

Module 12: Building Database Apps with PostgreSQL & Python

1. List of DB in PostgreSQL

```
SQL Shell (psql)
Server: [localhost]:
Database: [postgres]:
Port: [5432]:
Username: [postgres]:
Password for user postgres:
psql (17.6)
WARNING: Console code page (437) differs from windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

postgres=# \l

```

Name	Owner	Encoding	Locale Provider	Collate	Ctype	Locale	ICU Rules	Access privileges
postgres	postgres	UTF8	libc	English_United States.1252	English_United States.1252			
student	postgres	UTF8	libc	English_United States.1252	English_United States.1252			
template0	postgres	UTF8	libc	English_United States.1252	English_United States.1252			=c/postgres +
template1	postgres	UTF8	libc	English_United States.1252	English_United States.1252			postgres=CTC/postgres +
								=c/postgres +
								postgres=CTC/postgres

```
(4 rows)

postgres=#
```

2. Create Database

```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Password for user postgres:
psql (17.6)
WARNING: Console code page (437) differs from Windows code page (1252)
8-bit characters might not work correctly. See psql reference
page "Notes for Windows users" for details.
Type "help" for help.

postgres=# \l
      List of databases

```

Name	Owner	Encoding	Locale Provider	Collate	Ctype	Locale	ICU Rules	Access privileges
postgres	postgres	UTF8	libc	English_United States.1252	English_United States.1252			
student	postgres	UTF8	libc	English_United States.1252	English_United States.1252			
template0	postgres	UTF8	libc	English_United States.1252	English_United States.1252			+c/postgres postgres-CTC/postgres
template1	postgres	UTF8	libc	English_United States.1252	English_United States.1252			+c/postgres postgres-CTC/postgres

```
(4 rows)

postgres=# create database test;
CREATE DATABASE
postgres=#
```

```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Password for user postgres:
psql (17.6)
WARNING: Console code page (437) differs from Windows code page (1252)
8-bit characters might not work correctly. See psql reference
page "Notes for Windows users" for details.
Type "help" for help.

postgres=# \l
      List of databases

```

Name	Owner	Encoding	Locale Provider	Collate	Ctype	Locale	ICU Rules	Access privileges
postgres	postgres	UTF8	libc	English_United States.1252	English_United States.1252			
student	postgres	UTF8	libc	English_United States.1252	English_United States.1252			
template0	postgres	UTF8	libc	English_United States.1252	English_United States.1252			+c/postgres postgres-CTC/postgres
template1	postgres	UTF8	libc	English_United States.1252	English_United States.1252			+c/postgres postgres-CTC/postgres
test	postgres	UTF8	libc	English_United States.1252	English_United States.1252			

```
(5 rows)

postgres=#
```

3. Change database

```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Password for user postgres:
psql (17.6)
WARNING: Console code page (437) differs from Windows code page (1252)
8-bit characters might not work correctly. See psql reference
page "Notes for Windows users" for details.
Type "help" for help.

postgres=# \l
      List of databases
  Name | Owner | Encoding | Locale Provider | Collate | Ctype | Locale | ICU Rules | Access privileges
-----+-----+-----+-----+-----+-----+-----+-----+-----
postgres | postgres | UTF8 | libc | English_United States.1252 | English_United States.1252 |  |  | 
student | postgres | UTF8 | libc | English_United States.1252 | English_United States.1252 |  |  | 
template0 | postgres | UTF8 | libc | English_United States.1252 | English_United States.1252 |  |  | +c/postgres
template1 | postgres | UTF8 | libc | English_United States.1252 | English_United States.1252 |  |  | +postgres=CTC/postgres
(4 rows)

postgres=# create database test;
CREATE DATABASE
postgres=# \l
      List of databases
  Name | Owner | Encoding | Locale Provider | Collate | Ctype | Locale | ICU Rules | Access privileges
-----+-----+-----+-----+-----+-----+-----+-----+-----
postgres | postgres | UTF8 | libc | English_United States.1252 | English_United States.1252 |  |  | 
student | postgres | UTF8 | libc | English_United States.1252 | English_United States.1252 |  |  | 
template0 | postgres | UTF8 | libc | English_United States.1252 | English_United States.1252 |  |  | +c/postgres
template1 | postgres | UTF8 | libc | English_United States.1252 | English_United States.1252 |  |  | +postgres=CTC/postgres
test | postgres | UTF8 | libc | English_United States.1252 | English_United States.1252 |  |  | 
(5 rows)

postgres=# \c test
You are now connected to database "test" as user "postgres".
test=#
```

