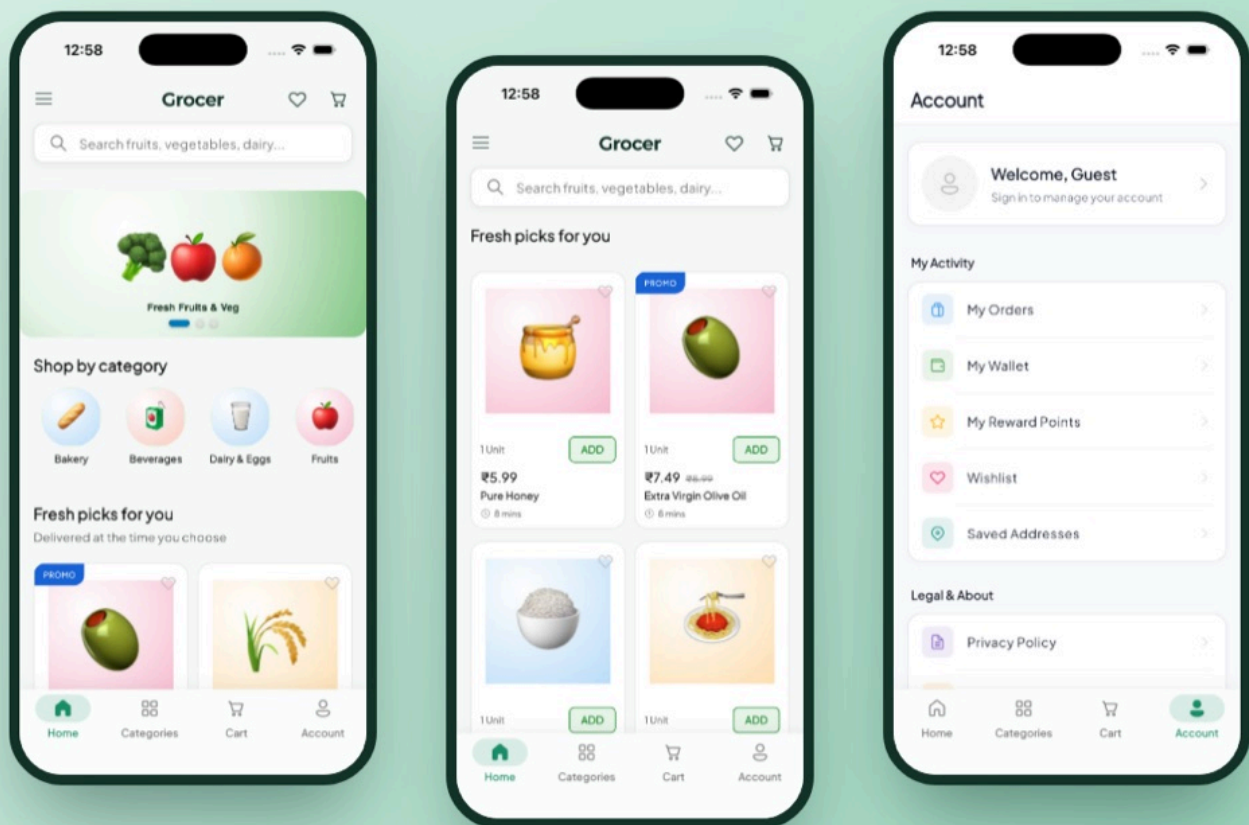


Mobile App

A step-by-step guide for everyone, no coding needed: set up your store, design the app, and build your Android app online. Developers will also find the source-code and REST API chapters.

1. Start here

In plain words: This product gives your online store its own **mobile shopping app** for Android phones and iPhones. Your customers install it, browse your products, add to the cart, pay and track orders. You manage everything from your normal WordPress dashboard.



The app, running on a phone: home screen, product list and account page.

What you bought: three parts that work together

Part	What it is	Who needs it
1. The plugin <code>mobile-app-for-woocommerce-plugin.zip</code>	Goes on your WordPress store. It lets you design the app's home screen, choose colours and login options, send notifications and build the app.	Everybody
2. The online App Builder (a tab inside the plugin)	Makes the Android app file for you on our servers. You fill in a form and press one button. No programming tools needed.	Most people
3. The app source code <code>flutter-app-source.zip</code>	The full code of the app (Flutter). Use it if you want an iPhone version, want to change how the app works, or prefer to build it yourself.	Developers

Which way should I go?

I want...	Do this
An Android app, quickly, with no technical work	Follow chapters 3 to 12. You never touch the source code.
An iPhone app	Chapters 3 to 9 for the store setup, then chapter 13 (needs a Mac and a developer).
To change the app's design or code	Chapters 13 and 15.

The big picture in 8 steps

- 1 Check your store is ready (chapter 3).
- 2 Install the plugin (chapter 4).
- 3 Design the home screen (chapter 5).
- 4 Choose colours, login and links (chapters 6 and 7).

- 5 Create a free Firebase project for notifications (chapter 8).
- 6 Build the Android app online (chapter 10).
- 7 Install it on your phone and test it (chapter 11).
- 8 Publish it to Google Play (chapter 12).

How long does it take? Plan about an hour for the setup the first time. The online build itself takes a short while and the download link arrives by email.

2. Words you will see

In plain words: Here is a short dictionary, so nothing in this guide feels like a foreign language.

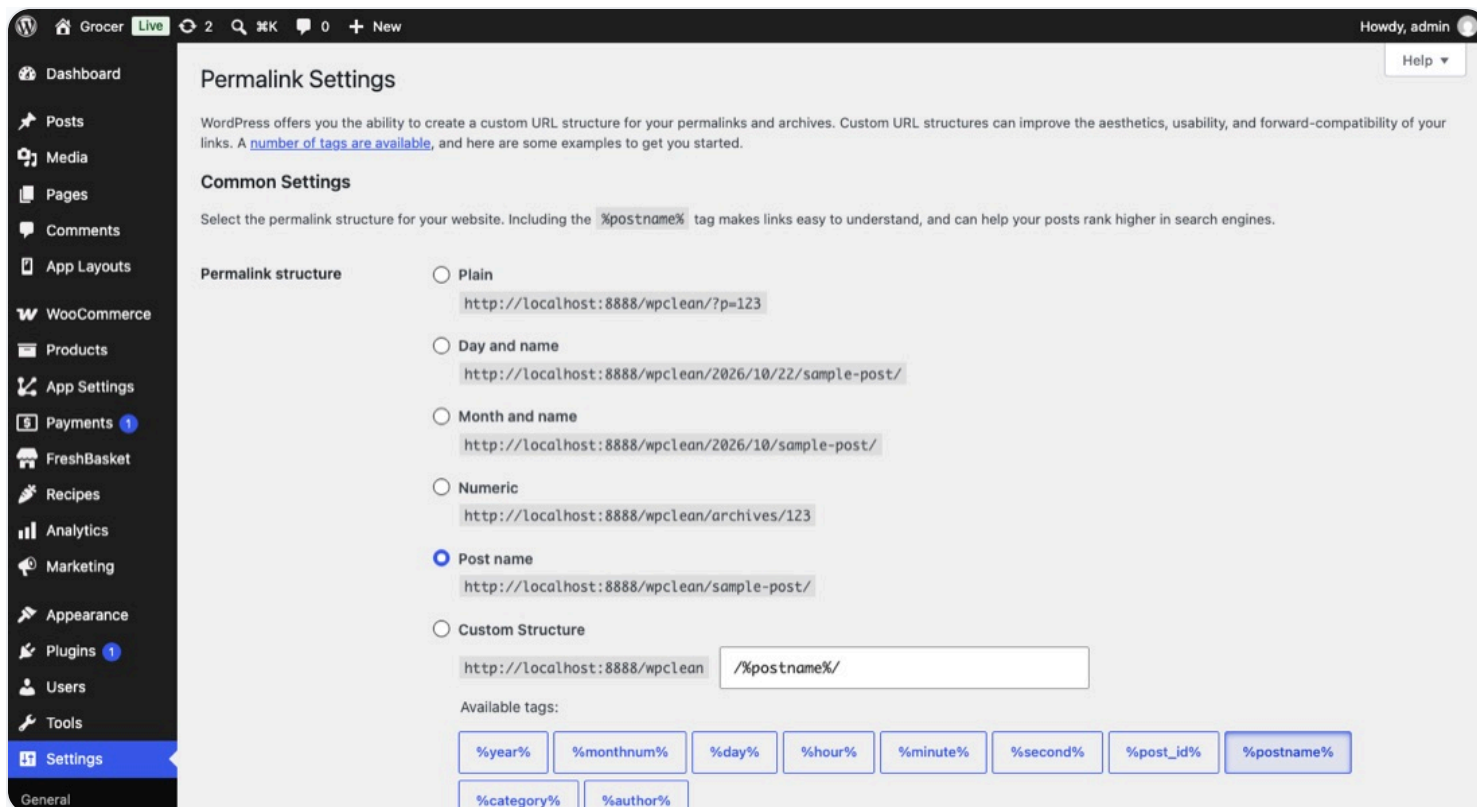
Word	What it means
App layout	One screen of your app (usually the home screen), designed in WordPress with blocks.
Block	A building piece of a page: a banner slider, a row of categories, a grid of products.
API	The "messenger" between your store and the app. The app asks your store for products; your store answers. You do not need to do anything with it.
Firebase	A free Google service that sends push notifications to phones and powers Google login.
Push notification	A message that pops up on the customer's phone, even when the app is closed.
APK	The Android app file. You install it on a phone to test.
AAB (App Bundle)	The Android file that Google Play asks for when you publish.
Package name	The app's unique ID, like <code>com.yourshop.app</code> . It never changes after you publish.
HTTPS	The padlock in the browser. Your store address must start with <code>https://</code> .
Permalinks	The way your store builds web addresses. Anything except "Plain" works.
Purchase code	The code that proves you bought this product. The builder asks for it.
OTP	A one-time code sent by SMS so people can log in with their phone number.
Flutter	The technology the app is written in. Only developers need it.

