TechXpress E-commerce Platform Database System

- A new user registers on the TechXpress platform by providing their name, email, password, phone number, and address. The system validates and stores this data in the User table. Once registered, the user can log in and browse available products.

- An admin manages the product catalog by adding new products with details such as name, description, price, stock quantity, discount (if applicable), color, and brand. Each product belongs to a specific category, ensuring proper classification. Products are stored in the Product table and linked to categories via the Category entity. The admin can also update or remove products when necessary.

- When a user selects a product, they add it to their **shopping cart**. The system temporarily stores the **product ID**, **quantity**, and **price at the time of selection** within the cart. The system also checks **stock availability** before confirming the addition to the cart.

- To place an order, the user reviews the cart and proceeds to **checkout**. The system generates a new record in the **Order** table, including the **total amount**, **order status** (e.g., Pending, Shipped, Completed), **creation date**, and **last update date**.

- The order is linked to the user via the **User ID**, and the **list of purchased products, along with their quantities and prices** are tracked through the relation between the relation between **Products** and **Orders**.

- Once the order is placed, an admin updates the order status based on processing stages (e.g., "Processing," "Shipped," or "Delivered"). The system updates the Order table's status and last update date. If an order is canceled, stock quantities are restored to prevent overselling.

- Additionally, stock management ensures that when an order is completed, the stock quantity (UnitsInStock) for each purchased product is reduced accordingly. If a product runs out of stock, it is marked as unavailable until restocked.

Entity Relationships:

➢ **User ↔ Order** → One user can place multiple orders.

➢ **Order ↔ Product** → Many-to-Many relationship **handled by storing product details within the Order**

➢ **Product ↔ Category** → Each product belongs to one category.

➢ **Order ↔ User** → Each order is linked to a single user.