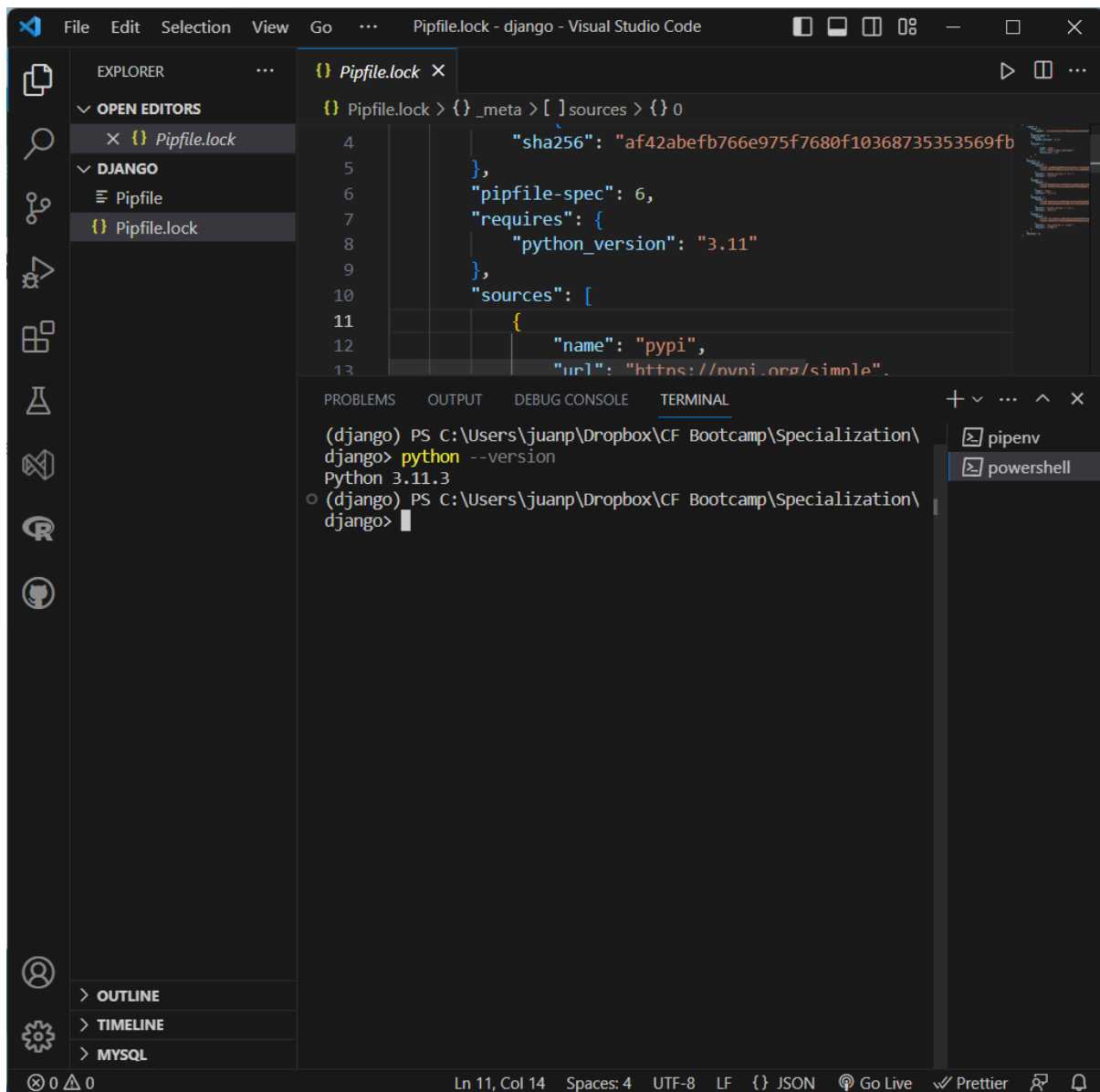


1. It is scalable, which means it is ideal for large websites. It executes the controller for you. There is a large programmer community behind it.
2. Scenarios:
 - a. Django -> The application is large and will probably need to be scaled at some point.
 - b. Django -> Since the framework is quick to generate and implement models and their functionality, you can deploy and maintain it quickly.
 - c. No Django -> It is unnecessary if there's no access to databases.
 - d. No Django -> If you want to maximize control over the app, you wouldn't use a pre-made tool like Django.
 - e. tool like Django.
 - f. Django -> Django has a large community of developers behind it. You can certainly find the help you need.
3. Python version and pipenv:

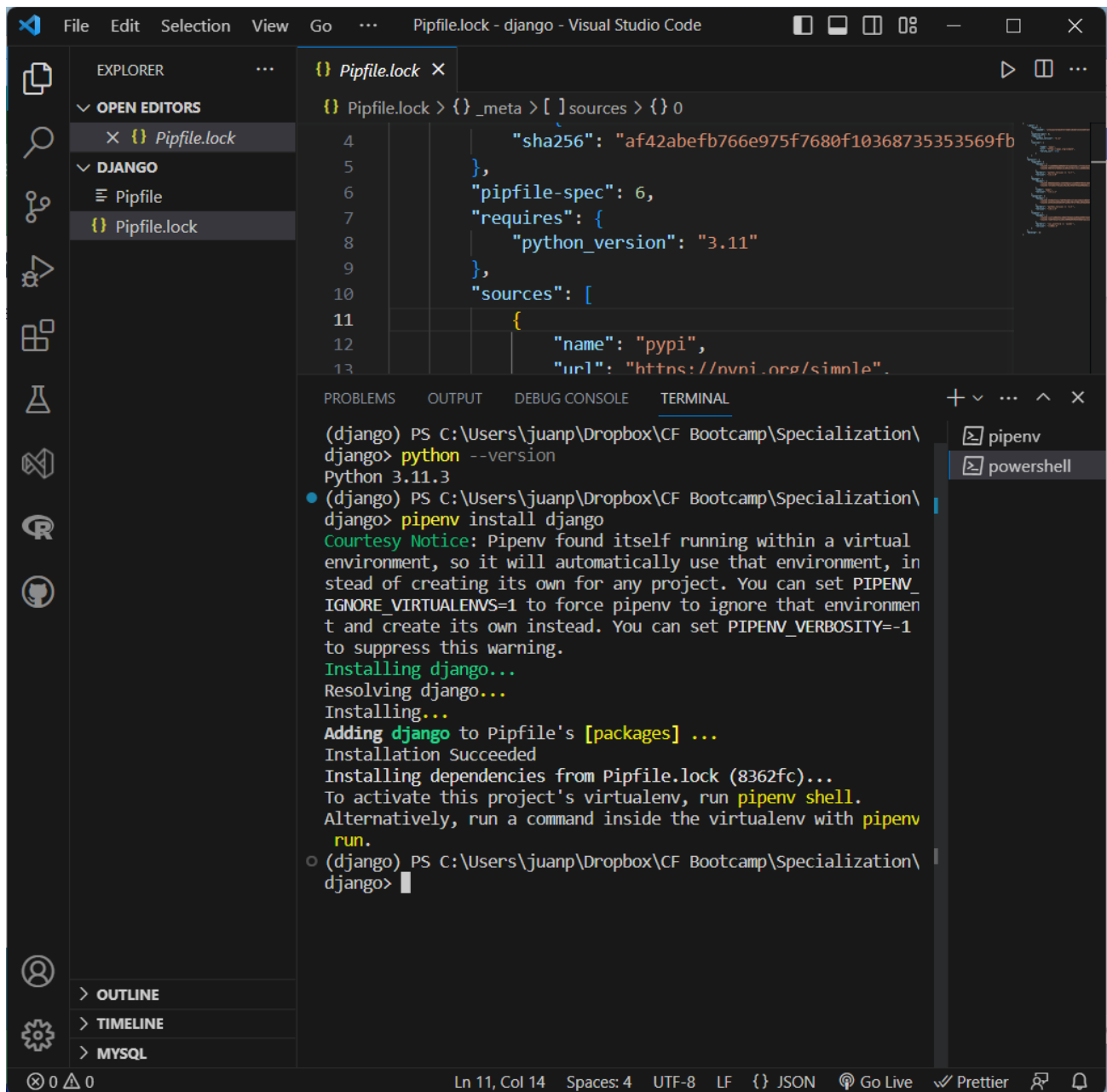


The screenshot shows the Visual Studio Code interface. The Explorer panel on the left shows the file structure with 'Pipfile.lock' selected. The main editor area displays the contents of 'Pipfile.lock', which is a JSON file. The terminal panel at the bottom shows the command prompt for a Django project, where the command 'python --version' has been executed, resulting in 'Python 3.11.3'. The status bar at the bottom indicates the current file is 'Pipfile.lock' at line 11, column 14, with 4 spaces, UTF-8 encoding, and LF line endings.

```
{
  "_meta": {
    "sources": [
      {
        "name": "pypi",
        "url": "https://pypi.org/simple"
      }
    ]
  },
  "requires": {
    "python_version": "3.11"
  },
  "sha256": "af42abefb766e975f7680f10368735353569fb"
}
```

```
(django) PS C:\Users\juanp\Dropbox\CF Bootcamp\Specialization\
django> python --version
Python 3.11.3
(django) PS C:\Users\juanp\Dropbox\CF Bootcamp\Specialization\
django>
```

4. Django installed:



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

- EXPLORER:** Shows the project structure with `Pipfile` and `Pipfile.lock` files.
- EDITOR:** Displays the `Pipfile.lock` file with the following content:

```
{
  "meta": {
    "sha256": "af42abefb766e975f7680f10368735353569fb",
    "pipfile-spec": 6,
    "requires": {
      "python_version": "3.11"
    },
    "sources": [
      {
        "name": "pypi",
        "url": "https://pypi.org/simple"
      }
    ]
  }
}
```
- TERMINAL:** Shows the output of the following commands:

```
(django) PS C:\Users\juanp\Dropbox\CF Bootcamp\Specialization\
django> python --version
Python 3.11.3
(django) PS C:\Users\juanp\Dropbox\CF Bootcamp\Specialization\
django> pipenv install django
Courtesy Notice: Pipenv found itself running within a virtual
environment, so it will automatically use that environment, in
stead of creating its own for any project. You can set PIPENV_
IGNORE_VIRTUALENVS=1 to force pipenv to ignore that environmen
t and create its own instead. You can set PIPENV_VERBOSITY=-1
to suppress this warning.
Installing django...
Resolving django...
Installing...
Adding django to Pipfile's [packages] ...
Installation Succeeded
Installing dependencies from Pipfile.lock (8362fc)...
To activate this project's virtualenv, run pipenv shell.
Alternatively, run a command inside the virtualenv with pipenv
run.
(django) PS C:\Users\juanp\Dropbox\CF Bootcamp\Specialization\
django>
```