3. Before you start: check your store

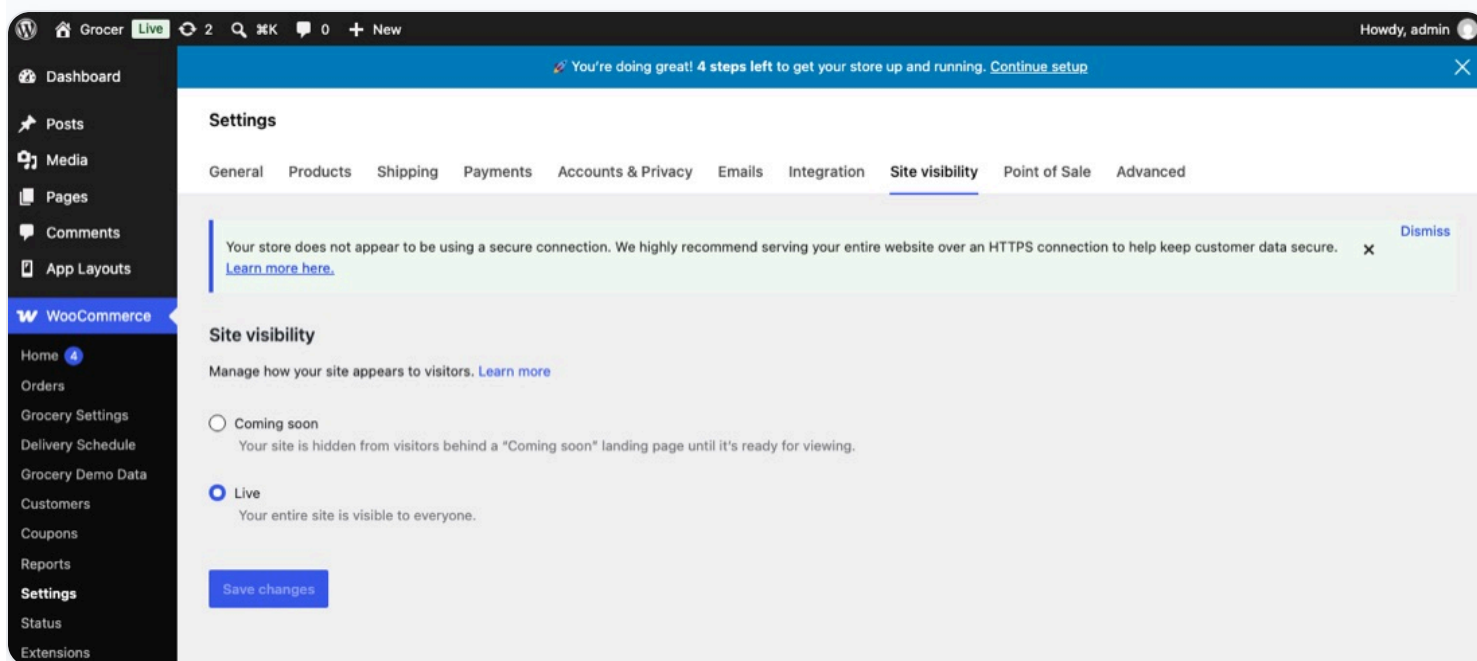
In plain words: A few quick checks now save a lot of time later. All of them take a minute.

Checklist

Check	Why	How
WordPress 6.0+, WooCommerce 8.0+, PHP 7.4+	Required versions	Dashboard → Tools → Site Health → Info
Your store address starts with https://	Phones refuse plain http	Ask your hosting company to enable the free SSL certificate
Permalinks are not "Plain"	The app reaches your store through <code>/wp-json/</code>	Settings → Permalinks → choose "Post name" → Save
The store is Live	New stores start in "Coming soon" mode and the app would see nothing	WooCommerce → Settings → Site visibility → Live
You have products	An empty store makes an empty app	Products → Add New



Settings → Permalinks: choose any option except Plain, then save.



WooCommerce → Settings → Site visibility: choose Live.

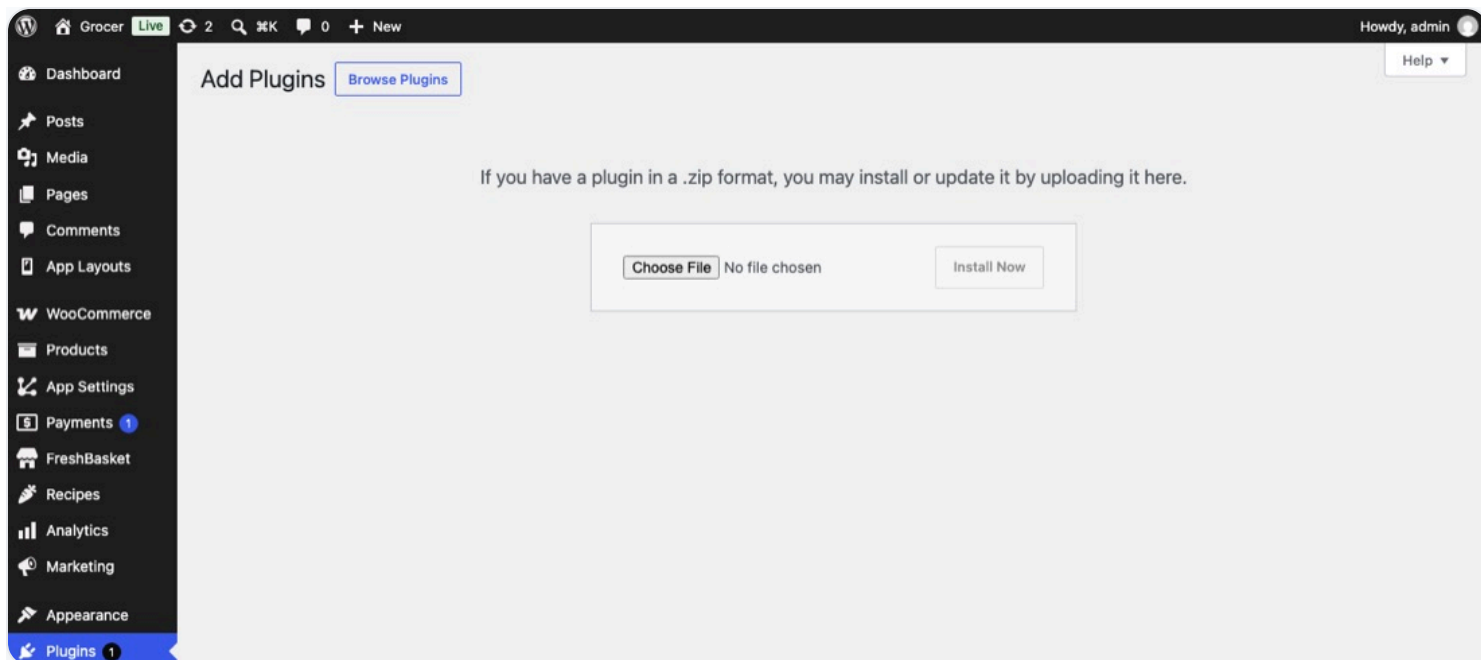
Accounts you may need

Account	For	Cost
Google (Gmail)	Firebase (notifications). Needed.	Free
Google Play Console	Putting the app on Google Play	One-time fee
Apple Developer	Putting an iPhone app on the App Store	Yearly fee
Facebook developer, Apple sign-in, SMS provider	Only if you want those login methods	Varies

