Task Management Application - ‘ToDo’

# Scope of the project:

Using Django, create a task management application ‘ToDo’ .

# Features of the project:

General

* Home
* Login
* Signup

User

* User Home/current todo
* Create todo
* Completed todo
* Show todo
* Delete todo
* Update todo
* Logout

# Template:

**Home page:**

Signup

Login

Logo ToDo

Hey! Welcome to ToDo. No more tensions ..Be organised by listing out your priorities..We help you for being organised. from now onwards you are not a mess! Acheive your Daily goals!

So Let’s start..

* ‘so Let’s start’ button will navigate to Login page

**Signup page:**

Please Sign up for getting started…

Name :

Email id :

:



Register

Confirm password :

Password :

* ‘Register’ button will navigate to user Home/current todo

**Login page:**

Please Login to view your todo’s

Name :

Password :

Login

**Database model:**

Title : CharField

Memo : TextField

Created\_date : DateTimeField

Completed\_date : DateTimeField

Important: Boolean Field

User : ForeignKey

**User Home or current todo page :**

create

completed

Current

Logo ToDo

Logout

Your Todo List

* Do exercise – at 3pm today
* Drink water – 8 glass a day
* **Buy food**

* Highlight with text area color and bold for those todo which are important True
* List down Title and some part of memo in current in checkbox
* Hyperlink each contents to show\_todo to view in form format(when click on content move to show\_todo page
* ‘Logo ToDo’ navigate to current\_todo
* ‘Completed’ navigate to complete\_todo
* ‘Create’ navigate to navigate\_todo
* ‘Logout’ navigate to home page.
* When no todo added display this. ‘Add ToDo ‘ will navigate to ‘Create todo’

You don’t have any current Todo’s Yet..

Add ToDo

**completed\_page :**

create

completed

Current

Logo ToDo

Logout

Your completed Todo List

* Do exercise – completed\_date
* Drink water – 8 glass a day -completed\_date
* Buy food – completed date
* Display completed date and title in completed\_todo

**create\_page :**

create

completed

Current

Logo ToDo

Logout

Title :

Memo:

Important

Create

* ‘Create’ button navigate to ‘current todo’

**show\_todo page :**

create

completed

Current

Logo ToDo

Logout

Do excercise

Title :

Memo:

Important

At 3 pm

Save

complete

Delete

* ‘complete’ button update current date in database and navigate to ‘complete todo’ .
* ‘Delete’ button delete record from database navigate to ‘current todo’
* Save – update the database and navigate to ‘current todo’

# 1.Setting up the Django-basic and check proper working:

### **1.Set up Django -project and app and check basic working of html file**

* Open command prompt and navigate to the folder you want to create the project (windows)

**Cd Desktop\django\_project**

**django-admin startproject Todo\_Project**

**cd Todo\_Project**

**django-admin startapp Todo\_app**

**atom .** ( to open atom software)

python manage.py runserver

### **2.Do updations in the project folder ‘Todo\_Project’**

* Todo\_Project/Todo\_Project /settings.py – add app name in ‘installed\_apps’. Also add static file location.

**INSTALLED\_APPS = [**

**'django.contrib.admin',**

**'django.contrib.auth',**

**'django.contrib.contenttypes',**

**'django.contrib.sessions',**

**'django.contrib.messages',**

**'django.contrib.staticfiles',**

**'Todo\_app',**

**]**

**TEMPLATES = [**

**{**

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

**'DIRS': ['Todo\_app/templates'],**

**'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',**

**],**

**},**

**},**

**]**

**import os.path**

**STATIC\_URL = '/static/'**

**STATICFILES\_DIRS=(os.path.join('static'),)**

* Todo\_Project/Todo\_Project /urls.py – add urls.py of app in path section

**from django.contrib import admin**

**from django.urls import path,include**

**from Todo\_app import views**

**urlpatterns = [**

**path('admin/', admin.site.urls),**

**path('',include(Todo\_app.urls)),**

**]**

### **3.Do updations in the app folder ‘Todo\_app’**

* Todo\_Project/Todo\_app/views.py – add details of html file

**from django.shortcuts import render**

**def home(request):**

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

* Create a file urls.py like Todo\_Project/Todo\_app/urls.py – add details of views

