**CODEBASE 2: E-Commerce System Backend using FastAPI:**

Link: [ACE\_Python\_codebases / codebase2 · GitLab](https://gitrepo.valuelabs.com/ace_python_codebases/codebase2) (Branch: Master)

### Basic Overview of Repo

This project is a backend API for an e-commerce application built using FastAPI. It supports features like product listing, user authentication, shopping cart, and order placement.

**1. Assignment 1:** Implement Coupon Code Functionality

**Objective of Assignment**

To implement a coupon code system that allows customers to receive discounts during checkout.

### Detailed Assignment

* Create a new model Coupon with fields:
* code (unique string)
* discount\_percent (integer 1-100)
* expires\_at (datetime)
* is\_active (boolean)
* Create endpoints to: Create a new coupon (admin only), Apply a coupon code to a user's cart, Remove a coupon from a cart
* Enforce that: Expired or inactive coupons cannot be applied, Only one coupon can be applied to a cart at a time.
* Modify the order total calculation to include the coupon discount if applied.

### End Goal Which Will Be Evaluated

* Clean integration with existing models and logic.
* Proper use of Pydantic and SQLAlchemy models.
* Accurate validation and enforcement of business rules.
* Secure and efficient endpoints.

**2. Assignment 2:** Implement Wishlist Feature

**Objective of Assignment**

To allow users to maintain a personal wishlist of products.

### Detailed Assignment

* Create a Wishlist model linking User and Product.
* Implement endpoints to: Add a product to wishlist, Remove product from wishlist, View wishlist
* Prevent duplicate entries for the same user-product pair.

### End Goal Which Will Be Evaluated

* Efficient relationship handling.
* Optimized queries for retrieval.
* Intuitive API structure.
* Proper use of authentication to restrict access.

**3. Assignment 3:** Implement Order Cancellation with Refund Eligibility

**Objective of Assignment**

To allow users to cancel their order and receive a refund if the order is not yet shipped.

### Detailed Assignment

* Extend the Order model with: status: pending, paid, shipped, delivered, cancelled and cancelled\_at datetime
* Create endpoint for user to cancel an order.
* Only allow cancellation if order is not shipped.

### End Goal Which Will Be Evaluated

* Proper state transitions with validations.
* Clean and secure refund logic.
* Accurate status handling and rollback.
* Tests for corner cases (e.g., cancelling shipped orders).

**4. Assignment 4:** Add Multi-Address Shipping Support

**Objective of Assignment**

To allow users to store and manage multiple shipping addresses and select one during checkout.

### Detailed Assignment

* Create an Address model: user\_id, address\_line\_1, address\_line\_2, city, state, postal\_code, country, is\_default (boolean)
* Users can: Add, update, delete, and list their addresses and Set an address as default (only one can be default).
* During checkout: Allow user to select an address or default to the is\_default one and Link the selected address to the created order.

### End Goal Which Will Be Evaluated

* Reusability of address logic across endpoints.
* Smooth integration with order creation.
* Correct enforcement of single default address.
* Data consistency and validation.