Tutorial: Building a Flutter E-Commerce App (Part 1: Setup, Landing Page, and Home Page)

Overview

In this tutorial series, you'll build a complete Flutter-based E-commerce app. We'll approach this step-by-step in waves, starting today with the basic project setup, landing page, and home page. I'll provide code snippets of each page, but the code itself won't be the complete version of the pages, which, if you are looking for that, can be found on my GitHub.

GitHub Link: https://github.com/ChristianSaenz/mini-project-comp375.git

---

Table of Contents:

Step 1: Setting Up Flutter in Android Studio

Before we dive into coding, let's ensure your environment is ready.

Prerequisites

- **Android Studio**: Download from [Android Studio Official Site]()
- **Flutter SDK**: Get the latest SDK from the [Flutter official website]()

Installing Flutter and Dart Plugins in Android Studio

1. Open **Android Studio**.
2. Navigate to `File > Settings` on Windows/Linux or `Android Studio > Preferences` on Mac.
3. Click on **Plugins**.
4. Search for and install:
   - **Flutter**
   - **Dart**

Restart Android Studio to apply changes.

Verify Flutter Installation

Open a terminal or command prompt and type:

flutter doctor

Resolve any issues indicated by `flutter doctor` output before proceeding. If the command `flutter doctor` doesn't work, it's likely because the Flutter SDK path is missing from your environmental variables. For troubleshooting and additional instructions on setting environmental variables, check [Flutter's official installation guide]().

Step 2: Create a New Flutter Project

1. In Android Studio, select `File > New > New Flutter Project`.
2. Choose `Flutter` as the project type.
3. Enter your project's name (e.g., `ecommerce_app`) and location.
4. Click **Finish**.

Step 3: Building the Landing Page

This app opens with a simple landing page featuring a logo, background, and navigation button to the home page.

Project Structure Setup

Inside `lib`, create:

- `pages` folder (landing and home pages will go here)
- `assets/img` folder within the `lib` folder for images
    - You can also host this outside the lib folder if you would like

Update `pubspec.yaml` to include:

- Any Images that you want to provide must be added to the assets like so

```
flutter:
  assets:
    - lib/assets/img/logo.png
    - lib/assets/img/background.png
```

Landing Page Code (`landing_page.dart`)

1. Importing Dependencies

```
import 'package:flutter/material.dart';
import 'package:timeless_toys/widgets/main_screen.dart';
```

Explanation:

- `flutter/material.dart` is the core library that provides widgets for building the UI.
- `main_screen.dart` is imported because the app will navigate to `MainScreen` after the user presses the "Get Started" button.

---

2. Defining the LandingPage Stateless Widget

```
class LandingPage extends StatelessWidget {
  const LandingPage({super.key});
```

Explanation:

- `LandingPage` is a **stateless widget** since it does not need to manage any internal state.
- The `super.key` helps in widget identity and optimizations.

---

3. Building the LandingPage UI

```
@override
 Widget build(BuildContext context) {
   return Scaffold(
     body: Stack(
       children: [
```

Explanation:

- `Scaffold` provides the basic structure of the screen.
- A `Stack` is used to layer multiple widgets on top of each other—important here because the background image needs to be behind the main content.

---

4. Adding the Background Image

```
          // Background Image covering the entire screen
          Container(
            decoration: const BoxDecoration(
              image: DecorationImage(
                image: AssetImage("assets/background.webp"),
                fit: BoxFit.cover, // Ensures image covers the whole screen
              ),
            ),
          ),
```

Explanation:

- A `Container` is used to set the **background image**.
- `BoxDecoration` applies an image that spans the full screen (`BoxFit.cover` ensures that it scales properly).

5. Placing the Logo in the Center

```
          // Column to center content on the screen
          Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Expanded(
                child: Center(
                  child: Column(
                    mainAxisSize: MainAxisSize.min, // Ensures column
takes only necessary space
                    children: [
                      Image.asset('assets/timeless_logo.jpg', height:
300), // Logo image
                    ],
                  ),
                ),
              ),
```

Explanation:

- A `Column` is used to **vertically align** elements on the screen.
- Expanded helps make sure content is centered properly.
- The `Image.asset()` widget displays the **logo image**.

