MERN Note

July 02, 2025

# 1 Day 1: Website Development

## 1.1 Frontend and Backend

Creating a website involves both frontend and backend development.

* **Frontend** - Designing the user interface using technologies like React.js, Angular, Vue.js, HTML, CSS.
* **Backend** - Managing the logic and data processing behind the scenes using frameworks like Django, Node.js, or .NET.

## 1.2 API Integration

API acts as an agent/medium to connect frontend and backend. A simple case: Nepal India, Nepal bata India rice pathaye jastai using helicopter, helicopter is API. Which is made by (backend).

## 1.3 Database

* **SQL**: Readability, Relation, (Godhamm/warehouse)
* **Postgres, mysql, sqlserver, mariadb, sqlite**
* **NOSQL**: Scalability
* **Mongodb, cassandra**

## 1.4 HTTP Status Codes

* 1xx - 199 - Informational
* 2xx - 299 - success ok, garna khojeko vayo
* 3xx - 399 - redirection, janu parne ek thau arko thau gayo
* 4xx - 499 - client / frontend/nepal le mistake garayo
* 5xx - 599 - backend/server/india ma kehi vayo

## 1.5 HTTP VERBS/Methods

* **POST** - kehi kura pathaunu paryo vane
* **GET** - kehi kura magnu paryo vane
* Cross site request forgery (CSRF) attack – OWASP Top 10
* **PUT** - kehi kura update garnu paryo vane
* **PATCH** - kehi kura update garnu pardaina la same xa tw ?
* **DELETE** - kehi kura delete garnu pardaina

**2 How to setup nodeJs project?**

1. Make an empty folder
2. Open folder in VSCode and open terminal in it.
3. Run npm init command, to automate you can run npm init -y too.
4. This will create package.json file(kundali ho project ko)
5. Create app.js file

## 2.1 Express - heart of nodeJs

Express framework hamro project lai cheyenne kura jastai routing, middleware, caching provides. Alternatives nestjs, hono, fastify.

## 2.2 Npm packages

Npm packages always follow semantic versioning(semver) → major.minor.patch → 2.1.3 Npm → bhatbhateni of javascript (registry of js packages)

* Npm install packageName
* Npm uninstall packageName

## 2.3 Nodemodules

Install gareko package basne thau plus install gareko package lai chayenne raw packages in basxa

**2.4 Package-lock.json**

Raw packages ko kundali which is used to make main package

# 3 JavaScript and NodeJs

JavaScript usually website lai interactive banauna only use hunthyo developed Brendan Eich. Made on 10 days.

But Ryan Dahl → why not js on server side or building backend?

## 3.1 2009 → NodeJs

NodeJsjs(cat)→→jadu ko vhada (environment) jas ma tapai ko js lai chai extra power aauxanodeJs → js(bagh) one task at a time

Tesova tw slow vayo hola ni ?NodeJs → single threaded raixa →

Nai testo hainw , libuv (made on C language) jasko through bata waiting garnu parne task easily handle garna milxo using event loop and thread pool

Event loop (main chef/manager), thread pool (assistant), libuv(kitchen)

Day 5:

Port numbers,

Node – watch app.js, watch mode 🡪 node v18.10 – restart server automatically after change

Script in package json

“start” = “node –watch app.js”

Yeti start rw test xa vane, *npm start*, *npm test* only hand ani bayo, but aru words huda chai jastai manish xa vane *npm run manish* ni hannu parcha

Localhost:3000  
127.0.0.1:3000  
[::1]:3000

For each req there should be response to complete request response

If response is not given, page will keep loading

**Day 6**

**JSON, API Testing using POSTMAN, POSTMAN documentationgeneration, Git and Github**

**Here is the exact text written in the image:**

**Day 7 :**API Documentation Generation Postman  
Collection → View Complete Documentation → Publish –> Website *kholinxa ani publish garne*  
CRUD Book Project  
Rest API – .lathalinga type ko unorganized jasto lekda ni vayo api ko naming  
RestFul API - rules huncha api naming garda eg  
GET - /books (books-resource name)  
POST - /books  
Patch - /books/:id  
Delete - /books/:id

Traditionally API communication, under fetching(tukra tukra gardai data *magnu*), over fetching( magey vanda *dherae aayunu* causes security loophole(information disclosure)

Graphql - solves both underfetching and over fetching problem , *dont* have get , post put, patch like traditional api, it *uses* three things : query, mutate, subscription.

Day 8 :  
Supabase → Backend as a service like firebase, but hami chai just postgres ko lagi matra use *garxam*, postgres database free ma *dinxa*  
Connection String → Mathi ko connect button ma click *garne* → ORMs vanne tab ma *gayerw* connection string copy *garne*  
Why ORM needed ?  
Raw query *sikiranu* paryo , security (sql injection)  
ORMs example : sequelize, prisma, drizzle, mongoose(ODM)  
Methods *dinxa* , hamile tehi methods use garda vayo raw query airaknu parena like users.findAll(), users.create()..... → translates into raw query (select \* from users) → DB