Setting Up the Django Web Server

CMPSC 305 – Database Systems

ALLEGHENY COLLEGE

Databases Online

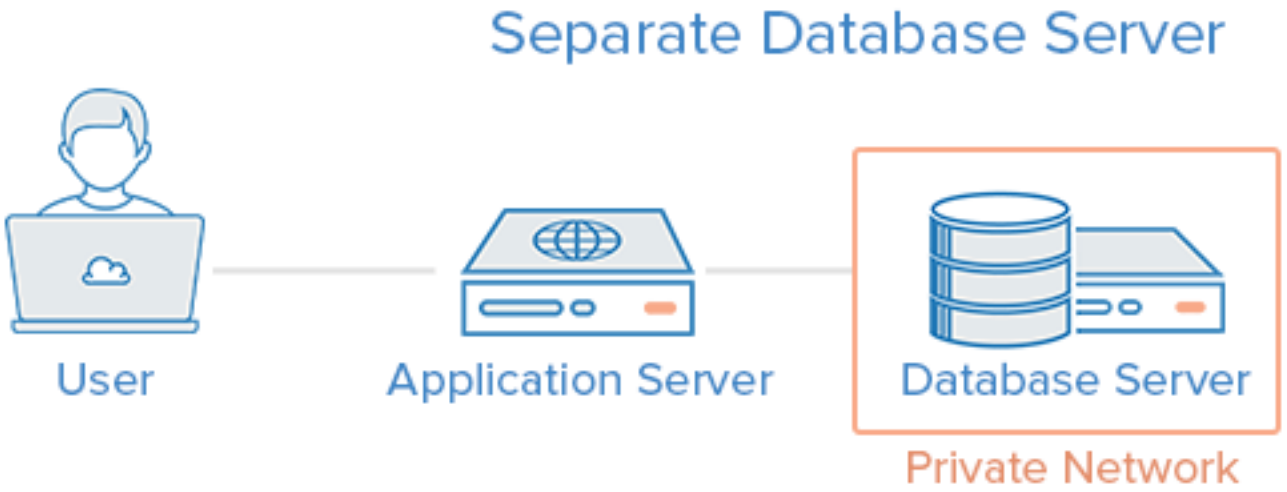
People use online data

- Purchasing: www.amazon.com
- Entertainment: www.netflix.com
- Social media: www.facebook.com
- Research: www.uniprot.org
- Weather: www.accuweather.com

And many other forms of information are online!

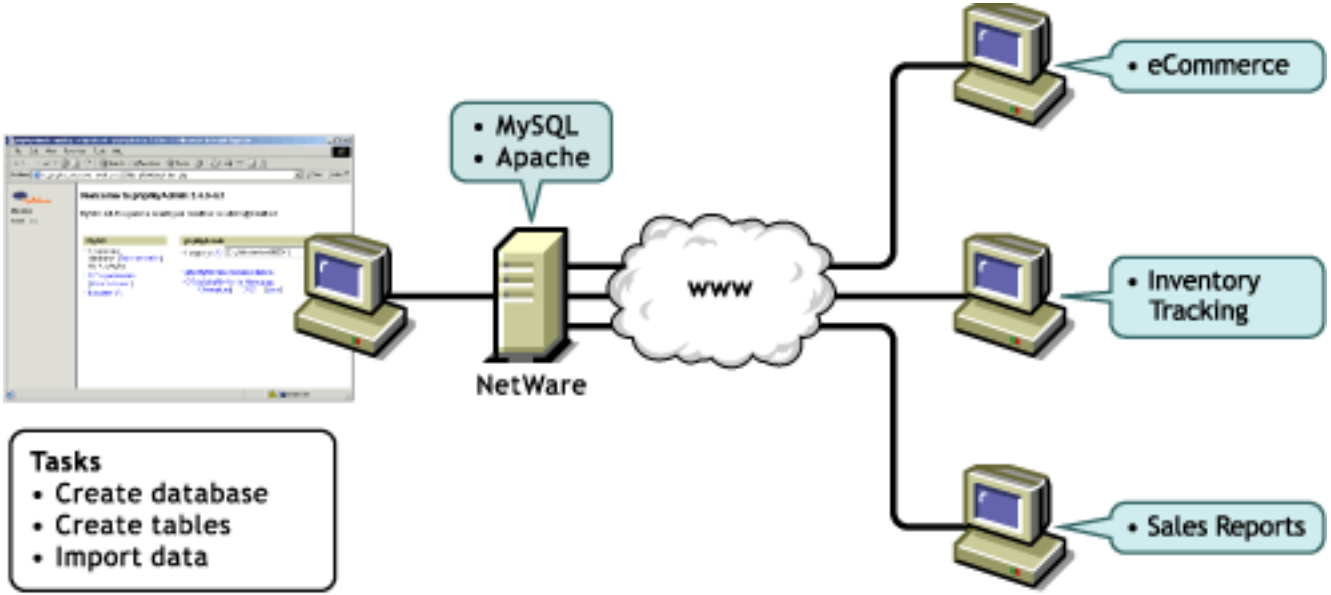
Where is the Database?

