

RECAP

Topics Covered:

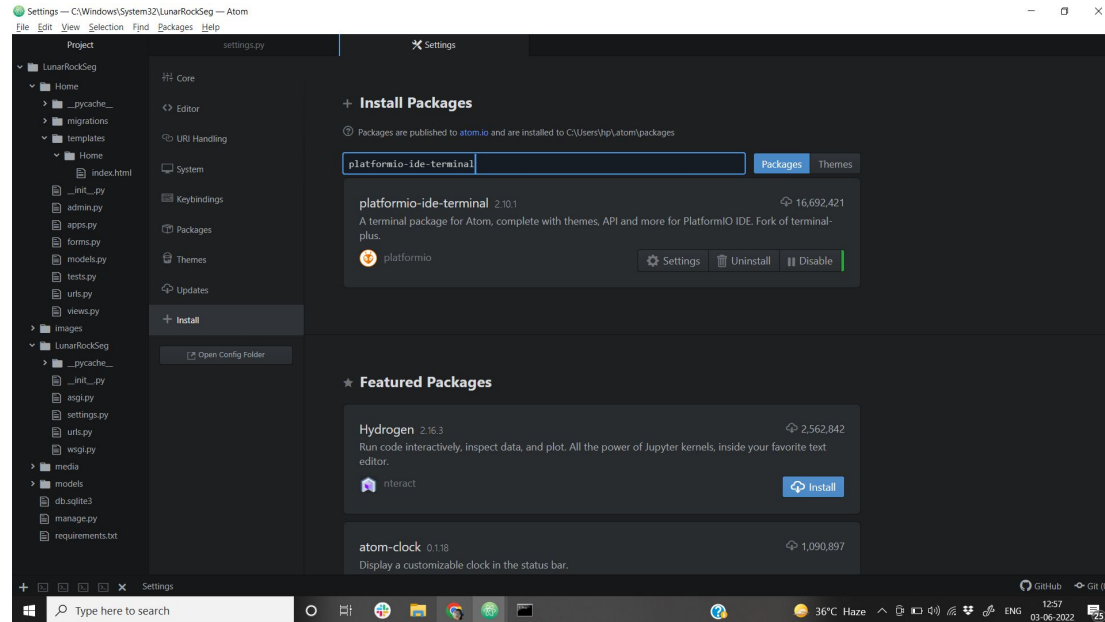
- ML Pipeline to segment Lunar Dataset Images
- Advanced ML Pipeline with `segmentation_models` and callbacks
- Best practices for image segmentation

Today we will learn

- Creating a Django Project and **Deploying** our trained model in it!

Install Atom and platformio-ide-terminal

- After installing Atom, go to File > Settings > Install > type: platformio-ide-terminal



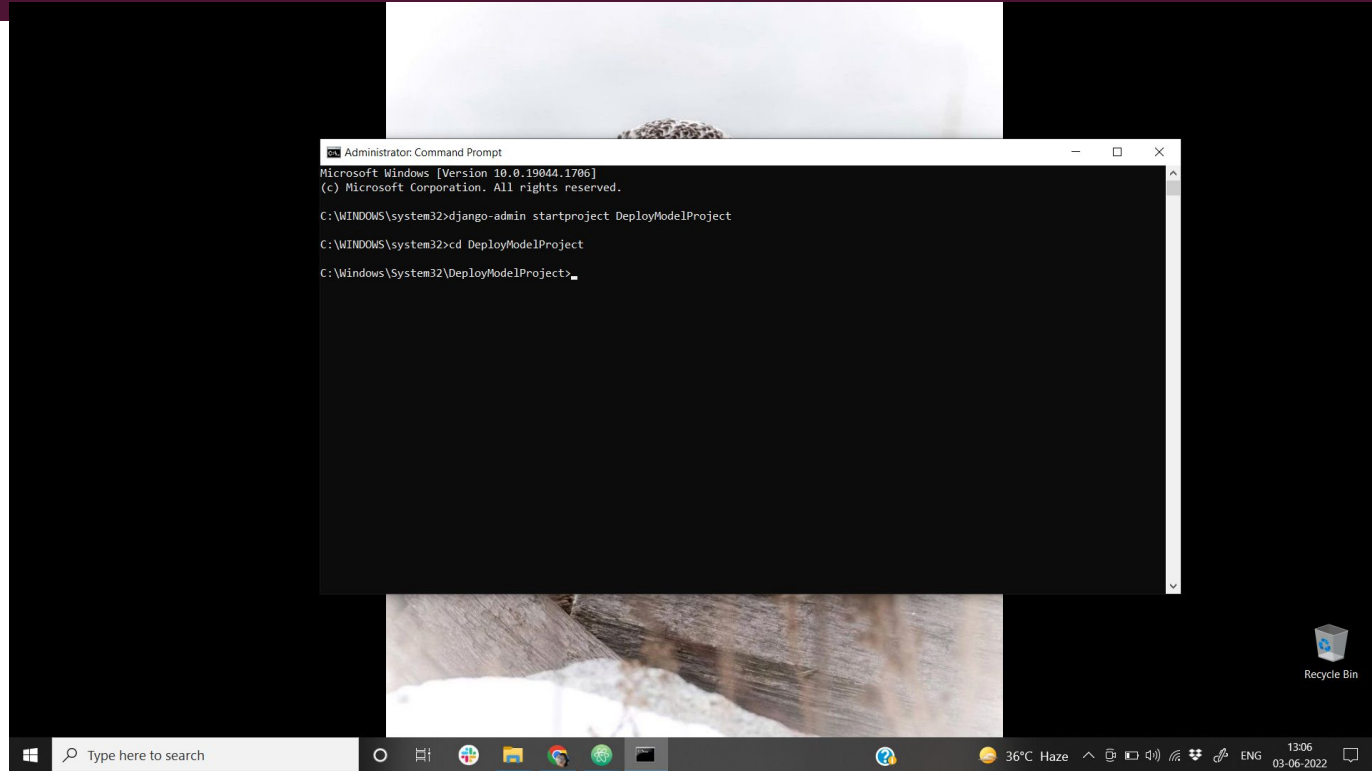
Install DJANGO

- Open CMD as administrator
- `pip install django`

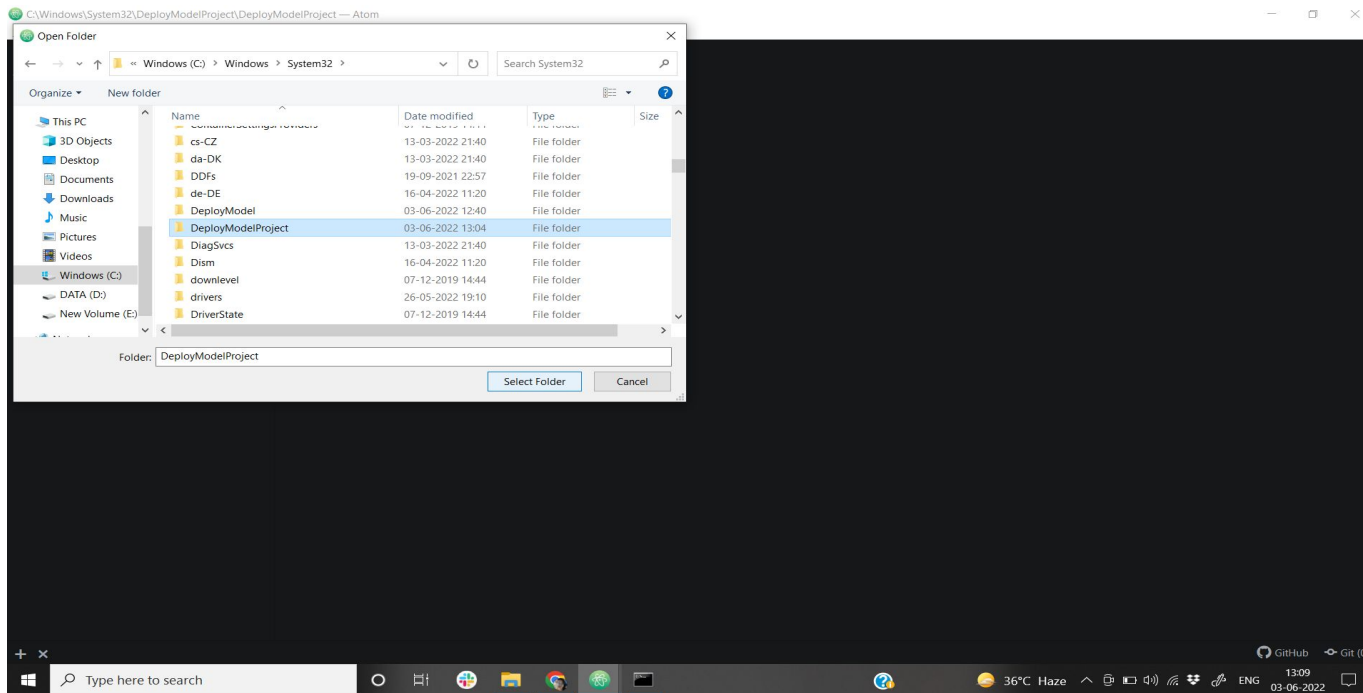
Start a new django project

- In the **CMD** as administrator window, type:
 - `django-admin startproject DeployModelProject`
 - `cd DeployModelProject`

