UE21CS351B - Cloud Computing

Mini Project Report

Building a Task Management Application with Raft Consensus Algorithm and MySQL

Submitted by:

Hamsini Ramesh	**PES2UG21CS183**
Harshitha R	**PES2UG21CS196**
Kashish KC	**PES2UG21CS223**
Khushi Kiran Sheelavantmath	**PES2UG21CS231**

6th Semester D Section

Prof. Kokila Paramanandam

Associate Professor

January - May 2024

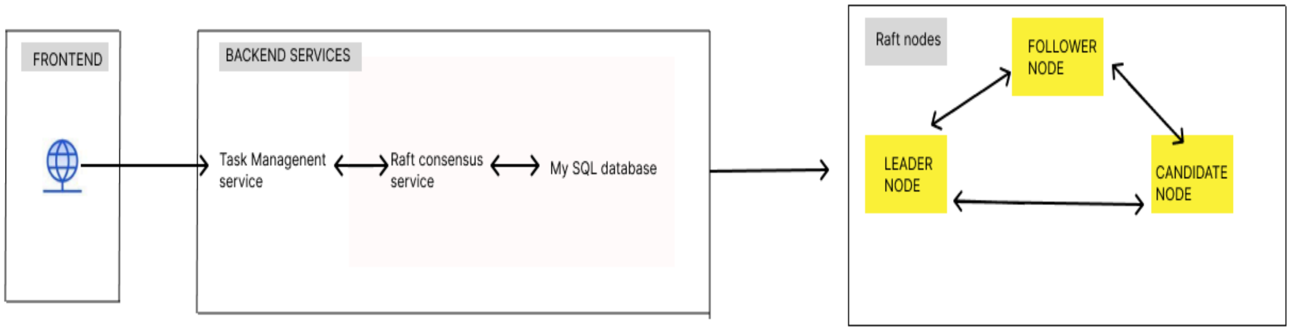
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

Weekly Deliverables:

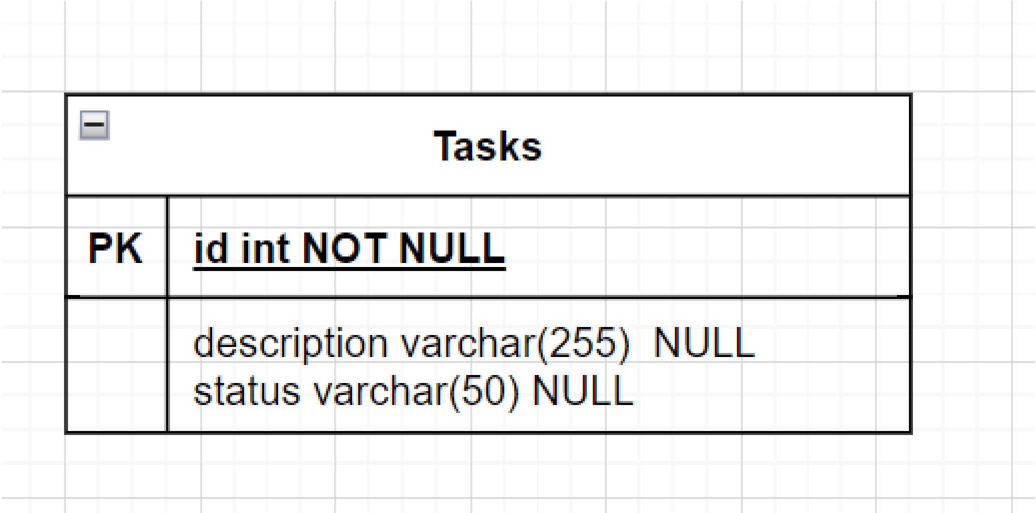
Week1:

1. **Designing the Application Architecture**

- **Architecture**



- **Schema diagram**



Tasks	
PK	id int NOT NULL
	description varchar(255) NULL status varchar(50) NULL

WEEK -2

1.Implementing Raft Consensus Algorithm:

In the Raft consensus algorithm, each node plays a specific role in the process of achieving distributed consensus. The roles include Leader, Follower, and Candidate. Here's a brief explanation of each role:

Leader:

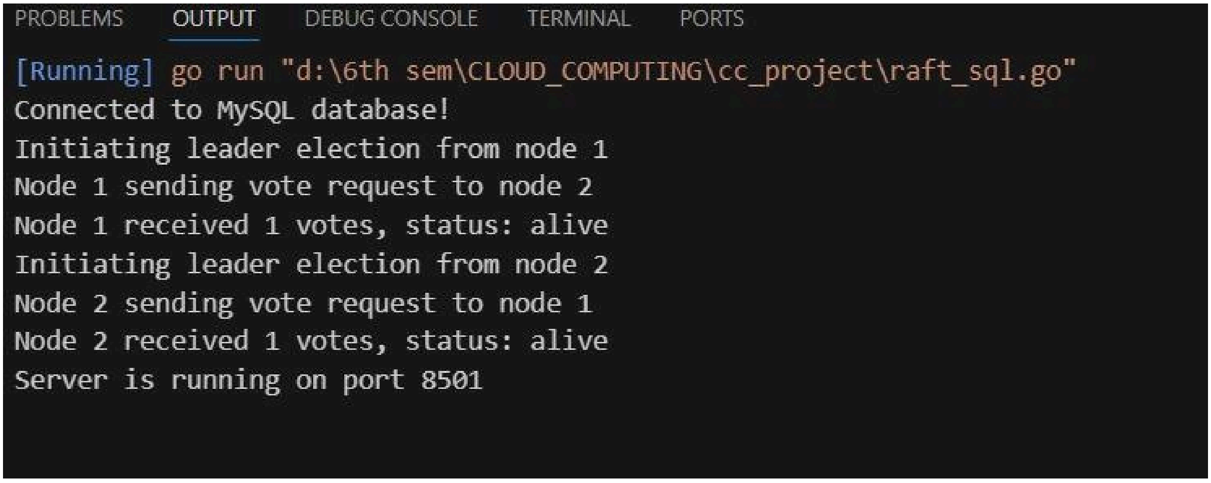
- The Leader is responsible for coordinating the operations of the Raft cluster.
- It receives client requests, appends them to the log, and replicates the log entries to all Followers in the cluster.
- The Leader is the only node that can accept client updates to the system.
- In the event of a split vote during an election, Raft will require a new election to select a new Leader.

Follower:

- Followers are passive nodes that replicate log entries from the Leader.
- They respond to RPC (Remote Procedure Call) requests from Candidates and Leaders, granting votes during leader election rounds.
- Followers do not initiate elections; they follow the leadership decisions made by the Leader or Candidate nodes.

Candidate:

- When a node wishes to become the Leader, it transitions to the Candidate state and initiates a leader election.
- During the election process, a Candidate requests votes from other nodes in the cluster.
- If a Candidate receives votes from a majority of the nodes, it becomes the new Leader.
- If no node wins the election, a new election is triggered after a randomised timeout period.



```
[Running] go run "d:\6th sem\CLOUD_COMPUTING\cc_project\raft_sql.go"
Connected to MySQL database!
Initiating leader election from node 1
Node 1 sending vote request to node 2
Node 1 received 1 votes, status: alive
Initiating leader election from node 2
Node 2 sending vote request to node 1
Node 2 received 1 votes, status: alive
Server is running on port 8501
```

In summary, the Raft consensus algorithm maintains a single Leader node responsible for managing the cluster's state and ensuring consistency among all nodes. Followers replicate the Leader's log and await instructions, while Candidates compete for leadership during leader election rounds. This division of roles ensures that the cluster remains stable, even in the presence of failures or network partitions.

2.Communication protocols implemented:

- Heartbeat Mechanism: Implemented via periodic AJAX calls from the leader to followers, ensuring continuous monitoring and leader authority.

