

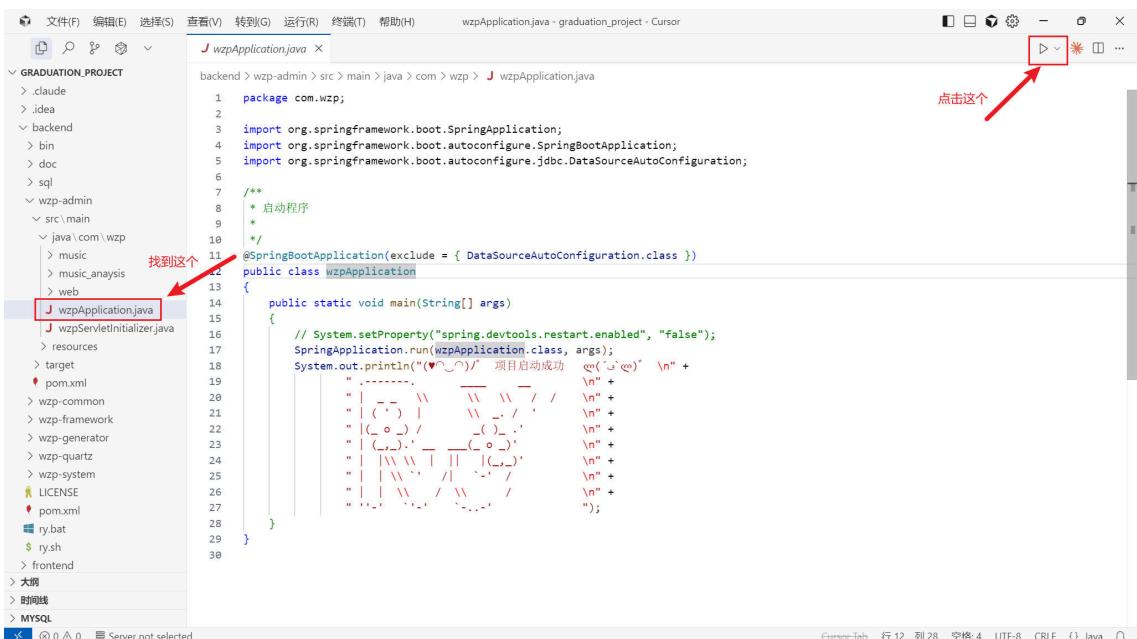
1. 打开redis(在D:\Redis目录下)

名称	修改日期	类型	大小
客户端	23/09/2025 9:47 PM	文件夹	
00-RELEASENOTES	23/09/2025 4:55 PM	文件	129 KB
dump.rdb	22/10/2025 6:51 PM	RDB 文件	14 KB
EventLog.dll	23/09/2025 4:55 PM	应用程序扩展	2 KB
README.txt	23/09/2025 4:55 PM	Text 源文件	1 KB
redis.windows.conf	23/09/2025 4:55 PM	CONF 文件	61 KB
redis.windows-service.conf	23/09/2025 4:55 PM	CONF 文件	61 KB
redis-benchmark.exe	23/09/2025 4:55 PM	应用程序	449 KB
redis-benchmark.pdb	23/09/2025 4:55 PM	Program Debug Da...	5,572 KB
redis-check-aof.exe	23/09/2025 4:55 PM	应用程序	1,813 KB
redis-check-aof.pdb	23/09/2025 4:55 PM	Program Debug Da...	9,620 KB
redis-check-rdb.exe	23/09/2025 4:55 PM	应用程序	1,813 KB
redis-check-rdb.pdb	23/09/2025 4:55 PM	Program Debug Da...	9,620 KB
redis-cli.exe	23/09/2025 4:55 PM	应用程序	614 KB
redis-cli.pdb	23/09/2025 4:55 PM	Program Debug Da...	5,980 KB
redis-server.exe	23/09/2025 4:55 PM	应用程序	1,813 KB
redis-server.pdb	23/09/2025 4:55 PM	Program Debug Da...	9,620 KB
RELEASENOTES.txt	23/09/2025 4:55 PM	Text 源文件	4 KB

