

EDGEDjango

EXPLORER

OPEN EDITORS

- # mystyle.css static/css
- myjs.js static/js
- settings.py EDGEDjango
- urls.py EDGEDjango
- views.py EDGEDjango
- homepage.html templates
- base.html templates 3**
- navbar.html templates

EDGEDJANGO

EDGEDjango

- __init__.py
- asgi.py
- settings.py
- urls.py
- views.py
- wsgi.py
- media
- static
- templates
 - base.html 3**
 - homepage.html
 - navbar.html
- manage.py

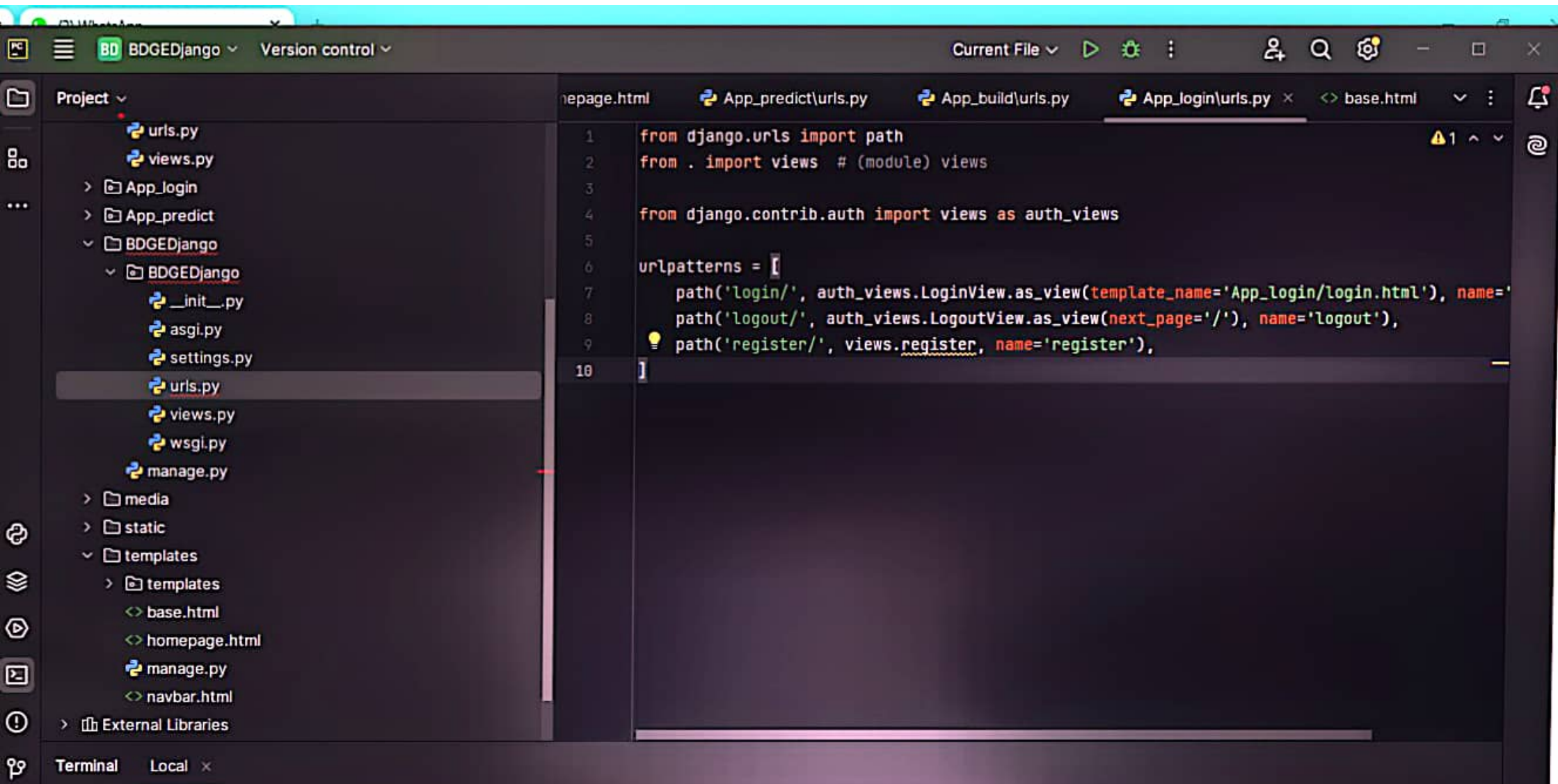
templates > base.html > ...

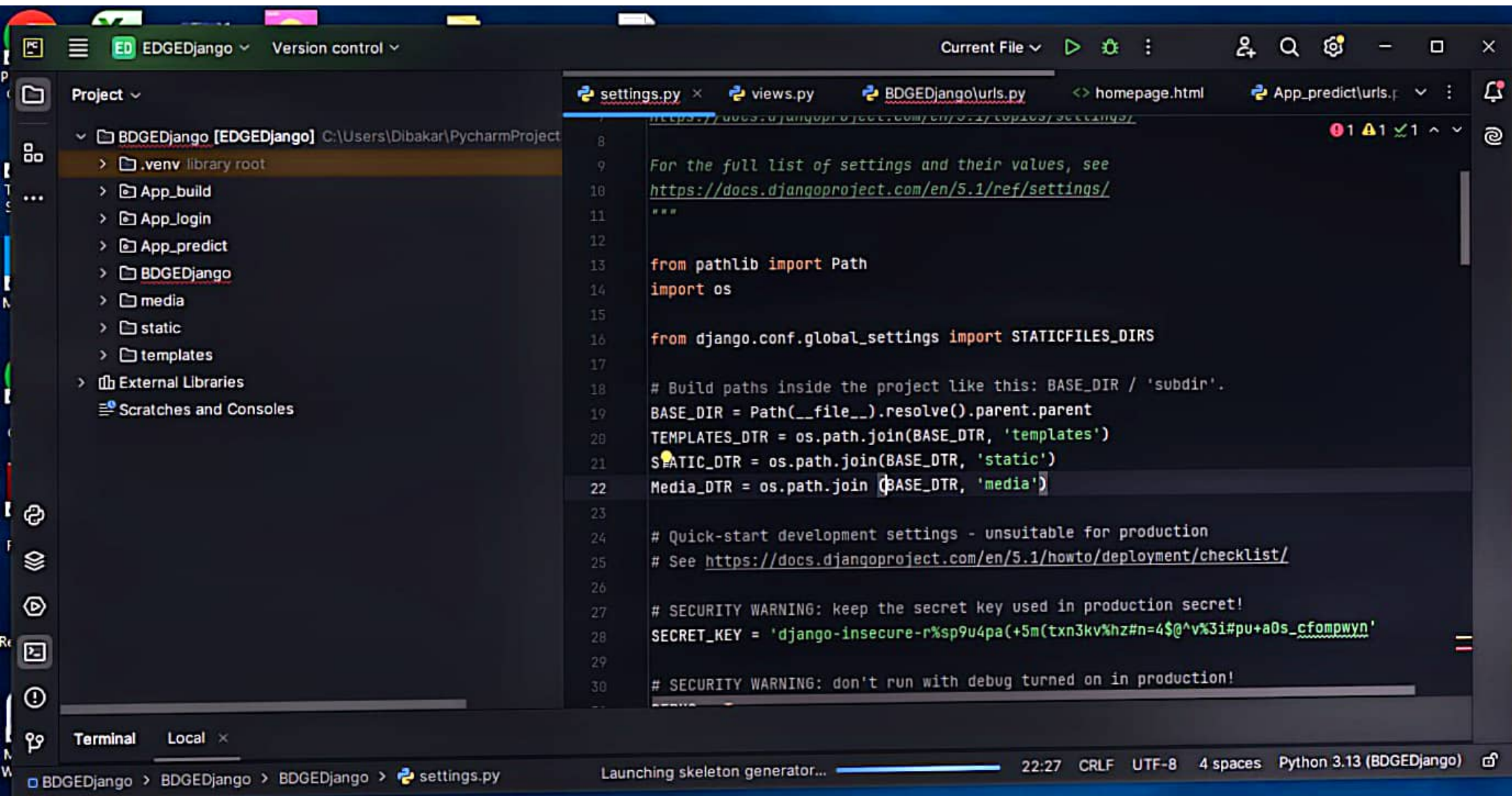
```
1 {% load static %}
2
3 <!DOCTYPE html>
4 <html lang="en" dir="ltr">
5
6 <head>
7   <meta charset="utf-8">
8   <script src="https://kit.fontawesome.com/4c53b33752.js" crossorigin="anonymous"></script>
9   <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02d8j0t6vLEHfe/JQGi
10   <script type="text/javascript" src="{% static 'js/myjs.js' %}"></script>
11   <link rel="stylesheet" href="{% static 'css/mystyle.css' %}">
12   <title>{% block title_block %}{% endblock %}</title>
13 </head>
14
15 <body style="background-color: #E0E0E0;">
16   {% include 'navbar.html' %}
17   <div class="container">
18     {% block body_block %}
19
20     {% endblock %}
21   </div>
22 </body>
23
24 </html>
```

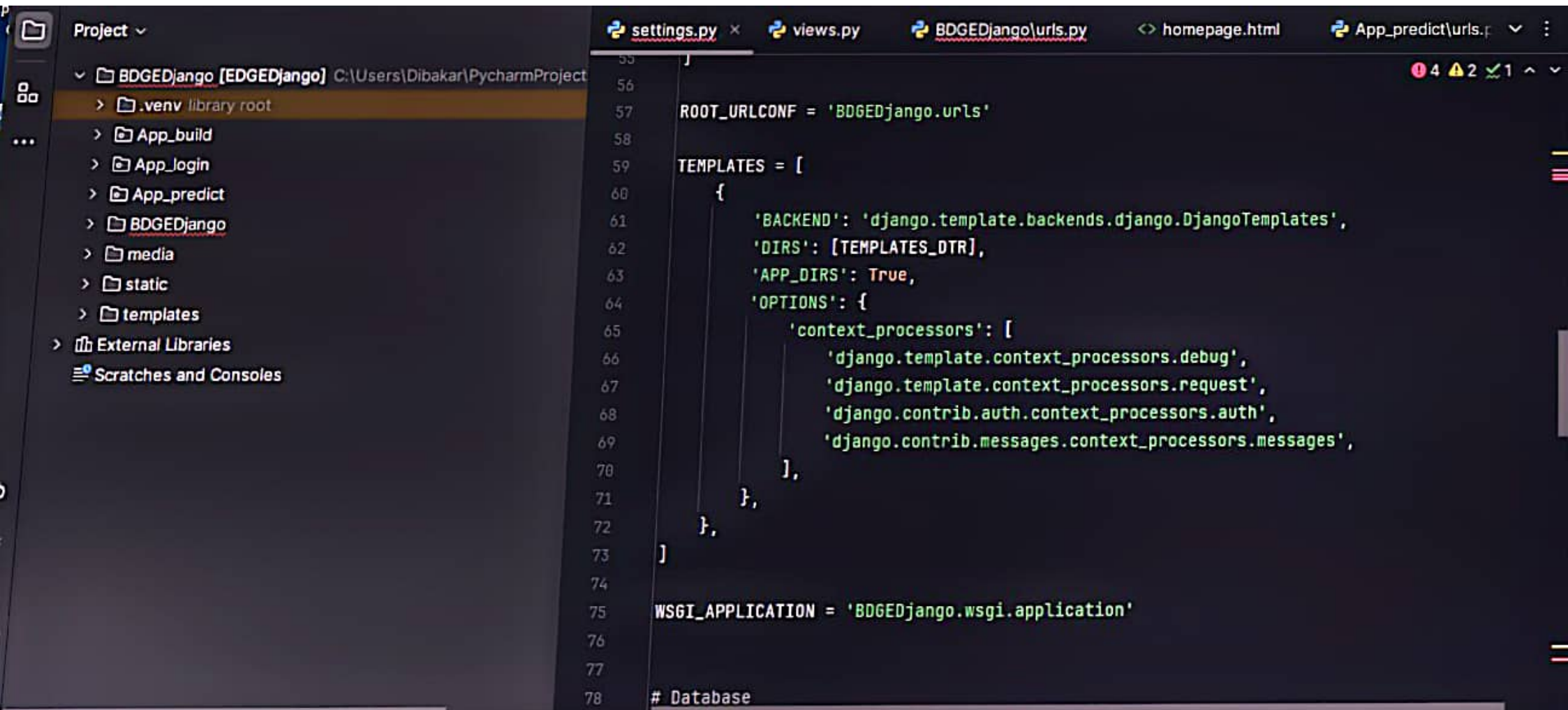
The script element allows authors to include dynamic script and data blocks in their documents. The element does not represent content for the user.