4. Deleting the database

```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Password for user postgres:
psql (17.6)
WARNING: Console code page (437) differs from Windows code page (1252)
8-bit characters might not work correctly. See psql reference
page "Notes for Windows users" for details.
Type "help" for help.

postgres=# \l
      List of databases
  Name | Owner | Encoding | Locale Provider | Collate | Ctype | Locale | ICU Rules | Access privileges
-----+-----+-----+-----+-----+-----+-----+-----+-----
postgres | postgres | UTF8 | libc | English_United States.1252 | English_United States.1252 |  |  | 
student | postgres | UTF8 | libc | English_United States.1252 | English_United States.1252 |  |  | 
template0 | postgres | UTF8 | libc | English_United States.1252 | English_United States.1252 |  |  | +c/postgres
template1 | postgres | UTF8 | libc | English_United States.1252 | English_United States.1252 |  |  | +postgres=CTC/postgres
(4 rows)

postgres=# create database test;
CREATE DATABASE
postgres=# \l
      List of databases
  Name | Owner | Encoding | Locale Provider | Collate | Ctype | Locale | ICU Rules | Access privileges
-----+-----+-----+-----+-----+-----+-----+-----+-----
postgres | postgres | UTF8 | libc | English_United States.1252 | English_United States.1252 |  |  | 
student | postgres | UTF8 | libc | English_United States.1252 | English_United States.1252 |  |  | 
template0 | postgres | UTF8 | libc | English_United States.1252 | English_United States.1252 |  |  | +c/postgres
template1 | postgres | UTF8 | libc | English_United States.1252 | English_United States.1252 |  |  | +postgres=CTC/postgres
test | postgres | UTF8 | libc | English_United States.1252 | English_United States.1252 |  |  | 
(5 rows)

postgres=# \c test
You are now connected to database "test" as user "postgres".
test=# \c postgres
You are now connected to database "postgres" as user "postgres".
postgres=# drop database test;
DROP DATABASE
postgres=#
```

5. Create a table and data in the table

```
SQL Shell (psql)
postgres=# create database test;
CREATE DATABASE
postgres=# \c test
You are now connected to database "test" as user "postgres".
test=# create table student(id int,name text,age int);
CREATE TABLE
test=# \d
          List of relations
Schema | Name  | Type  | Owner
-----+-----+-----+-----
public | student | table | postgres
(1 row)

test=#
```

```
SQL Shell (psql)
postgres=# create database test;
CREATE DATABASE
postgres=# \c test
You are now connected to database "test" as user "postgres".
test=# create table student(id int,name text,age int);
CREATE TABLE
test=# \d
          List of relations
Schema | Name  | Type  | Owner
-----+-----+-----+-----
public | student | table | postgres
(1 row)

test=# insert into student(id,name,age) value(1,'sanjeev',30);
ERROR:  syntax error at or near "value"
LINE 1: insert into student(id,name,age) value(1,'sanjeev',30);
                                           ^
test=# insert into student(id,name,age) values(1,'sanjeev',30);
INSERT 0 1
test=#
```

