Mar 30

App:

```
D:\Django_20MAR_7PM>django-admin startproject modelproject
D:\Django_20MAR_7PM>cd modelproject
D:\Django_20MAR_7PM\modelproject>py manage.py startapp testapp
```

-->Add app in settings.py

- **models.py**

```
class Employee(models.Model):
   eno = models.IntegerField()
   ename = models.CharField(max_length=30)
   esal = models.FloatField()
   eadddr = models.CharField(max_length=30)
```

Note:

This model class will be converted into database table. Django is responsible for this conversion.

table_name:appname_modelname
:testapp_employee

Fields:eno,ename,esal and eaddr and one extra field:id
Behaviours:eno is integer, ename is char.....
Model class:database table name + filed name + field behaviours

- **Converting Model class into database specific SQL code:**

Once we write Model class, we have to migrate the corresponding SQL code, for this we have to use makemiggrations.

```
D:\Django_20MAR_7PM\modelproject>py manage.py makemigrations
```

- **How to see the corresponding SQL code of migrations:**

```
D:\Django_20MAR_7PM\modelproject>py manage.py sqlmigrate testapp 0001
CREATE TABLE "testapp_employee" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "eno" integer NOT NULL, "ename" varchar(30) NOT NULL, "esal" real NOT NULL, "eadddr" varchar(30) NOT NULL);
COMMIT;
```

- **How to execute generated SQL code(migrate command)**

After generating sql code, we have to execute that sql code to create table in database. For this we have to use 'migrate' command.

```
D:\Django_20MAR_7PM\modelproject>py manage.py migrate
```

- **What is the advantage of creating tables with 'migrate' command**

If we use 'migrate' command, then all django required tables will be created in addition to our application specific tables. If we create table manually with sql code, then only our application specific table will be created and django may not work properly. Hence it is highly recommended to create tables with 'migrate' command.

- **How to check created table in Django admin interface:**

We have to register model class in 'admin.py' file

- **admin.py**

```
from django.contrib import admin
from  testapp.models import  Employee
admin.site.register(Employee)
```

- **Creation of superuser to login to admin interface**

We can create super user by using the command.

```
D:\Django_20MAR_7PM\modelproject>py manage.py createsuperuser
```

We can login to admin interface.

start server and login to admin interface by using

http://127.0.0.1:8000/admin/

Q.Difference between makemigrations and migrate?

'makemigrations' is responsible to generate SQL code for python model class where as 'migrate' is responsible to execute the SQL code so that tables will be created in the database.

- **Read data from the database and display for the end user?**

1).Start project
2).Start app
3).Add app in settings.py
4).Add database configurations in settings.py
5).Test database connection
6).Create Model class
7).Makemigrations and Migrate
8).Register model in admin.py
9).Create super user.
10).Login to admin interface and check table created or not.
11).Template file and static file and corresponding configuration in settings.py
12).view function to communicate with database and to get data and send this data to template file which is responsible to display to end user.