***DJANGO***

***Create a project***

**django-admin startproject mysite**

***run server***

**python manage.py runserver**

***directory structure***

mysite/

manage.py

mysite/

\_\_init\_\_.py // considered / treated as python packages

settings.py // all information about project settings

urls.py // url routing (url patterns)

asgi.py // ashynchronus server interface interface between webserver and web application (synchronous and astnchronus)

wsgi.py // web server getway interface . interface between webserver and web application (For only synchronus)

**Function views**

    1. Add an import:  from my\_app import views

    2. Add a URL to urlpatterns:  path('', views.home, name='home')

url.py

from django.contrib import admin

from django.urls import path

from . import views

urlpatterns = [

    path('admin/', admin.site.urls),

    path('', views.index, *name*='index'),

    path('about', views.about, *name*='about'),

]

Views.py

from django.http import HttpResponse

*def* index(*request*):

    return HttpResponse("hello world")

*def* about(*request*):

    return HttpResponse("hello bout")

***ADDING HTML TEMPLATES :-***

Create a templates folder

Setings.py > 'DIRS': [‘templates’]

Views.py > from django.shortcuts import render

*def* index(*request*):

    return render(request,'index.html')

***Send variables in templates :-***

*views.py*

*def* index(*request*):

    params = {'name':'spandan','title':'joshi'}

    return render(request,'index.html',params)

index.html

<h1>welcome {{name}} {{title}}</h1>

***GET and POST Request :-***

djtext = request.GET.get('text','default') // text is class of input field

djtext = request.POST.get('text','default')

***Projects VS Apps :-***

A project refers to the entire application and all its parts

Django-admin startproject projectname //create project

An app refers to a submodule of the project . pluggable web app

Py manage.py startapp appname

***Working with apps & projects viewsrouting :-***

Project file

Including another URLconf

    1. Import the include() function: from django.urls import include, path

    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))

Register App in Setinngs.py

INSTALLED\_APPS = [

    'student.apps.StudentConfig', // student is my app name

]

Created app / student folder

Student > urls.py

from django.urls import path

from . import views

urlpatterns = [

    path('', views.home, *name*='home'),

    path('student/<str:name>', views.student, *name*='student')

]

Student > views.py

from django.shortcuts import render

from django.http import HttpResponse

# Create your views here.

*def* home(*request*):

    return render(request , 'index.html')

*def* student(*request*,*name*):

    return HttpResponse("student "+ name)

***Adding Parameters in URL :-***

<type:name>

Type = str,int,uuid,slug

Name must specify in views

*def* student(*request*,*name*):

    return HttpResponse("student "+ name)

urlpatterns = [

path('articles/2003/', views.special\_case\_2003),

path('articles/<int:year>/', views.year\_archive),

path('articles/<int:year>/<int:month>/', views.month\_archive),

path('articles/<int:year>/<int:month>/<slug:slug>/', views.article\_detail),

]

***Django Templates:-***

// Built in Django templates

'BACKEND': 'django.template.backends.django.DjangoTemplates',

// Adding JinJa template

'django.template.backends.jinja2.jinja2'

// Adding templates directory

'DIRS': ['templates'],

Templating engine

{{ }}

{% for %} {% endfor %}

{% if %} {% endif %}

***Removing Hardcoded URL :-***

**{% url %}**  template tag used for removing hardcoded url

*the 'name' value as called by the {% url %} template tag*

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

<**li**><**a** href="{% **url** 'detail' question.id %}">{{ question.question\_text }}</**a**></**li**>

***Django Template Inheritance :-***

We can inharit template in 2 way

Using Extand & include

***Extending Templates :-***

Header.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <h1>Header</h1>

    {% block content %}

    {% endblock %}

Index.html

{% extends 'components/header.html' %}

{% block content %}

    <h1>welcome to my blog</h1>

    {% for i in data %}

    <li>{{i}}</li>

    {% endfor %}

{% endblock %}

</body>

</html>

***Include Templates :-***

Footer.html

<h1>Footer</h1>

</body>

</html>

Index.html

{% include 'components/footer.html' %}

***Working with Django Admin :-***

First migrate the auth model

**Py manage.py migrate**

**Py manage.py createsuperuser**

**NOTE - Password not visible so keep typing yor password**

Username (leave blank to use 'spandan'):

Email address: spandanj685@gmail.com

Password:

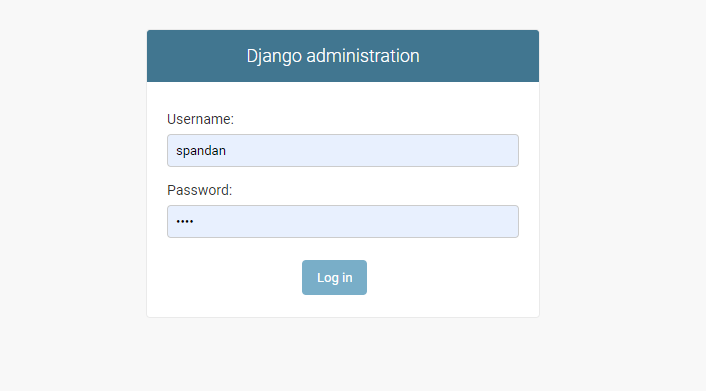
Password (again):

