Date: _____ Day: _____

Strat Backend

Node.js

Date: 3/21/2024 Lecture 85 Day: Thursday

Node.js and NPM

Q.1:- What is Node.js and where we use it.

Ans:- Node.js is an open-source cross platform javascript runtime environment and library for runing web application outside the client browser. Ryan Dahl developed it in 2009. and its latest iteration varison is 5.14 was released in April 2021. Developer use Node.js to create server-side web application and it is perfect for data intensive application since it uses an asynchronous, event-driven model.

Why Do we Use Node.js

There are many reasons for which we perfer using Nodejs for the server side of our application some of them are discussed in the flowing

Date: Day:

① Node.js is built on Google Chromes V8 engine, and for this reason its execution time is very fast and it run very quickly.

Note: What Google chrom V8 engine.

Overall the V8 engine play a crucial role in providing fast and efficient exection of Javascript code within web browser like Google Chrome, enabling the development of rich and interactive web application. It performance optimization and memory mangement techniques help deliver a smooth browsing experience of user.

② There are more than 50,000 bundles available in the Node package manager and for that reason developers can import any of the package any time according to thier needed functionality for a lot of time is saved.

Day:

Date:

Features of Node.js

1. Asynchoneous in Nature and Event-Driven.
2. Quick execution time for code.
3. Compatibility on the cross platforms. means that Different types of system like window, UNIX
4. Fast Data Streming.
5. No Buffering.

What is Node.js used for?

1. Backend for social media networking
2. Single-page application.
3. Chat application
4. Data streaming
5. IOT application.
   Internet of things

Which company used Node.js.

1. Netflix
2. LinkedIn
3. Uber
4. Nasa
5. Paypal
6. Twitter
7. Walmart
8. Go Daddy.

Date: Day:

Q.1/02 What is NPM?

npm stand for Node Package manager is the default package for Node.js. It is a command-line utility that allow developer to install, manage and share reuseable package of javascript code. npm also provides a centrel repository for publishing open-source Nod.js porject.

① Installing Package:

NPM allows you to easily install package from the npm registry or from a local directory.

npm install (package name)

npm install & express

This Command install the Express.js Framework for building web application.

② Using Install Package

once a package is install you can use it within your Node.js project by importing into your javascript.

Date: 3/22/2024 Day: Friday

code using the require() function or using ES6 import syntax if the package support it.
for example:-

```
const express = require('express')
```

② Managing Dependencies ->
When you install a package using npm, it automatically resolves and install any dependencies required by that package. npm also generates a package-look.json file to look down the version of each dependency install. ensuring consistency across different installation.

③ Listing installed Package.
You can view a list of all package installed in your project by running the npm ls or npm list commend. This command show your the dependency tree of the project including all installed package and their version.
npm --versison To find the version

Date: Day:

③ Updating Packages → you can update package to their latest version using the npm update command. The command update all packages listed in the 'package.json' file to their latest compatible version.

Write NPM init to initialize the package.json file.

What is Node module.

⇒ Node modouls are reuseble block of code that encapsulate related function. They are at the core of Node.js modular architecture. These modules can be built-in Node.js modules thrid-party modouls installed via npm or custom modoule by Developers.

- Built-in-modules ⇒ Node.js comes with a set of build-in-modules that provide essential functionality for task like file systems operation (fs) network communication ('Http') path manipulation

Date: 3/23/2024 Day: Saturday

- Third-party-modules => Developer can leverege a vest ecosystem of thrid-party-modules available on npm, the Node Package manager. These modules cover a wide raenge of function and can be easily installed into Node.js project using Npm.

- Custom modoule => Developer can create their own modules to encapulate specific pieces of functionlity. These module can be imported into other part of the application promoting code orgnization and reusebility.

What is Package.json.

Package.json is a matedata file used in Node.js project to define project setting including dependencis script and other metadata. It is typically located in the root directory of a Nodejs Project and is curcial for managing dependenice and project

Date: Day:

What is Express.js.

Express.js commonly referred to an Express is a web application framework for node.js It provides a robust set of feature for building web application and API including routing middleware support, template engines and more. Express simplify the process of creating web server and handling Http request and responce In Node.js applications.

To use Express in a Node.js project you would typically install using npm. Here how you would do it.

1. Initializing Node.js porject.
If you haven't already create a new directory for you project and initialize a new Node.js project using the following command in your terminal.

npm init -y.

This will create package.json in the default

Day:

Date:

Install Express -> once your project is initialized you can install Express as a Dependancy using NPM.

-> `NPM install express`

This command will download and install the latest version of Express in to you project 'node_modules' directory and update your package.json file with Express as a dependancy.

Date: 3/9/2024 Lecture 86 Day: Sunday

Commonjs modules. ECMAScript modules.

Commonjs and ECMAScript modules are both system for organizing and structuring code in javascript but they have some difference in How they are implemented and used.

