CODECANYON SETUP & DEPLOYMENT MANUAL

7 Web AI SaaS — Virtual Try-On

Complete Installation & Configuration Guide for the Laravel 13 Admin Panel and Flutter Android App

Laravel 13 (^13.0) **PHP 8.3+** **Flutter 3.27+ / Dart >=3.6.0**

Android SDK 36 **5 Commercial AI APIs**

Package Folder Structure

After extracting your download archive, your project is organized into four clean directories:

```
7_Web_AI_SaaS_v2.0/
├── Documentation/
│   ├── README.html          # Offline setup guide (open in browser)
│   ├── Documentation.pdf    # Printable visual manual (this file)
│   └── assets/              # Local annotated screenshots
├── Admin_Panel/             # Laravel 13 backend + vendor/ + composer.lock
├── Flutter_App/             # Flutter Android source (com.sevenweb.aiapp)
└── Demo_APK/                # Offline demo.apk for Android preview
```

Table of Contents

1. Server Requirements & Laravel 13 Admin Panel Setup 2
2. Admin Panel Configuration — All 13 Modules Explained 3
3. Flutter Android App Setup, Rebranding & APK/AAB Build 7
4. Third-Party Services, Usage Fees & AI Disclosure 9

1. Server Requirements & Laravel 13 Admin Panel Setup

The backend is built on **Laravel 13** with pre-compiled CSS (no Tailwind Play CDN) and includes both `composer.lock` and `vendor/` so you can deploy directly on shared hosting, cPanel, HestiaCP, or a VPS.

Requirement	Minimum Specification	Setup Notes
PHP Version	PHP 8.3 or 8.4	Select PHP 8.3+ in your hosting control panel.
PHP Extensions	pdo_mysql, mbstring, openssl, curl, gd, fileinfo, zip, xml, ctype	Checked automatically on Step 1 of the web installer.
Database	MySQL 8.0+ or MariaDB 10.6+	Create one empty database and user with full privileges.
SSL & Domain	Valid HTTPS SSL Certificate	Required for Google Sign-In OAuth and mobile API requests.
Dependencies	`composer.lock` + `vendor/` included	No SSH required; run `composer install` only if updating packages.

Step-by-Step Admin Panel Installation

1. **Upload & Extract:** Upload the contents of `Admin_Panel/` to your domain folder (e.g., `public_html/`) and extract the archive.
2. **Set Document Root:** Point your domain or subdomain to the `/public` directory (or keep the included root `.htaccess` redirect if your host uses `public_html/` directly). Ensure `storage/` and `bootstrap/cache/` are writable (`0775`).
3. **Run the 4-Step Web Installer:** Open `https://your-domain.com/install` in your browser:
   - *Step 1 (Requirements):* Confirms PHP 8.3+, extensions, and folder permissions.
   - *Step 2 (Database):* Enter your MySQL host, database name, username, and password, then click **Test Connection**.
   - *Step 3 (Admin Account):* Enter your Super Admin name, email, and strong password.
   - *Step 4 (Finish):* Click **Install Now**. All tables and defaults are created automatically and `storage/installed.lock` secures the installer.
4. **Configure the 1-Minute Cron Job:** Add a cron job in cPanel, HestiaCP, or Linux crontab so subscription expiry checks and reminder emails run automatically:

```
* * * * * /usr/bin/php /path/to/your/public_html/cron.php >/dev/null 2>&1
```

2. Admin Panel Configuration — All 13 Modules

Sign in at `https://your-domain.com/admin/login`. Every sidebar module maps directly to a live setting or mobile app screen:

