Final Project
Using SQL and Python

Dataset

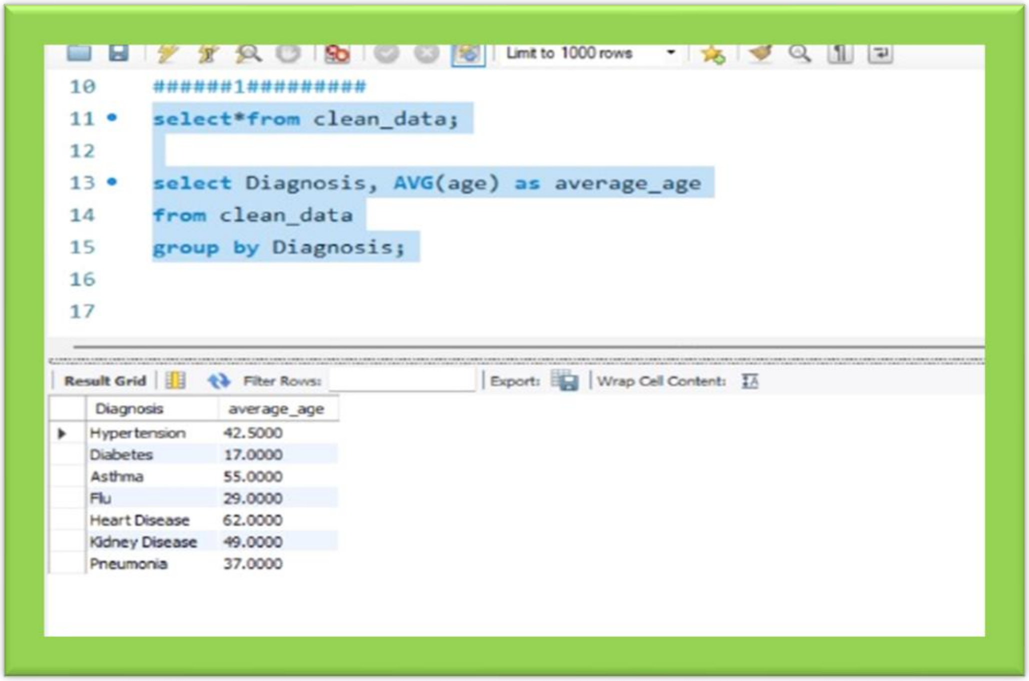
The dataset (hospital_patient_records.csv) contains the following columns:

- **PatientID**: Unique identifier for each patient.
- **Name**: Name of the patient.
- **Age**: Patient's age.
- **Gender**: Male or Female.
- **Diagnosis**: Primary diagnosis of the patient.
- **Medication**: Prescribed medication.
- **AdmissionDate**: Date of patient admission.
- **DischargeDate**: Date of discharge.
- **Doctor**: Assigned doctor’s name.
- **Department**: Hospital department (e.g., Cardiology, Orthopedics).
- **Status**: Patient status (e.g., Admitted, Discharged, Under Observation).

Data Exploration and Analysis (SQL and Python)

1. SQL Queries

- What is the **average age** of patients for each diagnosis?



```
######1#########
select*from clean_data;

select Diagnosis, AVG(age) as average_age
from clean_data
group by Diagnosis;
```

Diagnosis	average_age
Hypertension	42.5000
Diabetes	17.0000
Asthma	55.0000
Flu	29.0000
Heart Disease	62.0000
Kidney Disease	49.0000
Pneumonia	37.0000

Comment : The result average age of

'Hypertension' is '42.5000'

'Diabetes' is '17.0000"

Asthma' is '55.0000'

