Project Structure — Listmgr (SQM Platform — User + Admin Interface)

1. Overview

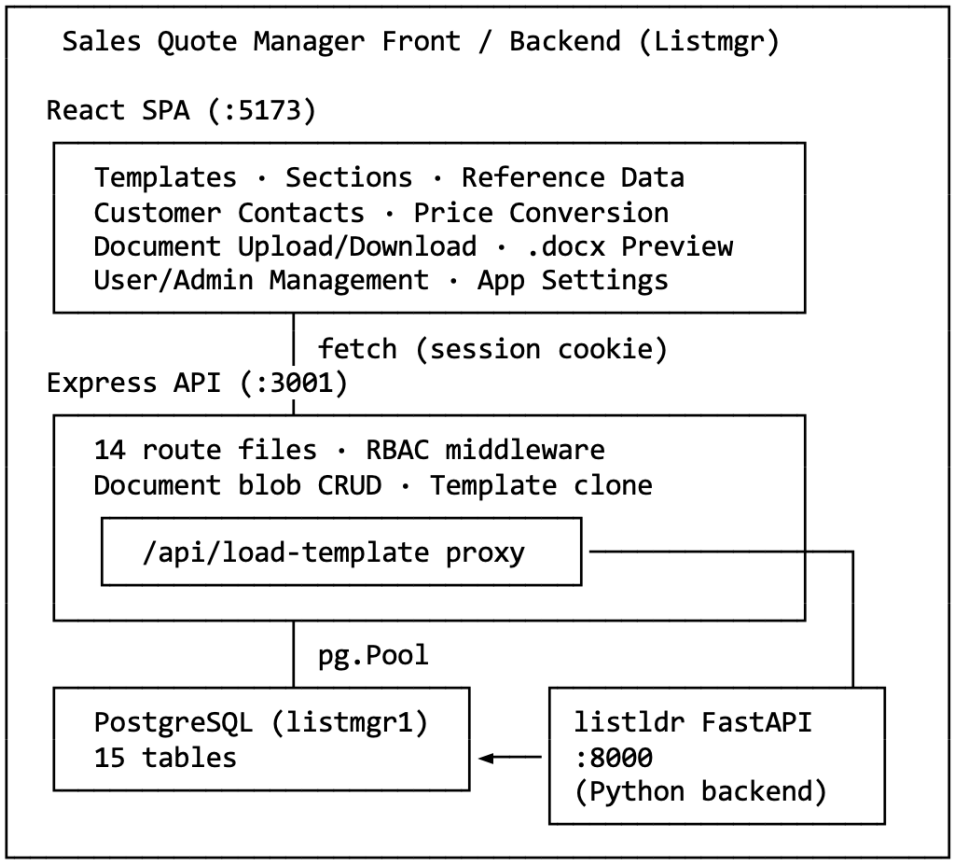
Listmgr is the primary user-facing UI/UX of the **Sales Quote Management (SQM) platform**. It provides a React SPA with an Express API backend for managing sales quote templates, reference data, customer contacts, and document workflows.

