

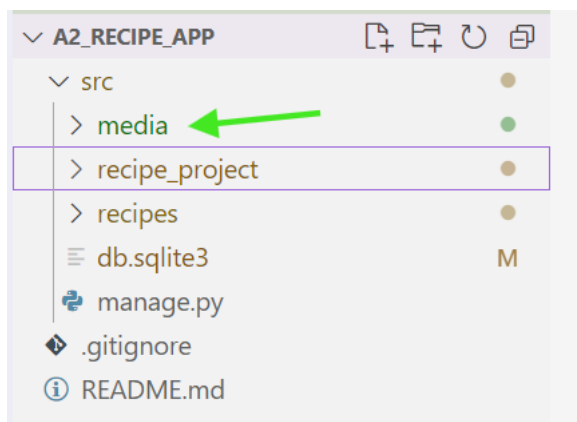
Task 2.5

Model/Table Attributes That Will Be Changed

For this task, I've updated the models.py file to include a 'pic' attribute in the table. This will allow users to view and include images with their recipes.

To accomplish this, I completed the following steps/alterations to the code:

1. Create media folder in root directory



2. Specify MEDIA_URL and MEDIA_ROOT in settings.py file

```
STATIC_URL = 'static/'

# Media files
# User-generated content

MEDIA_URL = '/media/'
MEDIA_ROOT = BASE_DIR / 'media'
```

3. Specify URL-View Mapping

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

# Include package allows you to use the include () function that will link the urls.py file in the app to the main urls.py file (this one)
from django.urls import include

# Settings allows you to access the MEDIA_URL and MEDIA_ROOT variables
from django.conf import settings

# Static provides access to the Django helper function static(), which allows you to create URLs from local folder name
from django.conf.urls.static import static

urlpatterns = [
    path('admin/', admin.site.urls), # URL pattern for admin site.
    path('', include('recipes.urls')), # Include recipe app URLs as the root URLconf.
    path("recipes/", include("recipes.urls")), # Explicit URL pattern for recipe app URLs.
]

# Extend the urlpatterns param to include the media info
urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

4a. Install Pillow

```
C:\Users\cpark\OneDrive\Documents\RockyCodingBootcamp\PythonScripts\src>workon web-dev
(web-dev) C:\Users\cpark\OneDrive\Documents\RockyCodingBootcamp\PythonScripts\src>pip install pillow
Collecting pillow
  Downloading pillow-10.4.0-cp38-cp38-win_amd64.whl.metadata (9.3 kB)
Downloading pillow-10.4.0-cp38-cp38-win_amd64.whl (2.6 MB)
2.6/2.6 MB 13.3 MB/s eta 0:00:00
Installing collected packages: pillow
Successfully installed pillow-10.4.0

(web-dev) C:\Users\cpark\OneDrive\Documents\RockyCodingBootcamp\PythonScripts\src>
```

(screenshot is from installation on a test project)

4b. Add pic attribute to the model

```
# Define class(table) and inherit from models.Model for basic functionality and attributes
class Recipe(models.Model):

    # Define attributes/columns in the table
    name= models.CharField(max_length=120)
    cooking_time= models.FloatField(help_text='minutes')
    ingredients= models.CharField(max_length=400, help_text='Ingredients must be separated by commas.')
    pic = models.ImageField(upload_to='recipes', default='no_picture.jpg')
```

Frontend Inspirations

I was mostly inspired by other student submissions, like the ones from [cjhart34](#), [Christian-Horban](#) and [TubaJordan](#). They created clean, user-friendly designs that make it easy to tell what kind of app it is and offer clear calls-to-action for the user. They created frontends that are easy to navigate and use, which I'm hopeful I can do as well once I include things like recipe search, user login, a

navigation bar, etc. For my recipe list and detail cards, I tried to keep them clean, easy to read and with enough contrast against the background image.