Humans indirectly use them

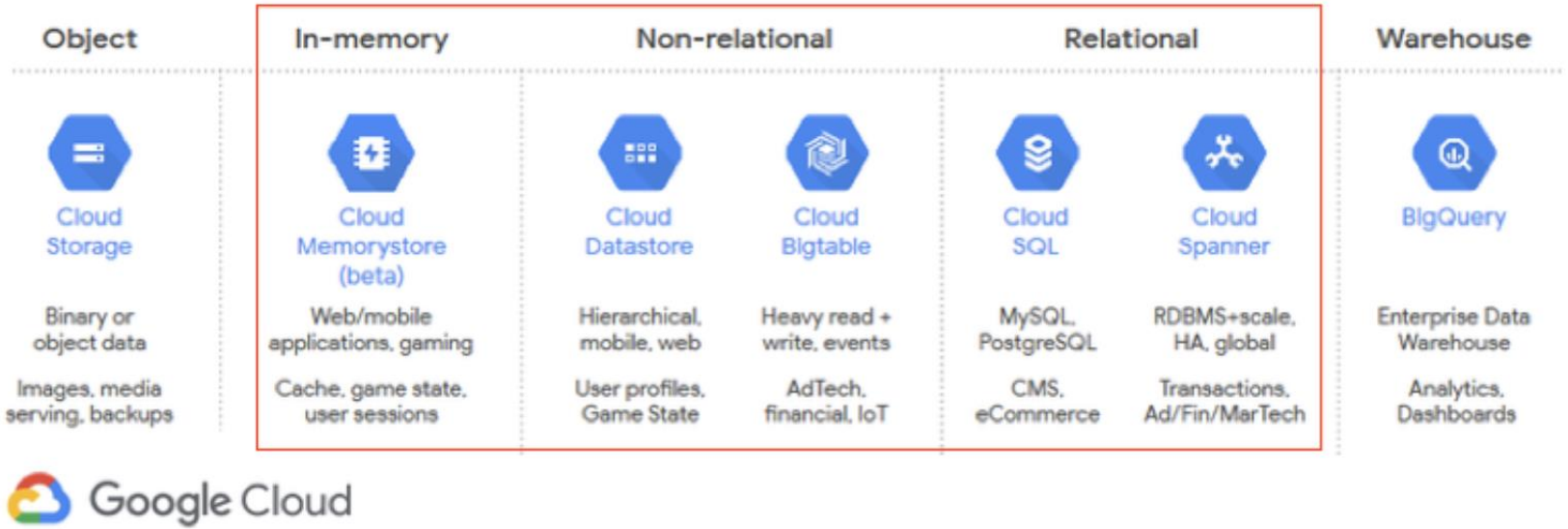


Where is the Database?

Computers use them



Cloud Resources



- Cloud technologies rely on databases

Django

An easy-to-create web site and online database server

- https://www.djangoproject.com/

Django: web server

Easy to build better Web apps more quickly and with less code.

- Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design.
- Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel.
- It's free and open source.

Django: web server

Easy to build better Web apps more quickly and with less code.

- Designed for rapid development of applications with quick concept to completion.

Includes dozens of extras:

- common Web development tasks
- user authentication
- content administration
- site maps
- RSS feeds
- DATABASE SUPPORT (more about that later)

Who Uses Django?

- Companies, organizations and governments
- Content management systems
- Social networks
- Scientific computing platforms ... and more!
- Specifically...
- https:// www.google.com/
- https://www.youtube.com/
- https://www.instagram.com/
- https://www.spotify.com/
- https://www.washingtonpost.com/ and many others!

TODO, Today

- We are going to install a virtual environment, from which to work
- Then install Django and setup an initial-state demo.

