Aim:

a.
   1. Installation and Configuration of Node Js.
   2. Learning about Call Backs and Event Loops in Node Js

b. Creating an Express Application.

Theory:

NODE:

Node.js (Node) is an open-source development platform for executing JavaScript code server-side. Node is useful for developing applications that require a persistent connection from the browser to the server and is often used for real-time applications such as chat, news feeds and web push notifications.

Node.js is intended to run on a dedicated HTTP server and to employ a single thread with one process at a time. Node.js applications are event based and run asynchronously. Code built on the Node platform does not follow the traditional model of receive, process, send, wait, receive. Instead, Node processes incoming requests in a constant event stack and sends small requests one after the other without waiting for responses. It interprets JavaScript code via Google's V8 JavaScript engine. It interprets JavaScript code via Google's V8 JavaScript engine.

Steps To Install Node Js:

1. Download the Node.js '.msi' installer from: https://nodejs.org/en/download/
2. Run the Node.js installer. Install it. Verify By typing 'node -v' in CMD

Event Loops –
The event loop is what allows Node.js to perform non-blocking I/O operations (despite the fact that JavaScript is single-threaded) by offloading operations to the system kernel whenever possible.

Callback –
A callback is a function which is called when a task is completed, thus helps in preventing any kind of blocking and a callback function allows other code to run in the meantime. Callback is called when task get completed and is asynchronous equivalent for a function. Using Callback concept, Node.js can process a large number of requests without waiting for any function to return the result which makes Node.js highly scalable.

EXPRESS:

It's a web framework that lets you structure a web application to handle multiple different http requests at a specific URL. Express is a minimal, open source and flexible Node.js web app framework designed to make developing websites; web apps, & APIs much easier.

- Express helps to respond to requests with route support so that you may write responses to specific URLs
- It Supports multiple templating engines to simplify generating HTML

Step to Install Express:

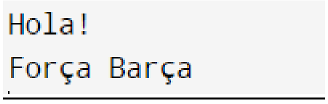
1. Install Node JS first. Check For npm package.
2. Open Vs Code & in terminal type ‘npm install express ‘
3. Express will be installed on the respective device

Code & Output:

5A

HelloWorld.js:

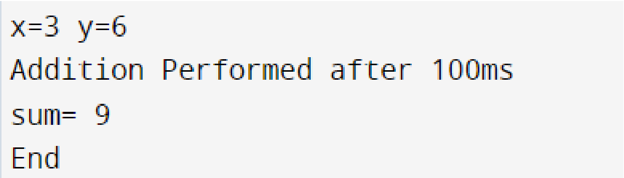
```
setTimeout(()=>{console.log('Força Barça')},0);

console.log('Hola!');
```



```
Hola!
Força Barça
```

CallBack.js:

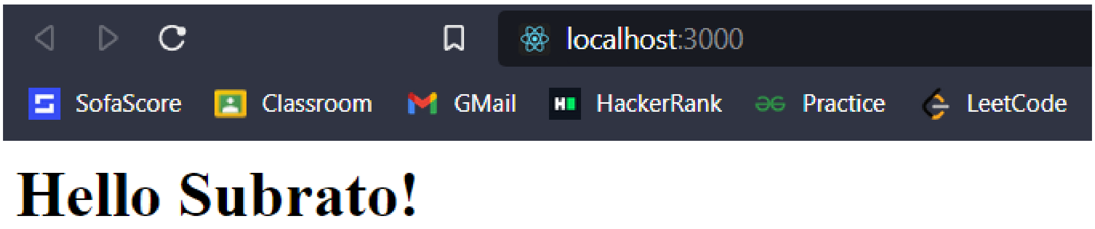
```
cal = function (x, y) {
  console.log("x=" + x, "y=" + y);
  let sum = add(x, y, function (result) {
    console.log("sum= " + result);
    console.log("End");
  });
};
add = function (x, y, callback) {
  setTimeout(function () {
    console.log("Addition Performed after 100ms");
    callback(x + y);
  }, 100);
};
cal(3, 6);
```



```
x=3 y=6
Addition Performed after 100ms
sum= 9
End
```

5B

```
const express = require('express')
const app = express()
const port = 3000

app.get('/', (req, res) => {
  res.send('<h1>Hello Subrato!</h1>')
})

app.listen(port, () => {
  console.log(`Example app listening on port ${3000}`)
})
```



Hello Subrato!

Conclusion:

Thus, we learnt:

a. To Install and Configure Node Js.
b. About Event Loops and Call Backs in Node Js.
c. To Install and Configure Express.
d. To Create Express Application