Hospital Management System

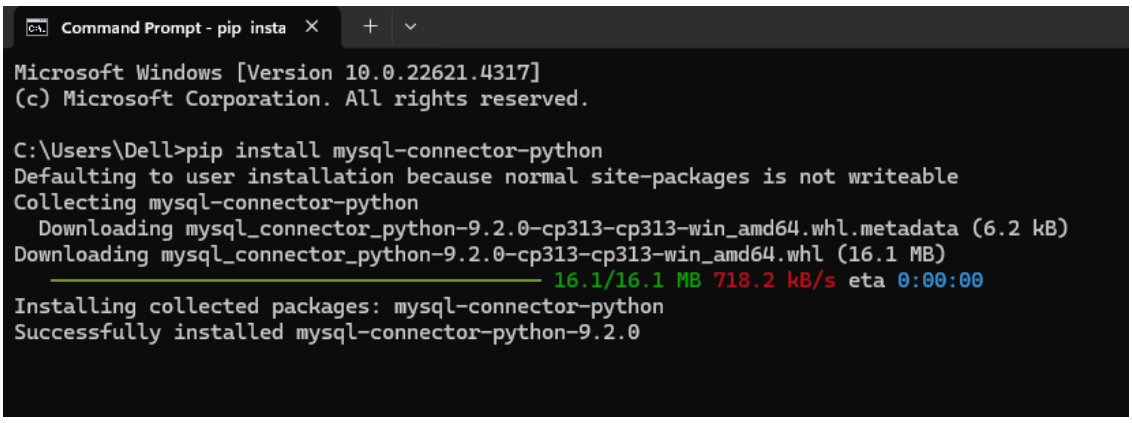
Project Description:

Developed a Hospital Management System using Python (Tkinter) for the frontend and MySQL for the backend. The system allows hospitals to manage patient records, including adding, updating, deleting, and viewing patient details. It ensures efficient record-keeping and retrieval of patient information.

Technologies Used: Python (Tkinter), MySQL, SQL, Database Management

Steps Include:

- Open the Python IDE and type the code for the hospital management system.
- Before this ,for connecting the python and MySQL we need to install the library **mysql.connector**
- Open the command prompt,type the command
  pip install mysql-connector-python
- After the successful installation , it shows



- Then import the mysql.connector library in the code.
- And then import **tKinter** library for Graphical User Interface(GUI).

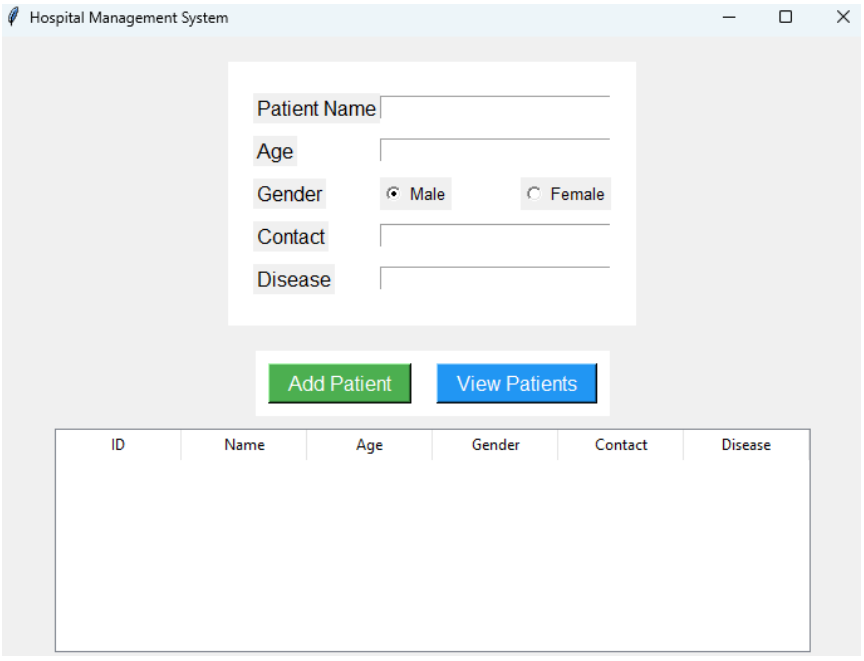
- After importing these library ,type the code the

projectHMS.py - C:\Users\Dell\projectHMS.py (3.11.4)