Start Django Project



Open Project in Atom - Fire up Atom as administrator



First Run

- `python manage.py migrate`
- `python manage.py runserver`

First Run

C:\Windows\System32\DeployModelProject\DeployModelProject — Atom

File Edit View Selection Find Packages Help

Project

- DeployModelProject
 - __pycache__
 - __init__.py
 - asgi.py
 - settings.py
 - urls.py
 - wsgi.py

You can toggle the Git tab with `Ctrl+Shift+9`

```
Applying admin.0003_logentry_add_action_flag_choices... OK
Applying contenttypes.0002_remove_content_type_name... OK
Applying auth.0002_alter_permission_name_max_length... OK
Applying auth.0003_alter_user_email_max_length... OK
Applying auth.0004_alter_user_username_opts... OK
Applying auth.0005_alter_user_last_login_null... OK
Applying auth.0006_require_contenttypes_0002... OK
Applying auth.0007_alter_validators_add_error_messages... OK
Applying auth.0008_alter_user_username_max_length... OK
Applying auth.0009_alter_user_last_name_max_length... OK
Applying auth.0010_alter_group_name_max_length... OK
Applying auth.0011_update_proxy_permissions... OK
Applying auth.0012_alter_user_first_name_max_length... OK
Applying sessions.0001_initial... OK
PS C:\Windows\System32\DeployModelProject>
```

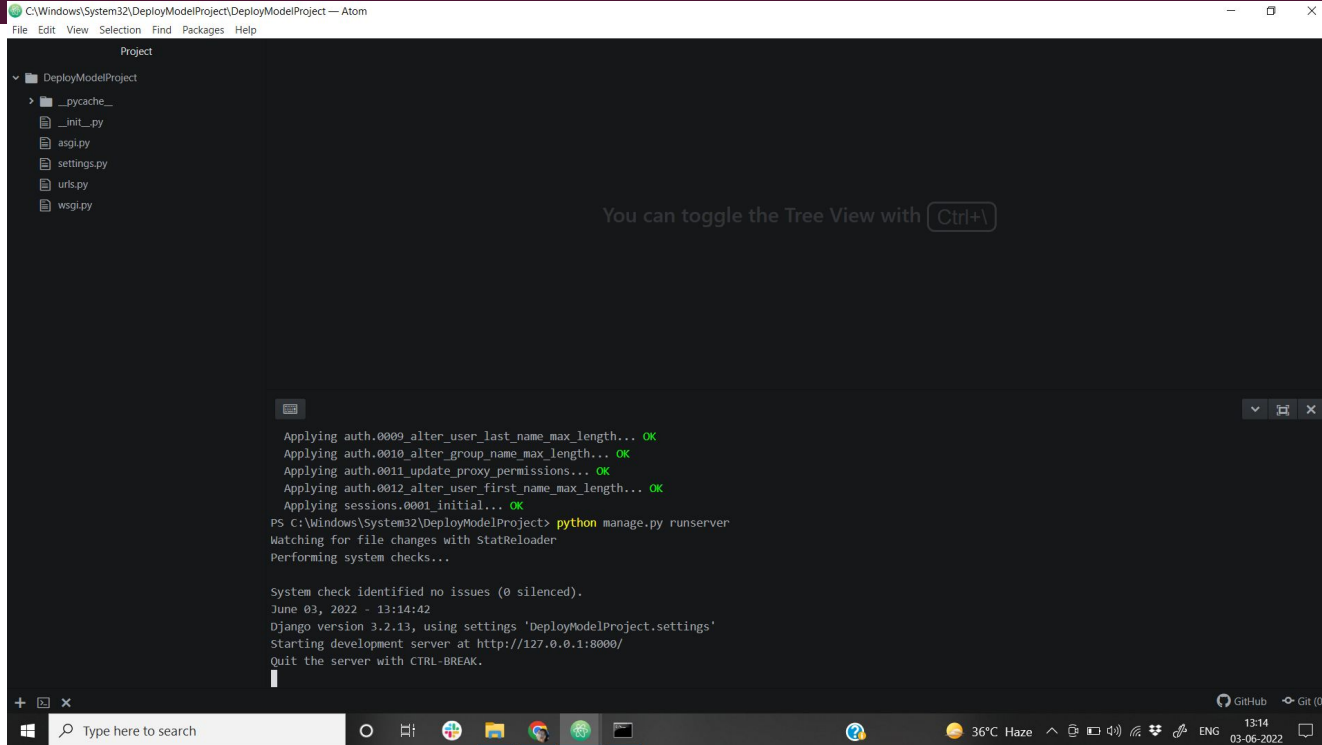
+ [x] x

Type here to search

GitHub Git (0)

13:13 03-06-2022

First Run



C:\Windows\System32\DeployModelProject\DeployModelProject — Atom

File Edit View Selection Find Packages Help

Project

- DeployModelProject
 - __pycache__
 - __init__.py
 - asgi.py
 - settings.py
 - urls.py
 - wsgi.py

