Full Stack Assignment - 1

Assignment: Build a Complete Web Application

Problem Statement:

Create a **Library Management System** where users (students and librarians) can:

1. Register and log in securely using hashed passwords.

2. Access features based on roles (e.g., librarian and student).
3. Manage library books with CRUD operations.
4. Maintain borrowing history and enforce borrowing limits.

Requirements:

Backend Tasks:

1. **User Authentication & Authorization**:
    - Use JWT for authentication.
    - Allow login and registration.
    - Restrict access based on roles:
        - **Student Role**: Can view and borrow books.
        - **Librarian Role**: Can add, update, or delete books.
2. **Database Management (MongoDB)**:
    - **User Schema**: Include fields for name, email, password, and role.
    - **Book Schema**: Include fields for title, author, ISBN, available copies, and borrowed count.
    - **Borrow Schema**: Store borrowing details (user, book, borrow date, and return date).
3. **Middleware**:
    - Protect routes with authentication middleware.
    - Create role-based authorization middleware to ensure specific features are accessible only to respective roles.
4. **Password Hashing**:
    - Use bcrypt for password storage and verification.
5. **Error Handling**:
    - Return proper status codes and error messages for unauthorized access, invalid inputs, and other failures.

Frontend Tasks:

1. **React Components**:

- Create a **Login Page** and **Register Page**.
- Build a **Dashboard** for both students and librarians with respective functionalities.
- Design a **Books Page** with search and filter options.

2. **API Integration**:
   - Use `axios` or `fetch` to interact with the backend.
   - Display error messages and success notifications dynamically.
3. **Role-Specific UI**:
   - Display features like "Add Book" or "Borrow Book" based on user roles.
4. **User-Friendly Design**:
   - Implement a responsive UI for a seamless experience.

Additional Real-World Enhancements:

1. **Borrowing Rules**:
   - Restrict students to borrow only 3 books at a time.
   - Automatically calculate overdue charges if the return date exceeds 15 days.
2. **Admin Role (Optional)**:
   - Allow admins to manage both students and librarians, approve librarian registrations, and reset book borrowing limits.
3. **Search and Filters**:
   - Add filters like genre, author, and publication year to improve the book search experience.
4. **Deployment**:
   - Deploy the backend on a cloud service like Heroku.
   - Use services like Netlify or Vercel for the React app.