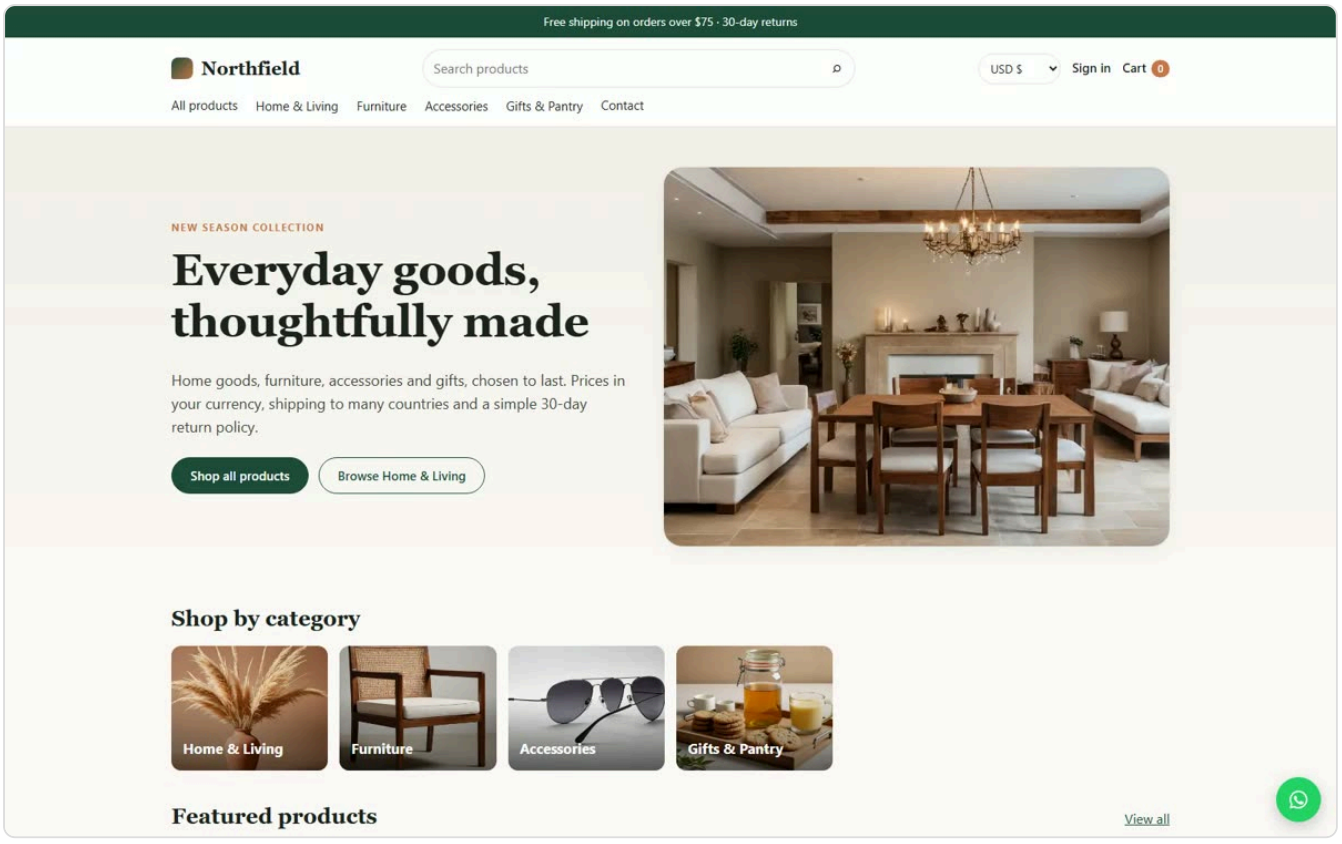
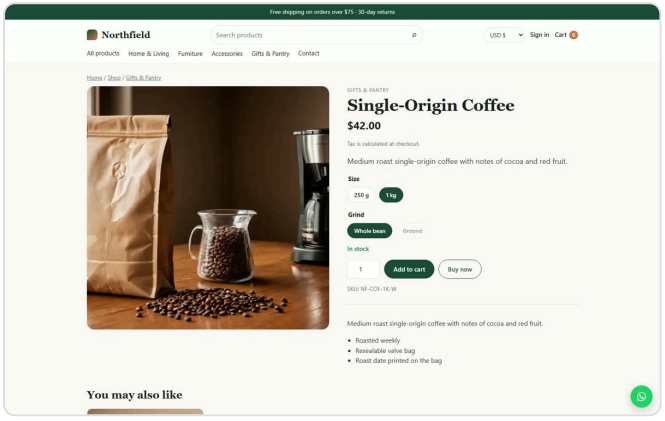
CODE CRAFT · INSTALLATION GUIDE

Node.js Ecommerce Store

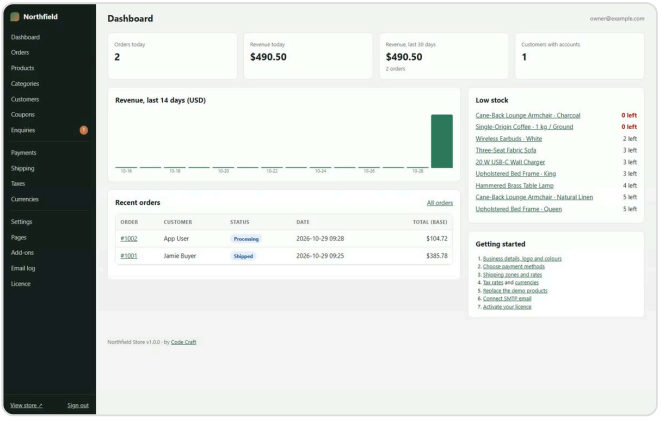
Northfield Store 1.0: Express + EJS + Prisma, with an admin panel, REST API and add-ons. This guide takes you from download to a live store on Hostinger, cPanel or a VPS.