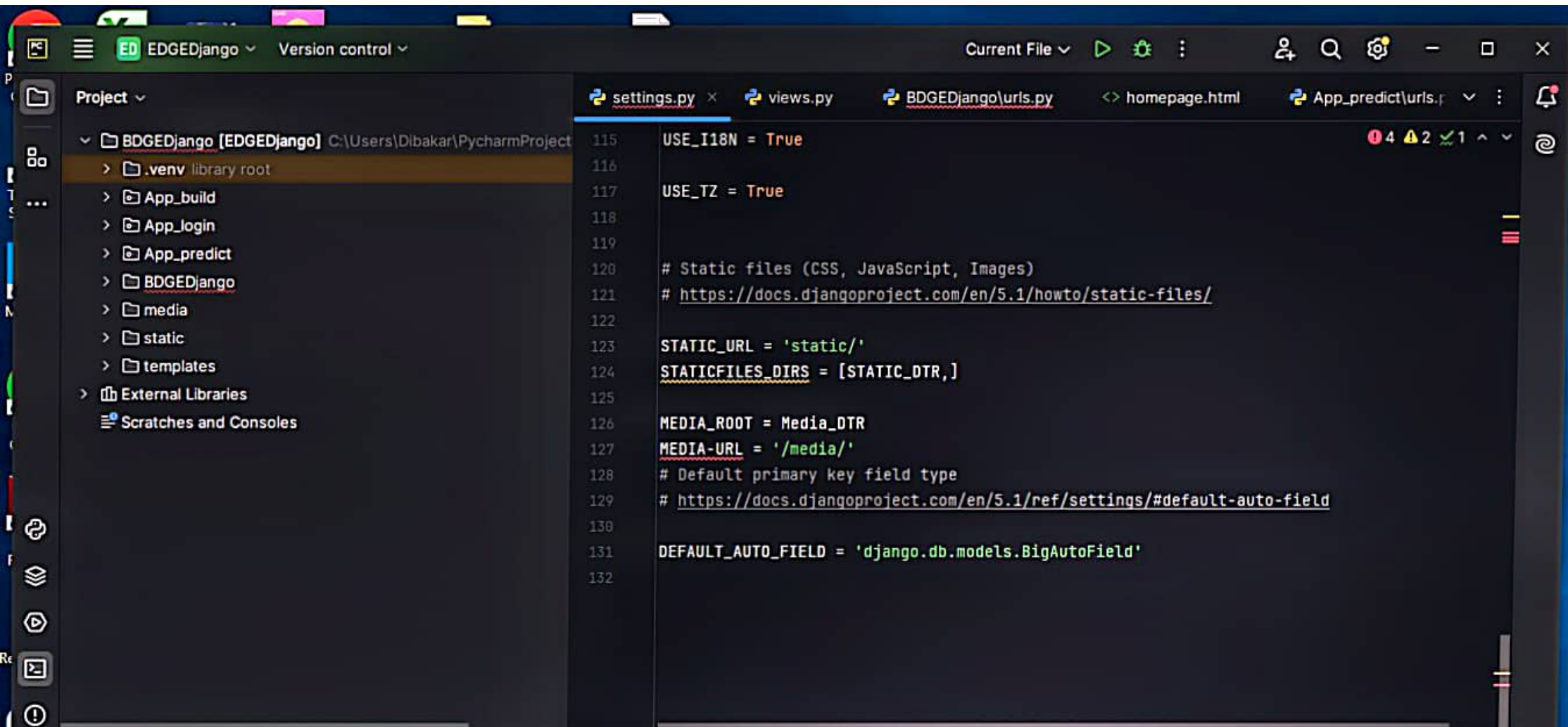
MDN Reference

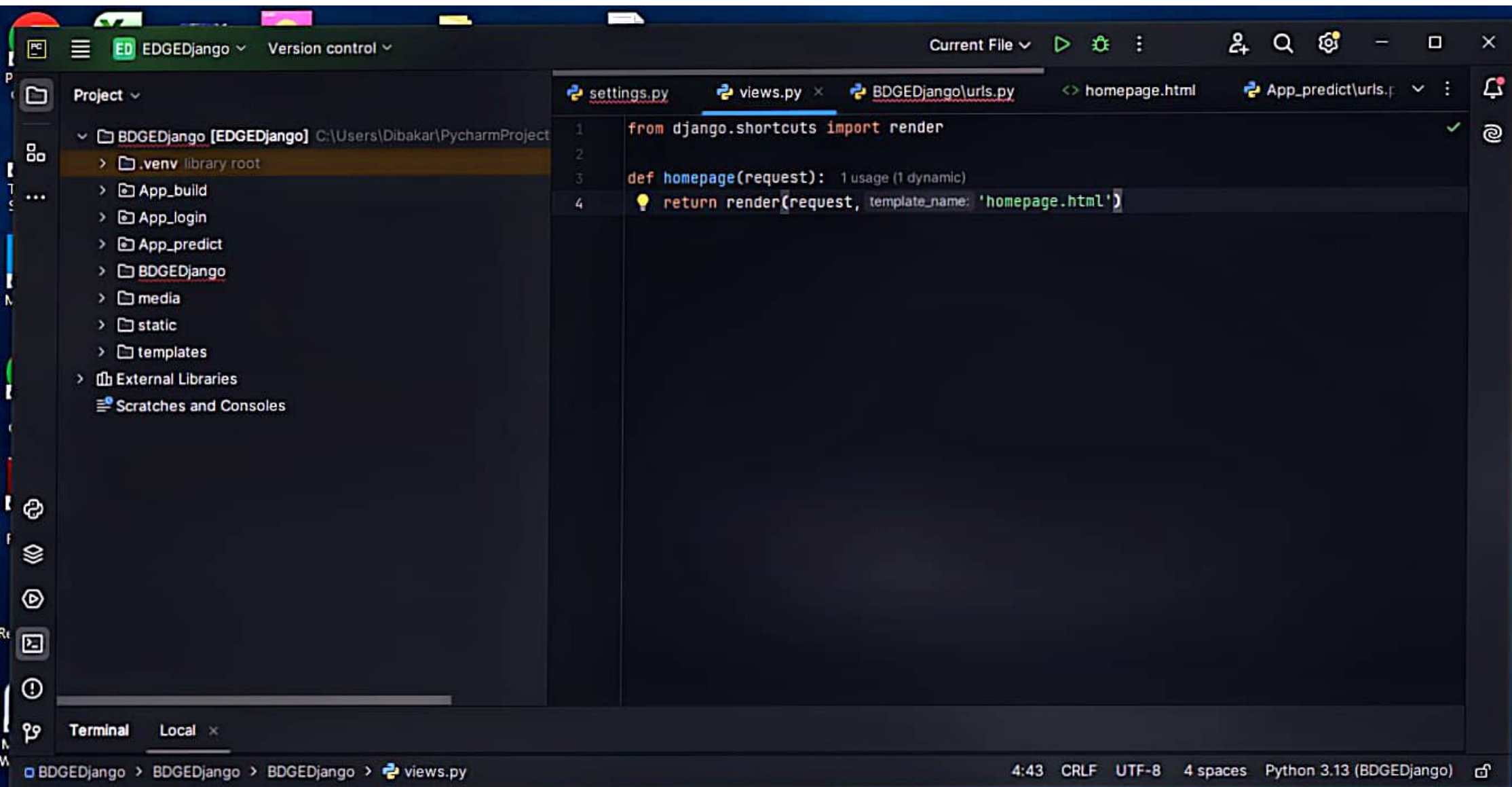
ity="sha384-GnS384xqQ1ac
/rE9/Qpg6aAZGJwFDMVNA/Gp
JB+Y1QKtV3Rn7W3mgPxhU9K/S

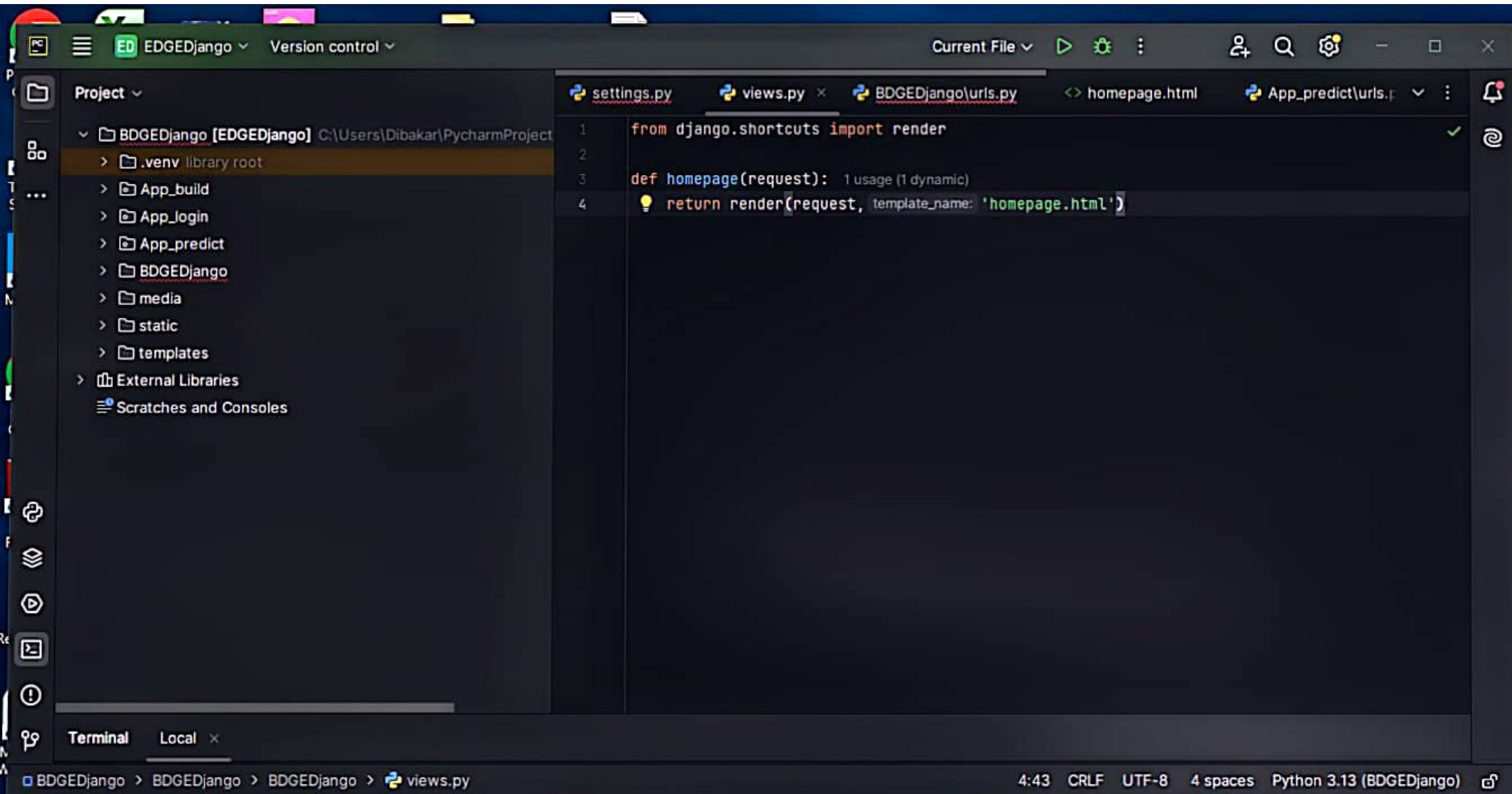












EDGEDjango Version control

Project

- BDGEDjango [EDGEDjango] C:\Users\Dibakar\PycharmProject
 - .venv library root
 - App_build
 - App_login
 - App_predict
 - BDGEDjango
 - media
 - static
 - templates
 - External Libraries
 - Scratches and Consoles

Current File

settings.py views.py BDGEDjango\urls.py x <> homepage.html App_predict\urls.py

https://docs.djangoproject.com/en/5.1/topics/http/urls/ 10

Examples:

Function views

1. Add an import: `from my_app import views`
2. Add a URL to urlpatterns: `path('', views.home, name='home')`

Class-based views

1. Add an import: `from ot`
2. Add a URL to urlpatterns: `path('', home.as_view(), name='home')`

Including another URLconf

1. Import the `include()` function: `from django.urls import include, path`
2. Add a URL to urlpatterns: `path('blog/', include('blog.urls'))`

