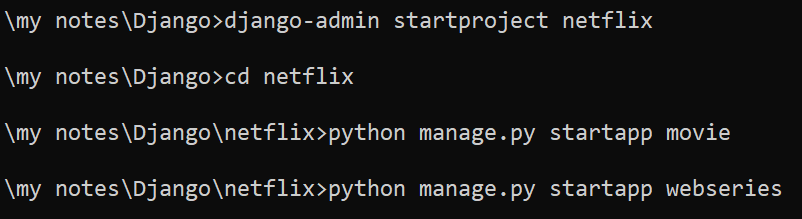
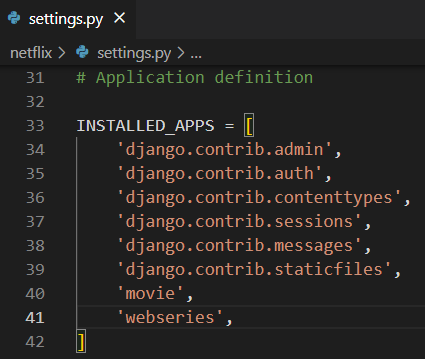
NetFlix project with MYSQL database

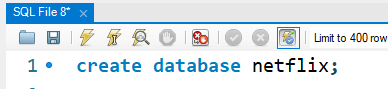
1. Create netflix project , Create 2 applications : movie and webseries and open project in vs code



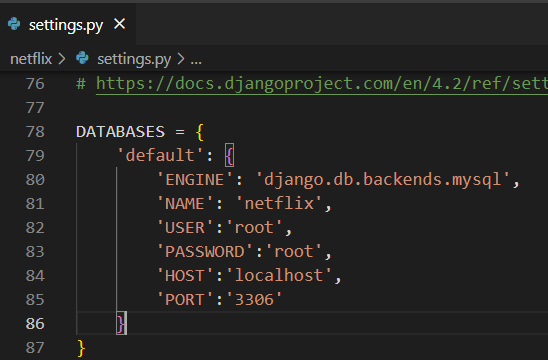
1. Register applications in settings.py

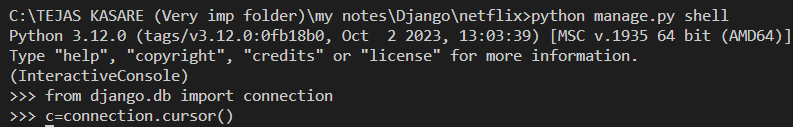


1. Working with mysql database
   1. Be ready with database name, username and password
   2. Here I am using default username and password of mysql i.e root and root
   3. Create new database for our project in workbench

