Full Stack - I - Lab Assignment Question

1. Assignment 1: Create a Node.js script demonstrating variables, functions, conditionals, loops, objects, arrays, and asynchronous code.

   A. Aim/Purpose of the Experiment:
   i. To develop a Node.js script that demonstrates the use of:
   o Variables
   o Functions
   o Conditionals
   o Loops
   o Objects
   o Arrays
   o Asynchronous programming
   B. Learning Outcomes:
   i. Understand the fundamentals of Node.js.
   ii. Implement core JavaScript concepts in a server-side environment.
   iii. Learn how asynchronous operations work in Node.js.
   iv. Gain hands-on experience with functions, conditionals, loops, objects, and arrays.
   C. Prerequisites:
   i. Basic understanding of JavaScript.
   ii. Familiarity with the Node.js runtime environment.
   iii. Knowledge of asynchronous programming concepts (callbacks, promises, async/await).
   D. Materials/Equipment/Apparatus/Devices/Software Required:
   i. Computer with Node.js installed (latest LTS version recommended).
   ii. Code editor (e.g., VS Code, Sublime Text, or any preferred IDE).
   iii. Terminal/Command prompt for executing Node.js scripts.
   E. Introduction and Theory:
   i. Node.js is a runtime environment that allows JavaScript to run on the server side.
   ii. It is non-blocking and event-driven, making it efficient for handling asynchronous operations.
   iii. The core concepts in this experiment include:
   o Variables: Used to store data values.
   o Functions: Blocks of reusable code.
   o Conditionals: Decision-making structures like if-else.
   o Loops: Iterative execution (for, while, etc.).
   o Objects & Arrays: Data structures for storing multiple values.
   o Asynchronous Code: Handling operations without blocking execution.
   F. Operating Procedure:
   i. Setup Node.js Environment:

o Install Node.js.
o Initialize a new project (optional: npm init).
ii. Create a Script (script.js) that includes:
o Declaring variables.
o Writing functions.
o Using conditionals and loops.
o Creating and manipulating objects and arrays.
o Implementing asynchronous operations using callbacks, promises, and async/await.
iii. Run the script using node script.js and analyze the output.
G. Precautions and/or Troubleshooting:
i. Ensure Node.js is installed correctly (node -v to check version).
ii. Use console.log() for debugging.
iii. Handle errors in asynchronous functions properly with try-catch or .catch().
H. Observations:
i. Note the behavior of synchronous vs. asynchronous code.
ii. Observe how loops, conditionals, and functions work in Node.js.
iii. Compare execution time for synchronous and asynchronous tasks.
I. Calculations & Analysis:
i. Analyze execution order in asynchronous code.
ii. Evaluate memory and performance usage for loops and data structures.
J. Result & Interpretation:
i. Successfully demonstrated fundamental Node.js concepts.
ii. Understood the working of asynchronous programming in Node.js.
K. Follow-up Questions:
i. What are the advantages of using asynchronous programming in Node.js?
ii. How do promises differ from callbacks?
iii. Why is Node.js single-threaded, and how does it handle concurrency?
L. Extension and Follow-up Activities (if applicable):
i. Extend the script to include file system operations using the fs module.
ii. Implement an HTTP server using the http module.
M. Assessments:
i. Write a script that reads a file asynchronously and processes the data.
ii. Modify the script to include exception handling and logging.
N. Suggested Reading:
i. Official Node.js Documentation
ii. Node.js Design Patterns by Mario Casciaro

Assignment 1: Create a Node.js script demonstrating variables, functions, conditionals, loops, objects, arrays, and asynchronous code.

Assignment 2: Node.js Programs for User Input and System Variables

Assignment 3: Command-Line Arguments & REPL in Node.js

Assignment 4: Write a Node.js Script for the REPL Environment, Including Functions, Array Methods, and Asynchronous Operations

Assignment 5: Write a Node.js Program to Calculate the Execution Time of Multiple Functions

Assignment 6: Write a Node.js Program to Explain Asynchronous and Synchronous Functions with Examples

Assignment 7: Core Path and FS Modules in Node.js

Assignment 8: Demonstrating the Use of Buffers in Node.js

Assignment 9: Describing Local Modules in Node.js

Assignment 10: Create a package.json file for a Node.js project and describe its setup with an external package like dotenv

Assignment 11: Create a Server Using the HTTP Core Module and Handle Routes with Different HTTP Methods

Assignment 12: Perform CRUD Operations on a File Using the FS Core Module in Node.js

Assignment 13: Create a Node.js Project for a Calculator with Multiple Routes

Assignment 13: Create a Node.js Project for a Calculator with Multiple Routes

Assignment 14: Create a Node.js Project Using Express to Respond with HTML and JSON

Assignment 15: Extend the Express Project