**from django.urls import path**

**from . import views**

**urlpatterns=[**

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

**]**

* Create a folder ‘templates’ in Todo\_Project/Todo\_app – create ‘index.html’ inside the template folder

**<!DOCTYPE html>**

**<html lang="en" dir="ltr">**

**<head>**

**<meta charset="utf-8">**

**<title>Todo app</title>**

**{%load static%}**

**<link rel="stylesheet" href={%static 'index.css'%}>**

**</head>**

**<body>**

**<p>Todo Application</p><br>**

**<button type="button" name="button" onclick="balert()">click here</button>**

**<script type="text/javascript" src={%static 'index.js'%}>**

**</script>**

**</body>**

**</html>**

* Create a folder ‘static’ in Todo\_Project/Todo\_app – create ‘index.css’ and ‘index.js’ inside the static folder

Index.js

**function balert()**

**{**

**alert("wow!")**

**}**

Index.html

**html{**

**background-color: green;**

**}**

* **Check the html page and make sure that css and js files works.Now we can go to the next step.**

# **2.Create Home page.**

Graphical user interface, text, application, email

Description automatically generated

* create ‘base.html’ file Todo\_Project/Todo\_app/templates/base.html

in base.html we can use repeating section of html and extend them to corresponding html file. Also check whether user authentication is done or not.

basic syntax –

{% if user.is\_authenticated %}

{% elif %}

{% endif %}

{% block content %} {% endblock%}

Main html –

{% extends ‘base.html’ %}

{% block content %}

<p> Welcome to app!</p>

{% endblock %}

base.html – repeating codes in html

{%load static%}

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- Bootstrap css -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">

</head>

<body>

<div class= “container”

{% block content %} {% endblock %} <!—run extended html code -->

</div>

<!—bootstrap -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p" crossorigin="anonymous"></script>

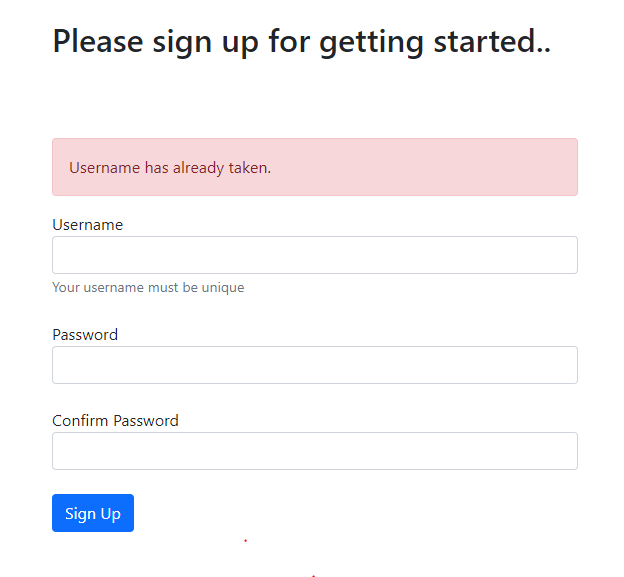
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js" integrity="sha384-7+zCNj/IqJ95wo16oMtfsKbZ9ccEh31eOz1HGyDuCQ6wgnyJNSYdrPa03rtR1zdB" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js" integrity="sha384-QJHtvGhmr9XOIpI6YVutG+2QOK9T+ZnN4kzFN1RtK3zEFEIsxhlmWl5/YESvpZ13" crossorigin="anonymous"></script>

</body>

</html>

# **2.Signup page.**



* in Todo\_Project/Todo\_app/views.py

import signup form - UserCreationForm.