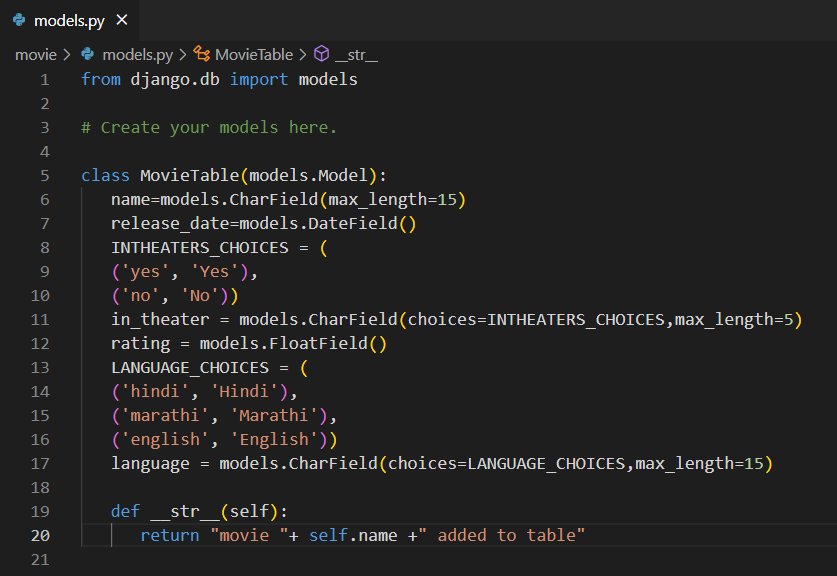


* 1. In setting.py register above database with user credentials

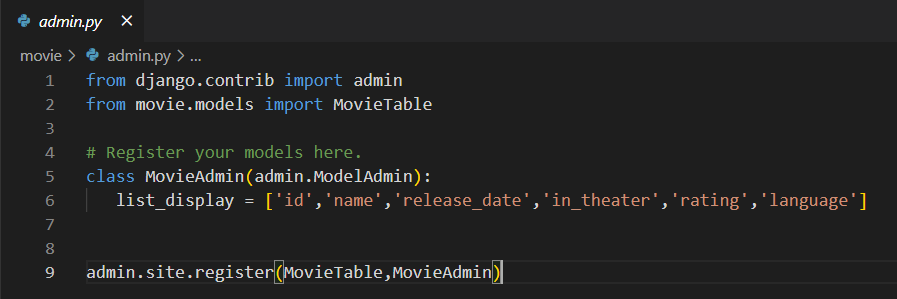


* 1. Install mysqlclient library
  2. Runserver : open terminal in vs code and run : >python manage.py runserver
  3. Open new terminal on vs code and check for django to mysql connetion by running following command
  4. 

4. Create model class in models.py



5. Register model class in admin.py

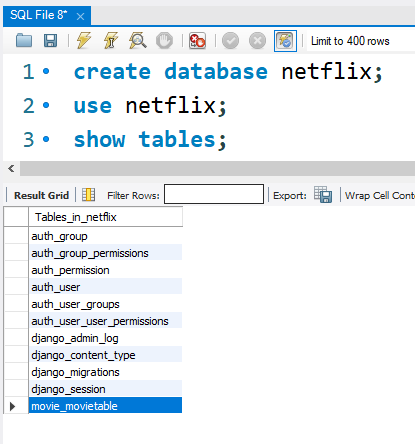


6. Makemigrations and migrate

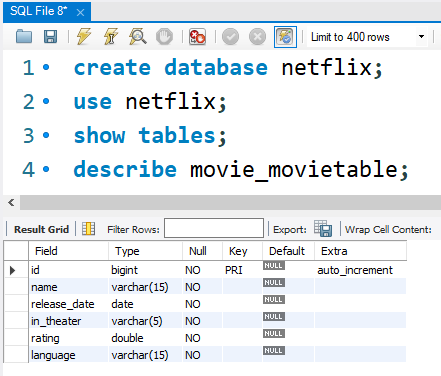
>python manage.py makemigrations

>python manage.py migrate

7. check tables in flipkart database



8. View structure of movie\_movietable table



9. DONE !!!!!!!!!!!!!!!

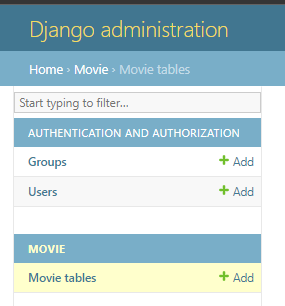
10. Creating superuser to access admin panel

>python manage.py createsuperuser

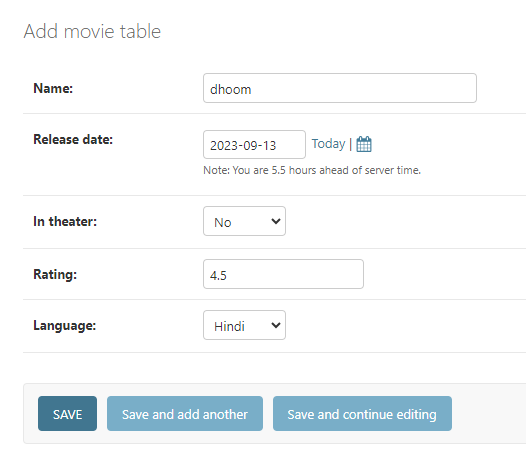
I have created with credentials : tejas, admin@123

