TEAM PRESENCE MONITOR

Introduction & Project Name

The individual project titled **"Team Presence Monitor"** is an enhancement of the existing **Work Hub** system — a team-based productivity and collaboration platform that enables users to manage projects, tasks, and deadlines efficiently.

While WorkHub allowed effective task and deadline management, it lacked a focused visibility feature for team leads to monitor which members were actively working within each project board. To address this limitation, the **Team Presence Monitor** feature was developed. It focuses on **real-time user presence detection** and **last-seen tracking**, allowing team heads to view who is currently active on a board and when a member was last active. This enhancement increases transparency, accountability, and coordination within the workspace.

Identifying the Business Problem

In the existing WorkHub system, users could efficiently manage tasks and deadlines, but **team heads had no direct way to see who was actively using a project board at any given moment**. This lack of visibility created several practical issues:

- Team heads could not tell who was available or currently engaged on a board.
- Important coordination and quick decisions were delayed due to uncertainty about member availability.
- Participation and engagement were harder to monitor.
- Communication and time-sensitive guidance from managers were less effective without knowing who was active.

Objective of the New Feature

The **Team Presence Monitor** was designed with these objectives:

1. **Show Real-Time Activity:** Display members who are currently active on each project board.
2. **Track Last Seen:** Maintain and display the last active time for members who are offline.
3. **Role-Based Visibility:** Restrict the activity dashboard so **only the team head** can view the active/last-seen status for their board.

4. **Improve Transparency and Responsiveness:** Help team leads make faster, better-informed coordination decisions.
5. **Secure and Persistent Logging:** Store activity records securely in PostgreSQL for auditability and analytics.

Requirements

Functional Requirements

- Show which users are currently active on a project board ("Active Now").
- Display "last seen" timestamps for users who are not currently active.
- Restrict access: only the **team head** of a board can view the activity dashboard for that board.
- Dynamically update presence and typing indicators in real time using Socket.IO.
- Persist activity logs and last-seen timestamps in the PostgreSQL database.
- Load recent activity history when a team head opens the dashboard.

Non-Functional Requirements

- User-friendly, responsive UI integrated into the WorkHub dashboard.
- Real-time updates with minimal latency and efficient Socket.IO usage for scalability.
- Secure backend integration using Flask and PostgreSQL; protect presence endpoints and data.
- Lightweight, performant front-end rendering so the dashboard does not affect board usability.

Tools & Technologies

- **Frontend:** HTML5, CSS3, JavaScript
- **Backend:** Flask (Python)
- **Database:** PostgreSQL
- **Real-Time Communication:** Flask-SocketIO (Socket.IO)

Technical Work and Analysis

Data Model

A new **Activity** (or Presence) model was added to the schema to store:

- user_id — ID of the user.
- board_id — ID of the project board.
- status — current presence state (e.g., online, offline, typing).
- last_seen — timestamp of the last activity.
- updated_at — timestamp of the latest presence update.

Indexes on board_id and user_id help queries for active members and recent history to be fast.

Backend Behavior

- When a user opens a board, the client emits a Socket.IO join_board event. The server:
    - Validates user and board membership.
    - Adds the socket to a room for that board.
    - Updates the user's presence status to online and stores last_seen as now.
- When a user disconnects or becomes idle, the server updates the status to offline and writes the last_seen timestamp.
- Typing indicators and short-lived presence updates are broadcast as lightweight Socket.IO events (typing, stop_typing) without heavy DB writes to reduce latency.
- Only sockets associated with the team head role receive the complete presence dashboard via a dedicated presence_dashboard event endpoint.

Frontend Behavior

- The Team Presence Monitor UI is integrated into the board view as a panel or modal visible to the team head.
- The UI listens to Socket.IO events to update active user lists and last-seen timestamps in real time.
- UI shows user avatars, names, presence status (active/typing/offline), and last seen (e.g., "last seen 12:34 PM, Aug 10, 2025").
- The dashboard fetches initial presence state via a REST endpoint when opened (role-checked) and then subscribes to presence updates over Socket.IO.

Security & Access Control

- Presence APIs and Socket.IO namespaces are secured: authentication tokens and session validation ensure the user is board-member and identifies whether they are the team head.
- Role checks are performed server-side; the server only emits detailed presence data to sockets belonging to the team head.
- Sensitive activity logs are stored securely in PostgreSQL with appropriate access controls.

How This Solves the Business Problem

The Team Presence Monitor directly addresses the earlier visibility gap by:

- Letting team heads instantly identify who is active on a board, enabling faster decisions and timely guidance.
- Providing last-seen context for inactive members so leads know when members were previously engaged.
- Reducing delays and confusion by removing the need to ask teammates their availability externally.
- Improving accountability and participation tracking within the same work environment where tasks are managed.

Result of the Feature Update

After integrating the Team Presence Monitor:

- Team leads can view current active members and last-seen timestamps for each project board.
- Real-time presence updates improve responsiveness and reduce coordination delays.
- Activity logs are preserved for future reference and simple analytics (e.g., participation trends).
- Team transparency and manager oversight have improved, driving better collaboration and faster resolution of blockers.

Conclusion

The **Team Presence Monitor** enriches WorkHub by adding a focused, role-restricted presence and last-seen capability that helps team heads maintain situational awareness of board-level

activity. By combining Flask-SocketIO for real-time events, PostgreSQL for persistent logging, and a lightweight JavaScript UI, this feature brings visibility and accountability into the project workspace without disrupting existing task workflows. The result is a more coordinated, transparent, and efficient team environment.