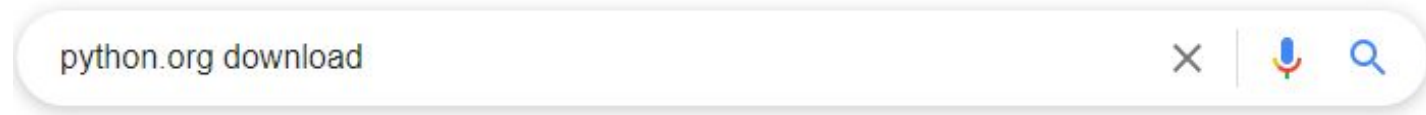


Python Installation & Environment Setup

Python Installation

- In your browser type Python.org download.



- Go to the page

<https://www.python.org> > downloads ⓘ

Download Python

See the main Documentation page. Information on tools for unpacking archive files provided on **python.org** is available. Tip: even if you **download** a ready-made ...

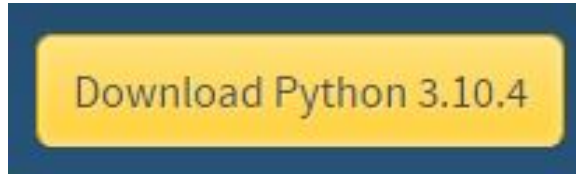
Python Installation

- Click on the first option.
- Then web appear as below.

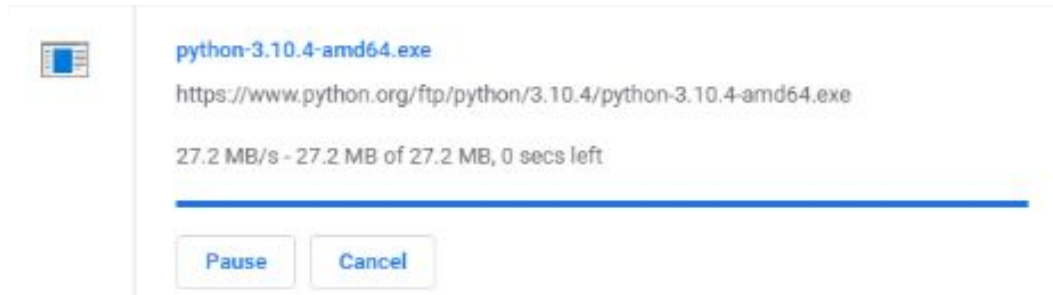


Python Installation

- Click on “Download Python 3.10.4”(Note: Make sure your python version is “3.9.x” Or above version.



