

Project 7

open window in atom and drag todo folder into it,

todo-app

cd mysite

cd myprojectserver

django-admin startproject mywebsite

cd mywebsite

python manage.py startapp todo

drag mywebsite into atom

mywebsite

Settings.py

APP

'todo'

urls.py

path, include

path("", include('todo'urls'))

app

(2) create this urls.py inside todo folder

copy from mywebsite → urls.py edit it

from django.urls import path

from . import import views

```
urlpatterns = [
    path(' ', views.index, name = 'index')
]
```

(2) todo → views.py

```
from django.shortcuts import render
from django.http import HttpResponse
```

```
def index(request):
    return HttpResponse('My TODO APP')
```

~~python manage.py runserver~~ →

(3) todo → models.py

```
from django.db import models
```

```
class Todo (models.Model):
    complete = models.BooleanField (default = False)
    todo_text = models.CharField (max_length = 50)
```

```
def __str__(self):
    return self.todo_text
```


(4) todo → admin.py and register model

from django.contrib import admin

from models import Todo
admin.site.register(Todo) → class

(5) migrate

python manage.py migrate ↵

python manage.py makemigrations todo ↵

python manage.py migrate ↵

(6) create superusers for admin:

python manage.py createsuperuser ↵

user

email

pswd

(7) python manage.py runserver ↵

(8) create new folders inside todo → templates → todo

todo → static → todo

(A) todo → templates → todo : why here because we have from
todo-asset folders

(B) todo → ^{Static} ~~templates~~ → ^{todo} ~~static~~ : copy here css and js folders
from todo-asset
folders

(9) templates → todo → index.html

load static files first —

{% load static %}

<head>

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

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

</head>

(10) Edit views.py

from django.shortcuts import render

def index(request):

return render(request, 'todo/index.html')

python manage.py runserver

✓ Read about python decorators from realpython.com website

— Adding todo and decorators —

create forms.py into todo app

(11) from django import forms

```
class TodoForm (forms.Form):
```

```
    text = forms.CharField (max_length = 50,
```

```
        widget = forms.TextInput (
```

```
            attrs = {
```

```
                'class' : 'form-control',
```

```
                'placeholder' : 'A Django Todo'
```

```
            }
```

```
    )
```

from
index.html

(12) Edit views.py

```
from django.shortcuts import render, redirect
```

```
from forms import TodoForm
```

```
from models import Todo
```

```
from django.views.decorators.http import
```

class name
→ require-POST

```
def index (request):
```

```
    mytodo = Todo.objects.order_by ('id')
```

```
    form = TodoForm ()
```

```
    context = { 'mytodo' : mytodo, 'form' : form }
```

```
    return render (request, 'todo/index.html', context)
```

@require_POST

```
def addNewTodo (request):
```

```
    form = TodoForm ( request.POST )
```

```
    if form.is_valid():
```

```
        my_new_todo = Todo ( todo_text = request.POST [ 'text' ] )
```

↓
models.py

```
        my_new_todo.save()
```

```
    return redirect ( 'index' )
```

↘ views.py

```
def CompleteTodo (request, todo_id):
```

```
    mytodo = Todo.objects.get ( pk = todo_id )
```

```
    mytodo.complete = True
```

```
    mytodo.save()
```

```
    return redirect ( 'index' )
```

↘ views.py

```
def deleteTodo (request):
```

```
    Todo.objects.filter ( complete_exact = True ).delete()
```

```
    return redirect ( 'index' )
```


(13) todo → urls.py

• Edit this

from django.urls import

:

path('add', views.addNewTodo, name='add'),

path('complete/<todo_id>', views.completeTodo, name='complete'),

path('delete', views.deleteTodo, name='delete'),

]

views.py

(14) index.html

<form action="{% url 'add %' %}" method="POST" style="border: 1px solid black; padding: 5px; width: 100%;">

{% csrf_token %}

<div class="form-group">

<div class="input-group">

{{ form.text }}

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

</div>

</div>

```
<div class="btn-toolbar">
```

```
<a href="#" class="btn btn-danger" data-toggle="delete">
```

```
<button type="button" class="btn btn-danger" data-toggle="delete">
```

```
<i class="fa fa-trash" data-toggle="delete"> DELETE COMPLETED
```

```
</button>
```

```
</a>
```

```
</div>
```

views by context name

```
{% for t in mytodo %}
```

```
<li class="list-group-item"> {{ t.todo_text }} </li>
```

models.py

```
{% if t.complete %}
```

```
{% for t in mytodo %}
```

```
{% if t.complete %}
```

```
<li class="list-group-item"> {{ t.todo_text }} </li>
```

models.py

```
{% else %}
```

```
<a href="#" class="btn btn-danger" data-toggle="complete">
```

```
<li class="list-group-item"> {{ t.todo_text }} </li>
```

```
{% endif %}
```

```
{% endfor %}
```

```
</ul>
```