

<프로젝트 사용설명서>

1. Ganache cli 설치

프로젝트를 사용할 이용자의 가상 네트워크로 쓰일 Ganache를 다운받아준다. 아래의 사이트에서 Ganache 어플을 설치하면 된다.

[Ganache - Truffle Suite](#)

2. GitHub에서 프로젝트 파일 다운로드

명령 프롬프트에 아래와 같은 명령어를 입력하여 프로젝트 파일을 다운로드한다.
(CoffeeBean-main파일)

- git clone <프로젝트 URL>

3. 프로젝트를 실행시키기 위한 각종 npm 장치들을 다운로드

다운 받은 폴더로 들어간 후 아래의 명령어를 입력해 필요한 장치들을 다운받는다.

(3-1)

npm install vue -g

npm install @vue/cli -g

(3-2)

npm install axios

npm install vue-router

npm install web3

npm install vuex

npm install vuex-persistedstate

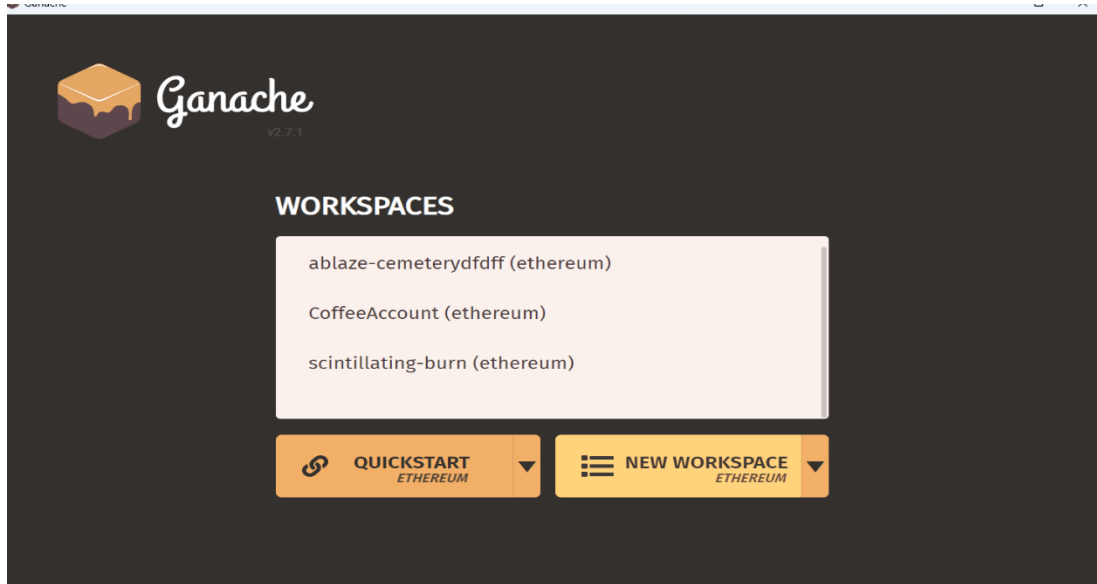
(명령 프롬프트에서 위의 내용에서 굵게 표시된 명령어들을 입력해주면 된다.)

이후 아래의 명령어를 입력하면 프로젝트를 실행시킬 수 있다.