Import user model – User

Import login

Import integrity error (username already taken)

basic syntax –

from django.contrib.auth.forms import UserCreationForm

from django.contrib.auth.models import User

from django.contrib.auth import login

from django.db import IntegrityError

def signup\_user(request):

if request.method=='GET':

return render(request,'signup\_user.html',{'form':UserCreationForm()})

else:

if request.POST['password1']==request.POST['password2']:

try:

user=User.objects.create\_user(request.POST['username'],password=request.POST['password1'])

user.save()

login(request,user)

return redirect('home')

except IntegrityError:

return render(request,'signup\_user.html',{'form':UserCreationForm(),'error':'Username has already taken.'})

else:

return render(request,'signup\_user.html',{'form':UserCreationForm(),'error':'password not match'})

* in Todo\_Project/Todo\_app/templates/signup\_user.html

{% if error %}

{{error }}

{% endif %}

<form action="{% url 'signup\_user' %}" method="post">

{% csrf\_token %}

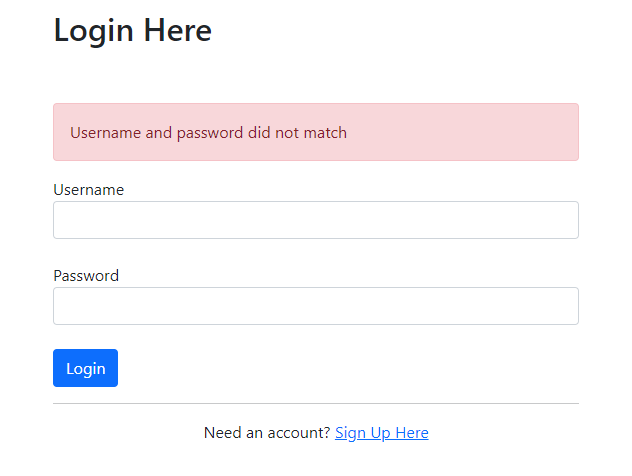
{{form.as\_p}}

<br>

<button type="submit" class="btn btn-primary">Sign Up</button>

</form>

# **3.Login page:**



* in Todo\_Project/Todo\_app/views.py

import login form - AuthenticationForm()

Import for authentication - authenticate

basic syntax –

from django.contrib.auth.forms import AuthenticationForm

from django.contrib.auth import login, authenticate

from django.db import IntegrityError

def login\_user(request):

if request.method=='GET':

return render(request,'login\_user.html',{'form':AuthenticationForm()})

else:

user=authenticate(request,username=request.POST['username'],password=request.POST['password'])

if user is None:

return render(request,'login\_user.html',{'form':AuthenticationForm(),'error':'Username and password did not match'})

else:

login(request,user)

return redirect('home')

