**Django – if statement**

We can see that for every listing we have special offer as shown below. We don’t want to provide special offer to every listing/destination. Let’s think of giving the offer only to Hyderabad or any one location.

**Graphical user interface, website

Description automatically generated**

If we see the code what we are doing is we have 3 objects/destinations created in views.py and we are simply adding them as highlighted below.

Text

Description automatically generated

In index.html we are using a loop, which will take the things from first destination to last destination. If we remove the special offer highlighted below it will be disappeared for all the destinations.

Text

Description automatically generated

A picture containing website

Description automatically generated

Remove the special offer save it and refresh the page and it will be disappeared

A bridge over a body of water

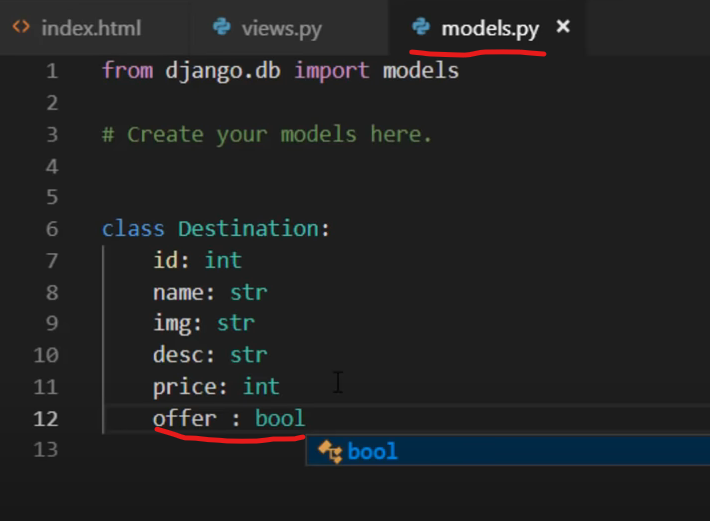
Description automatically generated with low confidence

So we need to make 3 changes

1) Go to **models.py** and specify **offer:bool**

2) Go to **views.py** and add a field dest1.offer = True , **dest2.offer = False** and **dest3.offer = False**

3) Let’s update our index.html accordingly (we will specify the condition if else)



Text

Description automatically generated

We know that by default it is true and we need not mention it separately. Otherwise you can mention **dest.offer == True** and it still works.

A screenshot of a computer

Description automatically generated with medium confidence

Now if we refresh the page we can see that the offer it applied only to the destination where it is set to true.

A collage of a bridge

Description automatically generated with low confidence

Now if you want to apply for another location we can just change the **dest1/2/3.offer = True in views.py**

A collage of a bridge

Description automatically generated with medium confidence

This is how we make things dynamic. If the data is coming from the database we don’t even have to change anything in the code. Just go to database change the offer field value from False to True or change it with the help of admin panel and it will get reflected on the front end.

Conclusion: we can use if else in the views as well not only views.py but index.html as well and it is what Django makes powerful. In the next section we will see how the data comes from the database. Let’s imagine this data is coming from the database(oracle/postgres) we will use postgres next. We will create a table with the help of models in postgres and we will connect it using ORM object relational mapper. We will map the objects and tables(relations)

**ORM (Object relational Mapping)**

What is ORM? When we think of virtual world we have data and on the other hand we have application. What is dynamic is application we use to access the data. Actually we want to use the data and we cannot do that directly. Ex: if you want to fetch some data from facebook you need to use either facebook app/web app.

Bubble chart

Description automatically generated

Imagine there is a database which is sitting somewhere and as a user if we want to access it we need an application in between and someone has to build that application.

Diagram

Description automatically generated

Let’s discuss about database – there are various types of databases and we are focusing on relational database here. If you expand your database, it will have some tables. Accounts, products, customer table

Diagram

Description automatically generated with medium confidence

Let’s take a look at what the customer table it might have id,name,add,phone num as columns. If that application is build in any of the object oriented programming maybe c#, java, python it will be having a class and that’s how we create an object.

Table

Description automatically generated

So basically to create an object you need a class. In our scenario about customer table we can create a customer class and that customer class will have 4 fields as shown below: cust\_id: ……

**Table

Description automatically generated with medium confidence**

So in database in customer table it has 4 columns and the class we have 4 properties. If we try to relate more we can have multiple objects of this class. Customer can have 5 objects, may be 10 objects depending upon the customer base for your business. Let’s say if we have 5 customers and for every customer will have different name, add, phone num and those things are stored in different objects. Every object will have it’s own data or we can also say that every object will have it's own state. State is basically the current values.

Graphical user interface, text, application, chat or text message

Description automatically generated

If you want to store this in database we will have a customer table as we know, one object represent one column. First two will be the first object data(uday’s data) and second row (rishi’s data) …..

A picture containing text, person, screenshot

Description automatically generated

How can we achieve that, how can we push the data from application to the database. That’s where we have the concept of **SQL**. We have to use SQL query as a programmer. We write the code/sql query and if you don’t want to do that(what’s the excuse here – maybe as a programmer you want to focus on application but not on the SQL language).

So you can simply say I got a class here and I don’t even want to create a table there and that table should be created automatically – yes, it is possible with the help of framework Django, hibernate it has the power to look at the class names and create a table for it. Example: class name is customer and table name will be customer

Table

Description automatically generated

Class has 4 properties and the table will have 4 columns

Table

Description automatically generated

This class depend upon how many objects you create, you create 5 objects I will have 5 rows there. The moment you create more objects and if you try to save them it will create a new row in the database automatically. That’s the power of ORM. This concept is not specific to any language we can use it with python has Django, java has hibernate. ORM is an awesome concept where you can directly create tables with the help of classes you create and you can add the data with the help of objects you create.