

SCHOOL EQUIPMENT LENDING SYSTEM

**SE ZG503 – FULL STACK APPLICATION
DEVELOPMENT**

AUTHOR: SARUMATHY G.

ROLL NO: 2024TM93153

GITHUB REPO: <https://github.com/mathysaru/SchoolLendingPortal>

DEMO VIDEO:

<https://drive.google.com/drive/folders/1BxeKmJNucYbVEoRIGYY-Yel-MTNcoy3K?usp=sharing>

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)

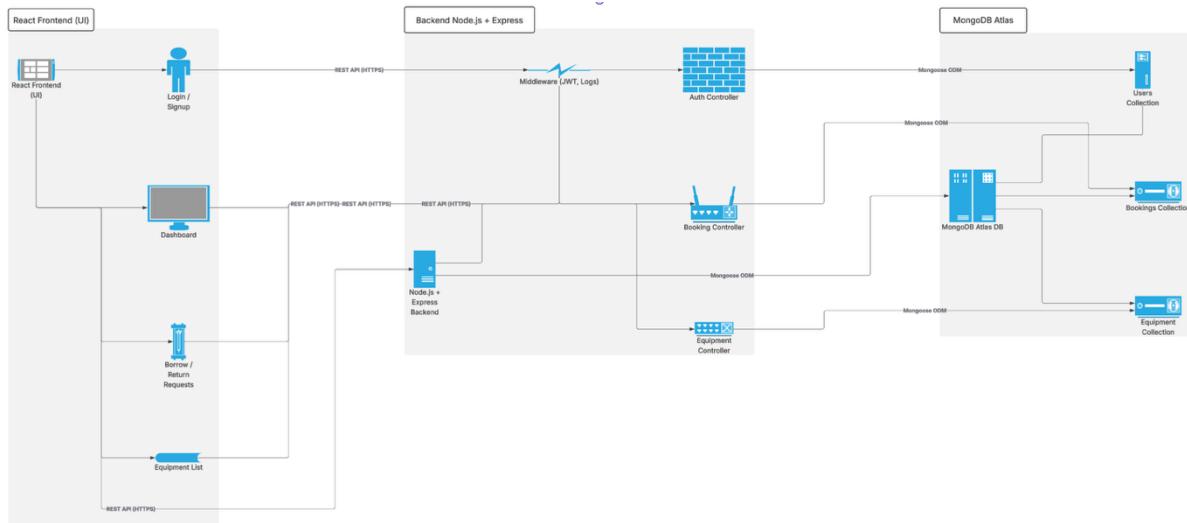
2. SYSTEM ARCHITECTURE

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



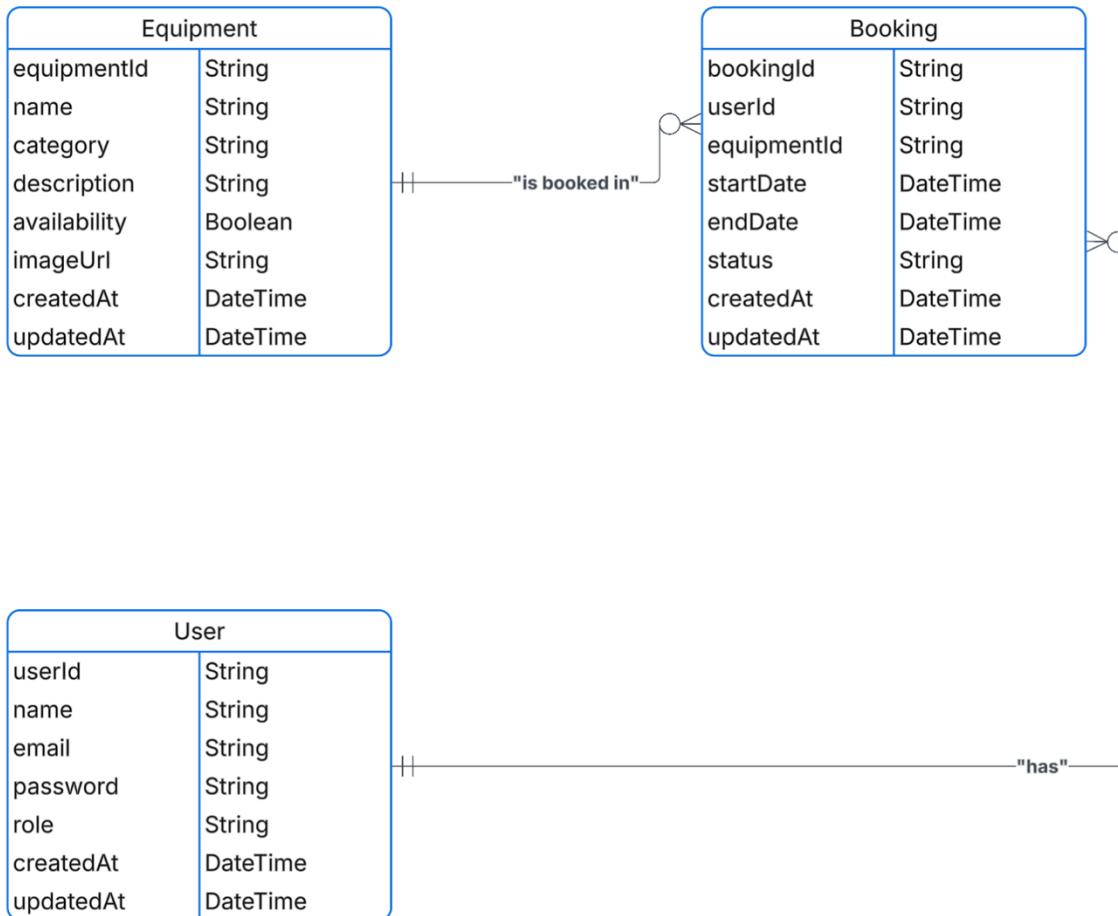
1. The **frontend client (React)** communicates with the backend REST API.
2. The **backend (Express server)** processes incoming HTTP requests and performs validation.
3. The **database layer (MongoDB)** handles CRUD operations for users, items, and bookings.
4. The **middleware layer** (auth, logging, 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**.