You can toggle the Tree View with `Ctrl+T`

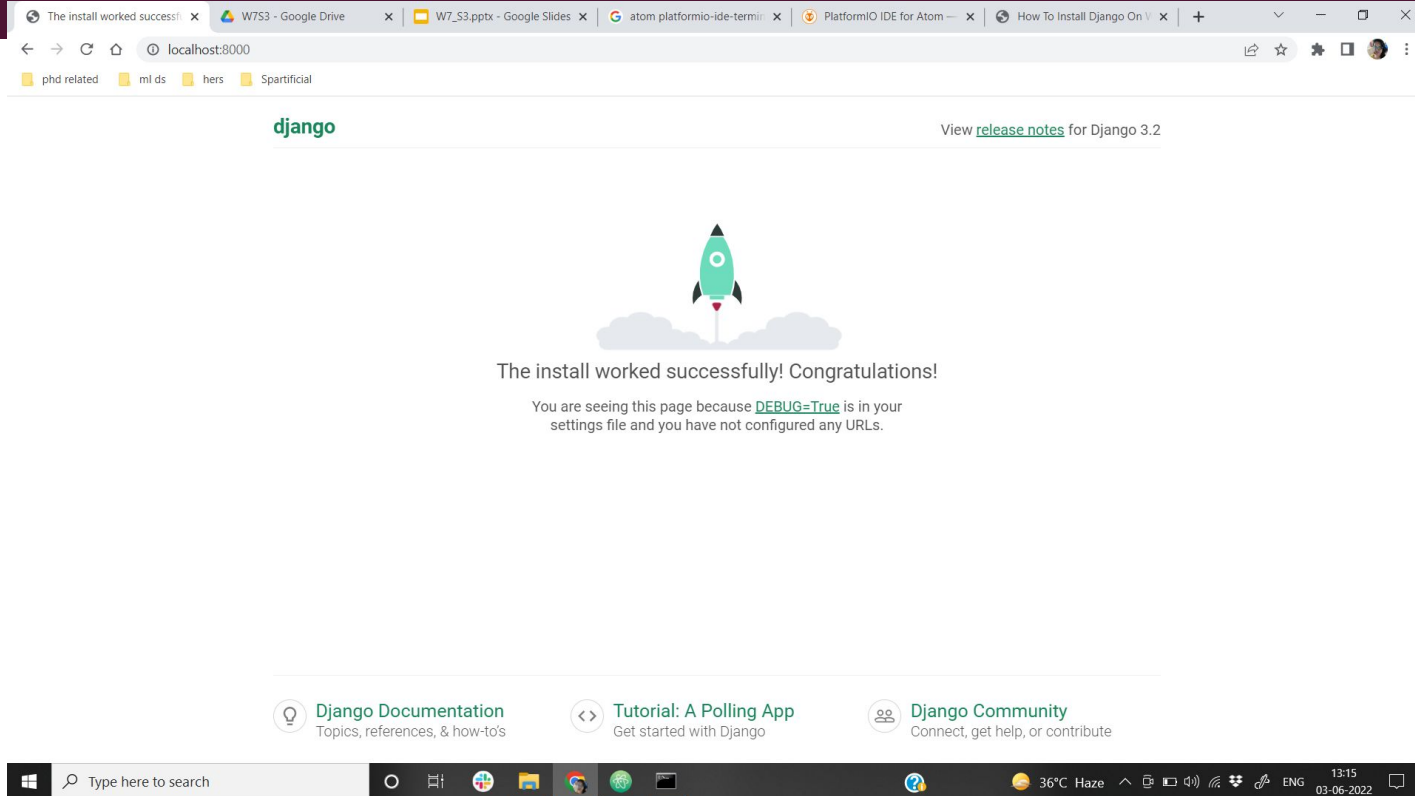
```
Applying auth.0009_alter_user_last_name_max_length... OK
Applying auth.0010_alter_group_name_max_length... OK
Applying auth.0011_update_proxy_permissions... OK
Applying auth.0012_alter_user_first_name_max_length... OK
Applying sessions.0001_initial... OK
PS C:\Windows\System32\DeployModelProject> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
June 03, 2022 - 13:14:42
Django version 3.2.13, using settings 'DeployModelProject.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

GitHub Git (0)

13:14 03-06-2022

First Run - open localhost:8000



The screenshot shows a web browser window with the address bar set to `localhost:8000`. The browser tabs include "The install worked successfully", "W753 - Google Drive", "W7_S3.pptx - Google Slides", "atom platformio-ide-termi...", "PlatformIO IDE for Atom", and "How To Install Django On...".

The Django 3.2 first-run page is displayed. At the top left is the **django** logo, and at the top right is a link to "View [release notes](#) for Django 3.2".

In the center, there is an illustration of a green rocket launching from a grey cloud. Below the illustration, the text reads: "The install worked successfully! Congratulations!"

Below the congratulations message, it states: "You are seeing this page because `DEBUG=True` is in your settings file and you have not configured any URLs."

At the bottom of the page, there are three links:

- Django Documentation**: Topics, references, & how-to's
- Tutorial: A Polling App**: Get started with Django
- Django Community**: Connect, get help, or contribute

The Windows taskbar at the bottom shows the search bar with "Type here to search", several application icons, and system information including "36°C Haze", "13:15", and "03-06-2022".

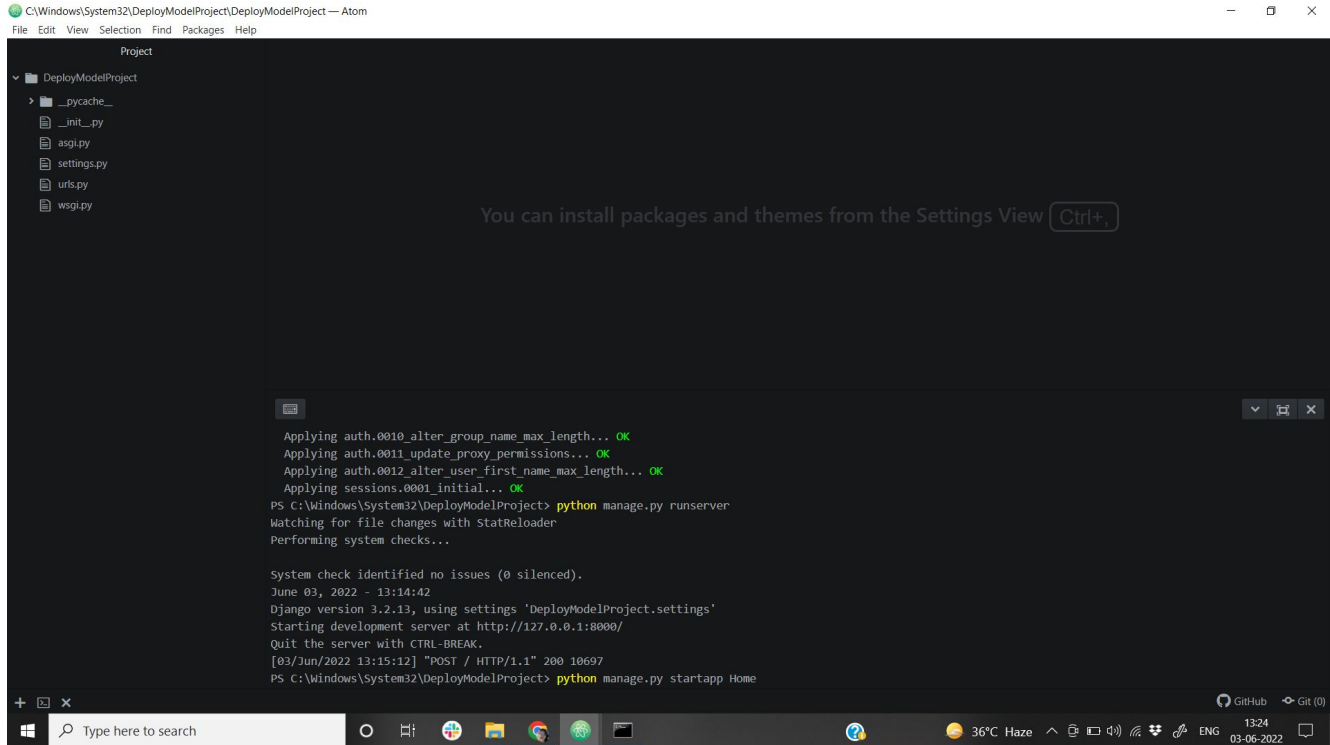
Progress Patch - I

- So far you have created a Django Project and started the Django Development Server which will run the application of your project.
- This is a lightweight Web server written purely in Python.
- By default the runserver command starts the development server on the port 8000.
- The server almost always automatically reloads - you have to manually reload if it does not.
- As the next step, we have to create an App - to hold/represent/host our ML model.

Create an App called Home

- First close the server by **CTRL+C**
- Type: `python manage.py startapp Home`

Create an App called Home - before



C:\Windows\System32\DeployModelProject\DeployModelProject — Atom

File Edit View Selection Find Packages Help

Project

DeployModelProject

- __pycache__
- __init__.py
- asgi.py
- settings.py
- urls.py
- wsgi.py

You can install packages and themes from the Settings View **Ctrl+,**

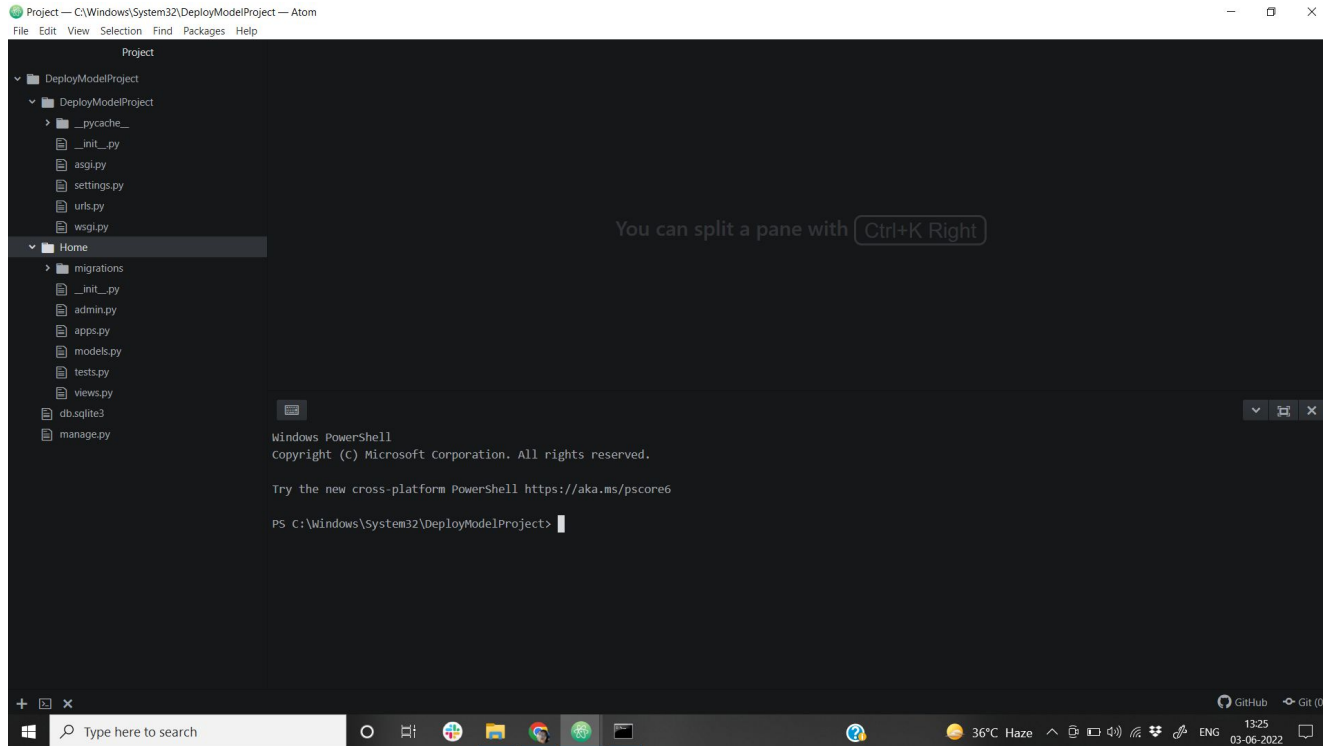
```
Applying auth.0010_alter_group_name_max_length... OK
Applying auth.0011_update_proxy_permissions... OK
Applying auth.0012_alter_user_first_name_max_length... OK
Applying sessions.0001_initial... OK
PS C:\Windows\System32\DeployModelProject> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
June 03, 2022 - 13:14:42
Django version 3.2.13, using settings 'DeployModelProject.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
[03/Jun/2022 13:15:12] "POST / HTTP/1.1" 200 10697
PS C:\Windows\System32\DeployModelProject> python manage.py startapp Home
```

Type here to search

36°C Haze 13:24 03-06-2022

Create an App called Home - after



Home App

- It has all python files to represent your app functionality.
- Develop your app and link everything properly.
- Steps:
 - Add 'Home' to `INSTALLED_APPS[]` in `settings.py`

settings.py — C:\Windows\System32\DeployModelProject — Atom

File Edit View Selection Find Packages Help

