Assignment 2: Fake Store - Final Submission

Overview

For the final submission of Assignment 2, you are required to complete and demonstrate a mobile application named "Fake Store." This app will integrate with both the fakestoreapi website and a specifically provided Fake Store API server to simulate a comprehensive online shopping experience. Your final application should encompass all functionalities as shown in the demo video, providing a thorough demonstration of your skills in mobile application development with React Native.

Submission Requirements

1. **Project Integration:**
   - Connect your application with the Fake Store API server provided. The API server details are available on GitHub: Fake Store Server.
   - Ensure all client-server interactions are handled efficiently and securely.

2. Final Documentation and Video Demonstration:

- Submit a document that includes:
  - The GitHub link to your project repository.
  - A link to a video demonstration showcasing all functionalities implemented in your app. The video should clearly demonstrate how the app operates and how each part of the specification has been met.
  - Please note that the video should not exceed 5 minutes in length. Content beyond 5 minutes may not be considered during the assessment.
  - For master's students, you are required to provide a project reflection of up to one page in length.

3. **App Requirements:**
   - Your application should consist of 9 different screens, each fulfilling specific functions. These screens and their functionalities will be detailed in the marking criteria section.

Marking Criteria

Your application will be evaluated based on the implementation and functionality of the following screens, all of which must be clearly demonstrated in your video submission:

1. Splash Screen (5 points):

- Upon launching the app, a splash screen should be displayed for a few seconds.
- This screen should prominently feature the name "Fake Store" or another suitable title that reflects the nature of the application.
- The splash screen should automatically disappear after a brief duration.
- You have the option to either implement a custom splash screen component or modify the `app.json` file to use a static image designed by you.

2. Product Category List Screen (5 points):

- Users can access the Product Category List Screen by clicking the 'Products' icon in the bottom tab navigation.
- This screen is only accessible to users who are logged in, either through signing up as a new user or signing in as an existing user. If a user is not logged in, clicking the icon should display an appropriate alert message prompting the user to sign in or sign up.
- Upon first display, the screen will fetch product categories from the fakestoreapi website. During this fetching process, an `ActivityIndicator` should be used to indicate that data retrieval is in progress.
- The fetched results should be cached within the app, allowing the categories to be displayed immediately on subsequent visits to this screen.
- In addition to the four categories fetched from the fakestoreapi, the screen should also display your name at the end of the category list as proof that you developed the app.
- The screen should have a title of `Product Categories`.
- Clicking on any category will navigate the user to the Product List Screen, which displays products specific to the selected category.

3. Product List Screen (5 points):

- Clicking on a category in the Category List Screen will navigate to the Product List Screen.
- The title of this screen should reflect the name of the selected category. Utilize a single Product List component to display varying lists of products for different categories.
- Upon first accessing a category, the products for that category should be fetched from the fakestoreapi website. During this fetching process, display an `ActivityIndicator` to indicate that the data is being retrieved.
- Subsequent visits to the same category should display the list of products immediately, utilizing cached data.
- Accessing a different category for the first time should trigger a new fetch request specific to that category.
- Each product listing should include a small-sized image of the product, along with its title and price.
- Display the product list using a `FlatList`, ensuring that users can scroll through if the products exceed one screen's capacity.
- Clicking on a product should navigate to the Product Detail Screen.
- Include a 'Back' button at the bottom of the screen, above the navigation bar, which when clicked, will return the user to the Product Category List Screen.

4. Product Detail Screen (10 points):

- Clicking on a product in the Product List Screen navigates to the Product Detail Screen.

- This screen should have the title "Product Details."

- It should prominently display the product’s image at a large scale, along with the product's title, rating, count (in my demo app, it uses sold for this field), price, and description.

- The description should be placed within a scrollable `View` to accommodate content that exceeds the visible area.

- The first visit to a product's detail screen should initiate a fetching process, indicated by an `ActivityIndicator`. Subsequent visits to the same product’s detail screen should display the details immediately, using cached data.

- The screen should feature two buttons:
  - A **Back** button that navigates back to the Product List Screen.
  - An **Add to Cart** button that adds the product to the shopping cart. This action should immediately be reflected in the badge count on the shopping cart icon in the bottom navigation bar. Continuing to click the **Add to Cart** button will add the product multiple times and increase the badge count with each click.
  - If the shopping cart is empty, the cart icon should display without a badge.

5. Shopping Cart Screen (10 points):

- The second item in the bottom navigation bar is the Shopping Cart icon. Users can click this icon to navigate to the Shopping Cart screen.
- Users must be logged in to access the Shopping Cart screen.
- This screen should display the title "Shopping Cart."
- If the cart is empty, a message stating "Your Cart is Empty" should appear.
- If there are products in the cart, the screen should display the total number of items and the total cost directly below the title.
- Beneath the cart summary, a list of products in the cart should be displayed.
- For each product, display a small image, the title, the price, and two buttons: "-" to decrease the quantity and "+" to increase it. If the quantity reaches zero, the product should be automatically removed from the cart.
- Each product should appear only once in the cart, with the quantity reflecting the total number of times it has been added.
- Changes in product quantity should immediately update the badge on the cart icon and the totals in the shopping cart summary.
- The badge on the shopping cart should be cleared when a user logs out.
- Shopping cart data should be updated on the Fake Store server via API calls whenever changes are made. Upon logging out and logging back in, the badge and the contents of the shopping cart should persist and reflect the state prior to logout.
- If the shopping cart is not empty, a "Check Out" button should be displayed at the bottom of the screen. Clicking this button should empty the cart, create a new order for the user with all items that were in the cart, and an alert should pop up to inform the user that a new order has been created.

