**Project Report 2 :**

**Fault Tolerant 2- phase distributed commit (2PC)**

I have neither given nor received unauthorized assistance or work

Mohit Kulkarni 1002031021 08/07/2023

Shaunak Amble 1002090283 08/07/2023

Implementation Overview :

The code simulates a distributed transaction scenario involving a Transaction Coordinator and two participants. It defines functions to mimic communication processes like receiving responses and commit messages, as well as sending prepare and commit messages. The **transaction\_manager** function orchestrates the process, managing potential coordinator failures and participant responses. Users interact by inputting operations, and the main loop controls the transaction flow. However, this simplified simulation lacks the complexity of real-world systems and serves as an educational example.

**What We Learned:**

1. **Distributed Transactions:** The implementation showcases the coordination required for distributed transactions. The Transaction Coordinator ensures that all participants agree to the transaction before it sends the commit message.
2. **Timeout Handling:** The implementation handles timeouts in both response reception and commit messages. If a participant doesn't respond within a specified time, the coordinator takes appropriate actions.
3. **Randomized Failures:** The implementation introduces random failure scenarios, such as Coordinator failure before sending prepare messages or participants failing to respond within time. This adds a level of realism to distributed systems.
4. **User Interaction:** The implementation interacts with the user to perform operations and initiate transactions, providing a way to observe the transaction process.

**Issues we faced:**

1. How to recover the threads after failure for coordinator and nodes.
2. Communication between coordinator’s thread and node’s thread.
3. After failure, how to read and write state file for recovery without exiting the program.
4. When coordinator fails, we must save the state for coordinator as well as the nodes, but we were not able to save it and because of that, threads used to get exist, we have fixed this by adding time delay.

**SCREENSHOTS: A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**