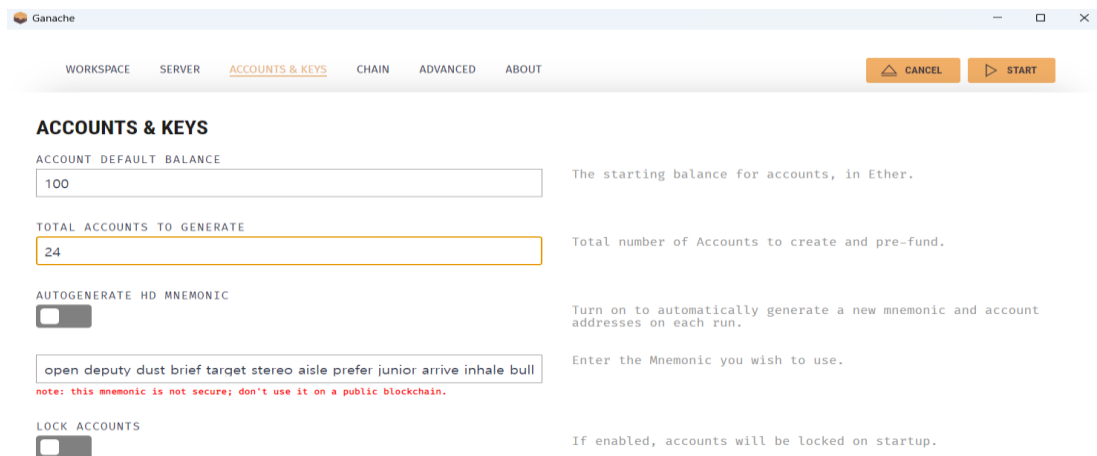
npm run serve

4. Ganache Test net 생성

Ganache를 실행시키면 아래와 같은 화면이 뜨고 여기서 NEW WORKSPACE 키를 눌러 새로운 블록체인 Test net을 생성한다.

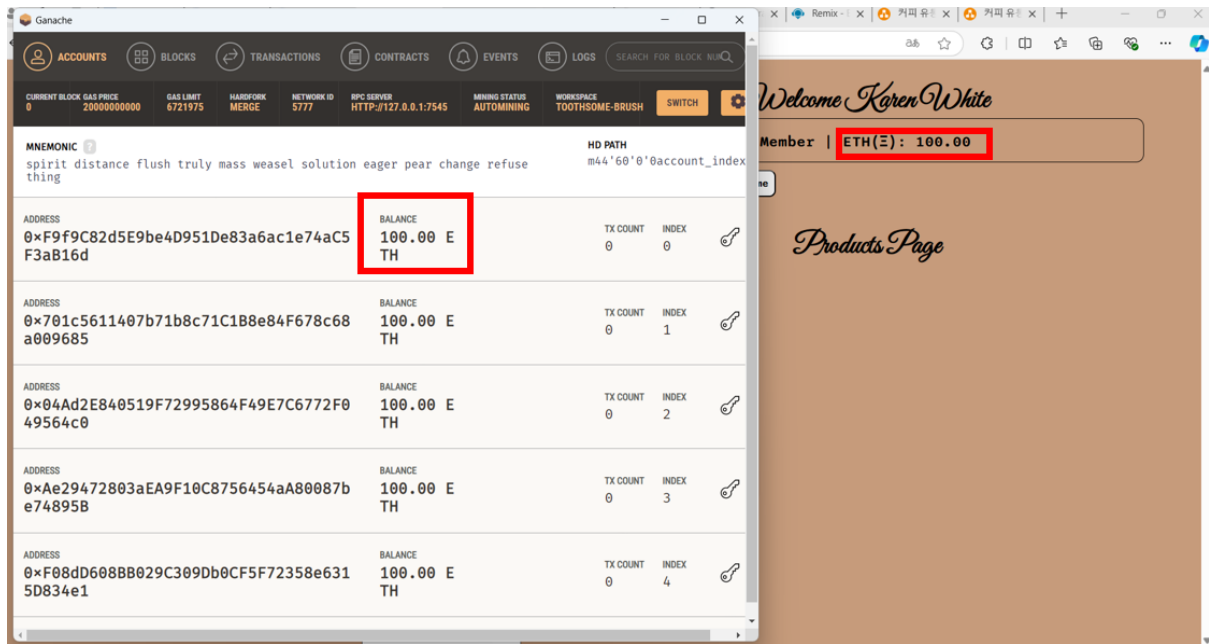


나머지는 기본 설정을 따르면 되지만 해당 프로젝트에선 블록체인 네트워크에 참여하는 참여자가 24명이기 때문에 아래와 같이 생성되는 계좌의 수를 24개로 설정해준다. 이후 왼쪽 상단의 START 버튼을 누르면 네트워크가 생성된다.