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

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.homepage, name='homepage'),
    path('App_build/', include('App_build.urls')),
    path('App_predict/', include('App_predict.urls')),
    path('App_login/', include('App_login.urls'))
]
```

Terminal Local x

BDGEDjango > BDGEDjango > BDGEDjango > urls.py 27:42 CRLF UTF-8 4 spaces Python 3.13 (BDGEDjango)

EDGEDjango Version control

Project

- BDGEDjango [EDGEDjango] C:\Users\Dibakar\PycharmProject
 - .venv library root
 - App_build
 - App_login
 - App_predict
 - BDGEDjango
 - media
 - static
 - templates
 - External Libraries
 - Scratches and Consoles

Current File

settings.py views.py BDGEDjango\urls.py homepage.html App_predict\urls.py

5 <https://docs.djangoproject.com/en/5.1/topics/http/urls/>

6 Examples:

7 Function views

8 1. Add an import: `from my_app import views`

9 2. Add a URL to urlpatterns: `path('', views.home, name='home')`

10 Class-based views

11 1. Add an import: `from other_app.views import Home`

12 2. Add a URL to urlpatterns: `path('', Home.as_view(), name='home')`

13 Including another URLconf

14 1. Import the `include()` function: `from django.urls import include, path`

15 2. Add a URL to urlpatterns: `path('blog/', include('blog.urls'))`

16 """

17 `from django.contrib import admin`

18 `from django.contrib.staticfiles.urls import staticfiles_urlpatterns`

19 `from django.urls import path`

20

21

22 urlpatterns = [

23 `path('admin/', admin.site.urls),`

24 `path('', views.homepage, name='homepage')`

25 `path('App_build/', include('App_build.urls'))`

26 `path('App_predict/', include('App_predict.urls'))`

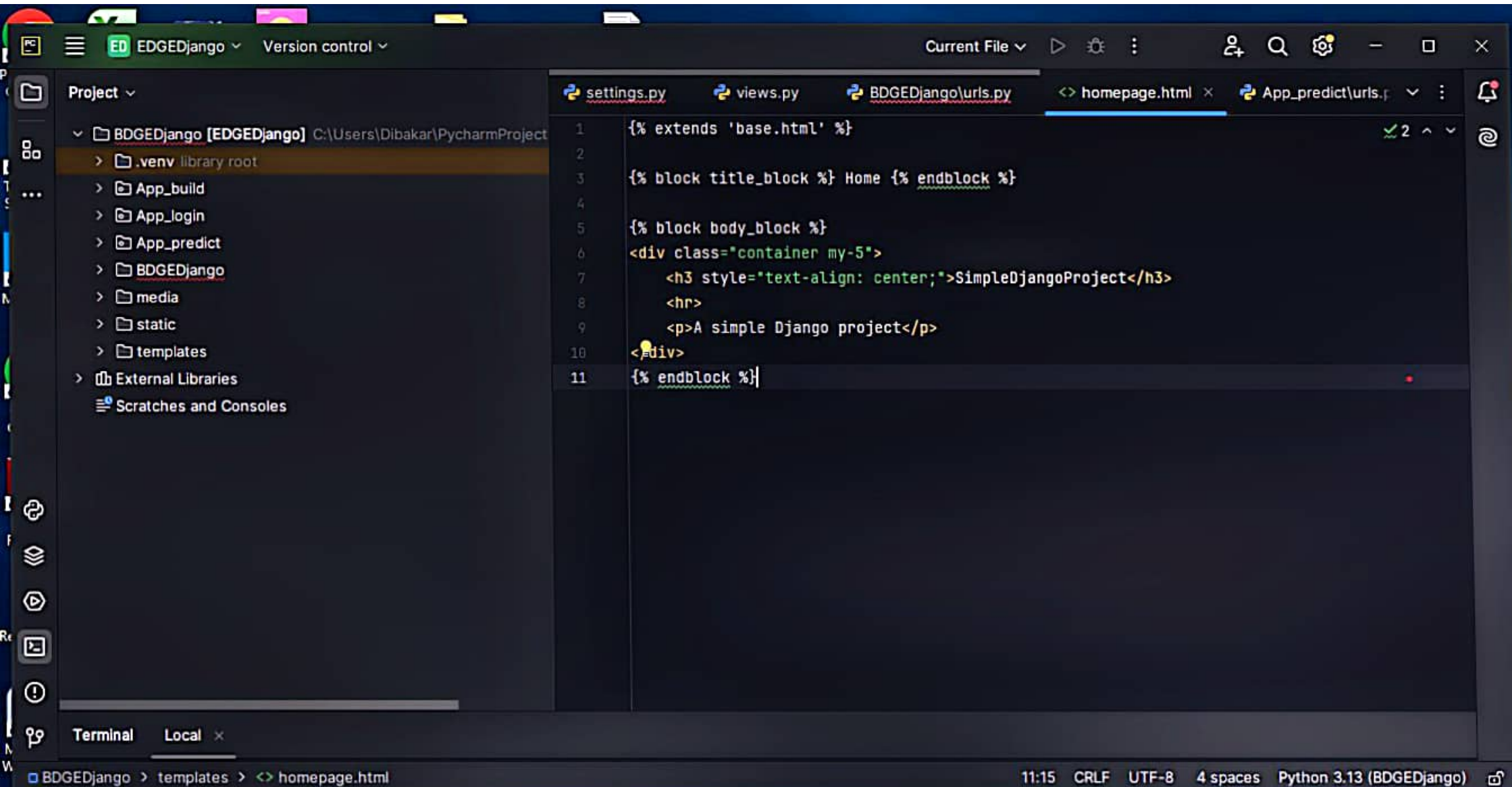
27 `path('App_login/', include('App_login.urls'))`

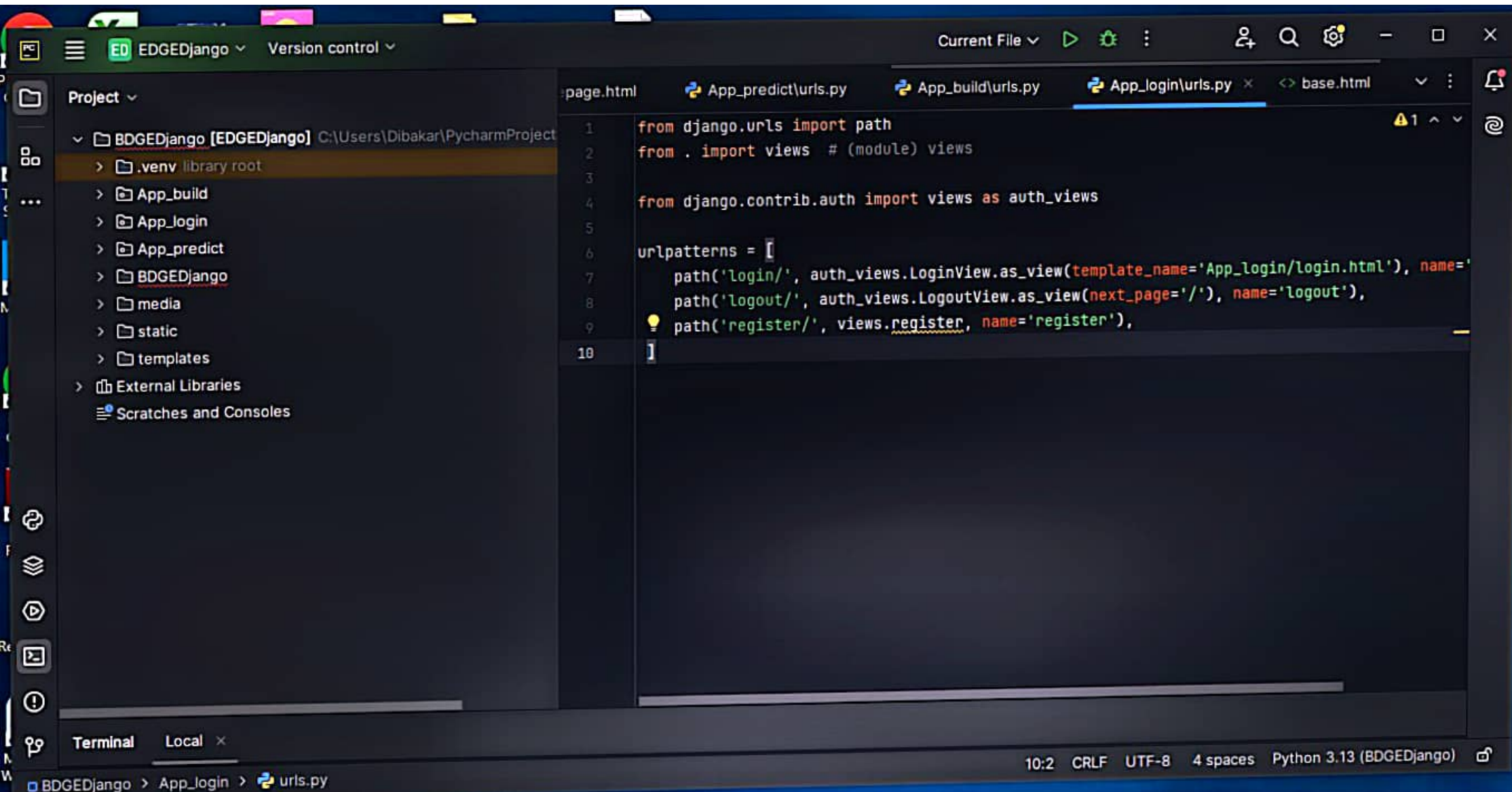
28]

Terminal Local

BDGEDjango > BDGEDjango > BDGEDjango > urls.py

27:42 CRLF UTF-8 4 spaces Python 3.13 (BDGEDjango)





EDGEDjango Version control

Project

- BDGEDjango [EDGEDjango] C:\Users\Dibakar\PycharmProject
 - .venv library root
 - App_build
 - App_login
 - App_predict
 - BDGEDjango
 - media
 - static
 - templates
- External Libraries
