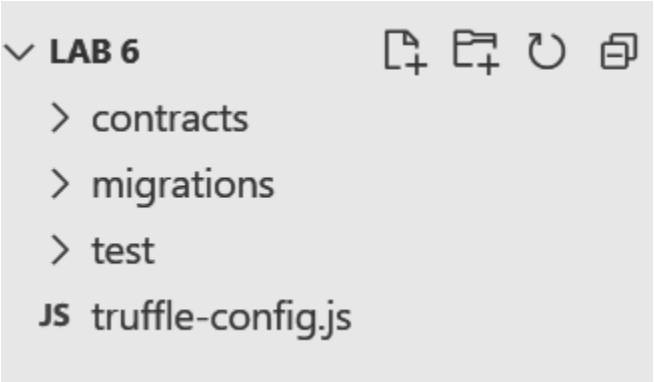
Step 1: Initialize your project

```
PS E:\Mikil\BE\Sem 8\Labs\Blockchain Lab\Lab 6> truffle init

Starting init...
================

> Copying project files to E:\Mikil\BE\Sem 8\Labs\Blockchain Lab\Lab 6

Init successful, sweet!

Try our scaffold commands to get started:
  $ truffle create contract YourContractName # scaffold a contract
  $ truffle create test YourTestName         # scaffold a test

http://trufflesuite.com/docs

PS E:\Mikil\BE\Sem 8\Labs\Blockchain Lab\Lab 6>
```

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

```
PS E:\Mikil\BE\Sem 8\Labs\Blockchain Lab\Lab 6> truffle compile

Compiling your contracts...
===========================
> Compiling .\contracts\IERC20.sol
> Compiling .\contracts\myToken.sol
> Artifacts written to E:\Mikil\BE\Sem 8\Labs\Blockchain Lab\Lab 6\build\contracts
> Compiled successfully using:
   - solc: 0.8.21+commit.d9974bed.Emscripten.clang
PS E:\Mikil\BE\Sem 8\Labs\Blockchain Lab\Lab 6>
```

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”.

```
const myToken = artifacts.require("myToken");

module.exports = function(deployer){
deployer.deploy(myToken)
};
```

Step 6: The ERC20 Token Smart contract will be deployed

```
PS E:\Mikil\BE\Sem 8\Labs\Blockchain Lab\Lab 6> truffle migrate --network development
```

```
Starting migrations...
======================
> Network name:    'development'
> Network id:      5777
> Block gas limit: 6721975 (0x6691b7)


37_Mikil_migrate_deploy.js
==========================

   Deploying 'myToken'
   -------------------
   > transaction hash:    0x1ca320bde88df659c2ebe4502c7e48930b80a105ad49b8acc04a61e4c6b55329
   > Blocks: 0            Seconds: 0
   > contract address:    0x17727B8770ad93904837Be111ABaD56b7fae1EE5
   > block number:        1
   > block timestamp:     1711859691
   > account:             0x59Ac7EbeABDd781f546DeA354852B0AC130259A2
   > balance:             99.997703653375
   > gas used:            680399 (0xa61cf)
   > gas price:           3.375 gwei
   > value sent:          0 ETH
   > total cost:          0.002296346625 ETH

   > Saving artifacts
   -------------------------------------
   > Total cost:      0.002296346625 ETH

Summary
=======
> Total deployments:   1
> Final cost:          0.002296346625 ETH
```