invent.ai

# Invent.ai Backend Case Study

A library management application will be developed to manage members and the borrowing of books by members. The operations that can be performed within the application are listed below:

- Listing users
- Accessing information about a user (name, books borrowed in the past with their user scores, and currently borrowed books)
- Creating a new user
- Listing books
- Accessing information about a book (name and average rating). Book viewing should be considered as a process much more frequent than borrowing and returning.
- Creating a new book
- Borrowing a book
- Returning a book and giving a rating

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

- Any code versioning tool should be used (git, svn etc.)
- Develop a REST API in a JavaScript environment with Node.js, utilizing the Express.js library.
- TypeScript or ES5+ with the option to employ libraries such as Webpack, Babel, etc. canbe used
- Ensure the application receives requests and returns responses compatible with the attached Postman Collection, with request/response examples provided within.
- Employ any relational database management system for the database, providing the DDL script along with the solution.
- Preferably use an ORM or query builder library for database operations (e.g., knex, sequelize, bookshelf, typeorm, etc.)
- Validate API request bodies using a validator (e.g., joi, express-validator, validator.js, etc.)
- Handle errors effectively, such as attempting to borrow a book by a non-existing user or borrowing a book already borrowed by someone else, indicating errors in the API response (at least 500 Internal Server Error).
- Optionally utilize utility libraries like Lodash, Underscore.js, moment, etc., if necessary