- Scratches and Consoles

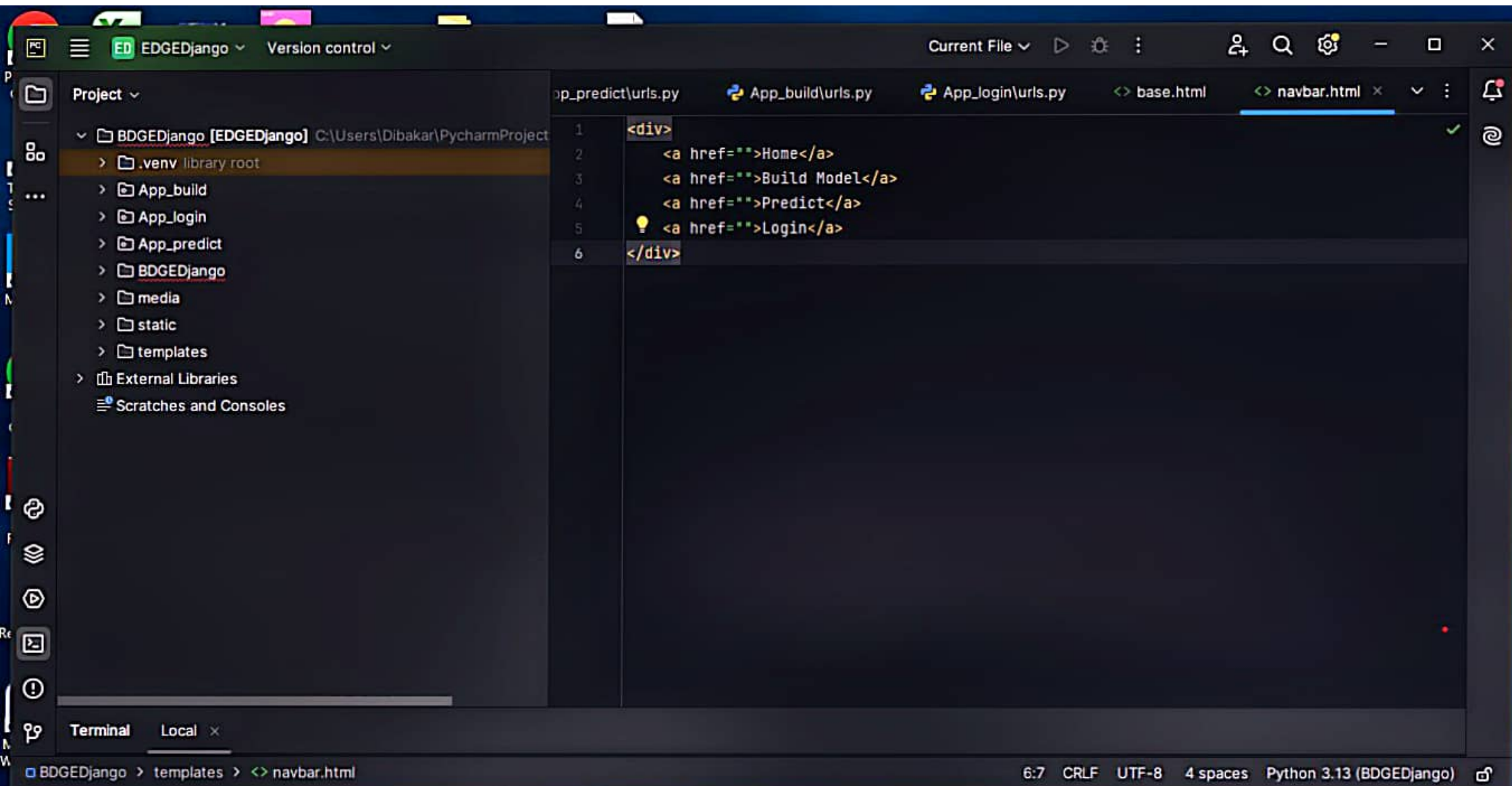
page.html App_predict\urls.py App_build\urls.py App_login\urls.py base.html

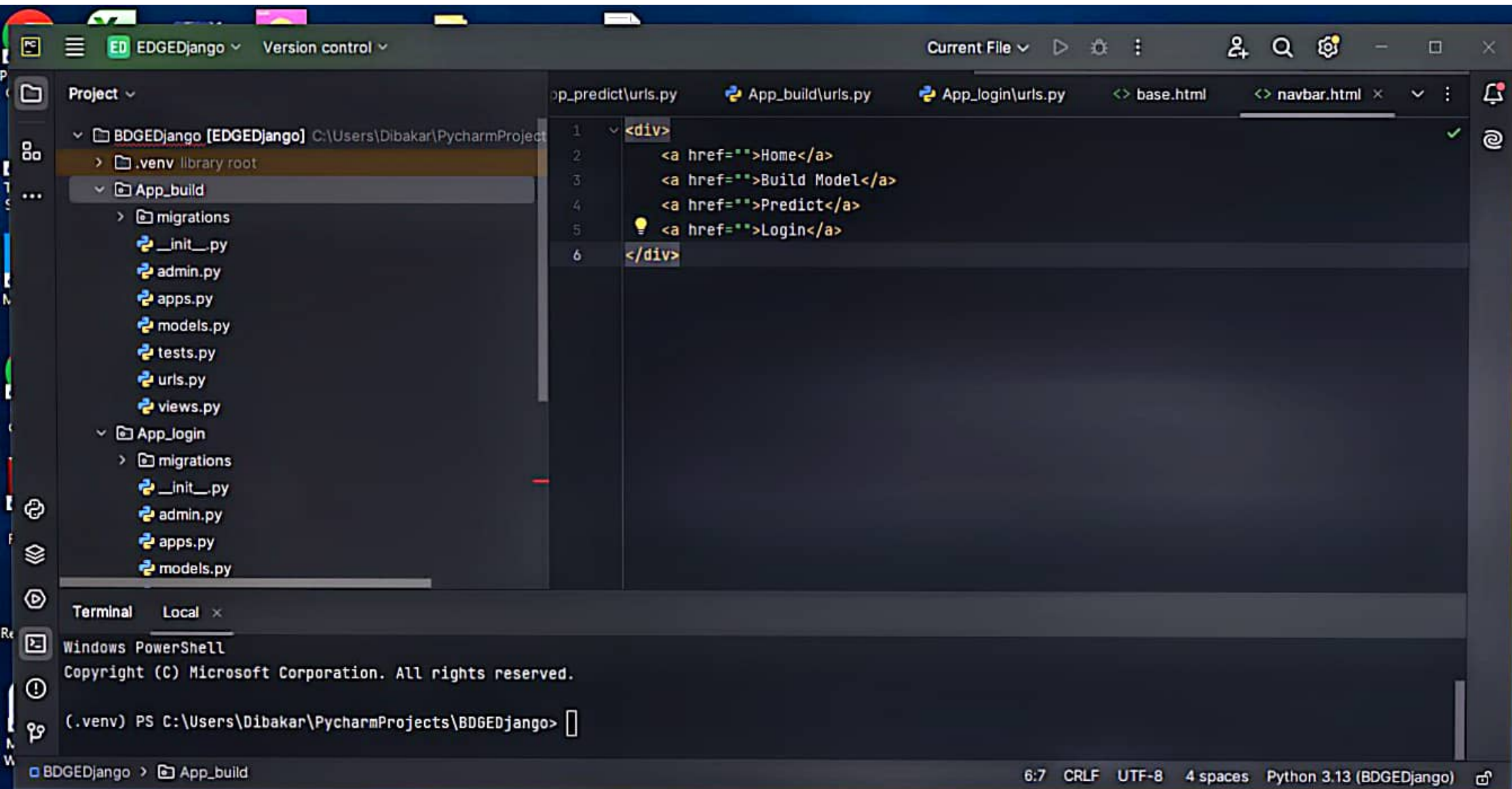
```
1 {% load static %}
2
3 <!DOCTYPE html>
4 <html lang="en" dir="ltr">
5
6 <head>
7     <meta charset="utf-8">
8     <meta name="viewport" content="width=device-width, initial-scale=1.0">
9     <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css"
10         integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
11         crossorigin="anonymous">
12     <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
13         integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCKR/rE9/QpG6aAZGJwFDMVNA/Gp6FF93hXpG5KkN"
14         crossorigin="anonymous"></script>
15     <script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min.js"
16         integrity="sha384-ApNbgh9B+Y1Q40z0gt8j00Xb6t7w3mgPxhU9Sx5zD5p+L6e6L6T2KdY6U6a6K"
17         crossorigin="anonymous"></script>
18     <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/js/bootstrap.min.js"
19         integrity="sha384-JZR6Spejh4U02d8j0t6vLEHfe/JQGiRRSQQx6z6T2KdY6U6a6K"
20         crossorigin="anonymous"></script>
21     <script src="https://kit.fontawesome.com/4c53b33752.js" crossorigin="anonymous"></script>
22     <script type="text/javascript" src="{% static 'js/myjs.js' %}"></script>
23     <link rel="stylesheet" href="{% static 'css/mystyle.css' %}">
24     <title>{% block title_block %}{% endblock %}</title>
```

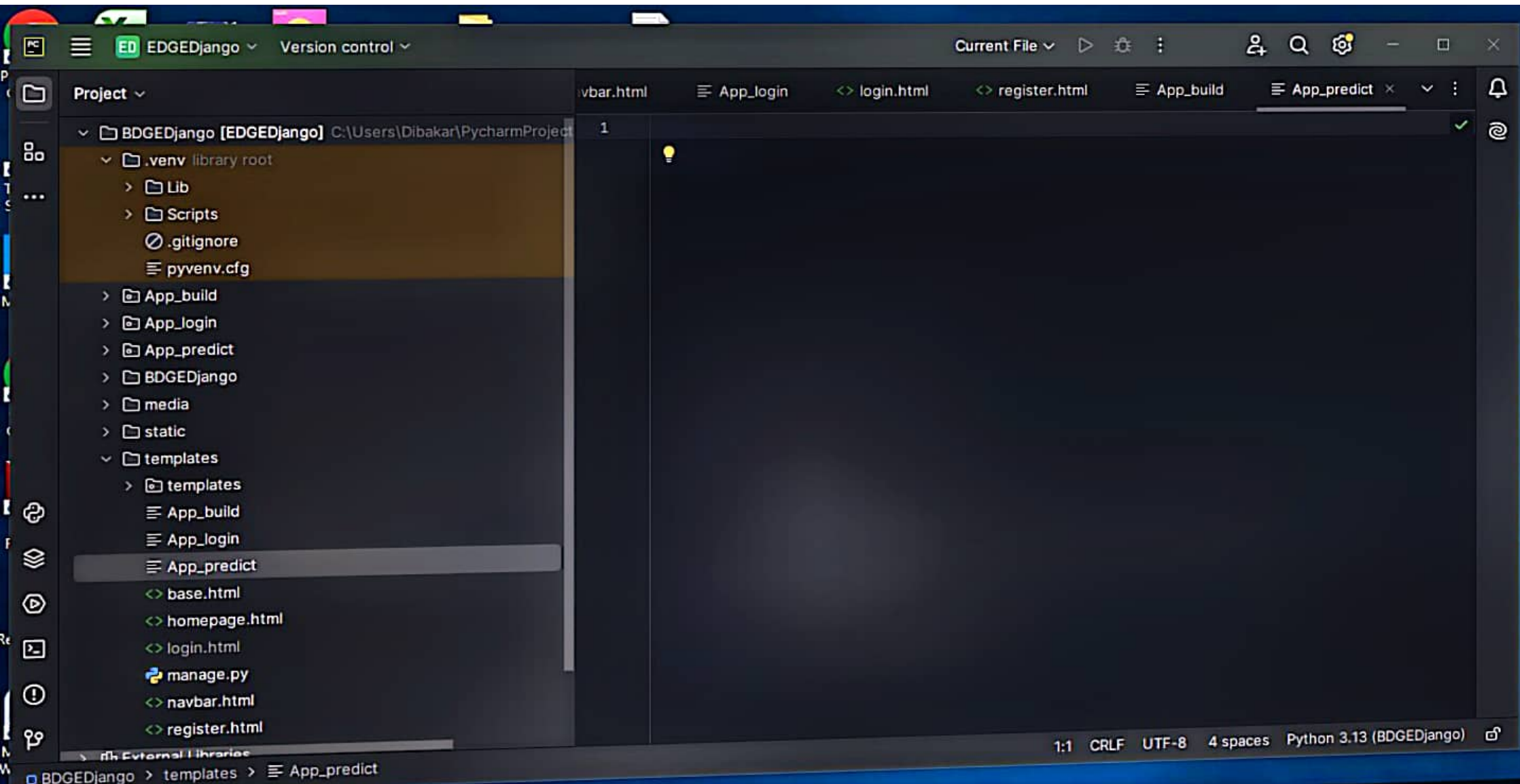
Terminal Local

BDGEDjango > templates > base.html

3:1 CRLF UTF-8 4 spaces Python 3.13 (BDGEDjango)







EDGEDjango Version control

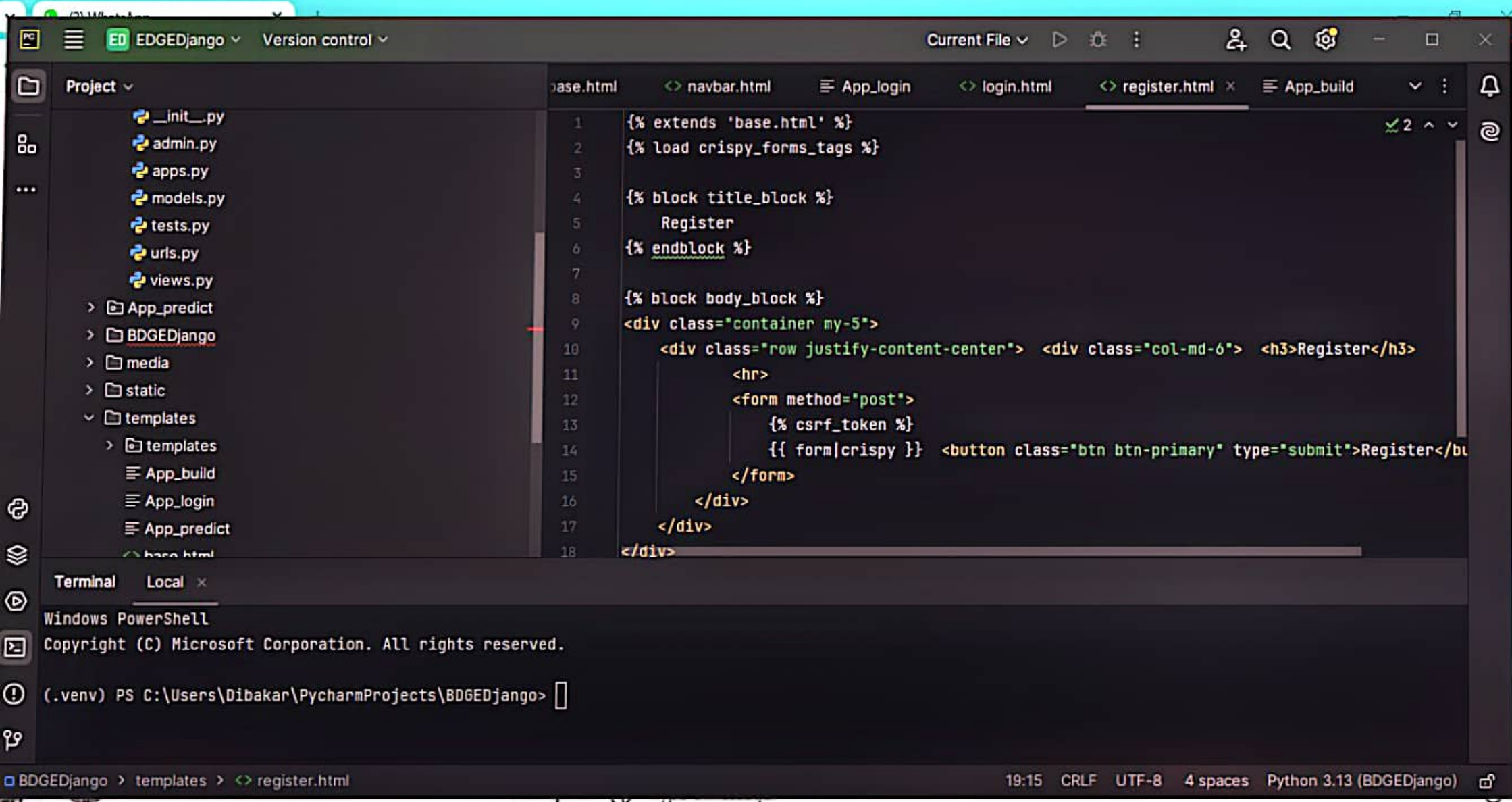
Project

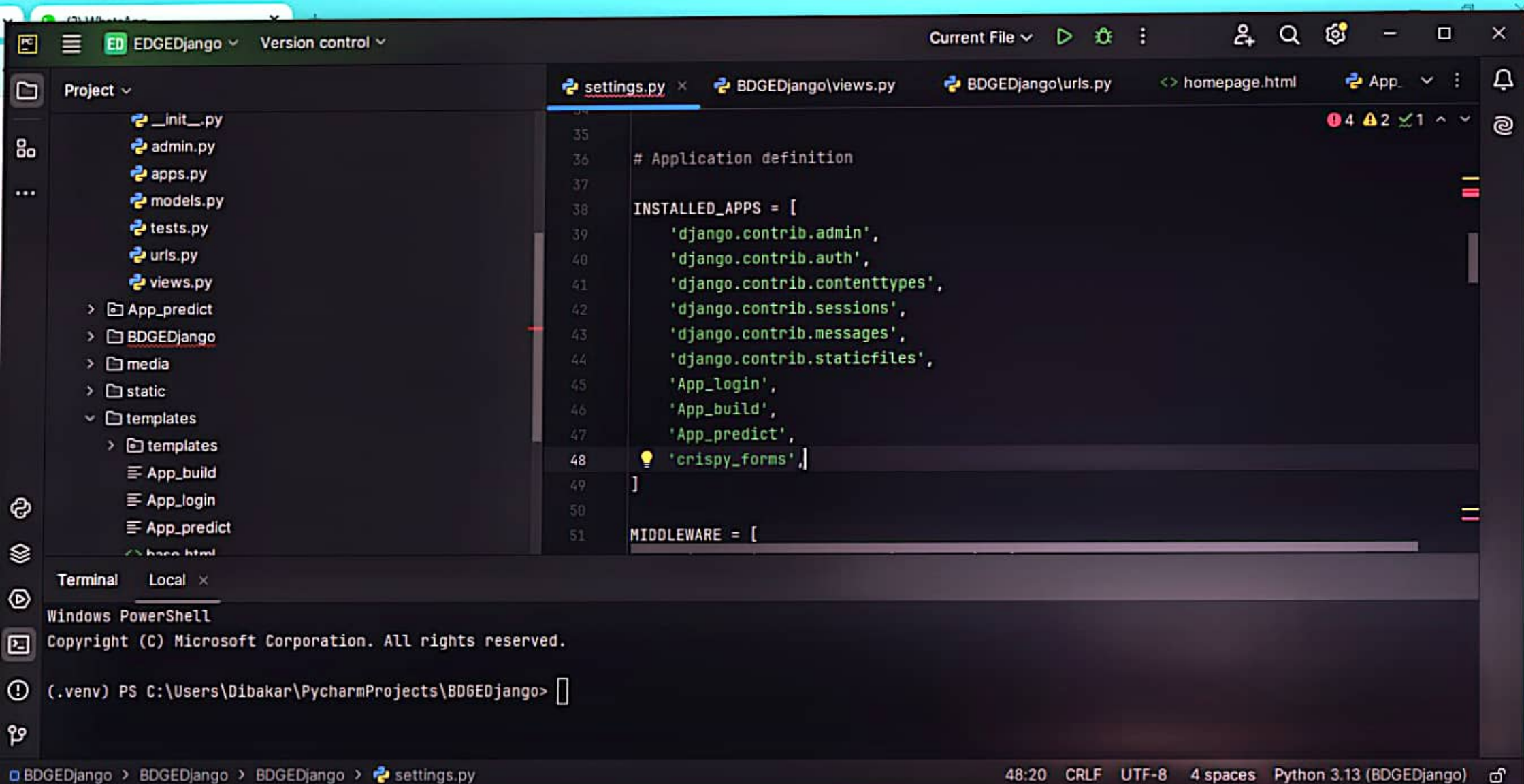
- BDGEDjango [EDGEDjango] C:\Users\Dibakar\PycharmProject
 - .venv library root
 - Lib
 - Scripts
 - .gitignore
 - pyvenv.cfg
 - App_build
 - App_login
 - migrations
 - __init__.py
 - admin.py
 - apps.py
 - models.py
 - tests.py
 - urls.py
 - views.py
 - App_predict
 - BDGEDjango
 - media
 - static
 - templates
 - templates
 - App_build
 - App_login

s.py App_build\urls.py App_login\urls.py App_login\views.py base.html