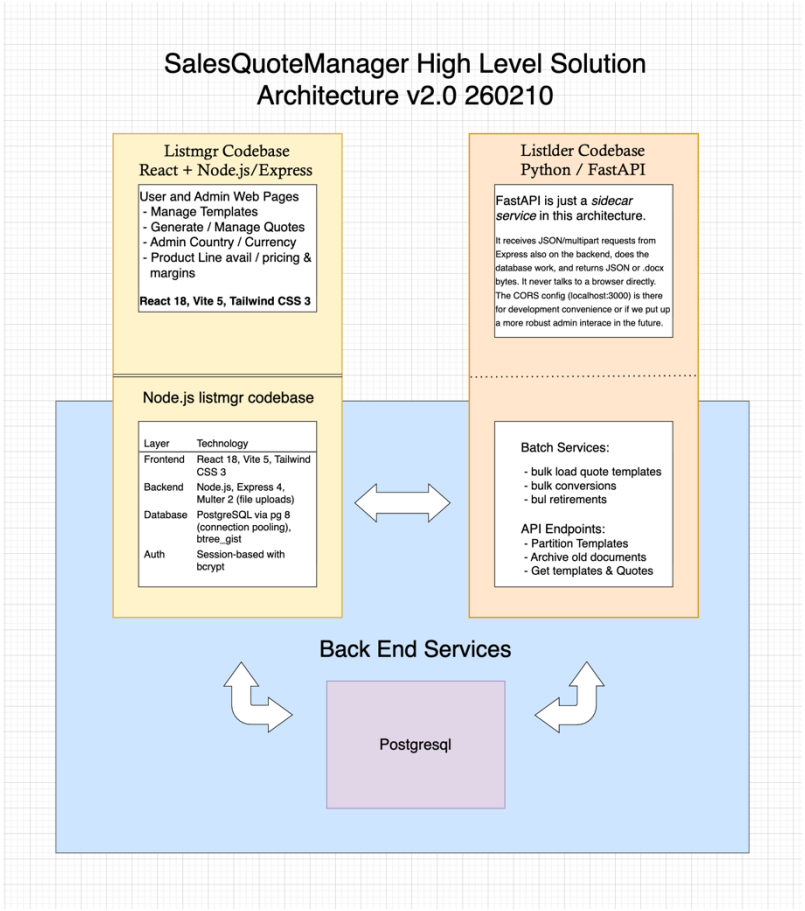
This project was **entirely generated by Claude Code and autoForge** (formerly Autocoder) — an autonomous AI coding test harness that tracks features in a SQLite database and drives implementation through iterative generation and verification cycles.

Listmgr shares the PostgreSQL database (`listmgr1`) with the companion **listldr** project (Python/FastAPI). For template loading and section extraction, the Express backend proxies requests to listldr's FastAPI service, ensuring a single codebase handles all document parsing, validation, and ingestion.

2. System Architecture

(Note Combination of React / Express API and FastAPI for shared backend capabilities)





3. Tech Stack

Layer	Technology	Version
Frontend	React	18.2
Build tool	Vite	5.0
Styling	Tailwind CSS	3.3
Routing	react-router-dom	6.20
Docx preview	docx-preview	0.3.7
Backend	Node.js / Express	4.18
Auth	express-session + bcryptjs	—
File uploads	multer	2.0
Database	PostgreSQL (pg)	8.11
Dev DB	SQLite (better-sqlite3)	—

---

4. Key Design Patterns

- **Enable/disable pattern**: Separate from `active` field; used for soft visibility control. Non-admin users never see disabled records. Cascading: disabled product categories hide their product lines.
- **Status workflow**: Templates progress through `not started → in process → in review → approved → cloned`. Status changes can cascade to all child sections.
- **Template cloning**: Deep copy (template + all sections) with status set to 'cloned' and ' (Clone)' appended to name.
- **Document deduplication**: SHA-256 hash uniqueness in `document_blob` prevents duplicate storage of identical files. Version history tracked in `document_blob_history`.
- **Proxy to listldr**: Template loading and section extraction are delegated to the listldr FastAPI service via `loadTemplate.js`, keeping document parsing logic in a single Python codebase.
- **Session auth**: 24-hour express-session with bcrypt password hashing. `requireAuth` and `requireAdmin` middleware guards.
- **FK constraint handling**: All delete operations catch PostgreSQL error 23503 (foreign key violation) and return user-friendly messages.
- **URL-persisted filters**: Template list filters are stored in URL search params for bookmarkability.

