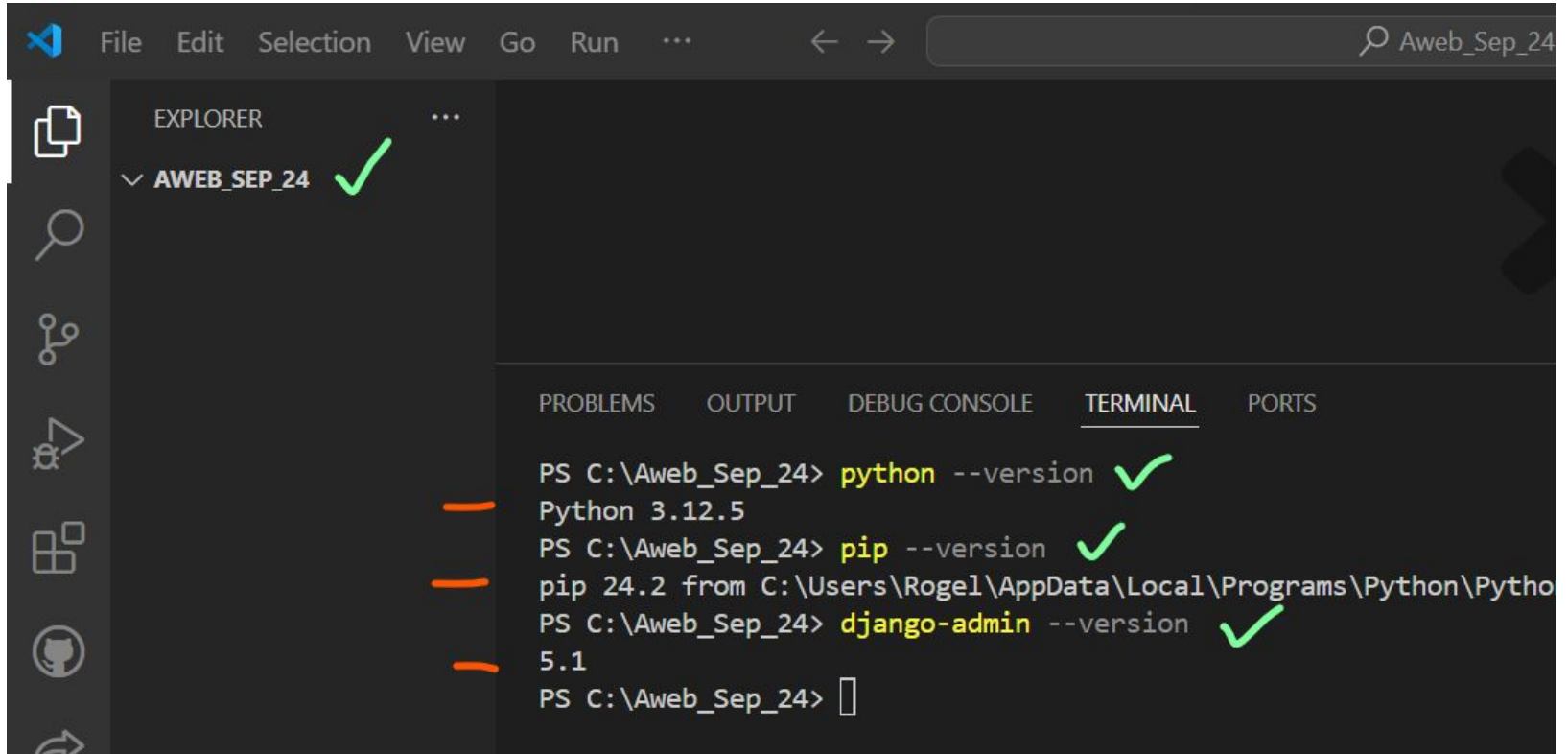


# verificando proyectos



The screenshot shows the Visual Studio Code interface with a dark theme. The Explorer sidebar on the left shows a project named 'AWEB\_SEP\_24' with a green checkmark next to it. The main editor area is divided into two panes. The top pane is empty. The bottom pane is the 'TERMINAL' tab, which shows the output of three commands executed in a PowerShell prompt. Each command is preceded by a red horizontal line. The commands and their outputs are: 1. 'python --version' returns 'Python 3.12.5'. 2. 'pip --version' returns 'pip 24.2 from C:\Users\Rogel\AppData\Local\Programs\Python\Python...'. 3. 'django-admin --version' returns '5.1'. Each output line is followed by a green checkmark.

