Topic: 4. Setting API EndPoints

Speaker: Personal | Notebook: API Development using Django Framework

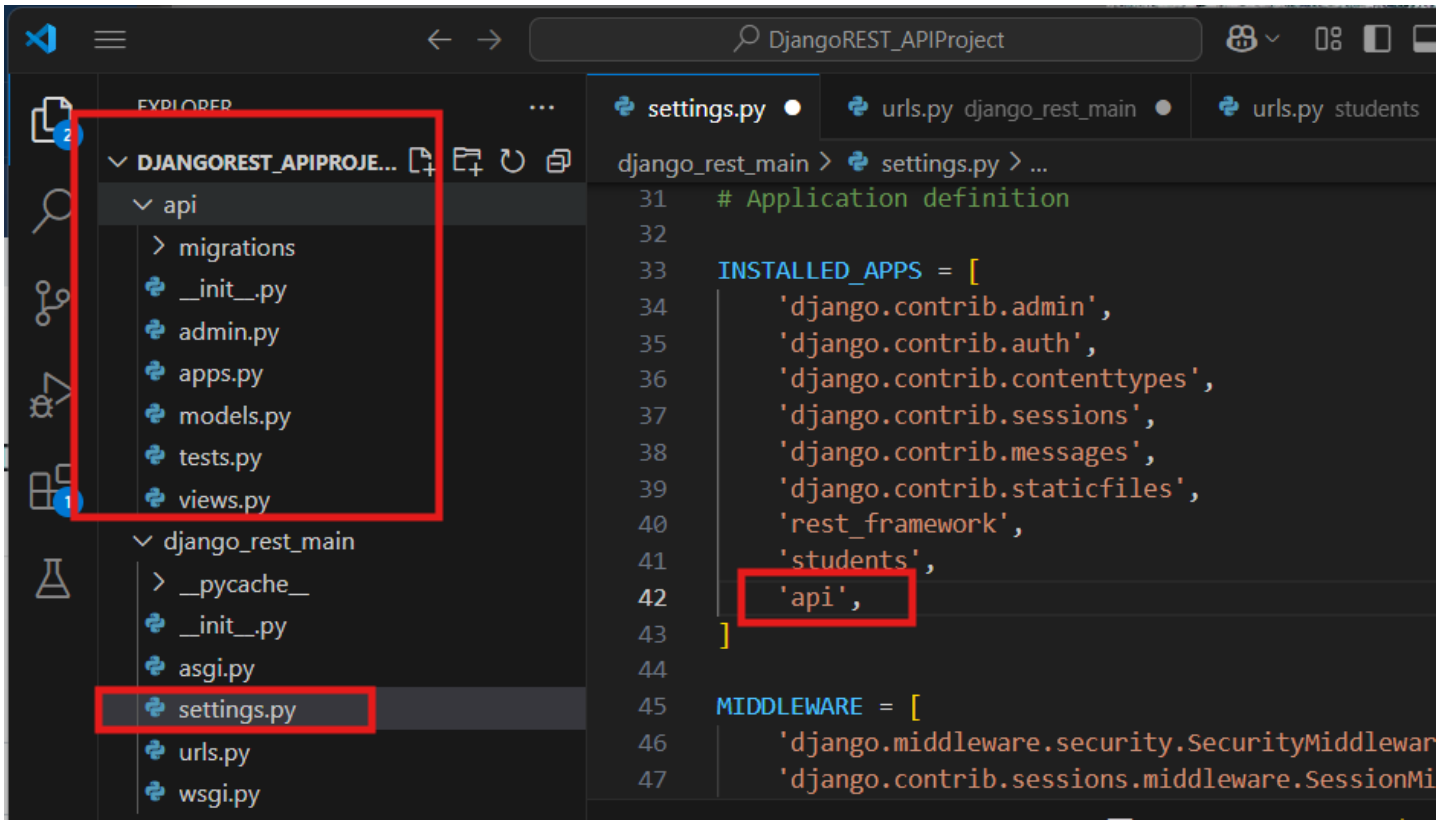
---

For other resources on how to create simple API endpoints using Django, use this reference in Medium.

