

## Sprint 3: Locations CRUD (Backend)

### Goal:

Enable users to create, read, update, and delete inventory locations that can be assigned to materials.

### User Story:

"As a user, I want to manage and assign locations to materials so I can organize where items are stored."

### Tasks:

#### 1. Create Location Model [x] Completed

- Fields: name, description

#### 2. Create Locations Routes [x] Completed

- POST /api/locations - Add location
- GET /api/locations - Get all locations
- GET /api/locations/:id - Get single location
- PUT /api/locations/:id - Update location
- DELETE /api/locations/:id - Delete location

#### 3. Create Controller Functions [x] Completed

- Handle logic for all routes (CRUD)
- Use protect middleware to restrict access

#### 4. Connect Routes to server.js [x] Completed

- Mount routes under /api/locations

## 5. Test Routes in Thunder Client [x] Completed

- Confirm full CRUD operations work
- Ensure token is required for protected actions

## Dependencies Needed:

- Already covered by previous sprints: express, mongoose, dotenv, jsonwebtoken

## Deliverables:

- Location model
- RESTful API with CRUD operations
- Protected routes using auth middleware

## Next Sprint:

- Frontend: Display Materials and Locations
- User Story: "As a user, I want to view and interact with materials and locations through a user interface."