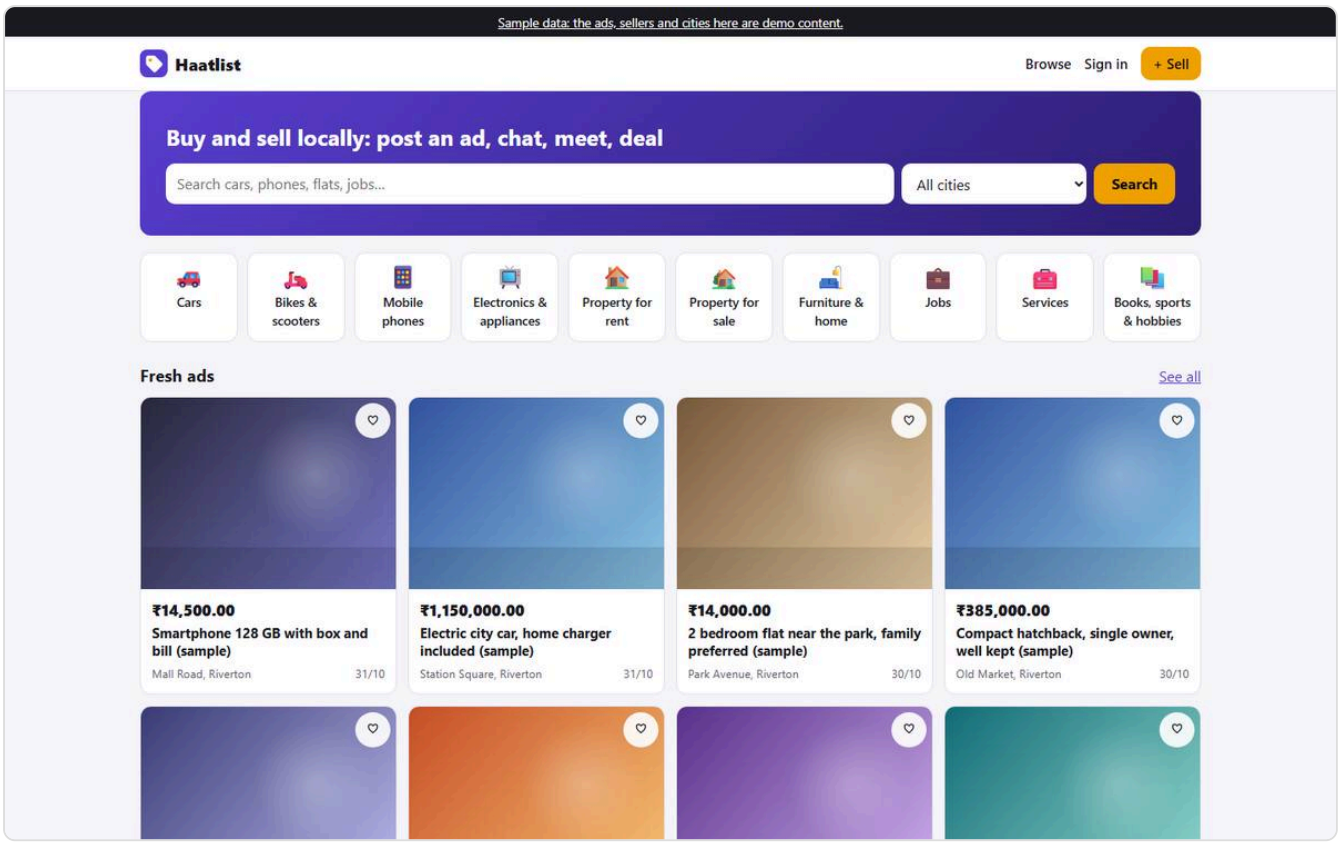
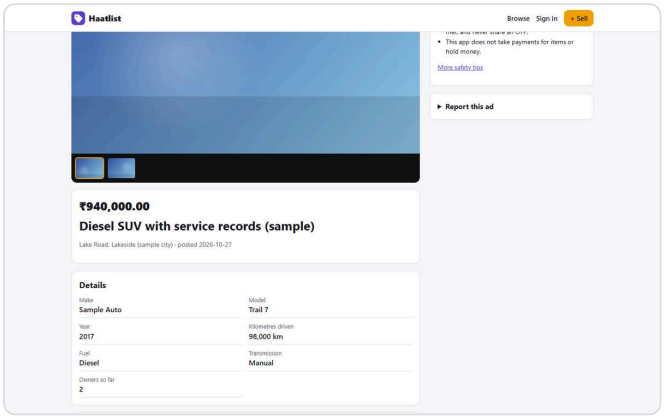
CODE CRAFT · INSTALLATION GUIDE