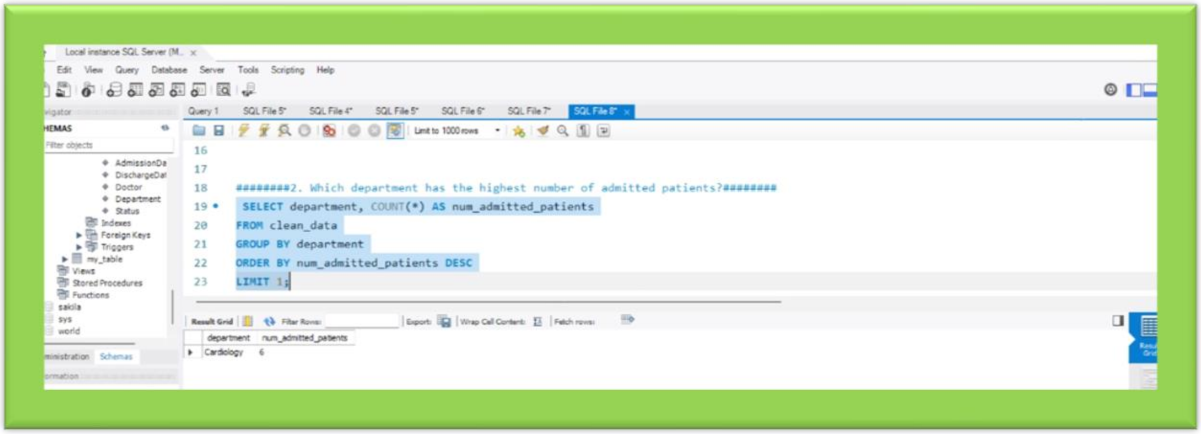
'Flu' is '29.0000'

'Heart Disease' is '62.0000'

'Kidney Disease' is '49.0000'

'Pneumonia' is '37.0000'

- Which department has the **highest number of admitted patients**?

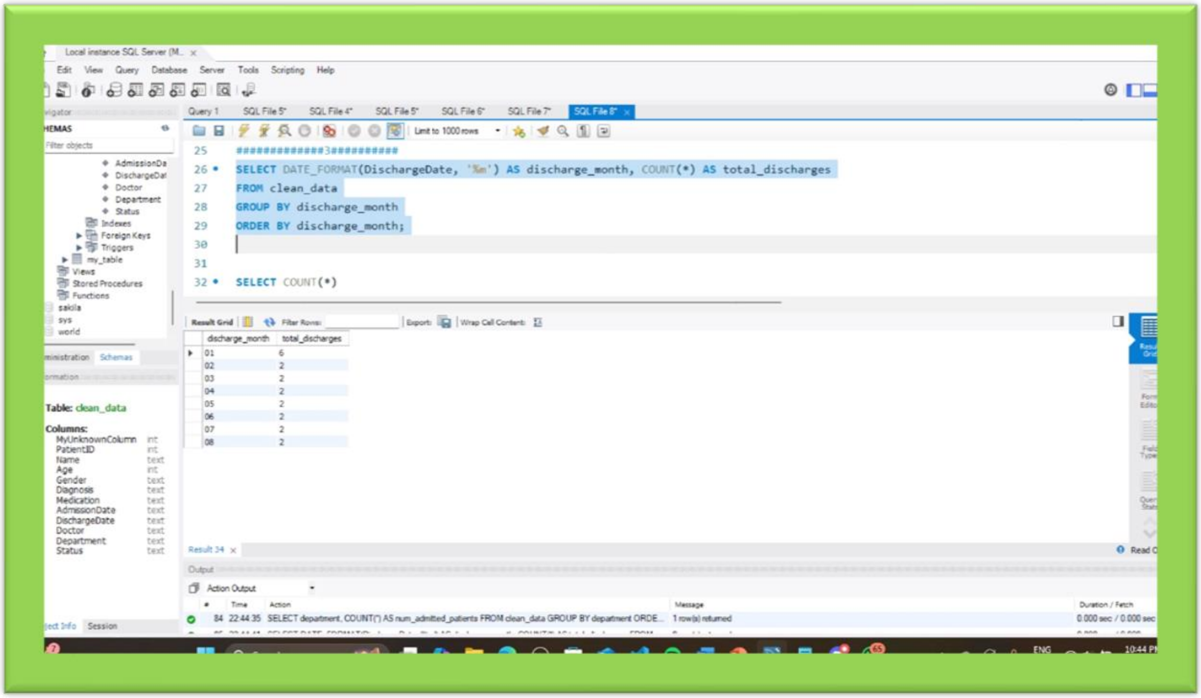


```
########2. Which department has the highest number of admitted patients?########
 SELECT department, COUNT(*) AS num_admitted_patients
FROM clean_data
GROUP BY department
ORDER BY num_admitted_patients DESC
LIMIT 1;
```

