**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

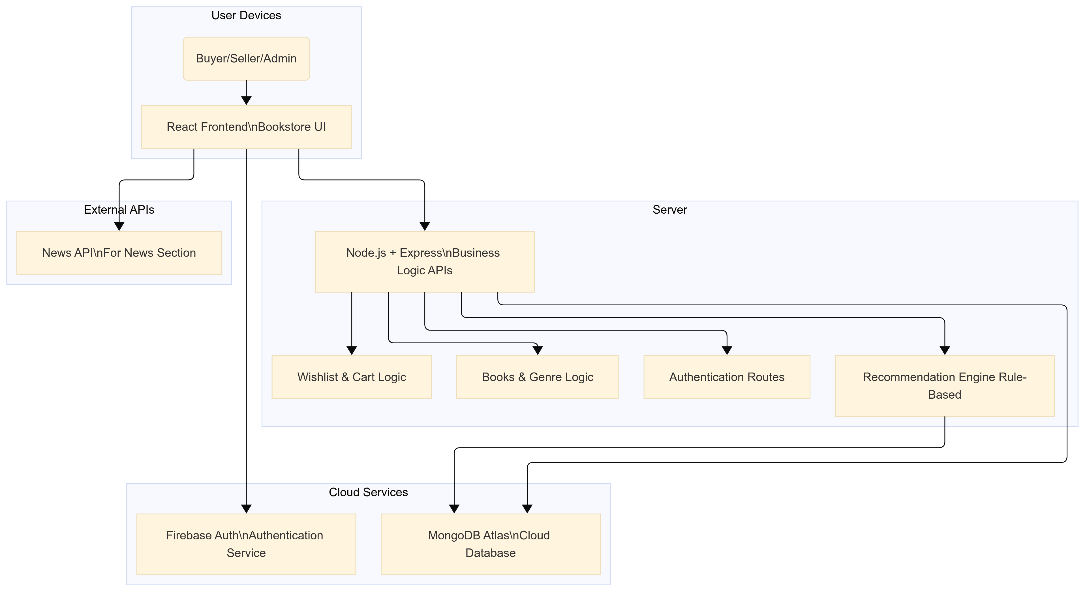
|  |  |
| --- | --- |
| Date | 18 March 2025 |
| Team ID | SWTID1743689974 |
| Project Name | Book Store |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

**Example: Order processing during pandemics for offline mode**

**Reference:** [**https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/**](https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/)

****

**Architecture Summary**

* **Users** (Admin, Seller, Buyer) interact via **React UI**
* **Firebase** handles authentication
* **MongoDB Atlas** is your cloud-hosted data layer
* The **Backend** (Node + Express) manages routes and logic
* **Optional ML** model can enhance book recommendations
* All frontend/backend run **locally** or via **cloud deployment** (like Vercel/Render)

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

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
| 1. | User Interface | Web-based interface for users to interact with the bookstore | HTML, CSS, JavaScript, React.js |
| 2. | Application Logic-1 | User Authentication and Role-based Access Management | Node.js, Firebase Authentication |
| 3. | Application Logic-2 | Book Management (CRUD by Seller/Admin) | Node.js, Express.js |
| 4. | Application Logic-3 | Wishlist, Cart, and Personalized Recommendation Logic | Node.js, Express.js |
| 5. | Database | Stores user profiles, books, wishlist, cart, orders | MongoDB (NoSQL) |
| 6. | Cloud Database | Cloud-hosted database instance | MongoDB Atlas |
| 7. | File Storage | Stores book images and related media | Firebase Storage / AWS S3 / Cloudinary |
| 8. | External API-1 | Book Metadata / Book Cover fetch (Optional) | Google Books API |
| 9. | External API-2 | News updates related to books/literature | NewsAPI.org / NY Times Books API |
| 10. | Infrastructure (Server/Cloud) | Deployed on cloud platform for scalability | Render / Vercel for frontend Railway / Heroku for backend |

**Table-2: Application Characteristics:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Characteristics** | **Description** | **Technology** |
| 1. | Open-Source Frameworks | Utilizes widely adopted open-source technologies for frontend, backend, and database management | React.js, Node.js, Express.js, MongoDB, Tailwind CSS |
| 2. | Security Implementations | Firebase Authentication for access control, HTTPS for secure data transfer, route protection | Firebase Auth, JWT, HTTPS, Firebase Rules, Helmet.js |
| 3. | Scalable Architecture | 3-tier architecture (Frontend-Backend-Database) with cloud deployment for horizontal scaling | MERN Stack, Render/Vercel, MongoDB Atlas (auto-scaling) |
| 4. | Availability | Hosted on cloud with automatic failover and high availability using distributed architecture | MongoDB Atlas (multi-region), Render, Firebase Hosting |
| 5. | Performance | React virtual DOM for fast UI rendering, API caching, CDN for static files, optimized queries | React.js, Express.js, MongoDB indexes, Vercel CDN |

**References:**

[**https://c4model.com/**](https://c4model.com/)

[**https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/**](https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/)

[**https://www.ibm.com/cloud/architecture**](https://www.ibm.com/cloud/architecture)

[**https://aws.amazon.com/architecture**](https://aws.amazon.com/architecture)

[**https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d**](https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d)