Haatlist Classifieds App

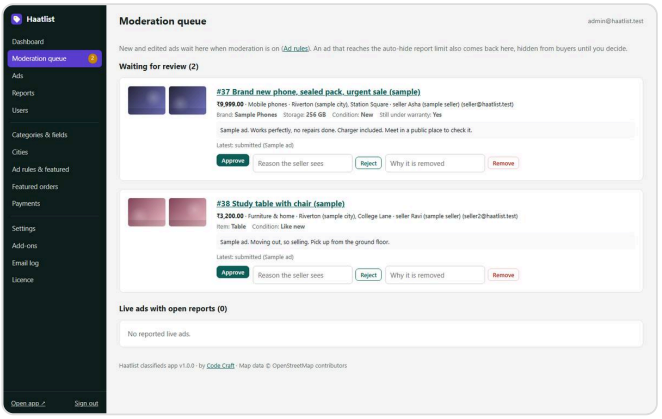
Version 1.0.0: a local buy-and-sell site with a customer web app, an admin with a moderation queue, a REST API and an Expo (React Native) mobile app. Node.js, Express, EJS and Prisma. This guide takes you from download to a live site on Hostinger, cPanel or a VPS.



The customer web app with the sample data. The name, logo, colours, categories and cities are all changed from the admin.



An ad page: photos, price and the category's own fields.



Admin moderation queue: approve, reject with a reason, or remove.

1. What is in the download

Folder / file	What it is
`server.js`	Start file. Hostinger, cPanel and PM2 run this.
`src/app.js`, `src/routes/`	The Express app: `web.js` (customer web app), `admin.js` (admin), `api.js` (REST API /api/v1).
`src/lib/`	Ads (validation, status, search, expiry, reports, moderation), category fields, photo checks, chat, featured ads, maps and distances, settings, licence, mailer, add-ons, UPI links.
`views/`, `public/`	EJS templates, CSS, small browser scripts, icons and the sample placeholder photos. Uploaded photos go to `public/uploads/`.
`prisma/`	`schema.prisma` (database tables) and `seed.js` (sample data).
`scripts/`	`setup.js` (creates .env, tables and sample data) and the placeholder-photo generator.
`addons/`	Drop add-on folders here (none are included).
`mobile/`	The Expo mobile app with its own README.
`test/`	8 unit tests, 16 flow tests and `api-smoke.sh`.
`README.md`, `API.md`	Reference documentation (every REST endpoint is in API.md).

Requirements

- Node.js 18.18 or newer.
- Database: nothing extra for SQLite (default). For MySQL 8 or PostgreSQL 13+, an empty database and a user.
- A domain with https for the live site.
- For the mobile app: Node.js on your computer, and for store builds an Expo account plus your own Android and Apple developer accounts.

2. Run it on your computer

1. Unzip the download and open a terminal in the folder.
2. Install and set up:

```
npm install
npm run setup      # creates .env with a random secret, the tables and the sample data
npm start          # http://localhost:3490
```

- Web app: `http://localhost:3490`
- Admin: `http://localhost:3490/admin`
- API: `http://localhost:3490/api/v1`

`npm run setup -- --no-seed` creates an empty app: the first visit to `/admin` then runs the setup wizard that creates your admin account. `npm run setup -- --reseed` wipes everything and loads the sample data again. `npm test` runs the 24 automated tests.

