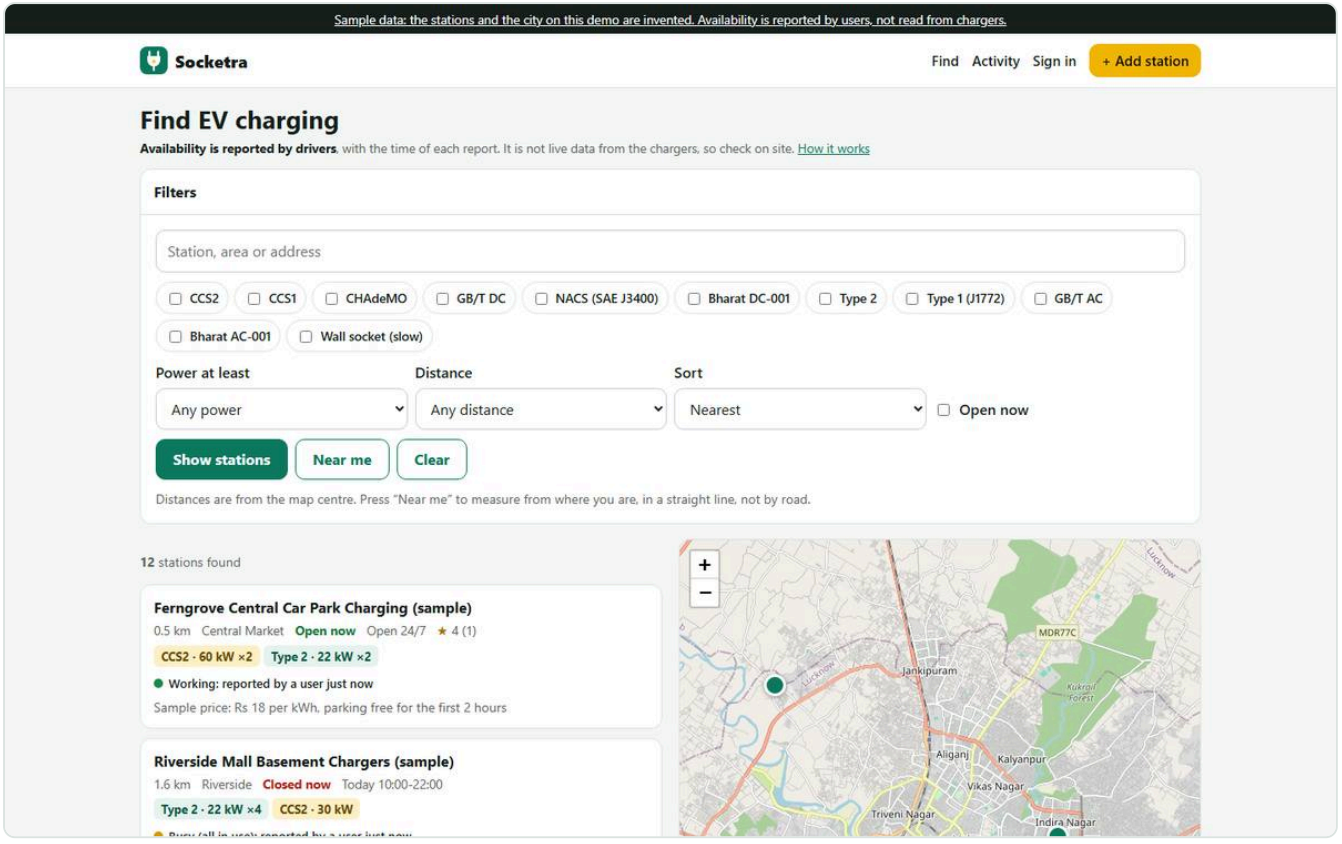
CODE CRAFT · INSTALLATION GUIDE

Socketra EV Charger Finder

Socketra 1.0: an EV charging station finder with a driver web app, an owner panel, an admin panel, a REST API and an Expo mobile app. Node.js, Express, EJS and Prisma. This guide takes you from the download to a live app on Hostinger, cPanel or a VPS.



The finder: connector, power, distance and open-now filters, the station list with each station's latest driver report, and the map.

