**truffle-config.js:**

* Import “babel” to get es6 features
* Import “dotenv” config to configure environment
* Configure contracts and abis (Json version of Smart Contracts) dir where truffle should look for
* Configure blockchain network to which app should connect
* Configure solidity compiler version
  + Solidity is a compiled language, all contracts src will be compiled into byte code that can be executed on Ethereum VM running on Ethereum node. Truffle is responsible for this.
* Compile smart contract to byte code using Solidity to run on Ethereum VM
  + ../capstone-project> **truffle compile**
* Deploy smart contract to block chain using migration. It is changing the state of blockchain to another when a new transaction is written to blockchain. Below command executes all defined migrations in the same order
  + ../capstone-project> **truffle migrate** (Automatically compiles too and deploy)
* Truffle console opens a node js console and loads truffle project to interact anything with truffle
  + ../capstone-project> truffle console
  + Get deployed copy of “Token” (smart contract) from blockchain in asynchronous mode using await
    - truffle (development)> **const token = await Token.deployed()**
    - truffle (development)> token.address
    - truffle (development)> const name = **await token.name()**
* Truffle allows to test smart contracts, gives testing framework based on java script and it uses Mocha testing framework and Chai assertion library to do that.
  + truffle (development)> **truffle test**