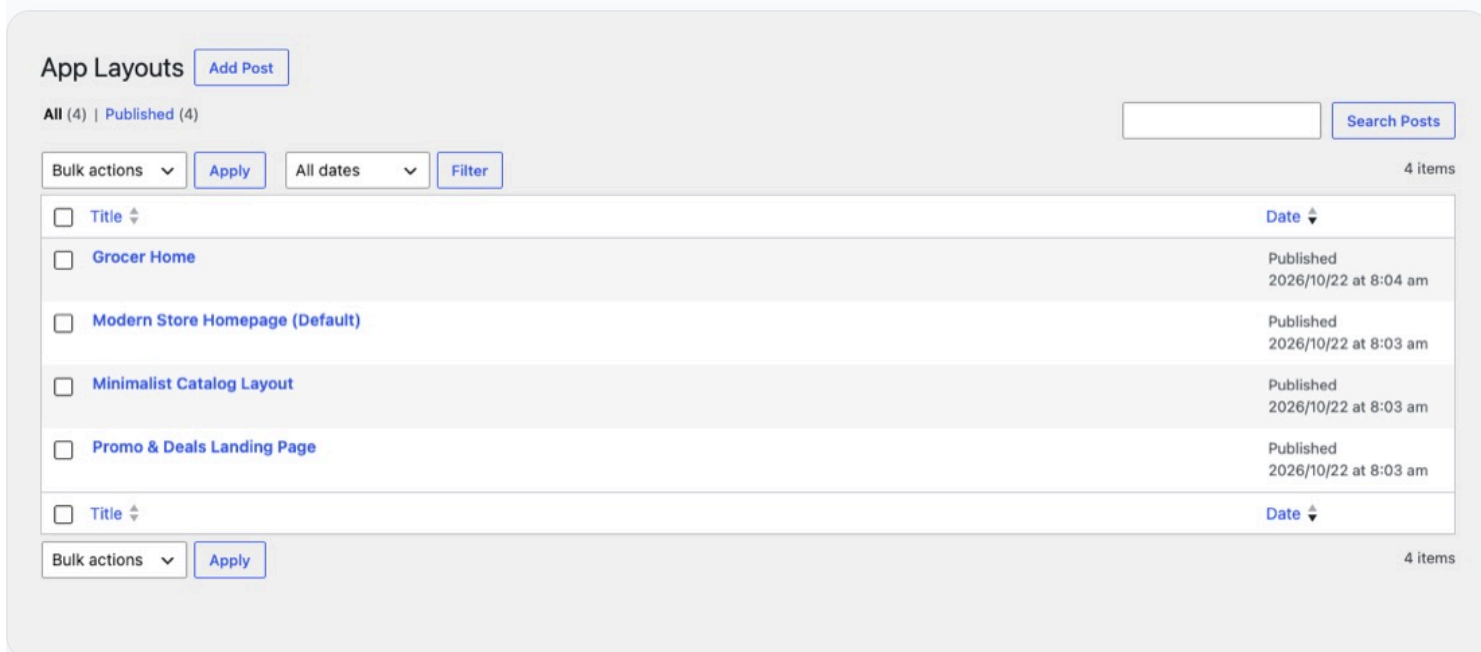
4. Install the plugin

- 1 In your WordPress dashboard open **Plugins → Add New → Upload Plugin**.
- 2 Choose the file `mobile-app-for-woocommerce-plugin.zip` and press **Install Now**, then **Activate**.

Use the *inner* zip files only. Do not upload the big complete-package zip.
- 3 Two new menu items appear on the left: **App Layouts** and **App Settings**. Three sample layouts were created for you.



Plugins → Add New → Upload Plugin.



App Layouts: your app screens. Three samples are included; edit one or add your own.

Check that it works

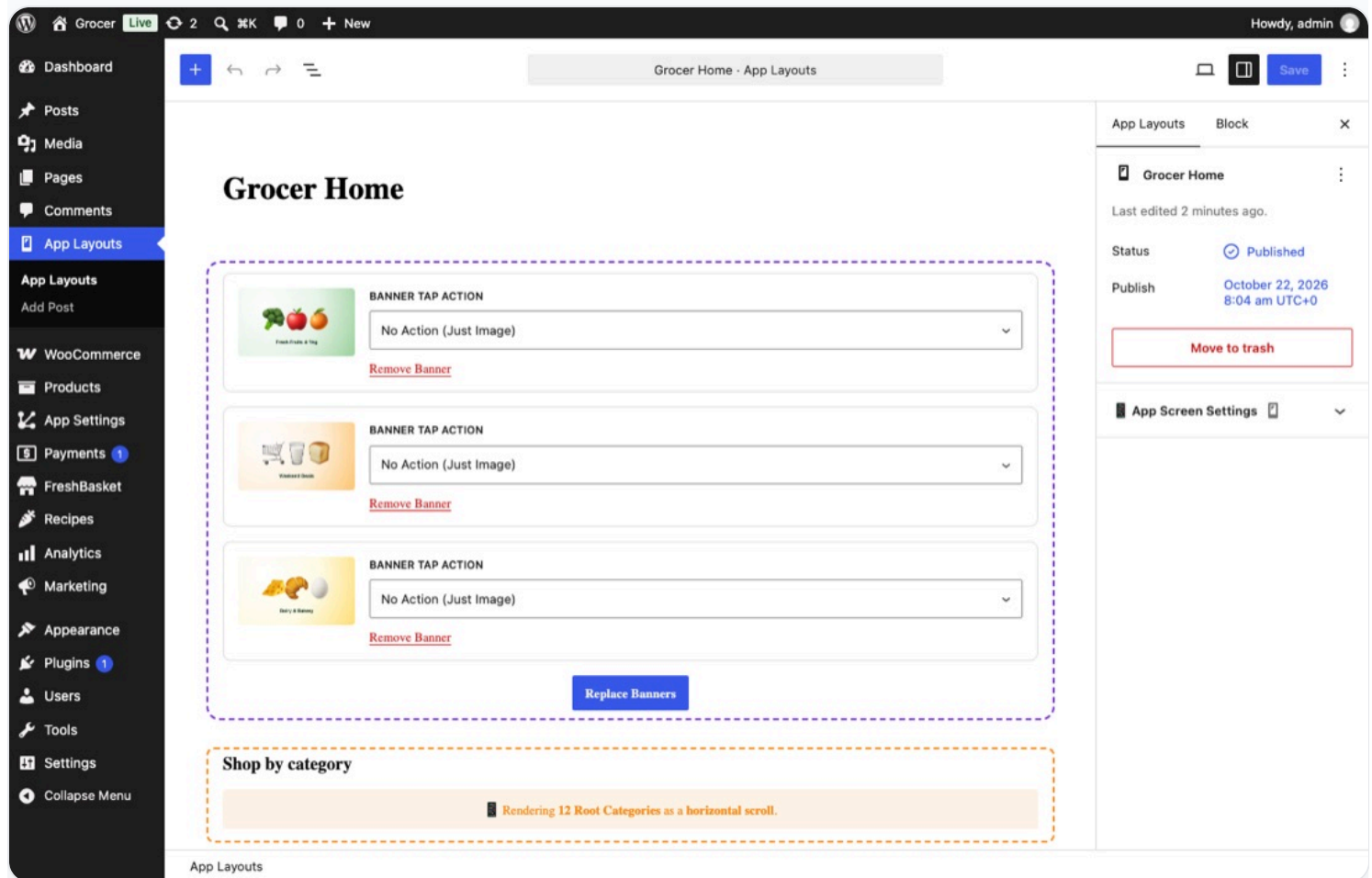
Open this address in your browser (replace with your own store address):

```
https://yourstore.com/wp-json/mo-app/v1/homepage
```

If you see a page of text full of curly brackets, the plugin works. If you see "Active App Layout not configured", that is normal for now: you choose the layout in chapter 6.

5. Design your home screen

In plain words: The home screen of your app is built like a normal WordPress page, but with special blocks. You stack blocks on top of each other: first a banner slider, then categories, then products. No coding.



