Assignment 2: Fake Store Milestone 2 specification

General Information

**Project Title: Fake Store**

In this project, you will further develop your mobile application "Fake Store." For this milestone, you will integrate Redux with reduxjs/toolkit to manage the application's state, particularly focusing on the shopping cart's functionality. The objective is to enhance your skills in state management within React Native applications and ensure seamless user interaction with the shopping cart.

**Submission Guideline, Milestones, and Deadlines**

The assignment is structured into three phases, with each concluding in a milestone submission. Deadlines are set as follows:

- **Milestone 1 Deadline**: End of Week 7
- **Milestone 2 Deadline**: End of Week 9
- **Final Submission Deadline**: End of Week 11

All submissions must be made through the provided submission link. Each student is allowed a total of four submission attempts, accommodating for any potential mistakes. It is crucial to check that the correct files are being uploaded before submission.

**Version Control and Submission Guidelines**

- Utilize GitHub for version control and repository hosting.
- Set your GitHub repository to private initially and invite your tutor as a collaborator for progress tracking and feedback. If your tutor is already a collaborator on your project, there is no need to reinvite them unless you change the repository.
- Clearly label your commits for each milestone with "Milestone 1", "Milestone 2", or "Final Submission" in your commit messages to facilitate organized assessment.

Milestone 2

**Requirements**

1. **State Management:**

   - It is compulsory to use Redux with reduxjs/toolkit for managing the state of the shopping cart.

2. **Bottom Tab Navigation:**

   - Implement a bottom tab navigation with two tabs (each tab should have a suitable icon):
     - **Products**: Links back to the original product category screen.
     - **Shopping Cart**: Opens a new screen that displays the contents of the shopping cart.
   - Implement a badge on the Shopping Cart tab to indicate the number of items in the cart, using Redux state to manage and display the count.

3. **Shopping Cart Screen:**

- Display a message "Your shopping cart is empty" when there are no items in the cart.
- Use a `FlatList` to display multiple products in the shopping cart when items are present.
- Include increase and decrease buttons to adjust the quantity of each product directly from the cart screen. Adjusting quantities should update the Redux state and reflect immediately in the UI.
- Remove items from the cart if their quantity drops to zero.
- If the shopping cart is not empty, display the total number of items and the total cost of the products at the top section of the screen.

**Marking Criteria**

1. **Bottom Tab Navigation (5 points):**

   - Successful implementation of bottom tab navigation with suitable icons, including the accurate display of a badge that indicates the total quantity of items in the shopping cart. For instance, if the shopping cart contains two products—Product A with a quantity of 1 and Product B with a quantity of 2—the badge should display a total of 3.

2. **Shopping Cart Screen Implementation (5 points):**

   - Correct implementation of the shopping cart screen using `FlatList`, handling both empty, single, and multiple product situations.

**Submission Guidelines**

You are required to submit:

- The URL link to your GitHub project repository.
- A screenshot of your GitHub commit history related to 'Milestone 2'.
- A screenshot of the Redux shopping cart slice from your project code.
- Screenshots of the shopping cart screen in the following states:
  - Empty cart
  - Cart with one type of product
  - Cart with multiple products
  - Ensure that in all scenarios, the badge displayed on the Shopping Cart tab, as well as the total cost and item count on the shopping cart screen, accurately reflect the contents of the shopping cart.

**Penalties:**

- Late submissions may incur a penalty in accordance with the university's relevant policies.
- Failure to demonstrate dynamic interactions or incomplete functionality may result in a deduction of up to 15% of the total marks for Milestone 2.
- Ensure that the text in your screenshots is visible. If they are too small to read, you must provide additional zoomed-in screenshots where critical information is clearly presented. Failure to do so will result in a penalty.

**Remark:**

Some students may have already implemented more functions required for Milestone 2, such as adding extra tabs in the bottom tab navigation. It is fine to include additional functions and features in Milestone 2; these enhancements will not incur any penalties.

A template is provided for Milestone 2, and its use is optional. If you choose not to use the template, please ensure that all required information specified in the template is included. Omissions may result in penalties. If you have any questions or need clarification, please do not hesitate to contact me.