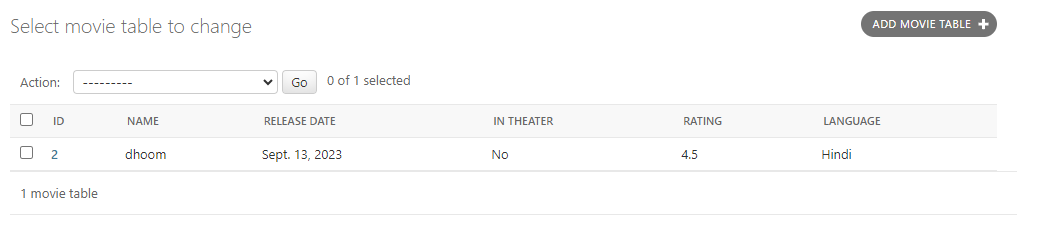
11. Open admin panel by : *http://127.0.0.1:8000/admin* and add above credentials

12. You will get movie table

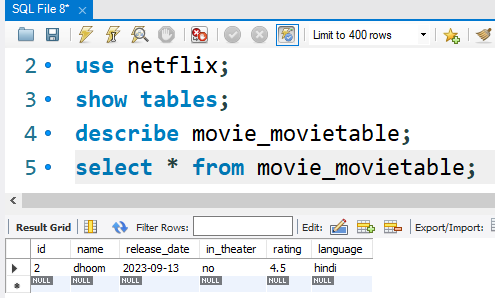


13. Add some data in movie table





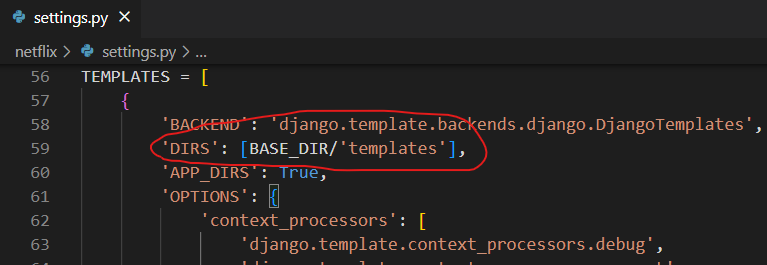
14. Check in mysql



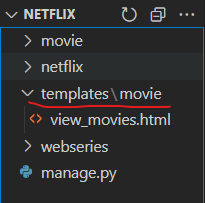
15. DONE !!!!!!!!!!!!!!!!!

16. Displaying products into HTML page

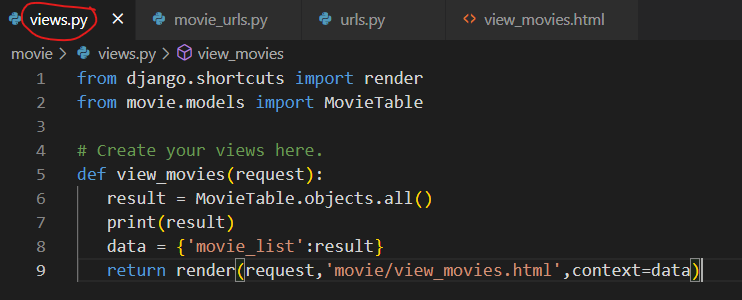
1. Create templates folder -> register templates folder in settings.py



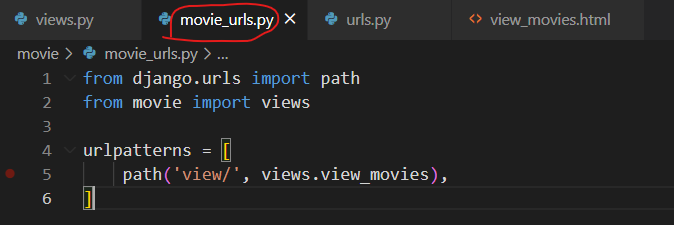
1. create movie folder inside templates folder -> create view\_movies.html in movie folder



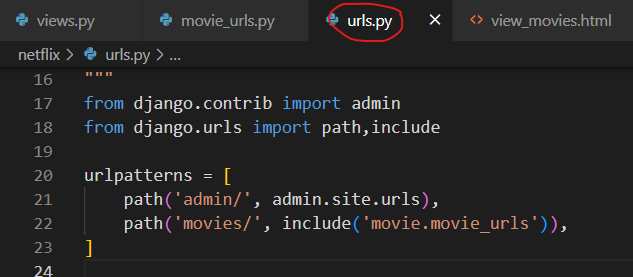
1. create view to fetch all data from model
   1. movie -> views.py



