

SCHOOL EQUIPMENT LENDING SYSTEM

AUTHOR: SARUMATHY G.

ROLL NO: 2024TM93153

1. INTRODUCTION

The SCHOOL EQUIPMENT LENDING PORTAL is a full-stack web application designed to simplify how schools manage the borrowing and lending of educational resources such as lab equipment, projectors, sports gear, and other assets.

The portal provides dedicated interfaces for students, staff, and administrators to perform their respective tasks—students can request or return equipment, staff can approve or monitor requests, and administrators can manage inventory and users efficiently.

The project adopts a **modular, scalable architecture**, built using:

- **Frontend:** React (with Neumorphism UI)
- **Backend:** Node.js + Express.js
- **Database:** MongoDB Atlas (cloud-based)
- **API Testing & Docs:** Swagger and Postman
- **Deployment Support:** Docker (optional)

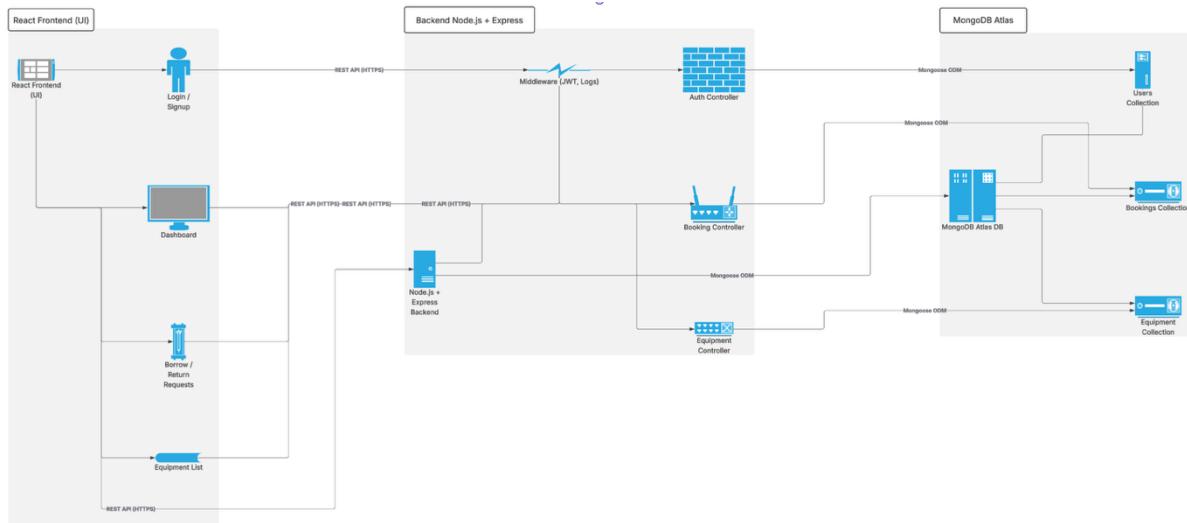
2. SYSTEM ARCHITECTURE

OVERVIEW

The architecture follows the **client-server model**, with clear separation between presentation, logic, and data layers.

- **Frontend (React):** Handles all UI/UX rendering, API calls, and state management.
- **Backend (Express + Node.js):** Manages authentication, routing, validation, and data transactions.
- **Database (MongoDB):** Stores users, equipment details, and booking records.
- **Middleware:** Handles authentication (JWT), logging, and error responses.