5. Running Locally

```
# Backend (port 3001)
cd backend && npm install && npm run dev

# Frontend (port 5173, Vite proxies API calls to backend)
cd frontend && npm install && npm run dev

# Template loader service (port 8000, required for Load Quote Template page)
cd ../1_listldr && uvicorn api.app:app -reload
```

6. Directory Layout

```
1_listmgr/
├── backend/                          # Express API server
│   ├── index.js                      # Entry point: middleware, CORS, session (24h),
│   │                                 #   requireAuth & requireAdmin guards
│   ├── package.json
│   ├── .env                          # DB_HOST, DB_PORT, DB_NAME, DB_USER, DB_PASSWORD
│   │
│   ├── db/
│   │   ├── index.js                 # pg.Pool wrapper, query helper with error logging
│   │   ├── schema.sql               # Full PostgreSQL schema (15 tables, incl. migrations
│   │   │                            #   105, 111, 114, 115, 116)
│   │   ├── seed.js                  # Seed data (users, reference tables)
│   │   ├── setup-postgresql.js      # PostgreSQL setup script
│   │   ├── setup.js                 # SQLite setup script (dev)
│   │   ├── migrate-105.js           # Column renames + VARCHAR→TEXT
│   │   └── migrate-111.js           # Enable/disable columns
│   │
│   ├── migrations/
│   │   ├── 114-database-changes.js      # TIMESTAMPTZ conversion, document_blob,
│   │   │                                #   document_blob_history, customer_quotes
│   │   ├── 115-price-conversion-tables.js  # btree_gist, price conversion tables
│   │   ├── 116-customer-contact.js      # Customer contact table
│   │   ├── add-enabled-columns.js
│   │   ├── add-template-section-enabled.js
```

```
│   │   ├── verify-114.js              # Post-migration verification
│   │   └── verify-115.js
│   │
│   ├── routes/                        # 14 route modules
│   │   ├── auth.js                   # POST login/logout, GET /me
│   │   ├── users.js                  # CRUD (admin only)
│   │   ├── templates.js              # CRUD, filter/search, clone, status cascade
│   │   ├── sections.js               # CRUD, clone, batch resequence
│   │   ├── sectionTypes.js           # CRUD for section type definitions
│   │   ├── documentBlobs.js          # Upload/download/remove .docx (SHA-256 dedup)
│   │   ├── loadTemplate.js           # Proxy to listldr FastAPI (load + section extract)
│   │   ├── currencies.js             # CRUD with enable/disable
│   │   ├── countries.js              # CRUD with joined currency info
│   │   ├── productCategories.js      # CRUD with cascading enable/disable
│   │   ├── productLines.js           # CRUD filtered by parent category state
│   │   ├── customerContacts.js       # CRUD for customer address book
│   │   ├── priceConversion.js        # Factors, pairs, values (admin only)
│   │   └── appSettings.js            # Theme color, version, client name
│   │
│   └── templates_docx/               # Uploaded test documents
│
├── frontend/                          # React SPA
│   ├── index.html
│   ├── package.json
│   ├── vite.config.js                # Dev server proxy → backend :3001
│   ├── tailwind.config.js
│   ├── postcss.config.js
│   │
│   └── src/
│       ├── main.jsx                  # ReactDOM.createRoot entry
│       ├── App.jsx                   # Router: PrivateRoute, AdminRoute guards
│       ├── index.css                 # Tailwind imports + global styles
│       │
│       ├── components/
│       │   └── Layout.jsx            # App shell: collapsible sidebar nav (4 menu
│       │                             #   groups), header (app name, client, user badge),
│       │                             #   footer (version info), mobile hamburger
│       │
```

```
│       ├── context/
│       │   ├── AuthContext.jsx     # Auth state, login/logout, isAdmin, session restore
│       │   └── SettingsContext.jsx  # App-wide settings from /api/app-settings
│       │
│       ├── pages/
│       │   ├── Login.jsx            # Login form with client branding
│       │   ├── Templates.jsx        # Template list: filter (country, product cat/line,
│       │   │                        #   active, enabled, search), sort, URL-persisted
│       │   │                        #   filters, status badges, clone/delete actions
│       │   ├── TemplateDetail.jsx   # Template view: status change with cascade,
│       │   │                        #   DocumentSection (upload/download/remove),
│       │   │                        #   collapsible section list, tabbed inline section
│       │   │                        #   editor, DocxViewerModal (in-browser .docx
│       │   │                        #   preview), ResequenceSectionsModal, SectionFormModal
│       │   ├── TemplateForm.jsx     # Create/edit template with reference dropdowns
│       │   ├── LoadQuoteTemplate.jsx # Upload .docx via listldr service: health check,
│       │   │                        #   availability banner, country/currency/product
│       │   │                        #   line form, proxied multipart upload
│       │   ├── Currencies.jsx       # Currency reference CRUD
│       │   ├── Countries.jsx        # Country reference CRUD
│       │   ├── ProductCategories.jsx # Product category CRUD
│       │   ├── ProductLines.jsx     # Product line CRUD (filtered by category)
│       │   ├── SectionTypes.jsx     # Section type management (admin)
│       │   ├── CustomerContacts.jsx  # Customer address book CRUD
│       │   ├── Users.jsx            # User management (admin)
│       │   ├── Settings.jsx         # App settings: theme color (22 Tailwind options),
│       │   │                        #   version, client name (admin)
│       │   ├── PriceConversion.jsx  # 3-tab UI: Factors, Pairs, Values (admin)
│       │   └── NotFound.jsx         # 404 page
│       │
│       ├── services/
│       │   └── api.js              # Fetch wrapper: get/post/put/upload/delete,
│       │                           #   credentials:'include', error normalization
│       │
│       └── utils/
│           └── colorPalette.js     # 22 Tailwind color constants for theme
│
├── docs/                           # Project documentation
```

```
│   ├── PROJECT_STRUCTURE.md
│   ├── document_blob_versioning_guide.md
│   └── add original file names to psql_template records.txt
│
├── png_save/                       # Tracked screenshots (*.png exempt from .gitignore)
│
├── CLAUDE.md                       # AI assistant project instructions
├── app_spec.txt                    # Original 103-feature application specification
├── features.db                     # SQLite — autoForge feature tracking
├── assistant.db                    # SQLite — AI conversation history
├── .gitignore
└── README.md
```