The storefront with the demo catalogue. Every name, colour, photo and text can be changed from the admin.



Product with two options (Size and Grind); each variant has its own price and stock.



Admin dashboard: orders today, revenue, a 14-day chart and low stock.

1. What is in the download

Folder / file	What it is
`server.js`	Start file. Hostinger, cPanel and PM2 run this.
`src/`	The application: storefront, admin, REST API, pricing, orders, add-on loader.
`views/`, `public/`	Page templates (EJS), CSS, JavaScript and demo images.
`prisma/`	Database schema and demo data (24 products, 12 categories, 9 currencies).
`addons/`	Drop add-on folders here (Stripe, Razorpay, PayPal, Auto currency rates are sold separately).
`README.md`, `API.md`, `ADDONS.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 store.

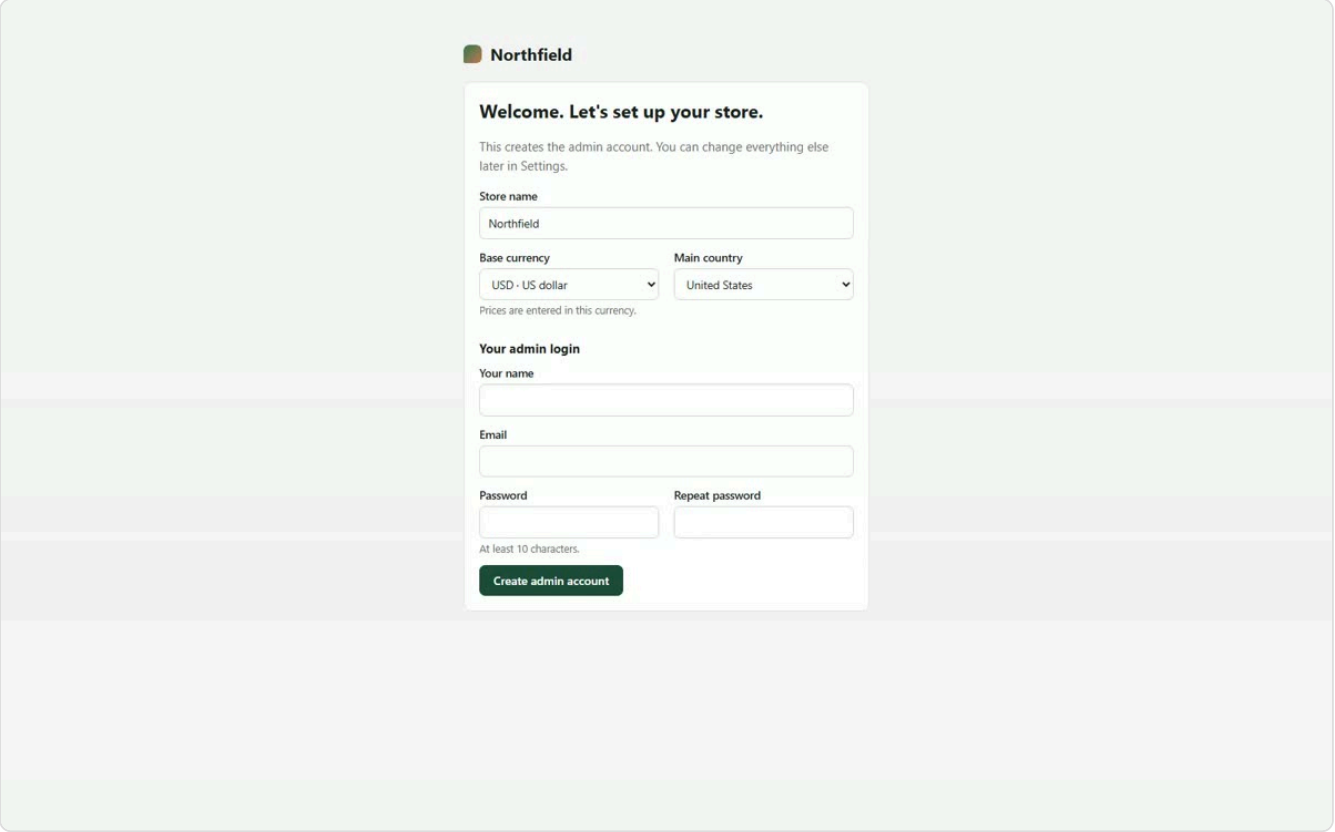
> The demo business details (123 Market Street, Your City, +1 555 010 0199) are placeholders. There are no reviews, ratings or sales numbers in the demo, real or invented.

2. Try it on your computer

1. Install Node.js from nodejs.org.
2. Unzip the download, open a terminal in the folder and run:

```
npm install        # packages + database client
npm run setup      # creates .env, the tables and the demo data
npm start          # open http://localhost:3420
```

3. Open `http://localhost:3420/admin`. The first visit shows the setup wizard: choose your store name, base currency and country, and create the admin login (password at least 10 characters).



