

Assignment-7

Module 12: Building Database Apps with PostgreSQL & Python

❖ Introduction to data

- Data is a collection of information.
- Datum – Single piece of information.
- Example – Data of a student: like (name, id, age, DOB, Gender, etc.

❖ Introduction to database

➤ Organized

- Tables by means of rows and columns.
- Can be easily Accessed, Managed & Updated

➤ Main purpose

- To operate a large amount of information by storing, retrieving, and managing them.
- Many databases are available like,
 - MySQL
 - Oracle
 - PostgreSQL
 - MongoDB, etc..
- Note: Types of Databases
 - ✚ Hierarchical database
 - ✚ NoSQL database
 - ✚ Object oriented database
 - ✚ Relational database
 - ✚ Network database

❖ Introduction to PostgreSQL

- Relational Database
 - Categorized by a set of tables
 - SQL(Structured Query Language)
 - Acts as the application interface
 - Easier to modify
 - Extending the database
 - Joining the database
- PostgreSQL
 - Object – relational database:
 - Object oriented database + relational database
 - Similar to relational database
 - Object oriented database -> object , classes & inheritance are supported.
 - Open source
 - Source code is available under PostgreSQL license
 - Building of commercial apps

❖ Installing PostgreSQL

- Go to PostgreSQL website and download the application
- Then install the app and don't forget to set of password and port no.
- The port no is: 5432 or 5433 (according to your device)
- And the password to help to login to PostgreSQL (SQL shell (psql))
Then you perform your tasks to psql.

- Open it PostgreSQL and the password are require to you created to installing time.

```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Password for user postgres:

psql (17.5)
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=#
```

Like that .

❖ Creating a database

- Open psql and enter your password.

Note: what are the databases that are available inside the shell.

You just type : \l

```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Password for user postgres:

psql (17.5)
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_India.1252 | English_India.1252 |
 postgres   | postgres | UTF8     | libc            | English_India.1252 | English_India.1252 |
 template0  | postgres | UTF8     | libc            | English_India.1252 | English_India.1252 |
 template1  | postgres | UTF8     | libc            | English_India.1252 | English_India.1252 |
(3 rows)

postgres=#
```

Like that.

- Create a database
 - ❖ The command is: `create database database_name;`

```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Password for user postgres:

psql (17.5)
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_India.1252 | English_India.1252 |  |  | =c/postgres +
 template0 | postgres | UTF8 | libc | English_India.1252 | English_India.1252 |  |  | postgres=CTc/postgres
 template1 | postgres | UTF8 | libc | English_India.1252 | English_India.1252 |  |  | =c/postgres +
                                     postgres=CTc/postgres
(3 rows)

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

Like that.

Note: to switch database one to another.

The command is: `\c database_name`

```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Password for user postgres:

psql (17.5)
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_India.1252 | English_India.1252 |  |  | =c/postgres +
 template0 | postgres | UTF8 | libc | English_India.1252 | English_India.1252 |  |  | postgres=CTc/postgres
 template1 | postgres | UTF8 | libc | English_India.1252 | English_India.1252 |  |  | =c/postgres +
                                     postgres=CTc/postgres
(3 rows)

postgres=# create database student;
CREATE DATABASE
postgres=# \c student
You are now connected to database "student" as user "postgres".
student=#
```

Like that.

❖ Deleting a database

- The command is : drop database database_name;
- For eample:

➤ Create a database (demo)

```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Password for user postgres:

psql (17.5)
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_India.1252 | English_India.1252 |  |  | =c/postgres +
template0 | postgres | UTF8 | libc | English_India.1252 | English_India.1252 |  |  | postgres=CTc/postgres +
template1 | postgres | UTF8 | libc | English_India.1252 | English_India.1252 |  |  | =c/postgres +
                                postgres=CTc/postgres
(3 rows)

postgres=# create database student;
CREATE DATABASE
postgres=# \c student
You are now connected to database "student" as user "postgres".
student=# create database demo;
CREATE DATABASE
student=#
```

➤ Check available databases

```
SQL Shell (psql)
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_India.1252 | English_India.1252 |  |  | =c/postgres +
template0 | postgres | UTF8 | libc | English_India.1252 | English_India.1252 |  |  | postgres=CTc/postgres +
template1 | postgres | UTF8 | libc | English_India.1252 | English_India.1252 |  |  | =c/postgres +
                                postgres=CTc/postgres
(3 rows)

postgres=# create database student;
CREATE DATABASE
postgres=# \c student
You are now connected to database "student" as user "postgres".
student=# create database demo;
CREATE DATABASE
student=# \l

      List of databases
  Name | Owner | Encoding | Locale Provider | Collate | Ctype | Locale | ICU Rules | Access privileges
-----|-----|-----|-----|-----|-----|-----|-----|-----
demo | postgres | UTF8 | libc | English_India.1252 | English_India.1252 |  |  | 
postgres | postgres | UTF8 | libc | English_India.1252 | English_India.1252 |  |  | 
student | postgres | UTF8 | libc | English_India.1252 | English_India.1252 |  |  | 
template0 | postgres | UTF8 | libc | English_India.1252 | English_India.1252 |  |  | =c/postgres +
                                postgres=CTc/postgres +
template1 | postgres | UTF8 | libc | English_India.1252 | English_India.1252 |  |  | =c/postgres +
                                postgres=CTc/postgres
(5 rows)

student=#
```

➤ Delete database

```
SQL Shell (psql)
postgres=# \l
```

| Name | Owner | Encoding | Locale Provider | Collate | Ctype | Locale | ICU Rules | Access privileges |
|-----------|----------|----------|-----------------|--------------------|--------------------|--------|-----------|--|
| postgres | postgres | UTF8 | libc | English_India.1252 | English_India.1252 | | | |
| template0 | postgres | UTF8 | libc | English_India.1252 | English_India.1252 | | | =c/postgres + postgres=CTc/postgres |
| template1 | postgres | UTF8 | libc | English_India.1252 | English_India.1252 | | | =c/postgres + postgres=CTc/postgres |

(3 rows)

```
postgres=# create database student;
CREATE DATABASE
postgres=# \c student
You are now connected to database "student" as user "postgres".
student=# create database demo;
CREATE DATABASE
student=# \l
```

| Name | Owner | Encoding | Locale Provider | Collate | Ctype | Locale | ICU Rules | Access privileges |
|-----------|----------|----------|-----------------|--------------------|--------------------|--------|-----------|--|
| demo | postgres | UTF8 | libc | English_India.1252 | English_India.1252 | | | |
| postgres | postgres | UTF8 | libc | English_India.1252 | English_India.1252 | | | |
| student | postgres | UTF8 | libc | English_India.1252 | English_India.1252 | | | |
| template0 | postgres | UTF8 | libc | English_India.1252 | English_India.1252 | | | =c/postgres + postgres=CTc/postgres |
| template1 | postgres | UTF8 | libc | English_India.1252 | English_India.1252 | | | =c/postgres + postgres=CTc/postgres |

(5 rows)

```
student=# drop database demo;
DROP DATABASE
student=#
```

➤ Check database are deleted or not

```
SQL Shell (psql)
CREATE DATABASE
postgres=# \c student
You are now connected to database "student" as user "postgres".
student=# create database demo;
CREATE DATABASE
student=# \l
```

| Name | Owner | Encoding | Locale Provider | Collate | Ctype | Locale | ICU Rules | Access privileges |
|-----------|----------|----------|-----------------|--------------------|--------------------|--------|-----------|--|
| demo | postgres | UTF8 | libc | English_India.1252 | English_India.1252 | | | |
| postgres | postgres | UTF8 | libc | English_India.1252 | English_India.1252 | | | |
| student | postgres | UTF8 | libc | English_India.1252 | English_India.1252 | | | |
| template0 | postgres | UTF8 | libc | English_India.1252 | English_India.1252 | | | =c/postgres + postgres=CTc/postgres |
| template1 | postgres | UTF8 | libc | English_India.1252 | English_India.1252 | | | =c/postgres + postgres=CTc/postgres |

(5 rows)