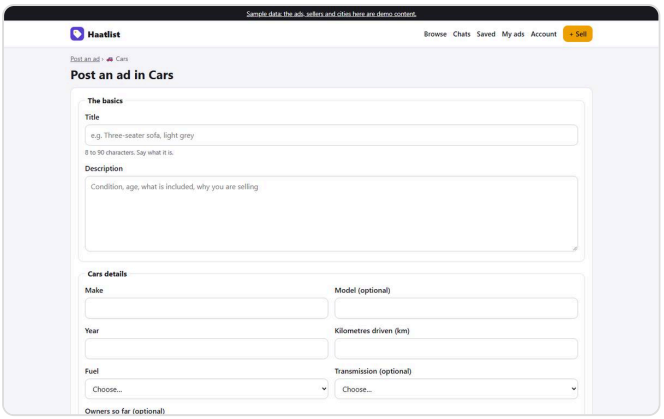
Demo logins (sample data only)

All demo accounts use the password `Demo@12345`. Change or delete them before going live.

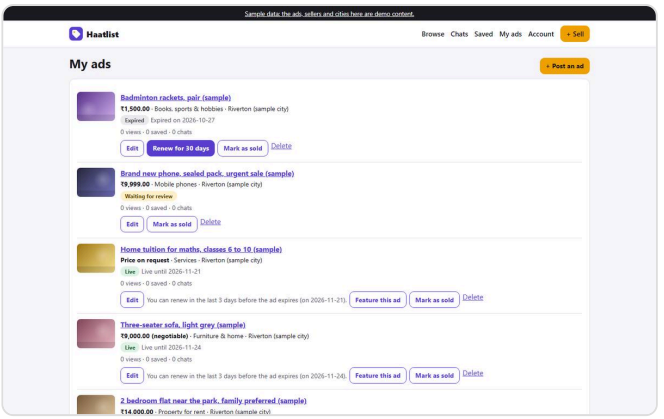
Role	Email	Where
Admin	`admin@haatlist.test`	`/admin`, or moderation mode in the mobile app
Seller (8 live ads, 1 waiting, 1 expired)	`seller@haatlist.test`	Web app or mobile app
Seller (8 live ads, 1 waiting, 1 sold)	`seller2@haatlist.test`	Web app or mobile app
Buyer	`buyer@haatlist.test`, `buyer2@haatlist.test`	Web app or mobile app

> The sample cities (Riverton, Lakeside, Hillcrest), ads and photos are invented placeholders. No chats, reports, views or payments are created by the seed.

3. How the app works

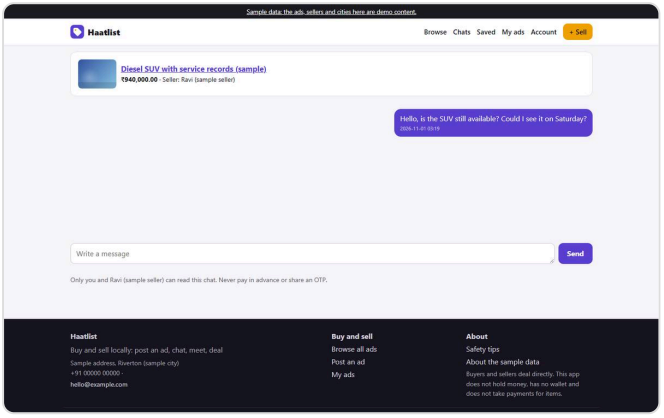


Posting an ad: the basics plus the category's own fields.

