The courtesy notice indicates that pipenv is detecting another virtual environment. To ensure that pipenv creates and uses its own virtual environment, follow these steps:

**Steps to Ensure pipenv Uses Its Own Environment**

1. **Deactivate Any Existing Virtual Environments**: Ensure that no other virtual environments are active:

sh

Copy code

deactivate

1. **Set the PIPENV\_IGNORE\_VIRTUALENVS Environment Variable**: You can set this environment variable to force pipenv to ignore any existing virtual environments and create its own.

**For Windows Command Prompt:**

sh

Copy code

set PIPENV\_IGNORE\_VIRTUALENVS=1

**For Windows PowerShell:**

sh

Copy code

$env:PIPENV\_IGNORE\_VIRTUALENVS=1

1. **Navigate to Your Project Directory**: Make sure you are in the correct project directory where your Pipfile is located.

sh

Copy code

cd C:\Users\akash\Desktop\todopy

1. **Initialize or Activate the pipenv Shell**: Initialize or activate the pipenv shell:

sh

Copy code

pipenv shell

1. **Install Dependencies (if not already installed)**: If you haven't already installed the necessary packages, do so within the pipenv environment:

sh

Copy code

pipenv install flask flask-marshmallow

1. **Run Your Script**: After ensuring the pipenv shell is active, run your script:

sh

Copy code

python main.py

**Example of Running Through PowerShell**

Here's an example sequence of commands:

sh

Copy code

# Deactivate any existing virtual environment

deactivate

# Set PIPENV\_IGNORE\_VIRTUALENVS in PowerShell

$env:PIPENV\_IGNORE\_VIRTUALENVS=1

# Navigate to your project directory

cd C:\Users\akash\Desktop\todopy

# Activate the pipenv shell

pipenv shell

# Install dependencies if needed

pipenv install flask flask-marshmallow

# Run your script

python main.py

**Persisting the Environment Variable**

To avoid setting PIPENV\_IGNORE\_VIRTUALENVS=1 every time, you can add it to your environment variables permanently:

1. **Open Environment Variables**:
   * Right-click on the Start button and select "System".
   * Click on "Advanced system settings".
   * In the System Properties window, click on the "Environment Variables" button.
2. **Add a New Environment Variable**:
   * In the Environment Variables window, click "New" under the "User variables" or "System variables" section.
   * Set the variable name to PIPENV\_IGNORE\_VIRTUALENVS and the value to 1.
   * Click "OK" to close all windows.

By following these steps, you can ensure that pipenv creates and uses its own virtual environment, allowing you to run your Python scripts without encountering module errors.