

<b>BookNest: Where Stories Nestle</b>	Team ID: LTVIP2026TMIDS24289
Team Leader: Keerthi Mekala	Member 1: Adil Mohammed
Member 2: Vasanth Metla	Member 3: Rahimtulla Shaik

## Project Design Phase - Solution Architecture

This document describes the solution architecture of BookNest — a full-stack MERN web application. The architecture bridges the gap between user requirements and the technical implementation across frontend, backend, and database layers.

### Architecture Overview

BookNest follows a 3-tier client-server architecture where the React frontend communicates with the Express.js backend through RESTful APIs, and the backend interacts with a MongoDB Atlas cloud database using Mongoose ODM.

TIER 1: Frontend	TIER 2: Backend	TIER 3: Database
<ul style="list-style-type: none"> <li>React.js (Vite)</li> <li>React Router DOM</li> <li>Bootstrap + CSS</li> <li>Axios HTTP Client</li> <li>Port: 5173</li> </ul>	<ul style="list-style-type: none"> <li>Node.js</li> <li>Express.js</li> <li>REST API Routes</li> <li>bcrypt (Auth)</li> <li>Port: 4000</li> </ul>	<ul style="list-style-type: none"> <li>MongoDB Atlas</li> <li>Mongoose ODM</li> <li>5 Collections</li> <li>Cloud Hosted</li> <li>Scalable</li> </ul>

*Browser ←→ React Frontend ←→ Express API ←→ MongoDB Atlas*

### Frontend Architecture

The frontend is a Single Page Application (SPA) built with React.js and Vite. It uses React Router DOM for client-side routing and Axios for API communication.

Module	Pages / Components	Description
User Module	Home, Login, Register, Wishlist, Orders, Book Detail	Buyer-facing pages
Seller Module	Seller Login, Register, Dashboard, Add Book, Orders	Seller management pages
Admin Module	Admin Login, Dashboard, Users, Sellers, Books	Admin control pages
Shared Components	Navbar, Footer, Book Card	Reusable UI components

## Backend Architecture

The backend is built with Node.js and Express.js, exposing a RESTful API on port 4000. It handles all business logic, authentication, and database operations.

Route Group	Endpoints	Description
User Auth	POST /signup, POST /login	Register and login buyers
Seller Auth	POST /ssignup, POST /slogin	Register and login sellers
Admin Auth	POST /asignup, POST /alogin	Register and login admins
Books	POST /items, GET /item, GET /item/:id, DELETE /itemdelete/:id	CRUD operations for books
Orders	POST /userorder, GET /getorders/:id, GET /getsellerorders/:id	Order management
Wishlist	POST /wishlist/add, GET /wishlist/:id, POST /wishlist/remove	Wishlist management
Admin	GET /users, GET /sellers, DELETE /userdelete/:id, DELETE /sellerdelete/:id	Admin platform management

## Database Architecture

BookNest uses MongoDB Atlas with Mongoose ODM. The database contains 5 collections:

Collection	Fields
users	_id, name, email, password
sellers	_id, name, email, password
admins	_id, name, email, password
items (books)	_id, title, author, genre, description, price, userId, userName, itemImage
orders	_id, userId, sellerId, bookId, title, price, buyerName, buyerAddress, createdAt
wishlists	_id, userId, bookId, title, author, price, itemImage

## Security Architecture

- Passwords are hashed using bcrypt before storing in MongoDB
- User session data (id, name, email) stored in browser localStorage after login
- Role-based routing: separate login and dashboard for User, Seller, and Admin
- CORS enabled on backend to allow cross-origin requests from frontend

- Environment variables managed via dotenv for database connection security

Team ID: LTVIP2026TMIDS24289 | Team Leader: Keerthi Mekala | Members: Adil Mohammed, Vasanth Metla, Rahimtulla Shaik