

## Contents

|  |   |
|--|---|
| Flutter Shopping App Documentation.....          | 1 |
| 1. Overview .....                                | 1 |
| 2. Project Structure.....                        | 1 |
| 3. App Features .....                            | 2 |
| 4. Responsive Design.....                        | 2 |
| 5. State Management.....                         | 2 |
| 6. Navigation Flow .....                         | 3 |
| 7. Asset Management .....                        | 3 |
| 8. Requirements .....                            | 3 |
| 9. Future Enhancements .....                     | 4 |
| 10. pubspec.yaml Summary .....                   | 4 |
| 11. How to run.....                              | 5 |
| 12. Sample Output in Medium Phone API 36.0 ..... | 6 |
| 13. Sample Output in Chrome Browser .....        | 9 |
| 14. GitHub Repository Name .....                 | 9 |

# Flutter Shopping App Documentation

## 1. Overview

This Flutter application is a simple shopping app with a home page listing product, a cart system, and a welcome screen. It demonstrates key Flutter principles like state management, navigation, UI design, and data modeling.

---

## 2. Project Structure

The app follows a modular structure:

- lib/data/: Contains sample product data.
  - lib/models/: Contains the product model.
  - lib/screens/: Contains main screen widgets like home, cart, and welcome screens.
  - lib/widgets/: Contains reusable UI components.
  - main.dart: Entry point and global state controller.
- 

### 3. App Features

- Display of predefined products with image, title, and price.
  - Add/remove items to/from the cart.
  - View total price and item count in the cart.
  - Drawer-based navigation between home and cart.
  - Banner section for visual appeal.
  - Responsive badge notification for cart items.
- 

### 4. Responsive Design

- Uses MediaQuery to adapt layout for mobile and tablet screens.
  - Dynamically adjusts GridView layout (2 columns for phones, 3 for tablets).
  - Button/icon sizes and text are scaled to maintain usability across devices.
- 

### 5. State Management

- **StatefulWidget** used in main.dart to manage cart state.
  - The addToCart and removeFromCart functions are passed down to children via constructors.
  - CartProvider (with ChangeNotifier) is implemented but not currently integrated. It's ready for enhancement if switching to Provider-based architecture.
-

## 6. Navigation Flow

- **WelcomeScreen:** Launch screen that routes to HomeScreen.
  - **HomeScreen:** Displays product list and drawer menu.
  - **CartScreen:** Shows selected items, total price, and remove options.
  - Uses Navigator.push() and pushReplacement() for screen transitions.
- 

## 7. Asset Management

Ensure the following image assets are included in the assets/images/ folder:

- bluebags.jpg
- redshoes.jpg
- wallet.jpg
- hat.jpg
- cart\_banner.jpg

Also, include them in the pubspec.yaml like:

```
flutter:  
  assets:  
    - assets/images/
```

---

## 8. Requirements

- Flutter SDK (latest stable)
  - Dart environment
  - IDE: Android Studio / VS Code
  - Emulators or Chrome browser
  - Asset images located in assets/images/  
(e.g., bluebags.jpg, redshoes.jpg, wallet.jpg, hat.jpg, cart\_banner.jpg)
-

## 9. Future Enhancements

- Implement CartProvider with Provider for cleaner state handling.
  - Add product details page.
  - Enable product quantity management.
  - Use a database or API for dynamic product data.
  - Add user authentication and checkout functionality.
- 

## 10. pubspec.yaml Summary

The pubspec.yaml file in the Flutter project defines key configurations for building and managing the app. The name is set to first\_flutter\_app, with a short description of the project, and the app version is specified as 1.0.0+1. Under environment, it specifies that the project requires Dart SDK version ^3.7.2 to ensure compatibility. The dependencies section includes core packages needed by the app: the flutter SDK itself and the cupertino\_icons package for using iOS-style icons. For development, the dev\_dependencies section includes flutter\_test for writing and running tests, and flutter\_lints to enforce code quality rules. The flutter section enables Material Design support with uses-material-design: true, which allows the use of built-in Material icons and components. Finally, under assets, it lists image directories and files that the app can access at runtime, including the entire assets/images/ folder and a specific file assets/images/cart\_banner.jpg, which must exist in my project directory.