- Download the setup file



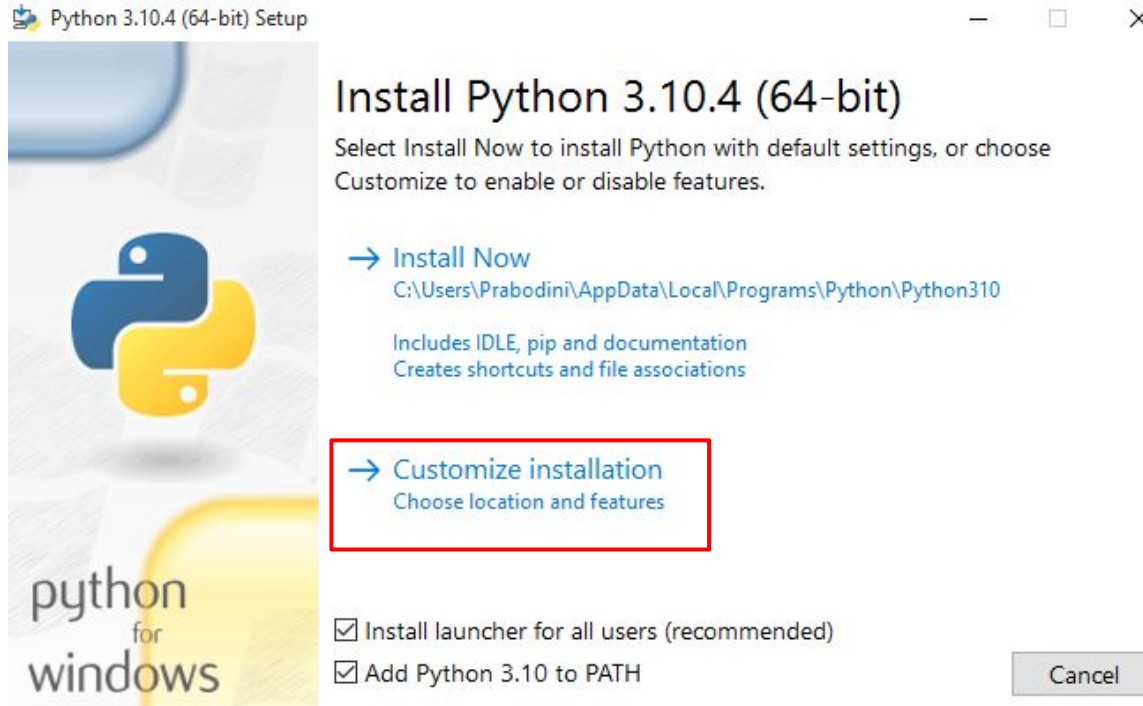
Python Installation

- After running the program the below dialog box is appeared.
- Put a tick in 'Add Python 3.10 to PATH'.



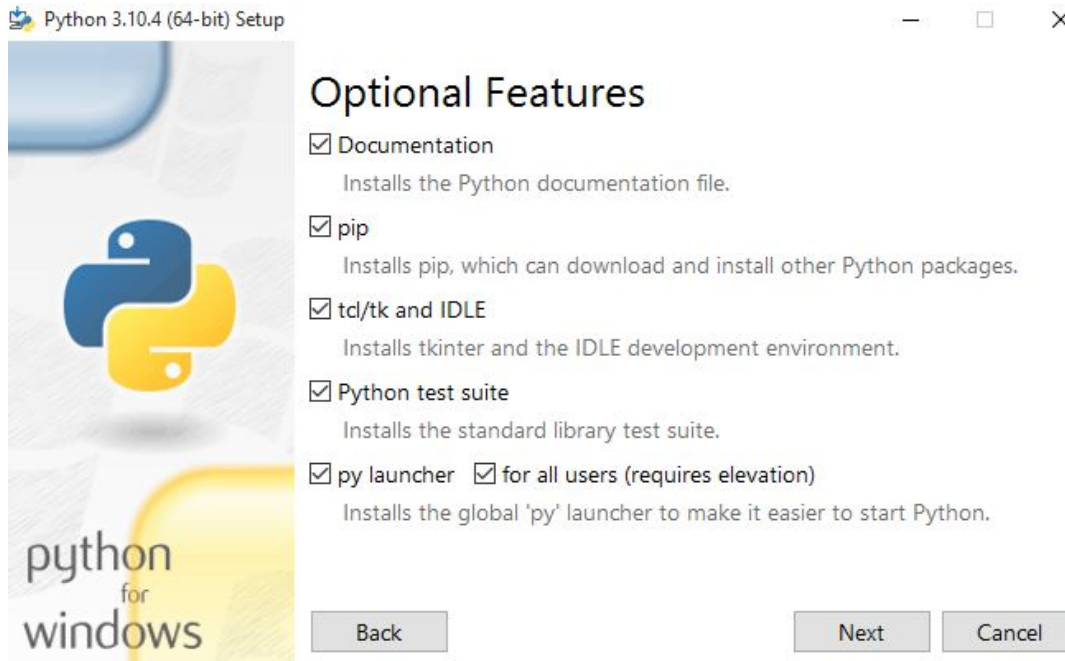
Python Installation

- Then select 'Customize Installation'



