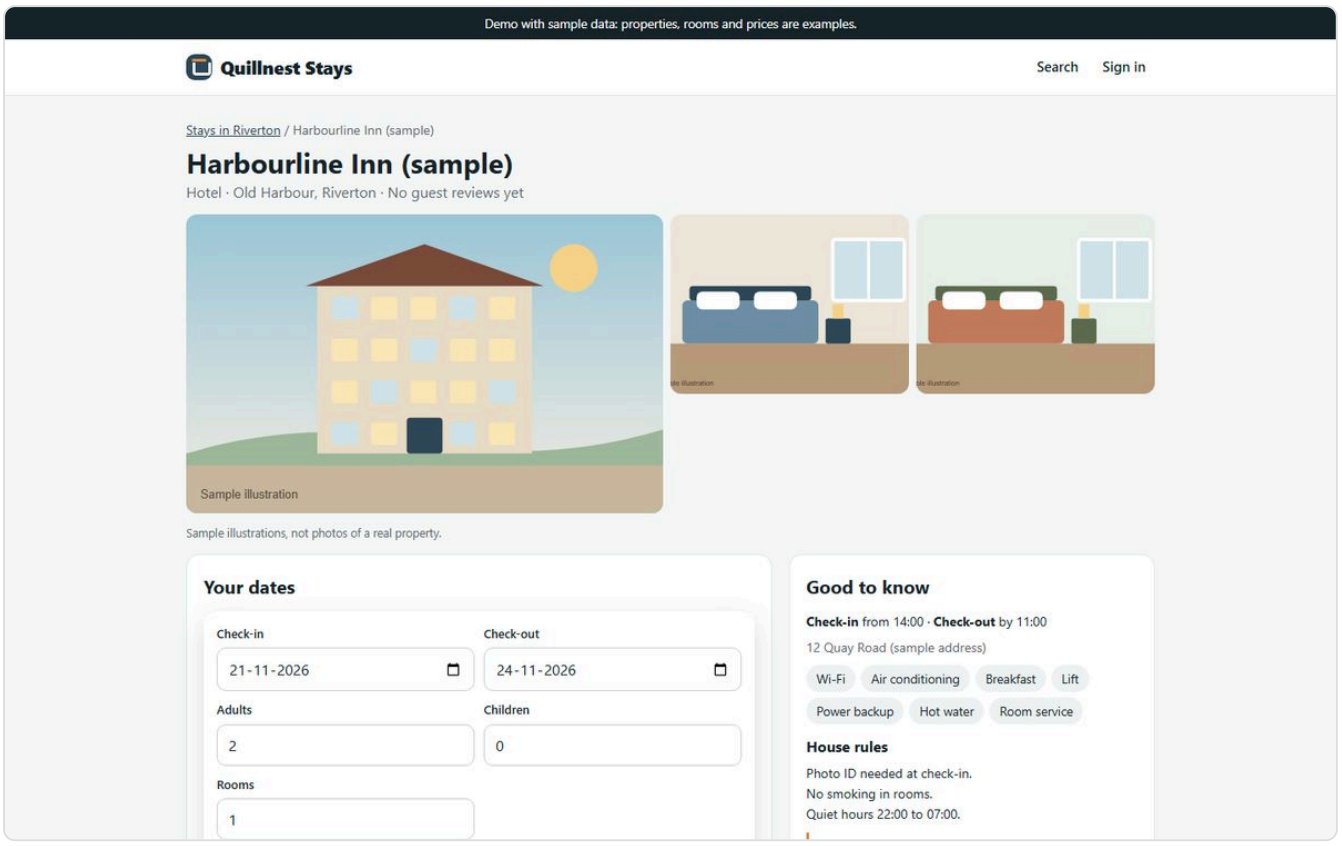
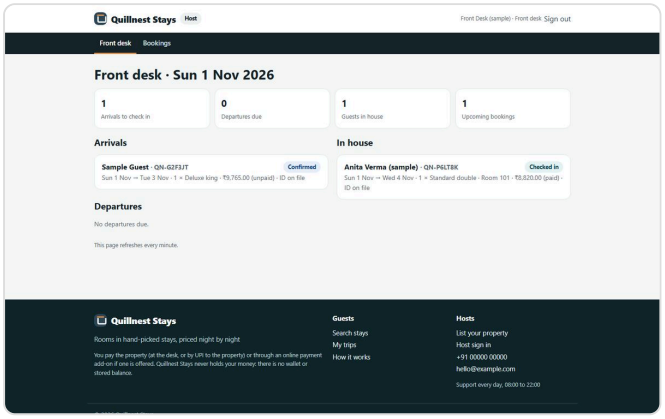
CODE CRAFT · INSTALLATION GUIDE