* login\_user.html - basic syntax –

# <form action="{% url 'login\_user' %}" method="post">

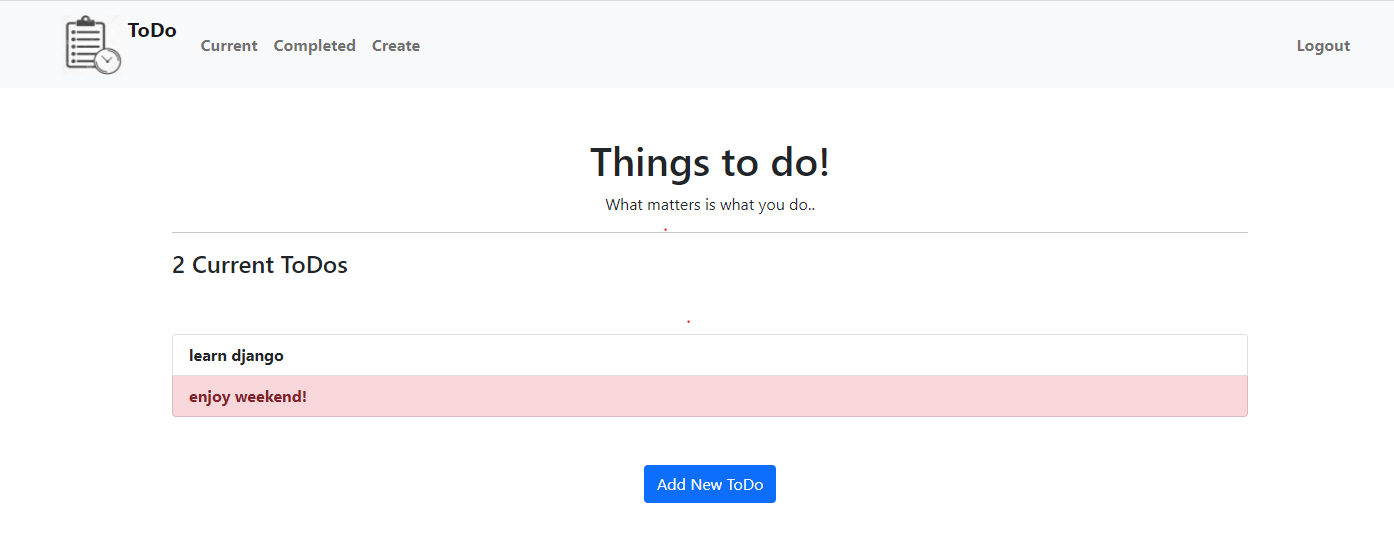
# {% csrf\_token %}

# {{form.as\_p}}

# <button type="submit" name="button">Login</button>

# </form>

# **4.Current todo page:**



* in Todo\_Project/Todo\_app/views.py

Import model- Todo db and fetch datas from the model and pass the data to html file to display when completed date is null

basic syntax –

from .models import todo\_db

def current\_todo(request):

data=todo\_db.objects.filter(user=request.user,completed\_date\_\_isnull=True)

return render(request,'current\_todo.html',{'todo\_data':data})

* in current\_todo.html

basic syntax –

from .models import todo\_db

def current\_todo(request):

<h1>Things to do!</h1>

<p>What matters is what you do..</p>

<ul>

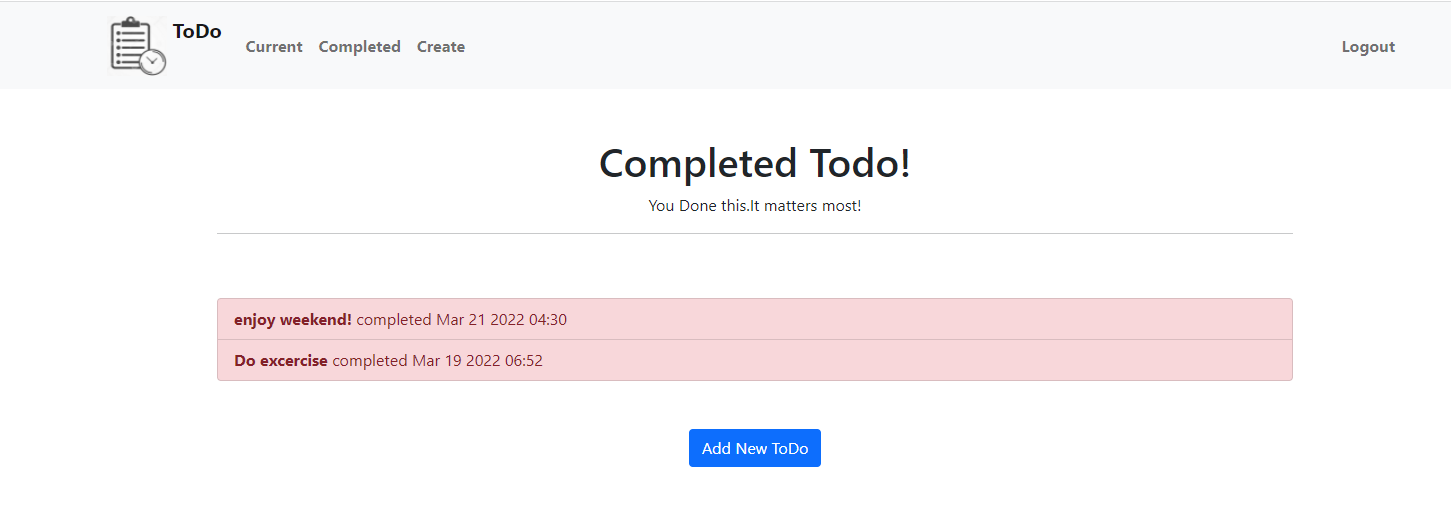
{% for data in todo\_data %}

<li>{{data.title}}</li>

{% endfor %}

</ul>

# **5.completed todo page:**



* in Todo\_Project/Todo\_app/views.py

Import model- Todo db and fetch datas from the model and pass the data to html file to display