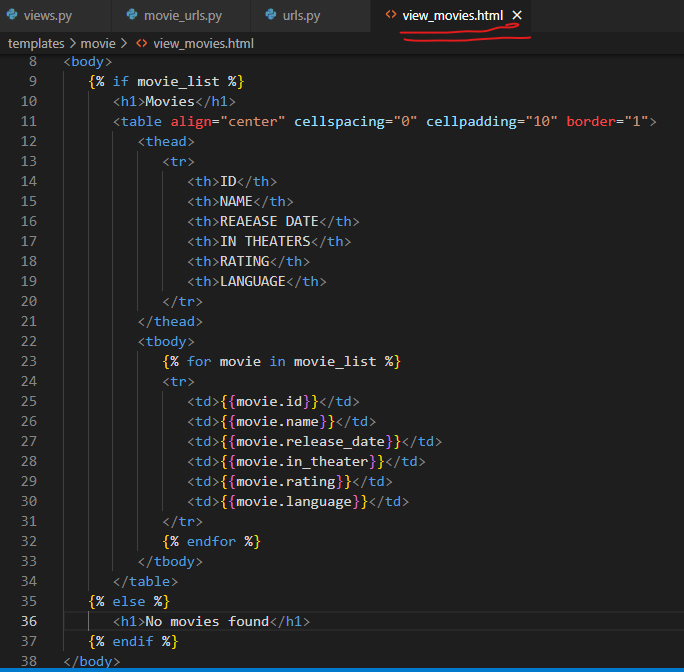
1. create application level url for above view
   1. create movie\_urls.py in movie folder



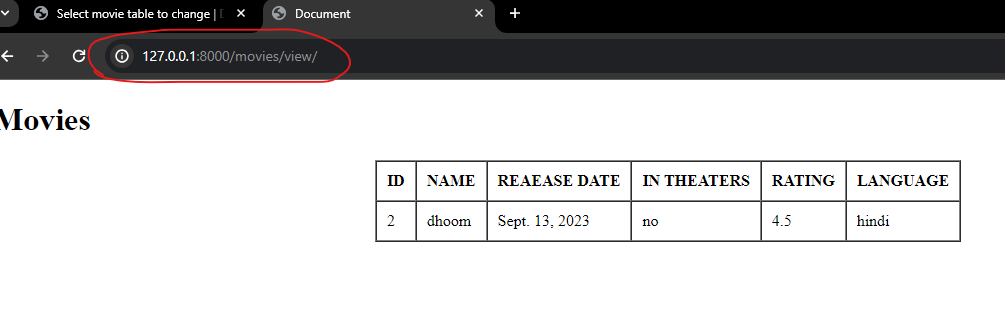
1. register above application level url in project level url : urls.py



1. create table in view\_movies.html to show all records



1. check for output



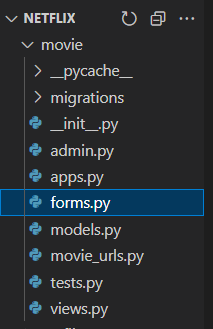
1. DONE !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

CREATING A FORM TO ADD MOVIES

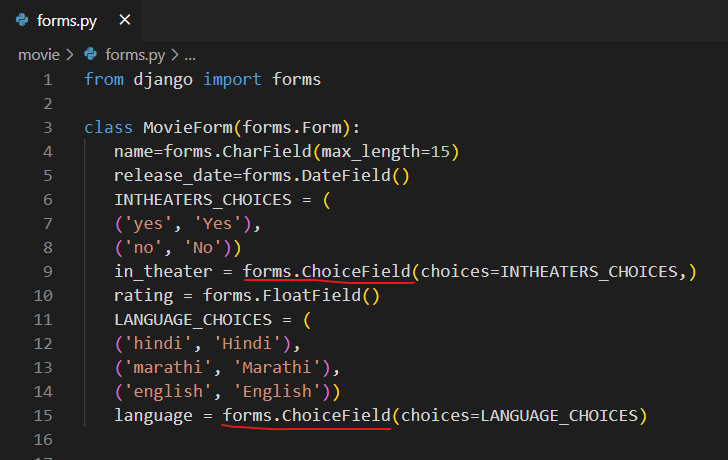
There are 2 types of forms available in django : django form and model form