```
1 from django.shortcuts import render, redirect
2 from django.contrib.auth.forms import UserCreationForm
3 from django.contrib import messages
4
5 def register(request): 1 usage
6     if request.method == 'POST':
7         form = UserCreationForm(request.POST)
8         if form.is_valid():
9             form.save()
10            messages.success(request, message: 'Your account has been created! You can now lo
11            return redirect('login') # Redirect to your login URL
12        else:
13            # Re-render the form with errors (important for user experience)
14            return render(request, template_name: 'App_login/register.html', context: {'form': fo
15
16 else: # Handle GET requests (when the user first visits the page)
17     form = UserCreationForm() # Create a blank form
18     return render(request, template_name: 'App_login/register.html', context: {'form': form})
```

BDGEDjango > App_login > views.py 18:74 CRLF UTF-8 4 spaces Python 3.13 (BDGEDjango)





EDGEDjango Version control

Project

- apps.py
- models.py
- tests.py
- urls.py
- views.py
- App_predict
- BDGEDjango
- media
- static
- templates
 - App_build
 - App_login
 - App_predict
 - base.html
 - homepage.html

base.html navbar.html App_login login.html register.html App_build

```
3
4 {% block title_block %}
5     Register
6 {% endblock %}
7
8 {% block body_block %}
9     <div class="container my-5">
10         <div class="row justify-content-center"> <div class="col-md-6"> <h3>Register</h3>
11             <hr>
12             <form method="post">
13                 {% csrf_token %}
14                 {{ form|crispy }} <button class="btn btn-primary" type="submit">Register</button>
15             </form>
16             <p>Don't have account? Register <a href="{% url 'register' %}">here</a></p>
17         </div>
18     </div>
19 </div>
20 {% endblock %}
```

Terminal Local x

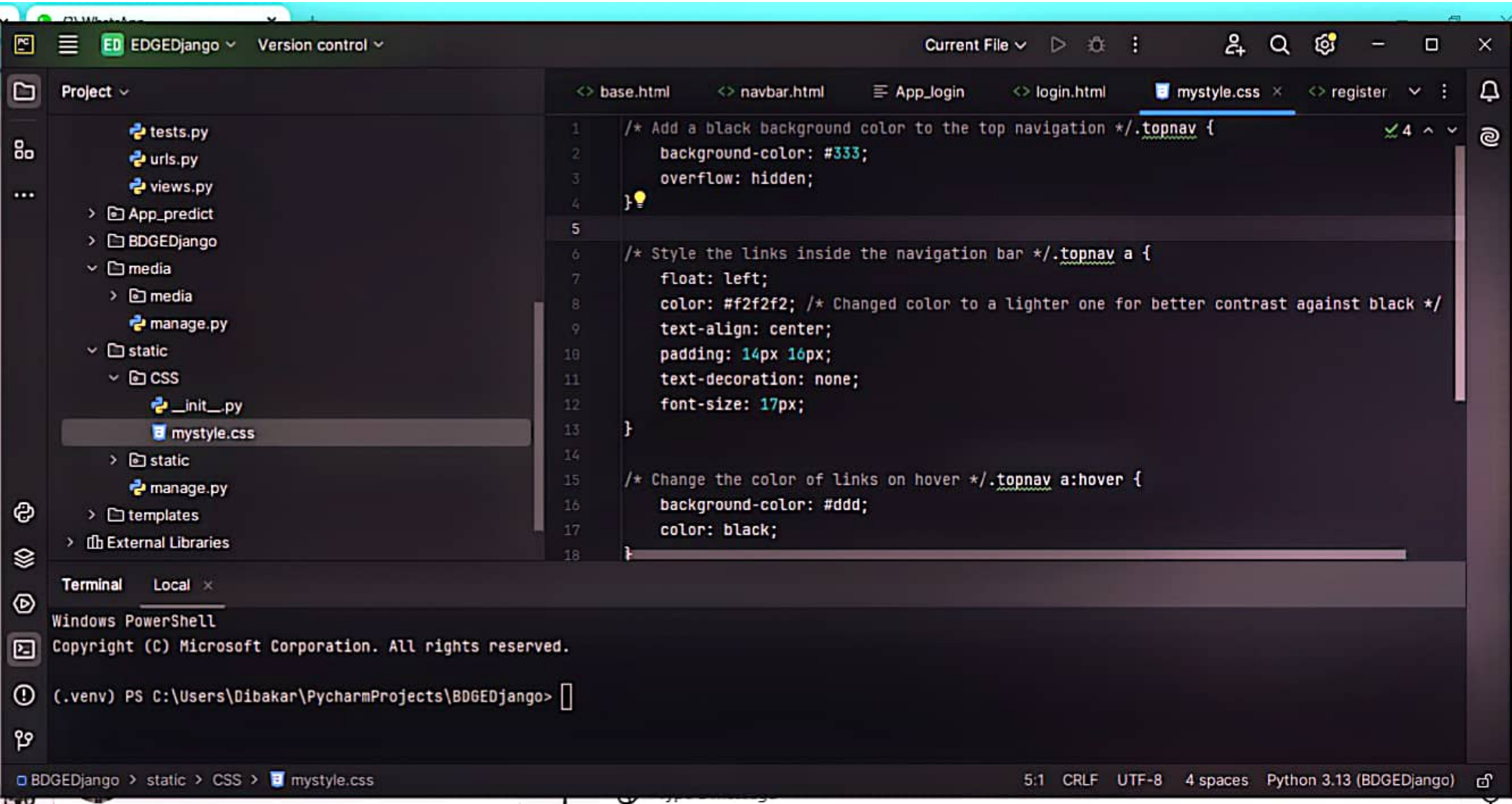
Windows PowerShell

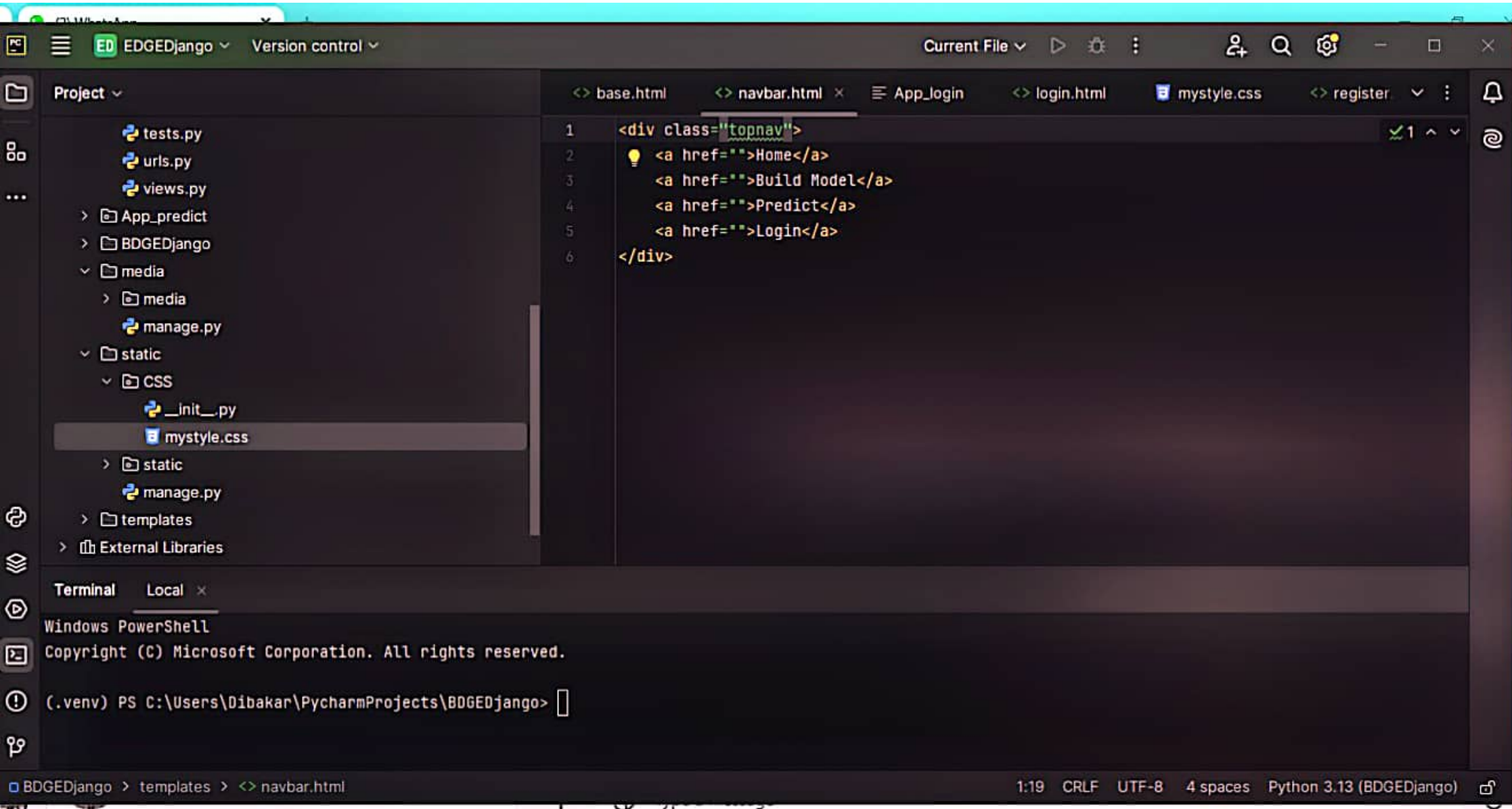
Copyright (C) Microsoft Corporation. All rights reserved.

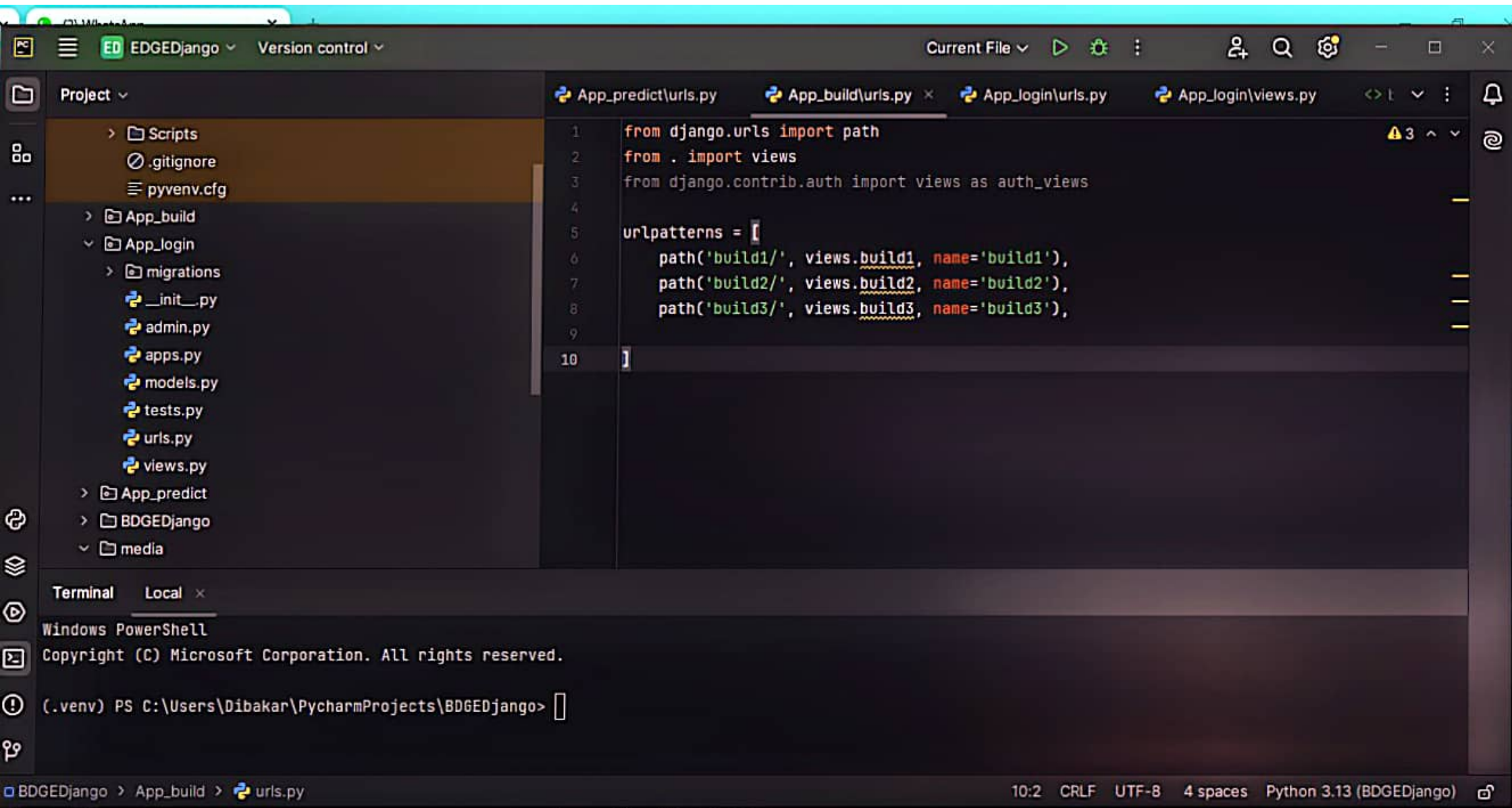
(.venv) PS C:\Users\Dibakar\PycharmProjects\BDGEDjango>

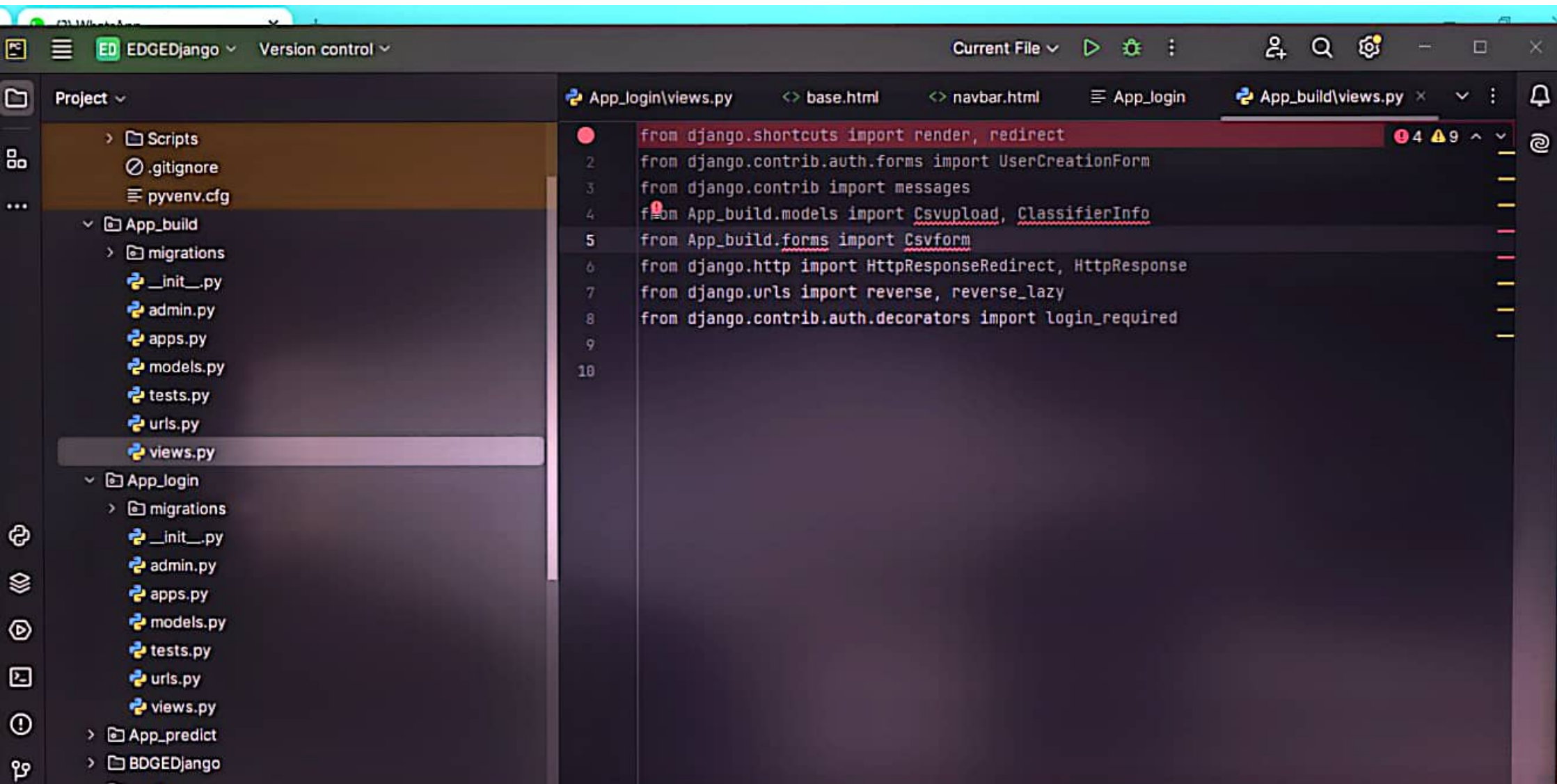
BDGEDjango > templates > login.html

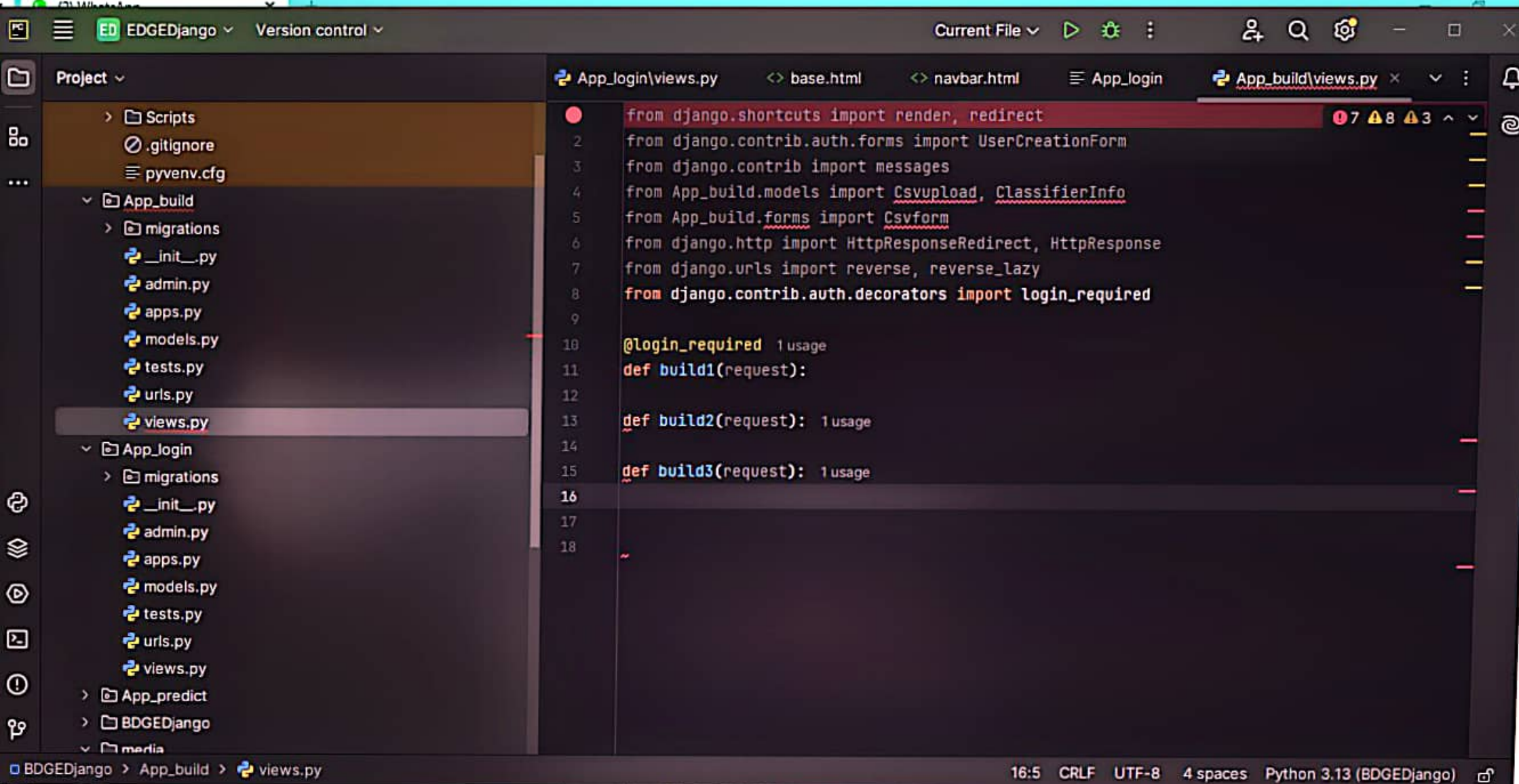
16:76 CRLF UTF-8 4 spaces Python 3.13 (BDGEDjango)





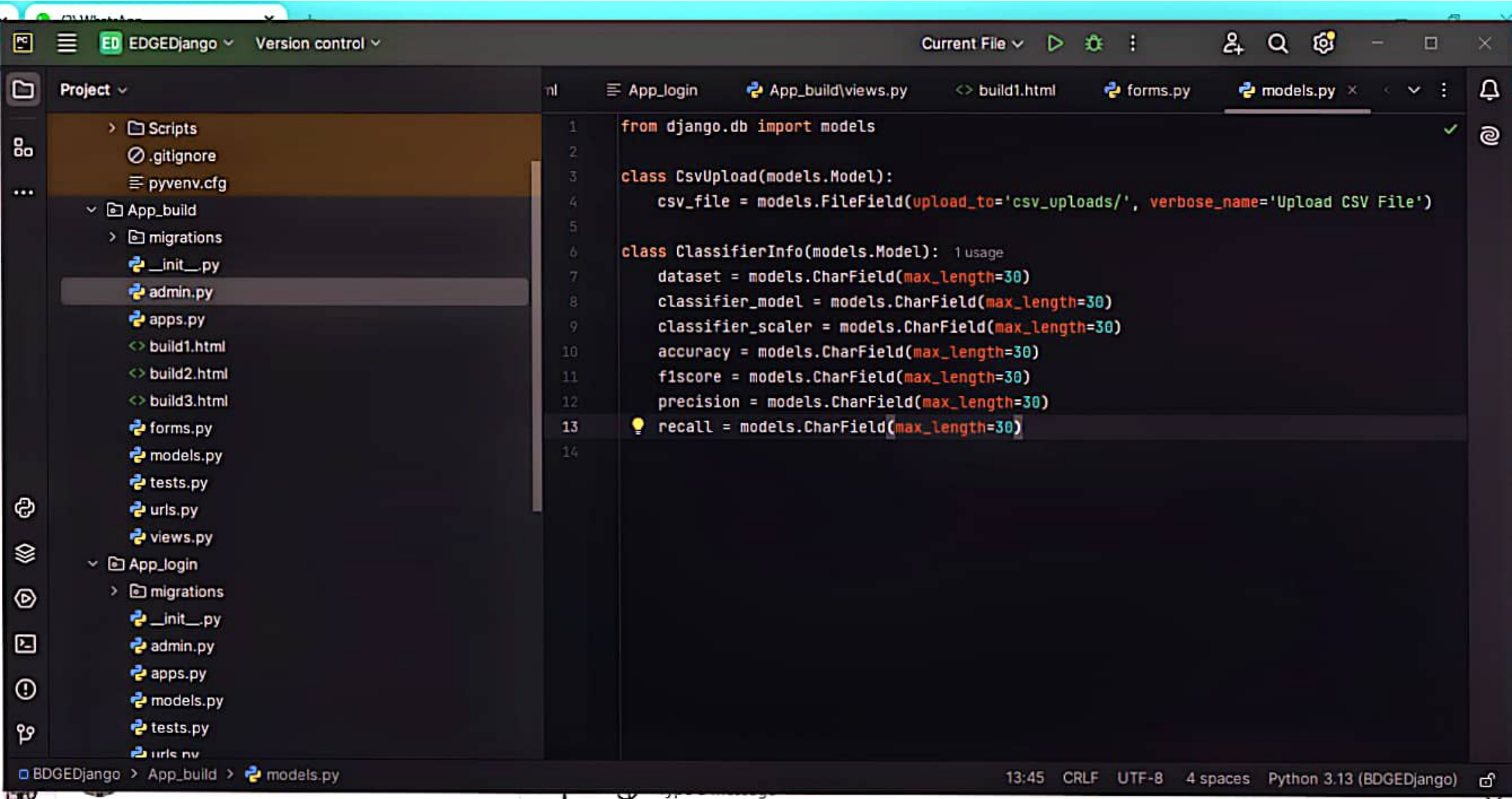






The screenshot shows a code editor with a Django project named 'BDGEDjango'. The left sidebar displays the project structure, including folders for 'Scripts', 'App_build', and 'App_login', and various Python files like 'migrations', 'models.py', 'tests.py', 'urls.py', and 'views.py'. The main editor area shows the content of 'build3.html' in the 'App_build' app. The template extends 'base.html' and loads 'crispy_forms_tags'. It defines a 'title_block' named 'Build1' and a 'body_block' containing a form for uploading a CSV file. The form uses the 'crispy' template engine and includes a submit button labeled 'upload'.