```
student=# drop database demo;
DROP DATABASE
student=# \l
```

| Name | Owner | Encoding | Locale Provider | Collate | Ctype | Locale | ICU Rules | Access privileges |
|-----------|----------|----------|-----------------|--------------------|--------------------|--------|-----------|--|
| postgres | postgres | UTF8 | libc | English_India.1252 | English_India.1252 | | | |
| student | postgres | UTF8 | libc | English_India.1252 | English_India.1252 | | | |
| template0 | postgres | UTF8 | libc | English_India.1252 | English_India.1252 | | | =c/postgres + postgres=CTc/postgres |
| template1 | postgres | UTF8 | libc | English_India.1252 | English_India.1252 | | | =c/postgres + postgres=CTc/postgres |

(4 rows)

```
student=#
```

Like that.

Now you see the demo database are deleted.

❖ Creating table and adding data

➤ Create a table

```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Password for user postgres:

psql (17.5)
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=# \c student;
You are now connected to database "student" as user "postgres".
student=# CREATE TABLE students( name text, rollno int, age int, gender text );
CREATE TABLE
student=#
```

Note: and you see all the relations using this command is:

```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Password for user postgres:

psql (17.5)
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=# \c student;
You are now connected to database "student" as user "postgres".
student=# CREATE TABLE students( name text, rollno int, age int, gender text );
CREATE TABLE
student=# \d
List of relations
Schema | Name      | Type  | Owner
-----+-----+-----+-----
public | students | table | postgres
(1 row)

student=#
```

Is that.

➤ Adding data

```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Password for user postgres:

psql (17.5)
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=# \c student;
You are now connected to database "student" as user "postgres".
student=# CREATE TABLE students( name text, rollno int, age int, gender text );
CREATE TABLE
student=# \d
          List of relations
Schema | Name   | Type  | Owner
-----+-----+-----+-----
public | students | table | postgres
(1 row)

student=# INSERT INTO students( name, rollno, age, gender ) VALUES ('Ujjwal',55,21,'male');
INSERT 0 1
student=#
```

And add one more student details

```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Password for user postgres:

psql (17.5)
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=# \c student;
You are now connected to database "student" as user "postgres".
student=# CREATE TABLE students( name text, rollno int, age int, gender text );
CREATE TABLE
student=# \d
          List of relations
Schema | Name   | Type  | Owner
-----+-----+-----+-----
public | students | table | postgres
(1 row)

student=# INSERT INTO students( name, rollno, age, gender ) VALUES ('Ujjwal',55,21,'male');
INSERT 0 1
student=# INSERT INTO students( name, rollno, age, gender ) VALUES ('Udit',54,22,'male');
INSERT 0 1
student=#
```

Like that.

❖ Retrieving data from database and deleting contents in the table.

to retrieve all the data from table.

```
SQL Shell (psql)
Port [5432]:
Username [postgres]:
Password for user postgres:

psql (17.5)
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=# \c student;
You are now connected to database "student" as user "postgres".
student=# CREATE TABLE students( name text, rollno int, age int, gender text );
CREATE TABLE
student=# \d
          List of relations
Schema | Name   | Type  | Owner
-----+-----+-----+-----
public | students | table | postgres
(1 row)

student=# INSERT INTO students( name, rollno, age, gender ) VALUES ('Ujjwal',55,21,'male');
INSERT 0 1
student=# INSERT INTO students( name, rollno, age, gender ) VALUES ('Udit',54,22,'male');
INSERT 0 1
student=# SELECT * FROM students;
 name | rollno | age | gender
-----+-----+-----+-----
Ujjwal |    55  |  21 | male
Udit   |    54  |  22 | male
(2 rows)

student=# |
```

And you also retrieve a specific data form the table like only names, age, rollno,etc.

```
SQL Shell (psql)
Type "help" for help.

postgres=# \c student;
You are now connected to database "student" as user "postgres".
student=# CREATE TABLE students( name text, rollno int, age int, gender text );
CREATE TABLE
student=# \d
          List of relations
Schema | Name   | Type  | Owner
-----+-----+-----+-----
public | students | table | postgres
(1 row)

student=# INSERT INTO students( name, rollno, age, gender ) VALUES ('Ujjwal',55,21,'male');
INSERT 0 1
student=# INSERT INTO students( name, rollno, age, gender ) VALUES ('Udit',54,22,'male');
INSERT 0 1
student=# SELECT * FROM students;
 name | rollno | age | gender
-----+-----+-----+-----
Ujjwal |    55  |  21 | male
Udit   |    54  |  22 | male
(2 rows)

student=# SELECT name FROM students;
 name
-----
Ujjwal
Udit
(2 rows)

student=# |
```

Like that.

And you also perform a compare command

```
SQL Shell (psql)
List of relations
Schema | Name | Type | Owner
-----+-----+-----+-----
public | students | table | postgres
(1 row)

student=# INSERT INTO students( name, rollno, age, gender ) VALUES ('Ujjwal',55,21,'male');
INSERT 0 1
student=# INSERT INTO students( name, rollno, age, gender ) VALUES ('Udit',54,22,'male');
INSERT 0 1
student=# SELECT * FROM students;
 name | rollno | age | gender
-----+-----+-----+-----
Ujjwal |    55 |  21 |   male
Udit   |    54 |  22 |   male
(2 rows)

student=# SELECT name FROM students;
 name
-----
Ujjwal
Udit
(2 rows)

student=# SELECT * FROM students WHERE age=21;
 name | rollno | age | gender
-----+-----+-----+-----
Ujjwal |    55 |  21 |   male
(1 row)

student=#
```

Like that and you perform many command like that.

❖ Delete the contents from inside the table.

Using truncate.

```
SQL Shell (psql)
public | students | table | postgres
(1 row)

student=# INSERT INTO students( name, rollno, age, gender ) VALUES ('Ujjwal',55,21,'male');
INSERT 0 1
student=# INSERT INTO students( name, rollno, age, gender ) VALUES ('Udit',54,22,'male');
INSERT 0 1
student=# SELECT * FROM students;
 name | rollno | age | gender
-----+-----+-----+-----
Ujjwal |    55 |  21 |   male
Udit   |    54 |  22 |   male
(2 rows)

student=# SELECT name FROM students;
 name
-----
Ujjwal
Udit
(2 rows)

student=# SELECT * FROM students WHERE age=21;
 name | rollno | age | gender
-----+-----+-----+-----
Ujjwal |    55 |  21 |   male
(1 row)

student=# TRUNCATE TABLE students;
TRUNCATE TABLE
student=#
```

Now check the content deleted or not.