department	num_admitted_patients
Cardiology	6

Comment: department has the **highest number of admitted patients** :Cardiology

- How many patients have been **discharged** per month?



```
#############3##########
SELECT DATE_FORMAT(DischargeDate, '%m') AS discharge_month, COUNT(*) AS total_discharges
FROM clean_data
GROUP BY discharge_month
ORDER BY discharge_month;


SELECT COUNT(*)
```

discharge_month	total_discharges
01	6
02	2
03	2
04	2
05	2
06	2
07	2
08	2

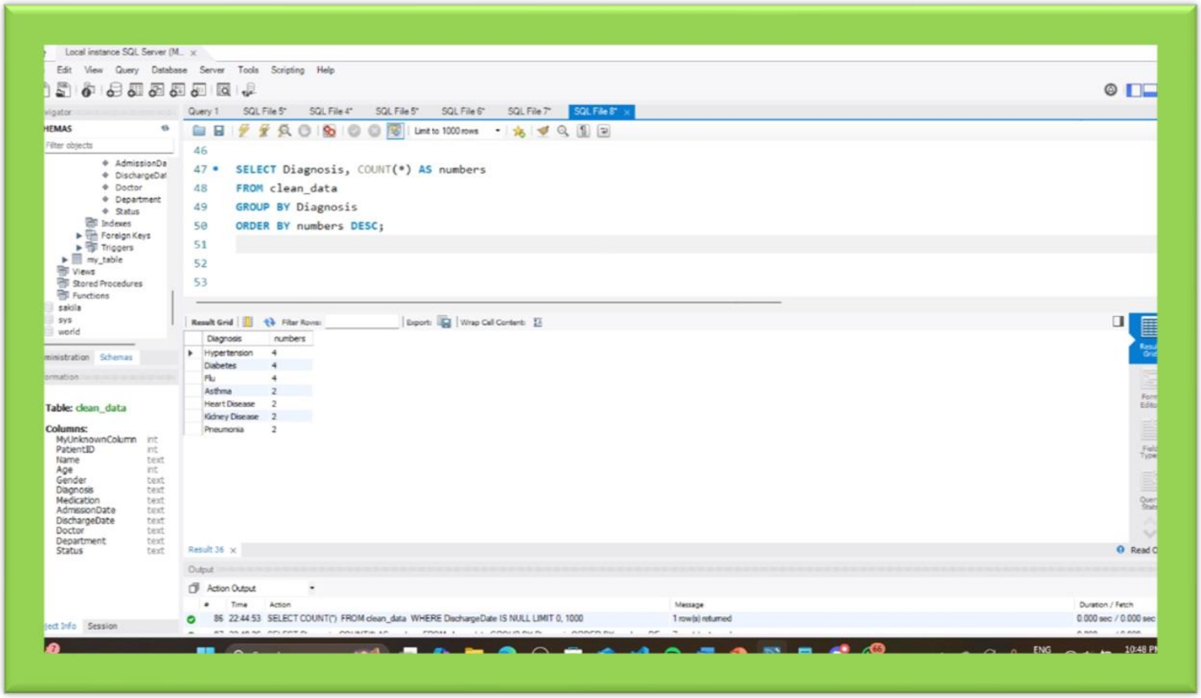
Table: clean_data

Columns:

MyUnknownColumn	int
PatientID	int
Name	text
Age	int
Gender	text
Diagnosis	text
Medication	text
AdmissionDate	text
DischargeDate	text
Doctor	text
Department	text
Status	text

```
#   Time      Action                                                                                       Message            Duration / Fetch
84  22:44:35  SELECT department, COUNT(*) AS num_admitted_patients FROM clean_data GROUP BY department ORDE...  1 row(s) returned  0.000 sec / 0.000 sec
```

- What is the **most common diagnosis** among patients?



```
SELECT Diagnosis, COUNT(*) AS numbers
FROM clean_data
GROUP BY Diagnosis
ORDER BY numbers DESC;
```

Diagnosis	numbers
Hypertension	4
Diabetes	4
Flu	4
Asthma	2
Heart Disease	2
Kidney Disease	2
Pneumonia	2