Lets understand how to create form using django form

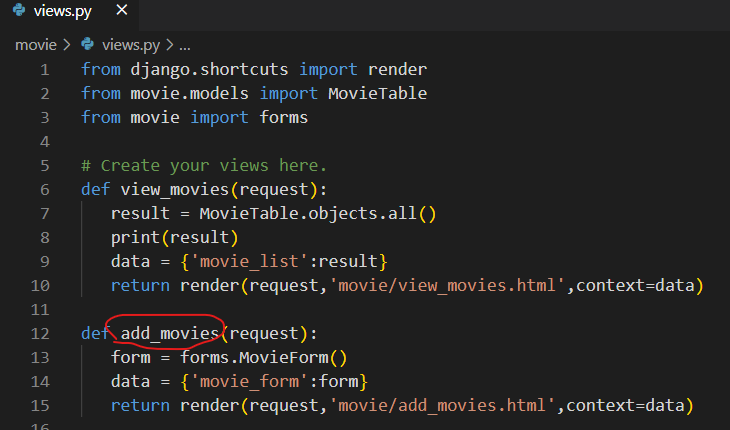
1. Create forms.py in application level (here inside movie folder)



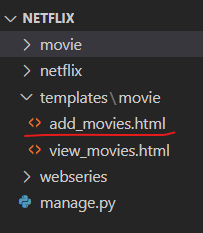
1. Add below code in it



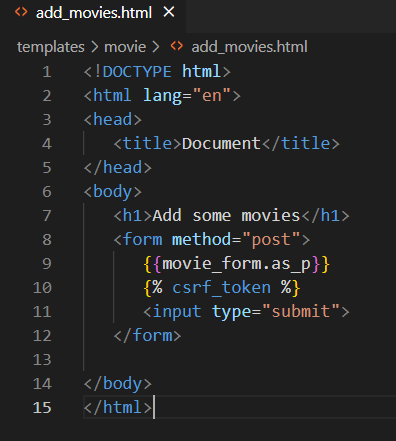
1. Register above form in views.py by create separate view to display form



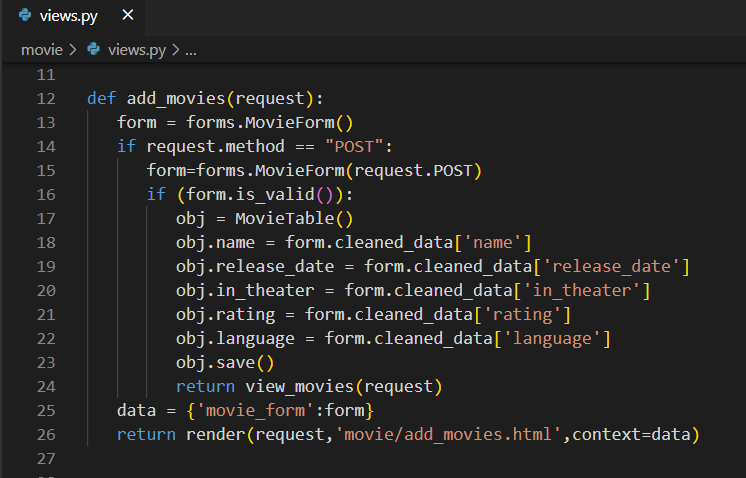
1. Create add\_movie.html in templates > movie > add\_movies.html



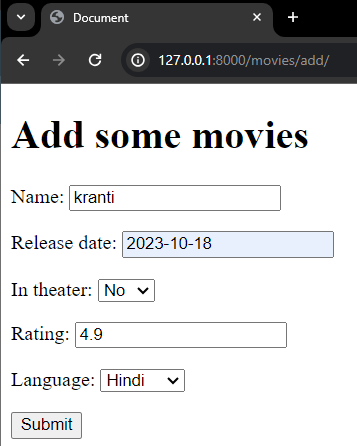
1. Add following code and csrf token to Solve csrf problem