```
SQL Shell (psql)
INSERT 0 1
student=# INSERT INTO students( name, rollno, age, gender ) VALUES ('Udit',54,22,'male');
INSERT 0 1
student=# SELECT * FROM students;
  name | rollno | age | gender
-----+-----+----+-----
 Ujjwal |    55 |  21 |   male
  Udit  |    54 |  22 |   male
(2 rows)

student=# SELECT name FROM students;
 name
-----
 Ujjwal
  Udit
(2 rows)

student=# SELECT * FROM students WHERE age=21;
  name | rollno | age | gender
-----+-----+----+-----
 Ujjwal |    55 |  21 |   male
(1 row)

student=# TRUNCATE TABLE students;
TRUNCATE TABLE
student=# SELECT * FROM students;
  name | rollno | age | gender
-----+-----+----+-----
(0 rows)

student=#
```

Now you see the table data has been deleted.

❖ Setting up virtualenv

- Virtualenv -> virtual environment
- Using vs code and command prompt
- Create a folder to you perform this task then open it to your vs code.
- And then open your command prompt (cmd) and go to your folder path(address of your folder).
- For example :

```
Command Prompt
Microsoft Windows [Version 10.0.26100.4202]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Ujjwal>E:

E:\>cd E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7
E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>
```

Like that it is my folder path.

- Then install pip virtualenv

```
Command Prompt
Microsoft Windows [Version 10.0.26100.4202]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Ujjwal>E:

E:\>cd E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7

E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>pip install virtualenv
Requirement already satisfied: virtualenv in c:\users\ujjwal\appdata\local\programs\python\python313\lib\site-packages (20.31.2)
Requirement already satisfied: distlib<1,>=0.3.7 in c:\users\ujjwal\appdata\local\programs\python\python313\lib\site-packages (from virtualenv) (0.3.9)
Requirement already satisfied: filelock<4,>=3.12.2 in c:\users\ujjwal\appdata\local\programs\python\python313\lib\site-packages (from virtualenv) (3.18.0)
Requirement already satisfied: platformdirs<5,>=3.9.1 in c:\users\ujjwal\appdata\local\programs\python\python313\lib\site-packages (from virtualenv) (4.3.8)

E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>
```

Like that I'm already installed .

- ➔ So you see your vs code the new folder has been created and the folder name is : env
- ➔ Then again go to you command prompt , and open env folder using cd (change directory) and then go to scripts inside the env.

And for example : Assignment-7>env>Scripts

```
Command Prompt
Microsoft Windows [Version 10.0.26100.4202]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Ujjwal>E:

E:\>cd

E:\>cd E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7

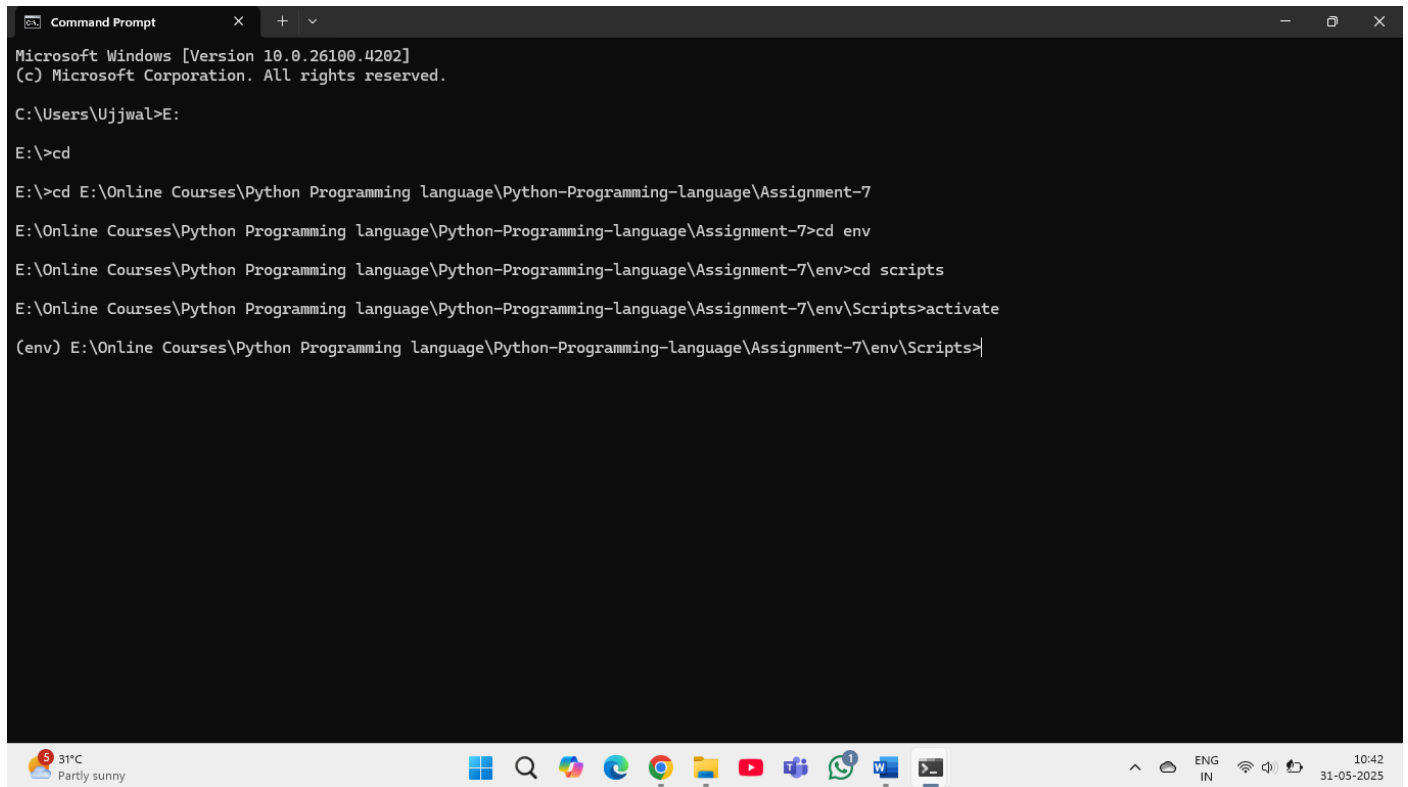
E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>cd env

E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7\env>cd scripts

E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7\env\Scripts>
```

Like that.

➔ The you activate the virtualenv



```
Microsoft Windows [Version 10.0.26100.4202]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Ujjwal>E:

E:\>cd

E:\>cd E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7

E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>cd env

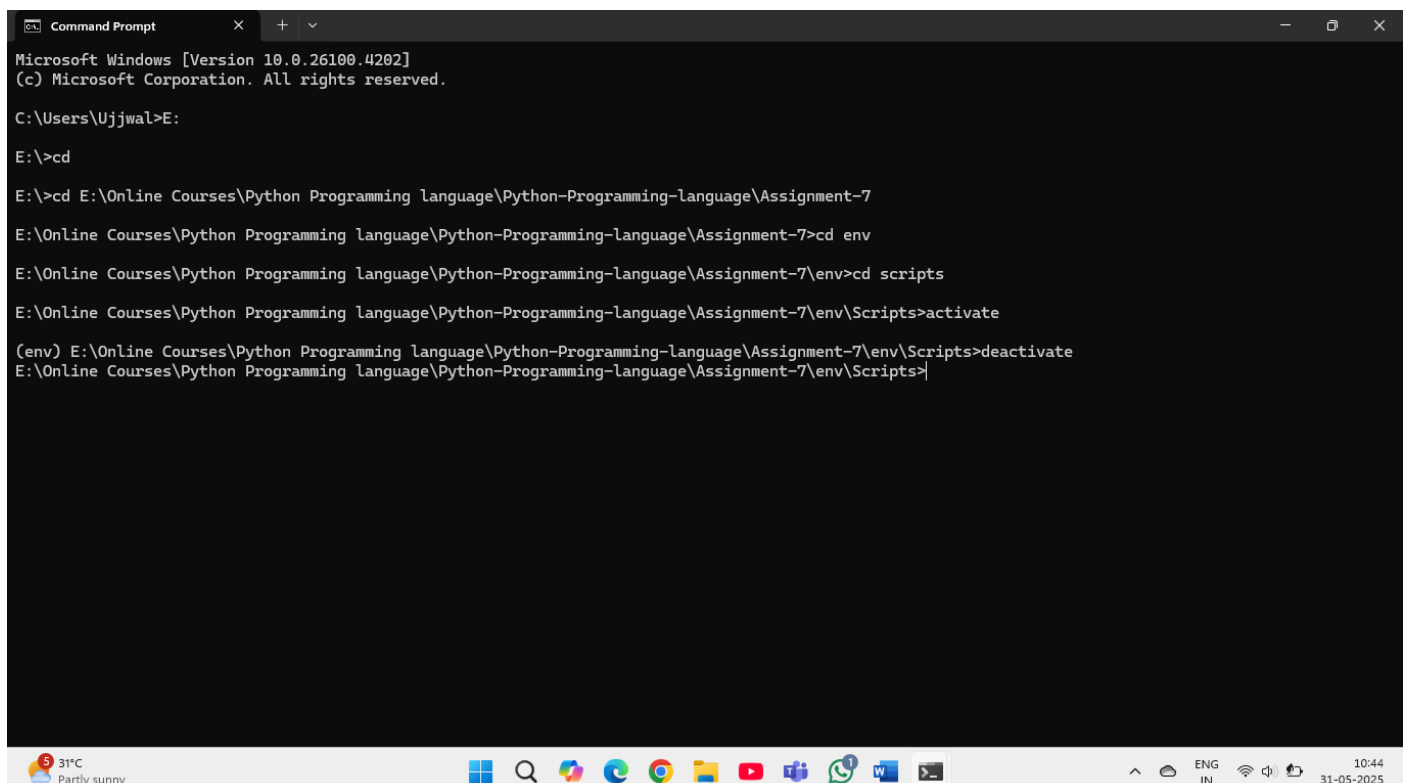
E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7\env>cd scripts