ARCHITECTURE DIAGRAM



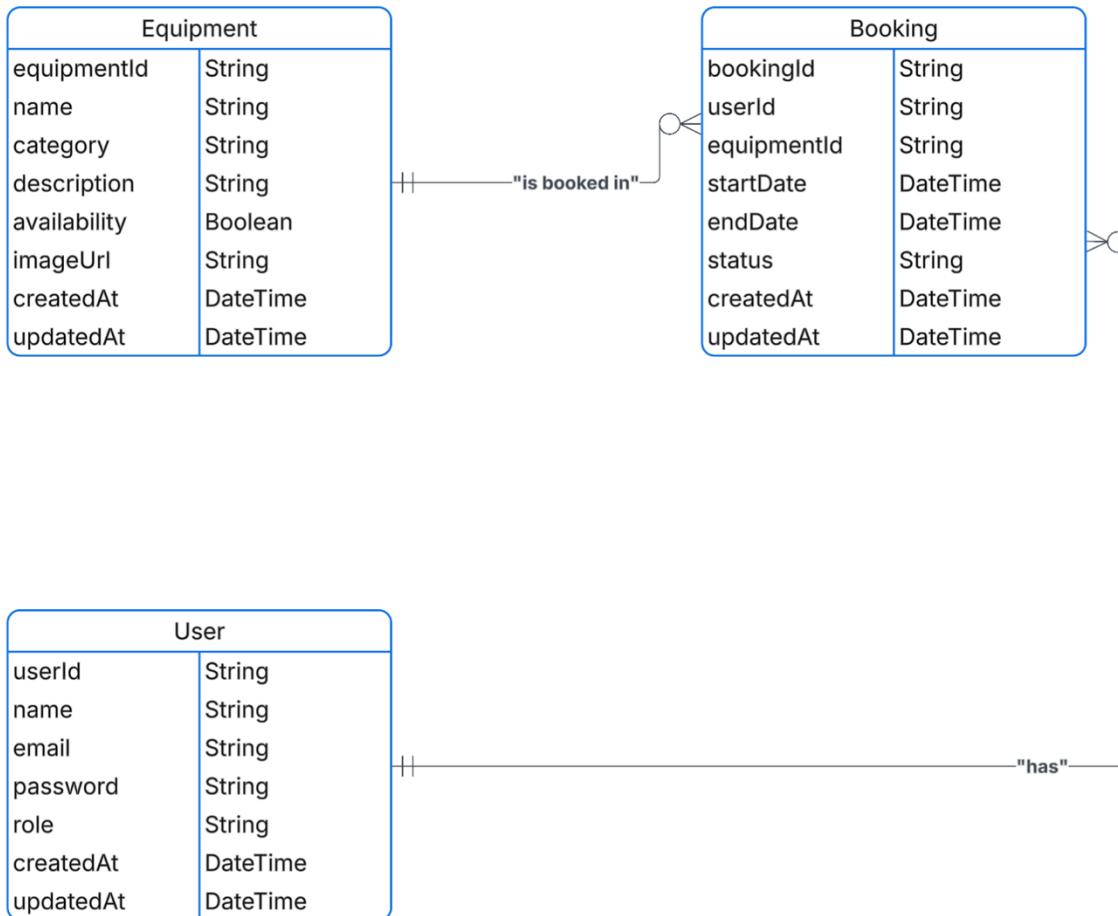
1. The **frontend client (React)** communicates with the backend REST API.
2. The **backend (Express server)** processes incoming HTTP requests and performs validation.
3. The **database layer (MongoDB)** handles CRUD operations for users, items, and bookings.
4. The **middleware layer** (auth, logging, error-handling) ensures security and traceability.
5. **JWT tokens** are used to protect endpoints and manage user sessions.

3. DATABASE SCHEMA & MODEL DESIGN

ENTITY MODELS

- **User** - name, email, password, role (student/staff/admin)
- **Item** - name, category, status, quantity, description, images[]
- **Booking** - userId, itemId, startDate, endDate, status (pending / approved / returned)

ER MODEL DIAGRAM



Relationships:

- One **User** → Many **Bookings**
- One **Item** → Many **Bookings**
- Admin manages both **Users** and **Items**.

DATABASE RULES

- Unique index on email for user collection.
- Prevent overlapping bookings using date validation logic on the backend.
- Timestamps auto-generated for creation and update tracking.

4. COMPONENT HIERARCHY

FRONTEND COMPONENTS

- **App.js:** Entry point controlling route access and global context.
- **Navbar:** Displays navigation links, profile dropdown, and logout confirmation modal.
- **LandingPage:** Public homepage showing school details, features, and call-to-action buttons.
- **Login & Signup Forms:** Manage authentication with Toastr notifications.
- **Admin Dashboard:** For managing equipment inventory and user activities.
- **Booking Section:** Lists borrowed items and booking history for users.
- **ConfirmModal / Dialog:** Used for actions like delete or logout confirmation.

BACKEND COMPONENTS

- **server.js:** Main entry point, connects to MongoDB, configures routes and middlewares.
- **Routes:**
 - /auth – User registration, login, token verification.
 - /items – CRUD for equipment (admin-only).
 - /bookings – Handle borrowing, returning, and conflict checks.
- **Middleware:**
 - authMiddleware – JWT authentication.
 - errorHandler – Unified API error responses.
- **Models:** Mongoose schema definitions for User, Item, and Booking.

