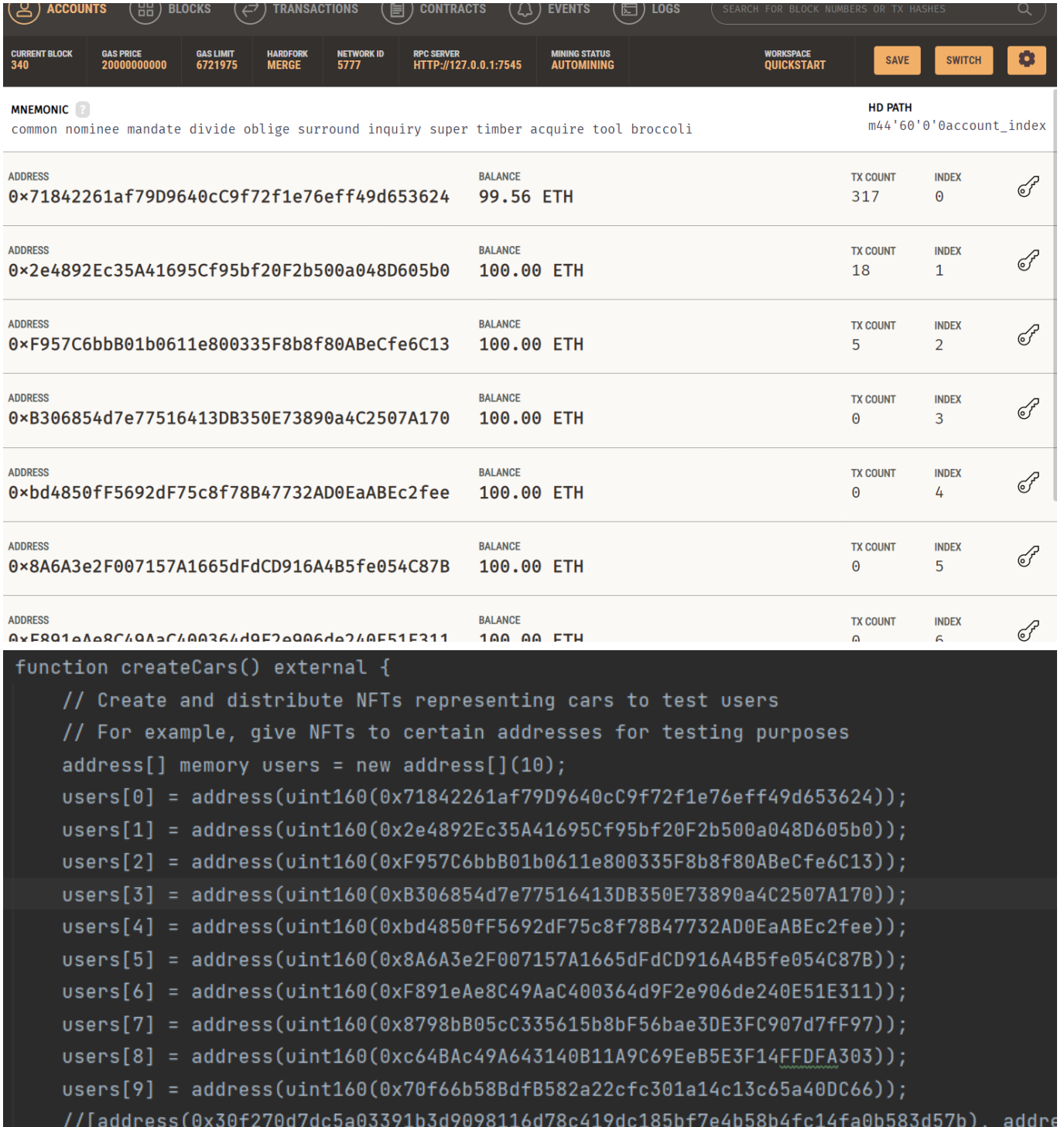


1.如何运行

1.在本地启动ganache应用 2.在 ./contracts 中安装依赖，运行如下命令：

```
npm install
```

3.在ganache中复制自己的账户地址，在./contracts/contracts/BorrowYourCar.sol中的createCars函数中和，粘贴自己的账户地址



The screenshot shows the Ganache application interface. At the top, there are tabs for ACCOUNTS, BLOCKS, TRANSACTIONS, CONTRACTS, EVENTS, and LOGS. Below the tabs, there's a search bar and several status indicators: CURRENT BLOCK (340), GAS PRICE (2000000000), GAS LIMIT (6721975), HARDFORK (MERGE), NETWORK ID (5777), RPC SERVER (HTTP://127.0.0.1:7545), MINING STATUS (AUTOMINING), and WORKSPACE (QUICKSTART). There are also buttons for SAVE, SWITCH, and a settings icon.

Below the status bar, there's a section for MNEMONIC and HD PATH. The MNEMONIC is "common nominee mandate divide oblige surround inquiry super timber acquire tool broccoli" and the HD PATH is "m44'60'0'0account_index".

The main part of the interface is a table listing accounts. Each row shows the ADDRESS, BALANCE, TX COUNT, and INDEX. The accounts are:

ADDRESS	BALANCE	TX COUNT	INDEX
0x71842261af79D9640cC9f72f1e76eff49d653624	99.56 ETH	317	0
0x2e4892Ec35A41695Cf95bf20F2b500a048D605b0	100.00 ETH	18	1
0xF957C6bbB01b0611e800335F8b8f80ABeCfe6C13	100.00 ETH	5	2
0xB306854d7e77516413DB350E73890a4C2507A170	100.00 ETH	0	3
0xbd4850fF5692dF75c8f78B47732AD0EaABEc2fee	100.00 ETH	0	4
0x8A6A3e2F007157A1665dFdCD916A4B5fe054C87B	100.00 ETH	0	5
0xF891eAe8C49AaC400364d9F2e906de240E51E311	100.00 ETH	0	6