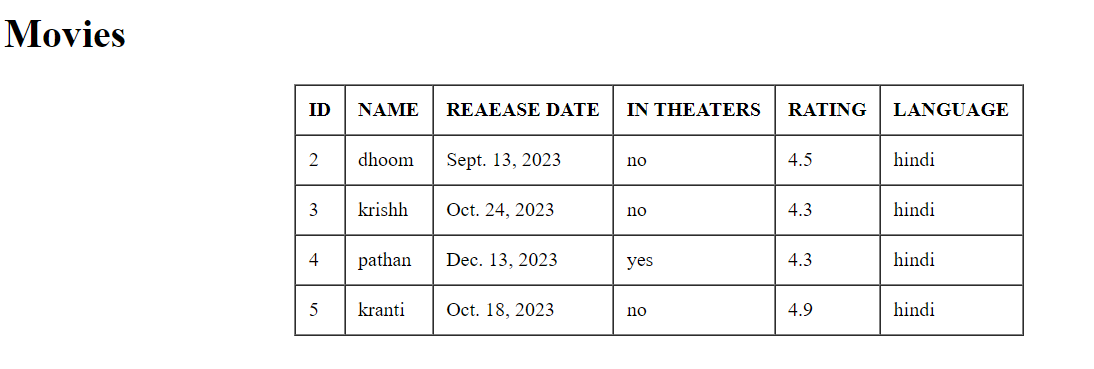


1. Logic to Collect data from form



1. Check for output

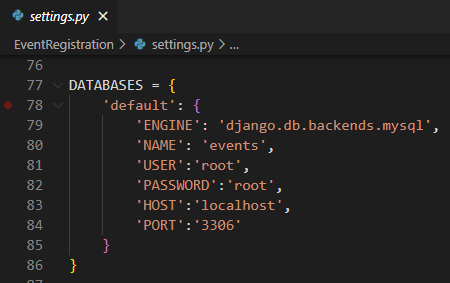




1. DONE !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

**MODEL FORM IN DJANGO**

1. Create django project : EventRegistration & open in VS code
2. Open VS code terminal and create Django app :bloodcamp
3. Register application in settings.py
4. Create database “events” in workbench
5. Add database details in settings.py



1. Start server
2. Check for database connection(open shell and do it)

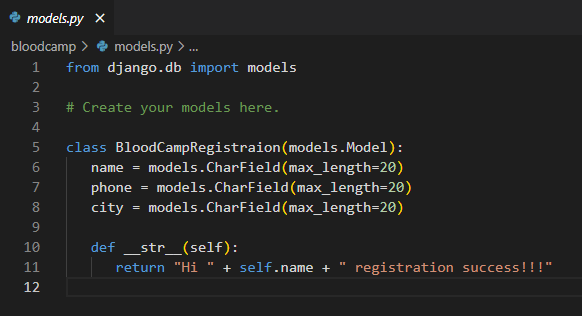
>python manage.py shell

>>>from django.db import connection

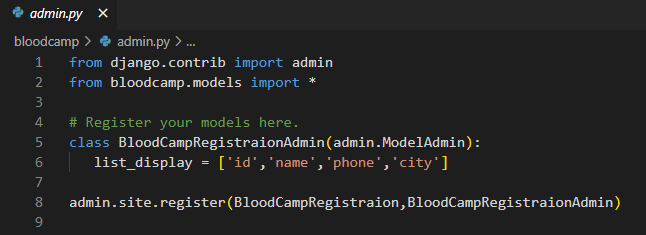
>>>c = connection.cursor()

IF NO ERROR THEN GOOD TO GO

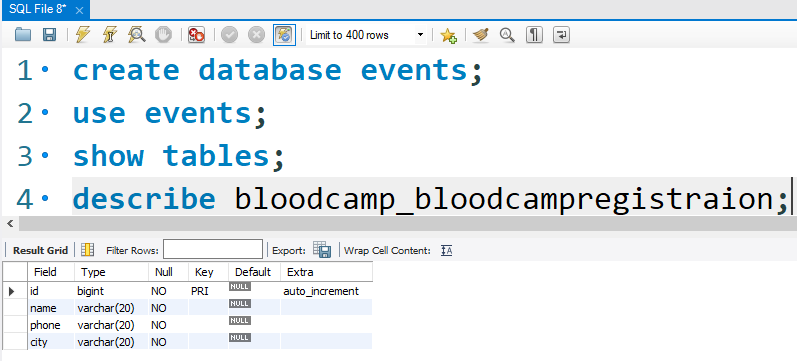
1. In bloodcamp application, create model class : BloodCampRegistraion (in model.py) with following fields : name, phone, city