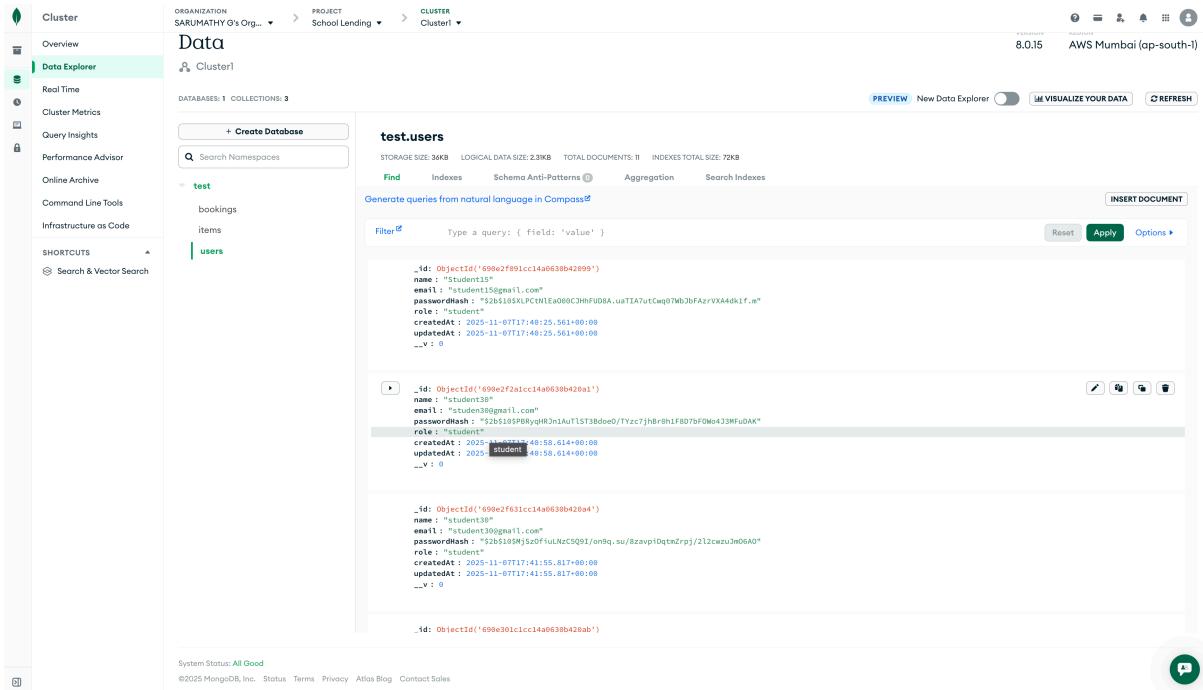
DATABASE RULES

- 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.

DATABASE SCREENSHOTS

The screenshot shows the MongoDB Compass interface for the `test.bookings` collection. The interface includes a sidebar with navigation links like Cluster, Overview, Data Explorer, and Performance Advisor. The main area displays the collection's storage size (36KB), logical data size (4.44KB), total documents (24), and index size (36KB). It features a search bar, a filter bar, and a results table. The results table lists documents with fields such as `_id`, `item`, `user`, `startDate`, `endDate`, `status`, `quantity`, `notes`, `createdAt`, and `updatedAt`. A preview bar at the top right indicates "New Data Explorer" and "REGION AWS Mumbai (ap-south-1) Version 8.0.15".