- **App Info**
  - name: Project name → first\_flutter\_app
  - description: Brief app description
  - version: App version → 1.0.0+1
- **Environment**
  - Requires Dart SDK version ^3.7.2
- **Dependencies**
  - flutter: Core Flutter SDK
  - cupertino\_icons: iOS-style icons (^1.0.8)

- **Dev Dependencies**
    - flutter\_test: For writing tests
    - flutter\_lints: Code linting (^5.0.0)
  - **Flutter Settings**
    - uses-material-design: true: Enables Material icons
  - **Assets**
    - assets/images/: Loads all images in this folder
    - assets/images/cart\_banner.jpg: Specific image included
- 

## 11. How to run

To run your Flutter project in **IntelliJ IDEA**, follow these steps:

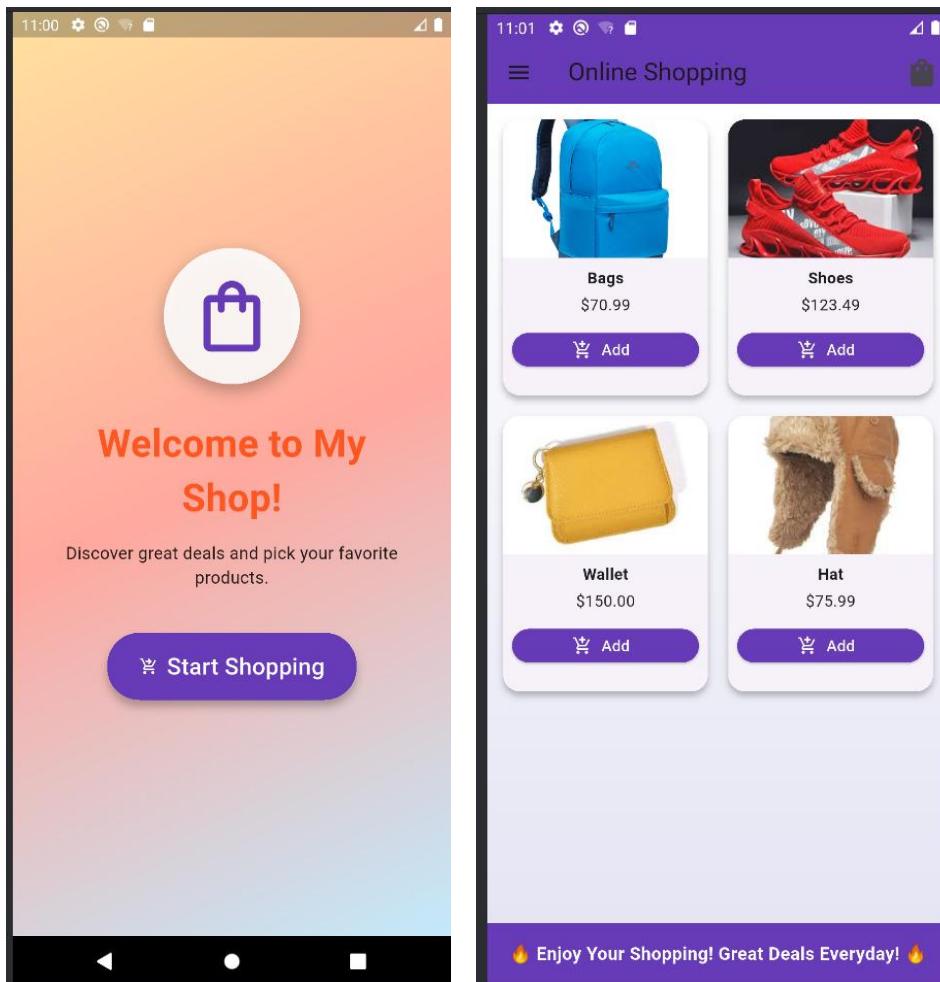
1. **Install Plugins:**
  - Open IntelliJ IDEA.
  - Go to File > Settings > Plugins.
  - Search for and install both **Flutter** and **Dart** plugins.
  - Restart the IDE after installation.
2. **Open Your Project:**
  - Go to File > Open, and select your Flutter project folder (e.g., first\_flutter\_app).
3. **Set Up Flutter SDK** (if not set already):
  - Go to File > Settings > Languages & Frameworks > Flutter.
  - Click on the Flutter SDK path and point it to your local Flutter SDK installation directory.
4. **Run the App:**
  - Connect a physical device or start an emulator.
  - In the top-right of IntelliJ, make sure the correct device is selected.

- Click the green **Run**  button or go to Run > Run 'main.dart'.

This will build and launch your Flutter app from the main.dart file, and you'll see it running on your selected device.

---

## 12. Sample Output in Medium Phone API 36.0



The above images displays a simple **e-commerce shop interface** with the following elements:

- **Screen Shot 1 - Header:**

- "Welcome to My Shop!"
- A subtitle: "Discover great deals and pick the favorite products."
- A "**Start Shopping**" button (likely a call-to-action).

- **Screen Shot 2 - Product Listings:**
  - Four products are displayed, each with:
    - **Name** (Bags, Wallet, Shoes, Hat)
    - **Price** (70.99,70.99,150.00, 123.49,123.49,75.99)
    - An "Add" button (presumably to add items to a cart).
  - **Footer:**
    - A closing message: "Enjoy Your Shopping! Great Deals Everyday!"