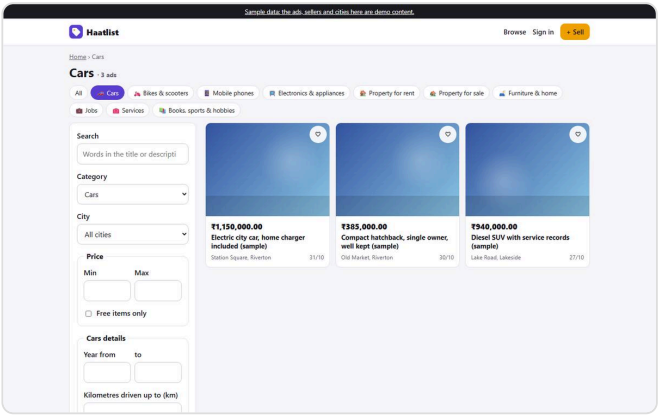


My ads: edit, renew, feature, mark as sold or delete.

1. A seller picks a category and fills in the title, description, price, the category's fields, the city (and optionally a map pin) and up to 8 photos. The server checks everything again.
2. With moderation on (the default) the ad waits in **Admin → Moderation queue**. Approve it (it goes live for 30 days) or reject it with a reason; the seller can fix and resend it. Editing a live ad sends it back to the queue (switchable).
3. Buyers find it with search and filters, save it and chat with the seller. Anyone signed in can report it; at 3 open reports (switchable) a live ad is hidden until you approve it again, remove it or dismiss the reports.
4. Three days before expiry the seller gets an email and can renew; an expired ad can be renewed any time, up to 12 renewals (switchable).
5. A seller can pay you to feature a live ad (UPI or bank transfer). It shows in the Featured strip on the home page and matching searches for the plan's days.