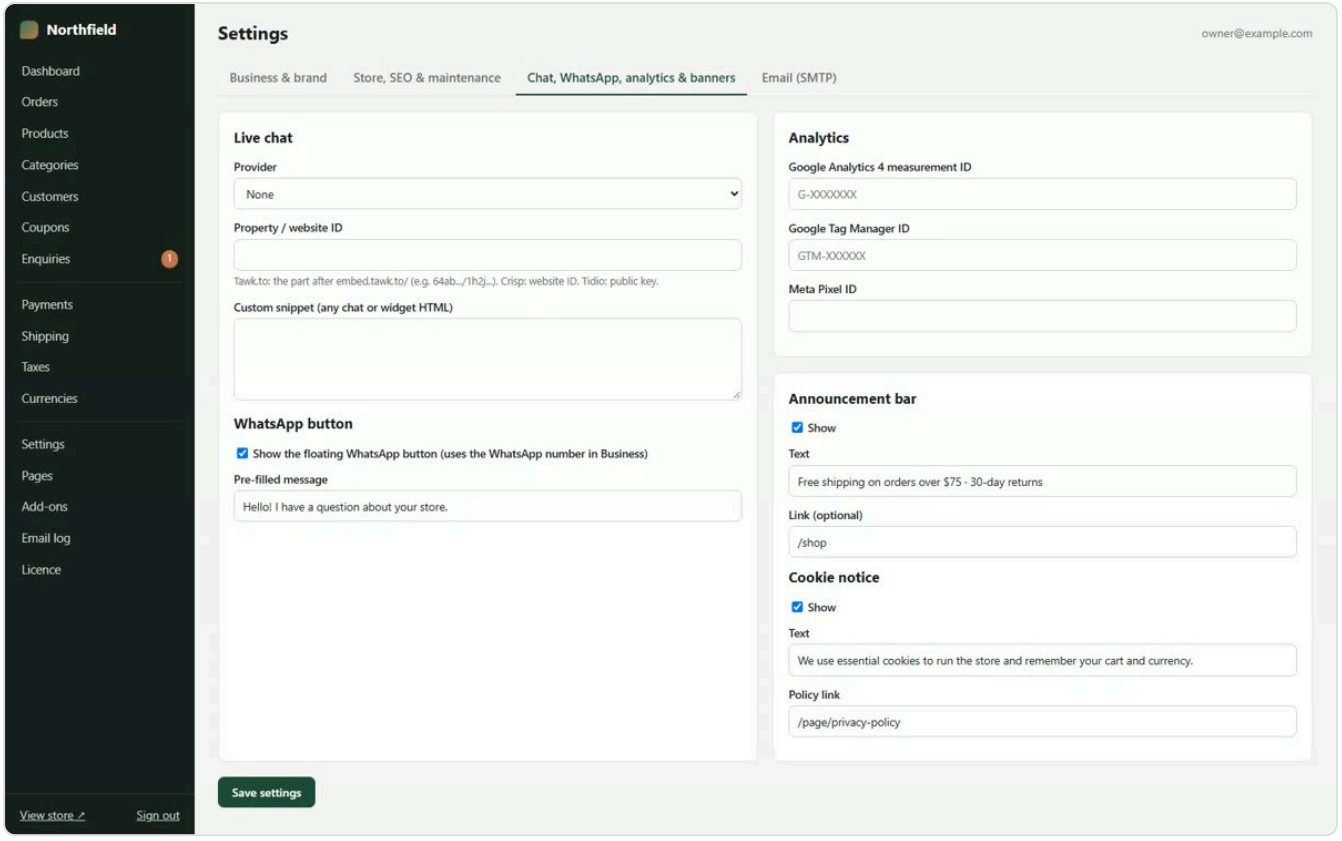
First-run setup wizard. It only appears while no admin account exists.

Useful variants: `npm run setup -- --no-seed` (empty store), `npm run setup -- --reseed` (reload the demo, keeps admin users), `npm run dev` (auto-restart while editing).

3. Set up your store in the admin

Business, brand and integrations (Settings)

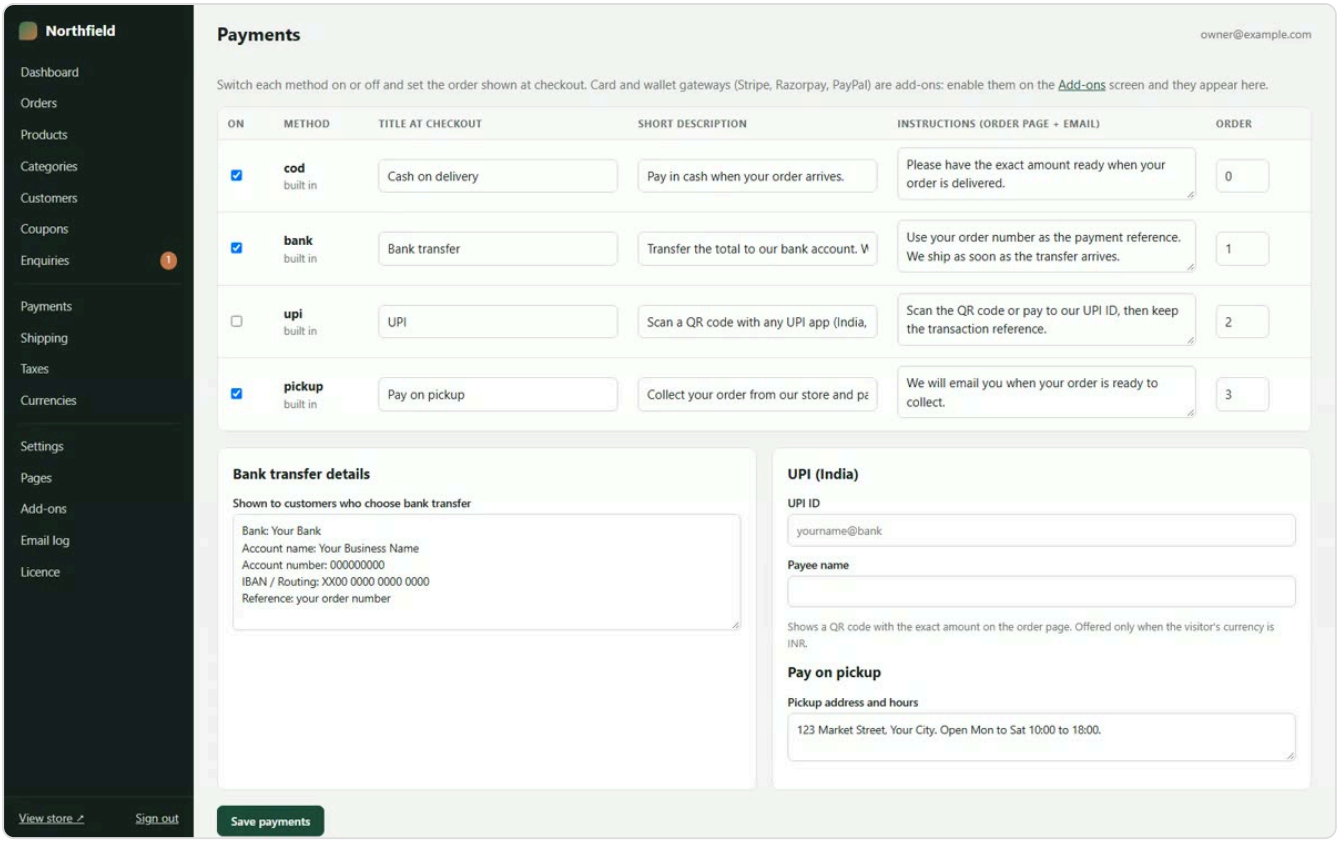
Store name, logo, colours, phone, WhatsApp number, email, address, map, hours and social links. The **Chat, WhatsApp, analytics & banners** tab holds live chat (Tawk.to, Crisp, Tidio or any snippet), the floating WhatsApp button, GA4 / GTM / Meta Pixel IDs, the announcement bar and the cookie notice. The **Store** tab has SEO defaults, guest checkout and maintenance mode.



