**Django template language part 1**

Now we should be here, where we can see the page shown below. Recall from the previous lecture how it navigates or works

Graphical user interface, application

Description automatically generated

We can do customization for the site, discuss about various fonts/styles/headings that can be applied to websites. Let’s try applying the **<h1> hello world </h1>** , refresh the page and observe the output. It should increase the size of the text on the web page.

What if you want to create a multiple tags, we cannot keep on adding multiple tags in the same function. What we should do is – we will create a separate page (home page) which will have lot of html tags. We know that if we create a page it will be a static page as once if we specify the content it cannot be changed, we want to make it dynamic and that’s where in Django we have this amazing concept called templates. Imagine in templates what we can do is we can have an html page and in that we can have dynamic content, the layout(bg color) will be fixed but the data will be changed.

In Django we have a concept called Django template language. **Take a look at settings.py in the main project folder and observe the templates section**. We can see that we are using templates, using these we can achieve template language. Now to see that we have to create a page. Normally we can create a page in the root directory or **we can create a folder and create our pages in the same folder and a home.html page as shown below and write something inside the page other than the previous message on the page.**

Graphical user interface, text, application

Description automatically generated

* Now if we refresh the page will it reflect the changes? Discuss

It will not reflect the changes as we are still returning our old ‘hello world’ message through the home function. So we need to make some changes to make this work.

1. **Change the settings.py in the main project folder – make change the to DIRS(directories) where we will specify the path of the template, where we will search it: which means Django should know where we have all the templates, do it by specifying the path. Make changes as shown below**.

Text

Description automatically generated

1. Now we also need to change the views.py file as we need to call the html page rather than hello world. We need to render the template, we will pass the request object as shown below and calling the html page.

A screenshot of a computer

Description automatically generated with medium confidence

If you refresh the browser we can see that we got the changes reflected as shown below.

Graphical user interface, application

Description automatically generated

We are discussing about dynamic content and yet we see simple static page, what if we want to use dynamic content. Instead of saying hello world we can change it to hello someone. Can we do it ? yes, we can. Instead of a static content, we can make the name dynamic. How do we do it? – add a dictionary and provide the content as shown below.

Graphical user interface, text

Description automatically generated

* Now go back to home.html and we want to use the dictionary provided in views file as shown below. If you change the name in views page and observe the output will change according to the value provided in the views file.

A screenshot of a computer

Description automatically generated with medium confidence

* Notice that we are not changing the home.html page, we are making changes to the views file and it is dynamically changing the page during the output. Double curly braces mentioned above means that the content mentioned inside this is dynamic. This is normal change and can we do operations here? Yes, we can let’s look at the next section for the same.

**DTL Part 2**

What if you want to apply the theme/color to all the pages in the project ? let’s see how to do it.

Create a new file in the templates folder as base.html 🡪 **! and hit enter will provide the basic code** as shown below and make changes highlighted accordingly which is changing colors.

In this body all the pages will replace their content here ex: data inside our home.html should go to the body of this base.html page. We will write it in DTL format or jinja format(if required we need to install the jinja extension)

Graphical user interface, text

Description automatically generated

Now if you refresh the home.html page it will not reflect the changes because we haven’t linked the base.html page.

In order to make it work, we need to make changes to the home.html as shown below. Jinja code extends to base.html and hello world will be inside the block content .

Graphical user interface, text, application

Description automatically generated

* Now if we refresh this, we will get the cyan color in the background for our main page.

Graphical user interface, website

Description automatically generated

* Now you can change the bgcolor value to our name which will give different colors as output. Our names will get converted into hex values and gives the corresponding color.