Positions: Full-Stack Laravel Developer

Task: Build a Task Management Dashboard

Description:

Imagine you are building a task management system that allows users to create, edit, and manage tasks. The system should have a responsive user interface developed with React on the frontend and a robust backend powered by Laravel. The goal is to create a user-friendly dashboard with CRUD operations for tasks, enabling users to easily create, read, update, and delete tasks. Implement task assignment to users and notifications for seamless communication

Requirements:

1. **Backend (Laravel):**
   - Set up a Laravel project with appropriate database migrations for tasks.
   - Create RESTful API endpoints to handle CRUD operations for tasks. **[With Name, Status, Start Date, End Date, Project, Team Members And Descriptions]**
   - Implement validation for task creation and updating to ensure data integrity.
   - Secure the API by implementing appropriate authentication mechanisms (e.g., JWT tokens).
   - Implement proper error handling and return meaningful error messages to the frontend.
   - Implement user notifications for assigning task.
   - Allow users to categorize tasks or assign them to different users.

2. **Frontend (React):**
   - Set up a React project and create a dashboard layout.
   - Implement a task listing page that fetches tasks from the Laravel API and displays them.
   - Create a form to add new tasks and validate the input before making a request to the API.
   - Enable users to edit and delete tasks directly from the dashboard.
   - Implement responsive design principles to ensure a seamless user experience on different devices.

3. **Integration:**
   - Connect the React frontend to the Laravel backend, ensuring smooth data flow.
   - Implement real-time updates for the task list using WebSockets or a suitable technology.
   - Optimize API requests to minimize data transfer and ensure efficient communication.
   - Implement proper handling for scenarios where the user is unauthorized or encounters network issues.
   - Implement sorting and filtering options for the task list.
   - Provide a search functionality to quickly find specific tasks.
   - Implement user authentication and authorization for different roles (e.g., admin, regular user).

Evaluation Criteria:

- Code structure, organization, and readability.
- Proper usage of Laravel for API development.
- Effective integration between React and Laravel.
- User interface design and responsiveness.
- Error handling and validation on both frontend and backend.
- Use of Git for version control.