```
Project
└─ DeployModelProject
   └─ DeployModelProject
      └─ __pycache__
         _init_.py
         asgi.py
         settings.py
         urls.py
         wsgi.py
      └─ Home
         └─ migrations
            _init_.py
            admin.py
            apps.py
            models.py
            tests.py
            views.py
            db.sqlite3
            manage.py
```

```
settings.py
29
30
31 # Application definition
32
33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40     'Home',
41 ]
42
43 MIDDLEWARE = [
44     'django.middleware.security.SecurityMiddleware',
45     'django.contrib.sessions.middleware.SessionMiddleware',
46     'django.middleware.common.CommonMiddleware',
```

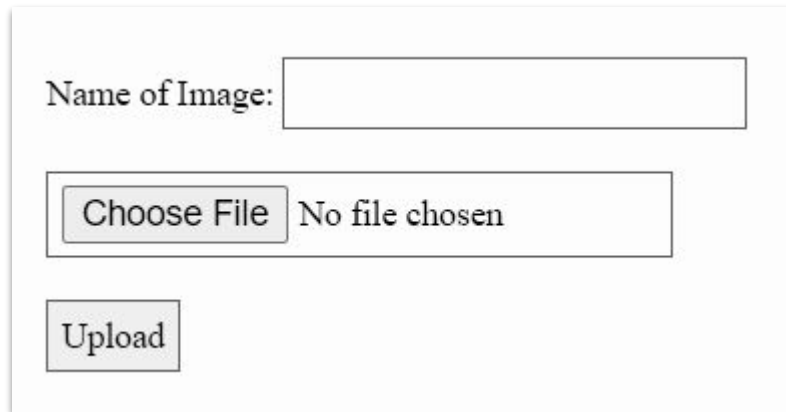
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

PS C:\Windows\System32\DeployModelProject>

Image Upload Form

- We're assuming that you want to save this image to a database so that you can permanently save it.
- So all images uploaded for prediction are saved to a database in Django - their pathways.
- The actual images are stored in the media directory of the website(project).

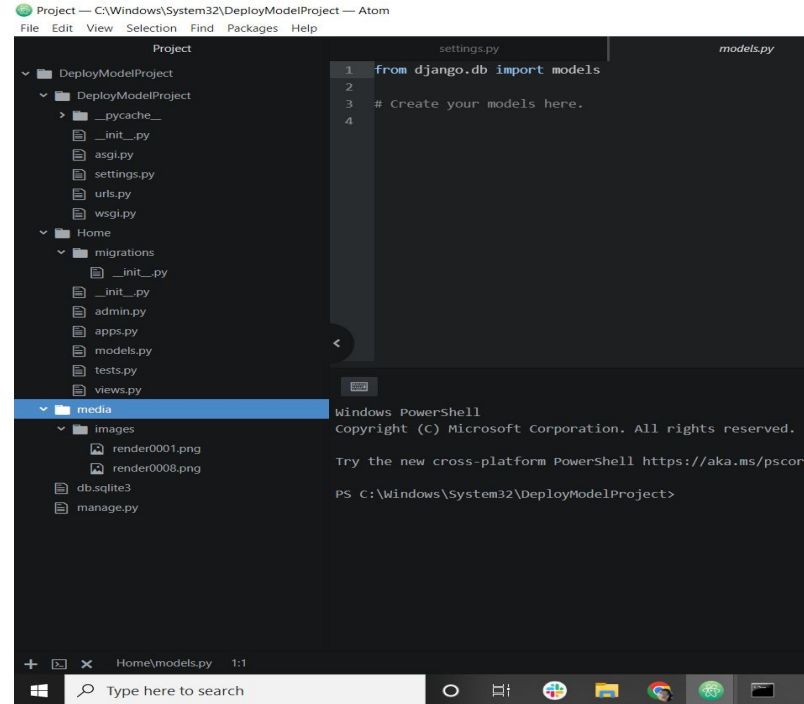


Name of Image:

No file chosen

Add Media Directory

- The first thing you should do is create a media directory in the root project directory of your website.
- In order to upload files such as images in Django, you have to make changes to the settings.py file.



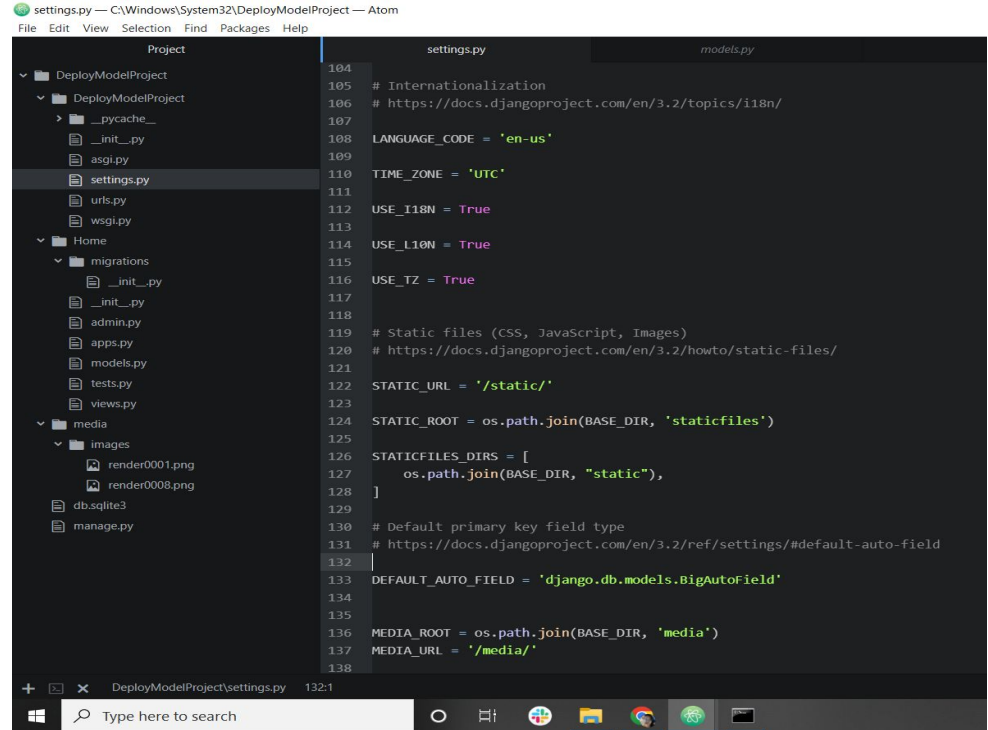
Specify the path to media directory

