

E-Commerce Admin Panel

A full-featured, responsive **Flutter Web Admin Dashboard** for managing an e-commerce platform. Built with **Flutter**, **GetX**, and **Supabase**, this admin panel provides real-time data management for products, orders, customers, categories, brands, banners, and more.

Table of Contents

- [Overview](#)
 - [Features](#)
 - [Tech Stack](#)
 - [Architecture](#)
 - [Project Structure](#)
 - [Getting Started](#)
 - [Environment Variables](#)
 - [Key Modules](#)
 - [State Management](#)
 - [Database Schema](#)
 - [Responsive Design](#)
 - [Security](#)
 - [PDF & QR Code Generation](#)
 - [Contributing](#)
 - [License](#)
-

Overview

This admin panel serves as the backend management interface for a complete e-commerce ecosystem. It connects to a **Supabase** backend and provides real-time CRUD operations across all business entities. The dashboard features live order tracking, weekly sales analytics, role-based access control, and a comprehensive media management system.

Target Platform: Flutter Web (also supports Android, iOS, macOS, Windows, Linux)

Features

Dashboard & Analytics

- Real-time order statistics with pie chart visualization
- Weekly sales bar chart with current/previous week toggle
- Live order stream via Supabase real-time subscriptions
- Order status breakdown (Placed, Processing, Shipped, Delivered, Cancelled)
- Paginated order data tables with search and sorting

Product Management

- Create, edit, and delete products
- Support for **single** and **variable** product types
- Product attributes and variations management
- Thumbnail and additional image uploads via media manager
- Brand and category assignment with multi-select
- Product visibility controls (Published/Hidden)
- Tab and offer assignment

Order Management

- View all orders with real-time updates
- Edit order status and details
- Order item breakdown with pricing
- Shipping address management
- PDF parcel label generation with QR codes
- Payment method tracking (PayHere, PayPal)
- Transaction fee calculation

Category Management

- Hierarchical category creation
- Category-product relationship linking
- Featured category support with image upload

Brand Management

- Brand CRUD with logo upload
- Brand-product association

- Featured brand designation

Banner Management

- Promotional banner creation
- Banner image upload and targeting
- Active/inactive banner toggle

Customer Management

- Customer profiles with contact details
- Shipping address management
- Customer activity tracking

Tab Management

- Custom navigation tab creation
- Tab-product association
- Active/inactive tab control

Media Management

- Centralized media library with folder organization
- Drag-and-drop file upload via Flutter Dropzone
- Image preview with metadata (name, URL, size)
- Copy URL to clipboard
- Delete media files
- Organized by category: Banners, Brands, Categories, Products, Users

Settings & Administration

- **General Settings:** App configuration and preferences
- **Payment Settings:** Payment gateway configuration
- **Email Settings:** Email template management
- **User Management:** Add/remove admin users
- **Role-Based Access Control:** Set permissions per role (Super Admin, Store Admin, Product Manager, Order Manager, Marketing Manager)
- **Change Password:** Secure password updates
- **User Activity Log:** Track admin actions
- **Database Management:** Database connectivity settings
- **Product Settings:** Shipping, delivery times, processing fees, reviews & ratings

Authentication

- Supabase Auth with PKCE flow
 - Login with email/password
 - Forgot password flow
 - Reset password
 - Session persistence via GetStorage
 - Route-based authentication guard
-

Tech Stack

Technology	Purpose
Flutter 3.x	Cross-platform UI framework
Dart 3.7+	Programming language
GetX	State management, routing, dependency injection
Supabase	Backend-as-a-Service (Auth, Database, Storage, Realtime)
GetStorage	Local key-value storage
fl_chart	Dashboard charts (pie charts, bar charts)
data_table_2	Advanced data tables with pagination
pdf + printing	PDF generation and printing
qr_flutter	QR code generation
file_picker	File selection
flutter_dropzone	Drag-and-drop file uploads
lottie	Loading and empty state animations
intl	Date and currency formatting
flutter_colorpicker	Color selection for product attributes
flutter_typeahead	Autocomplete search fields
multi_select_flutter	Multi-select dropdowns
clipboard	Copy-to-clipboard functionality
uuid	Unique ID generation

