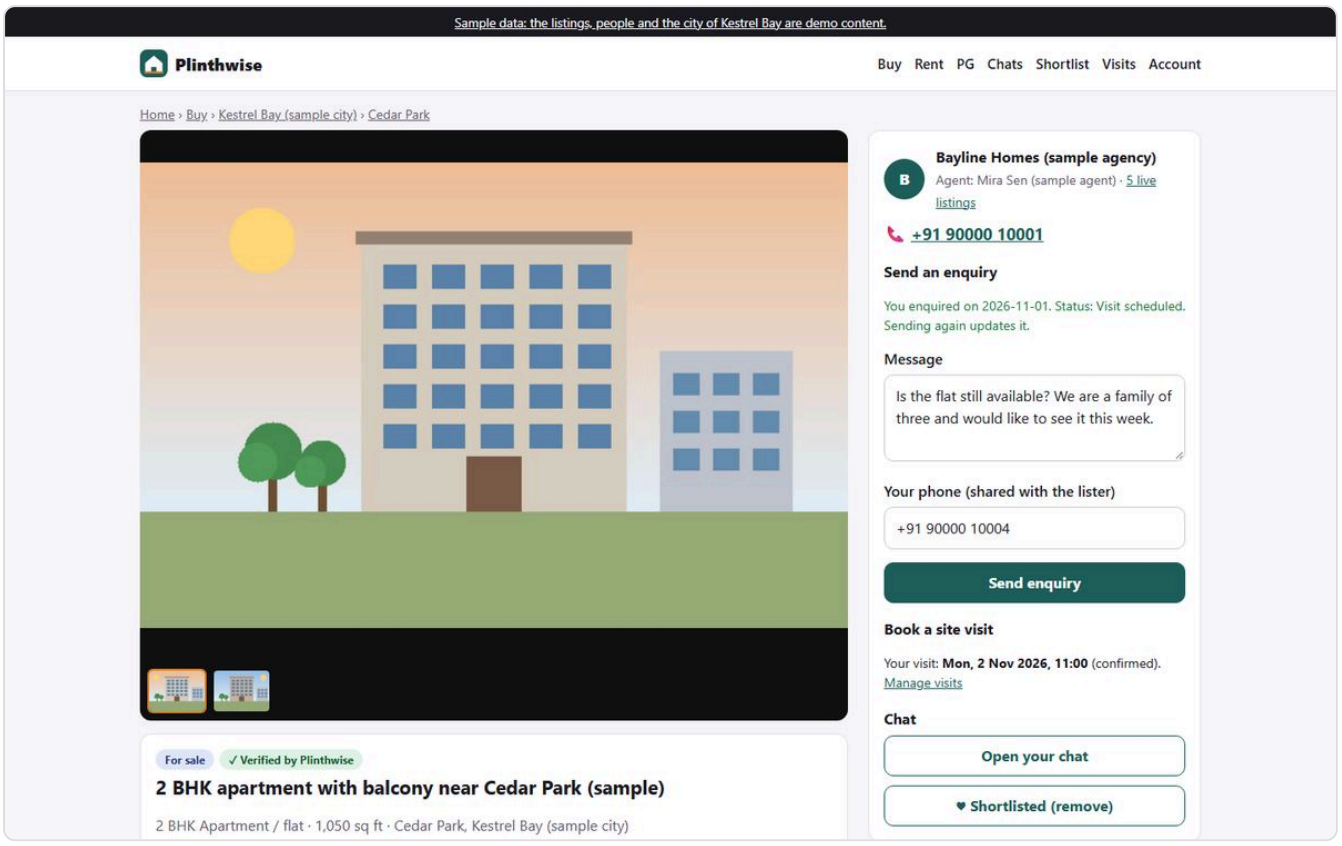
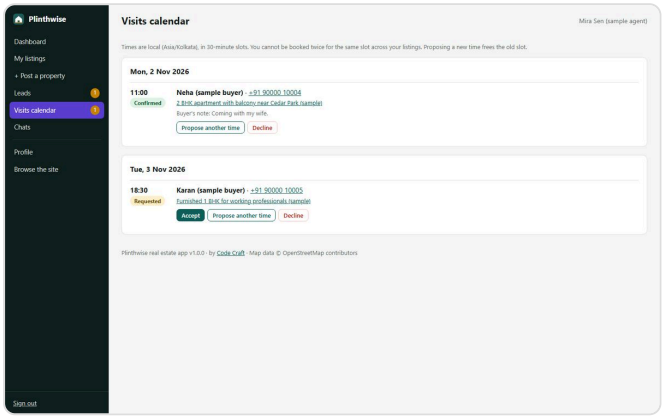
CODE CRAFT · INSTALLATION GUIDE

