

# Student\_Society

第二次作业要求（可以删除）：

去中心化学生社团组织治理应用

- 每个学生初始可以拥有或领取一些通证积分（ERC20）。
- 每个学生可以在应用中可以：
  1. 使用一定数量通证积分，发起关于该社团进行活动或制定规则的提案（Proposal）。
  2. 提案发起后一定支出时间内，使用一定数量通证积分可以对提案进行投票（赞成或反对，限制投票次数），投票行为被记录到区块链上。
  3. 提案投票时间截止后，赞成数大于反对数的提案通过，提案发起者作为贡献者可以领取一定的积分奖励。
- (Bonus) 发起提案并通过3次的学生，可以领取社团颁发的纪念品（ERC721）

以下内容作为作业仓库的README.md中需要描述的内容。请根据自己的需要进行修改并提交。

作业提交方式为：提交视频文件和仓库的连接到指定邮箱。

## 如何运行

1. 在本地启动ganache应用。
2. 在 `./contracts` 中安装需要的依赖，运行如下的命令：

```
npm install
```

3. 在 `./contracts` 中编译合约，运行如下的命令：

```
npx hardhat compile
```

4. 在 `./contracts/hardhat.config.ts` 中将ganache的url改为本地ganache网络的url地址，将ganache中账户的私钥写在 `accounts` 下
5. 在 `./contracts` 中部署合约，执行如下命令：

```
npx hardhat run ./scripts/deploy.ts --network ganache
```

6. 将合约编译好生成的 `abi` 文件复制到 `./frontend/src/utis/abis` 下，  
即 `./contracts/artifacts/contracts` 中的 `MyERC20.json` 和 `StudentSocietyDAO.json`
7. 将 `./frontend/src/pages/society/index.tsx` 中的ganache链ID、名称和rpc url改为本地ganache中的配置
8. 在 `./frontend` 中启动前端程序，运行如下的命令：

```
npm run start
```