E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7\env\Scripts>activate

(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7\env\Scripts>
```

Now you see the virtualenv has been activated

➔ And the your task has been done then you deactivate the virtualenv



```
Microsoft Windows [Version 10.0.26100.4202]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Ujjwal>E:

E:\>cd

E:\>cd E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7

E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>cd env

E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7\env>cd scripts

E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7\env\Scripts>activate

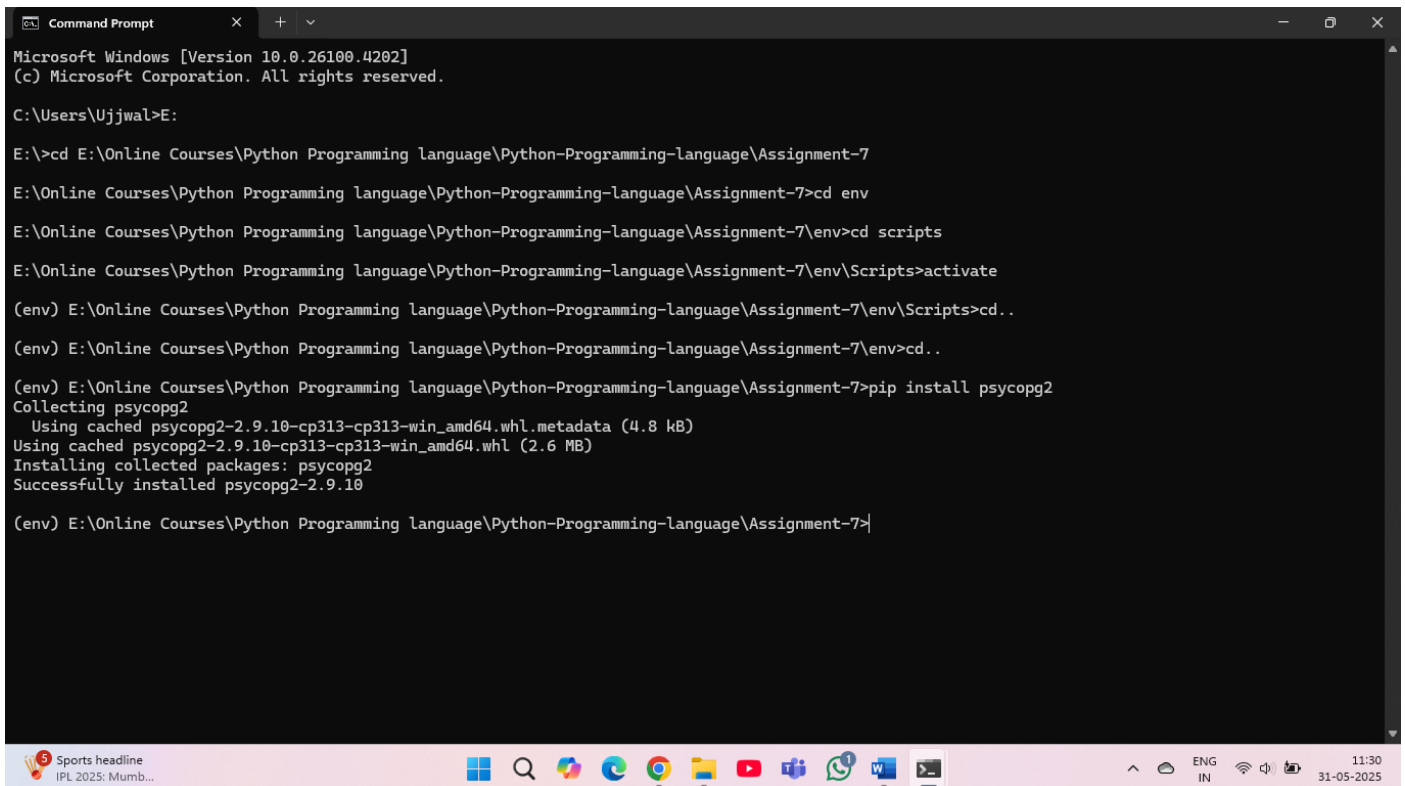
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7\env\Scripts>deactivate

E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7\env\Scripts>
```

Like that.

❖ Installing psycopg2

- ➔ To help the python and the Postgres database to communicate.
- ➔ To install package of psycopg2
- ➔ For example: `pip install psycopg2`



```
Microsoft Windows [Version 10.0.26100.4202]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Ujjwal>E:

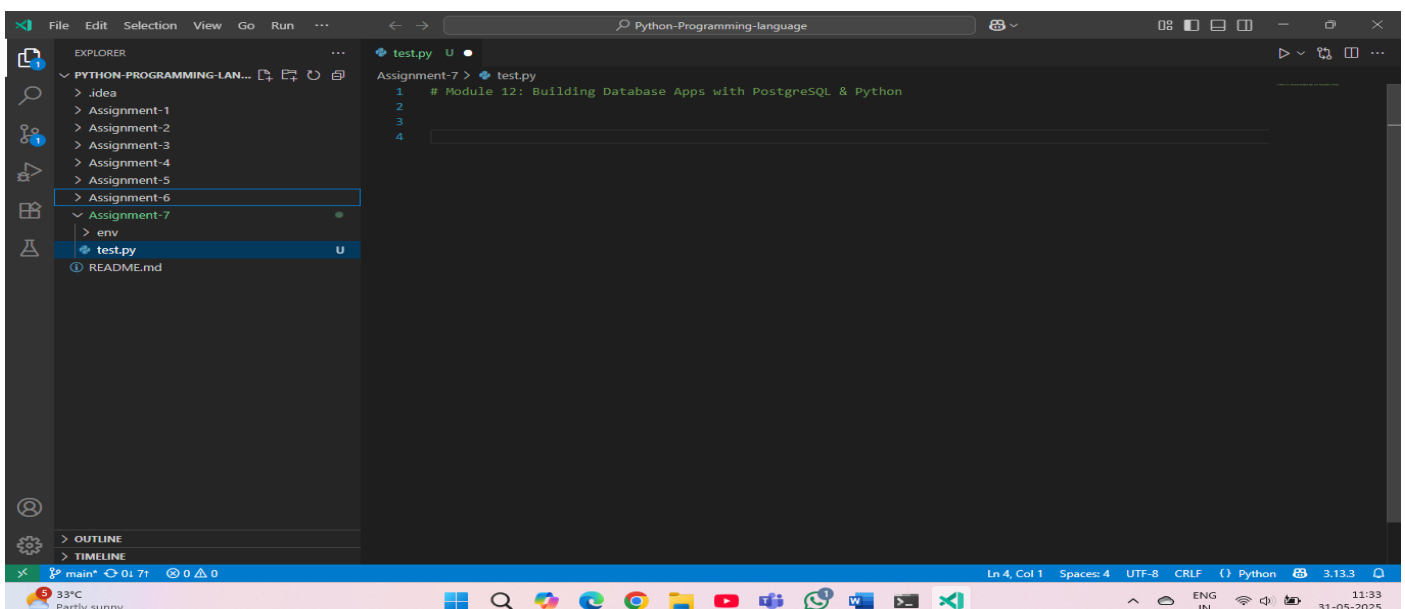
E:\>cd E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7
E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>cd env
E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7\env>cd scripts
E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7\env\Scripts>activate
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7\env\Scripts>cd..
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7\env>cd..
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>pip install psycopg2
Collecting psycopg2
  Using cached psycopg2-2.9.10-cp313-cp313-win_amd64.whl.metadata (4.8 kB)
Using cached psycopg2-2.9.10-cp313-cp313-win_amd64.whl (2.6 MB)
Installing collected packages: psycopg2
Successfully installed psycopg2-2.9.10

