**Objective 4 Report**

**Objective 4 - Users can share some form of multimedia content which is stored and hosted on your server**

* *Describe how you completed the objectives of the assignment, how your software is structured, and why you made the design decisions you made. A reader should have a good understanding of your software by reading this section without looking at a single line of your code*

**/forums/0001\_initial.py**

This file is related to Django’s engine called ORM Object Relational Mapper which translates python code into SQL code. Django’s ORM provides a way for the data in Django’s database to be accessed. All of the code here is created by Django. These are the “initial migrations” for our application that are following the order of how our model.py is structured.

Libraries and their links :

**Settings** -> We used AUTH\_USER\_MODEL

<https://docs.djangoproject.com/en/3.2/ref/settings/>

Models are used by Django to store an application’s data. This is used to return a string that will give the position of the User Model ( ‘auth.User’ which will then call get\_use\_model() to return the user model class that is specified. The user model contains fields that will be referencing the other models that are in our application. We are using this because it’s involved with migrations which is Django’s way of being able to make changes to our models such as adding fields, deleting models in our Django database.

**/Migrations** ->

<https://docs.djangoproject.com/en/3.2/topics/migrations/>

<https://github.com/django/django/blob/main/django/db/migrations/migration.py>

In our case, we are using migrations library and the function swappable.dependencies. We have to use this because Django creates a special key with the User Model and the AUTH\_USER\_MODEL and the relationship of each migration will be a reference to the settings in the AUTH\_USER\_MODEL and not the model directly. Django uses models to create tables, their fields and constraints within each model. Each model that we are creating with the function CreateModel function from the migrations library.

**CreateModel** ->

<https://docs.djangoproject.com/en/3.2/ref/migration-operations/>

Django uses models to create tables, their fields, and constraints associated with a model to store user data. Each model maps to a single database table. With this library, we are using the Create Model to create a table in the database to match with the fields given.

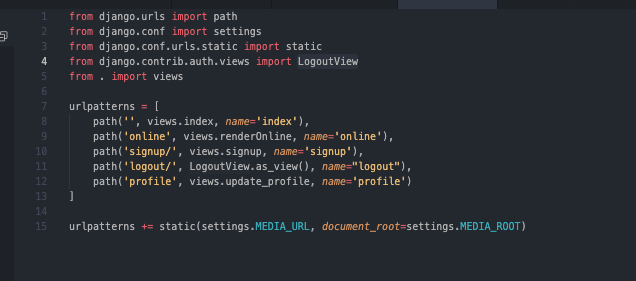
**Django.db.models.deletion (Used Set\_NULL)** Line 58 ->

<https://github.com/django/django/blob/main/django/db/models/deletion.py>

This function works by deleting the referenced object it corresponds to. So if it was on ‘updated\_by’ it would set the reference to NULL. It will replace the referenced object with “NULL” which is what SQL uses. Django will store the NULL in place of what was previously referenced there. This acts sort of like “deleting the data” in a more secure manner to allow for searches to be “User not found” instead of an error.

