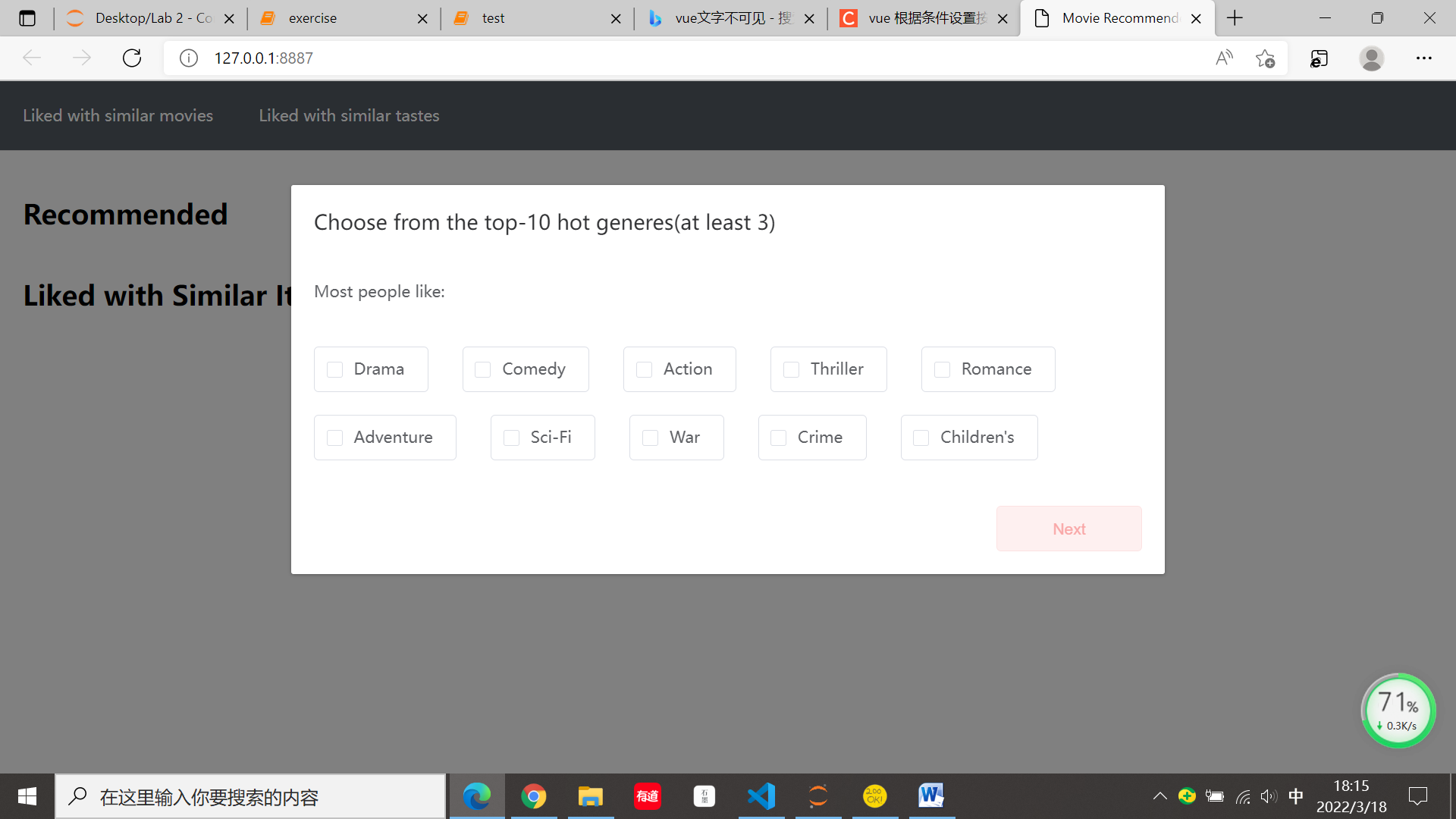
1. 初始化：显示top10受欢迎的genres，至少选3个

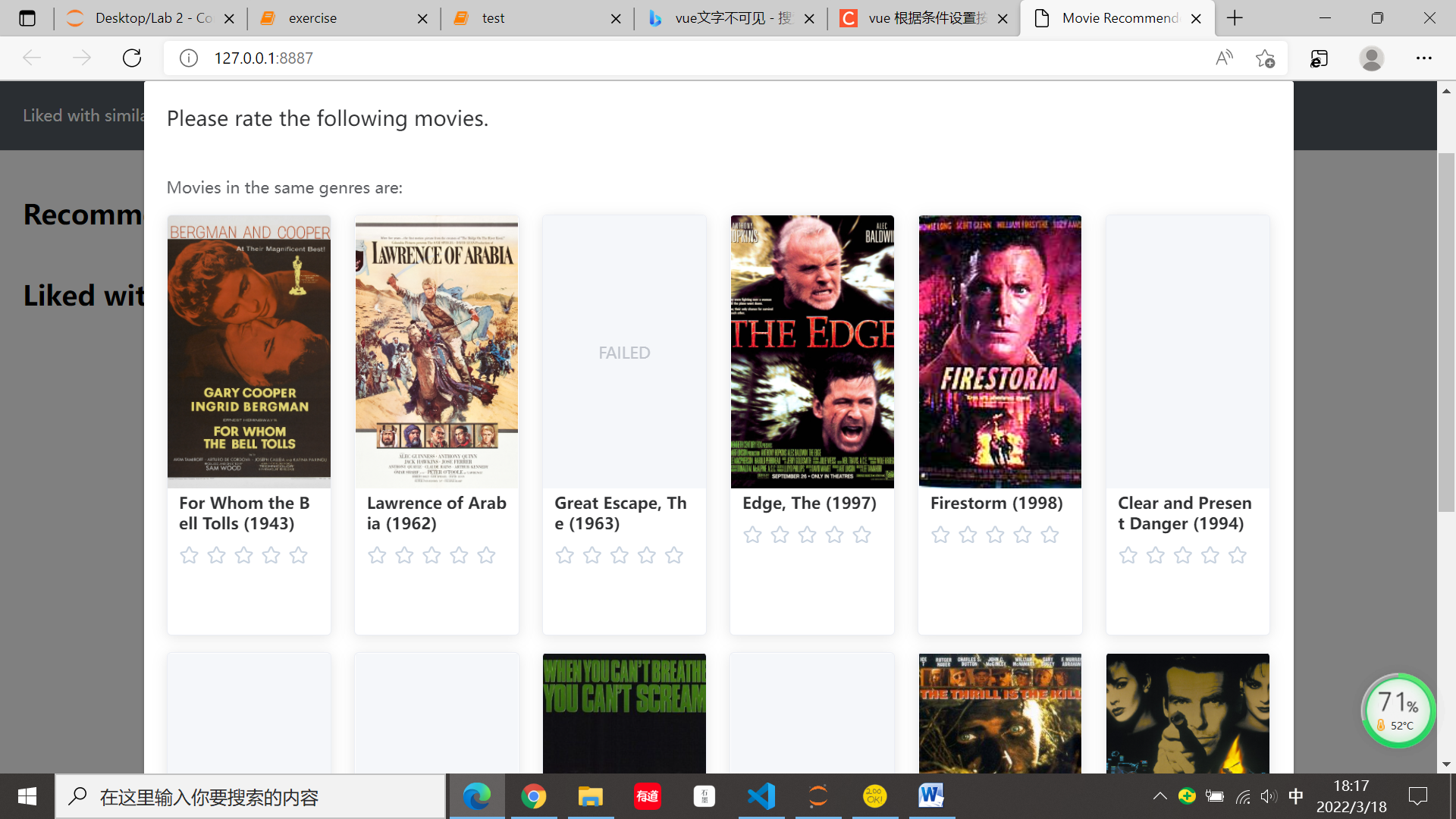
对应index.html的dialog0，和main.py的# show top-10 genres(@app.get("/api/genre"))



2. 根据勾选的genres，基于**content**显示cos相似度最高的12部电影

对应index.html的dialog1和main.py的# top-12 content-based similar movies(@app.post("/api/movies"))和#for dialog 0 & 1(def initial\_content\_based\_rec)

