Python Django – ToDo App

**Create new Project as Name ToDo-**

Django-admin startproject ToDo

Cd ToDo/

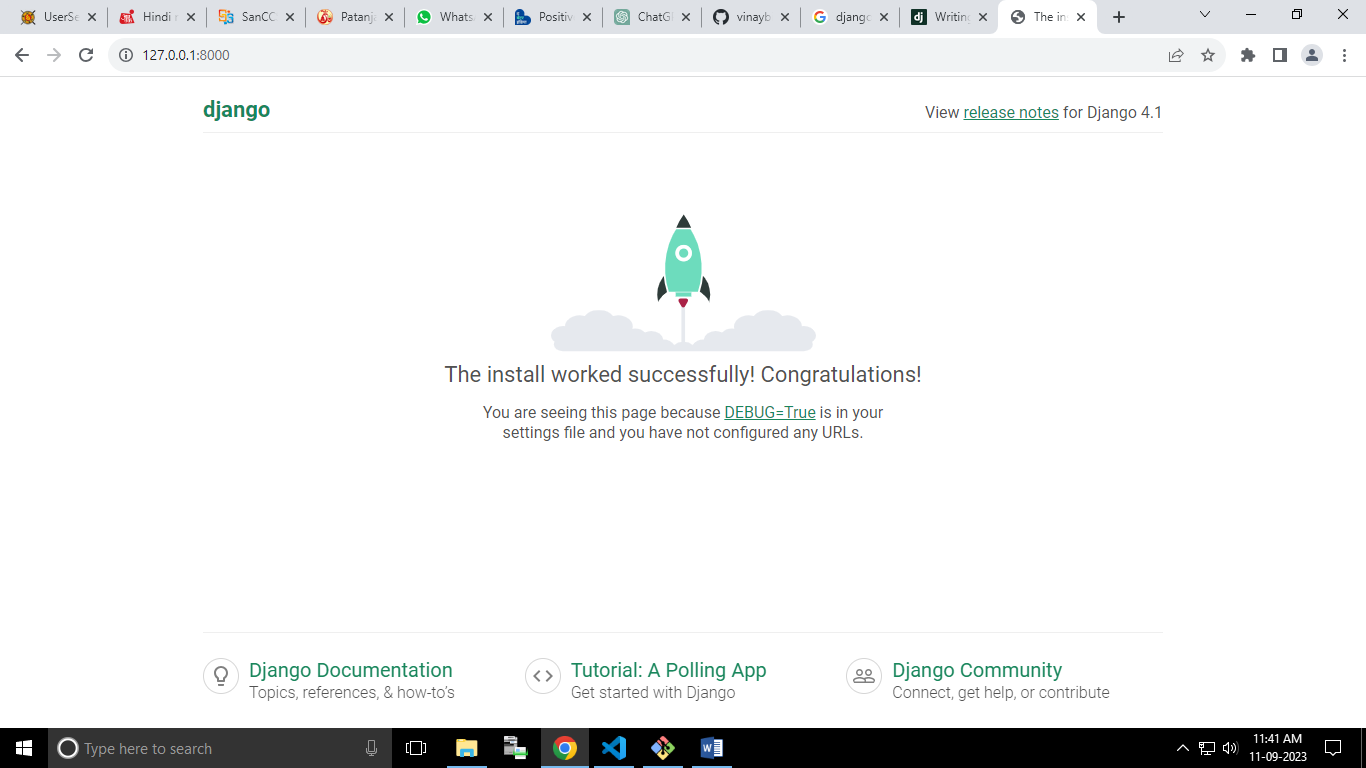
ls –a

now create new app inside main project

python manage.py startapp base

Run sever

Python manage.py runserver



Add this base app in settings.py file

INSTALLED\_APPS = [

    'django.contrib.admin',

    'django.contrib.auth',

    'django.contrib.contenttypes',

    'django.contrib.sessions',

    'django.contrib.messages',

    'django.contrib.staticfiles',

    'base',

]

Now Update url path in project url file.

This is

from django.contrib import admin

from django.urls import path

urlpatterns = [

    path('admin/', admin.site.urls),

]

Now update this one, add base url in project url

from django.contrib import admin

from django.urls import path, include

urlpatterns = [

    path('admin/', admin.site.urls),

    path('', include('base.urls')),

]

Before creating app urls.py file update views.py file and before update views.py file

Create folder structure for templates

Under app dir

Create Templates Folder under App dir

Base 🡪 templates 🡪 base 🡪 home.html

Now after that update views.py file

Creating new function name homepage

from django.shortcuts import render

# Create your views here.

def homepage(request):

    return render(request, 'base/home.html')