Pip: Used to install project software packages

- The latest versions of Python (version 3) come with pip
- To see if pip is already installed on your machine, type pip3 help
- If pip is not already installed, try upgrading your Python to get a newer version.
- Alternatively, install pip yourself from source: https://bootstrap.pypa.io/get-pip.py

Steps to set up a project

Steps

- Get into a working directory.
- Setup a virtual environment in which to work
- Activate the virtual environment
- Use pip/pip3 to install Django to build project
- Use Django to build a project
- Run the project

Setting Up Virtual Environment

- Create a project directory

```
mkdir week10
cd week10
```

- Create virtual environment using Python

```
python3 -m venv myenv
# see the file tree
find . -not -path '*/\.*'
```

- Activate myenv the virtual environment

```
source myenv/bin/activate  # macOS/Linux
myenv\Scripts\activate     # Windows
```

- Deactivate the virtual environment

```
deactivate
```

- Install the Streamlit software packages in the environment

```
pip3 install django
```

Setting-up Django

Your terminal should now say, (myenv)

- Install Django in your virtualenv

```
python -m django --version # check version
#or, python3 -m django --version # check version
```

- Create your first Django project!

```
• django-admin startproject mysite
```

- Use manage.py to run the webserver to se

```
cd mysite/
# we are now in: djangoWorking/myenv/mysite
python manage.py runserver
# or, python3 manage.py runserver
```

Setting-up Django

Your terminal should now say, (myenv)

- Use your browser to check your work

```
http://127.0.0.1:8000/
Control-c to exit
```

The Files of Your Project

Unix OS

```
find . -not -path '*/\.*'
-------------------------
./mysite
./mysite/manage.py
./mysite/mysite
./mysite/mysite/__init__.py
./mysite/mysite/settings.py
./mysite/mysite/urls.py
./mysite/mysite/wsgi.py
```

- Notable Files
  - manage.py: used to run the server
  - settings.py: configuration file
  - urls.py: used to connect url addresses to pages
  - wsgi.py: web server gateway interface utils

