EDGE Class 4

Titles: Django

Django is a back-end server side web framework.

Django is free, open source and written in Python.

Django makes it easier to build web pages using Python.

## What is Django?

Django is a Python framework that makes it easier to create web sites using Python.

Django takes care of the difficult stuff so that you can concentrate on building your web applications.

Django emphasizes reusability of components, also referred to as DRY (Don't Repeat Yourself), and comes with ready-to-use features like login system, database connection and CRUD operations (Create Read Update Delete).

Django is especially helpful for database driven websites.

How does Django Work?

Django follows the MVT design pattern (Model View Template).

* Model - The data you want to present, usually data from a database.
* View - A request handler that returns the relevant template and content - based on the request from the user.
* Template - A text file (like an HTML file) containing the layout of the web page, with logic on how to display the data.

## Model

The model provides data from the database.

In Django, the data is delivered as an Object Relational Mapping (ORM), which is a technique designed to make it easier to work with databases.

The most common way to extract data from a database is SQL. One problem with SQL is that you have to have a pretty good understanding of the database structure to be able to work with it.

Django, with ORM, makes it easier to communicate with the database, without having to write complex SQL statements.

The models are usually located in a file called models.py.

## View

A view is a function or method that takes http requests as arguments, imports the relevant model(s), and finds out what data to send to the template, and returns the final result.

The views are usually located in a file called views.py.

## Template

A template is a file where you describe how the result should be represented.

Templates are often .html files, with HTML code describing the layout of a web page, but it can also be in other file formats to present other results, but we will concentrate on .html files.

Django uses standard HTML to describe the layout, but uses Django tags to add logic:

<h1>My Homepage</h1>

<p>My name is {{ firstname }}.</p>

The templates of an application is located in a folder named templates.

## URLs

Django also provides a way to navigate around the different pages in a website.

When a user requests a URL, Django decides which view it will send it to.

This is done in a file called urls.py.

So, What is Going On?

When you have installed Django and created your first Django web application, and the browser requests the URL, this is basically what happens:

1. Django receives the URL, checks the urls.py file, and calls the view that matches the URL.
2. The view, located in views.py, checks for relevant models.
3. The models are imported from the models.py file.
4. The view then sends the data to a specified template in the template folder.
5. The template contains HTML and Django tags, and with the data it returns finished HTML content back to the browser.

Django can do a lot more than this, but this is basically what you will learn in this tutorial, and are the basic steps in a simple web application made with Django.

Django History

Django was invented by Lawrence Journal-World in 2003, to meet the short deadlines in the newspaper and at the same time meeting the demands of experienced web developers.

Initial release to the public was in July 2005.

Latest version of Django is 4.0.3 (March 2022).

## Virtual Environment

## For Windows

## py -m venv myworld

Unix/MacOS:

python -m venv myworld

django-admin --version

https://www.geeksforgeeks.org/how-to-create-a-django-project/

## What is an App?

An app is a web application that has a specific meaning in your project, like a home page, a contact form, or a members database.

In this tutorial we will create an app that allows us to list and register members in a database.

But first, let's just create a simple Django app that displays "Hello World!".

## Create App

I will name my app members.

Start by navigating to the selected location where you want to store the app, in my case the my\_tennis\_club folder, and run the command below.

If the server is still running, and you are not able to write commands, press [CTRL] [BREAK], or [CTRL] [C] to stop the server and you should be back in the virtual environment.

py manage.py startapp members

Django creates a folder named members in my project, with this content:

my\_tennis\_club  
    manage.py  
    my\_tennis\_club/  
    members/  
        migrations/  
            \_\_init\_\_.py  
        \_\_init\_\_.py  
        admin.py  
        apps.py  
        models.py  
        tests.py  
        views.py

## Views

Django views are Python functions that takes http requests and returns http response, like HTML documents.

A web page that uses Django is full of views with different tasks and missions.

Views are usually put in a file called views.py located on your app's folder.

There is a views.py in your members folder that looks like this:

A Django model is a table in your database.

## Django Models

Up until now in this tutorial, output has been static data from Python or HTML templates.

Now we will see how Django allows us to work with data, without having to change or upload files in the process.

In Django, data is created in objects, called Models, and is actually tables in a database.

## Create Table (Model)

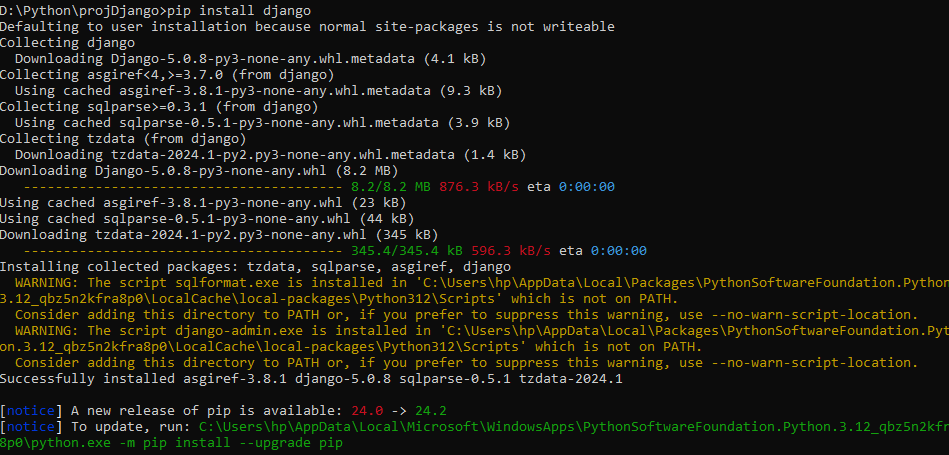
To create a model, navigate to the models.py file in the /members/ folder.