- 네트워크 연결 확인

이후 프로젝트를 실행시킨 후 로그인 해서 products 페이지에 들어간다. 거기서 나의 계정에 연결된 계좌의 ETH가 정상적으로 뜨면 연결이 된 것이다.



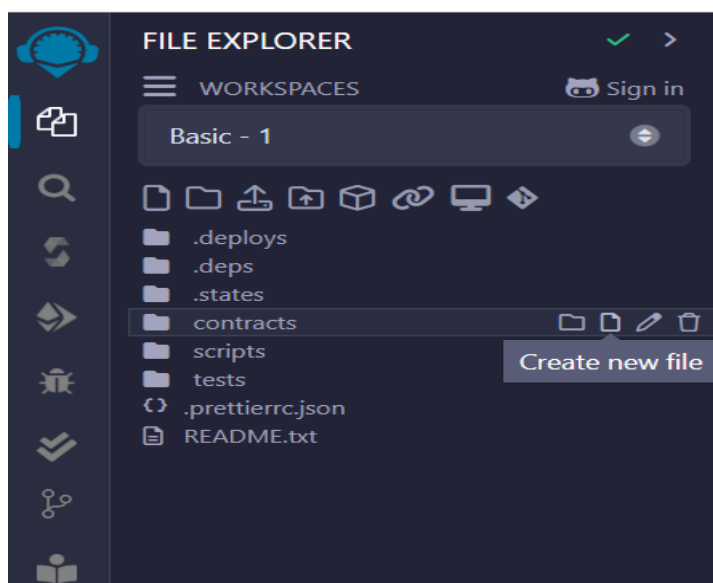
5. 자신의 블록체인 네트워크에 스마트 컨트랙트 배포

(1) Remix IDE 사이트에 접속

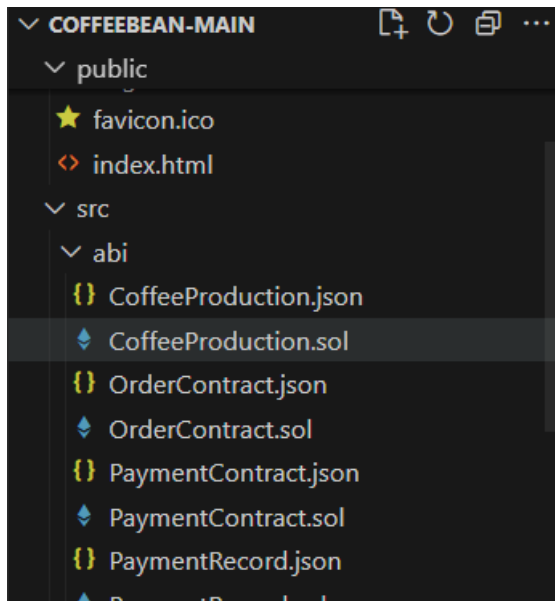
<https://remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null>

(2) WORKSPACES의 contracts 폴더에 프로젝트에 저장된 스마트 컨트랙트 파일들을 추가.

