# Distributed Systems Group Assignment

## Title: Designing a Fault-Tolerant Distributed File Storage System

## Scenario:

Your team has been tasked with designing a distributed file storage system. The system must ensure high availability, fault tolerance, and consistency while handling concurrent read/write operations from multiple clients. The system will be deployed across multiple servers. The uploaded files should be accessible by any client.

## Tasks:

Divide the tasks among the 4 team members, ensuring collaboration and integration of all components. Each member will focus on one core aspect of the system, but the team must work together to ensure the system functions as a whole.

### 1. Fault Tolerance (Member 1):

Objective: Ensure the system can continue operating even if one or more servers fail.

Tasks:

- Design a redundancy mechanism (e.g., replication or erasure coding) to tolerate server failures.

- Implement a failure detection system to identify when a server goes offline.

- Propose a recovery mechanism to restore data and services when a failed server comes back online.

- Evaluate the trade-offs between fault tolerance and system performance.

### 2. Data Replication and Consistency (Member 2):

Objective: Ensure data is replicated across multiple servers while maintaining consistency.

Tasks:

- Design a replication strategy (e.g., primary-backup, multi-master, or quorum-based replication).

- Choose a consistency model (e.g., strong consistency, eventual consistency) and justify your choice.

- Implement a mechanism to handle conflicts during concurrent read/write operations.

- Evaluate the impact of replication on system performance and consistency.

### 3. Time Synchronization (Member 3):

Objective: Ensure all servers in the system have synchronized clocks for consistent operations.

Tasks:

- Research and implement a time synchronization protocol (e.g., NTP or PTP) to synchronize clocks across servers.

- Analyze the impact of clock skew on system operations (e.g., ordering of events, consistency).

- Propose a mechanism to handle scenarios where time synchronization fails.

- Evaluate the trade-offs between synchronization accuracy and system overhead.

### 4. Consensus and Agreement Algorithms (Member 4):

Objective: Ensure all servers agree on the state of the system, even in the presence of failures.

Tasks:

- Research and implement a consensus algorithm (e.g., Paxos, Raft, or Zab) to achieve agreement among servers.

- Design a mechanism to handle leader election in case the current leader fails.

- Evaluate the performance of the consensus algorithm under different failure scenarios.

- Propose optimizations to reduce the overhead of achieving consensus.

- Test the system under different scenarios such as server failures and network partitions.

## Deliverables:

1. Report:  
 - A detailed report (10-12 pages) explaining the design choices, algorithms, and mechanisms used for each task. Include the test results conducted and include a discussion of the results.

2. Implementation:  
 - A prototype of the system using a programming language such as Python, Java, or Go.

3. Presentation:  
 - A 15 minute presentation slides explaining the design, implementation, and evaluation.

4. Code:

- Github link to the code in a text file.

5. Presentation video:

- Link to the Youtube video with the presentation.

## Evaluation Criteria:

1. Fault Tolerance:  
 - How well does the system handle server failures?  
 - Is the recovery mechanism effective?

2. Data Replication and Consistency:  
 - Does the system maintain consistency during concurrent operations?  
 - How does replication impact performance?

3. Time Synchronization:  
 - Are the servers’ clocks synchronized accurately?  
 - How does the system handle clock skew?

4. Consensus and Agreement:  
 - Does the system achieve consensus reliably?  
 - How does the consensus algorithm perform under failures?

5. Overall Integration:  
 - Do all components work together seamlessly?  
 - Is the system scalable and efficient?

## Submission Guidelines:

- Submit the report, source code, and presentation slides and text files with the links in a single zip file.

- Include a README file with the names, registration numbers and emails of the members and the instructions for running the prototype.

## Evaluation Criteria

|  |  |  |
| --- | --- | --- |
| Criteria | Weight | Description |
| Fault Tolerance | 20% | Effectiveness of redundancy, failure detection, and recovery mechanisms. |
| Data Replication & Consistency | 20% | Consistency model, conflict resolution, and replication strategy. |
| Time Synchronization | 20% | Accuracy of clock synchronization and handling of clock skew. |
| Consensus & Agreement | 20% | Reliability and performance of the consensus algorithm. |
| Overall Integration | 20% | Seamless integration of all components and scalability of the system. |