

搭建流程

搭建虚拟环境(jerry)

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.18363.900]
(c) 2019 Microsoft Corporation。保留所有权利。

C:\Users\Lenovo>mkvirtualenv -p C:\Python311\python.exe jerry
created virtual environment CPython3.11.4.final.0-64 in 1412ms
creator CPython3Windows(dest=C:\Users\Lenovo\Envs\jerry, clear=False, no_vcs_ignore=False, global=False)
seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=C:\Users\Lenovo\AppData\Local\pypa\virtualenv)
added seed packages: pip==23.1.2, setuptools==68.0.0, wheel==0.40.0
activators BashActivator, BatchActivator, FishActivator, NushellActivator, PowerShellActivator, PythonActivator

(jerry) C:\Users\Lenovo>
```

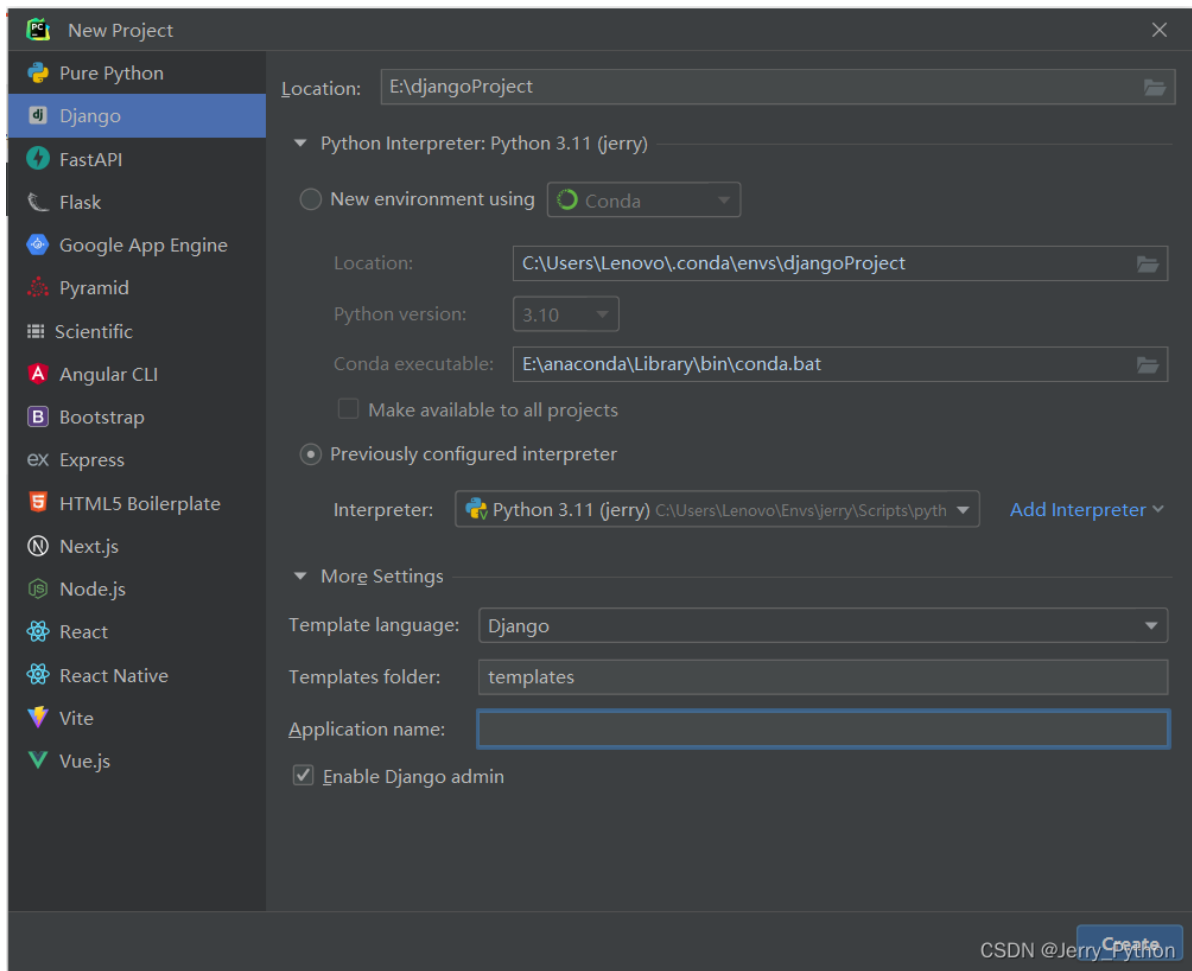
安装最新版本的Django(4.2.4)

