1.

Blockchain.js :

- Addblock() – Add a block(a class in block.js which is explained below) to the array of blocks (I.e. hence chain of blocks).

- getNextBlock()- The getNextBlock method creates a new block in the blockchain with the specified transactions, setting its index and previous hash values, generates the hash for a block that meets the target.

- generateHash() -The generateHash function takes a Block object and generates a SHA256 hash value with a specific difficulty target by incrementing the nonce property until a hash value that starts with three zeros is found. The function then returns the valid hash value.

- getPreviousBlock()- This method simply returns previous block object using array

Transaction.js :

This file only contains the constructor with parameters to, from, amount which creates a new transaction class object.

Block.js :

- key() - The key method returns a string that represents the contents of the block. It concatenates the JSON string representation of the transactions array, the index property, the previousHash property, and the nonce property of the block into a single string.

- addTransaction()- The addTransaction method takes a Transaction object as its input parameter and adds it to the transactions array of the block.

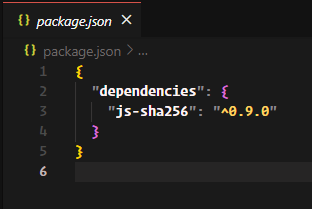
2.

In blockchain, the genesis block is the first block of a blockchain. It is unique as it does not reference any previous block, since there is no block before it. Therefore, the previous hash of the genesis block is set to a fixed value, usually "0000000000000000" or a similar value, to indicate that there is no previous block.

By setting the previous hash to a fixed value, it ensures that the genesis block is the first block in the blockchain, and any subsequent blocks can reference it as the previous block. Additionally, it also ensures that the hash of the genesis block can be used as the starting point for calculating the hash of the next block in the chain.

3.

Yes, it does show the dependency, for reference the below is the screenshot of dependency



4.

The first line of the Blockchain.js file is importing the js-sha256 module, which is a third-party library that provides an implementation of the SHA-256 cryptographic hash function in JavaScript.

The js-sha256 module provides a simple way to generate SHA-256 hashes, which are commonly used in blockchain applications to secure the data and ensure that it has not been tampered with.

By importing the js-sha256 module, the Blockchain class is able to use the sha256 function provided by the module to generate hashes for each block in the blockchain, as well as for other purposes such as verifying transactions and validating the integrity of the blockchain.

5.

When running the code with the initial difficulty target of finding a hash value that starts with three zeroes (i.e., hash.startsWith("000")), the task took a few seconds to complete.

6.

I changed to amount to two zeroes: Took less a second to complete

Increased the number of zeroes to 3: Took about 30 to 45 seconds to complete

Further, changed it to 7: Computer could compute as it is connected to power, took around 5 mins to produce the required hash.

Further changed to 15: Computer could not compute, and my terminal crashed

If we increase this to something above 50 zeroes, home desktops and laptops would probably not be able to solve even one problem due to hardware limitations.

7.

Overview of the project gives us idea on concept of blockchain, architecture briefly. Used double linked list in node js application as representation of how block chain works, also methods that are defined in multiples files what contracts do.