238'),
    name: 'Bob',
    age: 20,
    course: 'Physics'
  }
]
myDatabase> db.students.updateOne(
...   { name: "Alice" },
...   { $set: { age: 23 } }
... )
...
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
myDatabase> db.students.deleteOne({ name: "Bob" })
{ acknowledged: true, deletedCount: 1 }
myDatabase> db.students.drop()
true
myDatabase> exit
```

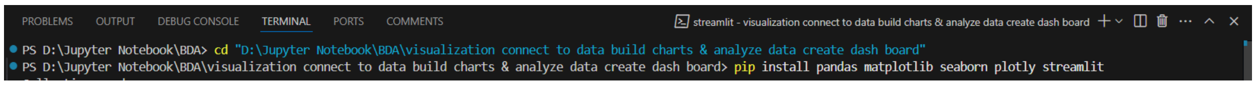
Visualization connect to data build charts & analyze data create dashboard

dashboard.py

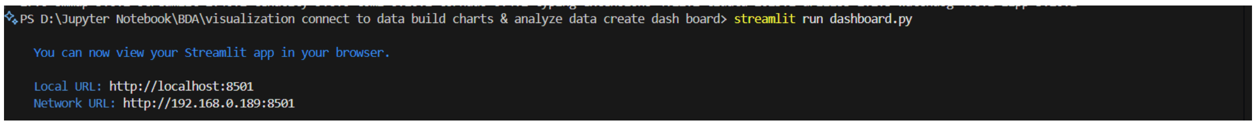
```
import streamlit as st
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
# 📌 Load Titanic dataset from a public URL
DATA_URL = "https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.csv"
data = pd.read_csv(DATA_URL)
# ------------------ □ STREAMLIT DASHBOARD START ------------------
# 📌 Set Power BI-like page layout
st.set_page_config(page_title="🚢 Titanic Analysis", layout="wide")
# 📌 Dashboard Header
st.markdown("<h1 style='text-align: center;'>🚢 Titanic Passenger Dashboard</h1>", unsafe_allow_html=True)
st.markdown("<h3 style='text-align: center;'>Analyze survival rates and passenger demographics</h3>", unsafe_allow_html=True)
# --------------- □ KPI METRICS (TOP ROW) -----------------
st.markdown("### 🚀 Key Statistics")
col1, col2, col3, col4 = st.columns(4)
# 📌 Display KPIs
with col1:
    st.metric(label="👥 Total Passengers", value=f"{len(data)}")
with col2:
    survived_count = data["Survived"].sum()
    st.metric(label="💖 Survived", value=f"{survived_count}")
with col3:
    avg_fare = round(data["Fare"].mean(), 2)
    st.metric(label="💰 Avg Fare", value=f"${avg_fare}")
with col4:
    avg_age = round(data["Age"].mean(), 1)
    st.metric(label="📊 Avg Age", value=f"{avg_age} yrs")
# --------------- □ VISUALIZATION GRID (MIDDLE ROW) -----------------
st.markdown("### 📊 Data Visualizations")
row1_col1, row1_col2 = st.columns(2)
# 📌 Bar Chart - Survival Count
```