The Output From manage.py

```
You have 18 unapplied migration(s). Your project may not work properly until
you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.

October 28, 2025 - 15:06:05

Django version 5.2.7, using settings 'mysite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

Yes! It Worked!

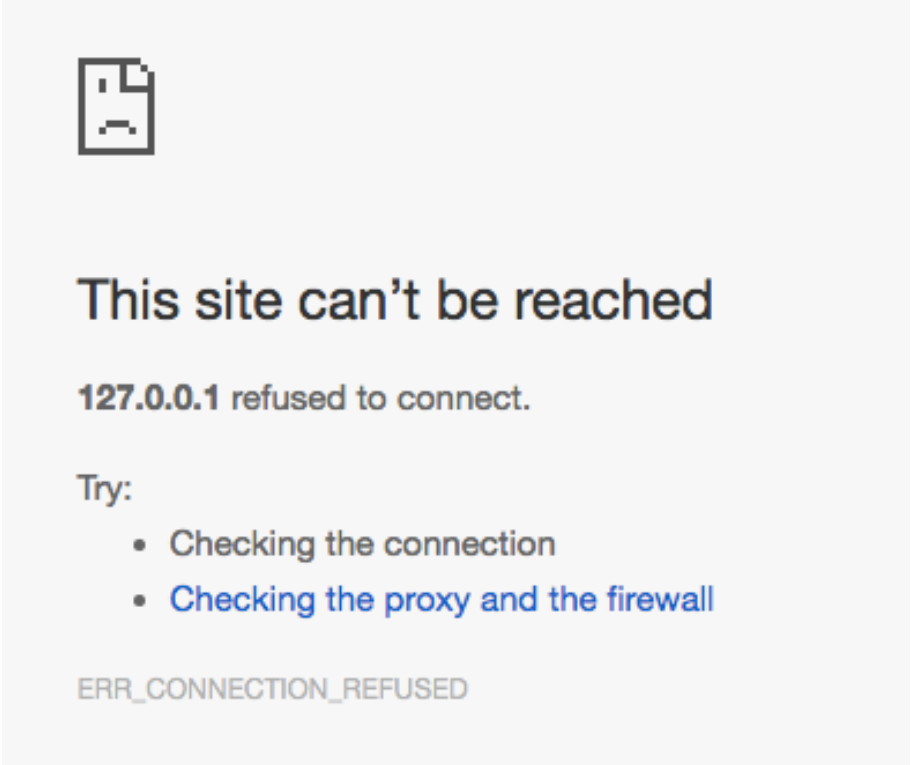
The install worked successfully! Congratulations!

View release notes for Django 5.2

You are seeing this page because DEBUG=True is in your settings file and you have not configured any URLs.

django

Rats! It Did Not Work!



- See: https://docs.djangoproject.com/en/3.1/ or
- https://www.djangoproject.com/start/

Create a Super User to View Database

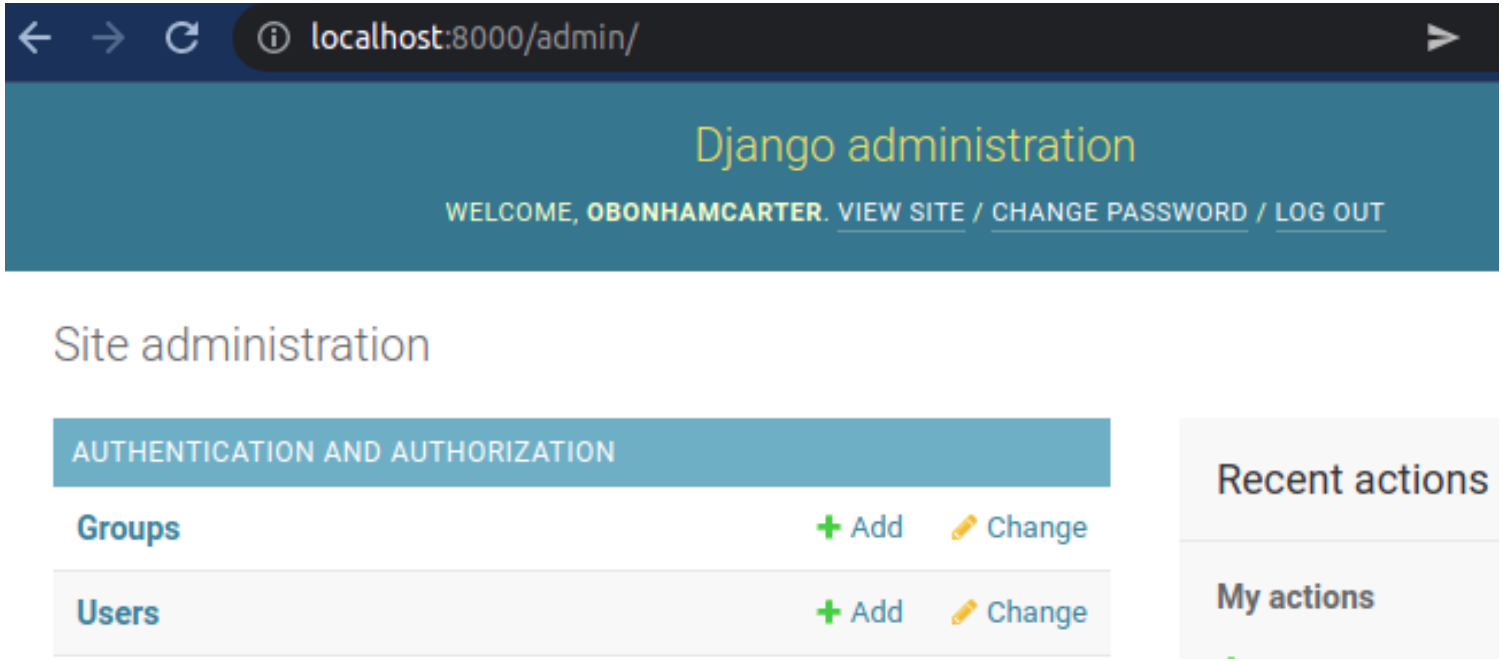
- Need to make a user for the site.

```
python3 manage.py migrate
python3 manage.py createsuperuser
```

- Migrate is to connect a database to the project to hold user (admin) data
- Username (leave blank to use 'user'): admin
- Email address: studentID@allegheny.edu
- Password: "pass1234"
- Password (again): "pass1234"
- Superuser created successfully.
- Now, look around the admin page, http://127.0.0.1:8000/admin

Admin

- Let's take a look at the "stuff" that Django built to manage the website!
- Visit the Admin controls at http://localhost:8000/admin/
- Then visit the File mysite/db.sqlite3 to check the database.



Django page

```
from django.contrib import admin
from django.urls import path
from django.http import HttpResponse

# Simple function that returns text to the browser
def home(request):
        return HttpResponse("<h1>Welcome to My Blog!</h1><p>This is your first
Django page 🍕</p>")

urlpatterns = [
        path('admin/', admin.site.urls),
        path('', home), # Root URL (http://127.0.0.1:8000/)
]
```

What could you do with a DB on a website!?!