```
File Edit Selection View Go Run ... < > Aweb_Sep_24
```

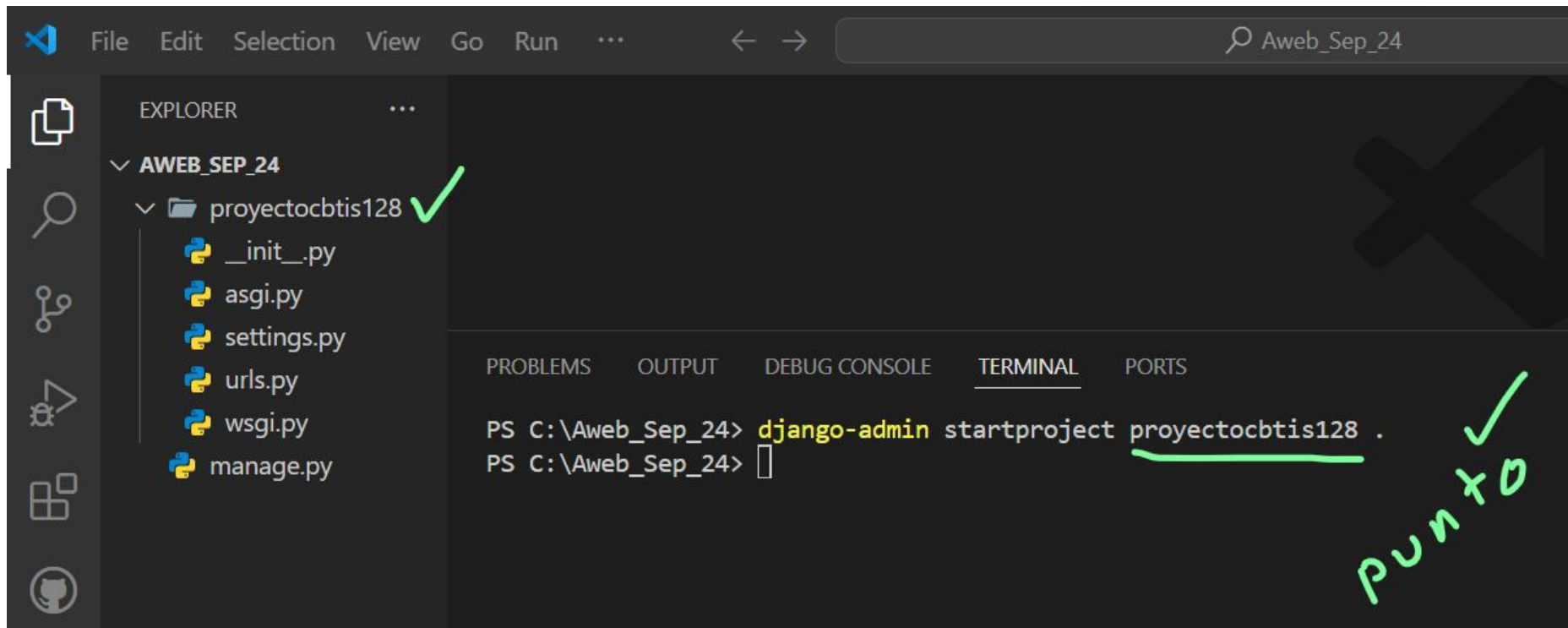
EXPLORER

✓ AWEB\_SEP\_24

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Aweb_Sep_24> python --version ✓
Python 3.12.5
PS C:\Aweb_Sep_24> pip --version ✓
pip 24.2 from C:\Users\Rogel\AppData\Local\Programs\Python\Python...
PS C:\Aweb_Sep_24> django-admin --version ✓
5.1
PS C:\Aweb_Sep_24> 
```

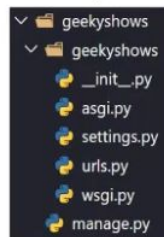
# Requerimientos para Django



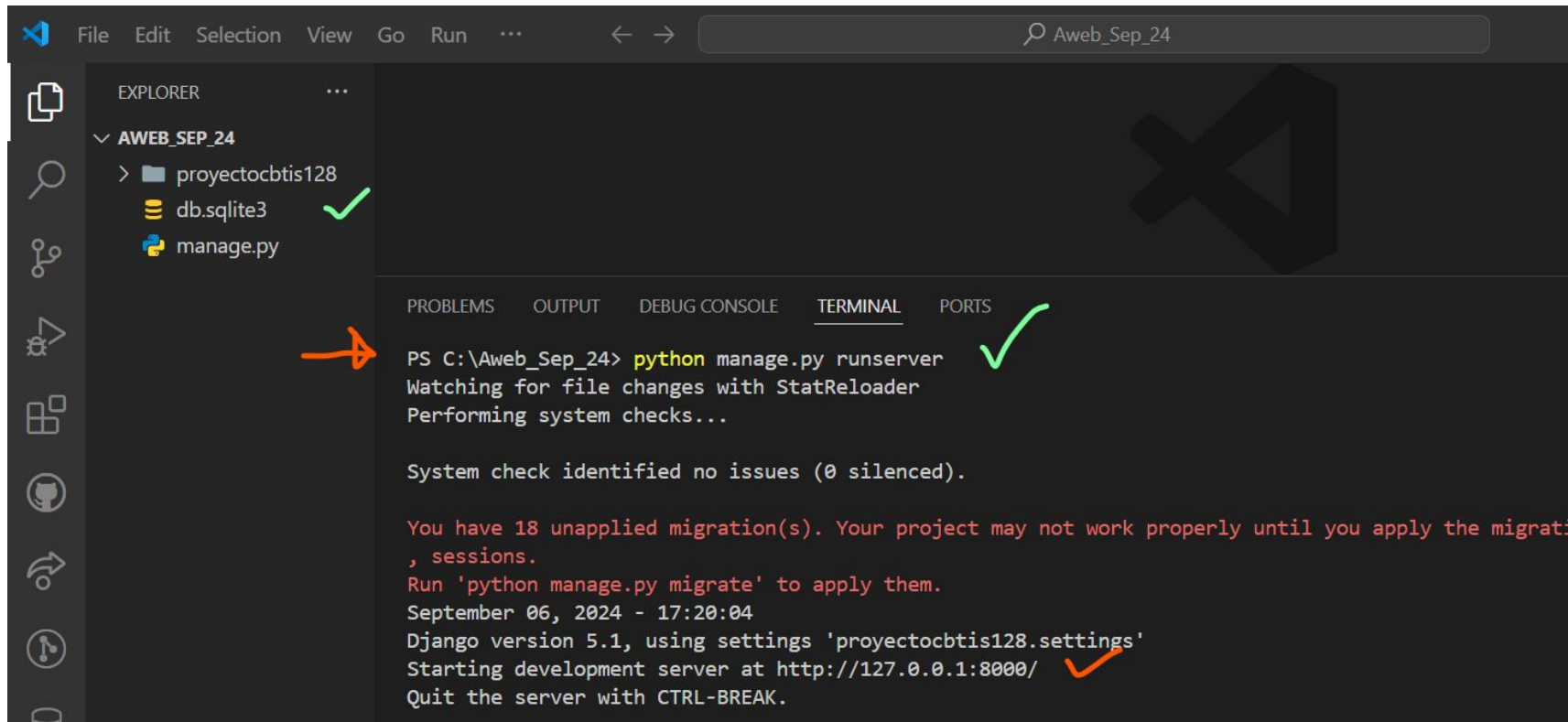
# Estructura del directorio del proyecto

## Django Project Directory Structure

- ✓ **\_\_init\_\_.py** – The folder which contains `__init__.py` file is considered as python package.
- ✓ **wsgi.py** – WSGI (Web Server Gateway Interface) is a specification that describes how a web server communicates with web applications, and how web applications can be chained together to process one request. WSGI provided a standard for synchronous Python apps.
- ✓ **asgi.py** – ASGI (Asynchronous Server Gateway Interface) is a spiritual successor to WSGI, intended to provide a standard interface between async-capable Python web servers, frameworks, and applications. ASGI provides standard for both asynchronous and synchronous apps.
- ✓ **settings.py** – This file contains all the information or data about project settings.  
E.g.:- Database Config information, Template, Installed Application, Validators etc.
- ✓ **urls.py** – This file contains information of url attached with application.
- ✓ **manage.py** – `manage.py` is automatically created in each Django project. It is Django's command-line utility also sets the `DJANGO_SETTINGS_MODULE` environment variable so that it points to your project's `settings.py` file. Generally, when working on a single Django project, it's easier to use `manage.py` than `django-admin`.



# ejecutar servidor



The screenshot shows the Visual Studio Code interface with the following components:

- Explorer Panel:** Displays the project structure under 'AWEB\_SEP\_24'. It includes a folder 'proyectocbtis128' and two files: 'db.sqlite3' (marked with a green checkmark) and 'manage.py' (indicated by an orange arrow).
- Terminal Panel:** Shows the output of the command `python manage.py runserver`. The output includes:
  - Confirmation of file changes being watched by StatReloader.
  - Successful system checks.
  - A warning about 18 unapplied migrations, with a suggestion to run `python manage.py migrate`.
  - Timestamp: September 06, 2024 - 17:20:04.
  - Django version 5.1 and settings path.
  - Server URL: `http://127.0.0.1:8000/` (marked with a green checkmark).
  - Instruction to quit the server with `CTRL-BREAK`.

