**Algorithm: Fetching Inventory, Price, User, admin and Flyer Data**

**1. Initialization:**

**Step 1:** Start the process.

**Step 2:** Initialize system databases (Product Database, User Database, Admin Database, Store Database, and Flyer Database).

**Step 3:** Retrieve the user's location (for nearby store detection) and logged-in status (User/Admin).

**Step 4:** Request user input for product search (keywords, categories, store names, or flyer search).

**2. Fetch Product Details (User Request):**

**Step 1:** User searches for a product or selects an item from a grocery list.

**Step 2:** Query the **Product Database** based on the search keywords or selected item.

* **Check:** If the product exists in the database.
  + **Yes:** Proceed to the next step.
  + **No:** Display an error message that the product is unavailable.

**Step 3:** Retrieve real-time product prices from the **Store Database** for all stores near the user’s location.

* Compare prices across stores.
* Display the list of stores with corresponding prices.

**3. Fetch Inventory Details:**

**Step 1:** Once the product is identified, query the **Store Database** for inventory availability.

* Check: If the product is available in the inventory of the selected stores.
  + **Yes:** Display available quantity for each store.
  + **No:** Notify the user of stock unavailability.

**Step 2:** If inventory exists, fetch additional details such as **product quantity offered for price matching**.

**4. Fetch Flyer Details:**

**Step 1:** Query the **Flyer Database** for the current flyer information related to the product or store.

* Retrieve details such as:
  + Flyer starts and end dates.
  + Flyer-specific prices or discounts.

**Step 2:** Display flyer details, including price validity period (start and end dates) and promotional pricing (if applicable).

**5. Admin-Specific Data Fetch (Admin Request):**

**Step 1:** If logged in as **Admin**, query additional admin-specific data such as:

* Current flyer status for the selected store.
* Existing product categories for updating or deleting.
* Price matching rules across stores.

**Step 2:** Retrieve current products and store information for **adding, updating, or deleting** items.

* Allow admins to modify price matching details, product quantities, and store records.

**6. User-Specific Data Retrieval:**

**Step 1:** Query the **User Database** for logged-in users to fetch:

* User account details.
* Personalized grocery list.
* Past searches or purchase history.

**Step 2:** Fetch **notifications** related to product status (availability, price drop, flyer start date, etc.).

**Step 3:** Display personalized recommendations based on current flyer data, inventory, and user preferences.

**7. End Process:**

**Step 1:** Finalize and display all retrieved information (prices, inventory, flyers, personalized notifications).

**Step 2:** End.

**Pseudocode for Fetching Inventory, Price, User, admin and Flyer Data**

BEGIN

// STEP 1: Initialize system databases and user session

INITIALIZE ProductDatabase, UserDatabase, AdminDatabase, StoreDatabase, FlyerDatabase

SET userLocation = GET\_USER\_LOCATION()

SET isLoggedIn = CHECK\_USER\_LOGIN\_STATUS()

// STEP 2: Get user input for product search or grocery list selection

PROMPT userInput FOR product search OR SELECT from grocery list

// STEP 3: Fetch product details based on user input

IF userInput EXISTS THEN

FETCH productDetails FROM ProductDatabase WHERE product = userInput

IF productDetails FOUND THEN

// Display product details to user

DISPLAY productDetails

ELSE

// Notify user that product is unavailable

DISPLAY "Product not found"

RETURN

END IF

ELSE

DISPLAY "No input provided"

RETURN

END IF

// STEP 4: Fetch real-time price comparison from nearby stores

FETCH nearbyStores FROM StoreDatabase WHERE storeLocation WITHIN userLocation

FOR EACH store IN nearbyStores DO

FETCH storeProductPrices FROM StoreDatabase WHERE product = userInput

END FOR

DISPLAY storeProductPrices FOR all nearby stores

// STEP 5: Fetch inventory details for each store

FOR EACH store IN nearbyStores DO

FETCH inventoryDetails FROM StoreDatabase WHERE product = userInput AND store = storeName

IF inventoryDetails AVAILABLE THEN

DISPLAY "Available quantity: " + inventoryDetails.quantity

ELSE

DISPLAY "Out of stock at " + storeName

END IF

END FOR

// STEP 6: Fetch flyer information for selected store and product

FETCH flyerDetails FROM FlyerDatabase WHERE product = userInput AND store = storeName

IF flyerDetails EXISTS THEN

DISPLAY "Flyer Price Validity: " + flyerDetails.startDate + " to " + flyerDetails.endDate

ELSE

DISPLAY "No current flyer available for this product"

END IF

// STEP 7: Admin-specific actions (if logged in as admin)

IF isLoggedIn = "Admin" THEN

// Admin can update product quantities for price match

FETCH adminOptions FROM AdminDatabase

IF adminOptions = "Manage Product Quantities" THEN

PROMPT admin FOR updateQuantity

UPDATE ProductDatabase SET quantity = updateQuantity WHERE product = selectedProduct

END IF

// Admin can manage stores for price matching

IF adminOptions = "Manage Stores for Price Matching" THEN

PROMPT admin FOR updateStore

UPDATE StoreDatabase SET storeName = updateStore WHERE product = selectedProduct

END IF

END IF

// STEP 8: User-specific data retrieval

IF isLoggedIn = "User" THEN

FETCH userDetails FROM UserDatabase WHERE userID = loggedInUserID

DISPLAY "Grocery List: " + userDetails.groceryList

DISPLAY "Notifications: " + userDetails.notifications

END IF

// STEP 9: Finalize and display retrieved data

DISPLAY "Final Price Comparison, Inventory Availability, and Flyer Details"

END