

Crud Operation Project Documentation.

সর্বপ্রথম ভার্চুয়াল এনভায়রনমেন্ট তৈরী করতে হবে.

```
H:\pythonProject\Crud_Operation_Final_Project>python -m venv env
H:\pythonProject\Crud_Operation_Final_Project>env\Scripts\activate
(env) H:\pythonProject\Crud_Operation_Final_Project>
```

তারপর এনভায়রনমেন্ট একটিভ করতে হবে।

```
Microsoft Windows [Version 10.0.17763.4645]
(c) 2018 Microsoft Corporation. All rights reserved.

H:\pythonProject\Crud_Operation_Final_Project>python -m venv env
H:\pythonProject\Crud_Operation_Final_Project>env\Scripts\activate
(env) H:\pythonProject\Crud_Operation_Final_Project>
```

তারপর Django ইনস্টল করতে হবে।

```
(env) H:\pythonProject\Crud_Operation_Final_Project>pip install django
Collecting django
  Downloading Django-4.2.4-py3-none-any.whl (8.0 MB)
----- 8.0/8.0 MB 72.9 kB/s eta 0:00:00
```

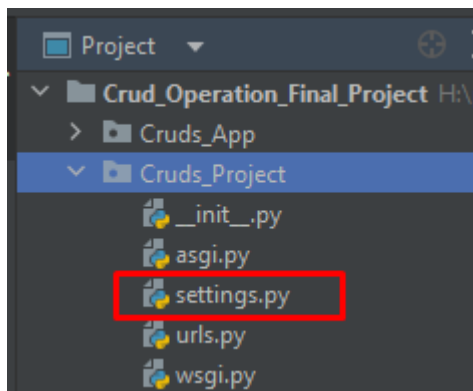
তারপর প্রজেক্ট তৈরী করতে হবে।

```
(env) H:\pythonProject\Crud_Operation_Final_Project>django-admin startproject Cruds_Project .
```

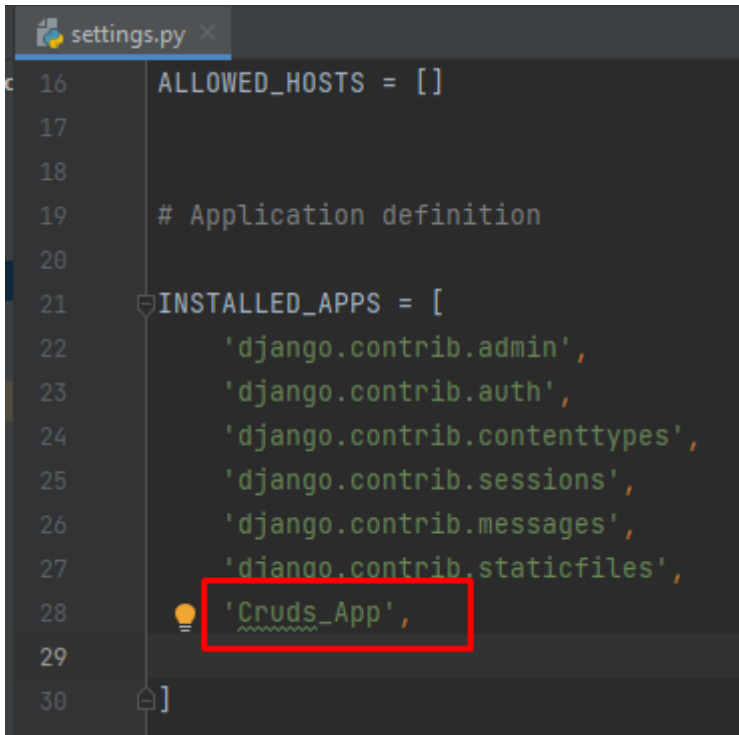
তারপর App তৈরী করতে হবে।

```
(env) H:\pythonProject\Crud_Operation_Final_Project>python manage.py startapp Cruds_App
```

এখন Pycharm ওপেন করে প্রজেক্ট এর সেটিং এ চলে যেতে হবে.