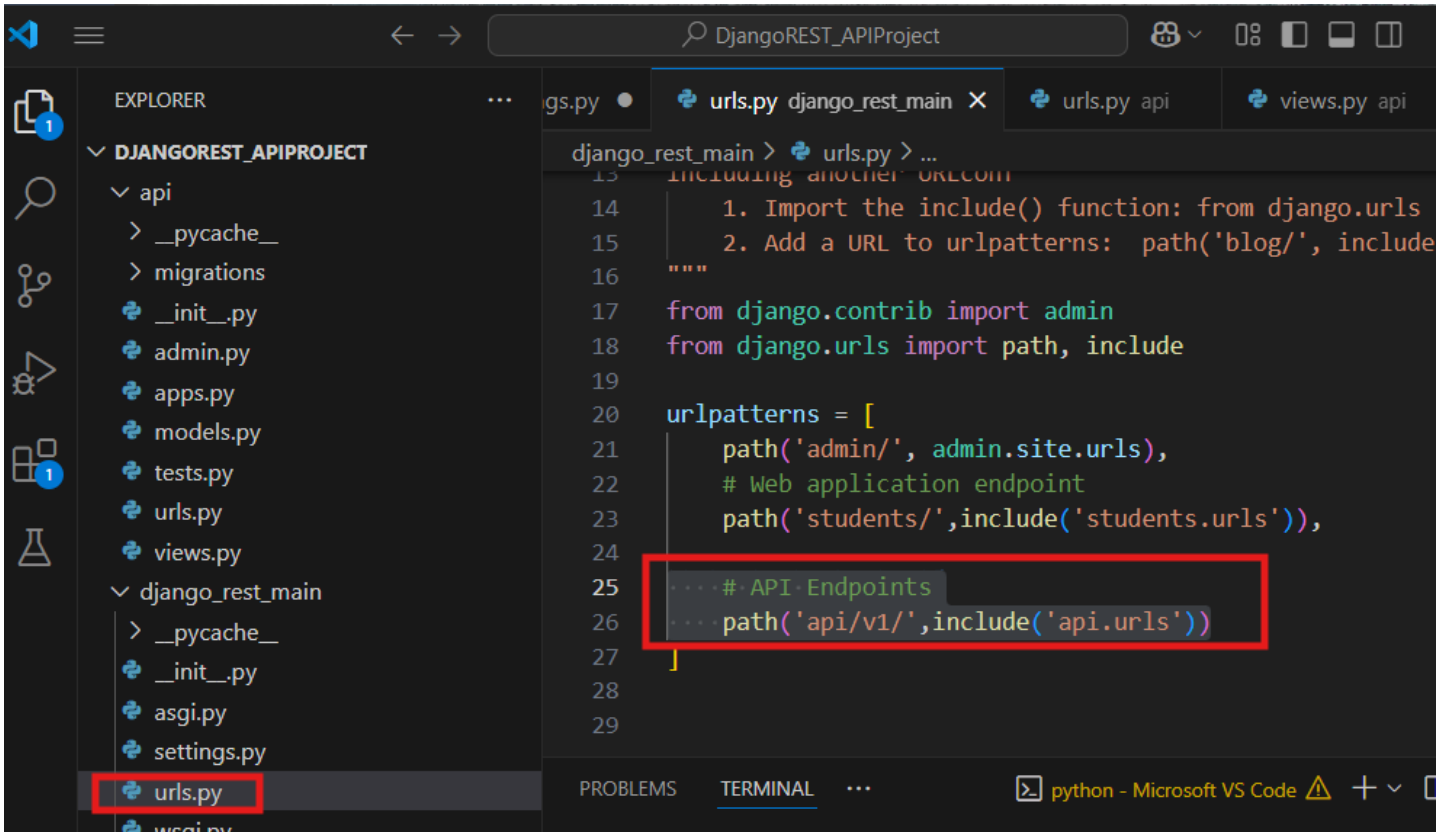
1. Create an API app in your Django project:

```
$ python manage.py startapp api
```

