**Project Design Phase**

**Solution Architecture**

|  |  |
| --- | --- |
| Date | 20 June 2025 |
| Team ID | LTVIP2025TMID53930 |
| Project Name | BookNest: Where Stories Nestle |
| Maximum Marks | 4 Marks |

**Solution Architecture:**

**The solution architecture of BookNest bridges the gap between traditional bookstore limitations and a scalable, digital-first MERN stack web platform. Its architecture is structured to serve three key user roles: Users, Sellers, and Admins, each with tailored capabilities. The goals of the architecture are:**

* **✅ Define role-based access and UI/UX flow for Users (browse/order), Sellers (add/manage books), and Admins (manage system-wide operations).**
* **✅ Design scalable, component-based frontend using React.js and Tailwind CSS with routing based on user roles.**
* **✅ Use Express.js for RESTful API development, enabling communication between frontend and backend with secured endpoints.**
* **✅ Integrate MongoDB database, modeling Users, Books, and Orders with Mongoose schemas.**
* **✅ Implement JWT-based authentication and middleware, providing secure, verified access for all operations.**
* **✅ Support functional modules like cart management, order placement, and status tracking with a clean, maintainable codebase.**

**Example - Solution Architecture Diagram:**

**+----------------------+**

**| React Frontend |**

**| (User, Seller, Admin)|**

**+----------+-----------+**

**|**

**v**

**+----------------------+**

**| Express.js Server |**

**| REST API + Middleware|**

**+----------+-----------+**

**|**

**+-----------+-----------+**

**| MongoDB Database |**

**| Users | Books | Orders|**

**+------------------------+**

** Frontend: React + React Router (Role-based Views)**

** Backend: Node.js + Express (API & Auth)**

** Database: MongoDB (Mongoose Schemas)**

** Security: JWT Tokens, Role Validation Middleware**