Real Estate App

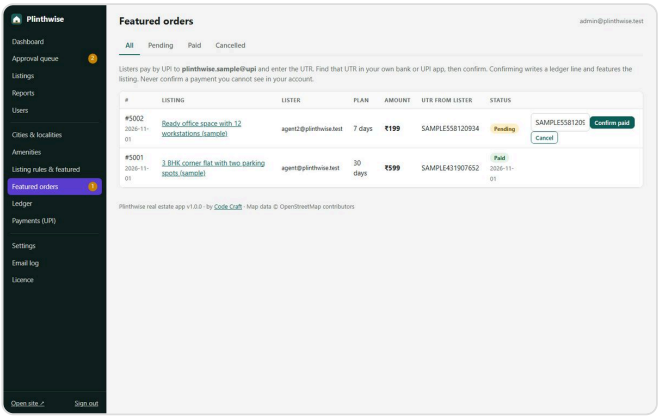
Plinthwise 1.0: a property listings web app for buying, renting and PG / co-living, a panel for agents and owners, an admin panel with moderation, a verified badge and featured listings, a REST API and an Expo mobile app, all in one Node.js project. This guide takes you from download to a live property portal on Hostinger, cPanel or a VPS.



A listing page as a signed-in buyer: drawn sample pictures, the lister, an enquiry, a confirmed site visit, chat and shortlist.



Agent: the visits calendar.



Admin: featured-listing payments by UPI.

No legal or title checks. The app does not verify titles, ownership, approvals, registrations or RERA numbers, and makes no such claim. RERA numbers are shown only as entered by the lister. No home loans, valuations or legal advice, and no money is held. "Plinthwise", the cities, listings and people in the demo are sample data, and the pictures are drawn placeholders.

1. What is in the download

Folder / file	What it is
`server.js`	Start file. Hostinger, cPanel and PM2 run this.
`src/routes/`	`web.js` (buyer web app), `partner.js` (agent / owner panel at `/partner`), `admin.js`, `api.js` (REST API).
`src/lib/`	Listing rules, moderation, expiry, search and the verified badge (`listings.js`); enquiries and the hidden-phone rule (`leads.js`); visit slots without double-booking (`visits.js`); private chat (`chat.js`); featured orders, UPI and the ledger (`featured.js`); settings, photos, mailer and licence.
`views/` , `public/`	EJS templates, CSS, scripts, the drawn sample pictures and the empty `uploads/` folder for listing photos.
`prisma/`	Database schema and the sample-data script.
`mobile/`	The Expo (React Native) app with a buyer mode and a lister mode.
`test/`	19 unit and full-flow tests.

Requirements

- Node.js 18.18 or newer (20 or 22 LTS recommended).
- Database: nothing extra for SQLite (default). MySQL 8+ or PostgreSQL 14+ use the same Prisma schema; only SQLite was tested.
- A domain with https, a writable `public/uploads/` folder, SMTP for emails and your own map tile provider for production.
- For the mobile app: an Expo account, and Google Play / Apple developer accounts to publish.

2. Run it on your computer

```
npm install
npm run setup      # .env with a random APP_SECRET, the tables and the sample data
npm start          # http://localhost:3560
```

Address	What opens
`http://localhost:3560`	Buyer web app
`/partner`	Agent / owner panel (sign in at `/account/login`)
`/admin`	Admin panel
`/api/v1`	REST API (see `API.md`)

`npm run setup -- --no-seed` creates empty tables and the first visit to `/admin` runs the setup wizard. `npm run setup -- --reseed` wipes everything and loads the sample data again.