Table: clean_data

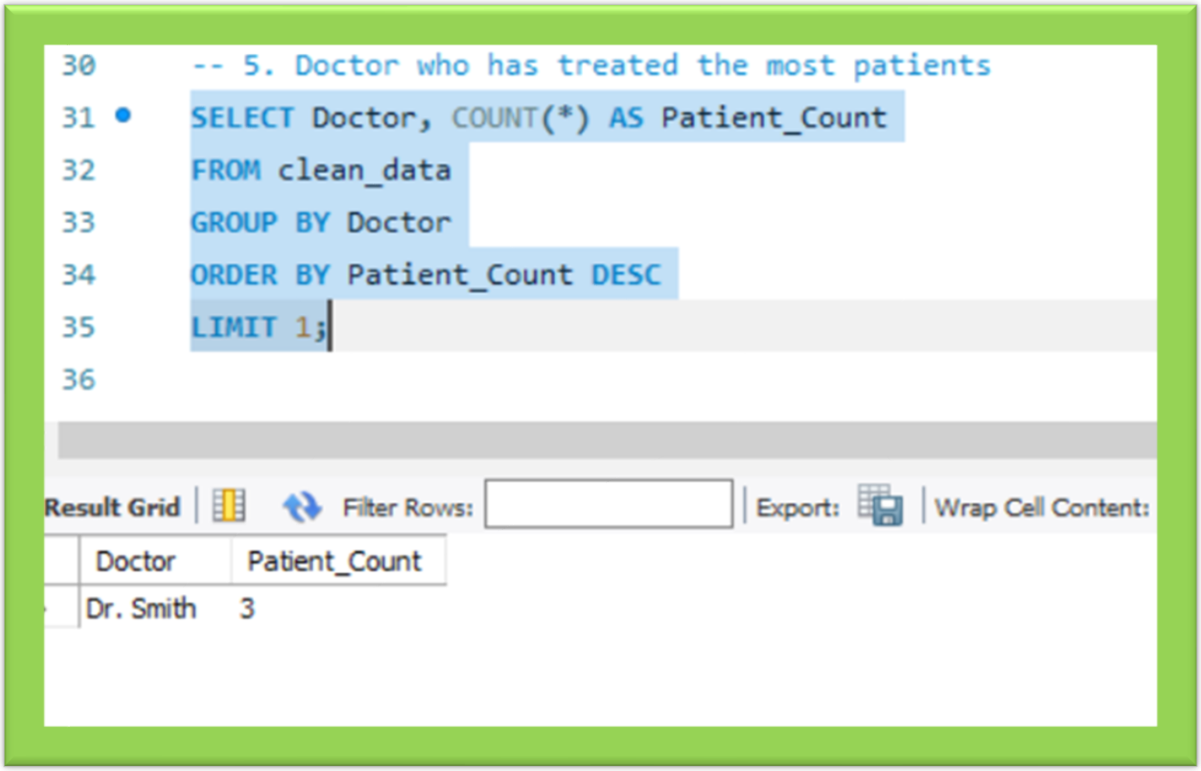
Columns:

MyUnknownColumn	int
PatientID	int
Name	text
Age	int
Gender	text
Diagnosis	text
Medication	text
AdmissionDate	text
DischargeDate	text
Doctor	text
Department	text
Status	text

```
#   Time      Action                                                            Message              Duration / Fetch
86  22:44:53  SELECT COUNT(*) FROM clean_data  WHERE DischargeDate IS NULL LIMIT 0, 1000   1 row(s) returned    0.000 sec / 0.000 sec
```

Comment: **most common diagnosis** among patients 'Hypertension'

- Which doctor has treated the **most patients**?



