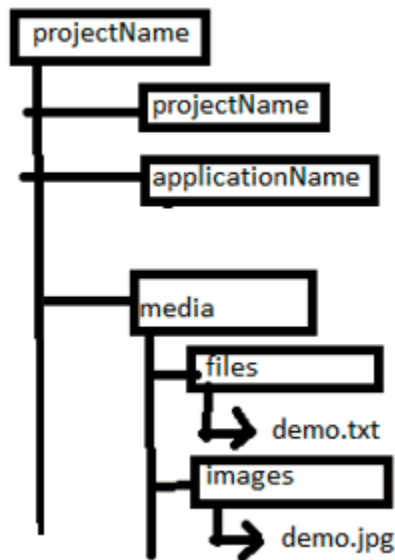


Uploading the Images and Files Using Django Templates :

- If you want to add Images and Files in django , we need to install "pillow" module by using pip comand. So open command prompt and install pillow module.

```
C:\Users\Lenovo> pip install pillow
```

- Create one "media" folder inside Base or root project directory for storing media files and images.
For example:



- Configuring the media files settings variables code in the settings.py file.

```
MEDIA_URL = '/media/'
```

```
MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
```

Here **MEDIA_ROOT** is used to represents the Server path to store media files or media images in the computer. **MEDIA_URL** is the reference URL for browser to access the media files or media images over Http

- Open **urls.py** and create **media urls** along with required urls.

Note: In the urls.py we should create the media files urls and add with existing urls like bellow,

```
from django.conf import settings
```

```
from django.conf.urls.static import static
```

```
urlpatterns = [
```

```
    -----
```

```
    -----
```

```
    -----
```

```
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

Create a Project for working with Images and Files using Django

Step: Now create required models with required model fields along with image and file fields also.

models.py

```
from django.db import models
class Hotel(models.Model):
    name = models.CharField(max_length=50)
    address = models.TextField()
    image = models.ImageField(upload_to='images/')
    file = models.FileField(upload_to='files/')

    def __str__(self):
        return self.name

    class Meta:
        db_table = 'hotel'
```

Step: Now create the required ModelForm class for Hotel model in forms.py file

forms.py

```
from django import forms
from hotelapp.models import Hotel

class HotelModelForm(forms.ModelForm):
    class Meta:
        model = Hotel
        fields = '__all__'
```

Step: Now open views.py and create required views logics.

```
from django.shortcuts import render, redirect
from django.http import HttpResponseRedirect
from hotelapp.models import Hotel
from hotelapp.forms import HotelModelForm

def HotelCreateView(request):
    if request.method == 'POST':
        form = HotelModelForm(request.POST, request.FILES)
        if form.is_valid():
            form.save()
            return redirect(successView)
            #return HttpResponseRedirect('Hotel date created succesfully!')
        else:
            return HttpResponseRedirect('Hotel date not created succesfully!')
    else:
        form = HotelModelForm()
```

```

context = {"form" : form}
return render(request, 'hotelapp/hotel_create.html', context)

```

```

def successView(request):
    return HttpResponse('Form data saved successfully')

```

Note:

- Whenever the `HotelCreateView` hits and that request is POST, we are creating an instance of model form like `form = HotelModelForm(request.POST, request.FILES)`. Here image will be stored under `request.FILES` one.
- If it is valid, then save into the database and redirects to **success url** which indicates successful uploading of the image.
- If the method is not POST then we are rendering request to html template for creating Hotel information.

Step : Now create required template file name like `templates/hotel_create.html`

```

<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css"
rel="stylesheet">
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"></script>
    <title>Create Hotel</title>

    <style>
      form{
        border: 5px solid darkcyan; background-color: yellow; margin:auto 0;
      }
      td, th{ padding: 10px 20px;}
    </style>
  </head>
  <body>
    <div class="container">
      <div class="row">
        <div class="col-md-3"></div>
        <div class="col-md-6">
          <form method="POST" enctype="multipart/form-data">
            <h3 class="bg-success">Welcome to Django Media Files Concept</h3>
            {% csrf_token %}
            <table>
              {{ form }}
              <tr>
                <th></th>
                <td><input type="submit" value="Create Hotel"
                  class="btn btn-success"></td>
              </tr>
            </table>
          </form>
        </div>
        <div class="col-md-3"></div>
      </div>
    </div>
  </body>
</html>

```

```
</body>
</html>
```

- Here we need to add **enctype="multipart/form-data"** values to the <form> tag when our "form object" is having fields related to images and files.

Step: Now create required media urls inside project level urls.py file to work with media files concept.
Projectlevel/urls.py

```
from django.contrib import admin
from django.urls import path, include
```

```
from django.conf import settings
from django.conf.urls.static import static
```

```
urlpatterns = [
    path('admin/', admin.site.urls),

    path('', include('hotelapp.urls')),
```

```
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

Step: Now create urls.py file inside hotelapp for executing required views.
hotelapp/urls.py

```
from django.urls import path
from hotelapp import views
```

```
urlpatterns = [
    path('', views.HotelCreateView, name='hotel_create'),
    path('success/', views.successView, name='success'),
]
```

Step: Now Create a view to access Images from database and display on Templates
views.py

```
def AccesingHotelView(request):
    hotel_list = Hotel.objects.values()
    context = {
        "hotel_list" : hotel_list,
        "message" : "Hotels information not available"
    }
    return render(request, 'hotelapp/hotel_list.html', context)
```

Step: Now open urls.py file and create url inside hotelapp for executing required view.
hotelapp/urls.py

```
from django.urls import path
from hotelapp import views
```

```
urlpatterns = [
    '-----',
    path('hotels/', views.AccessingHotelView, name='hotel_list'),
]
```

Step : Now create required template file name like `templates/hotel_list.html`

`templates/hotel_list.html`

```
<!doctype html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <link
href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css"
rel="stylesheet">
        <script
src="https://cdn.jsdelivrivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"></scri
pt>
        <title>Diaplay Hotels</title>
    </head>
    <body>
        <div class="container">
            {% if hotel_list %}
                <ul>
                    {% for hotel in hotel_list %}
                        <li>Hotel Name : {{ hotel.name }}</li>
                        <li>Hotel Address : {{ hotel.address }}</li>
                        <li>Hotel Image : {{ hotel.image }}</li>
                        <li>Hotel File : {{ hotel.files }}</li>
                        <br><hr>
                    {% endfor %}
                </ul>
            {% else %}
                <h1>{{ message }}</h1>
            {% endif %}
        </div>

        <div class="container">
            <h2>Accessing images from media folder</h2>
            

            <h3>Accssing images from Context object display them. </h3>
            {% for hotel in hotel_list %}
                
            {% endfor %}

        </div>
    </body>
</html>
```

Step: Now execute the server and send requested url from browser and get response.