**ORM:-** Object Relational Mapper

First install **virtualenv** in python installed folder.Ex:- In mycomputer my python installed in

[**C:\Users\karth\AppData\Local\Programs\Python\Python310\Scripts**>] . Go to that folder in cmd prompt and install vertialenv. **Command:- pip install virtualenv**

* Createfolder with project name and go inside the folder in command prompt and create virtual environment for the project with command. **Command:-virtualenv environment\_name**
* Now activate the virtual env to create your main application or to add or delete or anything for the created application. To activate environment. **Command:- environment\_name\Scripts\activate**
* Now you can do anything to your project.\*\*You should activate environment to do any changes to the application.
* Now create website folder using the command. **Command:-django-admin startproject proj\_name**
* Goinside the project\_name container and type **Command:- python manage.py runserver** to start the applications in the project to run. To stop server press **Ctrl+c**
* Nowyou can create applications using **Command:- python manage.py startapp application\_name**

Template Tags:- we use {% %} Ex:- {% for i in range(1,10) %} {% endfor %}, {% if i > 10%} {% endif %}

Template Variables:- we use {{}} Ex:- {{id}}, {{name}}

Template Filters:- {{name | capfirst | add:’ is a good boy’ | slugify}}

**Python manage.py migrate** to migrate admin,session,content tables to sqlite3 database.

**python manage.py createsuperuser** to create user with all admin permissions.

**Username:- admin Password:-password** for sqlite3 superuser. [srikanth pass.464]

To create database table in sqlite db from django class write class in models.py file of belonging app.

Class class\_name(models.Model):

Sid=models.IntegerField(primary\_key=True)

Sname=models.CharField(max\_length=50)

def \_\_str\_\_(self): 🡪To show sname instead object field

return self.Sname

Fields:- IntegerField(),CharField(max\_length=),DateField(),EmailField(),FloatField()

Now

**Python manage.py makemigrations** :- Django will take class from model file and change into django db table.

**python manage.py migrate:- Django w**ill create table in the backend database.

If you want admin people to insert data in database. Register this class which is in models file to admin.py file. **From** models import class\_name. **admin.site.register(class\_name).**

**Day-9:-**

If you don’t want django to create additional id with primary key constraint, we should add primary key constraint for any of the column name, so django will not add additional column\_name.

If you want to add new column in the existed table which already contained data. Then we should add default value for that column, so that old record column will be updated with given default value.

**Scname=models.CharField(max\_length=50,default=’Python’)**

By default Not Null = True in the database table to change that constraint we have to type

phno= models.IntegerField(**blank=True**,**null=True**)🡪 It is ok to not fill this field, so blank is for django to accept even there is no field entered, null is for sql to set null will be true for this table.

uname=models.CharField(max\_length=50,**help\_text=’Enter your Username’**)

To tell user to what to enter in this field, we can use help\_text

courses=[(unix,UNIX),(python,PYTHON),(oracle,ORACLE),(django,DJANGO)]

cname=models.CharField(choices=courses) 🡪 It will show us the courses in the form and we can select one and submit to databasetable.

Sid=models.IntegerField(unique=True)🡪 To give unique values in this field.

**Day\_10:-**

If you want to show all the columns in admin page of the table(**class**) we created in the model.

We should write define class and type all the column names in list format and register that class\_name with the class\_name from model.

**Admin File:**

**class StudentAdmin(admin.ModelAdmin):**

**list\_display=[column\_name1,column\_name2..]**

**admin.site.register(Student,StudentAdmin)**

If you want to render data from database.

Step:1:- create model class and give permission to admin to insert data from admin page.

Step2:- create function in views to retrieve the data.

From .models import Students

def student(request):

context={‘students’: Students.objects.all()}

return render(request,’students.html’,context)

step3:-create url for the function in view

step4:- create html file with for loop for students.

**Day\_11:-**

Django Filters for Sql Students tables:-

Students.objects.all().filter(fee\_\_gte=15000)🡪It will get the records where fee >= 15000

Students.objects.all().filter(fee\_\_lte=15000)🡪It will get the records where fee <= 15000

Students.objects.all().filter(cname=’Python’)🡪get records where cname = ‘Python’

Students.objects.all().exclude(cname=’Python’)🡪get all records except cname = ‘Python’

Students.objects.all().get(cname=’Django’)🡪**If there is only one record get works**, else we get error.So **get method only works on primary key columns and unique constraint columns**.

Students.objects.all().filter(cname=’Python’,fee=15000)🡪select \* from Students where cname=’Python’ and fee=15000

Students.objects.only(‘id’,‘name’)🡪 It will get all records with only id and name column.

**python manage.py runserver portnumber 🡪** To run server in specified portnumber.[8000-8007]

**python manage.py shell 🡪** To go into python shell prompt [exit(0)🡪 to exit from shell prompt]

**python manage.py help 🡪** To get help on different commands,subcommands.

**Python manage.py dbshell 🡪** To go into db

**Python manage.py inspectdb 🡪 To** get all the tables in db [every app tables]

**Day\_12:-**

Make sure to have STATIC\_URL=’/static/’ and static app in Installed apps list in settings.py file.

Create static folder in every app and in static folder create your app name folder and in that folder save css files.

Ex:- project/mywebsite/blog/static/blog/styles.css

In html files add template tag

**{% load static %}**

**<link href= “{% static ‘blog/styles.css’ %}” type=text/css rel= “{% static stylesheet %}” >**

**Day\_13:-**

**If you want to have all template files in different apps should get the same css styles then create static files in website/project\_static folder. So django will go first to project\_static folder if not found will go to its own app static folder.**

Create Static files in the project folder for using common static files for the apps in the project.

For that first we need to create a folder in mywebsite folder mywebsite/project\_static

Now add this folder in the settings file.

**STATICFILES\_DIRS = (os.path.join(BASE\_DIR,”project\_static”),)**

Now add css file in the project\_static folder and include this link address in applications html files.

**<link href= “{% static ‘base.css’ %}” type= ‘text/css’ rel= “stylesheet“>**

**At the time of hosting website** we have to copy all the static files into one folder, create a folder with your own name under mywebsite mywebsite/hosting\_static and in settings file add

**STATIC\_ROOT = os.path.join(BASE\_DIR,’hosting\_static’)**

**python manage.py collectstatic 🡪** Now all the static files(images,styles.css,javasriptfiles) will be copied into hosting\_static folder.

Develop the html code required for all pages in the project and save it in project/templates/base.html file.

Now in the apps template files write the below code in the html file to inherit that page into the present html file.

**{% extends ‘base.html’ %}**

**N**ow split base.html file by creating bootstrap folder under template folder and save different html files like header, footer, nav bar html files. So we can include all this files in base.html file using flwng cmd.

**{% include ‘bootstrap/head.html’ %}, {% include ‘bootstrap/foot.html’ %},etc....**