6. Retrieving data from the database and deleting content from it.

```
SQL Shell (psql)
test=# insert into student(id,name,age) values(1,'sanjeev',30);
INSERT 0 1
test=# insert into student(id,name,age) values(2,'sanjeev 2',30);
INSERT 0 1
test=# insert into student(id,name,age) values(3,'sanjeev 3',30);
INSERT 0 1
test=# select * from student;
 id | name      | age
----+-----+----
  1 | sanjeev   |  30
  2 | sanjeev 2 |  30
  3 | sanjeev 3 |  30
(3 rows)

test=#
```

```
SQL Shell (psql)
test=# insert into student(id,name,age) values(1,'sanjeev',30);
INSERT 0 1
test=# insert into student(id,name,age) values(2,'sanjeev 2',30);
INSERT 0 1
test=# insert into student(id,name,age) values(3,'sanjeev 3',30);
INSERT 0 1
test=# select * from student;
 id | name      | age
----+-----+----
  1 | sanjeev   |  30
  2 | sanjeev 2 |  30
  3 | sanjeev 3 |  30
(3 rows)

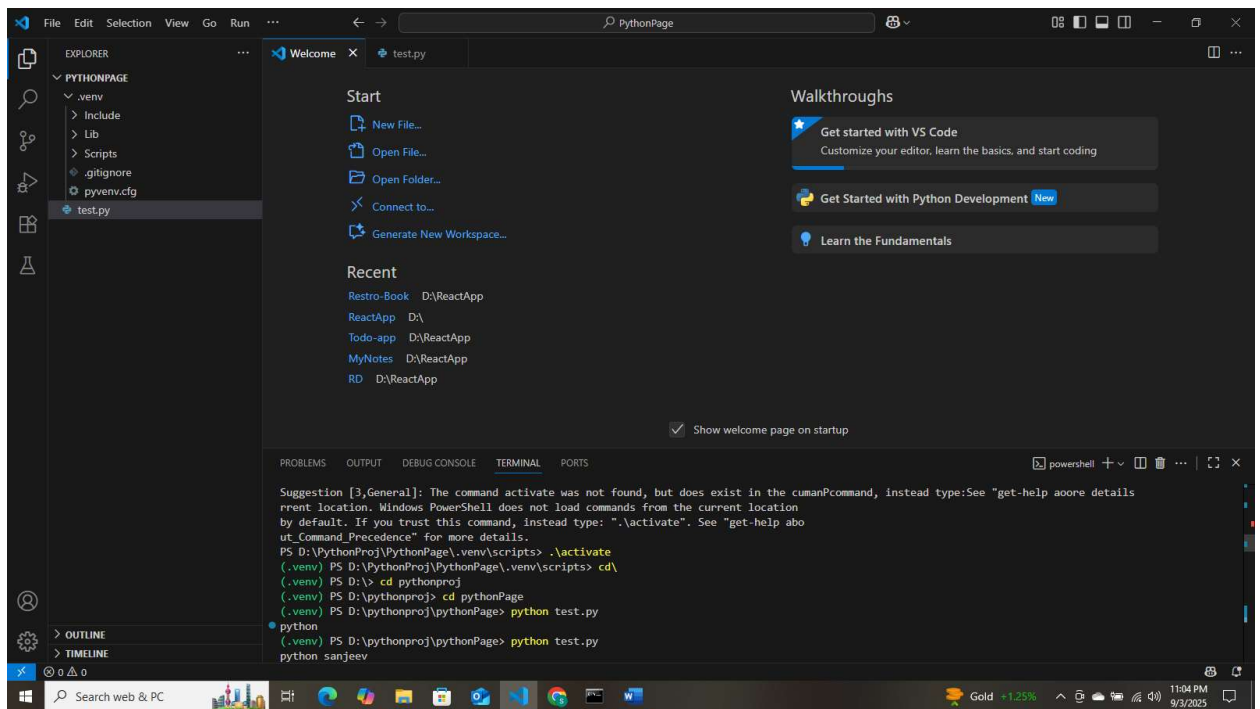
test=# delete from student where id=3;
DELETE 1
test=# select * from student;
 id | name      | age
----+-----+----
  1 | sanjeev   |  30
  2 | sanjeev 2 |  30
(2 rows)

test=#
```