(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>
```

Like that.

❖ Connection to the database

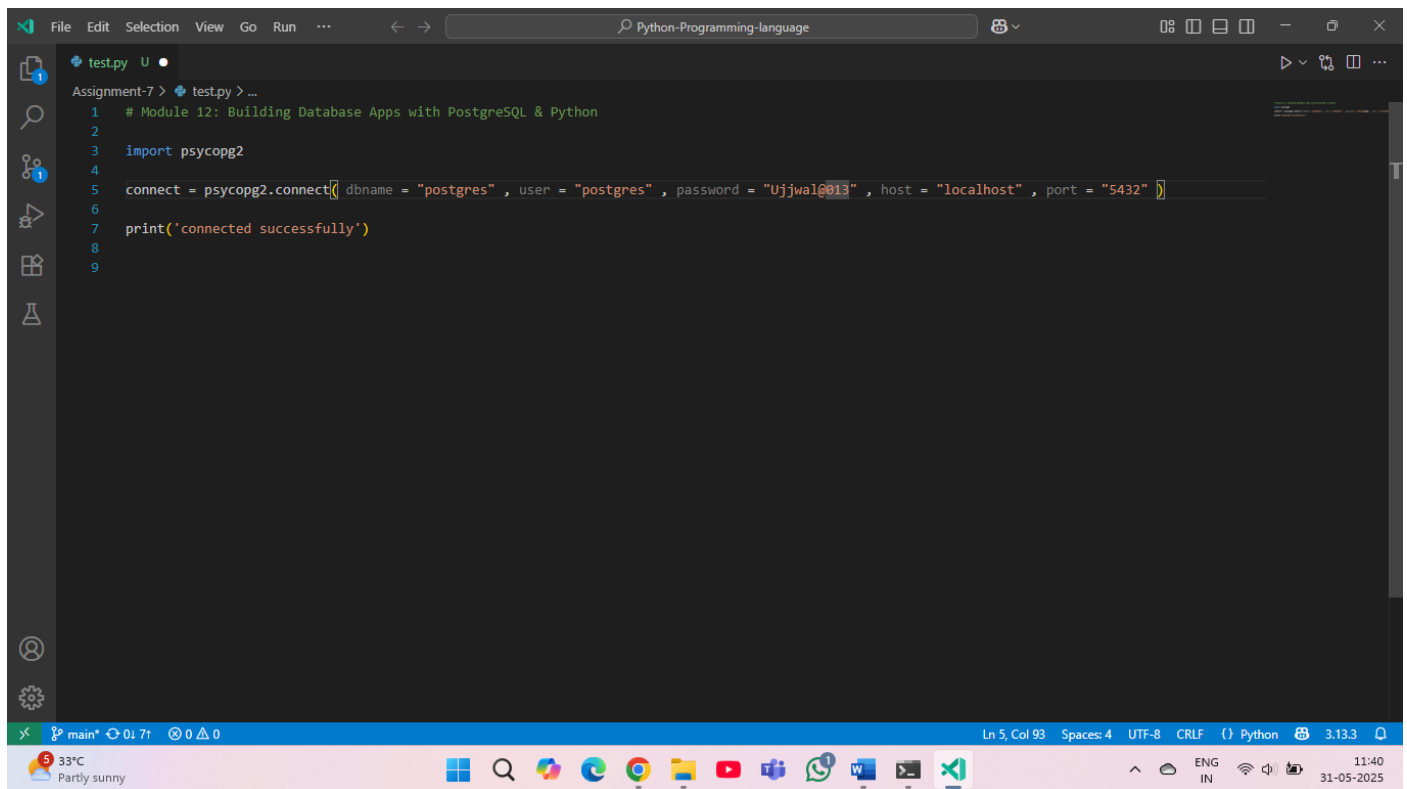
- ➔ Open vs code # file name for example , test.py



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays a file tree for a project named 'PYTHON-PROGRAMMING-LAN...'. The tree includes folders for 'Assignment-1' through 'Assignment-6', and 'Assignment-7'. Under 'Assignment-7', there are subfolders 'env' and 'scripts', and files 'test.py' and 'README.md'. The 'test.py' file is selected and highlighted. The main editor area on the right shows the content of 'test.py', which begins with a comment: '# Module 12: Building Database Apps with PostgreSQL & Python'. The status bar at the bottom indicates the file is 'test.py' at line 4, column 1, with a UTF-8 encoding and CRLF line endings.

Like that.

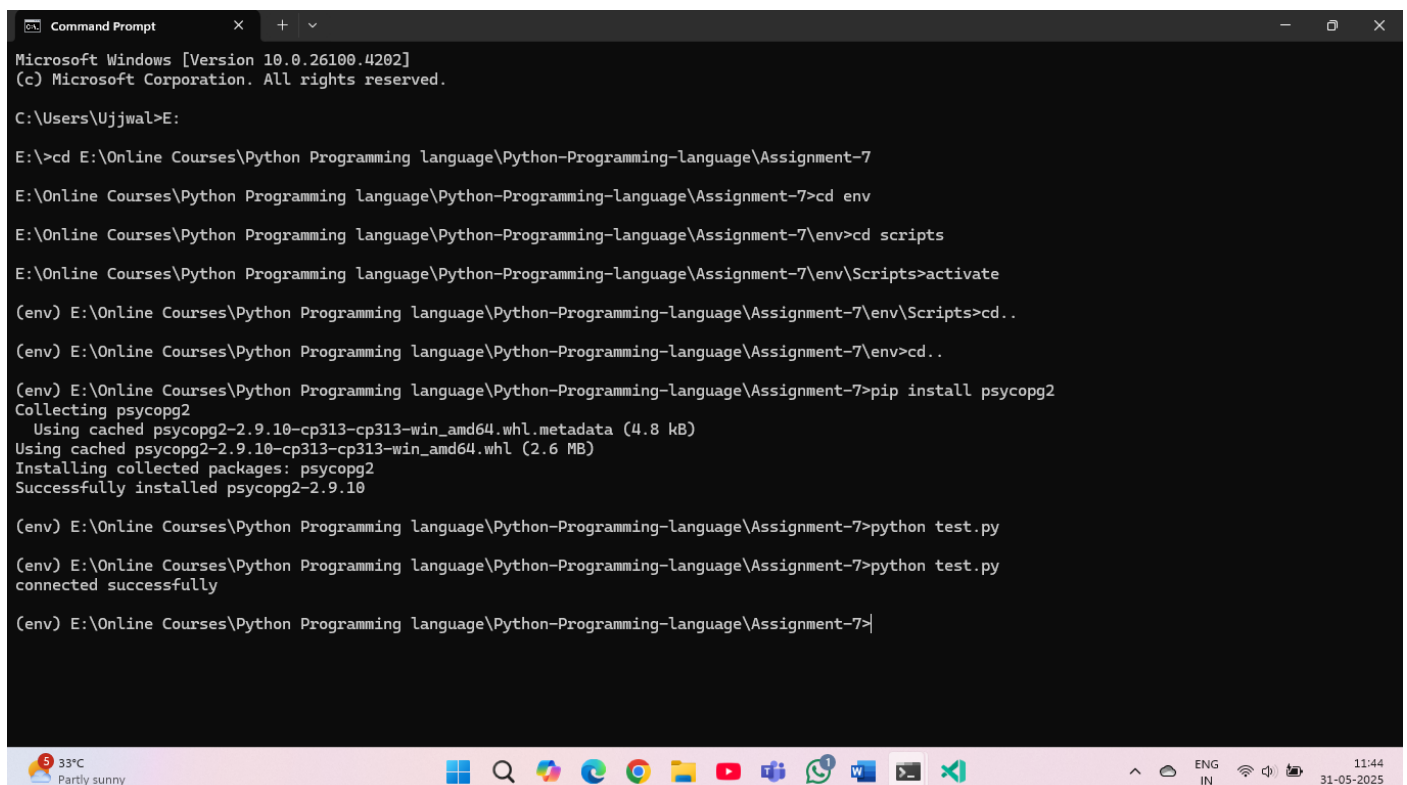
➔ Connect to database