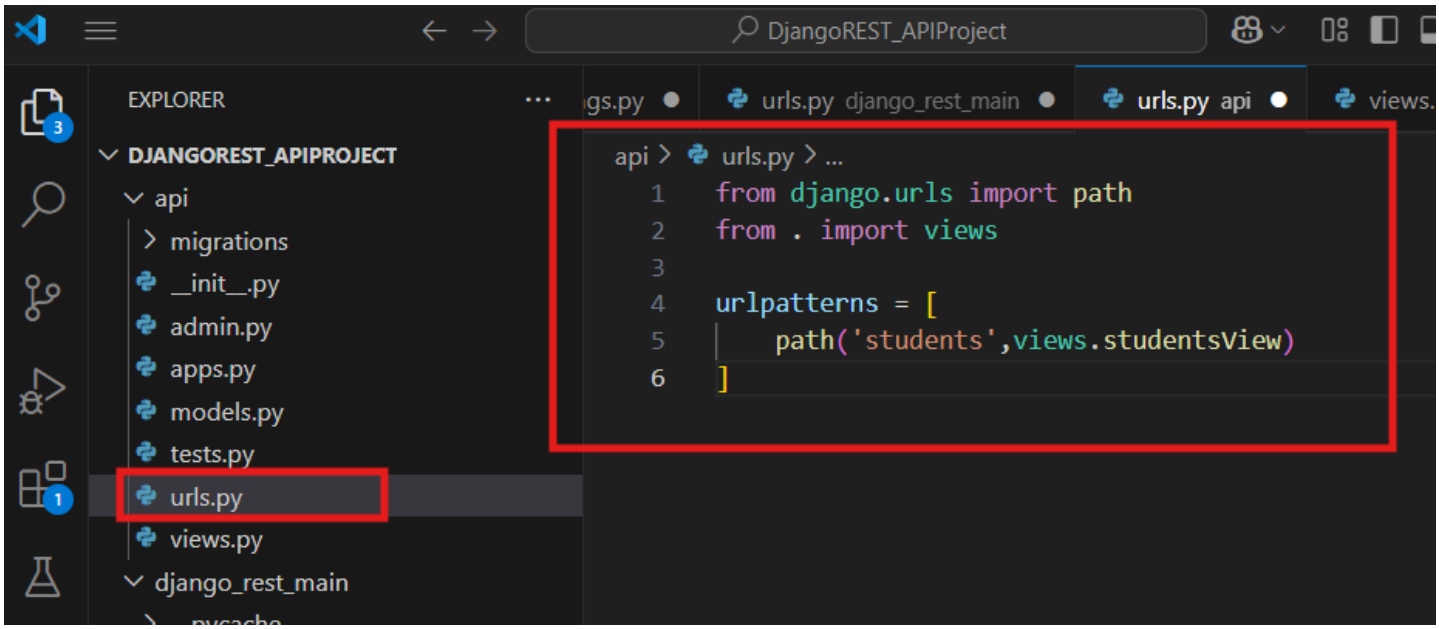
2. Register this new app in your SETTINGS.PY



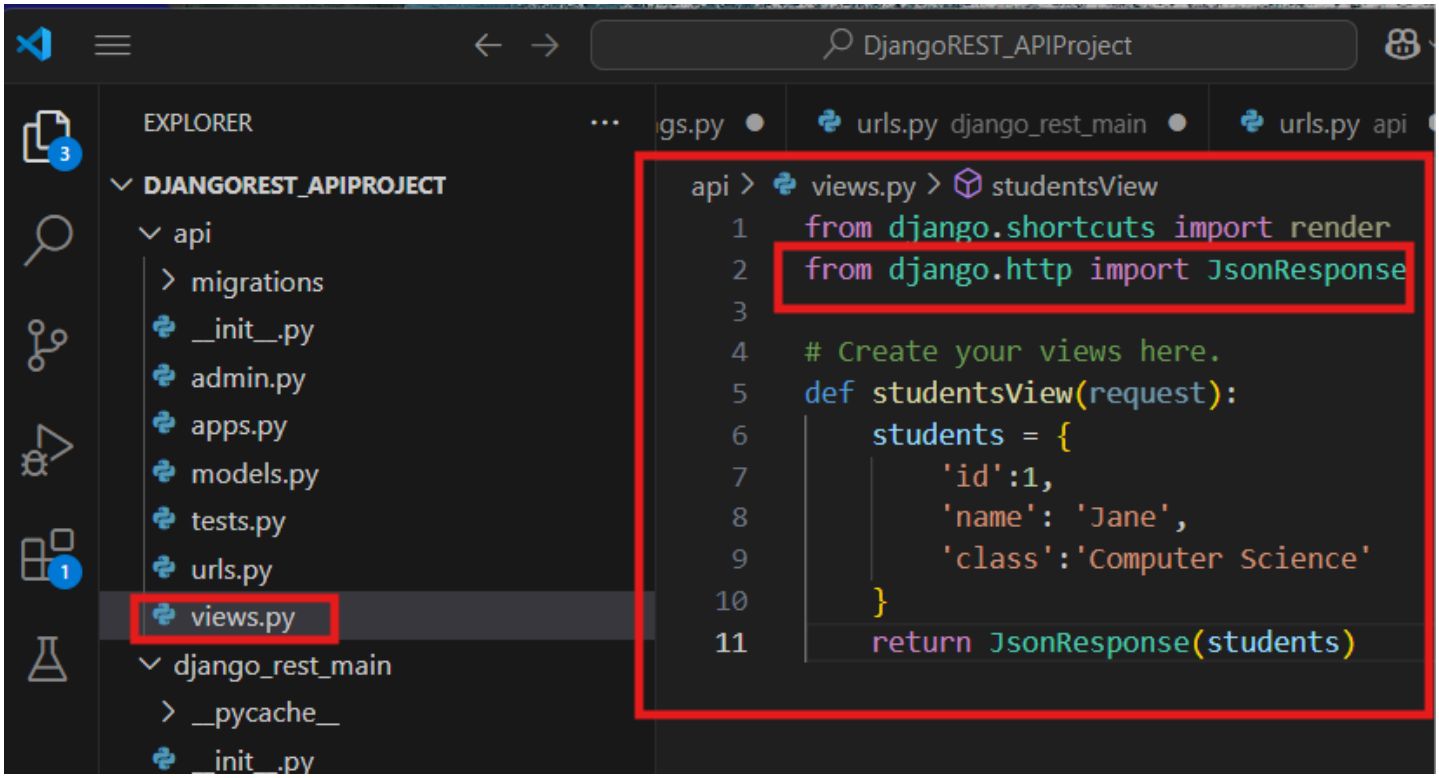
3. Now, update the main project's URLS.PY to include the URLS.PY of the newly created app.