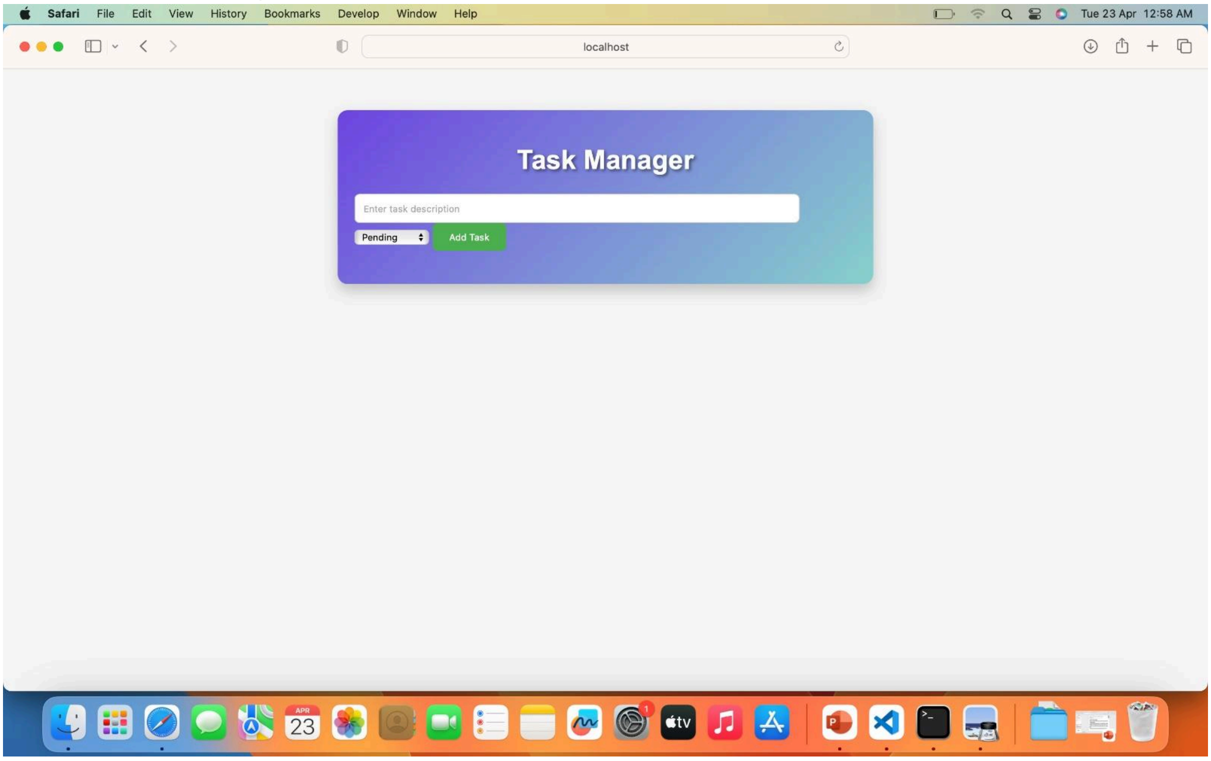
- Log Replication Protocol: Utilises HTTP POST requests for the leader to replicate logs to followers, with JSON payloads containing log entries.

- Election Process: Involves a combination of HTTP GET requests for voting and POST requests for announcing election results with timeouts.

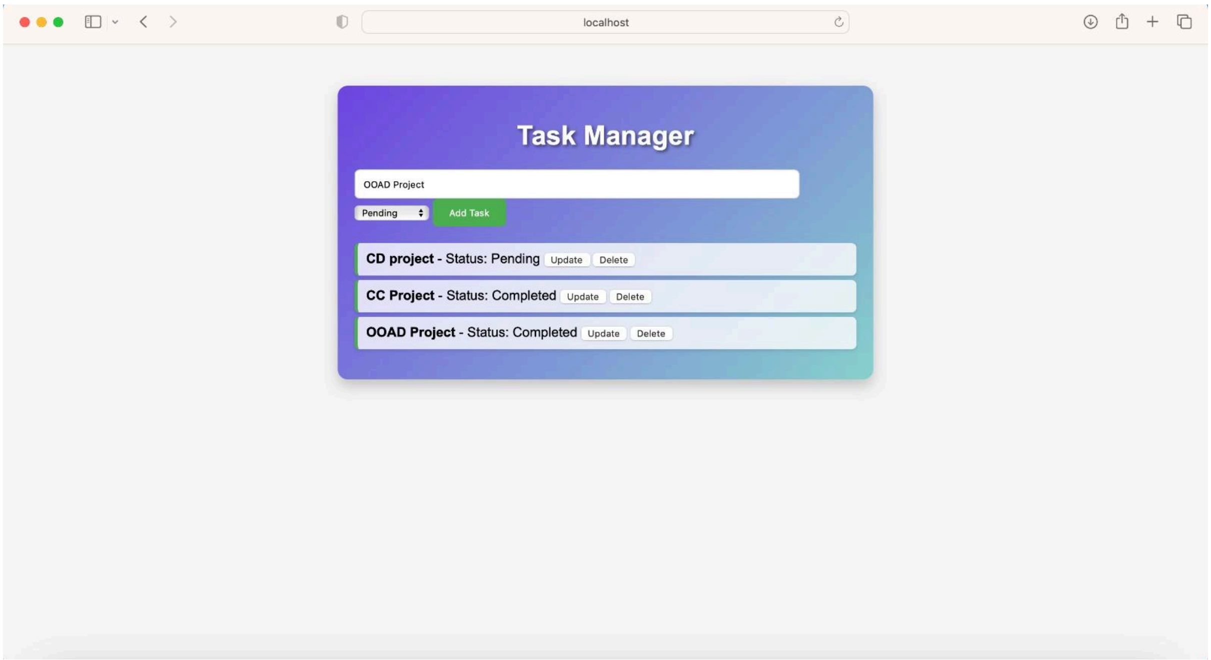
3. Developed Backend :