Admin Module	Purpose & Configuration Summary
1. Dashboard	Displays live KPIs (Total Users, Paid Subscribers, Total Generations, Revenue) and a 7-day earnings chart.
2. Manage Users	Search users, adjust credit balances, assign or extend subscription plans, block/unblock accounts, or permanently delete users.
3. Outfit Catalog	Create categories, set per-category Hidden AI Prompts, Good/Bad photo guides, 2 Sample Models, Store Buttons, and upload garment photos.
4. Virtual Try-On AI	Select your active commercial AI provider, pick from 23 licensed 2-image models, set credits per try-on, and test your API key.
5. Pricing Plans	Configure weekly, monthly, or yearly credit packs, RevenueCat Package IDs, and preview the mobile paywall live.
6. Payment Gateways	Choose active gateway (RevenueCat, Stripe Live, or PayPal Live) and save API credentials. All payments are verified server-side.
7. Mail Templates	Customize Purchase Confirmation, 3-Day Expiry Reminder, and Subscription Expired emails with tags like `{user_name}` and `{credits}`.
8. Transactions	Audit verified payment records with customer email, plan name, USD amount, gateway, and transaction ID.
9. Manage FAQs	Create, edit, and reorder FAQ entries displayed live inside the mobile app Help section.
10. Policy Page	Edit your Privacy Policy & Terms in plain text; publishes live to `/page/privacy-policy` and the mobile app.
11. Theme & Settings	Set App Brand Name, New Customer Starting Credits, Google Play Delete Account visibility toggle, and Contact Us info.
12. Social Login	Enable Google Sign-In and enter your Google Cloud OAuth 2.0 Web Client ID & Client Secret (verified server-side against audience).

Admin Module	Purpose & Configuration Summary
13. SMTP & Security	Configure outgoing SMTP credentials for transactional emails and optional Google Authenticator 2FA for admin login.

2.1 Setting Up Outfit Categories & Pre-Built Garments

Open **Outfit Catalog** to organize your clothing categories (for example: Women's Full Outfits, Men's Full Outfits, Suits & Party Wear, Jeans & Trousers, Shirts, Jackets). Clicking **Edit** on any category lets you upload a *Good Photo* and *Bad Photo* guide, up to 2 *Sample Models*, a Hidden AI Prompt, and an optional **Shop Button** (Amazon or 13 other store presets) with your product URL.

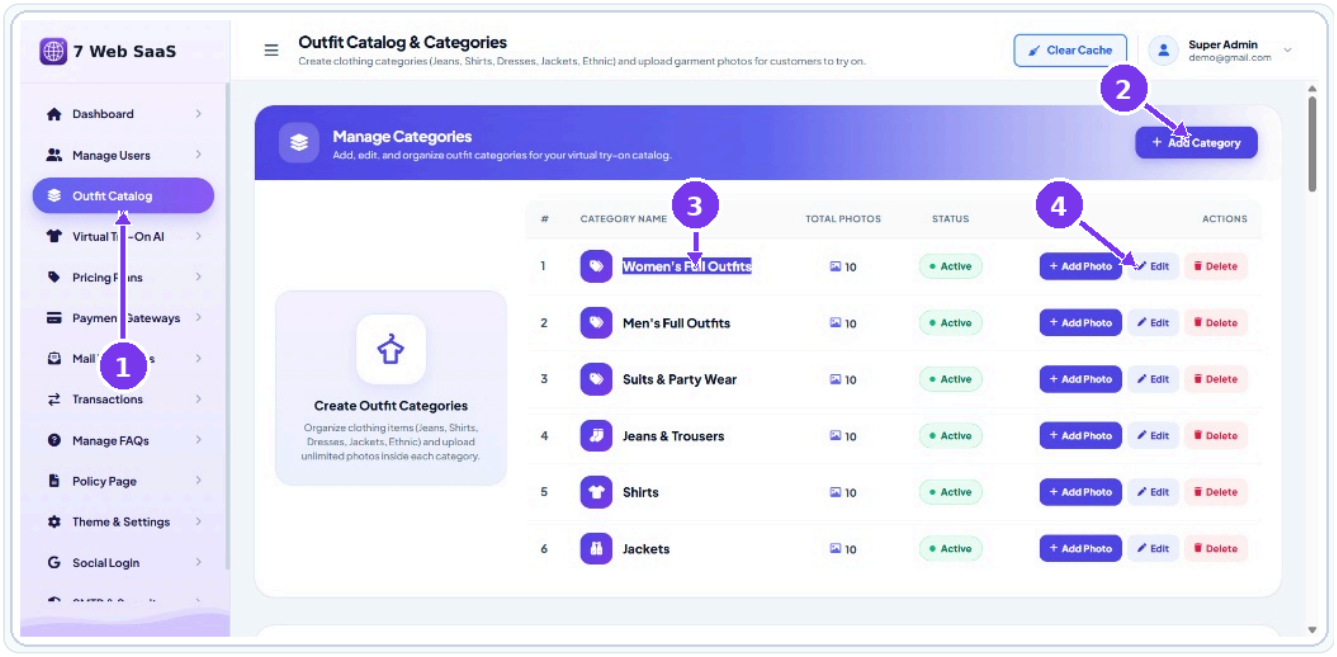


Figure 1 — Outfit Catalog & Categories: Manage categories, photo counts, and per-category try-on defaults.