after that update app urls.py file

from django.urls import path

from . import views

urlpatterns = [

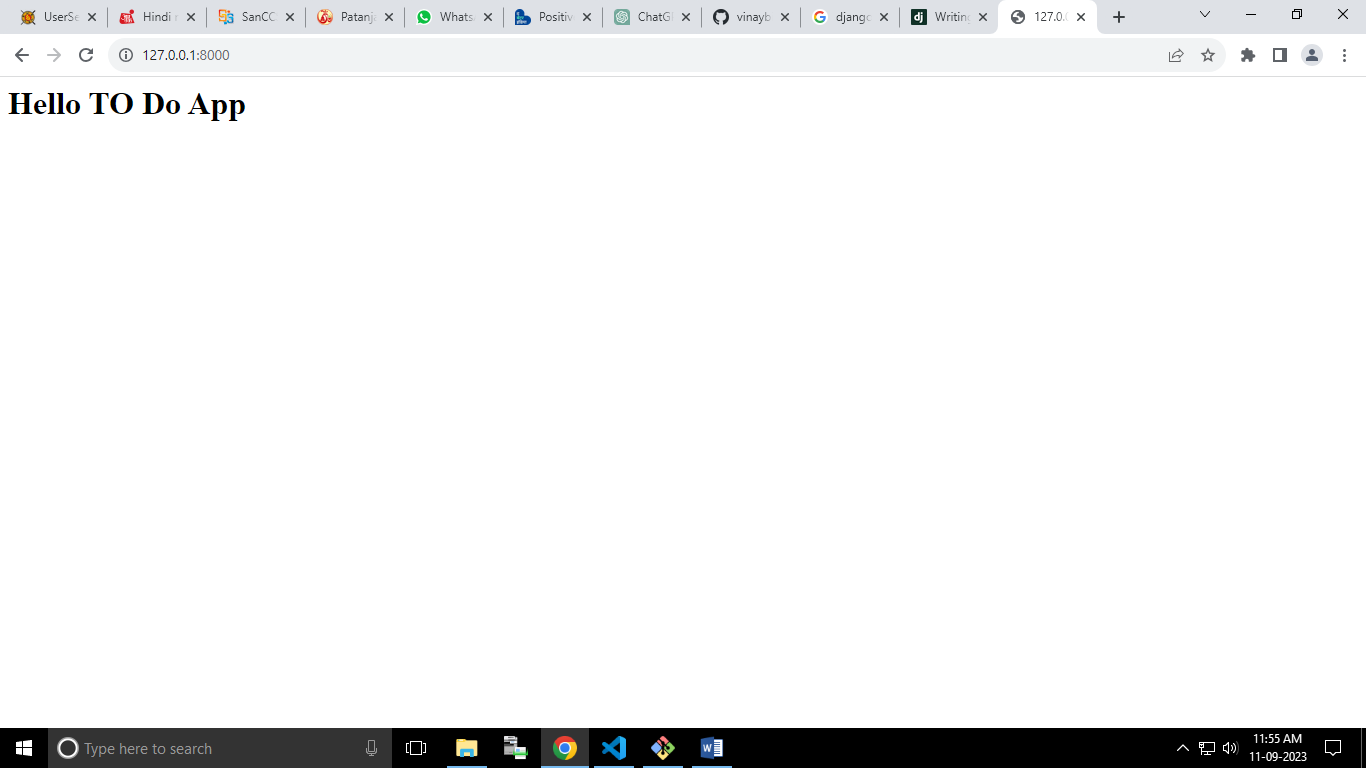
    path('', views.homepage, name='home'),

]

