TO DO LIST

VIEWS.PY

from django.shortcuts import render

from django.http import HttpResponse, HttpResponseRedirect

from .models import Todolist, Item

from .forms import CreateNewItem

def base(response, id):

    ls = Todolist.objects.get(id=id)

    item = ls.item\_set.all()

    return render(response, 'mama/list.html', {"ls":ls})

def create(response):

    if response.method == "POST":

        form= CreateNewItem(response.POST)

        if form.is\_valid():

            n = form.cleaned\_data["name"]

            t=Todolist(name=n)

            t.save()

        return HttpResponseRedirect("/%i" %t.id)

    else:

        form= CreateNewItem()

    return render(response, 'mama/create.html', {"form":form})

def home(response):

    return render(response, 'mama/home.html')

# Create your views here.

MODELS.PY

from django.db import models

class Todolist(models.Model):

    name = models.CharField(max\_length=200)

    def \_\_str\_\_(self):

        return self.name

class Item(models.Model):

    todolist = models.ForeignKey(Todolist, on\_delete=models.CASCADE)

    text = models.CharField(max\_length=300)

    complete = models.BooleanField()

    def \_\_str\_\_(self):

        return self.text

# Create your models here.

URLS.PY OF AN APP

from django.urls import path

from  .import views

urlpatterns = [

    path('', views.home, name="home-page"),

    path('<int:id>', views.base, name="home-page"),

    path('create/', views.create, name="home-page"),

]

FORMS.PY

from django import forms

class CreateNewItem(forms.Form):

    name = forms.CharField(label="name", max\_length=300)

    check = forms.BooleanField(required=False)

ADMIN.PY

from django.contrib import admin

from .models import Todolist, Item

admin.site.register(Todolist)

admin.site.register(Item)

# Register your models here.

TEMPLATES

BASE.HTML

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>{% block title %} base  page {% endblock %}</title>

    <style>

        .nav{

        background-color:black;

        color:black;

        display:flex;

        flex-direction:column;

        width:50cm;

        }

        body{

        overflow-x:hidden;

        }

        .nav ul li{

          display:inline-block;

          margin-left:20px;

        }

        .nav ul li:hover{

        background-color:blue;

        }

        .nav a{

        display:inline;

        text-decoration:none;

        color:white;

        font-size:20px;

        }

    </style>

</head>

<div style="display:flex;">

<div>

<div class="nav" >

    <ul>

<li><a href="/">Home</a></li>

<li><a href="/create">Create</a></li>

<li><a href="/1">View List</a></li>

        <li><a href="/register">Register</a></li>

    </ul>

</div>

<div style="margin-left:5cm;display:flex;flex-direction:column;">

<h1>WELCOME TO MY SITE --DIMOSO EL JUNIOR</h1>

{% block content %}

WELCOME TO BASE PAGE

{% endblock %}

    </div>

</div>

</body>

</html>

HOME.HTML

{% extends 'mama/base.html' %}

{% block title %} home {% endblock %}

{% block content %}

WELCOME TO  MY HOME PAGE

{% endblock %}

LIST.HTML

{% extends 'mama/base.html' %}

{% block title %}View Your Lists{% endblock %}

{% block content %}

View Your List  Here

<div style="color:blue;"><h1>{{ls.name}}</h1></div>

<ul>

    {% for item in ls.item\_set.all %}

    <li>{{item.text}}</li>

    {% endfor %}

</ul>

{% endblock %}

CREATE.HTML

{% extends 'mama/base.html' %}

{% block title %} create list {% endblock %}

{% block content %}

CREATE LISTS

<form method="post" action="/create/">

    {% csrf\_token%}

    {{form.as\_p}}

    <button type="submit" name="save" value="create">Create New</button>

</form>

<a href="/">click here</a>

{% endblock %}

URLS.PY OF THE PROJECT

from django.contrib import admin

from django.urls import path, include

from register import views as v

urlpatterns = [

    path('', include('mama.urls')),

    path('base/', include('mama.urls')),

    path('create/', include('mama.urls')),

    path('register/', v.register, name="register"),

    path('admin/', admin.site.urls),

]

SETTINGS.PY

"""

Django settings for todo project.

Generated by 'django-admin startproject' using Django 3.2.5.

