/docs/component-registry.json

{

"$schema": "https://json-schema.org/draft/2020-12/schema",

"registryVersion": "1.0.0",

"components": [

{

"sectionId": "hero",

"component": "Hero",

"requiredProps": ["eyebrow", "title", "subtitle"],

"optionalProps": ["primaryCta", "secondaryCta", "media", "theme", "backgroundEffect"],

"propSchema": {

"eyebrow": { "type": "string", "maxLength": 80 },

"title": { "type": "string", "maxLength": 120 },

"subtitle": { "type": "string", "maxLength": 280 },

"primaryCta": { "type": "object", "properties": { "label": {"type":"string"}, "href": {"type":"string"} }, "required": ["label","href"] },

"secondaryCta": { "type": "object", "properties": { "label": {"type":"string"}, "href": {"type":"string"} } },

"media": { "type": "object", "properties": { "assetId": {"type":"string"}, "alt": {"type":"string"} } },

"theme": { "type": "string", "enum": ["light","dark","brand","contrast"] },

"backgroundEffect": { "type": "string", "enum": ["none","stars","globe","grid"] }

}

},

{

"sectionId": "features",

"component": "FeatureGrid",

"requiredProps": ["title", "items"],

"optionalProps": ["kicker", "columns"],

"propSchema": {

"title": { "type": "string" },

"kicker": { "type": "string" },

"columns": { "type": "integer", "minimum": 2, "maximum": 4 },

"items": {

"type": "array",

"minItems": 3,

"items": {

"type": "object",

"properties": {

"icon": { "type":"string" },

"title": { "type":"string" },

"copy": { "type":"string", "maxLength": 220 },

"cta": { "type": "object", "properties": { "label":{"type":"string"}, "href":{"type":"string"} } }

},

"required": ["title","copy"]

}

}

}

},

{

"sectionId": "testimonials",

"component": "Testimonials",

"requiredProps": ["items"],

"optionalProps": ["title"],

"propSchema": {

"title": { "type": "string" },

"items": {

"type": "array",

"minItems": 2,

"items": {

"type": "object",

"properties": {

"quote": { "type":"string", "maxLength": 320 },

"author": { "type":"string" },

"role": { "type":"string" },

"logoAssetId": { "type":"string" }

},

"required": ["quote","author"]

}

}

}

},

{

"sectionId": "pricing",

"component": "Pricing",

"requiredProps": ["plans"],

"propSchema": {

"plans": { "type":"array", "items": {

"type":"object",

"properties": {

"name": {"type":"string"},

"price": {"type":"string"},

"period": {"type":"string","enum": ["mo","yr"]},

"features": {"type":"array","items":{"type":"string"}},

"cta": { "type":"object","properties":{"label":{"type":"string"},"href":{"type":"string"}}}

},

"required": ["name","price","features"]

} }

}

},

{

"sectionId": "cta",

"component": "Callout",

"requiredProps": ["title","primaryCta"],

"propSchema": {

"title": {"type":"string"},

"copy": {"type":"string"},

"primaryCta": { "type":"object","properties":{"label":{"type":"string"},"href":{"type":"string"}},"required":["label","href"] }

}

},

{

"sectionId": "blogList",

"component": "BlogList",

"requiredProps": ["items"],

"propSchema": {

"items": { "type":"array","items": {

"type":"object",

"properties": {

"slug": {"type":"string"},

"title": {"type":"string"},

"excerpt": {"type":"string"},

"date": {"type":"string", "format": "date"},

"readingTime": {"type":"integer"},

"coverAssetId": {"type":"string"}

},

"required": ["slug","title","excerpt","date"]

} }

}

}

],

"rules": {

"rejectUnknownProps": true,

"maxSectionsPerPage": 12

}

}

---

/docs/routes/redirects.csv

from,to,type,updated\_at,notes

/old-home,/ ,301,2025-09-20,Migrated to new homepage

/blog,/insights,302,2025-09-20,Temporary during content merge

/case-study-ai,/work/ai-automation,301,2025-09-20,Canonicalized

---

/docs/forms/contact.json

{

"formId": "contact",

"version": "1.0.0",

"method": "POST",

"endpoint": "/v1/forms/contact",

"idempotency": true,

"fields": [

{"name":"name","type":"text","label":"Your Name","required":true,"minLength":2},

{"name":"email","type":"email","label":"Email","required":true},

{"name":"company","type":"text","label":"Company","required":false},

{"name":"message","type":"textarea","label":"How can we help?","required":true,"minLength":20},

