**DJANGO (MODEL) / DATABSE CREATION**

Model represents the database table and its through this model that our view will perform all the database operations.

Each model will represent a database table and every model class we create will inherit or extend as below:

django.db.models.Model

The highlighted is the name of the class, which will have the corresponding fields.

Using Django, we can create that database structure from our models using commands like :

make migrations 🡺 Will generate the sql code from the model class what we created.

migrations 🡺 Within the sql; using sql we can create the database tables.

From that point, we can maintain all our database tables through these two comments.

If we add a new column later on to our table model, then we can execute these command again, which will update the structure of the table.

Along with application tables, Django also creates the below other tables:

admin: For manging the admin application that comes with Django

security: These are used for authentication and authorization.

session management : store the session information

**CREATE A PROJECT, ADD AN APP IN THE PROJECT;**

PS C:\Users\kamal\django\_projects> django-admin startproject modelDemo

PS C:\Users\kamal\django\_projects> cd .\modelDemo\

PS C:\Users\kamal\django\_projects\modelDemo> py -m manage startapp hospitalApp

**TO CHECK THE DATABASE (SQLLITE3) IS WORKING;**

PS C:\Users\kamal\django\_projects\modelDemo> py -m manage shell

Python 3.12.3 (tags/v3.12.3:f6650f9, Apr 9 2024, 14:05:25) [MSC v.1938 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license" for more information.

(InteractiveConsole)

>>> from django.db import connection

>>> c = connection.cursor()

>>> exit

Use exit() or Ctrl-Z plus Return to exit

>>> ^Z

now exiting InteractiveConsole...

PS C:\Users\kamal\django\_projects\modelDemo>

A screen shot of a computer

Description automatically generated

**TO CONFIGURE MYSQL DATABASE;**

The components required are;

Engine :

Name :

User :

Password :

Host :

Port :

In settings.py file, the default is for sqlite3, if we want to change to any others;

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.sqlite3',

'NAME': BASE\_DIR / 'db.sqlite3',

}

}

change to (MySQL)

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.mysql',

'NAME': 'pythondb',

'USER': 'root',

'PASSWORD': 'Jerrick2911'

}

}

Host and Port name is not required if its localhost.

Validate the above in the Shell as below:

PS C:\Users\kamal\django\_projects\modelDemo> py -m manage shell

Python 3.12.3 (tags/v3.12.3:f6650f9, Apr 9 2024, 14:05:25) [MSC v.1938 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license" for more information.

(InteractiveConsole)

>>> from django.db import connection

>>> c = connection.cursor()

>>>

**INSTALLING MYSQL PYTHON DRIVER IN THE PROJECT;**

PS C:\Users\kamal\django\_projects\modelDemo> pip install mysqlclient

Collecting mysqlclient

Downloading mysqlclient-2.2.4-cp312-cp312-win\_amd64.whl.metadata (4.6 kB)

Downloading mysqlclient-2.2.4-cp312-cp312-win\_amd64.whl (203 kB)

---------------------------------------- 203.3/203.3 kB 1.2 MB/s eta 0:00:00

Installing collected packages: mysqlclient

Successfully installed mysqlclient-2.2.4

PS C:\Users\kamal\django\_projects\modelDemo>

**MAKEMIGRATIONS;**

PS C:\Users\kamal\django\_projects\modelDemo> py -m manage makemigrations

Migrations for 'hospitalApp':

hospitalApp\migrations\0001\_initial.py

- Create model Doctors

PS C:\Users\kamal\django\_projects\modelDemo>

The below folder is created in the project,

A screenshot of a computer

Description automatically generated which the details of 0001\_initial.py as below:

from django.db import migrations, models

class Migration(migrations.Migration):

initial = True

dependencies = [

]

operations = [

migrations.CreateModel(

name='Doctors',

fields=[

('id', models.BigAutoField(auto\_created=True, primary\_key=True, serialize=False, verbose\_name='ID')),

('drFirstName', models.CharField(max\_length=30)),

('drLastName', models.CharField(max\_length=30)),

('drDegree', models.CharField(max\_length=20)),

('drSpecilized', models.CharField(max\_length=30)),

('drExperience', models.IntegerField()),

('drDOJ', models.DateField()),

],

),

]

