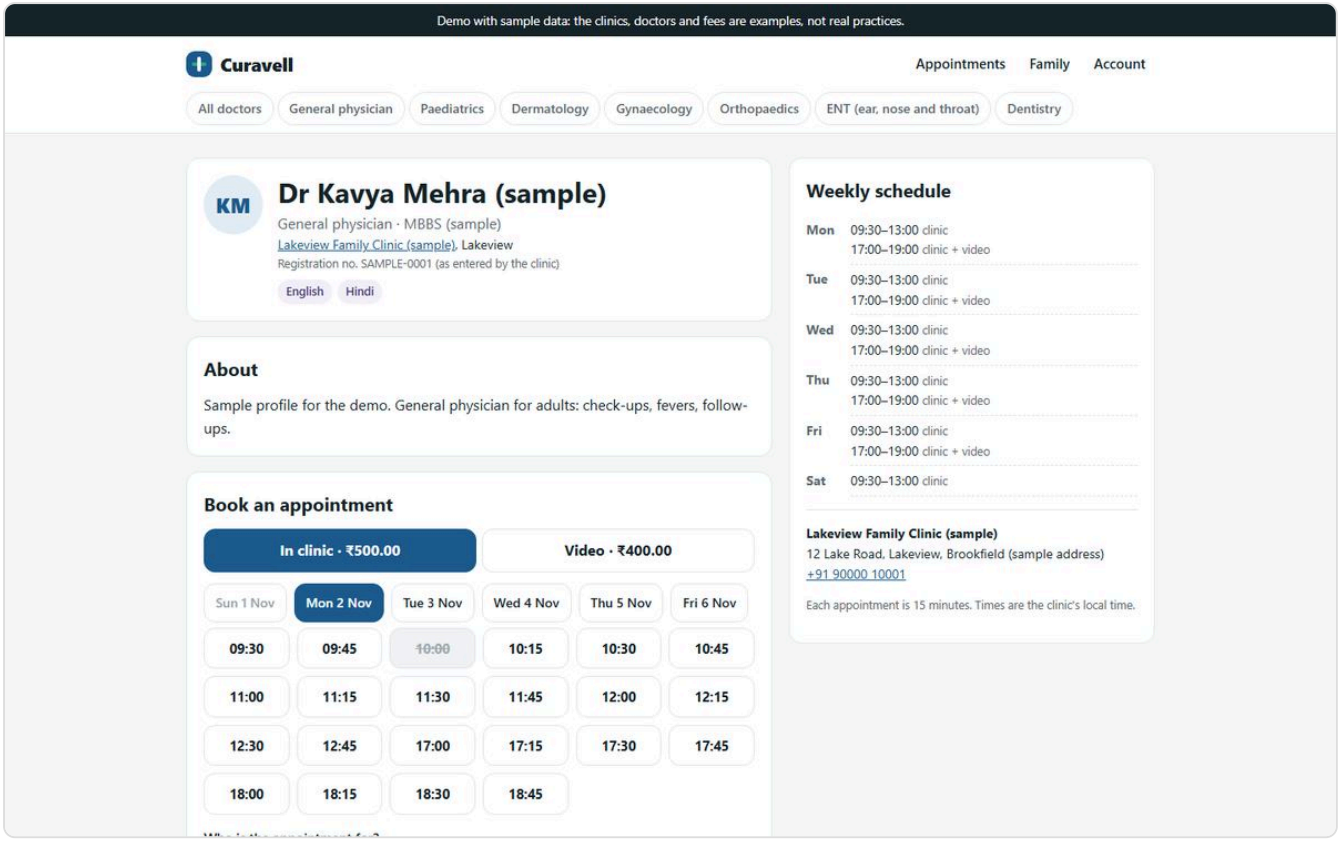
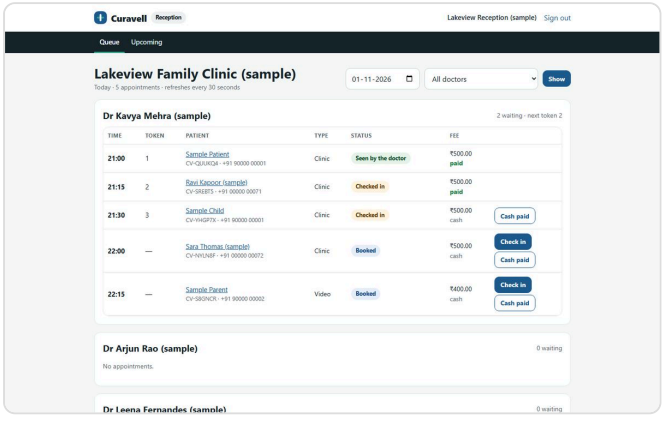
CODE CRAFT · INSTALLATION GUIDE

