**Android Application for Product Sale**

This application allows users to browse products, add them to a cart, and purchase them. It can utilize a remote database accessed through a RESTful API.

**Here's a breakdown of the functionalities in mobile app:**

**1. Authentication (Sign Up/Login): 10%**

* User Registration: Enable users to sign up by creating an account with a username and password, and optionally input their email, phone number, and address.
* Login: Allow existing users to log in using their registered credentials.
* Password Security: Ensure secure storage of user passwords using hashing techniques to prevent unauthorized access.

**2. List of Products: 15%**

* Product Fetching: The app should fetch product data from the remote database via a RESTful API.
* Product Display: Show a list of products, including product name, image, price, and a brief description.
* Sorting and Filtering: Provide options for users to sort products by price, popularity, or category, and filter products by specific attributes such as brand, price range, or ratings.

**3. Product Details: 15%**

* Detailed View: Show all product details, such as product description, technical specifications, and multiple images.
* Add to Cart: Allow users to add products to the cart from this screen with an option to select the desired quantity.

**4. Product Cart: 15%**

* Cart Overview: Display a list of products added to the cart along with the product image, name, price per item, selected quantity, and the total price.
* Cart Management: Allow users to adjust the quantity of products, remove items, or clear the entire cart.
* Cart Total: Dynamically update the total cart amount based on changes in the cart.

**5. Billing: 10%**

* Payment Integration: Integrate with a payment gateway like VNPay, ZaloPay, or PayPal to securely process payments.
* Billing Details: Allow users to choose a payment method and enter their billing and shipping information.
* Order Confirmation: Show a confirmation screen after successful payment, providing users with an order summary and payment status.

**6. Notification (Cart Badge): 15%**

* Cart Badge Notification: Display a notification badge on the app icon to show the number of items in the cart when the app is closed.
* Notification Library: Utilize libraries like NotificationCompat to handle Android notifications for this feature.

**7. Map Screen: 10%**

* Store Location: Integrate Google Maps to show the store's location on a map.
* Directions: Provide users with the option to get directions to the store using their current location.

**8. Chat Screen: 10%**

* Real-Time Chat: Allow users to chat with store representatives in real-time through an integrated chat service.
* Chat API: Use APIs like Firebase or implement a custom chat server to handle communication between users and the store.

**Database Choice:**

* MySQL or SQL Server

**Backend:**

* REST APIs are required to implement in any language.