1. **Sidebar Navigation:** Access all 13 admin modules.
2. **+ Add Category:** Create a new clothing category with defaults.
3. **Photo Counter:** Tracks uploaded garment photos per category.
4. **+ Add Photo / Edit:** Upload outfits or edit guides, samples & store links.

2.2 Connecting Your Commercial Virtual Try-On AI Provider

Open **Virtual Try-On AI**, select one of the five supported commercial 2-image providers, choose a verified model, set **Credits Per Try-On**, paste your provider API key, and click **Test API** before saving.

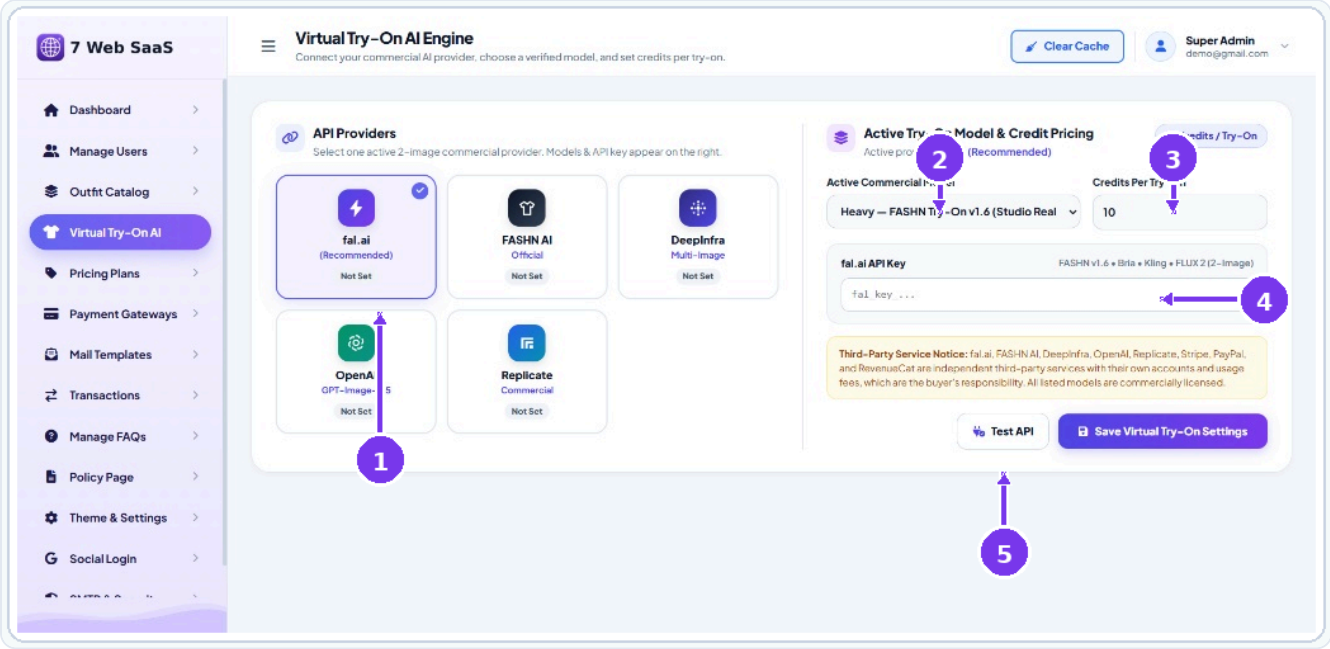


Figure 2 — Virtual Try-On AI Engine: Five commercial providers with 23 verified 2-image try-on models.

1 **API Providers:** Switch between fal.ai, FASHN AI, DeepInfra, OpenAI, or Replicate.