4. We then create a URLS.py in our API app and create the path.



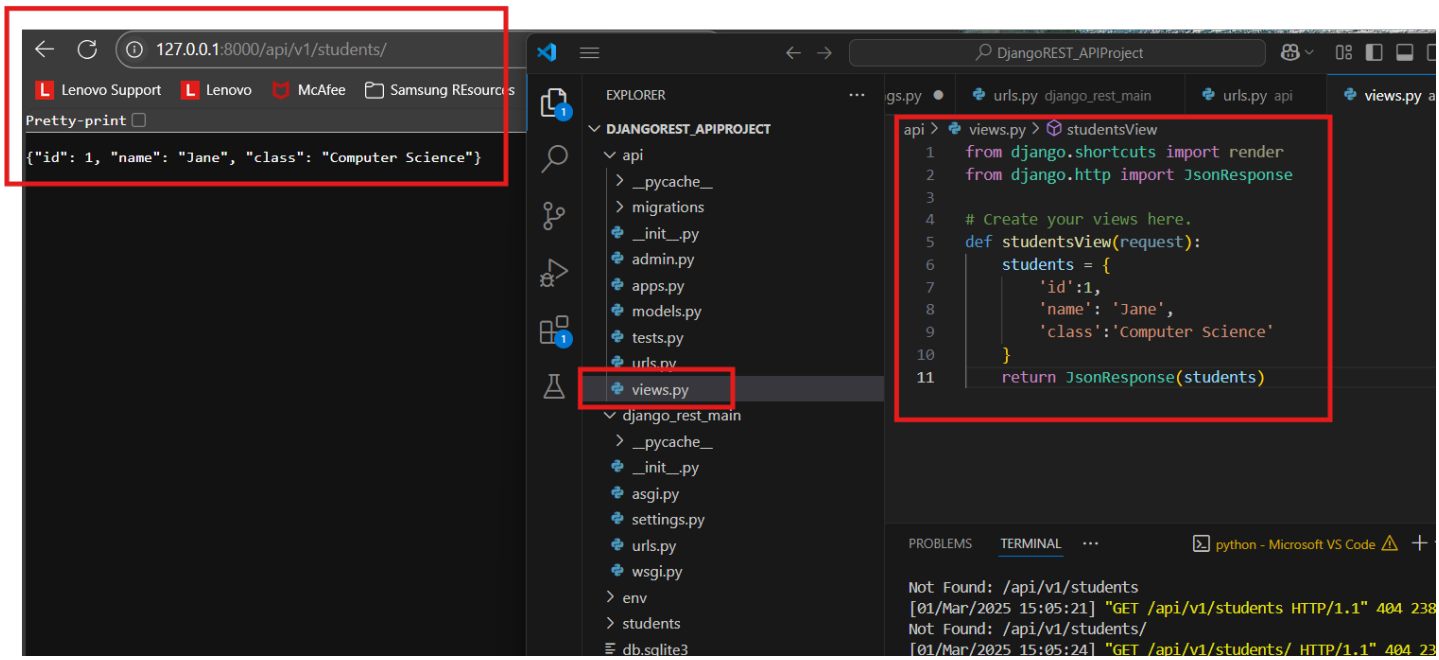
5. Update the API APP'S VIEWS.PY . Unlike regular views of web-based endpoints, APIs return JSON data.



