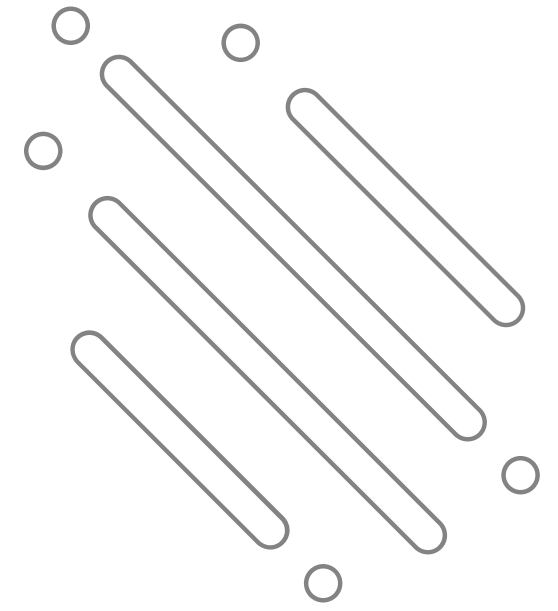
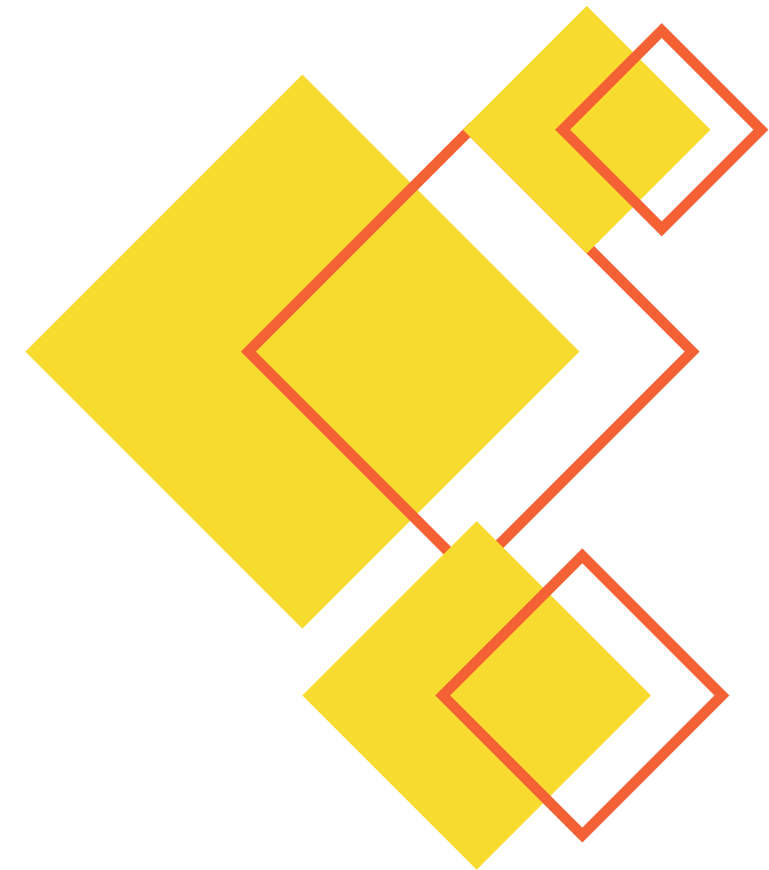
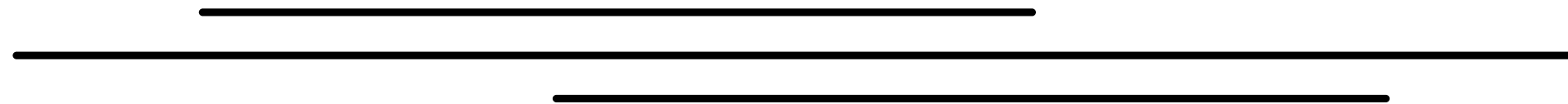


# SYSTEM DESIGN



সিস্টেম পারফরম্যান্স মেট্রিক্স: অ্যাভেইলিবিলাটি,  
স্কেলেবিলিটি, রিডাবলিটি



সিস্টেম ডিজাইন করার সময় শুধু একটি ফাংশনাল সিস্টেম তৈরি করলেই হয় না, তার চেয়েও গুরুত্বপূর্ণ হলো যে সিস্টেমটি আমাদের কে সেবা পারফরম্যান্স প্রোভাইড করছে, এবং বাস্তব সকল কাস্টমারদের চাহিদা পূরণ করতে পারছে। এখানেই আসলে পারফরম্যান্স মেট্রিক্সের গুরুত্ব। এই মেট্রিক্সগুলো আমাদের সিস্টেমের মান এবং কার্যকারিতা পরিমাপ করার জন্য একটি পারফেক্ট সল্যুশন।

