Project Development Prompts

This README outlines the series of development prompts used for guiding the creation of a Flutter-based e-commerce application.

Table of Contents

Prompt #1: Initial Setup

You are taking on the role of a Flutter developer tasked with creating an offline e-commerce application from scratch. This prompt will lay the groundwork for the project, focusing on the overall structure and key components.

Project Overview

The aim is to develop a Flutter-based e-commerce app. This prompt provides a general overview of the app's structure and main functionalities to set the context for detailed development in subsequent interactions.

Overview of Key Components

- **User Authentication:** Implement a basic login interface with username and password fields.
- **Product Listing Interface:** Create a scrollable list view that will display products. The list of products is hardcoded.
- **Shopping Cart Functionalities:** Allow users to interact with the shopping cart, adding and removing products as needed.

Technical Framework

- **State Management:** The app will utilize the Provider package to ensure modularity and testability.
- **Testing:** The structure will be designed to facilitate straightforward unit testing for all components.
- **Code Quality:** The development will adhere to the latest Flutter best practices.

Objective

The goal of this prompt is to establish a solid foundational understanding and prepare for detailed feature development.

Prompt #2: Simple Login Interface

Development Task: Simple Login Interface

Objective

Create a user interface that allows users to log in with a username and password. The login should always navigate to the next view without validating the credentials.

Requirements

User Interface

- Design a login page with fields for username and password.
- Include a login button that navigates to the next view when clicked.

Functionality

- Input fields accept user input, but no verification is performed.
- On clicking the login button, the app navigates directly to the next view.

Expected Outcome

A functional login page that lets users enter a username and password and proceed to the next view by clicking the login button, with no input validation.

Prompt #3: Displaying a List of Products

Task: Implement a feature to display a list of products in a scrollable list view within your Flutter app.

Requirements

Product List Interface

Create a screen to display a list of products.

List Item Layout

Design each list item to show a product image on the left, followed by the product name. Include an "Add" button at the end of each list item.

Functionality

Simulate retrieving the product data to populate the list but the list itself is hardcoded. Use Provider package for state management to handle the product list.

Expected Outcome

A screen displaying a scrollable list of products, with each item showing a product image, name, and an "Add" button at the end.

Prompt #4: Adding Products to Shopping Cart

Objective: Enable users to add products to the shopping cart by clicking the "Add" button associated with each product in your Flutter app.

Requirements

Adding Products

Implement functionality so that clicking the "Add" button adds the corresponding product to the shopping cart.

Updating UI

After adding a product to the cart, replace the "Add" button with an icon to signify successful addition.

Functionality

Ensure that the shopping cart is updated with the added product. Update the UI dynamically to reflect changes after adding a product. Manage the state of the shopping cart using the Provider package.

Expected Outcome

Users should be able to add products to the shopping cart by clicking the "Add" button associated with each product. After successful addition, the "Add" button should be replaced with an icon indicating that the product has been added to the cart.

Prompt #5: Adding Header Title and Cart Button to the AppBar in the Product List View

Objective: Enhance the product list view in your Flutter app by adding a header title in the AppBar and a cart button at the end of the AppBar.

Requirements

Header Title

Display a title/header at the top of the AppBar in the product list view.

Cart Button

Place a cart button at the end of the AppBar in the product list view.

Functionality

Opening the Cart

Implement functionality so that clicking the cart button opens the cart view.

Expected Outcome

The AppBar in the product list view should contain a header title at the top and a cart button at the end. Clicking the cart button should open the cart view.

Prompt #6: Implementing the Shopping Cart View

Objective: Create a view in your Flutter app to display the products added to the shopping cart. Each product should be shown with an icon and product name on the left, and a "Remove" button with an icon at the end of the row.

Requirements

Shopping Cart View

Develop a new screen to show the products added to the shopping cart. Display each product with an icon and product name on the left side of the row. Include a "Remove" button with an icon at the end of each row.

Back Button

Add a "Back" button to the AppBar of the shopping cart view, enabling users to return to the overall product view.

Functionality

Removing Products

Implement functionality so that clicking the "Remove" button removes the corresponding product from the shopping cart.

Expected Outcome

The shopping cart view should display all the products added to the cart, with each product showing an icon, name, and a "Remove" button. Clicking the "Remove" button should remove the respective product from the cart. The AppBar in the shopping cart view should include a "Back" button allowing users to return to the overall product view.