Open it, and add a Member table by creating a Member class, and describe the table fields in it:

Installation:

C:\Users\hp\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.12\_qbz5n2kfra8p0\LocalCache\local-packages\Python312\Scripts

pip install django



add an environment variable to system

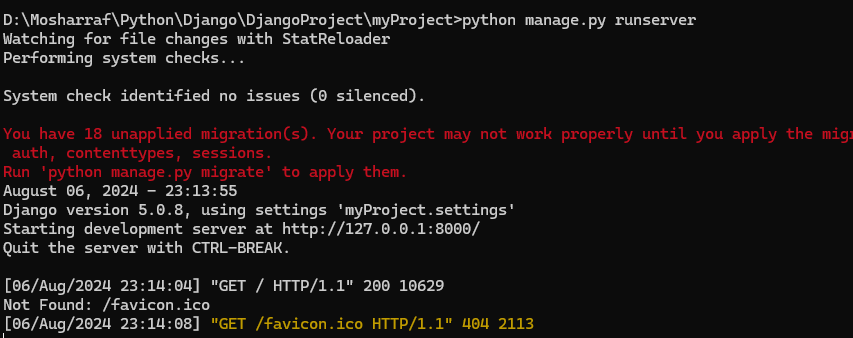
if not found django-admin, please go to installation folder

