

In order for developers to be able to contribute to the Peergrading Project, they must have a local copy of the existing code up and running in localhost. Only then can they understand how the different parts of the code function together to produce the functionality of the product.

While I tried to get the code locally running, I in particular faced two hardships:

1. Requirements.txt installation
2. PostgreSQL installation

1. Requirements.txt installation

This is done by running the command

```
pip install -r requirements.txt
```

The command is run in the same directory where the GitHub Repository of Peergrading Project has been cloned.

During the run, there is something called “building the wheels” as can be seen in the following screenshot:

So, in order for the system to be able to build wheels for the various Python Libraries, make sure that you have Microsoft Visual Studio 2022 installed. Keep in mind, this is NOT Microsoft Visual Studio Code - the favorite IDE for editing and running code.

To show the difference as for as symbol is concerned, this is how Microsoft Visual Studio Code's Symbol looks like:

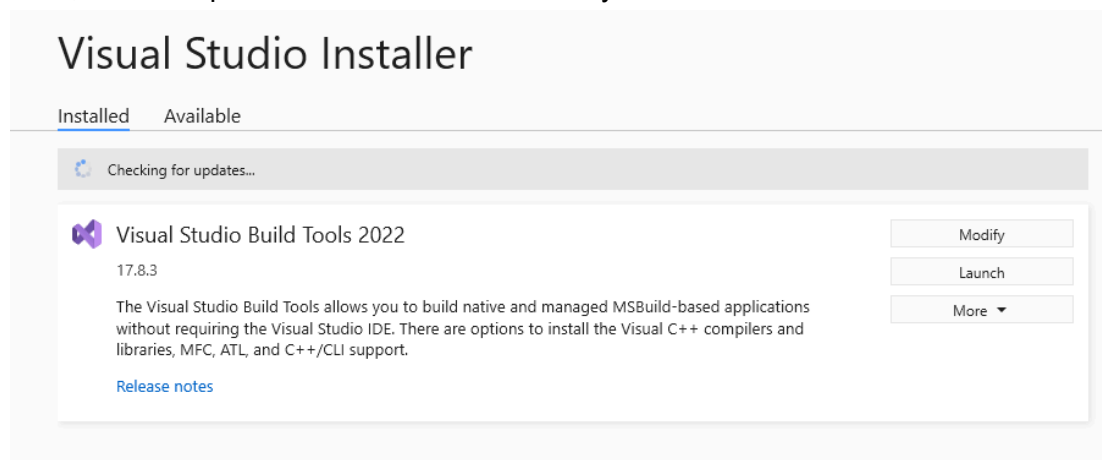


And this is how Microsoft Visual Studio 2022's Symbol looks like:



And this is how the Installer Interface of Microsoft Visual Studio 2022 looks like:

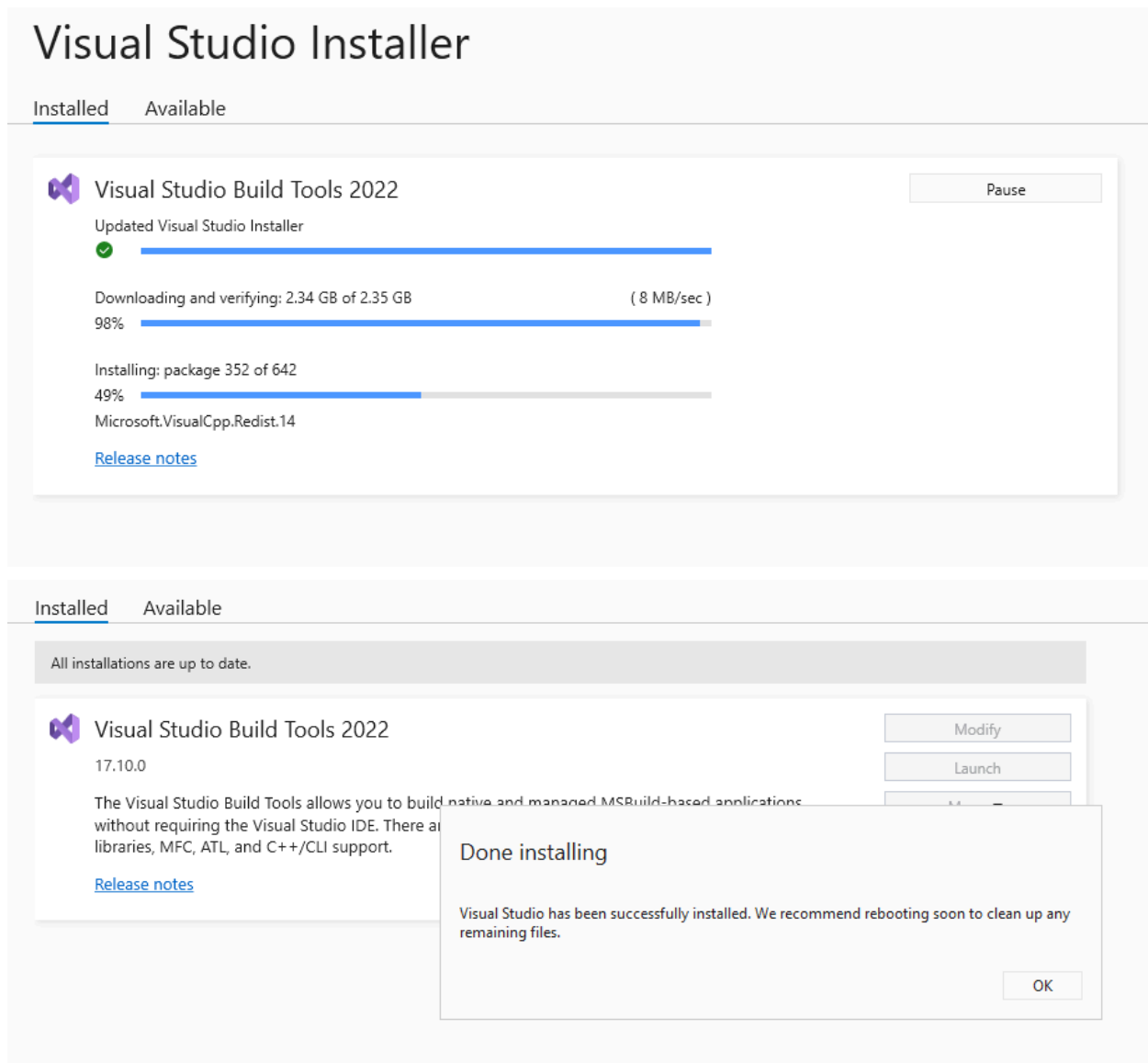
The “Installed” tab shows the installed applications. Here, we have Visual Studio Build Tools 2022, which helps to build the wheels of the Python Libraries.