---

6. Creating the "Get Started" Button

```
              // Padding around the button for spacing
              Padding(
                padding: const EdgeInsets.symmetric(horizontal: 20,
vertical: 40),
                child: SizedBox(
                  width: 250, // Button width
                  height: 50, // Button height
                  child: ElevatedButton(
                    onPressed: () {
                      // Navigates to the main screen and replaces the
current page
                      Navigator.pushReplacement(
                        context,
                        MaterialPageRoute(builder: (context) => const
MainScreen()),
                      );
                    },
```

Explanation:

- `Padding` ensures proper spacing around the button.
- `SizedBox` sets **fixed dimensions** for the button.
- `ElevatedButton` is used to **trigger navigation** to `MainScreen` when clicked.
- `Navigator.pushReplacement()` is used so that the user **cannot navigate back** to the landing page.

---

7. Styling the Button

```
                    style: ElevatedButton.styleFrom(
                      backgroundColor: Colors.blueAccent, // Button
color
                      shape: RoundedRectangleBorder(
                        borderRadius: BorderRadius.circular(10), //
Rounded corners for the button
                      ),
                    ),
                    child: const Text(
                      "Get Started", // Button text
                      style: TextStyle(
                        fontSize: 16,
                        fontWeight: FontWeight.bold,
                        color: Colors.white, // White text color for
contrast
                      ),
                    ),
                  ),
                ),
              ),
            ],
          ),
        ],
      ),
    );
  }
}
```

Explanation:

- The button is styled with:
  - A **blue accent color**.
  - **Rounded corners** for a modern look.
  - A **bold, white "Get Started" text**.

Understanding Key Widgets

Here's a breakdown of the primary widgets used in the landing page:

- `Scaffold`: Provides a basic layout structure for your page (app bars, floating action buttons, etc.).
- **Stack**: Allows you to overlay widgets. Useful for backgrounds or layered UI elements.
- **Container**: A versatile widget for styling (e.g., backgrounds, padding).
  - **DecorationImage**: For placing and fitting images as backgrounds.
- **Column**: Organizes widgets vertically.
- **Image.asset**: Loads images from your project's assets.
- **ElevatedButton**: A styled button for user interactions.
- **Navigator.pushReplacement**: This is used to navigate to a new page and remove the current page from the navigation stack.

Step 4: Building the HomePage

This is where the user will go after clicking Get Started. On the home page, the user can see the different types of items we have available.

Installing Required Packages

Our home page uses the **Carousel Slider** and **Smooth Page Indicator**. Add the following dependencies to your `pubspec.yaml` file:

dependencies:

  carousel_slider: ^4.2.1

  smooth_page_indicator: ^1.1.0

Run the command below in your terminal to fetch the dependencies:

- flutter pub get

Why We Use These Packages:

- **carousel_slider**: This package provides a simple way to create image sliders with auto-play, transitions, and customization.
- **smooth_page_indicator**: Helps add elegant page indicators to the carousel to improve the user experience.

Where to Find Flutter Packages:

If you're looking for additional Flutter packages, browse the official Dart Pub Repository. This website hosts thousands of open-source Flutter packages for UI components, networking, database management, animations, and more.

Home Page Code (`home_page.dart`)

1. Importing Dependencies

```
import 'package:flutter/material.dart';

import 'package:carousel_slider/carousel_slider.dart';

import 'package:smooth_page_indicator/smooth_page_indicator.dart';
```

Explanation:

- `flutter/material.dart`: Provides core UI components.
- `carousel_slider.dart`: Enables a carousel slider for displaying rotating images.
- `smooth_page_indicator.dart`: Provides a page indicator for the carousel.

