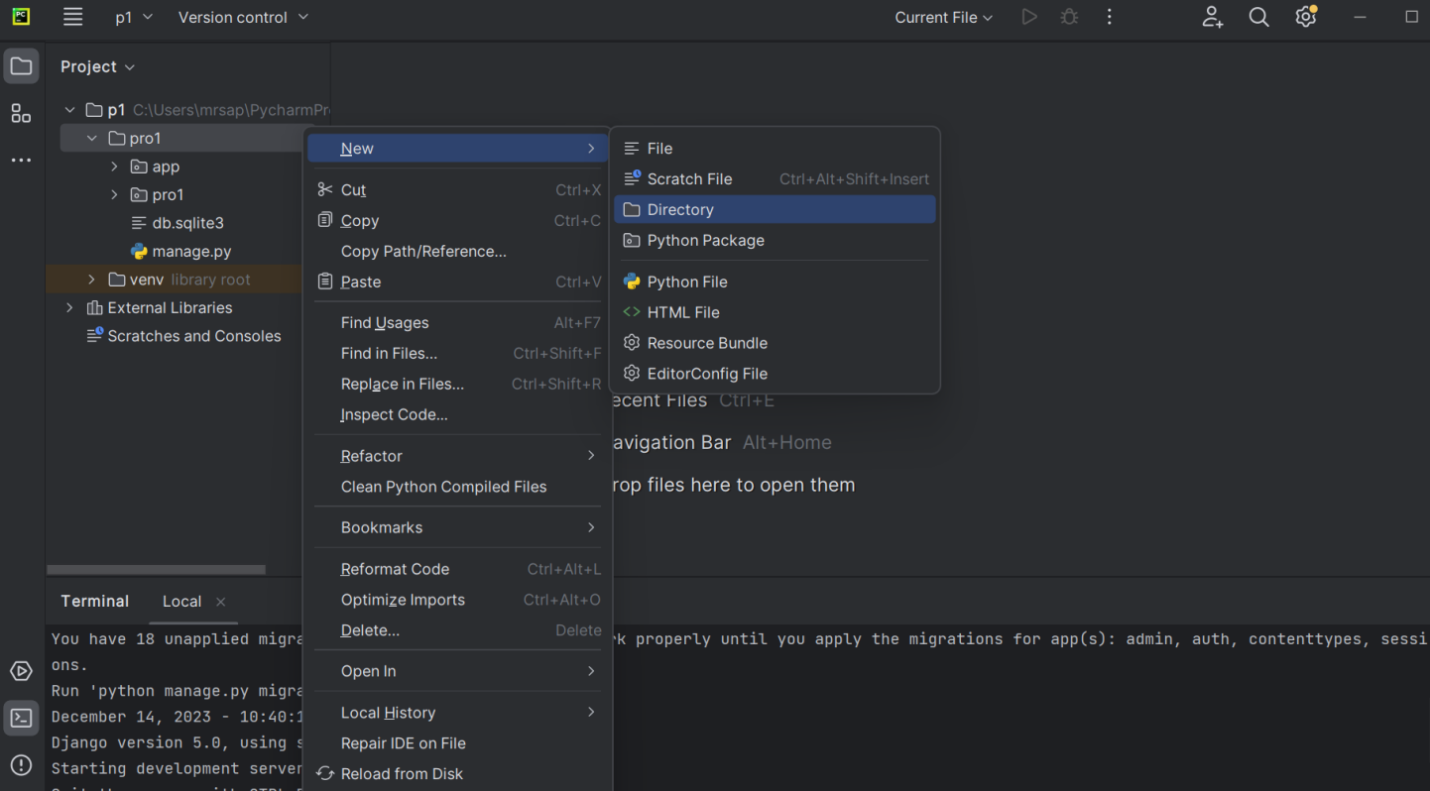
**Templates in Django**

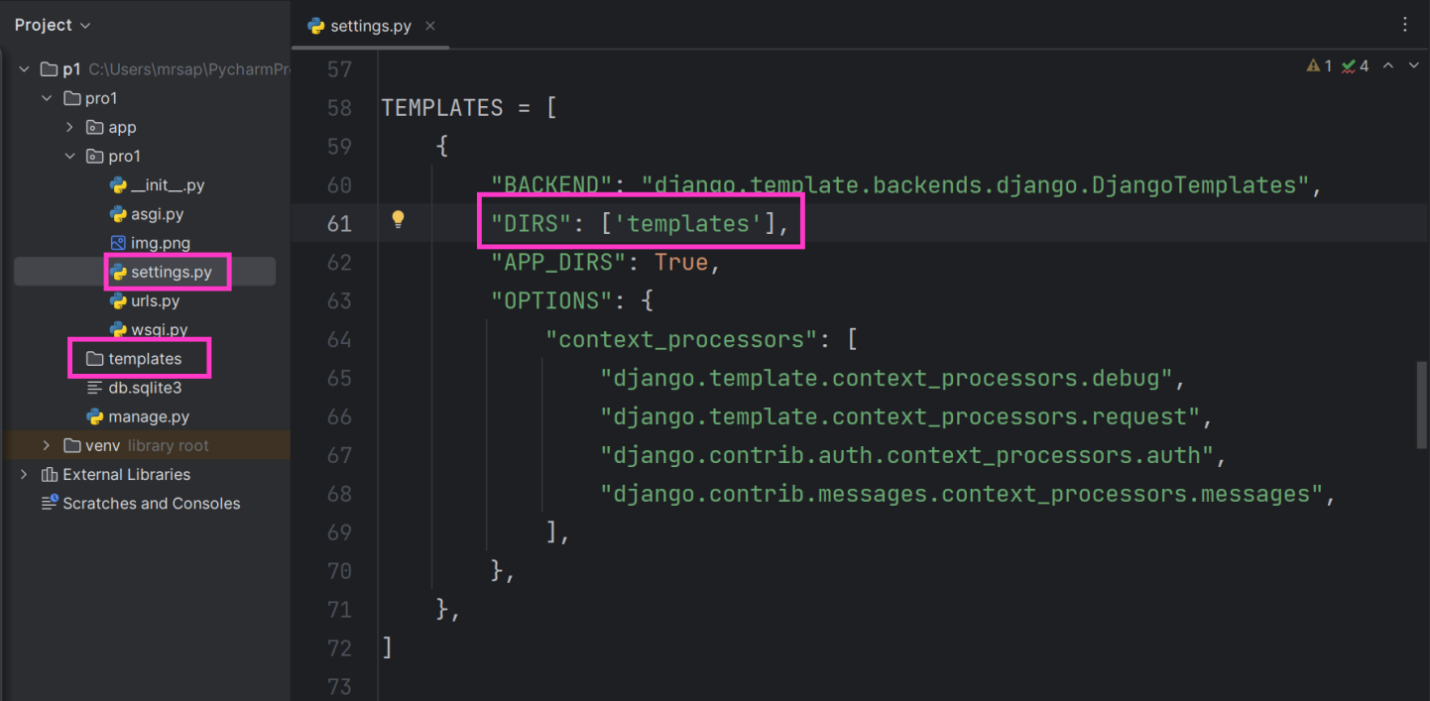
We learned how to return a HttpResponse with a h1 tag from a view. To return a full HTML page, you’ll need to use a templates.

**Step 1:-** we need to create a folder named **templates** inside root directory.



**Step 2:-** Register our templates folder in setting.py inside the TEMPLATES list .

So that Django will search for html files inside templates directory



**Step 3 :-** Click on templates directory create new folder named it same like app so that we can keep html files separate required for diff. apps for maintenance purpose.