The layout editor: three banners, then a row of categories (a products block comes below).

- 1 Go to **App Layouts** and open *Modern Store Homepage* (or press **Add Post** for a new one).
- 2 Click the blue **+** button and search for **Mobile**. You will find three blocks: *Mobile: App Banners*, *Mobile: App Categories*, *Mobile: App Product Collection*.
- 3 Add the blocks you want and drag them into the order you like.
- 4 Click a block and use the settings on the right to change it (see the tables below).
- 5 Press **Update** (or **Publish**).

The three blocks

Block	Shows	Main settings you will use
App Banners	Pictures that slide, or a grid of promotional pictures	Add images; choose what happens when tapped (open a product, a category, a web link, or nothing); slider or grid; height; rounded corners; autoplay
App Categories	Your product categories as round pictures	Style (scroll sideways, grid or list); item size; how many; choose a parent category to show its sub-categories
App Product Collection	A grid or a scrolling row of products	Choose a category; how many products; card size; columns; an optional countdown timer for sales

Make it pretty (all three blocks)

- **Title and description** with their own colours.
- **Background:** a colour, a gradient or a picture.
- **Spacing:** padding above and below, rounded corners.

Use pictures of the right shape. Banner pictures around 1200 × 500 pixels look good. Square pictures suit category circles.

The side panel: App Screen Settings

In the right-hand panel you will find **App Screen Settings**. It controls the top bar and the search bar of this screen:

Setting	What it does
App Bar Title / Logo	The name or logo at the top of the screen
Hide App Bar	Full-screen look with no top bar
Background / Text colour	Colours of the top bar
Show Bottom Search Bar	Adds a search box and lets you set its hint text
Pull to refresh	Customers can swipe down to reload
Pinned / Floating	Whether the top bar stays or hides when scrolling

What is saved behind the scenes (for the curious)

Each layout is saved as simple block text like this. You never have to type it, it is only here so you know there is no magic:

```
<!-- wp:wc-mobile/banners {"layoutType":"slider","bannerHeight":170,"showDots":true,"auto
<!-- wp:wc-mobile/categories {"title":"Shop by category","displayStyle":"horizontal_scrol
<!-- wp:wc-mobile/products {"title":"Fresh picks for you","layoutType":"grid","gridColumn
```

You can keep several layouts, for example a normal one and a sale one. Only the one you choose in App Settings is shown in the app. Changes appear in the app the next time it opens or is refreshed. **No new app build is needed.**

6. App Settings, tab by tab

In plain words: Open **App Settings** in the left menu. Everything that is not the home screen lives here, in tabs along the top. After changing anything press the blue **Save All Settings** button.

General Settings

Mobile App API Configuration

General Settings

Drawer Menu

Legal & Policies

Support & Social

App Info & Updates

Theme (Flex)

Integrations & APIs

Send Notification

App Builder

Basic settings that control how your mobile app functions and appears.

Active App Homepage:

Grocer Home

Select the design layout you want to show on the main home screen of your mobile app.

App Layout Direction:

System Default

Forces the app's text and menus to flow in a specific direction. 'System Default' automatically adapts to the customer's phone language.

Legacy Accent Color:

Your brand's main color. This is used as a backup color if the advanced 'Theme (Flex)' settings are turned off.

Save All Settings

General Settings: choose which layout is the app's home screen.

Setting	What to do
Active App Homepage	Choose your layout here. This is the step that switches the home screen on.
App Layout Direction	Leave on <i>System Default</i> . Choose <i>Right-to-Left</i> to force Arabic-style layout for everyone.
Legacy Accent Color	Only used if you turn the theme options off. You can ignore it.

Drawer Menu

Mobile App API Configuration

General Settings

Drawer Menu

Legal & Policies

Support & Social

App Info & Updates

Theme (Flex)

Integrations & APIs

Send Notification

App Builder

Customize the slide-out navigation menu (the "hamburger" menu) inside your app.

Enable Drawer Menu: ☒ Show the slide-out side menu in the app.

Enable My Wallet Menu: ☐ Show the "My Wallet" button so customers can see their store credit.

Enable Reward Points Menu: ☐ Show the "Reward Points" button for loyalty programs.

Menu Background Style: Solid Color

Choose how the background of the slide-out menu should look.

Background Color:

Only applies if "Solid Color" is selected above.

Background Gradient: linear-gradient(135deg, #0693e3 0%, #9b51e0 10)

A CSS code for color blending. Only applies if "Color Gradient" is selected. Example: `linear-gradient(to right, red, blue)`

Background Image URL:

Paste the full web link to an image. Only applies if "Image Background" is selected.

The side menu that slides in from the left of the app.

Switch the slide-out menu on or off, show or hide *My Wallet* and *Reward Points* (they need extra plugins, see chapter 14), and choose its background: a colour, a gradient or a picture address.

Legal & Policies

Mobile App API Configuration

General Settings

Drawer Menu

Legal & Policies

Support & Social

App Info & Updates

Theme (Flex)

Integrations & APIs

Send Notification

App Builder

These links will be displayed in the app's settings menu so customers can read your policies.

Privacy Policy URL:

Terms & Conditions URL:

Return Policy URL:

Save All Settings

Links to your privacy, terms and returns pages.

Paste the web addresses of your Privacy Policy, Terms & Conditions and Return Policy pages. Stores such as Google Play require a privacy policy.

Support & Social

Mobile App API Configuration

General Settings

Drawer Menu

Legal & Policies

Support & Social

App Info & Updates

Theme (Flex)

Integrations & APIs

Send Notification

App Builder

Provide contact methods for your customers. These appear in the "Contact Us" section of the mobile app.

Support Email:

hello@grocer.example

WhatsApp Number:

+1234567890

Include the country code. When tapped, it opens a chat with your business.

Facebook Page URL:

Instagram Profile URL:

Twitter/X Profile URL:

Contact details shown in the app's Contact Us area.

Your support email, WhatsApp number (with country code, for example +971...) and your Facebook, Instagram and X pages.

App Info & Updates

Mobile App API Configuration

General Settings

Drawer Menu

Legal & Policies

Support & Social

App Info & Updates

Theme (Flex)

Integrations & APIs

Send Notification

App Builder

Manage where customers can download your app and force them to update to new versions.

iOS App Store Link:

Android Play Store Link:

Minimum App Version:

If a customer's app version is lower than this number (e.g., **1.0.5**), the app will lock and force them to update from the store.

[Save All Settings](#)

Store links and the minimum app version.

After you publish, paste the Play Store and App Store links. **Minimum App Version:** if a customer has an older version than this number (for example 1.0.5), the app asks them to update before continuing. Leave it empty until you really need it.

Theme (Flex)

Mobile App API Configuration

- General Settings
- Drawer Menu
- Legal & Policies
- Support & Social
- App Info & Updates
- Theme (Flex)
- Integrations & APIs
- Send Notification
- App Builder

App Visual Identity

These settings automatically generate a beautiful, modern theme inside the mobile app using FlexColorScheme.

Primary Brand Color:

The dominant color of your brand (used for main buttons, app headers).

Secondary/Accent Color:

A contrasting color used to draw attention to special items (like sale badges or floating action buttons).

Surface Blend Mode:

Level Surfaces, Low Scaffold (Recommended)

This setting subtly mixes your "Primary Brand Color" into the app's white/gray backgrounds, giving the app a highly customized, premium feel.

Color Blend Strength (0-40):

7

How heavily the primary color should tint the backgrounds. 0 is pure white/gray. 7 is a gentle tint.

Top Bar Style:

Surface (Matches Background smoothly)

Changes the color of the very top header bar in the app.

Roundness (Border Radius):

14

Controls how round the corners are for buttons, images, and cards. (0 = Sharp squares, 12 = Soft curves, 30+ = Pill shaped).

Main Button Style:

Filled (Material 3 Default)

Select the primary visual style for action buttons (e.g., Add to Cart, Checkout).

Button Base Color Role:

Primary Brand Color

Defines which flex_color_scheme role the main buttons will map to.

Custom Button Color:

Only applies if Custom Specific Color is selected above. Explicitly overrides the default theme button layout colors.

Button Border Radius:

14

Overrides the general roundness setting specifically for buttons to create distinct interactive elements. (0 = Sharp squares, 20+ = Pill shaped).

Theme settings: colours, corners and buttons of the whole app.

