**📁 SizeWise Suite – Expert-Sorted Project Folder Layout**

This document outlines the complete folder structure for the SizeWise Suite HVAC application, organized in the most maintainable and scalable format as recommended by expert developers. It includes both active and future-proof directories, each with clear explanations.

**✅ Root Structure**

/app/ # Core application folder (all logic lives here)

├── assets/ # Raw assets (fonts, SVGs, videos, etc.)

├── config/ # Static configurations (defaults, freemium settings)

├── core/ # Shared logic, validation engines, registrars, schemas

├── data/ # Optional: seed data, backups, sample jobs

├── docs/ # Tool-specific developer documentation

├── hooks/ # Reusable UI logic handlers (for state/event control)

├── locales/ # Translation files (en.json, ph.json, etc.)

├── plugins/ # Optional third-party or in-house add-on tools

├── services/ # Backend/frontend service logic (e.g., API integration)

├── simulations/ # Canvas-based simulation engine (Phase 2+)

├── static/ # JS logic, images, CSS themes

├── templates/ # Jinja2 HTML templates per screen/tool

├── tests/ # Global & tool-specific test suites (unit + Cypress)

└── tools/ # Self-contained modules per HVAC tool (e.g. duct-sizer)

**📦 Folder-by-Folder Description**

**/tools/**

Each tool lives in its own folder with:

* logic.js – Calculation engine
* ui.js – UI interaction logic
* events.js – Event binding (input/output)
* validators/ – Rule validators
* rules/ – JSON rule definitions (e.g., velocity limits)
* tests/ – Tool-specific test cases

**/core/**

Shared logic and core engines:

* validatorEngine.js – Core validation orchestrator
* schemaValidator.js – JSON schema validation runner
* toolRegistrar.js – Auto-registration logic
* sharedCalculations.js, pressureLossFormulas.js, etc.

**/config/**

Static configuration JSONs:

* defaultSettings.json
* freemiumPlan.json
* toolConfig.json

**/static/**

* /css/ – Tailwind-style main and themed CSS
* /js/ – Client-side logic per tool (mirrors /tools/)
* /images/ – Icons, UI assets

**/templates/**

Jinja2-based UI layout templates:

* base.html – Universal layout base
* home.html, duct-sizer.html, etc.

**/tests/**

* /unit/ – JS logic unit tests
* /integration/ – End-to-end Cypress tests

**/services/**

Reusable service logic:

* Auth handlers, future API fetchers, or data transformers

**/hooks/**

Reusable interaction logic (future use):

* State sync, theme togglers, or auto-formatting inputs

**/assets/**

* Fonts, raw SVGs, and videos separated from /static/

**/locales/**

Language packs (future multilingual support)

* en.json, ph.json, es.json, etc.

**/admin/**

Internal-facing tools (future):

* JSON rule editors
* Audit dashboard
* User or role controls

**/simulations/**

Interactive airflow visualizations (planned for Phase 2):

* Canvas draw logic
* Flow animations
* Duct connection models

**/docs/**

Markdown or HTML developer documentation per tool:

* duct-sizer.md
* validator-guide.md
* Contribution notes for devs

**/plugins/**

Experimental or optional plug-ins:

* Drop-in validators or external calculators

**✅ Expert Notes:**

* All folders are modular and replaceable.
* Shared logic lives in /core/, while tool-specific logic stays in /tools/.
* Folder naming is kebab-case for clarity and consistency.
* Test folders are colocated near their tool/module for easier debugging.
* Asset organization separates raw sources (/assets/) from production UI content (/static/).

This folder structure ensures maintainability, future extensibility, and a clean development experience even as the SizeWise Suite expands.