

System Design

🧠 System Design আসলে কী?

System Design মানে হলো —

👉 একটা software system এমনভাবে পরিকল্পনা করা, যাতে সেটা

- লক্ষ লক্ষ user handle করতে পারে
- দ্রুত কাজ করে (low latency)
- সহজে বড় করা যায় (scalable)
- নষ্ট হলেও পুরো system না ভেঙে পড়ে (reliable & fault-tolerant)

✦ সহজ ভাষায়:

“System Design = বড় সমস্যা কে ছোট ছোট অংশে ভেঙে বাস্তবসম্মতভাবে বানানোর চিন্তাধারা”

✗ System Design কী নয়?

অনেকেই ভুল বোঝে।

- ✗ শুধু code লেখা
- ✗ শুধু diagram আঁকা
- ✗ শুধু tools জানা (AWS, Docker ইত্যাদি)

👉 এগুলো tools, System Design হলো **thinking process**।

✅ System Design আসলে কী নিয়ে কাজ করে?

1 Scale (সবচেয়ে গুরুত্বপূর্ণ)

“User 100 থেকে 10 million হলে কী হবে?”

System Design বলে:

- কোথায় bottleneck হবে
- কীভাবে system survive করবে

2 Components কিভাবে কথা বলবে

- Client
- Backend services
- Database
- Cache
- Queue

📌 কে কার সাথে কথা বলবে, কখন, কীভাবে — এটিই design।

3 Trade-offs

সব কিছু একসাথে পাওয়া যায় না।

উদাহরণ:

- Strong consistency ❌ High availability
- Low latency ❌ 100% accuracy

System Designer সিদ্ধান্ত নেয়:

“এই problem-এর জন্য কোনটা বেশি দরকার?”

4 Failure handle করা

Real world এ সব কিছু fail করে।

- DB down
- Server crash
- Network slow

System Design শেখায়:


“Fail হলেও service চলবে কীভাবে?”

Real-Life Example (Simple)

Problem: “একটা Online Food App বানাতে হবে”

Code চিন্তা করার আগে System Designer ভাবে:

- কত user?
- Peak time কখন?
- Order data কোথায় যাবে?
- Payment fail হলে কী হবে?
- Notification কীভাবে যাবে?

 এগুলোর উত্তরই **System Design**।

Interview Perspective (FAANG)

Interviewer আসলে জিজ্ঞেস করে না:

“তুমি Redis জানো?”

সে দেখে:

- তুমি প্রশ্ন করছো কিনা
- Assumption ঠিক কিনা

- Logical flow আছে কিনা
- Decision explain করতে পারছে কিনা

System Design = 5-Step Thinking Framework

- 1 Requirements বোঝা
- 2 High-level architecture
- 3 Data design
- 4 Scaling strategy
- 5 Failure handling

এই structure follow করলে **ANY system** design করা যায়।

One-line Summary

System Design is not about drawing boxes — it's about making correct decisions under constraints.