# অ্যাভেইলিবিলাটি (AVAILABILITY) - প্রাপ্যতা

অ্যাভেইলিবিলাটি বলতে আমরা সাধারণত বুঝি যে একটি সিস্টেম কত শতাংশ সময় সচল এবং ব্যবহারকারীদের কাছে অ্যাক্সেসযোগ্য থাকে, যেটাকে আমরা আমাদের ভাষায় বলি লাইভ আছে। এটি সাধারণত শতাংশে প্রকাশ করা হয়, যেমন "99.9% অ্যাভেইলিবিলাটি" (থ্রি নাইনস) বা "99.999% অ্যাভেইলিবিলাটি" (ফাইভ নাইনস)।

গাণিতিক উপস্থাপনা:

$$\text{অ্যাভেইলিবিলাটি} = \frac{\text{মোট আপটাইম}}{\text{মোট আপটাইম} + \text{মোট ডাউনটাইম}} \times 100\%$$

উচ্চ অ্যাভেইলিবিলাটি অত্যন্ত গুরুত্বপূর্ণ কারণ ব্যবহারকারীরা আশা করে যে সিস্টেমটি যখনই তাদের প্রয়োজন হবে তখনই অ্যাক্সেস করতে পারবে। ব্যাংকিং, ই-কমার্স বা স্বাস্থ্যসেবার মতো গুরুত্বপূর্ণ সিস্টেমগুলির জন্য কয়েক মিনিটের ডাউনটাইম থাকলে সেটা কোম্পানির জন্য আর্থিক ক্ষতি, সুনামের ক্ষতি এবং ব্যবহারকারীর অসন্তুষ্টি হতে পারে।

কোথায় গুরুত্বপূর্ণ ?

1. ওয়েব সার্ভিসেস: ই-কমার্স ওয়েবসাইট, সোশ্যাল মিডিয়া প্ল্যাটফর্ম, সার্চ ইঞ্জিন।
2. ক্লাউড ইনফ্রাস্ট্রাকচার: ভার্চুয়াল মেশিন, স্টোরেজ, ডেটাবেস।
3. গুরুত্বপূর্ণ ব্যবসায়িক অ্যাপ্লিকেশন: এন্টারপ্রাইজ রিসোর্স প্ল্যানিং (ERP) সিস্টেম, কাস্টমার রিলেশনশিপ ম্যানেজমেন্ট (CRM) সিস্টেম

না জানলে কী সমস্যা হয়?

- আয়ের ক্ষতি: আপনার ই-কমার্স সাইট ডাউন থাকলে আপনি কিছু বিক্রি করতে পারবেন না।
- গ্রাহকের অসন্তুষ্টি ও পরিষেবা ত্যাগ: যে পরিষেবা প্রায়শই অনুপলব্ধ থাকে, ব্যবহারকারীরা সেটি ছেড়ে দেবেন।
- সুনামের ক্ষতি: সিস্টেম বিভ্রাটের খবর দ্রুত ছড়িয়ে পড়ে এবং বিশ্বাস নষ্ট করে।
- নিয়ন্ত্রণমূলক জরিমানা: কিছু শিল্পের জন্য, ডাউনটাইম জরিমানা ডেকে আনতে পারে।

# RELIABILITY (নির্ভরযোগ্যতা) কী?

Reliability হচ্ছে কোনো system কত সময় ধরে কোনো রকম error ছাড়াই নির্ভুলভাবে কাজ করতে পারে – সেটি বুঝায়।

একটি system reliable যদি হয় সেটি বারবার ঠিক output দিবে এবং কোনো সময়ে ভুল না করে কাজ চালাতে পারবে তখন আপনি সিস্টেমটিকে Reliability ভালো আখ্যায়িত করতে পারেন।

চলুন একটি উদাহরণ দেখি :

ধরুন, আপনি একটি online payment system বানালেন। যেখানে ১০০ জন payment করলো

৮ জনের টাকা কেটে গেল, কিন্তু order confirm হলোনা  
তাহলে আপনার সিস্টেম reliable নয়

অথবা ধরুন, ৫ বার কেউ search করলো ২ বার ই timeout হয়ে গেলো – তাহলে কি আপনি আপনার সিস্টেমকে reliability ভালো বলতে পারবেন কখনোই না

তাহলে আমরা বুঝলাম Reliability  $\neq$  Availability কিন্তু তাহলে এক জিনিস না

Availability == সিস্টেম চালু আছে কিনা

Reliability == চালু থাকলেও ঠিকমতো কাজ করছে কিনা

**Reliability এর মূল বৈশিষ্ট্য:**

Consistency  $\rightarrow$  প্রতি বার একই ধরনের সঠিক result দিবে

Fault Tolerance  $\rightarrow$  কিছু error হলেও system fail করে না

Durability  $\rightarrow$  কাজ করে গেলে data হারায় না

Error-free Operation  $\rightarrow$  অনাকাঙ্ক্ষিত bug/crash থাকে না



**Reliability না থাকলে কী সমস্যা হয়?**

1. User Trust কমে যায়
2. Data corruption হয়
3. Revenue loss হয় (e.g., payment system ভুল করলেই টাকা নষ্ট)
4. Debugging কষ্টকর হয়
5. SLA breach হয়