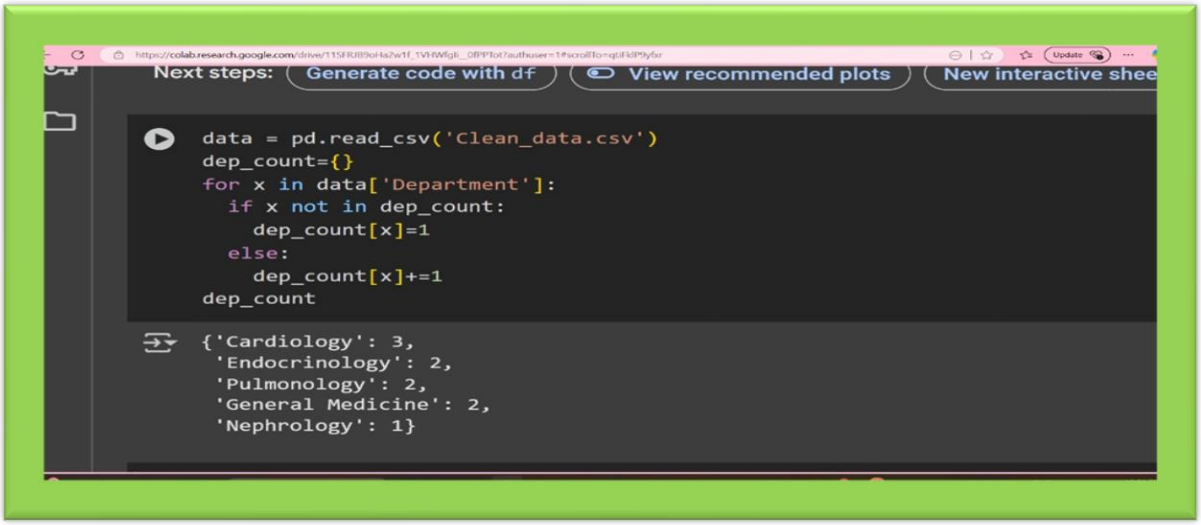
```
-- 5. Doctor who has treated the most patients
SELECT Doctor, COUNT(*) AS Patient_Count
FROM clean_data
GROUP BY Doctor
ORDER BY Patient_Count DESC
LIMIT 1;
```

Doctor	Patient_Count
Dr. Smith	3

Comment: doctor has treated the **most patients 'Dr. Smith'**

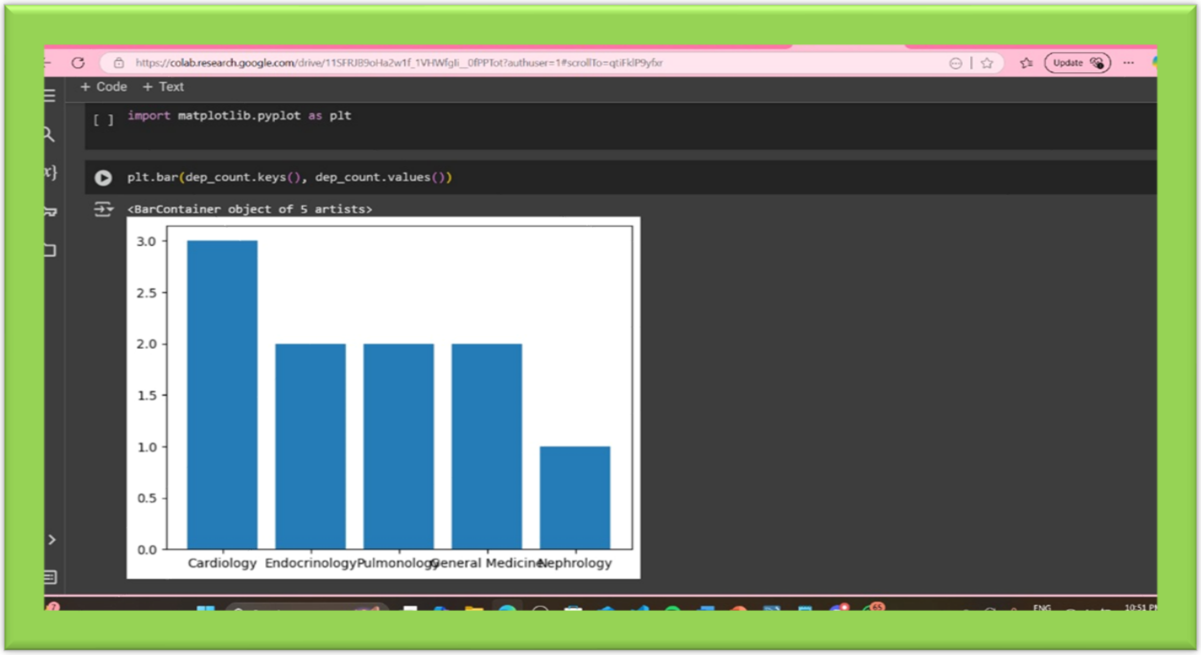
2. Python Analysis

- **Visualize the number of patients per department** using a bar chart.



```
data = pd.read_csv('Clean_data.csv')
dep_count={}
for x in data['Department']:
  if x not in dep_count:
    dep_count[x]=1
  else:
    dep_count[x]+=1
dep_count
```

```
{'Cardiology': 3,
 'Endocrinology': 2,
 'Pulmonology': 2,
 'General Medicine': 2,
 'Nephrology': 1}
```



```
import matplotlib.pyplot as plt
```

