

Step-by-step guide to create a simple To-Do List app in Django, save it, and run it locally.

BEST PRACTICE (Recommended): Use Virtual Environment

This avoids many errors.

1 Create virtual environment

```
python -m venv venv
```

2 Activate it (Windows)

```
venv\Scripts\activate
```

You should now see:

```
(venv) D:\todo_project>
```

3 Install Django inside venv

```
pip install django
```

4 Run your command

```
python manage.py startapp todo
```

Step 5: Register the App

Open:

```
todo_project/settings.py
```

Find `INSTALLED_APPS` and add:

```
INSTALLED_APPS = [
    ...
    'todo',
]
```

Save the file.

□ Step 6: Create the Model (Database Table)

Open:

```
todo/models.py
```

Replace with:

```
from django.db import models

class Task(models.Model):
    title = models.CharField(max_length=200)
    completed = models.BooleanField(default=False)
    created = models.DateTimeField(auto_now_add=True)

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

Save the file.

□ Step 7: Create Database Tables

Run:

```
python manage.py makemigrations
python manage.py migrate
```

□ Step 8: Create a View

Open:

todo/views.py

Add:

```
from django.shortcuts import render, redirect, get_object_or_404
from .models import Task
```

```
# READ + CREATE
```

```
def task_list(request):
    if request.method == 'POST':
        title = request.POST.get('title')
        if title:
            Task.objects.create(title=title)
        return redirect('task_list')

    tasks = Task.objects.all().order_by('-created')
    return render(request, 'task_list.html', {'tasks': tasks})
```

```
# UPDATE
```

```
def update_task(request, pk):
    task = get_object_or_404(Task, pk=pk)

    if request.method == 'POST':
        task.title = request.POST.get('title')
        task.completed = 'completed' in request.POST
```

```

        task.save()

        return redirect('task_list')

    return render(request, 'update_task.html', {'task': task})

# DELETE

def delete_task(request, pk):
    task = get_object_or_404(Task, pk=pk)

    if request.method == 'POST':
        task.delete()

        return redirect('task_list')

    return render(request, 'delete_task.html', {'task': task})

```

Save it.

□ Step 9: Create URLs

1 □ Create `todo/urls.py`

Create a new file:

`todo/urls.py`

Add:

```
from django.urls import path
from . import views
```

```
urlpatterns = [
    path('', views.task_list, name='task_list'),
    path('update/<int:pk>/', views.update_task, name='update_task'),
    path('delete/<int:pk>/', views.delete_task, name='delete_task'),
]
```

2 Connect App URLs to Project

Open:

```
todo_project/urls.py
```

Modify it:

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('todo.urls')),
]
```

Save.

Step 10: Create Template Folder

Inside `todo` folder create templates:

Then create file inside templates:

Create `task_list.html`

add your file with code:

```
<!DOCTYPE html>
<html>
<head>
    <title>To-Do List</title>
</head>
<body>

<h1>My To-Do List</h1>

<!-- CREATE -->
<form method="POST">
    {%- csrf_token %}
    <input type="text" name="title" placeholder="Enter task">
    <button type="submit">Add</button>
</form>

<hr>

<!-- READ -->
<ul>
    {%- for task in tasks %}
        <li>
            {%- if task.completed %}
                <strike>{{ task.title }}</strike>
            {%- else %}
                {{ task.title }}
            {%- endif %}

            <!-- UPDATE -->
            <a href="{% url 'update_task' task.id %}">Edit</a>

            <!-- DELETE -->
            <a href="{% url 'delete_task' task.id %}">Delete</a>
        </li>
    {%- empty %}
        <li>No tasks yet.</li>
    {%- endfor %}
</ul>

</body>
</html>
```

Create `update_task.html`

Create file:

```
todo/templates/update_task.html
```

Add:

```
<h2>Update Task</h2>

<form method="POST">
    {% csrf_token %}
    <input type="text" name="title" value="{{ task.title }}">
    <label>
        Completed:
        <input type="checkbox" name="completed" {% if task.completed %}checked{% endif %}>
    </label>
    <button type="submit">Update</button>
</form>

<a href="{% url 'task_list' %}">Back</a>
```