> **Availability is reported by drivers.** It is not live data from chargers. The app is not connected to any charger or charging network, and it says so next to every status.

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/lib/`	The Express app and its logic: stations (validation, search, availability, approval), opening hours, reports, check-ins, ratings, claims, CSV import, settings, email, licence, add-ons.
`src/routes/`	`web.js` (driver web app), `partner.js` (owner panel at /owner), `admin.js` (admin), `api.js` (REST API /api/v1).
`views/`, `public/`	EJS templates, CSS, small browser scripts and icons.
`prisma/`	Database schema and the sample-data seed.
`mobile/`	Expo (React Native) app: finder, owner mode and approval-queue mode.
`addons/`	Drop add-on folders here (none are included).
`test/`	22 unit and flow tests and a curl smoke test.
`README.md`, `API.md`	Reference documentation.

Requirements

- Node.js 18.18 or newer (20 or 22 LTS recommended).
- Database: nothing extra for SQLite (default). For MySQL 8+ or PostgreSQL 13+, an empty database and user.
- A domain with https for the live app, and your own map tile provider (see section 8).
- For the mobile app: Node.js, the Expo tools and an Expo account for EAS builds.

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:3510
```

Address	What opens
`http://localhost:3510`	Driver web app (finder)
`http://localhost:3510/owner`	Owner panel
`http://localhost:3510/admin`	Admin
`http://localhost:3510/api/v1`	REST API