2 **Active Model:** Pick from commercially licensed 2-image try-on models.

3 **Credits Per Try-On:** Credits deducted per successful generation.

4 **Provider API Key:** Paste your own key from the provider dashboard.

5 **Test API & Save:** Verify connectivity before saving changes.

2.3 Configuring Pricing Plans & Payment Gateways

In **Pricing Plans**, set each plan's title, billing interval, USD price, credits awarded, and RevenueCat Package ID. The right-side phone preview updates immediately to match the Flutter Paywall screen.

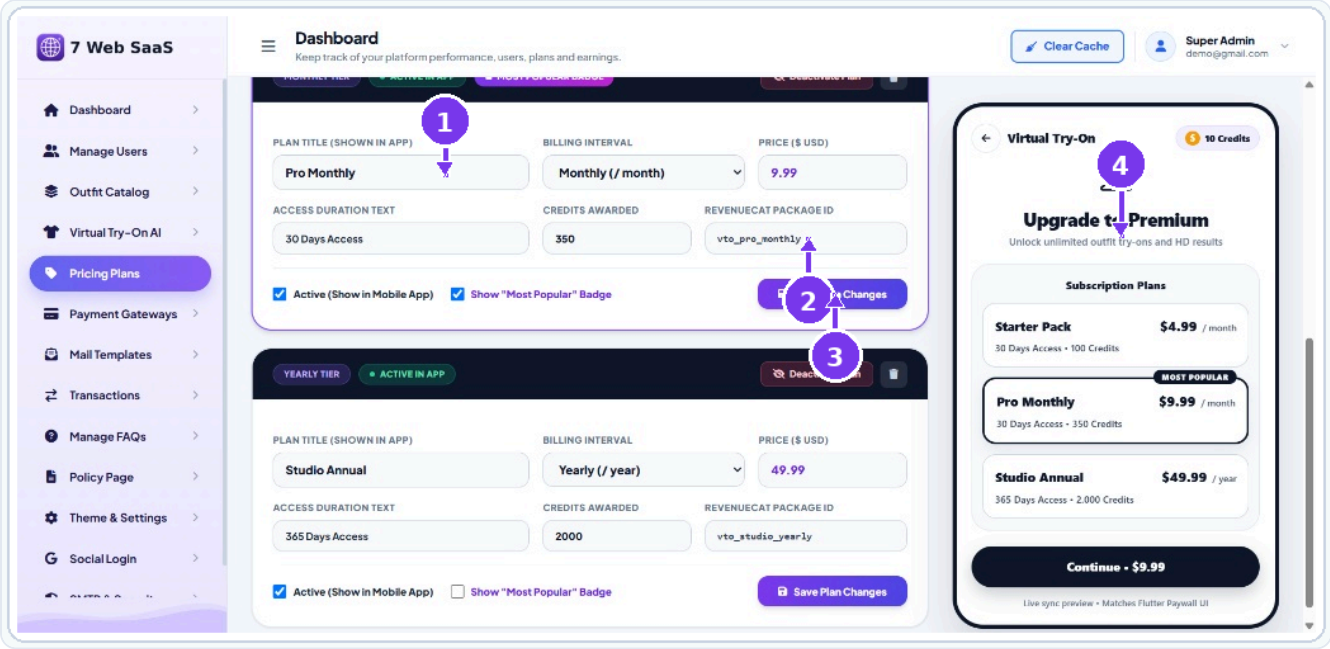


Figure 3 — Pricing Plans & Live Paywall Preview: Changes sync directly to the Flutter mobile app.

1 Plan Pricing & Credits: Edit title, interval, USD price, and credit allotment.

2 RevenueCat Package ID: Match your Google Play / RevenueCat product ID.

3 Save Plan Changes: Publish updated pricing live to the app.

4 Live Paywall Preview: Real-time preview of the mobile upgrade screen.

Next, open **Payment Gateways** and enable your preferred billing provider:

- **RevenueCat (Recommended for Google Play):** Enter your RevenueCat Public SDK Key, Secret REST API Key, and Webhook Secret. Purchases are verified server-side against RevenueCat before credits are granted.
- **Stripe (Live Mode):** Enter your Live Publishable Key (`pk_live_...`) and Live Secret Key (`sk_live_...`). The backend verifies exact paid amount, currency, and plan match.
- **PayPal (Live Mode):** Enter your Live Client ID and Live Secret Key. Production always validates against the live PayPal API.

