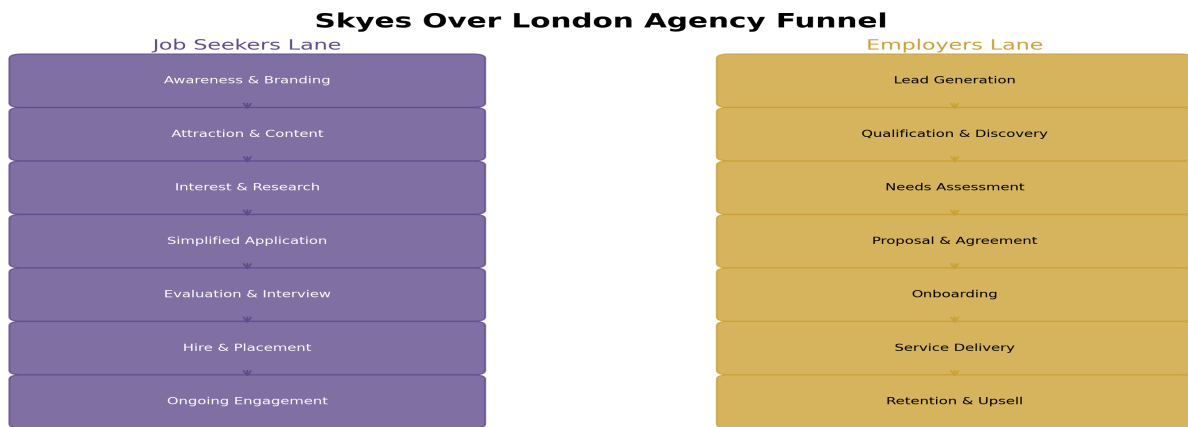


Skyes Over London

Dual-Lane Conversion Funnel System

Job Seekers + Employers | Netlify Forms + Functions + Blobs + Neon



What you have in hand

- Two separate conversion lanes: Job Seekers and Employers.
- Frictionless intake: single-page forms, honeypot, success routing.
- Drop-ready: Netlify Forms capture works immediately.
- Full stack: Functions write to Netlify Blobs (JSON retention) and Neon Postgres (structured records).

Build date: 2026-02-26

Executive Summary

This build is a production-ready, conversion-first intake website for Skyes Over London LC. It routes visitors into two distinct lanes, captures submissions via Netlify Forms, and (when deployed with Functions enabled) duplicates intake into Netlify Blobs and Neon Postgres for operational-grade retention and reporting.

Lane Map

- **Job Seekers** - awareness, attraction, research, simplified application, evaluation, hire, engagement.
- **Employers** - lead generation, qualification, needs assessment, proposal, onboarding, service delivery, retention.

System Architecture

- Front end: static pages in /public with a shared CSS + JS bundle.
- Intake: Netlify Forms (form names: jobseekers-intake, employers-intake).
- Backend: Netlify Functions (/intake and /health).
- Storage: Netlify Blobs store 'sol-intake' (JSON) + Neon table 'intake_submissions' (rows).

Endpoints

Method	Path	Purpose
POST	/ (Netlify Forms)	Stores submissions in Netlify Forms
POST	/.netlify/functions/intake	Writes JSON to Blobs + row to Neon (when env set)
GET	/.netlify/functions/health	Diagnostics: checks Functions + Blobs + DB env

What's Included

Website pages

- /index.html (lane chooser + diagram)
- /jobseekers.html (job seeker funnel + intake form)
- /employers.html (employer funnel + intake form)
- /thank-you.html (post-submit landing)
- /diagnostics.html (live health panel)
- /privacy.html and /terms.html (baseline legal pages)

Repository structure (as shipped)

```
- .env.example
- README.md
- README_DEPLOY.md
- netlify.toml
- package-lock.json
- package.json
- public/diagnostics.html
- public/employers.html
- public/index.html
- public/jobseekers.html
- public/privacy.html
- public/terms.html
- public/thank-you.html
- public/assets/funnel_diagram.png
- public/assets/intake.js
- public/assets/style.css
- netlify/functions/health.js
- netlify/functions/intake.js
```

Branding rules implemented

- Purple + gold system palette.
- Floating logo placement on pages (in site HTML) plus watermark layer (in site HTML).
- Clean enterprise layout: high contrast, large buttons, single primary action per screen.

Deployment and Environment

Deployment modes

- **Drop deploy:** Fastest. Publishes static /public. Netlify Forms capture works. Functions/Neon/Blobs may not run.
- **Git-connected deploy:** Recommended for full system. Functions are bundled; Blobs + Neon writes work.
- **Netlify CLI (netlify dev / deploy):** Great for Codespace testing and controlled deploys.

Environment variables

Variable	Meaning
NEON_DATABASE_URL	Preferred Neon Postgres connection string
DATABASE_URL	Alternate name (supported)
NODE_VERSION	Optional; recommended 20

Codespace quickstart

```
npm install
cp .env.example .env (paste your Neon connection string)
npm i -g netlify-cli
netlify dev
```

Verification checklist

- Submit each lane once; confirm entries appear in Netlify Forms.
- Open /diagnostics.html; confirm /health returns JSON when Functions are live.
- If DB enabled: verify Neon table intake_submissions has rows.
- If Blobs enabled: confirm Blobs store sol-intake is available in diagnostics.

Security and Operational Notes

- Spam protection: Netlify honeypot field (bot-field) on both forms.
- Headers: X-Frame-Options DENY, X-Content-Type-Options nosniff, strict referrer policy, and permissions policy baseline (netlify.toml).
- PII discipline: forms collect the minimum needed for first contact; richer details can be collected after qualification.
- Retention: dual-write pattern (Forms + Blobs + Neon) supports auditability and reporting.

Data model (Neon)

Table: `intake_submissions`

Key columns: `lane, name, email, phone, company, role, ip, user_agent, payload (JSONB), created_at`

Created automatically on first successful write from `/.netlify/functions/intake`.

Blob retention

Blob store: `sol-intake`

Usage: JSON payloads are saved with timestamped keys to provide redundant retention and easy export.