# **Hackathon Problem Statement: Digital Kudos Wall (Web App)**

## **Background**

## 

Some reading about Kudos cards if you are not familiar with it - [Kudo Box & Kudo Cards Nurture Intrinsic Motivation - Management 3.0 Practice](https://management30.com/practice/kudo-cards/)[Gratitude is True Intrinsic Motivation, Send a Kudo Today!](https://www.youtube.com/watch?v=0OotzTcZ2a4&ab_channel=Management3.0)

We want to foster a culture of appreciation by building a digital kudos wall that lets colleagues publicly recognise each other’s great work. The system should be simple to use, visually appealing, and easy to extend.

**Note**: The requirements outlined below are intentionally kept high-level and not overly detailed. Teams are encouraged to extend or adapt them to meet the overarching goals. A basic high-fidelity desktop design is provided as a reference for UI and UX.

## **Core Requirements**

1. **Web App**
   * Develop a straightforward web application (no Slack, mobile, or AI integrations).
   * **User Roles**:Introduce two user roles:
     + **Tech Lead**: Can create and view kudos.
     + **Team Member**: Can only view kudos.
   * **Kudos Entry**: When a Tech Lead creates a kudos, they must specify (all mandatory):
     + **Recipient’s Name** (the individual’s name). (Text)
     + **Team Name** (the department or team). (Static dropdown of items)
     + **Category** (e.g., Teamwork, Innovation, Helping Hand) (Static dropdown of items)
     + **Short message** explaining why they deserve recognition (Text).
   * **Kudos Wall:** A main page that displays all kudos publicly. Users can filter or search kudos by recipient, team, or category
2. **Authentication**
   * **Simple Email/Password Signup/Login**:
     + Users can register using their company email address and password.
     + After logging in, they can create and view kudos.
   * Ensure basic security measures for password storage and user sessions.
3. **Layered Architecture & SOPs(standard operating procedures)**
   * Implement a layered architecture (e.g., presentation, business logic, data access).
   * Define and follow SOPs covering design patterns, naming conventions, file/folder organization,coding guidelines and writing tests which Cursor can follow when writing code.
   * Keep the code structure maintainable and easy to extend.
4. **Analytics Dashboard**
   * Include a basic analytics view that shows:
     + Top recognised individuals or teams.
       - Provide options to view results over specific periods (weekly, monthly, quarterly, yearly) to see who’s getting the most kudos.
     + Trending words or categories.
       - For example, if certain keywords (like “collaboration” or “customer-first”) often appear in kudos messages, or if a particular category (e.g., “Teamwork”) dominates, highlight these as trending.
   * This dashboard should provide useful insights into how kudos are being used.
5. **Automated Testing**
   * Implement automated testing to ensure the reliability and correctness of core features.
     + Include unit tests to verify individual functions and components.
     + Include integration tests to ensure different parts of the system (for ex: test APIs like in rev-proxy, top level integration tests in lancer) work smoothly together.
   * The tests should provide reasonable coverage of the main functionalities (e.g., kudos creation, retrieval, filtering, authentication, and analytics).
6. **Deployment / Demo**
   * Provide a working demo by the end of the hackathon.
   * A local container (Docker) or a quick cloud deployment is recommended for demonstration.

## **Brownie Points (Optional Enhancements)**

* **Send Notifications to Basecamp**: When a kudo is given, a small message will automatically be posted in a designated Basecamp project.
* **User Profiles**: Each user has a profile page listing the kudos they’ve received.

## **Evaluation Criteria**

1. **Feature Completion (40%)**
   * Core functionalities (create & display kudos, basic auth, analytics) must be finished and demo-ready.
2. **Layered Architecture (20%)**
   * Cleanly separated concerns, maintainable structure, design patterns where appropriate.
3. **Automation Testing (20%)**
   * Quality and coverage of automated tests.
4. **Standard Operating Procedures Adherence (20%)**
   * Well-defined coding and testing standards, consistently implemented using Cursor AI Editor

## **Summary**

The main focus is building a simple web-based Kudos Wall where employees can sign up, log in, and post kudos. The kudos creation requires a recipient, a team name, a mandatory category, and a short appreciation message. We also need a basic analytics dashboard for insights. While the authentication is kept simple (email/password), we emphasise best practices—layered architecture, SOP-driven coding, and automated testing—to keep the project functional and future-proof. Brownie points for integrating notifications to Basecamp, user profiles, and other fun extras to enhance engagement.

## **Sample UI**