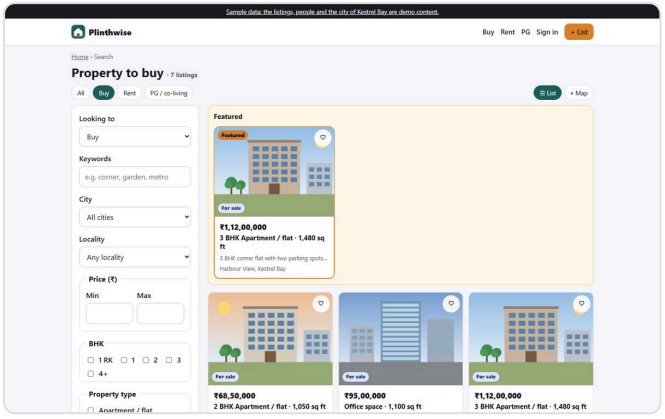
Demo logins (sample data, on your own copy only)

Role	Email	Password
Admin	`admin@plinthwise.test`	`Demo-Admin-2026!`
Agent (sample agency)	`agent@plinthwise.test`	`Demo-Agent-2026!`
Agent (second sample agency)	`agent2@plinthwise.test`	`Demo-Agent-2026!`
Owner	`owner@plinthwise.test`	`Demo-Owner-2026!`
Buyer	`buyer@plinthwise.test`	`Demo-Buyer-2026!`

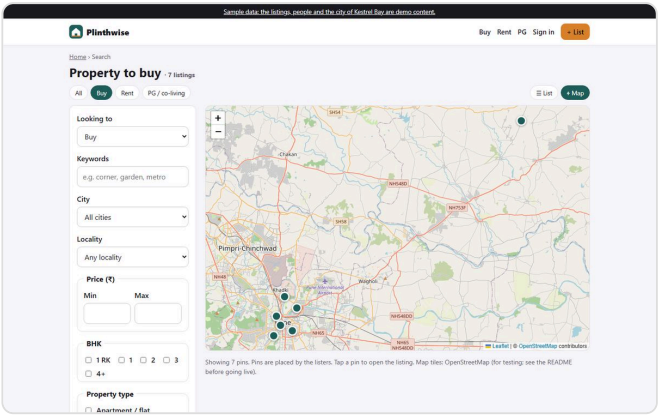
The sample data has 17 listings in two invented cities: 13 live (7 for sale, 4 for rent, 2 PG), 2 waiting for approval, 1 expired and 1 sold. It has no enquiries, chats, visits or payments, and no invented view counts, ratings or reviews. The screenshots in this guide were taken after a few test enquiries, visits, a chat and featured orders were made.

3. Buyers

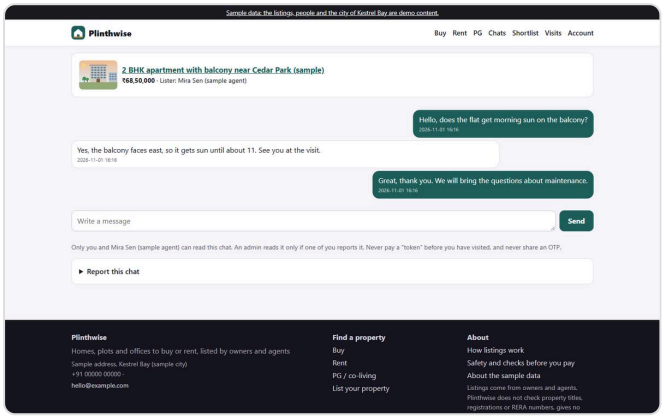
1. **Search** by buy, rent or PG, city, locality, price or rent range, BHK, property type, furnishing, area range, amenities, verified only and keywords, as a list or on a map. Featured listings are labelled.
2. **Listing page:** photos, facts, amenities, the map pin, the price per area worked out from the lister's own price and area, and RERA numbers "as entered by the lister" with a note to check them on the state RERA website.
3. **Enquiry and phone number:** with "hide my phone" on (the default), the lister's phone is shown only to buyers who have sent an enquiry. It is never shown to visitors who are not signed in.
4. **Site visits:** pick a date and a free 30-minute slot. The lister accepts, declines or proposes another time. One lister can never be booked twice for the same slot across all their listings, even by two requests at the same moment.
5. **Chat** with the lister, one thread per buyer and listing. Only the two of them can read it; an admin can read it only after one of them reports it. Buyers can also shortlist and report listings.