5. UI/UX CONCEPTUAL GUIDE

DESIGN PHILOSOPHY

The interface is developed using the **Neumorphism design style**, which blends modern flat design with subtle depth effects. It emphasizes a soft, tactile look to make interaction feel natural and calm—ideal for educational environments.

UI FLOW

1. Landing Page:

- Describes the purpose of the portal and has buttons for LOGIN or SIGN UP.
- Includes attractive images of school equipment and modern classroom activities.

2. Authentication:

- Toastr notifications appear on successful signup/login.
- Errors like “invalid credentials” or “existing email” are displayed clearly.

3. Dashboard (After Login):

- Personalized view depending on role:
- **Student:** Browse and borrow items
- **Staff:** Approve or return equipment
- **Admin:** Add/edit/delete items, manage users

4. Modals and Interactions:

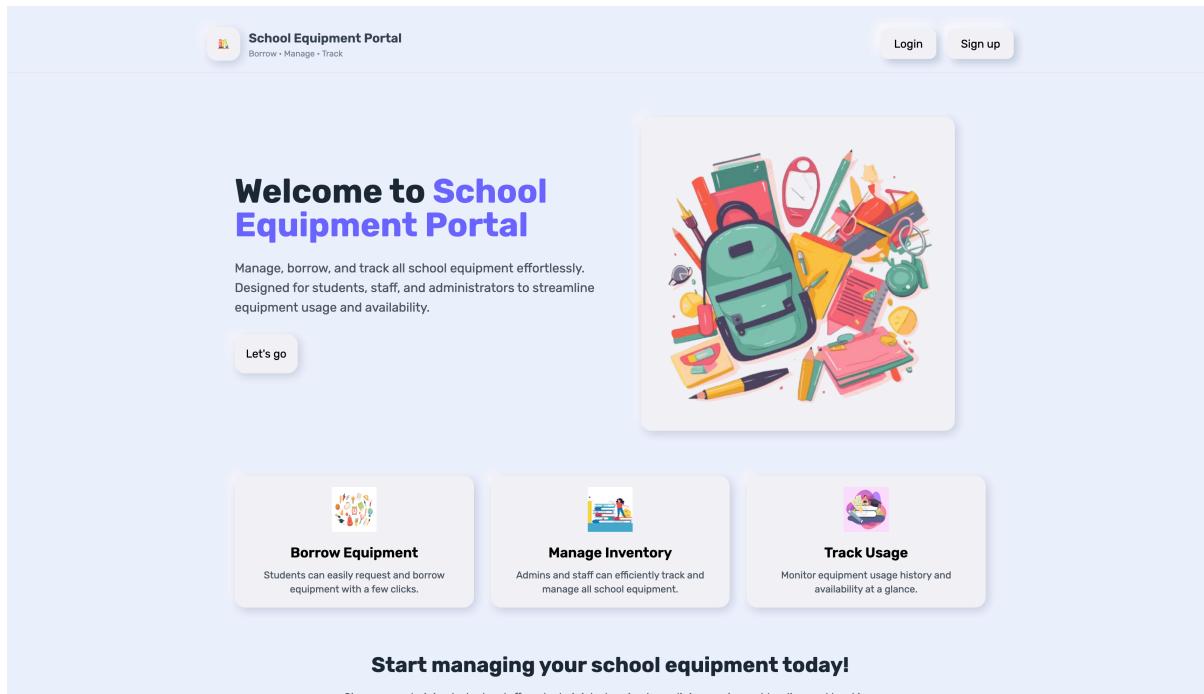
- Logout confirmation modal to prevent accidental sign-outs
- Delete confirmation modal before removing items
- Toastr success/failure alerts for all critical actions