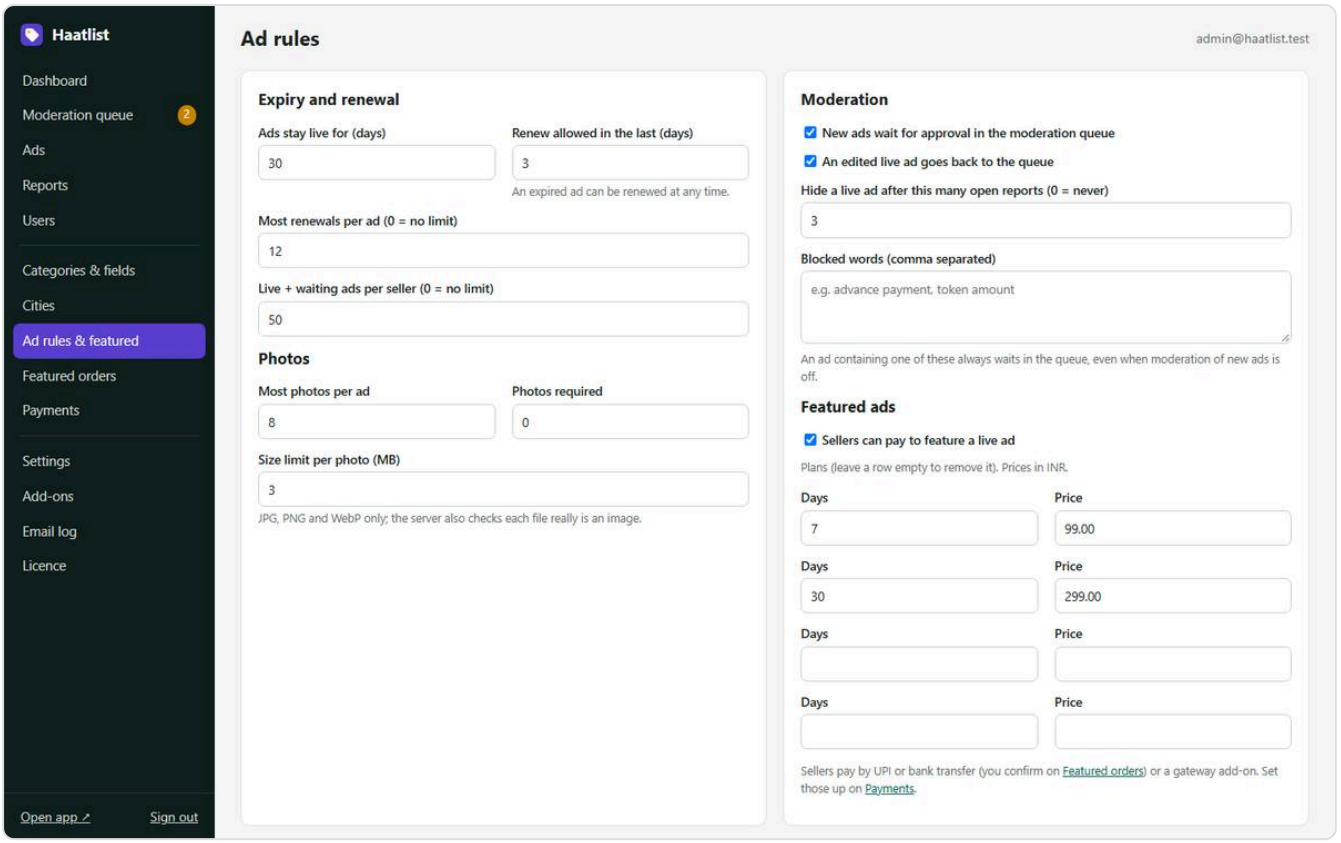


Chat: only the buyer and the seller can read it, not even the admin.



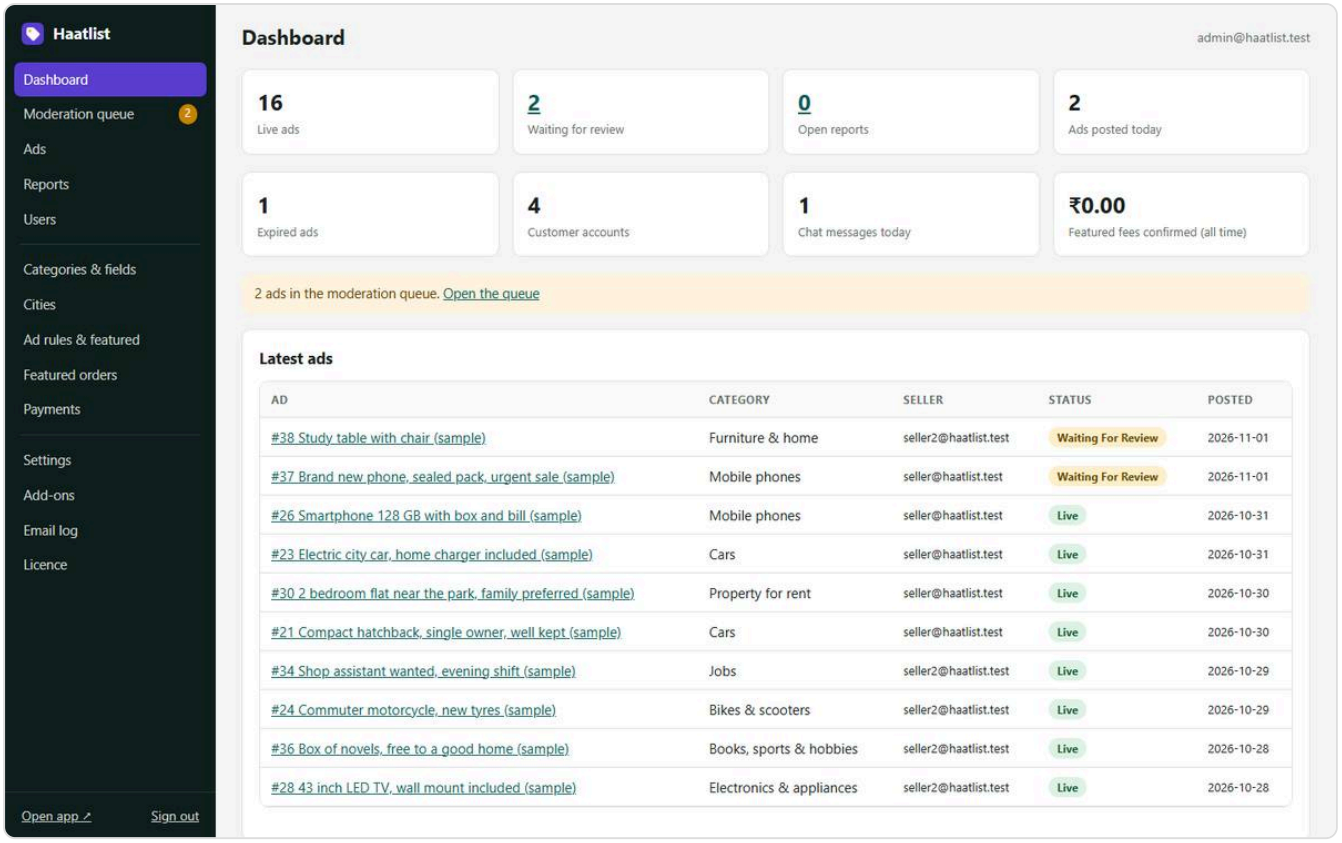
Search with price, free items and the category's own filters.