For more information on this file, see

https://docs.djangoproject.com/en/3.2/topics/settings/

For the full list of settings and their values, see

https://docs.djangoproject.com/en/3.2/ref/settings/

"""

from pathlib import Path

# Build paths inside the project like this: BASE\_DIR / 'subdir'.

BASE\_DIR = Path(\_\_file\_\_).resolve().parent.parent

# Quick-start development settings - unsuitable for production

# See https://docs.djangoproject.com/en/3.2/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!

SECRET\_KEY = 'django-insecure-@np7gi@)\_^2@b+\*e$(lfg+eds-#e2&\*v1\_t^\_ty=oe-eoonbg8'

# SECURITY WARNING: don't run with debug turned on in production!

DEBUG = True

ALLOWED\_HOSTS = []

# Application definition

INSTALLED\_APPS = [

    'django.contrib.admin',

    'django.contrib.auth',

    'django.contrib.contenttypes',

    'django.contrib.sessions',

    'django.contrib.messages',

    'django.contrib.staticfiles',

    "crispy\_forms",

    'mama.apps.MamaConfig',

    'register.apps.RegisterConfig',

]

MIDDLEWARE = [

    'django.middleware.security.SecurityMiddleware',

    'django.contrib.sessions.middleware.SessionMiddleware',

    'django.middleware.common.CommonMiddleware',

    'django.middleware.csrf.CsrfViewMiddleware',

    'django.contrib.auth.middleware.AuthenticationMiddleware',

    'django.contrib.messages.middleware.MessageMiddleware',

    'django.middleware.clickjacking.XFrameOptionsMiddleware',

]

ROOT\_URLCONF = 'todo.urls'

TEMPLATES = [

    {

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

        'DIRS': [],

        'APP\_DIRS': True,

        'OPTIONS': {

            'context\_processors': [

                'django.template.context\_processors.debug',

                'django.template.context\_processors.request',

                'django.contrib.auth.context\_processors.auth',

                'django.contrib.messages.context\_processors.messages',

            ],

        },

    },

]

WSGI\_APPLICATION = 'todo.wsgi.application'

# Database

# https://docs.djangoproject.com/en/3.2/ref/settings/#databases

DATABASES = {

    'default': {

        'ENGINE': 'django.db.backends.postgresql',

        'NAME': 'myapp',

        'USER': 'postgres',

        'PASSword': '1234',

        'HOST': 'localhost',

        'PORT': '5432',

    }

}

# Password validation

# https://docs.djangoproject.com/en/3.2/ref/settings/#auth-password-validators

AUTH\_PASSWORD\_VALIDATORS = [

    {

        'NAME': 'django.contrib.auth.password\_validation.UserAttributeSimilarityValidator',

    },

    {

        'NAME': 'django.contrib.auth.password\_validation.MinimumLengthValidator',

    },

    {

        'NAME': 'django.contrib.auth.password\_validation.CommonPasswordValidator',

    },

    {

        'NAME': 'django.contrib.auth.password\_validation.NumericPasswordValidator',

    },

]

# Internationalization

# https://docs.djangoproject.com/en/3.2/topics/i18n/

LANGUAGE\_CODE = 'en-us'

TIME\_ZONE = 'UTC'

USE\_I18N = True

USE\_L10N = True

USE\_TZ = True

# Static files (CSS, JavaScript, Images)

# https://docs.djangoproject.com/en/3.2/howto/static-files/

STATIC\_URL = '/static/'

CRISPY\_TEMPLATES\_PACK = "bootstrap4"

# Default primary key field type

# https://docs.djangoproject.com/en/3.2/ref/settings/#default-auto-field

DEFAULT\_AUTO\_FIELD = 'django.db.models.BigAutoField'

FORMS.PY OF ANOTHER APP REGISTER

from django .contrib .auth import login, authenticate

from django .contrib .auth .forms import UserCreationForm

from django import forms

from django .contrib .auth .models import User

class RegisterForm(UserCreationForm):

    email = forms.EmailField()

    phone = forms.IntegerField()

    class Meta:

        model = User

        fields = ["username", "password", "email", "phone"]

VIEWS.PY FOR ANOTHER APP REGISTER

from django.shortcuts import render

from django .contrib .auth import login, authenticate

from django .contrib .auth .forms import UserCreationForm

from .forms import RegisterForm

def register(response):

    if response.method == "POST":

        form = RegisterForm(response.POST)

        if form.is\_valid():

            form.save()

    else:

        form = RegisterForm()

    return render(response, 'register/register.html', {"form":form})

# Create your views here.

TEMPLATES -> REGISTER

REGISTER.HTML

<style>

body{

background-color:yellow;

}

    .form-group{

    background-color:lightblue;

    }

    input{

    width:10cm;

    }

</style>

{% extends 'mama/base.html' %}

{% block title %} registration form {% endblock %}

{% load crispy\_forms\_tags %}

{% block content %}

USER REGISTRATION FORM

<form method="post" class="form-group">

    {% csrf\_token %}

    {{form|crispy}}

    <button type="submit" name="save" value="create">Register</button>

</form>

<a href="/">click here</a>

{% endblock %}