The screenshot shows the MongoDB Compass interface for the `test.items` collection. The layout is similar to the previous screenshot, with a sidebar and a main data exploration area. The collection has 13 documents. The results table shows items with fields like `_id`, `name`, `category`, `condition`, `quantity`, and `images`. One document is expanded to show its internal structure, including nested arrays for `images` and `description`. The interface includes a preview bar at the top right indicating "New Data Explorer" and "REGION AWS Mumbai (ap-south-1) Version 8.0.15".



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

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:

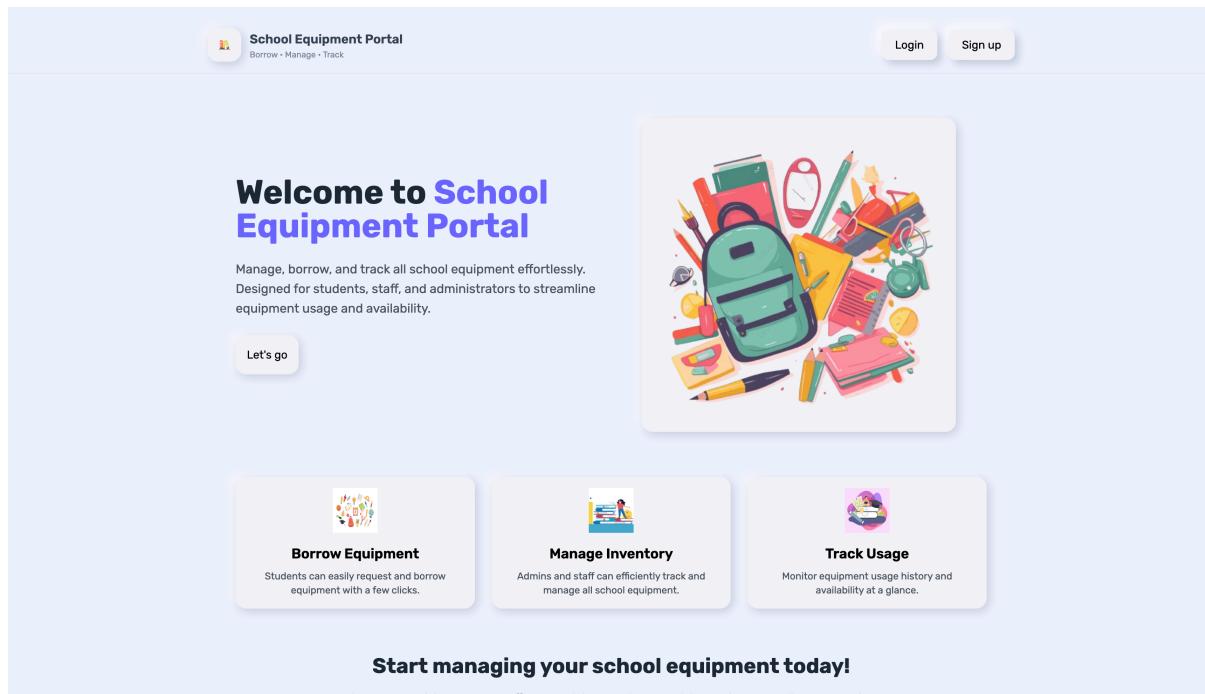
- Logout confirmation modal to prevent accidental sign-outs
- Delete confirmation modal before removing items
- Toastr success/failure alerts for all critical actions