```
1 {% extends 'base.html' %}
2 {% load crispy_forms_tags %}
3
4 {% block title_block %}
5     Build1
6 {% endblock %}
7
8 {% block body_block %}
9     <div class="container my-3">
10         <div class="row">
11             <div class="col-8 my-3">
12                 <h3>Upload a csv file.</h3>
13                 <hr>
14                 <form method="POST" enctype="multipart/form-data" action="{% url 'build1' %}">
15                     {% csrf_token %}
16                     <fieldset>
17                         {{ form|crispy }}
18                         <input type="submit" value="upload" class="btn btn-primary">
19                     </fieldset>
20                 </form>
21             </div>
22         </div>
23     </div>
24 {% endblock %}
```

EDGEDjango Version control

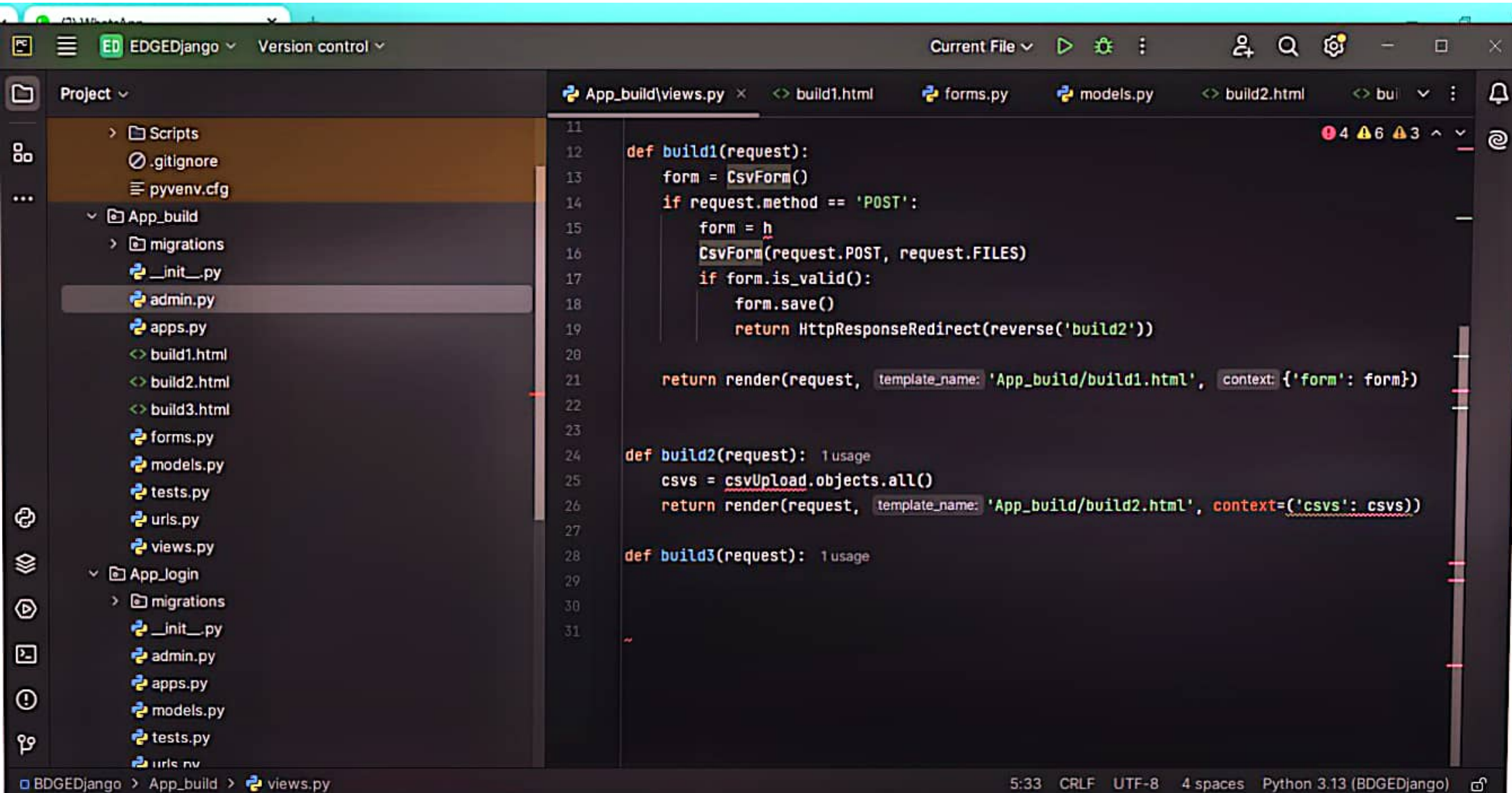
Project

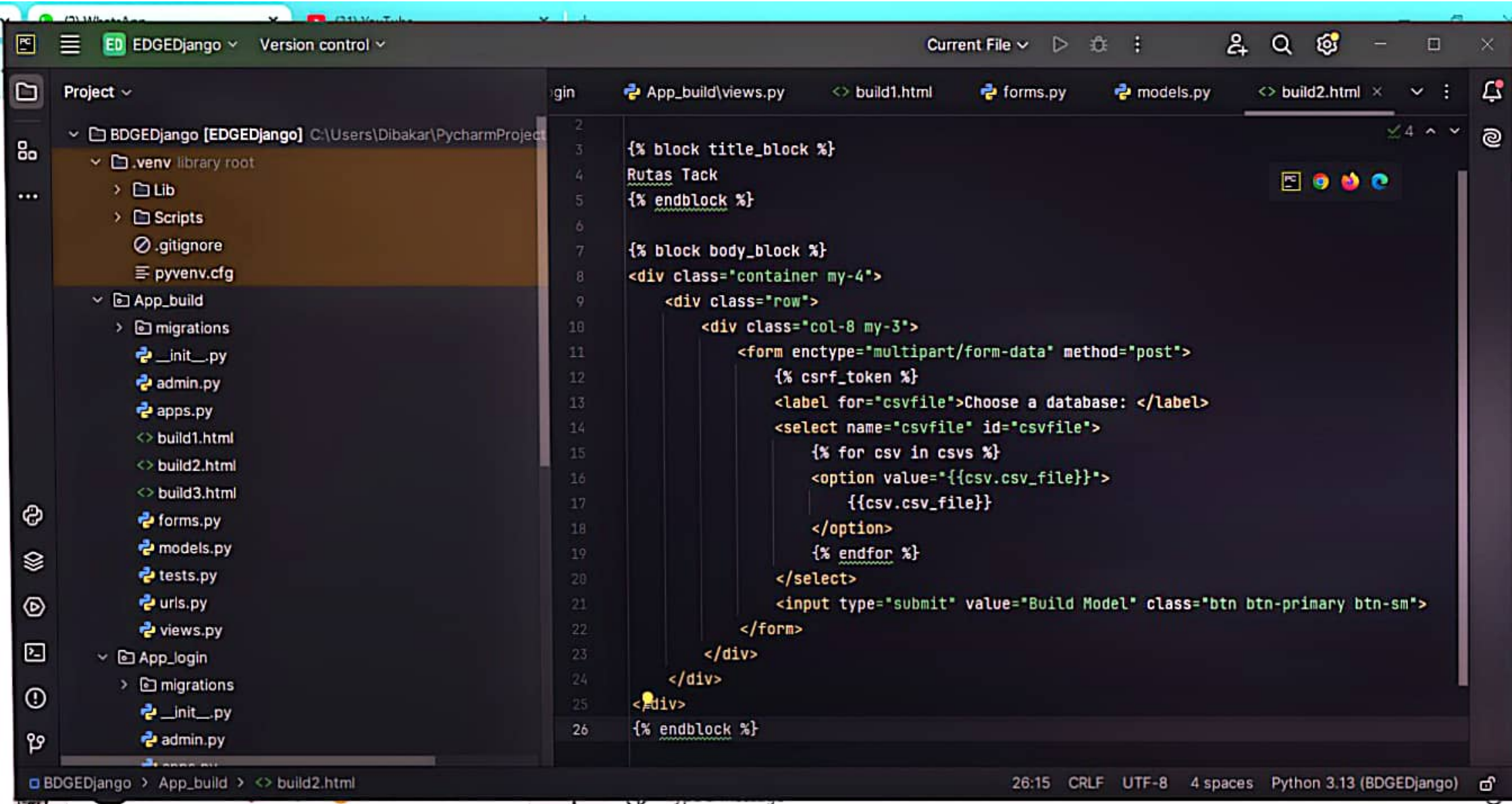
- Scripts
- .gitignore
- pyenv.cfg
- App_build
 - migrations
 - __init__.py
 - admin.py
 - apps.py
 - build1.html
 - build2.html
 - build3.html
 - forms.py
 - models.py
 - tests.py
 - urls.py
 - views.py
- App_login
 - migrations
 - __init__.py
 - admin.py
 - apps.py
 - models.py
 - tests.py
 - urls.py

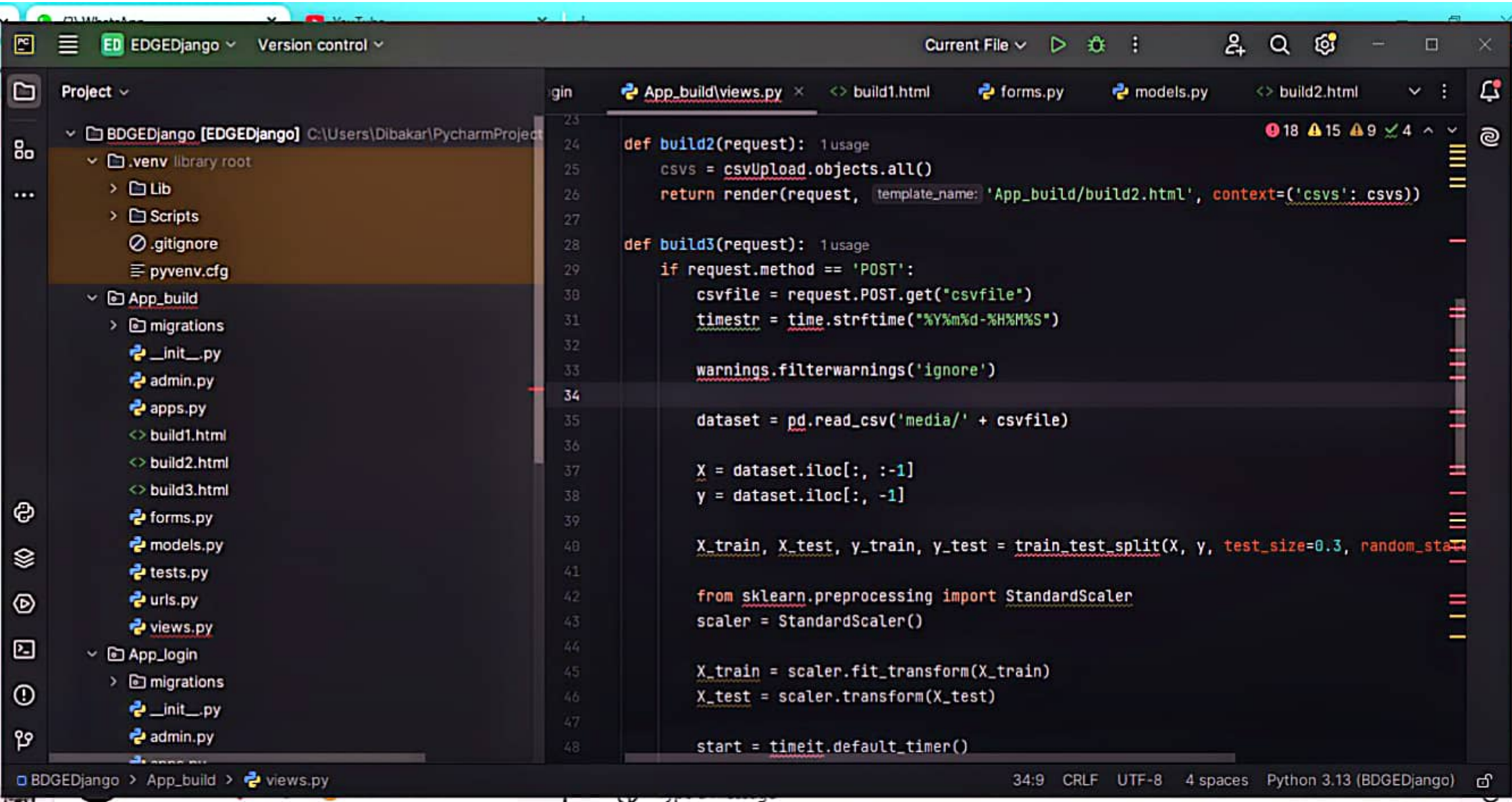
App_login App_build\views.py build1.html forms.py models.py

