SLIIT

COMPUTING | BUSINESS | ENGINEERING

SE3040 – Application Framework

Tutorial – Node JS

01. How does Node JS differ from JavaScript?

JavaScript: Runs in the browser. used for frontend development, DOM manipulation, event handling, and animations.
.

Node.js: Runs outside the browser. backend development, handling databases, APIs, file systems, and building web servers.

02. What is the event-driven model in Node JS?

programming paradigm where the flow of execution is determined by events, such as user interactions , I/O operations, or messages from other processes. Instead of executing code sequentially, Node.js listens for events and triggers corresponding callbacks when those events occur.

03. What are the key components of event-driven model?

Event, Event Producer, Event Listener, Event Handler, Event Loop, Callback/Subscriber, Message Queue, Middleware/Event Broker

04. Write a simple event driven program using node JS?

```
const EventEmitter = require('events');

const eventEmitter = new EventEmitter();

eventEmitter.on('message', () => {
   console.log('New message received!');
});

eventEmitter.emit('message');
```

05. What is an event loop and what does an event loop demonstrate in node JS?

The event loop is a core mechanism in Node.js that allows non-blocking, asynchronous operations. It continuously listens for events and executes queued tasks efficiently using a single-threaded, event-driven architecture.

Timers (setTimeout, setInterval)
I/O operations (File system, Network requests)
Promises & Async/Await
Event Listeners

06. Explain the role of Node Package Manager (npm) in managing dependencies in a Node.js project?

npm (Node Package Manager) is used to install, update, and manage dependencies in a Node.js

Installing Packages, Tracking Dependenciesm, Managing Versions,Local vs Global Install,

Updating & Removing Packages, Running Scripts, Managing Dev vs Prod Dependencies

07. Describe the different ways to install modules in Node.js, and explain how Node.js resolves installed modules when required in a project?

Local Installation (Default) - Installs the package in the node_modules/ directory of the current project. -npm install express

Global Installation - Installs the package system-wide and makes it accessible anywhere. - npm install -g nodemon

Installing Specific Versions - Install a particular version of a package. - npm install express@4.17.1

08. Explain how require() function works in Node.js, including how it handles module caching and different module types?

Resolve Module Path

****

Checks if the module is a core (built-in) module (e.g., fs, http).
Searches in the node_modules/ directory for installed third-party modules.
If a relative or absolute path is provided, it looks for the module in that location

Load the Module

If the module is a JavaScript file, Node.js executes it.
If it's a JSON file, Node.js parses it into an object.
If it's a C++ addon, it loads it as a binary file.

Cache the Module

Node.js stores the module in memory after the first load.
If the same module is required again, it is retrieved from the cache instead of being reloaded.
This improves performance by avoiding redundant executions.