Steps:

1. Create Database using Terminal using psql for Postgres
2. Enter Database credentials in settings.py
3. Create App in Django - python manage.py startapp blog (blog is name of app)
4. In the models.py file – create the data model –
5. In Admin.py import this model
6. In settings.py – insert the model in the app in INSTALLED APPS
7. Make Migration – python manage.py makemigrations –(prepares models for migration)
8. Migrate – python manage.py migrate – this creates the table for the data structure defined in the models.py file (data model)
9. Now in the web page (admin page) – we can input the image and it gets saved to the Database.

Steps to get Data from Database:

1. In views.py of the Blogs App, import Blog 🡪 from .models import Blog
2. Make a variable blogs = Blog.objects => gets blogs from postgres
3. render the blog objects in an html and pass blogs variable. This will pass blogs to html.

def allblogs(request):

    blogs = Blog.objects

    return render(request, 'blog/allblogs.html',{'blogs': blogs})

1. In the HTML make a for loop

 {% for blog in blogs.all%}

Going to a single URL:

1. In Blogs app, make urls.py

from django.urls import path

from .import views

urlpatterns = [

    path('',views.allblogs ,name='allblogs'),

    path('<int:blog\_id>/',views.detail,name='detail')

]

1. Here we have to pass blog id in the URL and so we are passing the int as blog id.
2. Each blog will have blog id in the database. We have defined detail function in views.py which will return the detail html for each blog id

def detail(request, blog\_id):

    detailblog = get\_object\_or\_404(Blog, pk = blog\_id)

    return render(request, 'blog/detail.html',{'blog': detailblog})

1. In all Blog html, we should give the link of blog id along with detail html

<a href="{%url 'detail' blog.id %}"><h3>{{blog.title}}</h3></a>