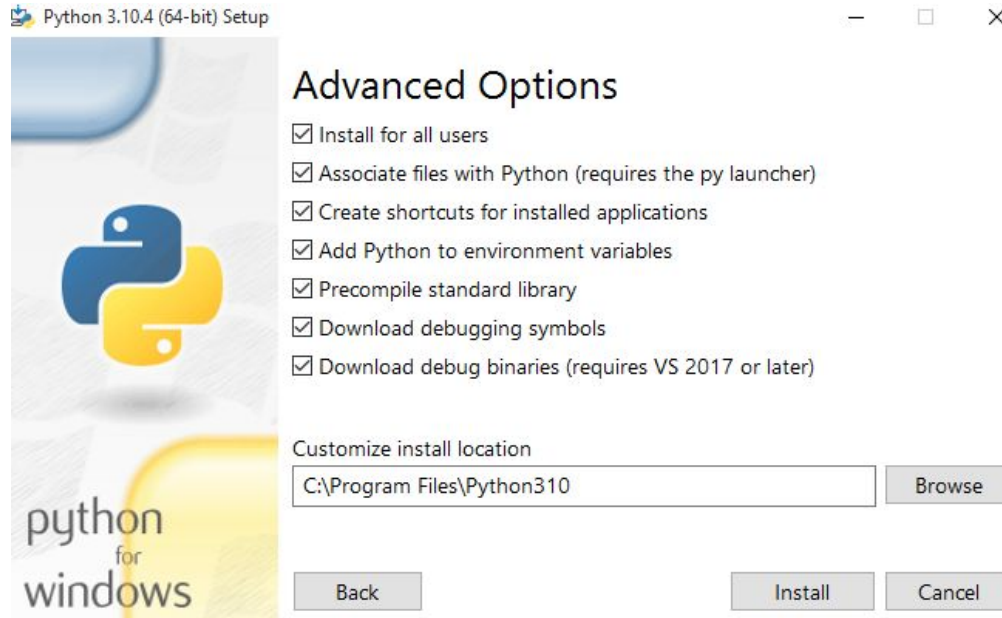
Python Installation

- After that put a tick all the 'Optional Features'.
- Then click on 'Next'.



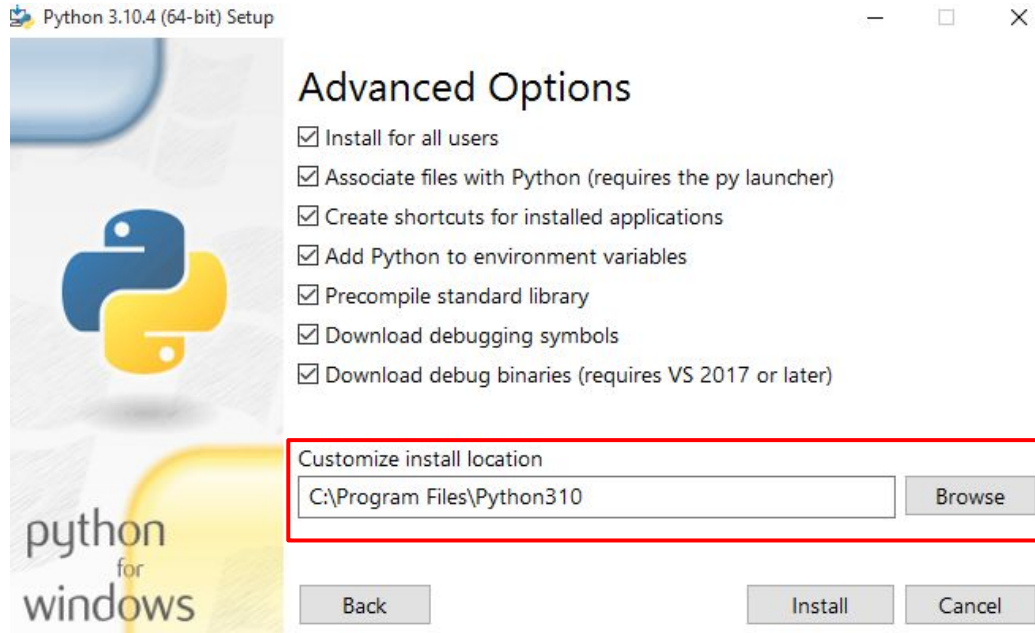
Python Installation

- Then the below window appear.
- Tick on all 'Advanced Options'.