---

2. Defining the HomePage Stateful Widget

```
class HomePage extends StatefulWidget {
  const HomePage({super.key});
  @override
  _HomePageState createState() => _HomePageState();
}
```

Explanation:

- `HomePage` is a **stateful widget** because it manages dynamic data like the active index of the carousel.
- `_HomePageState` is the associated state class that handles UI updates.

---

3. Managing State in `_HomePageState`

```
class _HomePageState extends State<HomePage> {
  int activeIndex = 0; // Tracks current image index in the carousel
  // List of images for the carousel
  final List<String> carouselImages = [
    "assets/banner1.webp",
    "assets/banner2.jpg",
    "assets/banner3.jpg",
 ];
```

Explanation:

- `activeIndex` tracks which carousel image is currently displayed.
- `carouselImages` is a list storing the file paths of the images used in the carousel.

---

4. Defining the List of Toy Items

```
// List of toy items for the grid display
 final List<Map<String, String>> toys = [
   {"name": "Teddy Bear", "image": "assets/teddy.webp"},
   {"name": "Race Car", "image": "assets/toy_car.jpg"},
   {"name": "Lego Set", "image": "assets/toy_lego.jpg"},
   {"name": "Doll", "image": "assets/toy_doll.jpg"},
   {"name": "Robot", "image": "assets/toy_robot.jpg"},
   {"name": "Train Set", "image": "assets/toy_train.webp"},
 ];
```

Explanation:

- `toys` is a **list of maps**, where each toy has a `name` and an `image` path.
- This list is used to populate the toy grid display.

---

5. Building the Home Page Layout

```
@override
 Widget build(BuildContext context) {
   return Scaffold(
     appBar: AppBar(
       backgroundColor: Colors.blueAccent,
       centerTitle: true,
       elevation: 4, // Adds shadow effect to the AppBar
```

Explanation:

- `Scaffold` provides the **basic layout** structure.
- `AppBar` acts as the top navigation bar, with a blue accent color and an elevation for a **shadow effect**.

---

6. Creating the AppBar with a Logo and Shopping Cart

```
       // Title section of the AppBar containing logo, title, and cart button
        title: Row(
          mainAxisAlignment: MainAxisAlignment.spaceBetween,
          children: [
            Image.asset(
              "assets/timeless_logo.jpg",
              height: 40, // Logo size
            ),
            const Text(
              "Timeless Toys",
              style: TextStyle(fontWeight: FontWeight.bold, fontSize: 18),
            ),
            IconButton(
              icon: const Icon(Icons.shopping_cart, size: 28, color: Colors.white),
              onPressed: () {
                // Placeholder action for shopping cart button
                ScaffoldMessenger.of(context).showSnackBar(
                  const SnackBar(content: Text("Shopping Cart 
Clicked")),
                );
              },
            ),
          ],
        ),
      ),
```

Explanation:

- Row is used to arrange the logo, title, and shopping cart **horizontally**.
- `IconButton` represents the **shopping cart** and shows a `SnackBar` when clicked.

---

7. Implementing the Carousel Slider

```
     body: Column(
        children: [
          // Carousel slider for banner images
          Stack(
            alignment: Alignment.bottomCenter,
            children: [
              CarouselSlider.builder(
                itemCount: carouselImages.length,
                },
                options: CarouselOptions(
                  height: 180, // Carousel height
                  autoPlay: true, // Enables auto-sliding
                  autoPlayInterval: const Duration(seconds: 3), // Duration per slide
                  enlargeCenterPage: true, // Applies zoom effect on active slide
                  viewportFraction: 0.9, // Determines visible portion of next image
                  onPageChanged: (index, reason) {
                    setState(() {
                      activeIndex = index;
                    });
                  },
                ),
                itemBuilder: (context, index, realIndex) {
                  return ClipRRect(
                    borderRadius: BorderRadius.circular(15),
                    child: Image.asset(
                      carouselImages[index],
                      width: double.infinity,
                      fit: BoxFit.cover, // Ensures full image coverage
                    ),
                  );
                },
            ),
```

Explanation:

- `CarouselSlider.builder()` is used to create a **sliding banner**.
- `autoPlay: true` makes the images transition automatically every 3 seconds.
- `onPageChanged()` updates the **active index** when a new image is displayed.