Doctor Appointment App

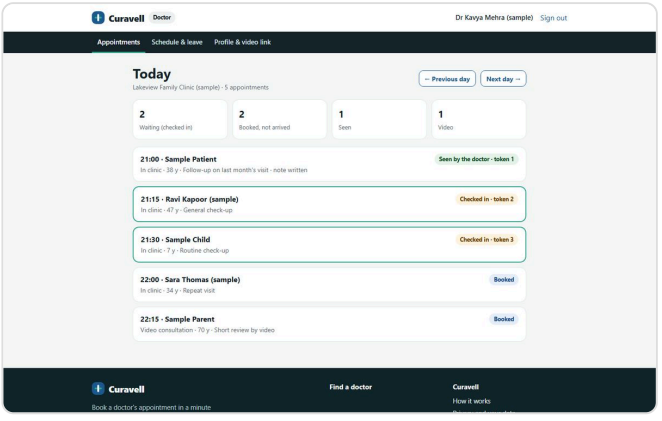
Curavell 1.0: a patient web app, a reception panel, a doctor panel, an admin panel, a REST API and an Expo mobile app, all in one Node.js project. This guide takes you from download to a live booking site on Hostinger, cPanel or a VPS.



A doctor's page: in-clinic or video, a date strip, the times that are really free, and the weekly schedule.



Reception: the day's queue with token numbers.



Doctor panel: the doctor's own day.

> **This app books appointments. It does not give medical advice**, does not diagnose or check what a doctor writes, and is **not a telemedicine platform**: a video appointment is a time slot plus a meeting link

> the clinic supplies. "Curavell", the sample city, clinics, doctors and patients are placeholders; change the name and colours in Admin → Settings.

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` (patient web app), `reception.js`, `doctor.js`, `admin.js`, `api.js` (REST API).
`src/lib/`	Booking, time locks, check-in tokens, notes, follow-ups and leave (`appointments.js`), schedule arithmetic (`slots.js`), family members, the prescription-note PDF, UPI links, settings, licence, mailer, add-on loader.
`views/`, `public/`	EJS templates, CSS, small browser scripts and icons.
`prisma/`	Database schema and the sample-data script.
`addons/`, `ADDONS.md`	The add-on system and payment contract. No gateway add-on is included.
`mobile/`	The Expo (React Native) app with a patient mode and a doctor mode.
`test/`	Unit, API flow and web-page tests (24 in all) and `api-smoke.sh`.
`README.md`, `API.md`	Reference documentation, including privacy and local-law notes.