`npm run setup -- --no-seed` creates an empty app: the first visit to `/admin` runs a setup wizard for your admin account. `npm run setup -- --reseed` wipes everything and loads the sample data again.

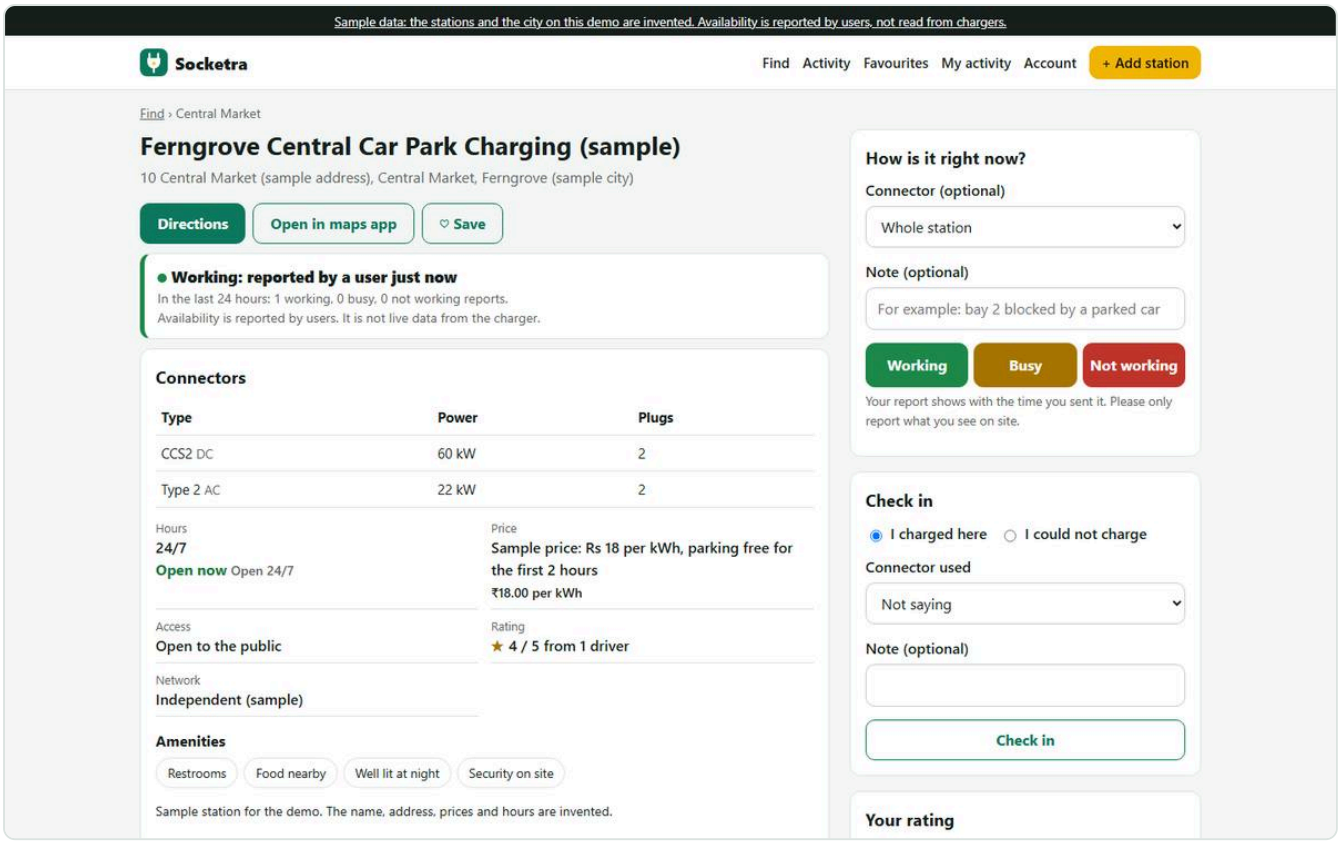
Demo logins (sample data only)

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

Role	Email	Where
Admin	`admin@socketra.test`	/admin, or the approval-queue mode in the mobile app
Station owner (2 sample stations)	`owner@socketra.test`	/owner, or owner mode in the mobile app
Driver (no station yet: try claiming one)	`owner2@socketra.test`	/ or the mobile app
Driver (added 1 station waiting for approval)	`driver@socketra.test`	/ or the mobile app
Driver (added 1 station waiting for approval)	`driver2@socketra.test`	/ or the mobile app

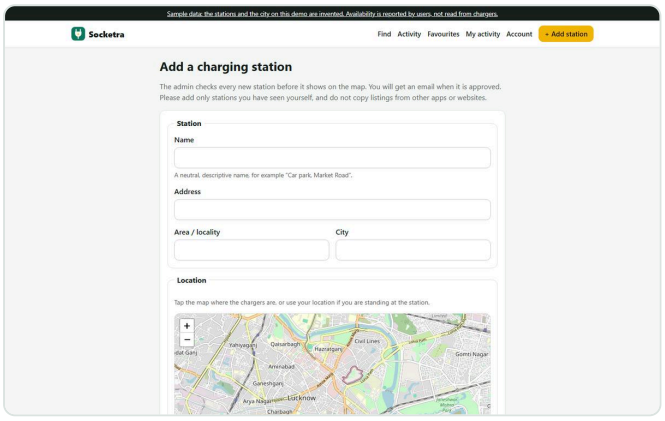
The sample data is an invented city, "Ferngrove (sample city)", with 14 sample stations (12 live, 2 waiting for approval). No reports, check-ins, ratings or favourites are seeded.

3. How drivers use it

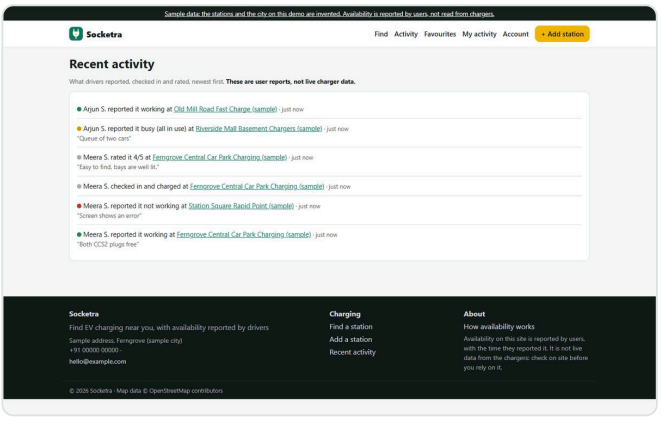


A station page: the latest driver report with its age, connectors with power and plugs, hours, price note, access and rating.

- **Find:** search, connector types, minimum power, distance from the map centre or "Near me", amenities, open now; sort by distance, power or rating. Distances are straight-line, not by road.
- **Report:** Working, Busy or Not working, for the station or one connector, with a note. One report per driver per station every 10 minutes.
- **Check in and rate:** "I charged here" or "I could not charge"; then rate 1 to 5 stars once (changeable later).