A screenshot of a computer

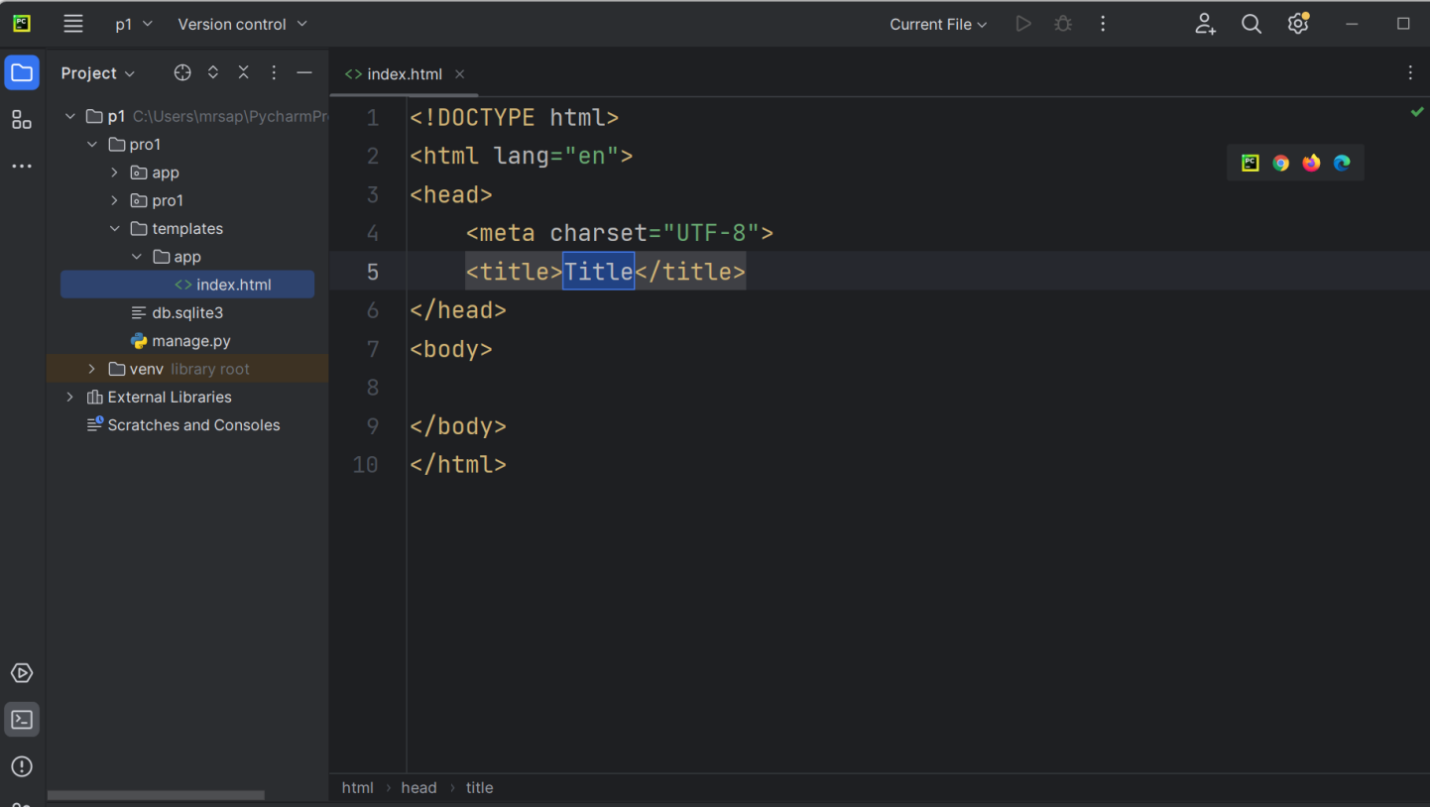
Description automatically generated

**Step 4:-** Right click on app create new html file index.html(you can name it anything)

A screenshot of a computer

Description automatically generated

After creation of html file you will get default code (boilerplate) modify that code.