---

8. Adding a Smooth Page Indicator

```
            Positioned(
              bottom: 10, // Adjusts position of indicator
              child: AnimatedSmoothIndicator(
                activeIndex: activeIndex,
                count: carouselImages.length,
                effect: ExpandingDotsEffect(
                  dotWidth: 10,
                  dotHeight: 10,
                  activeDotColor: Colors.blue,
                  dotColor: Colors.grey.shade300,
                ),
              ),
            ),
          ],
        ),
```

Explanation:

- `AnimatedSmoothIndicator` adds **dot indicators** below the carousel.

- `ExpandingDotsEffect` makes the active dot expand, improving **user experience**.

---

9. Displaying the Toy Grid

```
         const SizedBox(height: 20), // Adds spacing between carousel and grid


          // Grid display for toy products
          Expanded(
            child: Padding(
              padding: const EdgeInsets.all(8.0),
              child: GridView.builder(
                gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(
                  crossAxisCount: 2, // Displays 2 items per row
                  crossAxisSpacing: 10, // Spacing between columns
                  mainAxisSpacing: 10, // Spacing between rows
                  childAspectRatio: 0.8, // Aspect ratio of grid items
                ),
                itemCount: toys.length,
```

Explanation:

- `GridView.builder()` dynamically generates a **grid layout** for toys.
- `crossAxisCount: 2` ensures **two items per row**.
- `childAspectRatio: 0.8` sets the **height-to-width ratio**.

---

10. Implementing Tap Functionality for Toy Cards

```
              itemBuilder: (context, index) {
                return GestureDetector(
                  onTap: () {
                    // Displays a snackbar when a toy is clicked
                    ScaffoldMessenger.of(context).showSnackBar(
                      SnackBar(content: Text("Clicked on
${toys[index]['name']}")),
                    );
                  },
                  child: Card(
                    elevation: 5, // Adds shadow effect to cards
                    shape: RoundedRectangleBorder(
                      borderRadius: BorderRadius.circular(10),
                    ),
                    child: Column(
                      children: [
                        Expanded(
                          child: ClipRRect(
                            borderRadius: BorderRadius.circular(10),
                            child: Image.asset(
                              toys[index]["image"]!,
                              width: double.infinity,
                              fit: BoxFit.cover, // Ensures full
coverage of image
```

```
                              ),
                            ),
                          ),
                          Padding(
                            padding: const EdgeInsets.all(8.0),
                            child: Text(
                              toys[index]["name"]!,
                              style: const TextStyle(fontSize: 16, fontWeight: FontWeight.bold),
                              textAlign: TextAlign.center,
                            ),
                          ),
                        ],
                      ),
                    ),
                  );
                },
```

Explanation:

- `GestureDetector` makes each toy **clickable**.
- `ScaffoldMessenger.showSnackBar()` **notifies the user** when a toy is clicked.

Key Concepts Used in the Home Page

- **Stateful Widget (`HomePage`)**: Allows dynamic changes, such as updating the carousel indicator.

- `super.key` **in Constructors**: Ensures widget uniqueness and proper state management when Flutter rebuilds widgets.
- **AppBar**: Provides a header with a title for the page.
- **CarouselSlider**: Enables a horizontally scrolling image carousel.
- **Stack**: Overlays elements, used here for placing the carousel and its indicator.
- **AnimatedSmoothIndicator**: Displays page indicators for the carousel.
- **GridView.builder**: Dynamically displays items in a structured grid layout.
- **GestureDetector**: Detects taps on the product cards.
- **SnackBar**: Provides instant feedback when an item is clicked.

Step 5 Building the NavBar:

The Main Screen (`main_screen.dart`) serves as the navigation hub for our Flutter E-Commerce app. Unlike the Landing Page and Home Page, which display content directly, this screen manages navigation between multiple pages using a Bottom Navigation Bar.

