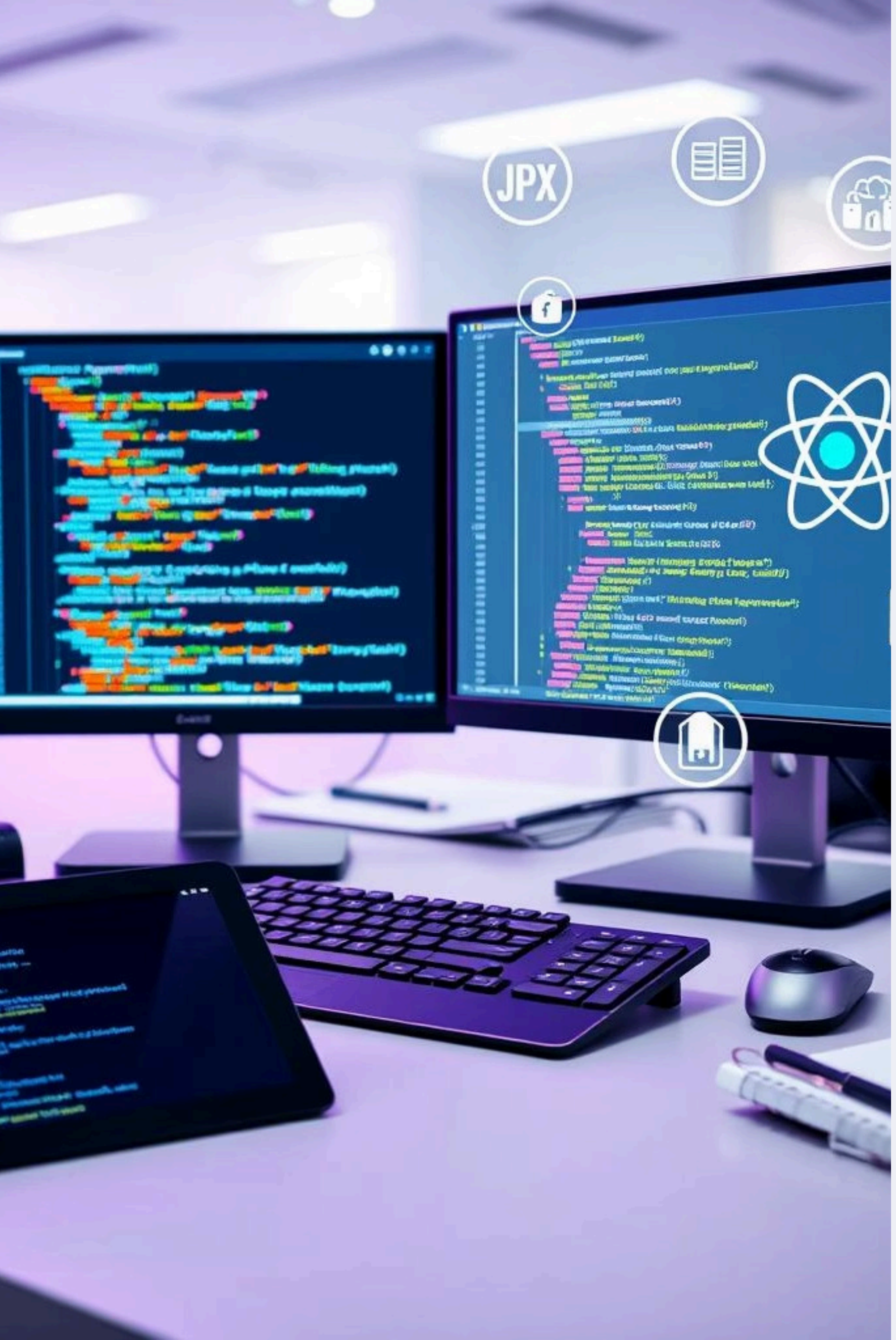

Mastering the MERN Stack: Your Journey Begins Here

Welcome

Made with GAMMA

Day 1 Learning Objectives

Understand MERN Stack Interactions

Grasp how MongoDB, Express.js, React.js, and Node.js work together.

Differentiate Frontend & Backend

Clearly distinguish between client-side and server-side responsibilities.

Set Up Node.js + Express Server

Configure a foundational backend server for your applications.

Test API Endpoints with Postman

Learn to send requests and inspect responses to validate your APIs.

Understand MongoDB's Role

Comprehend how MongoDB serves as the backbone for data storage.

Made with GAMMA

What is MERN? Why MERN?

The MERN stack is a powerful collection of technologies for building full-stack applications. Each component plays a crucial role in creating seamless web experiences.

MERN Stack Components

- **M**ongoDB: Database
- **E**xpress.js: Web Application Framework
- **R**eact.js: Frontend JavaScript Library
- **N**ode.js: JavaScript Runtime

Analogy: Building a Product

React is the storefront, handling what users see.

Express + Node are the warehouse managers, processing requests.

MongoDB is the inventory system, storing all your data.

Made with GAMMA

The Full-Stack Web: Frontend vs. Backend

Understanding the division of labor between frontend and backend is fundamental to full-stack development. They are two sides of the same coin, working in harmony.

Frontend (React)

- UI Components & Layout
- Client-Side Form Validation
- State Management
- Making API Requests
- Responsive Design

Backend (Node + Express)

- API Endpoints Definition
- Data Validation & Business Logic
- Database Operations
- Authentication & Authorization
- Error Handling & Logging

> **Key Point:** The frontend **asks** (sends requests), and the backend **acts** (processes and responds).

Made with GAMMA

Setting Up Your Project: Best Practices

A well-structured project is the foundation of efficient development. We'll walk through the essential steps to set up your Node.js and Express.js backend, ensuring a smooth workflow from the start. (Pointer for NodeJs Basics)

This includes initializing your project, organizing folders, and installing key dependencies like Express. We'll demonstrate a basic server configuration.

Made with GAMMA

```
Fargistan:
  etettems rerile:
    Imstert fiostiable = Todnlg/Apl)
     Mpll:
    Teall haing inster/lopertation:
      The respunized stalr restartionstaple,
```

Installing Postman & Testing APIs

Postman is an indispensable tool for API development. It allows you to send various types of requests and inspect responses without needing a frontend interface.

We'll perform our first GET request to a basic endpoint, observe the server's response, and discuss exactly what happens behind the scenes when you click "Send".

Made with GAMMA

Introduction to MongoDB: Your Flexible Database

MongoDB is a NoSQL database that stores data in flexible, JSON-like documents. This schema-less approach offers great versatility for evolving applications.

Analogy: Google Sheets

Think of MongoDB like Google Sheets: each document is a row, and each collection is a sheet within a spreadsheet.

Installation & Setup

Learn to install MongoDB locally or configure a cloud-hosted instance using MongoDB Atlas for easy access.

JSON Structure

Explore the power of JSON documents and the flexibility that a schema-less database offers for diverse data.

Made with GAMMA

Key Takeaways & Next Steps

1

MERN Foundations

Today, we laid the groundwork for understanding the MERN stack and the distinct roles of frontend and backend.

2

Practical Tools

You've set up your first server and learned to test APIs using Postman, essential skills for any developer.

3

Database Basics

We introduced MongoDB's document-based nature, paving the way for data management in your applications.

4

Take Home: Practise the stuff we learnt today, Attempt Kahoot, Try creating a backend server in Vanilla NodeJS and understand the nuances. Try exploring MongoDB.

Made with GAMMA