```
2 from django.contrib.auth.forms import UserCreationForm
3 from django.contrib import messages
4 from App_build.models import Csvupload, ClassifierInfo
5 from App_build.forms import Csvform
6 from django.http import HttpResponseRedirect, HttpResponse
7 from django.urls import reverse, reverse_lazy
8 from django.contrib.auth.decorators import login_required
9
10 @login_required
11
12 def build1(request):
13     form = CsvForm()
14     if request.method == 'POST':
15         form = h
16         CsvForm(request.POST, request.FILES)
17         if form.is_valid():
18             form.save()
19             return HttpResponseRedirect(reverse('build2'))
20
21     return render(request, template_name: 'App_build/build1.html', context: {'form': form})
22
23
24 def build2(request):
25
26 def build3(request):
```

BDGEDjango > App_build > views.py 20:9 CRLF UTF-8 4 spaces Python 3.13 (BDGEDjango)







EDGEDjango Version control

Current File

Project

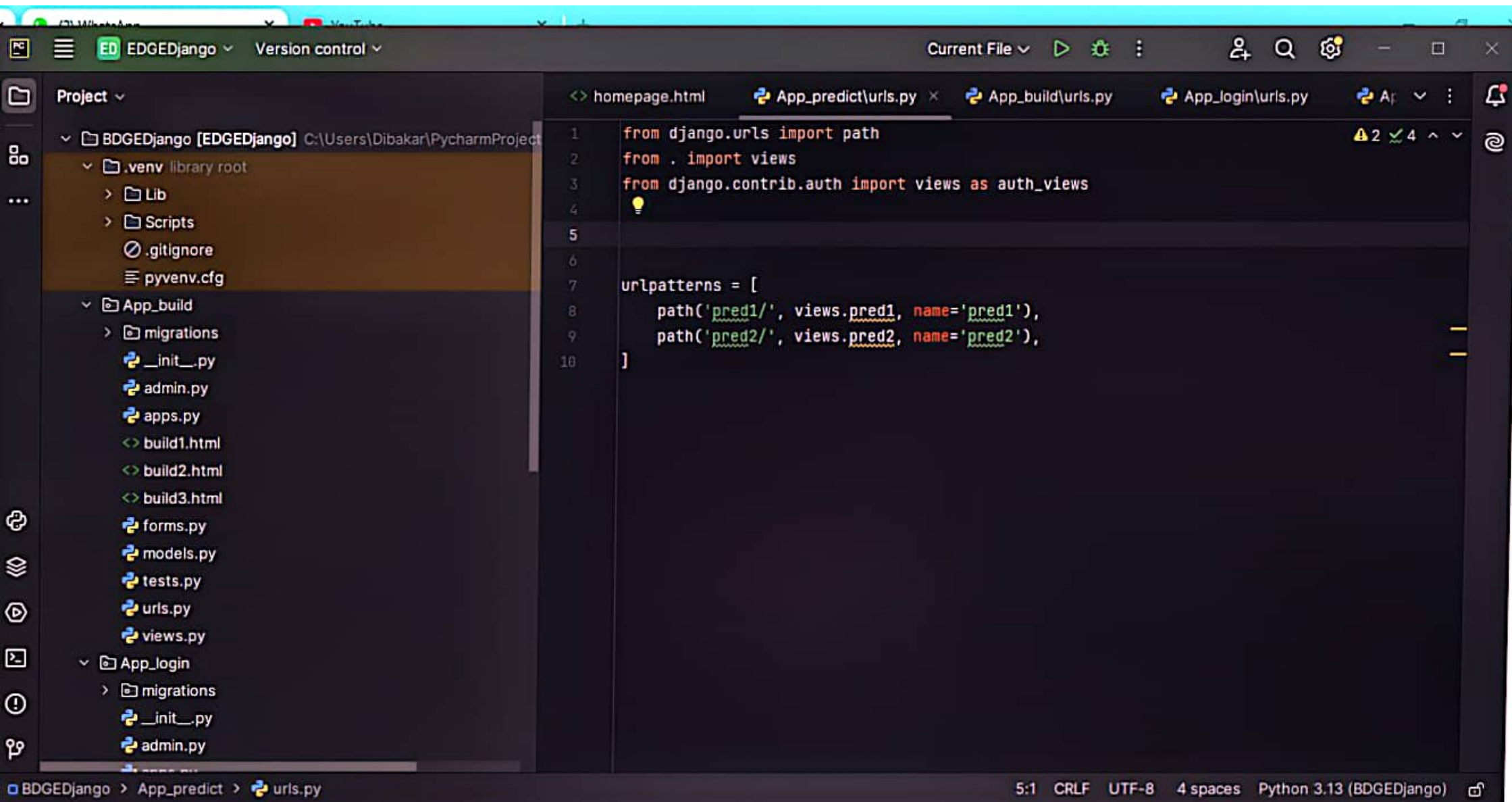
- BDGEDjango [EDGEDjango] C:\Users\Dibakar\PycharmProject
 - .venv library root
 - Lib
 - Scripts
 - .gitignore
 - pyvenv.cfg
 - App_build
 - migrations
 - __init__.py
 - admin.py
 - apps.py
 - build1.html
 - build2.html
 - build3.html
 - forms.py
 - models.py
 - tests.py
 - urls.py
 - views.py
 - App_login
 - migrations
 - __init__.py
 - admin.py

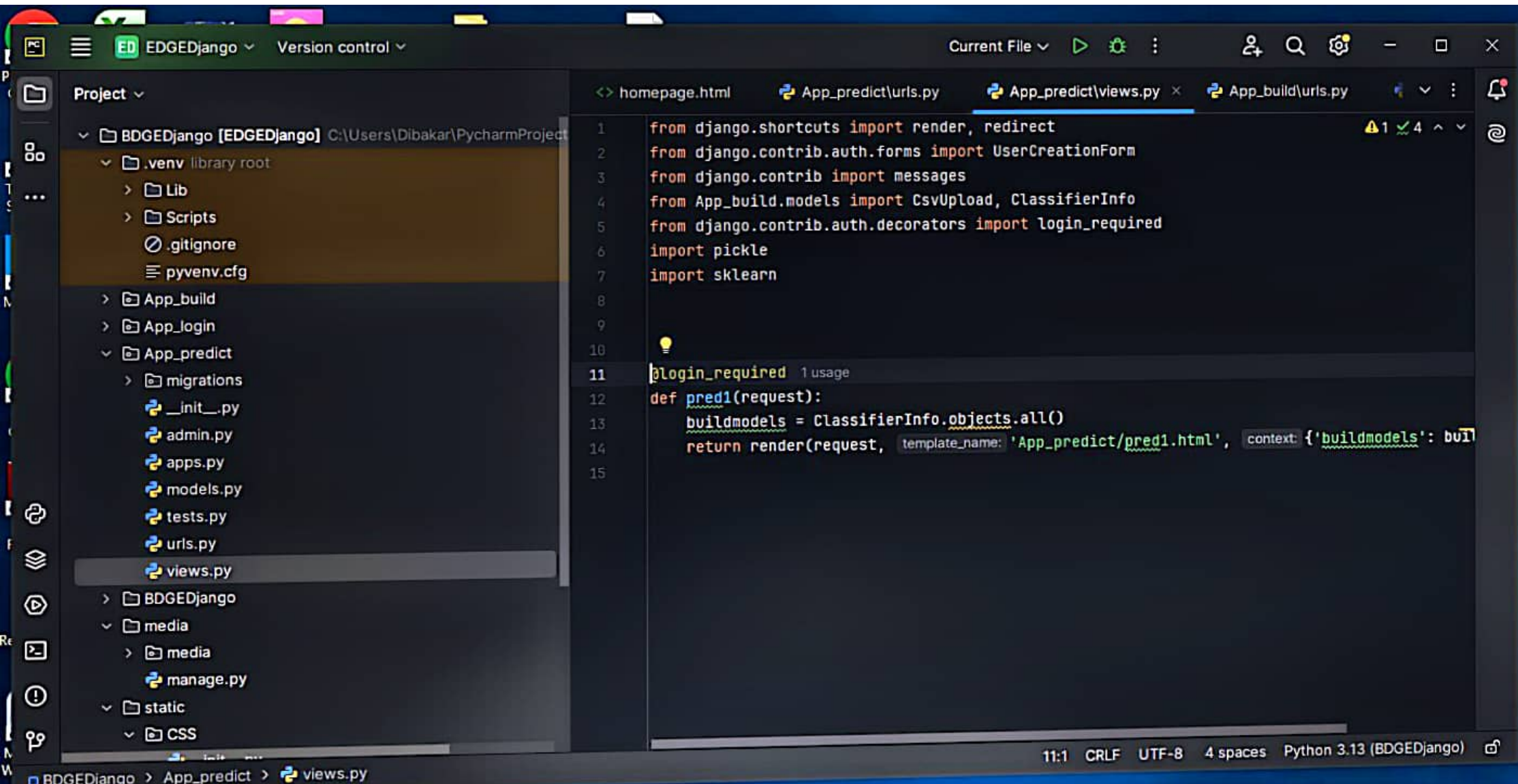
forms.py models.py build2.html build3.html login.html mys

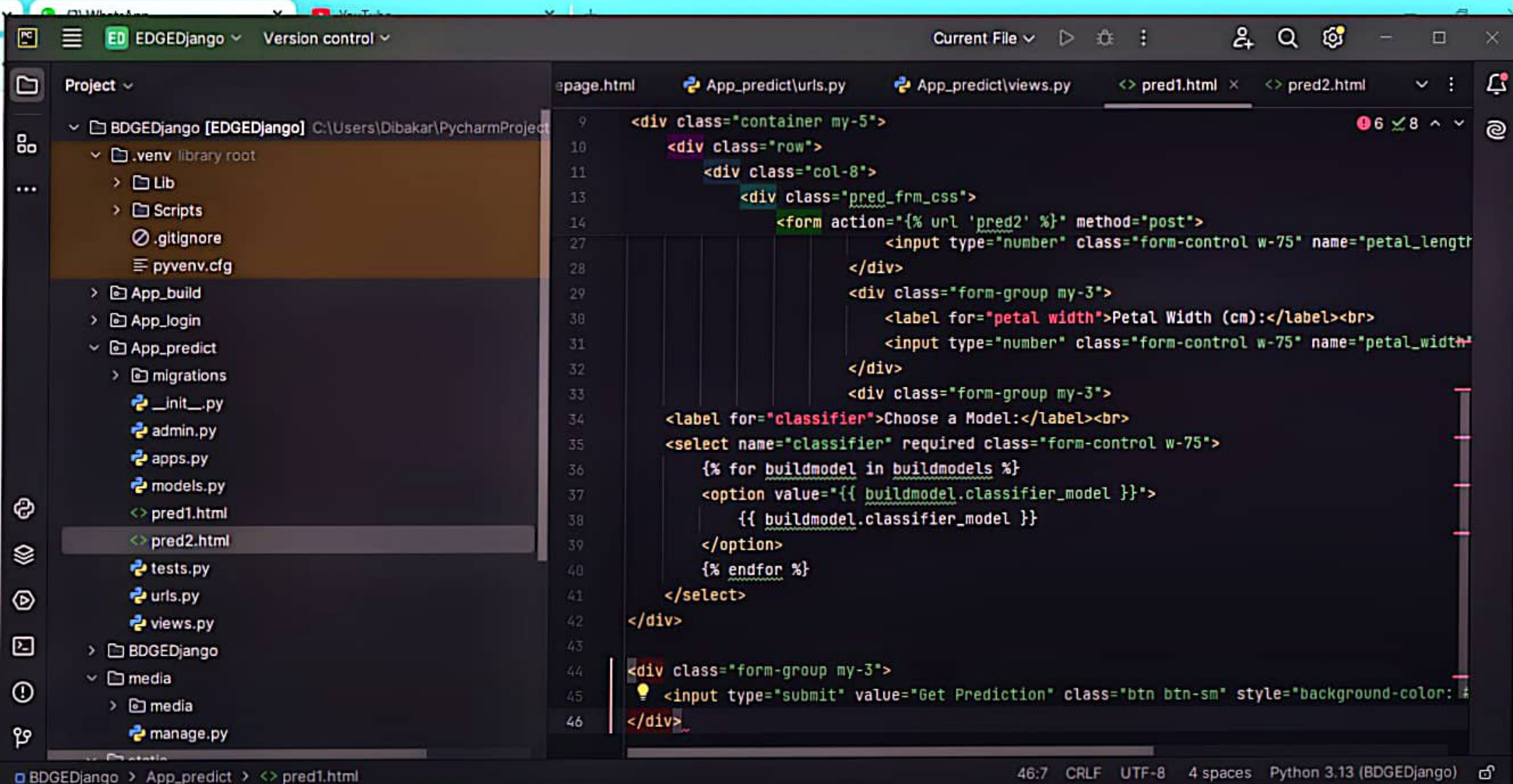
```
1 {% extends 'base.html' %}
2
3 {% block title_block %}
4 Build3
5 {% endblock %}
6
7 {% block body_block %}
8 <div class="container my-4">
9     <h3 style="text-align: center;">Result</h3>
10    <div class="row">
11        <div class="col-8 p-4" style="margin: 8px auto; background-color:#C2DED6; border-top-
12            <h4>Model name: {{model}}</h4>
13            <h5>Precision: {{precision}}</h5>
14            <h5>Recall: {{recall}}</h5>
15            <h5>F1score: {{f1score}}</h5>
16            <h5>Accuracy: {{accuracy}}</h5>
17        </div>
18    </div>
19 </div>
20 {% endblock %}
```

BDGEDjango > App_build > build3.html

20:15 CRLF UTF-8 4 spaces Python 3.13 (BDGEDjango)







EDGEDjango Version control

Project

- BDGEDjango [EDGEDjango] C:\Users\Dibakar\PycharmProject
 - .venv library root
 - Lib
 - Scripts
 - .gitignore
 - pyvenv.cfg
 - App_build
 - App_login
 - App_predict
 - migrations
 - __init__.py
 - admin.py
 - apps.py
 - models.py
 - pred1.html
 - pred2.html
 - tests.py
 - urls.py
 - views.py
 - BDGEDjango
 - media
 - media
 - manage.py

App_predict\urls.py App_predict\views.py pred1.html pred2.html