5. Responsiveness:

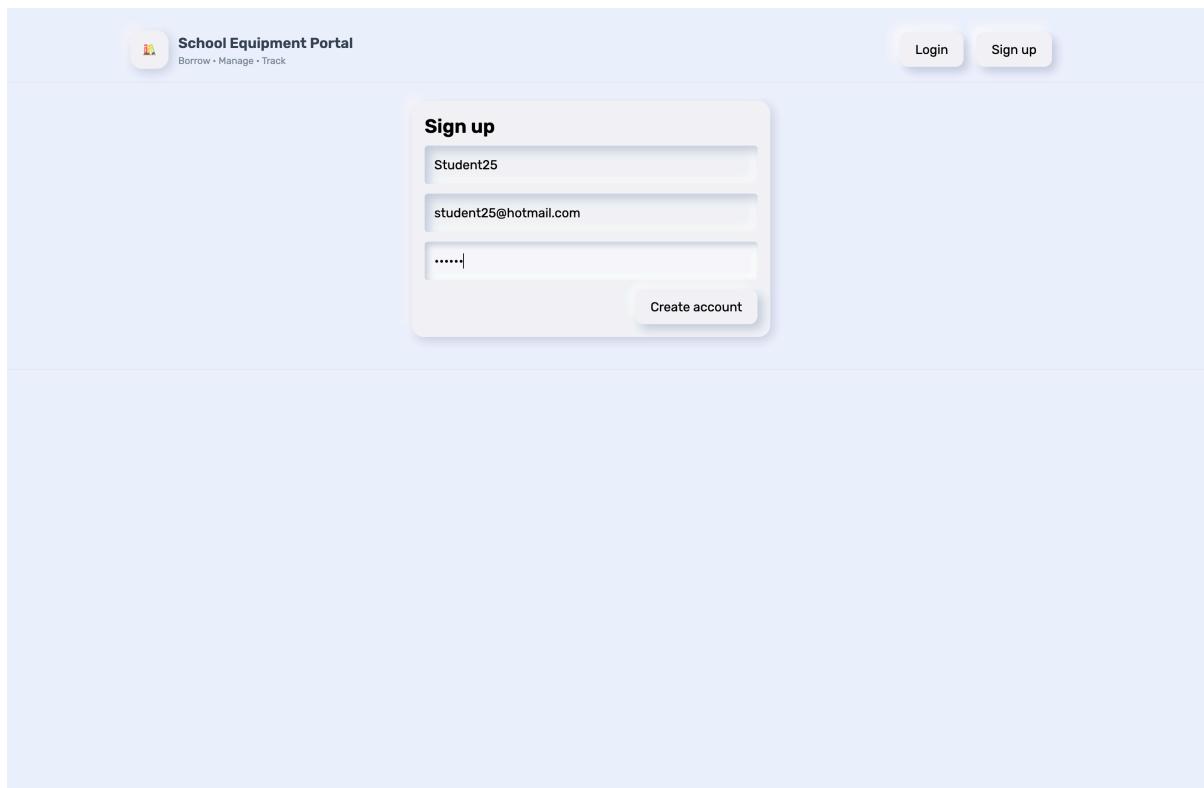
- Fully responsive layout optimized for desktops, tablets, and mobile devices.

UI SCREENSHOTS

1. Landing Screen



2. Sign-up Screen



3. List of Items

School Equipment Portal
Borrow • Manage • Track

Home Student25 (student)

School Equipment Catalog

Browse equipment. Use live search to quickly find what you need.

Search equipment (name, description)...

All categories Any availability Clear

Item	Category	Condition	Availability	Action
Shuttle Rackets	Sports	good	Available (25)	Request
Geometry Items	Maths	good	Available (1)	Request
Drafter	EG	good	Available (10)	Request
DevOps	Eng	good	Available (178)	Request
Software Architecture	SA	good	Available (1)	Request
Secure Software Engineering	SSE	good	Available (1)	Request

4. Search & Filter

School Equipment Portal
Borrow • Manage • Track

Home Student25 (student)

