**Chages in Schema:**

**Addition of the Wishlist model:**

* + The user model was added to store the wishlist items for each user. It has a wishlist that links to the User model, and an array of items that stores the products the user has added to their wishlist.

1. **Referencing Products:**
   * Each item in the items array of the Wishlist references a productId, which is linked to the Product model using mongoose.Schema.Types.ObjectId. This allows us to easily associate the products in the wishlist with their full details by using the .populate() method.
2. **Handling Wishlist Operations:**
   * The routes in the wishlist.routes.js handle three main operations:
     1. **Adding a product to the wishlist:** A product is added to the items array in the Wishlist model.
     2. **Viewing the wishlist:** The items array is populated with full product details using populate('items.productId').
     3. **Removing a product from the wishlist:** A product is removed from the items array by filtering out the product using its productId.

**Database Flow:**

1. **Adding to the Wishlist:**
   * A user adds a product to their wishlist via the POST /add-to-wishlist/:productId route.
   * If the user doesn’t already have a wishlist, one is created. The product is then added to the items array in their Wishlist document.
2. **Viewing the Wishlist:**
   * The user can view their wishlist using the GET /wishlist route. This retrieves the Wishlist for the user and populates the productId field in the items array with product details.
3. **Removing from the Wishlist:**
   * If the user wants to remove a product, the POST /remove-from-wishlist/:productId route removes that product from the items array of their Wishlist.

**Example of Database Structure:**

1. **User Collection:**
   * Each user has a unique userId that is referenced in the Wishlist collection.
2. **Wishlist Collection:**
   * The Wishlist collection stores the user's wishlist with a reference to the user (userId) and an array of items. Each item contains a reference to a productId, linking it to a specific product.
3. **Product Collection:**
   * The Product collection stores details about each product (e.g., name, price, description, etc.).

**Conclusion:**

This structure allows users to have personalized wishlists, with the ability to add or remove products as needed. By linking the wishlist to the User and Product collections via references, the application can efficiently manage and display the wishlist items with full product details.