**Static files 1**

Till this point we were able to create a simple application to add two numbers, but the User Interface (UI) was basic as shown below.

Graphical user interface, text, application

Description automatically generated

When you are creating a professional website we will want or need a better UI, there are multiple ways of creating UI from scratch by having a team with designer(s) who can create good UI or if you are a full stack developer you can develop a backend and front end as well.

Let’s use some existing templates, let’s go to google which will give you templates for free or you can buy them and they are not that costly. Let’s go to google and download a simple template

Graphical user interface, text, application, email

Description automatically generated

Access the url and proceed further to download the template and unzip the folder.

Graphical user interface

Description automatically generated

* At the moment our focus is on index.html and when you access it we can see the page in browser which will have amazing UI. We want to use this design.

Graphical user interface, text, application, email

Description automatically generated

Why we should focus on the static content - ? when you want to make the content dynamic, the content available on the page we are going to make it dynamic ex: image on the page or names or price. If you want to add one more image/place to the existing one and that’s where you will edit the database and you will add one more location and you will set the price and we will set the data as well. All this data, image, price will be coming from the database dynamically.

Graphical user interface, website

Description automatically generated

* How to use this in our project, when we focus on index.html we will need to take care of all the supporting/other static files like images/js…. as well because these are important for the layout. These files bring animation, images and we need all of them in our project.

Graphical user interface, text, application, email

Description automatically generated

* Now let’s create a new app as shown below.

Graphical user interface, text, application

Description automatically generated

* This will create a new application as shown below in our IDE, if not restart the IDE or check the folder where you have created. **We want to place our index.html in the TEMPLATES folder. Drag and drop it.**

Graphical user interface, text

Description automatically generated

* We can see that we had placed it in the templates folder, now to access this page we need to make some changes.

Text

Description automatically generated

* In travello we don’t have the URLs.py so let’s copy it from calc and paste it in the travello folder and update the file as shown below.

Text

Description automatically generated

* We only have one path which is views.index, now let’s modify the views file in travello as shown below. Copy the file from calc views and paste it in travello views file

A screenshot of a computer

Description automatically generated with medium confidence

* Now if you run the server and access the page we will still see the same old page which is picking it up from the calc.

Graphical user interface, application

Description automatically generated

* Now let’s go the main urls.py from myproject where when we are calling the home page it is navigating to calc.urls. Hence we will update it to travello.urls as shown below.

A screenshot of a computer

Description automatically generated with medium confidence

--Now if you refresh the page we will see the page as shown below which renders the index.html from travello as our home page. Inspect element the page and we can see that all the elements are missing.

Graphical user interface, application

Description automatically generated

Recap:

**Static files 2**

Now we should have a page which is not properly formatted, to have a proper layout we need css. Try inspect element and we can see that there are some things missing. We will copy the folders from the extracted template folder to the newly created static folder as shown in the below screenshot. Notice that in static folder on VSCode we have the folders being displayed

A screenshot of a computer

Description automatically generated

Now if you refresh the browser to see if we were able to render the page properly, we can notice that it is still not showing as Django should know where it has to look for the files which means it should know the path.

Graphical user interface, application, Word

Description automatically generated

To make it work we will make the changes to **settings.py** and perform the mentioned changes. 1) we will specify where we will find all the static files(we are informing Django where we will find all the static files) **STATICFILES\_DIRS** where we are joining the base path**(BASE\_DIR)** with **/static/**(where we will find all the static files).

What Django will do is it will make it simple by taking into consideration of our static files along with its location and I will take them and place it in my particular folder which is a django folder **(assets)** to improve the process. **BASE\_DIR** step

Text

Description automatically generated

Now if you refresh the page it will not work and give the same unformatted page without the supporting files.

Graphical user interface, application

Description automatically generated

In the above step we are mentioning the **assests** folder for Django. Do we have that folder in our folder structure. Wo don’t have the folder, now we have to create a folder and collect the static files with the help of below command.

Text

Description automatically generated

This command will result in creating a folder as shown with all the folders what we had placed in static folder earlier.

Graphical user interface, text, application

Description automatically generated

Now if you refresh the browser still it will not work

Graphical user interface, application, Word

Description automatically generated

First we had mentioned Django that it is responsible to handle all the files, but the moment we say Django will handle everything we also have to edit a file here, which is **index.html** which we are loading, observe the error highlighted below – not able to find the css file which is on line no 9.

Graphical user interface, text

Description automatically generated

Let’s take a look at index.html line 9. We can see we are trying to use a CSS but then it is not in a proper or expected location to access it by the index.html. so how do we provide the link here.

Text

Description automatically generated

Whenever we work with static files we have to change the URL which means we have to change the href to the Django understandable language, just before every path we need to make two changes – 1) first load the static as shown in the next screenshot

A screenshot of a computer

Description automatically generated with medium confidence

**2)** make changes to lines **{% static** ‘just before the path’ **%}** as shown below. We had configured static before if you can recollect. This change needs to be done for all the links in the index.html. yes, it is a tedious task but we have to do it everywhere

Text

Description automatically generated

Now if you refresh the page we can see that error in line 9 will disappear and the page loads with some content as shown below which changed its layout.

Graphical user interface, text, application, email

Description automatically generated

Do the same step for 2 or more lines and observe that the site gets better with its layout.

The template we are using has many images/css files hence we have to do it for all the links. But when we are working on it we will do it form the starting. Anyway let’s do it for all the hundreds of links and refresh the site.



This is a dynamic page, but it is still static. At the moment we are sending all this data from Django we can make it dynamic by sending the values from views.py which we will see in the next lecture.

Recap: we are copying a template from internet which has some static files but Django needs to handle the static files and to handle it we have to create a folder which we had named ‘static’ and pasted all our folders. Now in settings.py we mentioned from where we will get the static content and how do we refer it and we placed them in ‘assets’ folder. Updated the index.html with static tag and updated the links with load static accordingly.