## 功能实现分析

1. 每个学生初始可以拥有或领取一些通证积分（ERC20）

使用ERC20中的 `_mint` 函数实现某个账户领取通证积分的功能

2.使用一定数量通证积分，发起关于该社团进行活动或制定规则的提案（Proposal）

利用ERC20中的 `transferFrom` 函数，将一定的积分转给合约，成功后将创建新的提案

3.提案发起后一定支出时间内，使用一定数量通证积分可以对提案进行投票（赞成或反对，限制投票次数），投票行为被记录到区块链上。

利用ERC20中的 `transferFrom` 函数，将一定的积分转给合约，完成后提案的赞成/反对票增加

4.提案投票时间截止后，赞成数大于反对数的提案通过，提案发起者作为贡献者可以领取一定的积分奖励。

提案投票时间截止后，合约检测其通过状态，并利用ERC20中的 `transfer` 函数从合约转给提案发起者一定数量的通证积分

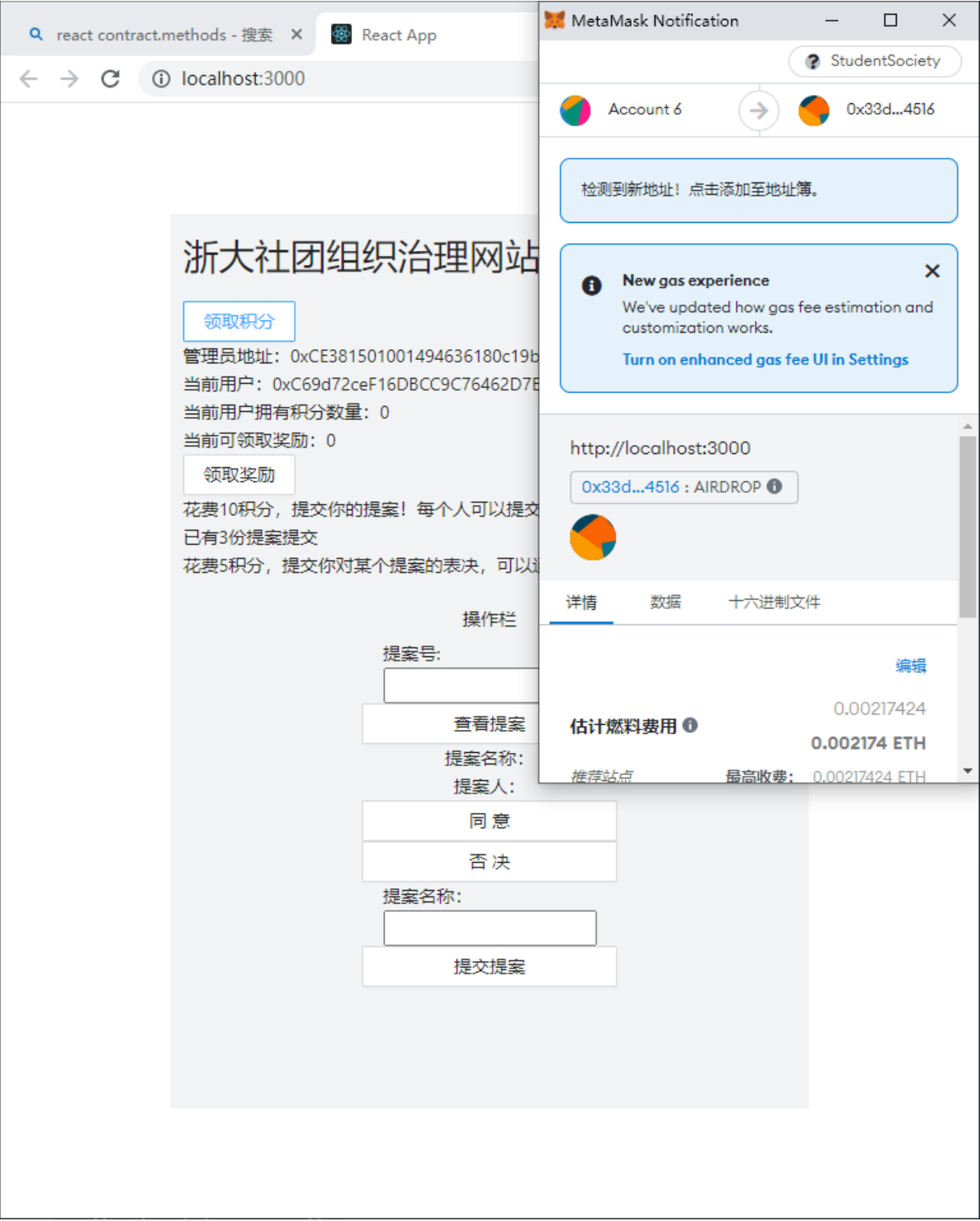
## 项目运行截图

---

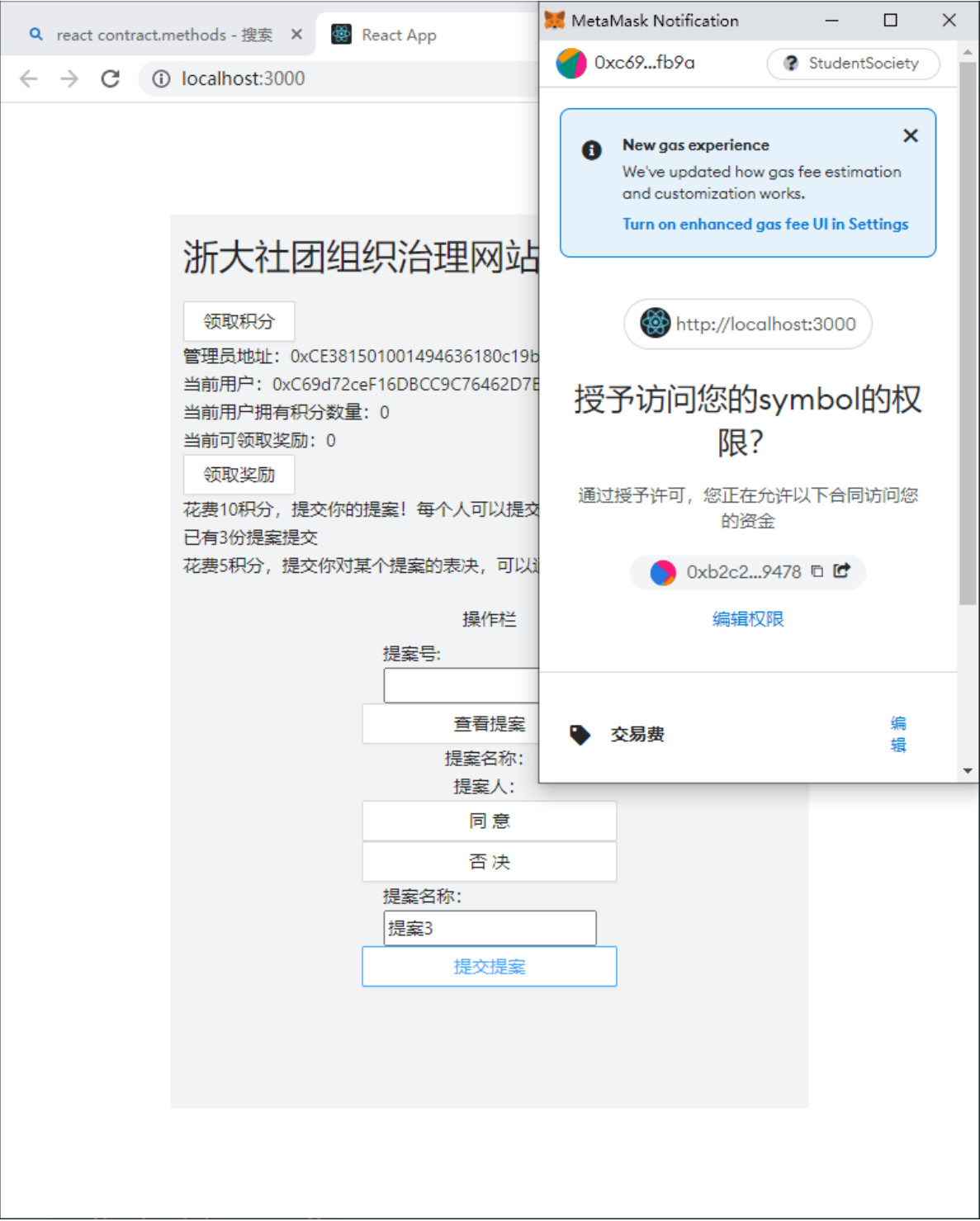