# en el navegador



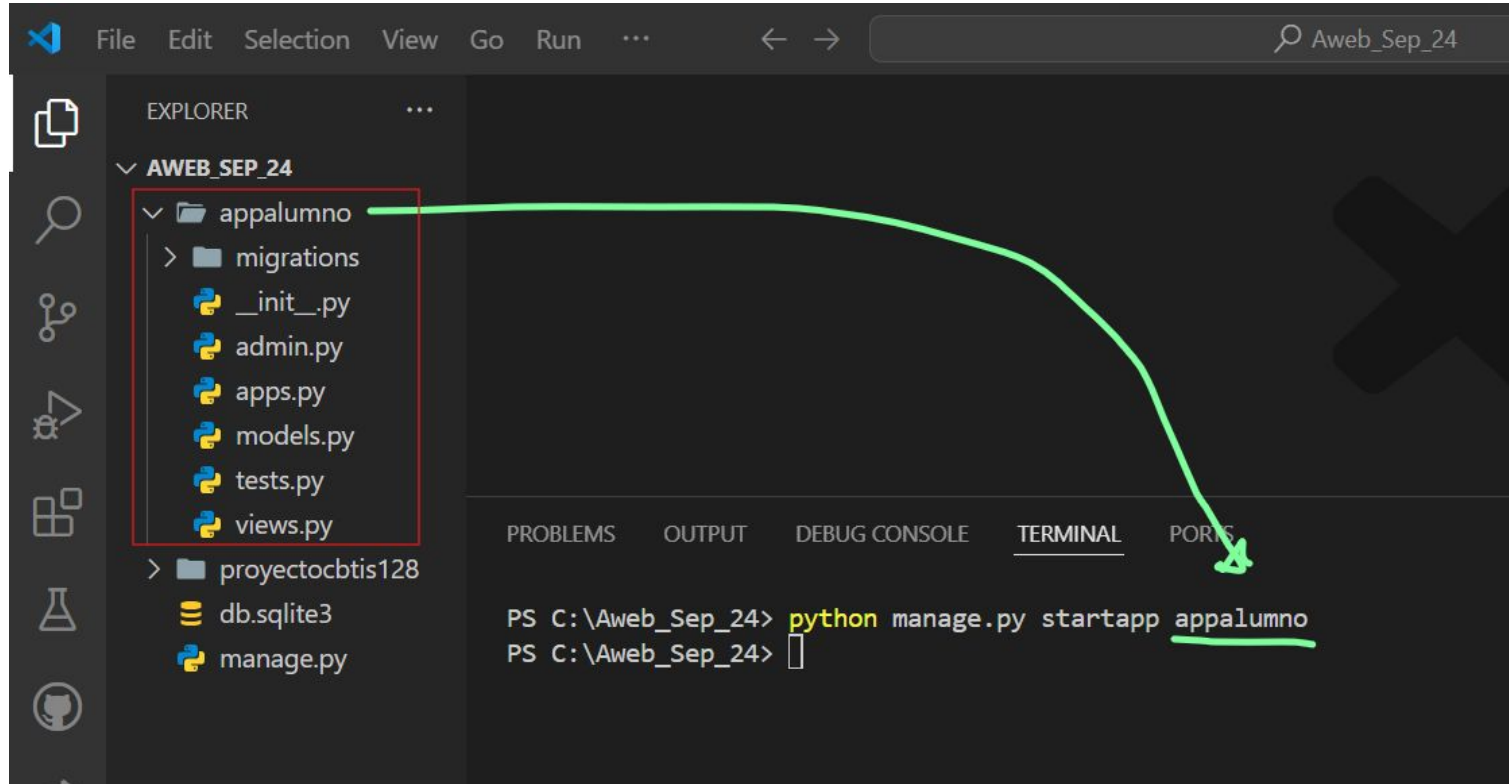
The install worked successfully! Congratulations!

View [release notes](#) for Django 5.1

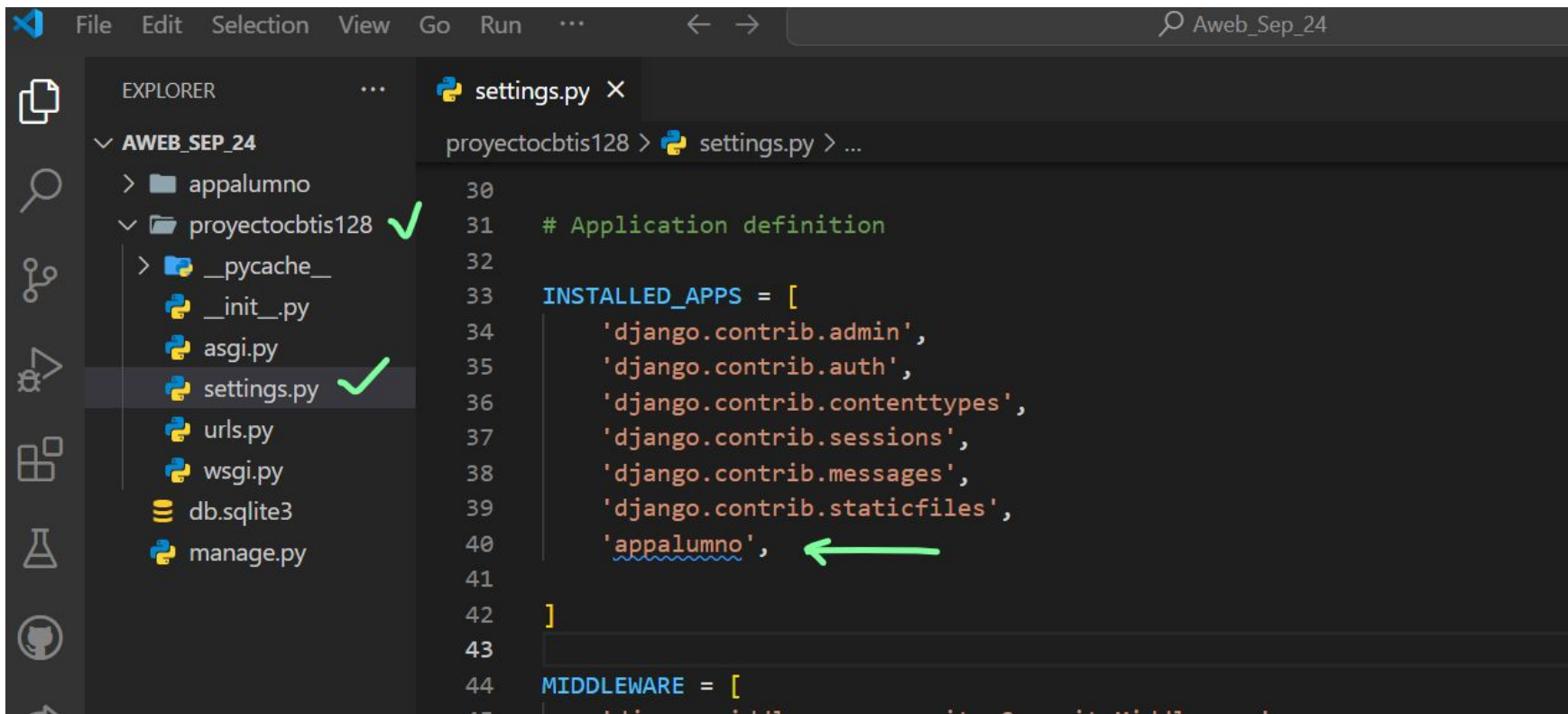
You are seeing this page because `DEBUG=True` is in your settings file and you have not configured any URLs.

**django**

# Creando la aplicación alumno **appalumno**



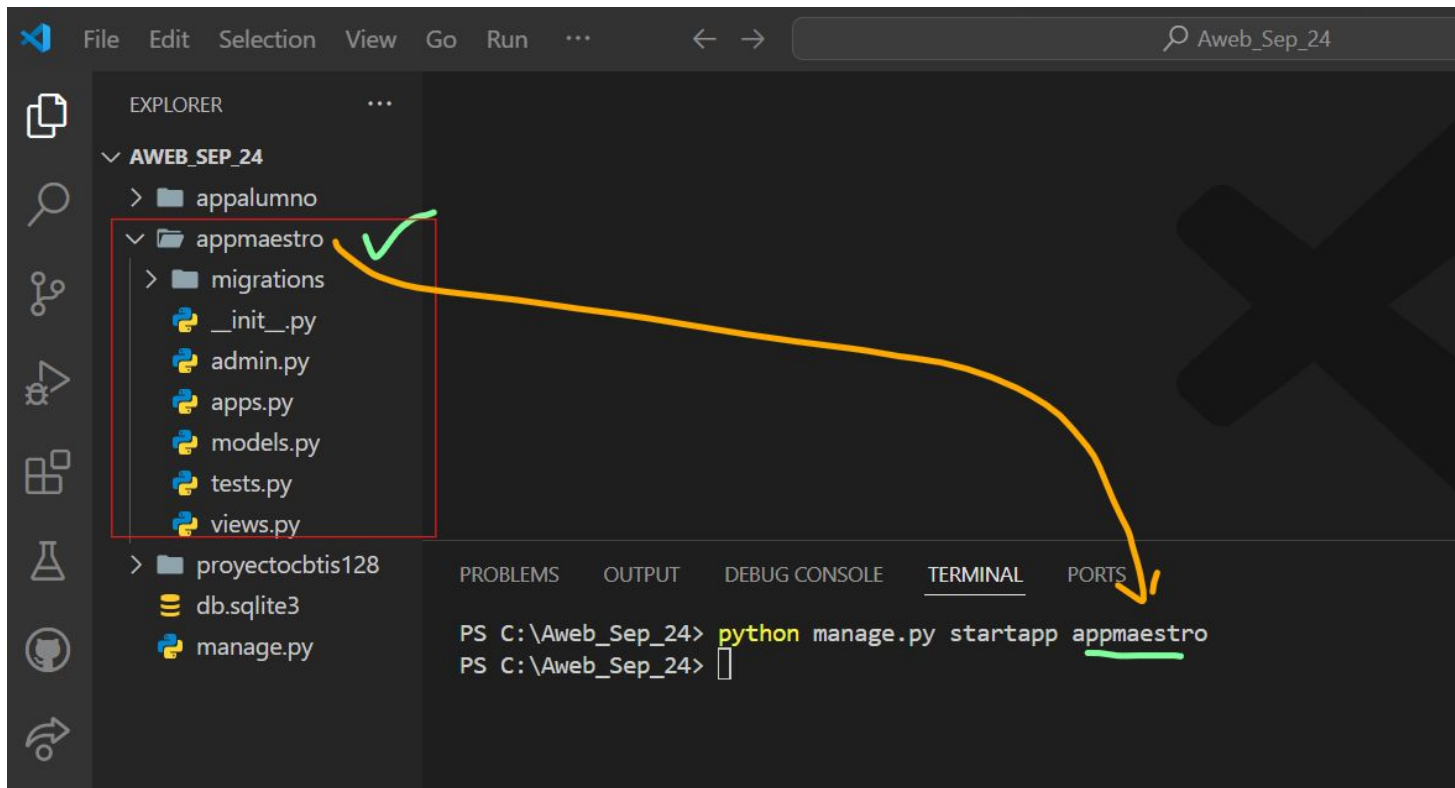
En el archivo settings.py del **proyectocbtis128** agrega la aplicacion **appalumno**,



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays the project structure under 'AWEB\_SEP\_24'. The 'proyectocbtis128' folder is expanded, showing files like '\_\_pycache\_\_', '\_\_init\_\_.py', 'asgi.py', 'settings.py' (highlighted with a green checkmark), 'urls.py', 'wsgi.py', 'db.sqlite3', and 'manage.py'. The main editor window shows the 'settings.py' file. The 'INSTALLED\_APPS' list is defined, and 'appalumno' has been added to the list, indicated by a green arrow pointing to the underlined text on line 40. The code is as follows:

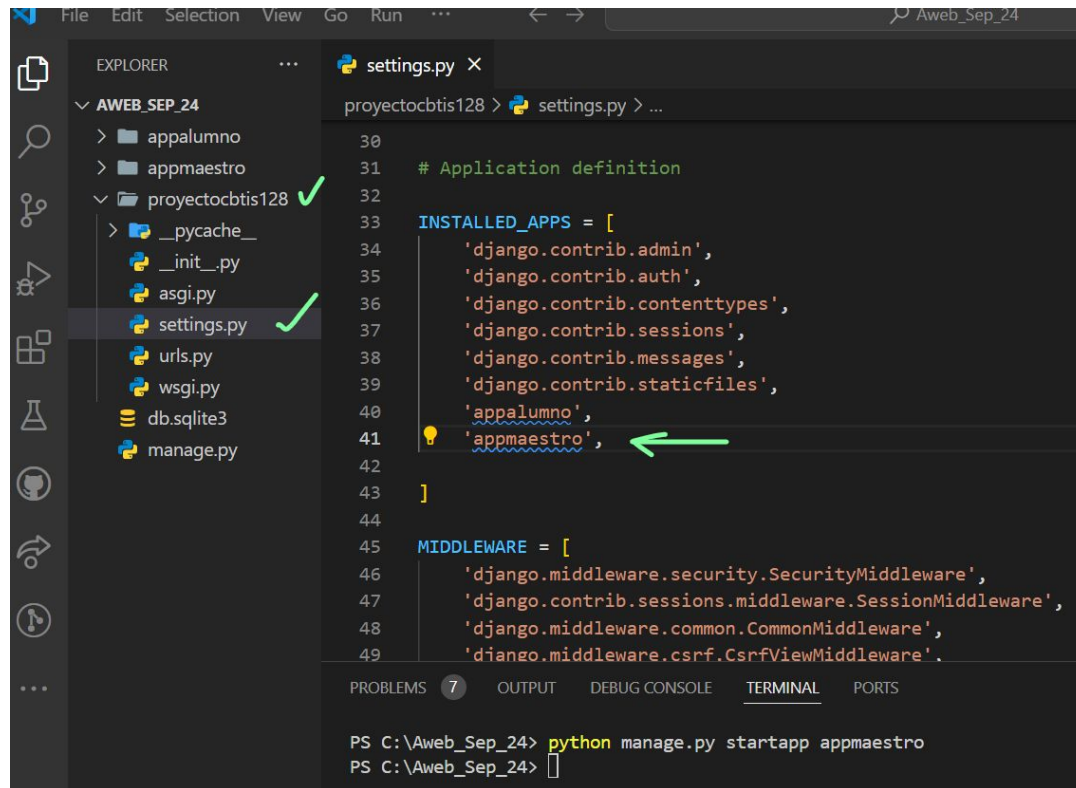
```
30
31 # Application definition
32
33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40     'appalumno',
41
42 ]
43
44 MIDDLEWARE = [
45     'django.middleware.security.SecurityMiddleware',
46     'django.contrib.sessions.middleware.SessionMiddleware',
47     'django.middleware.common.CommonMiddleware',
48     'django.middleware.csrf.CsrfViewMiddleware',
49     'django.contrib.auth.middleware.AuthenticationMiddleware',
50     'django.contrib.messages.middleware.MessageMiddleware',
51     'django.middleware.clickjacking.XFrameOptionsMiddleware',
52 ]
```

En el archivo setting.py del **proyectocbtis128** agrega la aplicacion **appmaestro**,





En el archivo `setting.py` del **proyectocbtis128** agrega la aplicacion **appmaestro**,



The screenshot shows the Visual Studio Code interface with the Explorer view on the left and the Editor view on the right. The Explorer view shows the project structure for 'Aweb\_Sep\_24', with 'proyectocbtis128' expanded. The 'settings.py' file is selected and highlighted with a green checkmark. The Editor view shows the content of 'settings.py' for 'proyectocbtis128'. The file contains the following code:

```
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     'appalumno',
41     'appmaestro',
42 ]
43
44 MIDDLEWARE = [
45     'django.middleware.security.SecurityMiddleware',
46     'django.contrib.sessions.middleware.SessionMiddleware',
47     'django.middleware.common.CommonMiddleware',
48     'django.middleware.csrf.CsrfViewMiddleware',
49 ]
```

A green arrow points to the 'appmaestro' entry in the 'INSTALLED\_APPS' list. The bottom of the interface shows the Terminal view with the following commands:

```
PS C:\Aweb_Sep_24> python manage.py startapp appmaestro
PS C:\Aweb_Sep_24>
```

# estructura de la appmaestro

**migrations** – This folder contains `__init__.py` file which means it's a python package. It also contains all files which are created after running `makemigration` command .