2.4 Theme, Settings, Delete Account Toggle & Google Login

Open **Theme & Settings** to set your mobile app header brand name, free starting credits for new accounts, Contact Us details, and the **Delete Account** visibility switch.

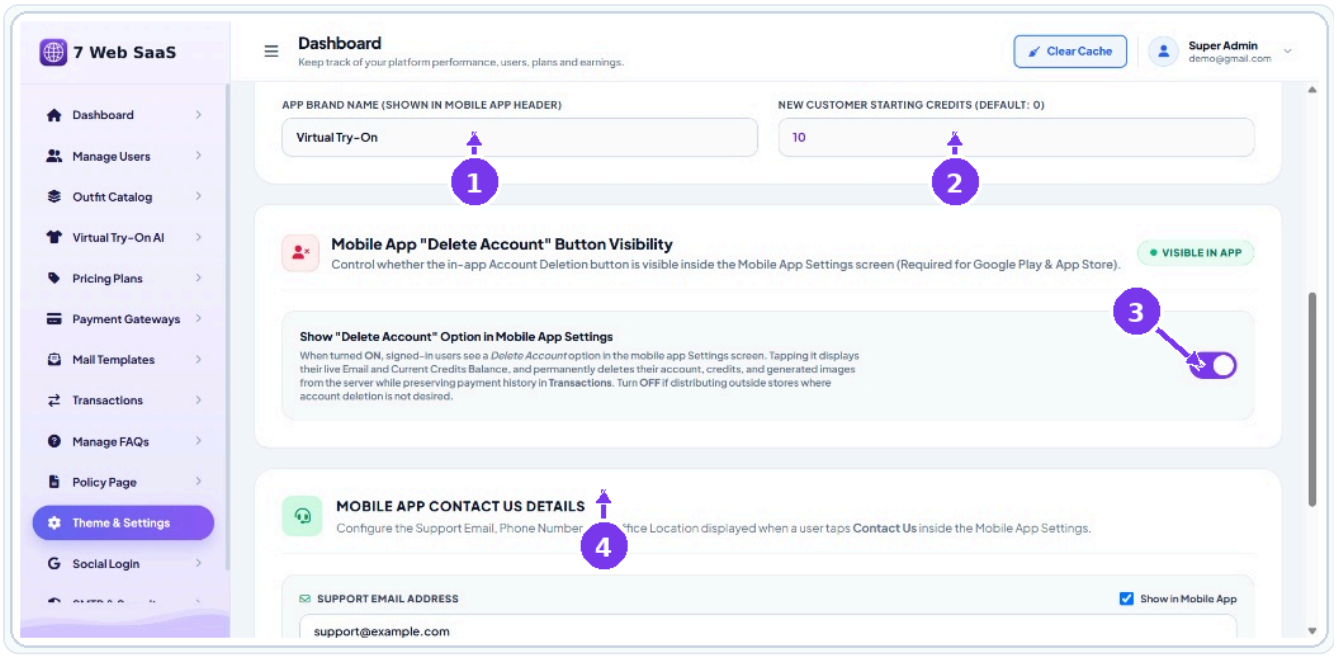


Figure 4 — Theme & Settings: Brand name, starting credits, Google Play compliance toggle, and support info.

1 **App Brand Name:** Updates the mobile app header text live.

2 **Starting Credits:** Free credits automatically given to new users.

3 **Delete Account Toggle:** Keep ON for Google Play compliance; turn OFF only for private outside-store distribution.

4 **Contact Us Details:** Support email, phone, and address shown in-app.

Google Social Login Setup: In Google Cloud Console, create an **OAuth 2.0 Web Application Client ID** with Authorized Redirect URI `https://your-domain.com/auth/google/callback`, plus an **Android OAuth Client ID** matching your package name and SHA-1 fingerprint. Paste the Web Client ID and Secret into **Admin Panel → Social Login**. The backend strictly verifies the Google ID/access token audience before issuing a session token.

