Full Stack Developer Assignment – Hirekarma Pvt Ltd

Project Title:

Event Management System

Objective

The goal of this assignment is to evaluate your full-stack development skills (frontend + backend + database integration), your ability to design a clean, responsive UI, and your understanding of modern development practices.

You are expected to build a mini Event Management System where there are two types of users:
1. Admin User → Can perform full CRUD (Create, Read, Update, Delete) operations on events.
2. Normal User → Can view events only.

Tech Stack Requirements

- Frontend: Next.js (or React.js) + Tailwind CSS
- Backend: Python (FastAPI)
- Database: PostgreSQL
- Version Control: Git (host your project on GitHub)

Core Features

Authentication & Authorization

- Signup & Login functionality (for both admin and normal users).
- Maintain user roles (admin, normal).
- Only admins can access event management features.
- Normal users should only see the event list and details.

Event Management

- Admin User
  - Add new events (with details like event name, date, time, description, and image).
  - Edit existing events.
  - Delete events.
  - View all events.

- Normal User
  - Can only view the list of events.
  - Should see event details in a clean, modern UI.

UI Expectations

- Clean, modern design with Tailwind CSS.
- Fully responsive UI (mobile, tablet, desktop).
- Intuitive navigation for both admin and normal users.

Backend Expectations

- Build APIs using FastAPI.
- Use PostgreSQL for event and user storage.
- JWT/session-based authentication (basic implementation is enough).
- Separate routes for admin and normal users.
- Follow clean and modular code structure.

Deliverables

1. Frontend (Next.js/React + Tailwind CSS):
   - Pages:
     - Signup / Login
     - Event List Page (for all users)
     - Event Management Page (for admin)
   - Integration with backend APIs

2. Backend (FastAPI):
   - REST APIs for:
     - Authentication (Signup & Login)
     - Event CRUD operations
   - Database integration with PostgreSQL

3. Database (PostgreSQL):
   - Table: users (fields: id, name, email, password, role [admin/normal])
   - Table: events (fields: id, title, description, date, time, image_url, created_at, updated_at)

4. GitHub Repository:
   - Push your project to a public GitHub repository.
   - Maintain clean commit history.
   - Include a README.md with:
     - Project setup instructions

- Tech stack used
- Screenshots (if possible)

5. Deployment:
- You may use Vercel/Netlify for frontend and AWS/Railway/Render /Heroku for backend.

Time Limit

You will have 1 day to complete this assignment.

Evaluation Criteria

- Signup & Login works (with role-based access: admin vs normal user)
- Completion of core features (admin CRUD + normal user view)
- Code quality, structure, and readability
- UI/UX (clean, modern, responsive)
- Proper use of Git (commits, branches)
- Working integration of frontend, backend, and database
- Bonus: Deployment of the application

Submission

- Share your GitHub repository link with us.
- Share a video demo (upload to drive and share the view link) of your application.
- Share live demo links for frontend and backend.

Note

This is not about building a production-level system but about showing your practical skills, learning speed, and problem-solving mindset.