**`__init__.py`** – The folder which contains `__init__.py` file is considered as Python Package.

**admin.py** – This file is used to register sql tables so we could perform CRUD operation from Admin Application. Admin Application is provided by Django to perform CRUD operation.

**apps.py** – This file is used to config app.

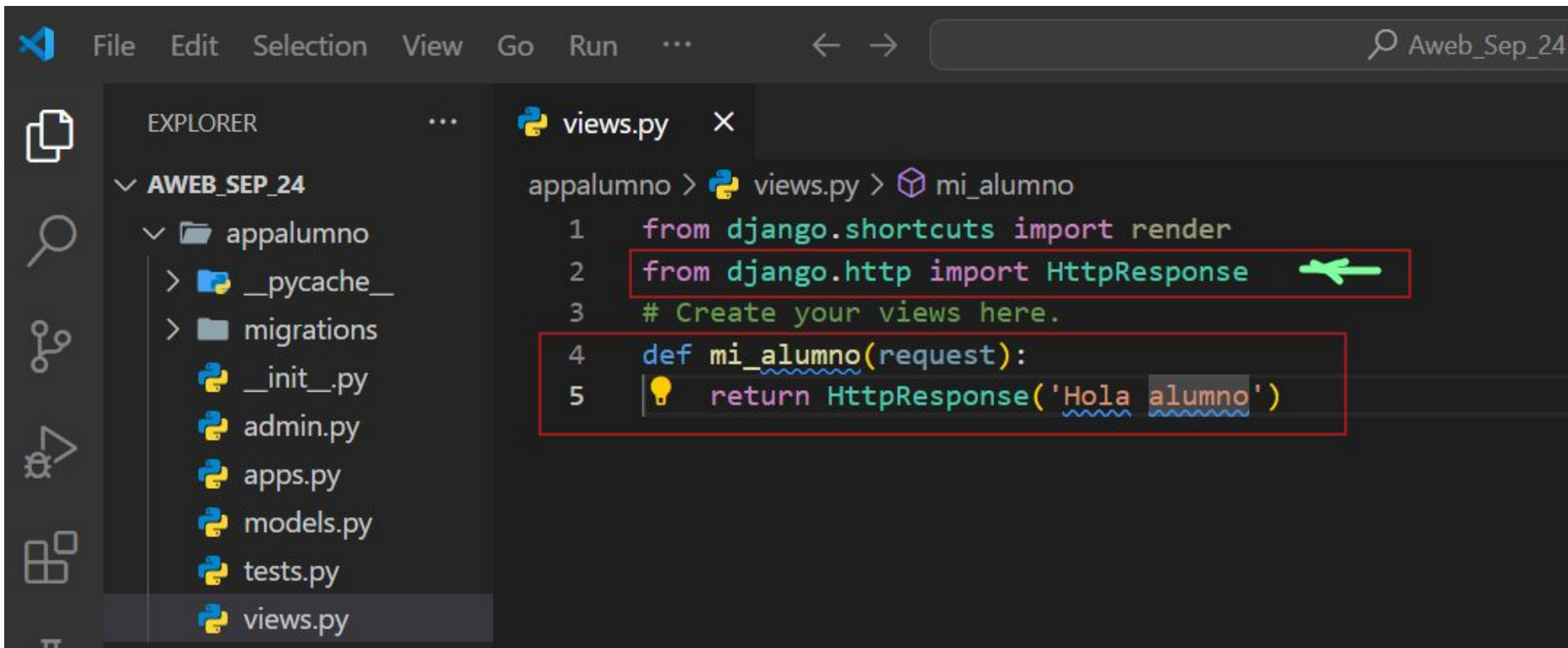
**models.py** – This file is used to create our own model class later these classes will be converted into database table by Django for our application.

