**Passing dynamic data in HTML part 1**

We are here now as shown in the below screenshot and the data is static, the images, words and prices are static. If you want to change something in future we have to edit the HTML file. We might need to update the file continuously based on the prices as they can go up and down as they depend on various factors. So without touching the html file can we change the images or prices which comes from the database.

**Graphical user interface, website

Description automatically generated**

So we need to make this page dynamic which means we need to make sure that these images or prices are coming from the database but not from the actual html page, keeping in mind that the layout, colors will be fixed and we can change it if we want to but we are not going to touch it.

Now we want to pass the data from python code, because python is a programming language in which we can do some processing and we can fetch the data from various locations like database, livestream. Once you have data in python we can pass it to html.

But how? – first of all we need to generate data in **views.py** and change the price of bali **679$ to 700$** so first we will update the price in **views.py**

Graphical user interface, application, website

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Now let’s go to index.html and find **679$ and replace it to {{price}}** which is jinja format as shown in the below screenshot

A person standing in front of a screen with text

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with medium confidence

Now if you refresh the home page we should observe the updated price for Bali as $700 which we are passing it from the actual python code

Graphical user interface, website

Description automatically generated

Now we don’t want to pass the price alone and in fact we want to pass the images, name from the python code. Think about it, there are many destinations their images, prices and every destination will have certain data which is the image, name, description and then price. We have to pass these 4 details for each destination which means we have 6 destinations and we have to pass 6\*4 = 24 fields.

Graphical user interface, application, website

Description automatically generated

We can say first destination image, name, description, and price so we have to pass 24 values in a list, what makes sense is to create a class object. Whenever we have combination of data instead of having 24 variables we can have 1 object which will hold 4 pieces of data. So we can have 6 objects for 6 destinations and each object will have 4 values(the image, name, description and price) and **YES, we can do it with the help of object oriented programming.**

To do it we will use **models.py** where we will create models, the models available in Django is for specific purpose like to connect with database and stuff(db is not in our topic at the moment so we can ignore it).

Text

Description automatically generated with medium confidence

In **models.py** we will create a class and create an object from the class. Our class name is destination 4 variables(for 4 fields) and one more for database connection which adds up to 5 variables. Database will have a primary key which is id, other variables are image, name, description, price. So we need to create a class which holds 5 variables as shown in the below screenshot and if you instantiate them or create an object we need to do it in **views.py** because that is where we will render the **index.html** page

Graphical user interface

Description automatically generated

In **views.py** we are not passing 24 values we are going to pass objects, we will create an object for **destination(dest1 for bali)** as we will do it for one destination and update the bali part. To create an object from Destination class we need to import it as shown below, and pass the **dest1** in the return as highlighted below

Text

Description automatically generated

To recollect we got an object which have no values to it, this obj has multiple variables with default values(name, price …) which we are not interested in at the moment. Now we are going to pass this to index.html

In the index.html how do you access it, let’s first find where the bali.

A screenshot of a computer

Description automatically generated with medium confidence

Now let’s make it dynamic by changing the name from **Bali to {{dest1.name}}**

A screenshot of a computer

Description automatically generated with medium confidence

Let’s check if it works, refresh our home page and we will see as shown below.

Graphical user interface, application, website

Description automatically generated

We notice that Bali disappeared from the page. The thing is we are not passing the value for the name as we had mentioned **dest1.name** for this field. So let’s fix it by making the below changes in **views.py**

A screenshot of a computer

Description automatically generated with medium confidence

Let’s refresh our home page and we can notice that Mumbai is on our home page.

Graphical user interface, application, website

Description automatically generated

Let’s update description, by making changes in **views.py** **(dest1.desc)** and **index.html(jinga format)**

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

Now if we refresh the page we can see that Mumbai and it’s description

Graphical user interface, application, website

Description automatically generated

Since we know the step update the price to some value and check the result.

* Let’s remove the 4,5 and 6 locations from the page, so that we can work on only the 3 locations and after removing the 3 locations we can see that there are only 3 locations remain as shown below.

Graphical user interface, application, website

Description automatically generated

Homework: let’s update the remaining two destinations with any of your favorite locations and update their prices, descriptions.

We will look at updating the images in our next lecture.