```
with row1_col1:
    st.markdown("#### 📌 Survival Count")
    fig, ax = plt.subplots()
    data["Survived"].value_counts().plot(kind="bar", color=["red", "green"], ax=ax)
    plt.xlabel("Survival (0 = No, 1 = Yes)")
    plt.ylabel("Count")
    st.pyplot(fig)
# 📌 Histogram - Age Distribution
with row1_col2:
    st.markdown("#### 📌 Age Distribution of Passengers")
    fig, ax = plt.subplots()
    sns.histplot(data["Age"].dropna(), bins=20, kde=True, ax=ax)
    plt.xlabel("Age")
    plt.ylabel("Frequency")
    st.pyplot(fig)
# --------------- □ INTERACTIVE SCATTER PLOT (FULL WIDTH) -----------------
st.markdown("### 🔍 Passenger Insights")
fig = px.scatter(data, x="Age", y="Fare", color="Survived",
            title="Fare Paid vs Age",
            labels={"Fare": "Fare Paid", "Age": "Passenger Age"},
            width=900, height=500)
st.plotly_chart(fig, use_container_width=True)
# --------------- □ FILTERED DATA TABLE (BOTTOM ROW) -----------------
st.markdown("### 🎯 Filter & Explore Data")
pclass_filter = st.selectbox("🔍 Select Passenger Class", sorted(data["Pclass"].unique()))
filtered_data = data[data["Pclass"] == pclass_filter]
st.dataframe(filtered_data)
# ------------------ □ STREAMLIT DASHBOARD END ------------------
```

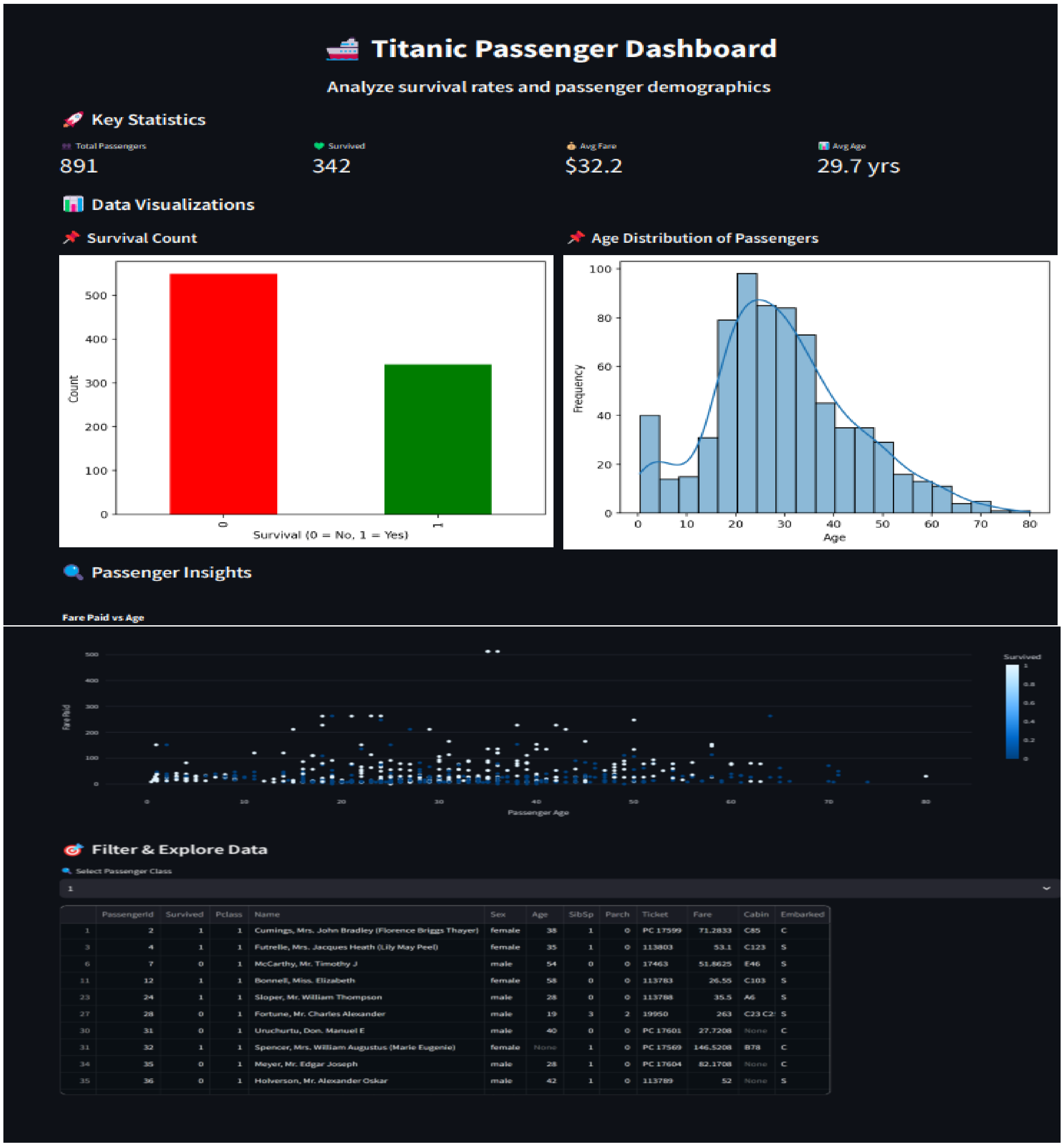
OUTPUT:-


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
COMMENTS
streamlit - visualization connect to data build charts & analyze data create dash board

```
PS D:\Jupyter Notebook\BDA> cd "D:\Jupyter Notebook\BDA\visualization connect to data build charts & analyze data create dash board"
PS D:\Jupyter Notebook\BDA\visualization connect to data build charts & analyze data create dash board> pip install pandas matplotlib seaborn plotly streamlit
```



```
PS D:\Jupyter Notebook\BDA\visualization connect to data build charts & analyze data create dash board> streamlit run dashboard.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://192.168.0.189:8501
```



Titanic Passenger Dashboard

Analyze survival rates and passenger demographics

Key Statistics

Total Passengers
891

Survived
342

Avg Fare
$32.2

Avg Age
29.7 yrs

Data Visualizations

Survival Count

Age Distribution of Passengers

Passenger Insights

Fare Paid vs Age

Filter & Explore Data

Select Passenger Class

1

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.2833	C85	C
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1	C123	S
6	7	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463	51.8625	E46	S
11	12	1	1	Bonnell, Miss. Elizabeth	female	58	0	0	113783	26.55	C103	S
23	24	1	1	Sloper, Mr. William Thompson	male	28	0	0	113788	35.5	A6	S
27	28	0	1	Fortune, Mr. Charles Alexander	male	19	3	2	19950	263	C23 C2	S
30	31	0	1	Uruchurtu, Don. Manuel E	male	40	0	0	PC 17601	27.7208	None	C
31	32	1	1	Spencer, Mrs. William Augustus (Marie Eugenie)	female	None	1	0	PC 17569	146.5208	B78	C
34	35	0	1	Meyer, Mr. Edgar Joseph	male	28	1	0	PC 17604	82.1708	None	C
35	36	0	1	Holverson, Mr. Alexander Oskar	male	42	1	0	113789	52	None	S