ASIAN IT INC.

MERN STACK DEVELOPER (FULL TIME, REMOTE)

SKILL ASSESSMENT TASK

Project Overview: Develop a robust medicine e-commerce platform using industry-standard technologies and practices. Implement features such as user authentication with JWT, comprehensive product management including CRUD operations with Redux and RTK Query with Axios, dynamic category management. Ensure seamless user experience with responsive design, efficient data handling with Express.js, Mongoose, and extensive error handling. Utilize SSR and SSG for optimized performance, implement SEO strategies for visibility, and create an intuitive dashboard for efficient admin management of orders, users, and inventory.

Technology Stack

- **Programming Language:** TypeScript
- **Web Framework:** Next.js
- **Backend:** Express.js, Mongoose
- **Authentication:** JWT (JSON Web Tokens)
- **State Management:** Redux, RTK Query, Axios
- **Email Service:** Nodemailer
- **Styling:** Tailwind CSS
- **Notification:** React Toastify
- **Modal:** React Portal

Integration of Backend Functionality:

To integrate the backend functionality for the medicine e-commerce platform, implement authentication, CRUD operations for various entities, and ensure a secure and well-structured approach. Here's a breakdown:

1. Authentication:

User Registration:

- Inputs: Name, Email, Password, and Photo.
- Storage: Store the photo publicly in a static directory.
- Validation: Implement rigorous email and password validation.
- Email Verification: Send a verification code to the user's email and ensure that the user cannot log in until their email is verified. Display a countdown timer for 59 seconds after which the user can request a new OTP if needed.

Login:

- Mechanism: Users log in with email and password.
- Tokens: Implement JWT for authentication.
    - Access Token: Valid for 1 hour.
    - Refresh Token: Valid for 1 day.
    - Token Refresh: Allow generating a new Access Token using the Refresh Token. If the Refresh Token expires, prompt the user to log in again.

2. User Roles:

- Roles: "Super admin," "admin," and "user."
- Functionality: Different functionalities based on user roles.

3. CRUD Operations: In Every CRUD Operations Use: Redux, RTK Query with Axios

Categories:

- Primary Category: CRUD operations for Name, Slug, and Thumbnail.
- Secondary Category: CRUD operations for Name, Slug, and Thumbnail.
- Tertiary Category: CRUD operations for Name, Slug, and Thumbnail.

Variants:

- CRUD Operations: Manage Name and Price for each variant.

Products:

- CRUD Operations: Manage Name, Slug, Photos (multiple), Description, Meta Key, Price, Discount, Stock Status (true/false), Status (active/inactive), and Categories.
- Relation: Products relate to Primary, Secondary, and Tertiary Categories, and Variants.
- Pricing: Different prices for different variants, default price if no variant is selected.

Shipping Address:

- CRUD Operations: Manage Division, District, Sub-District, Address, Name, and Phone.

Orders:

- CRUD Operations: Manage order details including status and filtering by date.

4. Backend Implementation:

Technology Stack:

- Framework: Express.js (for RESTful API), Mongoose.
- Authentication: Implement JWT-based authentication using libraries like jsonwebtoken and bcrypt for hashing passwords.
- Database: MongoDB with Mongoose to manage data.

Code Structure:

- Controllers: Handle the logic for user authentication, CRUD operations, etc.
- Models: Define the schema for users, products, categories, variants, and orders.
- Routes: Define endpoints for accessing different functionalities.
- Middleware: Implement authentication middleware for protecting routes.

5. Frontend Integration:

Registration and Login Forms:

- Handle form submissions and validations.

Admin Dashboard:

- Create pages for managing users, products, orders, and categories.

Product Pages: Show product according to category

- Implement product detail pages, add-to-cart functionality, and related products.

6. Add to Cart Functionality:

Button State Change:

- Initially, display "Add to Cart" on the product page.
- Add to Cart: When clicked, add the product to the shopping cart and change the button to "View Cart."
- View Cart: Clicking "View Cart" opens a modal displaying the shopping cart.

Product Variants:

- MG Options: If the product has different mg options, allow the user to select a specific mg.
- Price Update: Update the price based on the selected mg before adding the product to the cart.

7. Shopping Cart Functionality:

Manage Quantities:

- Allow users to increment or decrement product quantities using input fields and buttons.

Stock Handling:

- Display "Stock Out" if a product's quantity reaches zero. Prevent addition or increment of such products.

8. Cart Summary:

- Summary Calculation: Include the total price, discounts applied, and shipping charges in the cart summary.

9. Modal Interaction:

- Cart Modal: Clicking on a shopping cart item opens the Shopping Cart modal.

10. State Management:

- Redux & Redux-Persist: Use Redux for state management and redux-persist to persist shopping cart data.

11. Checkout Process:

- Checkout Redirection: Clicking "Proceed to Checkout" will redirect users to the shipping page if logged in; otherwise, prompt login.

12. Product Details Page:

- Information Display: Show all details about the product, with options to add it to the cart.
- Quantity Management: Allow users to increase or decrease product quantities.
- Button State: Change the "Add to Cart" button to "View Cart" once the product is added.

Redux Integration:

- Use Redux for managing cart state and actions.
- Configure redux-persist to keep cart data across sessions.

13. SEO and Performance:

- SEO: Implement static and dynamic meta tags using Next.js's next/head.
- Performance: Use SSR and SSG for pages where appropriate.

14. Deployment and Testing:

- Deployment: Ensure the application is properly deployed on a platform like Vercel or Heroku.
- Testing: Write tests for API endpoints and frontend components to ensure functionality and security.

15. Documentation:

- README: Provide clear instructions for running the application locally and any necessary environment variables.

Note:

1. Pagination should be used on all admin dashboard tables and product pages.
2. Ensure 100% error handling on all functions and pages.
3. All components must be reusable.
4. Follow industry-standard coding practices.
5. Maintain an industry-standard file structure.
6. This task should reflect best practices regarding code organization, security, and user experience. It should demonstrate proficiency in using the specified technologies.
7. The .env file cannot be ignored and must be provided.
8. Include meaningful comments in the code.
9. All variables must use types.
10. Use Redux for all state management.

Submission:

Send your completed job tasks to recruit@asianitinc.com

- GitHub Repository: Ensure it includes all code and a README file.
- You must provide .env file
- You must provide frontend and backend github repository separately.
- You must provide the credentials of super admin and admin.
- Live Deployment: Provide a link to the deployed application.
- Database Design: Include a PDF file outlining the database schema.

Deadline:

- **August 15, 2024, 11:59 PM**

Submission Rule:

- Task-related assistance: Asking for task-related help from any team members of ASIAN IT INC. is strictly prohibited. Such behavior will result in immediate rejection of your assessment.
- Ensure adherence to the submission deadline and rules mentioned above.
- For any questions or further clarification, please reach out. We eagerly anticipate reviewing your submissions and assessing your skills for the MERN Stack Developer (Full Time, Remote) at ASIAN IT INC.