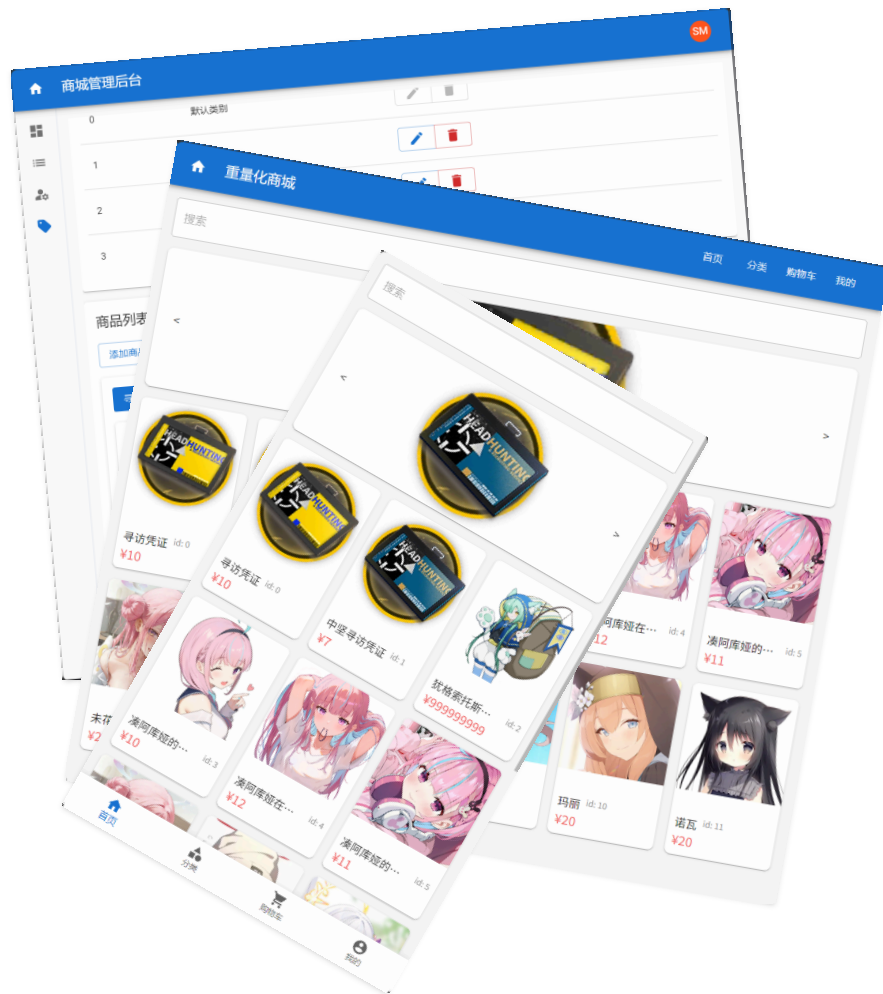


# 重量化商城系统









肖斌 俞贤皓 付家齐 杨鹏 谷雅丰











# 目录

1. 重量化商城系统
2. 目录
3. 所用技术
4. Solidjs 与 React
5. 实现功能与分工
6. 演示
7. 完结撒花

# 所用技术


-  NodeJS +  pnpm
-  React +  Mui
-  TailwindCSS
-  Java +  Spring +  MySQL

# 所用技术

-  NodeJS +  pnpm
-  React +  Mui
-  TailwindCSS
-  Java +  Spring +  MySQL

答辩不用

# 所用技术

-  NodeJS +  pnpm
-  React +  Mui
-  TailwindCSS
-  Golang +  Gin + GORM +  PostgreSQL

优雅

# 所用技术

-  NodeJS +  pnpm
-  React +  Mui
-  TailwindCSS
-  Golang +  Gin + GORM +  PostgreSQL





# 不用技术

-  NodeJS +  pnpm
-  React +  Mui
-  TailwindCSS
-  Golang +  Gin + GORM +  PostgreSQL

## 不用技术

-  NodeJS +  pnpm
  -  React +  Mui
  -  TailwindCSS
  -  Golang +  Gin +  GORM +  PostgreSQL
- 那就真重量化了

## 所用技术

-  bun
-  SolidJS +  SUID (Mui in SoliJS)
-  unocss
- 简易 key-value 存储后端: express + lowdb
- 前后端交互: axios



Slidev

生命的价值在于从无意义中「建构」意义



# Solidjs 与 React

```
1 // Solid
2 const Hello: Component = () => {
3   const [count, setCount] = createSignal(0);
4
5   return (
6     <button onClick={() => setCount(count => count + 1)}>
7       Clicked {count()} times
8     </button>
9   );
10 }
```

条件渲染、列表渲染、Context、ref...

但是！没有虚拟 DOM，使用 fine-grained reactivity。

---

# 实现功能与分工

- 肖斌
  - Github 仓库维护 (Issue、分支、PR 审查)
  - 技术选择
  - 数据模型、db、api 设计与实现
  - 后台实现
  - 整体样式调整 & 修复疑难杂症
- 俞贤皓
  - 主页、商品详情页、分类页
  - 商品组件、商品组选择
  - 后端实现
- 谷雅丰
  - 浏览商品添加历史记录逻辑
  - 历史记录页、前台登录组件
- 付家齐
  - 提交订单 -> 支付 -> 支付成功组件
  - 订单列表页、订单详情页
  - 商品页提交订单逻辑
  - 测试仙人
- 杨鹏
  - 商品页添加至购物车、购物车页结算逻辑
  - 购物车页面
  - 我的页面

演示

# 完结撒花

GitHub