Below the table, there's a code editor showing the createCars function in BorrowYourCar.sol:

```
function createCars() external {
    // Create and distribute NFTs representing cars to test users
    // For example, give NFTs to certain addresses for testing purposes
    address[] memory users = new address[](10);
    users[0] = address(uint160(0x71842261af79D9640cC9f72f1e76eff49d653624));
    users[1] = address(uint160(0x2e4892Ec35A41695Cf95bf20F2b500a048D605b0));
    users[2] = address(uint160(0xF957C6bbB01b0611e800335F8b8f80ABeCfe6C13));
    users[3] = address(uint160(0xB306854d7e77516413DB350E73890a4C2507A170));
    users[4] = address(uint160(0xbd4850fF5692dF75c8f78B47732AD0EaABEc2fee));
    users[5] = address(uint160(0x8A6A3e2F007157A1665dFdCD916A4B5fe054C87B));
    users[6] = address(uint160(0xF891eAe8C49AaC400364d9F2e906de240E51E311));
    users[7] = address(uint160(0x8798bB05cC335615b8bF56bae3DE3FC907d7fF97));
    users[8] = address(uint160(0xc64BAc49A643140B11A9C69EeB5E3F14FFDFA303));
    users[9] = address(uint160(0x70f66b58BdfB582a22cfc301a14c13c65a40DC66));
    // [address(0x30f270d7dc5a03391b3d9098116d78c419dc185bf7e4b58b4fc14fa0b583d57b), address(0x30f270d7dc5a03391b3d9098116d78c419dc185bf7e4b58b4fc14fa0b583d57b), address(0x30f270d7dc5a03391b3d9098116d78c419dc185bf7e4b58b4fc14fa0b583d57b), address(0x30f270d7dc5a03391b3d9098116d78c419dc185bf7e4b58b4fc14fa0b583d57b), address(0x30f270d7dc5a03391b3d9098116d78c419dc185bf7e4b58b4fc14fa0b583d57b), address(0x30f270d7dc5a03391b3d9098116d78c419dc185bf7e4b58b4fc14fa0b583d57b), address(0x30f270d7dc5a03391b3d9098116d78c419dc185bf7e4b58b4fc14fa0b583d57b), address(0x30f270d7dc5a03391b3d9098116d78c419dc185bf7e4b58b4fc14fa0b583d57b), address(0x30f270d7dc5a03391b3d9098116d78c419dc185bf7e4b58b4fc14fa0b583d57b), address(0x30f270d7dc5a03391b3d9098116d78c419dc185bf7e4b58b4fc14fa0b583d57b)]
}
```

4.在ganache中复制账户密钥，粘贴到.\contracts\hardhat.config.ts中：

```
url: http://127.0.0.1:7545 ,
// the private key of signers, change it according to your ganache user
accounts: [
  '0xbe941220748bdaa97dfd4f3838f326b04a110cda2d08d07646aded6bd787f15e',
  '0xff3a08282582b0c760476ddef837f3aa53a7a5010b4cd8f4e996e81644df17d1',
  '0xffc1a02b9952db5d0e1d7717b948f4cfc6948e24b7864e2facfa32ae35659f83',
  '0xd109dd3d194d9d2dde8a780663d941f66f217c140dcdf53f0b632854faa62d45',
  '0xc661ea86d0323c970c7d6dc459afacbff62fb6b0fe3dd89f70268972f45ae851',
  '0x4ea5fd8c2f092518ec5fc1f4155714a5d1c6a55a1c39d8afb3a2bd2f3fcd5c5d',
  '0x95275c9a371eb87eb7daf1b03701611292c7d5a07f764628a2f7ddab0425110f',
  '0x7fe17b2728daf646504a6ba87940f347592d618c5977a137a1c7178ac9ec2d90',
  '0xdf98e826582f001f24d9a1f9f6a3394ecbe8ce3defd3296c05342f1d6593e6a0',
  '0x34365603b96420772b6505ccb5e47fb19fa8c19c8a2e74e6db2aa57c6ad93b0'
]
```

5.在 ./contracts 中编译合约，运行如下命令：

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

6.复制打印出来的合约地址到 ./fronted/src/utils/contract-addresses.json 中

```
src > utils > {} contract-addresses.json > ...
1   {
2     "BorrowYourCar": "0xA052AF9BEE5A3331ee980C87F9e26a3fFC37A4ed",
3     "myERC20": "0xb43dd352B4Fc93ccac402eab6F2feBb4EAB51a10"
4   }
```

7.在 ./fronted 中安装依赖，运行如下命令：

```
npm install
```

8.在 ./ fronted 中启动前端程序，运行如下命令：

```
npm run dev
```

2.功能实现分析

2.1查看自己拥有的汽车列表

在合约中实现了getOwnedCar函数：

```
function getOwnedCars(address owner) external view returns (uint256[] memory) {
    // Get the list of car token IDs owned by a specific user
    uint256 counter = 0;
    for (uint256 i = 0; i < totalCars; i++) {
```

```
        if (ownerOf(i) == owner) {

            counter++;

        }
    }
    uint256[] memory ownedCars=new uint256[](counter) ;
    counter=0;
    for (uint256 i = 0; i < totalCars; i++) {
        if (ownerOf(i) == owner) {
            ownedCars[counter] = i;
            counter++;
        }
    }
    return ownedCars;
}
```

在前端中通过按钮触发并且调用它，渲染数据到listOfResult中：

连接钱包

当前用户: 0x71842261af79d9640cc9f72f1e76eff49d653624

当前用户积分: 0

刷新并显示未借用车辆

刷新并显示我已拥有的车辆

刷新并显示我已租借的车辆

刷新并显示所有车辆

查询

发行汽车

刷新

获得空投积分

汽车ID	汽车拥有者	汽车借用者	是否被借用	归还时间	操作
0	0x71842261af79d9640cc9f72f1e76eff49d653624	无	未被借用	∅无	借用
2	0x71842261af79d9640cc9f72f1e76eff49d653624	无	未被借用	∅无	借用
7	0x71842261af79d9640cc9f72f1e76eff49d653624	无	未被借用	∅无	借用
14	0x71842261af79d9640cc9f72f1e76eff49d653624	无	未被借用	∅无	借用

```
async showOwnedVehicles() {
    // 处理显示自己已拥有车辆的逻辑
    const tempList = Object.assign([], await BorrowYourCarContract.methods.getOwnedCars(this.currentUser).call())
    // console.log(tempList3)
    this.listOfMyOwnedCars = this.getCarDetailsArray(tempList)
    // console.log(this.listOfMyOwnedCars)
    this.listData = Object.assign([], this.listOfMyOwnedCars)
    // console.log(this.listData)
    Promise.all(this.listData)
        .then((result) => {
            // 将结果存储在一个同步数组中
            this.listOfResult = result
            console.log(this.listOfResult) // 输出同步数组
        })
        .catch((error) => {
            console.error(error) // 处理错误情况
        })
},
```