```
11 @login_required 1 usage
12 def pred1(request):
13     buildmodels = ClassifierInfo.objects.all()
14     return render(request, template_name: 'App_predict/pred1.html', context: {'buildmodels': buildmodels})
15 # custom method for generating predictions
16 def getPredictions(model, scaled, sepal_length, sepal_width, petal_length, petal_width): 1 us
17     prediction = model.predict(scaled.transform([[sepal_length, sepal_width, petal_length, pe
18     return prediction
19
20 # our result page view
21 def pred2(request): 1 usage
22     import sklearn
23     import pickle
24     sepal_length = float(request.GET['sepal_length'])
25     sepal_width = float(request.GET['sepal_width'])
26     petal_length = float(request.GET['petal_length'])
27     petal_width = float(request.GET['petal_width'])
28     classifier = request.GET['classifier']
29
30     model = pickle.load(open("media/" + classifier, "rb"))
31     scaled = pickle.load(open("media/" + "scaler" + classifier, "rb"))
32
33     result = getPredictions(model, scaled, sepal_length, sepal_width, petal_length, petal_wi
34
35     return render(request, template_name: 'App_predict/pred2.html', context: {'result': result})
```

BDGEDjango > App_predict > views.py 35:73 CRLF UTF-8 4 spaces Python 3.13 (BDGEDjango)