Requirements

- Node.js 18.18 or newer (20 or 22 LTS recommended).
- Database: nothing extra for SQLite (default). MySQL 8+ or PostgreSQL 13+ use the same Prisma schema but were not part of our tests: only SQLite was tested.
- A domain with https. The clinic's own meeting service if you offer video appointments.
- 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:3530
```

Address	What opens
`http://localhost:3530`	Patient web app
`/reception`	Reception panel (staff sign in at `/login`)
`/doctor`	Doctor panel (doctors sign in at `/login`)
`/admin`	Admin panel
`/api/v1`	REST API

`npm run setup -- --no-seed` creates empty tables and the first visit to `/admin` runs the setup wizard. `npm run setup -- --reseed` loads the sample data again (admin users are kept). `npm run seed -- --no-demo-users` loads the sample clinics and specialities without demo logins.

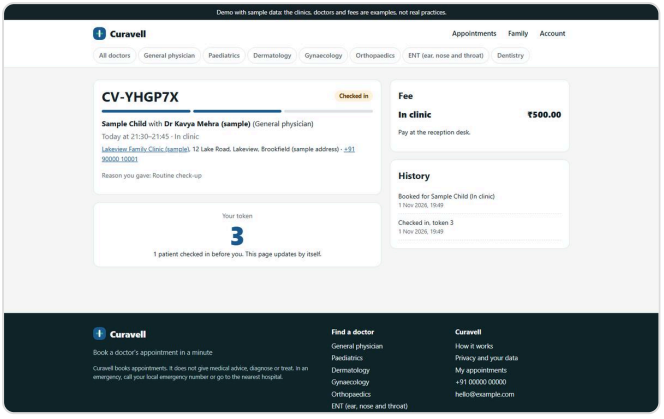
Demo logins (sample data, on your own copy only)

Role	Email	Password
Admin	`admin@example.com`	`Demo-Admin-2026!`
Patient (with a child and a parent as family members)	`patient@example.com`	`Demo-Patient-2026!`
Reception, Lakeview Family Clinic	`reception.lakeview@example.com`	`Demo-Reception-2026!`
Reception, Northgate Health Centre	`reception.northgate@example.com`	`Demo-Reception-2026!`
Doctor, general physician (clinic + video)	`dr.mehra@example.com`	`Demo-Doctor-2026!`

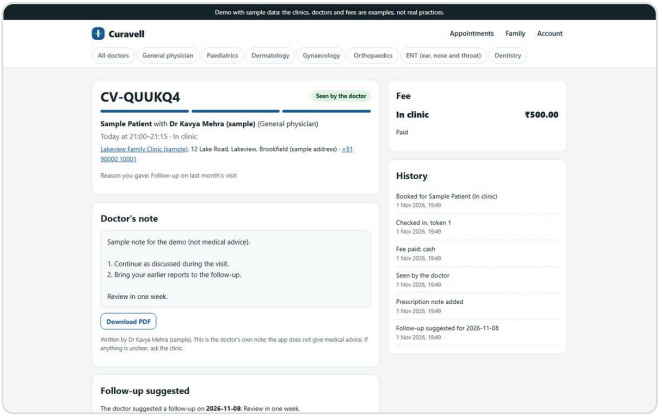
Four more sample doctors are listed in README.md. All sample names carry "(sample)", registration numbers are `SAMPLE-0001` …, and no appointments, notes, ratings or figures are created. The app shows no star ratings or reviews at all.