{"name":"consent","type":"checkbox","label":"I agree to the privacy policy","required":true}

],

"spam": {"honeypot":"website","timeMinMs":1500},

"successRedirect": "/thanks",

"notifications": {"to":["sales@hisl.ai"],"subject":"New Contact Form"}

}

/docs/forms/rfp.json

{

"formId": "rfp",

"version": "1.0.0",

"endpoint": "/v1/forms/rfp",

"idempotency": true,

"fields": [

{"name":"name","type":"text","label":"Name","required":true},

{"name":"email","type":"email","label":"Email","required":true},

{"name":"org","type":"text","label":"Organisation","required":true},

{"name":"budget","type":"select","label":"Budget","options":["<£25k","£25k–£75k","£75k–£150k",">£150k"],"required":true},

{"name":"timeline","type":"select","label":"Timeline","options":["<1 mo","1–3 mo","3–6 mo",">6 mo"],"required":true},

{"name":"brief","type":"textarea","label":"Project Brief","required":true,"minLength":80},

{"name":"attachments","type":"file","label":"Attachments","accept":[".pdf",".docx",".pptx"],"maxFiles":3}

],

"successRedirect": "/thanks-rfp",

"notifications": {"to":["newbusiness@hisl.ai"],"subject":"RFP Submission"}

}

/docs/forms/newsletter.json

{

"formId": "newsletter",

"version": "1.0.0",

"endpoint": "/v1/forms/newsletter",

"idempotency": true,

"fields": [

{"name":"email","type":"email","label":"Email","required":true},

{"name":"consent","type":"checkbox","label":"Receive updates","required":true}

],

"successRedirect": "/subscribed"

}

---

/docs/assets/asset\_manifest.schema.json

{

"$schema": "https://json-schema.org/draft/2020-12/schema",

"title": "Asset Manifest Schema",

"type": "object",

"properties": {

"version": { "type": "string" },

"assets": {

"type": "array",

"items": {

"type": "object",

"required": ["assetId","path","alt"],

"properties": {

"assetId": {"type":"string", "description":"Stable ID used in content"},

"path": {"type":"string", "description":"/public/images/..."},

"alt": {"type":"string"},

"caption": {"type":"string"},

"license": {"type":"string"},

"source": {"type":"string"},

"width": {"type":"integer"},

"height": {"type":"integer"},

"orientation": {"type":"string","enum":["landscape","portrait","square"]},

"focalPoint": {"type":"array","items":{"type":"number"},"minItems":2,"maxItems":2},

"colorProfile": {"type":"string"},

"tags": {"type":"array","items":{"type":"string"}}

}

}

}

},

"required": ["version","assets"]

}

/docs/assets/asset\_manifest.json (sample)

{

"version": "1.0.0",

"assets": [

{

"assetId": "globe-hero",

"path": "/images/hero/globe.webp",

"alt": "Wireframe globe representing AI-enabled operations",

"caption": "Realtime globe used on homepage",

"license": "HISL © 2025",

"width": 1920,

"height": 1080,

"orientation": "landscape",

"focalPoint": [0.52, 0.44],

"tags": ["hero","r3f","brand"]

}

]

}

---

/docs/api/v1\_contract.md

# IntegAI/HISL Website API – v1 Contract

## Principles

- Backwards compatible until v2.

- Deterministic GETs (cacheable).

- All POST endpoints support `Idempotency-Key`.

## Endpoints

### GET /v1/site/menu

Returns the global nav, footer and utility links.

200: { "menu": {...}, "footer": {...} }

### GET /v1/site/page/:slug

Returns the page definition (ordered sections) using component registry IDs.

200: {

"slug": "about",

"title": "About HISL",

"sections": [ { "sectionId":"hero", "props": {...} }, ... ],

"seo": { "title": "...", "description": "...", "imageAssetId": "..." }

}

### GET /v1/site/seo/:slug

Returns SEO fields for pre-render and sitemap.