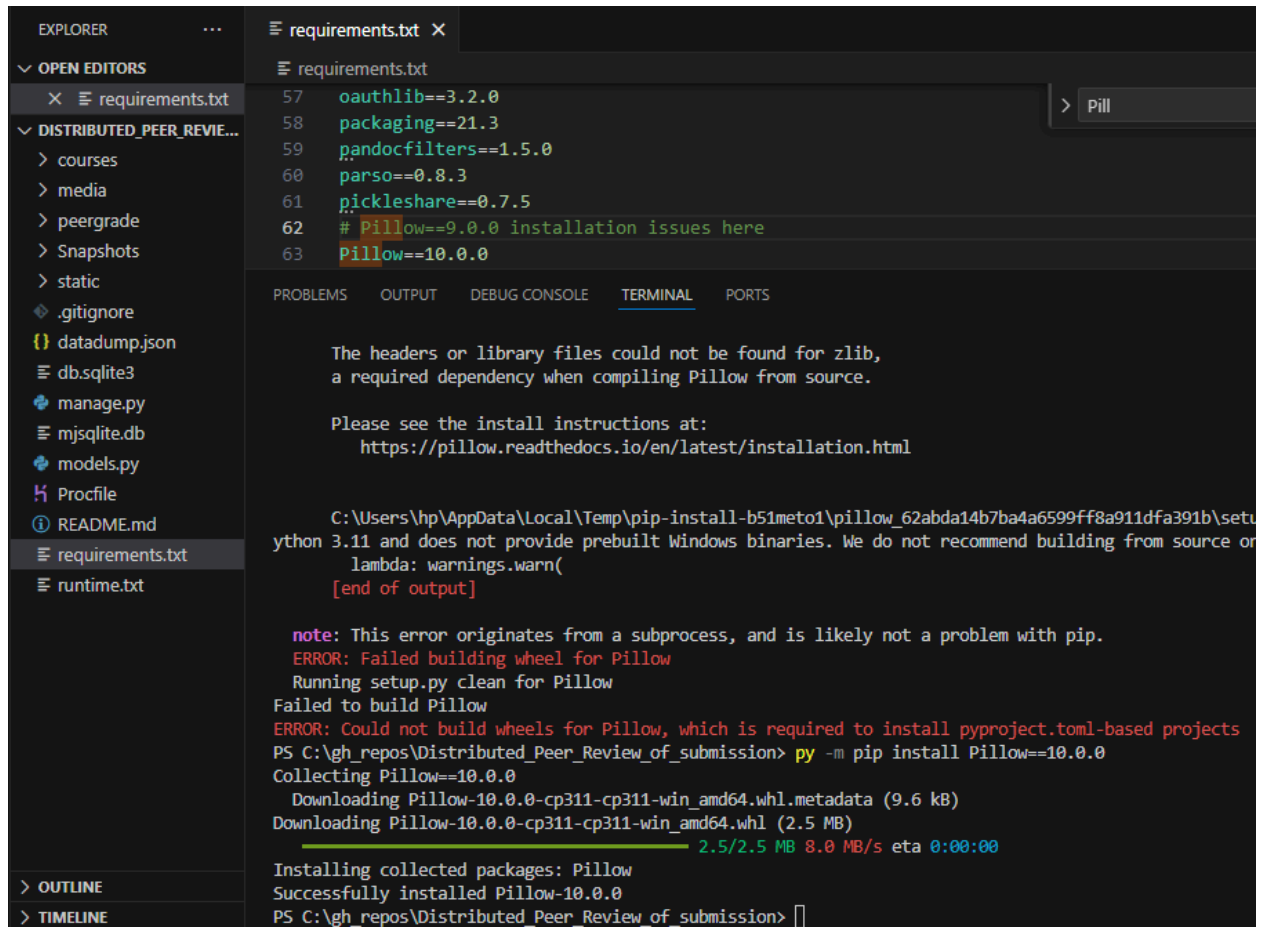
Despite having it installed, still the build was not successful:

```
error: command 'C:\\Program Files (x86)\\Microsoft Visual Studio\\2022\\BuildTools\\VC\\Tools\\MSVC\\14.38.33130\\bin\\x86_x-  
xe' failed with exit code 2  
[end of output]  
  
note: This error originates from a subprocess, and is likely not a problem with pip.  
ERROR: Failed building wheel for pyzmq  
Building wheel for tornado (setup.py) ... done  
Created wheel for tornado: filename=tornado-6.1-cp311-cp311-win_amd64.whl size=419557 sha256=d062196af31c8c27c4c7225dce  
Stored in directory: c:\users\hp\AppData\Local\Pip\Cache\wheels\f2\59\06\xa9c85c7b17ec0fc9b1e2ae0c59e3d39255d5c0a38492e3  
Successfully built cffi django-allauth kiwisolver MarkupSafe numpy pycpg2 pyrsistent tornado  
Failed to build matplotlib Pillow pyzmq  
ERROR: Could not build wheels for matplotlib, Pillow, pyzmq, which is required to install pyproject.toml-based projects
```

So what I tried is I went to the Installer, and checked the VS Build Tools 2022. I found that an update was required. So I clicked on "Update" and it went on as shown:



Turns out, there is no issue with MS VS 2022. The issue is with Pillow version.

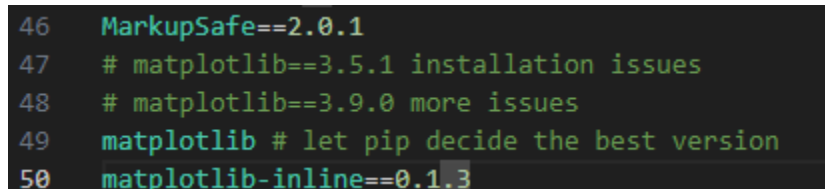


The screenshot shows a Visual Studio Code editor with a file explorer on the left and a terminal on the right. The file explorer shows a project structure with files like `requirements.txt`, `runtime.txt`, and various data files. The `requirements.txt` file is open in the editor, showing the following content:

```
57 oauthlib==3.2.0
58 packaging==21.3
59 pandocfilters==1.5.0
60 parso==0.8.3
61 pickleshare==0.7.5
62 # Pillow==9.0.0 installation issues here
63 Pillow==10.0.0
```

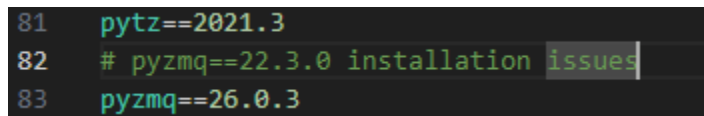
The terminal on the right shows the output of a `pip install` command. It displays an error message: "The headers or library files could not be found for zlib, a required dependency when compiling Pillow from source." It then provides a link to the installation instructions: <https://pillow.readthedocs.io/en/latest/installation.html>. The terminal also shows the command `python 3.11 and does not provide prebuilt Windows binaries. We do not recommend building from source or lambda: warnings.warn([end of output])` and the note: "This error originates from a subprocess, and is likely not a problem with pip." The terminal output continues with the command `py -m pip install Pillow==10.0.0` and shows the progress of downloading and installing the Pillow package.

Similarly the versions of Matplotlib:



```
46 MarkupSafe==2.0.1
47 # matplotlib==3.5.1 installation issues
48 # matplotlib==3.9.0 more issues
49 matplotlib # let pip decide the best version
50 matplotlib-inline==0.1.3
```

and pyzmq:



```
81 pytz==2021.3
82 # pyzmq==22.3.0 installation issues
83 pyzmq==26.0.3
```

are changed. The issues were discovered based on the error messages in the terminal. In the future, if you happen to face installation issues, you should do this:

1. Check the terminal message carefully. If required, copy-paste the terminal message to some chatbot like ChatGPT so that the work of identifying those critical lines which speak the most about the issue at hand is done for you. But be warned, do not copy-paste the whole error message lest the chatbot may not respond properly! Also, even before execution is over, error messages may appear in between. Be alert, and as soon as those intermediate error messages appear on the terminal, copy them and for the time being paste them to a notepad file. You can generate prompts to chatbot like

that. This is because the output message on the terminal is so large that the top lines including the error message get cropped and only the bottom (few thousand lines) are visible. So part of the context is missed.

2. Once the critical error lines are identified, tweak the versions of the libraries like I have shown above. Usually, we use the latest version. But sometimes that also does not work. Then, we let pip decide the best version rather than explicitly specifying.
3. Most importantly, do all this in a virtual environment so that there are no dependency conflicts elsewhere. This is a must.