3. How an appointment works

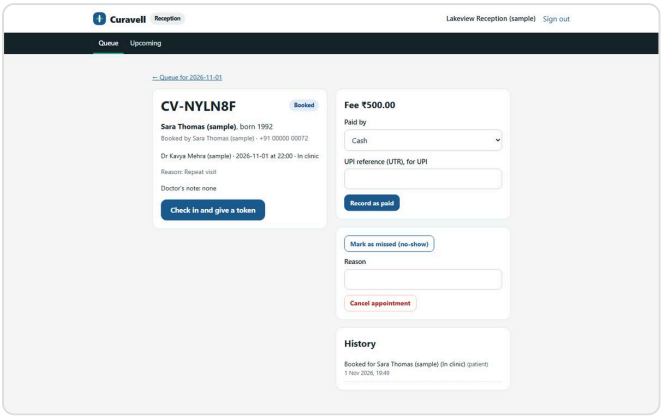
1. The patient picks a doctor, in clinic or video, a date and one of the free times, who it is for (themselves or a family member), an optional short reason and a payment method.
2. The server checks the time against the doctor's weekly sessions, leave days, lead time and booking window, sets the fee from the doctor's record and holds the time. Every 5-minute block of a doctor's day is a row with a unique index, so two appointments can never overlap.
3. **In clinic:** on the day, reception checks the patient in and the next token number for that doctor is given; the patient's page shows the token and how many are ahead. **Video:** the patient's page shows the clinic's meeting link.
4. The doctor marks the patient as seen, may write a plain-text note (the patient reads it and downloads the PDF) and may suggest a follow-up date, which the patient books in one tap.
5. Reception records the fee (cash, or UPI with the UTR), or marks a no-show. Patients can cancel until the cut-off (default 2 hours); reception and the admin can cancel later.



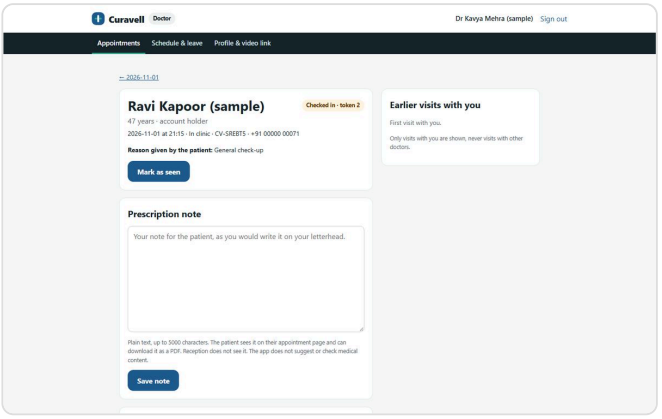
Patient: token number after check-in.



Patient: the doctor's note, its PDF and the follow-up.



Reception: check-in, fee and no-show. Reception never sees the doctor's note.

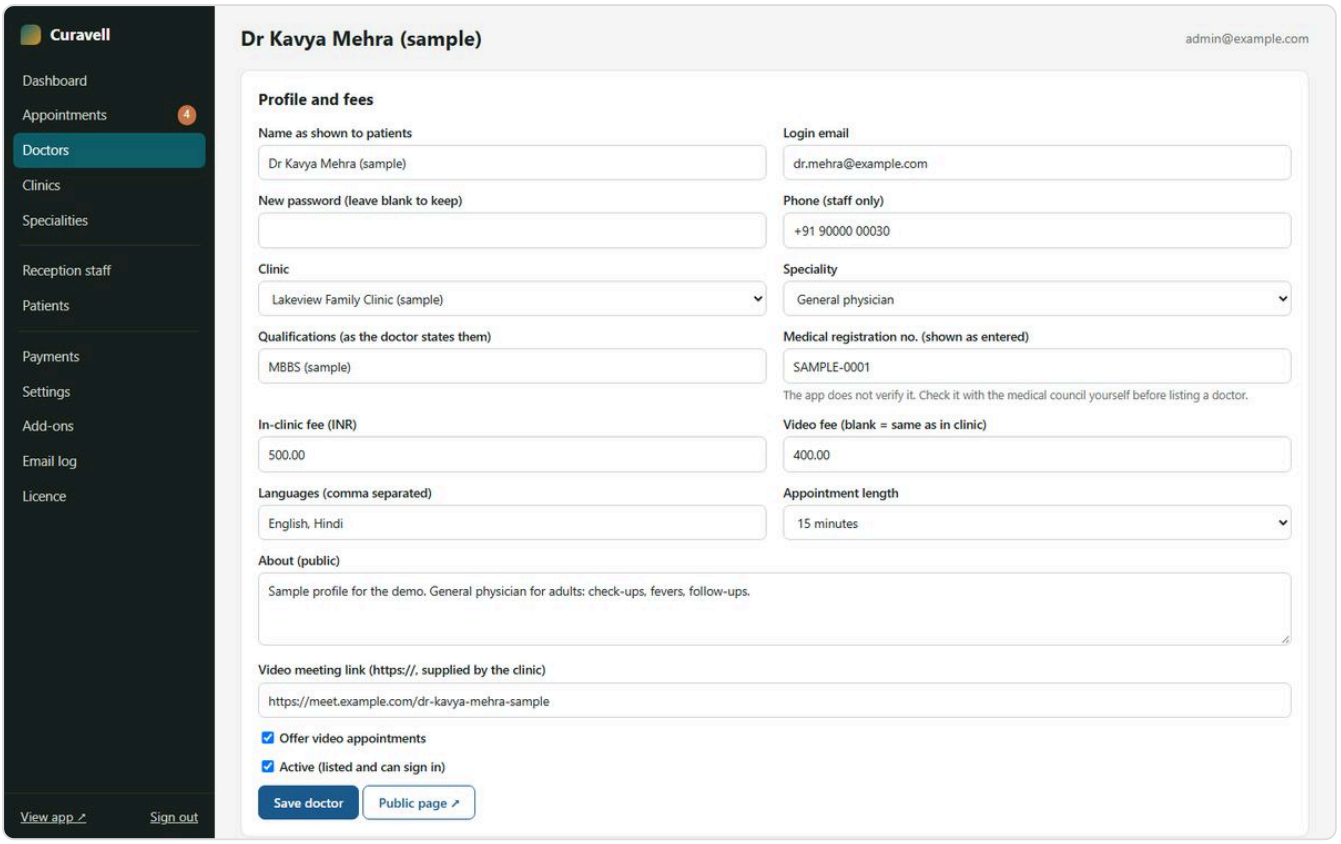


Doctor: a visit with the plain-text note box.

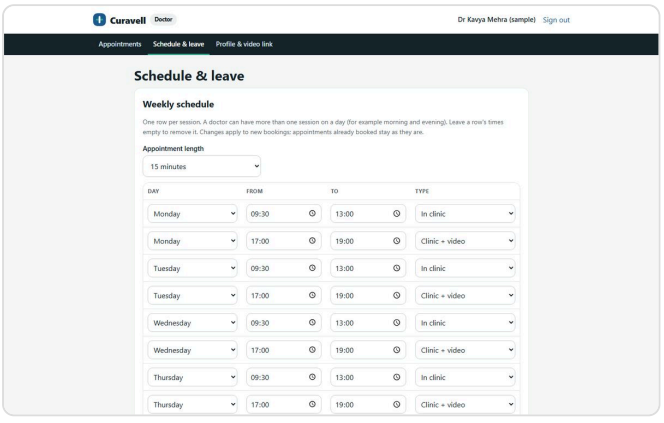
Reception cannot book on behalf of walk-in patients in this version: walk-ins book themselves on the website or the app.