双击

2. 把 graduation_project 文件夹拖入VScode图标打开

3. 运行后端程序E:\code\graduation_project\backend\wzp-admin\src\main\java\com\wzp\wzpApplication.java



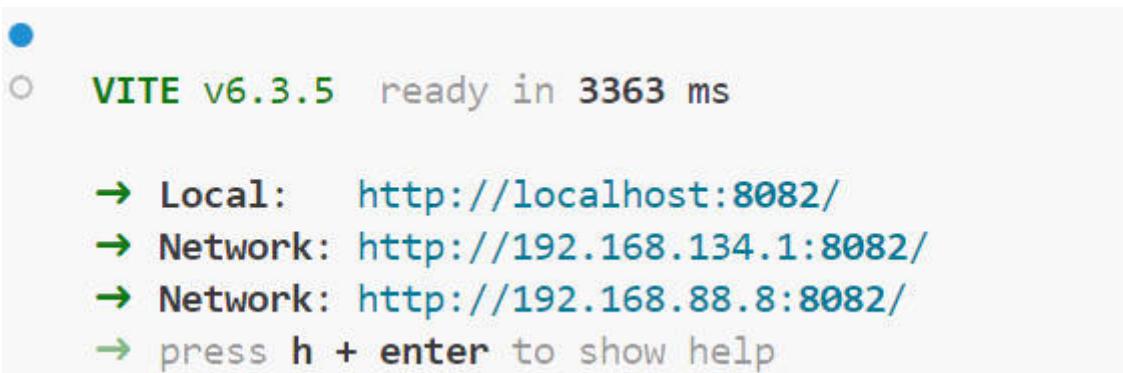
```
wzpApplication.java - graduation_project - Cursor
backend > wzp-admin > src > main > java > com > wzp > wzpApplication.java

1 package com.wzp;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5 import org.springframework.boot.autoconfigure.jdbc.DataSourceAutoConfiguration;
6
7 /**
8 * 启动程序
9 *
10 */
11 @SpringBootApplication(exclude = { DataSourceAutoConfiguration.class })
12 public class wzpApplication
13 {
14     public static void main(String[] args)
15     {
16         // System.setProperty("spring.devtools.restart.enabled", "false");
17         SpringApplication.run(wzpApplication.class, args);
18         System.out.println("（^o^）" 项目启动成功 "（`o`）" );
19         "-----";
20         " _ _ _ _ _ ";
21         " | ( ) | _ _ _ _ / ";
22         " | ( o ) / | ( o )_ ";
23         " | ( _ )_ / | ( _ ) ";
24         " | ( _ )_ / | ( _ ) ";
25         " | ( _ )_ / | ( _ ) ";
26         " | ( _ )_ / | ( _ ) ";
27         " | ( _ )_ / | ( _ ) ";
28     }
29 }

找到了这个 点击这个
```

4. 按window+r,输入cmd,打开终端, 输入 cd

C:\Users\guoda\Desktop\graduation_project\frontend 后, 运行前端代码 npm run dev 后回车, 等待一段时间直到出现这个。



