Online Quiz System

Introduction:

This documentation compares two approaches to implementing an online quiz system: **multithreading** and **sequential processing**. Each method offers unique advantages and trade-offs, suitable for different use cases.

1. Multithreading Implementation:

- Overview:
  A multithreaded quiz system handles multiple user requests simultaneously by creating separate threads for tasks such as question delivery, scoring, and logging.
- Key Features:
  - Concurrent handling of multiple users.
  - Efficient CPU utilization, especially on multi-core systems.
- Advantages:
  - High concurrency and scalability.
  - Faster response time for simultaneous requests.
  - Optimized resource usage.
- Disadvantages:
  - Complex to implement and debug.
  - Risk of concurrency issues like race conditions and deadlocks.
  - Higher maintenance effort.

---

2. Sequential Implementation:

- Overview:
  A sequential quiz system processes user requests one at a time, completing each task fully before starting the next.
- Key Features:
  - Simplicity in development and debugging.
  - Tasks are executed in a predictable and logical order.
- Advantages:
  - Easy to implement and maintain.
  - No concurrency issues, as only one task runs at a time.
  - Lower development costs.
- Disadvantages:

- Slower response time as users wait in a queue.
- Limited scalability and performance under heavy traffic.
- Inefficient resource utilization on modern systems.

Multithread implemented vs sequential implementation:

	Multithread-Implemented	Sequential-Implemented
Performance	High with multiple users	Slow with multiple users
Scalability	Supports many users	Limited to fewer users
Implementation	Complex	Simple
Response Time	Faster for multiple users	Slower for multiple users
Concurrency	Available	None