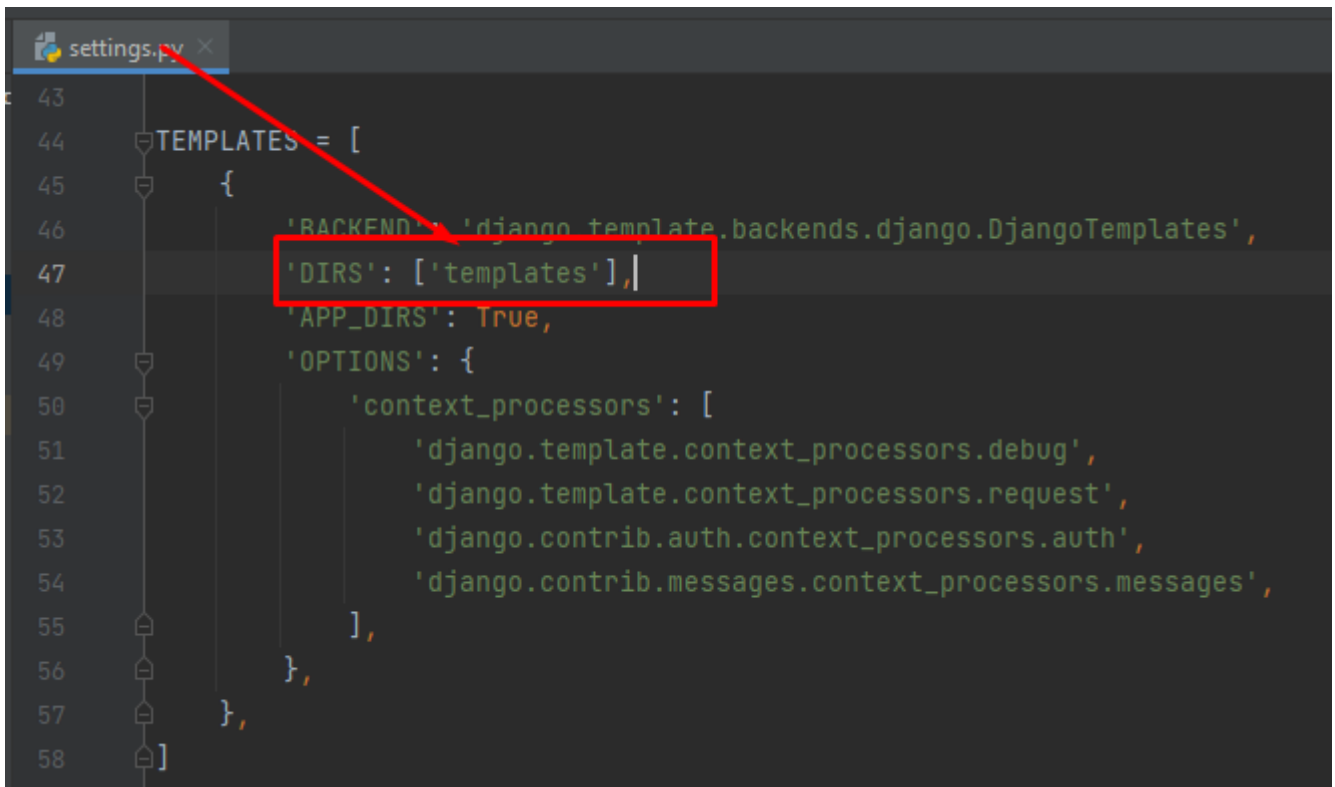


তারপর আমাদের প্রজেক্ট এর App টি সেটিংস এর INSTALLED_APPS এ চিনিয়ে দিতে হবে।



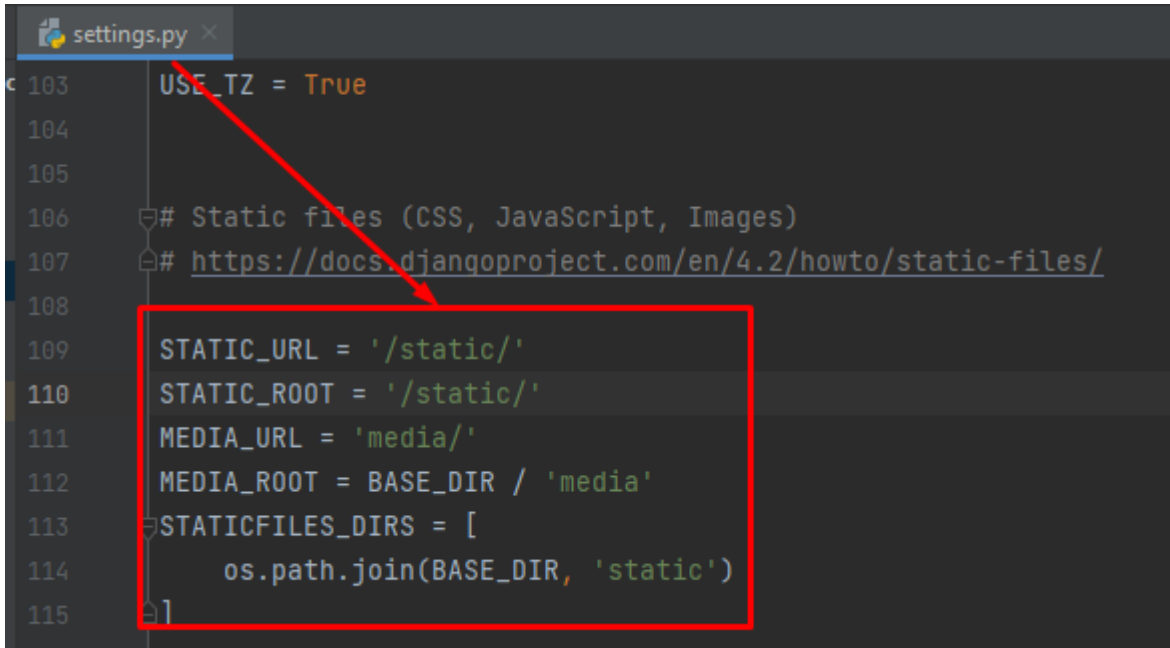
```
settings.py
16 ALLOWED_HOSTS = []
17
18
19 # Application definition
20
21 INSTALLED_APPS = [
22     'django.contrib.admin',
23     'django.contrib.auth',
24     'django.contrib.contenttypes',
25     'django.contrib.sessions',
26     'django.contrib.messages',
27     'django.contrib.staticfiles',
28     'Cruds_App',
29 ]
```

তারপর টেম্পলেট এ 'templates' চিনিয়ে দিতে হবে.



```
settings.py
43
44 TEMPLATES = [
45     {
46         'BACKEND': 'django.template.backends.django.DjangoTemplates',
47         'DIRS': ['templates'],
48         'APP_DIRS': True,
49         'OPTIONS': {
50             'context_processors': [
51                 'django.template.context_processors.debug',
52                 'django.template.context_processors.request',
53                 'django.contrib.auth.context_processors.auth',
54                 'django.contrib.messages.context_processors.messages',
55             ],
56         },