```
STATIC_ROOT = os.path.join(BASE_DIR, 'staticfiles')
```

```
STATICFILES_DIRS = [  
    os.path.join(BASE_DIR, "static"),  
]
```

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

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



settings.py — C:\Windows\System32\DeployModelProject — Atom

File Edit View Selection Find Packages Help

Project

- DeployModelProject
 - DeployModelProject
 - __pycache__
 - __init__.py
 - asgi.py
 - settings.py
 - urls.py
 - wsgi.py
 - Home
 - migrations
 - __init__.py
 - __init__.py
 - admin.py
 - apps.py
 - models.py
 - tests.py
 - views.py
 - media
 - images
 - render0001.png
 - render0008.png
 - db.sqlite3
 - manage.py

settings.py

```
104  
105 # Internationalization  
106 # https://docs.djangoproject.com/en/3.2/topics/i18n/  
107  
108 LANGUAGE_CODE = 'en-us'  
109  
110 TIME_ZONE = 'UTC'  
111  
112 USE_I18N = True  
113  
114 USE_L10N = True  
115  
116 USE_TZ = True  
117  
118  
119 # Static files (CSS, JavaScript, Images)  
120 # https://docs.djangoproject.com/en/3.2/howto/static-files/  
121  
122 STATIC_URL = '/static/'  
123  
124 STATIC_ROOT = os.path.join(BASE_DIR, 'staticfiles')  
125  
126 STATICFILES_DIRS = [  
127     os.path.join(BASE_DIR, "static"),  
128 ]  
129  
130 # Default primary key field type  
131 # https://docs.djangoproject.com/en/3.2/ref/settings/#default-auto-field  
132  
133 DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'  
134  
135  
136 MEDIA_ROOT = os.path.join(BASE_DIR, 'media')  
137 MEDIA_URL = '/media/'  
138
```

Update models.py

models.py — C:\Windows\System32\DeployModelProject — Atom

File Edit View Selection Find Packages Help

Project

- DeployModelProject
 - DeployModelProject
 - __pycache__
 - __init__.py
 - asgi.py
 - settings.py
 - urls.py
 - wsgi.py
 - Home
 - migrations
 - __init__.py
 - __init__.py
 - admin.py
 - apps.py
 - models.py
 - tests.py
 - views.py
 - media
 - images
 - render0001.png
 - render0008.png
 - db.sqlite3
 - manage.py

settings.py