4. Set it up in the admin



Admin → Ad rules & featured: lifetime, renewals, photo limits, moderation, blocked words and featured plans.

1. **Settings → Business & brand:** name, logo, colours, contact details and map centre.
2. **Categories & fields:** keep, edit or replace the sample categories. Each has a list of fields (text, number, select, yes/no) with rules and an optional search filter.
3. **Cities:** your cities or areas with a centre point.
4. **Ad rules & featured:** how long ads stay live, the renewal window, photo limits, moderation, blocked words, auto-hide after reports, featured plans and prices.
5. **Payments:** your UPI ID and bank details (only used for the featured fee). The seeded values are samples.
6. **Settings → Email (SMTP):** until it is set, emails are only written to the Email log.
7. **Licence:** paste your key on the live domain. Until it is active, posting new ads is switched off. One key covers one business on one live domain at a time. The app checks the key with coodecraft.com once a day and keeps a signed token, so it keeps working offline for at most 14 days. You can move the key to another domain up to 3 times a year (deactivate it on the old domain first). Local preview (`localhost`, `127.0.0.1`, `*.localhost` and `*.test`) needs no key.



Admin dashboard: every number comes from what people do in the app.

5. Put it online

Hostinger (Node.js Web App) or cPanel "Setup Node.js App"

1. Upload the folder (without `node_modules`, `.env` or `prisma/*.db`).
2. Set the start file to `server.js` and the Node version to 18 or newer.
3. Add the environment variables below (`TRUST_PROXY=true`, your `APP_URL`, a long `APP_SECRET`, your `DATABASE_URL`).
4. Run `npm install` and `npm run setup` from the panel's terminal, then start the app.

VPS with PM2 and Nginx

```
npm ci && npm run setup
pm2 start server.js --name haatlist && pm2 save
```

Point an Nginx `server` block at `http://127.0.0.1:3490` with `proxy_set_header X-Forwarded-Proto $scheme;` and `client_max_body_size 30m;` (photo uploads), and set `TRUST_PROXY=true`.

Key	What it is
`DATABASE_URL`	`file:./app.db` (SQLite, default), `mysql://…` or `postgresql://…`. After changing it, run `npm run setup`.
`PORT`	Default 3490 (Hostinger and cPanel set it for you).
`APP_SECRET`	Long random string that signs session cookies.
`APP_URL`	Your public https address, used in emails and payment return links.
`TRUST_PROXY`	`true` behind https on Hostinger, cPanel or Nginx.
`LICENSE_KEY`	Your licence key (also in Admin → Licence).
`UPLOADS_DIR`	Optional folder for ad photos (default `public/uploads`).

