**Clarity of Purpose (20%)**

**Objective:**  
The application will be a CART-based e-commerce platform where users can browse a variety of clothing items, add them to their shopping cart, and place orders. The application aims to simplify online clothing shopping by providing a seamless and user-friendly experience, allowing customers to quickly find, select, and purchase clothing.

**Category:**  
CART (e-commerce)

**User System Design (20%)**

**Objective:**  
The application will allow users to interact with the frontend (React app), and the client and server will communicate to fetch and manage data.

**Interactions:**

* **User to Client Side (React App):**
  + When users visit the homepage, they can browse clothing categories.
  + Users can log in to their accounts to view their profile, order history, and saved items.
  + Users can add items to their shopping cart and proceed to checkout.
* **Client to Server Communication:**
  + The client (React app) will make requests to the server (Node.js/Express) for data like available products, order details, and user information.
  + The server will send data back to the client, such as product lists, search results, and cart information.
* **Server to Database (MongoDB):**
  + The server will interact with a MongoDB database to store user data, product information, and order details.
  + The database will handle CRUD operations like adding products, updating cart items, and processing orders.

**User Interaction Diagram (Example):**

1. **Homepage:**
   * User visits the homepage, sees clothing categories.
   * User clicks on a category (e.g., Men’s Apparel) to view items.
   * User clicks on a product to view more details.
2. **Cart Management:**
   * User adds an item to the cart.
   * User proceeds to checkout, enters shipping details.
3. **Login/Signup:**
   * User signs up or logs in to manage personal data and past orders.

**Feature Justification (20%)**

**Objective:**  
Justify the importance of selected features for the application.

* **Product Search:**  
  The product search feature allows users to easily find clothing items based on various criteria, enhancing their shopping experience. This is critical for an e-commerce app, as users need to navigate a large inventory.
* **User Authentication (Login/Signup):**  
  By allowing users to register and log in, we ensure secure access to personal information, order history, and saved products. This also makes the shopping experience more personalized.
* **Shopping Cart & Checkout:**  
  The ability for users to add items to the cart, review them, and then proceed to checkout is a fundamental e-commerce feature. It facilitates the purchase process and ensures a smooth transaction flow.

**Feasibility (20%)**

* **Challenges:**
  + **Time Management:** Balancing project work with class assignments might be challenging, so breaking tasks into weekly sprints can help.
  + **Complex Features:** Implementing payment gateways and ensuring secure transactions might be complex, so I will look for tutorials and documentation online.
* **References:**
  + YouTube tutorials for building e-commerce websites with React and Node.js.
  + Existing e-commerce open-source projects on GitHub.

**Submission Documentation:**

* **Idea:** The e-commerce platform focuses on providing users with an intuitive shopping experience for clothing. It's a popular app type, with an emphasis on user interaction and ease of purchase.
* **Breakdown of Pages and Endpoints:**
  + **Pages:**
    - Homepage: Displays categories and featured products.
    - Product Detail Page: Detailed view of each clothing item.
    - Cart: Users can view and modify their cart items.
    - Checkout: Users can enter shipping details and complete the purchase.
    - Login/Signup: User registration and login pages.
  + **API Endpoints:**
    - GET /products: Fetch all products.
    - POST /cart: Add an item to the cart.
    - GET /cart: Retrieve the cart contents.
    - POST /checkout: Process the order.