### **Project Design Phase-II**

### **Technology Stack (Architecture & Stack)**

**Date**: April 14, 2025

**Team ID**: SWTID1743607143

**Project Name**: BookEase

**Maximum Marks**: 4 Marks

### **Technical Architecture**

The BookEase platform utilizes a MERN stack architecture to provide a local, scalable e-commerce solution for book trading, supporting user, seller, and admin functionalities with a responsive UI.

### **Table-1: Components & Technologies**

| **S.No** | **Component** | **Description** | **Technology** |
| --- | --- | --- | --- |
| 1 | User Interface | Web UI for browsing books, managing cart, admin tasks; homepage includes sparkle animations | React 18.x, Tailwind CSS, Lucide React |
| 2 | Application Logic-1 | Manages user authentication and role-based navigation | Node.js 20.x, Express.js 4.x, JWT |
| 3 | Application Logic-2 | Handles book catalog browsing and seller inventory management | Node.js, Express.js, Mongoose 8.x |
| 4 | Application Logic-3 | Processes admin tasks, including user and seller deletion with notifications | Node.js, Express.js |
| 5 | Database | NoSQL database storing users, books, orders; JSON-like documents | MongoDB 7.x |
| 6 | Cloud Database | Not implemented | None |
| 7 | File Storage | Stores book images and static assets locally | Local Filesystem |
| 8 | Infrastructure | Local development and deployment | Local: Node.js server |

### 

### 

### 

### **Table-2: Application Characteristics**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
| 1 | Open-Source Frameworks | Frameworks for frontend, backend, database, and styling | React, Express.js, MongoDB, Mongoose, Tailwind CSS |
| 2 | Security Implementations | Token-based authentication, password hashing, secure APIs | JWT, bcrypt |
| 3 | Scalable Architecture | 3-tier architecture (UI, logic, data) supports user growth with sharding | MongoDB sharding, Express.js |
| 4 | Availability | Local replication ensures data availability | MongoDB Replica Sets |
| 5 | Performance | Optimized queries and caching support 1000 requests per second | Redis, MongoDB indexes |

### **References**

* <https://c4model.com/>
* <https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/>
* <https://www.ibm.com/cloud/architecture>
* <https://aws.amazon.com/architecture>
* <https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d>