2.2查看自己已经借用的车辆

在合约中实现了getMyBorrowedCars函数：

```
function getMyBorrowedCars(address borrower1) external view returns (uint256[]
memory) {

    uint256 count = 0;
```

```

    for (uint256 i = 0; i < totalCars; i++) {
        if (cars[i].borrower == borrower1 && cars[i].isBorrowed) {
            count++;
        }
    }
    uint256[] memory MyBorrowedCars = new uint256[](count);
    count = 0;
    for (uint256 i = 0; i < totalCars; i++) {
        if (cars[i].borrower == borrower1 && cars[i].isBorrowed) {
            MyBorrowedCars[count] = i;
            count++;
        }
    }
    return MyBorrowedCars;
}

```

在前端中通过按钮触发并且调用它，渲染数据到listOfResult中：

Dashboard

连接钱包

当前用户: 0x71842261af79d9640cc9f72f1e76eff49d653624

当前用户积分: 19958

刷新并显示未借用车辆

刷新并显示我已拥有的车辆

刷新并显示我已租借的车辆

刷新并显示所有车辆

查询

发行汽车

刷新

获得空投积分

汽车ID	汽车拥有者	汽车借用者	是否被借用	归还时间	操作
1	0x2e4892Ec35A41695Cf95bf20F2b500a048D605b0	0x71842261af79d9640cc9f72f1e76eff49d653624	已被借用	🕒2023-11-12 06:11:11	归还
4	0xF957C6bbB01b0611e800335F8b8f80ABeCfe6C13	0x71842261af79d9640cc9f72f1e76eff49d653624	已被借用	🕒2023-11-12 06:11:31	归还

```

async showBorrowedCars() {
    // console.log('1')
    const tempList3 = Object.assign([], await BorrowYourCarContract.methods.getMyBorrowedCars(this.currentUser).call())
    // console.log(tempList3)
    this.listOfMyBorrowedCars = this.getCarDetailsArray(tempList3)
    // console.log(this.listOfMyOwnedCars)
    this.listData = Object.assign([], this.listOfMyBorrowedCars)
    // console.log(this.listData)
    Promise.all(this.listData)
        .then((result) => {
            // 将结果存储在一个同步数组中
            this.listOfResult = result
            console.log(this.listOfResult) // 输出同步数组
        })
        .catch((error) => {
            console.error(error) // 处理错误情况
        })
},

```

2.3查看当前还没有被借用的汽车列表

在合约中实现了getAvailableCars函数：

```

function getAvailableCars() external view returns (uint256[] memory) {
    // Get the list of available car token IDs (not currently borrowed)
    uint256[] memory availableCars = new uint256[](totalCars -

```

```
totalRentedCars);
    uint256 counter = 0;
    for (uint256 i = 0; i < totalCars; i++) {
        if (ownerOf(i) != address(0) && !cars[i].isBorrowed) {
            availableCars[counter] = i;
            counter++;
        }
    }
    return availableCars;
}
```

在前端中通过按钮触发并且调用它，渲染数据到listOfResult中：

Dashboard

连接钱包

当前用户: 0x71842261af79d9640cc9f72f1e76eff49d653624

当前用户积分: 19958

刷新并显示未借用的车辆

刷新并显示我已拥有的车辆

刷新并显示我已租借的车辆

刷新并显示所有车辆

查询

发行汽车

刷新

获得空投积分

汽车ID	汽车拥有者	汽车借用人	是否被借用	归还时间	操作
0	0x71842261af79d9640cc9f72f1e76eff49d653624	无	未被借用	∞无	借用
2	0x71842261af79d9640cc9f72f1e76eff49d653624	无	未被借用	∞无	借用
3	0x2e4892Ec35A41695Cf95bf20F2b500a048D605b0	无	未被借用	∞无	借用
5	0xB306854d7e77516413DB350E73890a4C2507A170	无	未被借用	∞无	借用
6	0xbd4850F5692dF75c8f78B47732AD0EaABEc2fee	无	未被借用	∞无	借用
7	0x71842261af79d9640cc9f72f1e76eff49d653624	无	未被借用	∞无	借用
◦	0x2e4892Ec35A41695Cf95bf20F2b	无	未被借用	∞无	借用

```
async showUnborrowedVehicles() {
    // 处理显示未借用车辆的逻辑
    const tempList = Object.assign([], await BorrowYourCarContract.methods.getAvailableCars().call())
    // console.log(tempList3)
    this.listOfUnborrowedCars = this.getCarDetailsArray(tempList)
    // console.log(this.listOfMyOwnedCars)
    this.listData = Object.assign([], this.listOfUnborrowedCars)
    // console.log(this.listData)
    Promise.all(this.listData)
        .then((result) => {
            // 将结果存储在一个同步数组中
            this.listOfResult = result
            console.log(this.listOfResult) // 输出同步数组
        })
        .catch((error) => {
            console.error(error) // 处理错误情况
        })
},
```