---

7. Navigation & UI Structure

The sidebar organizes all functionality into four collapsible menu groups:

Menu Group	Pages	Access
Templates	Manage Quote Templates, Load Quote Template, *Generate Quote Templates* (planned), *Export Quote Templates* (planned)	All users
Customer Quotes	*Manage Customer Quotes* (planned), *Generate Customer Quotes* (planned), *Export Customer Quotes* (planned), Manage Customers	All users
Reference Data	Currencies, Countries, Product Categories, Product Lines	All users
Admin	Section Types, Price Conversion, User Management, App Settings	Admin only

Italicized items are present in the navigation but disabled/grayed (future features).

A. Frontend Routes

Path	Component	Guard
`/login`	Login	Public
`/templates`	Templates	Auth
`/templates/new`	TemplateForm (create)	Auth
`/templates/:id`	TemplateDetail	Auth
`/templates/:id/edit`	TemplateForm (edit)	Auth
`/templates/load`	LoadQuoteTemplate	Auth
`/reference/currencies`	Currencies	Auth
`/reference/countries`	Countries	Auth
`/reference/product-categories`	ProductCategories	Auth
`/reference/product-lines`	ProductLines	Auth
`/customers/contacts`	CustomerContacts	Auth
`/admin/users`	Users	Admin
`/admin/settings`	Settings	Admin
`/admin/section-types`	SectionTypes	Admin
`/admin/price-conversion`	PriceConversion	Admin

B. Backend API Routes

1. Authentication

Method	Endpoint	Description
POST	`/api/auth/login`	Username/password login, bcrypt verify, checks user_enabled
POST	`/api/auth/logout`	Destroy session
GET	`/api/auth/me`	Return current user from session

Method	Endpoint	Description
GET	`/api/public-settings`	Client name, app/db version (unauthenticated, for login page)
GET	`/api/health`	Database connectivity check