放一些项目运行截图。

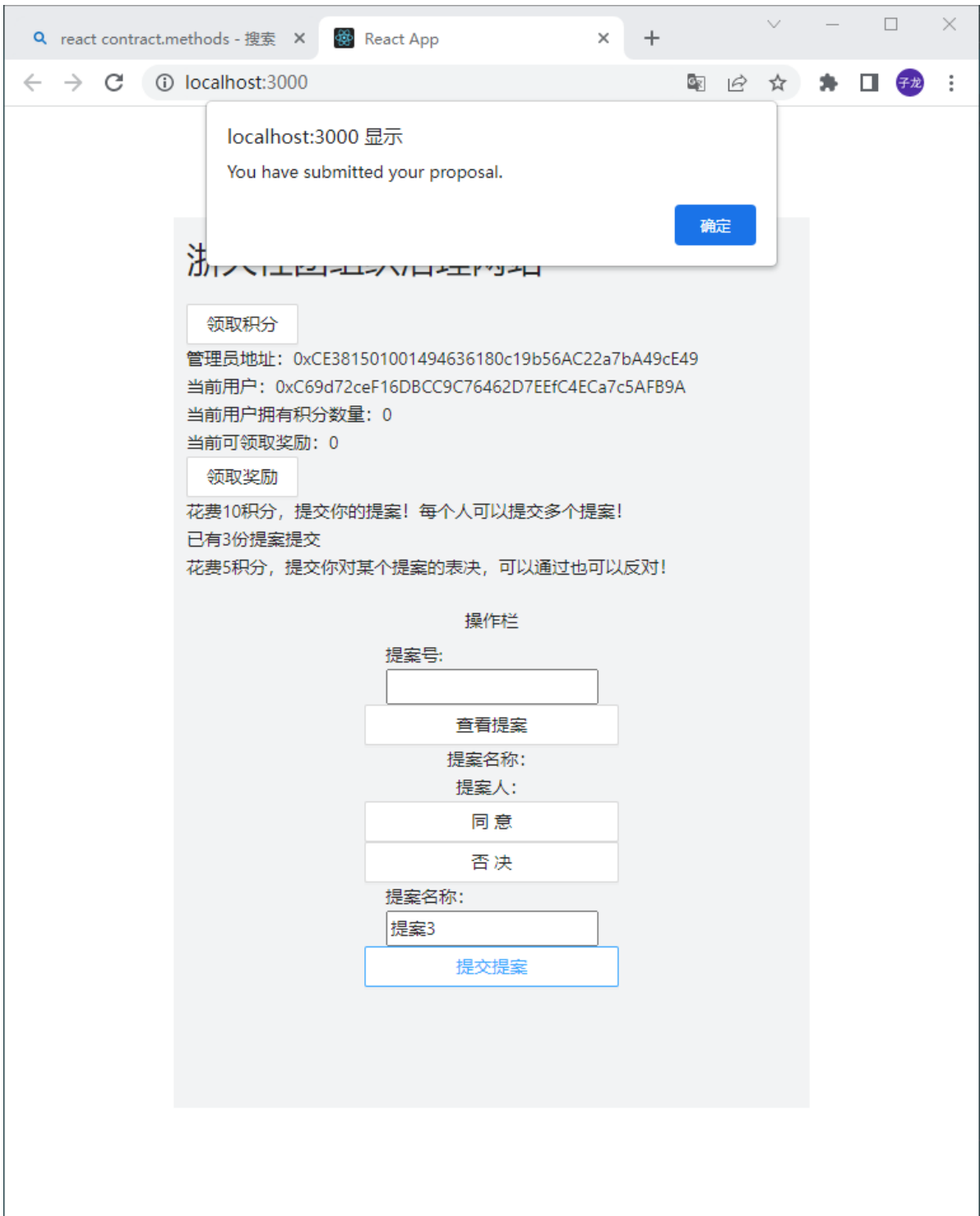
项目运行成功的关键页面和流程截图。主要包括操作流程以及和区块链交互的截图。

