**Module 64-Getting Started with Node, Express and Api**

**GETTING STARTED WITH NODE, EXPRESS AND API**

* Node express and install nodemon
* Dynamic api, api parameter, access params
* Query parametres and return search result
* Load data from server
* Middleware, handle cors
* Post API and Operations
* Git ignore foe node project

**MONGODb, DATABASE INTEGRATION, CRUD**

* Mongodb vs MySql
* Operaions with cloude database
* Save data to database and load from database
* CRUD operation

**JWT TOKEN**

* Introduction JWT token
* JWT token in client side
* JWT token in the server
* Filter user by email address
* Setup firebase admin

**YOU WILL ALSO HAVE**

* Node, Mongo , Crud Practise with defferent project
* HTTP methods, Get VS POST
* Knowledge about pagination
* Server deploy to heroku
* Image hosting

64-1 Module overview Why Node, node vs other language

**What is node js bangla?**

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser.

# Node.js কি?

অফিসিয়াল সাইটের সংজ্ঞা অনুযায়ী — “It is a JavaScript runtime built on Chrome’s V8 JavaScript engine. It uses an event-driven, non-blocking I/O model that makes it lightweight and efficient” অর্থাৎ — ক্রোমের V8 জাভাস্ক্রিপ্ট ইঞ্জিনের উপর তৈরি একটি জাভাস্ক্রিপ্ট রানটাইম যা কিনা ইভেন্ট-ড্রাইভেন ইনপুট/আউটপুট মডেল ব্যবহার করে এবং এর কারনে এটি অনেক ইফিসিয়েন্ট। উইকিপিডিয়াতে বলা — “Node.js is an open-source, cross-platform JavaScript run-time environment for executing JavaScript code server-side.” এটি একটি ওপেন সোর্স, ক্রস প্ল্যাটফর্ম জাভাস্ক্রিপ্ট রানটাইম এনভ্যায়র্নমেন্ট যার মাধ্যমে সার্ভার সাইডে জাভাস্ক্রিপ্ট কোড এক্সিকউট করা যায়” আমার যে সংজ্ঞাটি পছন্দ সেটা হচ্ছে — এটি একটি ওপেন সোর্স, কোর্স প্ল্যাটফর্ম, ইভেন্ট ড্রাইভেন রানটাইম এনভ্যায়র্নমেন্ট যার মাধ্যমে জাভাস্ক্রিপ্ট ব্যবহার করে এবং নন-ব্লকিং ইনপুট/আউটপুট মডেল ফলো করে সার্ভার সাইড, হাইব্রিড বা নেটওয়ার্কিং অ্যাপ্লিকেশন ডেভেলপ করা যায় :)

**কেন জাভাস্ক্রিপ্ট?** — Node.js এর ক্রিয়েটর Ryan Dahl এর মতে -

“JavaScript has certain characteristics that make it very different than other dynamic languages, namely that it has no concept of threads. Its model of concurrency is completely based around events.”

# Difference between Node.JS and Javascript:

| S.No | Javascript | NodeJS |
| --- | --- | --- |
| 1. | Javascript is a programming language that is used for writing scripts on the website. | NodeJS is a Javascript runtime environment. |
| 2. | Javascript can only be run in the browsers. | We can run Javascript outside the browser with the help of NodeJS. |
| 3. | It is basically used on the client-side. | It is mostly used on the server-side. |
| 4. | Javascript is capable enough to add HTML and play with the DOM. | Nodejs does not have capability to add HTML tags. |
| 5. | Javascript can run in any browser engine as like JS core in safari and Spidermonkey in Firefox. | V8 is the Javascript engine inside of node.js that parses and runs Javascript. |
| 6. | Javascript is used in frontend development. | Nodejs is used in server-side development. |
| 7. | Some of the javascript frameworks are RamdaJS, TypedJS, etc. | Some of the Nodejs modules are Lodash, express etc. These modules are to be imported from npm. |
| 8. | It is the upgraded version of ECMA script that uses Chrome’s V8 engine written in C++. | Nodejs is written in C, C++ and Javascript. |

# Difference between Node.js and React.js

| Node.js | React.js |
| --- | --- |
| Node.js used as a back-end framework | React is used for developing user interfaces. |
| It supports the Model–view–controller (MVC) framework. | Does not support the Model–view–controller (MVC) framework. |
| It runs on chrome’s v8 engine and uses an event-driven, non-blocking I/O model, which is written in C++. | It uses Node.js to compile and optimize the JavaScript code and easy to create UI Test cases. |
| Node.js handles requests and authentication from the browser, make database calls, etc. | It makes API calls and processes in-browser data. |
| Here the Real-time data streaming is handled easily. | In React complex architecture makes it hard to keep track of the traditional approach. |
| Framework for JavaScript execution having the largest ecosystem of open source libraries. | Facebook-backed Open Source JS library. |
| The language used whichonly JavaScript. | The language used is JSX and JavaScript. |
| There is no DOM (Document Object Model) concept that is Used. | Here the Virtual DOM (Document Object Model) is Used that makes it faster. |

**When should i use node.js?**

Node. js is primarily used for **non-blocking, event-driven servers**, due to its single-threaded nature. It's used for traditional web sites and back-end API services, but was designed with real-time, push-based architectures in mind.

#### 64-2 Getting started with Node and Express, your first api

**Link : <https://expressjs.com/en/starter/hello-world.html>**

const express = require("express");

const app = express();

const port = process.env.PORT || 5000;

app.get("/", (req, res) => {

  res.send("Hello World!");

});

app.listen(port, () => {

  console.log(`Example app listening on port ${port}`);

});

#### 64-3 (Recap) Node express and install nodemon for auto restart

### node server auto run

1. npm install -g nodemon # or using yarn: yarn global add nodemon
2. optional (npm install --save-dev nodemon)
3. pakage.json -> script-> "start": "node index.js", "start-dev": "nodemon index.js",
4. cmd -> nodemon index.js

#### 64-4 Create dynamic api, api parameter, access params

#### 64-5 Use fetch to load data from server, middleware, handle cors

#### 64-6 Create React form and Post API and send data to the server

  const handleAddUser=(event)=>{

    event.preventDefault();

    const name = event.target.name.value;

    const email = event.target.email.value;

    // console.log(name,email);

    const user = { name, email };

    // post data to server

      fetch('http://localhost:5000/user', {

      method: 'POST',

      headers: {

        'content-type': 'application/json'

      },

      body: JSON.stringify(user)

    })

      .then(res => res.json())

      .then(data => {

        // const newUsers = [...users, data];

        // setUsers(newUsers);

        console.log(data);

      })

  }

#### 64-7 Display POST data on the UI and explore query parameter