2.4显示所有的汽车列表

在合约中getAllCars函数：

连接钱包

当前用户: 0x71842261af79d9640cc9f72f1e76eff49d653624

当前用户积分: 19958

刷新并显示未借车辆	刷新并显示我已拥有的车辆	刷新并显示我已租借的车辆	刷新并显示所有车辆	查询	发行汽车	刷新	获得空投积分
汽车ID	汽车拥有者	汽车借用者	是否被借用	归还时间	操作		
0	0x71842261af79D9640cC9f72f1e76eff49d653624	无	未被借用	⌚无	借用		
1	0x2e4892Ec35A41695Cf95b20F2b500a048D605b0	0x71842261af79D9640cC9f72f1e76eff49d653624	已被借用	⌚2023-11-12 06:11:11	归还		
2	0x71842261af79D9640cC9f72f1e76eff49d653624	无	未被借用	⌚无	借用		
3	0x2e4892Ec35A41695Cf95b20F2b500a048D605b0	无	未被借用	⌚无	借用		
4	0xF957C6bbB01b0611e800335F8b8f80ABeCfe6C13	0x71842261af79D9640cC9f72f1e76eff49d653624	已被借用	⌚2023-11-12 06:11:31	归还		
5	0xB306854d7e77516413DB350E73890a4C2507A170	无	未被借用	⌚无	借用		
0xbd4850fF5692dF75c8f78B47732A							

```
function getAllCars() external view returns (uint256[] memory) {
    uint256[] memory allCars = new uint256[](totalCars);
    for (uint256 i = 0; i < totalCars; i++) {
        allCars[i] = i;
    }
    return allCars;
}
```

在前端中通过按钮触发并且调用它，渲染数据到listOfResult中：

```
async showAllCars() {
    const tempList = Object.assign([], await BorrowYourCarContract.methods.getAllCars().call())
    // console.log(tempList3)
    this.listOfAllCars = this.getCarDetailsArray(tempList)
    // console.log(this.listOfMyOwnedCars)
    this.listData = Object.assign([], this.listOfAllCars)
    // console.log(this.listData)
    Promise.all(this.listData)
        .then((result) => {
            // 将结果存储在一个同步数组中
            this.listOfResult = result
            console.log(this.listOfResult) // 输出同步数组
        })
        .catch((error) => {
            console.error(error) // 处理错误情况
        })
},
```

2.5查询一辆汽车的主人，以及该汽车当前的借用者

通过输入需要查询的汽车的ID号，在汽车列表中查找并且返回它，最后渲染到listOfResult中：

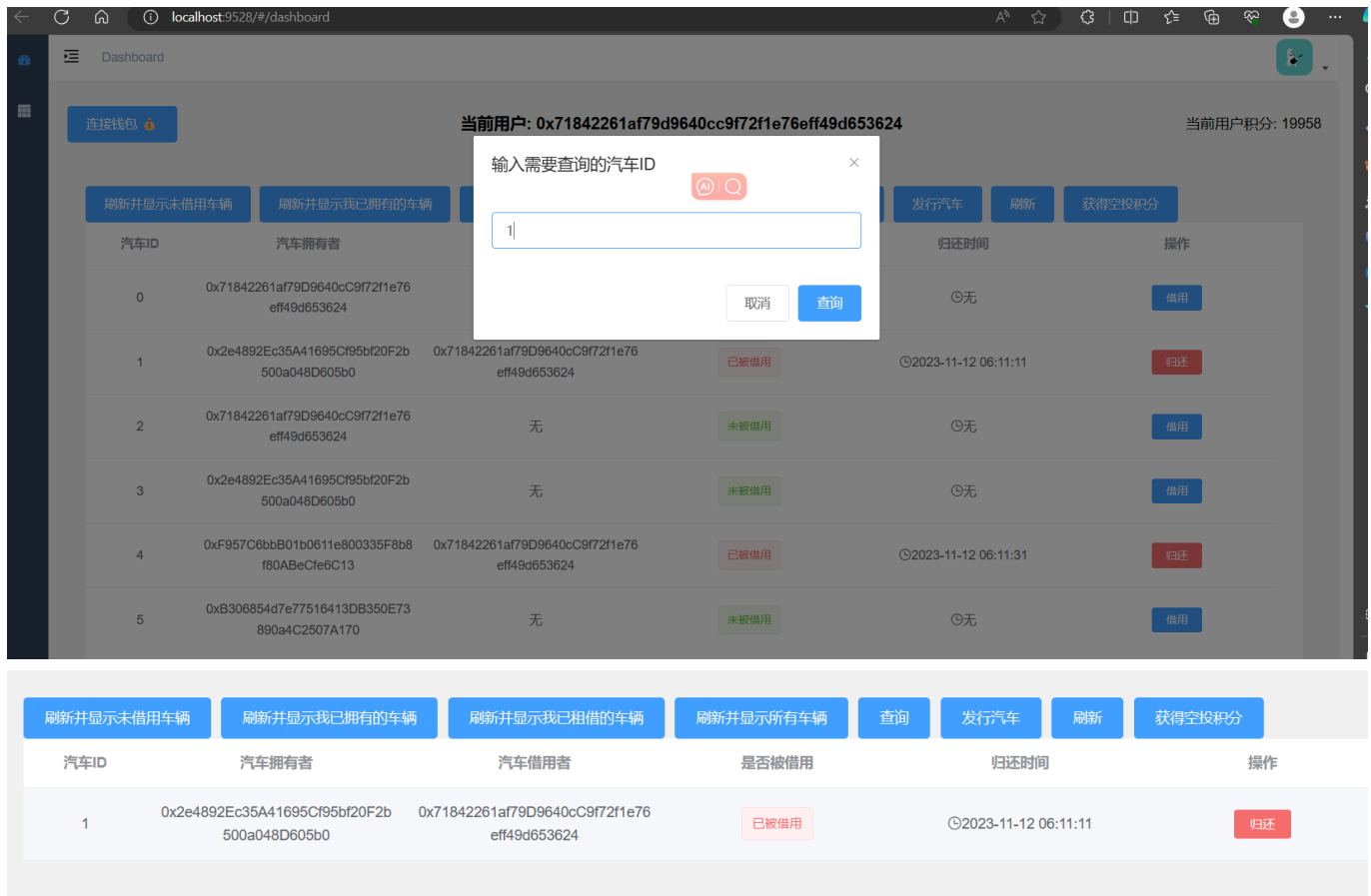
```
async performSearch() {
  // 处理查询逻辑

  const tokenId = +this.searchValue

  const tempList = [tokenId]

  this.listOfMyBorrowedCars = this.getCarDetailsArray(tempList)
  // console.log(this.listOfMyOwnedCars)
  this.listData = Object.assign([], this.listOfMyBorrowedCars)
  // console.log(this.listData)
  Promise.all(this.listData)
    .then((result) => {
      // 将结果存储在一个同步数组中
      this.listOfResult = result
      console.log(this.listOfResult) // 输出同步数组
    })
    .catch((error) => {
      console.error(error) // 处理错误情况
    })
  this.searchDialogVisible = false
},
```