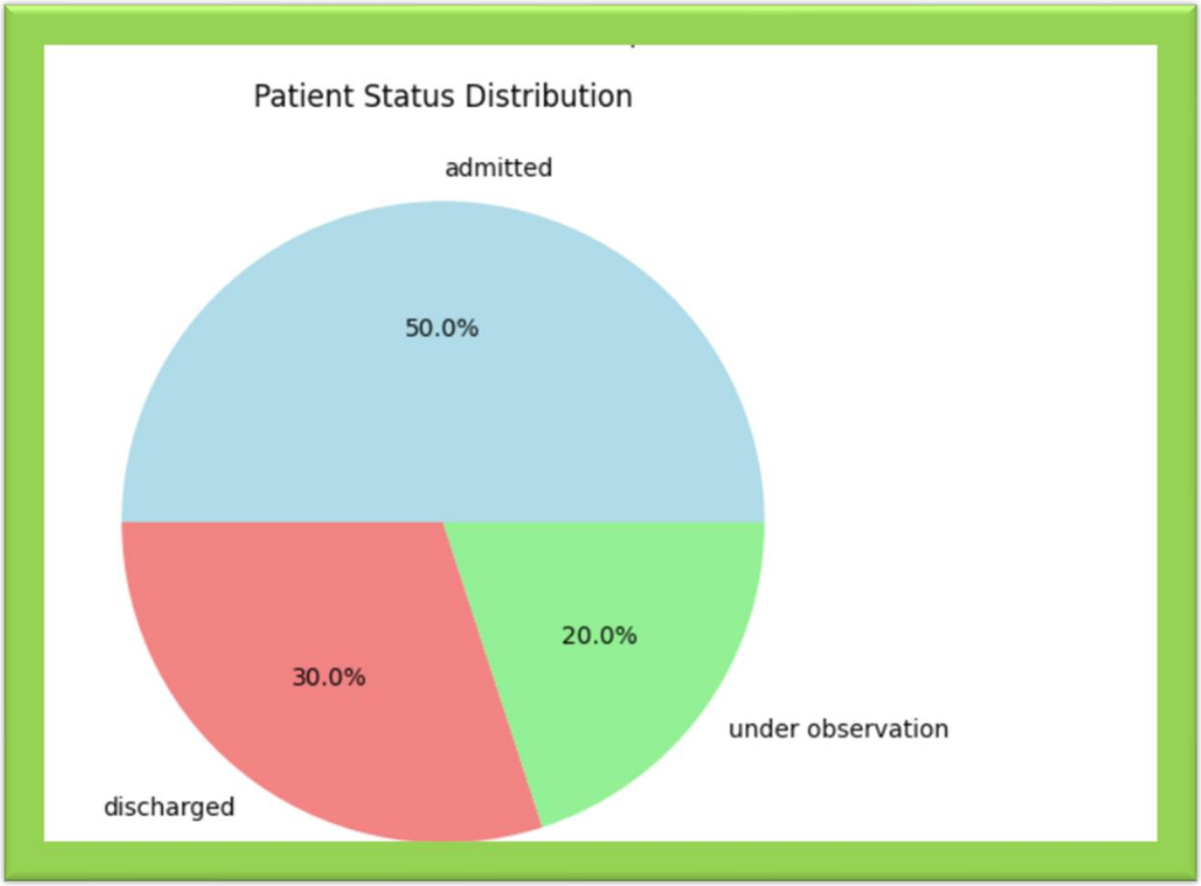
```
plt.bar(dep_count.keys(), dep_count.values())
```

```
<BarContainer object of 5 artists>
```

Comment: Cardiology has the highest number of patients (3), followed by Endocrinology, Pulmonology, and General Medicine (each with 2 patients).

