**Products & Discounts Manager with Choice of Backend Stack**

**Objective**

Build a Full Stack microservice API + frontend app with either Core PHP or Laravel on the backend, and Angular 19 on the frontend.

---

**Backend (Your Choice: Core PHP or Laravel)**

The backend should offer RESTful APIs with JSON responses. Key responsibilities:

- Authentication
    - User login with token-based authentication.
    - Seed users for testing; no registration needed.
- Products API
    - List products with pagination.
    - Single product details including price and applicable discounts.
    - Correct calculation and display of discount amounts and final prices.
    - **(Bonus)** Products API to allow search, and sorting
    - **(Bonus)** CRUD operations for products.
- Discounts API
    - List discounts with pagination.
    - Discount types: percentage or fixed amount.
    - **(Bonus)** Discounts API to allow search, and sorting
    - **(Bonus)** CRUD operations for discounts.
- Database
    - Use the given schema for Users, Products, Discounts, and their relations.
    - Implement according to chosen backend (manual SQL with PDO for Core PHP, or Eloquent ORM with Laravel).
- Code Quality
    - Clear separation of concerns (routing, business logic, data).
    - Use coding standards relevant to chosen stack.
    - **(Bonus)** Unit tests, code quality tools (phpcs, psalm).
    - **(Bonus)** API documentation using Swagger or Scribe (Laravel).

---

**Frontend (Angular 19)**

- Build a modern SPA that consumes the backend API.
- Functionalities:
    - Login page with token authentication.
    - Product listing page with pagination, sorting, search.
    - Product detail page showing discounts and calculated prices.
    - Discount listing page.
    - **(Bonus)** CRUD operations for products and discounts.
- Use Angular best practices (HttpClient, services, components).
- **(Bonus)** Responsive, clean UI (Angular Material optional).

| Feature | Core PHP Backend | Laravel Backend | Angular 19 Frontend |
|---|---|---|---|
| Authentication | Manual token with PDO | Laravel Passport/JWT | Token-based login and guards |
| Routing & Controllers | PHP files and manual routes | Laravel MVC + routing | Angular routing and modules |
| ORM/Database Access | PDO + manual SQL | Eloquent ORM | HTTP client calls to API |
| Pagination/Search/Sort | Hand-coded SQL queries | Eloquent query scopes | Client-side UI controls |
| Discount Logic | Custom PHP logic | Laravel service classes | Display calculated results |
| Code Quality & Testing | PHPUnit, code standards manual | PHPUnit, phpcs, psalm | Jasmine/Karma tests |
| API Documentation | Written/readme or Swagger JSON | Scribe or Swagger integration | N/A (focus on UI) |

Database Schema

```
-- USERS TABLE (seeded users)
CREATE TABLE users
 (
   id       BIGINT UNSIGNED auto_increment PRIMARY KEY,
   name     VARCHAR(100) NOT NULL,
   email    VARCHAR(150) UNIQUE NOT NULL,
   password  VARCHAR(255) NOT NULL,
   created_at TIMESTAMP NULL DEFAULT NULL,
   updated_at TIMESTAMP NULL DEFAULT NULL
 );

-- PRODUCTS TABLE
CREATE TABLE products
 (
   id       BIGINT UNSIGNED auto_increment PRIMARY KEY,
   name     VARCHAR(255) NOT NULL,
   description TEXT,
   price    DECIMAL(12, 2) NOT NULL,
   created_at  TIMESTAMP NULL DEFAULT NULL,
   updated_at  TIMESTAMP NULL DEFAULT NULL
 );

-- DISCOUNTS TABLE
CREATE TABLE discounts
 (
   id       BIGINT UNSIGNED auto_increment PRIMARY KEY,
   title    VARCHAR(255) NOT NULL,
   type     ENUM('percentage', 'fixed') NOT NULL,
   -- percentage (e.g. 10%), fixed (e.g. 100)
   value    DECIMAL(10, 2) NOT NULL,
   created_at TIMESTAMP NULL DEFAULT NULL,
   updated_at TIMESTAMP NULL DEFAULT NULL
 );

-- PRODUCT_DISCOUNT (pivot table: to allow multiple discounts per product)
CREATE TABLE product_discount
 (
   id       BIGINT UNSIGNED auto_increment PRIMARY KEY,
   product_id  BIGINT UNSIGNED NOT NULL,
   discount_id BIGINT UNSIGNED NOT NULL,
   FOREIGN KEY (product_id) REFERENCES products(id) ON DELETE CASCADE,
   FOREIGN KEY (discount_id) REFERENCES discounts(id) ON DELETE CASCADE
 );
```