


2023 네트워크최신기술

CAR

SHARING

20181640 안세홍

20181580 김동원



CONTENTS —

01 Car Sharing 어플 목적

02 메인 페이지

03 차 예약 페이지

04 세부 구현 방법



01

Car Sharing 어플 목적



어플리케이션의 목적

스마트 컨트랙트를 통해 쏘카와 같은 자동차 예약 어플리케이션을 개발하고자 합니다.

이를 통해 차량 예약 프로세스를 블록체인 기술로 관리하고, 사용자들 간의 투명하고 신뢰성 높은 거래를 보장할 수 있습니다.

또한, 블록체인의 불변성과 투명성을 활용하여 보안성과 안정성을 강화하며, 기존 시스템의 한계를 극복할 수 있습니다.





— 02

메인 페이지

CarSharing

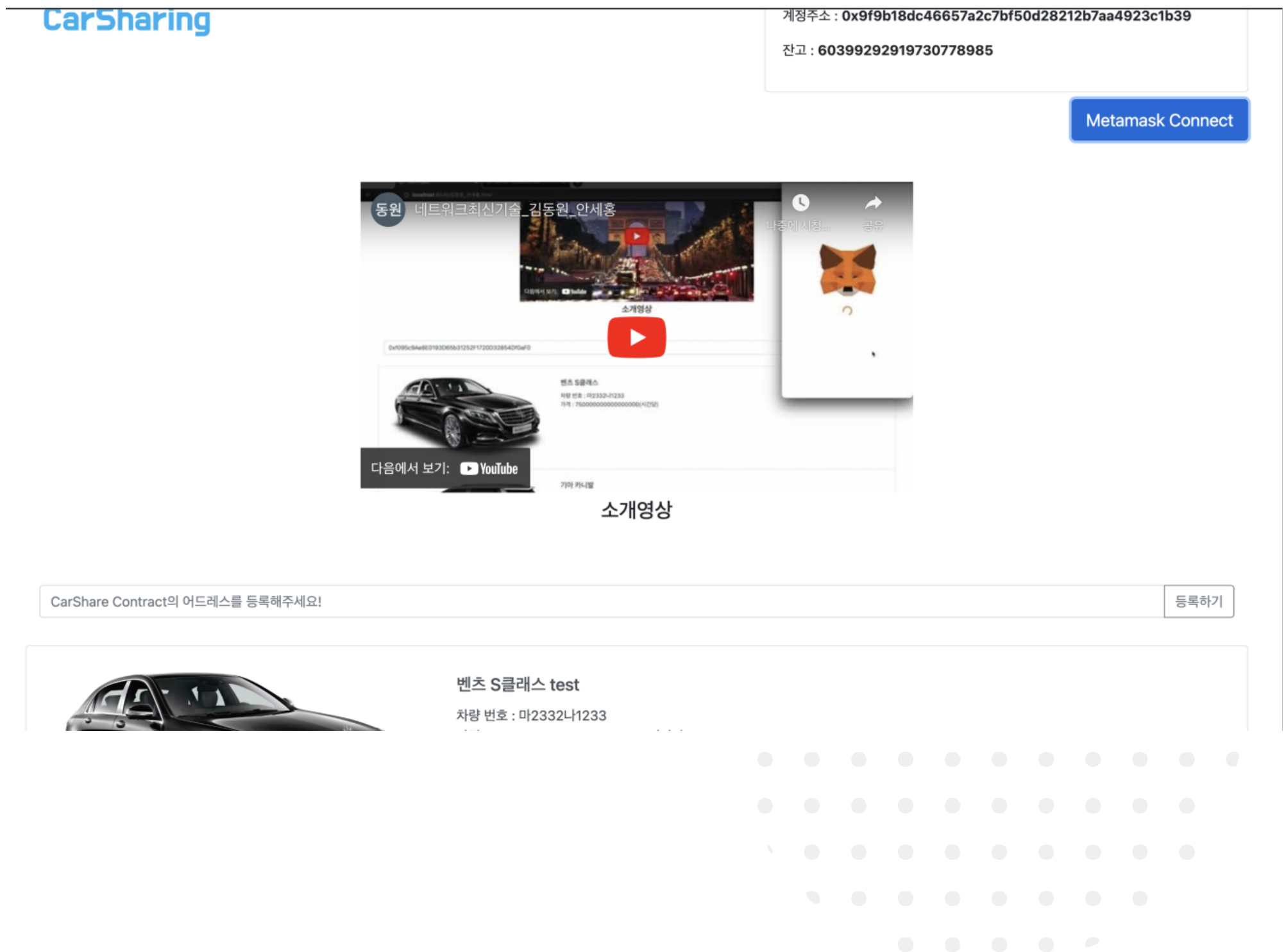
메인 페이지

메인 페이지는 CarShare Contract를 등록하고 예약할 수 있는 차량 리스트를 만들어주기 위해 기획되었습니다.

헤더에는 메인 로고와 계정 정보를 보여줄 수 있는 영역이 있습니다.

또한 배포된 CarShare Contract를 등록할 수 있는 UI가 존재합니다.

그리고 등록하기 영역 아래에는 차량 리스트를 보여줍니다. 차량 리스트 안에 있는 정보는 차량사진 차량이름, 차량 번호, 가격을 나타내 줍니다.





03

차량 예약 페이지






차량 예약 페이지

차량 리스트 버튼을 통해 차량 예약 페이지에 들어 오게 되면 먼저 왼쪽에는 **차량 정보**(차량 종류, 차량번호, 가격)를 확인하실 수 있습니다.

오른쪽에는 **대여 시작일**과 **시작 시간**, **대여 반납일**과 **반납 시간**을 입력 받아 **자동으로** **계산**을 해주도록 화면을 구성하였습니다.

마지막으로 해당 차량 상태가 **대여 기간 조건 만족** 시 **대여시작** 버튼이 **활성화** 되도록 구성하였습니다.





벤츠 S클래스 test
마2332나1233
900000000000000000 Wei



계정주소 : 0x9f9b18dc46657a2c7bf50d28212b7aa4923c1b39
잔고 : 60399292919730778985

현재 상태 : 대여중아님

상태 조회

Metamask Connect



벤츠 S클래스 test
마2332나1233

대여 시작일 2023. 05. 03.

대여 반납일 2023. 05. 05.

시간(0-23) 2

시간(0-23) 2

가격 : 900000000000000000(시간 당 가격) x 48시간 =
4320000000000000000 Wei

대여시작