Architecture

The project follows a **feature-first** architecture with clear separation of concerns:

```

lib/
+-- main.dart           # App entry point, dependency injection
+-- app.dart           # Root widget, theme & routing config
+-- screens/           # Feature modules (UI layer)
|   +-- dashboard/    # Dashboard with analytics
|   +-- products/     # Product CRUD
|   +-- orders/       # Order management
|   +-- category/     # Category management
|   +-- brands/       # Brand management
|   +-- banners/      # Banner management
|   +-- customers/    # Customer management
|   +-- tabs/         # Tab management
|   +-- media/        # Media library
|   +-- settings/     # App settings & admin
|   +-- login/        # Authentication
|   +-- forgot-password/ # Password recovery
|   +-- reset-password/ # Password reset
|   +-- layouts/      # Layout templates & headers
+-- common/           # Shared widgets
|   +-- widgets/      # Reusable UI components
|   +-- responsive/   # Responsive layout wrappers
+-- controllers/      # Global controllers
+-- util/             # Utilities & configuration
    +-- constants/    # App constants, colors, sizes, strings
    +-- formatters/   # Data formatters, enums, PDF creator
    +-- helpers/      # Helper functions
    +-- models/       # Data models
    +-- routes/       # Route definitions & middleware
    +-- theme/        # Theme configuration
    +-- device/       # Device utilities

```

Design Patterns Used

- **Repository Pattern** - Data access abstraction (e.g., ProductRepository)
- **Controller Pattern** - Business logic separation via GetX controllers
- **Observer Pattern** - Real-time data with Supabase streams
- **Singleton Pattern** - Service instances (e.g., JsonService)
- **Factory Pattern** - Model deserialization (fromJson factories)

Project Structure

Screens (Feature Modules)

Each feature module follows a consistent structure:

```

feature_name/
+-- feature_screen.dart # Main screen with responsive wrapper
+-- responsive_design/  # Platform-specific layouts
|   +-- feature_desktop.dart
|   +-- feature_tablet.dart
|   +-- feature_mobile.dart
+-- widgets/           # Feature-specific widgets
+-- controller/       # Feature controllers
+-- table/ or data/    # Data table sources

```

Common Widgets

Widget	Description
TRoundedContainer	Flexible container with border radius, shadow, and color
TRoundedImage	Image widget with Hero animation and placeholder
TCircularImage	Circular avatar-style image
TSectionHeading	Section header with optional action button
TChoiceChips	Color/text choice chips for attributes
TTabActionButton	Row of view/edit/delete action icons
TDialogs	Confirmation dialog utility
TFullScreenLoader	Loading overlay with progress indicator
ImageUploader	Image upload widget with preview
TSidebar	Navigation sidebar with role-based menu items

Data Models

Model	Description
ProductModel	Product with attributes, variations, pricing
ProductVariationsModel	Product variation with stock, price, images
ProductAttributesModel	Key-value product attributes
ProductCategoryModel	Product-category relationship
CategoryModel	Product category
BrandModel	Product brand
BannerModel	Promotional banner
OrderModel	Customer order
CustomerModel	Customer profile
AddressModel	Shipping/billing address
TabModel	Navigation tab
ImageModel	Media library image
AppConfigModel	Application configuration

Getting Started

Prerequisites

- Flutter SDK 3.7+ installed

- Dart SDK 3.7.2+
- A Supabase project (free tier works)
- Chrome browser (for web development)

Installation

1. Clone the repository:

```
git clone https://github.com/yourusername/ecommerce-admin-panel.git
cd ecommerce-admin-panel
```

2. Install dependencies:

```
flutter pub get
```

3. Configure environment variables (see [Environment Variables](#))

4. Run the app:

```
# Web (recommended)
flutter run -d chrome --dart-define=SUPABASE_URL=your_url --dart-define=SUPABASE_ANON_KEY=your_key

# Or with default configuration
flutter run -d chrome
```

5. Build for production:

```
flutter build web --dart-define=SUPABASE_URL=your_url --dart-define=SUPABASE_ANON_KEY=your_key
```

Environment Variables

The app uses compile-time environment variables via `--dart-define`:

Variable	Description	Required
SUPABASE_URL	Your Supabase project URL	Yes
SUPABASE_ANON_KEY	Your Supabase anonymous key	Yes

Usage:

```
flutter run -d chrome \
  --dart-define=SUPABASE_URL=https://your-project.supabase.co \
  --dart-define=SUPABASE_ANON_KEY=your-anon-key-here
```

Key Modules

Dashboard Module

The dashboard provides a real-time overview of business metrics:

- **Order Status Pie Chart** - Visual breakdown of order statuses with color coding
- **Weekly Sales Bar Chart** - Day-by-day sales with current/previous week toggle
- **Recent Orders Table** - Paginated table with latest orders
- **Real-time Updates** - Supabase stream subscription with debounced refresh