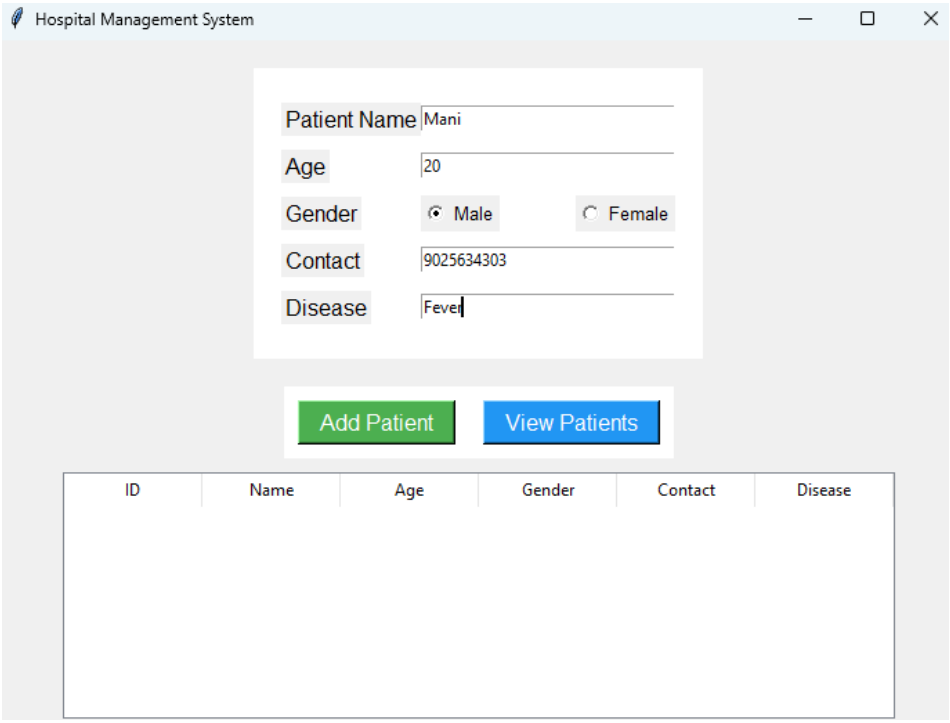
File Edit Format Run Options Window Help

```
import mysql.connector
from tkinter import *
from tkinter import messagebox, ttk

# Database Connection
conn = mysql.connector.connect(
    host='localhost',
    user='root',
    password='Mukund200@',
    database='hospital_management'
)
cursor = conn.cursor()

# Create Table if not exists
cursor.execute('''CREATE TABLE IF NOT EXISTS patients (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100),
    age INT,
    gender VARCHAR(10),
    contact VARCHAR(20),
    disease VARCHAR(100)
)''')
conn.commit()

# GUI Setup
root = Tk()
root.title("Hospital Management System")
root.geometry("700x500")
root.configure(bg="#f0f0f0")

frame = Frame(root, padx=20, pady=20, bg="#ffffff")
frame.pack(pady=20)

Label(frame, text="Patient Name", font=("Arial", 12)).grid(row=0, column=0, pady=5, sticky=W)
name_entry = Entry(frame, width=30)
name_entry.grid(row=0, column=1, pady=5)

Label(frame, text="Age", font=("Arial", 12)).grid(row=1, column=0, pady=5, sticky=W)
age_entry = Entry(frame, width=30)
age_entry.grid(row=1, column=1, pady=5)
```