- **Add a station:** name, map pin, connectors with power, hours, price note, access and amenities. It waits in the approval queue.
- **Claim:** a station's operator presses "Claim this station"; the admin verifies it.

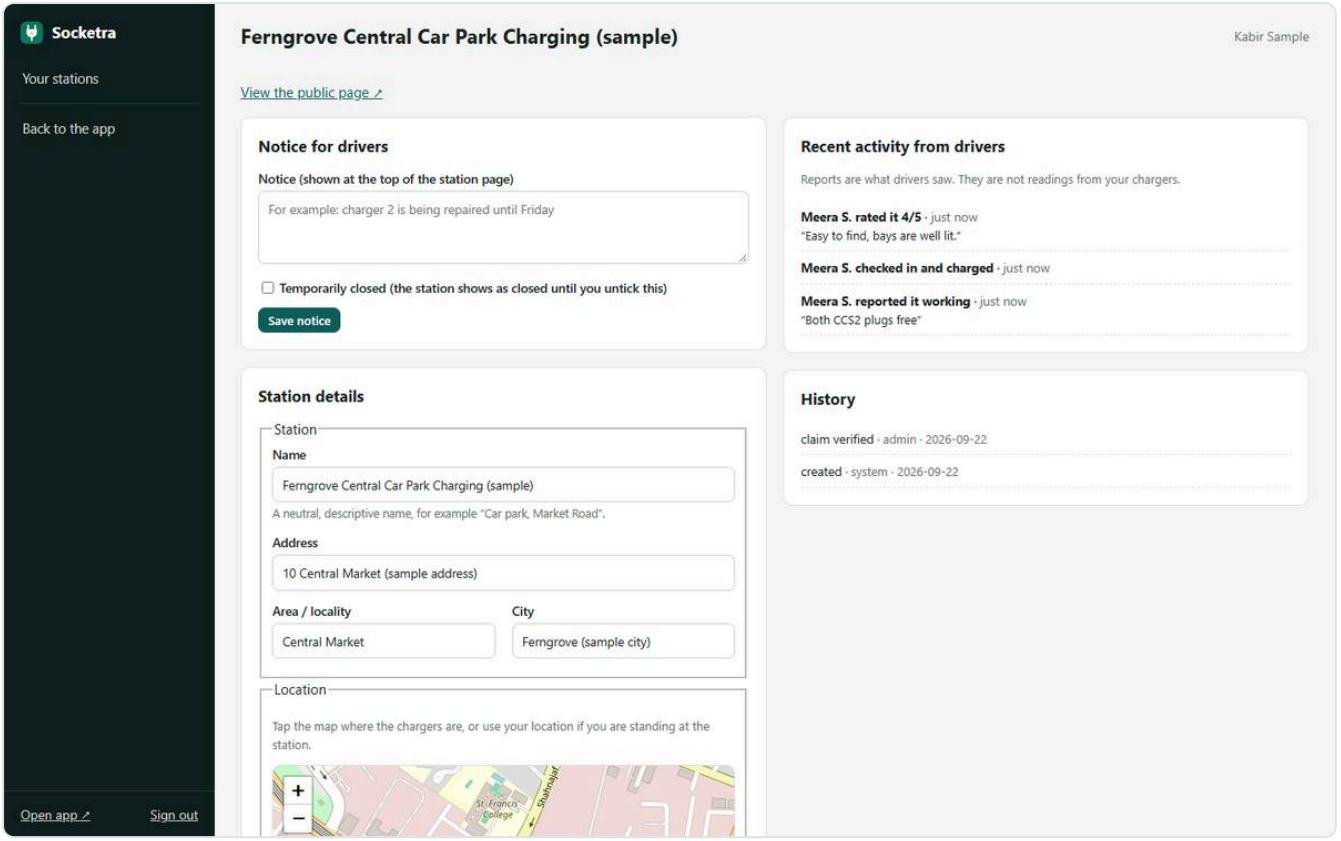


Adding a missing station.



Recent activity from drivers.