School Equipment Catalog

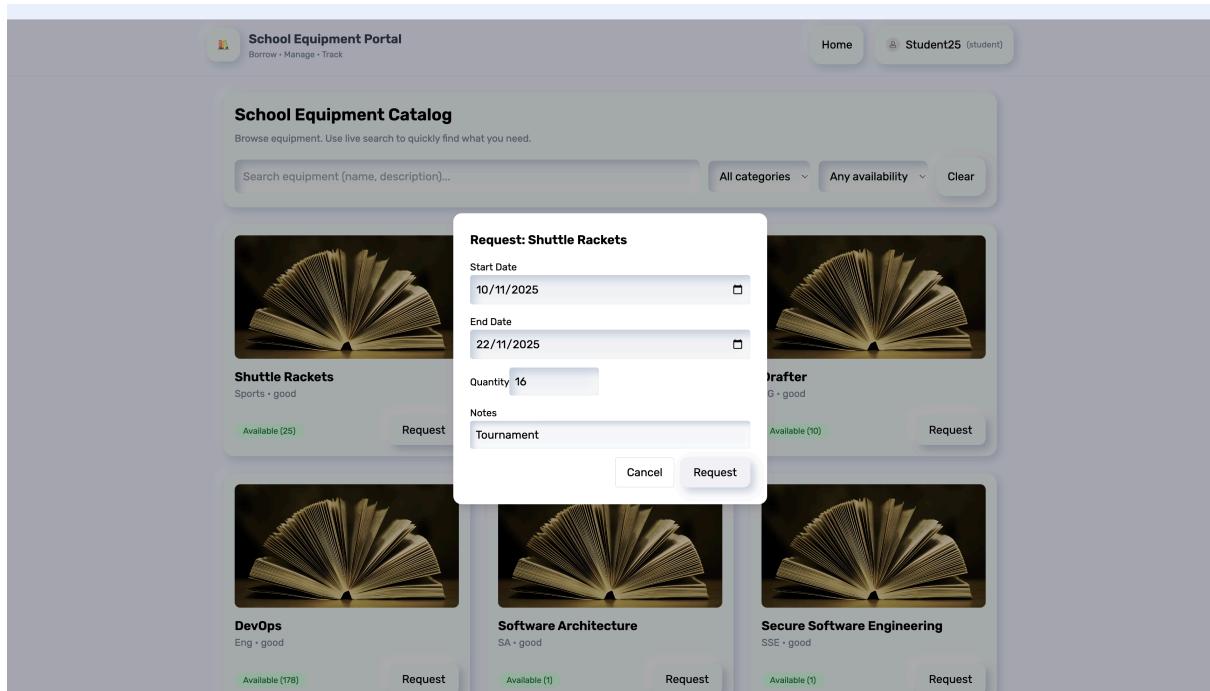
Browse equipment. Use live search to quickly find what you need.

