first part: views & urls

to contnue about view, we have to types of controllers: function_based and class_based:

in function_based: just get request and after works returns httprespnse(or render or jasonresponse or hrrprespnseredirect)

in class_based: it's sensitive to type of request(get, post , ...) and by intrhence from django classes can handle More precisely

about urls:

we have a urlpatterns in main app that has this form: urlpatterns=[path('developers/', include('app.urls'),]

first parameter is the route that starts with that and it helps django if url starts with that, continue link is in app.urls and go find it.

so in app/urls.py we have another urlpatterns=[path('ali/', views.show_name, name="show_name")]. and if continue url is ali th show_name function works.

(the name will be used in redirect and template so wait for it.)

let's see power of dynamic link:

we can write '<name>' or '<int:age>' or '<str:name>/<int: age>/' and the urlpattern will catch all this patterns and send to vew function.

don't forget view function arg has to be same name with part in url.

so our link mission finished here and the continue is related to view.

in view:

as sayed before, the view function gets request and any parameters we want from url it has to do somthing. for example get data from database.

and after that returns httprespnse(the_string_response).

but it's not very friendly to show to client so we can use html form. for this we can write html form in a string and return as a string.

so our proposed solution is template.

and last point, we can send the client to another page. for this we have 2 ways:

1- write httpresponseredirect("developers/"+the_name): so it's like we search it in serchbar and go to link

2- write httpresponseredirect(reverse("show_name", args=["ali", 10])

in second way we easily use link that we named(in the app) before and we are sure about link is valid.

so let's see template.

template:

at first in our app we make a template folder and in it make a folder with app name. and make a html file in it.

(becauase a parameter in settings.py all template folders will be recognized and our html file path is valid)

to continue, we make a html file for example I learn this:

```
<ul>
    <li> <a href="">name</a></li>
</ul>
```

and now we can use temlate ruls for examle tag and filter to take it dynamic. let's see them:

{{varible_name}}: it could be any type and from view or made in template(in for). it puts in page

for index or dictionry key have to use .blahblah

{%for skill in developer.skills%}.......{% endfor %}: for loop

` {% if a> 10%} ....{%endif%}: if elif else

{% url "show_developers" developer.username %}: like reverse we have url filter to put link

url name arg

and we can intrehence:

we make a template folder in main folder and put base.html in it.

and in settings.py in Templates, in dirs: [Base_Dir / "template"] to say to django don't forget this template folder to find my template.

in base.html parts we want to be diffrent in every template, put this:

{% block title_name %}{% endblock %} if i put sth between it's defualt if template has no use this block

and in template we put:

{% extends 'base.html'%}

{% block title_name %}

my code

{% endblock %}

and at the end in view function:

return render(request, "app/my.html", {"variable": my_list}: request, template path, variables used in template.

and finish.

(some new functions:)

.capitalize

.keys