

TERMINOLOGY

MVC *Model-View-Controller* is the “keep stuff separate” philosophy to separate out the code in a complicated web apps into 3 different categories

ORM *Object Relational Mapper* - the library used by the model to make special classes (called *models*) that can be saved and retrieved from the database (that is to say, *persisted*). Works by generating SQL.

migration Auto-generated code that uses the ORM to get the database sync'ed up with the latest additions to a project's models.

applying migration Using a *migration* to get the DB up-to-date and ready for use.

MVT *Model-View-Template* are the three categories of code in a Django project

app A single Django-powered *project* can consist of multiple *apps*, where each app can have a full vertical “slice” of models, views, and templates.

USING DJANGO FORMS

views.py

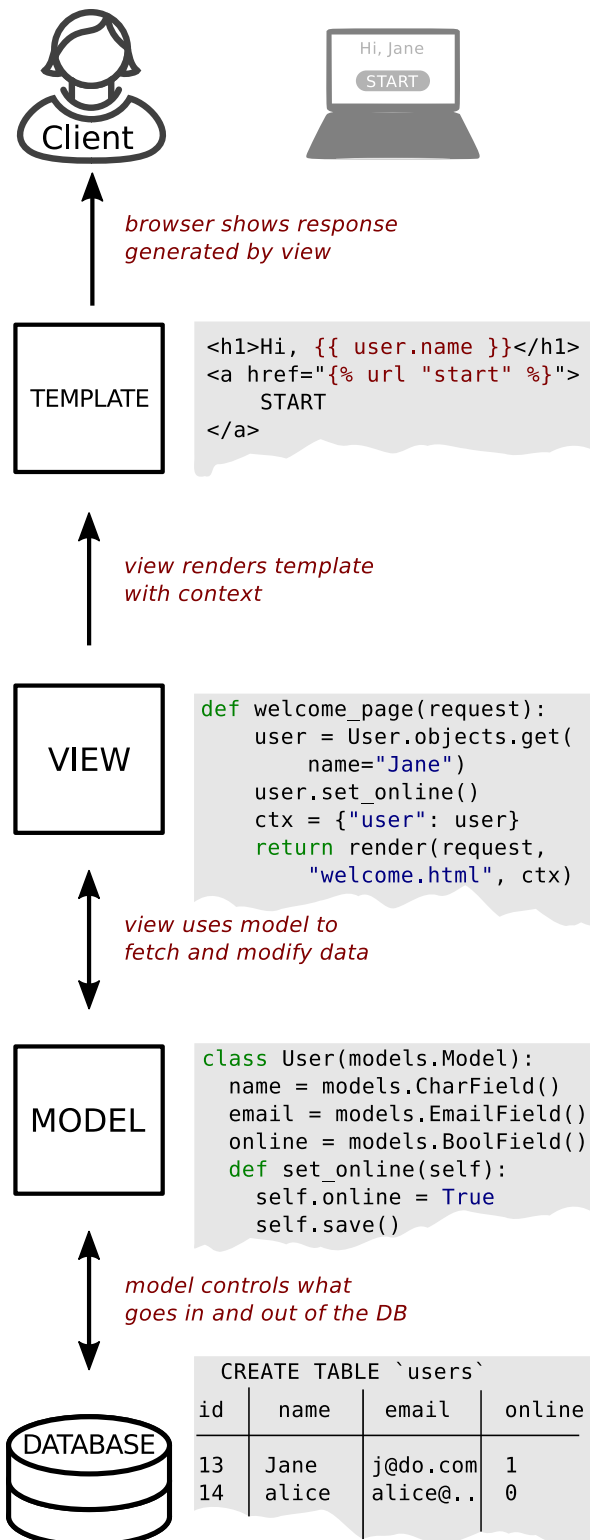
```
class NewUserForm(forms.Form):
    name = forms.CharField(required=True)
    email = forms.EmailField()

def get_name(request):
    if request.method == "GET":
        # Is initial GET: Create a blank form
        form = NewUserForm()
    else:
        # Is POST: Create a form based on POST data
        form = NewUserForm(request.POST)
        if form.is_valid():
            # If valid, create a new user & redirect
            user = User()
            user.username = form.cleaned_data["name"]
            user.email = form.cleaned_data["email"]
            user.save()
            return HttpResponseRedirect("/thanks/")

    ctx = {"form": form}
    return render(request, "users/create.html", ctx)
```

users/create.html

```
{% extend "base.html" %}
{% block content %}
<h1>Create new user</h1>
<form action="/new-user/" method="post">
    {% csrf_token %}
    {{ form }}
    <input type="submit" value="Submit" />
</form>
{% endblock content %}
```



model is the gate-keeper to data stored in the *database*

view defines *business logic* of your web app (called *controller* by Rails and others)

template is the appearance of your site in HTML (called *view* by Rails and others)