**Objective:**  
Develop a simple Personal Asset Management App for users to track their own assets (laptop, bike, phone, etc.). Each user should be able to register, log in, and manage their personal assets.

**Scope & Features**

**1. Backend (Spring Boot + Postgres)**  
Build REST APIs for:

* User Registration & Login (JWT-based authentication)
* CRUD APIs for Assets (each user manages their own assets only)
* APIs to fetch master data:
  + Asset Categories (stored in a separate table)
  + Asset Statuses (stored on a separate table)

**Data Model:**

**Users Table:**

* User id
* username
* password (hashed)
* email
* created at
* updated at

**Assets Table:**

* asset id
* user id (FK → Users)
* asset name
* category id (FK → Asset Categories)
* status id (FK → Asset Statuses)
* purchase date
* warranty expiry date (nullable)
* asset image URL (nullable)
* created at
* updated at

**Asset Categories Table:**

* category id
* category name (example: Laptop, Phone, Bike, Furniture)

**Asset Statuses Table:**

* status id
* status name (example: Active, Sold, Discarded)

**Backend Notes:**

* All Asset APIs must be secured — users should only access their own data
* Implement pagination for list APIs
* API endpoints should follow REST best practices

**2. Web App (React.js)**  
Build a responsive web app with:

* Login / Registration screens
* Dashboard showing my assets (table with pagination)
* Add Asset form (category and status dropdowns fetched from backend)
* Edit Asset form
* Delete Asset action
* View asset image (if URL is provided)
* API error and loading states handled gracefully

**3. Mobile App (React Native)**  
Cross-platform app supporting iOS and Android:

* Login / Registration screens
* My Assets list
* Add Asset
* Edit Asset
* Delete Asset
* Fetch categories/statuses from backend
* No offline support required
* Basic navigation and user experience

**Deliverables:**

* GitHub repository containing:
  + Spring Boot backend
  + React web app
  + React Native mobile app
* Postgres schema or migration scripts
* README.md with clear setup and run instructions

Note: Use MUI Components for front-end and pagination