4. First things to set in the admin

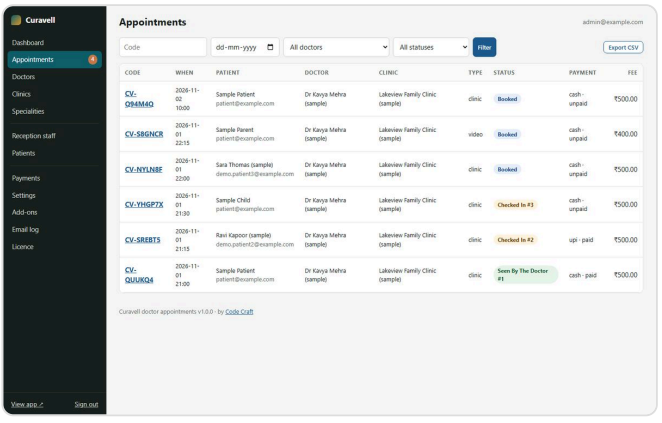
1. **Settings:** business name, logo, brand colours, contact details, SEO, booking rules (lead time, days ahead, cancellation cut-off), privacy contact and map centre.
2. **Clinics and Specialities:** your clinics (with a map pin) and the specialities you offer.
3. **Doctors:** login, clinic, speciality, qualifications and registration number (shown as entered; the app does not verify it), in-clinic and video fees, appointment length, languages, the meeting link, weekly sessions and leave days.
4. **Reception staff:** one login per clinic desk.
5. **Payments:** your UPI ID and which methods are on (only "pay at the clinic" is on in the sample).
6. **Settings → Email (SMTP):** until it is set, emails are only written to the email log.



Admin → Doctors: profile, fees, appointment length, video link.



Doctor → Schedule & leave: several sessions per weekday, each in clinic, video or both.



Admin → Appointments with filters and CSV export (no reasons or notes in it).

5. Put it online

1. Upload the files without `node_modules`, `.env` and `*.db`.
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`, `TZ` and your `MAP_TILE_URL`.
4. Run `npm install` and `npm run setup -- --no-seed` (or with the sample data, then delete the demo logins).
5. Open `https://your-domain/admin` and add your licence key in Admin → Licence. On a real domain new appointments stay switched off until the licence 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 curavell && pm2 save
```