2.6选择并借用某辆还没有被租借的汽车一段时间，使用自己发行的ERC20积分支付

在合约中实现了borrowCar函数：

```
function borrowCar(uint256 carTokenId, uint256 duration) external {
    require(ownerOf(carTokenId) != address(0), "Car does not exist");
    require(cars[carTokenId].borrower == address(0), "Car is already borrowed");

    address borrower = msg.sender;
    uint256 startTime = block.timestamp;
    uint256 borrowUntil = startTime + duration;
    calculateBorrowCost(duration);
    // Update the car's information
    cars[carTokenId].borrower = borrower;
    cars[carTokenId].borrowUntil = borrowUntil;
    cars[carTokenId].isBorrowed = true;
    totalRentedCars++;
    myERC20.transferFrom(msg.sender, address(this), tokenAmount);
    myERC20.transfer(cars[carTokenId].owner, myERC20.balanceOf(address(this)));
    // emit CarBorrowed(carTokenId, borrower, startTime, duration);
}
```


在前端中通过按钮触发并调用它：

3	0x2e4892Ec35A41695Cf95bf20F2b500a048D605b0	无	未被借用	☹无	借用
---	--	---	------	----	----

当前用户积分: 19958

汽车ID	汽车拥有者
0	0x71842261af79D9640cC9f72f1e76eff49d653624
1	0x2e4892Ec35A41695Cf95bf20F2b500a048D605b0
2	0x71842261af79D9640cC9f72f1e76eff49d653624
3	0x2e4892Ec35A41695Cf95bf20F2b500a048D605b0

输入借用时长

1天

5分

10小时

0秒

取消 确认

归还时间	操作
☹无	借用
☹2023-11-12 06:11:11	归还
☹无	借用
☹2023-11-12 20:33:27	归还

当前用户积分: 19923

```
async confirmBorrow(id, duration) {
  // 在这里处理借用时间的逻辑，例如发送请求或更新数据等
  // console.log(this.selectedRow.borrowUntil)
  // 关闭对话框
  const costAmount = parseInt(+duration / 3600 + 1)
  await myERC20Contract.methods.approve(BorrowYourCarContract.options.address, costAmount).send({
    from: this.currentUser
  })
  console.log('3')
  await BorrowYourCarContract.methods.borrowCar(id, duration).send({ from: this.currentUser })
  this.userScore = await myERC20Contract.methods.balanceOf(this.currentUser).call()
  this.dialogVisible = false
}
```

2.7归还汽车

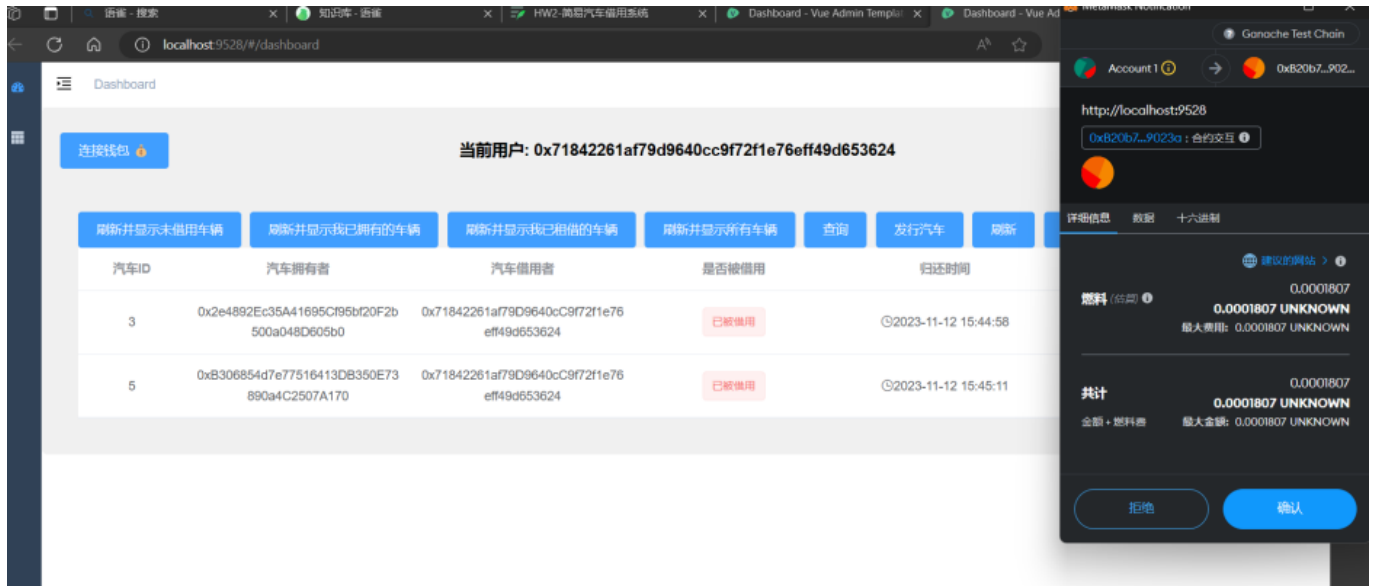
在合约中实现了returunCar函数：

```
function returnCar(uint256 carTokenId) external {

    // Update the car's information
    cars[carTokenId].borrower = address(0);
    cars[carTokenId].borrowUntil = 0;
    cars[carTokenId].isBorrowed = false;
    totalRentedCars--;

}
```

