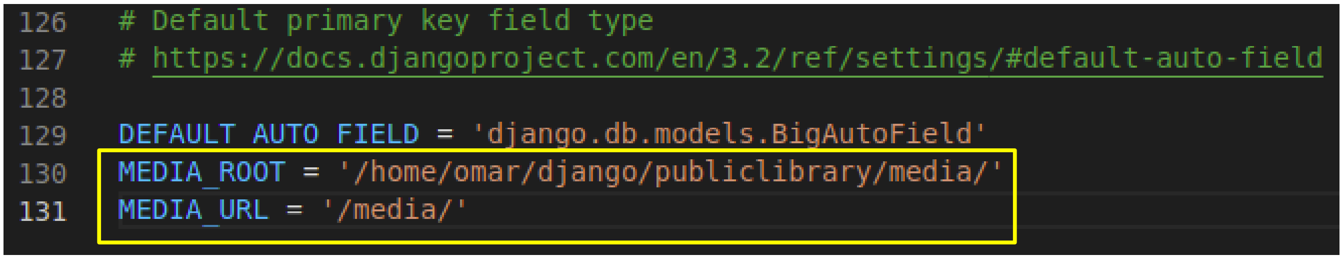
How to deal with images

1. first, setup some files(setting.py and urls.py)

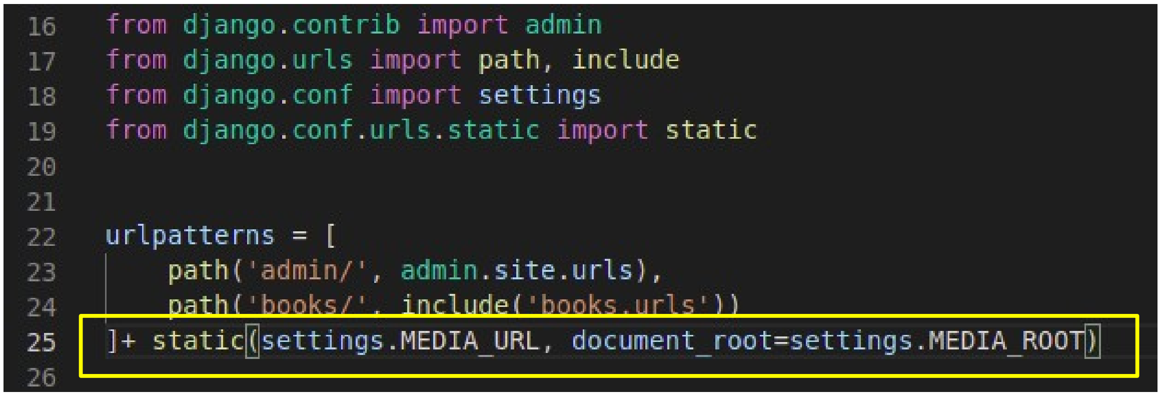
- Add MEDIA_ROOT and Media_URL to setting.py file :



```
# Default primary key field type
# https://docs.djangoproject.com/en/3.2/ref/settings/#default-auto-field

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
MEDIA_ROOT = '/home/omar/django/publiclibrary/media/'
MEDIA_URL = '/media/'
```

- Add the url path in urls.py as follows:

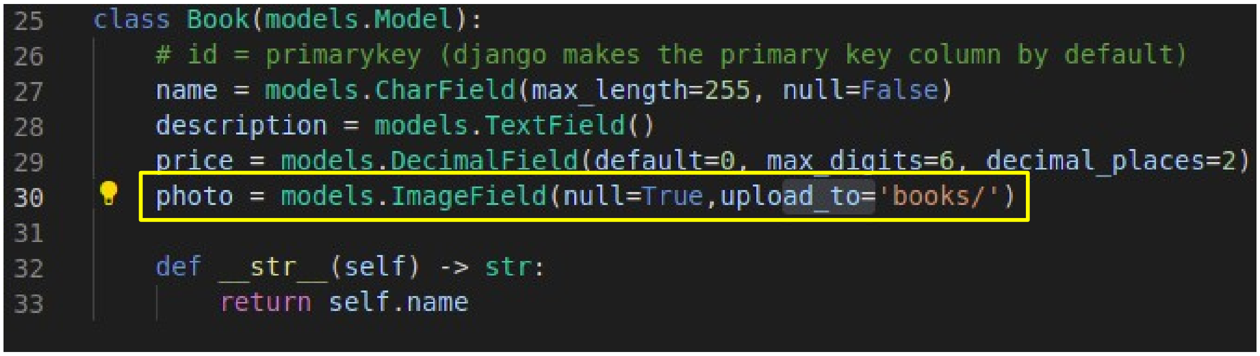


```
from django.contrib import admin
from django.urls import path, include
from django.conf import settings
from django.conf.urls.static import static


urlpatterns = [
    path('admin/', admin.site.urls),
    path('books/', include('books.urls'))
]+ static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)

```

2. Add your image object to the database interface class inside models.py:

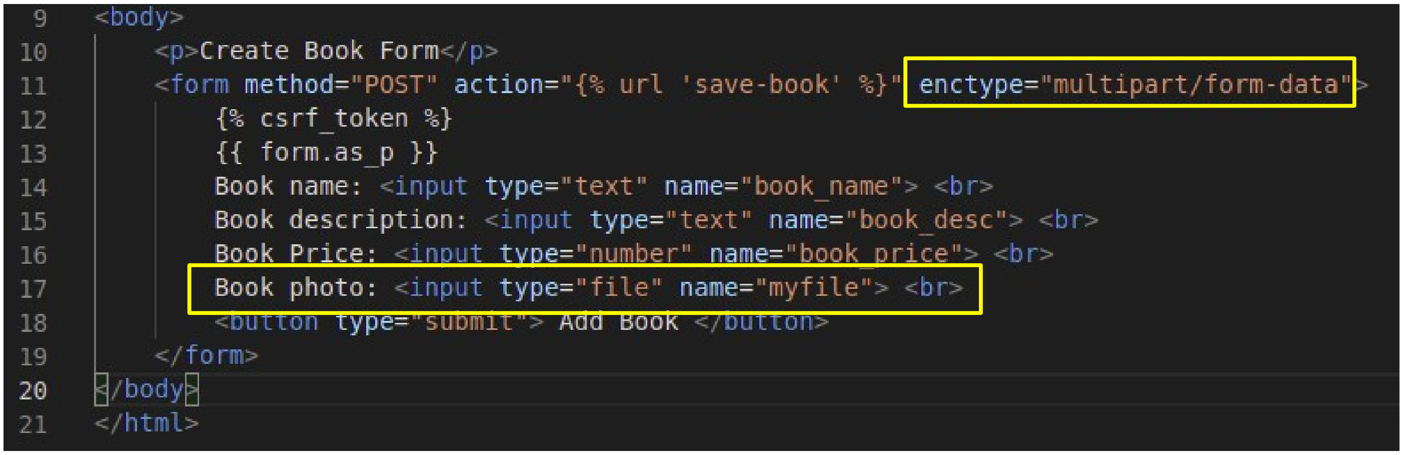


```
class Book(models.Model):
    # id = primarykey (django makes the primary key column by default)
    name = models.CharField(max_length=255, null=False)
    description = models.TextField()
    price = models.DecimalField(default=0, max_digits=6, decimal_places=2)
    photo = models.ImageField(null=True,upload_to='books/')

    def __str__(self) -> str:
        return self.name
```

