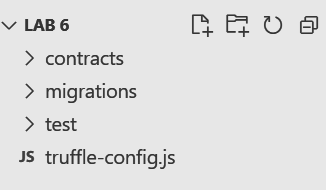
**Step 1:** Initialize your project



**Step 2:** Once this operation is complete, you’ll now have a project structure with the following items:



**Step 3:** Create a file named “myToken.sol” in the contract directory.

// SPDX-License-Identifier: GPL-3.0

pragma solidity 0.5.16;

import "@openzeppelin/contracts/token/ERC20/IERC20.sol";

contract Mikil is IERC20{

string private \_name;

string private \_symbol;

uint256 private \_decimals;

uint256 private \_totalSupply;

mapping (address => uint) private \_balances;

mapping (address => mapping(address => uint)) private \_allowances;

constructor(){

\_name = "Mikil";

\_symbol = "MKL";

\_decimals = 18;

\_totalSupply = 1000000 \* 10\*\*\_decimals;

\_balances[msg.sender] = \_totalSupply;

}

function name() public view returns(string memory){

return \_name;

}

function symbol() public view returns(string memory){

return \_symbol;

}

function decimals() public view returns(uint256){

return \_decimals;

}

function totalSupply() public view returns(uint256){

return \_totalSupply;

}

function balanceOf(address account) public view returns(uint256){

return \_balances[account];

}

function transfer(address recipient, uint256 amount) public returns(bool){

require(balanceOf(msg.sender) >= amount, "Lack of Funds.");

\_balances[msg.sender] -= amount;

\_balances[recipient] += amount;

return true;

}

function allowance(address owner, address spender) public view returns(uint256){

return \_allowances[owner][spender];

}

function approve(address spender, uint256 amount) public returns(bool){

\_allowances[msg.sender][spender] = amount;

return true;

}

function transferFrom(address sender, address recipient, uint256 amount) public returns(bool){

require(\_allowances[msg.sender][sender]>=amount);

\_balances[sender] -= amount;

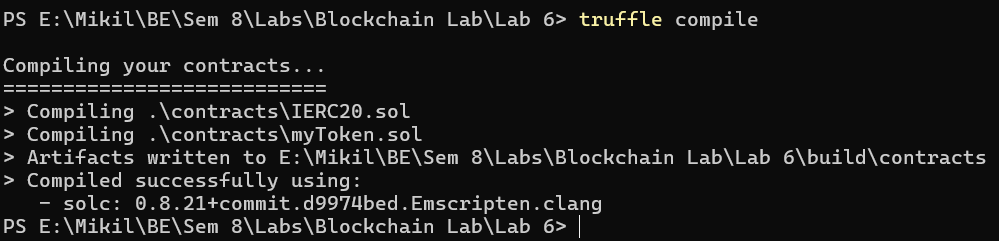
\_balances[recipient] += amount;

return true;

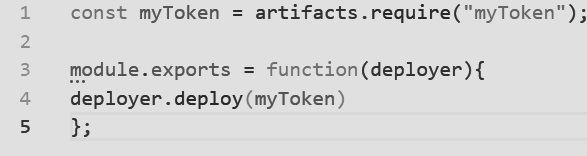
}

}

**Step 4:** Compile a Truffle project



**Step 5:** Create a migration to get the contract on the network. Create a file in the migrations folder named “37\_Mikil\_migrate\_deploy.js”.



**Step 6:** The ERC20 Token Smart contract will be deployed