1. Commonjs Modules =>
   - Commonjs modules are the format initially used in Node.js for structuring code.
   - They follow a synchronous loading mechanism, meaning dependensie are loading and exected before the execution of the module.
   - They use require() function to import modules. exports or exports to export functionality.
   - Commonjs modules are primarily used in server side javascript enviroment like Node.js.

Date: 3/3/2024 Day: Sunday

Note=> for example if make server using node:Http package to Host our website on server. There is problem genrate if we chage something in Node.js they can't change automatically again we run in terminal. So we install a package ~~re~~ change automatically. If we =>Nodemon restart the file when we do change.

Package Name
NPM --global nodemon.

Note=> What is NVM.
NVM is stand for Node version manager if working with a previous version write in ~~therminal~~ terminal ~~Note~~ NVM use 16

Now move to our main Topic Common js modules=> common is is default module ~~we~~ which we import a file using Require function. for example if we make Build-in-modules

Date: Day:

name i: mymodules.js

code:- mymodules.js

How we use Require in default

```
const myfunc = () => {
    console.log("Hi we check")
}
module.export = myfunc;
```

Now we import the code in main file.

```
const func = require('./mymodules');
func();
```

ECMAScript Modules:-

⇒ ECMAScript modules are the standardized modules format introduced in ECMAScript 6.

⇒ They support asynchronous loading allowing for dynamic imports.

⇒ ES modules use import and Export statement to import/Export functionality.

⇒ They are natively supported in modren web browser and in recent version of Node.js with the appropriate flag.

Date: Day:

Example =>

```
// Exporting modules
export const myfunc = () => {
console.log("Hello sir")
};

// Importing modules
import [myfunction] from './mymoduses'
myfunc();
```

we can export multipee function are varibee.

for exampee;

```
// export modouls
export const a = "Hi"        → name export
export const b = "sir"
export const c = 5
// import modules b, c so on.
import [a] from './mymodules';
```

There is a defult Export to export using default keyword.

```
const obj = { x:5, y:10}
export default obj; // default

import Phy from 'mymodules';
console.log('Phy')
```

Date: Day:

In default Export there is no need the brecket and fix name.

Note: If you are using Esb modules you must be change the type in package.json, write "type": "module";

Note=> we can also import module in Html and

Intervive Question.

There is one question rise in our mind why we can't get error with out initializing this key word (require, export import, dir-name --filename module)

There is also a bulit in function in Node.js file which take all these argumet in packet:-

```
(function( export, requirt, module
--filename, --dirname){
//Module code actually
live here.
});
```

Date: 1/4/2024 Lecture 87 Day: Monday

Working with file : fs and Path modules

What is fs and How we use it :

In Node.js fs stand for file system It a built-in modules that provides function for interacting with the file system on your computer, With the 'fs' module you can perform various file related operation such as reading from file, writting to file creating new file, deleting file and more.

① Write a file =>

There is two way to write a file ① fs.writeFile()
② fs.writeFileSync()

To write a file is a asynchrously use fs.writeFile() its recommend. fs.writFileSync() used for synchnously.

for example => const fs = require('fs')
Console.log('start')
fs.writeFile("Phy.txt", "Hi sir",()=>
{ console.log("data massage")
})
consal.log('end')

Date: Day:

same method to use the fs. writeFileSync().

② Reading files => Use fs.readFile() or fs.readFileSync() to read the content of a file asynchronously or synchrously.

for example=>

```
fs.readFile('Phy.text',(err,data)=>
{ console.log(err, data.toString())
})
```

output=> null, Hi sir

to remove the null in output

```
fs.readFile('Phy.text',(err,data)=>
{ ~~console.log(err, data.t~~
  if (err) {
    console.log(err)
    return;
  }
  clog(data),
```

What is callback hell.

If repeatly use read file and write file they will be make callback hell to ignore the callback hell use the promise in fs.

Page:

Day:

How we use promise in fs

```
import fs from "fs/promises"
let a = await fs.readFile("my.text")
```

what is appendFile. To add extra data in the file.

```
let b = await fs.appendFile("my.text
," In this is amazing")
console.log(a.toString(), b)
```

what is Path modules in Node.js

In Node.js the path modules is a built-in modules that provide utilities for working with file path and directories. It help in constructing and manipulating file path in a platform-independent manner meaning it abstract away the different in file path syntax between different operating system (such as window Linex and macOS)

Date: Day:

Here are some common functionalities provided by the Path module:

① Joining Path Segments => The path.join() method is used to join multiple path segment together to from a single path string =>

```
Const Path = require ('Path')
Const fullPath= Path.join ('/Path', 'to', 'file.text');
Console.log (fullpath).
```

Output => '/Path/to/file.text

Other example

```
Let Path =>" C:// User // Titel // Downloads // video "
        extract name
Console.log (Path.extname(mypath)
        base name
clog ( Path.basename(a))
        directry name
        ( Path.dirname(a).
```