Hotel Booking App

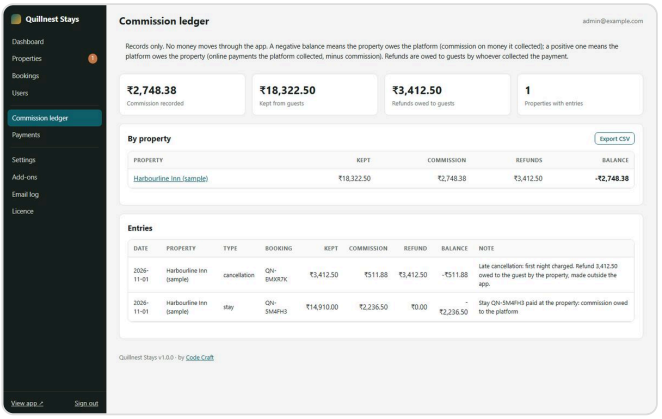
Quillnest Stays 1.0: a guest web app, a host panel for owners and front-desk staff, an admin panel with a commission ledger, a REST API and an Expo mobile app, all in one Node.js project. This guide takes you from download to a live booking site on Hostinger, cPanel or a VPS.



A property page: sample illustrations, your dates, room types with each rate's total, and what to know before you book.



Front desk: arrivals, departures and guests in house.



Admin: commission ledger.

No channel manager and no OTA sync. Rooms and prices are not synchronised with other booking sites or a property-management system. If a property also sells rooms elsewhere, the owner closes nights or lowers the allotment here by hand. "Quillnest Stays", the towns, properties and people are placeholders, and the pictures are labelled sample illustrations.

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` (guest web app), `partner.js` (host panel at `/host`), `admin.js`, `api.js` (REST API).
`src/lib/`	Availability, booking with inventory, cancellation, check-in/out, ID files and the ledger (`stays.js`); price calendar, seasons, rate plans, tax and policies (`rates.js`); the invoice PDF; UPI links; settings, licence, mailer, add-on loader.
`views/`, `public/`	EJS templates, CSS, scripts and the sample illustrations.
`storage/guest-ids/`	Private guest ID files, outside the public folder. Empty in the download.
`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 guest mode and a front-desk mode.
`test/`	22 unit, API and web-page tests.