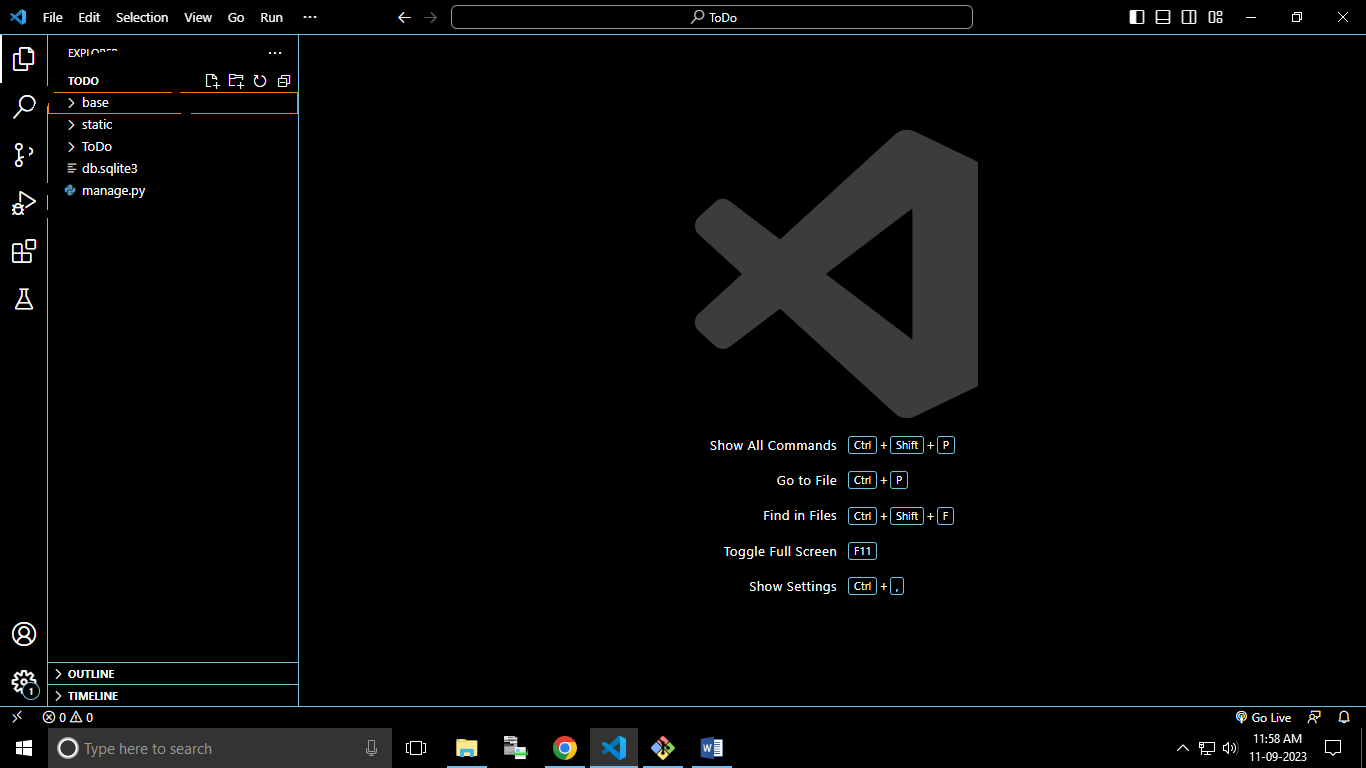
Now update home.html file

<h1>Hello TO Do App</h1>

Now check the page



Now create new folder name static in under project dir



Under static folder create folder

Images

Css

Js

After that configuration static folder in settings.py file

STATICFILES\_DIRS **=** [BASE\_DIR **/** "static"]

Now in templates start with

{% load static %}

<!DOCTYPE html>

<html>

<head>

  <title>Learn Django</title>

  <link rel="stylesheet" href="{% static 'css/base.css' %}">

</head>

    <body>

        <h1 class="todo">This is New To Do App</h1>

    </body>

</html>

And update css file

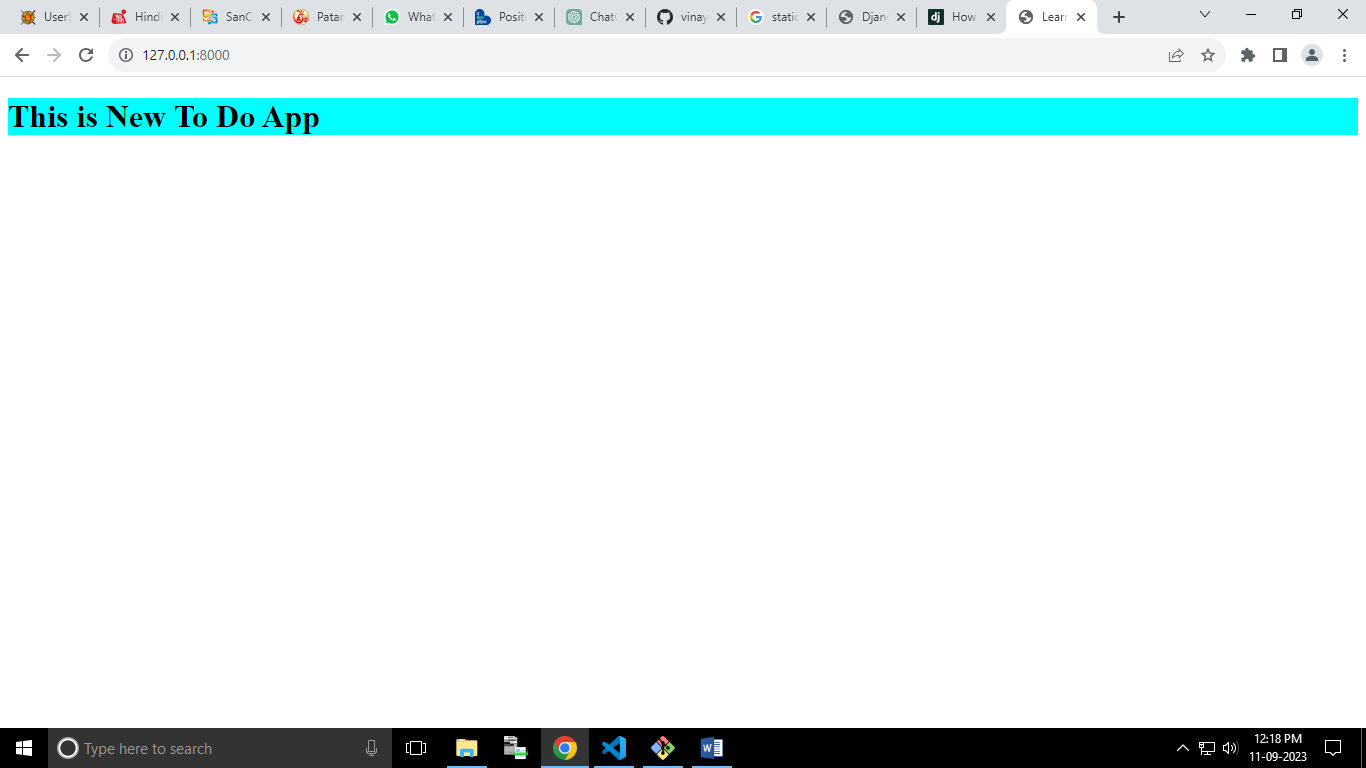
.todo{

    background-color: aqua;

}

For checking that css fle work or not

Well-done here is working css file



All the basic work completed

Now working on backend

Create new filed in database table in models.py file

from django.db import models

# Create your models here.

class ToDo(models.Model):

    text = models.TextField(max\_length=500)

    completed = models.BooleanField(default=False)

    created\_date = models.DateTimeField(auto\_now\_add=True)

    def \_\_str\_\_(self):

          return self.created\_date

After create database table migrate all table in default sqlite database.

vinay\_bijalwan@DESKTOP-00IQAHH MINGW64 /f/develpment\_work/ToDo

$ python manage.py makemigrations

Migrations for 'base':

base\migrations\0001\_initial.py

- Create model ToDo

vinay\_bijalwan@DESKTOP-00IQAHH MINGW64 /f/develpment\_work/ToDo

$ python manage.py migrate

Operations to perform:

Apply all migrations: admin, auth, base, 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 base.0001\_initial... OK

Applying sessions.0001\_initial... OK

Create super login for admin login

PS F:\develpment\_work\ToDo> python .\manage.py createsuperuser

Username (leave blank to use 'vinay\_bijalwan'): vinay

Email address: bijalwan.vinay@gmail.com

Password:

Password (again):

Superuser created successfully.