评分结果保存到this.movies

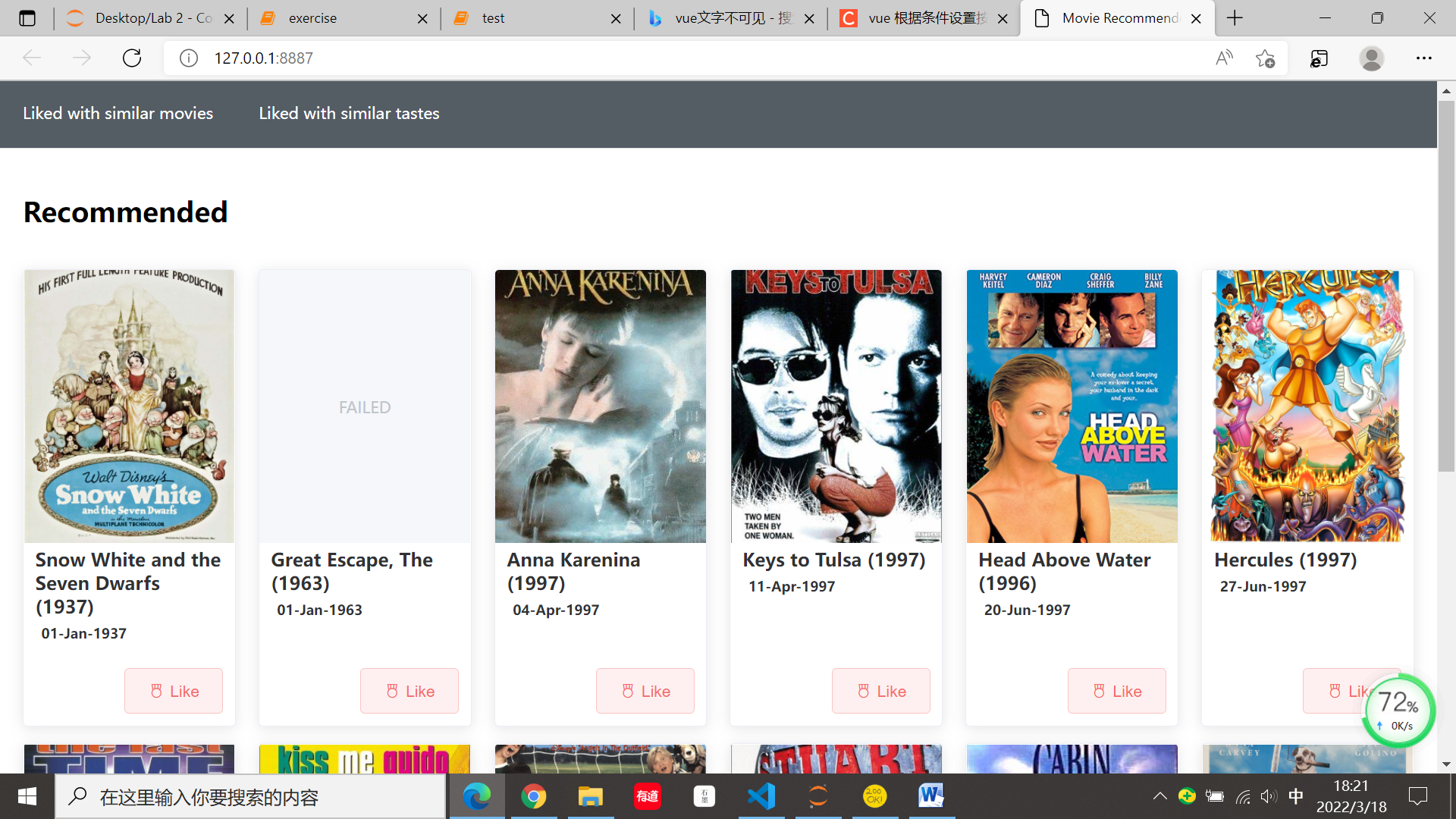


3. 主页面默认展示item-based recommendation结果（对应<!-- item-based recommendation -->）和explanation

