BACKEND😎

DAY 1 ✨

🚀 Topic 1: Node.js aur V8 Engine Kya Hai?

Node.js Kya Hai?

- Node.js koi programming language **nahi** hai.
- Yeh ek **JavaScript Runtime Environment** hai.
- Simple bhasha mein: Yeh ek aisa environment hai jo aapke JavaScript code ko **browser ke bahar** (jaise aapke computer ke server par) run karne deta hai.

V8 Engine Kya Hai?

- V8 Google ka banaya hua ek high-performance **JavaScript engine** hai.
- Yeh C++ mein likha gaya hai.
- Iska main kaam hai:
  1. Aapka JavaScript code lena.
  2. Use Machine Code (jo computer seedhe samajhta hai) mein convert karna.
- **Important:** V8, JS code ko C++ mein convert **nahi** karta. V8 *khud* C++ mein bana hai, aur woh JS ko seedha machine code mein badalta hai.

V8 Engine C++ mein Kyun Bana Hai?

- **Speed:** C++ ek bahut hi fast aur powerful language hai. Ek engine banane ke liye (jo code ko compile karta hai), aapko best possible performance chahiye.
- **Control:** C++ developers ko memory management aur low-level system resources par poora control deta hai, jo ek fast engine banane ke liye zaroori hai.

"V8 can be embedded into C++ application" ka Matlab

- Iska matlab hai ki V8 engine (jo C++ mein hai) ko aap apne C++ application ke andar "fit" ya "daal" sakte hain.
- **Example:** Node.js yahi karta hai! Node.js basically ek C++ program hai jiske andar V8 engine ko embed kiya gaya hai. Iske saath hi Node.js kuch aur features (jaise file system access) add karta hai.

🖥️ Topic 2: Server, Database, aur Business Logic

Server Kya Hai?

- Server ek computer ya software hai jo "clients" (jaise browser ya mobile app) ko service deta hai.
- Yeh requests (jaise "login karo" ya "data dikhao") sunta hai aur response (jaise "login successful" ya "yeh raha data") bhejta hai.

Database (DB) Kya Hai?

- Database (jaise MongoDB) ek jagah hai jahan aapka saara data permanently aur organized tareeke se store hota hai (users, products, posts, etc.).

Frontend, Server, aur Database Saath Mein Kaise Kaam Karte Hain?

Yeh ek simple diagram hai:

Frontend (Browser) 📱 (User "Login" button click karta hai)

↓ (Request jaata hai)

Server (Node.js) ⚙️ (Server request ko process karta hai, "Business Logic" run karta hai)

↓ (Server DB se data maangta hai)

Database (MongoDB) 🗄️ (DB check karta hai user hai ya nahi, aur server ko data wapas bhejta hai)

↑ (DB se response aata hai)

Server (Node.js) ⚙️ (Server final response taiyaar karta hai)

↓ (Response wapas frontend ko bhejta hai)

Frontend (Browser) 📱 (User ko "Welcome" message dikhata hai)

Business Logic Server par Kyun Likhte Hain?

- **Security! Security! Security!**
- **Example:** Maan lo aapke paas 5 lakh ki limit hai. Agar yeh logic frontend (browser) par likha, toh koi bhi smart user browser mein code badal kar usse 10 lakh kar sakta hai.
- Agar yeh logic server par hai, toh user use **touch nahi kar sakta**.

- **Database Credentials:** Aapke database ka password (DB credential) hamesha server par rehta hai. Yeh kabhi bhi frontend ko nahi diya jaata.

🎁 Topic 3: Node.js ke Extra Features

- **Global Object:** Jaise browser mein `window` object hota hai, waise hi Node.js mein `global` object hota hai.
- **`setTimeout`, `setInterval`:** Yeh functions Node.js mein bhi available hain.
- **DOM (Document Object Model)?:** Node.js mein **koi DOM nahi hota**. `document` ya `window` jaise objects yahan nahi milenge, kyunki Node.js server par chalta hai, browser mein nahi.
- **`fetch()`:** Naye Node.js versions mein `fetch()` available hai, bilkul browser ki tarah.