Settings → Chat, WhatsApp, analytics & banners.

Payments

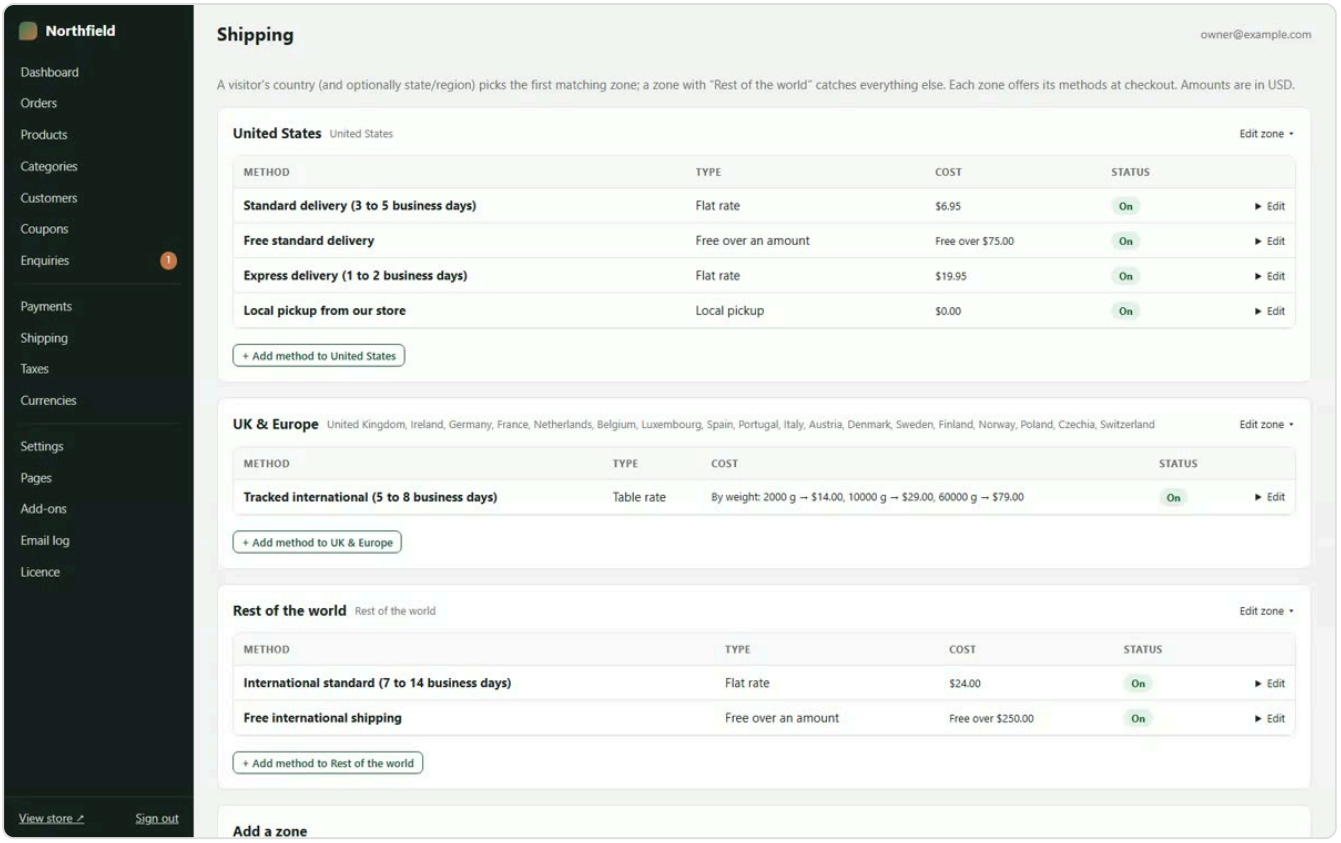
Switch Cash on delivery, Bank transfer, UPI (QR code with the exact amount, INR only) and Pay on pickup on or off, rename them and set their order. Enter your bank details and UPI ID here. Gateway add-ons add their own rows.



Admin → Payments.

Shipping zones and methods

A zone is a list of countries (optionally limited to regions/states). Each zone offers methods: flat rate, free over an amount, a table by weight or by order value, and local pickup. A "Rest of the world" zone catches every other country.