PS F:\develpment\_work\ToDo>

User login – vinay

Paswd – Vinay@1991

Showing the miodels table in Admin section

Update in admin.py file

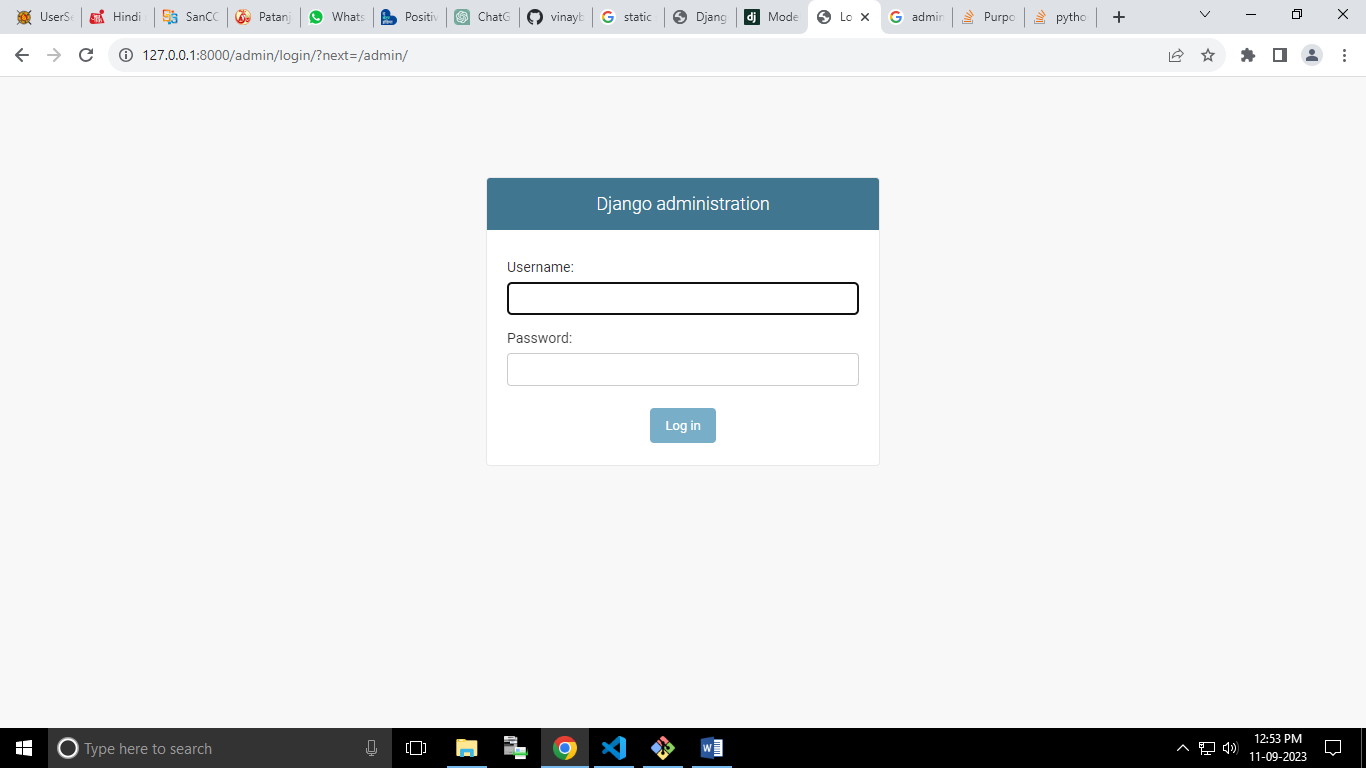
from django.contrib import admin

from .models import ToDo

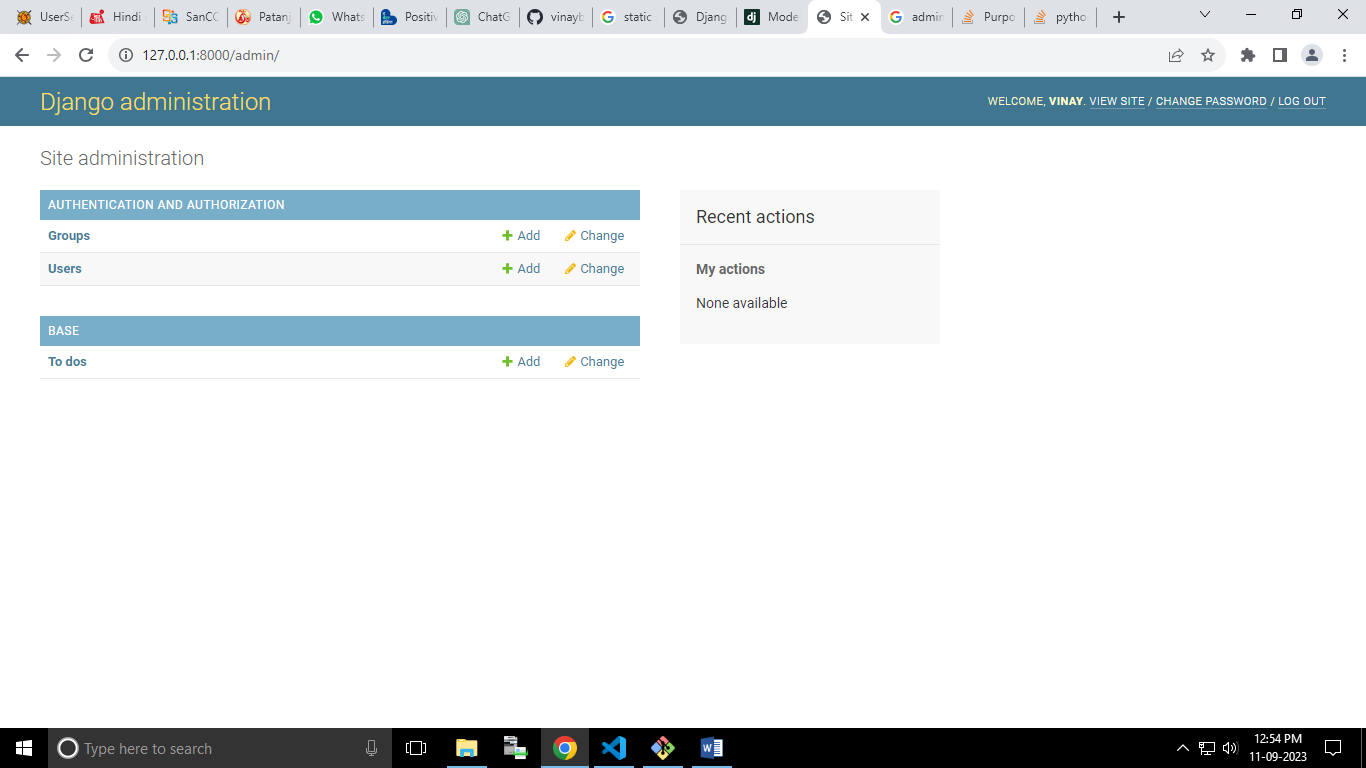
# Register your models here.

admin.site.register(ToDo)

now login admin



Login as Vinay



Now submit data from html form

We use ModelForm

Create new python file name as forms.py

from django.forms import ModelForm

from .models import ToDo

class ToDoForm(ModelForm):

    class Meta:

        model = ToDo

        fields = '\_\_all\_\_'

After that update views.py file for logic to submit data to database

def homepage(request):

    if request.method == "POST":

        form = ToDoForm(request.POST)

        if form.is\_valid():

            form.save()

            # Redirect to a different page after saving the form

            return redirect('home')

    else:

        form = ToDoForm()

    return render(request, 'base/home.html', {'form': form})

