process

System Process Workflow

- ***User Interface Layer (React):*** The process initiates with a Single Page Application (SPA) architecture where React components manage the Document Object Model (DOM) to provide a seamless user experience without page refreshes.
- ***Request Management:*** User actions, such as book queries or authentication, trigger asynchronous API calls to the backend using the Fetch API .
- ***Backend Logic Layer (Node.js & Express):***
    - ***Routing:*** The Express.js server intercepts requests and routes them to specific controllers based on the RESTful endpoint accessed.
    - ***Middleware:*** The system executes security checks and data validation—similar to the logic used in your PHP login systems—before processing the core request.
- ***Data Interaction (JSON):*** The server interacts with a structured JSON-based data store to perform CRUD (Create, Read, Update, Delete) operations on book and user records.
- ***Response & State Update:*** The backend returns a JSON payload; the React frontend then updates the global state, triggering the notification system to alert the user of successful actions or errors.
- ***Cloud Deployment Pipeline:*** The entire lifecycle is managed through a Continuous Deployment (CD) pipeline on **Netlify**, ensuring real-time synchronization between the codebase and the live application.

Proposed Design of product….

The "Product" design focuses on the final result—the software itself and its static structure.

- **Software Architecture:** A high-level view of how the system is organized. For vookapp, this is a **decoupled Full-Stack architecture** where the React client (Netlify) is separated from the Node.js server (Render).
- **Database Design:** Since you are using a **`data.json` flat-file database** on the Render server, the design specifies the schema for book metadata (ISBN, title, author) and user records.
- **User Interface (UI) Design:** The conceptual layout of the web application, focusing on a responsive, single-page experience for both library staff (admin) and students.
- **Module Decomposition:** Breaking the system into manageable parts, such as:
  - **User Management:** Handling account creation and secure authentication.
  - **Catalog Management:** Providing logic for adding, updating, and searching book records.
  - **Transaction Management:** Managing the state of borrowed, reserved, or returned books.