

# Clotree Full Stack Project - Interview Q&A

## Frontend (React.js) - Questions & Answers

Q: What is the tech stack used in this project?

A: I used the MERN stack - MongoDB for database, Express.js for backend API, React.js for frontend UI, and Node.js for the server. I also used Cloudinary for image uploads and JWT for authentication.

Q: How did you manage the global state in your app (e.g., cart, user auth)?

A: I used React Context API (or Redux if used) to manage global states like cart items, user login status, and admin roles across the app.

Q: How did you implement routing in your app?

A: I used React Router DOM for client-side routing. I created public routes and protected routes with PrivateRoute components.

Q: How did you handle conditional rendering for login/logout/admin?

A: I checked user status using values from global state or localStorage. Based on isLoggedIn or isAdmin, I conditionally rendered buttons or redirected the user to proper pages.

Q: Did you use any styling method or library?

A: I used Tailwind CSS for fast and responsive styling. It helped in building a clean and mobile-friendly UI quickly.

Q: How did you fetch and display product data?

A: I used Axios to fetch data from backend APIs and displayed it using React state and useEffect.

Q: How does your app handle form validation?

A: I used HTML5 validation for basic forms and added extra validation using JavaScript before submitting the form.

## Backend (Node.js, Express, MongoDB)

Q: How is your backend structured?

## Clotree Full Stack Project - Interview Q&A

A: My backend follows the MVC pattern - with models for MongoDB schemas, controllers for logic, routes for endpoints, and a separate config folder for DB connection and settings.

Q: How is authentication and authorization implemented?

A: I use bcrypt to hash passwords and JWT tokens for secure login sessions. The token is stored in localStorage and used for protected routes.

Q: How is the admin role handled?

A: Each user has a field like isAdmin. In protected routes, I check req.user.isAdmin === true before allowing actions like add/edit/delete product.

Q: How are images uploaded?

A: I used Cloudinary to upload images and store the returned secure\_url in MongoDB.

Q: How is error handling done?

A: I use middleware functions in Express to catch errors using try-catch and send proper JSON error messages to the frontend.

Q: How are your API routes protected?

A: I use middleware to verify JWT tokens before allowing users to perform protected actions.

## Full Stack / Integration

Q: How is frontend connected to backend?

A: I use Axios in the frontend to make HTTP requests to Express API. I use .env files to manage API URLs securely.

Q: How is cart/order data synced?

A: When users add items to cart, it's stored in local state or localStorage. On checkout, the cart is sent to backend and stored in MongoDB.

## Clotree Full Stack Project - Interview Q&A

Q: How is user session maintained?

A: JWT token is stored in localStorage and used in headers for API requests to keep the user logged in.

Q: Where is your app deployed?

A: Frontend is deployed on Vercel, and the backend on Render. MongoDB is hosted on MongoDB Atlas.

### Advanced / Bonus

Q: What would you improve?

A: I want to add a payment gateway, product reviews, search with filters, and a better admin dashboard.

Q: How would you scale this app?

A: I would implement pagination, lazy loading, use Redis for caching, and move to cloud platforms like AWS.

Q: How would you implement search and filter?

A: By sending a query to the backend using MongoDB regex or \$text search. Filters via query parameters.

Q: How do you protect environment variables?

A: I store secrets in .env files and never push them to GitHub. On deployment, I set them in the platform's settings.