```
1 from django.db import models
2
3 # Create your models here.
4
5
6 class Image(models.Model):
7     image = models.ImageField(upload_to='images/')
8
9     def __str__(self):
10         return self.image
11
```

models.py

Create forms.py

forms.py — C:\Windows\System32\DeployModelProject — Atom

File Edit View Selection Find Packages Help

The screenshot shows the Atom code editor interface. On the left is the 'Project' sidebar with a file tree. The tree shows a project named 'DeployModelProject' with subfolders like 'migrations' and 'media', and files like 'forms.py', 'models.py', 'tests.py', and 'views.py'. The 'forms.py' file is selected and highlighted. The main editor area on the right shows the code for 'forms.py'. The code defines a Django form class named 'ImageForm' that inherits from 'forms.ModelForm'. It includes a docstring 'Form for the image model' and a 'Meta' class that sets 'model = Image' and 'fields = ('image',)'. The line numbers 1 through 10 are visible on the left side of the code editor.

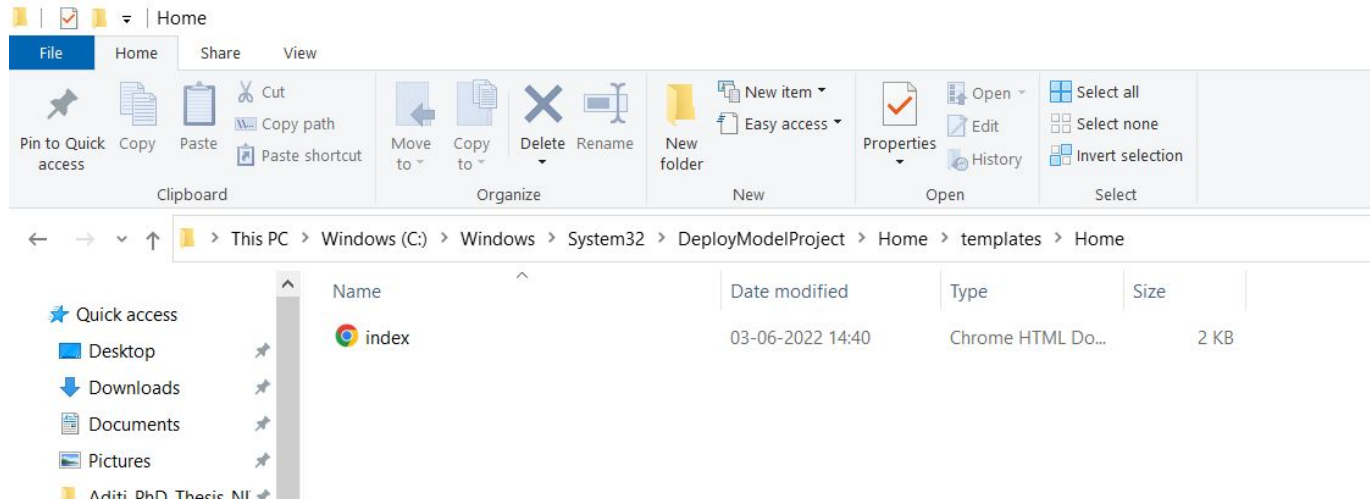
```
1 from .models import Image
2 from django import forms
3
4
5 class ImageForm(forms.ModelForm):
6     """Form for the image model"""
7     class Meta:
8         model = Image
9         fields = ('image',)
10
```

Progress Patch - 2

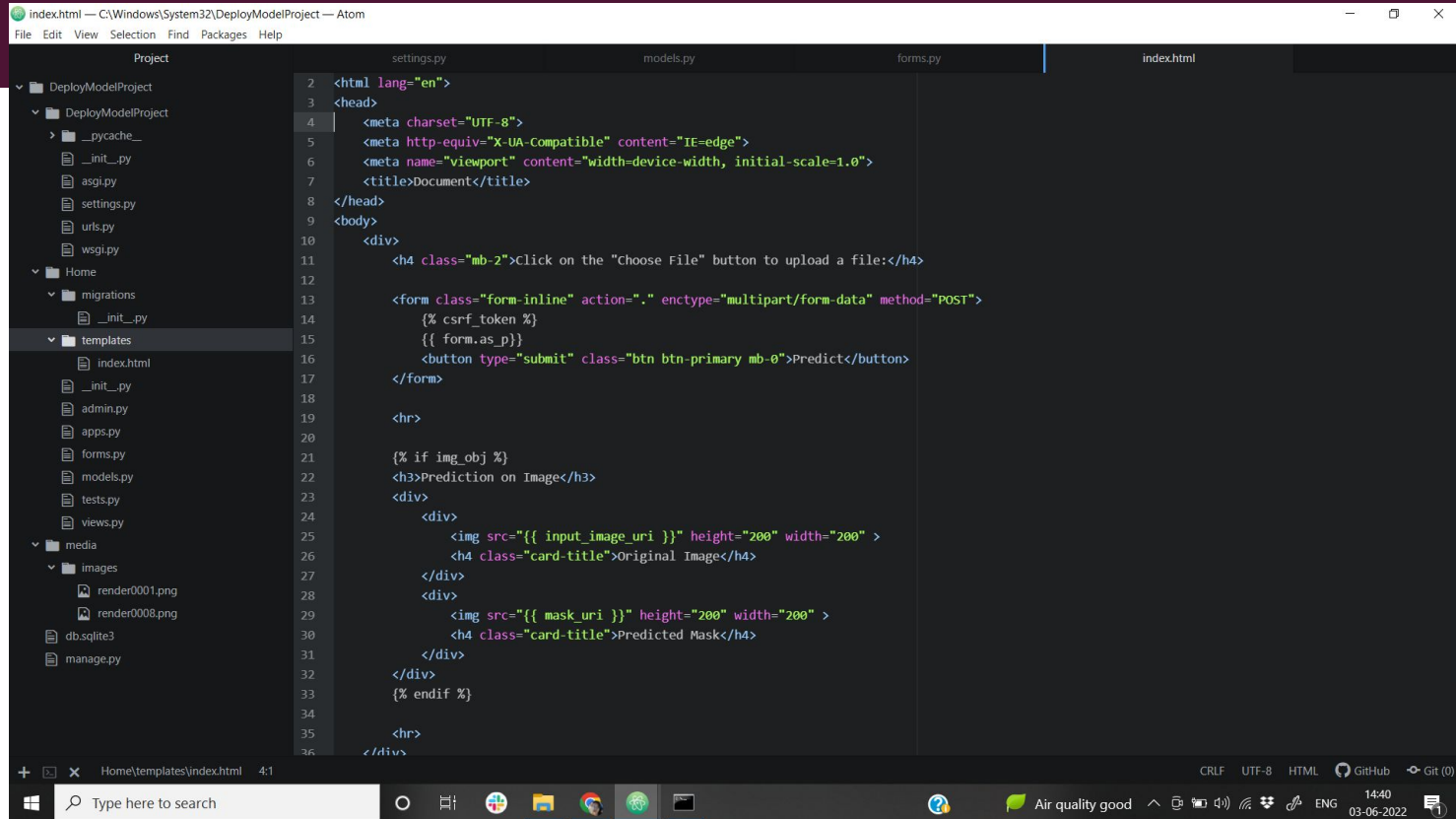
- **Now we have developed our app structure - added/updated forms.py, models.py**
- **Next we have to add our HTML page and link it to our ML model**

Create templates folder under Home

Create index.html under templates/Home



Index.html will display the upload file option and results

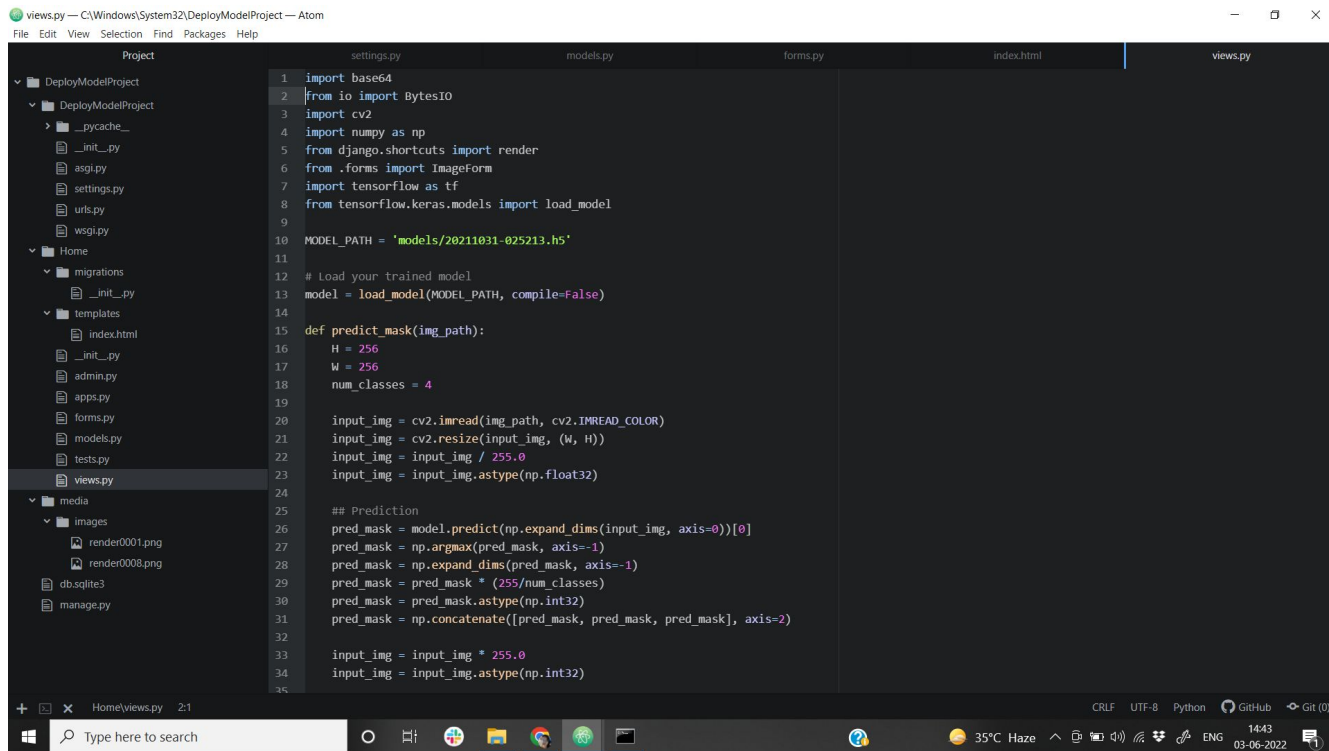


```
index.html — C:\Windows\System32\DeployModelProject — Atom
File Edit View Selection Find Packages Help

Project
  DeployModelProject
    DeployModelProject
      _pycache_
      _init_.py
      asgi.py
      settings.py
      urls.py
      wsgi.py
    Home
      migrations
      _init_.py
      templates
        index.html
        _init_.py
        admin.py
        apps.py
        forms.py
        models.py
        tests.py
        views.py
      media
        images
          render0001.png
          render0008.png
        db.sqlite3
        manage.py

2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8 </head>
9 <body>
10   <div>
11     <h4 class="mb-2">Click on the "Choose File" button to upload a file:</h4>
12
13     <form class="form-inline" action="." enctype="multipart/form-data" method="POST">
14       {% csrf_token %}
15       {{ form.as_p }}
16       <button type="submit" class="btn btn-primary mb-0">Predict</button>
17     </form>
18
19     <hr>
20
21     {% if img_obj %}
22     <h3>Prediction on Image</h3>
23     <div>
24       <div>
25         
26         <h4 class="card-title">Original Image</h4>
27       </div>
28       <div>
29         
30         <h4 class="card-title">Predicted Mask</h4>
31       </div>
32     </div>
33     {% endif %}
34
35     <hr>
36   </div>
```


views.py has index function - which will fetch the image object and pass it to predict function



```
views.py — C:\Windows\System32\DeployModelProject — Atom
File Edit View Selection Find Packages Help

Project settings.py models.py forms.py index.html views.py
DeployModelProject
  DeployModelProject
    _pycache_
    _init_.py
    asgi.py
    settings.py
    urls.py
    wsgi.py
  Home
    migrations
    _init_.py
    templates
    index.html
    _init_.py
    admin.py
    apps.py
    forms.py
    models.py
    tests.py
    views.py
  media
    images
      render0001.png
      render0008.png
    db.sqlite3
    manage.py