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; only SQLite was tested.
- A domain with https, and writable `storage/guest-ids/` and `public/uploads/` folders.
- 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:3540
```

Address	What opens
`http://localhost:3540`	Guest web app
`/host`	Host panel for owners and front desk (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).

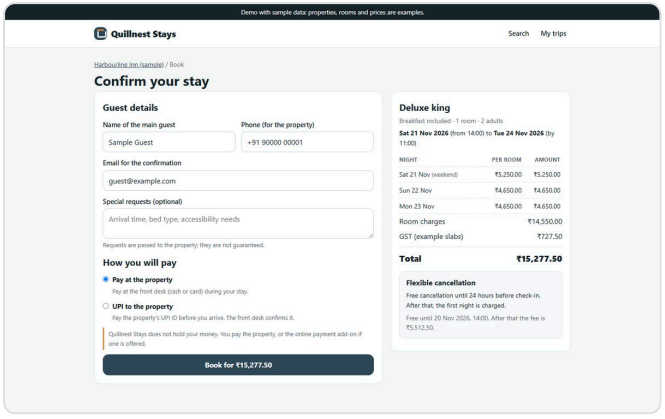
Demo logins (sample data, on your own copy only)

Role	Email	Password
Admin	`admin@example.com`	`Demo-Admin-2026!`
Guest	`guest@example.com`	`Demo-Guest-2026!`
Owner of Harbourline Inn and Old Mill Guest House	`owner.riverton@example.com`	`Demo-Owner-2026!`
Owner of Pinecrest Homestay and Lakeside Lodge (pending)	`owner.pinehollow@example.com`	`Demo-Owner-2026!`
Front desk at Harbourline Inn only	`desk.harbourline@example.com`	`Demo-Desk-2026!`

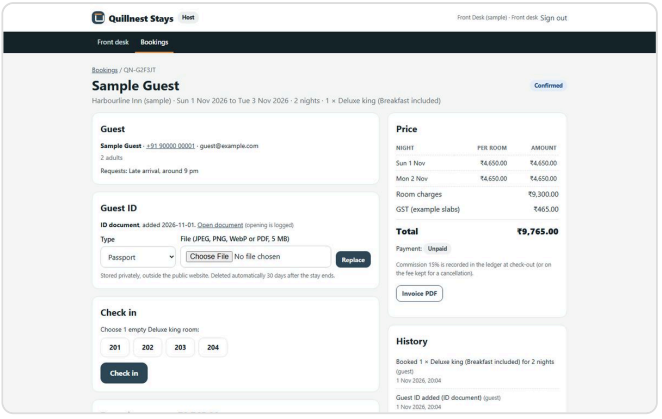
No bookings, reviews, ratings or sales figures are created. The tax slabs are an example ("GST (example slabs)"): check the current rates for your country. Old Mill's "Attic twin" has a single room, which makes the sold-out flow easy to try.

3. How a stay works

1. The guest searches by city, dates, adults, children and rooms. Results show only room types free on every night, with the total for the stay. The server prices every night: special price, room-type season, property season, weekend, base, then the rate plan, extra guests and the tax slab.
2. The guest books. Each room type has a row per night with a sold counter; every night is taken in one transaction, and if any night is full the whole booking rolls back, so the last room cannot be sold twice.
3. Before arrival the guest can upload their ID on the website. On the check-in date the front desk sees the arrival, checks the ID, picks empty rooms of the booked type and checks the guest in.
4. Payment: cash or card at the desk, UPI to the property's own UPI ID confirmed with the UTR, or a gateway add-on. Check-out needs the payment recorded first; leaving early gives the remaining nights back to sale. The ledger records the commission.
5. The guest can review the stay once, after check-out.



Booking page: night-by-night price, tax, and the fee if you cancel.



Front desk: guest ID, empty-room chips and check-in.

