**后端代码说明**

**1. API 实现及 WebSocket 实时推送逻辑**

* GET /api/poll

查询数据库，返回当前投票问题和所有选项的最新票数。

* POST /api/poll/vote

接收客户端传来的 option\_id，更新数据库中对应选项的票数。

更新成功后，触发 WebSocket 广播，通知所有已连接客户端最新投票数据。

* WebSocket /ws/poll

负责实时向客户端推送当前投票问题和所有选项的票数。

新客户端连接时，立即发送当前最新数据。

任何投票更新都会通过广播推送最新数据给所有客户端。

**2. 数据存储与票数更新逻辑**

* 使用 SQLite 数据库存储投票问题和选项数据。
* 数据库表：

poll 表：存储投票问题（id, question）

option 表：存储投票选项（id, poll\_id, text, votes）

* 读取投票数据时，查询投票问题和选项及对应票数。
* 投票时，数据库对应选项的票数字段 votes 加 1，并保存。