6. Run the server and add the API endpoint:

```
$ python manage.py runserver
```

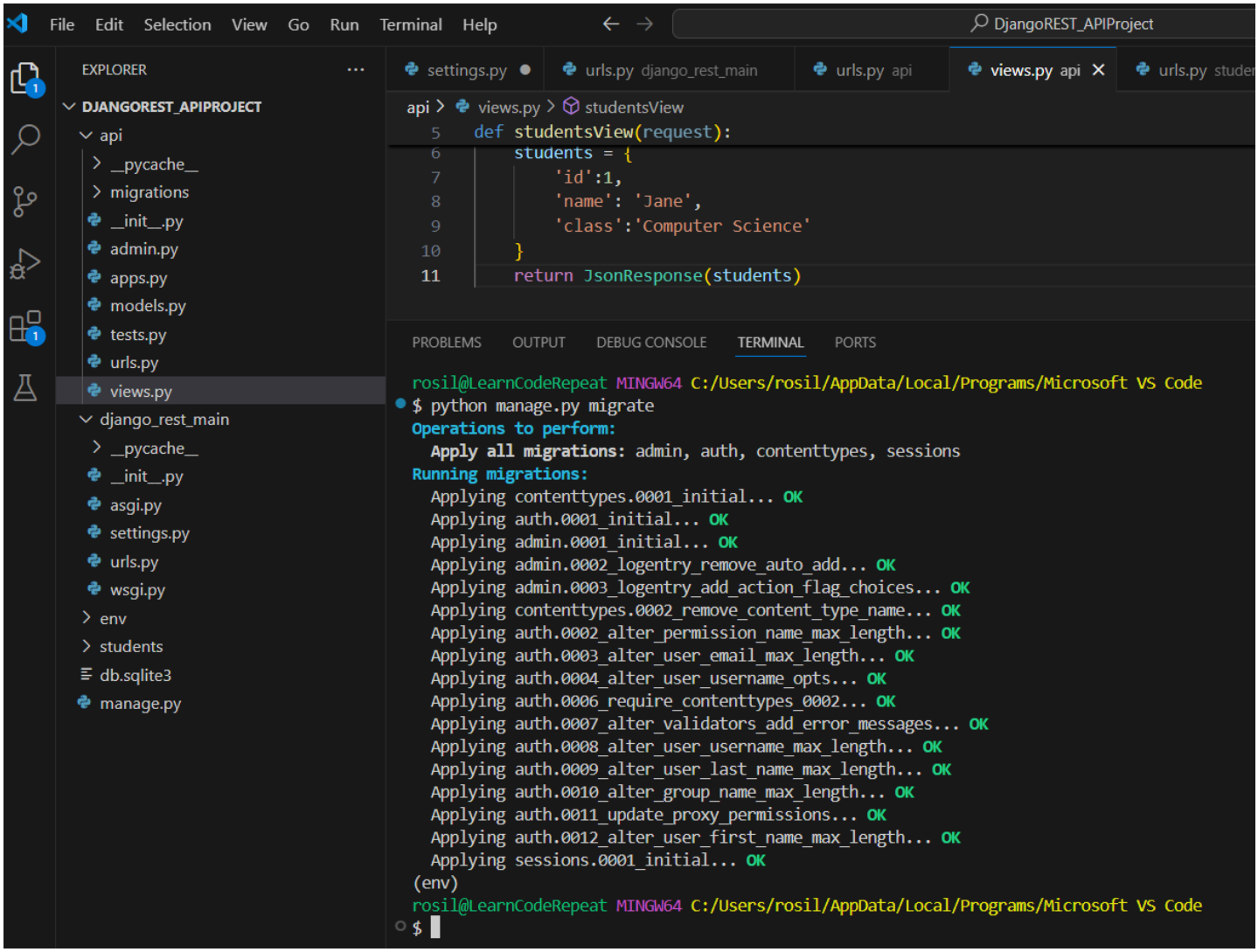
In your browser: 127.0.0.1:8000/api/v1/students/