```
(jerry) C:\Users\Lenovo>pip install django
Collecting django
  Using cached Django-4.2.4-py3-none-any.whl (8.0 MB)
Collecting asgiref<4, >=3.6.0 (from django)
  Using cached asgiref-3.7.2-py3-none-any.whl (24 kB)
Collecting sqlparse>=0.3.1 (from django)
  Using cached sqlparse-0.4.4-py3-none-any.whl (41 kB)
Collecting tzdata (from django)
  Using cached tzdata-2023.3-py2.py3-none-any.whl (341 kB)
Installing collected packages: tzdata, sqlparse, asgiref, django
Successfully installed asgiref-3.7.2 django-4.2.4 sqlparse-0.4.4 tzdata-2023.3
```

安装openai(0.27.8)

```
(jerry) C:\Users\Lenovo>pip install openai
```

使用Pycharm新建Django项目



如果你不使用Pycharm，请查阅官方文档进行项目的建立

Django 文档 | Django 文档 | Django

Run manage.py创建应用程序

```
manage.py@djangoProject > startapp chatgpt
C:\Users\Lenovo\Envs\jerry\Scripts\python.exe "E:\PyCharm 2022.3.3\plugins\python\helpers\pycharm\django_manage.py" startapp chatgpt
Tracking file by folder pattern: migrations

Following files were affected
E:\djangoProject\chatgpt\migrations\__init__.py
Process finished with exit code 0
```

将应用chatgpt配置到INSTALLED_APPS中:

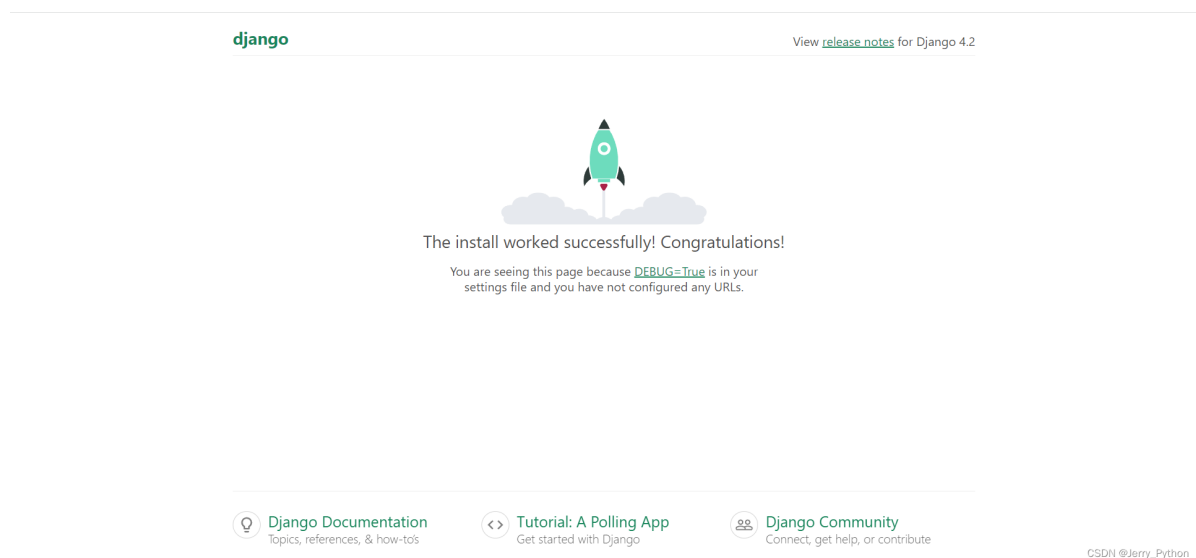
```
INSTALLED_APPS = [  
    ...  
    "django.contrib.admin",  
    "django.contrib.auth",  
    "django.contrib.contenttypes",  
    "django.contrib.sessions",  
    "django.contrib.messages",  
    "django.contrib.staticfiles",  
    "chatgpt",  
]
```

CSDN @Jerry_Python

测试服务器能否运行成功

```
manage.py@djangoProject > runserver  
C:\Users\Lenovo\Envs\jerry\Scripts\python.exe "E:\PyCharm 2022.3.3\plugins\python\helpers\pycharm\django_manage.py" runserver E:/dja  
Tracking file by folder pattern: migrations  
Watching for file changes with StatReloader  
Performing system checks...  
  
System check identified no issues (0 silenced).  
  
You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, conten  
Run 'python manage.py migrate' to apply them.  
August 06, 2023 - 10:50:36  
Django version 4.2.4, using settings 'djangoProject.settings'  
Starting development server at http://127.0.0.1:8000/  
Quit the server with CTRL-BREAK.  
  