Nephrology has the least number of patients (1)

- **Create a pie chart** showing the distribution of patient statuses (Admitted, Discharged, Under Observation).

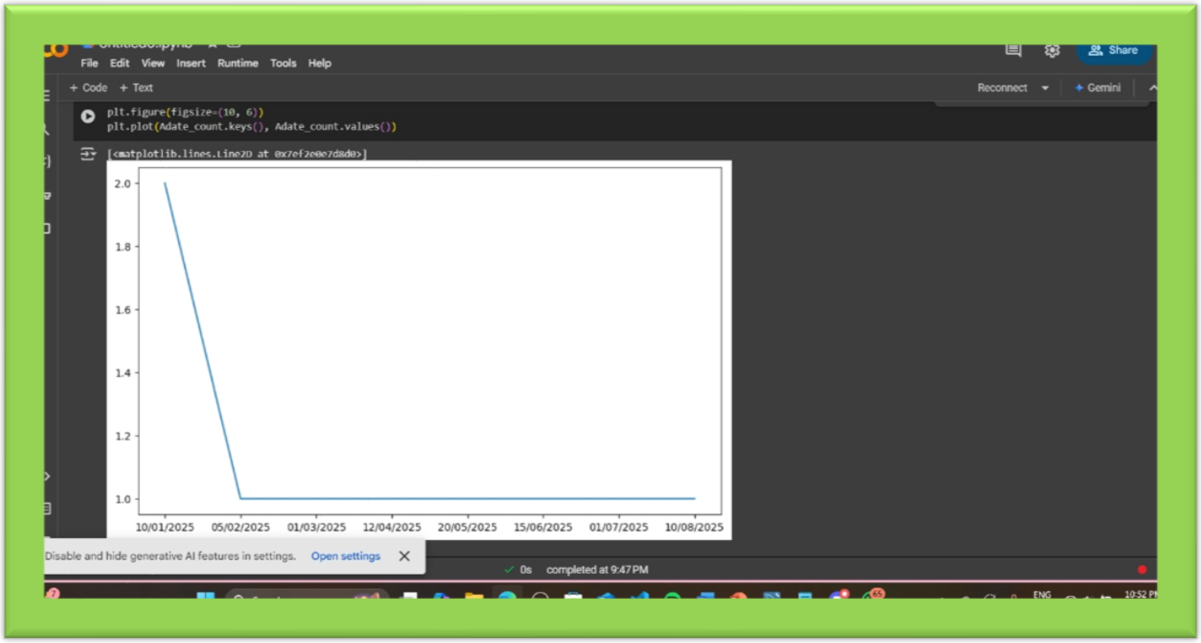


50% of patients are admitted, indicating a high hospitalization rate.

30% of patients are discharged

20% of patients are under observation

Generate a line chart showing **monthly hospital admissions trends**.



```
plt.figure(figsize=(10, 6))
plt.plot(Adate_count.keys(), Adate_count.values())
```

```
[<matplotlib.lines.Line2D at 0x7ef2e0e7d8d0>]
```

Theoretical Questions

1. Data Cleaning

- What are the common issues you might encounter in a messy dataset?

  1. Missing values
  2. Duplicate records
  3. Inconsistent data types
  4. Outliers
  5. Formatting errors

- How would you handle missing values in a dataset?

  1. Remove rows or columns with too many missing values
  2. Fill missing values with the mean, median, or mode

- What is the importance of data type consistency in data analysis?

  1. Ensuring accurate calculations
  2. Preventing errors during analysis and modeling
  3. Improving performance and efficient memory usage

2. SQL Queries

- What is the difference between INNER JOIN and LEFT JOIN?

    1. INNER JOIN returns only the matching records from both tables.
    2. LEFT JOIN returns all records from the left table and matching records from the right table

- How would you use the GROUP BY clause to aggregate data?

    1. The GROUP BY clause groups data by specific columns

- What is the purpose of the HAVING clause in SQL?

    1. The HAVING clause to apply conditions to the aggregated data

Python Analysis

- How would you use Pandas to clean a dataset with mixed data types?

    1.Convert columns with wrong data types to consistent types

    2. Handle invalid values
    3. Fill missing values or handle errors

- What are the benefits of using visualizations in data analysis?

    1. Making patterns, trends, and outliers easy to identify
    2. Providing a clearer understanding