Hemut Trial Project: Real-Time Q&A Dashboard

Build a Q&A dashboard (3 pages) where users (non-logged-in guests) can submit questions via a web interface, and see questions pop up in real time. All users can respond to questions, but only admins (logged in) can mark questions answered & get notified when new ones arrive.

Frontend (React Native or Next.js):

- A form to submit a question (text input + submit button). Include form validation using AJAX XLMHttpRequest Objects to ensure submitted questions are not blank.

- A live-updating dashboard to view questions.
  - Each question displays:
    - Timestamp: Current time of posting (use system time)
    - Status: **"Pending"**, **“Escalated”**, or **"Answered"**

  - Escalated Messages move to the top of the page, all posts ordered by timestamp
  - Guests have the ability to view, answer, and post questions, but only admins (logged in users) have the ability to mark questions as answered

Backend (FastAPI + AWS or local):

- REST API or gRPC endpoints for:

  - Submitting a question.

  - Getting the current list of questions.

  - Marking a question as answered.

- Real-time updates using:

  - **WebSockets** for pushing new questions to clients.

  - Optional: include **Webhooks** to ping an external service when a question is answered.

Infrastructure:

- Push all code to a GitHub repository, make it public and share the link in the e-mail reply.

- Login & Register pages with input validation, queries DB for valid credentials.

- Forum page where all questions are posted (all users can view/answer)

- Store data in-memory or optionally in a DBMS.
  - Users table: user_id (PK), username, email, password
  - Questions table: question_id (PK), user_id (FK), message
  - Anything else you come up with

- Deploy backend in SQL locally (or AWS if you prefer)

- Use environment variables or a .env file for the config.

Bonus (if you have time):

- Add RAG or Langchain integration to auto-suggest answers to questions (mocked response is fine).

- Add your own major creative feature to the product to showcase your skills!

Requirement	Tech Used
Fast API	Backend API
Next.js or React Native	Frontend
gRPC / WebSockets / Webhooks	Realtime updates & integrations
AWS	Optional backend deployment
Git/GitHub + CI/CD	Source control + deploy simulation
HTTP/HTTPS	API communications
Langchain or Pipecat (Optional)	RAG/Agent integration