Back up the database (`prisma/app.db` for SQLite) and `public/uploads/` regularly.

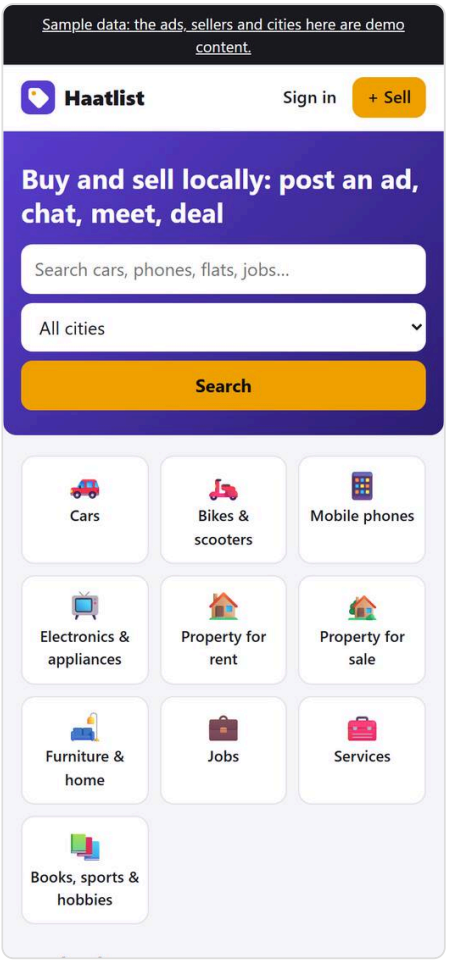
6. The REST API and the mobile app

The API at `/api/v1` uses bearer tokens from `/auth/login` (stored hashed, valid 90 days, revoked on sign-out, password change or when an admin blocks the account). It covers app info, categories, cities, search, posting with photos, renew, sold, delete, favourites, reports, chat, featured orders and the admin moderation queue. Every endpoint is documented in `API.md`.

The Expo app in `mobile/` has two modes, picked by the account that signs in: **buy and sell** (browse, saved, chats, my ads, account, posting with photos, featured payments) and **moderation** for admins.

```
cd mobile
cp .env.example .env   # set SERVER_URL
npm install
npx expo start
```

Set `APP_NAME`, `APP_SLUG`, `APP_SCHEME`, `BUNDLE_ID` and `BRAND_COLOR` in `mobile/.env` and replace the images in `mobile/assets/`. Store builds are made with EAS; the steps are in `mobile/README.md`.



The web app at phone width.

7. Before you go live, and honest limits

Checklist

- Delete or change the demo accounts and the sample categories, cities and ads (or start with `--no-seed`).
- Set your UPI ID and bank details in Admin → Payments, and SMTP in Settings.
- Switch map tiles to your own provider (see below).
- Write your site rules, terms and privacy policy.
- Activate the licence on your live domain.

Limits

- **Mobile app:** its TypeScript check passes and its web build was tried against the server. Native Android and iOS builds have not been run; build them with EAS and test on real phones before publishing.
- **Maps:** Leaflet with OpenStreetMap's public tiles, which are not meant for production traffic. Change `TILE` in `public/js/app.js` to your own tile or map provider before launch. Ad pages show an approximate circle only; distances are straight-line.
- **No wallet, no money held:** buyers and sellers settle sales themselves. A prepaid wallet needs a licence (in India, an RBI PPI licence).
- **Payment gateways** are add-ons and none are included. Add-ons in this product family are tested only against mocked responses; test any gateway in its sandbox first.
- **Chat** updates by polling every 5 seconds; there are no push notifications.
- **Moderation** is done by people; blocked words and auto-hide are simple rules, and photos are checked to be real image files, not for their content.
- Our tests ran on SQLite; MySQL and PostgreSQL are supported through Prisma.

Support: **coodecraft.com/contact**. Packages with setup include installation on your hosting by us.