□ Create `delete_task.html`

Create file:

```
todo/templates/delete_task.html
```

Add:

```
<h2>Are you sure you want to delete "{{ task.title }}"?</h2>

<form method="POST">
    {% csrf_token %}
    <button type="submit">Yes, Delete</button>
</form>

<a href="{% url 'task_list' %}">Cancel</a>
```

Step 11: Run the Server

Inside project folder:

```
python manage.py runserver
```

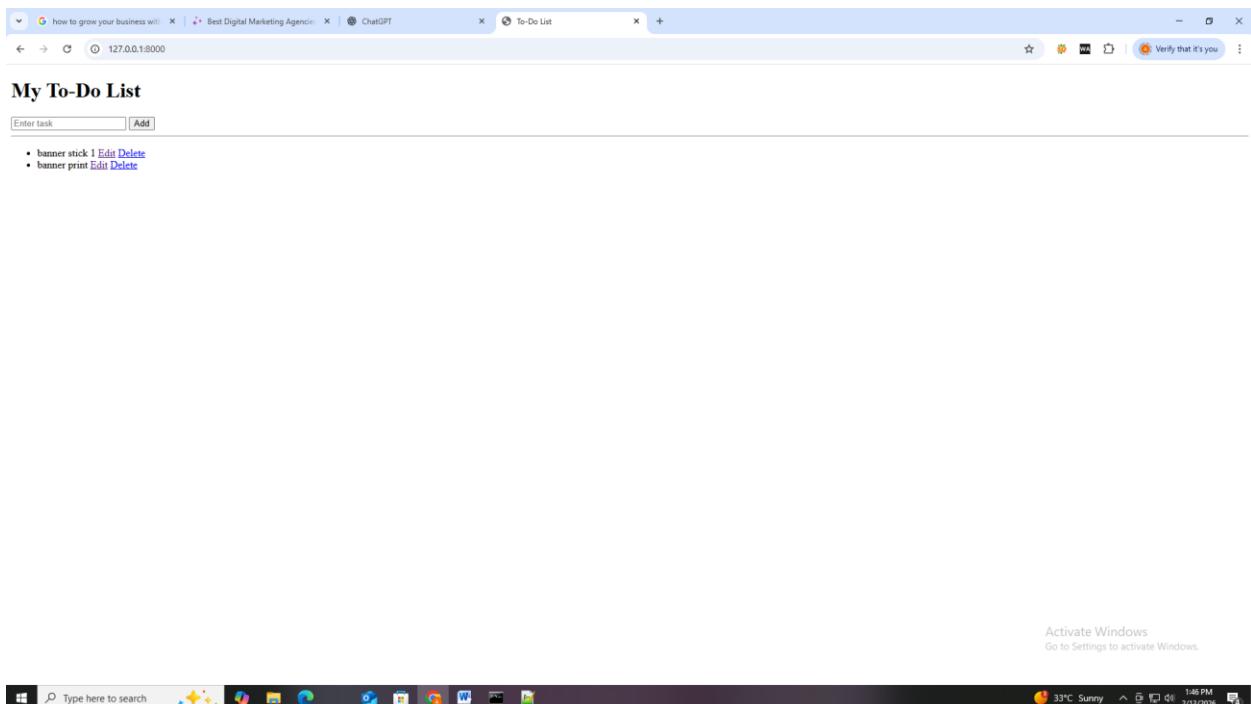
You'll see something like:

```
Starting development server at http://127.0.0.1:8000/
```

Open your browser and go to:

```
http://127.0.0.1:8000/
```

Your To-Do List app is running!



Optional: Create Admin User

To manage tasks from admin panel:

```
python manage.py createsuperuser
```

Then register model in:

```
todo/admin.py
from django.contrib import admin
from .models import Task

admin.site.register(Task)
```

Visit:

```
http://127.0.0.1:8000/admin/
```

Output:-

