Exercise 3

Scenario

You're developing an ecommerce for a startup selling shoes.

Each shoe model has a unique name and material (e.g. leather, rubber) and can be produced in different colors and sizes.
Each tuple (model, color, size) is unique.

Shoes are grouped into categories. Each shoe can be part of multiple categories and categories are structured as an ordered 1-N tree. For example:

- Winter Collection
  - Boots
    - Leather Boots
    - Rubber Boots
    - High Boots

So each category has a name, an optional parent and a weight/order.

A shoe can be part of multiple categories. For example, a shoe can be part of both "Leather Boots" and "High Boots" at the same time, but it could also be directly part of "Boots".

Also, the Marketing department wants to be able to manual order the shoes in each category.

Listing all shoes in a category should return all shoes in that category and all its subcategories avoiding duplicates.

Problem

Design the database for the catalog application and develop a web application which exposes a GraphQL API to query the catalog and navigate the categories tree, allowing to:

- paginate results;
- filter results by name, material, color and size;
- sort results by natural ordering (the order provided by the marketing) or by name.

The application should be covered by tests. BDD is preferred.

You can use any library and/or framework to solve the problem.