Same as current except display when completed date is not null

basic syntax –

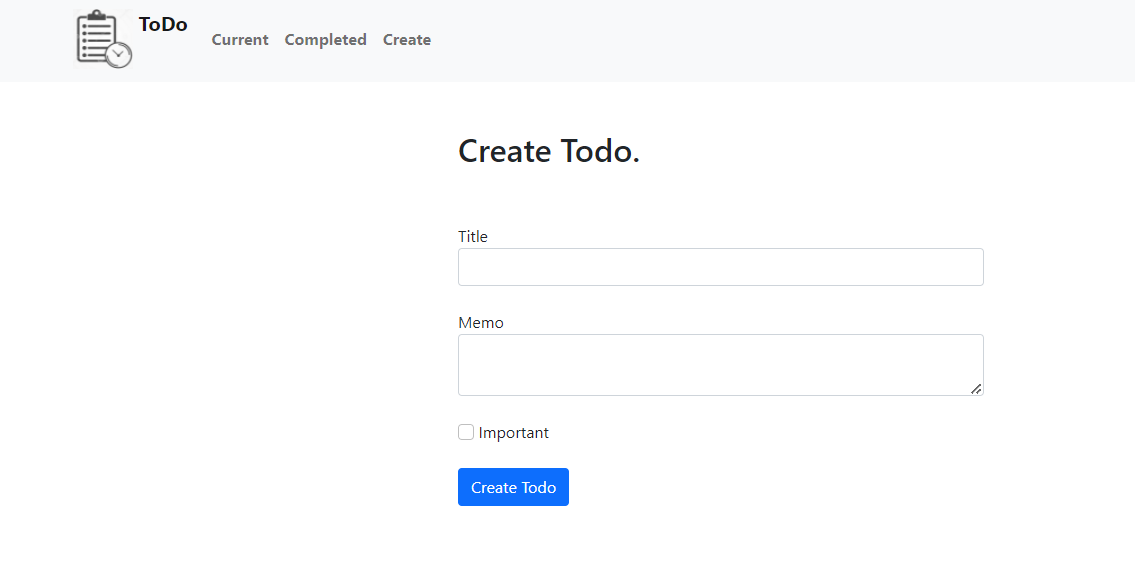
from .models import todo\_db

def completed\_todo(request):

data=todo\_db.objects.filter(user=request.user,completed\_date\_\_isnull=False).oredr\_by(‘-completed\_date’)

return render(request,'completed\_todo.html',{'todo\_data':data})

# **6.create todo page:**



* in Todo\_Project/Todo\_app/views.py

pass the form created for the model. When user add todo add the user details along the data and save to the database.

basic syntax –

def create\_todo(request):

if request.method=="GET":

return render(request,'create\_todo.html',{'form':todo\_form()})

else:

try:

form=todo\_form(request.POST)

new\_todo=form.save(commit=False)

new\_todo.user=request.user

new\_todo.save()

return redirect('current\_todo')

except ValueError:

return render(request,'create\_todo.html',{'form':todo\_form(),'error':"Invalid data passed In.Please try again"})