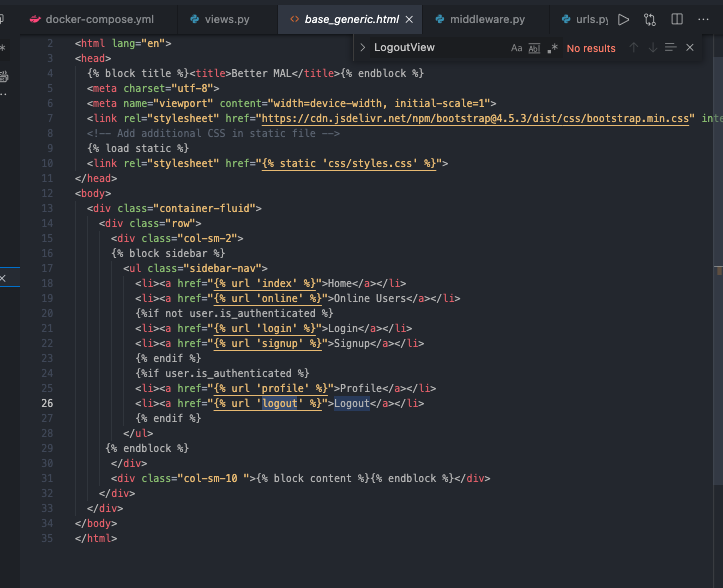
**/forum/urls.py**

*\* Path library is already talked about previously*



**What is happening here?**

Django utilizes the use of views which is a web request that generates some sort of web response. In our example, we are using the library path which is taking in a (route,view,kwargs,name). A route is the path that we want to have things rendered to. The view is a function that we have created in another document called view.py and the name is a custom name for a url pattern. A url pattern name is used so that our html will render the correct url in our templates.

****

**Libraries Used :**

**django.contrib.auth.views.logoutView :**

[**https://github.com/django/django/blob/7582d913e7db7f32e4cdcfafc177aa77cbbf4332/django/contrib/auth/views.py#L123**](https://github.com/django/django/blob/7582d913e7db7f32e4cdcfafc177aa77cbbf4332/django/contrib/auth/views.py#L123) Line : 111

* This view is from the django library which will render a logged\_out html page for the user when they click log out. It uses a variable called REDIRECT\_FIELD\_NAME which is bound in django.contrib.auth which binds REDIRECT\_FIELD\_NAME to ‘next’. ‘Next’ is used to specify the URL to be redirected to after the page. Since for logout it is NONE, then it will redirect to the logout.html we have in the template

**django/conf/urls/static.py :**

[**https://github.com/django/django/blob/7582d913e7db7f32e4cdcfafc177aa77cbbf4332/django/conf/urls/static.py**](https://github.com/django/django/blob/7582d913e7db7f32e4cdcfafc177aa77cbbf4332/django/conf/urls/static.py)Line: 10

* This returns a URL pattern for serving files in debug mode. We didn’t really use this because it’s associated for images during deployment aka for the admin
* This static library is used for when we are serving static files in development, because when we use the settings library and append MEDIA\_URL, we want to save the files for deployment.

**django/conf/settings.py /** *Not a library but is involved with settings.py*

* We are using these libraries for MEDIA\_ROOT and MEDIA\_URL
* MEDIA\_ROOT is also for stored files. MEDIA\_ROOT is a filesystem path to the directory that will hold all user-uploaded files. Kind of like the master branch while media\_url is just a branch.
* MEDIA\_URL is a url under MEDIA\_ROOT that handles stored files. These files are configured during deployment.

**/forum/apps.py**

**Github Library :**

[**https://github.com/django/django/blob/main/django/contrib/auth/apps.py**](https://github.com/django/django/blob/main/django/contrib/auth/apps.py)

**More Documentation :** [**https://docs.djangoproject.com/en/3.2/ref/applications/**](https://docs.djangoproject.com/en/3.2/ref/applications/)

The purpose of this file is to help developers include any configuration needed when creating a Django project. This way, when our application loads, AppConfig will specify our app to be “forum” so that when our INSTALLED\_APPS are called, we can make use of the name “forumConfig” without having to change the INSTALLED\_APP settings manually. This is used to promote easier configuration by changing the class in the INSTALLED\_APPS settings.

**/forums/forms.py**

This part of our project deals with the database and how Django utilizes forms as a container for the user to send data to our application. Django provides a helper class that allows us to create a Form class from an already finished Django mode. Fields are attributes that are associated with each model. If you look at our code, we have a UserForm implemented from the Django library .forms

**Library Used:**

from django.contrib.auth import get\_user\_model

**Github Library to get\_user\_model :** <https://github.com/django/django/blob/c240ceea7d88c6a8058dcacb37356c93e0a3618f/django/contrib/auth/__init__.py#L155>

This library is used because we need to do an authentication process before we can retrieve the User’s profile. We will be