```
test.py U •
Assignment-7 > test.py > ...
1 # Module 12: Building Database Apps with PostgreSQL & Python
2
3 import psycopg2
4
5 connect = psycopg2.connect(dbname = 'postgres', user = 'postgres', password = 'Ujjwal@013', host = 'localhost', port = '5432')
6
7 print('connected successfully')
8
9
```

Like that.

➔ Then open command prompt and virtualenv has been activated then execute the code.



```
Command Prompt
Microsoft Windows [Version 10.0.26100.4202]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Ujjwal>E:

E:\>cd E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7
E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>cd env
E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7\env>cd scripts
E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7\env\Scripts>activate
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7\env\Scripts>cd .
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7\env>cd .
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>pip install psycopg2
Collecting psycopg2
  Using cached psycopg2-2.9.10-cp313-cp313-win_amd64.whl.metadata (4.8 kB)
Using cached psycopg2-2.9.10-cp313-cp313-win_amd64.whl (2.6 MB)
Installing collected packages: psycopg2
Successfully installed psycopg2-2.9.10

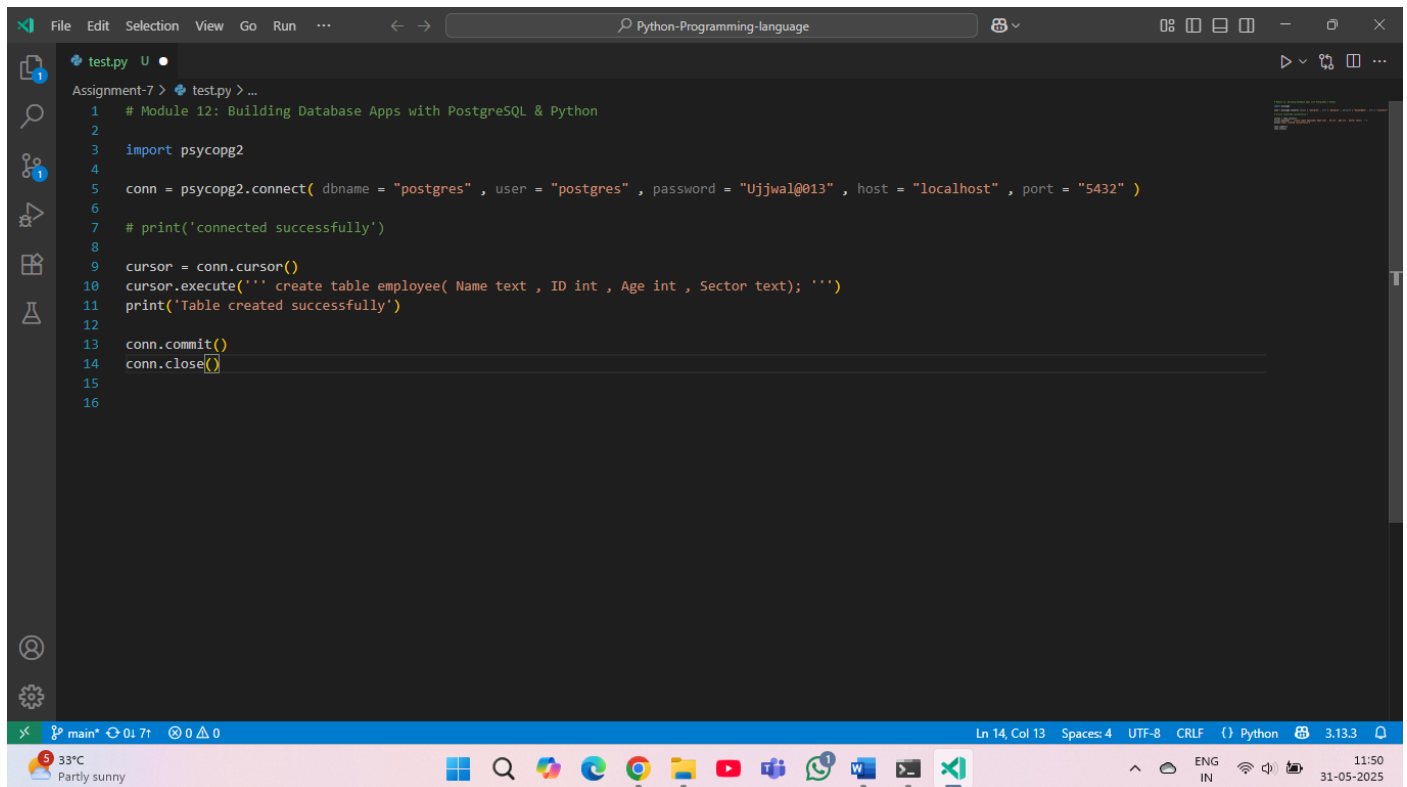
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>python test.py
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>python test.py
connected successfully