Finally, you should see something like this:

```
Installing collected packages: webencodings, wcwidth, Send2Trash, pywin32, pytz, pyasn1, pickleshare, mistune, json5, ipython-genutils, django-js-asset, dj-database-url, distlib, certifi, backcall, whitenoise, websocket-client, urllib3, tzdata, traitlets, tornado, toml, testpath, sqlparse, sniffio, six, rsa, pyzmq, pywinpty, pyparsing, PyJWT, Pygments, pycparser, pycodestyle, pyasn1-modules, pycpg2, prompt-toolkit, prometheus-client, platformdirs, Pillow, parso, pandocfilters, oauthlib, numpy, nest-asyncio, MarkupSafe, kiwisolver, idna, gunicorn, fonttools, filelock, entrypoints, django-crispy-forms, defusedxml, decorator, debugpy, cyclers, colorama, charset-normalizer, Babel, attrs, asgiref, WTForms, virtualenv, terminado, requests, python3-openid, python-dateutil, packaging, matplotlib-inline, jupyterlab-pygments, jupyter-core, jsonschema, Jinja2, jedi, Django, contourpy, cffi, captcha, autopep8, anyio, requests-oauthlib, nbformat, matplotlib, jupyter-client, ipython, django-heroku, cryptography, bleach, argon2-cffi-bindings, nbclient, ipykernel, argon2-cffi, nbconvert, djan
go-allauth, notebook, jupyter-server, nbclassic, jupyterlab-server, jupyterlab
Successfully installed Babel-2.9.1 Django-4.0.2 Jinja2-3.0.3 MarkupSafe-2.0.1 Pillow-10.0.0 PyJWT-2.3.0 Pygments-2.11.2 Send2Trash-1.8.0 WTForms-2.3.3 anyio-3.4.0 argon2-cffi-21.3.0 argon2-cffi-bindings-21.2.0 asgiref-3.5.0 attrs-21.4.0 autopep8-1.6.0 backcall-0.2.0 bleach-4.1.0 captcha-0.3 certifi-2021.10.8 cffi-1.15.0 charset-normalizer-2.0.10 colorama-0.4.4 contourpy-1.2.1 cryptography-36.0.1 cyclers-0.11.0 debugpy-1.5.1 decorator-5.1.1 defusedxml-0.7.1 distlib-0.3.4 dj-database-url-0.5.0 django-allauth-0.49.0 django-crispy-forms-1.14.0 django-heroku-0.3.1 django-js-asset-1.2.2 entrypoints-0.3 filelock-3.5.1 fonttools-4.28.5 gunicorn-20.1.0 idna-3.3 ipykernel-6.6.1 ipython-7.31.0 ipython-genutils-0.2.0 jedi-0.18.1 json5-0.9.6 jsonschema-4.3.3 jupyter-client-7.1.0 jupyter-core-4.9.1 jupyter-server-1.13.1 jupyterlab-3.2.6 jupyterlab-pygments-0.1.2 jupyterlab-server-2.10.3 kiwisolver-1.3.2 matplotlib-3.8.4 matplotlib-inline-0.1.3 mistune-0.8.4 nbclassic-0.3.4 nbclient-0.5.9 nbconvert-6.4.0 nbformat-5.1.3 nest-asyncio-1.5.4 notebook-6.4.6 numpy-1.22.0 oauthlib-3.2.0 packaging-21.3 pandocfilters-1.5.0 parso-0.8.3 pickleshare-0.7.5 platformdirs-2.5.0 prometheus-client-0.12.0 prompt-toolkit-3.0.24 pycpg2-2.9.3 pyasn1-0.4.8 pyasn1-modules-0.2.8 pycodestyle-2.8.0 pycparser-2.21 pyparsing-3.0.6 pyparsing-0.18.0 python-dateutil-2.8.2 python3-openid-3.2.0 pytz-2021.3 pywin32-306 pywinpty-2.0.13 pyzmq-26.0.3 requests-2.27.1 requests-oauthlib-1.3.1 rsa-4.8 six-1.16.0 sniffio-1.2.0 sqlparse-0.4.2 terminado-0.12.1 testpath-0.5.0 toml-0.10.2 tornado-6.1 traitlets-5.1.1 tzdata-2021.5 urllib3-1.26.8 virtualenv-20.13.1 wcwidth-0.2.5 webencodings-0.5.1 websocket-client-1.2.3 whitenoise-6.0.0

[notice] A new release of pip is available: 23.2.1 -> 24.0
[notice] To update, run: python.exe -m pip install --upgrade pip
(env) PS C:\gh_repos\Distributed_Peer_Review_of_submission> |
```

2. PostgreSQL Installation

I was facing difficulty here primarily because my previous attempts, over a year before attempting again this time, were messy. Even though I had run the uninstaller, which is supposed to have removed all files, still, it should be noted that the uninstaller DOES NOT remove all files, especially from folders which are hidden from view.


Let's go from Step 1 of the installation:

Step 1: Go to <https://www.postgresql.org/download/>

Step 2: This guide is based on Windows. So, click on Windows. Although I have not tried it, the procedure should be similar for other OSs like Mac, Linux, BSD, etc.

Step 3: Now, click on "Download the installer". You will be redirected to <https://www.enterprisedb.com/downloads/postgres-postgresql-downloads>


Step 4: Download the latest version in accordance with your computer's OS (Linux, Mac or Windows). Check first if your computer is 64-bit architecture (most modern computers are), or 32-bit architecture (most old computers are).


[Products](#)
[Solutions](#)
[Services & Support](#)
[Developers](#)
[Resources](#)
[Company](#)
[Sign In](#)
[Talk to an Expert](#)

Download PostgreSQL

Open source PostgreSQL packages and installers from EDB

PostgreSQL Version	Linux x86-64	Linux x86-32	Mac OS X	Windows x86-64	Windows x86-32
16.3	postgresql.org	postgresql.org			Not supported
15.7	postgresql.org	postgresql.org			Not supported
14.12	postgresql.org	postgresql.org			Not supported
13.15	postgresql.org	postgresql.org			Not supported


[Products](#)
[Solutions](#)
[Services & Support](#)
[Developers](#)
[Resources](#)
[Company](#)
[Sign In](#)
[Talk to an Expert](#)

Your download should begin in a few seconds

[Click here if your download does not start automatically.](#)

[Installation Instructions](#)

```

-> psql postgres://demo.pasnerdbp-cf4ake.dgga.edbcloud.io:5432/chinook?sslmode=require
psql (12.8 (Ubuntu 12.8-1) pgsql128-1), server 12.8.4 (Ubuntu 12.8.4-1deb10)
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
Type "help" for help.

chinook=# \dt
          List of relations
   Schema |      Name      | Type | Owner
-----|-----|-----|-----
 public | album          | table | chinook_owner
 public | artist         | table | chinook_owner
 public | customer       | table | chinook_owner
 public | employee       | table | chinook_owner
 public | genre          | table | chinook_owner
 public | invoice        | table | chinook_owner
 public | invoiceitem    | table | chinook_owner
 public | mediastore     | table | chinook_owner
 public | playlist       | table | chinook_owner
 public | playlisttrack  | table | chinook_owner
 public | track         | table | chinook_owner
 sys    | callback_queue_table | table | postgres
 sys    | dual           | table | postgres

chinook=# \d employee