SMTP & 2FA Setup: In **SMTP & Security**, enter your mail host (e.g., `smtp.gmail.com`, port `587`, TLS), username, and app password so purchase and expiry emails deliver reliably. You may also enable optional Google Authenticator 2FA for admin sign-in.

3. Flutter Android App Setup, Rebranding & Build

The mobile application is written in **Flutter 3.27+ (Dart >=3.6.0 <4.0.0)** targeting **Android SDK 36** (`compileSdk = 36` , `targetSdk = 36` , `minSdk = 24`) and stores user session tokens securely with `flutter_secure_storage` .

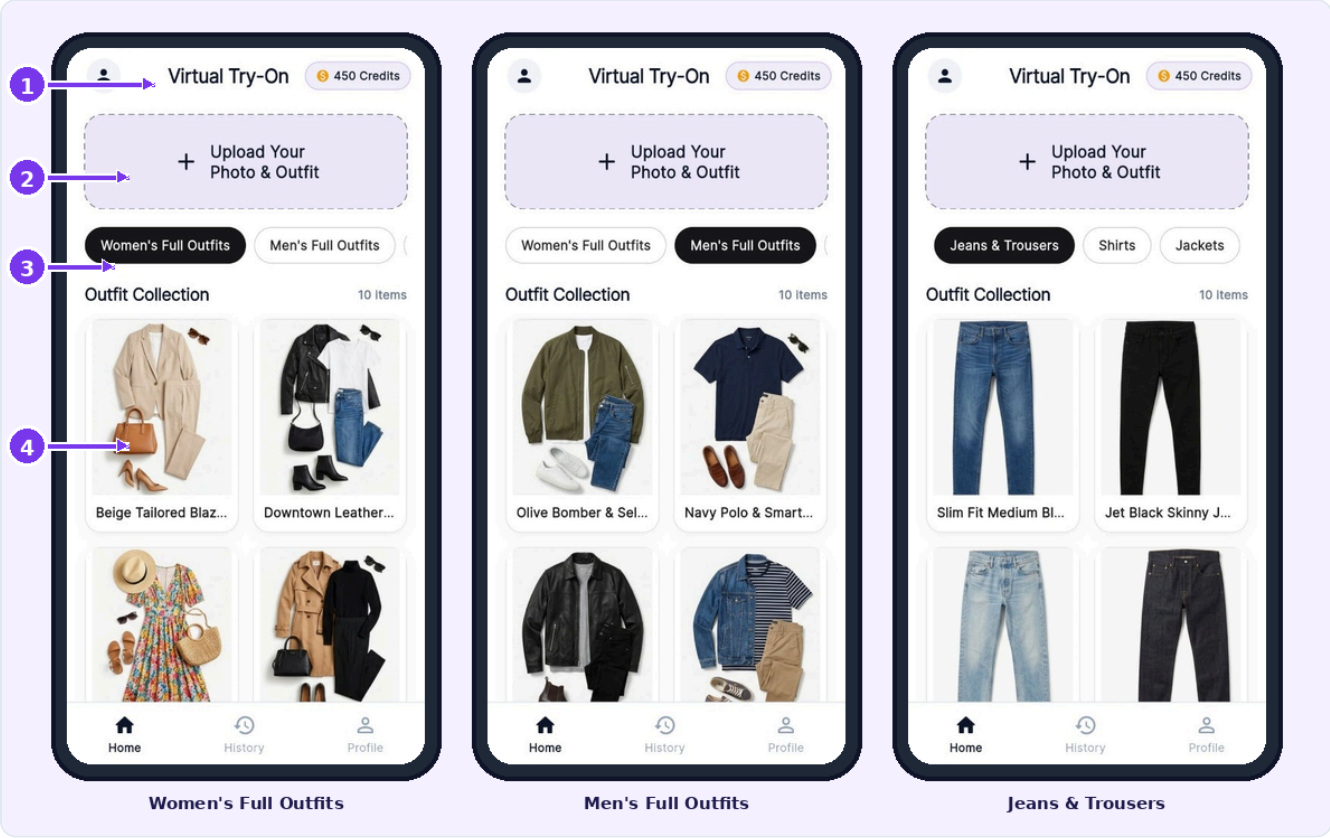


Figure 5 — Flutter Android App Home Catalog: Category-separated outfit browsing synced live with the Admin Panel.

