**Exam Instruction: Web System Development**

**GENERAL INSTRUCTIONS**

1. **Development Platform**:
   * You will use PHP, JavaScript, CSS (optional), and any relevant libraries.
   * Use API-based communication between the front-end and back-end of your system.
2. **Deadline**:
   * The system must be completed and submitted by December 6.
3. **Pitch Presentation**:
   * You will present your project on December 7.
   * Prepare a brief pitch (8 minutes at max) explaining your system’s functionality, user interface, and any special features you’ve implemented.
4. **Key Features to Implement**:
   * User management page for adding/editing Librarians.
   * Book management page for adding/editing books and viewing their status.
   * Report generation page where Admins can filter books by status and export the report as a PDF.
   * Borrow/Return system for Librarians to update the status of books.

**MANAGEMENT**

**User Management**

* The system supports two roles: **Admin** and **Librarian**.
* Only the **Admin** can manage users.
  + **Add User:** Create new Admin or Librarian accounts.
  + **Edit User:** Update the username, password, or role of existing users.
  + **Delete User:** Remove a user from the system.

**Book Management**

* The admin manages books:
  + **Add Book:** Add new books to the library database, including details like title, author, and initial status ("Available").
  + **Edit Book:** Update book details as needed.
  + **Delete Book:** Remove books from the library database.
* **Librarian** interacts with books by updating their status:
  + **Borrowing:** Change the book's status to "Borrowed."
  + **Returning:** Change the book's status back to "Available."

**Business Rules**

1. **Book Availability:**
   * A book must be "Available" to be borrowed.
   * A book can only be marked "Available" again after being returned.
2. **Role Permissions:**
   * **Admin:** Full access to User Management, Book Management, and Reporting.
   * **Librarian:** Limited to updating book statuses for borrowing and returning.
3. **Data Validation:**
   * Usernames must be unique.
   * Book statuses should update only if the status allows the operation (e.g., cannot borrow a book already "Borrowed").
4. **System Integrity:**
   * Deleting a book should ensure it is not currently borrowed.
   * User deletion should only be allowed for non-critical accounts (e.g., an Admin cannot delete their account).

**Description of Key Files**

**/assets/js**

**/user.js**: JavaScript for handling user management tasks like adding, editing, or deleting users via API calls.

**/book.js**: JavaScript for handling book operations like adding books, changing status, etc.

**/main.js**: General utility scripts for the entire application, such as validation and modal handling.

**/report.js**: Manages the report generation interface, fetching data via API calls and optionally exporting reports to PDF.

**/api**

**/user.php**: Handles CRUD operations for users *(Admin-only functionality)*.

**/book.php**: Handles CRUD operations and status updates for books.

**/auth.php**: Manages login and authentication requests.

**/report.php**: Fetches data for reports based on user-selected filters (e.g., book status). Returns data in JSON format for dynamic rendering.

**/config/database.php**

* Contains the database connection code (e.g., PDO or MySQLi).

**/includes**

**/header.php**: Common header with navigation and user session checks.

**/footer.php**: Common footer for consistent UI.

**/auth.php**: Functions to manage role-based access (e.g., check if the user is Admin or Librarian).

**/views/admin**

**/dashboard.php**: Displays reports, quick actions, and stats for the Admin.

**/manage\_users.php**: Form and table for managing users.

**/manage\_books.php**: Interface for managing book details.

/**reports.php**: Generates pdf.

**/views/librarian**

**/borrow\_return.php**: Interface for Librarians to manage book borrowing and returning.

**index.php**

* Acts as the application’s entry point, redirecting users to the login page or their respective dashboard based on their role.

**Application directory guide:**

A screenshot of a computer program

Description automatically generated