Lab 3

Part 1:

Done in Repository on GitHub.

Part 2:

here's a basic example of unit tests for a lending contract in Solidity. We assume that you have a lending contract named LendingContract:

Part 3:

Explanation of Ganache-Time-Traveller:

Ganache-Time-Traveller is a feature in Ganache that allows you to manipulate the Ethereum block timestamp during development and testing. This feature is actually particularly useful in order to test time-dependent functions in your smart contracts without having to wait for actual time to pass.

For instance, suppose you have a smart contract with a time-dependent function like a vesting period or an interest calculation. You are able to use Ganache-Time-Traveller to fast-forward time to test how the contract behaves at different points in the future.

≡ WORKSPACES

default_workspace ⇕

📁 .deps
📁 contracts
📁 artifacts
⚡ 1_Storage.sol
⚡ 2_Owner.sol
⚡ 3_Ballot.sol
⚡ AssemblyLoops.sol
⚡ Assignment 1 cont.sol
⚡ Assignment 1.sol
⚡ Enum.sol
⚡ ERC20.sol
⚡ Escrow.sol
⚡ Exam prep.sol
⚡ Lab 1draft.sol
⚡ lab 1 Submission.sol
⚡ Lab 2 Advanced.sol
⚡ Lab 2 part 2.sol
⚡ lab 2 Submission.sol
⚡ Lab 3 draft.sol   1
⚡ Lab 4 draft.sol
⚡ Lab 5 draft.sol
⚡ MyArray.sol
⚡ MyLastWill.sol
⚡ MyMultiSig.sol
⚡ MyWhiteList.sol
⚡ test.sol
⚡ test2.sol
⚡ TestArray.sol
⚡ Todos.sol

```javascript
1   const { time } = require('@openzeppelin/test-helpers');
2
3   // ...
4
5   it("should release vested tokens after a certain time", async () => {
6       const contract = await MyVestingContract.deployed();
7
8       // This is to Get the current timestamp
9       const currentTimestamp = await time.latest();
10
11      // This is to Move forward in time by 30 days
12      await time.increase(time.duration.days(30));
13
14      // Thiss is to Perform actions and assertions related to the vesting period
15      // ...
16  });
17
```

✔ listen on all transactions   🔍 Search with transaction hash or address

• ethers.js
• remix

Type the library name to see available commands.

>