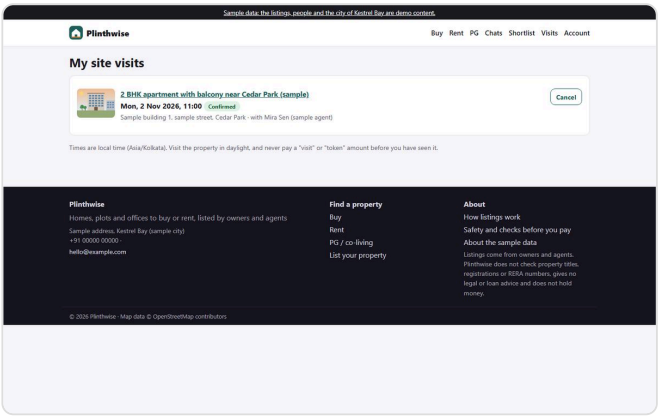
Search with filters.



The same search on the map (OpenStreetMap tiles for testing).



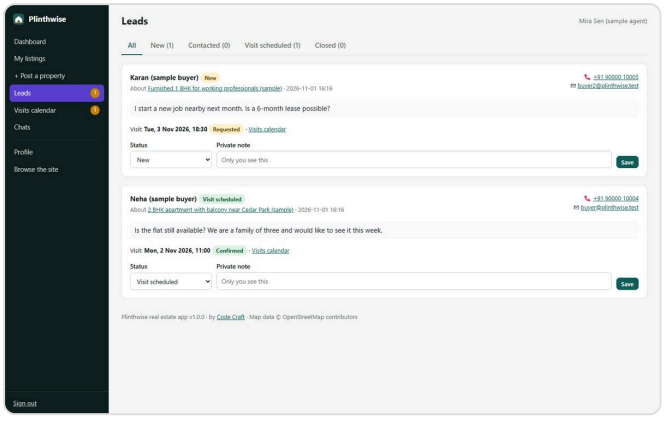
Buyer and lister chat.



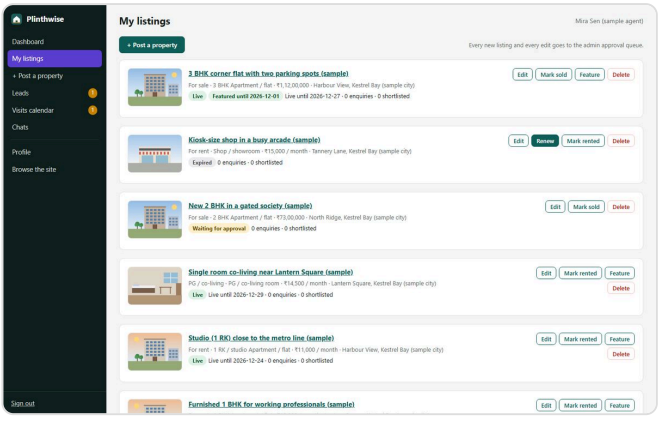
A buyer's site visits.