5. 运行E:\code\graduation_project\process_interface下的所有算法代码

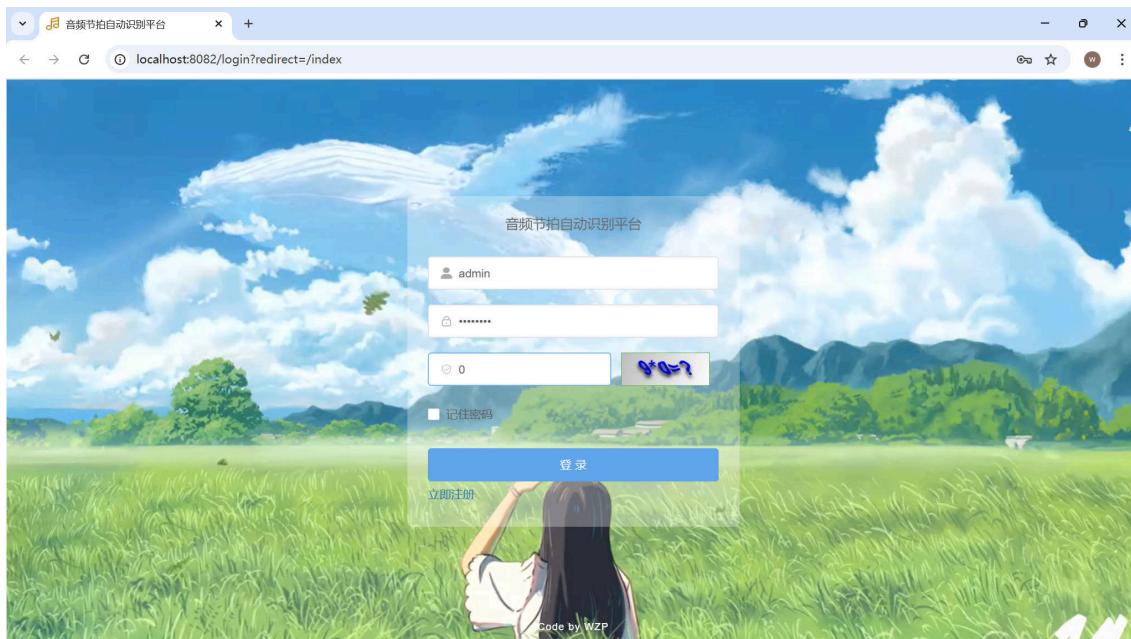
```

analyze.py

1  from flask import Flask, request, jsonify
2  import librosa
3  import numpy as np
4  import os
5  import re
6
7  app = Flask(__name__)
8
9  @app.route('/get_beatData', methods=['GET', 'POST'])
10 def analyze_audio():
11     # 处理GET和POST两种请求方式
12     if request.method == 'GET':
13         # 从查询参数获取路径
14         audio_path = request.args.get('audio_path')
15     else:
16         # 从JSON体获取路径
17         data = request.get_json(silent=True)
18         audio_path = data.get('audio_path') if data else None
19
20     if not audio_path:
21         return jsonify({'error': 'No audio path provided'}), 400
22
23     # 处理路径中的转义问题 - 使用字符串替换代替正则表达式
24     audio_path = audio_path.replace("\\", "\\\\") # 替换单反斜杠为双反斜杠
25     audio_path = audio_path.replace('/', os.path.sep) # 替换正斜杠为系统分隔符
26
27     # 处理路径中的引号问题
28     audio_path = audio_path.strip('"')
29
30     if not os.path.exists(audio_path):
31         return jsonify({'error': 'File not found: {}'.format(audio_path)}), 404
32
33     try:

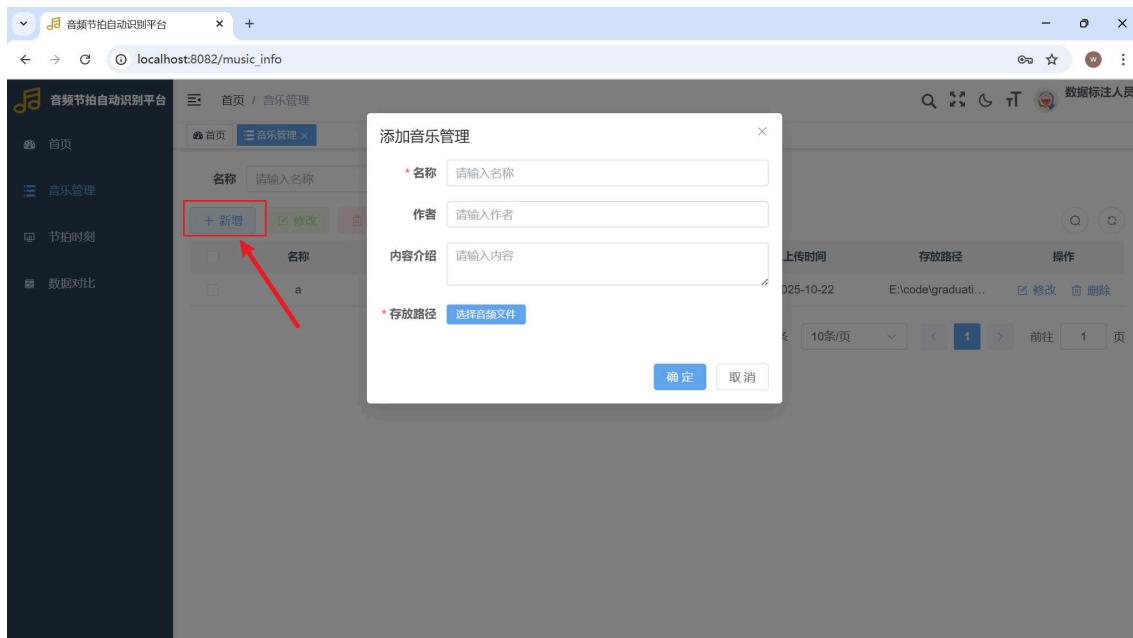
```