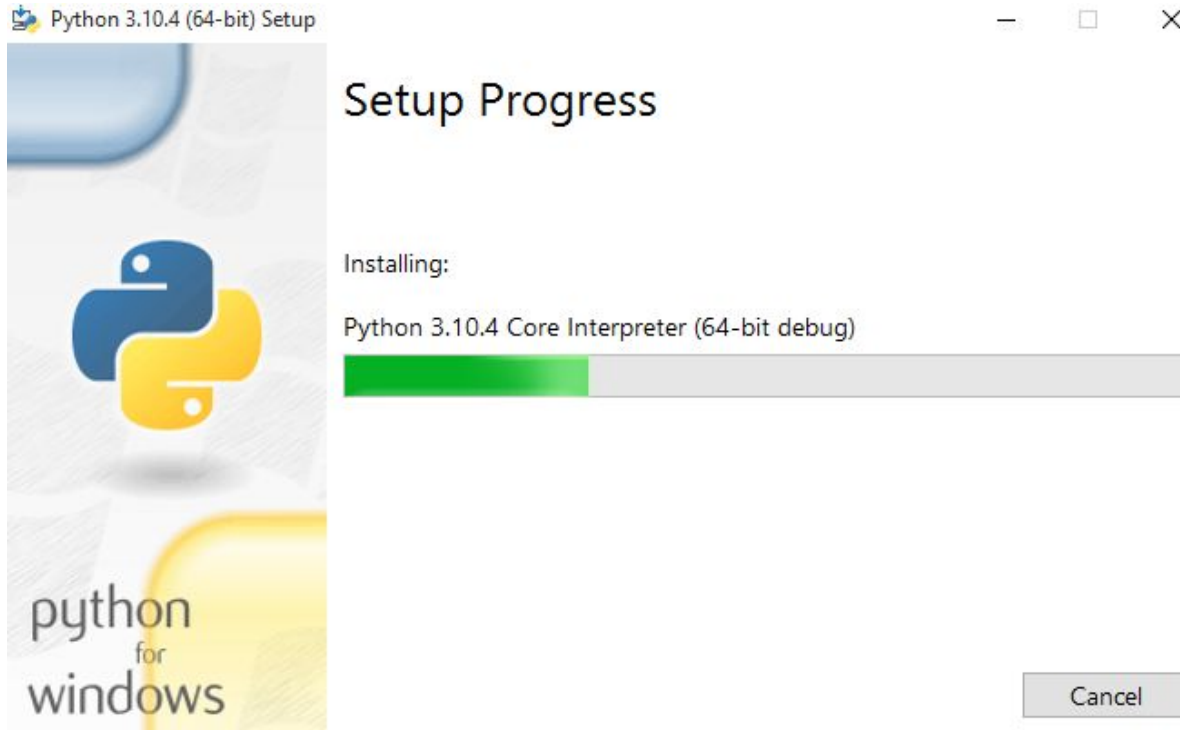
Python Installation

- Browse the destination in 'C:\Program Files'.
- Then click on 'Install'.



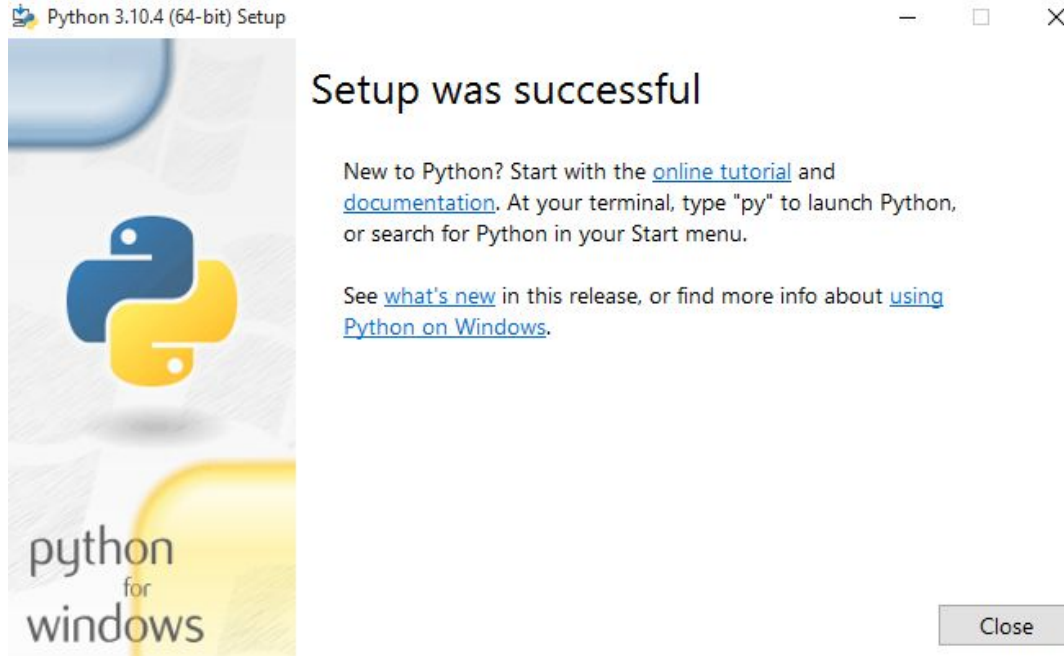
Python Installation

- Wait till the 'Setup Progress'.