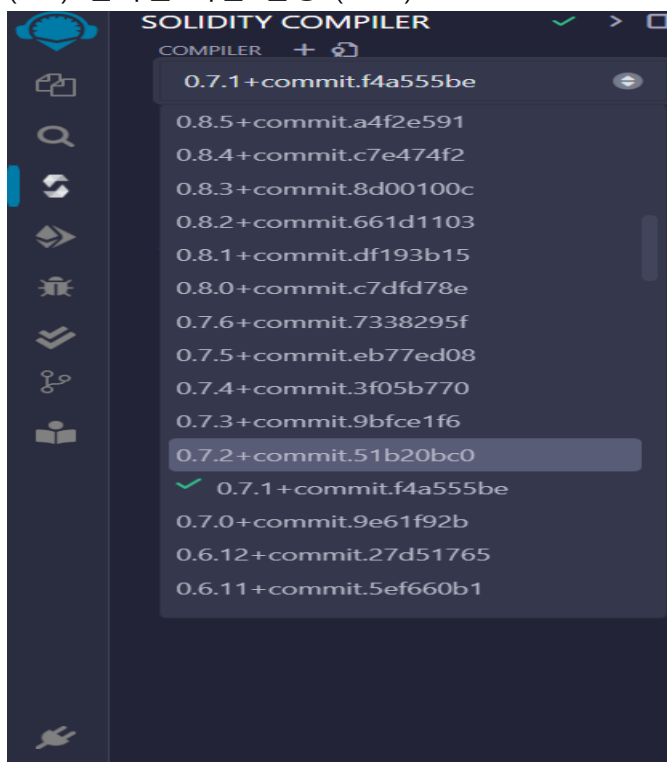
(2-1) contracts 폴더에서 Create new file 기능을 이용해 새로운 파일 생성



(2-2) 프로젝트 파일 중 /src/abi에 저장되어 있는 스마트 컨트랙트 파일 코드(.sol 파일)들을 모두 복사 + 붙여넣기 해 줌.

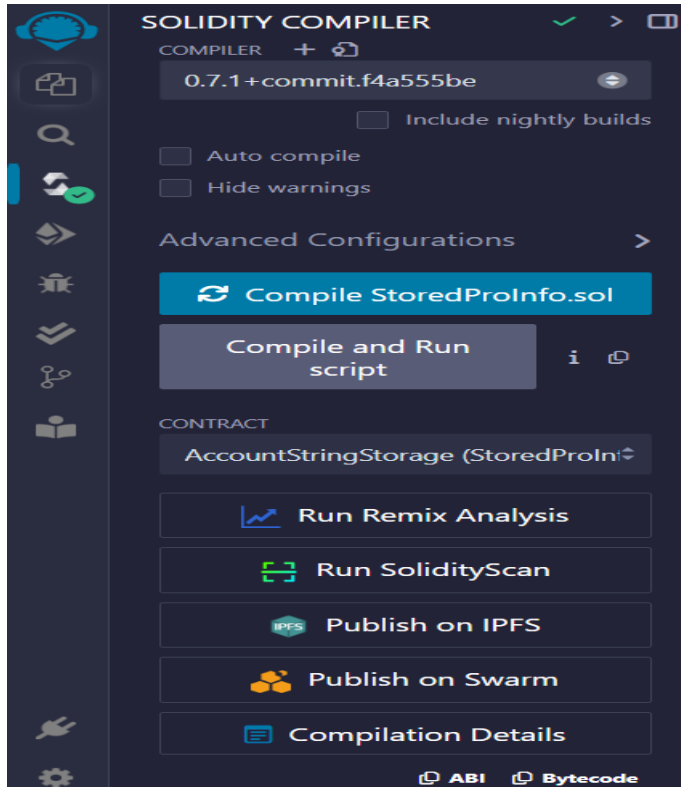


(2-3) 컴파일 버전 설정 (0.7.1)