**tests.py** – This is files is used to create tests.



**views.py** – This file is used to create view. We write all the business logic related code in this file.

En el archivo **views.py** de la carpeta **appalumno** escribe lo que se indica en rojo



The screenshot shows a code editor with the following structure:

- EXPLORER
  - AWEB\_SEP\_24
    - appalumno
      - \_\_pycache\_\_
      - migrations
      - \_\_init\_\_.py
      - admin.py
      - apps.py
      - models.py
      - tests.py
      - views.py

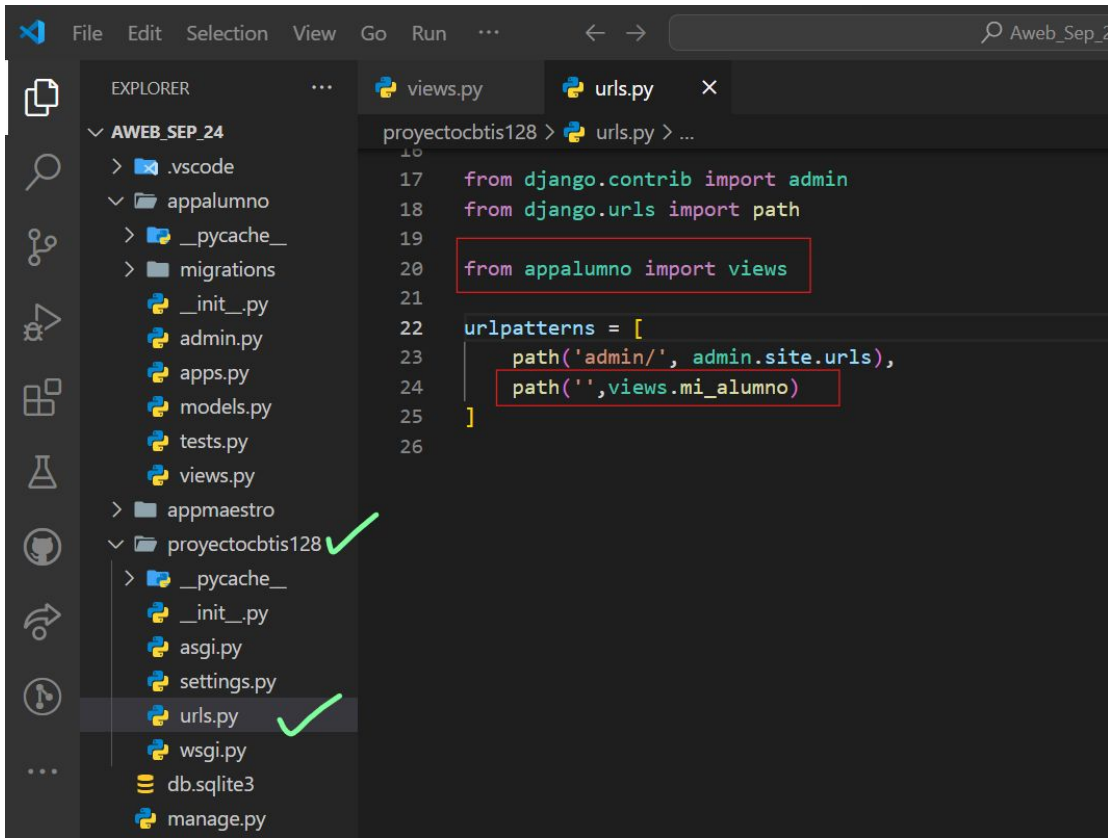
The **views.py** file is open, showing the following code:

```
appalumno > views.py > mi_alumno
1 from django.shortcuts import render
2 from django.http import HttpResponseRedirect
3 # Create your views here.
4 def mi_alumno(request):
5     return HttpResponseRedirect('Hola alumno')
```

Red boxes highlight the following code segments:

- Line 2: `from django.http import HttpResponseRedirect`
- Lines 4-5: `def mi_alumno(request):` and `return HttpResponseRedirect('Hola alumno')`

En el archivo **urls.py** de la carpeta **proyectocbtis128** escribe lo que se indica en rojo



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays the project structure. The 'proyectocbtis128' folder is expanded, and the 'urls.py' file is selected, indicated by a green checkmark. The main editor area shows the content of 'urls.py'. Two lines of code are highlighted with red boxes: 'from appalumno import views' on line 20 and 'path('', views.mi\_alumno)' on line 24. The code in the editor is as follows:

```
16
17 from django.contrib import admin
18 from django.urls import path
19
20 from appalumno import views
21
22 urlpatterns = [
23     path('admin/', admin.site.urls),
24     path('', views.mi_alumno)
25 ]
26
```