Lab 6: Solidity Mappings & Structs Tutorial

Name: Priyanka, stu-id: 20179

Step 1 & Step 2 & Step 3 & Step 4 & Step 5: Creating a Struct &

Mapping, Add & Get & Count from Mapping

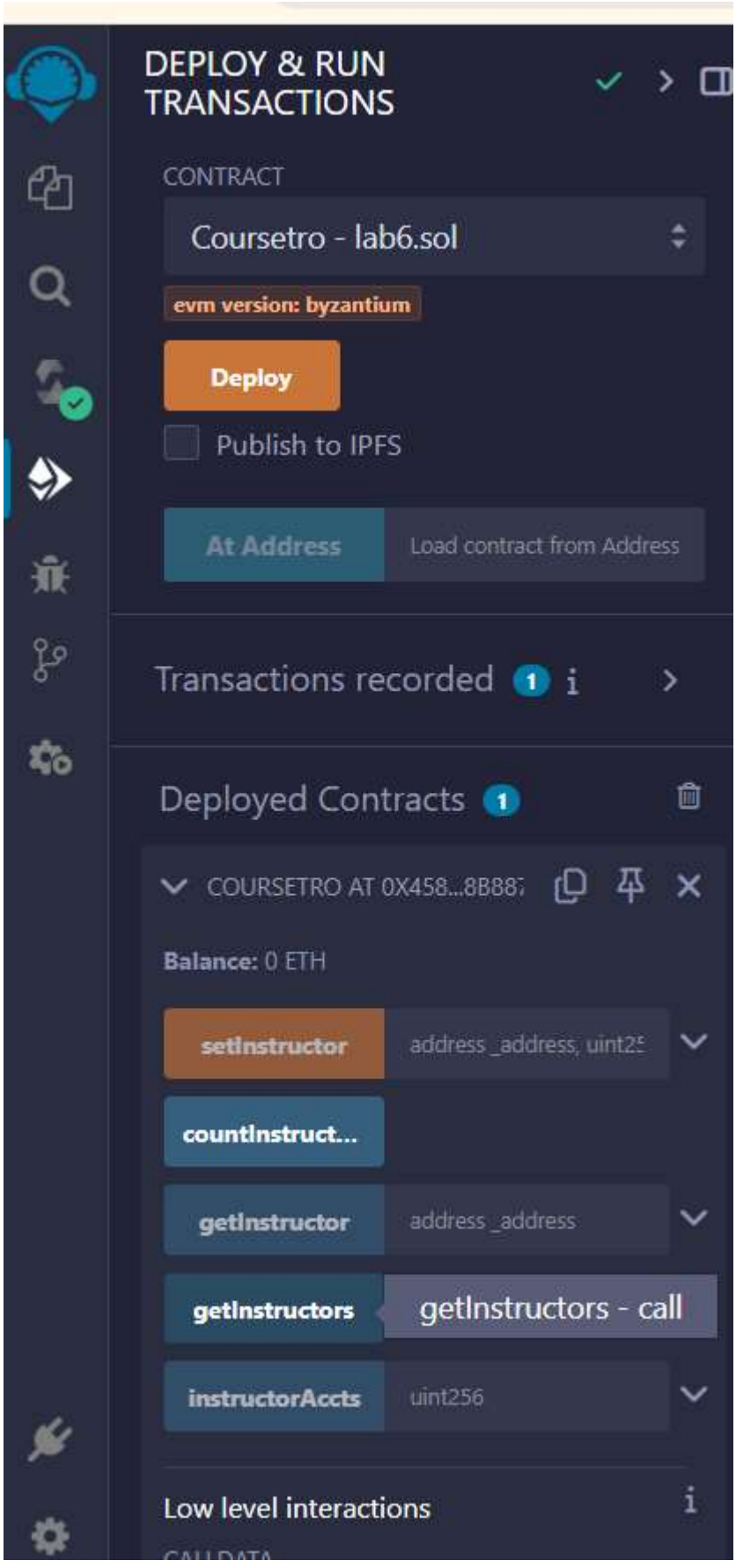
Go to Remix Ethereum IDE at http://remix.ethereum.org and create smart contract called lab6.sol

final lab6.sol content

```
pragma solidity ^0.4.18;

contract Coursetro {
struct Instructor {
uint age;
string fName;
string lName;
}
// Mapping Instructor struct to an Ethereum address.
mapping (address => Instructor) instructors;
address[] public instructorAccts;
func                                    uint _age, string memory _fName, string memory _lName) public {
Inst   address[] public instructorAccts  rs[_address];
inst
inst   lab6.sol 10:0
inst
instructorAccts.push(_address);
}
// Return a list of addresses from instructorAccts
function getInstructors() view public returns (address[] memory) {    infinite gas
return instructorAccts;
}
// Retrieve a specific instructor based on a provided address
function getInstructor(address _address) view public returns (uint, string memory, string memory) {
return (instructors[_address].age, instructors[_address].fName, instructors[_address].lName);
}
// Count how many instructors
function countInstructors() view public returns (uint) {    498 gas
```

compile and deploy the contract


DEPLOY & RUN TRANSACTIONS
CONTRACT
Coursetro - lab6.sol
evm version: byzantium
Deploy
Publish to IPFS
At Address
Load contract from Address
Transactions recorded 1
Deployed Contracts 1
COURSETRO AT 0X458...8B887
Balance: 0 ETH
setInstructor
address _address, uint25
countInstruct...
getInstructor
address _address
getInstructors
getInstructors - call
instructorAccts
uint256
Low level interactions

test the methods like this, the account address was chose from the ganache available accounts when you start the RPC server