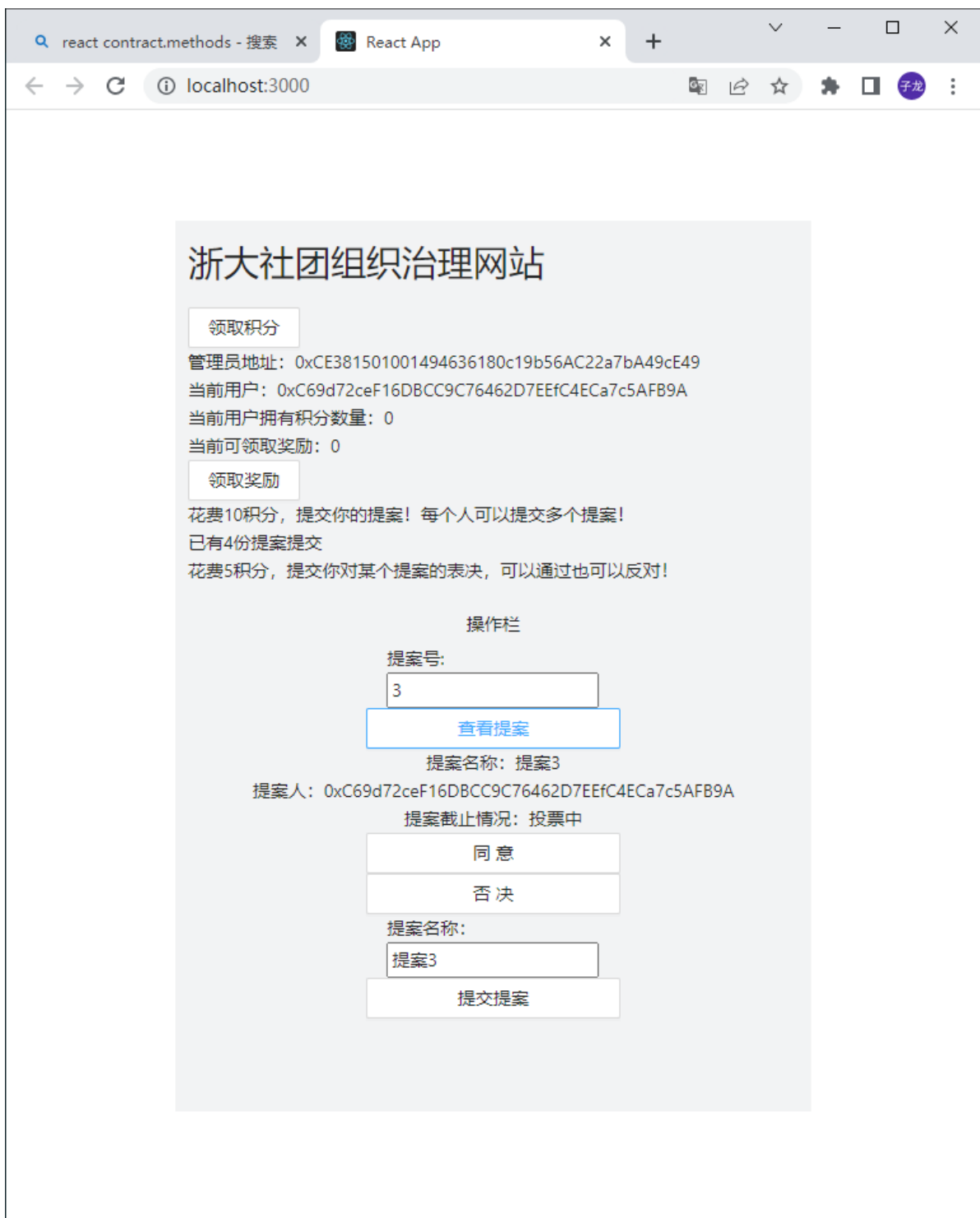




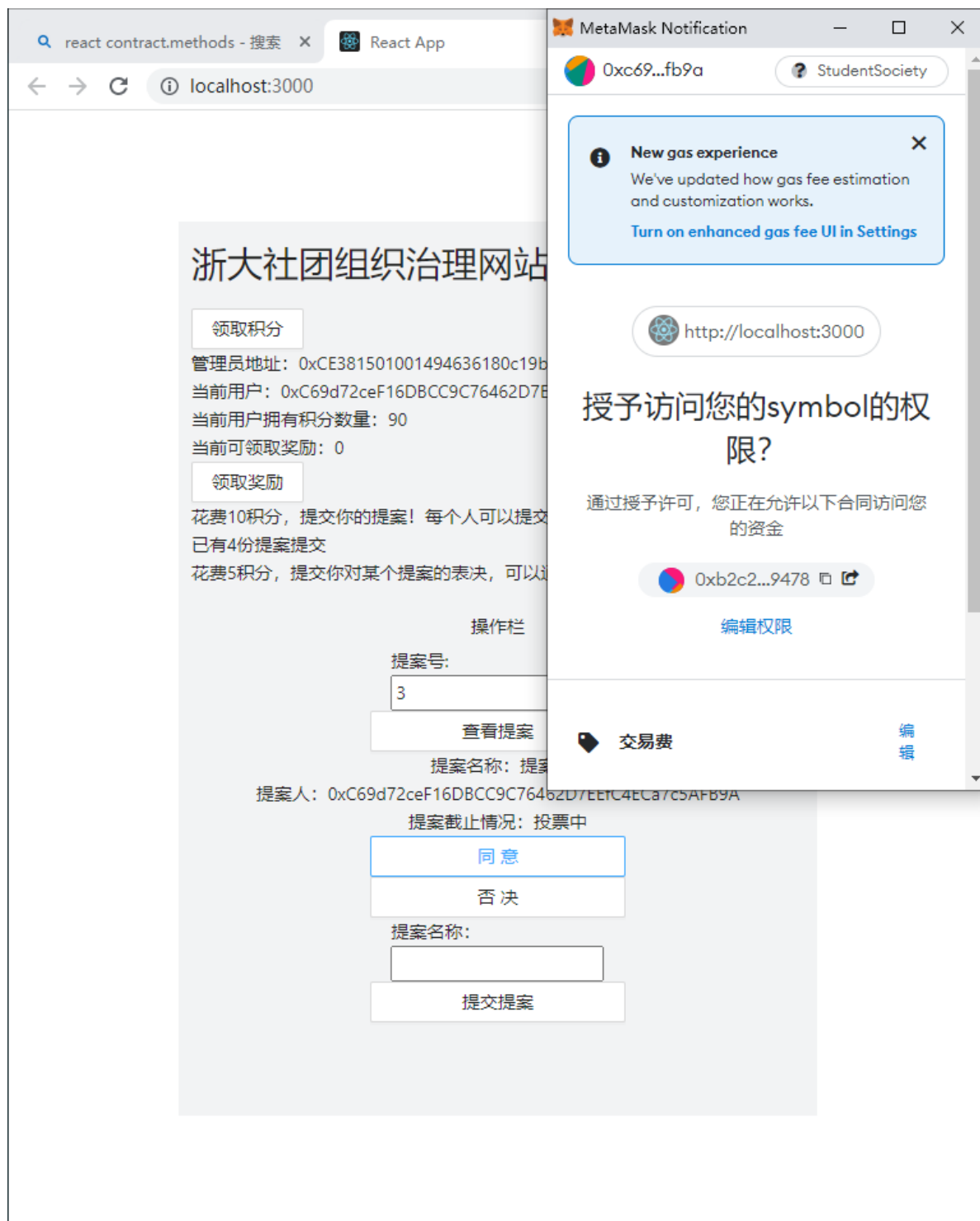


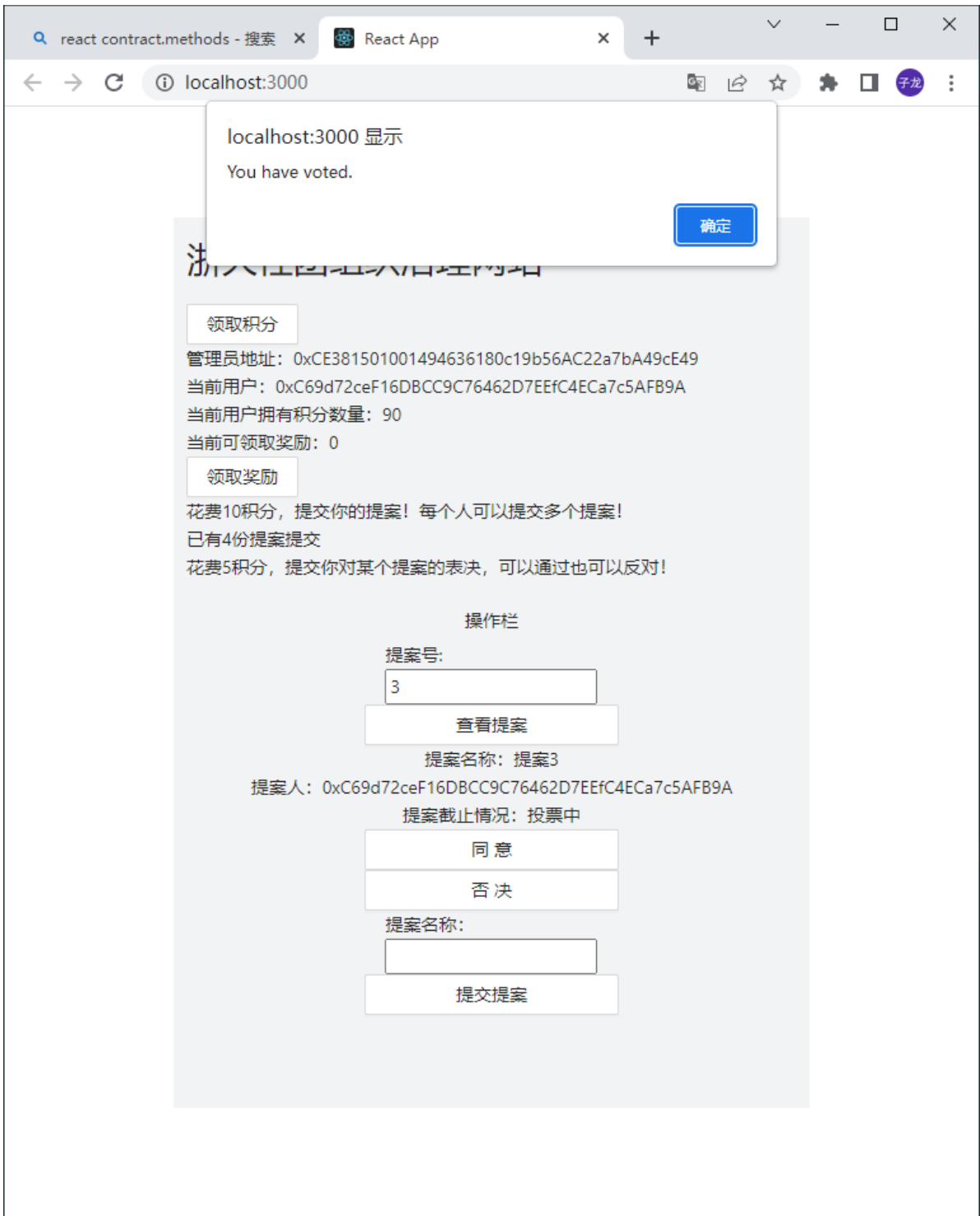