6. My Orders Screen (10 points):

- The "My Orders" icon, located next to the shopping cart icon on the bottom navigation bar, is labeled accordingly.
- Clicking this icon navigates to the My Orders Screen, which is accessible only to logged-in users.
- Orders can have one of three statuses: new orders, paid orders, and delivered orders. Orders are initially created as new orders through the checkout process in the Shopping Cart Screen. A new order can be paid to become a paid order, which can then be received to become a delivered order.
- When a new order is created via the checkout button on the Shopping Cart Screen, an API call is made to the Fake Store server, which returns the order ID of the new order.

- The My Orders Screen features three labels for the different order statuses. For example, the first label is "New Order" and displays the number of new orders for the user.
- Each label has a small caret icon on the right end. Clicking the caret toggles the expansion and contraction of the display. When expanded, all orders of the associated status are listed under the label.
- By default, each order displays in compact mode, showing only the order ID, number of items, and total price, along with a caret to expand the order for more details.
- Clicking the caret of an order toggles between compact and expanded modes. In expanded mode, details of the order, including a small image, title, and quantity of each product, are displayed beneath the order summary.
- In expanded mode, each new order includes a "Pay" button, and each paid order includes a "Receive" button. Clicking the "Pay" button triggers an API call to update the order's status to paid, while clicking the "Receive" button updates a paid order to delivered.
- After clicking the button and initiating the API call, an alert displays confirming the new status (e.g., "Your order is paid" or "Your order is delivered"). Following this, the entire order list and all labels are refreshed to reflect the new order statuses.
- Similar to the shopping cart icon, the My Orders icon displays a badge indicating the number of new orders. If there are no new orders, the icon shows no badge. This badge updates automatically whenever the number of new orders changes.
- Upon user logout, the badge on the My Orders icon disappears. When the user logs back in, the badge should immediately reflect the status as it was before logout.

7. User Profile Screen (10 points):

- The last icon on the bottom navigation bar is for the user profile.
- When a logged-in user clicks this icon, they are navigated to the User Profile Screen.
- This screen displays the user’s name and email address along with an "Update" button and a "Sign Out" button.
- Clicking the "Update" button prompts a pop-up form where the user can enter a new name and password.
- Below the input fields, there are two buttons: "Confirm" and "Cancel."
- Clicking "Confirm" sends the update request to the Fake Store server.
  - If the update is successful, an alert notifies that the name and password have been successfully updated, and the screen returns to the main profile view.
  - If the update fails, an alert displays the error message, and the update form remains displayed.
- The "Cancel" button discards any changes, returning the display from the update form back to the user profile screen.
- Clicking the "Sign Out" button logs the user out of the application.

8. Sign In Screen (form) (5 points):

- If a user is not logged in, navigating to the user profile icon will display the Sign In form.
- The form should be titled "Sign in with your email and password" and include two input fields for the user's email and password.
  - The text entered in the password field should be masked for security.
- Below the input fields, there should be two buttons: "Clear" and "Sign In."
  - The "Clear" button clears all input fields.
  - The "Sign In" button submits the email and password to the Fake Store server via an API call.

- If the credentials match the server's records, the user will be logged in with an authorization token received from the server. This token will expire after one hour, and the display will transition to the User Profile Screen.
- If the credentials do not match the server records, an alert displaying "Wrong email or password" should appear.
- At the bottom of the form, a label stating "Switch to: sign up" should be provided. Clicking this label navigates the user to the Sign Up Screen (form).

9. Sign Up Screen (form) (10 points):

- An unlogged user can toggle between the Sign In and Sign Up forms by clicking the label at the bottom of the forms. This label should toggle between "Switch to: sign up" and "Switch to: sign in" respectively.
- The title of the Sign Up form should be "Sign up a new user."
- This form includes three input fields: Name, Email, and Password, and two buttons: "Clear" and "Sign Up."
    - Input in the Password field should be masked for security purposes.
- The "Clear" button clears all the input fields.
- Clicking the "Sign Up" button submits all input fields to the Fake Store server via an API call.
- Server-side form validation checks for errors. If an error occurs, it should be displayed through an alert.
- If the registration is successful without any errors, a new user account will be created, and the display will transition to the User Profile Screen.

Additional Notes

- Please review the requirements carefully and compare them with the demo video provided on the course website to ensure a complete understanding. If any part is unclear, feel free to contact me.

- In your demo video, please create and demonstrate at least two different user accounts to show that, upon logging in with different users, the user profiles, orders, and shopping cart contents vary accordingly.