1 import base64
2 from io import BytesIO
3 import cv2
4 import numpy as np
5 from django.shortcuts import render
6 from .forms import ImageForm
7 import tensorflow as tf
8 from tensorflow.keras.models import load_model
9
10 MODEL_PATH = 'models/20211031-025213.h5'
11
12 # Load your trained model
13 model = load_model(MODEL_PATH, compile=False)
14
15 def predict_mask(img_path):
16     H = 256
17     W = 256
18     num_classes = 4
19
20     input_img = cv2.imread(img_path, cv2.IMREAD_COLOR)
21     input_img = cv2.resize(input_img, (W, H))
22     input_img = input_img / 255.0
23     input_img = input_img.astype(np.float32)
24
25     ## Prediction
26     pred_mask = model.predict(np.expand_dims(input_img, axis=0))[0]
27     pred_mask = np.argmax(pred_mask, axis=-1)
28     pred_mask = np.expand_dims(pred_mask, axis=-1)
29     pred_mask = pred_mask * (255/num_classes)
30     pred_mask = pred_mask.astype(np.int32)
31     pred_mask = np.concatenate([pred_mask, pred_mask, pred_mask], axis=2)
32
33     input_img = input_img * 255.0
34     input_img = input_img.astype(np.int32)
```

Progress Patch - 3

- **We added our HTML file and linked it to our ML model prediction function!**
- **Now we can register our app and perform migrations.**
- **After that we will link our app with our project directory via urls.py files.**
- **We also need to copy our trained model to our project directory.**
- **Then we can run our ML model on a website!**

Register app on admin.py

admin.py — C:\Windows\System32\DeployModelProject — Atom

File Edit View Selection Find Packages Help

Project

- DeployModelProject
 - DeployModelProject
 - __pycache__
 - __init__.py
 - asgi.py
 - settings.py
 - urls.py
 - wsgi.py
 - Home
 - migrations
 - __init__.py
 - templates
 - index.html
 - __init__.py
 - admin.py
 - apps.py
 - forms.py
 - models.py
 - tests.py
 - views.py
 - media
 - images
 - render0001.png
 - render0008.png
 - db.sqlite3
 - manage.py

```
1 from django.contrib import admin
2
3 from .models import Image
4
5 # Register your models here.
6 admin.site.register(Image)
7
```

settings.py models.py forms.py index.html views.py admin.py

Home\admin.py 7:1

CRLF UTF-8 Python GitHub Git (0)

Type here to search 35°C Haze 14:45 03-06-2022

Migrations

Project — C:\Windows\System32\DeployModelProject — Atom

File Edit View Selection Find Packages Help

Project

- DeployModelProject
 - DeployModelProject
 - __pycache__
 - __init__.py
 - asgi.py
 - settings.py
 - urls.py
 - wsgi.py
 - Home
 - __pycache__
 - migrations
 - __pycache__
 - __init__.py
 - 0001_initial.py
 - templates
 - index.html
 - media
 - images
 - render0001.png
 - render0008.png
 - db.sqlite3
 - manage.py

settings.py

```
1 import base64
2 from io import BytesIO
3 import cv2
4 import numpy as np
5 from django.shortcuts import render
6 from .forms import ImageForm
7 import tensorflow as tf
8 from tensorflow.keras.models import load_model
9
10 MODEL_PATH = 'models/20211031-025213.h5'
11
12 # Load your trained model
13 model = load_model(MODEL_PATH, compile=False)
14
15 def predict_mask(img_path):
16     H = 256
17     W = 256
18     num_classes = 4
```

views.py

```
1 from django.shortcuts import render
2 from .forms import ImageForm
3 from .models import Image
4 from .migrations import predict_mask
5
6 def home(request):
7     if request.method == 'POST':
8         form = ImageForm(request.POST, request.FILES)
9         if form.is_valid():
10             image = form.save()
11             return render(request, 'home.html', {'image': image})
12     else:
13         form = ImageForm()
14     return render(request, 'home.html', {'form': form})
15
16 def about(request):
17     return render(request, 'about.html')
18
19 def contact(request):
20     return render(request, 'contact.html')
```

Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

```
PS C:\Windows\System32\DeployModelProject> python manage.py makemigrations
Migrations for 'Home':
  Home\migrations\0001_initial.py
    - Create model Image
PS C:\Windows\System32\DeployModelProject> python manage.py migrate
Operations to perform:
  Apply all migrations: Home, admin, auth, contenttypes, sessions
Running migrations:
  Applying Home.0001_initial... OK
PS C:\Windows\System32\DeployModelProject>
```

Home\views.py 2:1

Type here to search

CRLF UTF-8 Python GitHub Git (0)

14:47 03-06-2022

Link via urls.py inside Home

urls.py — Home — C:\Windows\System32\DeployModelProject — Atom

File Edit View Selection Find Packages Help

Project

- DeployModelProject
 - DeployModelProject
 - __pycache__
 - _init_.py
 - asgi.py
 - settings.py
 - urls.py
 - wsgi.py
 - Home
 - __pycache__
 - migrations
 - __pycache__
 - _init_.py
 - 0001_initial.py
 - templates
 - _init_.py
 - admin.py
 - apps.py
 - forms.py
 - models.py
 - tests.py
 - urls.py
 - views.py
 - media
 - images
 - render0001.png
 - render0008.png
 - dbsqlite3
 - manage.py

```
1 from django.urls import path
2 from Home import views
3
4 urlpatterns = [
5     path('', views.index, name='home_index'),
6 ]
7
```

Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
June 03, 2022 - 14:48:19
Django version 3.2.13, using settings 'DeployModelProject.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

[03/Jun/2022 14:48:25] "POST / HTTP/1.1" 200 10697
[03/Jun/2022 14:48:25] "GET /static/admin/css/fonts.css HTTP/1.1" 304 0
[03/Jun/2022 14:48:25] "GET /static/admin/fonts/Roboto-Regular-webfont.woff HTTP/1.1" 304 0
[03/Jun/2022 14:48:25] "GET /static/admin/fonts/Roboto-Bold-webfont.woff HTTP/1.1" 304 0
[03/Jun/2022 14:48:25] "GET /static/admin/fonts/Roboto-Light-webfont.woff HTTP/1.1" 304 0
[03/Jun/2022 14:48:33] "POST / HTTP/1.1" 200 10697
PS C:\Windows\System32\DeployModelProject>

Home\urls.py 7:1

CRLF UTF-8 Python GitHub Git (0)

14:49 03-06-2022

Link via urls.py inside project

The screenshot shows the Atom editor interface with a Django project named 'DeployModelProject'. The file explorer on the left shows the project structure, including folders like 'migrations', 'templates', and 'media', and files like 'urls.py'. The main editor window displays the 'urls.py' file with the following code:

```
1 from django.contrib import admin
2 from django.urls import path, include
3
4 urlpatterns = [
5     path('admin/', admin.site.urls),
6     path('', include('Home.urls')),
7 ]
8
```

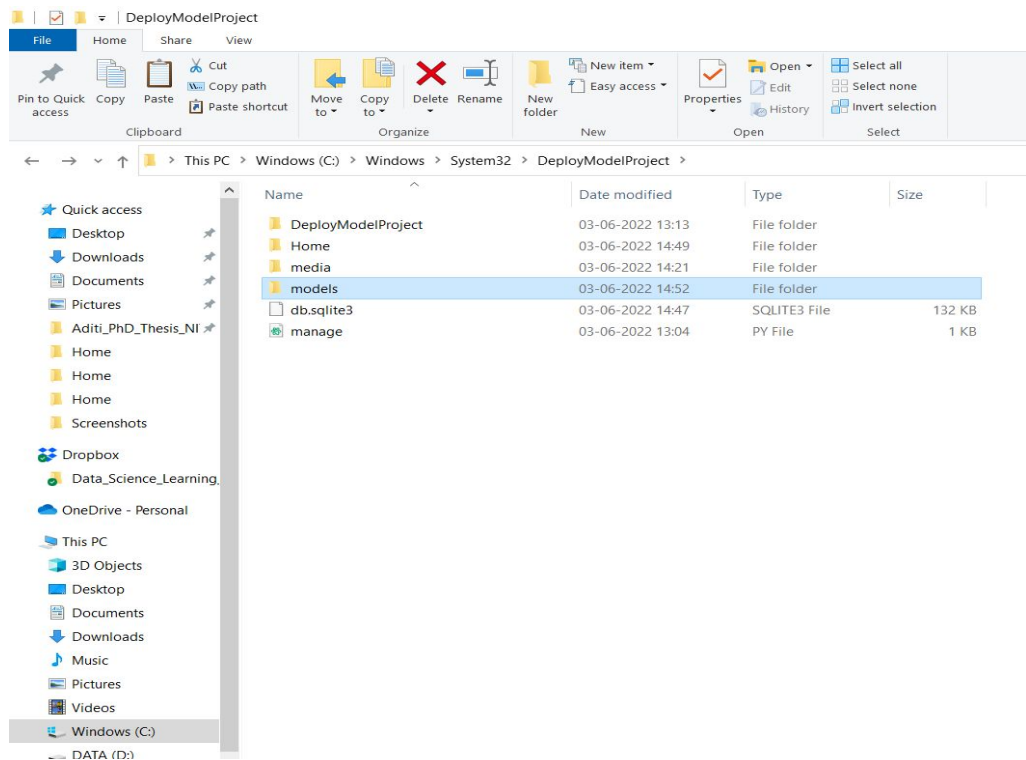
Below the code editor, the terminal output shows the Django development server running. It includes system checks, the Django version (3.2.13), and a list of HTTP requests and responses:

```
Watching for file changes with StatReloader
Performing system checks...

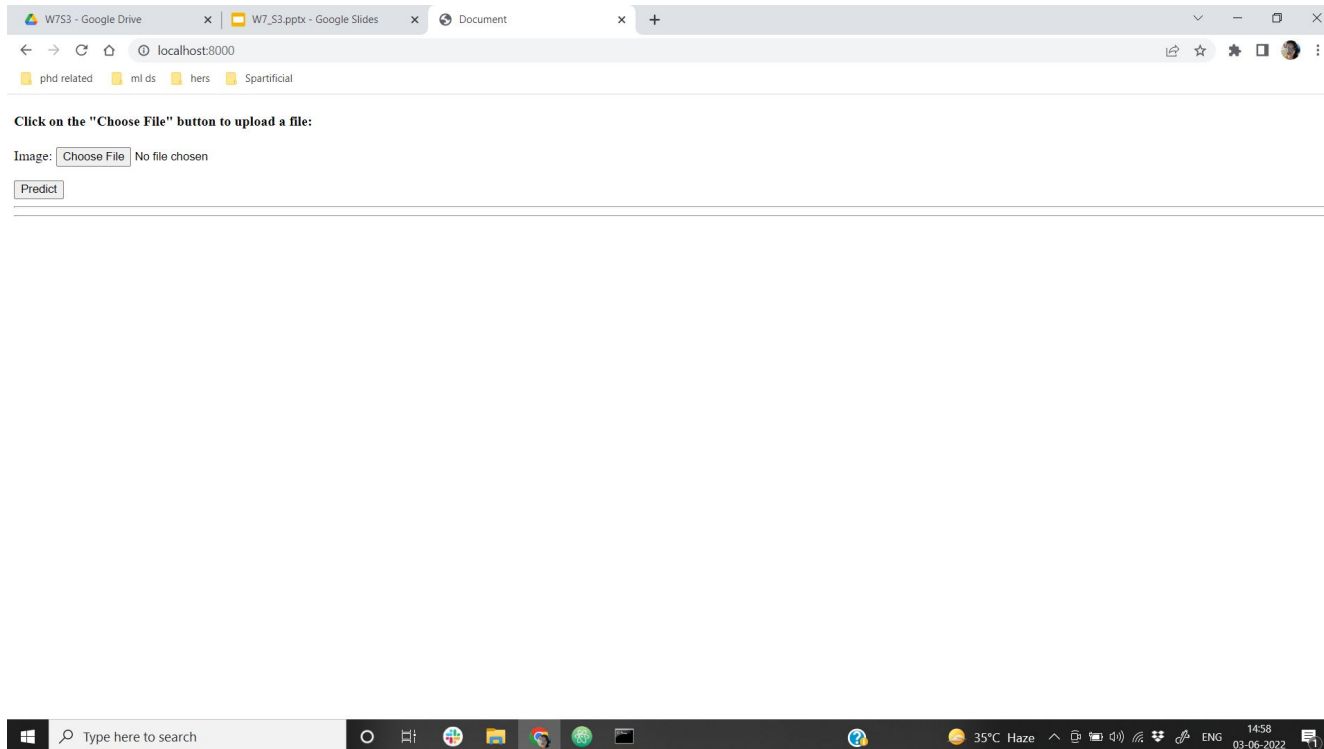
System check identified no issues (0 silenced).
June 03, 2022 - 14:48:19
Django version 3.2.13, using settings 'DeployModelProject.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
[03/Jun/2022 14:48:25] "POST / HTTP/1.1" 200 10697
[03/Jun/2022 14:48:25] "GET /static/admin/css/fonts.css HTTP/1.1" 304 0
[03/Jun/2022 14:48:25] "GET /static/admin/fonts/Roboto-Regular-webfont.woff HTTP/1.1" 304 0
[03/Jun/2022 14:48:25] "GET /static/admin/fonts/Roboto-Bold-webfont.woff HTTP/1.1" 304 0
[03/Jun/2022 14:48:25] "GET /static/admin/fonts/Roboto-Light-webfont.woff HTTP/1.1" 304 0
[03/Jun/2022 14:48:33] "POST / HTTP/1.1" 200 10697
PS C:\Windows\System32\DeployModelProject>
```

The status bar at the bottom indicates the file encoding (CRLF), text encoding (UTF-8), and the active language (Python). It also shows the current time (14:50) and date (03-06-2022).

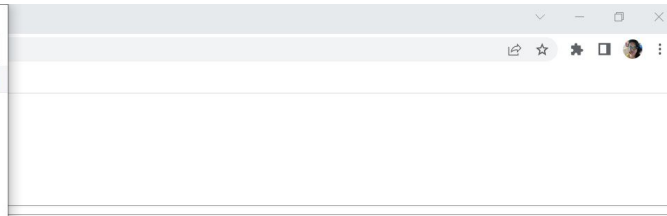
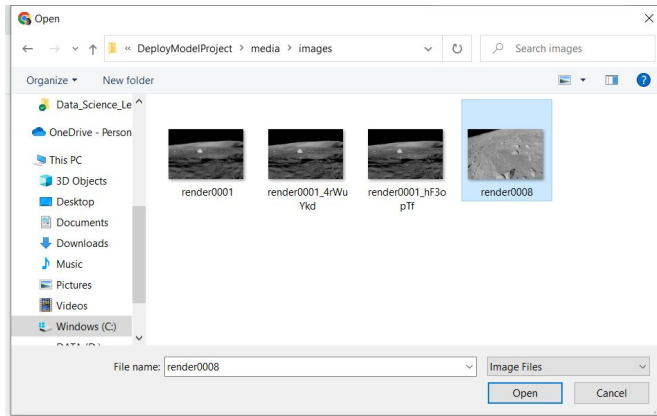
Copy model into project directory



```
python manage.py makemigrations
python manage.py migrate
python manage.py runserver
```




```
python manage.py makemigrations  
python manage.py migrate  
python manage.py runserver
```



```
python manage.py makemigrations
python manage.py migrate
python manage.py runserver
```

W7S3 - Google Drive x W7_S3.pptx - Google Slides x Document x +


localhost:8000

phd related ml ds hers Spartificial


Click on the "Choose File" button to upload a file:

Image: No file chosen

Prediction on Image



Original Image



Predicted Mask

Type here to search

35°C Haze 14:58 03-06-2022

General Guidelines

- **Always make sure you choose the parent project folder while opening in Atom**
- **Run makemigrations and migrate regularly**

References

- <http://www.learningaboutelectronics.com/Articles/How-to-insert-images-into-a-database-table-with-Python-in-Django.php>
- <https://docs.djangoproject.com/en/1.11/intro/tutorial01/>
- https://www.youtube.com/watch?v=zcALUNZNBuk&ab_channel=SaiPrakashReddy

Time to practice some question!

Q - Create a django app called School which includes a form to insert your school image on it. Also add link to your school website at the top before adding images. Also take care of the necessary files which need to be created to work it efficiently.

Note : Go through this PPT before attempting the question.