Python Installation

- After that below window 'Setup was successful' appear.
- Then click on 'Close'.

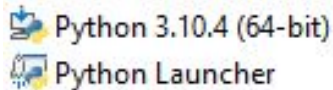


Python Installation

- To check your installation,
 - Go to search bar and type 'Control Panel'.
 - Then go to '**Program and Features**'.



- If we install Python correctly, we can see Python 3.10.4 (64-bit) in Program and Features.



Python Software Foundation
Python Software Foundation

Python Installation(Check Path)

- Again in the search bar type 'cmd' and open a command prompt.
- In command prompt type '**python --version**' and Enter. (To check the PATH is set and current python version)

```
.>python --version
```

- If you install correctly, after we enter the version should be appear as below.

```
Python 3.10.4
```

Create Project Folder

- Change the current directory to that directory where the project folder will be created. Type '**cd *directory_path***' in command prompt.

```
>cd project_class
```

- Create a project folder by Command Prompt. Type '**mkdir *folder_name***' in command prompt.

```
>mkdir project_class
```

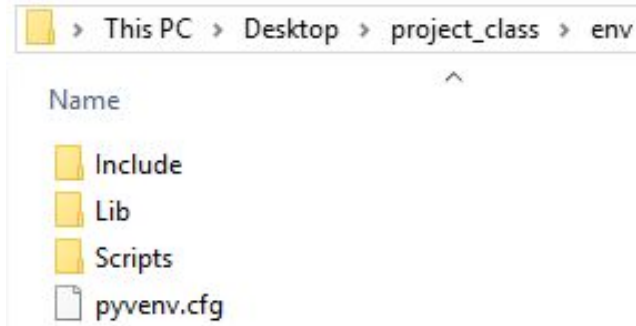
Create a Virtual Environment

- Create a Virtual Environment by command '**python -m venv env**'.

```
>python -m venv env
```

Name of Virtual Environment

- After creating Virtual Environment will get a subfolder in project folder named '**env**' as follows.



Activate the Virtual Environment

- Activate the Virtual Environment by 'env\Scripts\activate' command.

```
C:\Users\Prabodini\Desktop\project_class>env\Scripts\activate
```

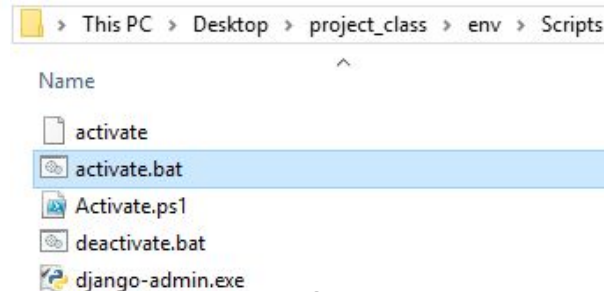
- Once a Virtual Environment is activated its name is appeared at the beginning of the current directory.

```
(env) C:\Users\Prabodini\Desktop\project_class>
```


Activate the Virtual Environment

- Activate the Virtual Environment by 'env\Scripts\activate' command.

```
C:\Users\Prabodini\Desktop\project_class>env\Scripts\activate
```



- Once a Virtual Environment is activated its name is appeared at the beginning of the current directory.

```
(env) C:\Users\Prabodini\Desktop\project_class>
```

Install Django

- Install Django by '**pip install django**' command.

```
(env) C:\Users\Prabodini\Desktop\project_class>pip install django
Collecting django
  Using cached Django-4.0.4-py3-none-any.whl (8.0 MB)
Collecting tzdata
  Using cached tzdata-2022.1-py2.py3-none-any.whl (339 kB)
Collecting asgiref<4,>=3.4.1
  Using cached asgiref-3.5.1-py3-none-any.whl (22 kB)
Collecting sqlparse>=0.2.2
  Using cached sqlparse-0.4.2-py3-none-any.whl (42 kB)
Installing collected packages: tzdata, sqlparse, asgiref, django
Successfully installed asgiref-3.5.1 django-4.0.4 sqlparse-0.4.2 tzdata-2022.1
```

- Django is installed your virtual environment.