6. 在浏览器打开网址 http://localhost:8082/

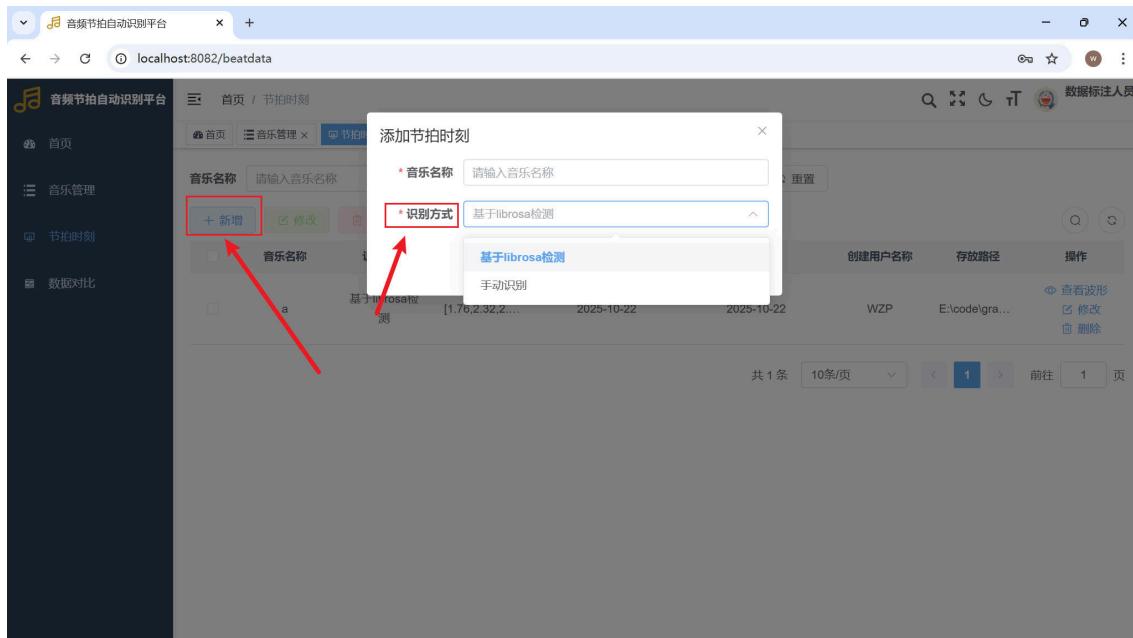


7. 数据标注人员账户: annotator, 密码: 123456; 超级用户账户admin,密码: admin123;我推荐老师用数据标注人员账户, 这样我好区分。

8. 音乐管理页面。点击新增按钮添加音频文件信息。



9. 节拍时刻页面。点击新增按钮，选择手动识别（没有节拍时刻标签）或者基于librosa检测自动识别（有节拍时刻标签，可能需要微调）



10. 点击查看波形，进入分析页面。点击“asdw”中的任意一个键都可以插入节拍时刻标签。鼠标左键可拖动节拍时刻标签或者靠近按“<-”或“->”键移动节拍时刻标签（点击一次移动0.01s,紧按可加速），鼠标右键可选择删除节拍时刻标签。



11. 数据对比页面。可选择多个同一音频的节拍时刻数据进行比对，目前不支持在这里修改。不同音频的数据不能进行比较！