CSDN @Jerry_Python
```

在浏览器中输入URL: <http://127.0.0.1:8000/> 并访问, 若出现下图则表示服务器运行成功

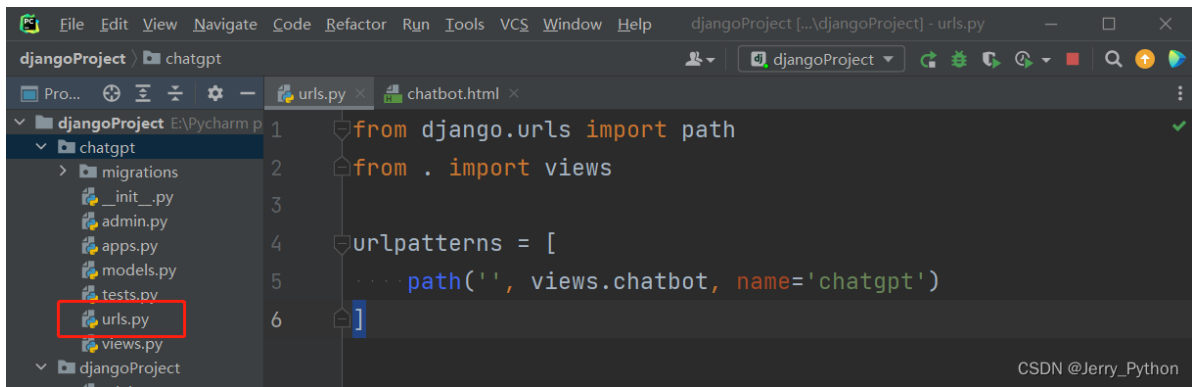


下载html文件, 复制到templates文件夹中

GitHub网址:

[GitHub - Jerrygjr/django-chatbot](https://github.com/Jerrygjr/django-chatbot)

在chatgpt应用下新建urls.py

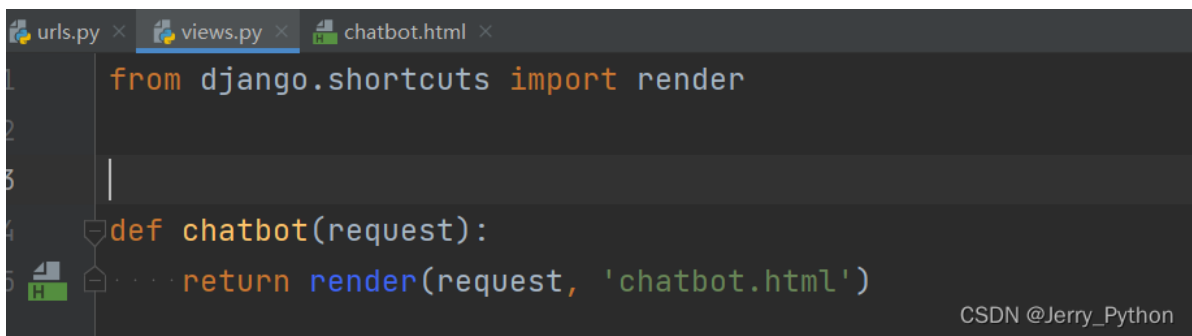


The screenshot shows the PyCharm IDE with a Django project named 'djangoProject'. The 'chatgpt' app directory is expanded in the left sidebar, showing files like migrations, __init__.py, admin.py, apps.py, models.py, tests.py, and views.py. A new file 'urls.py' has been created and is highlighted with a red box. The main editor shows the content of 'urls.py' with the following code:

```
1 from django.urls import path
2 from . import views
3
4 urlpatterns = [
5     path('', views.chatbot, name='chatgpt')
6 ]
```

The bottom right corner of the IDE window displays 'CSDN @Jerry_Python'.

在view.py中渲染html文件:

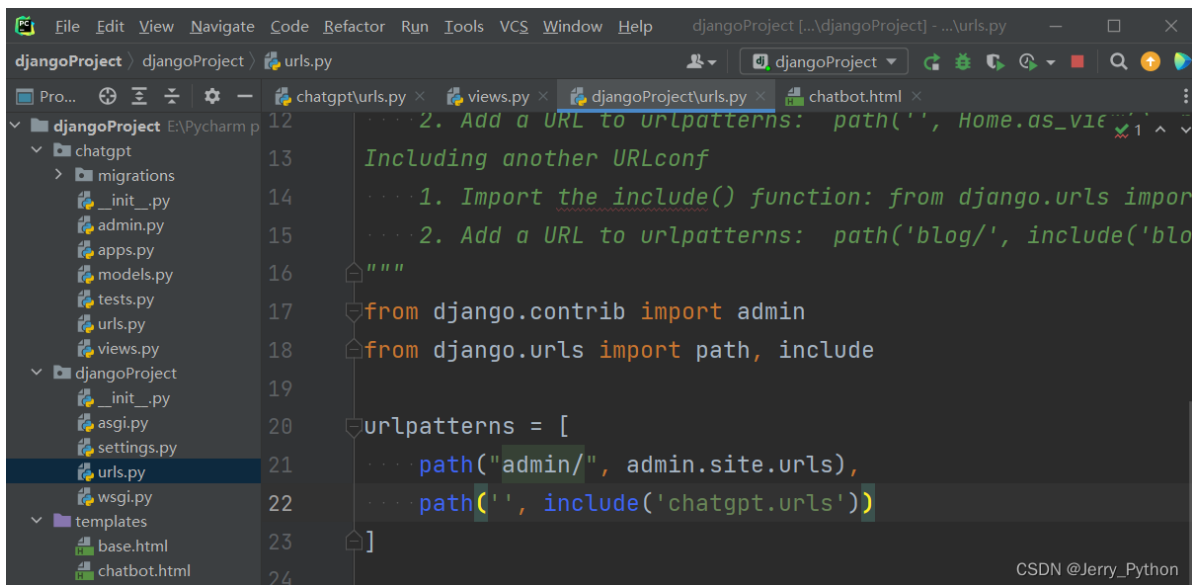


The screenshot shows the PyCharm IDE with the 'views.py' file open in the 'chatgpt' app directory. The code in 'views.py' is as follows:

```
1 from django.shortcuts import render
2
3
4 def chatbot(request):
5     return render(request, 'chatbot.html')
```

The bottom right corner of the IDE window displays 'CSDN @Jerry_Python'.

主文件夹下配置url:

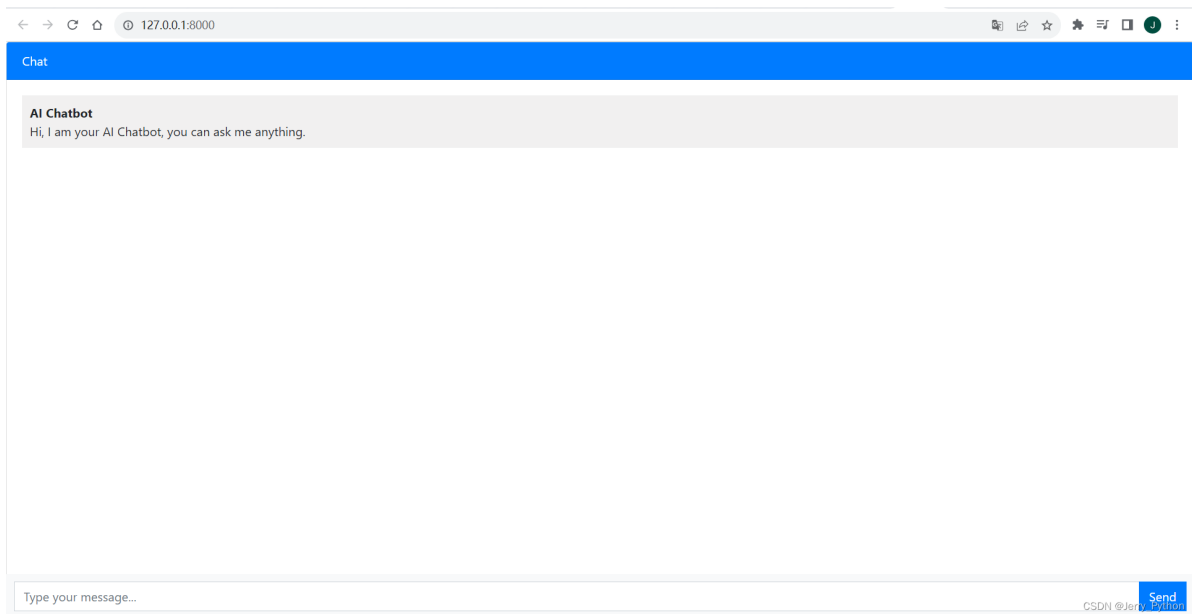


The screenshot shows the PyCharm IDE with the 'djangoProject/urls.py' file open. The code in this file is as follows:

```
12 """
13     2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
14     Including another URLconf
15     1. Import the include() function: from django.urls import include
16     2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
17 """
18 from django.contrib import admin
19 from django.urls import path, include
20
21 urlpatterns = [
22     path("admin/", admin.site.urls),
23     path("", include('chatgpt.urls'))
24 ]
```

The bottom right corner of the IDE window displays 'CSDN @Jerry_Python'.

在浏览器中再次输入URL: <http://127.0.0.1:8000/> 并访问



实现用户与chatbot交互功能:

1) 表单提交功能:

```
<form class="message-form">
  {%csrf_token%}
  <div class="input-group">
    <input type="text" class="form-control message-input" placeholder="Type your message...">
    <div class="input-group-append">
      <button type="submit" class="btn btn-primary btn-send">Send</button>
    </div>
  </div>
</form>
</div>
```

CSDN @Jerry_Python

2) 选择特定的查询或特定的标签

```
<script>
const messagesList = document.querySelector('.messages-list');
const messageForm = document.querySelector('.message-form');
const messageInput = document.querySelector('.message-input');
```

CSDN @Jerry_Python

3) 事件监听器

```
messageForm.addEventListener('submit', (event) => {
  event.preventDefault(); //按下按钮后不刷新页面

  const message = messageInput.value.trim(); //删除空格
  if (message.length === 0) { //消息长度为0时, 不返回任何内容
    return;
  }
});
```

CSDN @Jerry_Python

4) 按下按钮后在页面中附加消息项，显示用户输入的文本，并清空输入框。

```
const messageItem = document.createElement('li');
messageItem.classList.add('message', 'sent');
messageItem.innerHTML = `
  <div class="message-text">
    <div class="message-sender">
      <b>You</b>
    </div>
    <div class="message-content">
      ${message}
    </div>
  </div>`;
messagesList.appendChild(messageItem); //将messageItem添加到messagesList
messageInput.value = ''; //发送后文本框清空
```

CSDN @Jerry_Python

5) 用fetch来向后端发送请求，同上，将response附加到消息项中

```
fetch('', {
  method: 'POST',
  headers: { 'Content-Type': 'application/x-www-form-urlencoded' },
  body: new URLSearchParams({
    'csrfmiddlewaretoken': document.querySelector('[name=csrfmiddlewaretoken]').value,
    'message': message
  })
})
.then(response => response.json())
.then(data => {
  const response = data.response;
  const messageItem = document.createElement('li');
  messageItem.classList.add('message', 'received');
  messageItem.innerHTML = `
    <div class="message-text">
      <div class="message-sender">
        <b>AI Chatbot</b>
      </div>
      <div class="message-content">
        ${response}
      </div>
    </div>
  `;
  messagesList.appendChild(messageItem);
});
```

CSDN @Jerry_Python

6) 修改view.py返回json响应

```
1 from django.shortcuts import render
2 from django.http import JsonResponse
3
4
5 def chatbot(request):
6     if request.method == 'POST':
7         message = request.POST.get('message')
8         response = 'This is my response'
9         return JsonResponse({'message': message, 'response': response})
10    return render(request, 'chatbot.html')
```

CSDN @Jerry_Python

You
hello

AI Chatbot

This is my response

CSDN @Jerry_Python

新建一个API key:

API keys

Your secret API keys are listed below. Please note that we do not display your secret API keys again after you generate them.

Do not share your API key with others, or expose it in the browser or other client-side code. In order to protect the security of your account, OpenAI may also automatically disable any API key that we've found has leaked publicly.

NAME	KEY	CREATED	LAST USED ⓘ	
Secret key	sk-...EdPw	2023年8月11日	2023年8月11日	✎ 🗑
chatbot	sk-...HD7G	2023年8月12日	Never	✎ 🗑

+ Create new secret key

CSDN @Jerry_Python

不过我的API免费额度到期了，在淘宝上买了一个，可参考我的上一篇文章。

[GitHub - Jerryqjr/chatgpt-django: 使用Django开发的chatbot，调用gpt3.5-API](#)

修改view.py接入chatgpt API

```
import openai

openai_api_key = 'sk-0q2lI1IqKZEfjXgzG63KT3B1bkFJm8QZLTQaNbGigffZHD'
openai.api_key = openai_api_key
```

CSDN @Jerry_Python

```
def ask_openai(message):
    openai.Model.list()
    response = openai.ChatCompletion.create(
        model='gpt-3.5-turbo',
        messages=[
            {"role": "system", "content": "You are an helpful assistant."},
            {"role": "user", "content": message},
        ],
    )