7. Run the migrations command and create superuser.

```
$ python manage.py migrate
```

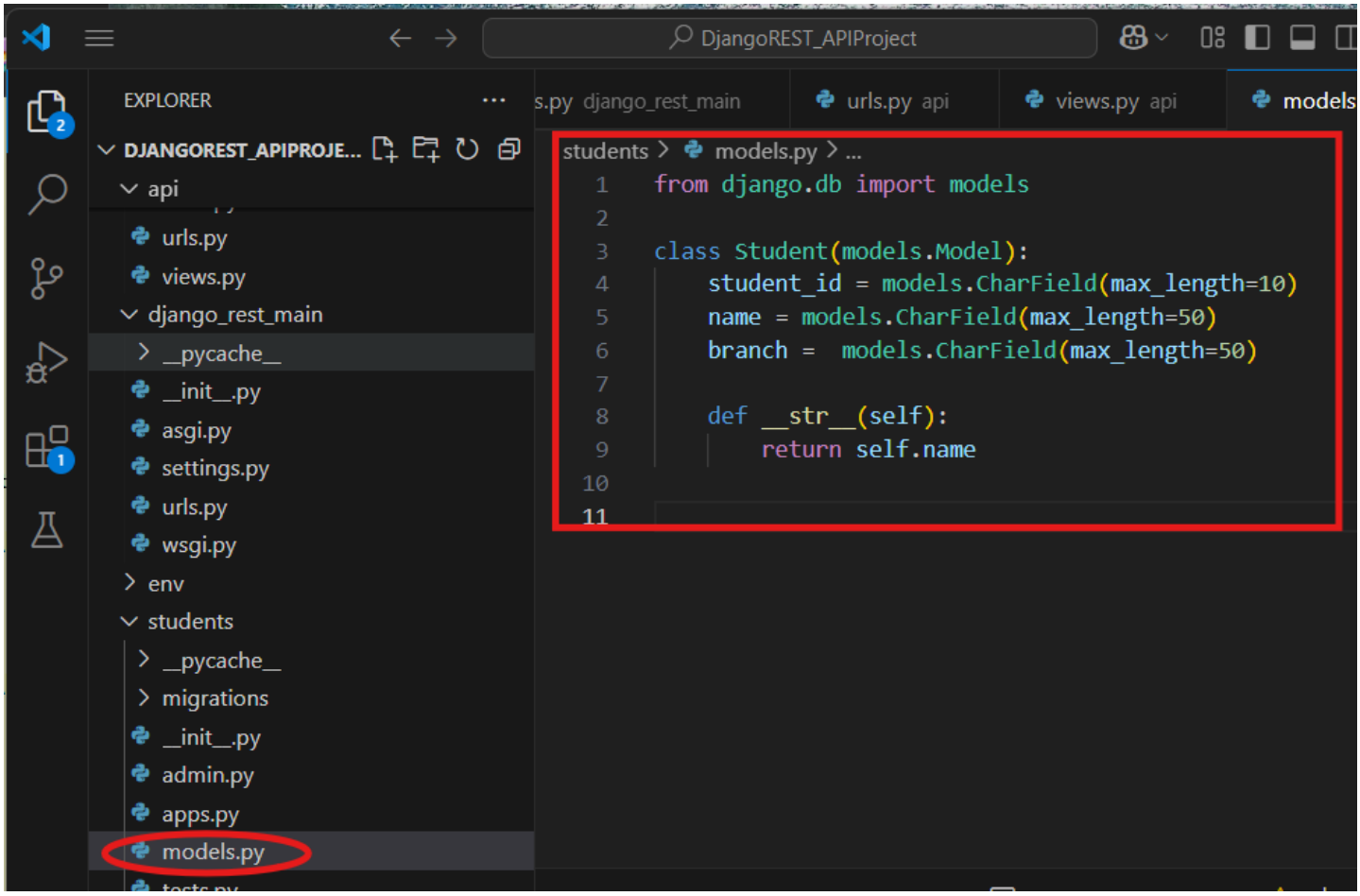
This will create `user.auths` default table.



8. Access the admin panel by running the server. Access it using:

`http://127.0.0.1:8000/admin/`

9. Create a student model in the STUDENTS app to create a new table.



10. Run the migrations.

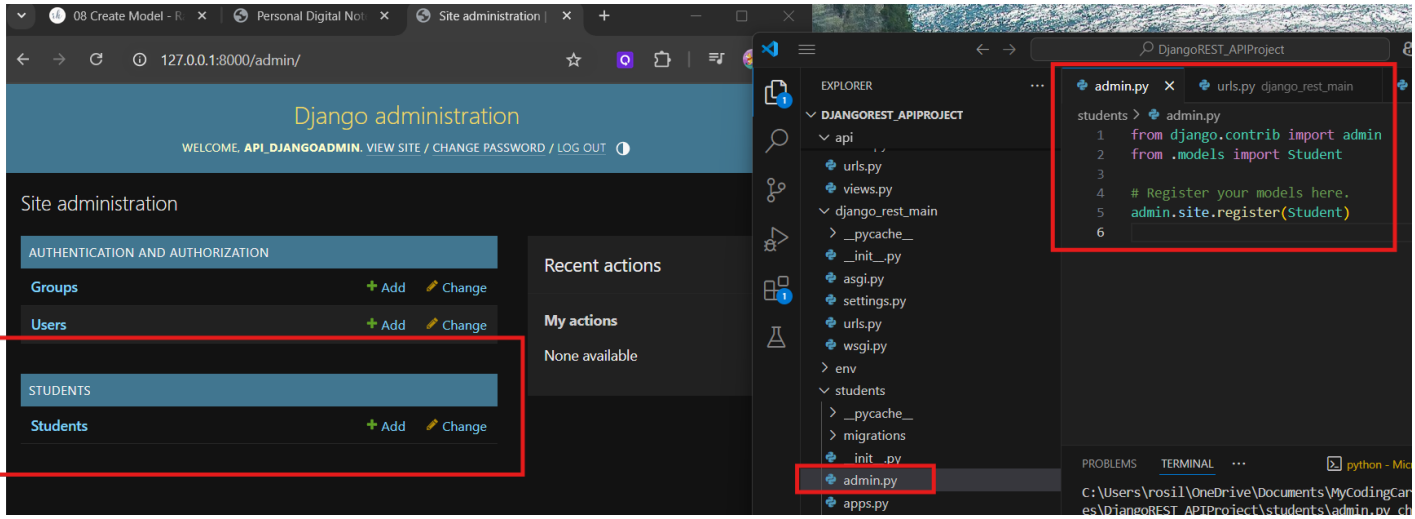
```
$ python manage.py makemigrations
```

```
$ python manage.py migrate
```