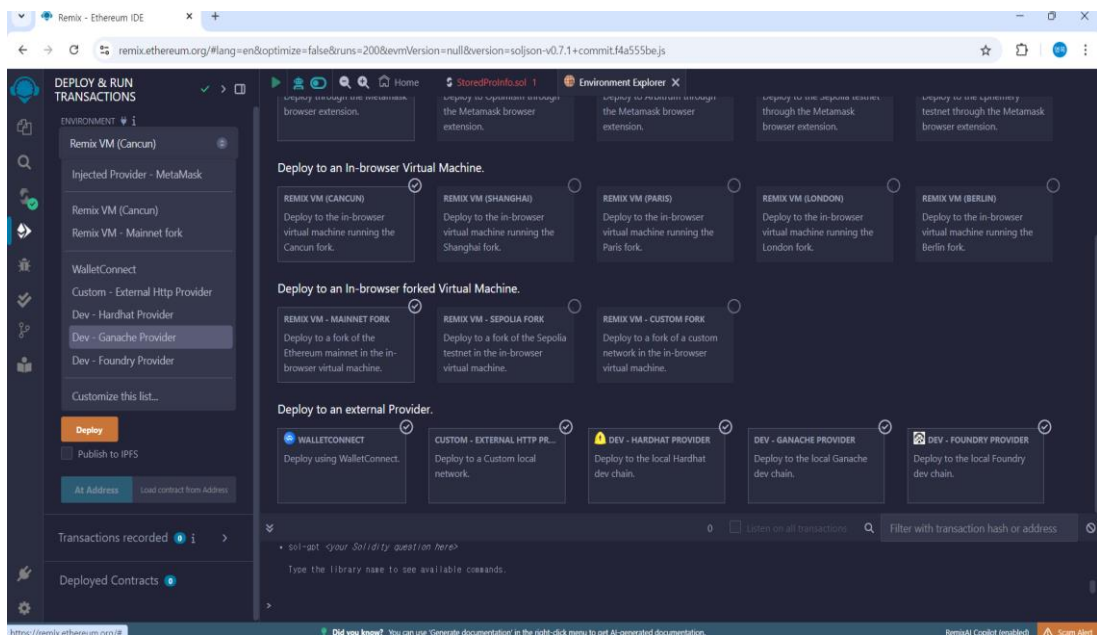
(2-4) 컴파일

Compile 버튼 혹은 Ctrl+S 키를 눌렀을 때 초록색 체크 아이콘이 뜨면 성공적으로 컴파일 된 것이다.