    answer = response.choices[0].message.content.strip() #strip() 删除字符串前置和后置空格
    return answer
```

CSDN @Jerry_Python

<

AI Chatbot

Hi, I am your AI Chatbot, you can ask me anything.

>

You

聊聊你对中国房地产的看法

AI Chatbot

作为一个智能助手，我可以为您提供一些对中国房地产市场的一般观点。中国房地产市场在过去几十年里经历了快速发展和持续增长。这一行业对中国经济的发展起到了关键的推动作用，为就业提供了大量机会，并为国内各种工业和服务业提供了支持。然而，也存在一些问题和挑战需要面对。首先，房地产市场的高速增长造成了房价过高的问题。在一些热门城市，房价上涨过快，使得购房成为了许多人难以负担的负担。这也导致了部分资金注入房地产，而不是在其他有利于经济发展的行业和领域投资。此外，房地产市场的投资热潮也存在一定的风险。一些人为了追求投资回报，过度依赖于购买房产。这使得市场价格容易受到投机者的影响，同时也增加了市场的波动性和不确定性。最近几年，中国政府采取了一系列措施来调控房地产市场。例如，限制房贷额度、加强土地供应和推出房地产税等政策。这些措施旨在遏制房价过快上涨，并推动市场的健康发展。总体而言，中国房地产市场在经历了快速发展的同时，也面临着一些挑战。政府的调控措施和加强监管有助于确保市场的平稳运行。这一市场对中国经济的发展仍然具有重要意义，但需要注意平衡各种利益和风险，以实现可持续发展。

CSDN @Jerry_Python

完成用户注册、登录、注销功能，并把聊天记录保存到数据库：

```
def chatbot(request):...

def login(request):
    ... return render(request, 'login.html')

def register(request):
    ... return render(request, 'register.html')

def logout(request):
    auth.logout(request)
```

CSDN @Jerry_Python

Login

Username

Jerry

Password

.....

Login

CSDN @Jerry_Python

实现注册并自动登录功能：

- 1) 先判断两次输入密码是否相同

```
def register(request):  
    if request.method == 'POST':  
        username = request.POST['username']  
        email = request.POST['email']  
        password1 = request.POST['password1']  
        password2 = request.POST['password2']  
  
        if password1 == password2:  
            pass  
        else:  
            error_message = 'Password dont match'  
            return render(request, 'register.html', {'error_message': error_message})  
    return render(request, 'register.html')
```

CSDN @Jerry_Python

Register

Password dont match

Username

Email

Password

Confirm Password

Register

CSDN @Jerry_Python

2) 补全创建用户功能并迁移表之后

```
def register(request):
    if request.method == 'POST':
        username = request.POST['username']
        email = request.POST['email']
        password1 = request.POST['password1']
        password2 = request.POST['password2']

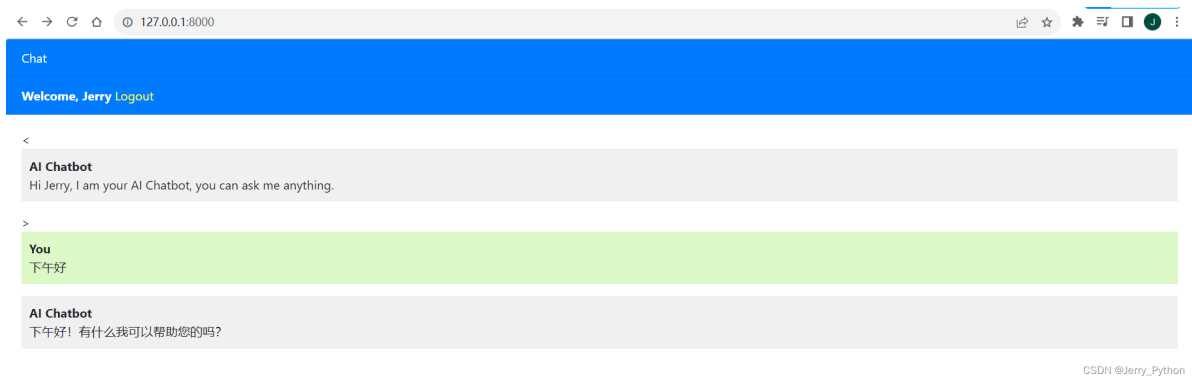
        if password1 == password2:
            try:
                user = User.objects.create_user(username, email, password1)
                user.save()
                auth.login(request, user)
                return redirect('chatbot')
            except:
                error_message = 'Error creating account'
                return render(request, 'register.html', {'error_message': error_message})
        else:
            error_message = 'Password dont match'
            return render(request, 'register.html', {'error_message': error_message})
    return render(request, 'register.html')
```

CSDN @Jerry_Python

```
<div class="message-content">
    Hi {{user.username}}, I am your AI Chatbot, you can ask me anything.
</div>
```

CSDN @Jerry_Python

注册成功后:



CSDN @Jerry_Python

修改chatbot.html中的前端展示效果:

```
{% block content %}
<div class="chat-container">
    <div class="card flex-grow-1">
        <div class="card-header bg-primary text-white">Chat</div>
        {% if user.is_authenticated %}
        <div class="card-header bg-primary text-white"><b>Welcome, {{user.username}}</b> <a style="color: yellow;" href="logout">Logout</a></div>
        {% else %}
        <div class="card-header bg-primary text-white"><a style="color: yellow;" href="login">Login</a> <a style="color: yellow;" href="register">Register</a></div>
        {% endif %}
        <div class="card-body messages-box">
```

CSDN @Jerry_Python

3) 新建数据表来保存聊天记录

```
oProject\urls.py x manage.py x chatgpt\urls.py x chatgpt\views.py x models.py x tests.py x settings.py x chatbot\views.py x regis
from django.db import models
from django.contrib.auth.models import User

class Chat(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    message = models.TextField()
    response = models.TextField()
    created_at = models.DateTimeField(auto_created=True)

    def __str__(self):
        return f'{self.user.username}: {self.message}'
```

CSDN @Jerry_Python

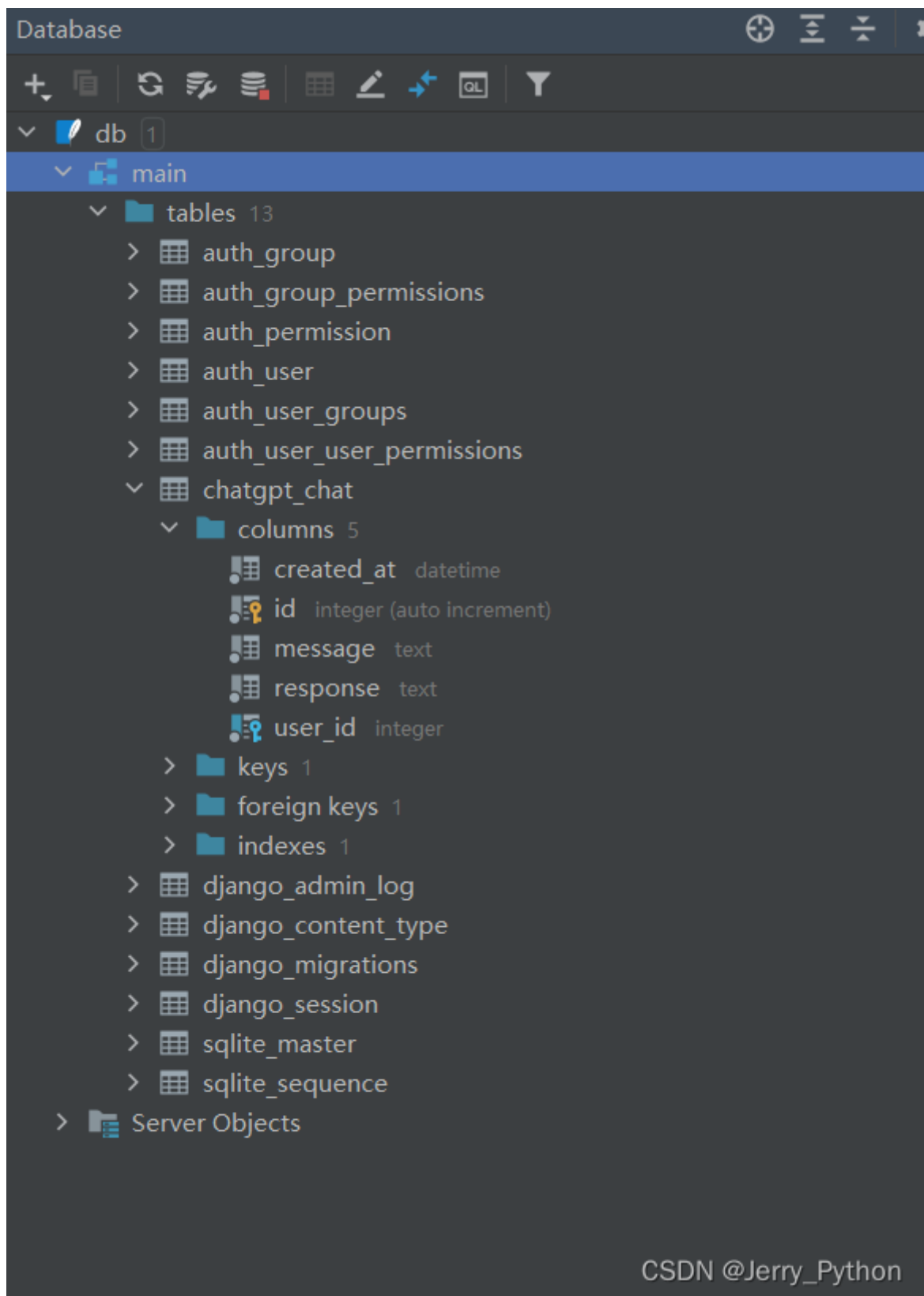
迁移表