**SQL QUERY FOR THE ABOVE MIGRATION;**

PS C:\Users\kamal\django\_projects\modelDemo> py -m manage sqlmigrate hospitalApp 0001

--

-- Create model Doctors

--

CREATE TABLE `hospitalApp\_doctors` (`id` bigint AUTO\_INCREMENT NOT NULL PRIMARY KEY, `drFirstName` varchar(30) NOT NULL, `drLastName` varchar(30) NOT NULL, `drDegree` varchar(20) NOT NULL, `drSpecilized` varchar(30) NOT NULL, `drExperience` integer NOT NULL, `drDOJ` date NOT NULL);

PS C:\Users\kamal\django\_projects\modelDemo>

**MIGRATE;**

PS C:\Users\kamal\django\_projects\modelDemo> py -m manage migrate

Operations to perform:

Apply all migrations: admin, auth, contenttypes, hospitalApp, sessions

Running migrations:

Applying contenttypes.0001\_initial... OK

Applying auth.0001\_initial... OK

Applying admin.0001\_initial... OK

Applying admin.0002\_logentry\_remove\_auto\_add... OK

Applying admin.0003\_logentry\_add\_action\_flag\_choices... OK

Applying contenttypes.0002\_remove\_content\_type\_name... OK

Applying auth.0002\_alter\_permission\_name\_max\_length... OK

Applying auth.0003\_alter\_user\_email\_max\_length... OK

Applying auth.0004\_alter\_user\_username\_opts... OK

Applying auth.0005\_alter\_user\_last\_login\_null... OK

Applying auth.0006\_require\_contenttypes\_0002... OK

Applying auth.0007\_alter\_validators\_add\_error\_messages... OK

Applying auth.0008\_alter\_user\_username\_max\_length... OK

Applying auth.0009\_alter\_user\_last\_name\_max\_length... OK

Applying auth.0010\_alter\_group\_name\_max\_length... OK

Applying auth.0011\_update\_proxy\_permissions... OK

Applying auth.0012\_alter\_user\_first\_name\_max\_length... OK

Applying hospitalApp.0001\_initial... OK

Applying sessions.0001\_initial... OK

PS C:\Users\kamal\django\_projects\modelDemo>

**TABLES CREATED IN MYSQL FOR THE PYTHON PROJECT;**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**DISPLAY THE VALUES IN THE VIEW;**

A screenshot of a computer

Description automatically generated

**CREATING THE SUPERUSER;**

PS C:\Users\kamal\django\_projects\modelDemo> py -m manage createsuperuser

Username (leave blank to use 'kamal'): augusta

Email address: augustaraju@hotmail.com

Password:

Password (again):

Superuser created successfully.

PS C:\Users\kamal\django\_projects\modelDemo>

Use this link of the browser : [Log in | Django site admin](http://localhost:8000/admin/login/?next=/admin/)

Login in with superuser details.

**Output**

A screenshot of a computer

Description automatically generated

**DISPLAY THE CLASS / MODEL IN THE ABOVE ADMIN PAGE;**

(ADDING MODELS TO ADMIN UI)

Code to be added in admin.py

from django.contrib import admin

from hospitalApp.models import Doctors

# Register your models here.

admin.site.register(Doctors)

A screenshot of a computer

Description automatically generated

If we click on the Doctors Object (1), then we can see the details, (tab of right hand side) A screenshot of a computer

Description automatically generated

Here we can update / add / remove this (CRUD) process can be done here.

A screenshot of a computer

Description automatically generated

A screenshot of a medical application

Description automatically generated

This will bring with the records added in the UI of admin, with user augusta

A screenshot of a computer

Description automatically generated

By changing the code to below, instead of showing the model (Doctors) as object, this will display all the fields, which we require in the grid format. The highlighted (list\_display), in the code below has to the same variable, which cant be changed.

**Code:**

from django.contrib import admin

from hospitalApp.models import Doctors

# Register your models here.

class DoctorsAdmin(admin.ModelAdmin):

list\_display=[

'drFirstName',

'drLastName',

'drDegree',

'drSpecilized',

'drExperience',

'drDOJ'

]

admin.site.register(Doctors, DoctorsAdmin)

**Output in Admin UI**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated