

Project Overview

You will develop a basic ordering system where users can browse items, place orders, and receive applicable offers. Staff can manage inventory and offers. Authentication and real database persistence are out of scope (use in-memory storage).

Implementation Options

Choose one of the following:

- **Frontend:** Web-based using React.
- **Backend:** FAST API.

Features to Implement

1. **Browse Items**
 - Endpoint: `GET /items`
 - Users can view available items, stock levels, and prices.
2. **Order Items**
 - Endpoint: `POST /orders`
 - Users can add items to a cart and confirm an order.
3. **Apply Offers**
 - Endpoint: `GET /offers`
 - Users can see available discounts.
 - Discount applied automatically if conditions are met.
4. **Manage Inventory** (for staff)
 - Endpoints:
 - `POST /items-management` (Add/Update items)
 - `DELETE /items-management` (Remove items)
5. **Manage Offers** (for staff)
 - Endpoints:
 - `POST /offers-management` (Add/Update offers)
 - `DELETE /offers-management` (Remove offers)

Technology Guidelines

- Use a framework (React for frontend, Spring Boot or .NET for backend).
- Implement observability (logging, error handling).
- Keep code modular and production-ready.

Delivery

- Provide a GitHub repository with your implementation.
- Include a `README.md` explaining your approach, setup, and trade-offs.

Scope Reduction

- Authentication and authorization are not required.

- In-memory storage is sufficient.
- UI can be minimal but functional.