Hackathon Task: Todo Task Management Web Application

Objective

Build a full-stack Todo Task Management Web Application that allows users to sign in using social logins (Google, GitHub, Facebook) and manage their personal and shared tasks. Users should be able to perform full CRUD operations, filter tasks, and collaborate with others by sharing tasks. The app should support real-time updates, responsive design, and user-friendly UI/UX. The goal is to showcase your complete engineering skills—from authentication, API design, and database modeling to real-time communication, cloud deployment, and DevOps best practices.

Tech Stack

Backend	Node JS stack - (Express, Nest, etc)
Frontend	React
Database	Any SQL or NoSQL
Deployment	Any free deployment platforms

Scope & Requirements:

Backend:

- RESTful API using your preferred stack
- User authentication via OAuth 2.0 (Google, GitHub, Facebook) using JWT-based session management.
- CRUD operations for tasks, scoped per user and shared users.
- Real-time updates using WebSockets or Server-Sent Events
- Pagination and sorting of tasks.

- Rate limiting & input validation.

Frontend:

- Social media login (Google, GitHub, Facebook or any other) (Any 1 is fine)
- Responsive UI (desktop + mobile)
- Dashboard:
  - Task list with filters (e.g., due today, overdue, by priority/status).
- Task Forms:
  - Create/edit tasks
  - Share tasks with other users via email/username
  - Mark tasks complete/in progress
- Real-time updates to task list without refreshing (using WebSockets or polling).
- Toast messages or modals for user actions (e.g., task updated, deleted).
- Error boundaries and basic offline support.

Deployment:

- Backend:
  - Deploy to AWS Lambda (serverless), Fly.io, Render, or Railway.
- Frontend:
  - Host on Vercel, Netlify, or Firebase.
- Database:
  - Use any hosted database (Supabase, MongoDB Atlas, Neon, etc.)
- Ensure the live app is reachable via a public URL.
- The above given are just examples. You can choose any platform of your choice.

In your submission, you must

1. Create a public repository in github and add a README.md file to follow the instructions and do the setup
2. Add a line "**This project is a part of a hackathon run by https://www.katomaran.com** " in your README.md file's bottom
3. Only publicly deployed submission will be considered for evaluation.

4. Assume anything you want which is not explicitly mentioned in the document and mention your assumption in the README.md
5. Attach an architecture diagram of your app along with your submission
6. Add a loom video(or similar) link in the README.md file where you explain and demonstrate your working app
7. Additional Instructions
    - Your code must be structured, modular, scalable and well-named.
    - Good looking user interfaces will be valued very highly.
    - You will be asked to run the code and explain the application in your final round of interview.
    - Although we would recommend everyone to follow the instructions above, you may decide to ignore a few for an innovative reason, but you must ensure your submission is good so you get a chance in the next round to explain it to us.
    - Use of AI tools like ChatGPT, Replit, Lovable or cursor etc., **is encouraged** and if used, please ensure you retain the prompting you have done to be displayed during the next round of interview.
8. You have 48 hours from receipt of this brief to complete and submit your repository link. Submit your github repository in the google form before **10 AM Wednesday, July 2nd, 2025**

We look forward to seeing your full-stack craftsmanship—from database schema to deployed UI—and hearing you explain your design and implementation choices in the interview.