## তাহলে কিভাবে Reliable System বানানো যায়?

- ◆ 1. Redundancy ব্যবহার করুন
  - একাধিক component রাখলে failure হলেও অন্যটি কাজ করে
- ◆ 2. Retry Mechanism
  - Request fail করলে আবার চেষ্টা করে
- ◆ 3. Circuit Breaker Pattern
  - System overload হলে কিছু অংশ থামিয়ে দেয় যাতে পুরো system না পড়ে
- ◆ 4. Monitoring & Alerting
  - কোনো error হলে সঙ্গে সঙ্গে alert পাওয়া যায় (e.g., Datadog, Prometheus)
- ◆ 5. Testing and Validation
  - Proper Unit test, Integration test, Load test
- ◆ 6. Graceful Degradation
  - System এর কোনো অংশ fail করলেও বাকি অংশ কাজ চালিয়ে যা

# কিছু বাস্তব উদাহরণ:

System	Reliability কি ভাবে গুরুত্বপূর্ণ
 Banking System	1 টাকা হারালেও trust নষ্ট
 Hospital Software	ভুল prescription দিলে জীবন ঝুঁকিতে
 E-commerce Checkout	Order fail হলে user চলে যাবে
 SpaceX Systems	hardware + software সবই reliable হতে হয়

## Reliability মাপার উপায় (MTTF, MTTR, MTBF)

Metrics	অর্থ
MTTF (Mean Time To Failure)	system কত সময় কাজ করে failure এর আগে
MTTR (Mean Time To Repair)	failure এর পরে ঠিক হতে কত সময় লাগে
MTBF (Mean Time Between Failures)	একবার fail হওয়া থেকে পরেরবার fail হওয়া পর্যন্ত সময়

বেশি MTBF = বেশি reliable  
কম MTTR = দ্রুত ফিক্স করা যায়



# REDUNDANCY (অতিরিক্ত ব্যাকআপ / প্রতিস্থাপন ক্ষমতা)

## Redundancy (রিডানডেন্সি) কী?

Redundancy হচ্ছে এমন একটি ডিজাইন কৌশল, যেখানে প্রধান কোনো কম্পোনেন্ট fail করলে, আরেকটি ব্যাকআপ কম্পোনেন্ট তাৎক্ষণিকভাবে সচল হয়ে যায় – ফলে আপনার system কখনও একদম বন্ধ হয়ে যাবে না।

সুতরাং এক কোথায় বলতে পারেন,

"একটা না থাকলে আরেকটা কাজ করবে – এটাই Redundancy!"

## লক্ষ্য:

1. High Availability নিশ্চিত করা
2. Single Point of Failure (SPOF) দূর করা
3. Fault Tolerant System তৈরি করা

## চলুন একটি উদাহরণ দেখি:

সিস্টেম	Without Redundancy	With Redundancy
Server	1 টা সার্ভার থাকলে crash হলেই সব বন্ধ	2 টা সার্ভার থাকলে backup চালু হয়ে যায়
Database	একটাই DB থাকলে তা fail করলে data access সম্ভব না	Replica DB বা cluster থাকলে smooth থাকে
Power Supply	একটাই power source থাকলে বিদ্যুৎ গেলে সব নষ্ট	UPS বা generator থাকলে চালু থাকে

# Redundancy এর ধরণ (Types)

## ◆ 1. Hardware Redundancy

Extra server, power supply, network switch

📌 উদাহরণ: Dual power supply in servers

## ◆ 2. Software Redundancy

Application-level fallback, retry mechanism

📌 উদাহরণ: Payment gateway fail হলে অন্য gateway use

## ◆ 3. Data Redundancy

Data multiple জায়গায় রাখা

📌 উদাহরণ: Master-slave DB, RAID setup

## ◆ 4. Network Redundancy

Multiple network paths


📌 উদাহরণ: Load balancer with two ISPs

# Redundancy কিভাবে বাস্তবে implement করা যায়?

Scope	কীভাবে করবেন
Server	Load balancer + multiple web servers (Nginx, HAProxy)
Database	Master-slave, replica sets (MySQL, PostgreSQL, MongoDB)
File Storage	AWS S3 + Backup to another region
Deployment	Blue-Green deployment strategy
Network	Multiple ISPs or VPN tunnels

## বাস্তব উদাহরণ:

- 🛒 Amazon : হাজার হাজার server আছে redundancy সহ – একটি crash হলেও হাজারটি চালু থাকে
- 🏥 Hospital System : Life-support বা medical record system কখনও ডাউন হওয়া চলবে না – তাই full redundancy
- 🚀 NASA Space Systems : spacecraft এ triple redundancy থাকে – দুইটি fail করলেও তৃতীয়টি চালু থাকে



# শেষ কথা:

সুতরাং এই ছিল আজকের আলোচনা অ্যাভেইলিবিলিটি, রিলায়েবিলিটি, রিডানডেন্সি নিয়ে সামনে  
আমরা জানাবো  
থ্রুপুট এবং ল্যাটেন্সি in system design