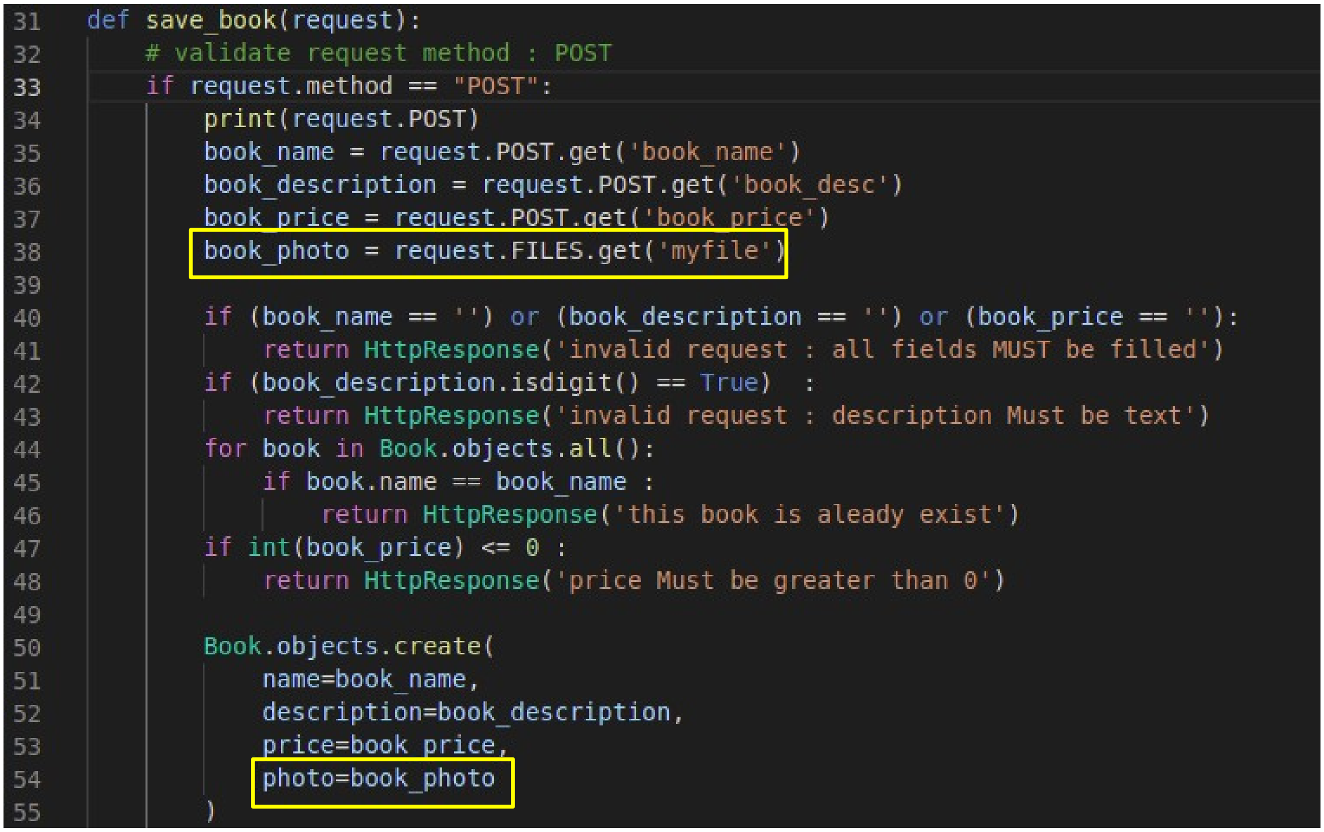
- ✔ do not forget to make sure that all this directories exist. For me I made (media) directory inside project directory then made another directory (books) inside it.