5. Responsiveness:

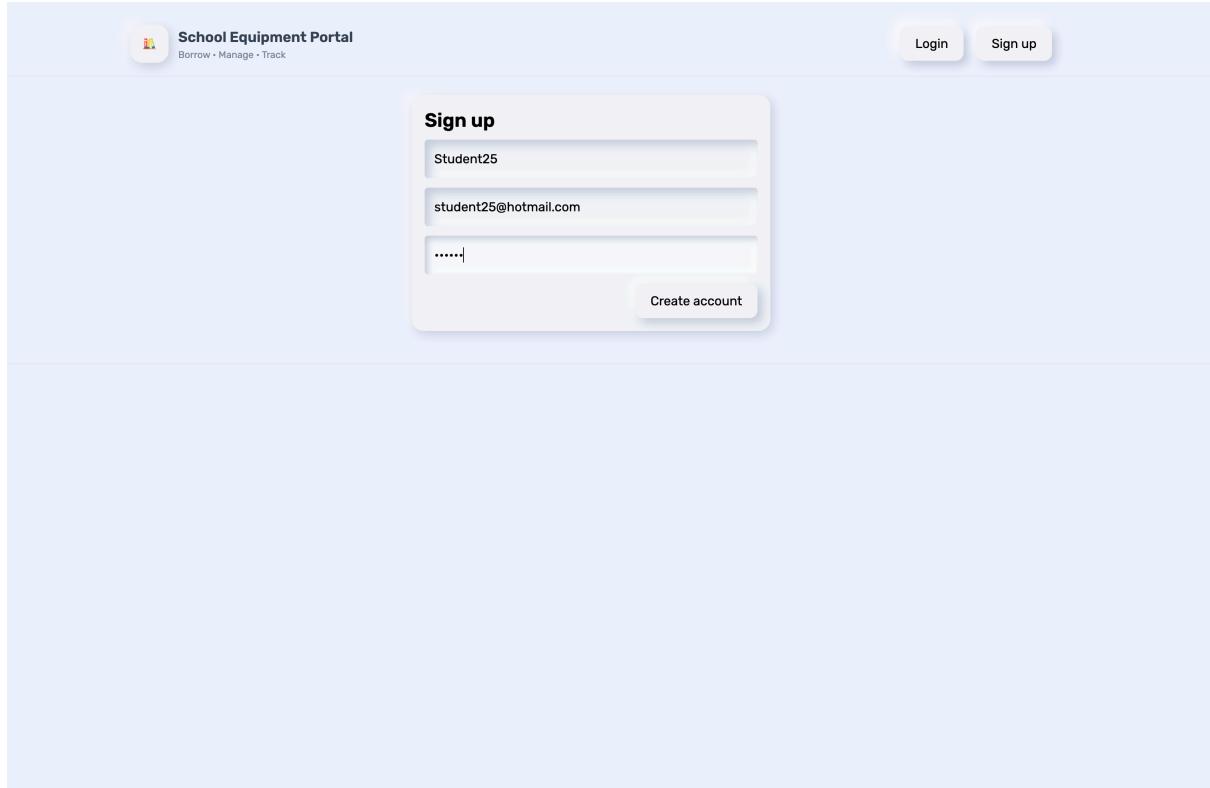
- Fully responsive layout optimized for desktops, tablets, and mobile devices.