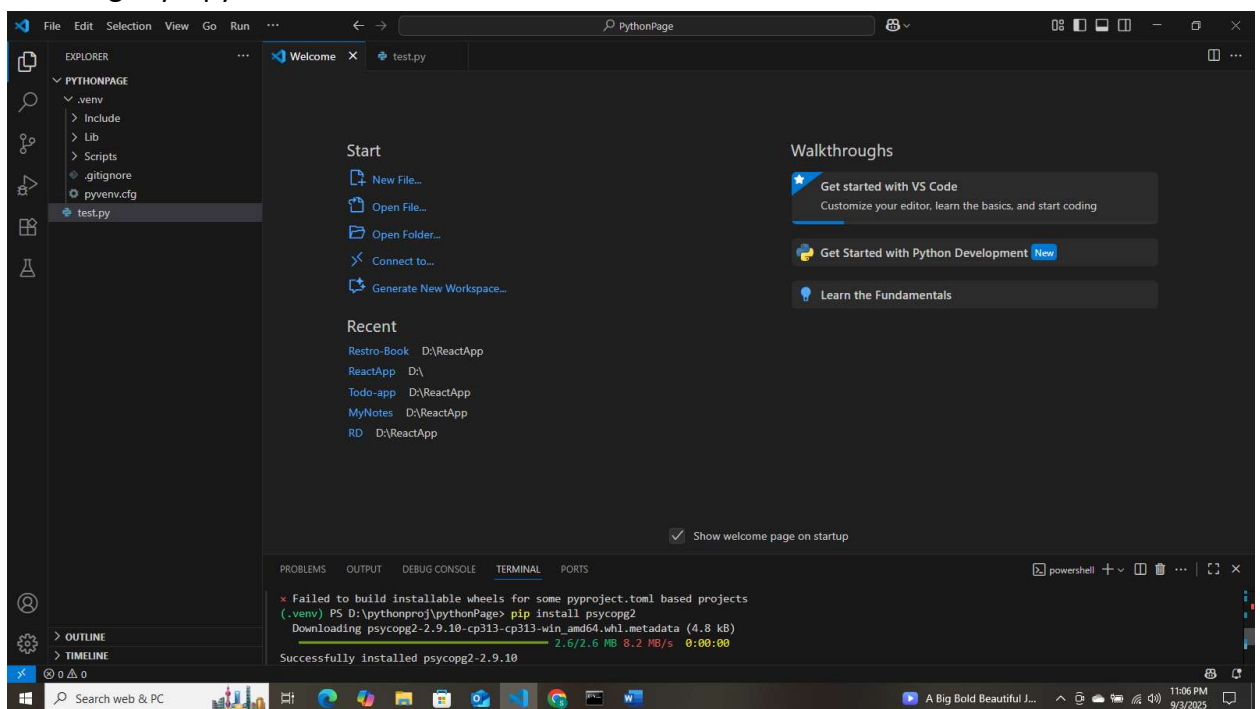
7. Setting Up Virtual env

The screenshot shows a Windows 10 desktop with a VS Code editor window open. The editor is displaying a Python project named 'PythonPage'. The Explorer sidebar on the left shows a file tree with 'PYTHONPAGE', '.venv', and 'test.py'. The main editor area displays the 'test.py' file with a 'Start' button and a 'Walkthroughs' section. The bottom status bar shows the file path 'D:\PythonProj\PythonPage\test.py' and the Python version 'Python 3.9.0'.

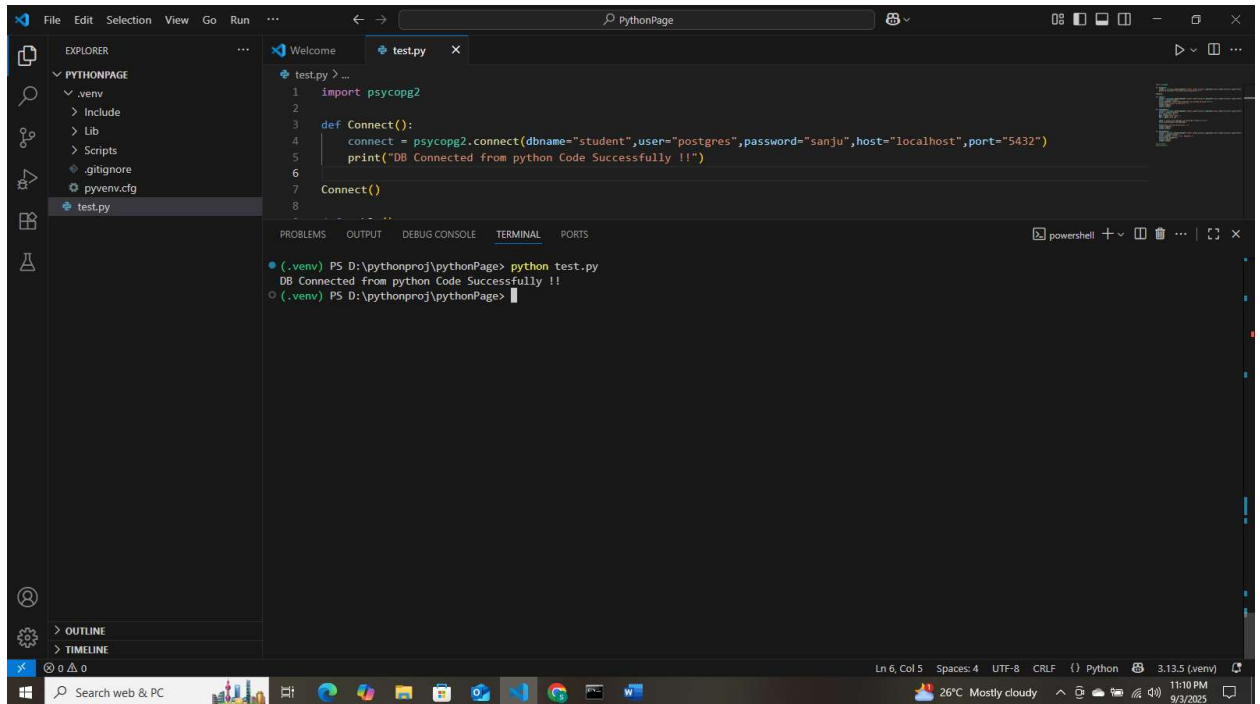
The screenshot shows the Visual Studio Code interface with a Python project. The Explorer pane on the left shows a file structure with 'test.py'. The main editor area displays the terminal output of a virtual environment setup, including package installations like distlib, filelock, platformdirs, and virtualenv. The bottom status bar shows the file encoding as UTF-8.



