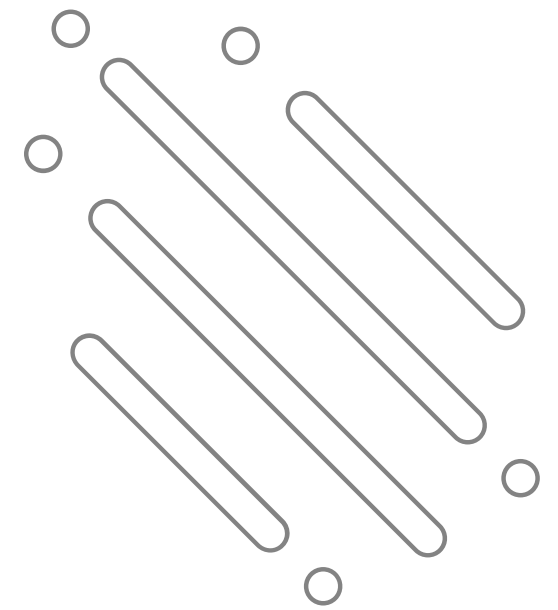
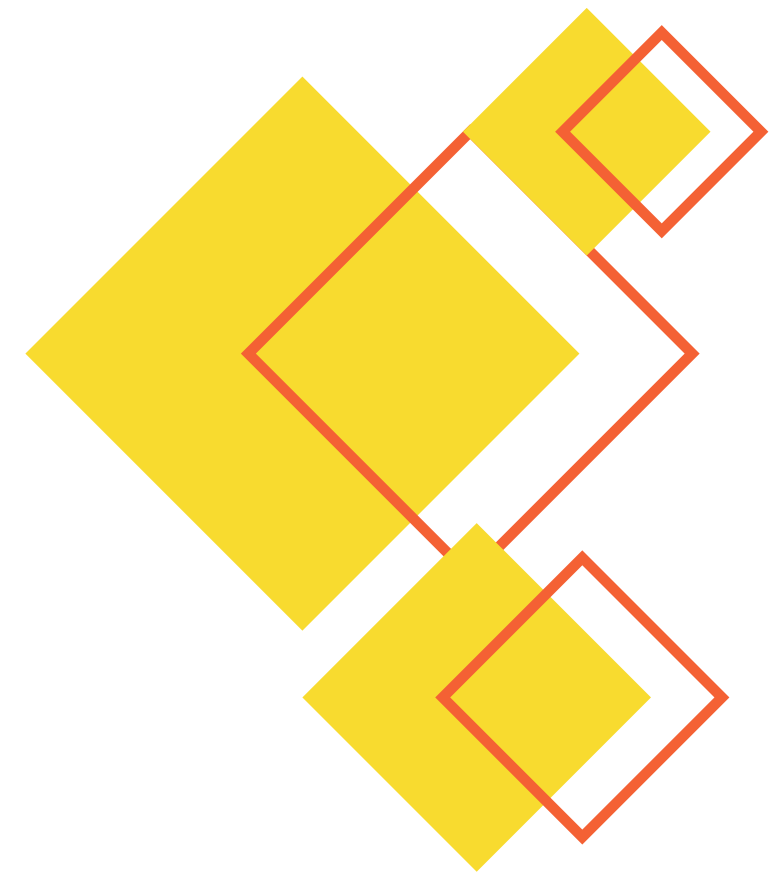
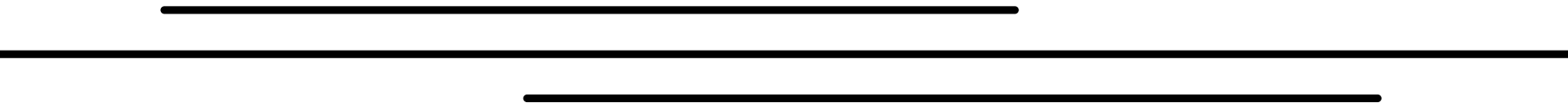




SYSTEM DESIGN



MONOLITHIC VS
MICROSERVICES
ARCHITECTURE



ডিডিওর মূল উদ্দেশ্য

- Monolithic এবং Microservices architecture কী?
- কোনটা কখন ব্যবহার করবেন?
- System design-এ এই দুইটি স্টাইলের গুরুত্ব কী?
- বাস্তব উদাহরণ ও তুলনামূলক বিশ্লেষণ।

MONOLITHIC ARCHITECTURE কী?

- একক কোডবেজ (single codebase), একসাথে deploy হয়।
- সব ফিচার একসাথে একটি অ্যাপ্লিকেশনেই থাকে।
- উদাহরণ:
- একটি Laravel বা Django অ্যাপ যেখানে frontend, backend, authentication, payment – সব একই অ্যাপে থাকে।

সুবিধা:

- Development শুরু করা সহজ।
- Deploy করা সহজ (একটি ফাইল বা সার্ভারেই সব থাকে)।
- Performance সাধারণত ভালো থাকে কম ইন্টারনাল communication থাকার কারণে।

অসুবিধা:

- অ্যাপ বড় হলে কোড maintain করা কঠিন হয়ে পড়ে।
- এক জায়গার bug পুরো অ্যাপকে প্রভাবিত করে।
- Continuous deployment/scale কঠিন হয়ে পড়ে।

MICROSERVICES ARCHITECTURE কী?

