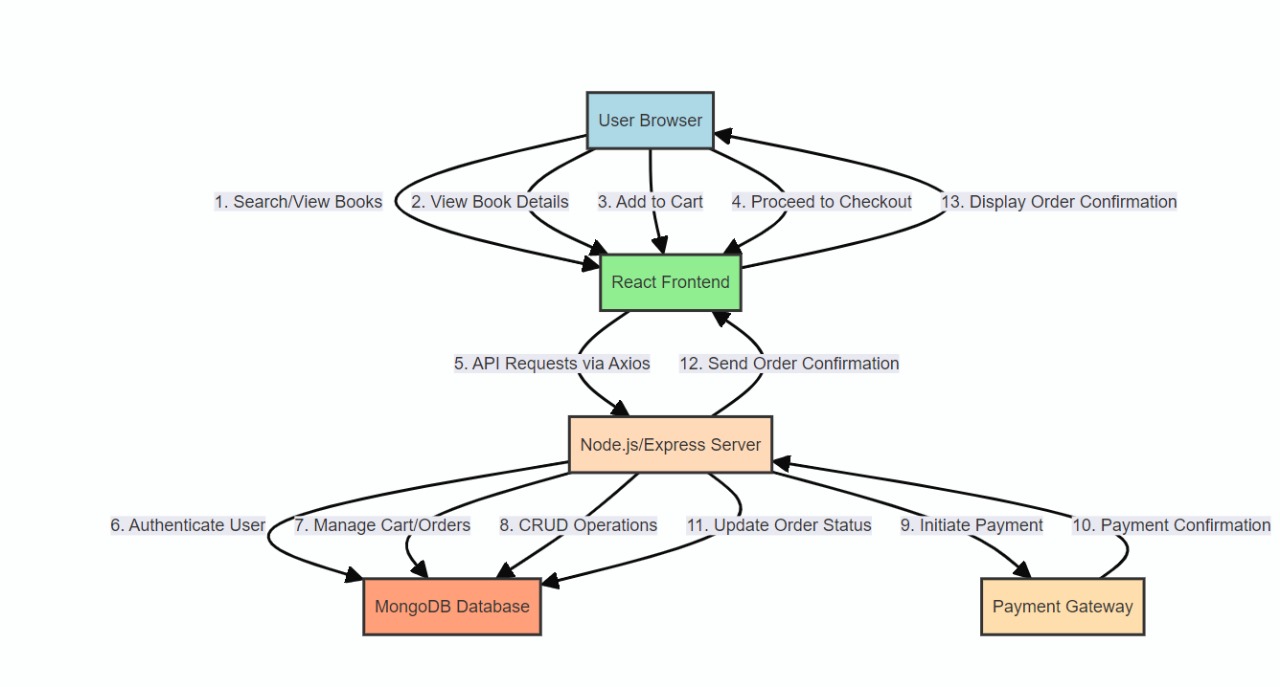
**Requirement Gathering and Analysis Phase**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 06 July 2024 |
| Team ID | SWTID1720110658 |
| Project Name | Book Store |
| Maximum Marks | - |

**Technical Architecture:**

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



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

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
|  | User Interface | How user interacts with application e.g.  Web UI. | HTML, CSS, JavaScript, React.js, Tailwind CSS |
|  | Application Logic-1 | User Authentication | Node.js, Express.js, Passport.js |
|  | Application Logic-2 | |  |  | | --- | --- | |  | Book Data Operations | | Node.js, Express.js |
|  | Application Logic-3 | Search & Filter Functionality | Elasticsearch |
|  | Database | Storing user and book data | MongoDB, Mongoose |
|  | Cloud Database | Database service on the cloud | MongoDB Atlas |
|  | File Storage | |  |  | | --- | --- | |  | Storing book covers and media | | AWS S3 |
|  | External API-1 | Fetching book details | Google Books API |
|  | External API-2 | Fetching additional book data | Open Library API |
|  | Infrastructure | Application deployment | Local: Docker, Cloud: Heroku, Vercel |
|  | CI/CD | Continuous Integration and Deployment | GitHub Actions |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | Using free and open-source frameworks to build the application | React.js, Node.js, Express.js, Mongoose |
|  | Security Implementations | Measures to protect the application from security threats such as user authentication | Passport.js |
|  | Scalable Architecture | Designed to handle increasing user loads and data volume efficiently | Horizontal Scaling: Node.js and Express.js  Database Scalability: MongoDB  Cloud Scalability: Heroku |
|  | Availability | Ensuring the application is available and operational | Heroku, MongoDB |
|  | Performance | Ensuring the application runs efficiently and quickly by using fast NoSQL database and a efficient search functionality | Node.js, MongoDB, Elasticsearch |