# **POC Title:**

**Collaborative Task Management System**

### **Objective:** Design and implement a **full-stack task management platform** where users can create, assign, and track tasks in real-time. The system should support **multi-user roles (Admin, Manager, Team Member)**, incorporate robust backend APIs, asynchronous updates, and a modern web-based UI.

### **Project Requirements:** Backend (Java + Spring Boot + Database)

* Implement REST APIs for **Users, Teams, and Tasks**.
* Features:
  + Task CRUD operations (create, update, delete, mark complete).
  + Role-based access control (Admin manages users, Manager assigns tasks, Members update status).
* Use **MySQL or MongoDB** for persistence.
* Provide endpoints for analytics (e.g., tasks completed per user/team).
* Containerize backend with **Docker** (bonus).

#### Middleware / Business Logic

* Auto-assign tasks to team members in round-robin fashion if no assignee is provided.
* Prevent deletion of tasks already marked as completed.
* Implement notification logic (simulate via console logs/API response) when tasks are assigned/updated.
* Ensure concurrency handling for task updates.

#### Frontend (React.js / Next.js + HTML/CSS/JS)

* **User Dashboard:**
  + View assigned tasks, update status, add comments.
* **Manager Dashboard:**
  + Create and assign tasks, view team progress.
* **Admin Dashboard:**
  + Manage users and teams.
* Use **React Router/Next.js routing** for navigation.
* Create a clean, responsive UI with tables and forms.

### **Final Deliverables:**

1. Source code on **GitHub** with clear commit history.
2. SQL/NoSQL script for database setup.
3. README with installation & execution steps.
4. Screenshots or short demo video (optional but preferred).

### **Evaluation Criteria (with marks):**

* Backend APIs & Database Design - 25 marks
* Middleware & Business Logic - 20 marks
* Frontend React/Next.js Implementation -25 marks
* Code Quality & Structure - 15 marks
* Documentation & Deployment Instructions - 10 marks
* Testing & Error Handling - 5 marks
* Creativity & UI/UX Design - 5 marks

### **Guidelines:**

* Follow coding best practices with proper project structure.
* Use Git for version control and maintain meaningful commit messages.
* Ensure APIs are tested using Postman.
* Handle error responses gracefully and validate inputs.
* Complete and submit the project within **7 days**.
* Bonus points for deploying on a free cloud platform (Heroku/Render/Netlify).