Variable	What it is
`DATABASE_URL`	`file:./app.db` (SQLite), `mysql://…` or `postgresql://…`
`PORT`	Default 3530 (hosts usually set it).
`APP_SECRET`	A long random string that signs session cookies.
`APP_URL`	Your public https address (emails, the mobile app).
`TRUST_PROXY`	`true` behind Hostinger, cPanel or Nginx with https.
`TZ`	Time zone for schedules and slots, for example `Asia/Kolkata`.
`MAP_TILE_URL`, `MAP_ATTRIBUTION`	Your own map tile provider (see below).

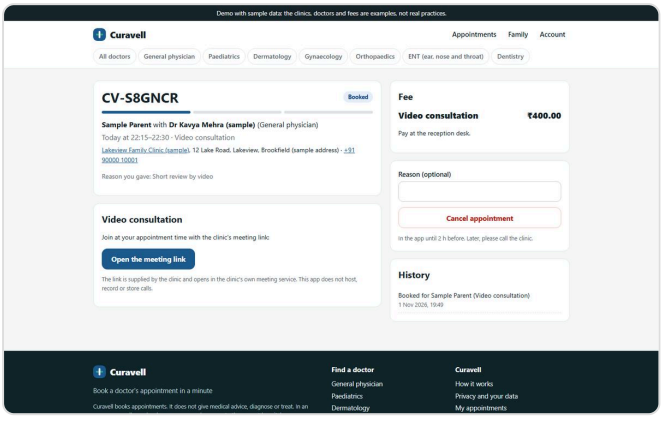
6. Privacy, payments and maps

The app stores personal and health-related data: names, years of birth, phone numbers, appointment history, the reasons patients type and the notes doctors write. You are responsible for the law where you and your patients are (for example India's DPDP Act, GDPR for health data, HIPAA in the US). README.md has a starting checklist; it is not legal advice.

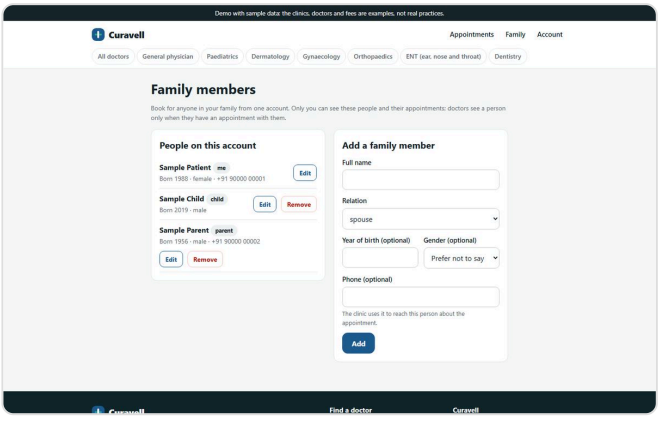
- Role and ownership checks on every route: a patient sees their own family only, a doctor their own appointments, reception their own clinic and never the note. Admin screens and the CSV show that a note exists, not its text.
- The privacy page is a template: replace it with a policy reviewed for your country. Analytics tags are not loaded on patient account pages or the panels.
- Not included: encryption at rest, audit logs of who viewed a record, two-factor sign-in and backups. Use your host for those.

Payments: patients pay per visit, at the desk (cash), by UPI to your UPI ID (a link and QR for the exact fee, confirmed by reception with the UTR) or through a gateway add-on. No gateway is bundled, and there is no wallet or prepaid credit.

Maps: clinic pages use Leaflet with OpenStreetMap's public tiles, which are for testing only. Set `MAP_TILE_URL` and `MAP_ATTRIBUTION` to your own provider before you launch.



Video appointment: the clinic's meeting link. The app does not host or record calls.



Family members on one patient account.

7. REST API and the mobile app

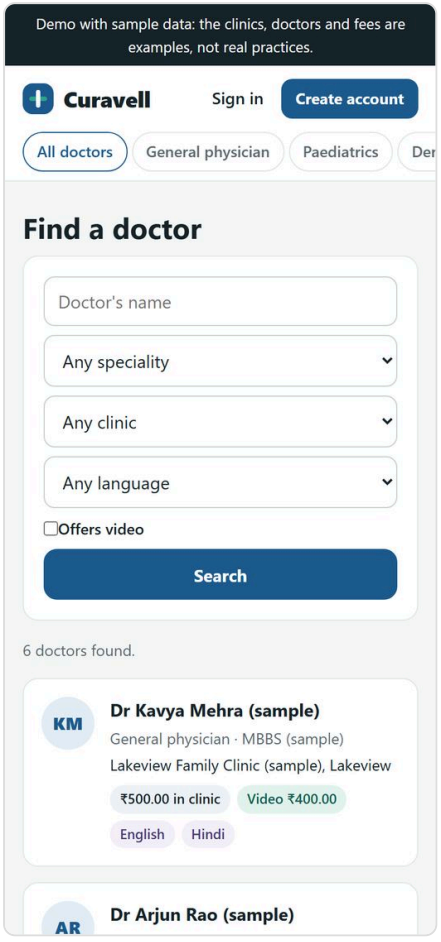
`/api/v1` uses bearer tokens from `/auth/login` (90 days, revoked on sign-out). It covers doctors, clinics, specialities and free times, family members, booking, cancelling and the note PDF for patients; the doctor's day, seen, note, follow-up, schedule and leave; and the reception queue, check-in, payment, no-show and cancel. See `API.md`.

```
curl http://localhost:3530/api/v1/doctors
curl "http://localhost:3530/api/v1/doctors/1/slots?date=2026-11-02&mode=clinic"
```