8. Installing Psycopy2



9. Connecting DB from Python



The screenshot shows the Visual Studio Code editor with a file named `test.py` open. The code in the editor is as follows:

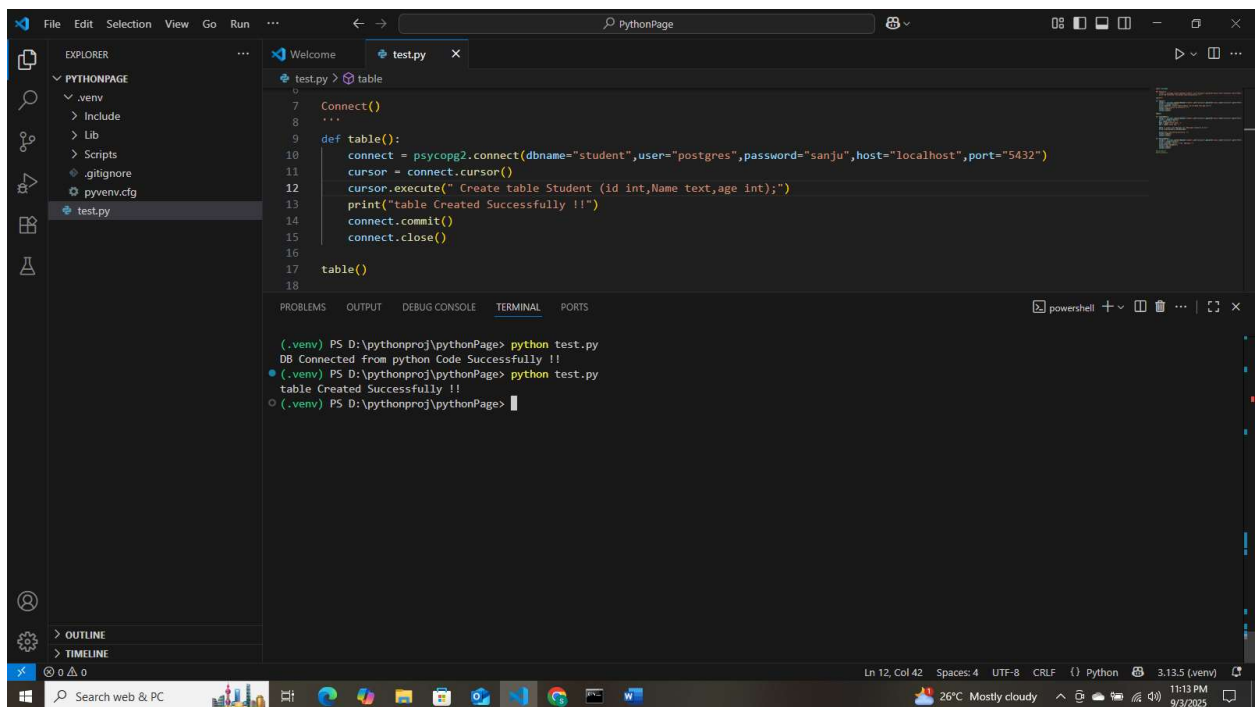
```
1 import psycopg2
2
3 def Connect():
4     connect = psycopg2.connect(dbname="student",user="postgres",password="sanju",host="localhost",port="5432")
5     print("DB Connected from python Code Successfully !!")
6
7 Connect()
8
```

The terminal at the bottom shows the execution of the script:

```
(.venv) PS D:\pythonproj\pythonPage> python test.py
DB Connected from python Code Successfully !!
(.venv) PS D:\pythonproj\pythonPage>
```

The Explorer sidebar on the left shows the project structure for `PYTHONPAGE`, including `.venv`, `include`, `Lib`, `Scripts`, `.gitignore`, and `pyvenv.cfg`. The `test.py` file is selected.