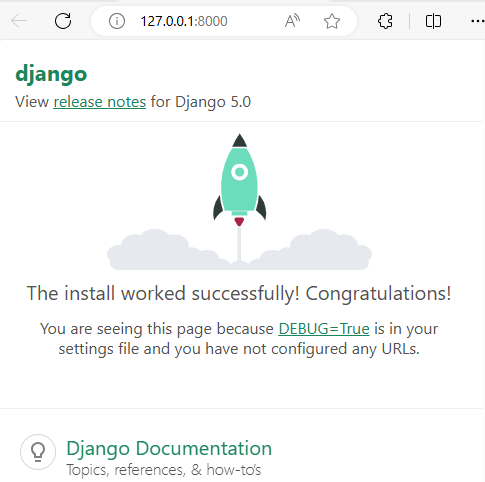
run django-admin



django-admin startproject myProject

python manage.py runserver

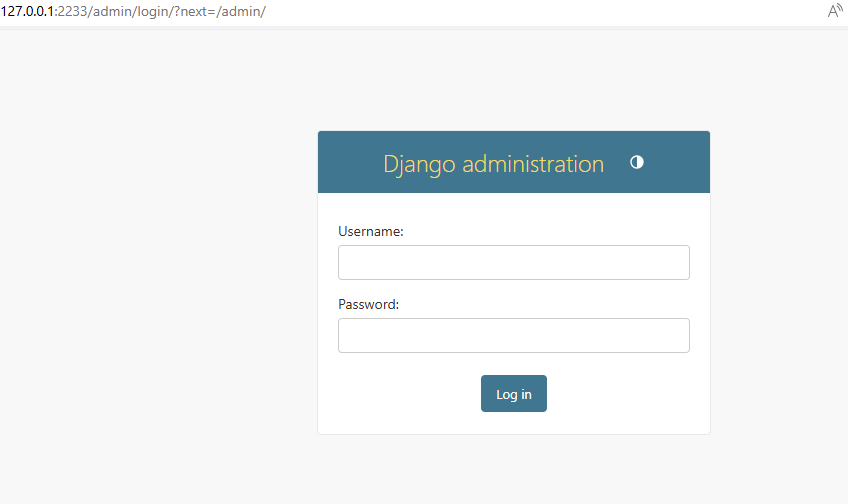


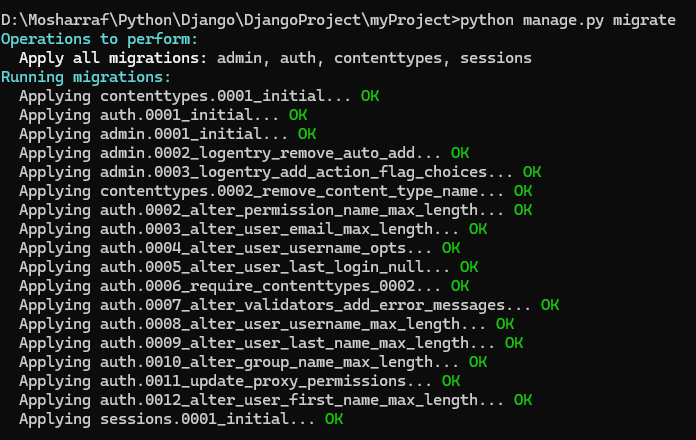


If you change port number, Run following code

python manage.py runserver 2233

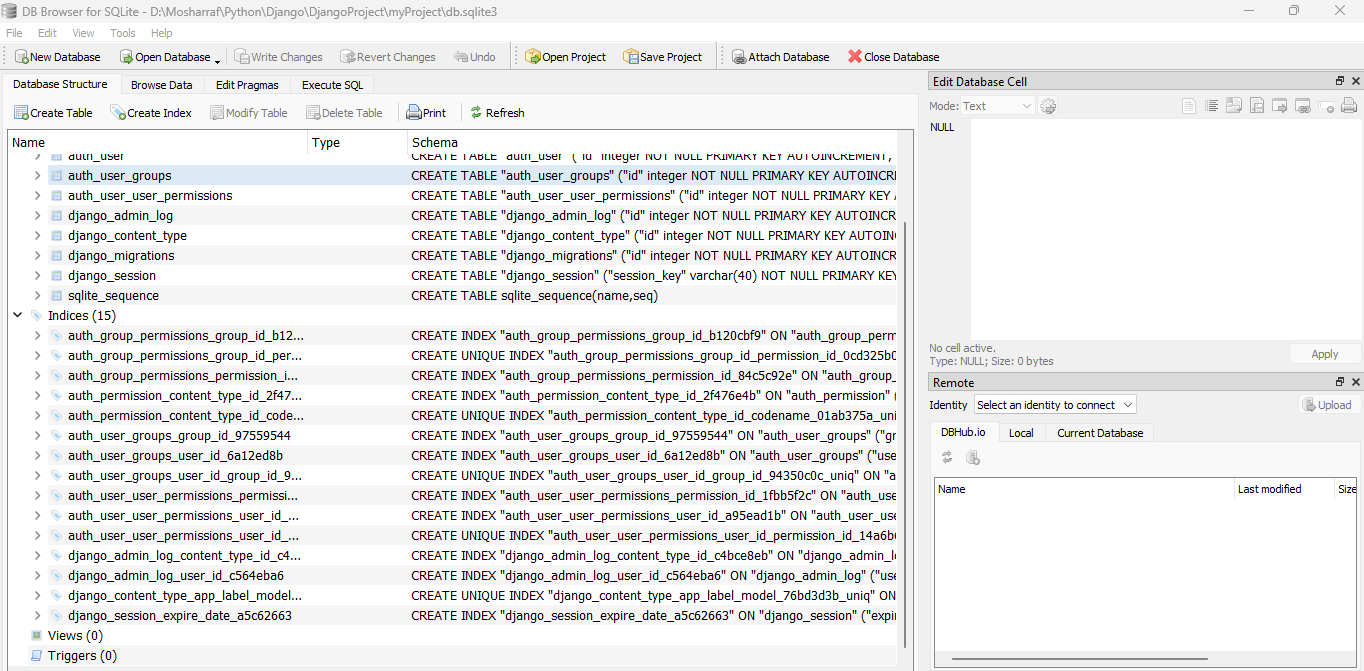
Admin panel: <http://127.0.0.1:2233/admin>

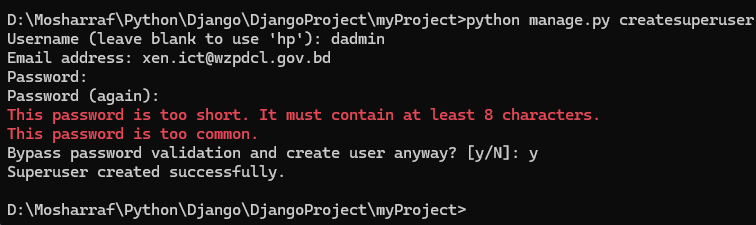


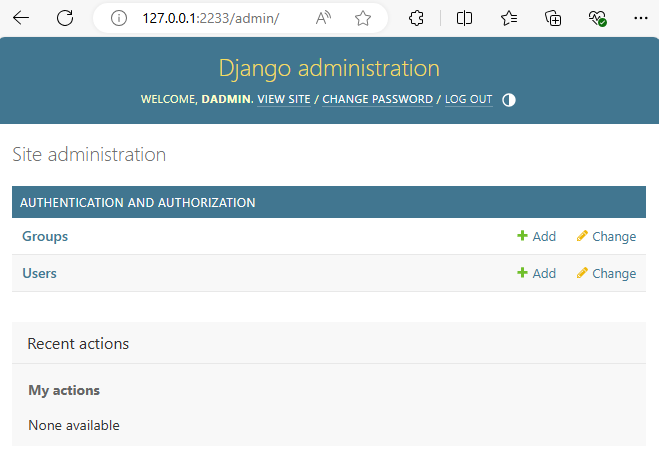


Now Download SQLite DB Browser

Run and Open SQLite Database as follows







https://www.youtube.com/watch?v=ZtiPjEarJyU