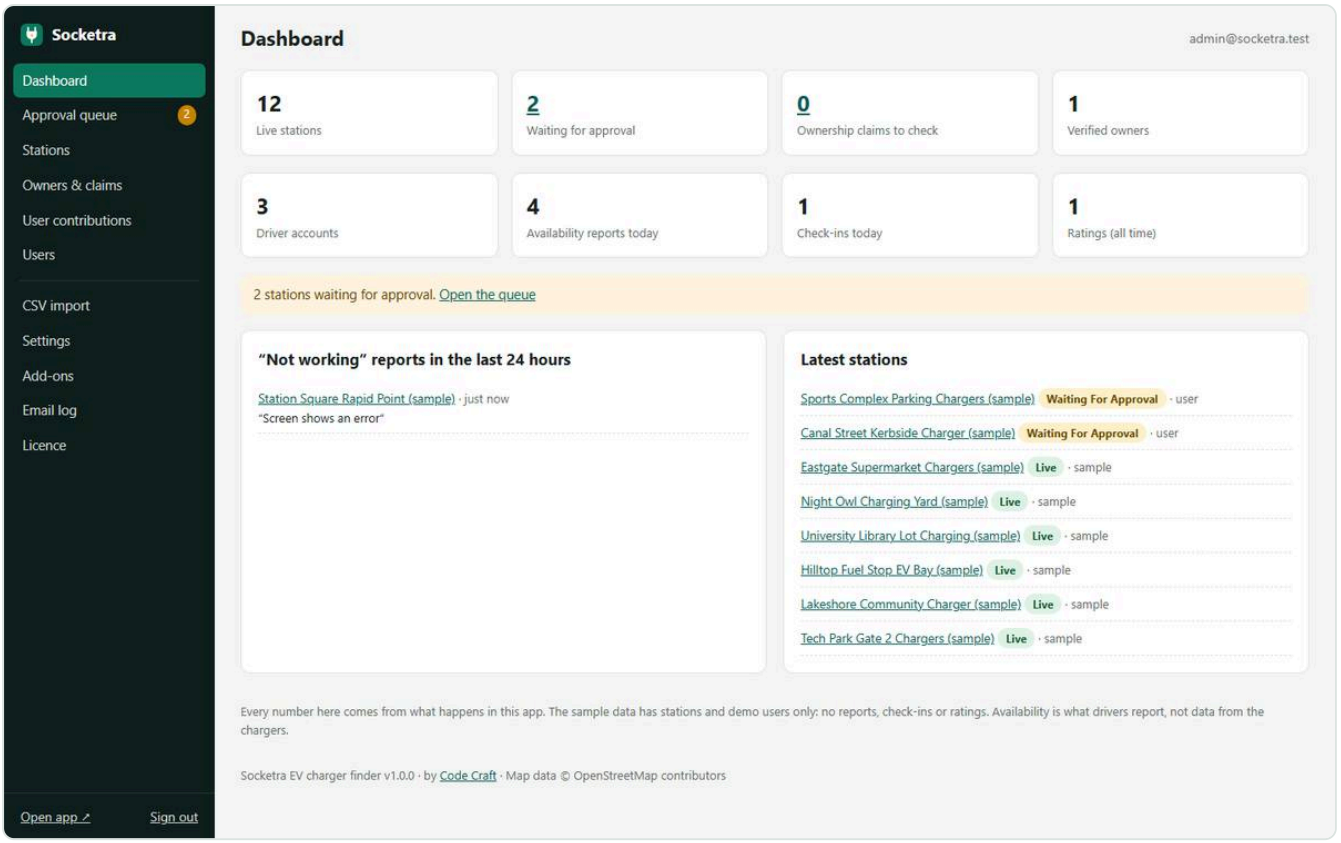
4. The owner panel



Owner station screen: a notice for drivers, the temporarily-closed switch, the station details and recent driver activity.

A driver becomes an owner only when the admin verifies their claim. The owner panel lists their stations with reports, check-ins, saves and rating. For each station the owner edits the details, connectors, hours and price note, posts a notice, and marks it temporarily closed. Owners cannot report on, check in at or rate their own station.

5. The admin panel



Admin dashboard: live stations, the approval queue, claims, reports today and recent not-working reports.