在前端中通过按钮触发并调用它：



2.8发行汽车

在合约中实现createCars函数：

```
function createCars() external {
  // Create and distribute NFTs representing cars to test users
  // For example, give NFTs to certain addresses for testing purposes
  address[] memory users = new address[](10);
  users[0] = address(uint160(0x71842261af79D9640cC9f72f1e76eff49d653624));
  users[1] = address(uint160(0x2e4892Ec35A41695Cf95bf20F2b500a048D605b0));
  users[2] = address(uint160(0xF957C6bbB01b0611e800335F8b8f80ABeCfe6C13));
  users[3] = address(uint160(0xB306854d7e77516413DB350E73890a4C2507A170));
  users[4] = address(uint160(0xbd4850ff5692dF75c8f78B47732AD0EaABeC2fee));
  users[5] = address(uint160(0x8A6A3e2F007157A1665dFdCD916A4B5fe054C87B));
  users[6] = address(uint160(0xF891eAe8C49AaC400364d9F2e906de240E51E311));
  users[7] = address(uint160(0x8798bB05cC335615b8bF56bae3DE3FC907d7ff97));
  users[8] = address(uint160(0xc64BAc49A643140B11A9C69EeB5E3F14FFDFA303));
  users[9] = address(uint160(0x70f66b58BdfB582a22cfc301a14c13c65a40DC66));

  //[[address(0x30f270d7dc5a03391b3d9098116d78c419dc185bf7e4b58b4fc14fa0b583d57b),
  address(0xf13d4b0ab910c8d78bb22629c858ea73ee22d9b0a5e8bd462d48190c49bd14b3)]];
  for (uint256 i = 0; i < 2; i++) {
    // Mint a car NFT and assign it to the user
    uint256 tokenId = mintCar(users[i]);
    // Update the car's owner in the cars mapping
```

```

        cars[tokenId].owner = users[i];
    }
    for (uint256 i = 0; i < 5; i++) {
        // Mint a car NFT and assign it to the user
        uint256 tokenId = mintCar(users[i]);
        // Update the car's owner in the cars mapping
        cars[tokenId].owner = users[i];
    }
    for (uint256 i = 0; i < 7; i++) {
        // Mint a car NFT and assign it to the user
        uint256 tokenId = mintCar(users[i]);
        // Update the car's owner in the cars mapping
        cars[tokenId].owner = users[i];
    }
    for (uint256 i = 0; i < 10; i++) {
        // Mint a car NFT and assign it to the user
        uint256 tokenId = mintCar(users[i]);
        // Update the car's owner in the cars mapping
        cars[tokenId].owner = users[i];
    }
}

```

查询

发行汽车

刷新

在前端中通过按钮触发并调用它：

```

async createCars() {
    await BorrowYourCarContract.methods.createCars().send({ from: this.currentUser })
},

```

3.项目运行截图

按照以下流程操作：编译部署合约，启动前端，连接钱包，发行汽车，显示所有汽车，显示自己拥有的汽车，获得空投积分，借用一些汽车，显示已借用的汽车，显示未借用的汽车，归还汽车，显示已经借用的汽车，查询一辆汽车 下面是按照此流程运行的截图： 编译部署合约：

```

PS C:\Users\17875\Desktop\courses\区块链\HW2\ZJU-blockchain-course-2023\contracts> npx hardhat run ./scripts/deploy.ts --network ganache
BorrowYourCar deployed to 0xB20b7b1425C789C768f785AA2087FA170C69023a
erc20 contract has been deployed successfully in 0xE609FB420A695725716f879E0aE00b204D50bDfB

```

```

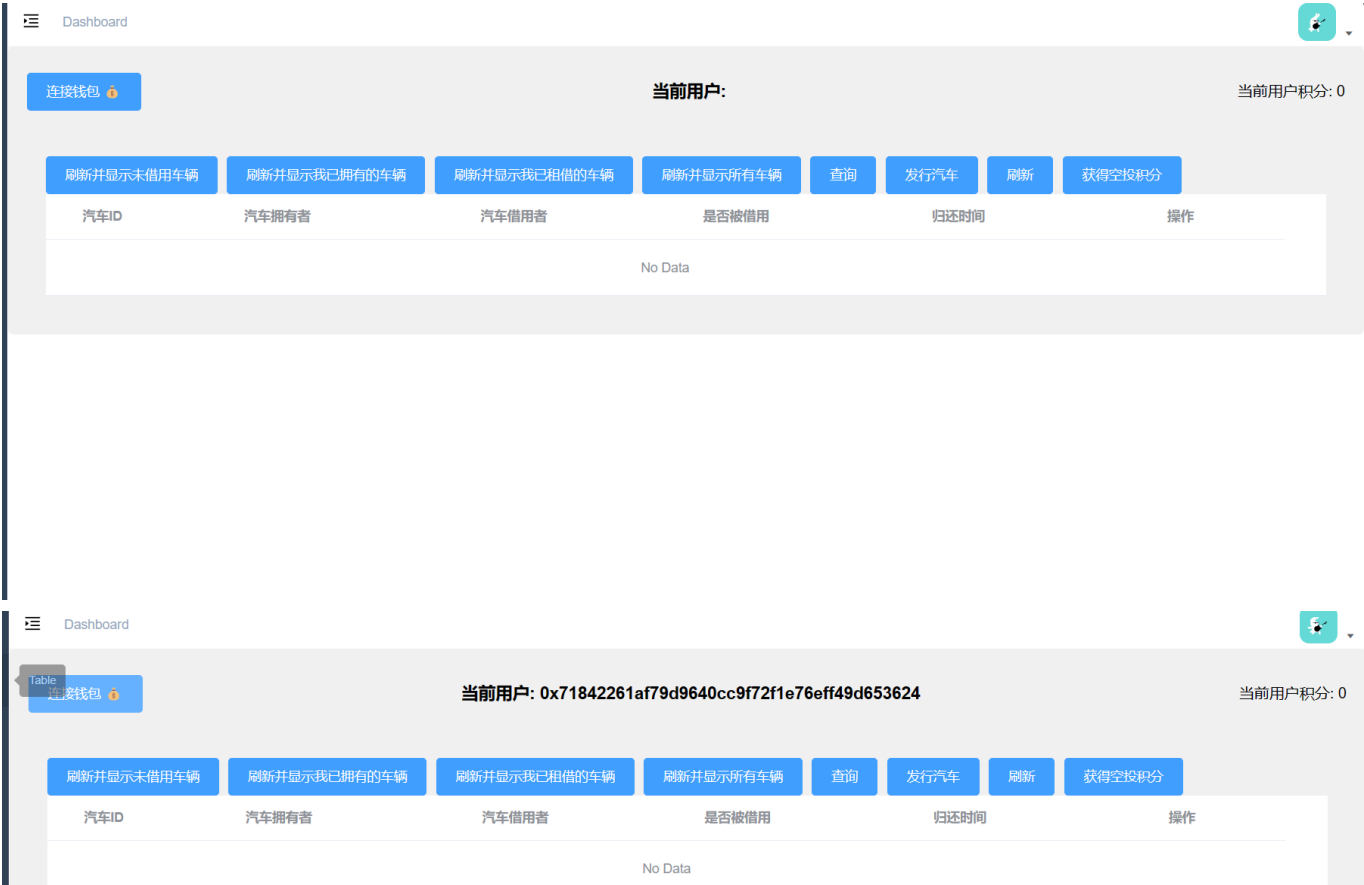
{
  "BorrowYourCar": "0xB20b7b1425C789C768f785AA2087FA170C69023a",
  "myERC20": "0xE609FB420A695725716f879E0aE00b204D50bDfB"
}