WEEK-3 PROGRESS:

1) Frontend Design:



- Added task OOAD project



Task Manager

OOAD Project

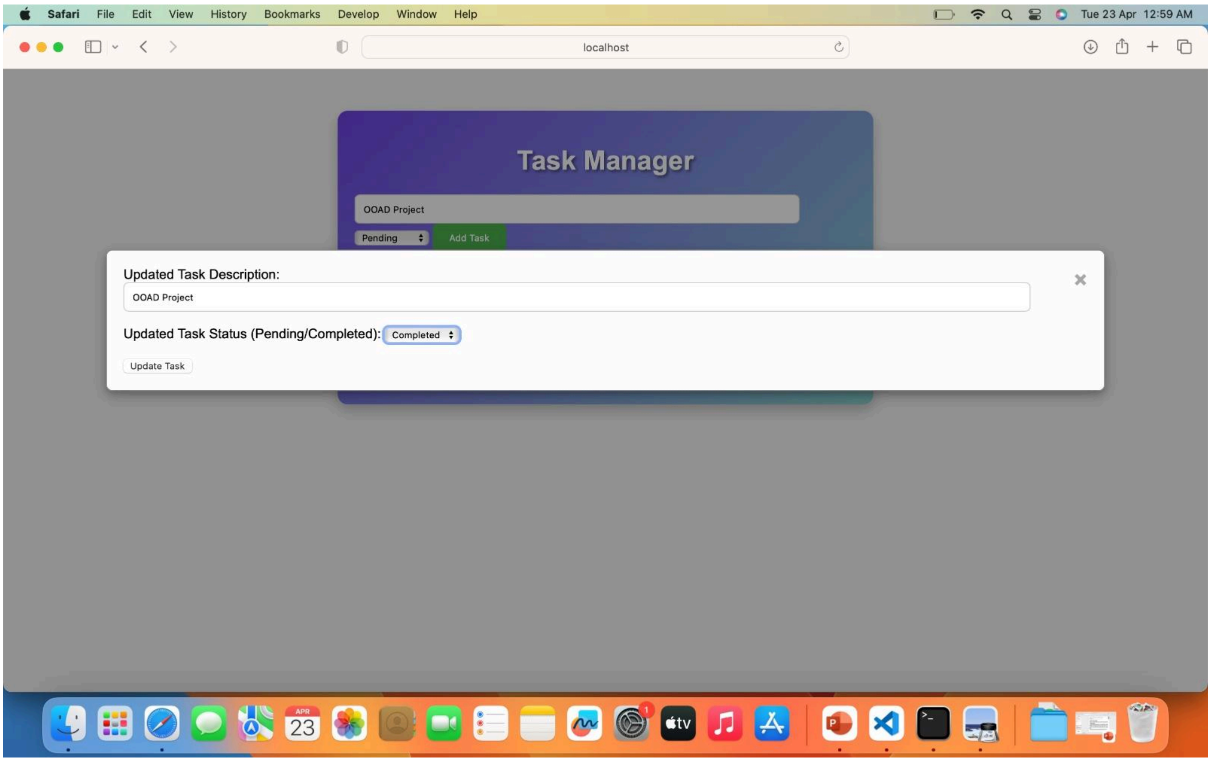
Pending Add Task

CD project - Status: Pending Update Delete

CC Project - Status: Completed Update Delete

OOAD Project - Status: Completed Update Delete

- Updated task OOAD project from Completed to Pending



Task Manager

OOAD Project

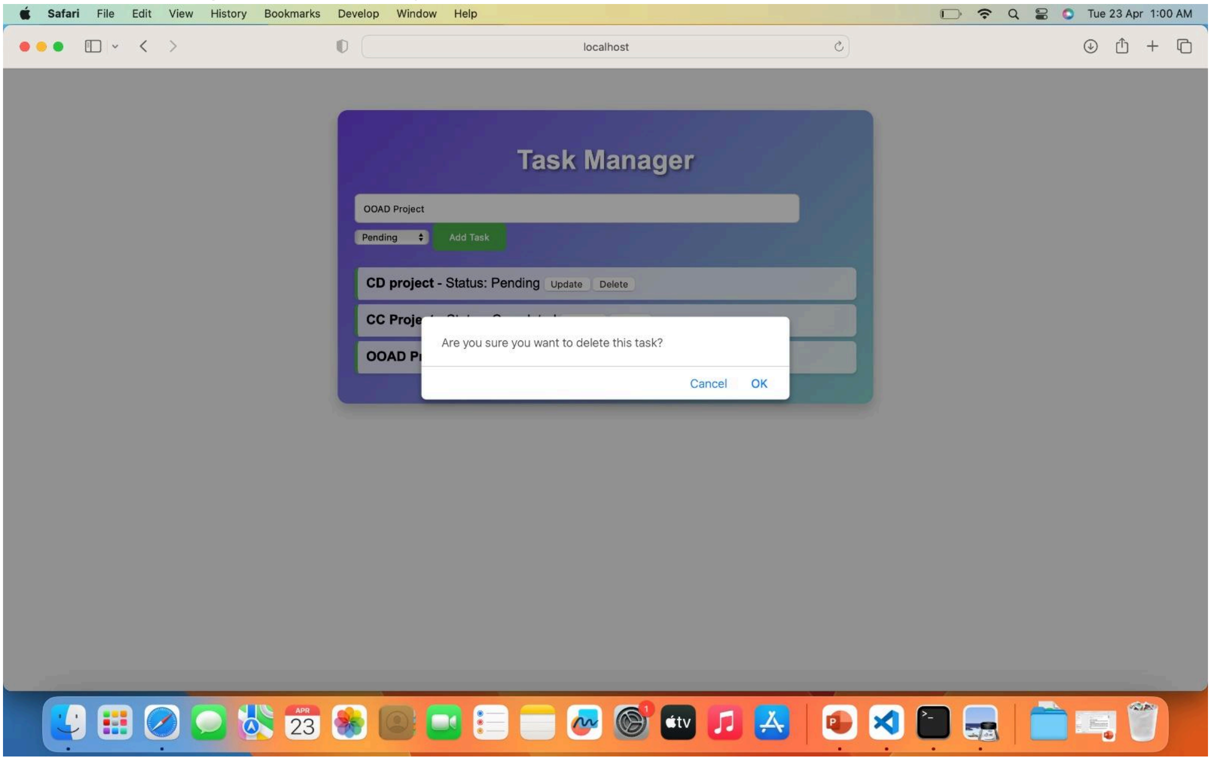
Pending Add Task

Updated Task Description: OOAD Project

Updated Task Status (Pending/Completed): Completed

Update Task

- Deleting task CD Project



Task Manager

OOAD Project

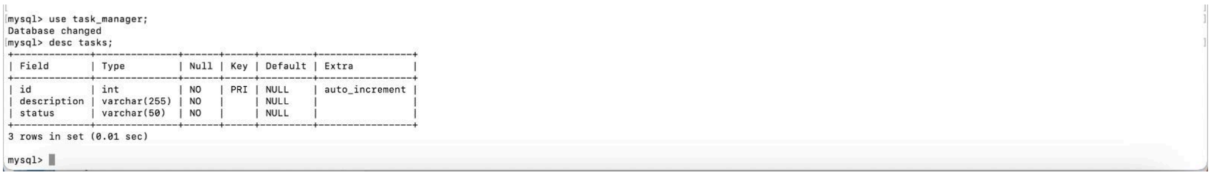
CD project - Status: Pending

CC Proje

OOAD Pr

Are you sure you want to delete this task?

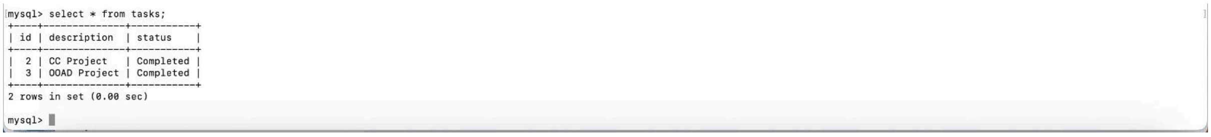
- Table Structure



```
mysql> use task_manager;
Database changed
mysql> desc tasks;
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| id          | int          | NO   | PRI | NULL    | auto_increment |
| description | varchar(255) | NO   |     | NULL    |                |
| status      | varchar(50)  | NO   |     | NULL    |                |
+-------------+--------------+------+-----+---------+----------------+
3 rows in set (0.01 sec)

mysql> 
```

- Updated table after performing CRUD operations



```
mysql> select * from tasks;
+----+--------------+-----------+
| id | description  | status    |
+----+--------------+-----------+
|  2 | CC Project   | Completed |
|  3 | OOAD Project | Completed |
+----+--------------+-----------+
2 rows in set (0.00 sec)

mysql> 
```