10. Create a table using Python



The screenshot shows the Visual Studio Code editor with the `test.py` file updated to include a `table()` function. The code is as follows:

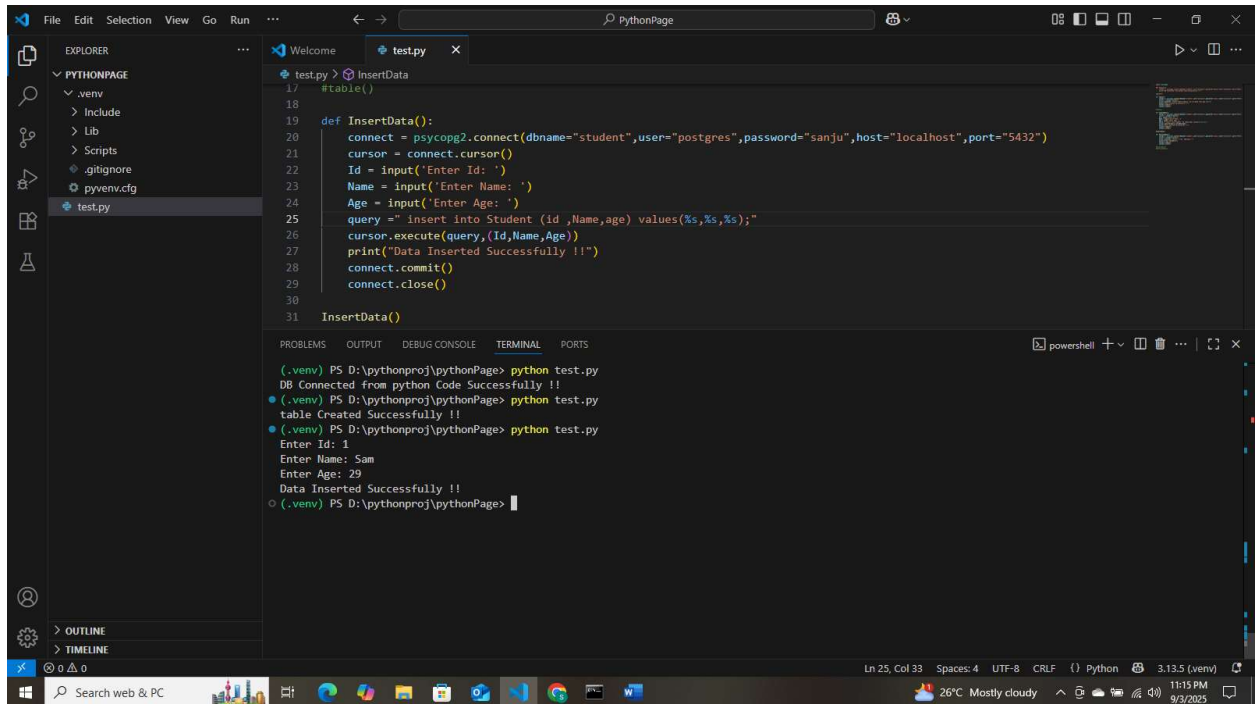
```
6
7 Connect()
8 '''
9 def table():
10     connect = psycopg2.connect(dbname="student",user="postgres",password="sanju",host="localhost",port="5432")
11     cursor = connect.cursor()
12     cursor.execute(" Create table Student (id int,Name text,age int);")
13     print("table Created Successfully !!")
14     connect.commit()
15     connect.close()
16
17 table()
18
```

The terminal shows the execution of the script twice:

```
(.venv) PS D:\pythonproj\pythonPage> python test.py
DB Connected from python Code Successfully !!
(.venv) PS D:\pythonproj\pythonPage> python test.py
table Created Successfully !!
(.venv) PS D:\pythonproj\pythonPage>
```

The Explorer sidebar on the left shows the project structure for `PYTHONPAGE`, including `.venv`, `include`, `Lib`, `Scripts`, `.gitignore`, and `pyvenv.cfg`. The `test.py` file is selected.