Product Module

Full product lifecycle management:

- **Single Products** - Fixed price, stock, and attributes
- **Variable Products** - Multiple variations with individual pricing, stock, and images
- **Attribute System** - Dynamic key-value attributes (Color, Size, Material, etc.)
- **Image Management** - Thumbnail + additional images via media library integration

Order Module

Comprehensive order processing:

- **Order Tracking** - Status progression (Placed > Processing > Shipped > Delivered)
- **PDF Labels** - Generate printable parcel labels with QR codes
- **Payment Integration** - PayHere and PayPal support with fee calculation
- **Address Management** - Shipping and billing address handling

Media Module

Centralized asset management:

- **Folder Organization** - Categorized storage (Banners, Brands, Products, etc.)
- **Drag & Drop Upload** - Flutter Dropzone integration
- **Image Preview** - Full-size preview with metadata
- **URL Management** - Copy public URLs for use across the platform

Settings Module

Administrative configuration:

- **Role-Based Access** - 5 role levels with granular permissions
 - **User Management** - Add/remove dashboard users
 - **Activity Logging** - Track all admin actions
 - **Payment Configuration** - Gateway settings and processing fees
 - **Shipping & Delivery** - Shipping cost and delivery time configuration
-

State Management

The app uses **GetX** for state management, providing:

- **Reactive State** (RX variables) - Auto-updating UI on data changes
- **Dependency Injection** (Get . put, Get . lazyPut) - Controller lifecycle management
- **Route Management** (GetMaterialApp, Get . toNamed) - Named routing with URL strategy
- **Snackbar/Dialog** - Built-in UI feedback utilities

Key Controllers

Controller	Responsibility
DashboardController	Order stats, weekly sales, real-time streams
CreateProductController	Product form state, validation, submission
ProductsController	Product listing, search, filtering
MediaController	Media upload, folder management
SidebarController	Navigation state, active menu tracking
HeaderController	Header state and user info
SettingsController	App configuration management
RolesController	Role-based permission management
TabsController	Tab CRUD operations
MonthlyStatsController	Monthly analytics data
UserActivityController	Admin action logging

Database Schema

The app interacts with the following Supabase tables:

Table	Description
products	Product catalog with pricing, stock, attributes
product_categories	Many-to-many product-category relationships
categories	Product categories with hierarchy
brands	Product brands
banners	Promotional banners
orders	Customer orders with status tracking
customers	Customer profiles
tabs	Navigation tabs
dashboard_users	Admin user accounts

Table	Description
user_activity_log	Admin action audit trail
app_config	Application settings

Responsive Design

The app implements a **three-tier responsive layout**:

Breakpoint	Layout	Description
$\geq 1336\text{px}$	Desktop	Full sidebar + content area
$\geq 768\text{px}$	Tablet	Drawer sidebar + content
$< 768\text{px}$	Mobile	Drawer sidebar + compact content

Layout Components

- **TSiteTemplate** - Root layout wrapper with responsive switching
 - **DesktopLayout** - Fixed sidebar + scrollable content
 - **TabletLayout** - AppBar with drawer + content
 - **MobileLayout** - AppBar with drawer + compact content
 - **TResponsiveWidget** - Breakpoint-based widget switcher
-

Security

- **Authentication:** Supabase Auth with PKCE flow for web security
 - **Route Guards:** Middleware redirects unauthenticated users to login
 - **Session Management:** Automatic session detection on app start
 - **Role-Based Access:** Menu items and features restricted by user role
 - **Environment Variables:** Sensitive keys passed via `--dart-define` (not hardcoded)
-

PDF & QR Code Generation

The admin panel can generate **printable parcel labels** for orders:

- **PDF Generation** using the `pdf` package
- **QR Codes** linking to review pages via `qr_flutter`

- **Print Support** via the `printing` package
 - **Label Content:** Order ID, customer details, item table, billing summary, QR code
-

Contributing

1. Fork the repository
 2. Create a feature branch (`git checkout -b feature/new-feature`)
 3. Commit your changes (`git commit -m 'Add new feature'`)
 4. Push to the branch (`git push origin feature/new-feature`)
 5. Open a Pull Request
-

License

This project is licensed under the MIT License - see the [LICENSE](#) file for details.

Author

Mohamed Afras

- GitHub: [@yourusername](#)
 - Email: mohamedafraai@gmail.com
-

Built with Flutter & Supabase