### GET /v1/site/search?q=

Basic site search for Insights/Work.

### POST /v1/forms/:formId

Headers: `Idempotency-Key: <uuid>`

Body: { "fields": {...}, "meta": {...} }

201 on first accepted, 200 on duplicate idempotent replay.

### POST /v1/publish

Auth required. Triggers ingestion of a `.integpkg` and cache invalidation.

## Status Codes

- 200 OK / 201 Created / 202 Accepted

- 400 ValidationError (with JSON pointer details)

- 401/403 Auth errors

- 404 Not Found

- 409 Conflict (Idempotency key re-use with different payload)

- 500 ServerError (errorId)

---

/docs/ops/cache.md

# Cache & Invalidation

- HTML/SSR: 5 minutes TTL, stale-while-revalidate 60 minutes.

- /v1/site/page/\* and /v1/site/seo/\*: 10 minutes TTL, SWR 60 minutes.

- /v1/forms/\*: no-store.

- Static assets under /\_next/static and /images: immutable, hash-named.

## Invalidation

- On publish, purge the following keys:

- page:\* for changed slugs

- seo:\* for changed slugs

- sitemap.xml and / if nav changed

- Warm following slugs post-publish (see `cache-warm-list.txt`).

---

/src/styles/tokens.json

{

"version": "1.0.0",

"color": {

"background": { "base": "#0B1020", "elevated": "#0F152A", "inverted": "#FFFFFF" },

"text": { "primary": "#E6E9F2", "secondary": "#A7B0C0", "inverted": "#0B1020", "muted": "#8A93A5" },

"brand": { "primary": "#24D17E", "secondary": "#F7C948", "accent": "#3BC7FF" },

"border": { "soft": "#1D2746", "strong": "#2B3760" },

"states": { "success": "#2BD576", "warning": "#FFC857", "danger": "#FF5A5F", "info": "#5FB3FF" }

},

"typography": {

"fontFamily": { "sans": "Inter, ui-sans-serif, system-ui", "mono": "JetBrains Mono, ui-monospace" },

"sizes": { "xs":"12px","sm":"14px","md":"16px","lg":"18px","xl":"22px","2xl":"28px","3xl":"36px","4xl":"48px" },

"lineHeights": { "tight": 1.2, "snug": 1.35, "normal": 1.5 }

},

"radius": { "sm":"8px","md":"12px","lg":"16px","xl":"24px","pill":"9999px" },

"shadow": { "sm":"0 1px 2px rgba(0,0,0,.2)","md":"0 6px 18px rgba(3,6,23,.35)","lg":"0 12px 40px rgba(3,6,23,.45)" },

"spacing": { "xs":4,"sm":8,"md":12,"lg":16,"xl":24,"2xl":32,"3xl":48 },

"motion": {

"duration": { "fast": 120, "base": 220, "slow": 420 },

"easing": { "standard":"cubic-bezier(0.2, 0, 0.2, 1)", "emphasized":"cubic-bezier(0.2, 0, 0, 1)" },

"maxParallaxPx": 18

}

}

---

/scripts/generate-imagery-manifest.mjs

#!/usr/bin/env node

import { readdirSync, statSync, writeFileSync } from "fs";

import { join, parse } from "path";

const ROOT = process.cwd();

const IMAGES\_DIR = join(ROOT, "public", "images");

const OUT = join(ROOT, "docs", "assets", "asset\_manifest.json");

function walk(dir, out = []) {

for (const entry of readdirSync(dir)) {

const p = join(dir, entry);

const s = statSync(p);

if (s.isDirectory()) walk(p, out);

else if (/\.(png|jpe?g|webp|svg)$/i.test(p)) out.push(p);

}

return out;

}

const files = walk(IMAGES\_DIR);

const assets = files.map((p) => {

const rel = p.replace(ROOT, "").replace(/\\/g, "/");

const id = parse(p).name.toLowerCase().replace(/[^a-z0-9]+/g,"-");

return { assetId: id, path: rel, alt: "", tags: [] };

});

writeFileSync(OUT, JSON.stringify({ version: "1.0.0", assets }, null, 2));

console.log(`Wrote ${assets.length} assets to ${OUT}`);

---

/docs/ux/motion.md

# Motion Guardrails

- Respect `prefers-reduced-motion`: disable parallax, particle fields, and heavy R3F scenes; swap to static images.