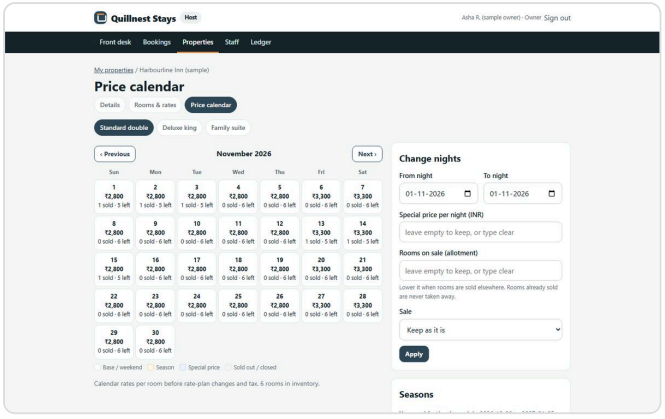
Cancellation policies

Policy	Rule
Flexible	Free until 24 hours before check-in, then the first night.
Moderate	Free until 5 days before, then 50%.
Non-refundable	No refund if the guest cancels, changes or does not arrive.

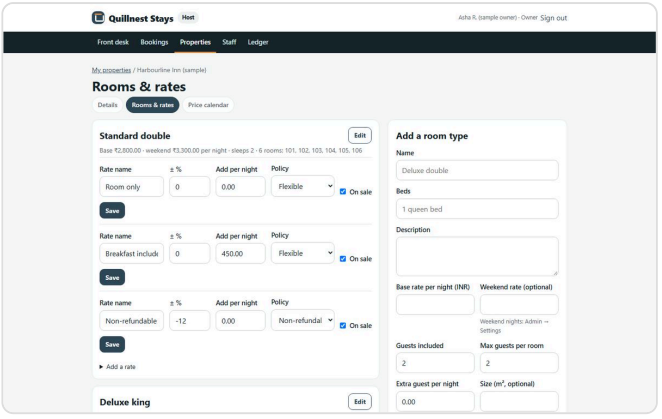
The fee and the refund (what was paid minus the fee) are calculated on the server, shown before cancelling and **recorded in the ledger only**. The refund itself is made outside the app and recorded with its reference. The owner can waive the fee.

4. Owners: rooms, rates and the calendar

1. **Properties:** details, amenities, map pin, photos, the property's UPI ID, check-in and check-out times. New properties wait for admin approval.
2. **Rooms & rates:** room types (base and weekend rate, guests included and maximum, extra-guest charge) and their room numbers; inventory follows the room list. Rate plans per room type: a percentage up or down, an add-on per night such as breakfast, and the cancellation policy.
3. **Price calendar:** seasons (a date range, a percentage or a fixed rate), special prices, rooms on sale (allotment) and stop sale for any night or date range.
4. **Staff:** front-desk logins that only see this owner's chosen property.
5. **Ledger:** what was kept from guests, the commission and refunds owed.



Owner → Price calendar: nights, sold counts and changes for a date range.



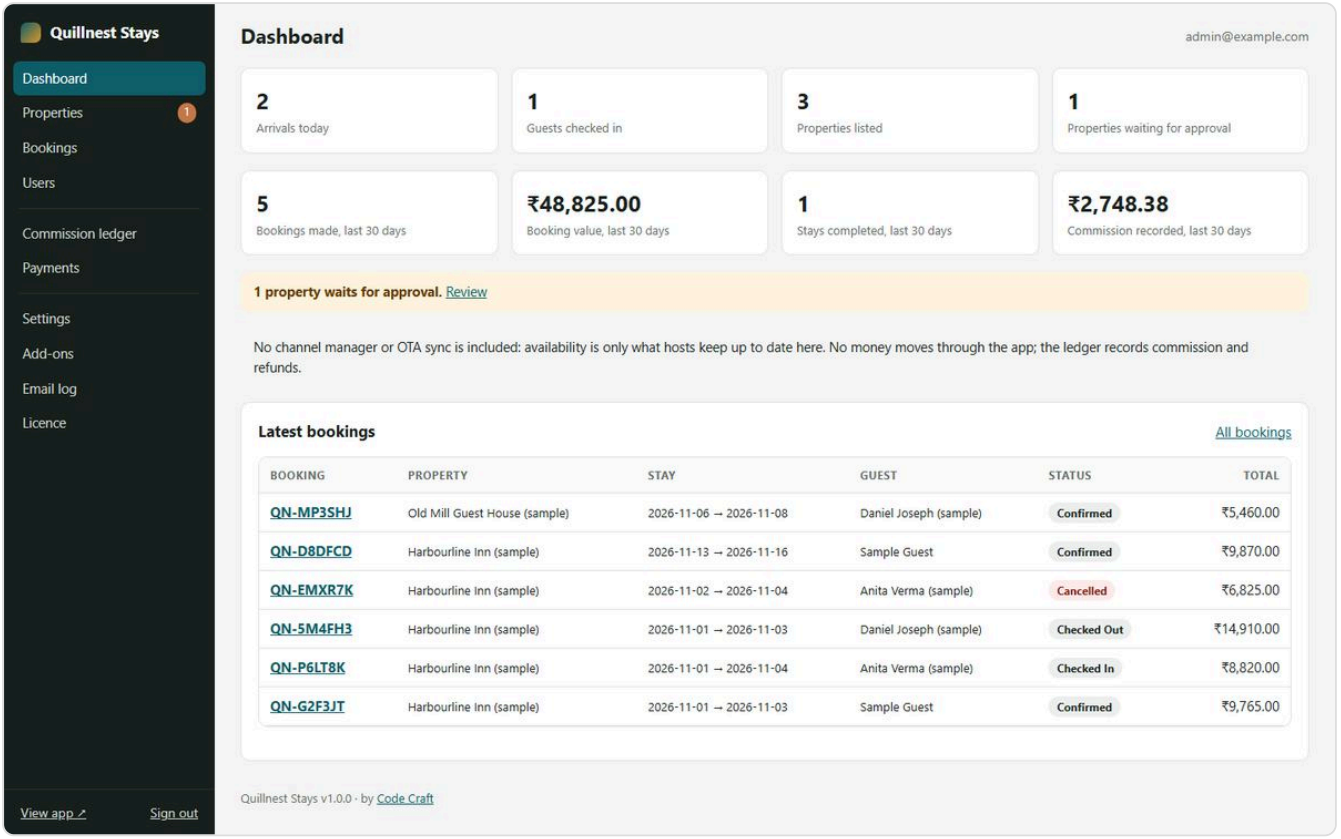
Owner → Rooms & rates.

5. Admin and going live

The admin approves, suspends or returns properties to pending, sets commission per property, sees all bookings (cancel, refunds, invoice, ID view), manages users, keeps the commission ledger (CSV) and records settlements. Settings cover business details, brand colours, SEO, maintenance, commission, tax slabs, stay rules, weekend nights, guest-ID rules (retention days), map, chat, WhatsApp, analytics, banner, cookie notice and SMTP.

1. Upload the files without `node_modules`, `.env`, `*.db` and the contents of `storage/guest-ids/`.
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`.
5. Open `/admin` and add your licence key in Admin → Licence. On a real domain new bookings stay 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 quillnest && pm2 save
```



