Note – I have since decided that the blog would be too much for my scope and removed the code. I can still be found on the 'blog' branch or in previous commits.

Products > Model.py

The purpose of the function was to add a product and create a blog instance at the same time. I wanted to fill the blog fields in with the details from the form.

I tried to fill the image fields (which are foreign keys) with the product_id.

This is my code.

```python
def add_product(request):
    """ Add a product to the store """
    if request.method == 'POST':
        form = ProductForm(request.POST, request.FILES)
        if form.is_valid():
            product = form.save()
            # http://www.learningaboutelectronics.com/Articles/How-to-retrieve-data-from-a-Django-form-Python.php#:~
            sku = form.cleaned_data.get("sku")
            title = form.cleaned_data.get("name")
            retreive_product = get_object_or_404(Product, sku=sku)
            Blog.objects.create(title=title, sku=sku, image_mobile_url=retreive_product.image_url_mobile)
            return redirect(reverse('products'))
        else:
```

I got this error.

AttributeError at /products/add/

'Product' object has no attribute 'product_id'

Request Method:	POST
Request URL:	http://localhost:8000/products/add/
Django Version:	3.1.1
Exception Type:	AttributeError
Exception Value:	'Product' object has no attribute 'product_id'
Exception Location:	/workspace/ms4/products/views.py, line 90, in add_product
Python Executable:	/home/gitpod/.pyenv/versions/3.8.6/bin/python3
Python Version:	3.8.6
Python Path:	['/workspace/ms4', '/home/gitpod/.pyenv/versions/3.8.6/lib/python38.zip', '/home/gitpod/.pyenv/versions/3.8.6/lib/python3.8', '/home/gitpod/.pyenv/versions/3.8.6/lib/python3.8/lib-dynload', '/workspace/.pip-modules/lib/python3.8/site-packages', '/home/gitpod/.pyenv/versions/3.8.6/lib/python3.8/site-packages']
Server time:	Sun, 25 Oct 2020 22:18:38 +0000

Next, I tried to retrieve the product and pull it's product_id to assign it to the blog image_url_mobile field. This is my code.

```python
def add_product(request):
    """ Add a product to the store """
    if request.method == 'POST':
        form = ProductForm(request.POST, request.FILES)
        if form.is_valid():
            product = form.save()
            # http://www.learningaboutelectronics.com/Articles/How-to-retrieve-data-from-a-Django-
            sku = form.cleaned_data.get("sku")
            title = form.cleaned_data.get("name")
            image_mobile_url = product.product_id
            Blog.objects.create(title=title, sku=sku, image_url_mobile=image_mobile_url)

            return redirect(reverse('products'))
        else:
            messages.error(request, 'Failed to add product. Please ensure the form is valid.')
    else:
        form = ProductForm()
```

I got this error.

ValueError at /products/add/

Cannot assign "'https://imagizer.imageshack.com/v2/640x480q90/924/yyf0l1.jpg'": "Blog.image_mobile_url" must be a "Product" instance.

Request Method:	POST
Request URL:	http://localhost:8000/products/add/
Django Version:	3.1.1
Exception Type:	ValueError
Exception Value:	Cannot assign "'https://imagizer.imageshack.com/v2/640x480q90/924/yyf0l1.jpg'": "Blog.image_mobile_url" must be a "Product" instance.
Exception Location:	/workspace/.pip-modules/lib/python3.8/site-packages/django/db/models/fields/related_descriptors.py, line 215, in __set__
Python Executable:	/home/gitpod/.pyenv/versions/3.8.6/bin/python3
Python Version:	3.8.6
Python Path:	['/workspace/ms4', '/home/gitpod/.pyenv/versions/3.8.6/lib/python38.zip', '/home/gitpod/.pyenv/versions/3.8.6/lib/python3.8', '/home/gitpod/.pyenv/versions/3.8.6/lib/python3.8/lib-dynload', '/workspace/.pip-modules/lib/python3.8/site-packages', '/home/gitpod/.pyenv/versions/3.8.6/lib/python3.8/site-packages']
Server time:	Sun, 25 Oct 2020 22:30:03 +0000

I Googled what it meant that it had to be an instance and here is the reference.

https://stackoverflow.com/questions/51380780/django-foreign-key-cannot-assign-must-be-a-instance

I change the code to this. I realised that I could just assign the entire product entry and the foreign key would sort out the details. The code below works and the added blogs now show with an image.

```python
def add_product(request):
    """ Add a product to the store """
    if request.method == 'POST':
        form = ProductForm(request.POST, request.FILES)
        if form.is_valid():
            product = form.save()
            # http://www.learningaboutelectronics.com/Articles/How-to-retrieve-data-from-a-Django
            sku = form.cleaned_data.get("sku")
            title = form.cleaned_data.get("name")
            Blog.objects.create(title=title, sku=sku)
            retreive_blog = get_object_or_404(Blog, sku=sku)
            retreive_prod product _object_or_404(Product, sku=sku)
            retreive_blog.product = retreive_product
            retreive_blog.image_mobile_url = retreive_product
            retreive_blog.image_desktop_url = retreive_product
            retreive_blog.save()
            return redirect(reverse('products'))
        else:
            messages.error(request, 'Failed to add product. Please ensure the form is valid.')
```