German International University
Faculty of Informatics and Computer Science
Dr. Nada Sharaf
Eng. Donia Ali
AL. Amany Hussein
Eng. Hania Ashraf
Eng. Omar Ashraf

Software Project I, Winter 2024
Practice Assignment 4

Exercise 4-1 React

Design and build a custom user interface for a shopping system using React, connecting it to the backend endpoints developed in practice assignment 2.
Github Link: https://github.com/DoniaAli77/ShoppingExample

Guidelines:

Freedom of Design: Students are free to design the structure and layout of their components, pages, and functionalities as they see fit. Creativity is encouraged in UI/UX design.
Core Requirements: The application should communicate effectively with the backend to retrieve, display, create, update, and delete data for both products and users.
Implement essential CRUD operations for products and users.
Include functionalities to manage a shopping cart and perform a checkout process.
Your Project should be composed of minimum 2 pages.

Submission: This practice assignment is an individual PA.
Each student should push his work to his own github and submit the github link via mail to your tutorial TA.
Students must submit before the end of next week (before Thursday 21 November) .

Suggested (but not mandatory) Features:

Components:
Think about modular components like ProductList, ProductDetail, UserList, UserDetail, ShoppingCart, and Checkout etc., that can be reused throughout the application.

Pages:
A homepage that could showcase featured products or highlight various categories.
Detailed product and user pages for viewing and editing information.
A shopping cart page that displays items and allows for checkout.

Example:
Home Page: Display all products using the GET /products endpoint.
Product Detail Page: Display details for a specific product using GET /products/:id.
User Page: Display all users with GET /users.
User Detail Page: Display details of a specific user using GET /users/:id.
Shopping Cart Page: Display the shopping cart of a specific user using GET /users/cart/:id.

Interactivity:
Buttons for actions like adding products to the cart, editing user details, and processing checkouts.

Forms:
Implement forms for user input where necessary (e.g., creating a new product, updating user info).

State Management:
Use React's useState and useEffect hooks for state management.

Routing:
Use react-router-dom for seamless navigation between different parts of the application.

Error Handling:
Show appropriate error messages or toasts for API call failures and edge cases.

Design:
Students are encouraged to use CSS frameworks or libraries (e.g., Bootstrap, Tailwind, Material-UI) or create custom styling. Additional Features: Students can add any extra features they think would enhance the user experience (e.g., search bars, product filters).