4. Agents and owners

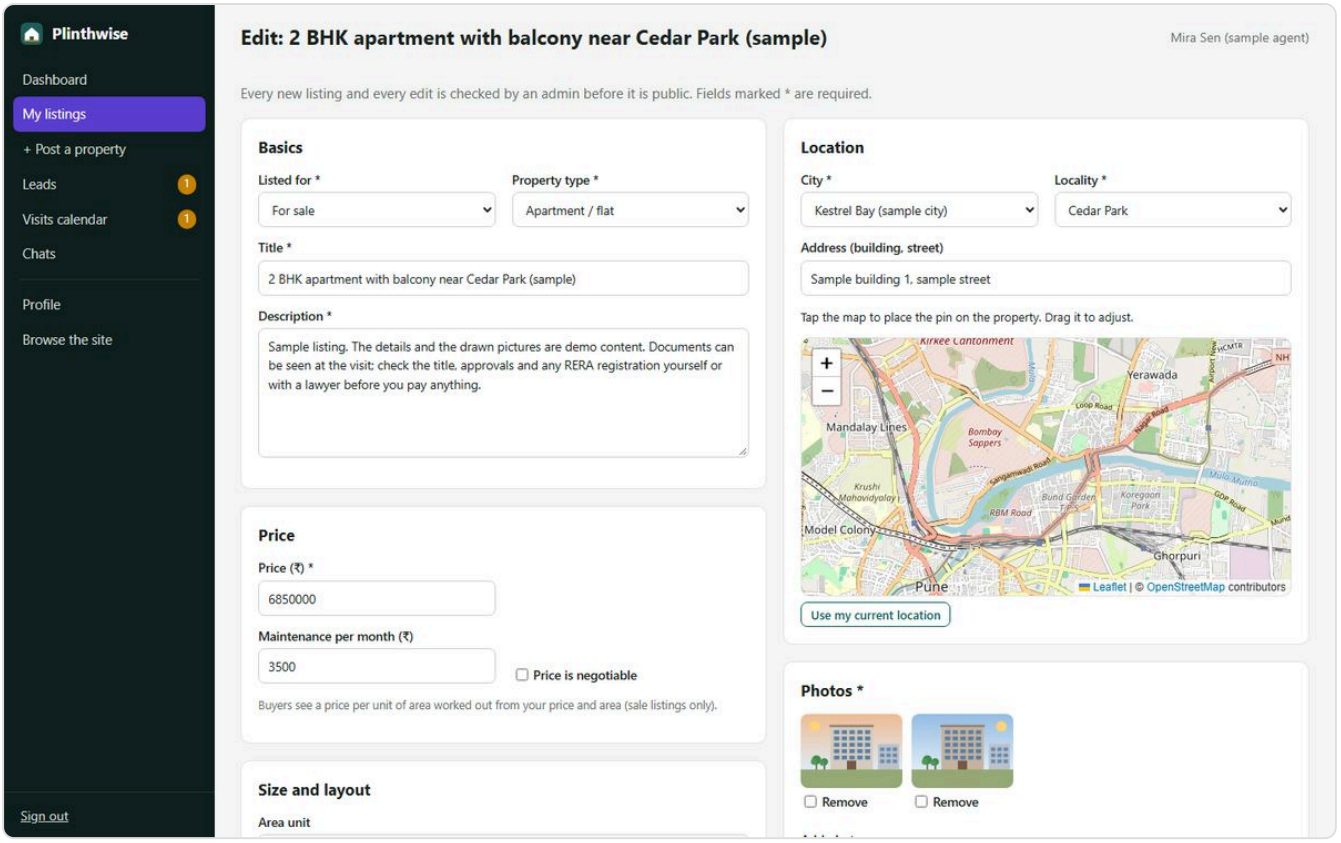
1. **Post a listing:** buy, rent or PG; property type; price or monthly rent with deposit and maintenance; built-up, carpet or plot area with a unit; BHK, bathrooms, floor, facing, furnishing, age and available-from date; amenities; address, city, locality and a map pin; a project RERA number if any; up to 12 photos (JPG, PNG or WebP, checked by their real bytes).
2. **Approval:** every new listing and every edit goes to the admin's approval queue. Nothing is public before an admin approves it, and an edit removes the verified badge.
3. **Expiry:** listings stay live for 60 days (you can change this), a reminder goes out 7 days before, and the lister can renew. Mark a property sold or rented, which also cancels its open visits.
4. **Leads:** each enquiry is a lead (new, contacted, visit scheduled, closed) with a private note.
5. **Visits calendar:** accept, decline, propose another time or mark done.
6. **Featured listing:** pick a plan, pay the site's UPI ID by link or QR and enter the UTR. The admin confirms it by hand.