Setting	What it changes
Primary Brand Color	Main buttons, active menu item, highlights. Pick your brand colour.
Secondary / Accent Color	Small highlights such as badges
Surface Blend Mode and Strength	How much of your brand colour tints the backgrounds. "Level Surfaces, Low Scaffold" with strength 7 is a gentle, recommended look.
Top Bar Style	Whether the top bar is plain or coloured
Roundness	Corner roundness of cards and pictures (0 sharp, 12 soft, 30 very round)
Main Button Style	Filled, with shadow, or outline only
Button colour and corners	Use the brand colour or choose a special one
Modern Design (Material 3)	Leave on

Currency, number of decimals and tax display are **not** set here: the app copies them from WooCommerce → Settings automatically.

7. Let customers log in

In plain words: Customers can always create an account with an email and password. You can also offer faster ways to log in. Switch on only the ones you are ready to set up.

Mobile App API Configuration

[General Settings](#)[Drawer Menu](#)[Legal & Policies](#)[Support & Social](#)[App Info & Updates](#)[Theme \(Flex\)](#)[Integrations & APIs](#)[Send Notification](#)[App Builder](#)

Login & Authentication Methods

Select which login methods are available to your customers in the mobile app.

Facebook Login: ☒ Enable Facebook Login

Google Login: ☒ Enable Google Login

Apple Login: ☒ Enable Apple Login

Phone Number Login: ☒ Enable Phone Login (Requires SMS setup below)

Phone Login ONLY: ☐ **Strict Mode:** Hide email and all social options. The app will **only** ask for a phone number.

Firebase Configuration

Firebase is a free service from Google that powers Push Notifications and Social Logins for your app. You can find these keys in your [Firebase Console](#) under Project Settings.

API Key:

Project ID:

App ID:

Messaging Sender ID:

Firebase Service Account JSON

This is a highly secure key that allows your website to send Push Notifications directly to your customers' phones.

How to get this: Go to Firebase Console -> Project Settings -> Service Accounts -> Click "Generate new private key". Open the downloaded file in a text editor (like Notepad), copy everything, and paste it below.

```
{
  "type": "service_account",
  "project_id": "your-app-id",
  "private_key_id": "...",
  "private_key": "-----BEGIN PRIVATE KEY-----\n...\n-----END PRIVATE KEY-----\n",
  "client_email": "...",
}
```

Automated Push Notification Triggers

Order Placed: ☒ Send a push notification to the customer's phone immediately after they place an order.

Order Status Updated: ☒ Send a push notification when an order status changes (e.g., changes to "Processing" or "Shipped").

WhatsApp API Configuration

App Settings → Integrations & APIs: login methods at the top, then Firebase and notification options.

Method	What you need	Difficulty
Email and password	Nothing, always on	None
Google	A Firebase project (chapter 8) with Google sign-in switched on, and the SHA fingerprints of your app added to Firebase	Medium
Facebook	A Facebook developer app: its name, ID and client token go into the App Builder (chapter 10)	Medium
Apple	Apple Developer account. Required by Apple if you offer other social logins in an iPhone app	Medium
Phone number (OTP)	An SMS provider: 2Factor (Indian +91 numbers only) or OTPlless	Medium
Phone Login ONLY	Hides email and social buttons: customers only enter their phone number	Easy once OTP works

Setting up phone login (OTP)

- 1 Open the **SMS / OTP** section further down in the Integrations tab.
- 2 Choose **2Factor** or **OTPlless** as the provider.
- 3 Paste the key(s) from your provider into the boxes that appear.
- 4 Tick *Enable Phone Login* at the top, then **Save All Settings**.

Important: 2Factor sends codes to Indian numbers only. For other countries choose OTPlless.

The login buttons of Google, Facebook and Apple appear in the app only when switched on *and* configured. Test each one on a real phone before you publish.

8. Set up Firebase (notifications and Google login)

In plain words: Firebase is Google's free service that delivers push notifications to your customers' phones. You create a Firebase project once, then download two files and copy a few values. It looks long, but each step is a click.

Part A: create the project and register the Android app

- 1 Go to console.firebase.google.com and sign in with your Google account.
- 2 Press **Add project**, give it your shop's name and finish the wizard (you can turn Google Analytics off).
- 3 Inside the project press the **Android** icon to add an Android app.
- 4 In **Android package name** type the package name you want for your app, for example `com.yourshop.app` . **Write it down: you will type the same one in the App Builder.**
- 5 Press **Register app**, then **Download google-services.json**. Keep this file safe. You can skip the other steps of the wizard.

Part B: copy four values into the plugin

- 1 In Firebase press the gear icon → **Project settings** → tab **General**.
- 2 Scroll to *Your apps*. Copy the **API key**, **Project ID**, **App ID** and **Messaging Sender ID** (the Sender ID is under the *Cloud Messaging* tab).
- 3 In WordPress open **App Settings** → **Integrations & APIs** → **Firebase Configuration** and paste each value into its box.

Part C: the key that lets your store send notifications

- 1 In Firebase project settings open the **Service accounts** tab.
- 2 Press **Generate new private key**. A small file ending in `.json` downloads.
- 3 Open that file with a text editor (Notepad, TextEdit), select everything and copy it.

- 4 Paste it into the big box **Firestore Service Account JSON** in the plugin and press **Save All Settings**.

The service account file is like a password to your Firebase project. Never send it by email or post it publicly.

Part D (only for Google login)

- 1 In Firebase open **Authentication** → **Sign-in method** and switch on **Google**.
- 2 After your first build, add the app's **SHA-1** and **SHA-256** fingerprints in Project settings → your Android app. For Google Play builds use the fingerprints that Google Play shows under *App integrity*.
- 3 Download `google-services.json` again and use the new file for the next build.

Keep the same package name everywhere: in Firebase, in the App Builder and (for source builds) in the app code. If they differ, notifications and Google login will not work.

9. Notifications and messages to customers

In plain words: Your app can tell customers about their orders automatically and you can send your own announcements.

Automatic order notifications

In **Integrations & APIs** tick **Order Placed** and **Order Status Updated** to send a push notification to the customer's phone whenever an order is placed or changes status (processing, shipped...).

Below, **Notification Message Templates** let you write your own wording. You can use these placeholders, which are replaced with real details:

Type	Becomes
<code>{order_id}</code>	The order number
<code>{order_status}</code>	The new status

Title: Order #{order_id} update

Message: Your order is now {order_status}. Thank you for shopping with us!

WhatsApp and SMS (optional)

- **WhatsApp:** sends order receipts and updates through Meta Cloud API, Twilio or UltraMsg. Paste the provider's token and the sender number.
- **SMS:** sends order updates with 2Factor, MSG91, Twilio or Unifonic. Tick *Order Updates via SMS* and paste the provider's key.

Send your own announcement

Mobile App API Configuration

General Settings

Drawer Menu

Legal & Policies

Support & Social

App Info & Updates

Theme (Flex)

Integrations & APIs

Send Notification

App Builder

Send Custom Push Notification

Manually dispatch push notifications to your mobile app audience.

Notification Title

Flash Sale! 🎉

Message Body

Get 20% off all products for the next 24 hours.

Target Audience

☒ All App Users
 ☐ Specific Users

Send Notification Now

Save All Settings

Send Notification: write a title and message and choose who receives it.

- 1 Open the **Send Notification** tab.

- 2 Type a **title** and a **message** (for example "Weekend sale: 20% off fruit").
- 3 Choose **All users**, or **Selected users** and search by name or email.
- 4 Press send. A green message means Firebase accepted it. A red message tells you what to fix.

Customers only get notifications if they allowed them on their phone and opened the app at least once after logging in.