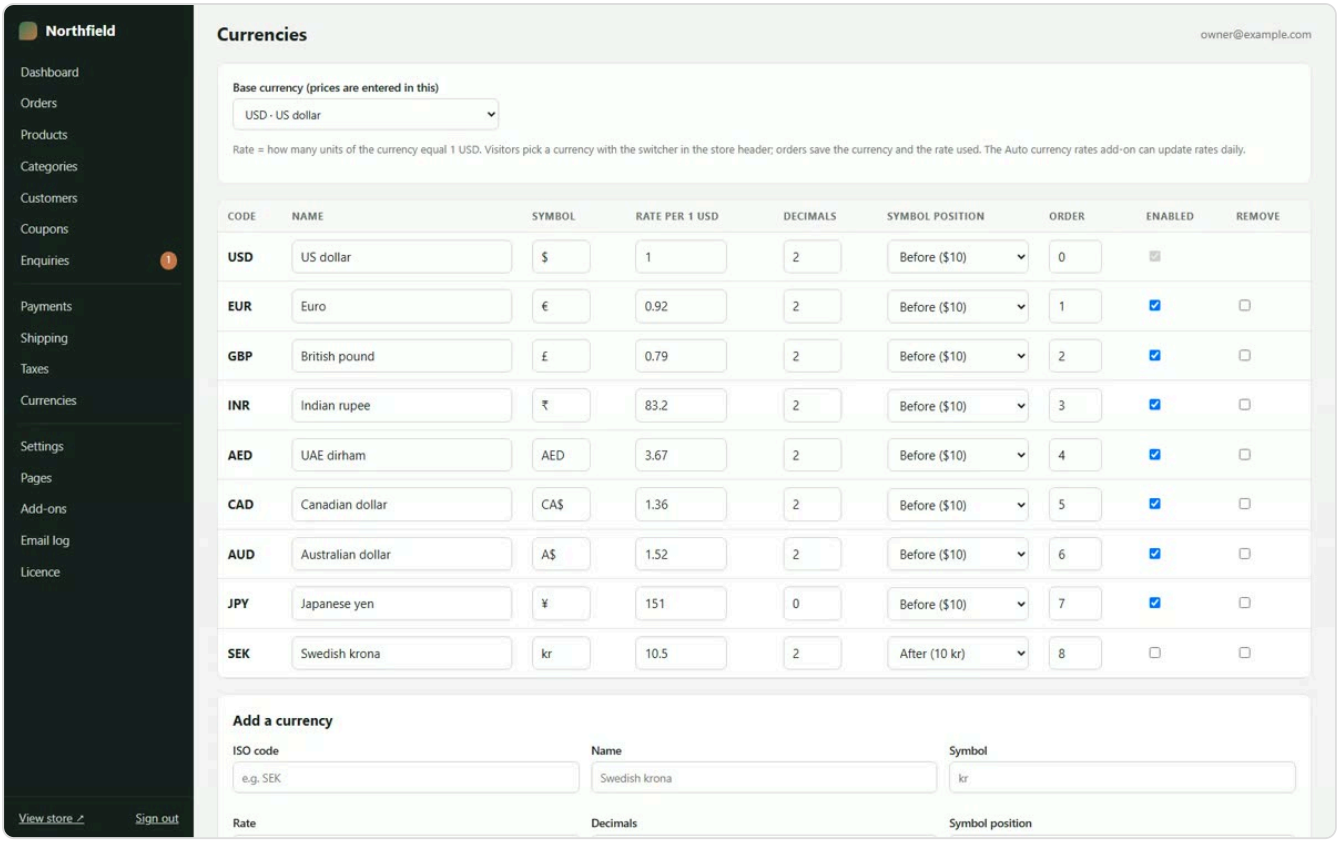
Admin → Shipping with the three demo zones.

Taxes

Add a rate per country, or per region inside a country (e.g. US state). Choose whether your prices already include tax and whether shipping is taxed.

Currencies

Pick the base currency (prices are entered in it) and enable more currencies, each with its rate, symbol, decimals and symbol position. Visitors choose with the switcher in the header; every order records the currency and the rate. The Auto currency rates add-on updates rates daily.