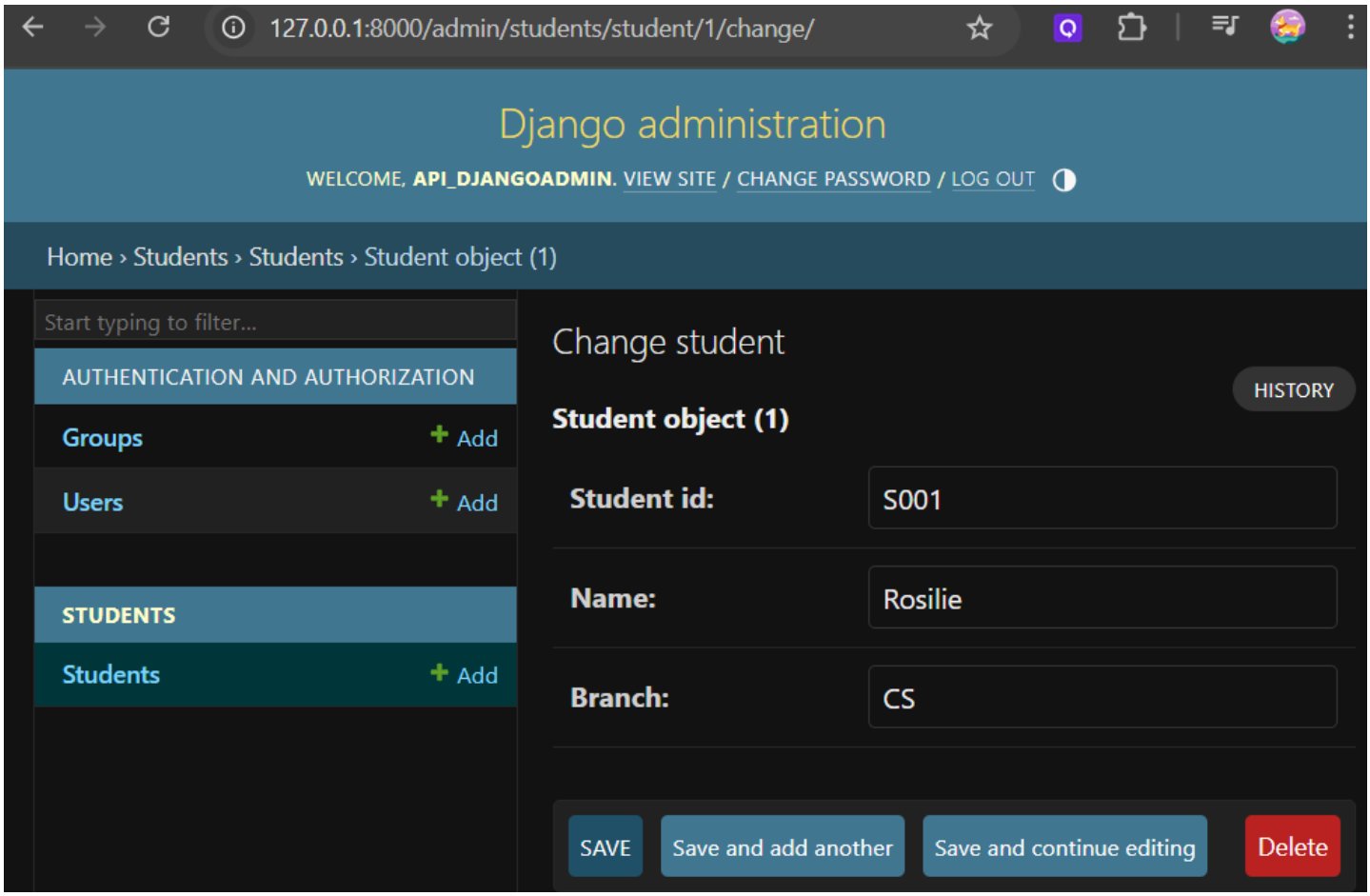
```
$ python manage.py makemigrations
Migrations for 'students':
  students\migrations\0001_initial.py
    + Create model Student
(env)
```

```
(env)
rosil@LearnCodeRepeat MINGW64 C:/Users/rosil/AppData/Local/Programs/Mi
Code
$ python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions, students
Running migrations:
  Applying students.0001_initial... OK
(env)
```