11. Inserting the data using Python



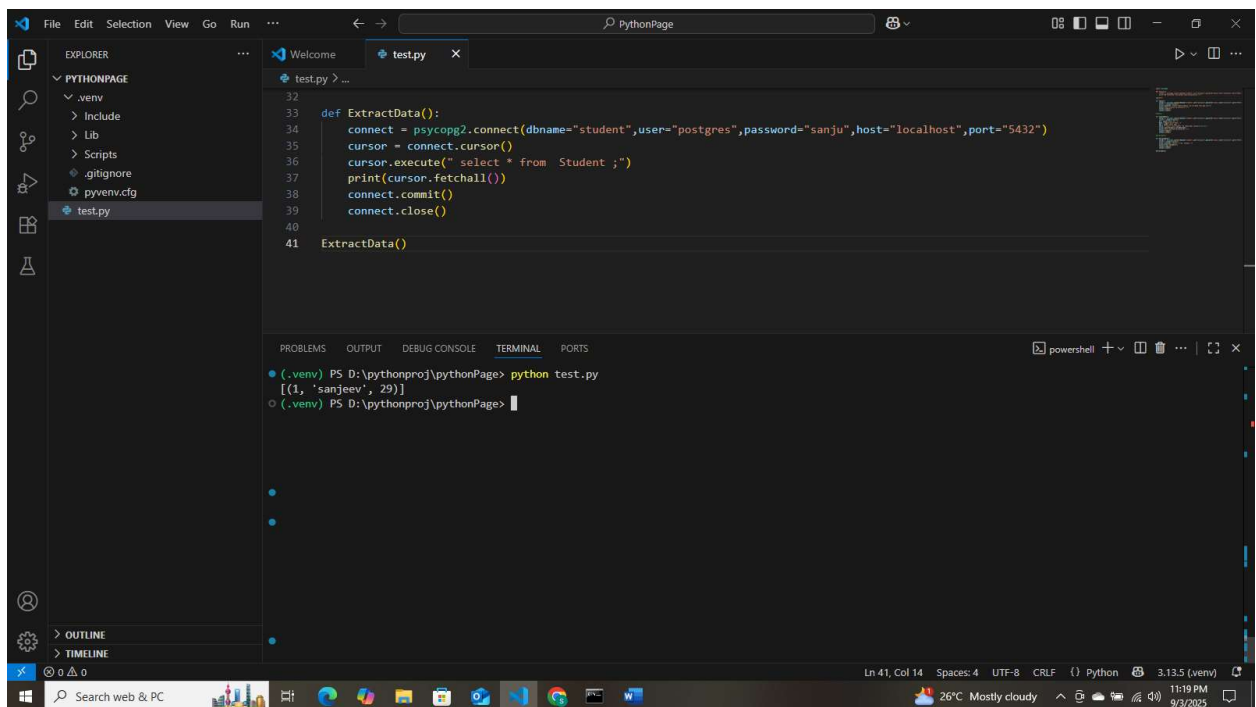
The screenshot shows a Visual Studio Code editor with a Python file named `test.py`. The code defines a function `InsertData()` that connects to a PostgreSQL database using `psycopg2`, prompts the user for an ID, name, and age, and inserts the data into a table named `Student`. The terminal output shows the successful execution of the script, including the database connection, table creation, and data insertion.

```
17 def InsertData():
18     #table()
19
20     connect = psycopg2.connect(dbname="student",user="postgres",password="sanju",host="localhost",port="5432")
21     cursor = connect.cursor()
22     Id = input('Enter Id: ')
23     Name = input('Enter Name: ')
24     Age = input('Enter Age: ')
25     query = " insert into Student (id ,Name,age) values(%s,%s,%s);"
26     cursor.execute(query,(Id,Name,Age))
27     print("Data Inserted Successfully !!")
28     connect.commit()
29     connect.close()
30
31 InsertData()
```

Terminal Output:

```
(.venv) PS D:\pythonproj\pythonPage> python test.py
DB Connected from python Code Successfully !!
(.venv) PS D:\pythonproj\pythonPage> python test.py
table Created Successfully !!
(.venv) PS D:\pythonproj\pythonPage> python test.py
Enter Id: 1
Enter Name: Sam
Enter Age: 29
Data Inserted Successfully !!
(.venv) PS D:\pythonproj\pythonPage>
```