NavBar Code Overview

1. Importing Dependencies

```
import 'package:flutter/material.dart';

import 'package:timeless_toys/pages/account_page.dart';

import 'package:timeless_toys/pages/home_page.dart';

import 'package:timeless_toys/pages/shop_page.dart';
```

Explanation:

- The `flutter/material.dart` package provides the core widgets and themes needed for building a Flutter app.
- The other three imports (`account_page.dart`, `home_page.dart`, and `shop_page.dart`) are for different pages in the app that will be used in the main screen.

---

2. Defining the MainScreen Stateful Widget

```
class MainScreen extends StatefulWidget {
  const MainScreen({super.key});

  @override
  _MainScreenState createState() => _MainScreenState();
}
```

Explanation:

- `MainScreen` is a **stateful widget** because the UI needs to change dynamically when the user selects different tabs.
- `createState()` links it to the `_MainScreenState` class, which manages the state of the widget.

---

3. Managing State with `_MainScreenState`

```
class _MainScreenState extends State<MainScreen> {
  int _selectedIndex = 0; // Tracks the selected tab index

  // List of pages (screens) displayed in the main screen
  final List<Widget> _screens = [
    HomePage(),   // Home Page screen
    ShopPage(),   // Shop Page screen
    AccountPage() // Account Page screen
  ];
```

Explanation:

- `_selectedIndex` is used to track which tab is currently active.

- `_screens` is a list containing the pages that correspond to each tab in the bottom navigation bar.

---

4. Handling Tab Selection

```
// Updates the selected tab index when a tab is tapped
 void _onItemTapped(int index) {
   setState(() {
     _selectedIndex = index;
   });
 }
```

Explanation:

- `_onItemTapped()` is called when the user taps a tab in the bottom navigation bar.
- `setState()` updates `_selectedIndex`, triggering a UI rebuild to display the selected page.

---

5. Building the Main Screen UI

```
@override
 Widget build(BuildContext context) {
   return Scaffold(
     // Displays the currently selected screen
     body: _screens[_selectedIndex],
```

Explanation:

- `Scaffold` is the base structure for the screen.
- body dynamically displays the selected page based on `_selectedIndex`.

---

6. Implementing the Bottom Navigation Bar

```
     // Bottom Navigation Bar for switching between pages
      bottomNavigationBar: BottomNavigationBar(
        currentIndex: _selectedIndex, // Highlights the selected tab
        onTap: _onItemTapped, // Handles tab selection
        backgroundColor: Colors.white, // Background color of the bar
        selectedItemColor: Colors.blueAccent, // Color of the selected
tab icon and label
        unselectedItemColor: Colors.grey, // Color of unselected tab
icons and labels
        items: const [
          BottomNavigationBarItem(
            icon: Icon(Icons.home), // Home tab icon
            label: 'Home', // Home tab label
          ),
          BottomNavigationBarItem(
            icon: Icon(Icons.shopping_cart), // Shop tab icon
            label: 'Shop', // Shop tab label
          ),
          BottomNavigationBarItem(
            icon: Icon(Icons.person), // Account tab icon
            label: 'Account', // Account tab label
          ),
        ],
      ),
    );
  }
}
```

Explanation:

- `BottomNavigationBar` provides a navigation menu at the bottom of the screen.
- `currentIndex` highlights the selected tab.
- `onTap` calls `_onItemTapped()` to update the selected tab.
- `items` defines the icons and labels for each tab.

How the Main Screen Works Compared to Other Screens?

- The Main Screen is a StatefulWidget because it needs to track which tab is selected and update the UI accordingly.
- Unlike the Landing Page, which only serves as an introduction, or the Home Page, which displays content, the Main Screen provides persistent navigation, allowing users to switch between different sections of the app seamlessly.