2. Templates & Sections

Method	Endpoint	Description
GET	`/api/templates`	List/filter templates (country, product cat/line, active, enabled, search, sort)
POST	`/api/templates`	Create template
GET	`/api/templates/:id`	Get template detail with document info (has_document, filename, size)
PUT	`/api/templates/:id`	Update template
PUT	`/api/templates/:id/status`	Change status with optional cascade to all sections
DELETE	`/api/templates/:id`	Delete template and sections (admin)
POST	`/api/templates/:id/clone`	Deep clone: template + all sections, status → 'cloned'
GET	`/api/sections/template/:id`	List sections for template (joined with section type)
GET	`/api/sections/:id`	Get single section
POST	`/api/sections/template/:id`	Create section (auto-assigns sequence if omitted)
PUT	`/api/sections/:id`	Update section
POST	`/api/sections/:id/clone`	Clone section within same template
DELETE	`/api/sections/:id`	Delete section (admin)
PUT	`/api/sections/template/:id/resequence`	Batch update sequence numbers

3. Document Management

Method	Endpoint	Description
POST	`/api/templates/:id/document`	Upload .docx (max 4MB, SHA-256 dedup, archives old blob)
GET	`/api/templates/:id/document`	Download current document as attachment
DELETE	`/api/templates/:id/document`	Remove document (admin, archives to history)
GET	`/api/templates/:id/document/info`	Document metadata without bytes
GET	`/api/templates/:id/document/history`	Blob version history for template

4. Template Loading (proxied to listldr FastAPI)

Method	Endpoint	Description
GET	`/api/load-template/health`	Check listldr service availability
POST	`/api/load-template`	Proxy multipart upload to FastAPI `/api/v1/templates/load` (60s timeout)
GET	`/api/load-template/:id/sections/:seqn/docx`	Proxy section extraction to FastAPI (30s timeout)

5. Reference Data (CRUD pattern: GET list, GET/:id, POST, PUT/:id, DELETE/:id)

Resource	Base Endpoint	Notes
Currencies	`/api/currencies`	Duplicate symbol check (case-insensitive), enable/disable
Countries	`/api/countries`	Joined with currency info, enable/disable
Product Categories	`/api/product-categories`	Non-admin filtered by enabled, cascading to product lines
Product Lines	`/api/product-lines`	Filtered by parent category enabled state
Section Types	`/api/section-types`	All users see all types (no enable filter)
Customer Contacts	`/api/customer-contacts`	Address book (name, company, phone, email, address)