UI SCREENSHOTS

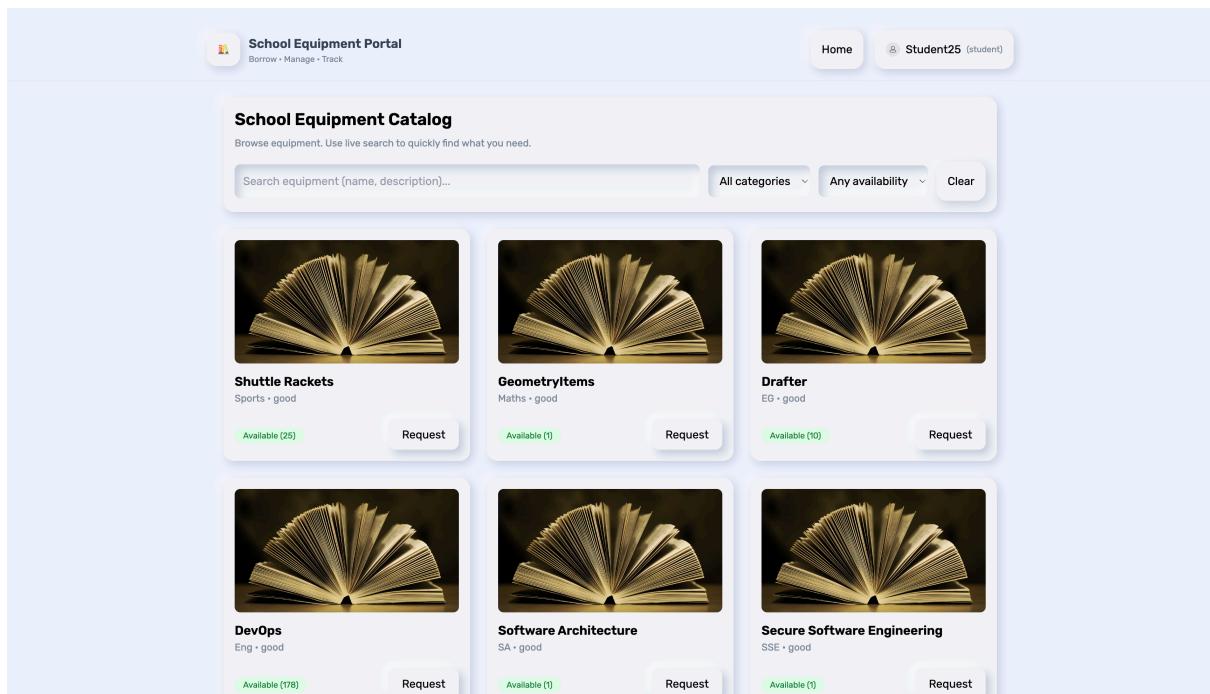
1. Landing Screen



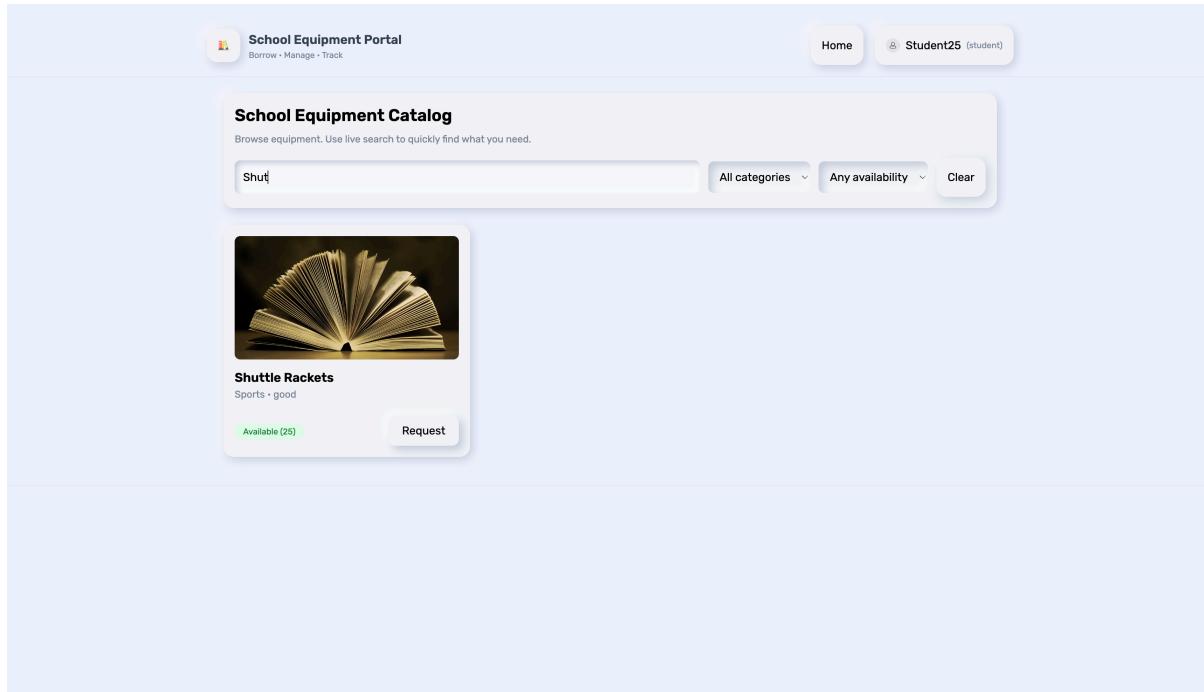
2. Sign-up Screen



