Introduction To NodeJs

For week 9, we'll be diving into **Introduction to Nodes**, where we'll explore what nodes are, their role in data structures, and how they connect to form structures like linked lists, trees, and graphs.

By the end of the week, you'll understand how nodes hold and manage data, connect to other nodes, and enable efficient organization in systems like databases and networks.

Let's look at the breakdown of key topics and tasks ahead! How does this sound as an intro? We can now work on learning objectives and tasks.

Software Development Week 9

Learning Objectives for the week

At the end of this week you should be able to;

- **Understand Node.js Architecture**: Explain the fundamental concepts of Node.js, including its event-driven architecture and non-blocking I/O model, and how they differ from traditional server-side technologies.
- **Set Up Development Environment**: Install Node.js and use basic command-line tools to set up a Node.js development environment on your local machine.
- **Module Management with npm**: Learn to manage modules using npm, including searching for, installing, uninstalling, and managing package dependencies.
- **Build Web Servers**: Develop simple web servers using Node.js core modules and popular frameworks like Express.js, handling HTTP requests and responses.
- **Create APIs with Node.js**: Build basic APIs that handle data requests and responses in JSON format, implementing core API functionalities.

Software Development Week 9

Online Learning Modules

This is the online module that you have to complete this week.

- Introduction to Nodejs - Full Course: Click Here 1
- Snippet on Nodejs: Click Here 2

Additional Resource links
1. Click here - Text
2. Article - Text
3. Download - Ebook

Software Development Week 9

Weekly Applied Learning Assignment

Build a Simple API with Node.js

1. **Instructions:**
    - **Set Up Your Environment**: Install Node.js on your local machine and initialize a new project using npm.
    - **Create a Web Server**:
        - Using Node.js and Express.js, create a basic web server that listens on a specified port.
        - Your server should respond with a welcome message when a user accesses the root URL ("/").
    - **Build an API**:
        - Create a simple API with two endpoints:
            - A **GET** request that returns a list of fictional users in JSON format.
            - A **POST** request that allows you to add a new user to the list.
    - **Handle Errors**:
        - Implement error handling for invalid requests (e.g., when a required field is missing or the request format is incorrect).

Submission:

- Submit your code and summary via CodePen.