```

启动前端：



连接钱包：



发行汽车：



显示所有汽车：

连接钱包

当前用户: 0x71842261af79d9640cc9f72f1e76eff49d653624

当前用户积分: 0

刷新并显示未借用车辆

刷新并显示我已拥有的车辆

刷新并显示我已租借的车辆

刷新并显示所有车辆

查询

发行汽车

刷新

获得空投积分

汽车ID	汽车拥有者	汽车借用者	是否被借用	归还时间	操作
0	0x71842261af79d9640cC9f72f1e76eff49d653624	无	未被借用	无	借用
1	0x2e4892Ec35A41695Cf95bf20F2b500a048D605b0	无	未被借用	无	借用
2	0x71842261af79d9640cC9f72f1e76eff49d653624	无	未被借用	无	借用
3	0x2e4892Ec35A41695Cf95bf20F2b500a048D605b0	无	未被借用	无	借用
4	0xF957C6bbB01b0611e800335F8b8f80ABeCfe6C13	无	未被借用	无	借用
5	0xB306854d7e77516413DB350E73890a4C2507A170	无	未被借用	无	借用
6	0xbd4850fF5692dF75c8f78B47732Ad0Fa8Fcf7fe	无	未被借用	无	借用

显示自己拥有的汽车：

连接钱包

当前用户: 0x71842261af79d9640cc9f72f1e76eff49d653624

当前用户积分: 0

刷新并显示未借用车辆

刷新并显示我已拥有的车辆

刷新并显示我已租借的车辆

刷新并显示所有车辆

查询

发行汽车

刷新

获得空投积分

汽车ID	汽车拥有者	汽车借用者	是否被借用	归还时间	操作
0	0x71842261af79d9640cC9f72f1e76eff49d653624	无	未被借用	无	借用
2	0x71842261af79d9640cC9f72f1e76eff49d653624	无	未被借用	无	借用
7	0x71842261af79d9640cC9f72f1e76eff49d653624	无	未被借用	无	借用
14	0x71842261af79d9640cC9f72f1e76eff49d653624	无	未被借用	无	借用

获得空投积分：

Dashboard

当前用户: 0x71842261af79d9640cc9f72f1e76eff49d653624

当前用户积分: 0

刷新并显示未借用车辆

刷新并显示我已拥有的车辆

刷新并显示我已租借的车辆

刷新并显示所有车辆

查询

发行汽车

刷新

获得空投积分

汽车ID	汽车拥有者	汽车借用者	是否被借用	归还时间	操作
0	0x71842261af79d9640cC9f72f1e76eff49d653624	无	未被借用	无	借用
2	0x71842261af79d9640cC9f72f1e76eff49d653624	无	未被借用	无	借用
7	0x71842261af79d9640cC9f72f1e76eff49d653624	无	未被借用	无	借用
14	0x71842261af79d9640cC9f72f1e76eff49d653624	无	未被借用	无	借用

Account 1

0xE609F...0bD...

http://localhost:9528

0xE609F...0bDfB : AIRDROP

详细信息 数据 十六进制

建议的网站

燃料 (估计) 0.0002546

很可能在 15 秒 以内 最大费用: 0.0002546 UNKNOWN

共计 0.0002546

金额 + 燃料费 最大金额: 0.0002546 UNKNOWN

拒绝 确认

localhost:9528 显示

You have claimed ZJU Token.