10. Build your Android app online (App Builder)

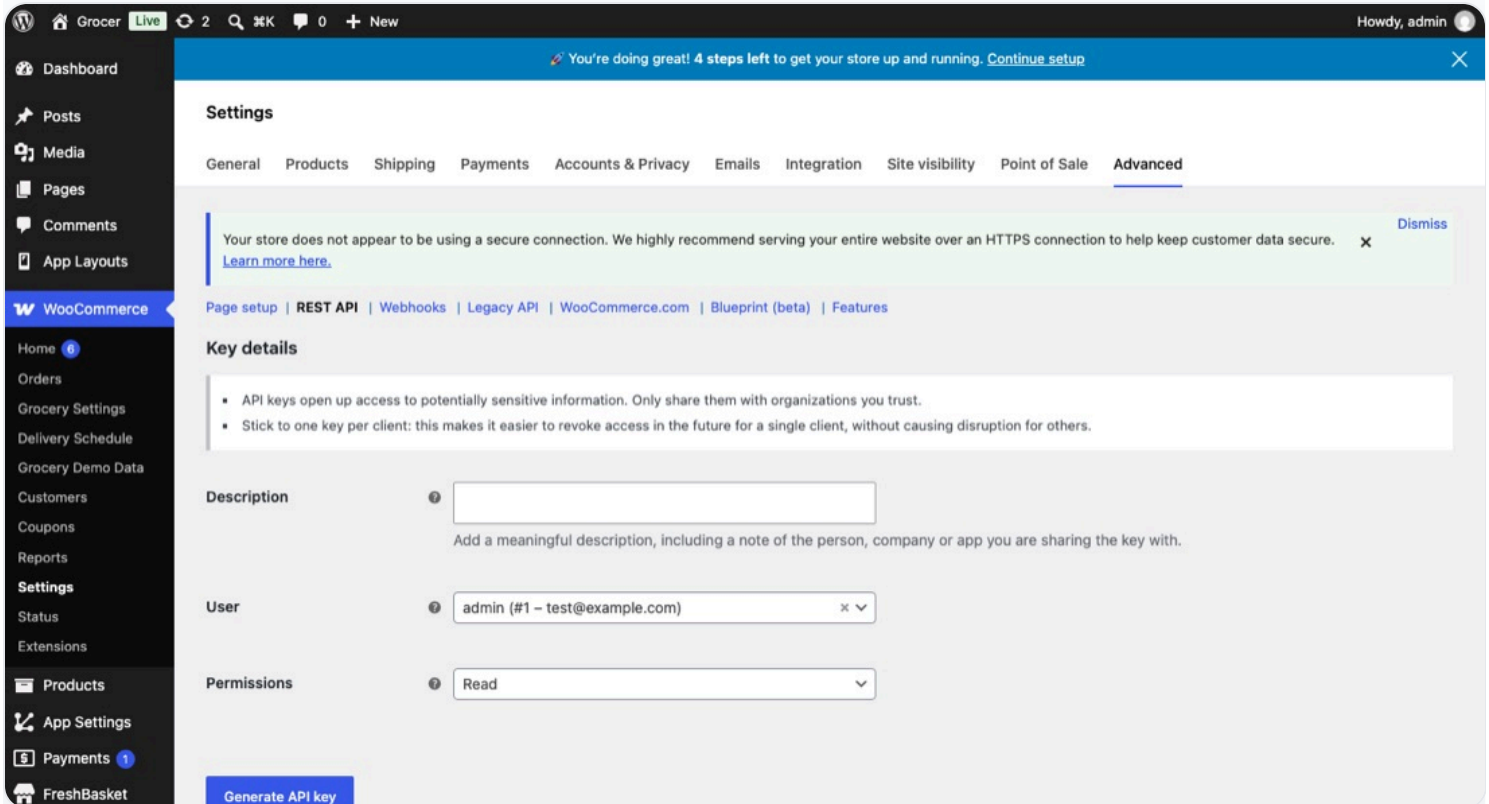
In plain words: This is the easiest way to get your app. You type your shop's details into a form inside WordPress and press one button. Our server builds the Android file and **emails you a download link**. You do not install anything.

What you need on the table first

Item	Where to get it
Your purchase code	From your purchase receipt. Each purchase code builds one app for one store address.
Your store address	For example https://yourshop.com
An app name and package name	You decide: for example "My Shop" and com.myshop.app . The package name must be the same one you typed in Firebase.
google-services.json	Downloaded in chapter 8, Part A
WooCommerce keys	Created below (2 minutes)
Three pictures	App icon (square, 1024 × 1024), a loading logo, and a splash picture
Your email address and phone number	So we can send you the download link

Create the WooCommerce keys (one time)

- 1 Open **WooCommerce** → **Settings** → **Advanced** → **REST API** → **Add key**.
- 2 Description: *Mobile app*. User: your admin account. **Permissions: Read/Write**.
- 3 Press **Generate API key**. Copy the **Consumer key** and **Consumer secret** into a safe note. The secret is shown only once.



WooCommerce → Settings → Advanced → REST API → Add key.

Fill in the App Builder

- 1 Open **App Settings** → **App Builder**.
- 2 Under **Connect to Cloud Builder** type your **Site URL** and **Purchase Code**, then press **Fetch App Details**. A green message says the connection works and the form below opens.
- 3 **Basic App Information:** app name, package name, your email, phone and two-letter country code (for example AE, IN, US).
- 4 **Security Keys:** paste the Consumer Key and Consumer Secret. Add a Google Maps key only if you want customers to pick their address on a map.
- 5 **Facebook Login Setup:** fill only if you use Facebook login.

- 6 **Branding Images:** pick the splash colour, then choose the **App Icon, Loading Logo, Splash Image** and the **Firestore Config** file (`google-services.json`). Each file uploads the moment you choose it and shows a green *Uploaded Successfully*.
- 7 Choose the **Build Output Type: APK** to test on your own phone, **App Bundle (AAB)** for Google Play.
- 8 Press **Save Details**, then **Start Android Build**.

Mobile App API Configuration

[General Settings](#)[Drawer Menu](#)[Legal & Policies](#)[Support & Social](#)[App Info & Updates](#)[Theme \(Flex\)](#)[Integrations & APIs](#)[Send Notification](#)[App Builder](#)[Save All Settings](#)

Connect to Cloud Builder

To automatically build your APK or iOS file, enter your site URL and your purchase code (from CodeCanyon or your receipt).

Site URL

Purchase Code

[Fetch App Details](#)

Basic App Information

App Name

The title that appears underneath the app icon on the customer's phone.

Package Name

A unique technical ID required by Apple and Google. Usually formatted like:
`com.yourbrandname.app`

Developer Email

Developer Phone

Country Code

Two-letter country code (e.g., US, UK, IN, AE).

Security Keys

WooCommerce Consumer
Key

Go to WooCommerce > Settings > Advanced > REST API to generate this.

WooCommerce Consumer
Secret

Google Maps API Key

App Settings → App Builder: connect, fill the form, upload the pictures and press the build button.

What happens next

Status	Meaning
Pending	Your build is in the queue. The server starts the next waiting build every few minutes.
In progress	Your app is being made.
Completed	Done. You receive an email with a download link ("Download APK" or "Download AppBundle").
Failed	You receive an email saying so. Check the picture, Firebase file and package name, then build again.

After you press the button you will see: *"Your build is processing. You will receive an email with the download link once the build is completed."* You can close the page. Check your spam folder if the email is slow. Download links are kept for a limited time (about 30 days), so download and keep your file.

Rules to know

- **All four files are required:** icon, loading logo, splash image and `google-services.json`.
- The package name must appear inside your `google-services.json`. On your very first build, if they differ, the builder adopts the package name from the file.
- One purchase code allows the number of apps stated on your licence (normally **one app for one store address**). If you see "maximum number of apps reached", contact support.
- Some licences are limited to APK builds. The builder then makes an APK even if you chose App Bundle.
- The builder makes **Android** files. For iPhone see chapter 13.
- Your uploaded pictures and Firebase file are removed from the build server after about three months.

When something goes wrong

You see	Do this
Red message after <i>Fetch App Details</i>	Check the store address (with https) and the purchase code. Make sure your hosting allows outgoing connections.
"Email and Phone are required"	Fill both fields and save again.
"Icon / Splash / Logo / Google services file is missing"	Upload the missing file and build again.
"The google-services.json file does not contain the package name"	Your Firebase app was registered with a different package name. Fix one of them so both match.
No email after a long time	Check spam. Make sure the email in the form is correct, save, and contact support with your store address.

Privacy: to build your app, the plugin sends your store address, purchase code, app name, package name, contact details, the WooCommerce keys you typed, Google Maps and Facebook values, your pictures and your Firebase file to our build server (apps.taqnix.com) over a secure connection. Nothing is sent until you press the buttons in this tab. Create a key just for the app, and delete it under WooCommerce → Settings → Advanced → REST API when you no longer need the builder.

11. Install the app on your phone and test it

- 1 Open the email on your Android phone and tap **Download APK** (or download on a computer and send the file to the phone).
- 2 Open the downloaded file. Android may ask you to **allow installing from this source**. Allow it for this one time.
- 3 Open the app. Wait a few seconds: the splash screen shows, then your home screen.
- 4 Walk through this checklist.

Test	OK?
The home screen shows your banners, categories and products	☐
Colours and name look right	☐
Open a product, choose a variation, add to cart	☐
Create an account and log in	☐
Place a test order with cash on delivery	☐
The order appears in WooCommerce → Orders	☐
You receive the order notification on the phone	☐
Every login method you switched on works	☐
Privacy policy and contact links open	☐

Test with a real phone, not only a computer. Do your first tests with cash on delivery so no real money moves.

