

QuickQuote MVP (Monorepo Architecture) – Product Requirements Document (PRD)

Executive Summary

QuickQuote is the first module in the StackQuotes Monorepo, a SaaS platform for contractors to quote, propose, and get paid within one ecosystem. The MVP allows contractors to build, send, and track professional single-price estimates in minutes. It’s built as a standalone but extensible module inside the shared StackQuotes app architecture, designed for seamless integration with future modules like PayLink and ChangeFlow.

Stack: Vue 3 + TypeScript + TailwindCSS, Supabase, Vercel (Turborepo), Stripe Billing, Resend API.

Objectives

Primary Goals	Enable users to create, edit, and send estimates. Generate branded PDFs. Store client data.
Success Metrics	≤10 min from login to sent quote; ≥60% weekly returning users; 0 migration debt when adding

Monorepo Structure

stackquotes/ ■■■■ apps/ ■ ■■■■ web/ ■ ■ ■■■■ src/modules/quickquote/ ■ ■ ■■■■
src/modules/paylink/ ■ ■ ■■■■ src/modules/changeflow/ ■ ■ ■■■■ components/ ■ ■ ■■■■ pages/ ■
■■■■ api/ ■ ■■■■ estimates/ ■ ■■■■ proposals/ ■ ■■■■ payments/ ■■■■ packages/ ■ ■■■■ db/ ■
■■■■ ui/ ■ ■■■■ config/ ■ ■■■■ types/

Core Functional Requirements

Authentication	Supabase Auth for email/password (OAuth optional).
Estimate Builder	Create/Edit, Auto Totals, Save Drafts, Duplicate, Export PDF, Send via Email, Dashboard.
Constraints	Single total price only, 10 estimates per user for early phase.
PDF Export	Branded layout with logo, company info, client details, totals, and footer.
Client Database	Stores client contact data linked by user_id.
Dashboard	List/filter estimates by Draft/Sent/Accepted.

Frontend Components

Component	Purpose
-----------	---------

EstimateForm.vue	Main builder interface.
LineItemRow.vue	Dynamic line item management.
TotalsBar.vue	Displays subtotal, tax, total.
EstimateList.vue	Dashboard for all estimates.
PDFPreviewModal.vue	Preview before export.
ClientSelect.vue	Select or add client.
SettingsForm.vue	Company logo and footer preferences.

Backend Endpoints

/api/estimates/create	POST	Create new estimate.
/api/estimates/update	PATCH	Update existing estimate.
/api/estimates/list	GET	List estimates.
/api/pdf/generate	POST	Generate and upload PDF to Supabase.
/api/email/send	POST	Send estimate via Resend API.

Database Schema (Supabase)

Tables: - users (id, email, company_name, logo_url, created_at) - clients (id, user_id, name, email, phone, address) - estimates (id, user_id, client_id, project_title, line_items, subtotal, tax, total, notes, status, converted_to_proposal, created_at) - user_settings (user_id, default_tax_rate, footer_text, logo_url)

Extensibility Preparation

Future Module	Dependency	Preparation
PayLink	estimate_id → proposals	Add converted_to_proposal field.
ChangeFlow	job_id	Add nullable job_id.
ProfitPulse	line-item cost tracking	Add nullable cost field.
PermitPath	address	Keep normalized address schema.
TeamStack	org_id	Add org_id to users table.

Shared Packages

/packages/db	Supabase client, schema migrations.
/packages/ui	Shared components.
/packages/config	Environment loaders, constants.
/packages/types	Shared TS interfaces.

Non-Functional Requirements

Performance: <2s dashboard load, <5s PDF generation. Security: RLS enabled. UX: Mobile-first. Scalability: modular architecture. Analytics: track estimates_created, pdf_downloads, emails_sent.

Deliverables

- QuickQuote module built under /apps/web/src/modules/quickquote/.
- Supabase schema preloaded from /packages/db/schema.sql.
- PDF + email export.
- Deployed on Vercel with working auth.
- Ready for PayLink integration.

Success Criteria

10 Design Partners by Dec 2024. 40 Founders by Jan 2025. MVP stable and extensible for PayLink by Q2 2025.

Codex Build Command

Run inside monorepo: `codex build --from ./docs/PRDs/QuickQuote_Monorepo_MVP_PRD.pdf --path ./apps/web/src/modules/quickquote`