**Passing dynamic data in html part 2**

In part 1 we have seen how to pass the data dynamically and observed the changes happen in the front end. In the html we have seen the jinja tag has been used.

Let’s update the image which we haven’t done in the previous part **dest1.img =**

If you look at the **assets folder** we can see that there are images for different destinations destination1,destination2,…. So we will use first 3 images 1,2,3

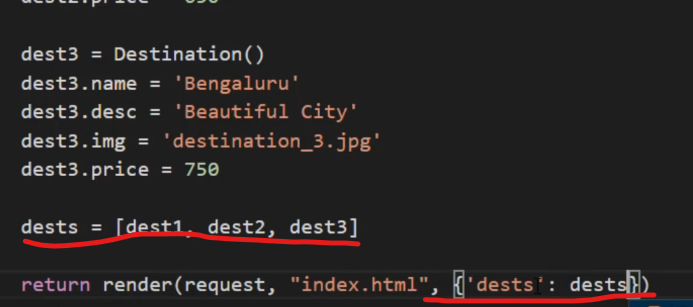
Text

Description automatically generated

Update our **views.py accordingly with the 3 objects dest1,dest2,dest3** as shown below, create a list with these 3 objects **dests = [dest1,dest2,dest3]** and update the **return key value pair to {‘dests’ : dests}**

Text

Description automatically generated



Now let’s go back to **index.html** and we need to make a change, here we are passing the data for 3 destinations. We can manually update the 3 destinations as well like we did in part 1. We know that we have the 3 destinations on the page because we are displaying them, but what if the amount of destinations we see on the screen depend on the database records(think of what if we have 10 destinations, 20 or more destinations) it is not easy task to keep on updating them every time on index.html and we might make some mistakes as well.

* Think of **using a loop**, we don’t have to worry about the destinations. Depending upon how many destinations we are passing from the database we can show them, here we want to work with 3 **so the loop should run for 3 times**.
* Let’s look at how to work on a loop here, since we know that we are going to run the loop let’s remove 2 other destinations related html from the index.html and have **ONLY ONE destination**. Refresh the page after making the changes and we will have only ONE on the page as shown below. Don’t worry data will not be shown at the moment because we have done some changes.

Graphical user interface, application, website

Description automatically generated

* Now let’s run a loop for our 3 objects in index.html using jinja format. 1) for dest in dests for starting the loop 2) endfor (ending the loop) check the below screenshot.

A screenshot of a computer

Description automatically generated with medium confidence

**Refresh the index.html** page and observe the output, we got 3 locations which means our loop is working. If you pass x number of objects in views.py we will see x number of destinations here. We are making it dynamic.

Graphical user interface, website

Description automatically generated

* We want to change data, the names, prices and descriptions. Let’s update dest1 to dest as we had used dest in our loop and save the file. Check the below screenshot for more understanding.

A screenshot of a computer

Description automatically generated with medium confidence

Refresh the index.html page and we should the output as shown below. Updated names, prices, descriptions EXCEPT image. Ignore the special offer at the moment

Graphical user interface, website

Description automatically generated

Let’s work on image, now if we go back to views.py we are also passing the img in our views.py file. We need to make some change to the image part in the index.html which means we need to make it dynamic, to make it dynamic update it as shown below

Text

Description automatically generated

Refresh the page and we will observe that the images are not visible anymore.

Graphical user interface, text, application

Description automatically generated

Inspect the element and console, we can see the below error. Image is not found 404 and the image it is trying to find is ***dest.img*** in the second line, which means the data is not getting passed.

Graphical user interface, application

Description automatically generated

The issue is in index.html image field ‘images/{{dest.img}} works inside the html, however we placed it inside the jinja format as shown below. We had done jinja inside jinja which won’t work(don’t ask me why as I too don’t know why, for the moment just know that it doesn’t work)

Text

Description automatically generated

To solve the above issue we need to perform 2 steps

1. On top of the index.html page we say static images as baseUrl – what happens is whenever you use images you simply use baseUrl and this baseUrl becomes a variable

A screenshot of a computer

Description automatically generated with medium confidence

1. In image field we remove the jinja format and we will mention baseUrl as variable and you can notice that we are writing it outside the jinja

Text

Description automatically generated

Refresh the page and we should see the page works as expected.

Graphical user interface, website

Description automatically generated

Recap: we had used for loop to give multiple iterations and every time it runs it will give different content.

In the next lecture we will look at how to address the special offer(which we will limit it to only one destination) part and pass the data from the database.