in html – to see form

{{form.as\_p}}

# **7.Logout page:**

* in Todo\_Project/Todo\_app/views.py

import logout then redirect to general home page

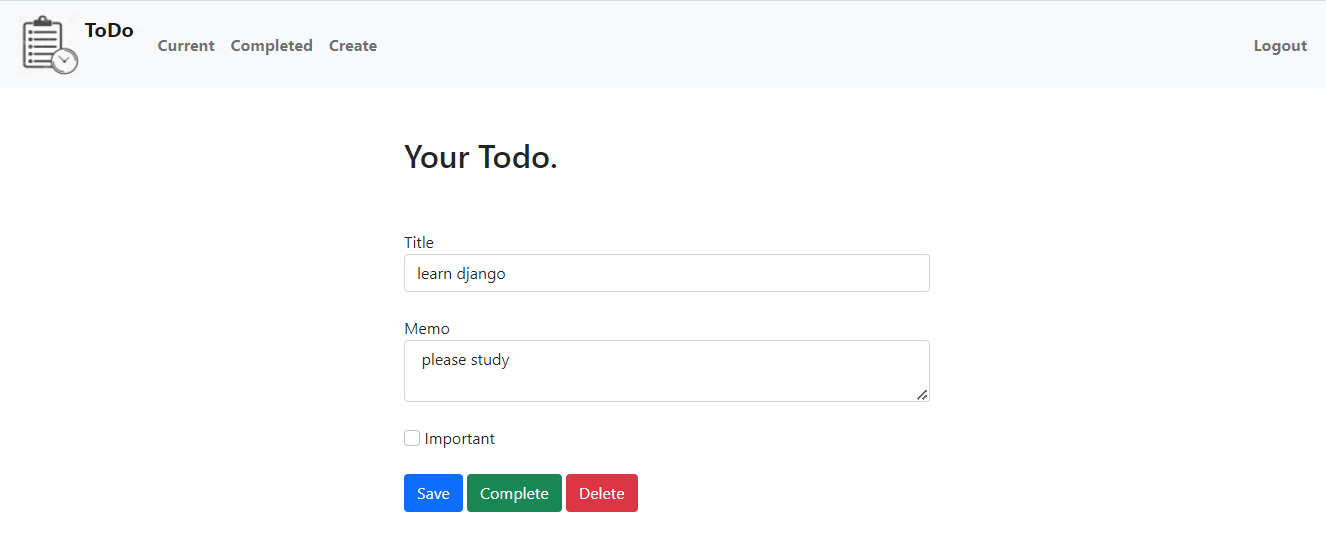
basic syntax –

def logout\_user(request):

logout(request)

return redirect('home')

# **7.show \_todo page:**



* Show\_todo page give a detailed view of a particular todo.todos in current\_todo and completed\_todo is hyperlinked with show\_todo page.we can update,delete and complete the todo from this page.
* In Todo\_Project/Todo\_app/views.py

import logout then redirect to general home page

basic syntax –

def show\_todo(request,todo\_pk):

data=get\_object\_or\_404(todo\_db, pk=todo\_pk, user=request.user)

if request.method=="GET":

form=todo\_form(instance=data)

return render(request,'show\_todo.html',{'todo\_data':data,'form':form})

else:

try:

form=todo\_form(request.POST,instance=data)

form.save()

return redirect('current\_todo')

except ValueError:

return render(request,'show\_todo.html',{'todo\_data':data,'form':form,'error':"data not valid"})

def complete\_todo(request,todo\_pk):

data=get\_object\_or\_404(todo\_db, pk=todo\_pk, user=request.user)

data.completed\_date=timezone.now()

data.save()

return redirect('completed\_todo')

def delete\_todo(request,todo\_pk):

data=get\_object\_or\_404(todo\_db, pk=todo\_pk, user=request.user)

data.delete()

return redirect('current\_todo')

* in Todo\_Project/Todo\_app/url.py

pass primary key of the todo

basic syntax –

urlpatterns=[

# indirect pages

path('show/<int:todo\_pk>', views.show\_todo,name="show\_todo"),

path('show/<int:todo\_pk>/complete/', views.complete\_todo,name="complete\_todo"),

path('show/<int:todo\_pk>/delete/', views.delete\_todo,name="delete\_todo"),

]

* In show\_todo html file display form with values filled.
* Inorder to avoid view of the user pages by url. In views we add ‘@login\_required’ above declaring function.

from django.contrib.auth.decorators import login\_required

@login\_required

def logout\_user(request):

logout(request)

return redirect('home')