Invent Analytics Software Solutions Engineer Case

A library management application will be developed to manage members and the borrowing of books by members. The system is intended to be used by an authorized staff member working in the book lending/return department of a library. There will be no authentication or authorization mechanisms implemented. The operations that can be performed within the application are listed below:

- Existing users should be listed in a grid. This grid will display all registered users.
- When a user is selected, the user's detail page should be accessible. This page will display:
    - Books currently borrowed by the user
    - Books previously borrowed by the user along with the ratings they provided
- On the user's detail page, there should be an option to return a book currently in the user's possession.
- Existing books should be listed in a grid. This grid will display all available books.
- When a book is selected, the book's detail page should be accessible. This operation should be considered as a process much more frequent than others. This page will display:
    - Information related to the book (author, year etc)
    - Current owner, if there is
    - The average rating of the book
- From the book's detail page, there should be an option to lend the book to a user.

Technical requirements are listed below. The solution should strive to meet these requirements as much as possible:

- The frontend should be developed using React.js.
- TypeScript or ES5+ can be used.
- The REST API should be developed using Node.js with Express.js.
- Any relational database can be used, and a DDL script to create the database should be included with the solution.
- An ORM or query builder should be used for database operations (e.g., Knex, Sequelize, TypeORM, Prisma, etc.).

- Build tools such as Webpack, Esbuild, or Babel can be utilized.
- Errors from the service should be handled appropriately on the frontend to inform the user in a meaningful way.
- The developed API must be compatible with the provided Postman collection.
- A version control tool must be used (e.g., Git, SVN, etc.).
- Utility libraries such as Lodash or Underscore can be used if necessary.
- Instructions to run the application must be included in a README file.
- Using SASS/LESS/SCSS will be considered a bonus.
- Using Redux will be considered a bonus.
- UI libraries such as Bootstrap, Bulma, Semantic UI, or Material UI can be used.