Full Stack Coding Assignment: Real-Time Collaborative To-Do Board

Objective

Build a web-based collaborative to-do board application where multiple users can log in, manage tasks, and see changes happen in real time—similar to a minimal Trello board but with live sync and some custom business logic.

Assignment Brief

1. Backend (Node.js/Express + MongoDB or PHP + MySQL)

- **User Registration & Login:**
  Secure sign-up/login with hashed passwords and JWT-based authentication.
- **Task API:**
  Tasks have title, description, assigned user, status (Todo, In Progress, Done), and priority.
- **Real-Time Sync:**
  Implement real-time updates using WebSockets (e.g., Socket.IO for Node.js) so all users see changes instantly.
- **Action Logging:**
  Every change (add/edit/delete/assign/drag-drop) should be logged with who did what, when. Expose a REST API to fetch the last 20 actions.
- **Conflict Handling:**
  If two users edit the same task at the same time, detect the conflict and prompt both users to resolve (see details below).

2. Frontend (React, no UI template libraries)

- **Login/Register Pages:**
  Custom-built forms (do not use Bootstrap or any form generator).
- **Kanban Board:**
  Three columns: Todo, In Progress, Done.
  Tasks can be dragged and dropped between columns and reassigned to any user.
- **Activity Log Panel:**
  Shows last 20 actions; updates live.
- **Unique UI:**
  Custom styling only—no third-party CSS frameworks.
- **Animations:**
  At least one custom animation (e.g., card flip, smooth drag-drop, etc.).

- **Responsiveness:**
  The app should work well on both desktop and mobile screens.

3. Unique Logic Challenges

- **Smart Assign:**
  Add a "Smart Assign" button—when clicked on a task, it assigns the task to the user with the fewest current active tasks. You must implement this logic yourself.

- **Conflict Handling:**
  If two users edit the same task at the same time, show both versions and let users choose to "merge" or "overwrite."

- **Validation:**
  Task titles must be unique per board, and must not match column names.

Submission & Deployment Instructions

1. **GitHub Repository**
   - Create a public GitHub repository for your project.
   - Make regular, meaningful commits showing your work and progress.
   - Include a detailed `README.md` with:
     - Project overview
     - Tech stack used
     - Setup and installation instructions (how to run both backend and frontend locally)
     - Features list and usage guide
     - Explanations for your Smart Assign and Conflict Handling logic
     - Link to your deployed live app and demo video
2. **Live Deployment**
   - Deploy both backend and frontend to a free hosting service.
     - **Frontend:** Vercel, Netlify, or similar
     - **Backend:** Render, Railway, Cyclic, Heroku, or similar (Node.js/MongoDB recommended)
   - Make sure your deployed app is accessible and works for demonstration.
   - Include all necessary environment variable setup instructions in your README (do not share secrets or private keys).

3. **Demo Video**
    - Record a 5–10 minute screen recording with voiceover (Loom, Zoom, OBS, or similar tools are fine).
        - Brief intro to your project and tech stack
        - Show login/register, Kanban features, real-time sync, Smart Assign, and conflict resolution
        - Highlight your favorite part or the most challenging part
    - Upload your video to Google Drive, YouTube (unlisted), Loom, or any accessible platform.
    - Include the video link in your README.
4. **Logic Document**
    - Create a 1-page PDF or Markdown file titled `Logic_Document.pdf` or `Logic_Document.md` explaining:
        - How you implemented Smart Assign (in your own words, not code)
        - How your conflict handling works, with examples if possible
5. **Final Submission**
    - Submit the following via the Internshala portal (in the assignment submission section):
        - The GitHub repository link
        - The deployed app URL
        - The demo video link
        - The logic document (uploaded as PDF/Markdown, or a link to it in your repo)
    - Make sure all links are accessible and permissions are set for viewing.

Deadline

Submit all requirements on the Internshala portal by:

11:59 PM IST, 7 days from the date you receive this assignment