Shut

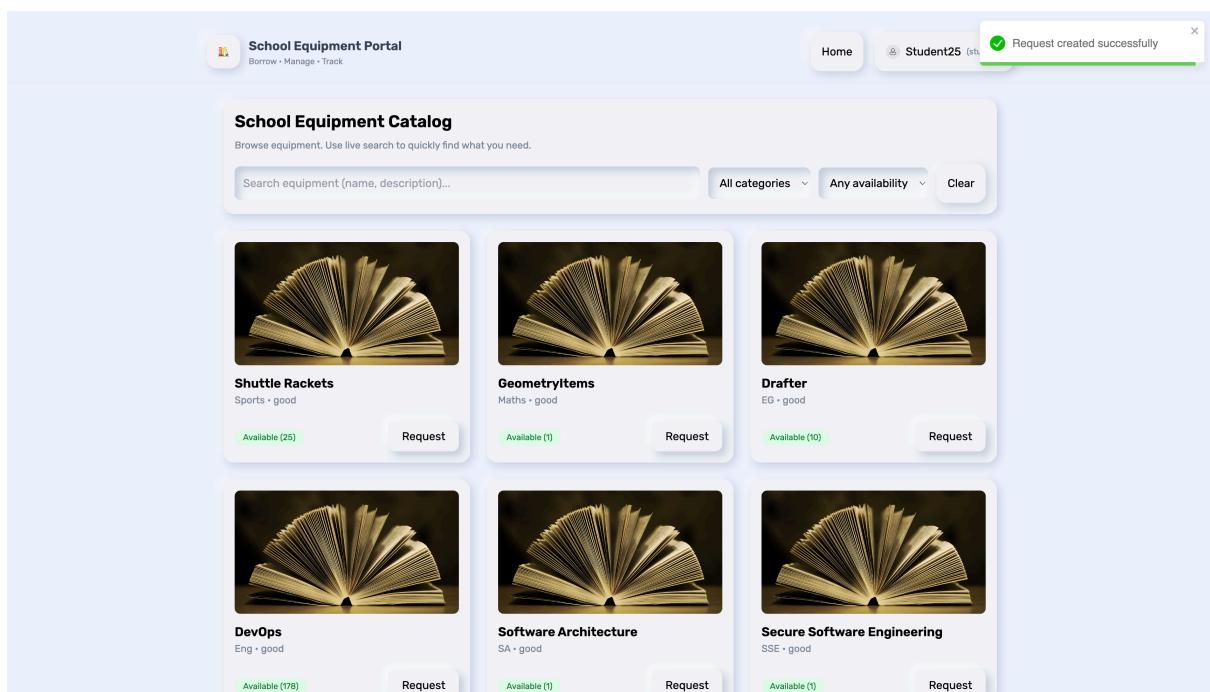
All categories Any availability Clear

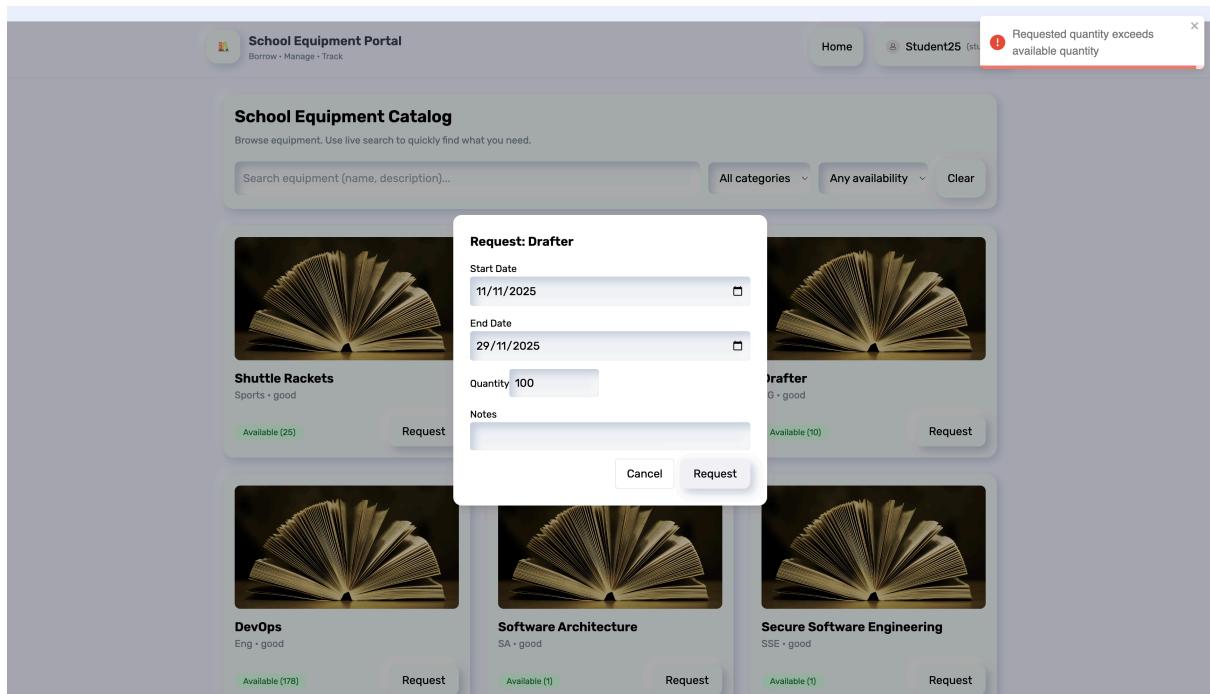
Item	Category	Condition	Availability	Action
Shuttle Rackets	Sports	good	Available (25)	Request