Logic Behind the Code

1. **Tracking the Active Tab:**
   - The `_selectedIndex` variable stores the currently selected tab.
   - It starts at `0`, meaning the Home Page is displayed initially.
2. **Handling Navigation:**
   - `_screens` is a list storing the three main pages (Home, Shop, Account).
   - The `Scaffold` widget sets `body` to `_screens[_selectedIndex]`, meaning it displays whichever page is currently selected.
3. **Bottom Navigation Bar Logic:**
   - The `BottomNavigationBar` widget provides a menu with three tabs.
   - `currentIndex: _selectedIndex` ensures that the correct tab is highlighted.
   - When a tab is tapped, `_onItemTapped` updates `_selectedIndex` using `setState()`, causing the UI to rebuild with the new page.
4. **Stateful Behavior:**

    - Since `_selectedIndex` is stored in the widget’s state, switching between tabs doesn't rebuild the entire app.
    - The `setState()` function ensures that only the relevant parts of the UI update.

Key Concepts Used in the Main Screen

- **Stateful Widget (`MainScreen`):** Allows dynamic tab selection.
- BottomNavigationBar: Provides persistent navigation at the bottom of the screen.
- **List of Screens (`_screens`):** Stores references to different app sections.
- **setState():** Updates the selected index when a tab is tapped, triggering a UI update.
- **Scaffold:** Wraps the page and includes a navigation bar.

Step 6 Building Main Page:

In this section, we'll focus on the **main entry point** of our Flutter E-Commerce app. The `main.dart` file is responsible for initializing the app and defining the app-wide theme and navigation structure.

Main Code (`main.dart`)

```
import 'package:flutter/material.dart';
import 'package:timeless_toys/pages/landing_page.dart';
import 'package:timeless_toys/widgets/main_screen.dart';

void main() {
  runApp(const MyApp()); // Starts the Flutter application
}

class MyApp extends StatelessWidget {
  // The main application widget that acts as the root of the app.
  // 'super.key' ensures widget uniqueness and helps maintain state during hot reloads.
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo', // App title
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple), // Defines the app-wide color scheme
        useMaterial3: true, // Enables Material 3 design system
      ),
      home: const LandingPage(), // Sets the initial screen when the app loads
    );
  }
}
```

Key Concepts Used in the Main Page

- **`void main()`**: This is the entry point of the app. The `runApp()` function launches the `MyApp` widget.
- **Stateless Widget (`MyApp`)**: Represents the root widget of the application. It doesn't change state once built.
- **`super.key` in Constructors**: Helps Flutter track widget state properly, especially during hot reloads.
- **`MaterialApp`**: The core widget that wraps the entire application and provides app-wide configurations.
- **`title`**: The name of the application, used in debugging tools.
- **`ThemeData`**: Defines the overall theme and styling of the app.
- **`home: LandingPage()`**: Specifies the first screen to load when the app starts.

Appendix

Landing Page: (Full Code)

```
import 'package:flutter/material.dart';

import 'package:timeless_toys/widgets/main_screen.dart';


class LandingPage extends StatelessWidget {
```

```
const LandingPage({super.key});

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: Stack(
      children: [
        // Background Image covering the entire screen
        Container(
          decoration: const BoxDecoration(
            image: DecorationImage(
              image: AssetImage("assets/background.webp"),
              fit: BoxFit.cover, // Ensures image covers the whole screen
            ),
          ),
        ),

        // Column to center content on the screen
        Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Expanded(
              child: Center(
                child: Column(
      mainAxisSize: MainAxisSize.min, // Ensures column takes only necessary space
      children: [
        Image.asset('assets/timeless_logo.jpg', height: 300), // Logo image
      ],
    ),
  ),
),

// Padding around the button for spacing
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 20, vertical: 40),
  child: SizedBox(
    width: 250, // Button width
    height: 50, // Button height
    child: ElevatedButton(
      onPressed: () {
        // Navigates to the main screen and replaces the current page
        Navigator.pushReplacement(
          context,
          MaterialPageRoute(builder: (context) => const MainScreen()),
        );
      },
      style: ElevatedButton.styleFrom(
        backgroundColor: Colors.blueAccent, // Button color
```

```
                      shape: RoundedRectangleBorder(
                        borderRadius: BorderRadius.circular(10), // Rounded corners for the button
                      ),
                    ),
                    child: const Text(
                      "Get Started", // Button text
                      style: TextStyle(
                        fontSize: 16,
                        fontWeight: FontWeight.bold,
                        color: Colors.white, // White text color for contrast
                      ),
                    ),
                  ),
                ),
              ),
            ],
          ),
        ],
      ),
    );
  }
}
```