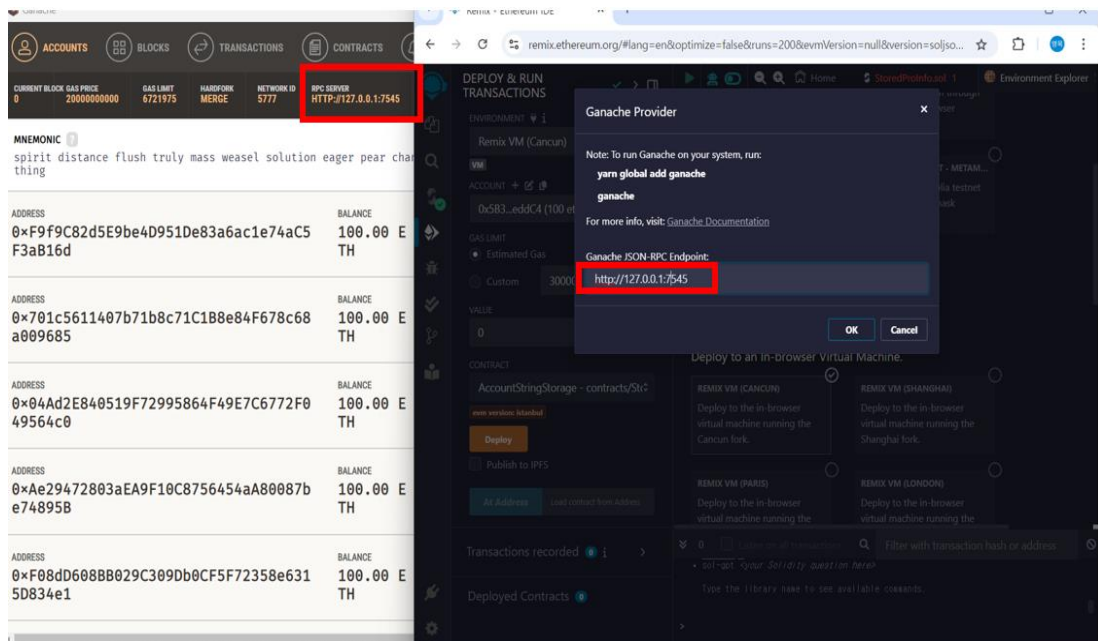


(2-5) 자신의 가상 네트워크에 REMIX IDE 연결 후 배포

ENVIRONMENT를 'Dev - Ganache Provider' 로 설정. (만약 WalletConnect에 Ganache Provider가 뜨지 않으면 맨 밑의 Customize this list에서 추가)

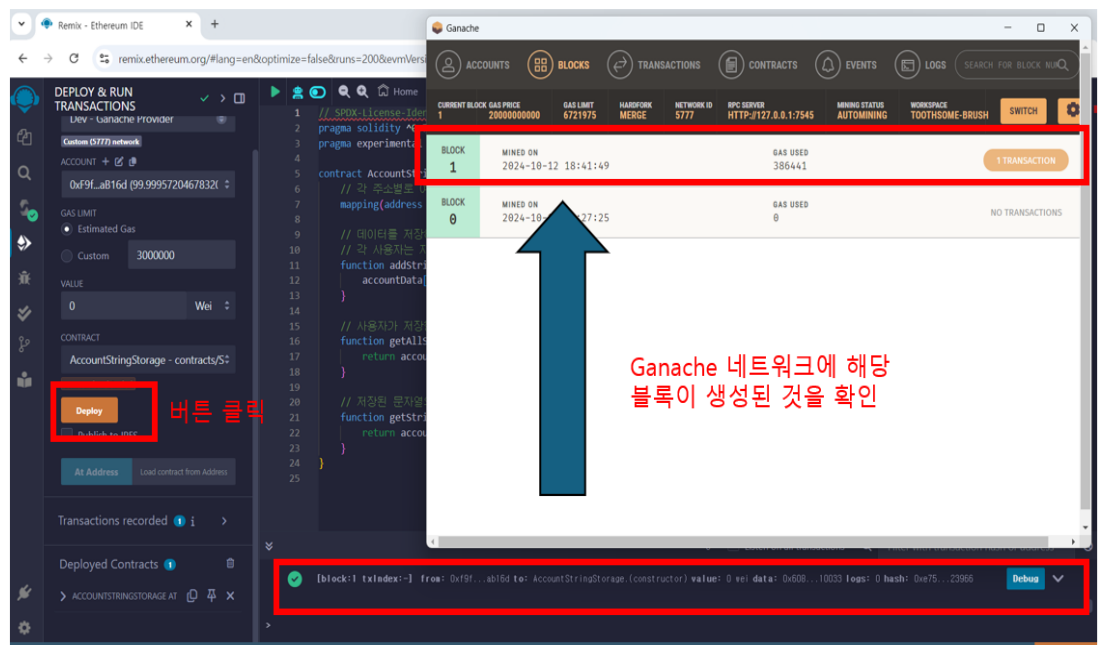


(2-6) 자신의 Ganache 네트워크 포트 번호에 맞춰서 연결



(2-7) 스마트 컨트랙트 배포

네트워크 연결 후 Deploy 버튼을 누르면 배포가 된다.