(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>
```

Like that.

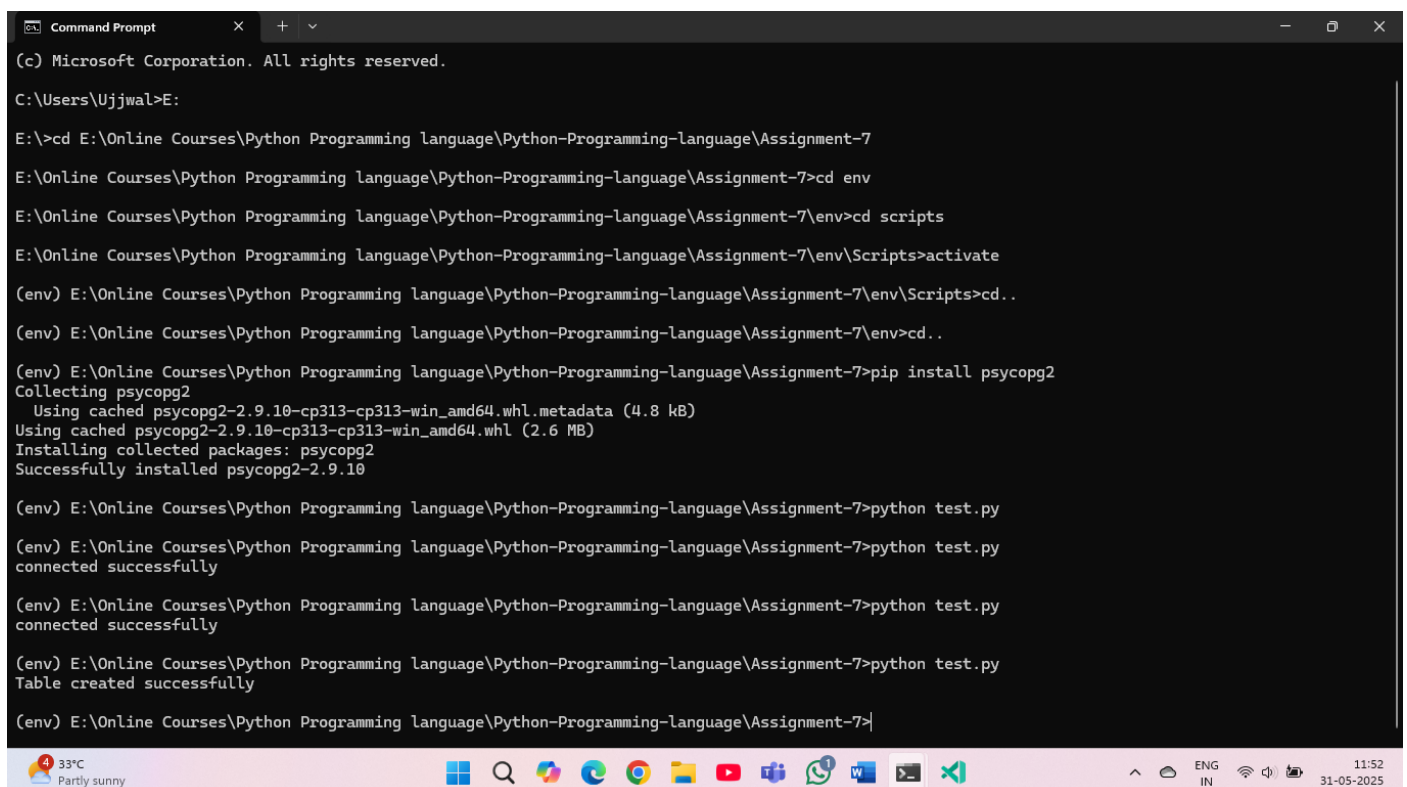
❖ Creating table using python

Eg ->



```
test.py U •
Assignment-7 > test.py > ...
1 # Module 12: Building Database Apps with PostgreSQL & Python
2
3 import psycopg2
4
5 conn = psycopg2.connect( dbname = "postgres" , user = "postgres" , password = "Ujjwal@013" , host = "localhost" , port = "5432" )
6
7 # print('connected successfully')
8
9 cursor = conn.cursor()
10 cursor.execute(''' create table employee( Name text , ID int , Age int , Sector text); ''')
11 print('Table created successfully')
12
13 conn.commit()
14 conn.close()
15
16
```

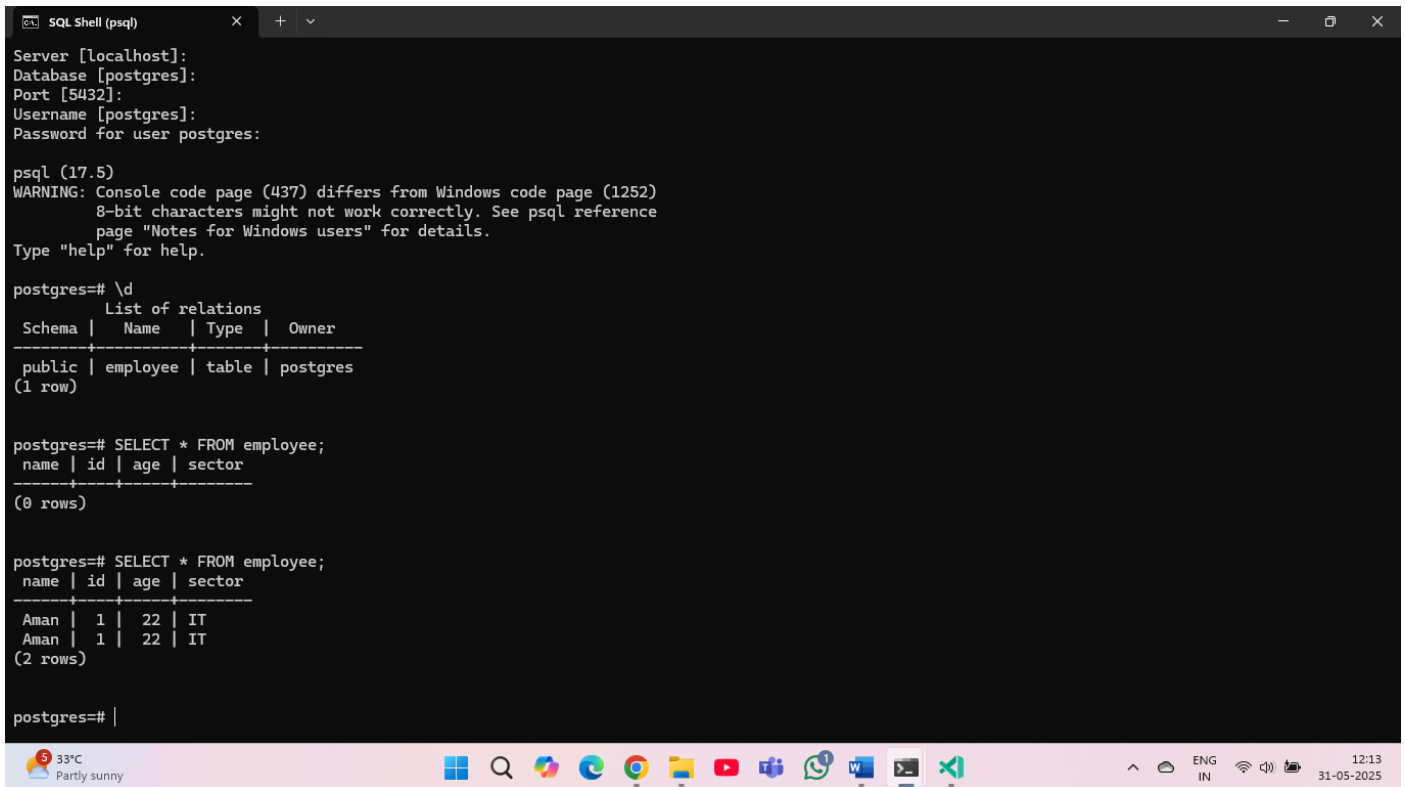
Like that then the output is :



```
Command Prompt
(c) Microsoft Corporation. All rights reserved.
C:\Users\Ujjwal>E:
E:\>cd E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7
E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>cd env
E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7\env>cd scripts
E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7\env\Scripts>activate
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7\env\Scripts>cd..
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7\env>cd..
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>pip install psycopg2
Collecting psycopg2
  Using cached psycopg2-2.9.10-cp313-cp313-win_amd64.whl.metadata (4.8 kB)
Using cached psycopg2-2.9.10-cp313-cp313-win_amd64.whl (2.6 MB)
Installing collected packages: psycopg2
Successfully installed psycopg2-2.9.10
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>python test.py
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>python test.py
connected successfully
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>python test.py
connected successfully
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>python test.py
Table created successfully
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>
```

Like that. Then check to psql shell

➔ then check to psql shell