Home Page: (Full Code)

```
import 'package:flutter/material.dart';

import 'package:carousel_slider/carousel_slider.dart';

import 'package:smooth_page_indicator/smooth_page_indicator.dart';


class HomePage extends StatefulWidget {

  const HomePage({super.key});


  @override

  _HomePageState createState() => _HomePageState();

}


class _HomePageState extends State<HomePage> {

  int activeIndex = 0; // Tracks current image index in the carousel


  // List of images for the carousel

  final List<String> carouselImages = [

    "assets/banner1.webp",

    "assets/banner2.jpg",

    "assets/banner3.jpg",

  ];
```

```
// List of toy items for the grid display
final List<Map<String, String>> toys = [
  {"name": "Teddy Bear", "image": "assets/teddy.webp"},
  {"name": "Race Car", "image": "assets/toy_car.jpg"},
  {"name": "Lego Set", "image": "assets/toy_lego.jpg"},
  {"name": "Doll", "image": "assets/toy_doll.jpg"},
  {"name": "Robot", "image": "assets/toy_robot.jpg"},
  {"name": "Train Set", "image": "assets/toy_train.webp"},
];

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      backgroundColor: Colors.blueAccent,
      centerTitle: true,
      elevation: 4, // Adds shadow effect to the AppBar

      // Title section of the AppBar containing logo, title, and cart button
      title: Row(
        mainAxisAlignment: MainAxisAlignment.spaceBetween,
        children: [
          Image.asset(
            "assets/timeless_logo.jpg",
        height: 40, // Logo size
      ),
      const Text(
        "Timeless Toys",
        style: TextStyle(fontWeight: FontWeight.bold, fontSize: 18),
      ),
      IconButton(
        icon: const Icon(Icons.shopping_cart, size: 28, color: Colors.white),
        onPressed: () {
          // Placeholder action for shopping cart button
          ScaffoldMessenger.of(context).showSnackBar(
            const SnackBar(content: Text("Shopping Cart Clicked")),
          );
        },
      ),
    ],
  ),
),

body: Column(
  children: [
    // Carousel slider for banner images
    Stack(
      alignment: Alignment.bottomCenter,
children: [
  CarouselSlider.builder(
    itemCount: carouselImages.length,
    options: CarouselOptions(
      height: 180, // Carousel height
      autoPlay: true, // Enables auto-sliding
      autoPlayInterval: const Duration(seconds: 3), // Duration per slide
      enlargeCenterPage: true, // Applies zoom effect on active slide
      viewportFraction: 0.9, // Determines visible portion of next image
      onPageChanged: (index, reason) {
        setState(() {
          activeIndex = index;
        });
      },
    ),
    itemBuilder: (context, index, realIndex) {
      return ClipRRect(
        borderRadius: BorderRadius.circular(15),
        child: Image.asset(
          carouselImages[index],
          width: double.infinity,
          fit: BoxFit.cover, // Ensures full image coverage
        ),
      );
      },
    ),
    Positioned(
      bottom: 10, // Adjusts position of indicator
      child: AnimatedSmoothIndicator(
        activeIndex: activeIndex,
        count: carouselImages.length,
        effect: ExpandingDotsEffect(
          dotWidth: 10,
          dotHeight: 10,
          activeDotColor: Colors.blue,
          dotColor: Colors.grey.shade300,
        ),
      ),
    ),
  ],
),

const SizedBox(height: 20), // Adds spacing between carousel and grid

// Grid display for toy products
Expanded(
  child: Padding(
    padding: const EdgeInsets.all(8.0),
child: GridView.builder(
  gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(
    crossAxisCount: 2, // Displays 2 items per row
    crossAxisSpacing: 10, // Spacing between columns
    mainAxisSpacing: 10, // Spacing between rows
    childAspectRatio: 0.8, // Aspect ratio of grid items
  ),
  itemCount: toys.length,
  itemBuilder: (context, index) {
    return GestureDetector(
      onTap: () {
        // Displays a snackbar when a toy is clicked
        ScaffoldMessenger.of(context).showSnackBar(
          SnackBar(content: Text("Clicked on ${toys[index]['name']}")),
        );
      },
      child: Card(
        elevation: 5, // Adds shadow effect to cards
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(10),
        ),
        child: Column(
          children: [
            Expanded(
                child: ClipRRect(
                  borderRadius: BorderRadius.circular(10),
                  child: Image.asset(
                    toys[index]["image"]!,
                    width: double.infinity,
                    fit: BoxFit.cover, // Ensures full coverage of image
                  ),
                ),
              ),
              Padding(
                padding: const EdgeInsets.all(8.0),
                child: Text(
                  toys[index]["name"]!,
                  style: const TextStyle(fontSize: 16, fontWeight: FontWeight.bold),
                  textAlign: TextAlign.center,
                ),
              ),
            ],
          ),
        ),
      );
    },
  ),
),
          ),
        ],
      ),
    );
  }
}
```

