

Overview

This project is a simple Shop App built with Flutter, allowing users to browse popular sports items, filter by categories, view item details, add to a shopping bag, and see the subtotal, tax, and total at checkout

Project Structure

```
lib/  
├── main.dart  
├── models/  
│   └── item.dart  
├── widgets/  
│   └── item_tile.dart  
└── screens/  
    ├── home_screen.dart  
    └── cart_screen.dart  
assets/  
└── images/  
    ├── soccer_ball.png  
    ├── basketball_ball.jpg  
    └── ... (other item images)  
pubspec.yaml
```

File Descriptions

lib/main.dart

Sets up the MaterialApp, defines the theme, and launches HomeScreen as the app's home

lib/models/item.dart

Defines the Item data model with fields:

name (String)

price (double)

assetImage (String path to local asset)

category (String for filtering)

lib/widgets/item_tile.dart

Renders each item in a nicely styled Container with:

Image from assets (via Image.asset)

Name (tappable to show full-size image in a dialog)

Price displayed as \ \$XX.XX

Add-to-cart IconButton

lib/screens/home_screen.dart

Manages the main shop UI:

Displays a ListView of either the top 5 “Popular” items or items filtered by category
Provides a horizontal ChoiceChip bar (Popular, Ball, Racket, Bat, Glove, Board) for filtering using ChoiceChip widgets
Shows the cart icon with item count badge in the bottom bar.

lib/screens/cart_screen.dart

Displays the shopping bag:

ListView of selected items with name, price, and a delete button.

Calculates and shows Subtotal, Tax (13%), and Total (with tax) dynamically

All product images (.png / .jpg) reside in assets/images/. Ensure pubspec.yaml includes:
flutter:

uses-material-design: true

assets:

- assets/images/

Running on a Physical Android Device

Enable Developer Options

On your device, open Settings → About phone and tap Build number seven times until you see “You are now a developer!”

Return to Settings, open System → Advanced → Developer options.

Allow USB Debugging

In Developer options, scroll to USB debugging and toggle it On.

Confirm the prompt “Allow USB debugging?” on your device.

Select USB Configuration → MIDI

Still in Developer options, find Networking (or Select USB configuration) and choose MIDI.

This routes your device’s USB port to act as a MIDI interface, which Android Studio and IntelliJ use to establish an ADB connection reliably

Connect Your Device

Use a data-capable USB cable to plug your phone into your computer.

When prompted, tap Allow on the “USB debugging” authorization dialog.

Run from IntelliJ IDEA

In IntelliJ’s Run/Debug configurations, select “No Device” or your project’s launch config; the IDE will now list your physical device in the target selector

Click the Run or Debug button to build and install the APK directly on your device