- Cap interaction-driven parallax to `tokens.motion.maxParallaxPx`.

- Duration: 120–420ms; avoid >600ms except page transitions.

- Easing: standard cubic-bezier(0.2,0,0.2,1). Avoid bounce.

- Idle CPU < 5%: pause effects when tab is hidden or out of viewport.

- Never block First Contentful Paint; load heavy scenes after LCP.

---

/lighthouserc.json

{

"ci": {

"assert": {

"assertions": {

"categories:performance": ["error", {"minScore": 0.9}],

"first-contentful-paint": ["warn", {"maxNumericValue": 1800}],

"largest-contentful-paint": ["error", {"maxNumericValue": 2200}],

"total-blocking-time": ["error", {"maxNumericValue": 150}],

"cumulative-layout-shift": ["error", {"maxNumericValue": 0.08}],

"uses-responsive-images": "error",

"color-contrast": "error",

"uses-rel-preload": "warn"

}

},

"collect": { "url": ["http://localhost:3000/","http://localhost:3000/insights"], "numberOfRuns": 1 }

}

}

---

/tests/load/k6-smoke.js

import http from 'k6/http';

import { sleep, check } from 'k6';

export const options = {

vus: 10, duration: '1m',

thresholds: { http\_req\_duration: ['p(95)<800'], http\_req\_failed: ['rate<0.005'] }

};

const pages = ['/', '/about', '/work', '/insights'];

export default function () {

for (const p of pages) {

const res = http.get(`${\_\_ENV.BASE\_URL}${p}`, { tags: { page: p }});

check(res, { 'status 200': (r) => r.status === 200 });

sleep(0.5);

}

}

---

/docs/ops/telemetry.md

# Telemetry & Analytics

## Sentry (errors)

- DSN via `SENTRY\_DSN` (optional).

- Capture frontend errors and SSR exceptions; scrub PII (emails, phone, free-text fields).

- Include release + commit SHA.

## PostHog (product analytics)

- Key `POSTHOG\_KEY`, host `POSTHOG\_HOST` (optional).

- Autocapture disabled; define explicit events only:

- `cta\_click` { label, href }

- `form\_submit` { formId, ok, code }

- `feature\_toggle\_seen` { flag, on }

- Respect DNT and cookie consent.

## Langfuse / LLM spans (if used)

- Keys `LANGFUSE\_PUBLIC\_KEY`/`LANGFUSE\_SECRET\_KEY`.

- Mask user-input fields, store prompts w/ hashing.

## Sampling

- Default sample rate: 10% for sessions, 100% for errors.

## Data retention

- Error data 90 days; analytics events 180 days.

---

/docs/sop/publish.md

# Publishing SOP

1. \*\*Prepare package\*\*

- Run `npm run build`.

- Run `node scripts/publish.mjs pack` → creates `dist/site.integpkg`.

2. \*\*Ship to storage\*\*

- Upload `site.integpkg` to `s3://hisl-prod/site/` (or MinIO bucket).

3. \*\*Trigger publish\*\*

- Call `/v1/publish` with package URL and changelog.

4. \*\*Invalidate caches\*\*

- Purge changed slugs (page:\* and seo:\*).

- Warm cache using `docs/routes/cache-warm-list.txt`.

5. \*\*Verify\*\*

- Check CI Lighthouse and k6 dashboards.

- Spot-check forms (idempotent replays return 200).

---

/scripts/publish.mjs

#!/usr/bin/env node

import { execSync } from "child\_process";

import { mkdirSync, writeFileSync } from "fs";

import { join } from "path";

const cmd = process.argv[2];

if (cmd === "pack") {

mkdirSync("dist", { recursive: true });

execSync("next build", { stdio: "inherit" });

// minimal package descriptor

const pkg = { createdAt: new Date().toISOString(), commit: process.env.GITHUB\_SHA || "" };

writeFileSync(join("dist","site.integpkg"), JSON.stringify(pkg, null, 2));

console.log("Wrote dist/site.integpkg");

} else {

console.log("Usage: node scripts/publish.mjs pack");

}

---

/docs/routes/cache-warm-list.txt

/

/about

/work

/insights

/contact

---

/docs/qa/a11y.md

# Accessibility (WCAG 2.2 AA)

- Color contrast ≥ 4.5:1 for body text; ≥ 3:1 for large text.