계정주소 : 0x9f9b18dc46657a2c7bf50d28212b7aa4923c1b39
잔고 : 60399292919730778985

현재 상태 : 대여중아님

상태 조회

Metamask Connect

대여 시작일 2023. 06. 07.

대여 반납일 2023. 06. 08.

시간(0-23) 2

시간(0-23) 2

가격 : 900000000000000000(시간 당 가격) x 24시간 =
2160000000000000000 Wei

대여시작

대여 시간 기준으로 자동 계산

The image is a screenshot of a web browser displaying a car reservation interface. The browser's address bar shows the URL: localhost:8545/carReservation.html?address=0xf095c9Ae8E0193D65b31252F172. The main content area features the 'CarSharing' logo on the left. Below it, a black Mercedes-Benz S-Class is shown. To the right of the car, the text reads '벤츠 S클래스 test', '마2332나1233', and '900000000000000000 Wei'. Further right, a box displays '계정주소 : 0x9f9b18dc46657a2' and '잔고 : 6039929291973077898'. Below this, it says '현재 상태 : 대여중아님' and '상태'. A reservation form shows '대여 시작일' as '2023. 06. 07.' and '시간(0-23)' as '2'. At the bottom, the price is listed as '가격 : 9000000000000000000' and '216000000000'. A dark wallet overlay is visible on the right side, showing 'CUSTOMER 1' with a balance of '21.6 ETH'. It also displays transaction details like '가스 (예상치)' and '합계'. At the bottom of the overlay are buttons for '거부' and '확인'. A blue button labeled '대여시작' is located at the bottom right of the main page.