确定

连接钱包

当前用户: 0x71842261af79d9640cc9f72f1e76eff49d653624

当前用户积分: 10000

刷新并显示未借用车辆

刷新并显示我已拥有的车辆

刷新并显示我已租借的车辆

刷新并显示所有车辆

查询

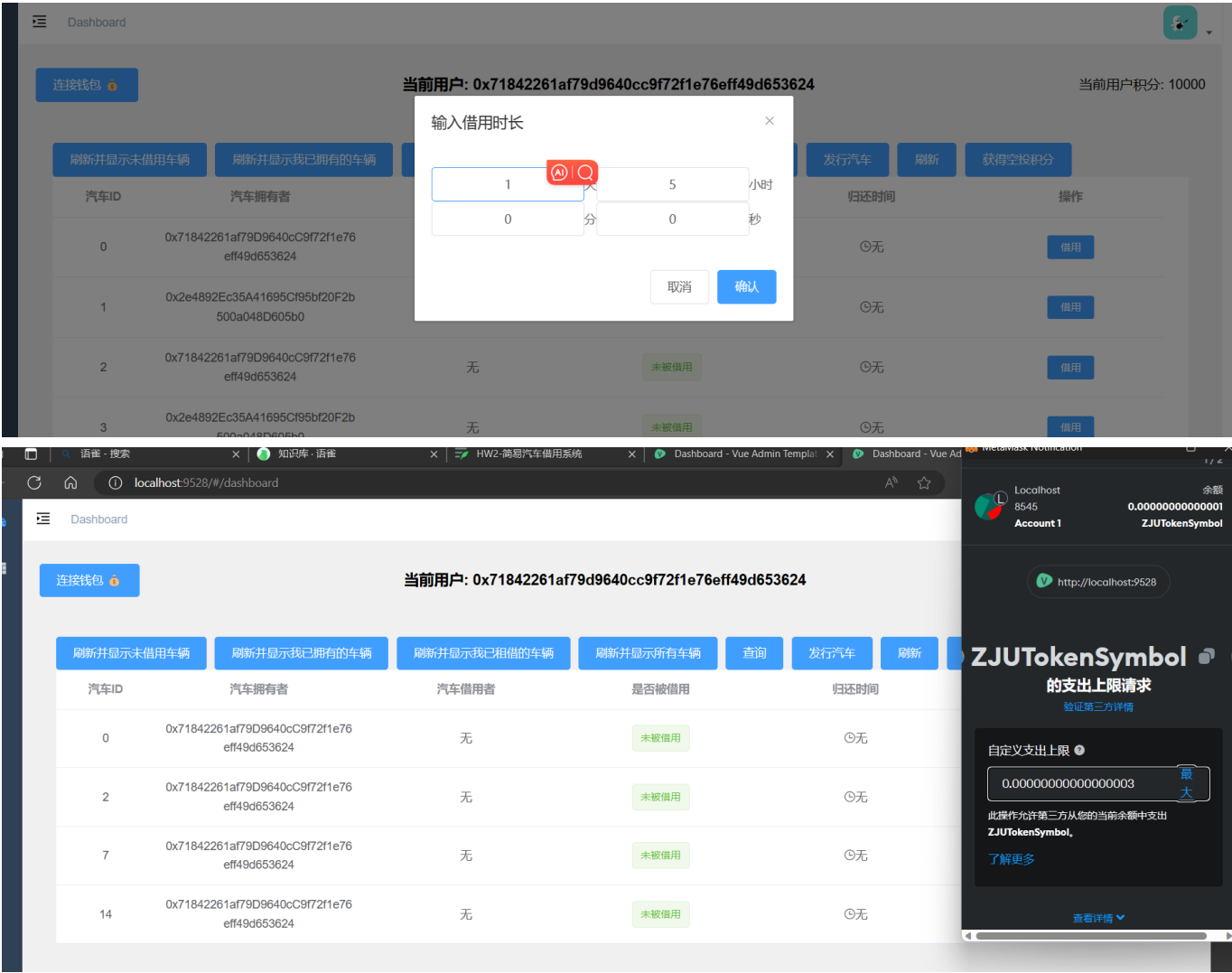
发行汽车

刷新

获得空投积分

汽车ID	汽车拥有者	汽车借用者	是否被借用	归还时间	操作
0	0x71842261af79d9640cC9f72f1e76eff49d653624	无	未被借用	无	借用
2	0x71842261af79d9640cC9f72f1e76eff49d653624	无	未被借用	无	借用
7	0x71842261af79d9640cC9f72f1e76eff49d653624	无	未被借用	无	借用
14	0x71842261af79d9640cC9f72f1e76eff49d653624	无	未被借用	无	借用

借用一些汽车（一共借了两辆，因为操作一样，只放了借用其中一辆时的截图）：



显示我已借用汽车列表：



显示所有未借用汽车

连接钱包

当前用户: 0x71842261af79d9640cc9f72f1e76eff49d653624

当前用户积分: 9940

刷新并显示未借用车辆

刷新并显示我已拥有的车辆

刷新并显示我已租借的车辆

刷新并显示所有车辆

查询

发行汽车

刷新

获得空投积分

汽车ID	汽车拥有者	汽车借用者	是否被借用	归还时间	操作
0	0x71842261af79d9640cC9f72f1e76eff49d653624	无	未被借用	⌚无	借用
1	0x2e4892Ec35A41695Cf95bf20F2b500a048D605b0	无	未被借用	⌚无	借用
2	0x71842261af79d9640cC9f72f1e76eff49d653624	无	未被借用	⌚无	借用
4	0xF957C6bbB01b0611e800335F8b8f80ABeCfe6C13	无	未被借用	⌚无	借用
6	0xbd4850fF5692dF75c8f78B47732AD0EaABEc2fee	无	未被借用	⌚无	借用
7	0x71842261af79d9640cC9f72f1e76eff49d653624	无	未被借用	⌚无	借用
8	0x2e4892Ec35A41695Cf95bf20F2b500a048D605b0	无	未被借用	⌚无	借用

归还汽车：

Dashboard

连接钱包

当前用户: 0x71842261af79d9640cc9f72f1e76eff49d653624

刷新并显示未借用车辆

刷新并显示我已拥有的车辆

刷新并显示我已租借的车辆

刷新并显示所有车辆

查询

发行汽车

刷新

获得空投积分

汽车ID	汽车拥有者	汽车借用者	是否被借用	归还时间	操作
3	0x2e4892Ec35A41695Cf95bf20F2b500a048D605b0	0x71842261af79d9640cC9f72f1e76eff49d653624	已被借用	⌚2023-11-12 15:44:58	
5	0xB306854d7e77516413DB350E73890a4C2507A170	0x71842261af79d9640cC9f72f1e76eff49d653624	已被借用	⌚2023-11-12 15:45:11	

Account 1

0xB20b7...902...

http://localhost:9528

0xB20b7...9023a : 合约交互

详细信息 数据 十六进制

建议的网站

燃料 (合约)

0.0001807 UNKNOWN

最大费用: 0.0001807 UNKNOWN

共计

0.0001807 UNKNOWN

金额 + 燃料费 最大金额: 0.0001807 UNKNOWN

拒绝 确认

显示我已借用汽车列表：

刷新并显示未借用车辆

刷新并显示我已拥有的车辆

刷新并显示我已租借的车辆

刷新并显示所有车辆

查询

发行汽车

刷新

获得空投积分

汽车ID	汽车拥有者	汽车借用者	是否被借用	归还时间	操作
5	0xB306854d7e77516413DB350E73890a4C2507A170	0x71842261af79d9640cC9f72f1e76eff49d653624	已被借用	⌚2023-11-12 15:45:11	归还

查询一辆汽车：