5. Request of Item

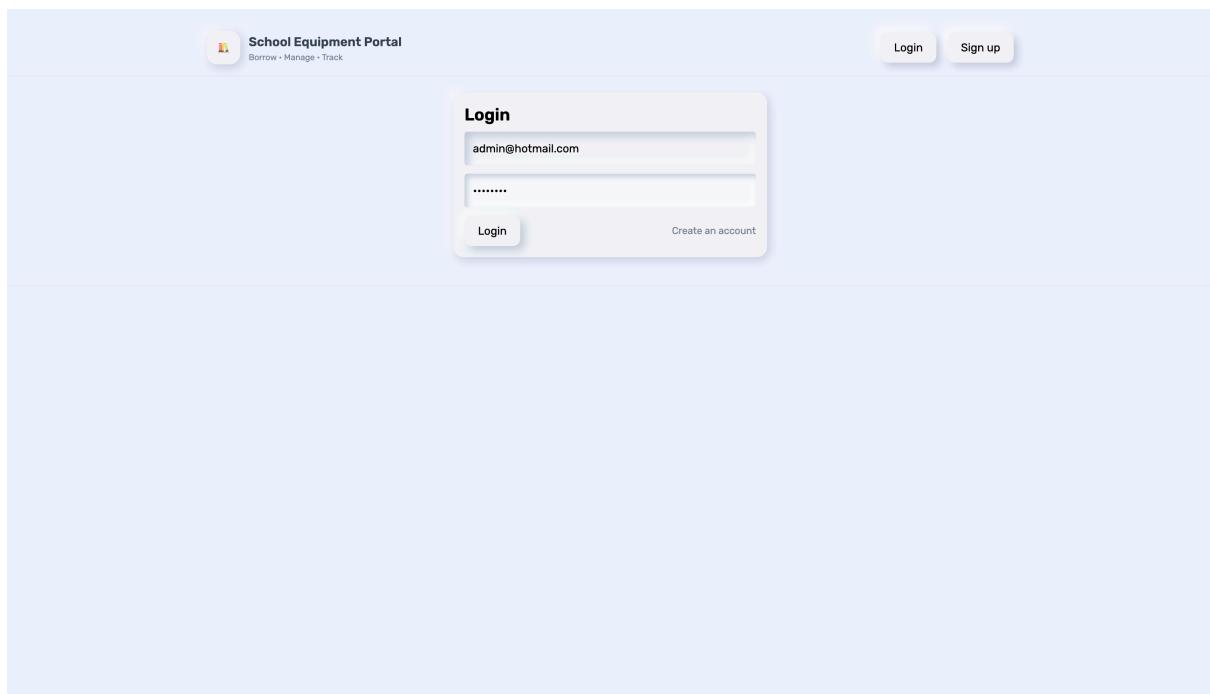


6. Toastr of requesting an item





7. Login Screen (Admin)



8. Manage Screen (Admin)

The screenshot shows the 'Manage Bookings' section of the portal. It lists several equipment bookings:

- Microscope**: Student25 - 11/11/2025 → 21/11/2025. Qty: 5. Status: **approved**. Action: **Mark Returned**.
- Microscope**: Admin - 18/11/2025 → 15/11/2025. Qty: 1. Status: **rejected**.
- Canon DSLR Camera**: Admin - 21/11/2025 → 29/11/2025. Qty: 1. Status: **approved**. Action: **Mark Returned**.
- DevOps**: Admin - 06/11/2025 → 14/11/2025. Qty: 1. Status: **rejected**.
- DevOps**: Admin - 07/11/2025 → 21/11/2025. Qty: 1. Status: **requested**. Actions: **Approve** (green) and **Reject** (red).
- Drafter**: Admin - 14/11/2025 → 22/11/2025. Qty: 1. Status: **approved**. Action: **Mark Returned**.
- DevOps**: Admin - 27/11/2025 → 29/11/2025. Qty: 20. Status: **requested**. Actions: **Approve** (green) and **Reject** (red).

9. Adding of Items (Admin)

The screenshot shows the 'Admin - Items' section. A new item is being added:

Shuttle Rackets
Sports
25
Description

Buttons: **Add item**, **Cancel**, **Save**.

Below the form, there are six items listed in a grid:

Image	Name	Description	Action
	GeometryItems Maths · good		Edit Delete
	Drafter EG · good		Edit Delete
	DevOps Eng · good		Edit Delete

10. Toastr Notification

The screenshot shows the 'Admin – Items' section of the School Equipment Portal. At the top right, there is a toastr notification with a green checkmark icon and the text 'Item Shuttle Rackets added'. Below the notification, there are nine items listed in a grid of three rows and three columns. Each item has a small image of books, a name, a category, and status, followed by 'Edit' and 'Delete' buttons.

Item Name	Category	Status
Shuttle Rackets	Sports	good
GeometryItems	Maths	good
Drafter	EG	good
DevOps	Eng	good
Software Architecture	SA	good
Secure Software Engineering	SSE	good

11. Confirmation dialog for deletion of an item

The screenshot shows the same 'Admin – Items' section as the previous image, but with a modal dialog box overlaid. The dialog box has a title 'Delete item' and the question 'Are you sure you want to delete this item?'. It contains two buttons: 'Cancel' and 'Delete'. The rest of the page is dimmed, and the items are visible through the semi-transparent overlay.

12. Swagger Docs

The screenshot shows two identical instances of the Swagger UI for the "School Equipment Lending API". Both instances are version 1.0.0 and follow the OpenAPI Specification 3.0. The top instance is for the "School Equipment Portal" and the bottom instance is for a general endpoint.