```

EDB Postgres® AI

Get Postgres from a Builder, Not a Borrower

EDB Postgres AI Fully Managed Database-as-a-Service

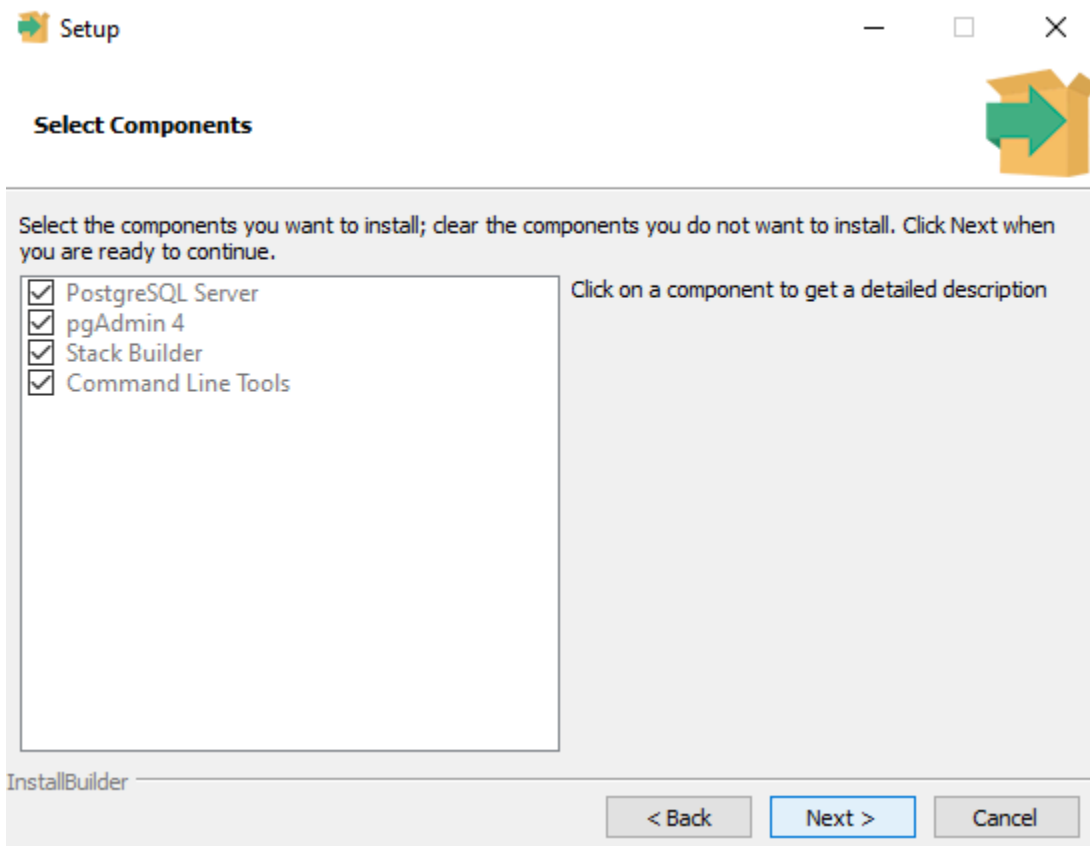
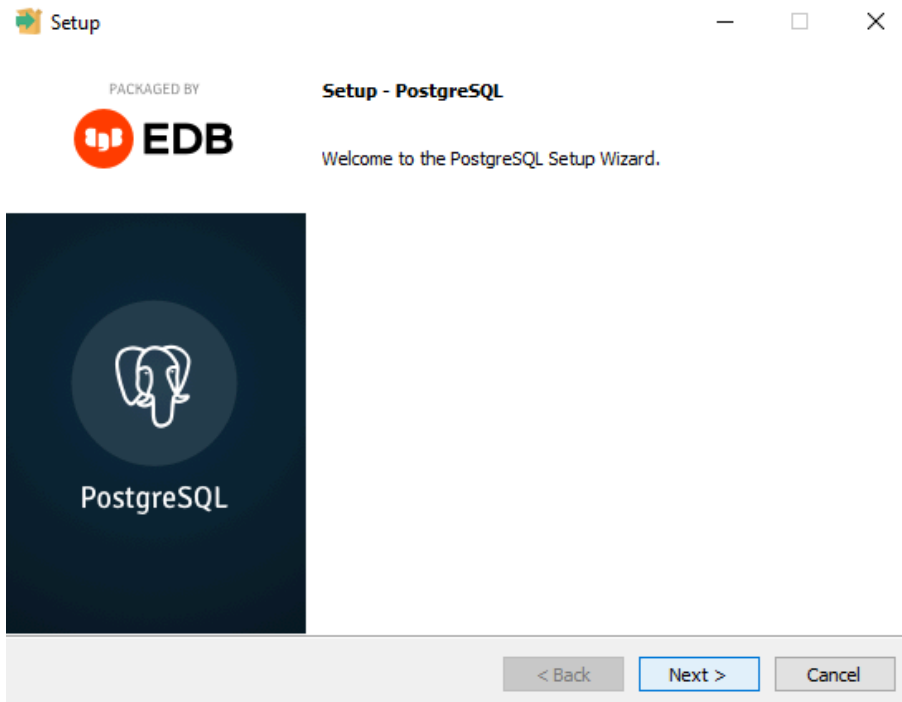
- Distributed Postgres
- Unmatched Oracle Compatibility
- No Vendor Lock-In

[Start Free](#)
[Learn More](#)

Activate Windows

Go to Settings to activate Windows.

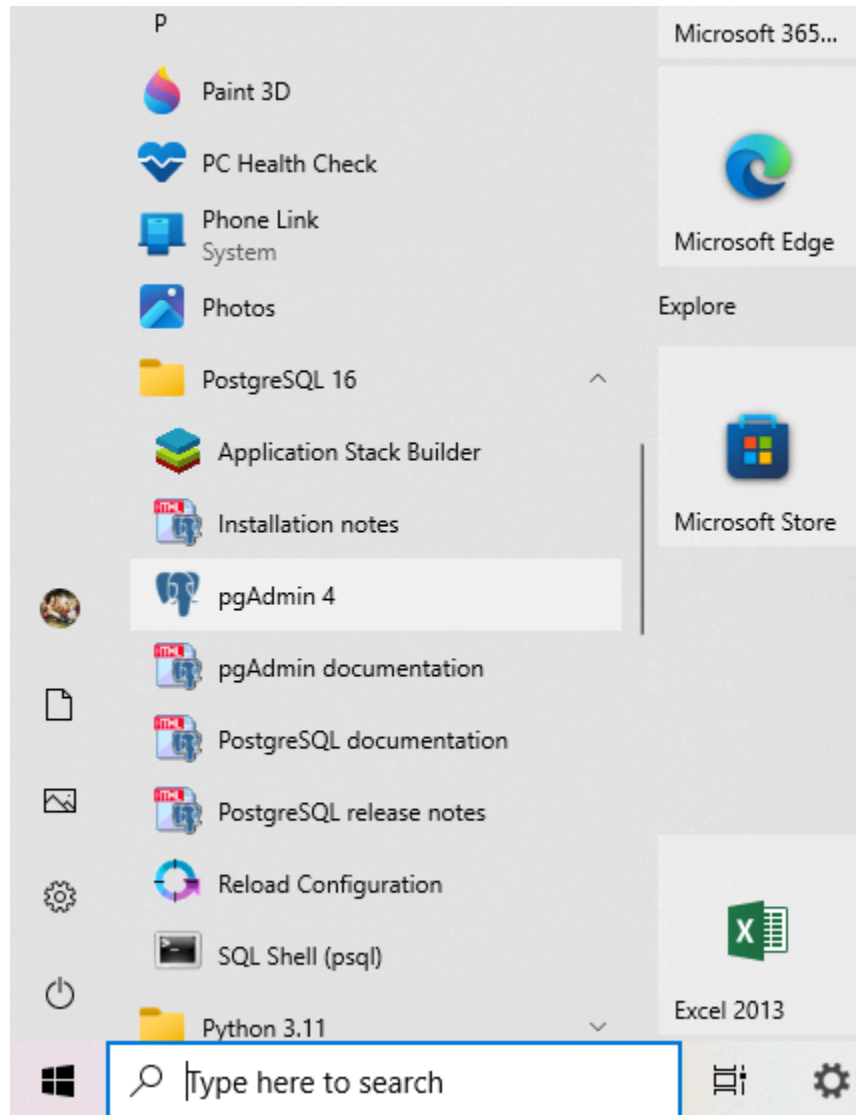
Step 5: Follow the installation steps by the installer



Next, mention the directory where PostgreSQL and components should be installed. I chose the D drive as it had more space.