57     },
58 ]
```

তারপর স্ট্যাটিক ফোল্ডার সেটিংস এ এভাবে লিখতে হবে।

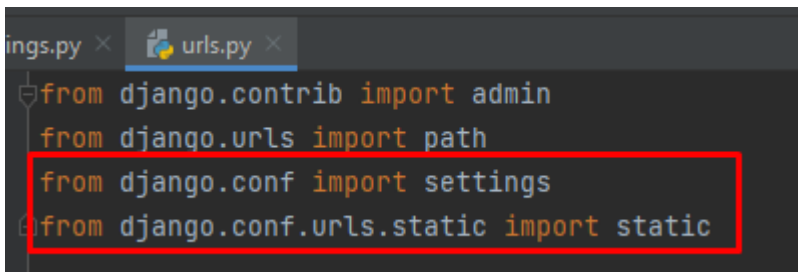


The screenshot shows a code editor with the file 'settings.py' open. The code is as follows:

```
103 USE_TZ = True
104
105
106 # Static files (CSS, JavaScript, Images)
107 # https://docs.djangoproject.com/en/4.2/howto/static-files/
108
109 STATIC_URL = '/static/'
110 STATIC_ROOT = '/static/'
111 MEDIA_URL = 'media/'
112 MEDIA_ROOT = BASE_DIR / 'media'
113 STATICFILES_DIRS = [
114     os.path.join(BASE_DIR, 'static')
115 ]
```

A red box highlights the static file configuration lines (109-115). A red arrow points from the text 'তারপর স্ট্যাটিক ফোল্ডার সেটিংস এ এভাবে লিখতে হবে।' to the highlighted section.

তারপর প্রজেক্ট এর urls এর সেটিংস এন্ড স্ট্যাটিক ফাইলকে ইম্পোর্ট করতে হবে।

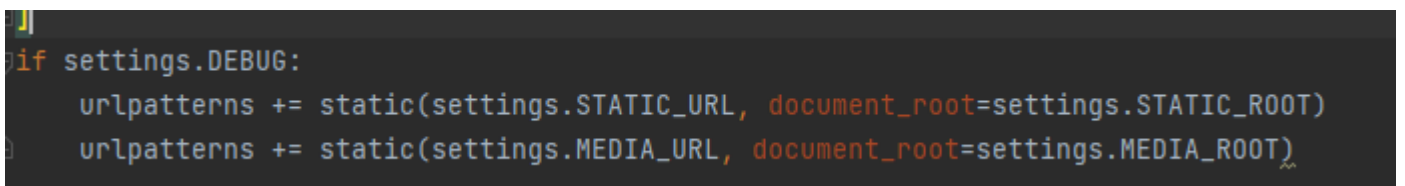


The screenshot shows a code editor with the file 'urls.py' open. The code is as follows:

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

A red box highlights the import statements for 'settings' and 'static' (lines 3-4).

