

EC Site — Tech Stack & Architecture

Portfolio Project · Full Stack Engineer

Frontend

Tech	Role	Why
Next.js	Main framework	SEO friendly, fast, SSR/SSG for product pages
TypeScript	Language	Type safety, fewer bugs, enterprise standard
Tailwind CSS	Styling	Fast UI development, utility-first, responsive
Zustand	Cart state	Simple, lightweight state management
React Query	Data fetching	Caching, loading states, clean API calls

Backend

Tech	Role	Why
NestJS	API framework	Structured, TypeScript native, enterprise-ready
Prisma	ORM	Type-safe DB access, auto migrations
JWT	Authentication	Stateless auth for API, role-based access
Stripe	Payments	Industry standard, great docs, webhook support
class-validator	Validation	DTO-based input validation with decorators

Database

Tech	Role	Why
PostgreSQL	Main database	Reliable, relational, widely used in production
Redis	Caching (optional)	Cache product lists, sessions for performance

Storage & CDN

Tech	Role	Why
AWS S3	Image storage	Scalable, cheap, industry standard
CloudFront	CDN	Fast image & asset delivery worldwide

DevOps & Hosting

Tech	Role	Why
AWS EC2	App hosting	Runs NestJS backend API
AWS RDS	DB hosting	Managed PostgreSQL — backups, scaling, security

Vercel	Frontend hosting	Best for Next.js, zero config, free forever
DuckDNS	Free DNS	Points free domain to EC2 IP address
GitHub Actions	CI/CD	Auto deploy on push to main branch
PM2	Process manager	Keep NestJS running 24/7 on EC2
Nginx	Reverse proxy	Route traffic to NestJS on EC2

■ Final Free Setup Summary

Layer	URL / Service	Platform	Cost
Frontend (Next.js)	yourapp.vercel.app	Vercel	Free forever
Backend (NestJS)	your-api.duckdns.org	AWS EC2	Free 12 months
Database	Internal (VPC)	AWS RDS	Free 12 months
Images	CDN URL	AWS S3 + CloudFront	~Free small usage
DNS for Backend	your-api.duckdns.org	DuckDNS	Free forever
Payment	Stripe API	Stripe	Free (% per txn)
CI/CD	Auto deploy	GitHub Actions	Free forever

■■ System Architecture

```

User (Browser)
|
yourapp.vercel.app (Vercel -- Next.js Frontend, free)
| API calls
your-api.duckdns.org (DuckDNS -- free domain)
| points to
EC2 Instance
+-- Nginx (reverse proxy :80 -> :4000)
+-- NestJS API :4000
| |
RDS S3
(PostgreSQL) (Product Images)
|
CloudFront CDN
|
Stripe
(Payment + Webhooks)

```

■■ EC2 IP Problem & DuckDNS Solution

Problem: EC2 public IP changes every time you restart the instance

Solution: DuckDNS (free dynamic DNS)

1. Register at duckdns.org
2. Create subdomain: your-api.duckdns.org
3. Point it to your EC2 IP
4. Set env variable in Vercel:
NEXT_PUBLIC_API_URL=http://your-api.duckdns.org:4000

If EC2 IP changes -> update DuckDNS -> everything works again

■ Deployment Flow (CI/CD)

```
Developer pushes code to GitHub
|
GitHub Actions (CI/CD pipeline triggers)
|
Run tests -> Build -> SSH into EC2
|
Pull latest code -> Restart PM2
|
Live at your-api.duckdns.org + yourapp.vercel.app
```

■ User Roles

Role	Access	Auth Required
Guest	Browse products, view details	No
Customer	Cart, checkout, order history	Yes (JWT)
Admin	Manage products & orders	Yes (JWT + Role Guard)