E-commerce Project Structure Plan: Website & Admin Panel

This document outlines a structured approach for your Flutter e-commerce application, separating the customer-facing "Website" from the "Admin Panel" and integrating Navigator 2.0 (go_router) for efficient navigation. We'll also account for your theme.dart and cwp.dart files being outside the lib directory.

1. Top-Level Project Directory Structure

Let's start with how your main project folder will be organized.

```
your_ecommerce_project/
├── .gitignore
├── pubspec.yaml
├── pubspec.lock
├── README.md
├── analysis_options.yaml
├── lib/                  # All your Flutter application code
│   ├── main.dart          # Main entry point of the application
│   ├── app.dart           # Where MaterialApp.router and GoRouter config lives
│   │
│   ├── common/            # Widgets, utilities, and services shared by BOTH website &
panel
│   │   ├── widgets/        # Reusable UI components (e.g., CustomButton, AppCard)
│   │   ├── models/         # Data models (e.g., Product, User, Order)
│   │   ├── services/       # API calls, authentication, database interactions
│   │   ├── utils/         # Generic helper functions (e.g., date formatters, validators)
│   │   └── constants/      # Global constants (e.g., API_BASE_URL, app names)
│   │
│   ├── website/           # Customer-facing website specific code
│   │   ├── pages/          # All top-level pages of the website
│   │   │   ├── home_page.dart
│   │   │   ├── products_list_page.dart
│   │   │   ├── product_detail_page.dart
│   │   │   ├── cart_page.dart
│   │   │   ├── checkout_page.dart
│   │   │   └── user_profile_page.dart
│   │   │
│   │   └── widgets/        # Widgets specific to the website (e.g., ProductCard for
website)
```

```
│   │
│   └── panel/            # Admin panel specific code
│       ├── pages/         # All top-level pages of the admin panel
│       │   ├── dashboard_page.dart
│       │   ├── orders_management_page.dart
│       │   ├── product_editor_page.dart
│       │   ├── user_management_page.dart
│       │   └── settings_page.dart # Panel settings
│       │
│       └── widgets/       # Widgets specific to the admin panel (e.g., AdminTable)
│
├── assets/               # Static assets like images, fonts, icons
│   ├── images/
│   ├── fonts/
│   └── icons/
│
├── web/                  # Web-specific files (index.html, manifest.json)
├── android/              # Android platform-specific code
├── ios/                  # iOS platform-specific code
├── windows/              # Windows platform-specific code
├── linux/                # Linux platform-specific code
├── macos/                # macOS platform-specific code
│
├── theme.dart            # Your custom theme definitions (OUTSIDE lib)
├── cwp.dart              # Your custom responsive widgets and other utilities
(OUTSIDE lib)
```

Key Points on Top-Level Structure:

- **lib/**: The standard location for all your Dart source code.
- **theme.dart & cwp.dart**: Explicitly placed *outside* lib as per your request. This means their import paths will be relative, e.g., import 'package:ecommerce_website/../theme.dart';. While unusual for package-based imports, we will accommodate this.

2. lib Folder Structure (Detailed)

2.1 main.dart

This file will be minimal. Its primary job is to kickstart your Flutter app and delegate to app.dart.

```
// lib/main.dart
import 'package:flutter/material.dart';
import 'package:ecommerce_website/app.dart'; // Import your main app widget

void main() {
  runApp(const MyApp());
}

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

  @override
  Widget build(BuildContext context) {
    // This is the app entry point. All router, theme setup will be in app.dart
    return const AppRoot();
  }
}
```

2.2 app.dart (GoRouter & Theme Configuration)