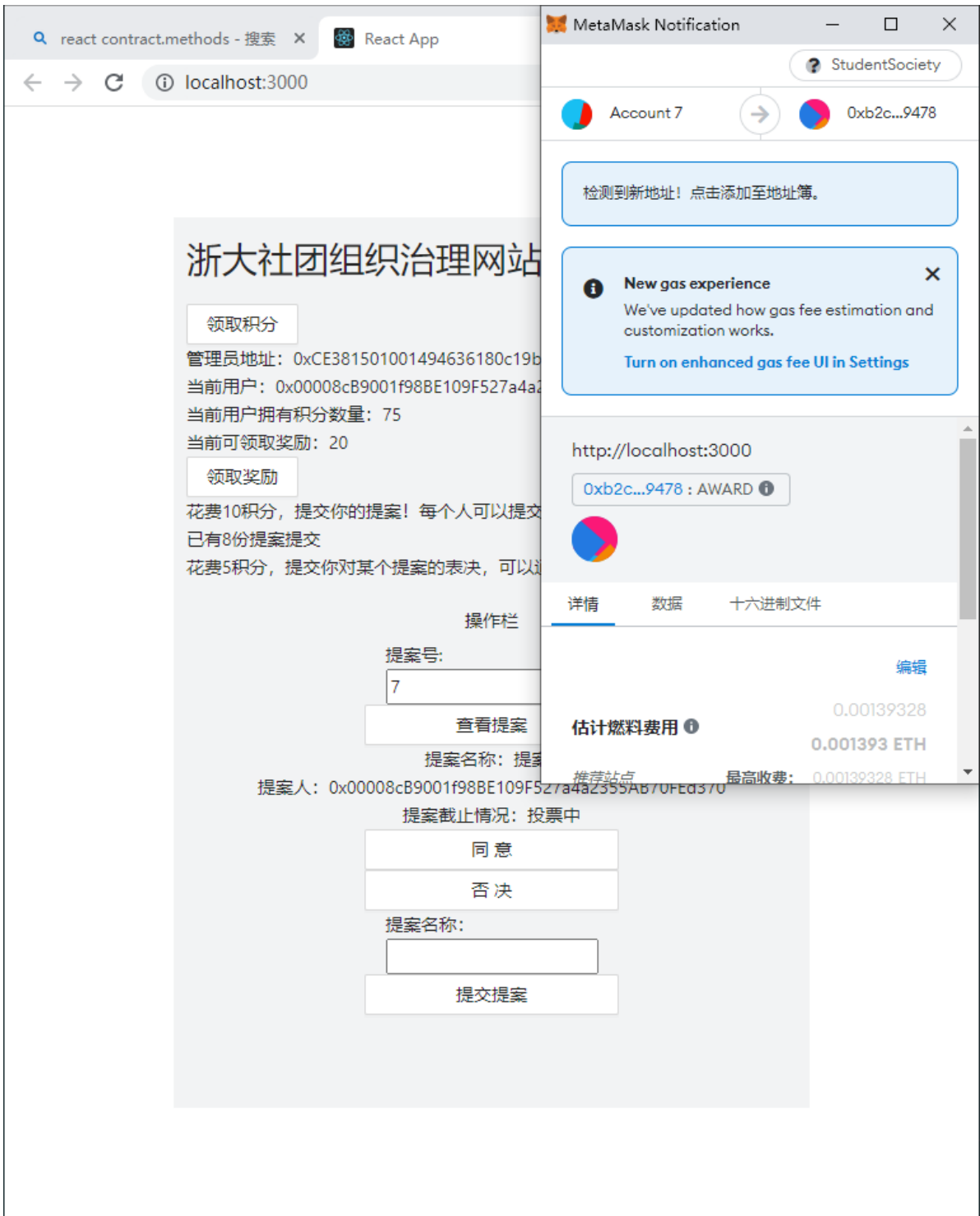


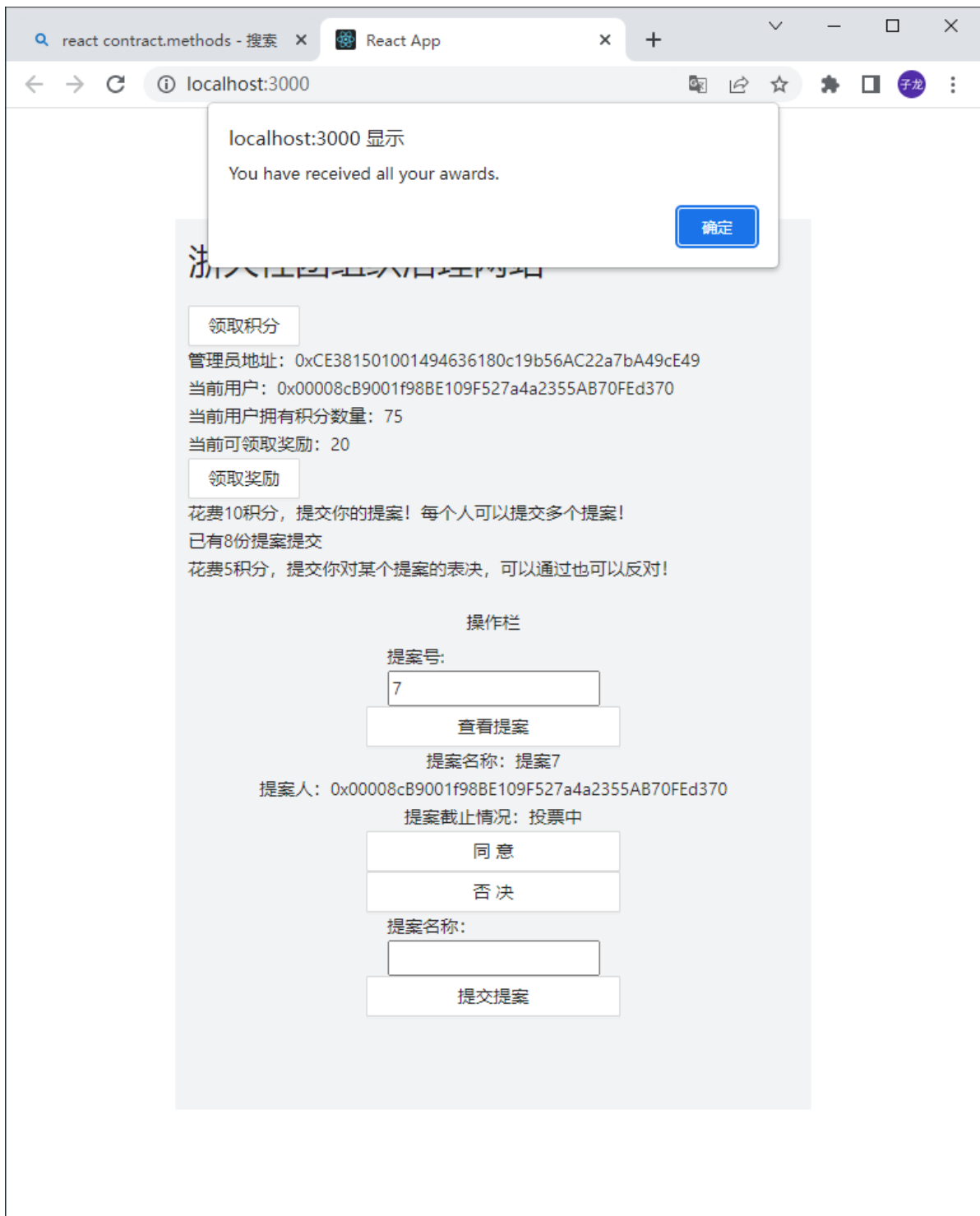












## 参考内容

课程的参考Demo见: [DEMOS](#)。