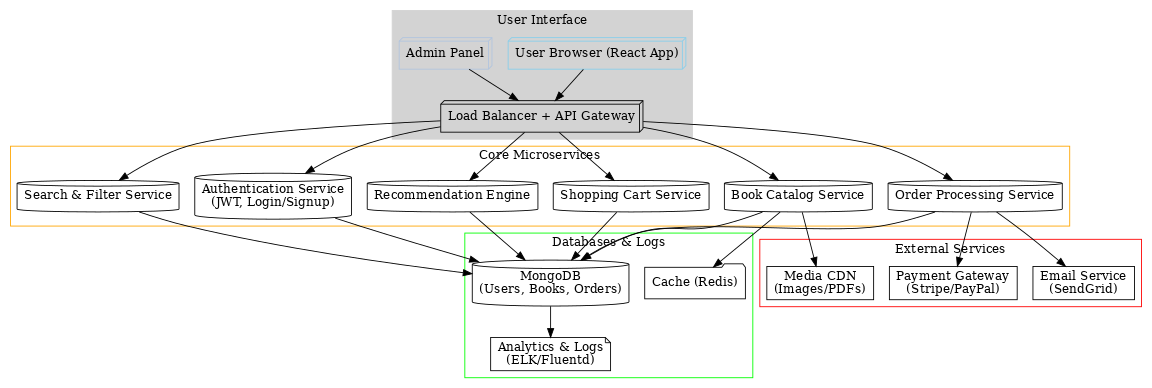
**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

| Date | 25 March 2025 |
| --- | --- |
| Team ID | **SWTID1744137899** |
| Project Name | Book-store Quasar Books |
| Maximum Marks | 4 Marks |

**Technical Architecture:**



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

| **S.No** | **Component** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | User Interface | Web-based UI for browsing books, user/admin dashboards, and cart management | React.js, Tailwind CSS |
|  | Application Logic-1 | User authentication (login/signup) | Node.js, Express.js, JWT, Bcrypt |
|  | Application Logic-2 | Book management (CRUD operations) | Node.js, Express.js, MongoDB |
|  | Application Logic-3 | Order processing and status updates | Node.js, Express.js |
|  | Database | Stores user data, books, and orders | MongoDB (NoSQL) |
|  | Cloud Database | Database hosted on MongoDB Atlas | MongoDB Atlas |
|  | File Storage | Book thumbnails and user avatars | Local Filesystem (or Cloudinary\*) |
|  | External API-1 | Payment gateway (future enhancement) | Stripe/PayPal API |
|  | External API-2 | Email notifications (future enhancement) | SendGrid/Mailchimp API |
|  | Machine Learning | Book recommendations (future enhancement) | Python, Scikit-learn |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology Used** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | Frontend: React, Redux Backend: Express.js, Mongoose | React, Express.js, Mongoose |
|  | Security | JWT authentication, password hashing (Bcrypt), CORS policies | JWT, Bcrypt, Helmet.js |
|  | Scalable Architecture | 3-tier architecture (Frontend ↔ Backend ↔ Database) | React, Node.js, MongoDB |
|  | Availability | Stateless backend, load balancing (future: Kubernetes) | Render, Netlify |
|  | Performance | Client-side rendering (React), async API calls, caching (future: Redis) | Axios, React Lazy Loading |

### 