Lab -1

1) The first question you will have to answer is explaining JS event loop, clarifying the synchronous and asynchronous parts, queues, priority, with examples. Write your answer as if it were for a real interview.

   JS is a single-threaded synchronous programming language, which means that it can only run one command at a time. So those synchronous commands can actually be blocked. In order to avoid blocking the main thread, asynchronous js comes along with things like callbacks, and async wait, etc.
   The event loop is responsible for checking the call stack in the callback queue or Macrotask Queue (event queue). If all the statements from the call stack have finished, the event loop will pop the statement from the callback queue and push it to the call stack.
   Asynchronous means more than one process running simultaneously. Synchronous means one process is executed at a time. Synchronous code is always executed first before asynchronous code.

```
console.log("1st Function") //1 sync

setTimeout(() => { //Web API
    console.log("2nd Function") //4 async
}, 1000);

console.log("3rd Function") //2 sync

const promise = new Promise((resolve, reject) => {
    console.log("Promise"); //3 sync
    setTimeout(() => {
        resolve("Async Promise") //5 async
    }, 1000)
})

promise.then((result) => {
    console.log(result) //5 async
})
```

   //output
   1st Function
   3rd Function
   Promise
   2nd Function
   Async Promise

2) The follow up question is how may we convert a sync operation/function to become asynchronous?

   An operation/function to become asynchronous by using one of the following ways:

   1. setTimeout()
   2. setImmediate()
   3. setInterval()
   4. Promise
   5. Async/await
   6. Fetch API