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.

Implementation of Proposed Design of product…

1. Frontend Implementation (Client-Side)

- **Environment:** Developed using **React.js** and deployed on **Netlify** for high availability and global content delivery.
- **Component Architecture:** Created a modular UI where independent components manage specific views (e.g., `BookList`, `SearchBar`, `Login`) to ensure a maintainable codebase.
- **State Management:** Integrated React Hooks to manage application state locally, ensuring the UI reflects data changes instantly without full page reloads.
- **API Integration:** Configured the frontend to perform cross-origin (CORS) asynchronous requests to the live Render backend URL, replacing local development endpoints.

2. Backend Implementation (Server-Side)

- **Environment:** Hosted on **Render**, utilizing a **Node.js** and **Express.js** runtime environment for the `server.js` file.
- **RESTful API Development:** Built server-side endpoints to handle business logic, such as validating user credentials and managing book borrowing transactions.
- **Security Middleware:** Implemented server-side logic to protect routes and validate incoming data payloads before they interact with the data layer.

3. Data Layer Implementation

- **Data Store:** Utilized a flat-file **`data.json`** database residing directly on the Render server environment.
- **File I/O Operations:** Implemented Node.js File System (`fs`) modules to perform real-time read and write operations on the `data.json` file for inventory updates.
- **Data Structure:** Organized data in a lightweight JSON format to ensure fast parsing and minimal "Round Trip Time" (RTT) during data exchange.

4. Branding & Finalization

- **Project Identity:** Integrated the professional signature **PriyanshuBejJVM** into the application's footer and code metadata to establish ownership and branding.
- **Deployment Pipeline:** Established a Git-based continuous deployment workflow where frontend updates trigger Netlify builds and server-side updates trigger Render redeployments.

5.Live Deployment: The application is fully deployed and accessible at https://bookmanagement7thsemester.netlify.app/.