**Book Sharing APP**

**User Roles**

* Readers
* Owners
* Viewers
* Administration

**Use cases**

* Book listing ( Dashboard)
* Search Directory
* Book details ( Author, past-ratings, Availability etc.)
* Request Management

**Elaborate description on User Roles**

1. **Readers**- There is a possibility that many employees are book enthusiasts looking for options to share and read. This platform helps them discover available books within their vicinity and also encourages community culture and bond.
2. **Owners**- Another set of users for this application are the owners, willing to share their books for reading. They can list their books on the platform, making them available for others. This encourages collaborative resource sharing and one-on-one interactions.
3. **Viewers** Some users may only want to browse the book list/dashboard to see what kinds of books are available or to check reviews and ratings of certain books. This interaction promotes comprehensive information exchange and exploration of various genres.
4. **Administration-** This user segment is the inter-link between the owners and buyers which is responsible for showcasing books availability, Reviews, send requests for borrowing etc.

**Use Cases**

* **Book Listing/ Dashboard:-**
  + This enables the users to see availability of books and their details
* **Search Directory**
  + This allows users to find the desired books easily without going through the entire book list
  + This can also be filtered based on genre, authors, availability, ratings etc.
  + Book details
* **Book Details**
  + This use case provides information about the book, including the author, ratings and also has the owner details. This can be used for a clearer understanding of the books and gives a brief idea about what is the book about
* **Request Management**
  + This is one of the main use cases where the readers will be shooting requests for the available books, if they are available, the owner and the user will be connected shortly and if not , the request will still be sent to the owner , turning it into a queue of who next will be getting this particular book.

**System Architecture**

**Final User Roles :- 2**

1. **Admin**
2. **Employee**

**Feature Distribution & Implementation**

**1. Admin:**

* **Admin Panel**
  + A centralized panel for managing the entire system.
  + Dashboard overview of club activities, user engagement, and book transactions.
  + Secure login with role-based access control.
* **Club Management**
  + Create, edit, and remove clubs.
  + Assign club admins and monitor club activities.
  + View and approve club-related reports.
* **User Management**
  + Add, remove, or modify user roles.
  + Track user activity and engagement within clubs.
  + Suspend or reactivate accounts if necessary.
* **Categories/Languages**
  + Manage book categories and supported languages.
  + Enable dynamic updates to category options available to users.
  + Provide filtering and sorting options for books based on categories.
* **Reports & Analytics**
  + **Book Transactions**
    - Monitor borrowing history, pending requests, and approval rates.
    - Generate detailed reports for club and system-wide transactions.
  + **Book Inventory**
    - Track the total number of books, available vs borrowed books, and book trends.
    - Generate automated alerts for book shortages or inactivity.
  + **User Engagement**
    - Analyze user behavior such as frequent borrowers, most popular books, and club participation.
    - Provide insights on platform growth and member interactions**.**

**2. Club Admin:**

* **Admin Dashboard**
  + Overview of club-specific activities, book listings, and member engagement.
  + Notification system for pending approvals and user requests.
* **Member Boarding**
  + Invite and approve new members.
  + Set membership rules and guidelines for book borrowing.
* **Request to Return**
  + Track pending return requests from members.
  + Approve or decline return requests based on club policies.
* **Listing for Approval**
  + **Approval/Reject**
    - Review book listing submissions by members.
    - Approve or reject listings based on quality and relevance.

**3. Members:**

* **Login with MFA**
  + Multi-factor authentication for secure access.
  + Email/OTP-based verification upon login.
* **Club Selection of Multiple**
  + Users can join and participate in multiple clubs.
  + A club-switching interface for seamless navigation.
* **User Preferences**
  + Customize preferred book genres, notification settings, and reading lists.
  + Store preferences in the user profile and dynamically update them.
* **Dashboard**
  + Personalized dashboard displaying borrowed books, pending requests, and recommendations.
  + Integrated book search and request tracking.
  + Notification system for book approvals, due dates, and return requests.

**Book Sharing System**

**Feature Distribution and Implementation (React & Node.js)**

| **User Role** | **Feature** | **Implementation** | **API Endpoints** |
| --- | --- | --- | --- |
| **Admin Panel** | Club Management | CRUD operations for managing clubs, assigning club admins | POST /admin/create-club |
|  | User Management | Handle user registrations, role assignments, and deactivations | GET /admin/users, PUT /admin/update-role/:id |
|  | Categories/Languages | Define and manage book categories and supported languages | - |
|  | Reports - Book Transactions | Generate reports on borrowed and returned books | GET /admin/reports/transactions |
|  | Reports - Book Inventory | Track available and borrowed books | - |
| **Club Admin Panel** | Member Boarding | Approve or reject member requests | POST /club-admin/approve-member/:id |
|  | Request to Return | Track book return requests | POST /club-admin/request-return/:bookId |
|  | Listing for Approval | Approve or reject books added by members | PUT /club-admin/approve-book/:bookId |
| **Member Features** | Login with MFA | Secure authentication using OTP/email verification | POST /auth/login |
|  | Multiple Club Selection | Join multiple clubs and switch between them | - |
|  | User Preferences | Save reading preferences and language settings | PUT /member/update-preferences |
|  | Dashboard - Book Listings | View available books | GET /member/dashboard |
|  | Dashboard - Request Books | Request books for borrowing | POST /member/request-book/:bookId |
|  | Dashboard - Track Borrowed Books | Track borrowed books & return deadlines | - |

**Tech Stack & Deployment Strategy**

| **Component** | **Technology** | **Deployment** |
| --- | --- | --- |
| **Frontend** | React.js (Material-UI/Ant Design, React Router, Redux/Context API) | Deployed on Vercel/Netlify, CI/CD using GitHub Actions |
| **Backend** | Node.js with Express.js | Deployed on AWS EC2/Heroku, Load balancing with Nginx/P M2 |
| **Database** | MongoDB Atlas/PostgreSQL | Managed cloud database |
| **Authentication** | Firebase/Auth0 (MFA, JWT-based) | Secure authentication & authorization |
| **Logging & Monitoring** | Winston, Morgan | Error tracking and logging |
| **Real-time Updates** | WebSocket | Live notifications and book status updates |