6. Admin-Only Routes

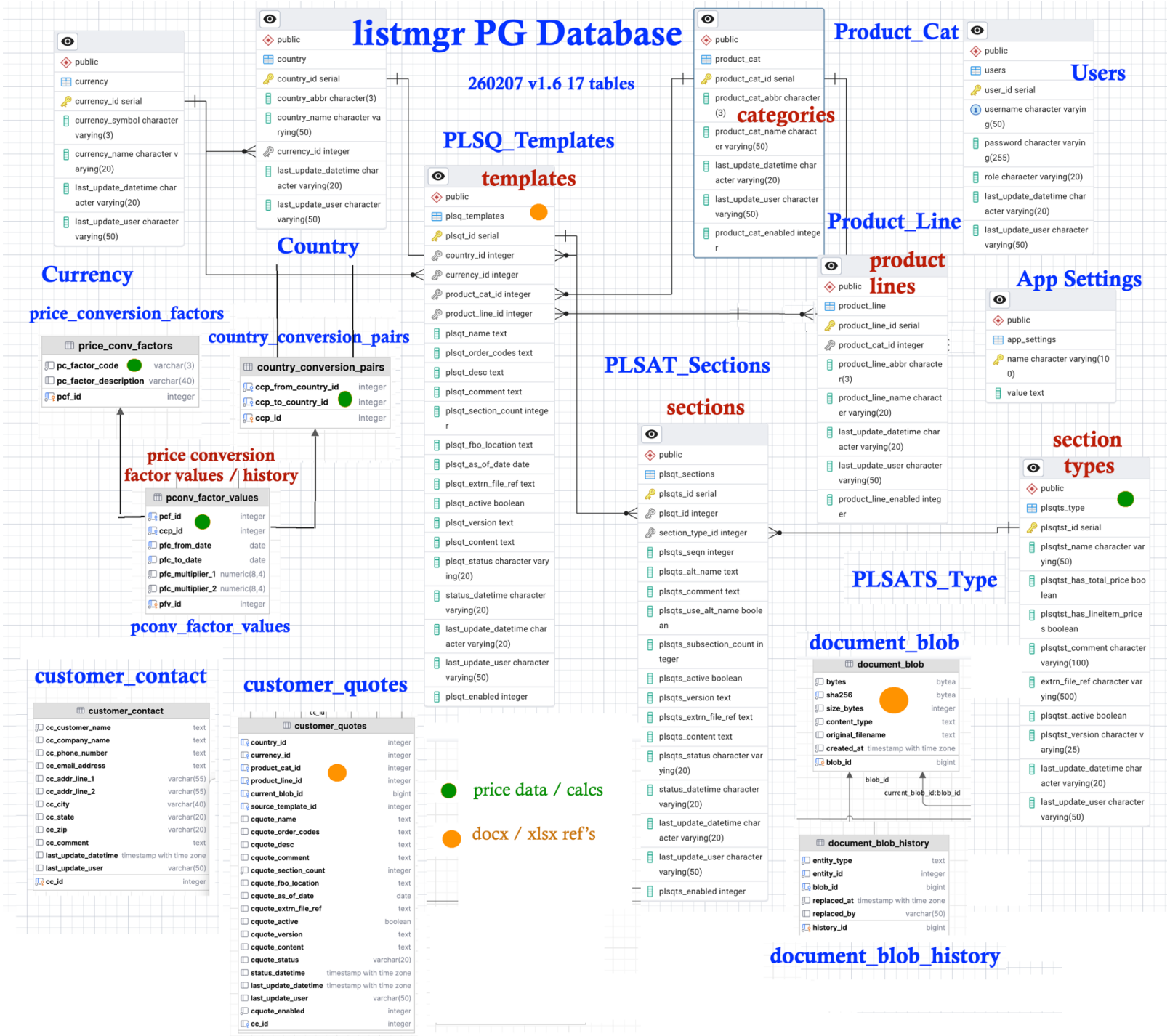
Resource	Base Endpoint	Notes
Users	`/api/users`	CRUD, bcrypt password hashing, enable/disable
App Settings	`/api/app-settings`	Theme color (22 options), version, client name. GET /colors for palette.
Price Conversion	`/api/price-conversion/*`	Factors (3-char codes), country pairs (directional), values (date-bounded, overlap-prevented)