Leads with status and private notes.



My listings with renew and featured.



The listing form with the map pin and photos.

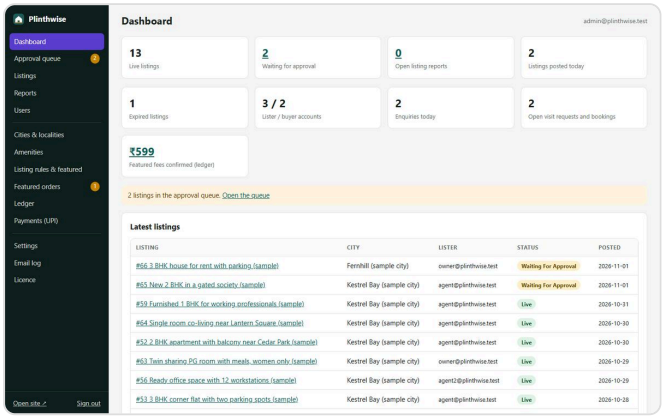
5. Admin and going live

The admin approves or rejects listings (a reason is required), removes or restores them, sets the verified badge with a note on what was checked, handles reports on listings and chats, manages users, cities and localities, amenities, listing rules and featured plans, confirms featured payments after matching the UTR, and keeps a ledger with CSV export. Settings cover business details and brand, the map centre and time zone, live chat, WhatsApp, analytics, banners, a cookie notice and SMTP.

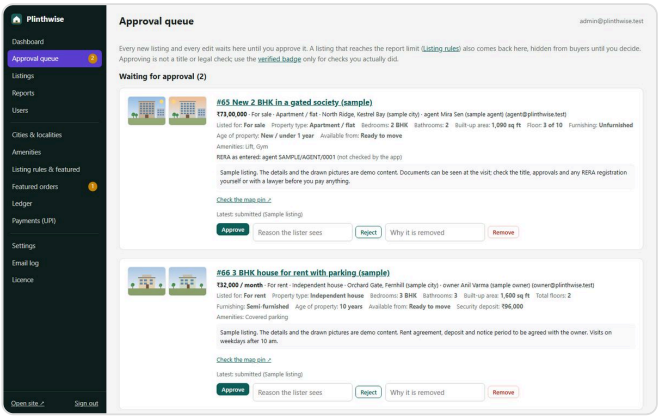
1. Upload the files without `node_modules`, `.env`, `prisma/*.db` and the contents of `public/uploads/`.
2. Node 20 or 22, startup file `server.js` (Hostinger Node.js app, cPanel "Setup Node.js App" or PM2 on a VPS).
3. Environment: `DATABASE_URL`, `APP_SECRET`, `APP_URL=https://your-domain`, `TRUST_PROXY=true` and optionally `UPLOADS_DIR`.
4. Run `npm install` and `npm run setup -- --no-seed`, then open `/admin` and add your cities, localities and amenities, your UPI ID and SMTP.
5. Add your licence key in Admin → Licence. On a real domain posting new listings stays switched off until it is active. 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.

```
npm ci && npm run setup -- --no-seed
pm2 start server.js --name plinthwise && pm2 save
```

Behind Nginx, set `client_max_body_size 80m;` for photo uploads. Back up the database and `public/uploads/`.



Admin dashboard.



Admin: the approval queue.

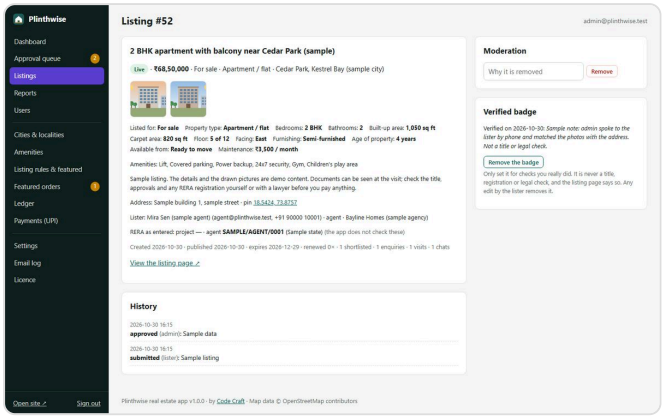
6. Verified badge, payments and maps

Verified badge: set only by an admin on the listing, always with a note on exactly what was checked (for example "spoke to the owner by phone; photos match the address"). Buyers see the note and a line saying it is not a title, registration or legal check. Any edit by the lister removes it.

