**JavaScript Console Assignment: Task Manager (CLI Simulation)**

**Objective**

Build a **console-based Task Manager** that allows the user to manage tasks directly through **JavaScript functions** and the **browser console** (or Node.js).

**Requirements**

**1. Data Structure**

* Tasks will be stored in an **array of objects**.
* Each task object will have:
  + id: Unique number (use Date.now() or a counter).
  + name: String - task description.
  + completed: Boolean - true if task is done, false otherwise.

**2. Functions to Implement**

**a) Add Task**

function addTask(name)

* Adds a new task object to the tasks array.
* Ignore empty or whitespace-only tasks.

**b) List All Tasks**

function listTasks()

* Logs all tasks with their status (Completed or Pending).

**c) Mark Task as Completed**

function markTaskCompleted(id)

* Marks a specific task as completed.

**d) Delete Task**

function deleteTask(id)

* Removes a task by id.

**e) Filter Tasks**

function filterTasks(status)

* Accepts completed or pending and logs only tasks matching the filter.

**Starter Code Template**

let tasks = [];

function addTask(name) {

const trimmedName = name.trim();

if (trimmedName) {

const newTask = {

id: Date.now(),

name: trimmedName,

completed: false

};

tasks.push(newTask);

console.log(`Task "${trimmedName}" added successfully.`);

} else {

console.log("Task name cannot be empty.");

}

}

function listTasks() {

if (tasks.length === 0) {

console.log("No tasks available.");

return;

}

console.log("Tasks List:");

tasks.forEach(task => {

console.log(`${task.id} - ${task.name} [${task.completed ? "Completed" : "Pending"}]`);

});

}

function markTaskCompleted(id) {

const task = tasks.find(task => task.id === id);

if (task) {

task.completed = true;

console.log(`Task "${task.name}" marked as completed.`);

} else {

console.log(`Task with ID ${id} not found.`);

}

}

function deleteTask(id) {

const initialLength = tasks.length;

tasks = tasks.filter(task => task.id !== id);

if (tasks.length < initialLength) {

console.log(`Task with ID ${id} deleted.`);

} else {

console.log(`Task with ID ${id} not found.`);

}

}

function filterTasks(status) {

let filteredTasks = [];

if (status.toLowerCase() === 'completed') {

filteredTasks = tasks.filter(task => task.completed);

} else if (status.toLowerCase() === 'pending') {

filteredTasks = tasks.filter(task => !task.completed);

} else {

console.log("Invalid status. Use 'completed' or 'pending'.");

return;

}

if (filteredTasks.length === 0) {

console.log(`No ${status} tasks found.`);

} else {

console.log(`${status.charAt(0).toUpperCase() + status.slice(1)} Tasks:`);

filteredTasks.forEach(task => {

console.log(`${task.id} - ${task.name}`);

});

}

}

// Example Test Flow

addTask("Learn JavaScript");

addTask("Complete Bootcamp Assignment");

listTasks();

markTaskCompleted(tasks[0].id);

listTasks();

filterTasks('completed');

filterTasks('pending');

deleteTask(tasks[0].id);

listTasks();

**How to Use**

* Open your browser console (or use Node.js).
* Copy-paste this code.
* Call the functions directly from the console.
* Example:
* addTask('Learn JavaScript');
* listTasks();
* markTaskCompleted(tasks[0].id);