Keep clicking on "Next" keeping the defaults.
Finally, install. It should take a few minutes.

Step 5: Now, Launch pgAdmin4



My problem: Even after doing a fresh installation, I faced issue like this:
pgAdmin Runtime Environment

Python Path: "D:\College Applications\PostgreSQL\16\pgAdmin 4\python\python.exe"
Runtime Config File: "C:\Users\hp\AppData\Roaming\pgadmin\runtime_config.json"
pgAdmin Config File: "D:\College Applications\PostgreSQL\16\pgAdmin 4\web\config.py"
Webapp Path: "D:\College Applications\PostgreSQL\16\pgAdmin 4\web\pgAdmin4.py"

pgAdmin Command: "D:\College Applications\PostgreSQL\16\pgAdmin4\python\python.exe -s D:\College Applications\PostgreSQL\16\pgAdmin4\web\pgAdmin4.py"

Environment:

- ALLUSERSPROFILE: C:\ProgramData
- APPDATA: C:\Users\hp\AppData\Roaming

This appears in a window, while trying to launch pgAdmin4.

A characteristic which lets us predict if this error window is going to open or not is that the loading time is long, and says that "Taking longer time to load...", something like that.

So what to do???

Step 5a: First, close pgAdmin4. Terminate Any Running pgAdmin or PostgreSQL Processes
Ensure no other instances of pgAdmin or PostgreSQL are running, which might be locking the database file.






To ensure that this action is thoroughly performed, do this:

- Open Task Manager:
- Press Ctrl + Shift + Esc to open the Task Manager.
- Look for any processes named pgAdmin, postgres, or similar.
- Select each of these processes and click End Task to terminate them.

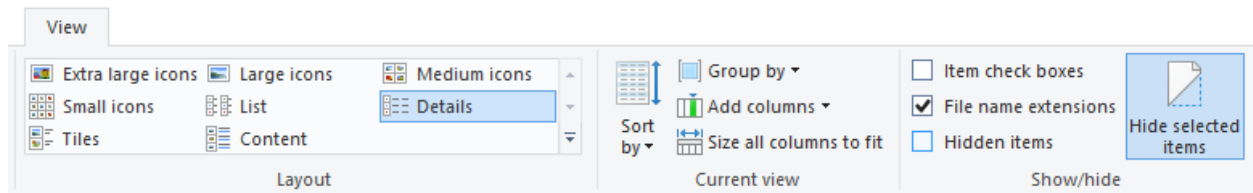
Step 5b: Backup and Delete Existing Database Files

Locate and back up the existing SQLite database file, then delete it to allow pgAdmin to recreate it afresh.

- Navigate to the pgAdmin Data Directory:
- Open File Explorer.
- Navigate to C:\Users\hp\AppData\Roaming\pgAdmin\.
- If you don't see AppData, it might be hidden. You can enable viewing hidden files by clicking on the View tab in File Explorer and checking Hidden items.

This PC > Local Disk (C:) > Users > hp >			
Name	Date modified	Type	Size
 3D Objects	18-Apr-22 0:40	File folder	
 AndroidStudioProjects	02-Sep-23 15:51	File folder	
 Cisco Packet Tracer 8.2.1	31-Oct-23 23:05	File folder	
 Contacts	26-Sep-23 16:55	File folder	
 Documents	07-May-22 13:58	File folder	

Where is AppData??



Check the “Hidden items” checkbox which can be found in the “View” tab.

This PC > Local Disk (C:) > Users > hp >

Name	Date modified	Type
3D Objects	18-Apr-22 0:40	File folder
AndroidStudioProjects	02-Sep-23 15:51	File folder
AppData	18-Apr-22 0:40	File folder
Cisco Packet Tracer 8.2.1	31-Oct-23 23:05	File folder
Contacts	26-Sep-23 16:55	File folder

Aha!

Next, go to “Roaming” folder.

> Local Disk (C:) > Users > hp > AppData >

Name	Date modified	Type
Local	25-May-24 13:47	File folder
LocalLow	22-Apr-24 1:24	File folder
Roaming	02-Apr-24 15:37	File folder

Now, go to “pgAdmin” folder.

This PC > Local Disk (C:) > Users > hp > AppData > Roaming >

Name	Date modified	Type
Macromedia	18-Apr-22 16:28	File folder
Microsoft	19-Jan-24 10:13	File folder
Mozilla	11-Feb-24 21:26	File folder
npm	16-Apr-23 16:40	File folder
pgAdmin	25-May-24 17:58	File folder

You will find:

This PC > Local Disk (C:) > Users > hp > AppData > Roaming > pgAdmin > ▼ 🔄 Search

Name	Date modified	Type	Size
azurecredentialcache	25-May-24 13:52	File folder	
sessions	25-May-24 17:50	File folder	
storage	25-May-24 13:52	File folder	
pgadmin4.db	25-May-24 17:58	Data Base File	168 KB
pgadmin4.db.prev.bak	25-May-24 17:25	BAK File	37 KB
pgadmin4.log	25-May-24 17:45	Text Document	22 KB
pgadmin4.log.2023-03-24_19-38-10	24-Mar-23 19:38	2023-03-24_19-38-...	4 KB
runtime_config.json	25-May-24 13:56	JSON Source File	1 KB

You may take backup of the highlighted file, or just delete it. This file is what is causing the problem.

Step 5c: Clear Cache and Temp Files

Clear temporary files that might interfere with the installation process.

Use Disk Cleanup:

- Type Disk Cleanup in the Windows search bar and open it.
- Select the drive where temporary files are stored (usually C:) and click OK.
- Check the boxes for Temporary files and other categories you wish to clean.
- Click OK and then Delete Files.

Manually Delete Temp Files:

- Navigate to C:\Users\hp\AppData\Local\Temp.
- Select all files and folders in this directory and delete them.

Alternately, I suggest you to not “Select all files and folders in this directory and delete them”.

What if some of the files are important for some other application?

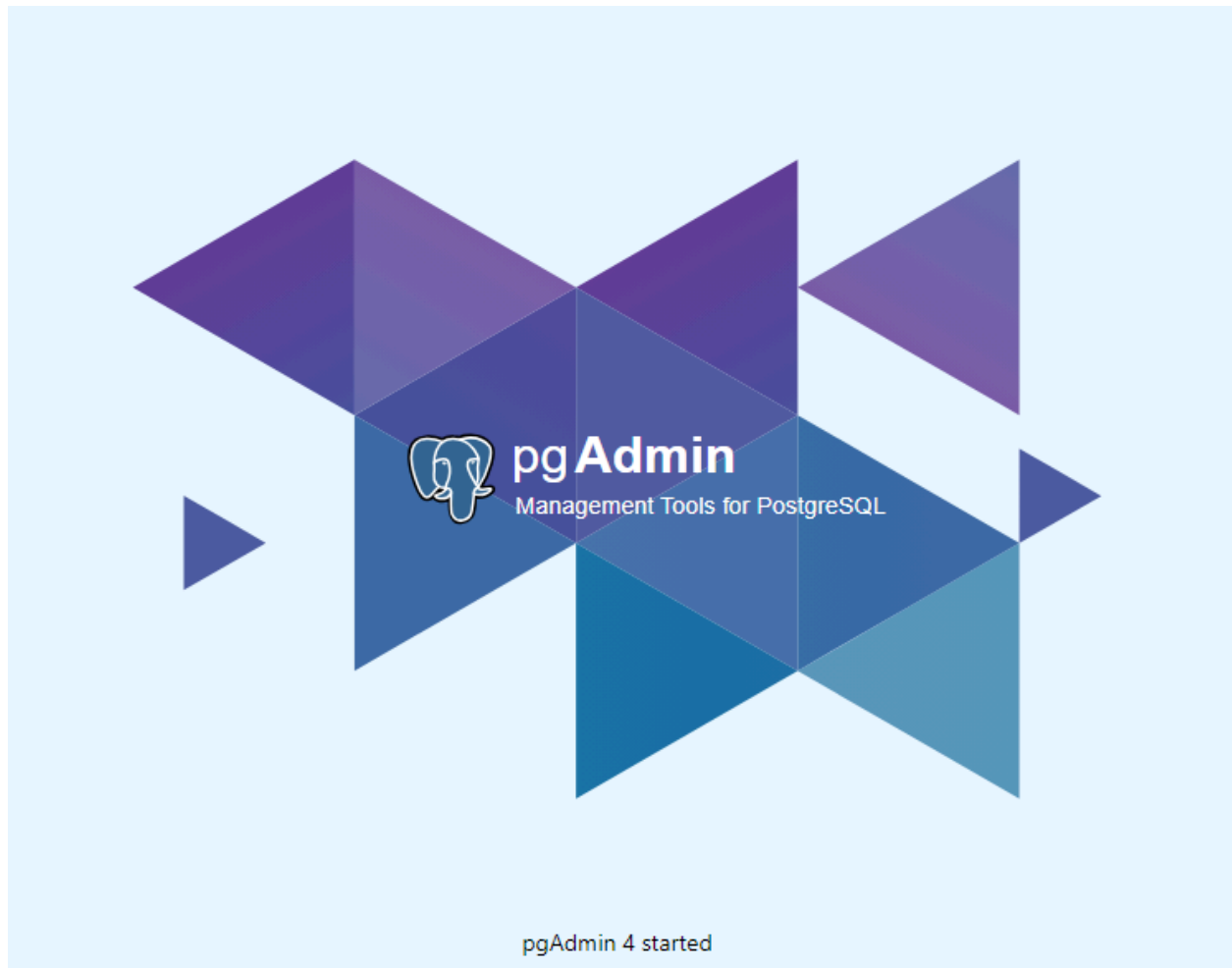
Here’s what I suggest:

This PC > Local Disk (C:) > Users > hp > AppData > Local > Temp ▼ 🔄 Search

Name	Date modified	Type	Size
bc608eeb-359b-4c46-a9b7-a94510fe79eb...	25-May-24 19:01	TMP File	0 KB
34d1f1e6-a572-4f11-b114-cfe47c5f87e9.t...	25-May-24 18:55	TMP File	11,769 KB
JET1C65.tmp	25-May-24 18:50	TMP File	0 KB
installbuilder_installer.log	25-May-24 18:49	Text Document	5 KB
a46aecbb-c06c-4ab7-8a49-ecedabeec37a...	25-May-24 18:47	TMP File	0 KB

Click on the “Date modified” to sort it according to descending order - the most recent files appear first. Delete the files of just a few days back, maybe, say, 3 days back. Now, try opening the application again.

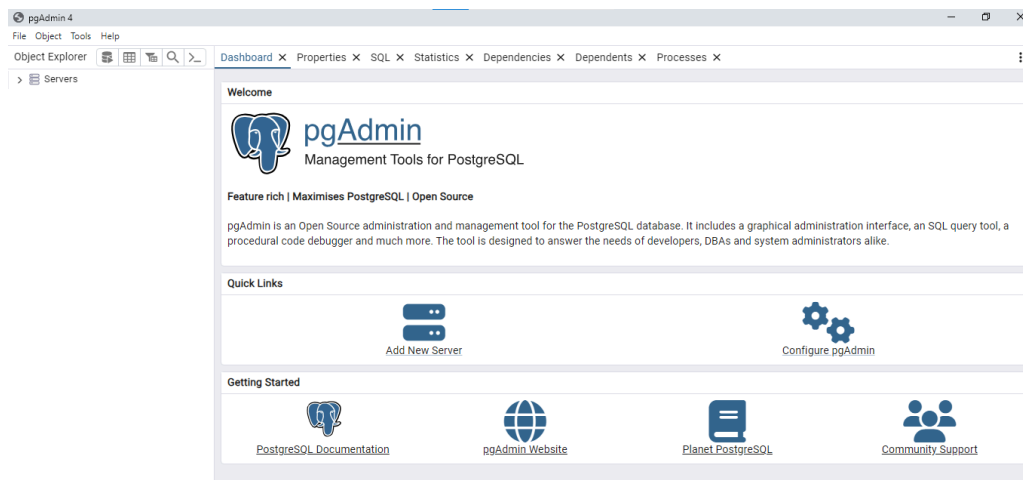
Step 5d: Retry opening the application



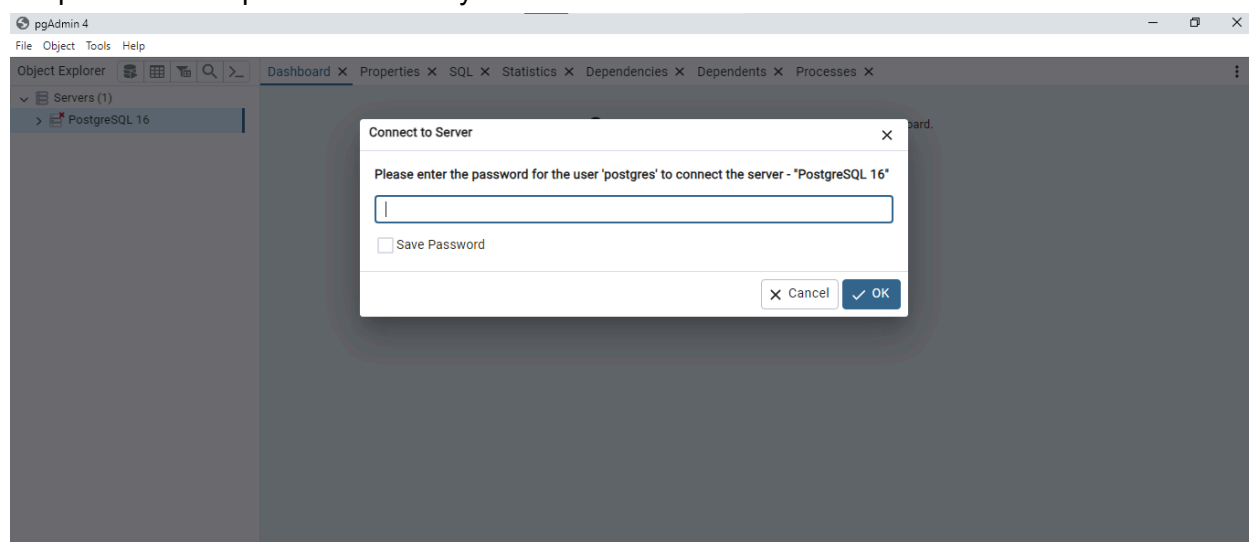
Two things you should notice while the pgAdmin 4 is starting:

1. It is much faster, and phrases like “Taking a long time...” won’t appear.
2. Most importantly, the message as shown above “pgAdmin 4 started” appears. Even if it takes a long time, don’t worry. But this second message must appear.