12. Extracting the data from the table in Python



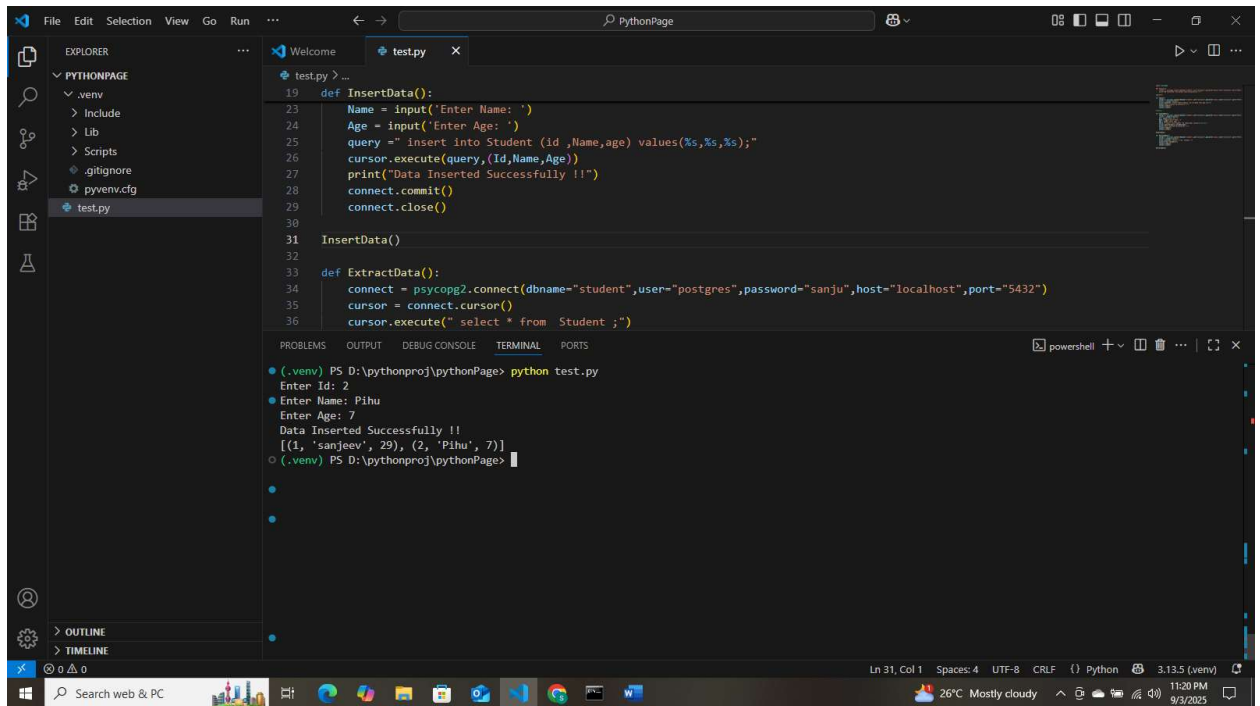
The screenshot shows a Visual Studio Code editor with a Python file named `test.py`. The code defines a function `ExtractData()` that connects to a PostgreSQL database using `psycopg2`, executes a `SELECT` query to retrieve all data from the `Student` table, and prints the results. The terminal output shows the successful execution of the script, including the database connection, query execution, and data extraction.

```
32 def ExtractData():
33
34     connect = psycopg2.connect(dbname="student",user="postgres",password="sanju",host="localhost",port="5432")
35     cursor = connect.cursor()
36     cursor.execute(" select * from Student ;")
37     print(cursor.fetchall())
38     connect.commit()
39     connect.close()
40
41 ExtractData()
```

Terminal Output:

```
(.venv) PS D:\pythonproj\pythonPage> python test.py
[(1, 'sanjeev', 29)]
(.venv) PS D:\pythonproj\pythonPage>
```

13. Adding the input from the User.



The screenshot shows a Visual Studio Code editor window with a Python file named `test.py` open. The file contains two functions: `InsertData()` and `ExtractData()`. The `InsertData()` function prompts the user for a name and age, constructs an SQL insert query, and executes it using a PostgreSQL connection. The `ExtractData()` function connects to the same database and retrieves all data from the `Student` table. The terminal at the bottom shows the execution of `python test.py`, where the user enters `2` for the ID, `Pihu` for the name, and `7` for the age. The output shows the data being inserted successfully and then retrieved as a list of tuples.

```
19 def InsertData():
23     Name = input('Enter Name: ')
24     Age = input('Enter Age: ')
25     query = "insert into Student (id ,Name,age) values(%s,%s,%s);"
26     cursor.execute(query,(Id,Name,Age))
27     print("Data Inserted Successfully !!")
28     connect.commit()
29     connect.close()
30
31 InsertData()
32
33 def ExtractData():
34     connect = psycopg2.connect(dbname="student",user="postgres",password="sanju",host="localhost",port="5432")
35     cursor = connect.cursor()
36     cursor.execute(" select * from Student ;")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
(.venv) PS D:\pythonproj\pythonPage> python test.py
Enter Id: 2
Enter Name: Pihu
Enter Age: 7
Data Inserted Successfully !!
[(1, 'sanjeev', 29), (2, 'Pihu', 7)]
(.venv) PS D:\pythonproj\pythonPage>
```

Ln 31, Col 1 Spaces: 4 UTF-8 CRLF Python 3.13.5 (.venv) 11:20 PM 9/3/2025