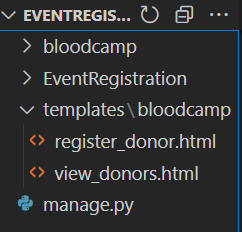


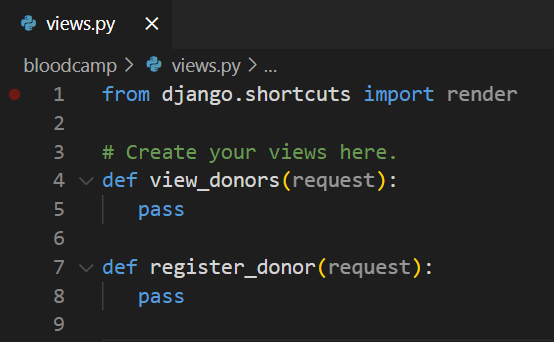
1. Register model class in admin.py



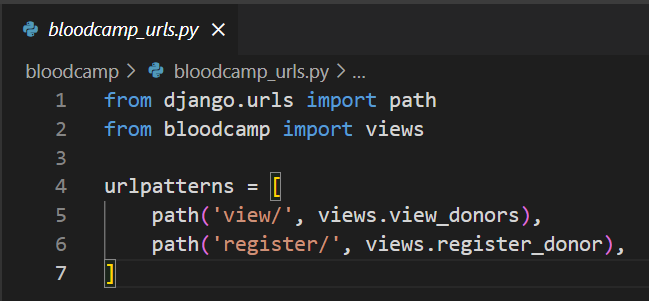
1. Makemigrations
2. Migrate
3. Check for table in workbench



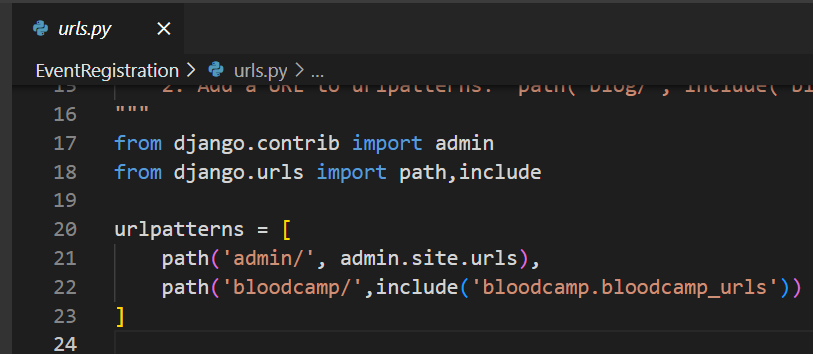
1. Create templates folder and register in settings.py
   1. Create bloodcamp folder in templates folder
      1. Create register\_donor.html and view\_donors.html
      2. 
2. Create 2 views in bloodcamp > views : view\_donors , register\_donor
   1. Initial code for views



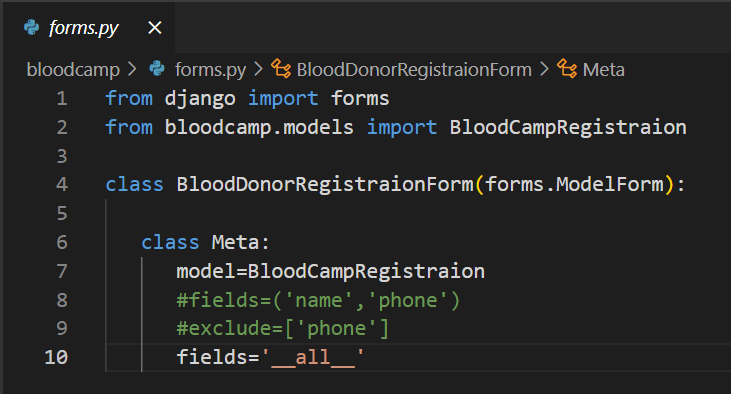
* 1. Create url pattern at application level for above view (create file bloodcamp\_urls.py)



