CUSTOM FORMS

In the last video we created a simple form using the djangos built in forms. This tutorial will show you how we can create custom forms on our own without using a form template like before. The reason we would do this is because sometimes we want to customize our forms extensively and djangos built in forms cannot do what we would like.

django

MODIFYING LIST.HTML

The custom form we are going to create will be within our list.hmtl file. There we will add a way for the user to add a new item to each To Do List and to save which items are completed by checking a check button. To do this we will replace our code with the following:

# list.html file

Modify code

```
{% extends "main/base.html" %}
{% block title %}View List{% endblock %}
{% block content %}
    <h1>{{ls.name}}</h1>

    <form method="post" action="#">
    {% csrf_token %}
        <ul>
            {% for item in ls.item_set.all %}
                {% if item.complete == True %}
                    <li><input type="checkbox", value="clicked", name="c{{item.id}}" checked>{{item.text}}</li>
                {% else %}
                    <li><input type="checkbox", value="clicked", name="c{{item.id}}">{{item.text}}</li>
                {% endif %}
            {% endfor %}
        </ul>
        <button type="submit", name="newItem", value="newItem">Add Item</button>
        <input type="text", name="new">
        <button type="submit", name="save", value="save">Save</button>
    </form>

{% endblock %}
```

Now we have setup a form where each item in our list has a check-button to the left of it, a save button, and a way to add a new item.

GETTING INFORMATION

Similarly to before we need a way to get information from our form. Since our form uses a POST request we use a similar approach to before. The main difference is that since we haven't used djangos forms we are going to need to validate the form ourselves and extract the information we need.

Inside our views.py file we will edit the function responsible for showing our list view. Since we have two buttons in out form we will need to determine which one was clicked so we know if we need to add a new item or simply update our items.

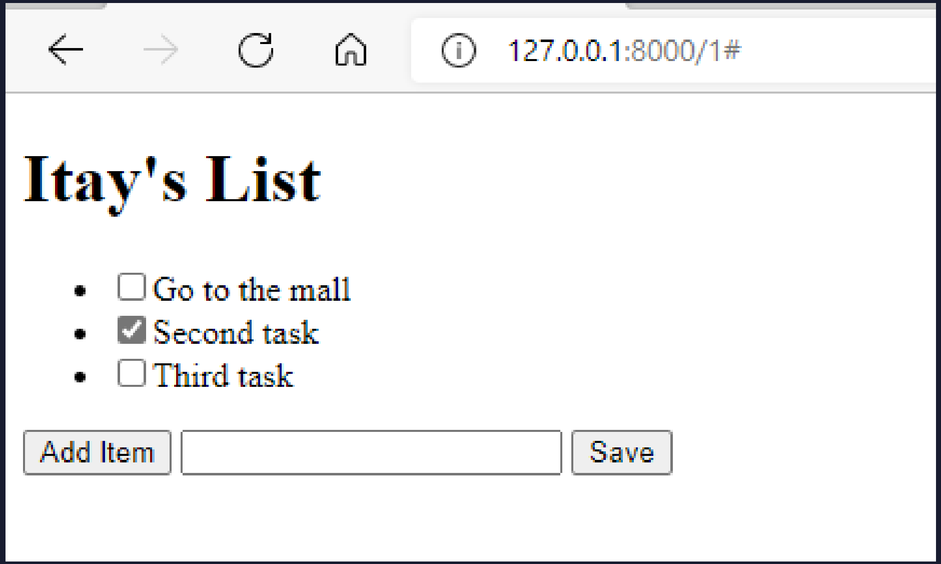
This is what our new function will look like:

```python
def index(response, id):
    ls = ToDoList.objects.get(id=id)

    if response.method == "POST":
        if response.POST.get("save"):
            for item in ls.item_set.all():
                if response.POST.get("c" + str(item.id)) == "clicked":
                    item.complete = True
                else:
                    item.complete = False

                item.save()

        elif response.POST.get("newItem"):
            txt = response.POST.get("new")

            if len(txt) > 2:
                ls.item_set.create(text=txt, complete=False)
            else:
                print("invalid")

    return render(response, "main/list.html", {"ls": ls})
```

Add code

FORM

Now we have a form that allows us to add new items to our To Do Lists and to update the state of each of our items.



127.0.0.1:8000/1#

Itay's List

- ☐ Go to the mall
- ✓ Second task
- ☐ Third task