all done these things finally we do update in HTML file

{% load static %}

<!DOCTYPE html>

<html>

<head>

  <title>Learn Django</title>

  <link rel="stylesheet" href="{% static 'css/base.css' %}">

</head>

    <body>

        <h1 class="todo">This is New To Do App</h1>

        <div class="container">

          <form method="post">

            {% csrf\_token %}  <!-- Include the CSRF token for security -->

            {{ form.as\_p }}    <!-- Render the form fields as paragraphs -->

            <button type="submit">Submit</button>

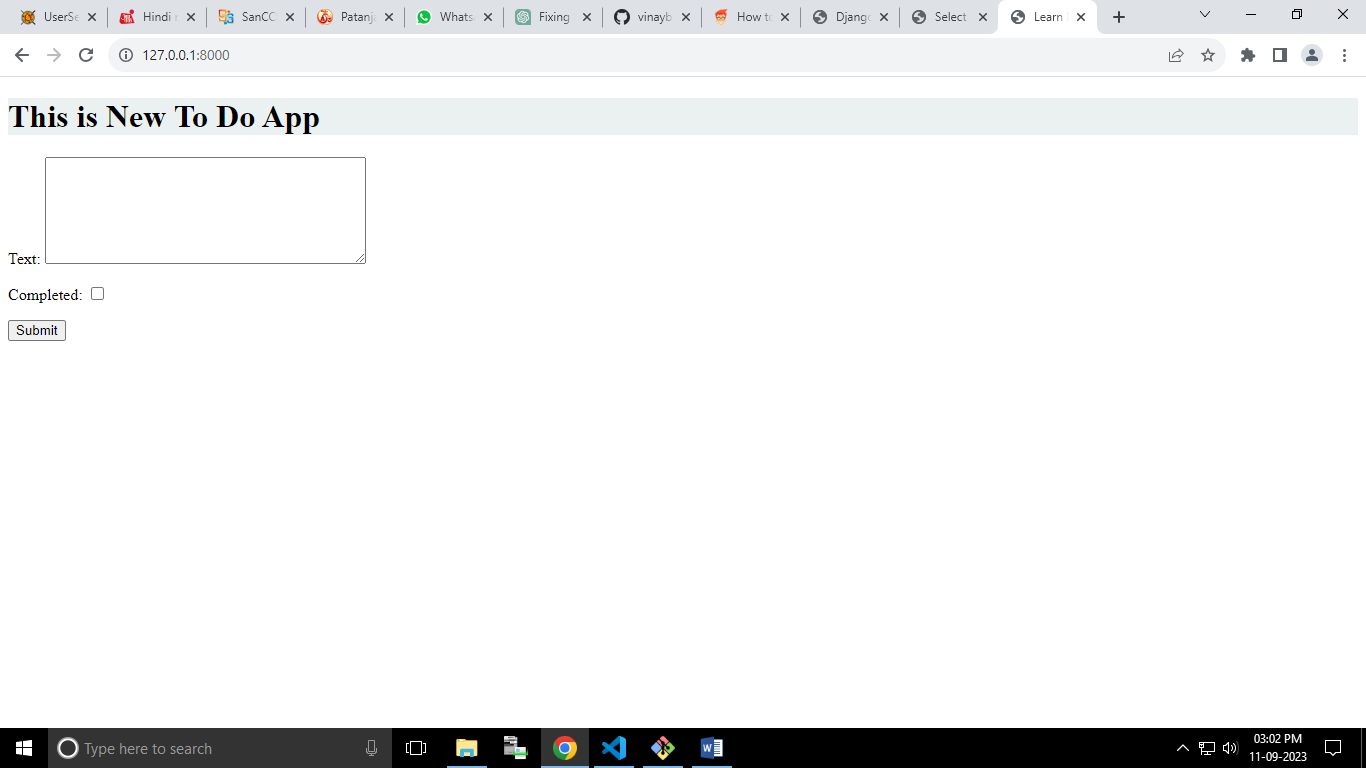
        </form>