차량 예약 페이지

마지막으로 확인 버튼을 클릭 후에는 대여
성공 alert가 뜨며 대여가 성공적으로 완료
됩니다.



localhost:8545 내용:

대여 성공!

확인

d28212b7aa4923c1b39

현재 상태 : 대여중아님

상태 조회

Metamask Connect



벤츠 S클래스 test

마2332나1233

900000000000000000 Wei

대여 시작일 2023. 06. 07.

대여 반납일 2023. 06. 08.

시간(0-23) 2

시간(0-23) 2

가격 : 900000000000000000(시간 당 가격) x 24시간 =
2160000000000000000 Wei

대여시작





차량 예약 페이지

만약 현재 시간 기준으로 대여중이라면 **상태 조회** 버튼을 클릭하여 현재 상태가 대여중아님에
서 **대여중**으로 변경됩니다.



벤츠 S클래스 test

마2332나1233

900000000000000000 Wei

잔고 : 38798976937344560155

현재 상태 : **대여중!**

상태 조회

Metamask Connect

대여 시작일 2023. 06. 07.



대여 반납일 2023. 06. 08.



시간(0-23)

2

시간(0-23)

2

가격 : 9000000000000000000(시간 당 가격) x 24시간 =
2160000000000000000 Wei

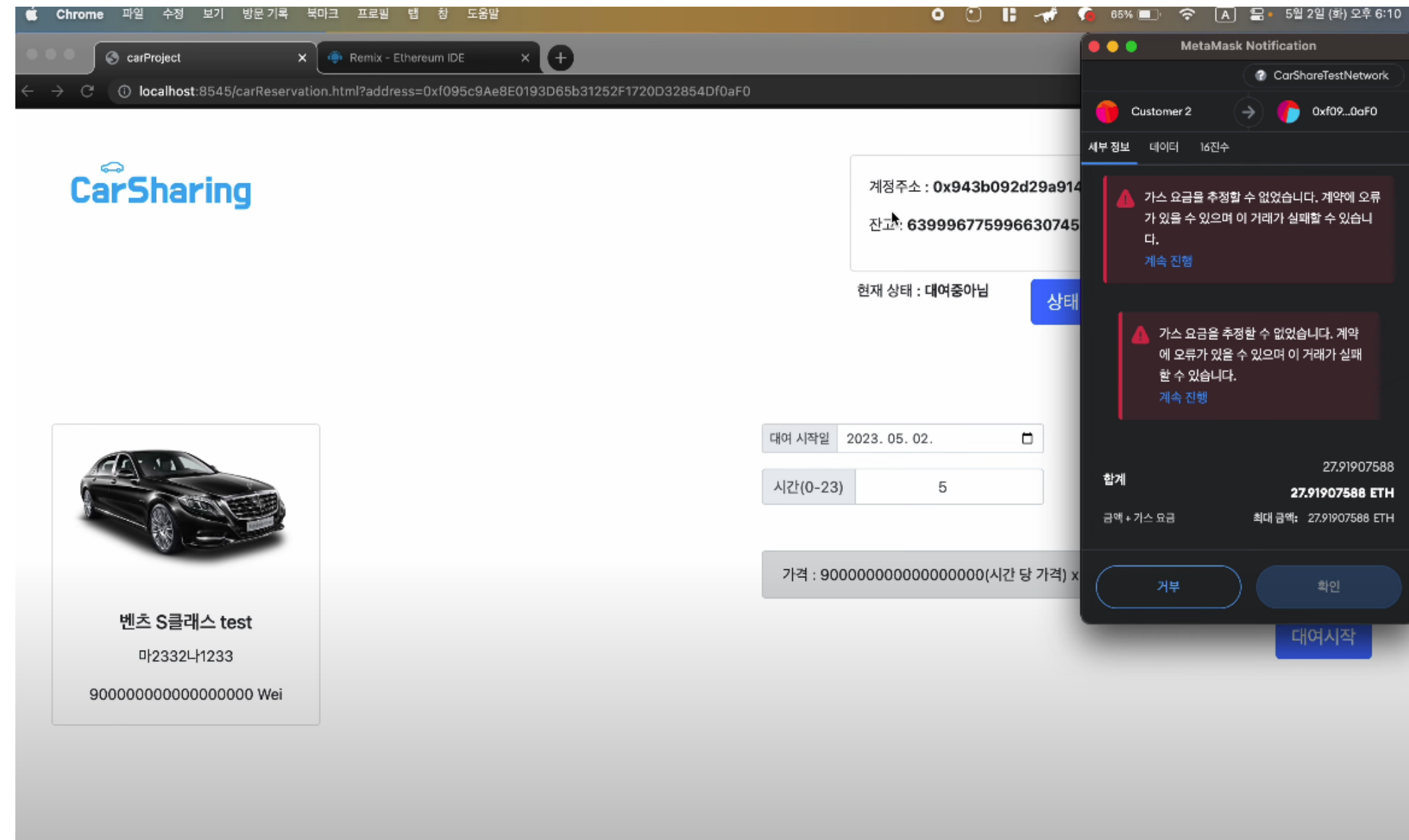
대여시작



CarSharing

차량 예약 페이지

그리고 타 고객의 예약 시간에 이미 대여중인 차량은 해당 시간에는 대여가 불가능하므로 대여 시작버튼을 클릭하더라도 대여가 불가능합니다.



— 04

세부 구현 방법 및 시연영상



세부 구현

ContractRegistry.sol

CarShare Contract를 등록할 수 있는 Smart Contract, CarShare Contract들을 배열로 저장하고 인덱스로 조회가 가능합니다.

CarShare.sol

owner와 타 계정 간의 차량 공유 예약 Smart Contract 입니다. validation이 가능하고, 차 정보 및 현재 계정이 대여중인지 판별할 수 있습니다. 예약 성공 시 바로 owner에게 예약금을 송금합니다.

sameAddress

이미 등록이 되어있는 컨트랙트인지 판별

enrollContract

컨트랙트를 등록하는 함수