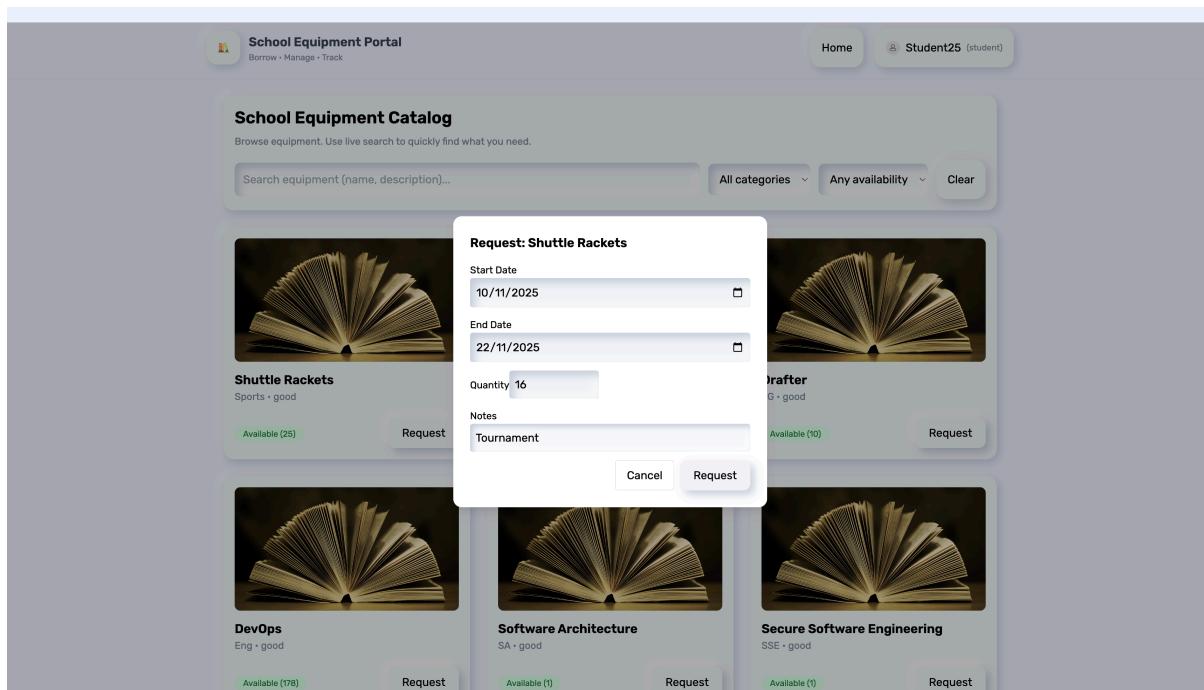
3. List of Items



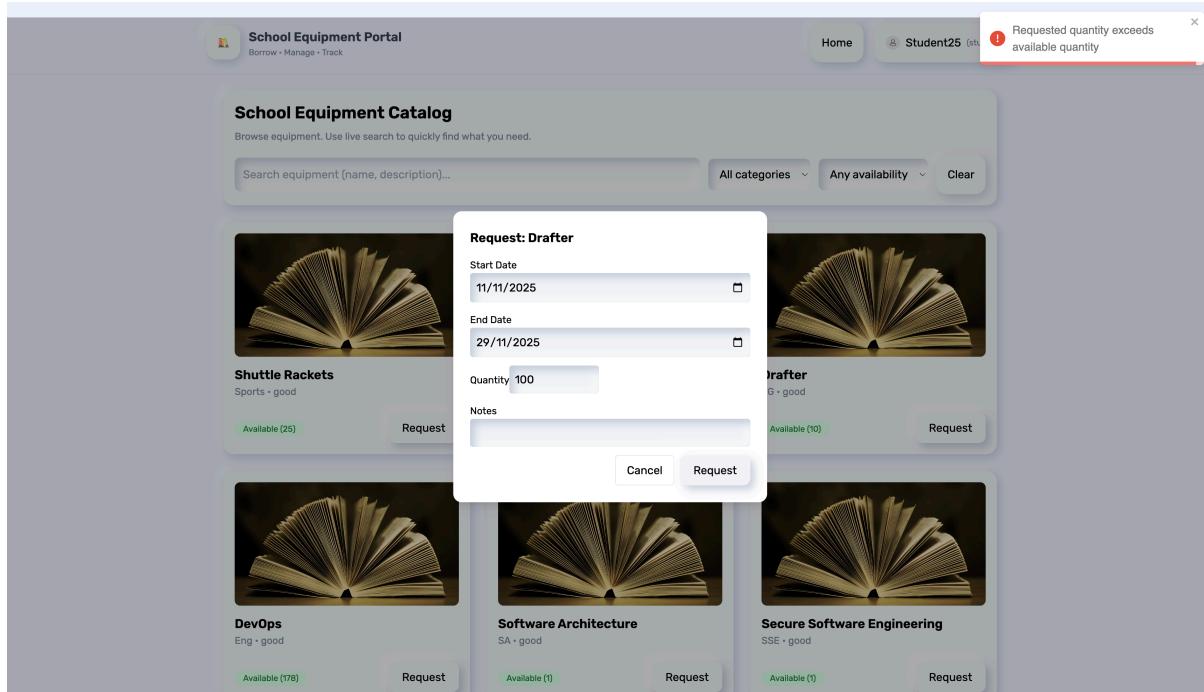
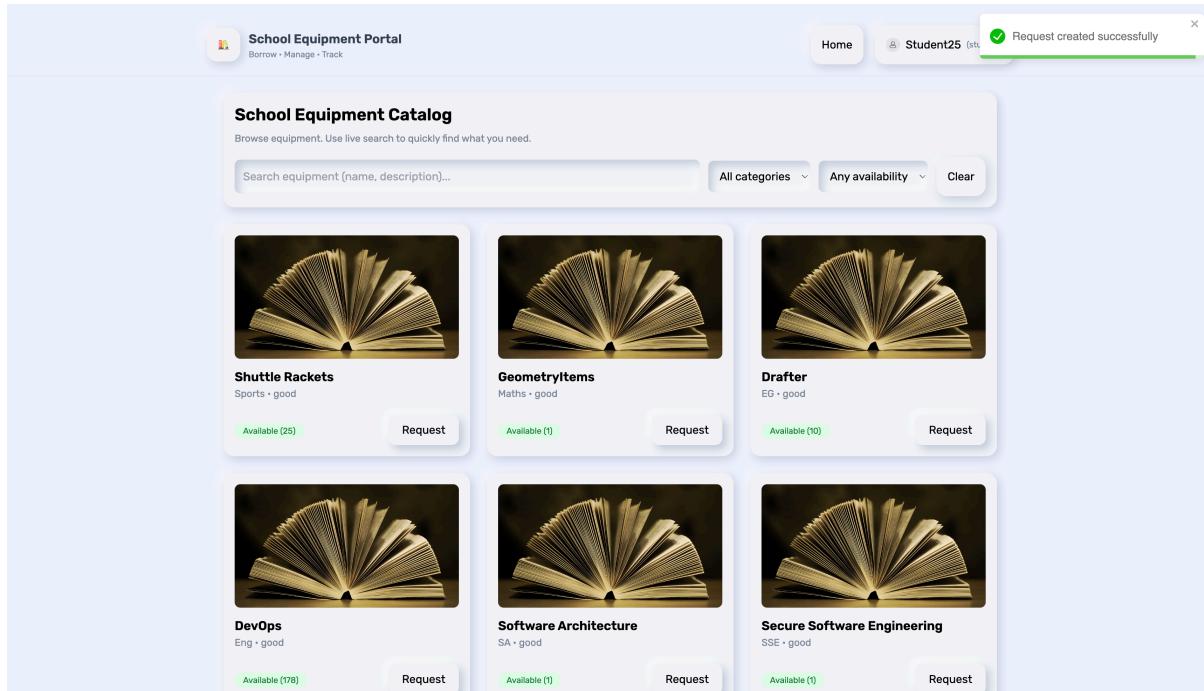
4. Search & Filter



