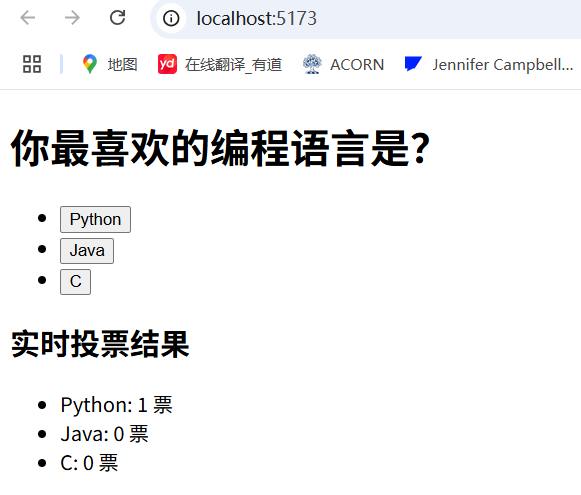
**前端代码说明**

**页面展示投票问卷与结果图表**



**投票交互与实时结果刷新逻辑**

1. 页面加载初始化数据

在组件mounted中使用 axios 向后端 API (/api/poll) 发送 GET 请求

获取当前投票问题和选项数据，赋值给 question和 options，用于渲染页面

调用 connectWebSocket建立 WebSocket 连接，用于接收服务器推送的实时投票结果更新

2. 投票交互

用户点击某个投票按钮时，触发 vote(optionId)方法

vote方法通过axios.post 向后端发送投票请求（携带选项 ID）

用户可以反复投票，方便多次互动

投票请求发送成功后，前端不直接修改投票数据，而是等待服务器通过 WebSocket 推送最新数据

3. WebSocket 实时结果刷新

connectWebSocket方法创建一个新的 WebSocket 连接，连接到后端的/ws/poll端点

连接建立后，服务器会立即推送数据库最新状态

每当后端接收到新的投票时，会广播最新的投票状态给所有连接的客户端

WebSocket 客户端通过 onmessage事件接收消息，将接收到的 JSON 数据解析后更新本地的question和options数据

Vue 的响应式系统会自动刷新视图，实时显示最新的投票结果

4. 连接断开及重连

当 WebSocket 连接意外关闭时，触发onclose事件

前端会等待 3 秒后自动尝试重新建立 WebSocket 连接，保证投票结果能够持续实时更新

5. 组件卸载时关闭连接

在 Vue 组件的 beforeUnmount中关闭 WebSocket 连接，释放资源