Prompt #7: Displaying Total Price and Buy Button in the Shopping Cart View

Objective: Enhance the shopping cart view in your Flutter app by displaying the total price of all products below the product list, along with a "Buy" button.

Requirements

Total Price and Buy Button

Show the total price of all products below the product list in the shopping cart view. Include a "Buy" button below the total price.

Snackbar Display

Upon clicking the "Buy" button, display a snackbar with the message "Purchased".

Functionality

Buy Button Action

Implement functionality so that clicking the "Buy" button displays a snackbar confirming the purchase.

Expected Outcome

The shopping cart view should display the total price of all products below the product list, along with a "Buy" button. Clicking the "Buy" button should trigger a snackbar with the message "Purchased".

Prompt #T8: Writing Unit Tests for User Authentication

Objective: Develop unit tests to ensure that users can log in with a username and password, and that successful login always navigates to the next view without logic verification.

Requirements

Test Scenarios

Write unit tests to verify that users can successfully log in with valid username and password combinations. Ensure that the login process navigates to the next view without performing logic verification.

Navigation Testing

Verify that navigation to the next view occurs after successful login, without any logic verification checks.

Expected Outcome

Unit tests should confirm that users can log in with valid credentials and navigate to the next view.

Prompt #T9: Generating Test Cases for Product Listing and Adding to Cart Features

Objective: Develop test cases to ensure the correct functionality of displaying a list of products in a ListView and enabling users to add products to the shopping cart in your Flutter app.

Test Case Formulation

Displaying Product List

Scenario: Verify that the product list is properly displayed in a ListView. Expected Outcome: Ensure that all products are visible and scrollable within the ListView.

Product List Entry Layout

Scenario: Check the layout of each item in the product list. Expected Outcome: Verify that each list item displays a product image on the left, followed by the product name, and an "Add" button at the end.

Adding Product to Cart

Scenario: Test the functionality of adding a product to the shopping cart. Expected Outcome: Ensure that clicking the "Add" button for a product successfully adds the product to the cart.

Updating UI After Adding to Cart

Scenario: After adding a product to the cart. Expected Outcome: Verify that the "Add" button is replaced by an icon to indicate successful addition to the cart.

Prompt #T10: Testing AppBar Title and Shopping Cart Button

Objective: Develop test cases to ensure the correct functionality of displaying a title and a shopping cart button in the AppBar, as well as managing the shopping cart view in your Flutter app.

Test Case Formulation

Testing AppBar Title

Scenario: Verify the presence and correctness of the title in the AppBar. Expected Outcome: Ensure that the AppBar at the top of the product list view contains the expected title.

Testing Shopping Cart Button in AppBar

Scenario: Check the presence and functionality of the shopping cart button in the AppBar. Expected Outcome: Verify that the AppBar at the top of the product list view contains a shopping cart button at the end, and clicking it opens the shopping cart view.

Testing Shopping Cart View Layout

Scenario: Ensure that the shopping cart view displays added products correctly. Expected Outcome: Verify that each product in the shopping cart view shows an icon and product name on the left, followed by a "Remove" button with an icon at the end of the row.

Testing Back Button in AppBar of Shopping Cart View

Scenario: Check the presence and functionality of the back button in the AppBar of the shopping cart view. Expected Outcome: Ensure that the AppBar in the shopping cart view contains a back button allowing users to return to the overall product view.

Prompt #T11: Removing Items from Shopping Cart

Objective: Develop test cases to ensure the correct functionality of removing items from the shopping cart and handling the purchase process in your Flutter app.

Test Case Formulation

Testing Removal of Items from Shopping Cart

Scenario: Verify the functionality of removing a product from the shopping cart. Expected Outcome: Ensure that clicking the "Remove" button for a product removes it from the cart.

Testing Display of Total Price and Buy Button

Scenario: Check the display of the total price of all products and the presence of the "Buy" button below the product list in the shopping cart view. Expected Outcome: Verify that the total price of all products is accurately displayed, and there is a "Buy" button available.

Testing Purchase Process

Scenario: Test the purchase process by clicking the "Buy" button. Expected Outcome: Ensure that clicking the "Buy" button triggers a snackbar displaying the message "Purchased".