Servers: http://localhost:5000

Auth User authentication

- POST /api/auth/signup** Register a new user
- POST /api/auth/login** Login and get JWT token

Bookings Equipment booking and returns

- POST /api/bookings** Create a new booking
- GET /api/bookings** Get all bookings (own bookings for students, all for admin/staff)
- PUT /api/bookings/{id}/approve** Approve a booking (Admin/Staff only)
- PUT /api/bookings/{id}/reject** Reject a booking (Admin/Staff only)
- PUT /api/bookings/{id}/return** Mark booking as returned

Items Equipment management

- GET /api/items** Get all equipment items

Auth User authentication

- POST /api/auth/signup** Register a new user
- POST /api/auth/login** Login and get JWT token

Bookings Equipment booking and returns

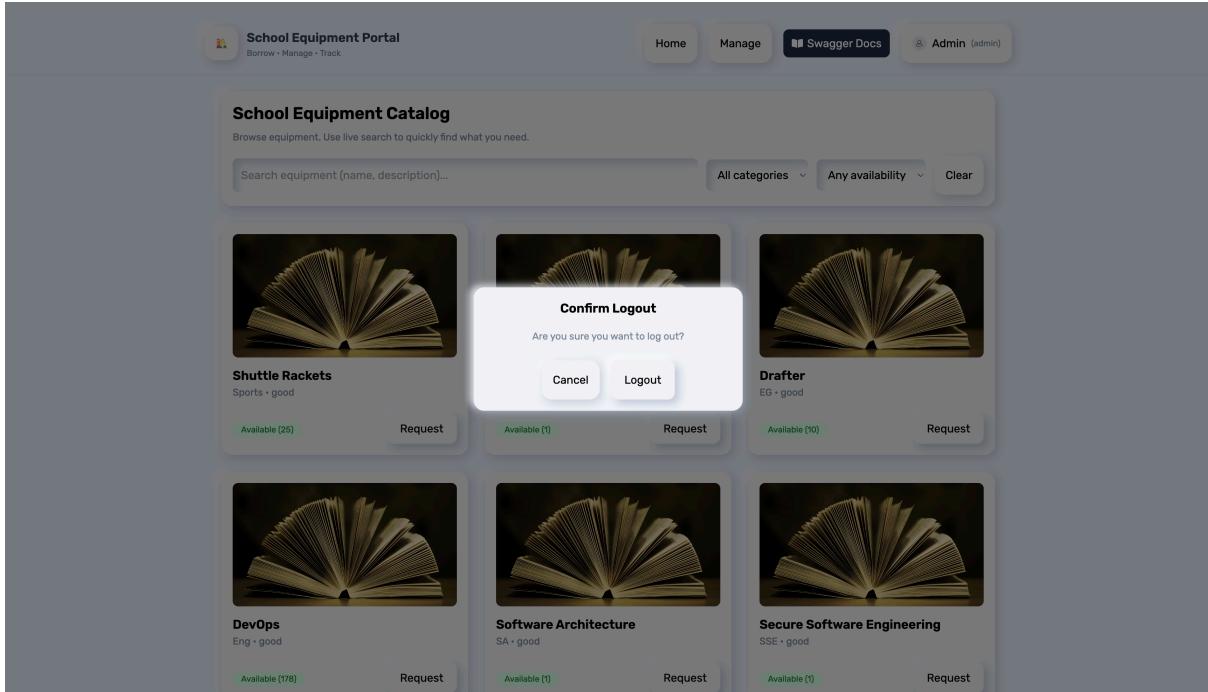
- POST /api/bookings** Create a new booking
- GET /api/bookings** Get all bookings (own bookings for students, all for admin/staff)
- PUT /api/bookings/{id}/approve** Approve a booking (Admin/Staff only)
- PUT /api/bookings/{id}/reject** Reject a booking (Admin/Staff only)
- PUT /api/bookings/{id}/return** Mark booking as returned

Items Equipment management

- GET /api/items** Get all equipment items
- POST /api/items** Add a new equipment item
- GET /api/items/{id}** Get item by ID
- PUT /api/items/{id}** Update an existing equipment item
- DELETE /api/items/{id}** Delete an equipment item

Authorize (green button)

13. Logout Confirmation



6. ASSUMPTIONS

- Users are registered by self-signup or by the admin.
- Each item has a unique identifier and optional image file.
- Booking requests are validated for date conflicts (no overlapping).
- The system supports multiple concurrent sessions with JWT-based authentication.
- Admin approval is mandatory for new item creation or modifications.

7. CONCLUSION

The SCHOOL EQUIPMENT LENDING PORTAL demonstrates a well-balanced integration of user experience, security, and functional scalability. By combining a **React frontend** and **Node.js backend** with a **MongoDB Atlas** cloud database, it ensures both flexibility and robustness.

The **Neumorphism-based interface** adds visual appeal and promotes intuitive navigation.

This solution provides a real-world, deployable system that can be expanded for institution-level asset management, helping schools transition from manual tracking to an intelligent, digital ecosystem.