1 Live Header & Credits: Syncs App Brand Name and user balance from Admin.

2 Custom Photo & Outfit: Lets users upload any garment from their gallery.

3 Category Pills: Clean per-category filtering synced from Outfit Catalog.

4 Pre-Built Outfit Cards: Tap any card to open the Try-On workspace & Shop link.

Step 1: Connect the App to Your Domain

Open `Flutter_App/lib/config/api_config.dart` and replace `siteRoot` with your live HTTPS Admin Panel domain (no trailing slash):

```
class ApiConfig {
  static const String siteRoot = 'https://your-domain.com';
  static const String baseUrl = '$siteRoot/api/v1';
```

Step 2: Change Package Name (Application ID) & App Name

1. **Application ID:** Open `android/app/build.gradle.kts` and replace `com.sevenweb.aiapp` in both `namespace` and `applicationId` with your unique reverse-domain ID (e.g., `com.yourcompany.tryon`).
2. **Kotlin Package:** Update the `package` line at the top of `android/app/src/main/kotlin/.../MainActivity.kt` to match your new Application ID.

3. **App Display Name:** Open `android/app/src/main/AndroidManifest.xml` and change `android:label="7 Web AI"` to your desired launcher name.

Step 3: Replace App Icon & Optional Splash / Onboarding Screens

- **Launcher Icon:** Replace the PNG icons inside `android/app/src/main/res/mipmap-mdpi` through `mipmap-xxxhdpi` with your own square logo.
- **Splash Screen & Onboarding (Optional):** You can keep the built-in splash screen and onboarding slides as-is, or optionally replace `assets/onboarding/slide_1.jpg`, `slide_2.jpg`, and `slide_3.jpg` and edit the titles in `lib/config/onboarding_config.dart`.

Step 4: Register Android SHA-1 Fingerprint for Google Sign-In

Generate your release keystore SHA-1 fingerprint and paste it into your Android OAuth Client ID in Google Cloud Console:

```
keytool -list -v -keystore your-release-key.jks -alias your-key-alias
```

Step 5: Build Release APK or Google Play App Bundle (.aab)

Open a terminal inside `Flutter_App/` and run:

```
flutter pub get
flutter build apk --release          # Output: build/app/outputs/flutter-
apk/app-release.apk
flutter build appbundle --release    # Output:
build/app/outputs/bundle/release/app-release.aab
```

> **Official Platform Support Policy (Android Only)**
>
> This Flutter codebase is **100% tested and officially supported on Android** (Android 7.0 / API 24 through Android 16 / API 36). Because Flutter is cross-platform, buyers may optionally compile for iOS or Web at their own discretion; however, official author technical support, updates, and liability apply exclusively to Android.

4. Third-Party Services, Usage Fees & AI Disclosure

Third-Party Services & Buyer Responsibility Notice

fal.ai, FASHN AI, DeepInfra, OpenAI, Replicate, Google Cloud (OAuth Sign-In), Stripe, PayPal, and RevenueCat are independent third-party services. Each service requires its own account and carries its own usage fees, rate limits, and terms of service. All third-party API, cloud, and payment processing costs are solely the buyer's responsibility.

All 23 Virtual Try-On models pre-configured in the Admin Panel are commercially licensed 2-image models. Buyers are responsible for reviewing their chosen provider's current pricing and terms before launching in production.

AI Usage & Sample Assets Disclosure

- **Runtime AI Functionality:** The application uses third-party generative AI APIs (configured by the administrator in *Admin Panel → Virtual Try-On AI*) to combine a user photo and a garment image into a Virtual Try-On preview.
- **Development & Documentation Assistance:** AI-assisted tools were used to help draft code refactoring, setup documentation, and demo catalog descriptions, all of which were reviewed and tested by the author.
- **Demo Sample Outfits & Model Photos:** All sample outfit flat-lays, demo model portraits, Good/Bad photo guide examples, and pre-built demo try-on results included in the offline Demo APK and Demo Admin Panel are synthetic AI-generated studio fashion images created specifically for demonstration. They do not depict real third-party individuals or copyrighted brand photography. For live commercial stores, buyers should replace demo images with their own licensed product catalogs.