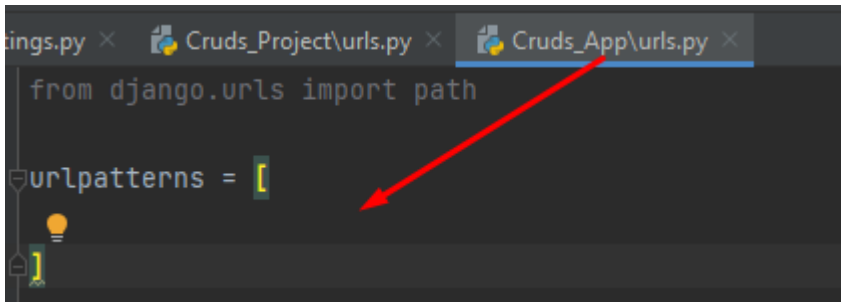
urls এর সেটিংস এন্ড স্ট্যাটিক ফাইলকে এভাবে চিনিয়ে দিতে হবে.



The screenshot shows a code editor with the file 'urls.py' open. The code is as follows:

```
if settings.DEBUG:
    urlpatterns += static(settings.STATIC_URL, document_root=settings.STATIC_ROOT)
urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

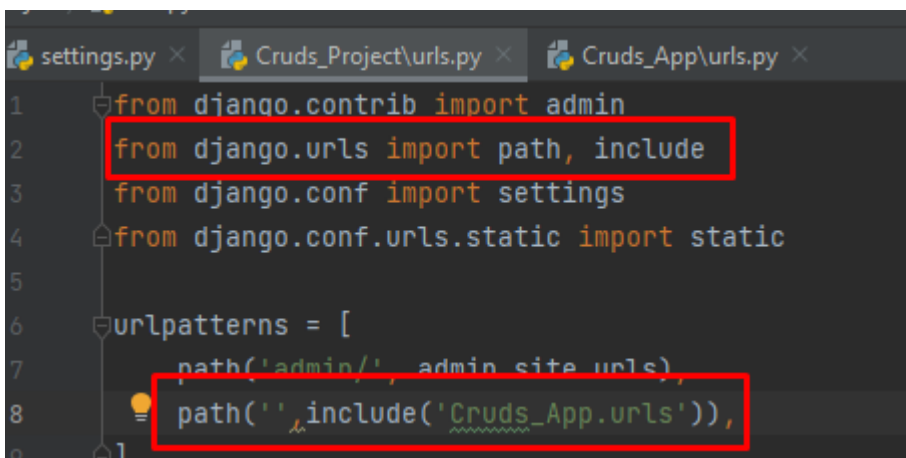
প্রজেক্ট urls এর সাথে include করে কাজ করার জন্য Cruds_App এ আরেকটি urls বানাবো, তারপর এভাবে import করবো এবং ব্ল্যাংক urlpatterns নিবো।



```
Cruds_App\urls.py
from django.urls import path

urlpatterns = []
```

তারপর main project urls এ include করবো এভাবে।

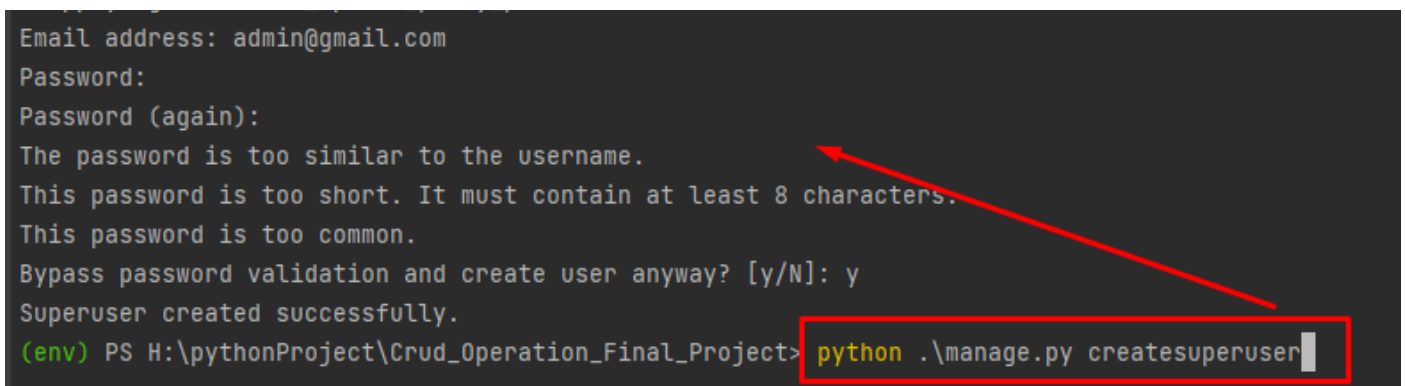