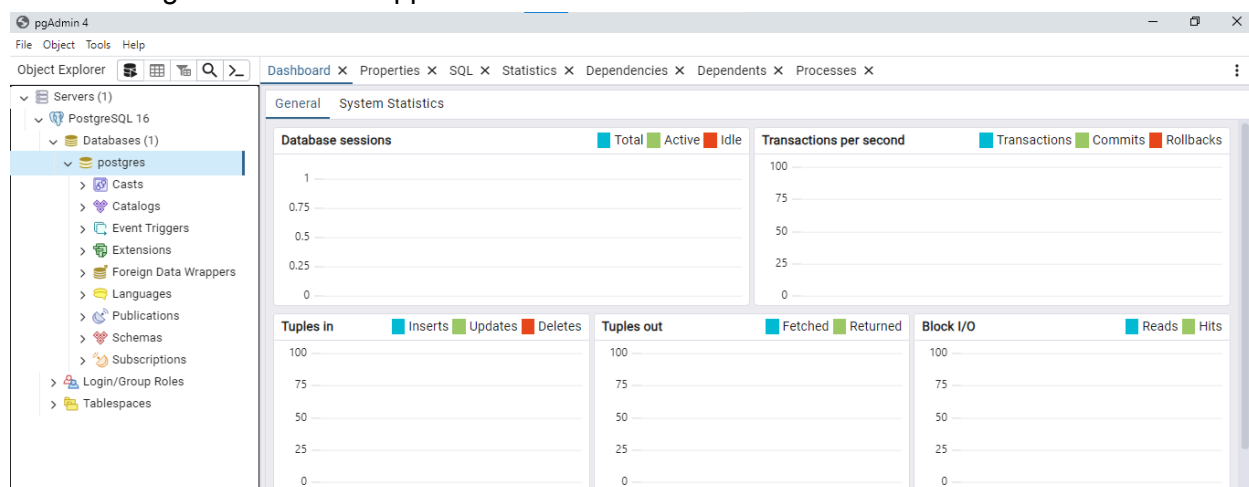
Step 6: The UI



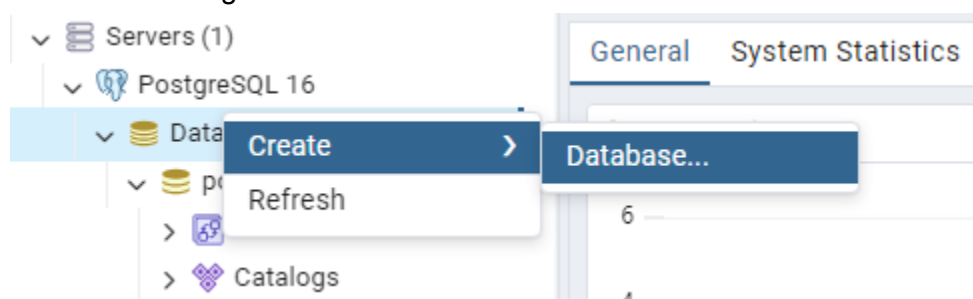
Step 7: Enter the password which you mentioned at the time of installation.



The following screen should appear:



Step 8: Create the peer_review database by right-clicking on “Databases” on the left-hand-side tree and selecting “Create Database”.



Step 9: Enter details like Name.

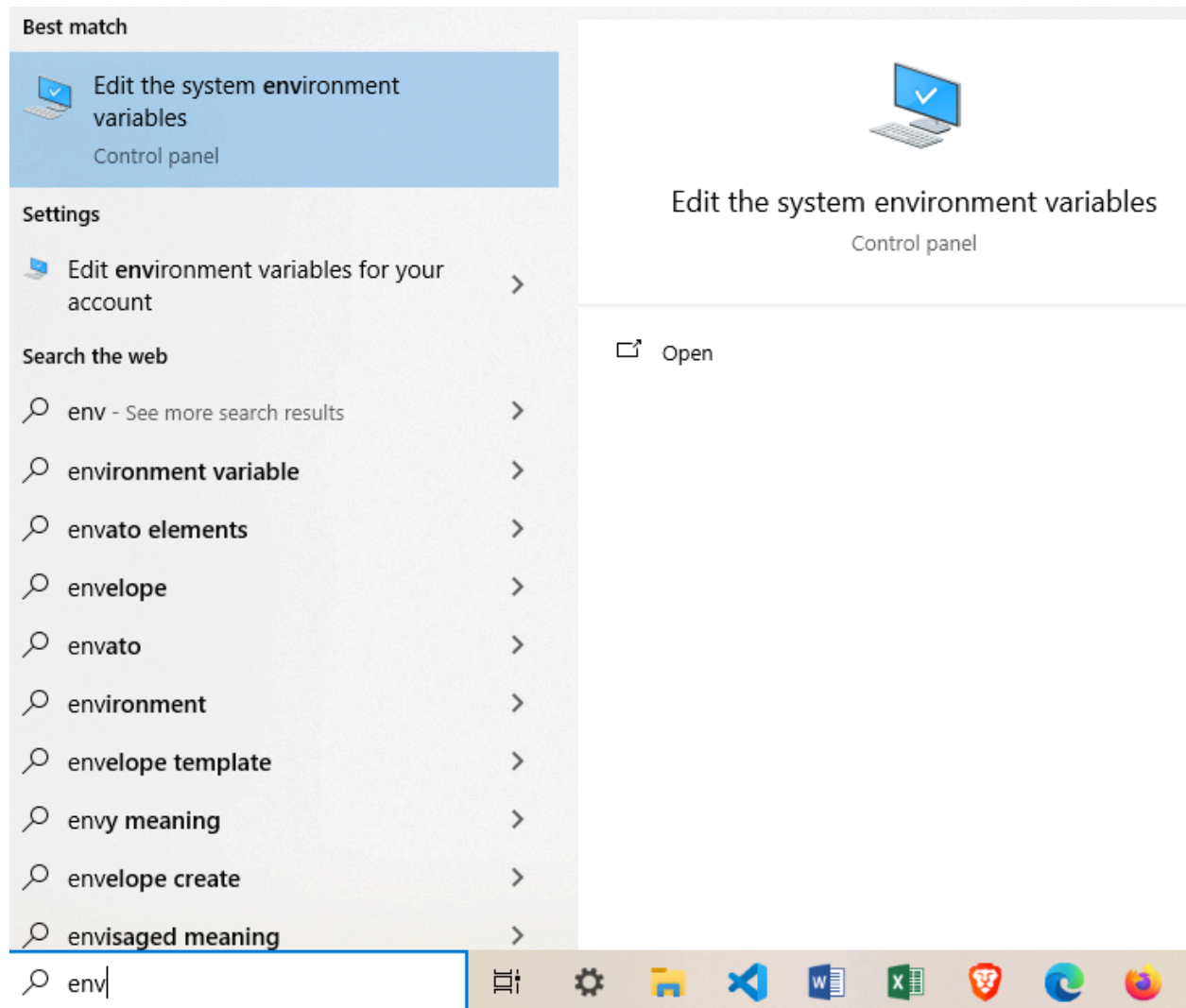
Despite all this,

```
File "C:\gh_repos\Distributed_Peer_Review_of_submission\env\Lib\site-packages\django\utils\connection.py", line 15, in __getattr__
    return getattr(self._connections[self._alias], item)
File "C:\gh_repos\Distributed_Peer_Review_of_submission\env\Lib\site-packages\django\utils\connection.py", line 62, in __getitem__
    conn = self.create_connection(alias)
File "C:\gh_repos\Distributed_Peer_Review_of_submission\env\Lib\site-packages\django\db\utils.py", line 204, in create_connection
    backend = load_backend(db['ENGINE'])
File "C:\gh_repos\Distributed_Peer_Review_of_submission\env\Lib\site-packages\django\db\utils.py", line 111, in load_backend
    return import_module('%s.base' % backend_name)
File "C:\Users\hp\AppData\Local\Programs\Python\Python311\Lib\importlib\_init_.py", line 126, in import_module
    return bootstrap._gcd_import(name[level:], package, level)
File "C:\gh_repos\Distributed_Peer_Review_of_submission\env\Lib\site-packages\django\db\backends\postgresql\base.py", line 29, in <module>
    raise ImproperlyConfigured("Error loading psycopg2 module: %s" % e)
django.core.exceptions.ImproperlyConfigured: Error loading psycopg2 module: DLL load failed while importing _psycopg: The specified module could not be found.

(env) PS C:\gh_repos\Distributed_Peer_Review_of_submission>
```

Oh no!

py manage.py makemigrations does not run!



System Properties



Computer Name Hardware **Advanced** System Protection Remote

You must be logged on as an Administrator to make most of these changes.