```
manage.py@djangoproject > makemigrations
C:\Users\Lenovo\Envs\jerry\Scripts\python.exe "E:\PyCharm 2022.3.3\plugins\p
Tracking file by folder pattern: migrations
Migrations for 'chatgpt':
  chatgpt\migrations\0001_initial.py
    - Create model Chat

Following files were affected
  E:\Pycharm project\djangoproject\chatgpt\migrations\0001_initial.py
Process finished with exit code 0
manage.py@djangoproject > migrate
C:\Users\Lenovo\Envs\jerry\Scripts\python.exe "E:\PyCharm 2022.3.3\plugins\p
Tracking file by folder pattern: migrations
Operations to perform:
  Apply all migrations: admin, auth, chatgpt, contenttypes, sessions
Running migrations:
  Applying chatgpt.0001_initial... OK

manage.py@djangoproject >
```

CSDN @Jerry_Python



4) 使用django自带的后台管理系统

createsuperuser

```
manage.py@djangoProject > createsuperuser
C:\Users\Lenovo\Envs\jerry\Scripts\python.exe "E:\PyCharm 2022.3.3\plu
Tracking file by folder pattern: migrations
Username (leave blank to use 'lenovo'): Jerryadmin
Email address: [REDACTED]@qq.com
Warning: Password input may be echoed.
Password: 123456
Warning: Password input may be echoed.
Password (again): 123456
This password is too short. It must contain at least 8 characters.
This password is too common.
This password is entirely numeric.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.

Process finished with exit code 0
```

CSDN @Jerry_Python

将数据表导入后台管理系统

```
oProject\urls.py x manage.py x admin.py x chatgpt\urls.py x chatgpt\views.py x
from django.contrib import admin
from .models import Chat

# Register your models here.
admin.site.register(Chat)
```

CSDN @Jerry_Python

127.0.0.1:8000/admin/

Django administration

WELCOME, JERRYADMIN VIEW SITE / CHANGE PASSWORD / LOG OUT

Site administration

AUTHENTICATION AND AUTHORIZATION	
Groups	+ Add Change
Users	+ Add Change

CHATGPT	
Chats	+ Add Change

Recent actions

My actions

None available

CSDN @Jerry_Python

5) 保存聊天记录:

```
def chatbot(request):  
    if request.method == 'POST':  
        message = request.POST.get('message')  
        response = ask_openai(message)  
        chat = Chat(user=request.user, message=message, response=response, created_at=timezone.now())  
        chat.save()  
        return JsonResponse({'message': message, 'response': response})  
    return render(request, 'chatbot.html')
```

CSDN @Jerry_Python

The screenshot displays the Django administration interface. The top navigation bar includes the text "Django administration" and "WELCOME, JERRYADMIN VIEW SITE / CHANGE PASSWORD / LOG OUT". The breadcrumb trail shows "Home > Chatgpt > Chats > Jerryadmin: 你喜欢打篮球吗". The left sidebar contains a search bar and two main sections: "AUTHENTICATION AND AUTHORIZATION" with links for "Groups" and "Users" (each with a "+ Add" button), and "CHATGPT" with a link for "Chats" (+ Add). The main content area is titled "Change chat" and shows details for a chat record titled "Jerryadmin: 你喜欢打篮球吗". It includes a "HISTORY" button. The "Created at" section shows the date "2023-08-14" (Today) and time "08:52:11" (Now). A note states "Note: You are 8 hours ahead of server time." The "User" field is set to "Jerryadmin". The "Message" field contains the text "你喜欢打篮球吗". The "Response" field contains the text: "作为一个虚拟助手，我没有个人喜好或能力来打篮球。但是我可以帮助你找到打篮球的信息，例如篮球场、比赛规则等。如果你有任何关于篮球的问题或需要帮助，请告诉我，我会尽力提供帮助。"

CSDN @Jerry_Python

6) 接下来实现显示历史记录在网页的功能

```
{% for chat in chats %}
    {% if chat.user == request.user %}

        <li class="message sent">
        <div class="message-text">
            <div class="message-sender">
                <b>You</b>
            </div>
            <div class="message-content">|
                {{chat.message}}
            </div>
        </div>
    </li>

    <li class="message received">
        <div class="message-text">
            <div class="message-sender">
                <b>AI Chatbot</b>
            </div>
            <div class="message-content">
                {{chat.response}}
            </div>
        </div>
    </li>

    {% endif %}
{% endfor %}
```

CSDN @Jerry_Python

刷新页面即显示历史记录，完成咯



