🔰 1. Introduction to Full-Stack Development

- Full-stack = Frontend + Backend + Database + Deployment
- Emphasis on **JavaScript-based stacks** due to popularity and ecosystem

---

🌐 2. Popular Full-Stack Development Stacks

Stack	Full Form / Components	Frontend	Backend	Database	Key Highlights / Use Case
MEAN	MongoDB, Express.js, Angular, Node.js	Angular	Express.js	MongoDB	All-JavaScript stack; suitable for SPA and enterprise web apps
MERN	MongoDB, Express.js, React, Node.js	React	Express.js	MongoDB	Popular for dynamic, component-based UI development
MEVN	MongoDB, Express.js, Vue.js, Node.js	Vue.js	Express.js	MongoDB	Lightweight frontend stack; easier learning curve
MEAN(R)	MongoDB, Express.js, Angular/React, Node.js	Angular + React	Express.js	MongoDB	Combines strengths of both Angular and React
JAMstack	JavaScript, APIs, Markup	JS + Static HTML	API (serverless)	Headless CMS / DB	Optimized for speed, security, scalability using CDNs and APIs
LAMP	Linux, Apache, MySQL, PHP	HTML/CSS/JS	PHP	MySQL	Traditional, robust stack; good for shared hosting environments

By: Ishan K Rajani

Stack	Full Form / Components	Frontend	Backend	Database	Key Highlights / Use Case
LEMP	Linux, Nginx (Engine-X), MySQL, PHP/Python	HTML/CSS/JS	PHP or Python	MySQL	Modern variant of LAMP with Nginx for improved performance
Django Stack	Python, Django, HTML/CSS/JS, PostgreSQL/MySQL	HTML/CSS/JS	Django (Python)	PostgreSQL/MySQL	High-level Python web framework; rapid dev and clean code
Ruby on Rails	Ruby, Rails, HTML/CSS/JS, PostgreSQL/MySQL	HTML/CSS/JS	Ruby on Rails	PostgreSQL/MySQL	Favors convention over configuration; great for startups
.NET Stack	ASP.NET Core, Blazor/JS, SQL Server	Blazor/JS	ASP.NET Core	SQL Server	Enterprise-ready; integrates well with Microsoft tools
Serverless Stack	JS Frontend, AWS Lambda/Firebase, Cloud DB	React/Vue/Angular	AWS Lambda / Firebase	DynamoDB / Firestore	Scales automatically; ideal for microservices and event-driven apps

By: Ishan K Rajani

⚙️ 3. Common Technologies Used Across Stacks

🌐 Frontend

- **React**, **Angular**, **Vue.js**, **Svelte**, **Next.js**, **Nuxt.js**

🖥️ Backend

- **Node.js**, **Express.js**
- **Django** (Python), **Flask** (Python)
- **Spring Boot** (Java), **ASP.NET Core** (C#)
- **Rails** (Ruby)

🗄️ Databases

- **MongoDB**, **MySQL**, **PostgreSQL**, **Firebase**, **Redis**, **SQLite**

☁️ DevOps & Deployment

- **Docker**, **Kubernetes**
- **CI/CD**: GitHub Actions, GitLab CI
- **Platforms**: AWS, Heroku, Vercel, Netlify

🧠 4. How to Choose a Stack?

- Based on project size, team expertise, scalability, performance
- JS-based stacks (like MERN) are ideal for rapid development
- Python or Java stacks are strong in enterprise & data-intensive apps

By: Ishan K Rajani