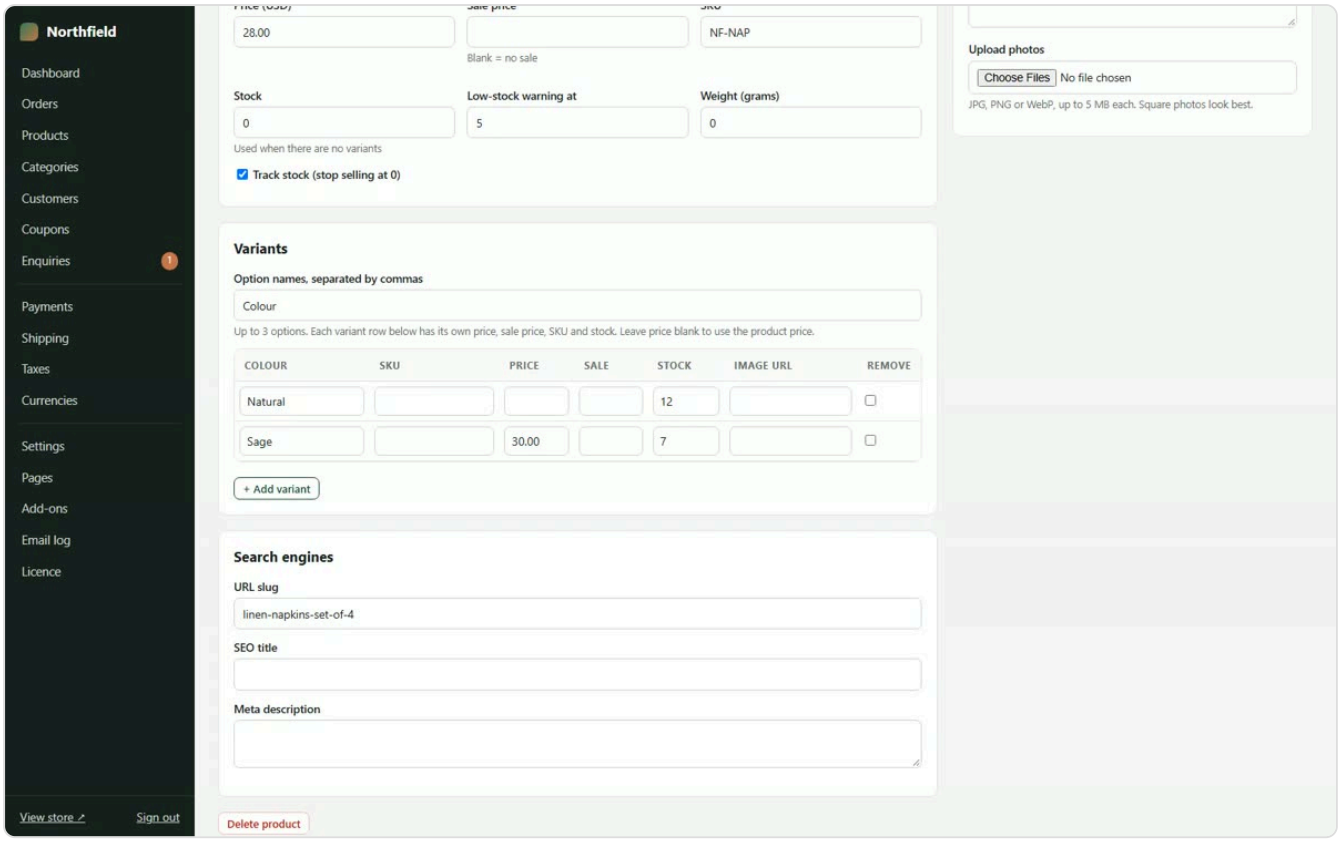


Admin → Currencies.

4. Products, coupons and orders

Products and variants

1. Products → Add product: name, descriptions, price, optional sale price, SKU, stock, weight (for weight-based shipping), category, photos, SEO title and description.
2. For options, type the option names (e.g. `Colour, Size`) and save. Then add one variant row per combination with its own price, sale price, SKU, stock and photo.



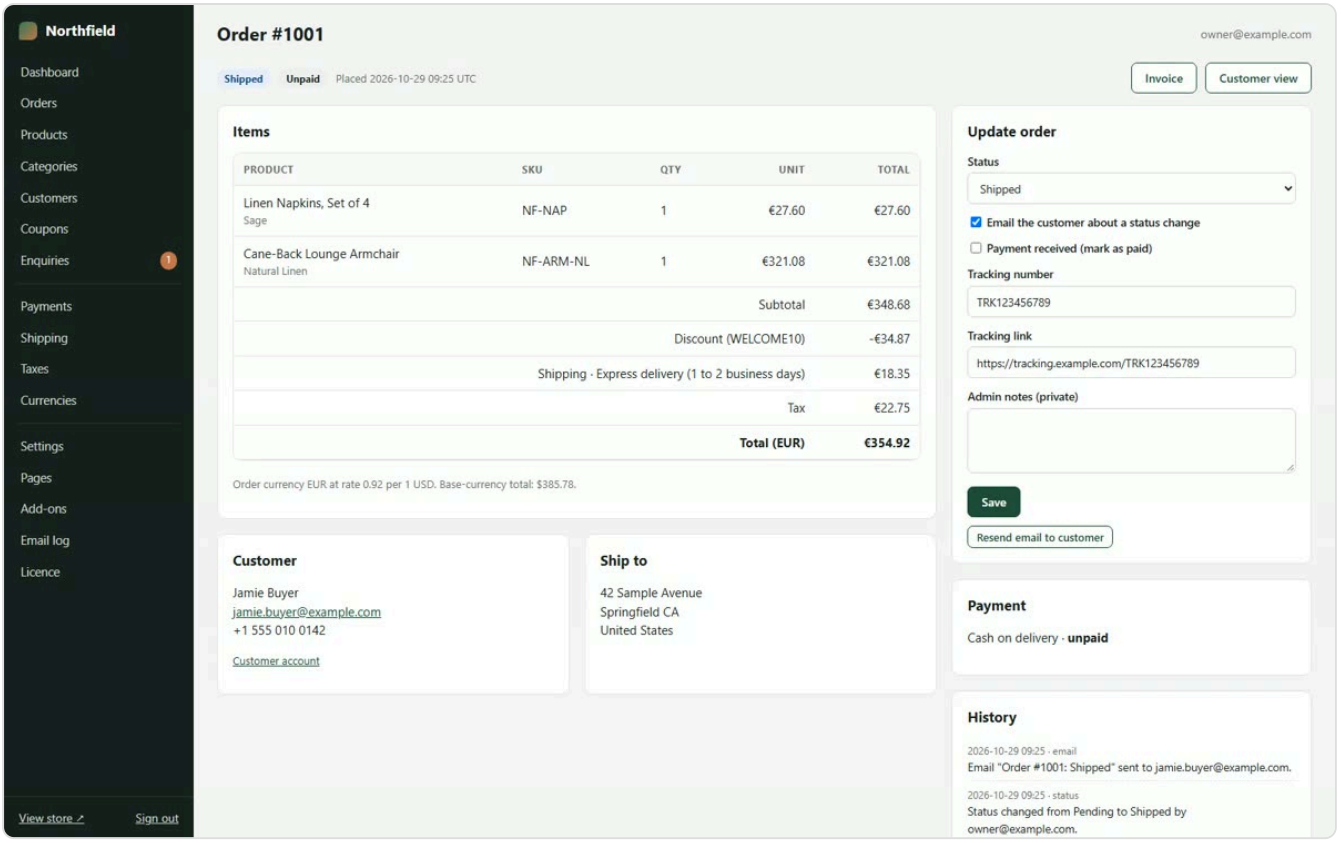
Variant editor: each row has its own SKU, price, sale price and stock.