3. In the template that creates new book, add new input to make the user able to browse his file system to get his photo.



```
<body>
    <p>Create Book Form</p>
    <form method="POST" action="{% url 'save-book' %}" enctype="multipart/form-data">
        {% csrf_token %}
        {{ form.as_p }}
        Book name: <input type="text" name="book_name"> <br>
        Book description: <input type="text" name="book_desc"> <br>
        Book Price: <input type="number" name="book_price"> <br>
        Book photo: <input type="file" name="myfile"> <br>
        <button type="submit"> Add Book </button>
    </form>
</body>
</html>
```

4. Then go to views.py to save this photo to database:

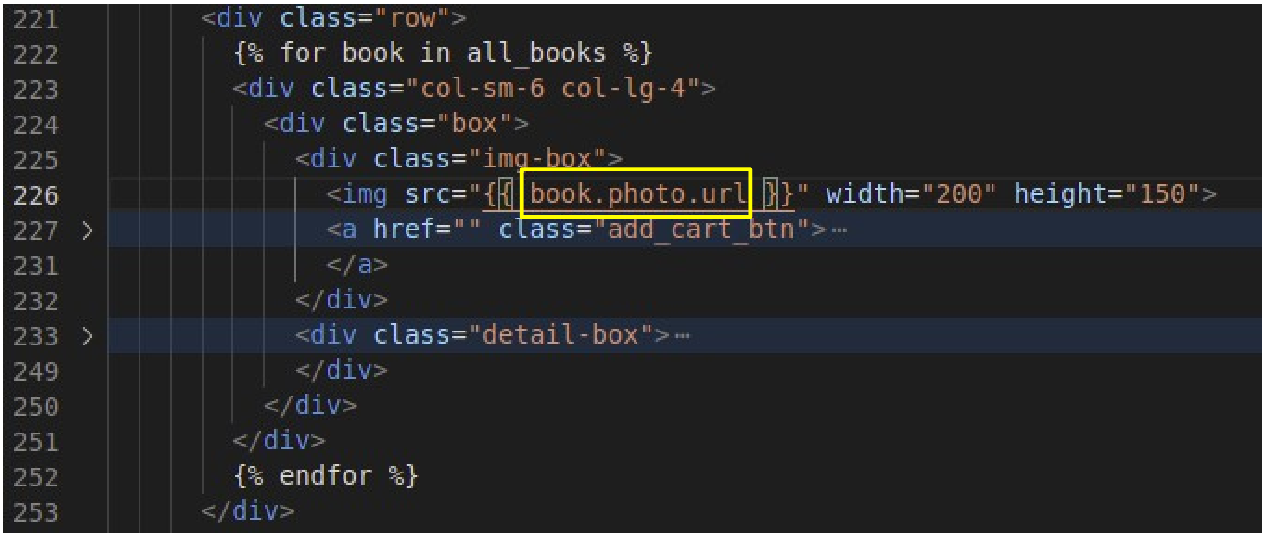


```python
def save_book(request):
    # validate request method : POST
    if request.method == "POST":
        print(request.POST)
        book_name = request.POST.get('book_name')
        book_description = request.POST.get('book_desc')
        book_price = request.POST.get('book_price')
        book_photo = request.FILES.get('myfile')

        if (book_name == '') or (book_description == '') or (book_price == ''):
            return HttpResponse('invalid request : all fields MUST be filled')
        if (book_description.isdigit() == True)  :
            return HttpResponse('invalid request : description Must be text')
        for book in Book.objects.all():
            if book.name == book_name :
                return HttpResponse('this book is aleady exist')
        if int(book_price) <= 0 :
            return HttpResponse('price Must be greater than 0')

        Book.objects.create(
            name=book_name,
            description=book_description,
            price=book_price,
            photo=book_photo
        )
```

5. Finally, if you want to show the photo in your template, just pass its url to Html file.



```html
<div class="row">
  {% for book in all_books %}
  <div class="col-sm-6 col-lg-4">
    <div class="box">
      <div class="img-box">
        <img src="{{ book.photo.url }}" width="200" height="150">
        <a href="" class="add_cart_btn">⋯
        </a>
      </div>
      <div class="detail-box">⋯
      </div>
    </div>
  </div>
  {% endfor %}
</div>
```