This file will contain the core MaterialApp.router setup, including the GoRouter configuration and your dynamic theme integration.

```
// lib/app.dart
import 'package:flutter/material.dart';
import 'package:go_router/go_router.dart';
import 'package:ecommerce_website/../theme.dart'; // Relative path to your
theme.dart
import 'package:ecommerce_website/../cwp.dart';  // Relative path to your cwp.dart
(for responsive widgets)

// Import your page files
import 'package:ecommerce_website/lib/website/pages/home_page.dart';
import 'package:ecommerce_website/lib/website/pages/products_list_page.dart';
import 'package:ecommerce_website/lib/website/pages/product_detail_page.dart';
import 'package:ecommerce_website/lib/website/pages/cart_page.dart';
import 'package:ecommerce_website/lib/website/pages/checkout_page.dart';
import 'package:ecommerce_website/lib/website/pages/user_profile_page.dart';

import 'package:ecommerce_website/lib/panel/pages/dashboard_page.dart';
import
'package:ecommerce_website/lib/panel/pages/orders_management_page.dart';
import 'package:ecommerce_website/lib/panel/pages/product_editor_page.dart';
import 'package:ecommerce_website/lib/panel/pages/user_management_page.dart';
import 'package:ecommerce_website/lib/panel/pages/settings_page.dart';

// A placeholder for an unknown route page
class UnknownRoutePage extends StatelessWidget {
  const UnknownRoutePage({super.key});
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text('Page Not Found', style:
context.headlineSmall?.copyWith(color: context.primaryText))),
      body: Center(
        child: Text(
          '404 - Oopsy! The page you are looking for does not exist.',
          style: context.displaySmall?.copyWith(color: context.errorColor),
          textAlign: TextAlign.center,
        ),
      ),
    );
  }
}


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

  @override
  State<AppRoot> createState() => _AppRootState();
}

class _AppRootState extends State<AppRoot> {
  // GoRouter configuration
  late final GoRouter _router = GoRouter(
    initialLocation: '/', // Default starting route
    routes: [
      // --- Website Routes ---
      GoRoute(
        path: '/',
        builder: (context, state) => const HomePage(),
      ),
      GoRoute(
        path: '/products',
        builder: (context, state) => const ProductsListPage(),
        routes: [
          GoRoute(
            path: ':productId', // Nested route for product detail
            builder: (context, state) => ProductDetailPage(
              productId: state.pathParameters['productId']!,
            ),
          ),
        ],
      ),
      GoRoute(
        path: '/cart',
        builder: (context, state) => const CartPage(),
      ),
      GoRoute(
        path: '/checkout',
        builder: (context, state) => const CheckoutPage(),
      ),
      GoRoute(
        path: '/profile',
        builder: (context, state) => const UserProfilePage(),
      ),

      // --- Admin Panel Routes (Nested under /panel) ---
      GoRoute(
        path: '/panel',
        builder: (context, state) => const DashboardPage(), // Default panel page
        routes: [
          GoRoute(
            path: 'orders',
            builder: (context, state) => const OrdersManagementPage(),
          ),
          GoRoute(
            path: 'products', // Admin product management
            builder: (context, state) => const ProductEditorPage(),
          ),
          GoRoute(
            path: 'users',
            builder: (context, state) => const UserManagementPage(),
          ),
          GoRoute(
            path: 'settings', // Admin settings
            builder: (context, state) => const PanelSettingsPage(),
          ),
        ],
      ),
    ],
    // Error handling for unknown routes
    errorBuilder: (context, state) => const UnknownRoutePage(),
   );

   @override
   Widget build(BuildContext context) {
    final Brightness platformBrightness = MediaQuery.platformBrightnessOf(context);

    return MaterialApp.router(
      title: 'E-commerce App',
      // Apply your dynamic theme
      theme: buildAppTheme(platformBrightness),
      debugShowCheckedModeBanner: false,
      routerConfig: _router, // Assign the GoRouter instance
    );
   }
  }
```

2.3 common/ Folder

This is for code that can be used by both your website and admin panel.

- **common/widgets/**:
    - custom_button.dart
    - app_card.dart
    - loading_indicator.dart
    - responsive_image.dart
    - empty_state_widget.dart
- **common/models/**:
    - product.dart (The Product class you previously used)
    - user.dart
    - order.dart
    - category.dart
- **common/services/**:
    - auth_service.dart (Login, Logout, User registration)
    - api_service.dart (Handles all API calls)
    - product_service.dart (Methods for fetching/updating products)
    - order_service.dart
- **common/utils/**:
    - app_helpers.dart (e.g., formatCurrency, validateEmail)
    - logger.dart (Simple logging utility)
- **common/constants/**:
    - app_constants.dart (e.g., APP_NAME, API_BASE_URL, PRODUCT_CATEGORIES)
    - app_colors_constants.dart (If you have static color constants not in theme.dart)

2.4 website/ Folder

Code specific to the customer-facing part of your e-commerce site.