Coupons

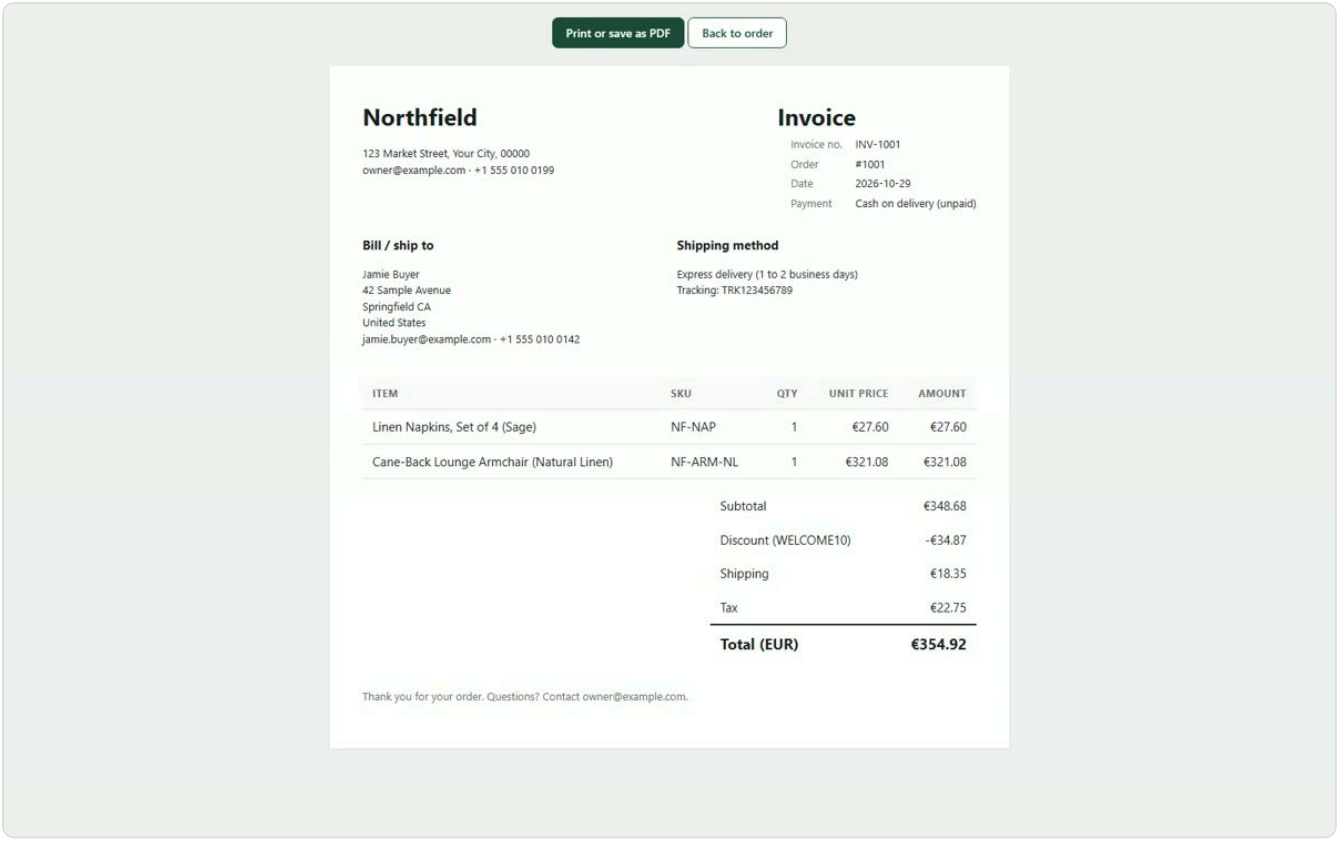
Percent or fixed amount, minimum spend, expiry date and usage limit. Codes are checked again on the server at checkout.

Orders

Statuses: pending → paid → processing → shipped → delivered, or cancelled / refunded (stock is put back). Add a tracking number and link, private notes, resend emails, print the invoice and export CSV.



Order screen with status, tracking, history and both currency amounts.



Printable invoice (use the browser's Save as PDF).

Email

Settings → Email (SMTP): host, port, username, password and From address from your mailbox provider. Until this is set, emails are written to Admin → Email log only.

5. Put it online

Hostinger (Node.js Web App)

1. hPanel → Websites → Add website → Node.js Web App. Upload the zip (or connect Git).
2. Node version 20 or 22. Entry file `server.js`. Build command `npm install && npm run setup`.
3. Environment variables: `APP_SECRET` (long random text), `APP_URL=https://your-domain`, `TRUST_PROXY=true`, and `DATABASE_URL` (SQLite `file:./store.db`, or the MySQL database you created in hPanel: `mysql://user:pass@localhost:3306/dbname`).
4. Deploy, then open `https://your-domain/admin` to run the setup wizard.

cPanel (Setup Node.js App)

1. Upload and extract the zip into a folder outside `public_html`, e.g. `~/store`.
2. cPanel → Setup Node.js App → Create application: Node 20/22, application root `store`, application URL your domain, startup file `server.js`.
3. Add the environment variables listed above. For MySQL, create the database and user in cPanel → MySQL Databases first.
4. Press **Run NPM Install**, then open the terminal (copy the "enter virtual environment" command shown at the top of the app page) and run `npm run setup`.
5. Restart the app and open `/admin`. Run AutoSSL so the site uses https.

VPS (Ubuntu, PM2 and Nginx)

```
cd /var/www/store            # the unzipped files
npm install && npm run setup
sudo npm i -g pm2
pm2 start server.js --name store && pm2 save && pm2 startup

# /etc/nginx/sites-available/store
server {
  server_name shop.example.com;
  client_max_body_size 10m;
  location / {
    proxy_pass http://127.0.0.1:3420;
    proxy_set_header Host $host;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
  }
}
sudo certbot --nginx -d shop.example.com
```

Using MySQL or PostgreSQL

Set `DATABASE_URL` to `mysql://…` or `postgresql://…` and run `npm run setup`; it switches the schema to that database, creates the tables and loads the demo data. (We tested on SQLite; MySQL 8+ and PostgreSQL run through the same Prisma schema.)