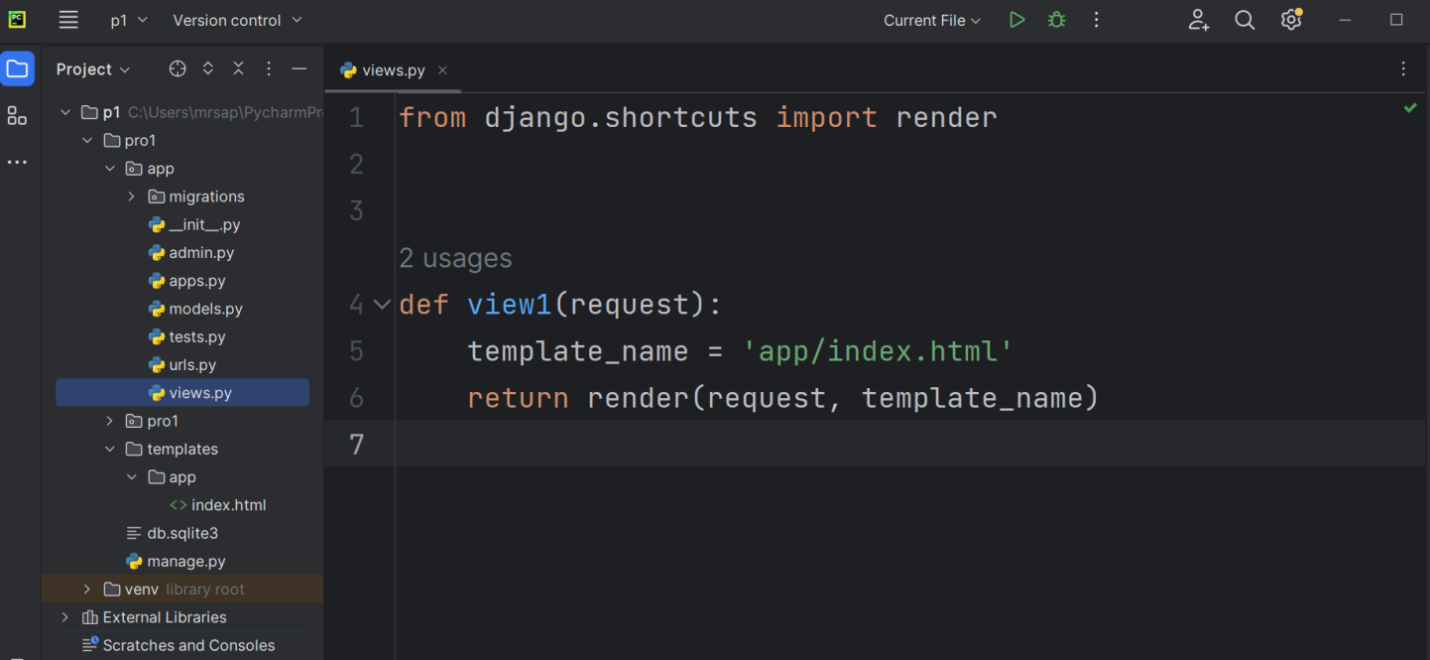


A screen shot of a computer program

Description automatically generated

Now instead of HttpResponse Objcet we are going to use render function which we again internally returning an HttpResponse.

There are two required arguments for render function request and template\_name.



Urls will remain same now runserver and template will get render on browser.

A screenshot of a computer

Description automatically generated

**DTL(Django template language)**

* DTL is a powerful and flexible language that allows you to create dynamic web pages using HTML templates.
* At runtime, the template engine replaces variables in a template file with actual values, and transforms the template into an HTML file sent to the client.