12. Publish your app to the stores

In plain words: The stores check every app before customers can find it. You prepare a listing (name, description, pictures) and upload the app file.

Google Play (Android)

- 1 Create a [Google Play Console](#) developer account.
- 2 Create a new app and fill in the store listing: short and long description, an icon, screenshots of the app and a feature graphic.

- 3 Add the link to your **privacy policy** (required).
- 4 Fill in the content questionnaires (ads, data safety, target age). Explain that the app uses **location** to fill in the delivery address, and does not show ads.
- 5 Build an **App Bundle (AAB)** in the App Builder, upload it under *Production* → *Create release*, and send it for review.
- 6 Review usually takes from a few hours to a few days. Afterwards paste the store link into **App Settings** → **App Info & Updates**.

Google Play re-signs your app with a Play key. If you use Google login, add the **SHA-1/SHA-256 of the Play signing key** to Firebase (chapter 8, Part D).

Apple App Store (iPhone)

An iPhone app needs a Mac with Xcode and an Apple Developer account, so this part is done from the source code with a developer's help (chapter 13). Apple also requires **Sign in with Apple** if the app offers other social logins, and a privacy policy.

Keep the same package name forever and keep a copy of the signing keys. You need them for every update.

13. For developers: build the app from the source code

In plain words: Skip this chapter if you use the online builder. It is for people who want an iPhone version, want to change the app, or like to compile it themselves.

One-time setup

- 1 Install [Flutter](#) 3.41 or newer (this installs Dart too).
- 2 Run `flutter doctor` and fix every item it reports (Android SDK and licences, Xcode, CocoaPods).

3 Unzip `flutter-app-source.zip` , open a terminal in the folder and run:

```
flutter pub get
```

Point the app to your store

Open `lib/core/constants/app_config.dart` and change two lines:

```
static const String siteUrl = 'https://yourstore.com'; // no slash at the end  
static const String appName = 'My Store';
```

The API path `/wp-json/mo-app/v1` is added for you. To test with a store on your own computer use an address the phone can reach: Android emulator `http://10.0.2.2:8888/wp` , iPhone simulator `http://localhost:8888/wp` .

Name, ID, icon, splash and version

What	Command or file
Package name (both platforms)	<pre>dart run change_app_package_name:main com.yourbrand.app</pre> <p>Default is <code>com.taqnix.groceryapp</code> . Then check <i>Bundle Identifier</i> in Xcode.</p>
Name under the icon	<p>Android: <code>android:label</code> in <code>android/app/src/main/AndroidManifest.xml</code> .</p> <p>iPhone: <i>Display Name</i> in Xcode.</p>
App icon	<p>Replace <code>assets/icon/icon_android.png</code> (1024 × 1024) then run</p> <pre>dart run flutter_launcher_icons</pre>
Splash screen	<p>Replace <code>assets/images/splash.png</code> , set the colour in <code>pubspec.yaml</code> (block <code>flutter_native_splash</code>) then run</p> <pre>dart run flutter_native_splash:create</pre>
Version	<p><code>version: 1.0.0+11</code> in <code>pubspec.yaml</code> (name + build number). Raise the number after the plus sign for every store upload.</p>

Firebase files

The package has **no Firebase files**: they belong to your own project (chapter 8).

- **Android:** put `google-services.json` into `android/app/` , then add the Google services plugin. In `android/settings.gradle.kts` add inside `plugins { }` :

```
id("com.google.gms.google-services") version "4.4.2" apply false
```

and in `android/app/build.gradle.kts` add inside `plugins { }` :

```
id("com.google.gms.google-services")
```

- **iPhone:** open `ios/Runner.xcworkspace` in Xcode, drag `GoogleService-Info.plist` into the *Runner* folder and tick *Copy items if needed*.

Without these files the app still shops normally, but push notifications and Google login are off.

Facebook, Google, Apple, maps

In `ios/Runner/Info.plist` replace the placeholders:

```
<string>fbYOURAPPID</string>      <!-- keep the fb prefix -->
<key>FacebookAppID</key><string>YOURAPPID</string>
<key>FacebookClientToken</key><string>CLIENT-TOKEN</string>
```

For Android follow the `flutter_facebook_auth` package guide. Switch on Google and Apple in Firebase Authentication; Apple sign-in also needs the *Sign in with Apple* capability in Xcode. Razorpay needs no key in the app: it is read from your WooCommerce Razorpay settings.

Run and build

```
flutter run                # on a phone or emulator
flutter build apk --release # Android APK
flutter build appbundle --release # Android AAB for Google Play
flutter build ipa --release  # iPhone (on a Mac)
```

Sign the Android app for Google Play

Release builds use the debug key until you add your own. Create a key once and keep it safe:

```
keytool -genkey -v -keystore ~/upload-keystore.jks -keyalg RSA -keysize 2048 -validity 10
```

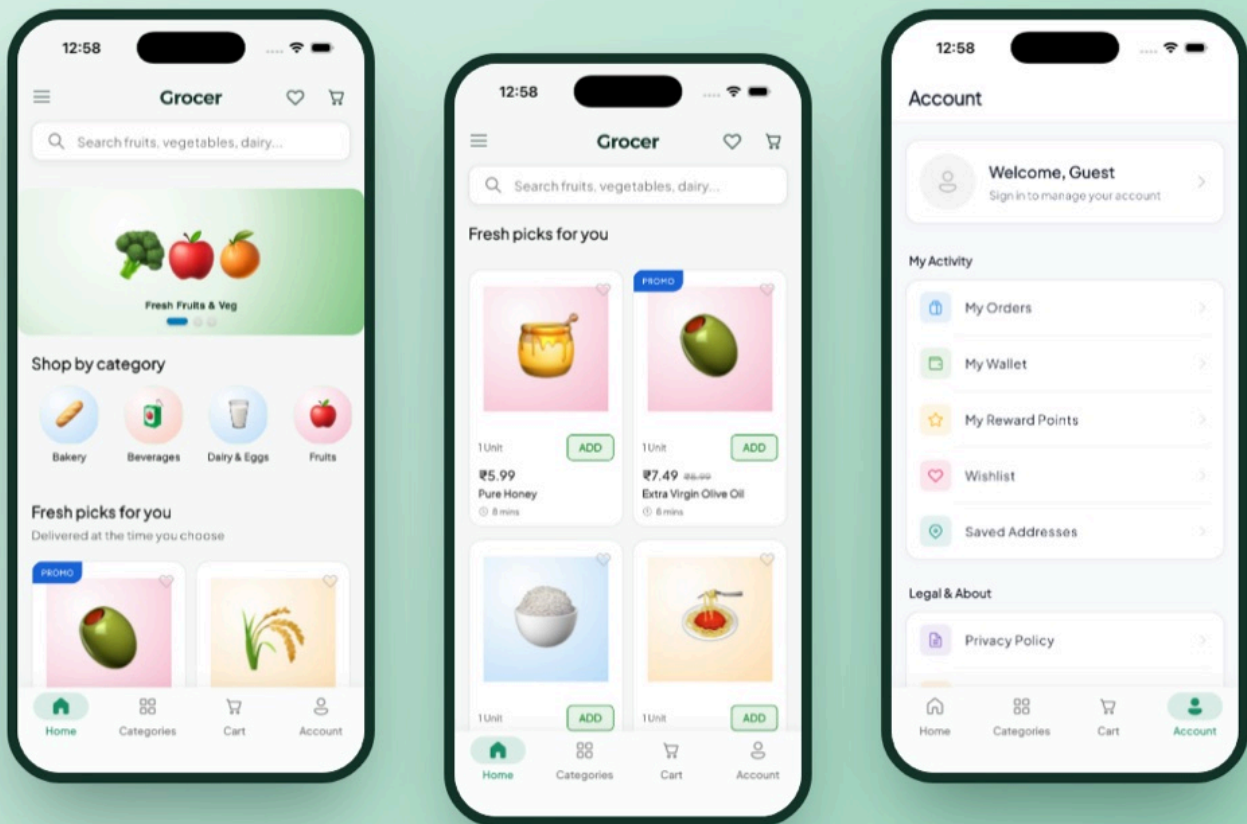
Create `android/key.properties` :

```
storePassword=YOUR_PASSWORD
keyPassword=YOUR_PASSWORD
keyAlias=upload
storeFile=/Users/you/upload-keystore.jks
```

and in `android/app/build.gradle.kts` load it and use it in `signingConfigs` / `buildTypes.release` as described in the [Flutter signing guide](#). For iPhone choose your team in Xcode → *Signing & Capabilities*.