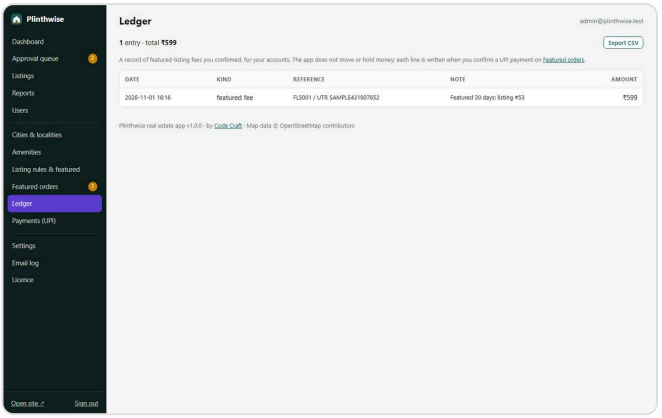
Payments: no wallet, no deposit escrow and no rent or price paid through the app. The only payment the app knows about is the optional featured-listing fee, paid by UPI straight to your UPI ID. The admin checks the UTR in their own bank or UPI app and confirms; confirming writes one ledger line and features the listing, and a second confirm is refused. No gateway is included.

Maps: Leaflet with OpenStreetMap's public tiles, which are for testing and light use only. Change `TILE` in `public/js/app.js` to your own provider before you launch and follow its attribution rules.

Local law: running a property portal may need registrations or disclosures (for example rules for advertising projects or for agents), and you need a privacy policy for the contact details people share. That is the site owner's duty.



Admin: a listing with the verified badge and its note.



Admin: the ledger of confirmed featured fees.

7. REST API and the mobile app

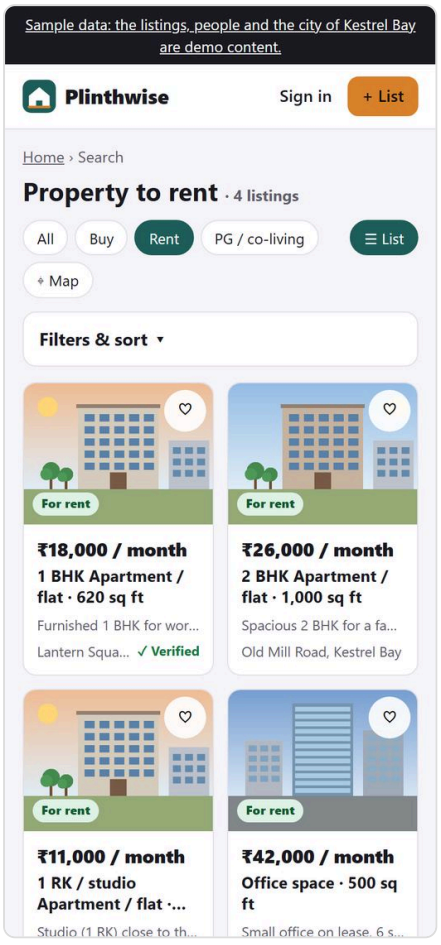
`/api/v1` uses bearer tokens (stored hashed, 90 days, revoked on sign-out). It covers search and listings, the day's visit slots, shortlist, enquiries, visits, chat and reports for buyers, and listings, photos, leads, visits and featured orders for listers. See `API.md`.

```
curl "http://localhost:3560/api/v1/listings?purpose=rent&bhk=2&furnishing=furnished"
```