Navbar: (Full Code)

```
import 'package:flutter/material.dart';
import 'package:timeless_toys/pages/account_page.dart';
import 'package:timeless_toys/pages/home_page.dart';
import 'package:timeless_toys/pages/shop_page.dart';

class MainScreen extends StatefulWidget {
  const MainScreen({super.key});

  @override
  _MainScreenState createState() => _MainScreenState();
}

class _MainScreenState extends State<MainScreen> {
  int _selectedIndex = 0; // Tracks the selected tab index

  // List of pages (screens) displayed in the main screen
```

```
final List<Widget> _screens = [
  HomePage(),  // Home Page screen
  ShopPage(),  // Shop Page screen
  AccountPage() // Account Page screen
];

// Updates the selected tab index when a tab is tapped
void _onItemTapped(int index) {
  setState(() {
    _selectedIndex = index;
  });
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    // Displays the currently selected screen
    body: _screens[_selectedIndex],

    // Bottom Navigation Bar for switching between pages
    bottomNavigationBar: BottomNavigationBar(
      currentIndex: _selectedIndex, // Highlights the selected tab
      onTap: _onItemTapped, // Handles tab selection
      backgroundColor: Colors.white, // Background color of the bar
    selectedItemColor: Colors.blueAccent, // Color of the selected tab icon and label
    unselectedItemColor: Colors.grey, // Color of unselected tab icons and labels
    items: const [
     BottomNavigationBarItem(
      icon: Icon(Icons.home), // Home tab icon
      label: 'Home', // Home tab label
     ),
     BottomNavigationBarItem(
      icon: Icon(Icons.shopping_cart), // Shop tab icon
      label: 'Shop', // Shop tab label
     ),
     BottomNavigationBarItem(
      icon: Icon(Icons.person), // Account tab icon
      label: 'Account', // Account tab label
     ),
    ],
   ),
  );
 }
}
```

Main.dart: (Full Code)

```
import 'package:flutter/material.dart';
import 'package:timeless_toys/pages/landing_page.dart';
import 'package:timeless_toys/widgets/main_screen.dart';
```

```
// Entry point of the Flutter application
void main() {
  runApp(const MyApp()); // Launches the application
}


class MyApp extends StatelessWidget {
  const MyApp({super.key});


  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo', // Application title
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple), // Sets primary theme color
        useMaterial3: true, // Enables Material 3 design elements
      ),
      home: const LandingPage(), // Sets the initial screen of the app
    );
  }}
```