结果保存到this.item\_recommended和this.item\_liked

调用item\_rec获取推荐结果（@app.post("/api/recommend")）

调用item\_liked\_btn添加解释和更新推荐（@app.get("/api/add\_recommend/{item\_id}")）

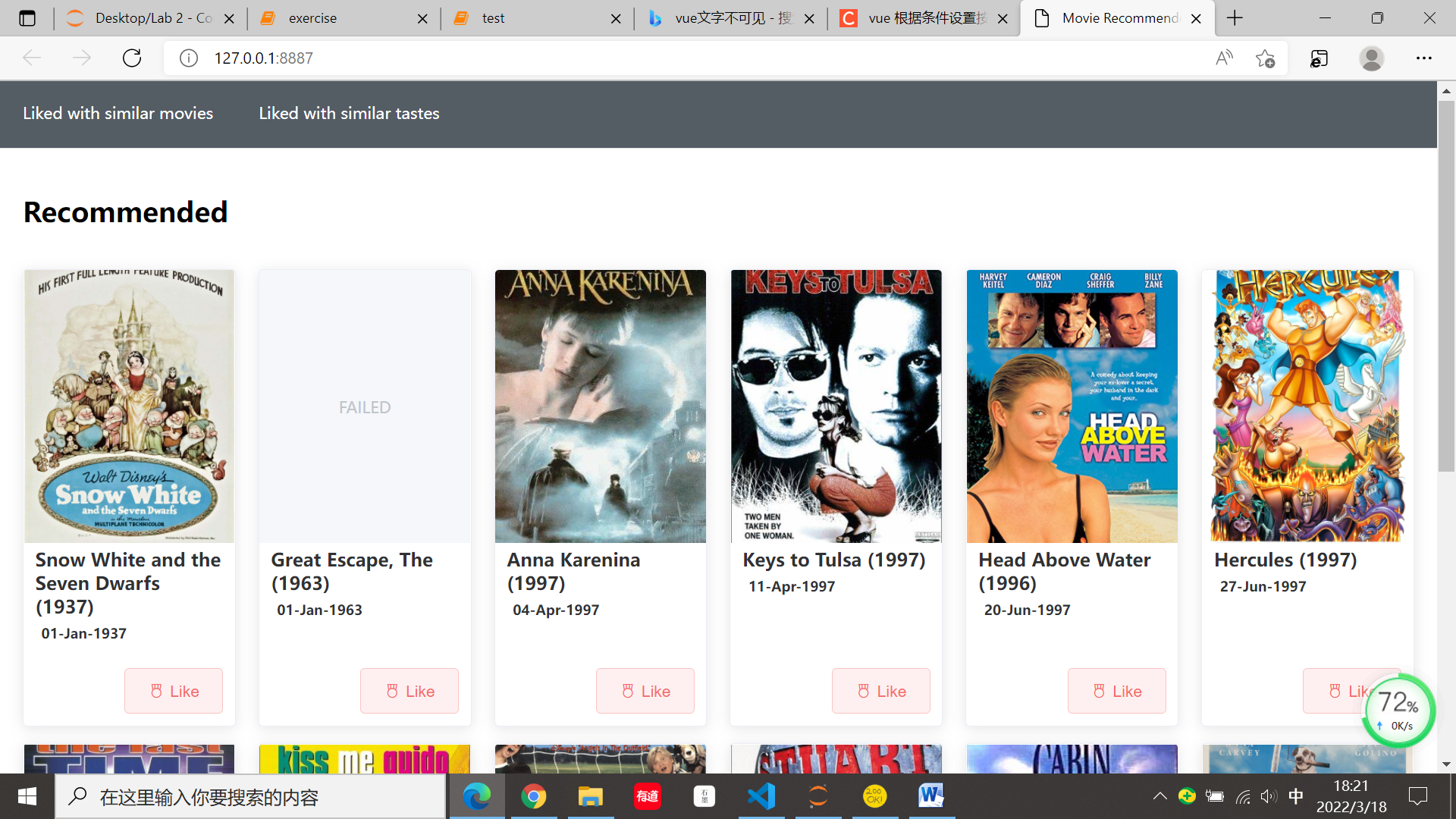


4. 点击liked with similar tastes，显示user-based recommendation结果（对应<!-- user-based recommendation -->）和explanation

结果保存到this.user\_recommended和this.user\_liked

调用user\_rec获取推荐结果（@app.post("/api/user\_recommend")）

调用user\_liked\_btn添加解释和更新推荐（@app.get("/api/add\_user\_recommend/{item\_id}")）



**注：目前item-based用的是demo里定义的函数，user-based没写，只建立了handler**