(2-8) 파일에 적힌 계약 주소를 자신의 네트워크에 배포된 계약 주소로 UPDATE

배포된 스마트 컨트랙트의 Address를 복사하여 프로젝트 파일 중 아래 목록을 참고하여 해당 파일들의 address 값들을 변경해준다. (/src/components)

- 각 smart contract 기능과 위치:

1. CoffeeProduction.sol (ProductionContractAddress)

기능: coffee 생산물 정보 저장

위치: ManageProduct.vue, TrackingPage.vue, DistributorDashboard.vue, ManagingPage.vue

2. OrderContract.sol (orderContractAddress)

기능: 주문 정보 저장

위치: CartTableCompo.vue, DistributorDashboard.vue, ManagingPage.vue, TrackingPage.vue

3. PaymentContract.sol (contractAddress)

기능: 결제 기능 (address값 변경 필요없음)

위치: CartTableCompo.vue,

4. StoredDelInfo.sol (StoredDelInfoContractAddress)

기능: 출고 정보 저장

위치: DistributorDashboard.vue, ManagingPage.vue

5. StoredProInfo.sol (StoredProInfoContractAddress)

기능: 생산자와 소비자 계좌에 본인이 생성한 TX Hash 값들 저장

위치: TrackinPage.vue, CartTableCompo.vue, ManageProduct.vue

6. StoredRecInfo.sol (StoredRecInfoContractAddress)

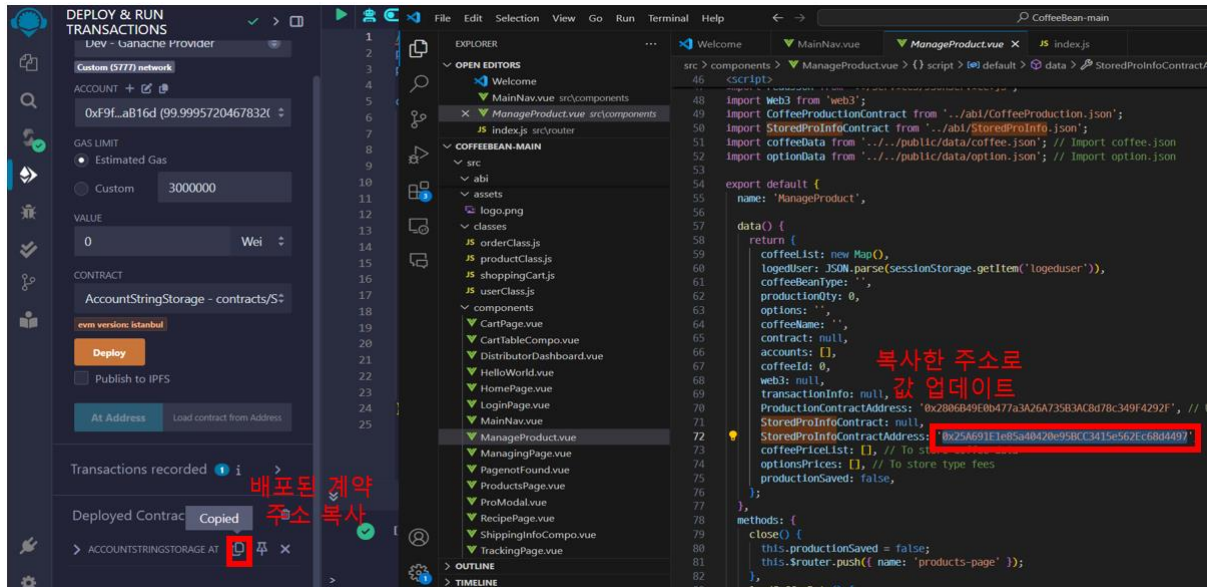
기능: 입고 정보 저장

위치: ManagingPage.vue, DistributorDashboard.vue

7. PaymentRecord.sol (paymentRecordAddress)

기능: ETH 출금, 입금 기록 저장

위치: DistributorDashboard.vue, ManagingPage.vue, CartTablCompo.vue



이렇게 모든 주소 값들을 변경해 준 후 프로젝트를 실행시켜서 주소 매칭이 잘되었는지 확인해본다.

6. 프로젝트 실행 후 동작 확인

모든 과정을 따라왔으면 `npm run serve` 명령어로 프로젝트를 실행시켜서 정상적으로 프로젝트의 기능들을 이용할 수 있다.