

House of Edtech

Fullstack Developer Fulltime Assignment 1

Dec 2025

We are seeking highly skilled and motivated individuals proficient in modern web development. Candidates must be good with Node.js, capable of building efficient, scalable, and secure backend solutions, including asynchronous programming, RESTful APIs, database integration (SQL/NoSQL), middleware, and package management.

Expertise in Next.js is crucial, encompassing server-side rendering (SSR), static site generation (SSG), API routes, optimization, routing, data fetching, and deployment.

A strong understanding of basic React.js development is essential, covering **component-based architecture**, **state management** (Hooks, Context API, Query Params), lifecycle methods, and the virtual DOM.

Overall, candidates should possess a comprehensive understanding of full-stack development, from concept to deployment. This includes building robust, scalable, fast, secure, and user-friendly applications while adhering to best practices for code quality and maintainability. Experience with AI technologies is a plus.

What do we expect from you?

We don't expect a basic CRUD application. We want you to showcase your intellect and approach to developing impactful applications, demonstrating sophisticated problem-solving and innovative solutions beyond just technical proficiency. Your project should reflect critical thinking, analytical capabilities, and strategic design for valuable, transformative outcomes, from ideation to scalable implementation.

Mandatory Skills:

Next JS 16, React.js, Git, Tailwind CSS, PostgreSQL/MongoDB

Optional Skills:

Hands-on Experience with AI libraries like AI-SDK, OpenAI, Gemini or Groq!

Task:

Develop a full-stack, secure, and user-friendly CRUD application using Next.js 16 that effectively addresses the requirements of a particular user group or scenario.

Key Requirements:**Technology Stack:**

- **Backend/Frontend:** Next.js 16 (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 16's component-based architecture and styling capabilities. You can use anything related to Tailwind CSS (eg. shadcn, radix-ui) for designing elements. 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.
- **Use AI for add-on features (Optional):** Leverage AI for a wide array of add-on features. AI is a game-changer, revolutionizing various industries and aspects of daily life. Its integration allows for enhanced capabilities, providing a significant competitive edge and opening up new possibilities for innovation and efficiency.

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 your name, Github Profile and LinkedIn profile is mentioned in the footer of the application.