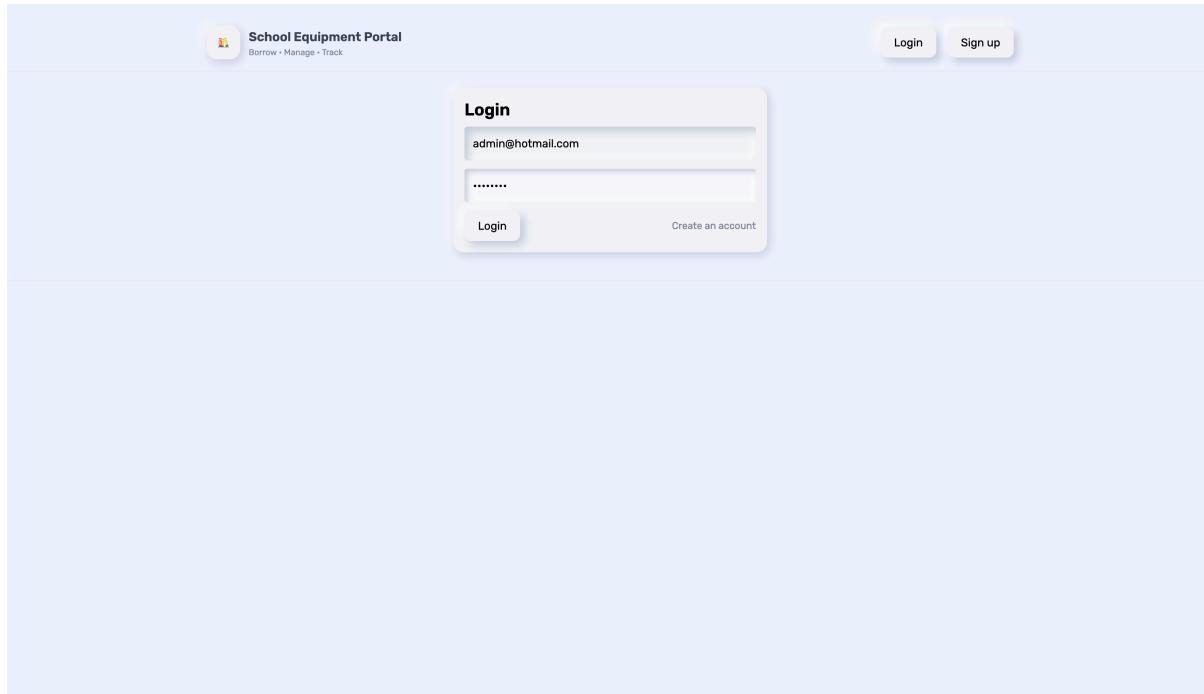
5. Request of Item



6. Toastr of requesting an item



7. Login Screen (Admin)



8. Manage Screen (Admin)

The screenshot shows the 'Manage Bookings' section of the portal. At the top, there are navigation links for 'Home', 'Manage', 'Swagger Docs', and 'Admin (admin)'. The main content area displays a list of equipment bookings:

- Microscope**
Student25 - 11/11/2025 → 21/11/2025
Qty: 5 • Status: **approved**
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**
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**
Approve Reject
- Drafter**
Admin - 14/11/2025 → 22/11/2025
Qty: 1 • Status: **approved**
Mark Returned
- DevOps**
Admin - 27/11/2025 → 29/11/2025
Qty: 20 • Status: **requested**
Approve Reject

9. Adding of Items (Admin)

The screenshot shows the 'School Equipment Portal' Admin - Items page. At the top, there are navigation links: Home, Manage, Swagger Docs, and Admin (admin). A modal window is open for adding a new item, titled 'Shuttle Rackets'. The modal contains fields for 'Sports' (set to '25') and 'Condition' (set to 'Good'). Below the modal, three items are listed in a grid:

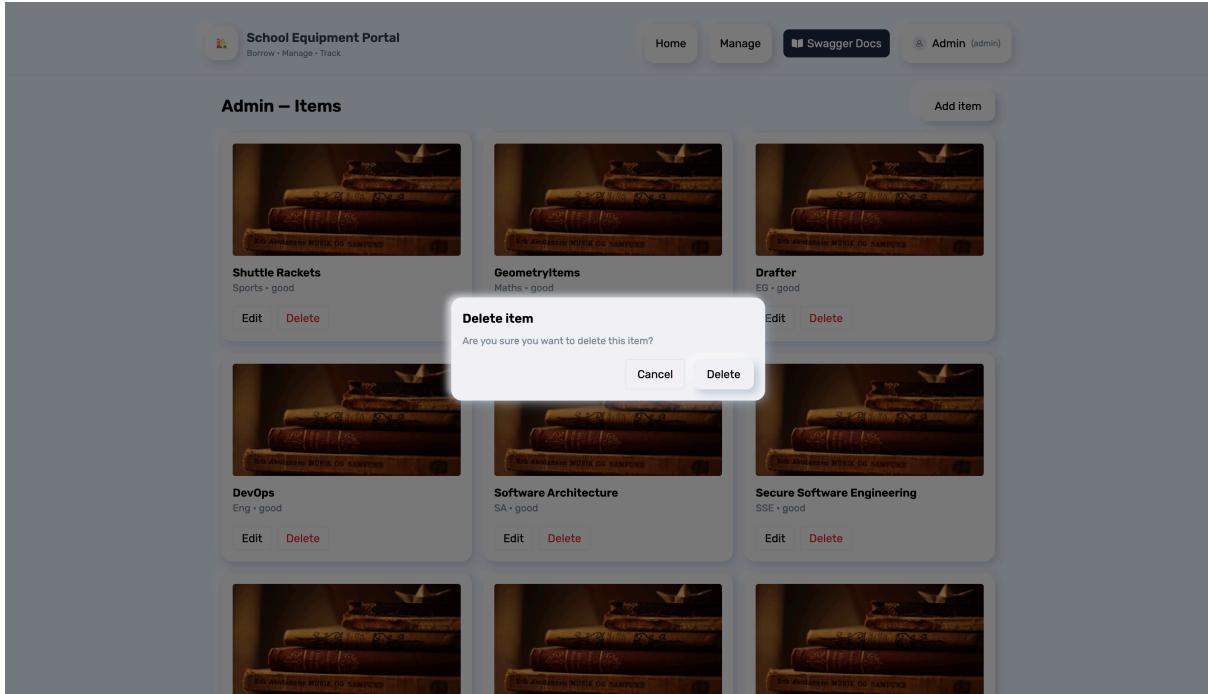
- GeometryItems**
Maths · good
[Edit](#) [Delete](#)
- Drafter**
EG · good
[Edit](#) [Delete](#)
- DevOps**
Eng · good
[Edit](#) [Delete](#)

Each item has a small thumbnail image of books.

10. Toastr Notification

The screenshot shows the 'School Equipment Portal' Admin - Items page. A green toastr notification at the top right says 'Item Shuttle Rackets added' with a checkmark icon. The rest of the page is identical to the previous screenshot, displaying the same list of items and their details.

11. Confirmation dialog for deletion of an item



12. Swagger Docs

A screenshot of the 'School Equipment Lending API' documentation generated by Swagger. The top navigation bar includes the 'Swagger' logo and links for 'Home', 'Manage', 'Swagger Docs', and 'Admin (admin)'. The main title is 'School Equipment Lending API 1.0.0 OAS 3.0'. Below the title, it says 'API documentation for the School Equipment Portal'. The interface shows several sections: 'Auth' (User authentication), 'Bookings' (Equipment booking and returns), and 'Items' (Equipment management). Each section lists specific API endpoints with their methods, URLs, and descriptions. For example, under 'Auth', there are 'POST /api/auth/signup' and 'POST /api/auth/login'. Under 'Bookings', there are 'POST /api/bookings', 'GET /api/bookings', 'PUT /api/bookings/{id}/approve', 'PUT /api/bookings/{id}/reject', and 'PUT /api/bookings/{id}/return'. Under 'Items', there is a single listed endpoint: 'GET /api/items'. On the right side of the interface, there is an 'Authorize' button with a lock icon.

Servers Authorize

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

13. Logout Confirmation

School Equipment Portal Borrow • Manage • Track

Home Manage Swagger Docs Admin (admin)

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

Search equipment (name, description)... All categories Any availability Clear

Shuttle Rackets
Sports • good
Available (25) 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 (9) Request

Confirm Logout
Are you sure you want to log out?
Cancel Logout

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.