This following image displays a **shopping cart interface** for an online store with the following sections:

- **Screen Shot 3 - Shopping Cart Summary:**
  - "**Total Items: 2**" (indicating 2 items are in the cart).
  - "**Total: \$194.48**" (calculated sum of selected items).
- **Screen Shot 4 - Footer:**
  - A closing message: "**Enjoy Your Shopping! Great Deals Everyday!**"
  - "**Total Items: 2**" (indicating 2 items are in the cart).

## Exercise 1 – Aye Kyi Kyi Cho (StdID-1276026)

11:01 11:03

My Shopping Cart Online Shopping

2

Shoes \$123.49

Bags \$70.99

Bags \$70.99

Shoes \$123.49

Wallet \$150.00

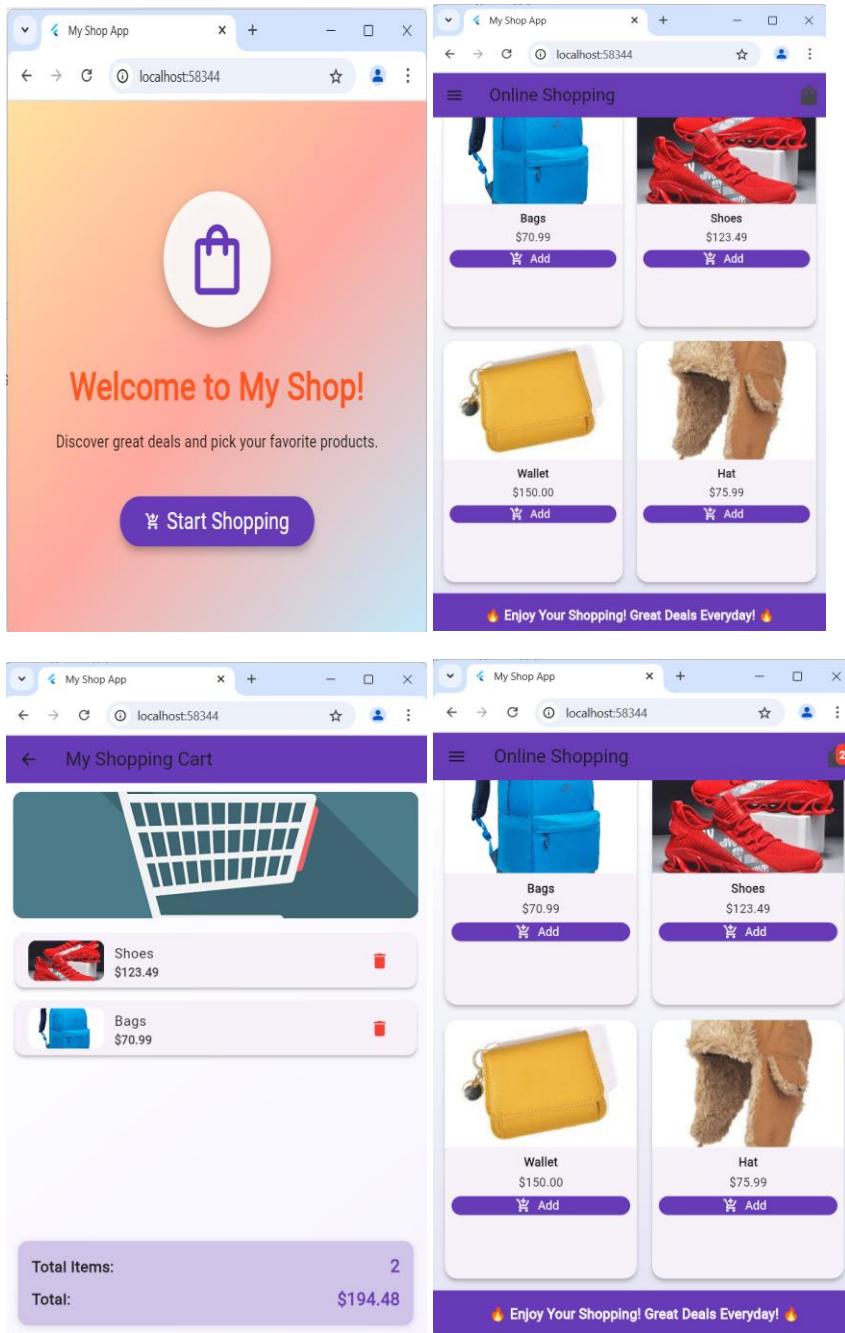
Hat \$75.99

Total Items: 2

Total: \$194.48

Enjoy Your Shopping! Great Deals Everyday!

## 13. Sample Output in Chrome Browser



---

## 14. GitHub Repository Name

[https://github.com/AyeKyiCho/mobile\\_fluuter\\_app\\_exercise1](https://github.com/AyeKyiCho/mobile_fluuter_app_exercise1)