📦 Topic 4: Modules in Node.js (CJS vs ESM)

CommonJS (CJS) Modules Kya Hain?

- Yeh Node.js ka **default** (puraana) module system hai.
- Ismein hum `require()` function se file import karte hain.
- Aur `module.exports` se data export (bhejte) hain.

Data Private Kyun Rehta Hai? (IIFE ka Jaadu)

- **Sawwal:** `second.js` mein banaya gaya function `first.js` mein direct kyun nahi milta?
- **Jawaab:** Kyunki Node.js har file ko ek private "wrapper" mein run karta hai. Yeh wrapper ek **IIFE** (Immediately Invoked Function Expression) jaisa hota hai.

IIFE Kya Hai? Yeh ek aisa function hai jo bante hi turant run ho jaata hai.

```
// Yeh ek IIFE hai

(function (){
   console.log("Main turant run ho gaya");
   // Yahan ka code bahar se access nahi ho sakta
})();
```

Is IIFE ke kaaran, har file ka apna private scope hota hai. Wahan banaye gaye variables tab tak bahar nahi jaate jab tak aap `module.exports` se unhe bhejte nahi.

`module.exports` Kaise Kaam Karta Hai?

- **`module.exports` default roop se ek khaali object hota hai.**
- Jab aap `console.log(module.exports);` likhte hain (bina kuch export kiye), aapko `{}` (empty object) milta hai.
- **Aap data kaise bhej sakte ho:**

Ek Object Bhejna (Recommended):

JavaScript

```
// second.js

function sum(a,b){ console.log(a+b); }

function sub(a,b){ console.log(a-b); }



// Aap object mein properties add kar rahe ho

module.exports.sum = sum;

module.exports.sub = sub;

// Ya short mein: module.exports = {sum, sub};
```

JavaScript

```
// first.js

// Import karte waqt object ko destructure karna padega

const {sum, sub} = require("./second");

sum(5, 5);
```

1.

Sirf ek Function Bhejna:

JavaScript

```
// sour/second.js

function sum(a,b){ console.log(a+b); }

module.exports = sum; // Yahan aapne poora object hi badal diya
```

JavaScript

```
// sour/first.js

// Ab 'sum' variable mein poora function aa gaya

const sum = require("./second");

sum(5, 3);
```

2.

ESM (import/export) Modules

- Yeh naya (modern) module system hai jo JavaScript ka standard hai.
- Yeh `import` aur `export` keywords ka use karta hai.
- By default, Node.js isse `.js` files mein **nahi** samajhta (woh CJS samajhta hai).

Node.js mein ESM Kaise Use Karein?

- **Option 1:** File ka extension `.mjs` (Module JS) kar do.
- **Option 2 (Aapke zip file jaisa):**
  - Apne project folder mein `package.json` file banao.
  - Uske andar likho: `{ "type": "module" }`
  - Bas! Ab aapki saari `.js` files `import`/`export` support karengi.
- **Agar `"type": "module"` hai, toh CJS (`require`) kaise chalayein?**
  - Us file ka extension `.cjs` (Common JS) kar do.
  - Ya `package.json` mein `"type": "commonjs"` likh do (jo default hota hai).

React `import`/`export` Bina Extension Change Kiye Kaise Use Karta Hai?

- React (aur Vite/Create React App) **Babel** aur **Bundlers** (jaise Webpack/Parcel) ka use karte hain.
- Jab aap `npm run dev` ya `npm run build` karte hain, toh yeh tools aapke modern `import`/`export` code ko "transpile" (convert) kar dete hain aise code mein jo saare browsers (aur Node.js) samajh sakein. Woh Node.js ka default module system direct use nahi kar rahe hote.