```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Password for user postgres:

psql (17.5)
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=# \d
          List of relations
Schema | Name   | Type | Owner
-----+-----+-----+-----
public | employee | table | postgres
(1 row)

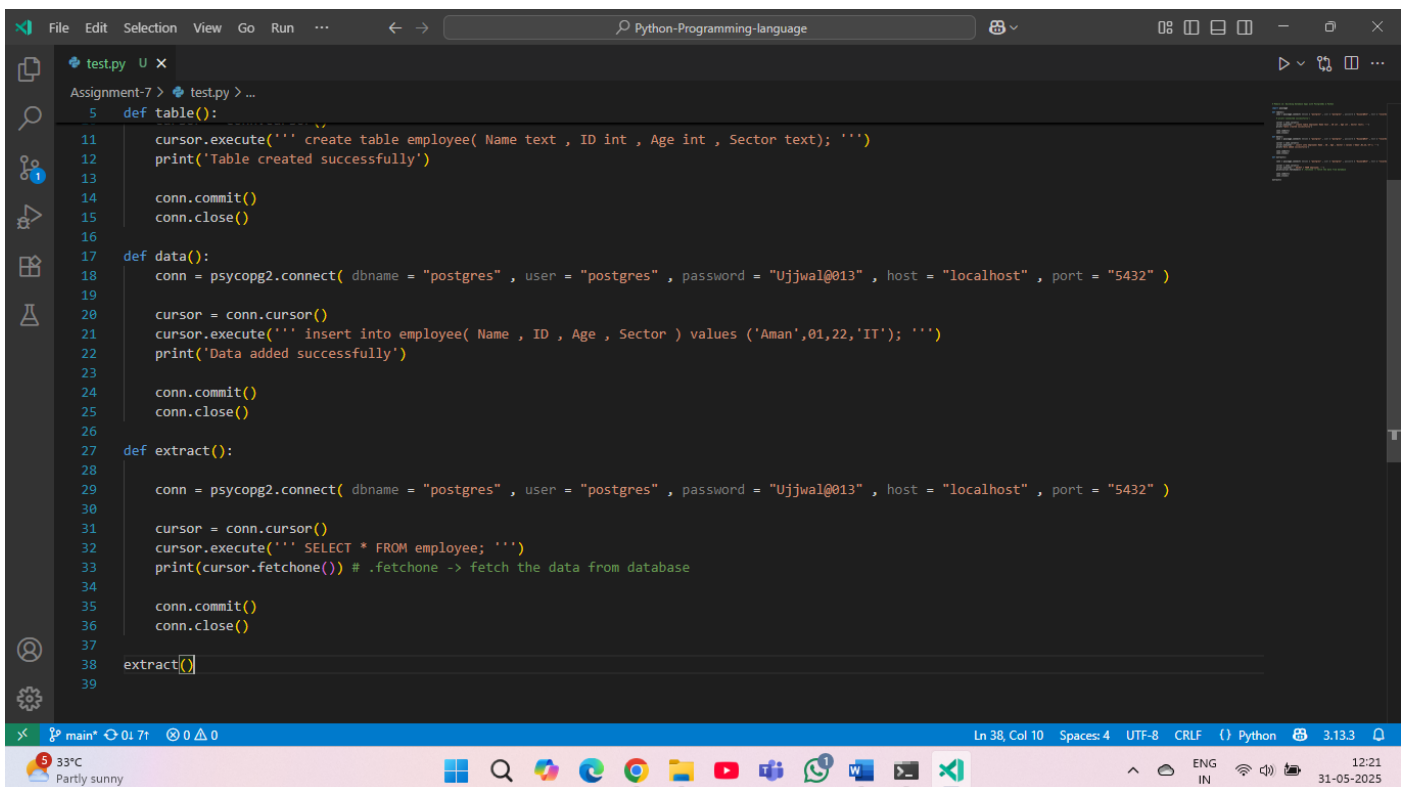
postgres=# SELECT * FROM employee;
 name | id | age | sector
-----+---+----+-----
(0 rows)

postgres=# SELECT * FROM employee;
 name | id | age | sector
-----+---+----+-----
 Aman |  1 |  22 |    IT
 Aman |  1 |  22 |    IT
(2 rows)

postgres=#
```

Like that.

❖ Extracting the data from the database



```
test.py
Assignment-7 > test.py > ...
5 def table():
11     cursor.execute('create table employee( Name text , ID int , Age int , Sector text); ')
12     print('Table created successfully')
13
14     conn.commit()
15     conn.close()
16
17 def data():
18     conn = psycopg2.connect( dbname = "postgres" , user = "postgres" , password = "Ujjwal@013" , host = "localhost" , port = "5432" )
19
20     cursor = conn.cursor()
21     cursor.execute('insert into employee( Name , ID , Age , Sector ) values ('Aman',01,22,'IT'); ')
22     print('Data added successfully')
23
24     conn.commit()
25     conn.close()
26
27 def extract():
28
29     conn = psycopg2.connect( dbname = "postgres" , user = "postgres" , password = "Ujjwal@013" , host = "localhost" , port = "5432" )
30
31     cursor = conn.cursor()
32     cursor.execute('SELECT * FROM employee; ')
33     print(cursor.fetchone()) # .fetchone -> fetch the data from database
34
35     conn.commit()
36     conn.close()
37
38     extract()
39
```

Like that

And the output is:


```
Command Prompt
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>
(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>python test.py
Data added successfully

(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>python test.py
('Aman', 1, 22, 'IT')

(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>python test.py
Aman

(env) E:\Online Courses\Python Programming language\Python-Programming-language\Assignment-7>
```

Like that.

❖ Adding the input from the user

```
File Edit Selection View Go Run ... Python-Programming-language
test.py U
Assignment-7 > test.py > ...
22 # print('Data added successfully')
23
24 # conn.commit()
25 # conn.close()
26 def data():
27     conn = psycopg2.connect( dbname = "postgres" , user = "postgres" , password = "Ujjwal@013" , host = "localhost" , port = "5432" )
28
29     cursor = conn.cursor()
30
31     name = input('enter name: ')
32     id = input('enter id: ')
33     age = input('enter age: ')
34     sector = input('enter sector: ')
35
36     query = ' insert into employee( Name , ID , Age , Sector ) values ( %s , %s , %s , %s); '
37     cursor.execute( query, (name , id , age , sector))
38     print('Data added successfully')
39
40     conn.commit()
41     conn.close()
42
43 > def extract(): ...
55
56 data()
57
```

Like that.

And the output is:

Now see the data from psql shell

```
SQL Shell (psql)

postgres=# SELECT * FROM employee;
 name | id | age | sector
-----+---+----+-----
 Aman |  1 |  22 | IT
 Aman |  1 |  22 | IT
 Ujjwal | 2 |  20 | it
(3 rows)

postgres=# truncate table employee;
TRUNCATE TABLE
postgres=# SELECT * FROM employee;
 name | id | age | sector
-----+---+----+-----
(0 rows)

postgres=# SELECT * FROM employee;
 name | id | age | sector
-----+---+----+-----
(0 rows)

postgres=# SELECT * FROM employee;
 name | id | age | sector
-----+---+----+-----
 Ujjwal | 1 |  21 | IT
 Udit   | 2 |  21 | HR
 Annu   | 3 |  20 | IT
 Anshika | 4 |  19 | IT
(4 rows)

postgres=#
```

Now you see all the employee information.

Thank you

Author -> Ujjwal