

house of edtech

WE BUILD BRANDS LIKE NO ONE ELSE COULD DO.

Full Stack Developer Assessment

Mandatory Skills:

Next JS, React.js, Git, Tailwind CSS, MongoDB.

So, you've cleared the first round, here comes the next deal breaker for you that will ensure the required skill sets are matched and you're an ideal candidate for us! It's time for a quick project round:

Task:

Develop a full-stack, secure, and user-friendly CRUD application using Next.js 14 that effectively addresses the requirements of a particular user group or scenario.

Key Requirements:

Technology Stack:

- Backend/Frontend: Next.js 14 (leveraging TypeScript for type safety and code maintainability).
- Database: Select a database management system based on project needs (e.g., PostgreSQL, MySQL, MongoDB).

- **CRUD Functionality:** Create, Read, Update, and Delete operations for data entities relevant to the chosen user group/scenario. Ensure robust data validation and sanitization to prevent security vulnerabilities.
- **User Interface:** Design a clean, intuitive, and responsive user interface that aligns with Next.js 14's component-based architecture and styling capabilities. You can use Tailwind for the CSS. Consider accessibility best practices to cater to users with diverse needs.
- **Deployment:** Deploy the application to a suitable hosting platform (e.g., Vercel, Netlify). Configure continuous integration and continuous delivery (CI/CD) for streamlined development and deployment processes.
- **Code Optimization:** Optimize code for performance and efficiency, considering techniques like code splitting, caching, and server-side rendering (SSR). Write clean, well-structured, and well-documented code that adheres to best practices and style guides.
- **Real-World Considerations:** Address potential real-world challenges that might arise in a production environment, such as scalability, error handling, and security.

Good to Have:

- **Authentication and Authorization:** Implement a secure authentication mechanism (e.g., JWT-based or other preferred approach). Enforce granular authorization rules to control user access to resources and actions.
- **Testing:** Consider integration and end-to-end testing for comprehensive coverage.
- **Security:** Discuss mitigation strategies and contingency plans for these challenges.

Evaluation Criteria:

- **Functionality:** Completeness and correctness of CRUD operations, data validation, authentication, and authorization.
- **User Interface:** User-friendliness, responsiveness, and adherence to accessibility best practices.

- Code Quality: Code structure, organization, documentation, and optimization techniques.
- Testing: Coverage and effectiveness of unit, integration, and end-to-end tests.
- Deployment: Successful deployment and CI/CD configuration.
- Real-World Considerations: Demonstrated understanding of potential challenges and proposed solutions.

Submission Guidelines:

- Share your GitHub Repository along with the Live Deployment.
- Make sure you're name, Github Profile and LinkedIn profile is mentioned in the footer of application.