8. Roles & Permissions

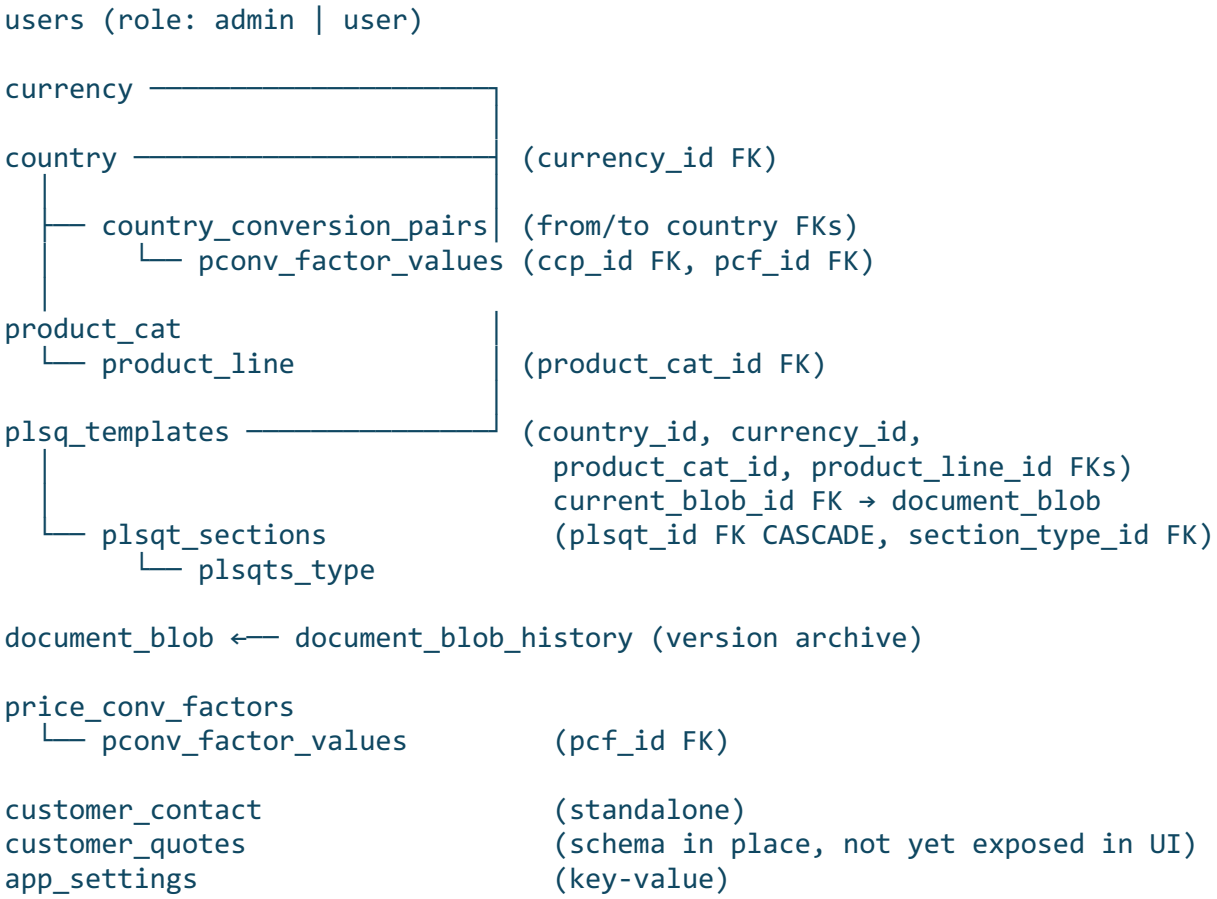
Action	Admin	User
Create / Read / Update	Yes	Yes
Delete records	Yes	No
User management	Yes	No
App settings	Yes	No
Price conversion management	Yes	No
Section type management	Yes	No
See disabled records	Yes	No
Edit disabled templates	Yes	No

Non-admin users can edit templates only when `plsqt_enabled = true` and status is ‘cloned’ or ‘not started’.

