**Architecture Decisions**

**a) Models & Relationships**

* **User**: Includes role field to differentiate customer vs admin.
* **Vendor**: Represents sellers, has many products.
* **Product**: Belongs to vendor.
* **Cart / CartItem**: Session-based or DB-based cart; items belong to a cart.
* **Order / OrderItem**: Orders created per vendor on checkout.
* **Payment**: Linked one-to-one with order, storing payment status (pending, paid).

**b) Cart & Checkout**

* **Cart grouped by vendor** ensures each vendor gets a separate order.
* **Stock validation** during cart add and checkout ensures inventory consistency.
* **Checkout**:
  1. Validate stock.
  2. Group items by vendor.
  3. Create orders & order items per vendor.
  4. Deduct product stock.
  5. Create a Payment record (marked as paid for simulation).

**c) Admin & Customer separation**

* **Middleware** (auth, admin) restricts routes.
* Admin can:
  + View all orders.
  + Filter by vendor, customer, or payment status.
  + See order details with items & totals.
* Customers can:
  + Browse products grouped by vendor.
  + Manage cart & checkout.

**d) Service Classes / Events / Listeners**

* **Services**: Encapsulate business logic for cart and checkout. Keeps controllers thin.
* **Events/Listeners**:
  + OrderPlaced → could send emails/notifications.
  + PaymentSucceeded → could trigger invoices or shipment workflows.

**e) Blade + Bootstrap**

* Blade templates separate logic and view.
* Bootstrap ensures responsive UI quickly.
* Tables for cart, orders, and products provide clean layouts.

**f) Routing & Controller Decisions**

* Resourceful routing for products, cart, orders.
* Custom routes for multi-vendor checkout.
* Controllers are divided:
  + CartController → cart operations.
  + OrderController → customer orders view.
  + AdminOrderController → admin management of orders.

**2️⃣ Assumptions Made**

* Each product belongs to **exactly one vendor**.
* Users have a single role (customer or admin).
* Payments are always successful.
* Cart belongs to a user; no guest cart.
* All orders are created immediately at checkout (no pending/processing state).
* No shipment, tax, or discount logic yet.
* Frontend uses Bootstrap for quick UI; advanced JS frameworks not used