- পুরো অ্যাপ ছোট ছোট সার্ভিসে ভাগ করা হয়।
- প্রতিটি সার্ভিস আলাদা ভাবে কাজ করে এবং আলাদাভাবে deploy হয়।
- সার্ভিসগুলো API বা Message Queue-এর মাধ্যমে যোগাযোগ করে।

সুবিধা:

- প্রতিটি সার্ভিস আলাদাভাবে scale করা যায়।
- আলাদা টিম আলাদা সার্ভিস নিয়ে কাজ করতে পারে।
- একটি সার্ভিসে সমস্যা হলেও পুরো অ্যাপ বন্ধ হয় না।

অসুবিধা:

- Infrastructure setup জটিল (Docker, Kubernetes, etc. দরকার হতে পারে)।
- সার্ভিসগুলোর মধ্যে communication, logging, monitoring complex হয়।
- Initial development comparatively slow এবং costly।

তুলনামূলক বিশ্লেষণ (COMPARISON TABLE)

বিষয়	Monolithic	Microservices
Deployment	একসাথে সব	আলাদা আলাদা
Maintenance	কষ্টকর (বড় হলে)	সহজ (প্রতিটি সার্ভিসে আলাদা কাজ)
Scaling	পুরো অ্যাপ একসাথে	নির্দিষ্ট সার্ভিস
Development	দ্রুত শুরু	শুরুতে জটিলতা
Fault Isolation	কম	বেশি
Suitable for	ছোট/মাঝারি প্রজেক্ট	বড়/কমপ্লেক্স প্রজেক্ট

কখন কোনটা ব্যবহার করবেন?

Monolithic ব্যবহার করবেন যদি:

- আপনার প্রজেক্ট ছোট বা MVP (minimum viable product)।
- টিম ছোট।
- আপনি দ্রুত কিছু তৈরি করে ইউজারদের সামনে দিতে চান।
-

Microservices ব্যবহার করবেন যদি:

- আপনার অ্যাপ অনেক বড় বা ভবিষ্যতে অনেক বড় হবে বলে ভাবছেন।
- আপনার টিম বড় বা future scalability দরকার।
- আপনি বিভিন্ন সার্ভিস বিভিন্ন ল্যাঙ্গুয়েজে বা টিমে আলাদা রাখতে চান।

SYSTEM DESIGN CONTEXT-এ গুরুত্ব

- System Design এ আপনি শুধু features plan করেন না, architecture design করেন।
- এই সিদ্ধান্তই নির্ধারণ করবে আপনি future scale করতে পারবেন কিনা।
- তাই Monolith vs Microservice বোঝা জরুরি: আপনি system-এর structure ও reliability নির্ধারণ করছেন।

বাস্তব উদাহরণ

- Monolithic: একটিটা Laravel অ্যাপ যেটা frontend/backend/DB সব এক জায়গায়।
- Microservice: Amazon-এর মতো প্ল্যাটফর্ম – যেখানে payment, product, cart, user সব আলাদা সার্ভিস।

আপনার জন্য আমার সাজেশন হলো :

- আপনার প্রজেক্টের stage, size, এবং team অনুযায়ী সিদ্ধান্ত নিন।
- Design করার সময় future vision মাথায় রাখুন।
- ভুল decision, System কে ধ্বংস করতে পারে, আর সঠিক decision তাকে scalable ও maintainable করে তোলে।


তাহলে আমরা যে লারাভেল বা DJANGO দিয়ে এপ্লিকেশন বানাই যেখানে ব্যাকএন্ড আমরা একটা পার্ট রাখি আর ফ্রন্টএন্ড এ আমরা NEXTJS দিয়ে ডেভেলপ করি কমিউনিকেশন এর জন্য আমরা এপিআই ব্যবহার করি এবং আমাদের ডাটাবেস কিন্তু ১টাই থাকে সেটাকে কি আমরা মাইক্রোসার্ভিস বলবো নাকি মনোলিথিক সিস্টেম বলবো ?

টা Monolithic না, আবার পুরোপুরি Microservices ও না। তবে এর closest সংজ্ঞা হবে:

API-based Decoupled Monolith Architecture

Microservices হলে:

1. Backend-এ ছোট ছোট services ভাগ করা হতো
2. প্রতিটা সার্ভিসের আলাদা কোডবেস, আলাদা deploy, আলাদা DB (বা অন্তত schema separation)
3. সার্ভিসগুলোর মাঝে communication হত API বা Message Queue দিয়ে



শেষ কথা:

সুতরাং এই ছিল আজকের আলোচনা Monolithic এবং Microservice Architecture নিয়ে সামনে আমরা
জানাবো

Computer Architecture in system design