```
Cruds_Project\urls.py
from django.contrib import admin
from django.urls import path, include
from django.conf import settings
from django.conf.urls.static import static

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

তারপর terminal থেকে makemigrations আর migrate করে নিবো।

তারপর superuser create করতে হবে।



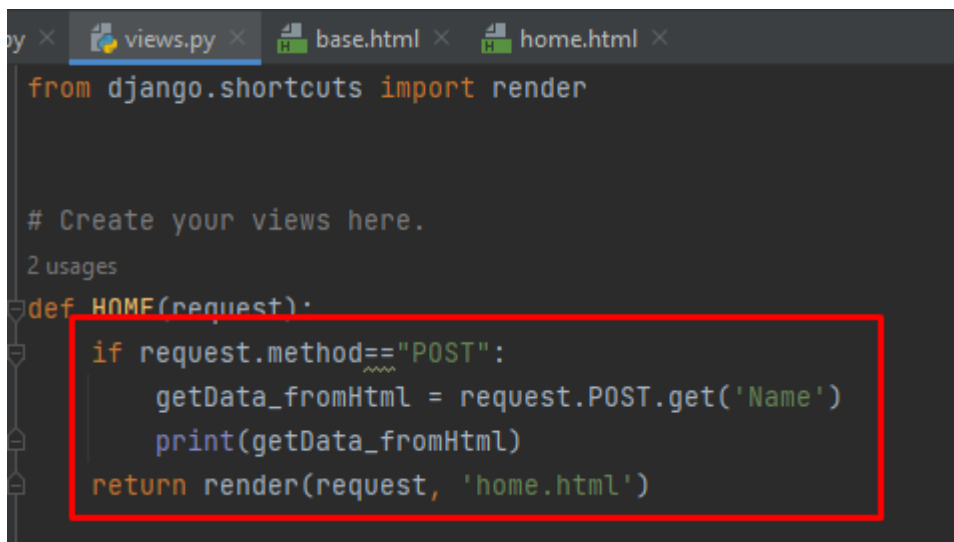
```
Email address: admin@gmail.com
Password:
Password (again):
The password is too similar to the username.
This password is too short. It must contain at least 8 characters.
This password is too common.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
(env) PS H:\pythonProject\Crud_Operation_Final_Project> python .\manage.py createsuperuser
```

তারপর runserver দিয়ে admin প্যানেল এ প্রবেশ করতে পারবো।

এরপর App urls এ from .views import..... এখানে function নাম তা বসে দিতে হবে।

এরপর bootstrap -5 থেকে একটি template নিয়ে নিবো।

Frontend থেকে কোন ডাটা backend এ (ডাটাবেজে) পাঠানোর জন্য নিম্নোক্ত function এর মাধ্যমে পাঠাতে হবে.