      </div>

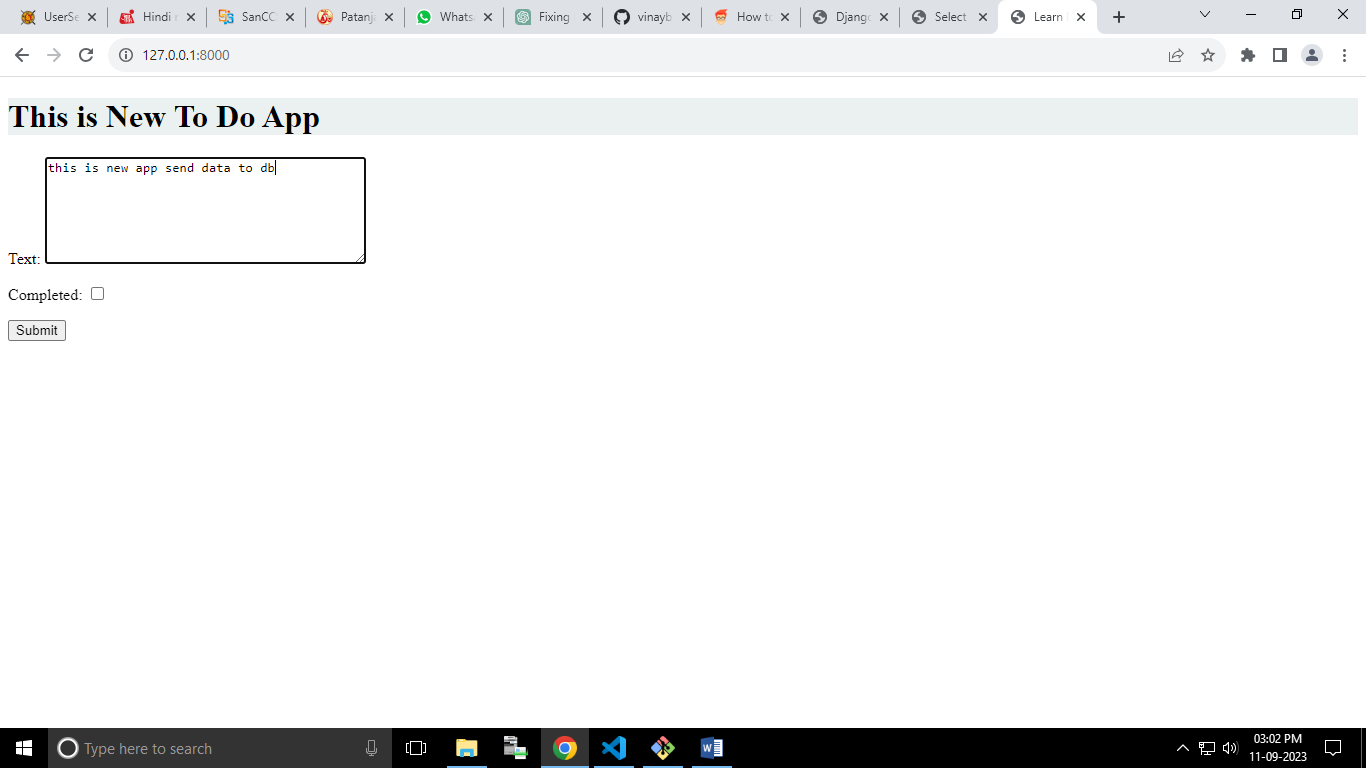
    </body>

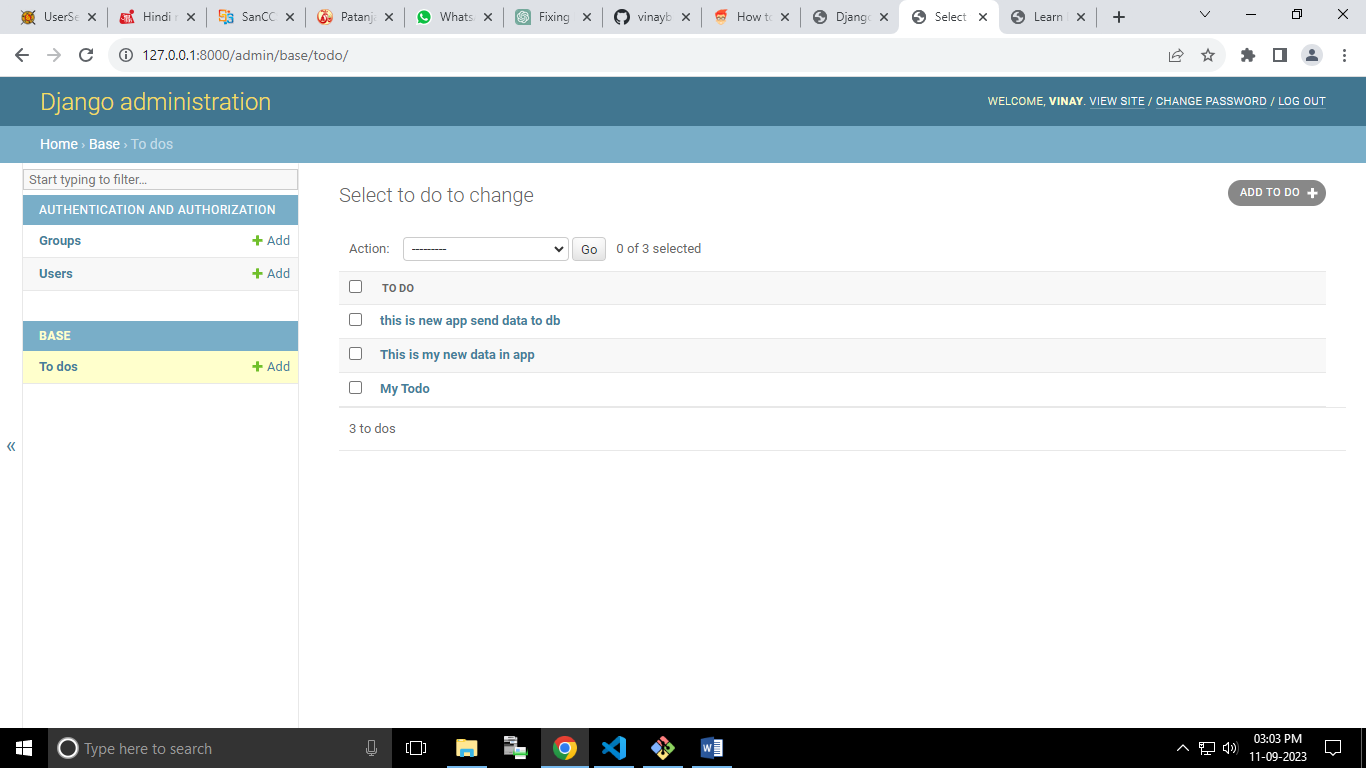
</html>

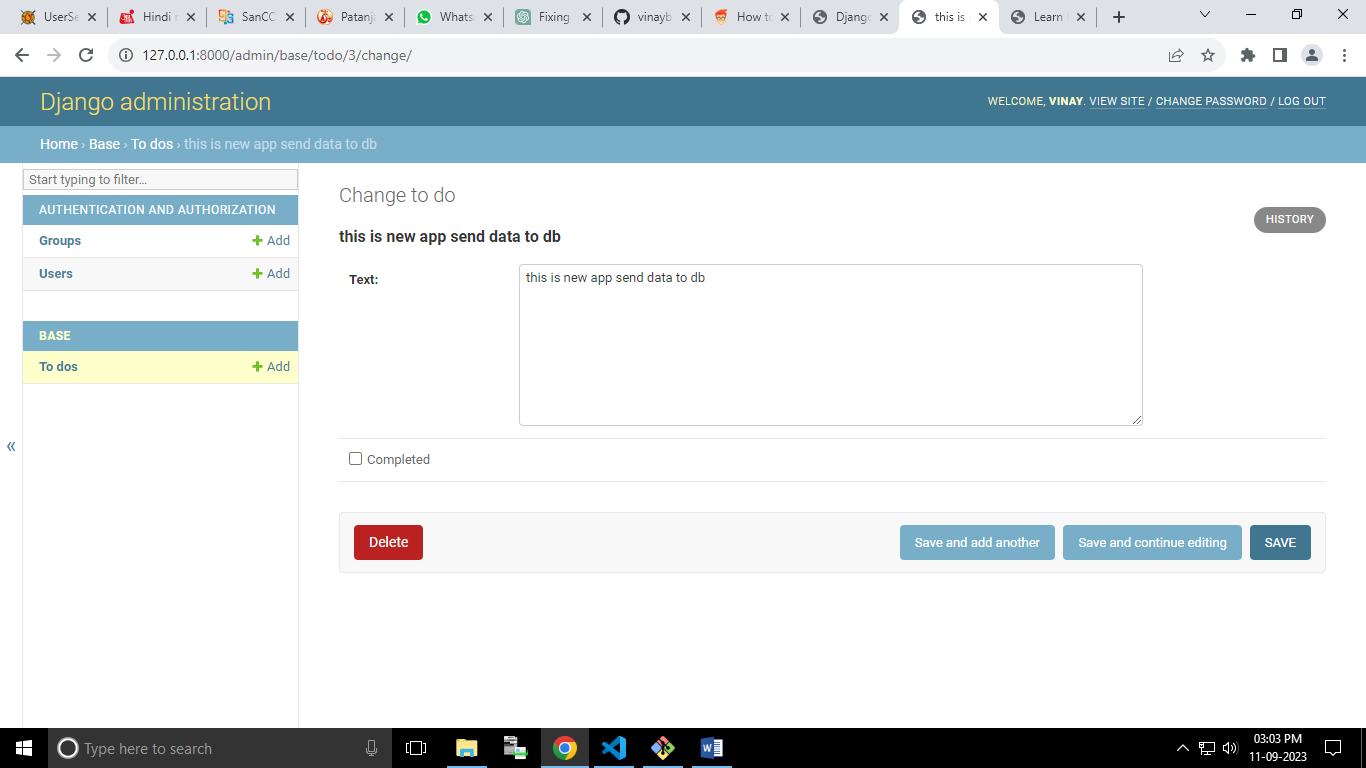
This one is my page



And filling the box data goes to my database







Render the save database

Update in views.py file

def homepage(request):