The Android manifest already contains the INTERNET, location and notification permissions that release builds need.

14. What the app can do



Home, product list and account screens (grocery demo).

Area	What customers can do
Home	See your banners, categories and products as you designed them; pull down to refresh.
Search and filters	Search, and filter by category, price and product options (colour, size...).
Product page	Pictures, price, options, description, reviews; write a review; add to wishlist.
Cart	Change quantities, remove items, use coupon codes.
Checkout	Fill the same address fields as on your website, choose shipping and payment. Cash on delivery finishes inside the app, Razorpay opens its own payment sheet, other payment methods open a secure web page and return to the app.
Account	Log in or register, edit profile, saved addresses (with map), orders and order details, log out.
Wallet and reward points	See balance and history. Needs the plugins TeraWallet and WooCommerce Points and Rewards . Switch the menu entries on in App Settings → Drawer Menu.
Side menu and info	Privacy, terms and returns pages, contact links, forced update screen.
Notifications	Order updates and your own announcements.

15. For developers: the REST API

In plain words: Everything the app does goes through a small set of web addresses on your store. Developers can use them to extend the app or build another one.

```
GET https://yourstore.com/wp-json/mo-app/v1/homepage
```

```
{
  "global_settings": {
    "accent_color": "#ff0055",
    "currency_settings": {
      "currency_code": "INR",
      "currency_symbol": "\u20b9",
      "currency_position": "left",
      "thousand_separator": ",",
      "decimal_separator": ".",
      "number_of_decimals": 2,
      "price_display_suffix": "",
      "calc_taxes": false,
      "prices_include_tax": false,
      "tax_display_shop": "excl",
      "tax_display_cart": "excl"
    },
    "theme_settings": {
      "primary_color": "#1b8f69",
      "secondary_color": "#fa8c16",
      "surface_mode": "levelSurfacesLowScaffold",
      "blend_level": 7,
      "app_bar_style": "surface",
      "border_radius": 14,
      "button_style": "filled",
      "button_color_role": "primary",
      "button_custom_color": "#000000",
      "button_border_radius": 14,
      "use_material3": true
    },
    "login_methods": {
      "facebook": true,
      "google": true,
      "apple": true,
      "phone": true,
      "phone_only": false
    }
  },
  "layout": {
```

A piece of the answer from the home endpoint.

Base address: <https://yourstore.com/wp-json/mo-app/v1> . Logged-in calls send the header `X-App-Token` that the login call returns.

Try it

```
# The home layout and global settings
curl https://yourstore.com/wp-json/mo-app/v1/homepage

# Products: page 1, 10 items, search for "milk"
curl "https://yourstore.com/wp-json/mo-app/v1/products?page=1&per_page=10&search=milk"

# Log in and keep the token
curl -X POST https://yourstore.com/wp-json/mo-app/v1/auth/login \
  -H "Content-Type: application/json" \
  -d '{"username":"customer@example.com","password":"secret"}'

# Use the token to read the customer's orders
curl https://yourstore.com/wp-json/mo-app/v1/orders -H "X-App-Token: YOUR_TOKEN"
```

Dart example (inside the app)

```
final res = await dio.get('/products', queryParameters: {'category': 'fruits', 'per_page': 10});
final products = res.data['products'];
```

All endpoints

Endpoint	Purpose
GET /homepage	Settings (theme, currency and tax, login methods, drawer, links, versions) and the active layout.
GET /products , GET /products/{id}	Products. Parameters: page , per_page , include , search , category , attributes , min_price , max_price , orderby , order .
GET /product-attributes	Filter options, optionally for one category.
GET /products/{id}/reviews , POST /write-review	Reviews.
GET /categories	Categories with pictures and counts.
POST /auth/register , /auth/login , /auth/logout	Email and password accounts.
POST /auth/send-otp , /auth/verify-otp	Phone login with a one-time code.
POST /auth/social-login	Google, Facebook or Apple login with the provider's token.
GET/POST /auth/profile , POST /auth/fcm-token	Profile and the phone's push token.
GET /cart , POST /cart/add , /cart/update , /cart/remove , /cart/apply-coupon , /cart/remove-coupon	Cart.
GET /checkout/form , POST /checkout/update-review , POST /checkout	Checkout fields, totals and placing the order.
GET /orders , GET /orders/{id}	The customer's orders.

GET/POST /addresses , PUT/DELETE
/addresses/{id}

Saved addresses.

GET /wishlist , GET /wishlist/details ,
POST /wishlist/toggle

Wishlist.

GET /wallet , GET /points

Wallet and points (extra plugins required).

The plugin's `includes/api` folder has one file per group with the exact parameters.

16. Problems and fixes

In plain words: Find what you see in the left column.

What you see	What to do
The app shows a loading error or no products	Open <code>https://yourstore.com/wp-json/mo-app/v1/homepage</code> in a browser. It must load without a login. If not: use https, set permalinks to anything but Plain, set the store to Live, and check that no security plugin or firewall blocks <code>/wp-json/</code> .
The web address says "Active App Layout not configured"	App Settings → General Settings → Active App Homepage: choose a layout and save.
I changed a colour or the layout and the app did not change	Press Save All Settings (or Update on the layout), then pull down to refresh or reopen the app.
The home screen is empty	The layout has no blocks, or the categories and products block found nothing. Add products and check categories are not empty.
The banners are grey	The banner picture addresses are wrong, or the pictures are not public. Re-add them from the Media Library.
Notifications do not arrive	Check the Firebase Service Account JSON is pasted in full, the package name is the same everywhere, the phone allowed notifications, and the customer logged in once.
Google login fails	Add the SHA-1 and SHA-256 of the signing key to Firebase, download <code>google-services.json</code> again, rebuild.
The OTP code does not arrive	2Factor works for Indian numbers only. Use OTPless for other countries. Check your provider's balance and key.
Wallet or points show nothing	Install and activate TeraWallet and WooCommerce Points and Rewards.
Fetch App Details fails	Check address and purchase code; see chapter 10.
My release app cannot connect (own build)	The Android manifest needs the INTERNET permission. It is included; keep it if you replace the file.

iPhone build fails in Xcode

Run `cd ios && pod install --repo-update`, add `GoogleService-Info.plist`, choose a signing team and a unique bundle ID.

The cart or checkout looks odd in the app

The checkout form comes from WooCommerce. Check your checkout fields and payment methods in WooCommerce → Settings.

17. Updating and removing

Update the plugin by uploading the new zip (**Replace current with uploaded**); your layouts and settings are kept. Update the app by building a new version and uploading it to the stores. To remove the plugin, deactivate and delete it; your app layouts stay in the database until you delete them from **App Layouts**.

18. Support

Installation help and bug fixes are included for 6 months, with lifetime updates. Contact us through the marketplace item page where you purchased the package. When you write, include your store address and what you see on the screen.

Changelog

1.0.2 – Android push fix

- Flutter app: the Firebase Gradle plugin is now applied automatically when `android/app/google-services.json` is present, so push notifications work in Android builds made from the source code (builds without the file still work)
- Version number bumped in the package, documentation and readme

1.0.1 – Illustrated guide

- New step-by-step documentation for non-developers: 18 chapters with screenshots of every settings tab, Firebase set-up, the online App Builder and publishing to the stores
- Code snippets for developers: app configuration, Gradle and Xcode set-up, signing, build commands and REST API examples
- The guide now explains how builds are queued and delivered by email, the required files and the licence limits
- Plugin 1.0.28: the App Builder sends the purchase code with every request (required by the updated build server)

1.0.0 – Initial release

- Flutter shopping app for Android and iOS: home from your layout, categories, search and filters, product pages with reviews, cart and coupons, checkout, orders, addresses, wishlist, drawer menu, forced update
- WordPress plugin 1.0.28: App Layouts with Banners, Categories and Products blocks and per-screen app bar settings, App Settings for Material 3 theme, login methods, drawer, legal and support links, store links and minimum version
- Login with email, Google, Facebook, Apple and phone OTP (2Factor, OTPless); Firebase push (FCM v1), custom notifications, WhatsApp and SMS order messages
- App Builder tab: build your Android APK or App Bundle online from the WordPress dashboard
- Documented REST API under /wp-json/mo-app/v1 and complete Flutter source with the Android release permissions added
- Security: App Builder actions require an administrator and a nonce