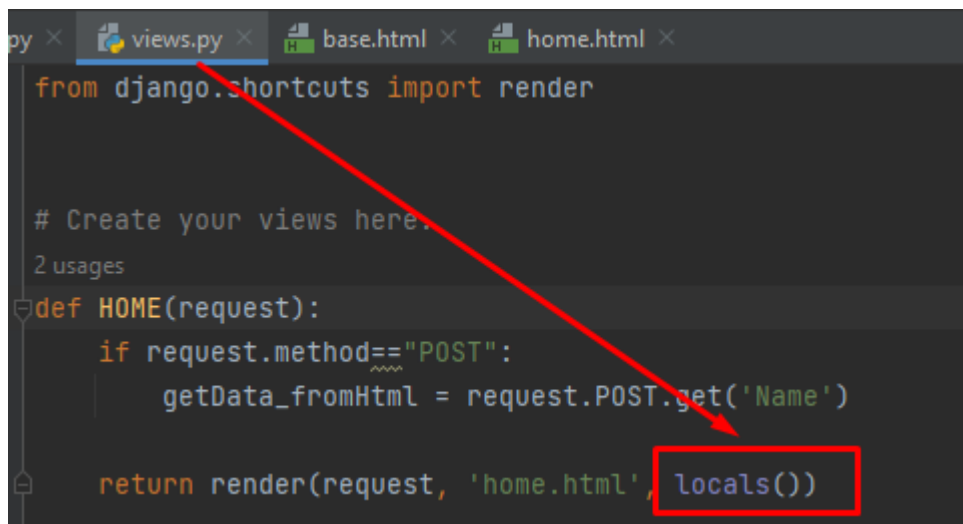


```
views.py x base.html x home.html x
from django.shortcuts import render

# Create your views here.
2 usages
def HOME(request):
    if request.method=="POST":
        getData_fromHtml = request.POST.get('Name')
        print(getData_fromHtml)
        return render(request, 'home.html')
```

Backend থেকে কোনো ডাটা Frontend এ show করানোর জন্য নিম্নোক্ত function (Locals এবং Context) এর মাধ্যমে show করানো যায়। function এ locals() মানে হচ্ছে ডেটাবেজের সবগুলো ডাটা নিয়ে show করাবে। আর context হচ্ছে স্পেসিফিক ডাটা ধরে ধরে show করবে। এতে সময় বাজবে, মেমরি কম খরচ হবে। function লিখার পরে html এ যে field এ ডাটা শো করতে চাই সেখানে ফাঙ্কশন এ ব্যবহৃত {{variable name }} লিখতে হবে।

Using Locals:



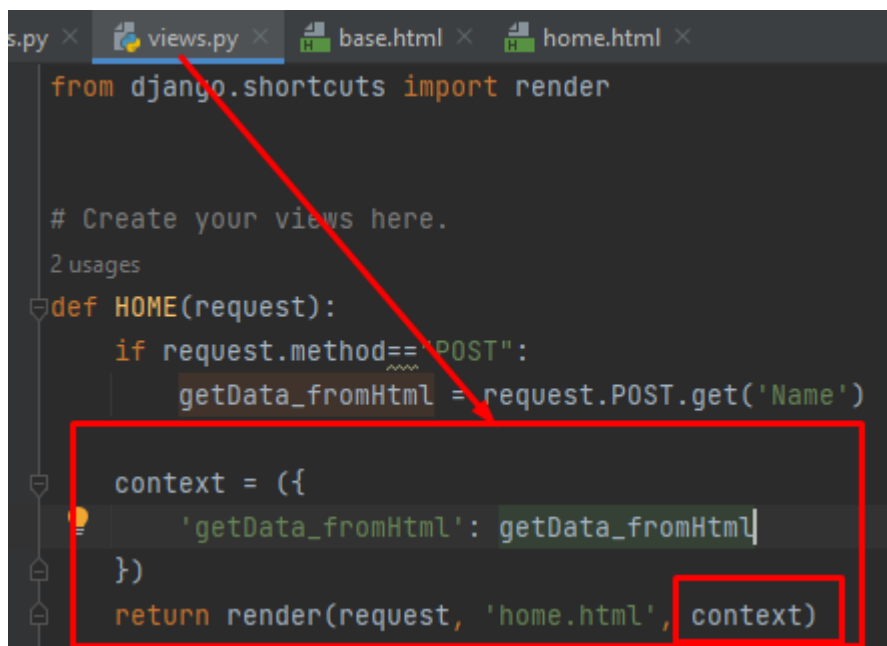
```
from django.shortcuts import render

# Create your views here.
2 usages
def HOME(request):
    if request.method=="POST":
        getData_fromHtml = request.POST.get('Name')

    return render(request, 'home.html', locals())
```

A red arrow points from the `render` function in the first line to the `locals()` argument in the `return` statement. The `locals()` is highlighted with a red box.

Using Context:



```
from django.shortcuts import render

# Create your views here.
2 usages
def HOME(request):
    if request.method=="POST":
        getData_fromHtml = request.POST.get('Name')

    context = ({
        'getData_fromHtml': getData_fromHtml
    })
    return render(request, 'home.html', context)
```

A red arrow points from the `render` function in the first line to the `context` argument in the `return` statement. The `context` is highlighted with a red box. The `context` variable is defined as a dictionary containing the `getData_fromHtml` variable.