Backups

Copy `prisma/store.db` (SQLite) or use your host's database backup, and always copy `public/uploads/`.

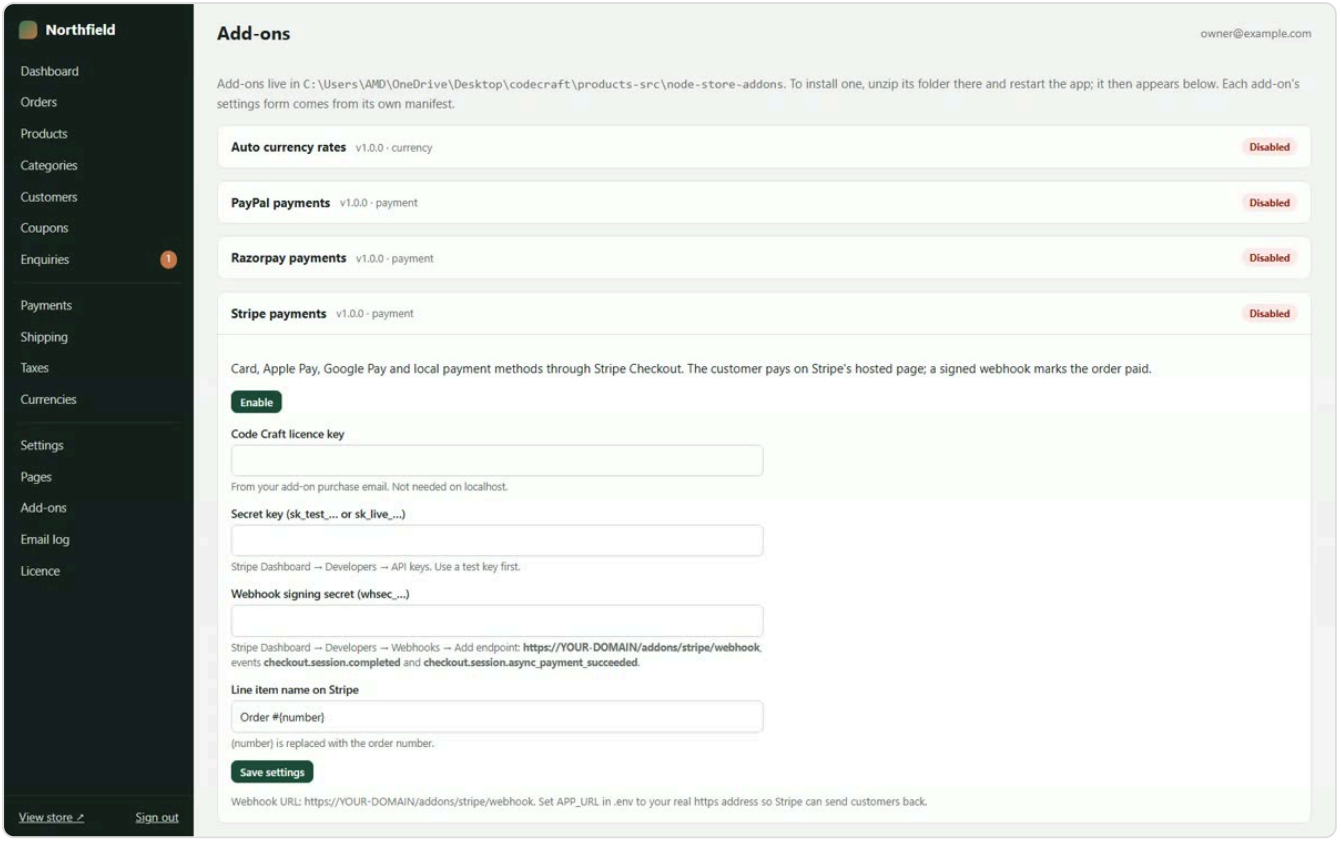
6. Licence, add-ons and the mobile API

Activate the licence

1. Admin → Licence: paste the key from your purchase email (also at coodecraft.com/account) and press **Save and activate**.
2. The store checks the key with coodecraft.com about once a week. If coodecraft.com cannot be reached, an active store keeps working for 30 days.
3. Until the licence is active on your domain, checkout and the contact form are switched off and the admin shows a banner. Localhost, 127.0.0.1, *.test and *.local never need a key.

Add-ons

Copy an add-on folder into `addons/`, restart, then enable it and fill in its settings in Admin → Add-ons. Payment add-ons appear in Admin → Payments and at checkout.



Admin → Add-ons with the Stripe add-on's settings form.

REST API

`/api/v1` serves store info, currencies, categories, products, a cart quote (shipping, tax, coupon and currency), checkout, customer accounts with bearer tokens, orders and the contact form. See `API.md` for every endpoint.

```
curl https://your-domain/api/v1/products?category=eyewear&currency=EUR
curl -X POST https://your-domain/api/v1/cart/quote -H "Content-Type: application/json" \
  -d '{"items":[{"productId":19,"qty":1}],"country":"GB","coupon":"WELCOME10"}'
```

7. Troubleshooting

Problem	Fix
"The database is not ready" at start	Run `npm run setup` in the app folder.
Checkout button disabled on the live site	Activate the licence in Admin → Licence.
Payment return links point to localhost	Set `APP_URL` to your https address and restart.
Logged out right away behind https	Set `TRUST_PROXY=true`.
No emails arrive	Check Admin → Email log; set SMTP in Settings → Email and use "Send a test email".
"Page expired" after a form	The form was open too long or cookies are blocked; reload and try again.
Forgot the admin password	Ask your developer to run a password reset with Prisma, or contact Code Craft support.

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