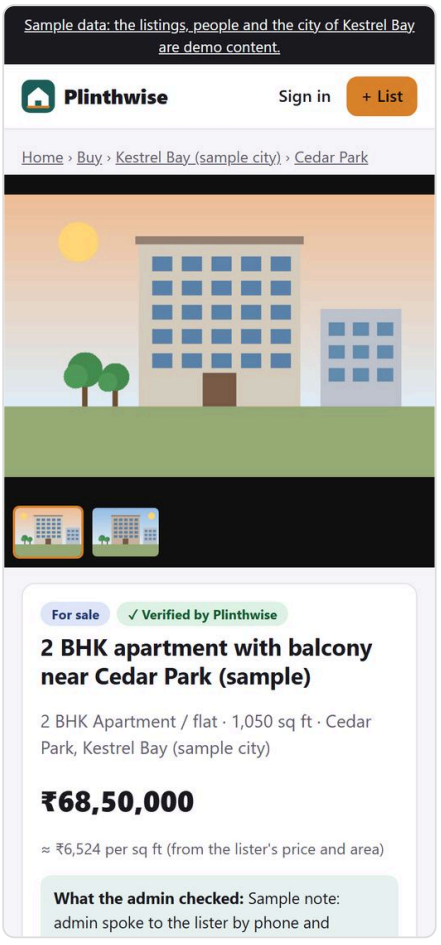
Run the mobile app

```
cd mobile
npm install
cp .env.example .env        # set SERVER_URL
npx expo start              # w = web, a = Android, i = iOS
```

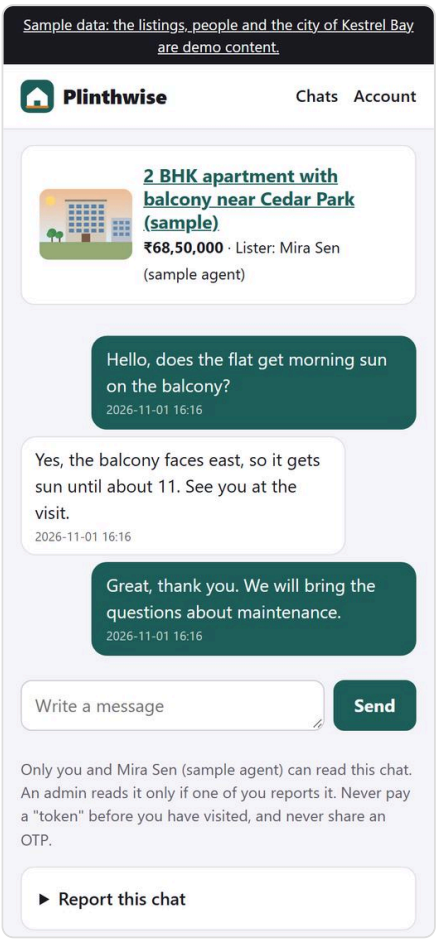
Buyer mode: search with filters, listing pages with a photo carousel, shortlist, enquiry, site visits with free slots, and chat. Lister mode for agents and owners: my listings with renew and mark sold or rented, leads, the visits calendar and chats. Posting and editing listings with photos and the map pin is done in the web panel; the map opens in the browser. Admins use the web admin.



Search at phone width.



A listing.



Chat.

8. Honest limits and troubleshooting

- No title, ownership, approval or RERA checks; no home loans, valuations, price estimates or legal advice.
- No wallet, escrow or rent collection; no payment gateway; UPI is confirmed by hand.
- No push notifications and no map inside the mobile app. Native Android and iOS builds have not been run by us: the app type-checks and its web export was tried in a browser. Test your EAS builds on real phones before publishing.
- Tests ran on SQLite only. OpenStreetMap tiles are for testing only.

Problem	Fix
A new or edited listing does not show	It is waiting in Admin → Approval queue, or it has expired.
A lister cannot request a featured listing	Set your UPI ID in Admin → Payments (UPI).
A visit slot is refused	The lister already has a visit at that time, or it is outside visiting hours or the booking window (Admin → Listing rules).
Posting is switched off on the live site	Activate the licence in Admin → Licence.
Logged out right away behind https	Set `TRUST_PROXY=true`.

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