- **website/pages/**:
    - home_page.dart: The main landing page.
    - products_list_page.dart: Displays all products.
    - product_detail_page.dart: Shows details for a single product (receives productId from router).
    - cart_page.dart: User's shopping cart.
    - checkout_page.dart: Checkout process.
    - user_profile_page.dart: User account details.
- **website/widgets/**:
    - website_app_bar.dart (Custom app bar for website)
    - product_grid_item.dart (Specific product display for grid on website)
    - category_filter_widget.dart

    - product_carousel.dart (If you abstract your PageView logic further)

2.5 panel/ Folder

Code specific to the admin dashboard.

- **panel/pages/**:
    - dashboard_page.dart: Admin dashboard overview.
    - orders_management_page.dart: Manage customer orders.
    - product_editor_page.dart: Add/edit products.
    - user_management_page.dart: Manage users.
    - settings_page.dart: Admin panel specific settings.
- **panel/widgets/**:
    - admin_sidebar.dart
    - data_table_widget.dart (Reusable table for admin data)
    - panel_header.dart

3. Navigation Strategy (go_router)

We are using go_router which simplifies Navigator 2.0 immensely.

- **Centralized Routes**: All your routes are defined in one place (lib/app.dart) within the GoRouter instance.
- **Nested Routes**: Notice how /products/:productId is nested under /products, and admin routes like /panel/orders are nested under /panel. This makes navigation hierarchy clear.
- **Navigation Calls**:
    - **context.go('/path/to/page')**: To navigate to a new route. This replaces Navigator.push and automatically manages the stack.
    - **context.pop()**: To go back to the previous route.
    - **context.goNamed('routeName', pathParameters: {'id': '123'})**: For more type-safe and readable navigation (requires naming your routes in GoRouter).

Example Page Implementation (e.g., home_page.dart)

Each page will be a standard StatelessWidget or StatefulWidget, and they will use context.go() for navigation.

```
// lib/website/pages/home_page.dart
import 'package:flutter/material.dart';
import 'package:go_router/go_router.dart';
import 'package:ecommerce_website/../theme.dart'; // Import theme
import 'package:ecommerce_website/../cwp.dart';  // Import custom widgets
```

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Welcome', style: context.headlineSmall?.copyWith(color:
context.primaryText)),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Text('Welcome to the E-commerce Website!', style:
context.bodyLarge?.copyWith(color: context.primaryText)),
            const SizedBox(height: 20),
            ElevatedButton(
              onPressed: () {
                context.go('/products'); // Navigate to products list
              },
              child: Text('View Products', style: context.labelLarge),
            ),
            const SizedBox(height: 10),
            ElevatedButton(
              onPressed: () {
                context.go('/panel'); // Navigate to admin panel
              },
              child: Text('Go to Admin Panel', style: context.labelLarge),
            ),
          ],
        ),
      ),
    );
  }
}
```

4. Integration of theme.dart and cwp.dart

Since theme.dart and cwp.dart are *outside* the lib folder, your imports will need to reflect this relative path.

- **Example Import in app.dart**:
  import 'package:ecommerce_website/../theme.dart';
  import 'package:ecommerce_website/../cwp.dart';

  This tells Flutter to go up one directory from lib/ (to your_ecommerce_project/) and then find theme.dart or cwp.dart.
- **Example Import in other lib files (e.g., home_page.dart)**:
  import 'package:ecommerce_website/../theme.dart';
  import 'package:ecommerce_website/../cwp.dart';

  This import path will be consistent across all your Dart files within the lib folder.

What to Build Next

1. **Set up pubspec.yaml**: Add go_router to your dependencies and run flutter pub get.
2. **Create Placeholders**: Create empty .dart files for each page and widget as per the structure.
3. **Implement app.dart**: Copy the AppRoot code into lib/app.dart.
4. **Update main.dart**: Make it minimal, just calling AppRoot.
5. **Start Building Pages**: Begin implementing the UI for your HomePage, ProductsListPage, etc., using your themed widgets and context.go() for navigation.

This detailed plan should give you a clear roadmap and prevent a lot of time-wasting as you build your robust e-commerce application. Let me know when you're ready to dive into the next step!