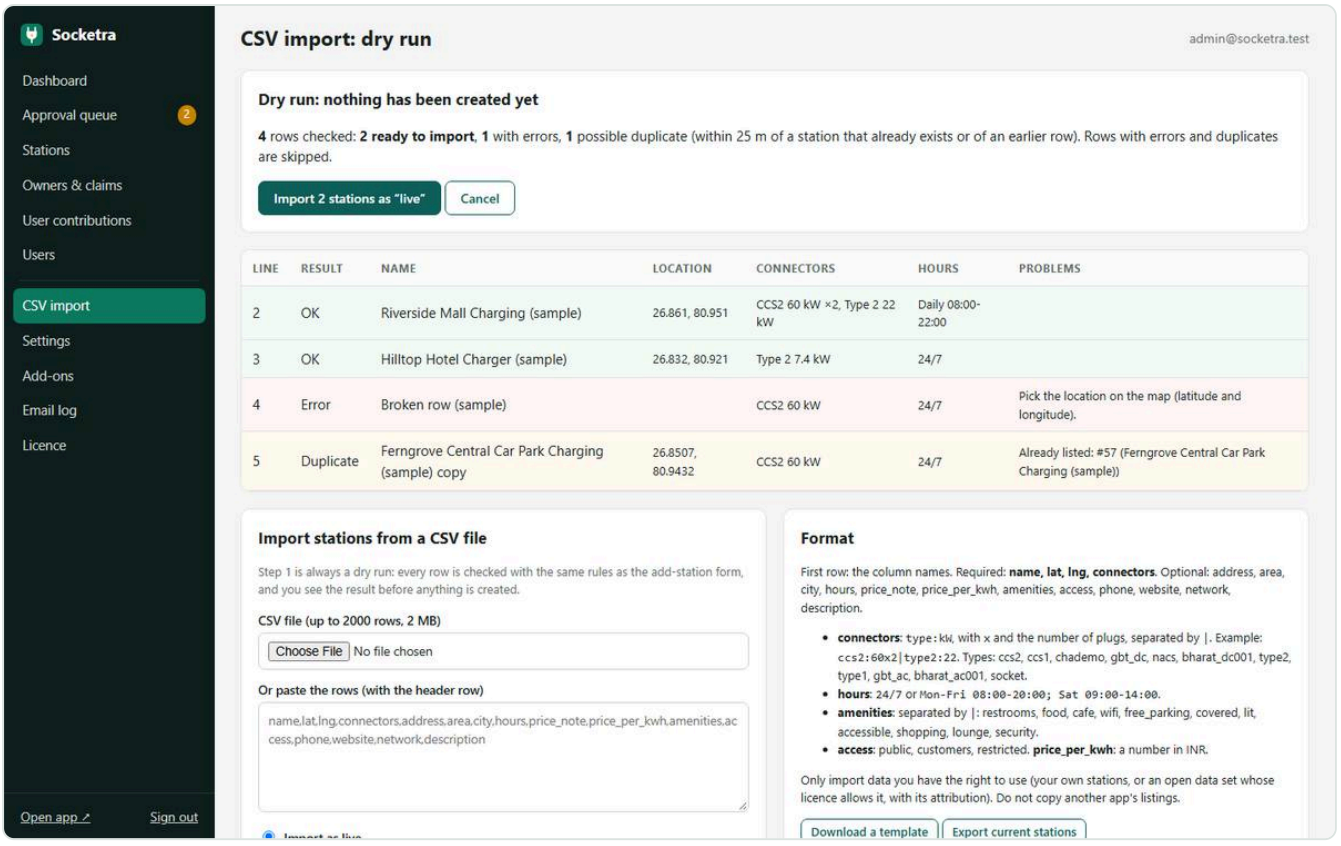
Screen	What you do there
Approval queue	Approve, edit first, or reject with a reason (emailed to the driver). Warns about stations within 150 m.
Stations	Add, edit, hide, delete, export CSV; post a notice; remove an owner.
Owners & claims	Verify or reject claims. Check each claim yourself, for example by calling the station's own number.
User contributions	Hide abusive or fake reports, check-ins and ratings. You cannot write or edit what users wrote.
Users	Block or unblock accounts (blocking revokes their app tokens).
CSV import	Dry run first, then import once (see below).
Settings	Business and brand, map centre, time zone, station and report rules, chat, WhatsApp, analytics, banners, SEO, maintenance, SMTP.
