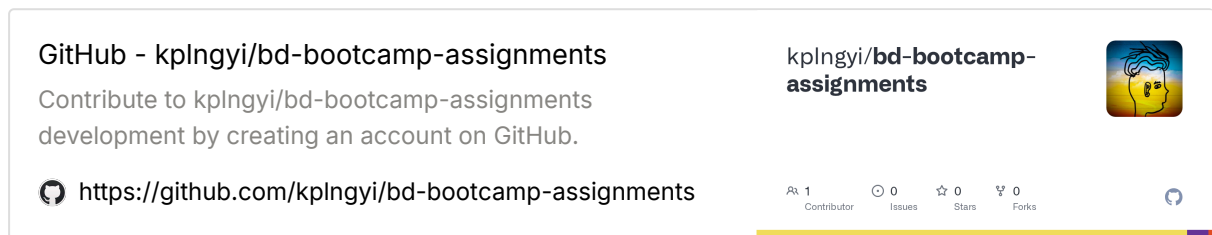


🚀 字节工程训练营项目总结 (Engineering Bootcamp Projects)



1. 电商商品列表页开发 (eComm List Page)
2. 实时投票系统 (Real-Time Voting System)

📁 作业一：电商商品列表页开发

🎯 概览 (Overview)

本项目旨在开发一个功能完整的电商商品列表页，重点实现高效的数据交互、状态管理和优良的用户体验。页面涵盖了电商核心的筛选、排序和分页功能。

⚙️ 技术栈 (Tech Stack)

- 前端框架：[React]
- 状态管理：[Redux Toolkit]
- 组件库：[Ant Design]
- 网络请求：WebSocket

✨ 核心功能 (Core Features)

- 商品展示：以卡片形式清晰展示商品信息。
- 多维筛选：支持按价格区间、商品分类、品牌等条件进行筛选。
- 动态排序：支持按销量、价格、上架时间等字段进行升序/降序排列。

- **分页加载**：实现高效的页码式数据加载和切换。
- **统一状态管理**：使用 [RTK] 集中管理筛选条件和商品列表数据。

加分项任务清单 (Bonus To-Do List)

以下是本项目实现的加分项和完成状态：

状态	任务描述	细节与实现方式
	对筛选功能进行 防抖 (Debounce) 处理。	针对价格 输入等筛选条件，使用 <code>lodash.debounce</code> 或自定义 Hook 避免短时间内频繁请求 API。
	使用 骨架屏 (Skeleton) 提升加载体验。	在列表数据请求过程中，使用组件库提供的 `` 组件替代实际内容。
	应用 虚拟滚动 (Virtual Scrolling) 优化大数据量列表性能。	采用 <code>react-window</code> 库，仅渲染视口内可见的商品项。
	在列表顶部增加 "猜你喜欢" 模块。	模拟 AI 推荐逻辑，根据当前筛选条件或随机逻辑展示推荐商品。

作业二：实时投票系统

概览 (Overview)

本项目旨在构建一个高响应、数据同步迅速的实时投票系统。用户投票后，数据排名和总票数能即时更新，无需手动刷新。

技术栈 (Tech Stack)

- **前端框架**：[React / Vue] (请根据您的选择填写)
- **实时通信**：WebSocket (原生或 [Socket.io](#))
- **后端服务**：Node.js / Go / Python (提供 WebSocket 服务和数据持久化)
- **图表库**：ECharts / AntV G2 (用于可视化大屏)

核心功能 (Core Features)

- **多选项投票**：用户可以选择多个选项中的一个进行投票。
- **实时数据同步**：基于 WebSocket，投票数据在所有连接客户端间毫秒级同步。
- **动态排名展示**：根据票数实时更新选项的排名和百分比。

- **防止重复投票**：基础的用户或 IP 限制机制。

💎 加分项任务清单 (Bonus To-Do List)

以下是本项目实现的加分项和完成状态：

状态	任务描述	细节与实现方式
✓	实现 高并发限流算法 (令牌桶/漏桶)。	后端实现 ：在投票 API 接口应用限流中间件，支持 1000+ 并发请求的稳定处理。
✓	WebSocket 消息批量合并 优化性能。	前端或后端 ：对短时间内的票数更新消息进行合并 (Batching)，减少推送频率和渲染次数。
✓	使用 Web Worker 处理数据计算。	将复杂的排名计算或大数据统计逻辑从浏览器主线程分离，确保 UI 渲染流畅。
✓	实现 异常投票检测 功能。	后端逻辑 ：检测异常的投票模式（如僵尸投票、机器刷票）并进行标记或过滤。
🔄	创新功能：开发 投票数据可视化大屏 。	使用 ECharts/G2 等库，展示投票趋势图、地域分布、用户活跃度等数据仪表盘。

🔗 运行和部署 (Running and Deployment)

请按照以下步骤运行任一项目：

1. 克隆本仓库：

```
git clone <https://github.com/kplngyi/bd-bootcamp-assignments>
```

2. 进入任一项目文件夹：

```
cd assignment-1-ecomm-list-page
# 或
cd assignment-2-realtime-voting-system
```

3. 通过项目文件README.md运行项目