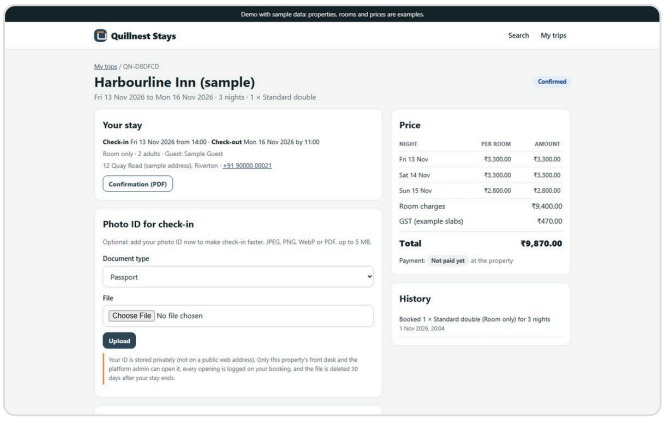
Admin dashboard.

6. Guest IDs, payments and maps

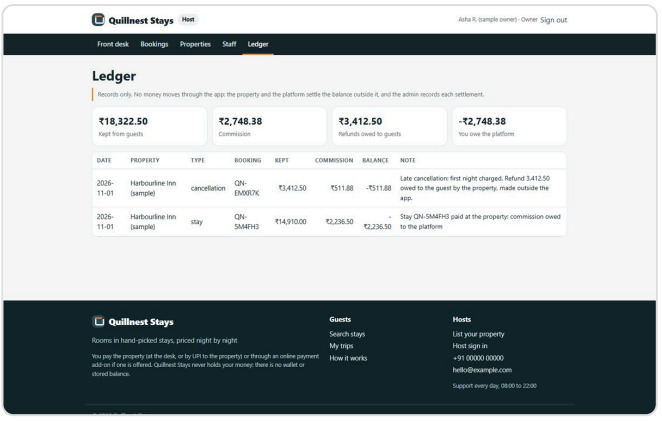
Guest IDs: JPEG, PNG, WebP or PDF up to 5 MB, stored in `storage/guest-ids/` under a random name. Only the property's owner and front desk (and the admin) can open it; every opening is logged on the booking. Files are deleted automatically `idRetentionDays` (default 30) after check-out or cancellation. The app does not file any police or immigration report for you: check the rules where your properties are.

Payments: no wallet and no stored balance. Guests pay the property at the desk, by UPI to the property's own UPI ID (a link and QR for the exact total, confirmed by the desk with the UTR) or through a gateway add-on. No gateway is bundled. The ledger is a record: paid at the property, the property owes the platform the commission; settlements are made outside the app.

Maps: property maps 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.



Guest → My trips: price, ID upload, confirmation PDF.



Owner → Ledger: a late cancellation and a completed stay.