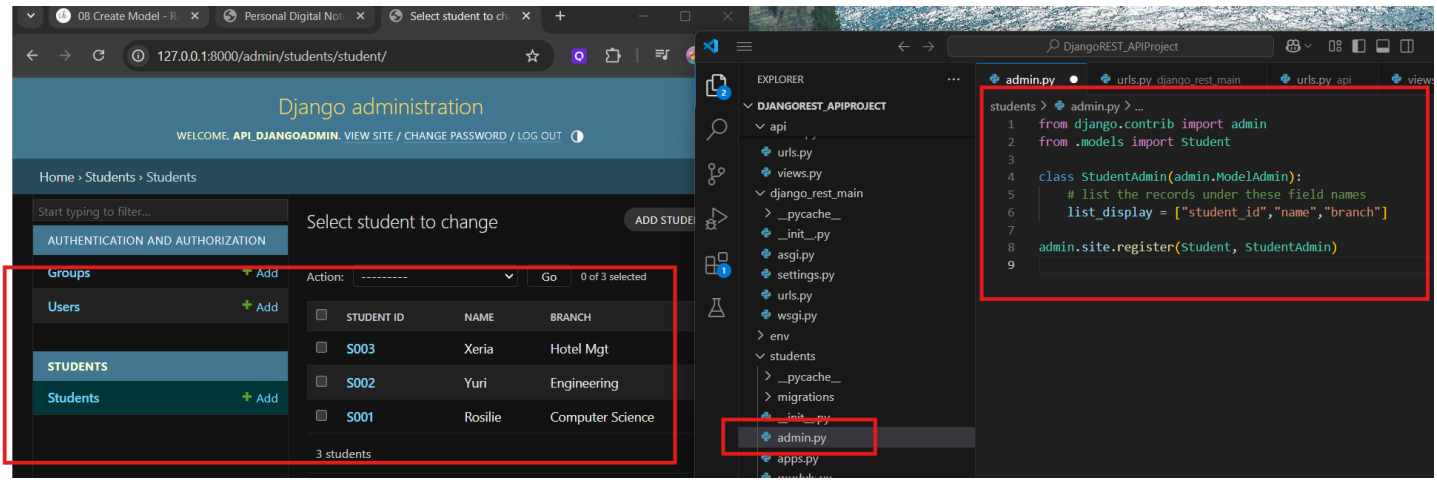
11. To make this table appear on your admin dashboard, register this in the ADMIN.PY of STUDENTS app.



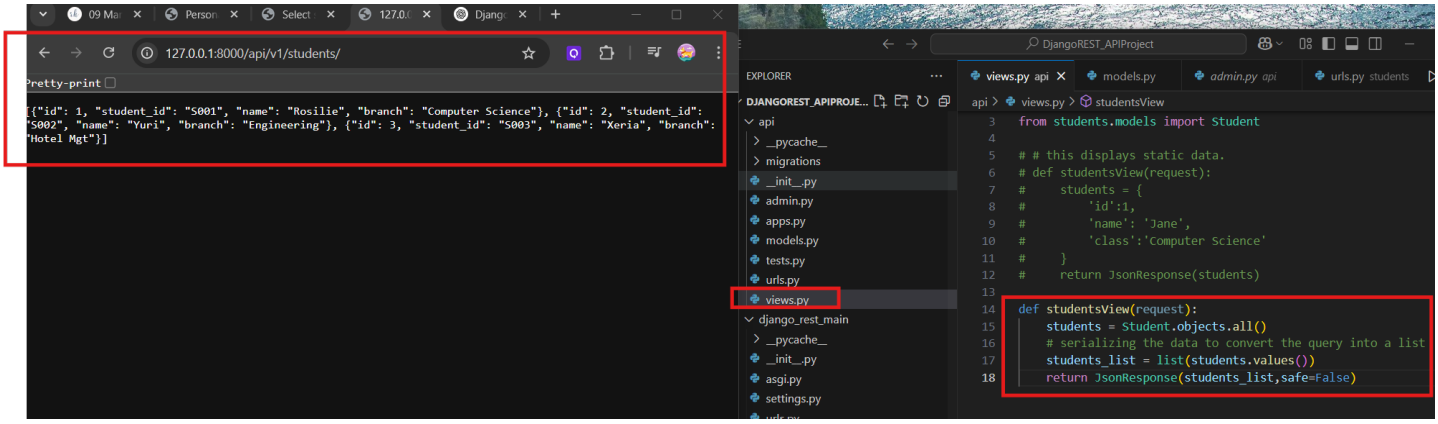
12. Create a sample record in the table.



13. To display the model student records using the admin dashboard:



14. To display the dynamic data from the database. We need to **SERIALIZE** to convert the returned query set into a list.



15. SERIALIZERS are like TRANSLATORS that convert certain data i.e QUERYSET from your database into other types of data like JSON data that can be used on HTML. While DESERIALIZERS will reverse the translation i.e from JSON file into Query set (database records).
