

Environment Setup

Drop Environment Configuration

Last updated: 2026-02-13 **Source:** [src/drop-app/package.json](#), [next.config.ts](#), [Dockerfile](#), [docker-compose.yml](#), [fly.toml](#)

Technology Stack

Layer	Technology	Version	Source
Runtime	Node.js	22 (Alpine)	Dockerfile:2
Framework	Next.js	16.1.6	package.json:14
UI	React	19.2.3	package.json:15-16
Database (all environments)	PostgreSQL 16 via Drizzle ORM	drizzle-orm	src/shared/db/schema.ts
Auth	JWT via jose	^6.1.3	package.json:8
Password hashing	bcryptjs	^3.0.3	package.json:5
Styling	Tailwind CSS	^4	package.json:33
UI Components	Radix UI	^1.4.3	package.json:13
Icons	Lucide React	^0.563.0	package.json:9
Theme	next-themes	^0.4.6	package.json:10
Toasts	Sonner	^2.0.7	package.json:17

Dev Dependencies

Tool	Version	Purpose	Source
Vitest	^4.0.18	Unit/integration testing	package.json:36
Playwright	^1.58.2	E2E testing	package.json:21
TypeScript	^5	Type checking	package.json:35
ESLint	^9	Linting	package.json:29

Tool	Version	Purpose	Source
shadcn	^3.8.4	UI component generation	package.json:32

NPM Scripts

Source: `src/drop-app/package.json:5-12`

Script	Command	Description
<code>dev</code>	<code>next dev</code>	Start development server (port 3000)
<code>build</code>	<code>next build</code>	Build for production (standalone output)
<code>start</code>	<code>next start</code>	Start production server
<code>lint</code>	<code>eslint</code>	Run ESLint
<code>test</code>	<code>vitest run</code>	Run unit/integration tests (single run)
<code>test:watch</code>	<code>vitest</code>	Run tests in watch mode

Next.js Configuration

Source: `src/drop-app/next.config.ts:1-49`

Setting	Value	Purpose
<code>output</code>	<code>"standalone"</code>	Self-contained server for Docker (<code>next.config.ts:4</code>)
<code>devIndicators</code>	<code>false</code>	Disable dev indicators (<code>next.config.ts:5</code>)

Security Headers

All responses include these headers (configured in `next.config.ts:6-58`):

Header	Value (Production)	Value (Development)	Purpose
--------	--------------------	---------------------	---------

Content-Security-Policy	<code>default-src 'self'; script-src 'self'; style-src 'self' 'unsafe-inline'; font-src 'self'; img-src 'self' data: blob;; connect-src 'self'; frame-ancestors 'none'</code>	<code>default-src 'self'; script-src 'self' 'unsafe-inline' 'unsafe-eval'; style-src 'self' 'unsafe-inline'; font-src 'self'; img-src 'self' data: blob;; connect-src 'self'; frame-ancestors 'none'</code>	XSS and injection protection
X-Frame-Options	<code>DENY</code>	<code>DENY</code>	Clickjacking prevention
X-Content-Type-Options	<code>nosniff</code>	<code>nosniff</code>	MIME sniffing prevention
Referrer-Policy	<code>strict-origin-when-cross-origin</code>	<code>strict-origin-when-cross-origin</code>	Referrer leakage prevention
Permissions-Policy	<code>camera=(self), microphone=(), geolocation=(self)</code>	<code>camera=(self), microphone=(), geolocation=(self)</code>	Feature restriction
Strict-Transport-Security	<code>max-age=63072000; includeSubDomains; preload</code>	<code>max-age=63072000; includeSubDomains; preload</code>	Force HTTPS

Note: CSP is stricter in production (no `unsafe-eval` for scripts). Development mode allows `unsafe-inline` and `unsafe-eval` for HMR (Hot Module Replacement) to work.

Environment Modes

Development

- `NODE_ENV=development` (default)
- Demo user seeded automatically
- Login page shows demo credentials hint
- In-memory rate limiting fallback
- PostgreSQL 16 via Docker (`docker compose up -d`), port 5433

Production

- `NODE_ENV=production`
- Demo seed data disabled
- `JWT_SECRET` required (fatal error if missing)
- Cookies set with `secure: true`
- PostgreSQL 16 on AWS RDS via `DATABASE_URL`

Test

- `NODE_ENV=test`
- PostgreSQL 16 test database (`drop_test`), created via `pg-test-db.ts` helper

- Tables truncated between tests; schema pushed via Drizzle before suite runs
- Mocked Next.js modules (server, headers)

Port Mapping

Service	Internal Port	External Port	Protocol
Drop App	3000	3000	HTTP
PostgreSQL (local dev)	5432	5433	TCP
PostgreSQL (production RDS)	5432	5432	TCP

Docker Image Details

Base: `node:22-alpine` **User:** `nextjs` (UID 1001) **Working dir:** `/app` **Exposed port:** 3000
Entrypoint: `node server.js` **Build context:** `src/drop-app/`

Image contents (runner stage):

- `/app/public/` -- Static assets
 - `/app/.next/standalone/` -- Next.js standalone server
 - `/app/.next/static/` -- Static build output
-

Revision #5

Created 2026-02-18 08:44:27 UTC by John

Updated 2026-05-23 10:58:10 UTC by John