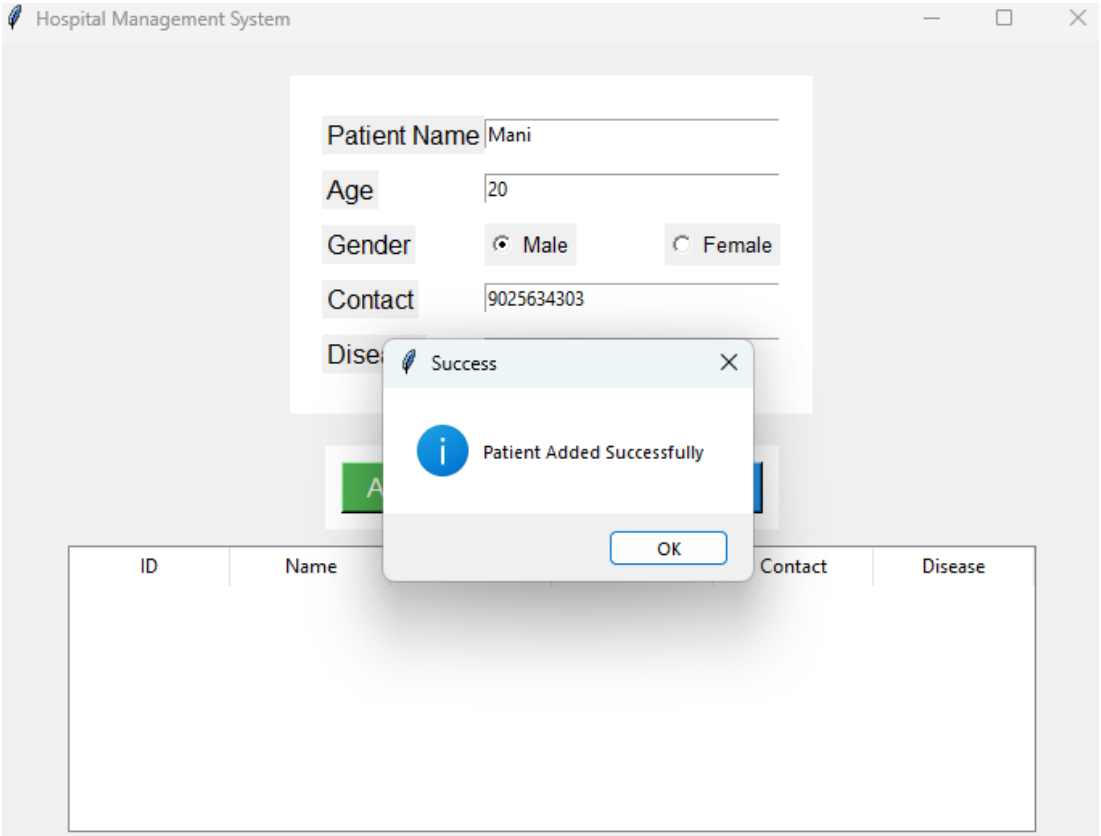
- After typing the code go to menu bar Click on **Run** option to run the code.
- After the successful compilation , it opens the GUI page



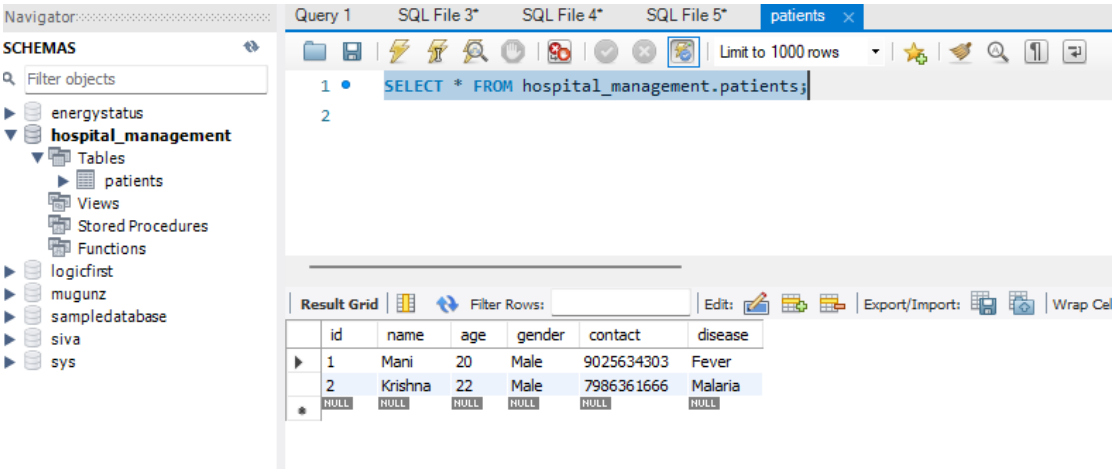
- Fill the details and Click on **Add Patient** button



- Then it shows **Patient Added successfully**



- We can add many details like this
- These datas are to be stored in a **MySQL** database.



- Using MySQL database we can manage patient records, including adding, updating, deleting, and viewing patient details.