Performance

Visual effects, processor scheduling, memory usage, and virtual memory

Settings...

User Profiles

Desktop settings related to your sign-in

Settings...

Startup and Recovery

System startup, system failure, and debugging information

Settings...

Environment Variables...

OK

Cancel

Apply

User variables for hp

Variable	Value
BEARER_TOKEN	AAAAAAAAAAAAAAAAAAAAAAKgXogEAAAAAgtf6ahOB2f9AqAUO...
OneDrive	C:\Users\hp\OneDrive - somaiya.edu
OneDriveCommercial	C:\Users\hp\OneDrive - somaiya.edu
Path	C:\Users\hp\AppData\Local\Programs\Python\Python311\Scripts\...
TEMP	C:\Users\hp\AppData\Local\Temp
TMP	C:\Users\hp\AppData\Local\Temp

New...

Edit...

Delete

System variables

Variable	Value
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	4
OS	Windows_NT
Path	C:\Program Files\Common Files\Oracle\Java\javapath;C:\Program ...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64

New...

Edit...

Delete

OK

Cancel

System variables

Variable	Value
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	4
OS	Windows_NT
Path	C:\Program Files\Common Files\Oracle\Java\javapath;C:\Program ...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64

New...

Edit...

Delete

Edit environment variable



C:\Program Files\Common Files\Oracle\Java\javapath
C:\Program Files (x86)\Common Files\Oracle\Java\javapath
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files\nodejs\
C:\Users\hp\AppData\Local\Programs\Python\Python311\Scripts
C:\Users\hp\DB Browser for SQLite
C:\Program Files\swipl\bin
C:\Program Files\Graphviz\bin
D:\College Applications\PostgreSQL\16\bin

New

Edit

Browse...

Delete

Move Up

Move Down

Edit text...

OK

Cancel

```
(env) PS C:\gh_repos\Distributed_Peer_Review_of_submission> pip uninstall psycopg2 psycopg2-binary
Found existing installation: psycopg2 2.9.3
Uninstalling psycopg2-2.9.3:
  Would remove:
    c:\gh_repos\distributed_peer_review_of_submission\env\lib\site-packages\psycopg2-2.9.3.dist-info\*
    c:\gh_repos\distributed_peer_review_of_submission\env\lib\site-packages\psycopg2\*
Proceed (Y/n)? Y
Successfully uninstalled psycopg2-2.9.3
WARNING: Skipping psycopg2-binary as it is not installed.
(env) PS C:\gh_repos\Distributed_Peer_Review_of_submission> pip install psycopg2-binary
Collecting psycopg2-binary
  Obtaining dependency information for psycopg2-binary from https://files.pythonhosted.org/packages/25/dc1de686/psycopg2_binary-2.9.9-cp311-cp311-win_amd64.whl.metadata
    Downloading psycopg2_binary-2.9.9-cp311-cp311-win_amd64.whl.metadata (4.6 kB)
  Downloading psycopg2_binary-2.9.9-cp311-cp311-win_amd64.whl (1.2 MB)
    1.2/1.2 MB 1.8 MB/s eta 0:00:00
Installing collected packages: psycopg2-binary
Successfully installed psycopg2-binary-2.9.9
```

```
(env) PS C:\gh_repos\Distributed_Peer_Review_of_submission> py manage.py makemigrations
C:\gh_repos\Distributed_Peer_Review_of_submission\env\Lib\site-packages\django\core\management\commands\makemigrations.py:105: RuntimeWarning: Got an error checking a consistent migration history performed for database connection 'default': connection to server at "localhost" (:::1), port 5432 failed: FATAL: database "peergradeview-system" does not exist
warnings.warn(
Migrations for 'auth':
  env\Lib\site-packages\django\contrib\auth\migrations\0013_alter_user_email.py
    - Alter field email on user
(env) PS C:\gh_repos\Distributed_Peer_Review_of_submission>
```

Activate Windows
Go to Settings to activate Windows.

Solution:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        # 'NAME': "peergradeview-system",
        'NAME': "peer_grade",
        'USER': "postgres",
        'PASSWORD': "<PUT THAT PASSWORD HERE WHICH YOU SET AT THE TIME OF INSTALLATION>",
        'HOST': "localhost",
        'PORT': "5432"
    }
}
```

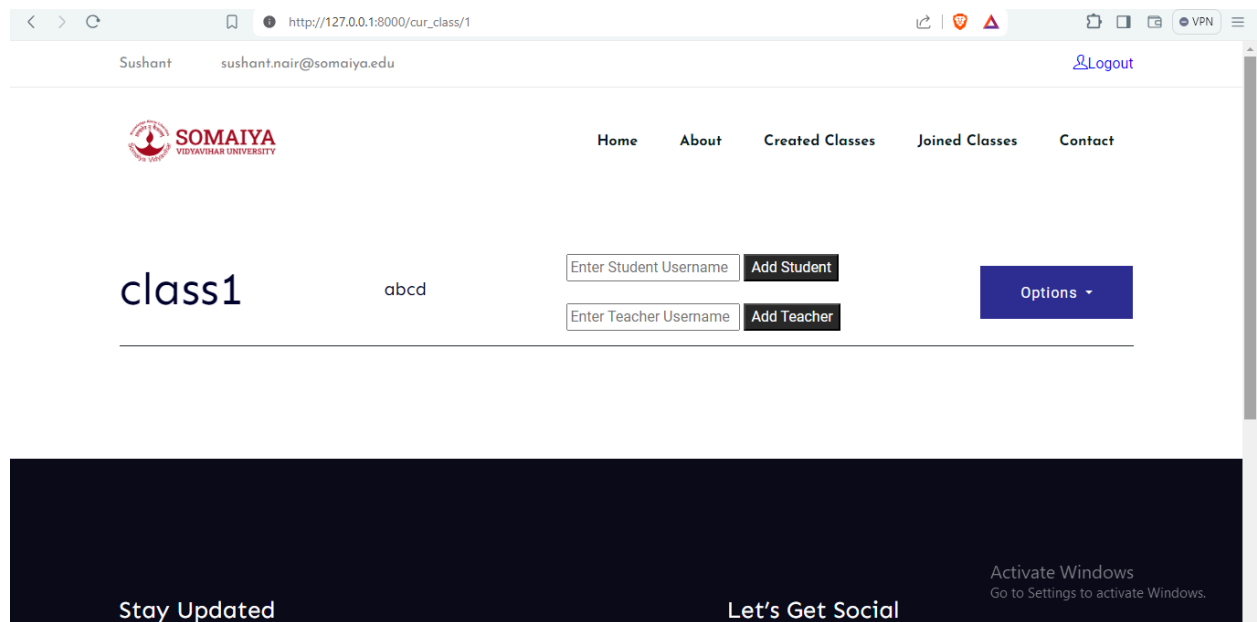
```
(env) PS C:\gh_repos\Distributed_Peer_Review_of_submission> py manage.py migrate
Operations to perform:
  Apply all migrations: account, admin, auth, contenttypes, courses, sessions, sites, socialaccount
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying account.0001_initial... OK
  Applying account.0002_email_max_length... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
```

1. peergrade/settings.py: This entry tells Git to ignore the specific file settings.py located in the peergrade directory. As a result, any changes made to peergrade/settings.py will not be tracked by Git. This is often done to avoid committing sensitive information like database credentials, API keys, or configuration settings that are specific to your local environment and should not be shared or exposed in the repository.
2. env/: This entry tells Git to ignore the entire env directory and all its contents. Typically, env/ is used for virtual environments in Python projects, where all the project-specific dependencies are installed. Ignoring this directory prevents the large number of files in the virtual environment from being tracked by Git. It also ensures that each developer

can create their own virtual environment locally without causing conflicts or bloat in the repository.

Finally,

```
py manage.py runserver
```



Phew! So gratifying that it ran 😊.