Assignment 3 – Part 3: Node Internals

1. What is the Node.js Event Loop?
The Event Loop is the mechanism that allows Node.js to perform non-blocking asynchronous operations. It monitors the Call Stack and pushes callbacks from the Event Queue to the Call Stack when it becomes empty.

2. What is Libuv and what role does it play?
Libuv is a C library used internally by Node.js to handle asynchronous operations such as file system, networking, timers, and the thread pool. It is responsible for managing the Event Loop.

3. How does Node.js handle asynchronous operations under the hood?
Node.js sends asynchronous tasks to Libuv. Libuv executes them using the operating system or the thread pool. Once completed, the callback is added to the Event Queue and executed by the Event Loop.

4. Difference between Call Stack, Event Queue, and Event Loop
Call Stack executes the current JavaScript code.
Event Queue holds completed callbacks waiting for execution.
Event Loop moves callbacks from the queue to the stack when the stack is empty.

5. What is the Node.js Thread Pool and how to set its size?
The thread pool is used for heavy operations like file system and crypto tasks. The default size is 4 threads and it can be changed using the UV_THREADPOOL_SIZE environment variable.

6. Blocking vs Non-Blocking execution
Blocking code stops execution until the task finishes. Non-blocking code allows Node.js to continue running while tasks execute asynchronously.