Email log, Add-ons, Licence	Every email the app sent (or would have sent), add-on settings, licence key.

First things to set

1. Settings → Business & brand: name, logo, colours, contact details, **map centre** and **time zone** (used for "open now").

2. Settings → Stations & reports: approval of new stations, how long a report counts (72 h), when it is labelled out of date (24 h), cooldowns, whether ratings need a check-in.
3. Stations: delete the sample stations, then add yours or use CSV import.
4. Settings → Email (SMTP): until it is set, emails are only written to the email log.

6. CSV import



The dry run: 4 rows checked, 2 ready, 1 error (no location) and 1 possible duplicate. Nothing is created until you press Import.

Header row required, any column order. Required: `name`, `lat`, `lng`, `connectors`. Optional: `address`, `area`, `city`, `hours`, `price_note`, `price_per_kwh`, `amenities`, `access`, `phone`, `website`, `network`, `description`.

- `connectors`: `type:kW` with optional `x` and the number of plugs, separated by `|`, e.g. `ccs2:60x2|type2:22`. Types: ccs2, ccs1, chademo, gbt_dc, nacs, bharat_dc001, type2, type1, gbt_ac, bharat_ac001, socket.
- `hours`: `24/7`, `Daily 06:00-23:00` or `Mon-Fri 08:00-20:00; Sat 09:00-14:00`.
- `amenities`: restrooms, food, cafe, wifi, free_parking, covered, lit, accessible, shopping, lounge, security, separated by `|`.
- Up to 2,000 rows and 2 MB per file. Rows within 25 m of an existing station or an earlier row are flagged as duplicates.

```
name,lat,lng,connectors,hours,price_note
Riverside Mall Charging,26.861,80.951,ccs2:60x2|type2:22,Daily 08:00-22:00,Rs 18 per kWh
```

> Only import stations you have the right to publish: your own, ones your users add, or an open data set whose licence allows it, with its attribution.

7. 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.
4. Run `npm install` and `npm run setup` from the panel's terminal, then start the app.

Key	Value
`DATABASE_URL`	`file:./app.db` (SQLite), `mysql://…` or `postgresql://…`
`PORT`	Default 3510 (Hostinger and cPanel set it for you)
`APP_SECRET`	A long random string (npm run setup makes one)
`APP_URL`	Your public https address
`TRUST_PROXY`	`true` behind https on Hostinger, cPanel or Nginx
`LICENSE_KEY`	Your licence key (or paste it in Admin → Licence)
`TILE_URL`	Optional: the tile URL the API tells apps to use

VPS with PM2 and Nginx

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

Point an Nginx server block at `http://127.0.0.1:3510` with `proxy_set_header X-Forwarded-Proto $scheme;`, set `TRUST_PROXY=true` and add https (for example with Certbot).

MySQL or PostgreSQL

Create an empty database, put its URL in `DATABASE_URL` and run `npm run setup`. The script switches the Prisma provider to match. Our tests ran on SQLite.

8. Maps, licence and backups

Map tiles

The web app uses Leaflet with OpenStreetMap tiles. OpenStreetMap's public tile servers are fine for testing and light use but are **not for production traffic**. Before launch, switch to your own tile or map provider by changing `TILE` in `public/js/app.js` and `TILE_URL` in `mobile/.env`, and keep the attribution your provider requires.