getLength

컨트랙트를 저장하는 배열의 크기

getContractAddr

해당 인덱스의 컨트랙트 어드레스 리턴

ContractRegistry.sol

constructor

차량의 정보(사진 / 모델 / 번호 / 시간당 가격)와 함께 생성

getCarInfo

컨트랙트의 차량 정보를 리턴

isRentOwner

해당 계정이 대여중인지 판별

canReservation

예약이 가능한지 판별

userReservation

해당 계정이 차량 대여를 위한 예약하는 함수, 예약 성공과 동시에 owner에게 송금

CarShare.sol

CarSharing

세부 구현

```
✓ async function gateWayPage(req, res) {  
  var fname = "." + decodeURIComponent(url.parse(req.url).pathname);  
  ✓ fs.readFile(fname, async function (err, data) {  
    ✓ if(err) {  
      ✓ errorPage(res, 404, `${fname}이 존재하지 않습니다.`);  
      ✓ return;  
    }  
    ✓ res.writeHead(200, {'Content-Type': 'text/html'});  
    ✓ if(url.parse(req.url, true).query.address) {  
      ✓ const htmlString = data.toString().replace('{{contractAddr}}', url.parse(req.url, true).query.address);  
      ✓ res.write(htmlString);  
      ✓ return res.end();  
    }  
    ✓ res.write(data);  
    ✓ return res.end();  
  }  
  });  
}  
  
http.createServer(gateWayPage).listen(8545);
```

해당 프로젝트는 Node 어플리케이션입니다. 해당 프로젝트 터미널에서 node index.js를 실행시켜서 어플리케이션을 실행하고 http://localhost:8545/김동원_안세홍.html를 통해서 메인화면에 접속 가능합니다.

해당 어플리케이션의 실행방식은 README.md를 참고하시면 감사하겠습니다.

2023 네트워크최신기술

**THANK
YOU**