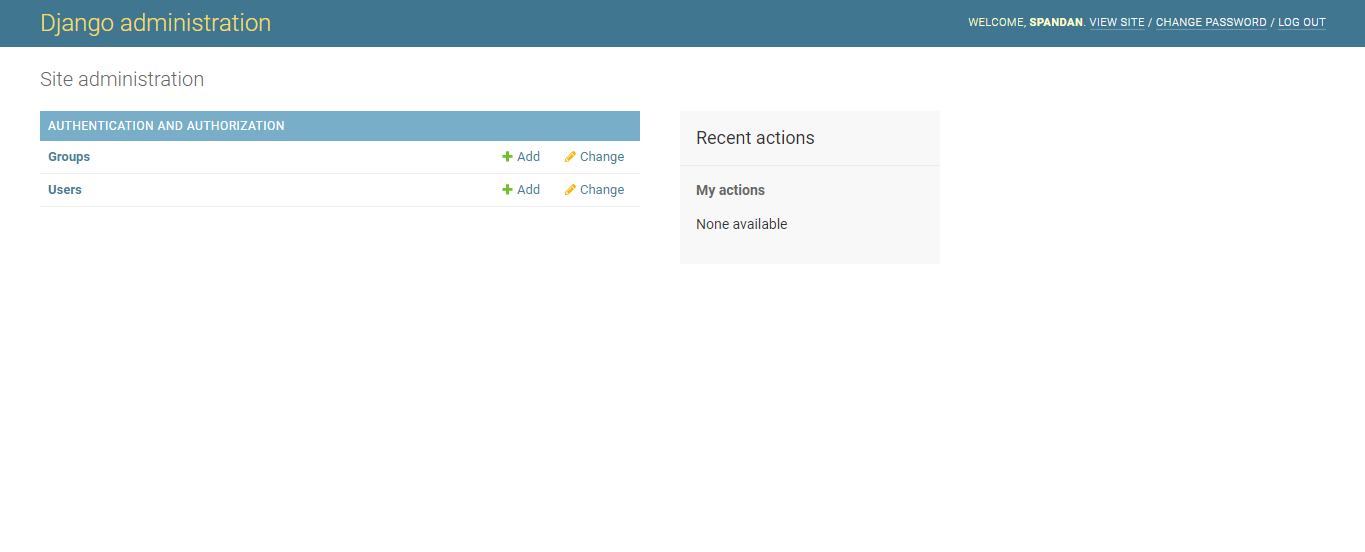
The password is too similar to the username.

Bypass password validation and create user anyway? [y/N]: y

Superuser created successfully

[**http://127.0.0.1:8000/admin/**](http://127.0.0.1:8000/admin/)

****

****

***Working with Django Models :-***

Models means blueprint of database.

Django Default database engine is **sqlite3** you can change it easily by following documentation

**Rules for creating a model**

1. Model name must be singular ( a ) User > Users
2. Model name must start with capital latter
3. Models inherit (models.Model)

from django.db import models

# Create your models here.

*class* Student(*models*.*Model*):

    name = models.CharField(("Student Name"), *max\_length*=50 , *blank*=False , *null*=False)

    age = models.IntegerField(("Student Age"),*null*=False)

    address = models.TextField(("Student Address"))

**py manage.py makemigrations student** // studebt is my app name

make migration create a schema

**py manage.py migrate** // create table in database

*def* \_\_str\_\_(*self*):

        return self.name

admin.py

from employee\_register.models import \*

# Register your models here.

admin.site.register(Position)

admin.site.register(Employee)

The [**migrate**](https://docs.djangoproject.com/en/3.0/ref/django-admin/#django-admin-migrate) command looks at the [**INSTALLED\_APPS**](https://docs.djangoproject.com/en/3.0/ref/settings/#std:setting-INSTALLED_APPS) setting and creates any necessary database tables according to the database settings in your **mysite/settings.py** file and the database migrations shipped with the app

**DJANGO MODEL RELATION :-**

***Working with Django ORM :-***

***Learn Django orm using shell :-***

**py manage.py shell**

**from Students.models import \***

**METHOD 1 FOR INSERT**

**object = Student()**

**>>> object = Student()**

**>>> object.name = "spandan joshi"**

**>>> object.age = 18**

**>>> object.address = "somthing"**

**>>> bject.save()**

**METHOD 2 FOR INSERT**

**>>> obj = Student(name="aman", age="18",address="gjhgjesg")**

**>>> obj.save()**

**FILTER :-**

**Student.objects.filter(age=18)**

**\_\_startswith=” ”**

**\_\_endswith=” “**

**\_\_contains=” ”**

**\_\_istartswith=” ” // I stands for case insensitive**

**\_\_exact = “ ” // Exact match**

**\_\_contains = “ “ // Like**

Entry.objects.get(headline\_\_contains='Lennon')

**SELECT** ... **WHERE** headline **LIKE** '%Lennon%';

**\_\_gt , \_\_lt , \_\_gte , \_\_lte**

**FETCH :-**

**>>> Blog.objects.all( )**

**>>> Blog.objects.get(name="food")**

**>>> Blog.objects.all().exclude(name="food")**

**UPDATE :-**

**Student.objects.filter(id=18).update(name='spandan')**

**DELETE :-**

**>>> Student.objects.filter(id=1).delete()**

**LIMITING QUERY SET :-**

**Entry.objects.all()[:5]**

**Entry.objects.all()[5:10]**

**>>> Entry.objects.order\_by('headline')[0:1].get()**

**FOR MORE INFO =** <https://docs.djangoproject.com/en/3.0/topics/db/queries/>

**LOGIN AUTHONTICATION**

**Get data from form :-**