    todos = ToDo.objects.all()

    if request.method == "POST":

        form = ToDoForm(request.POST)

        if form.is\_valid():

            form.save()

            # Redirect to a different page after saving the form

            return redirect('home')

    else:

        form = ToDoForm()

    context = {

        'todos': todos,

        'form': form,

    }

    return render(request, 'base/home.html', context)

Update in HTml file

<div>

        {% for data in todos %}

          <h3>{{ data.text }}</h3>

        {% endfor %}

      </div>

In Whole page

{% load static %}

<!DOCTYPE html>

<html>

<head>

  <title>ToDo App</title>

  <link rel="stylesheet" href="{% static 'css/base.css' %}">

</head>

    <body>

        <h1 class="todo">This is New To Do App</h1>

        <div class="container">

          <form method="post">

            {% csrf\_token %}  <!-- Include the CSRF token for security -->

            {{ form.as\_p }}    <!-- Render the form fields as paragraphs -->

            <button type="submit">Submit</button>

        </form>

      </div>

      <div>

        {% for data in todos %}

          <h3>{{ data.text }}</h3>

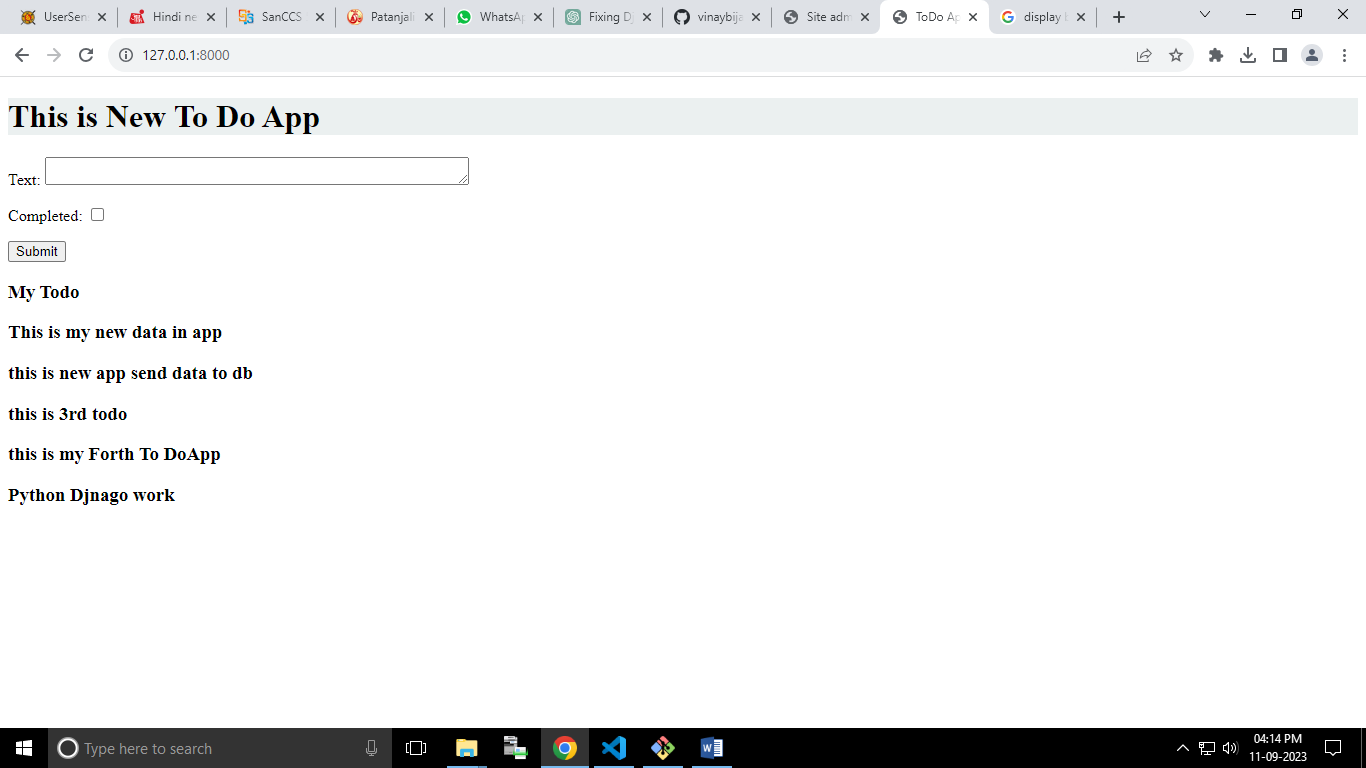
        {% endfor %}

      </div>

    </body>

</html>

Now HTml page look like this



<!--. <a href="{% url 'edit\_task' task.id %}">Edit</a> -->