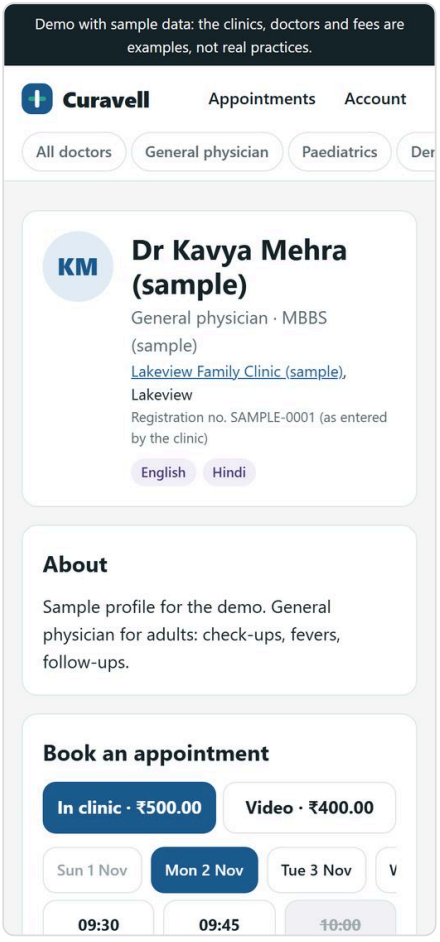
Run the mobile app

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

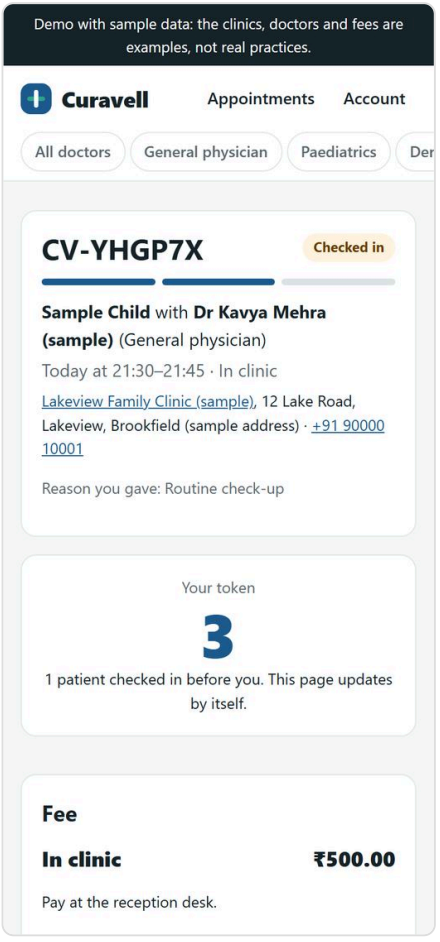
The app has a patient mode and a doctor mode, chosen by the signed-in role; reception and admin accounts use the website. Use `http://10.0.2.2:3530` for the Android emulator and your https domain for a published app. Build with EAS (`npx eas-cli@latest build -p android --profile preview`).



Find a doctor at phone width.



Booking.



Token after check-in.

8. Honest limits and troubleshooting

- Native Android and iOS builds have not been run by us: the app type-checks and its web export builds. Test your EAS builds on real phones before publishing.
- Not telemedicine: video is a meeting link only. No medical advice, no SMS or push reminders, no waiting-room display, no e-signatures on notes, no walk-in booking at reception.
- No payment gateway is bundled; UPI is confirmed by hand. Gateway add-ons in this product family are tested only against mocked responses.
- No wallet. Tests ran on SQLite only. Map tiles from OpenStreetMap are for testing only.

Problem	Fix
A doctor shows no free times	Check the weekly sessions, leave days, the lead time and days ahead in Settings, and the `TZ` variable.
"Check-in opens on the day"	Reception can only check in on the appointment date.
Booking is switched off on the live site	Activate the licence in Admin → Licence.
Logged out right away behind https	Set `TRUST_PROXY=true`.
No emails arrive	Check Admin → Email log; set SMTP in Settings.

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