7. REST API and the mobile app

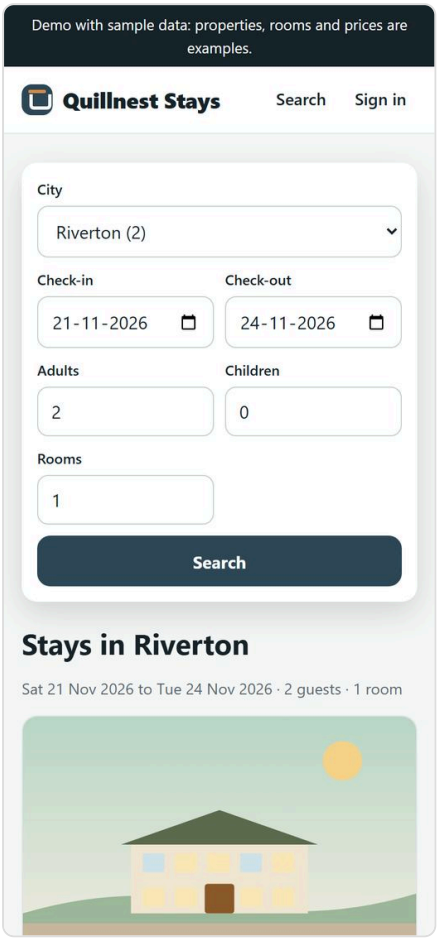
`/api/v1` uses bearer tokens (90 days). It covers cities, search, properties, the price calendar and quotes; a guest's bookings, cancellation, ID upload, review and invoice; the front desk's bookings, check-in, check-out, payment, no-show, refunds and ID; the owner's room calendar and ledger; and admin lists. See `API.md`.

```
curl "http://localhost:3540/api/v1/search?city=Riverton&checkIn=2026-12-01&checkOut=2026-12-03&adults=2"
```

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
```

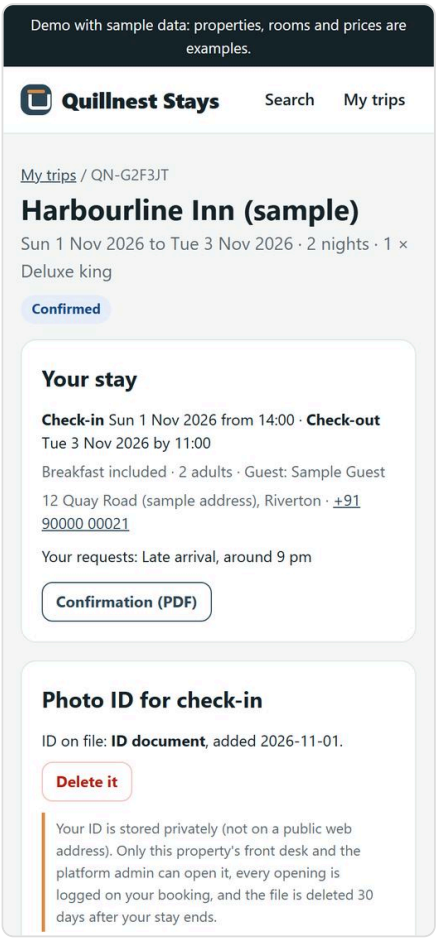
Guest mode: search, stay, price calendar, booking, my trips, cancellation preview and UPI. Front-desk mode for owners and staff: arrivals, departures, in house, check-in with room choice, payment with UTR, check-out and no-show. The mobile app cannot upload a guest ID (there is no file picker): guests and the desk upload it on the website, and check-in needs it when the ID rule is on. Owners manage rooms, prices and the ledger on the web.



Search at phone width.



Property.



My trip.

8. Honest limits and troubleshooting

- No channel manager or OTA sync, no housekeeping or live room-status module, no rate parity or yield management.
- Native Android and iOS builds have not been run by us: the app type-checks. Test your EAS builds on real phones before publishing.
- No payment gateway is bundled; UPI is confirmed by hand. No wallet and no automatic refunds. Gateway add-ons in this product family are tested only against mocked responses.
- Tests ran on SQLite only. OpenStreetMap tiles are for testing only.

Problem	Fix
A property does not show in search	It is pending approval, or a night in the range is full or closed.
Check-in refused: ID required	Upload the guest's ID on the booking page (website), or change the guest-ID rule in Settings.
Check-out refused	Record the payment first.
Bookings are 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.