**The Basic Structure of Django Template Language(DTL)**

The DTL syntax is very similar to Python. It consists of :

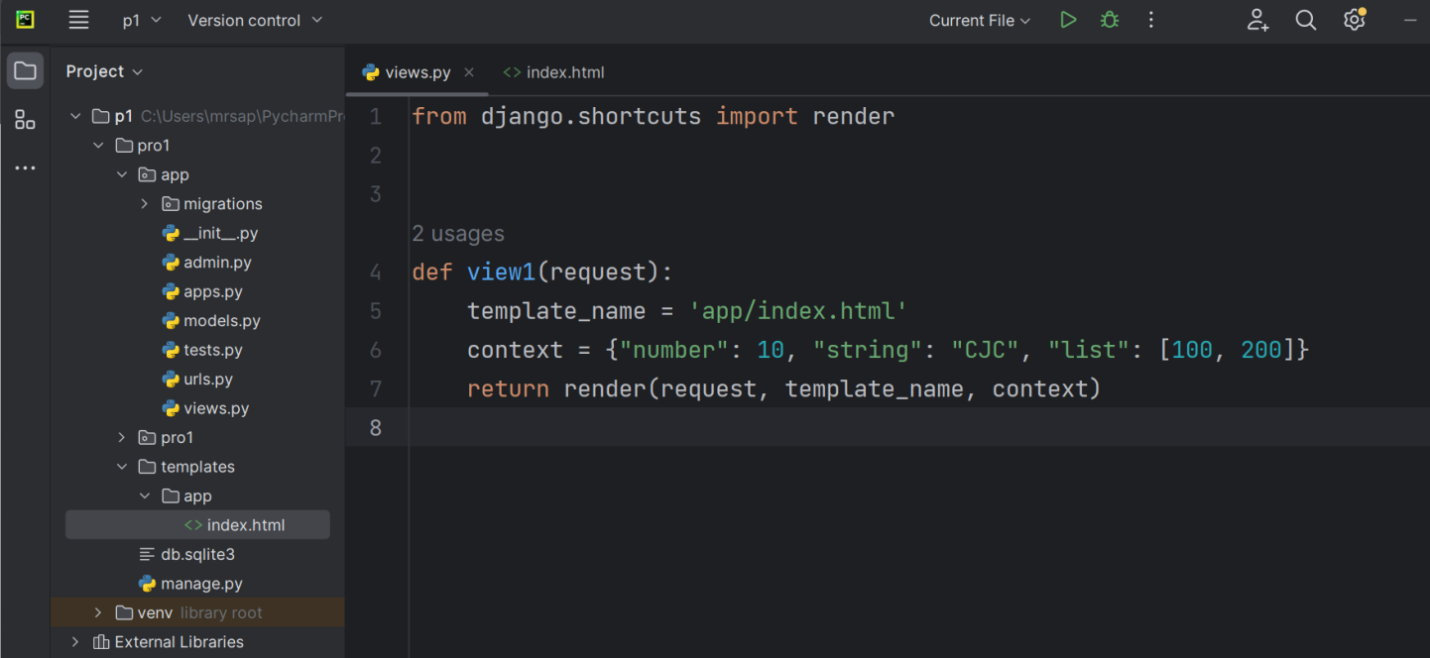
* Template tags --> {% tag %}
* Template variables --> {{ var }}
* Template Filters --> {{ var|lower }}
* Template Comments --> {# #}

The template tags are a way of telling Django that here comes something else than plain HTML. The template tags allows us to to do some programming on the server before sending HTML to the client.

Before using DTL lets learn about context dictionary which is third argument of our render function.

**[Context](https://docs.djangoproject.com/en/2.1/ref/templates/api/" \l "django.template.Context)**

* A [Context](https://docs.djangoproject.com/en/2.1/ref/templates/api/" \l "django.template.Context) is a dictionary with variable names as the **key** and their values as the **value**.
* Variable(Key) names consist of any combination of alphanumeric characters and the underscore ("\_")
* Do not start with an underscore, and number.
* You cannot have spaces or punctuation characters “.” in variable names.



If else and for loop demonstration in below code.

A screenshot of a computer program

Description automatically generated

Keep in mind that 🡪 this is invalid {{ } }, { % %} there should be space between curly braces and % symbol.

A screenshot of a computer

Description automatically generated