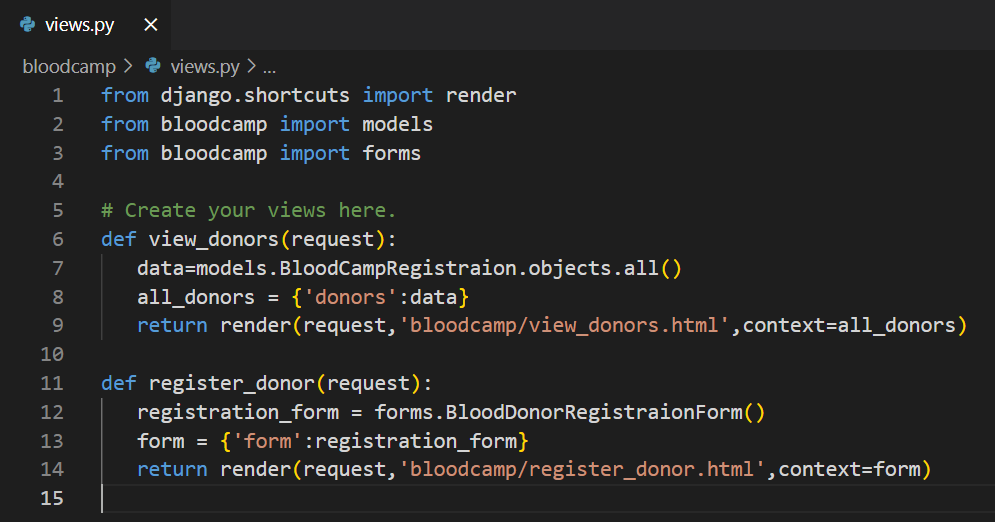
* 1. Include application level urls in project level urls.py



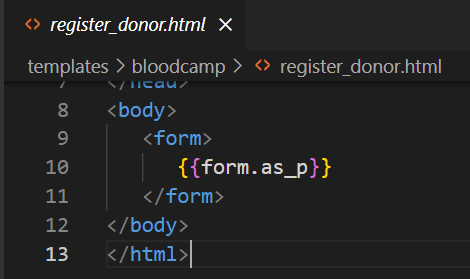
1. Create forms.py at application level (in bloodcamp folder) and add below code

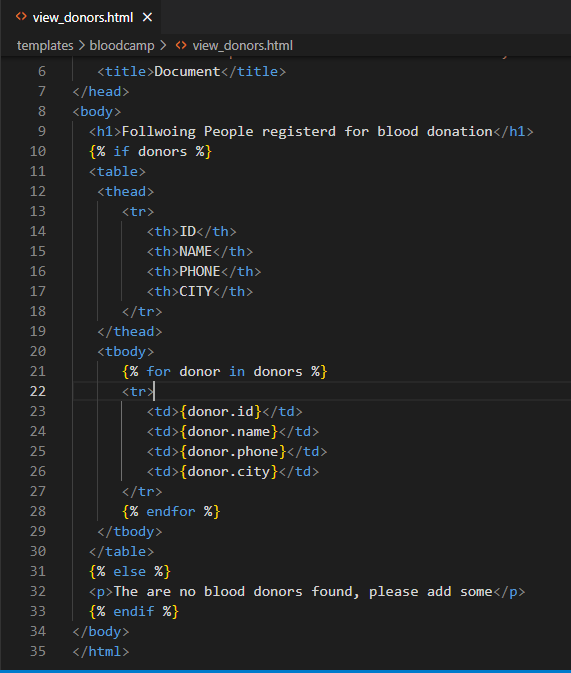


1. Code for above both views

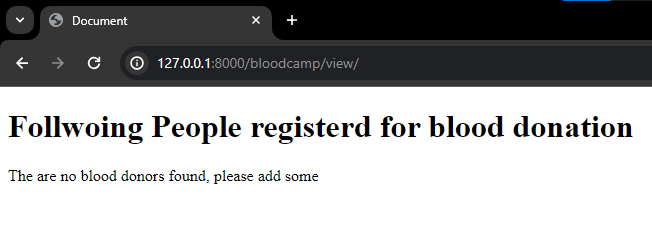


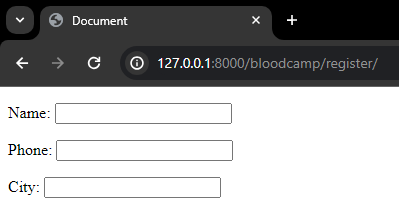
1. Code for above both html files





1. Restart sever
2. Final output





1. DONE
2. Sds