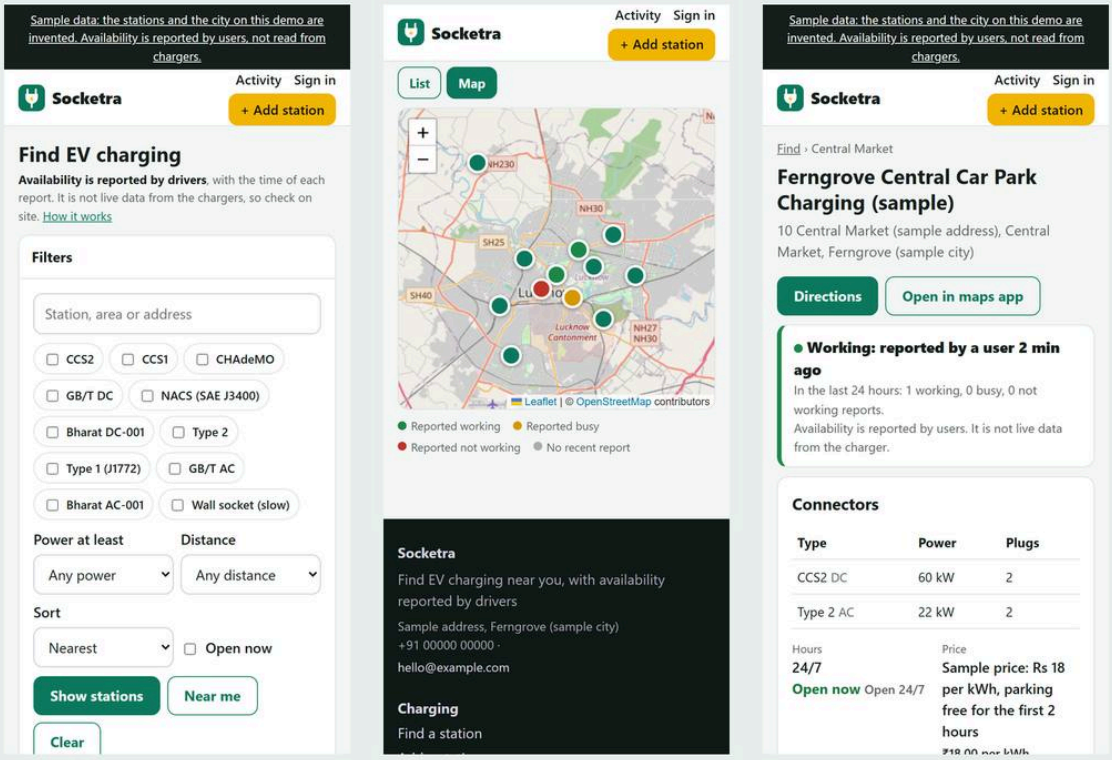
Licence

On a live domain, add your licence key in Admin → Licence. Until it is active, adding new stations is switched off and a banner shows. 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.

Backups

Back up the database regularly (`prisma/app.db` for SQLite, or your host's MySQL/PostgreSQL backup) and `public/uploads/`, which holds your logo.

9. REST API and mobile app



The web app at phone width: filters, the map with pins coloured by the latest report, and a station page.

The REST API at `/api/v1` uses bearer tokens (stored hashed, 90 days, revoked on sign-out, password change or blocking). It covers app info, stations and search, station details and ratings, the activity feed, adding stations, reports, check-ins, ratings, favourites, claims, the owner's stations and notice, and the admin queue and claims. `API.md` documents every endpoint.

Mobile app (mobile/)

```
cd mobile
npm install
cp .env.example .env        # set SERVER_URL to your server (https for published apps)
npx expo start              # w = web, a = Android, i = iOS, or Expo Go
```

- The account's role picks the mode: drivers get the finder, owners also manage their stations, admins get the approval queue.
- From the Android emulator use `http://10.0.2.2:3510`; from a phone on the same Wi-Fi use your computer's LAN address.
- The map is a simple tile map with pins and zoom and move buttons. There is no pinch or drag.
- Set the app name, bundle id and colours in `mobile/.env`, then build with EAS. **Native Android and iOS builds have not been run by us**: test on real phones before you publish.

10. Going-live checklist and limits

- Change or delete the demo accounts and the sample stations.
- Set the business details, logo, colours, map centre and time zone.
- Set `APP_URL`, `TRUST_PROXY=true` and a long `APP_SECRET`; serve over https.
- Switch the map tiles to your own provider (web and mobile).
- Enter your SMTP server and send yourself a test email.
- Activate the licence and set up regular backups.
- Decide your report and approval rules, and how you will check ownership claims.

What this app does not do

- Availability is reported by drivers. It is not live charger data, and the app is not connected to any charger or network.
- No payments: price notes are shown as written; nothing is charged or held. No payment add-on is included.
- Ownership claims are checked by a person, not automatically.
- The mobile map has buttons only (no pinch or drag), and native builds have not been run.
- Distances are straight-line; directions open on openstreetmap.org.

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