Create a Project Folder with initial files

- Create a project folder by 'django-admin startproject *project_name*' command.

```
>django-admin startproject day01
```

Project
Name

- Go to project folder.

```
>cd day01
```

> This PC > Desktop > project_class

Name

day01
env

> This PC > Desktop > project_class > day01

Name

day01
manage.py

Run the Project

- Run the project by **'python manage.py runserver'** command.

```
(env) C:\Users\Prabodini\Desktop\project_class\day01>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...
```

```
System check identified no issues (0 silenced).
May 12, 2022 - 11:24:53
Django version 4.0.4, using settings 'day01.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Copy the url &
open it in browser



- To stop the server by pressing **'Ctrl + C'** in cmd.

django

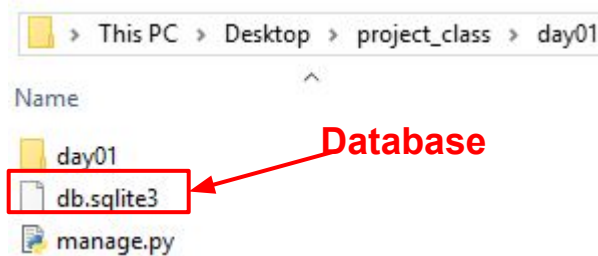
View [release notes](#) for Django 4.0



The install worked successfully! Congratulations!

Database

- After running the project, the database file (**db.sqlite3**) will be created as below.



- Download and install the '**DB Browser**' software to browse the database(store all the data which related your project).



Database

- Open the database using 'DB Browser'. You can see no data or table in initial stage as below.



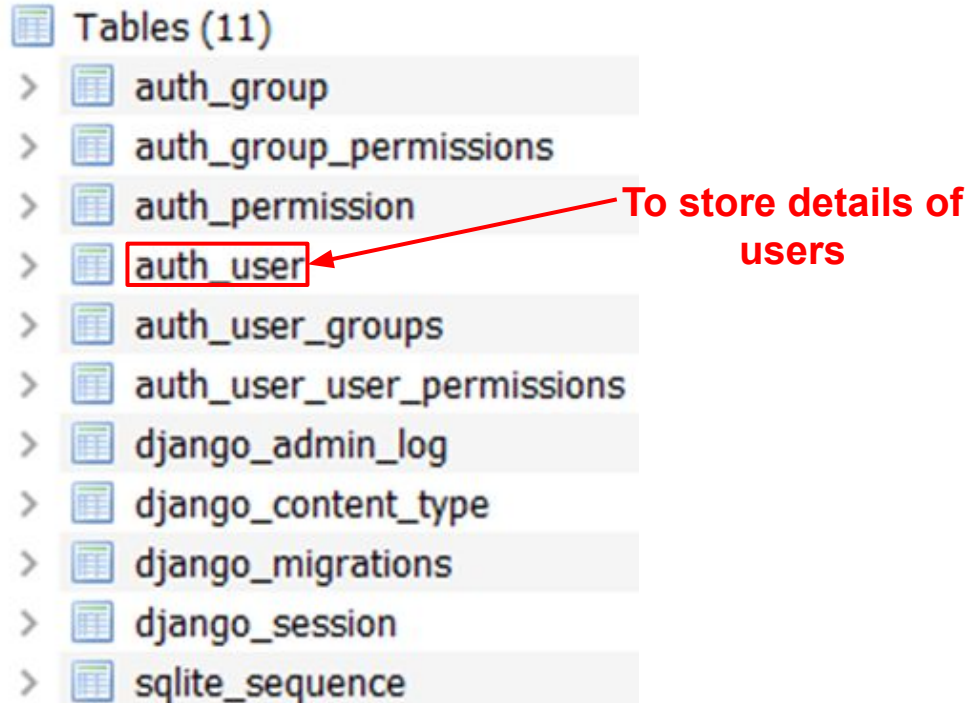
Create tables in Database

- Create tables by ‘python manage.py migrate’ command.