- All images have alt text from asset manifest.

- Keyboard: visible focus ring; all interactive elements reachable in DOM order.

- Motion: honors `prefers-reduced-motion`; no auto-playing background video with audio.

- Forms: labels + descriptions + error summaries; success/failure ARIA live regions.

- Headings: single H1 per page; logical hierarchy.

- Link text is descriptive (no "click here").

- 404/500 pages offer navigation back to safety.

---

/docs/security/secrets.md

# Secrets & Privacy

- Never commit secrets. Use environment vars or cloud secret store.

- `.env.local` for dev only; production via platform env.

- Rotate on incident; least-privilege keys per environment.

- PII defaults to OFF. Do not log form free-text or emails.

- Allowed outbound domains documented in `/docs/ops/flags.md` (news APIs etc.).

---

/.env.example

# Build

NODE\_ENV=development

NEXT\_PUBLIC\_SITE\_NAME=HISL

# Telemetry (optional)

SENTRY\_DSN=

POSTHOG\_KEY=

POSTHOG\_HOST=https://app.posthog.com

LANGFUSE\_PUBLIC\_KEY=

LANGFUSE\_SECRET\_KEY=

# Feature flags

NEXT\_PUBLIC\_FLAG\_GLOBE=false

NEXT\_PUBLIC\_FLAG\_NEWS=false

# API

API\_BASE\_URL=http://localhost:3000

---

/CODEOWNERS

# Require review for critical areas

/docs/ @owner

/src/app/(routes)/\*\* @owner

/src/styles/\*\* @owner

/scripts/\*\* @owner

---

/.github/workflows/ci.yml

name: CI

on:

push: { branches: [main] }

pull\_request:

jobs:

build-test:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v4

- uses: actions/setup-node@v4

with: { node-version: '20' }

- run: npm ci

- run: npm run lint

- run: npm run typecheck || true

- run: npm run test --if-present

- run: npm run build

/.github/workflows/lighthouse.yml

name: Lighthouse

on: [push]

jobs:

lhci:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v4

- uses: actions/setup-node@v4

with: { node-version: '20' }

- run: npm ci

- run: npm run start & npx wait-on http://localhost:3000

- run: npx @lhci/cli autorun --config=lighthouserc.json

/.github/workflows/schema-check.yml

name: Schema Check

on: [push, pull\_request]

jobs:

validate:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v4

- uses: actions/setup-node@v4

with: { node-version: '20' }

- run: npm ci

- name: Validate component registry & forms

run: |

node -e "const a=require('./docs/component-registry.json'); console.log('component-registry ok')"

node -e "for (const f of ['contact','rfp','newsletter']) { require(`./docs/forms/${f}.json`); console.log(f,'ok') }"

/.github/workflows/k6.yml

name: Load Test (k6)

on:

workflow\_dispatch:

jobs:

k6:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v4

- uses: grafana/k6-action@v0.3.1

with:

filename: tests/load/k6-smoke.js

env:

BASE\_URL: https://hisl.ai

---

/docs/ops/flags.md

# Feature Flags

## Flags

- `NEXT\_PUBLIC\_FLAG\_GLOBE` (bool): enable R3F globe on homepage.

- `NEXT\_PUBLIC\_FLAG\_NEWS` (bool): enable AI news widget.

## Usage

Read via `process.env.NEXT\_PUBLIC\_FLAG\_\*` at build time; ensure safe defaults off.

Emit `feature\_toggle\_seen` event when a flag-driven component renders.

---

/config/redirects.ts (optional helper for Next.js)

import { readFileSync } from "fs";

import { join } from "path";

export function loadRedirects() {

const csv = readFileSync(join(process.cwd(), "docs/routes/redirects.csv"), "utf8");

const lines = csv.trim().split(/\r?\n/).slice(1);

return lines.map((l) => {

const [from,to,type] = l.split(",").map(s => s.trim());

return { source: from === "/ " ? "/" : from, destination: to, permanent: type === "301" };

});

}