9. Database — `listmgr1`


listmgr PG Database
260207 v1.6 17 tables
public
currency
currency_id serial
currency_symbol character varying(3)
currency_name character varying(20)
last_update_datetime character varying(20)
last_update_user character varying(50)
Currency
public
country
country_id serial
country_abbr character(3)
country_name character varying(50)
currency_id integer
last_update_datetime character varying(20)
last_update_user character varying(50)
Country
price_conversion_factors
price_conv_factors
pc_factor_code varchar(3)
pc_factor_description varchar(40)
pcf_id integer
country_conversion_pairs
country_conversion_pairs
ccp_from_country_id integer
ccp_to_country_id integer
ccp_id integer
price conversion factor values / history
pconv_factor_values
pcf_id integer
ccp_id integer
pfc_from_date date
pfc_to_date date
pfc_multiplier_1 numeric(8,4)
pfc_multiplier_2 numeric(8,4)
pfv_id integer
pconv_factor_values
PLSQ_Templates
templates
public
plsq_templates
plsqt_id serial
country_id integer
currency_id integer
product_cat_id integer
product_line_id integer
plsqt_name text
plsqt_order_codes text
plsqt_desc text
plsqt_comment text
plsqt_section_count integer
plsqt_fbo_location text
plsqt_as_of_date date
plsqt_extrn_file_ref text
plsqt_active boolean
plsqt_version text
plsqt_content text
plsqt_status character varying(20)
status_datetime character varying(20)
last_update_datetime character varying(20)
last_update_user character varying(50)
plsqt_enabled integer
Product_Cat
categories
public
product_cat
product_cat_id serial
product_cat_abbr character(3)
product_cat_name character varying(50)
last_update_datetime character varying(20)
last_update_user character varying(50)
product_cat_enabled integer
Product_Line
product lines
public
product_line
product_line_id serial
product_cat_id integer
product_line_abbr character(3)
product_line_name character varying(20)
last_update_datetime character varying(20)
last_update_user character varying(50)
product_line_enabled integer
Users
public
users
user_id serial
username character varying(50)
password character varying(255)
role character varying(20)
last_update_datetime character varying(20)
last_update_user character varying(50)
App Settings
public
app_settings
name character varying(100)
value text
PLSAT_Sections
sections
public
plsqt_sections
plsqts_id serial
plsqt_id integer
section_type_id integer
plsqts_seqn integer
plsqts_alt_name text
plsqts_comment text
plsqts_use_alt_name boolean
plsqts_subsection_count integer
plsqts_active boolean
plsqts_version text
plsqts_extrn_file_ref text
plsqts_content text
plsqts_status character varying(20)
status_datetime character varying(20)
last_update_datetime character varying(20)
last_update_user character varying(50)
plsqts_enabled integer
PLSATS_Type
section types
public
plsqts_type
plsqtst_id serial
plsqtst_name character varying(50)
plsqtst_has_total_price boolean
plsqtst_has_lineitem_prices boolean
plsqtst_comment character varying(100)
extrn_file_ref character varying(500)
plsqtst_active boolean
plsqtst_version character varying(25)
last_update_datetime character varying(20)
last_update_user character varying(50)
document_blob
document_blob
bytes bytea
sha256 bytea
size_bytes integer
content_type text
original_filename text
created_at timestamp with time zone
blob_id bigint
blob_id
current_blob_id:blob_id
document_blob_history
entity_type text
entity_id integer
blob_id bigint
replaced_at timestamp with time zone
replaced_by varchar(50)
history_id bigint
document_blob_history
customer_contact
customer_contact
cc_customer_name text
cc_company_name text
cc_phone_number text
cc_email_address text
cc_addr_line_1 varchar(55)
cc_addr_line_2 varchar(55)
cc_city varchar(40)
cc_state varchar(20)
cc_zip varchar(20)
cc_comment text
last_update_datetime timestamp with time zone
last_update_user varchar(50)
cc_id integer
customer_quotes
customer_quotes
country_id integer
currency_id integer
product_cat_id integer
product_line_id integer
current_blob_id bigint
source_template_id integer
cquote_name text
cquote_order_codes text
cquote_desc text
cquote_comment text
cquote_section_count integer
cquote_fbo_location text
cquote_as_of_date date
cquote_extrn_file_ref text
cquote_active boolean
cquote_version text
cquote_content text
cquote_status varchar(20)
status_datetime timestamp with time zone
last_update_datetime timestamp with time zone
last_update_user varchar(50)
cquote_enabled integer
cc_id integer
price data / calcs
docx / xlsx ref's

C. Key Entity Relationships



10. Migration History

Migration	Changes
105	Column renames for consistency, VARCHAR → TEXT for content fields
111	Added `*_enabled` columns (INTEGER DEFAULT 1) to currency, country, users
114	VARCHAR datetime → TIMESTAMPTZ, document_blob with SHA-256 dedup, document_blob_history, customer_quotes table
115	btree_gist extension, price_conv_factors, country_conversion_pairs (UNIQUE), pconv_factor_values (EXCLUDE overlapping dates)
116	customer_contact table

11. Development Notes

This project was **entirely generated by Claude Code and autoForge** (formerly Autocoder). The `features.db` SQLite database tracks 103+ features from the original `app_spec.txt` specification. The `assistant.db` database stores AI conversation history from the generation process. The `CLAUDE.md` file provides project context and instructions for AI assistants working on the codebase.