**Project Proposal:**

**Project Title:** Work Distributor System

**Problem Statement:**  
Efficiently managing and distributing tasks across multiple workers in a distributed system is a complex challenge. Manual scheduling can lead to inefficiencies, delays, and errors, especially when handling a high volume of tasks. An automated system that can receive tasks from clients, schedule them appropriately, and distribute them to available workers for execution is essential to enhance productivity and ensure timely task completion. This project aims to develop such a system to streamline task management and ensure real-time communication between all participants.

**Proposed Solution:**  
To develop a task management system where clients can submit tasks to a central system that efficiently stores, organizes, and manages tasks. A coordinator will oversee task assignments by ensuring the availability of workers, and communication will be streamlined to enable real-time updates and collaboration.

**Proposed System Architecture:**  
The Work Distributor system will consist of the following components:

1. **Distributor:**
   * The distributor serves as the front-end server of the system.
   * Receives tasks from the clients and distributes them for execution.
   * Accepts tasks via an HTTP-based protocol.
   * Stores tasks in an SQLite database for processing.
   * Maintains a task log and tracks task statuses.
2. **Coordinator:**
   * Accesses the SQLite database to view and manage tasks.
   * Checks the availability of workers before assigning tasks.
   * Communicates task details to workers using WebSocket protocol for real-time updates.
3. **Workers:**
   * Receive task notifications from the coordinator in real time.
   * Update task statuses upon completion or if assistance is required.
   * Communicate directly with the coordinator through WebSockets.
4. **SQLite Database:**
   * Central storage for all tasks and their statuses.
   * Logs information about workers, their availability, and performance metrics.

**Features:**

* Task submission and tracking system.
* Real-time updates on task status using WebSocket protocol.
* Worker availability tracking and task assignment.
* Administrator dashboard for monitoring overall performance.
* Notifications and alerts for task deadlines and status updates.

**Technology Stack:**

* **Backend:** Go for handling task submissions, database interactions, and real-time communication.
* **Database:** SQLite for task storage.
* **Real-Time Communication:** WebSocket protocol for real-time updates.
* **HTTP Protocol:** For Client-Distributor communication.