Assignment ContentTop of Form

**Assignment 1: Implementing a very simple Library Management System**

**Overview**

In this assignment, you will develop a Library Management System using Node.js and Express for the backend, and Bootstrap for the frontend UI. The system will allow users to perform CRUD operations on books and manage borrow and return transactions. This project will help you understand full-stack development, RESTful API design, and CRUD operations in a web application.

**Objectives**

Build RESTful APIs using Node.js and Express to manage books and borrow operations.

Implement a frontend using HTML, CSS, and Bootstrap to interact with the backend.

Apply CRUD operations on books: Create, Read, Update, Delete.

Implement functionality to borrow and return books.

Use MongoDB as the database for storing book and transaction data.

**Requirements - Backend (Node.js & Express)**

1. **Setup and Initialization**:

* Initialize a new Node.js project.
* Install necessary NPM packages: **express**, **mongoose**, **body-parser**, and **cors**.

1. **Database Model**:

* Use **mongoose** to define models for Books and Transactions (Borrow/Return).
* The Book model should include fields such as title, author, and ISBN.
* The Transaction model should include fields such as bookID, userID, borrowDate, and returnDate.

1. **API Endpoints (you can add more if you like)**:

* Implement CRUD operations for books:
* GET **/api/books** to retrieve all books.
* POST **/api/books** to add a new book.
* GET **/api/books/:id** to retrieve a book by ID.
* PUT **/api/books/:id** to update a book by ID.
* DELETE **/api/books/:id** to delete a book by ID.
* Implement endpoints for borrowing and returning books:
* POST **/api/borrow** to borrow a book.
* POST **/api/return** to return a book.

1. **Middleware & Error Handling**:

* Use **body-parser** to parse request bodies.
* Implement error handling middleware for managing API errors gracefully.

**Requirements - Frontend (Bootstrap, HTML, CSS)**

1. **Bootstrap Setup**:

Integrate Bootstrap into your project to design the UI.

1. **UI Design**:

* Create a main page to list all books with options to add, edit, or delete books.
* Design a form to add or edit book details.
* Implement a borrow interface, allowing users to borrow and return books.

1. **Interaction with Backend**:

* Use JavaScript to fetch data from the backend and display it on the frontend.
* Implement form submissions to create or update books through the API.
* Handle borrow and return operations via the frontend, interacting with the backend.

1. **Deliverables**

* Source code for the backend and frontend.
* A README file documenting:
* Project setup instructions.
* API endpoints and their usage.
* A brief explanation of the frontend design.

**Evaluation Criteria**

* Functionality: All features work as expected.
* Code Quality: Clean, readable, and well-structured, and commented code.
* Error Handling: Proper error responses from the backend and user-friendly error messages on the frontend.
* Documentation: Clear and concise README documentation.

**Submission**

Submit your project as a zip file or a link to a GitHub repository. Ensure that your submission includes all necessary files to run the application and a detailed README. No need to include nod\_modules directories. This assignment requires a combination of backend and frontend skills. It's recommended to start with the backend API development, followed by the frontend UI. Test each part thoroughly before integrating them. Good luck!

Bottom of Form