```
(env) C:\Users\Prabodini\Desktop\project_class\day01>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... 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
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK
```












Created tables in Database

- Go to 'DB Browser' and observe the created tables as follow.



Data of 'auth_user' table

- Open 'auth_user' table to see what data is stored as below.

 id	integer	"id" integer NOT NULL
 password	varchar(128)	"password" varchar(128) NOT NULL
 last_login	datetime	"last_login" datetime
 is_superuser	bool	"is_superuser" bool NOT NULL
 username	varchar(150)	"username" varchar(150) NOT NULL UNIQUE
 last_name	varchar(150)	"last_name" varchar(150) NOT NULL
 email	varchar(254)	"email" varchar(254) NOT NULL
 is_staff	bool	"is_staff" bool NOT NULL
 is_active	bool	"is_active" bool NOT NULL
 date_joined	datetime	"date_joined" datetime NOT NULL
 first_name	varchar(150)	"first_name" varchar(150) NOT NULL

Browse data of 'auth_user' table

- Right click on 'auth_user' table -> Browse Table to see what data is stored now as below.

id	password	last_login	is_superuser	username	last_name	email	is_staff	is_active
Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter

Create a Super User

- Create a super user who has all the power by 'python manage.py createsuperuser' command.


```
(env) C:\Users\Prabodini\Desktop\project_class\day01>python manage.py createsuperuser
Username (leave blank to use 'prabodini'): admin ← Username
Email address: admin@abc.com ← Email
Password: ← Password - 1234
Password (again):
This password is too short. It must contain at least 8 characters.
This password is too common.
This password is entirely numeric.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
```

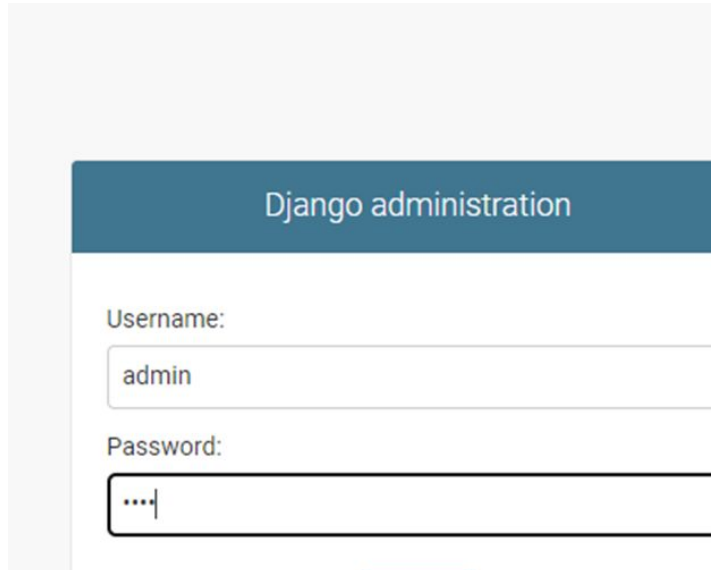
- Created super user in 'auth_user' table.

id	password	last_login	is_superuser	username
Filter	Filter	Filter	Filter	Filter
1	pbkdf2_sha256\$320000\$iP0niocSn9qQMImZvsb1...	NULL	1	admin

Create other users using admin panel

- Open admin panel.

 localhost:8000/admin



The screenshot shows the Django administration interface. At the top, there is a dark blue header with the text "Django administration". Below the header, there are two input fields. The first is labeled "Username:" and contains the text "admin". The second is labeled "Password:" and contains four dots, indicating a password field. Below the password field, there is a blue line.

Create other users using admin panel

- Store user details in user table. (Note: Regardless of the username password is “lnbti@123”)

Action: 0 of 5 selected

<input type="checkbox"/>	USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
<input type="checkbox"/>	admin	admin@abc.com			✓
<input type="checkbox"/>	kasun				✗
<input type="checkbox"/>	nimal				✗
<input type="checkbox"/>	rani				✗
<input type="checkbox"/>	saman				✗

5 users

Summary

1. Install python.
2. Check whether the path is set.
3. Create a folder to store all our project.
4. Create a Virtual Environment.
5. Activate the Virtual Environment.
6. Install Django to the virtual environment.
7. Create the project.
8. Run server and observe it in the browser.
9. Observe the empty database.
10. Migrate (create tables)
11. Create super user.
12. Log in to the system using super user table.
13. Create few other users to users table.

Thank You