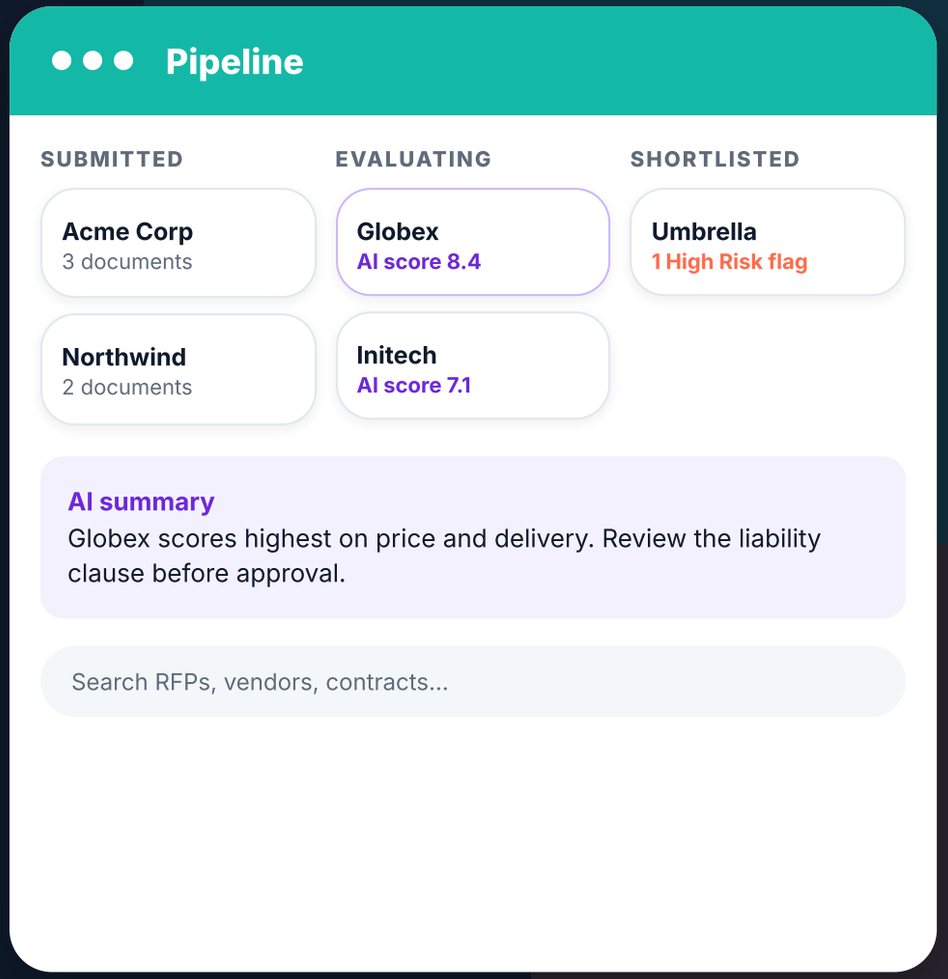
PORTFOLIO CASE STUDY | FULL-STACK WEB APPLICATION

ProcureMaster: AI Procurement Intelligence Platform

A web app that reads vendor proposals, scores them, flags risky clauses, and guides each deal through approval to a signed contract.

Next.js 14 | TypeScript | Supabase | Gemini AI | Tailwind | Vercel



AI-powered software development | Built in phased Claude Code sessions from a six-document spec

01 Project Overview

What the platform is, what it is for, and who it helps

PROJECT NAME

ProcureMaster: AI Procurement Platform

A full-stack web app for small and mid-size teams. It collects vendor proposals in one place, uses AI to pull out key terms, score each vendor, and flag risky clauses, then moves the winner through approval to a tracked contract.

MAIN OBJECTIVE

Cut the time from request for quote to signed contract, and make every vendor decision easy to review.

WHO IT HELPS

- **Small and mid-size companies**
  Teams of 10 to 200 people buying from vendors
- **Procurement managers**
  Run the full vendor pipeline
- **Department heads**
  Raise needs and review vendor choices
- **Finance approvers**
  Approve with the full report in hand
- **Vendors**
  Submit proposals through a secure portal

4 user roles with their own access

18 database tables with row-level security

39 API routes for the app and AI steps

12 build phases, each tested

02

The Business Pain

The problems teams face without procurement automation

Spreadsheet scoring

Vendors are scored in spreadsheets with no version history and no common criteria.

Proposals lost in email

Responses pile up in inboxes, so they are hard to find and compare.

Manual risk checks

People scan long contracts by eye, which is slow and easy to get wrong.

Approvals get chased

Approvers are pinged on Slack and email, with no record of who said yes.

Missed renewals

Contract end dates live in separate files, so renewals slip by.

Sound familiar?

When vendor data is scattered, deals slow down and risks slip through. One platform with AI help can take on the repeat work.

03 The Proposed Solution

An AI-assisted platform that reads documents and moves each deal forward

USER ACTION	PLATFORM DOES	RESULT
Department head submits a need	**Saves a structured requirement** `requirements form`	It shows up as a new pipeline item
Manager invites vendors	**Sends a unique link that expires in 72 hours** `/api/rfps/[id]/invite-vendor`	Vendors upload through a secure portal
Vendor uploads a proposal	**Reads PDF, Word, or Excel and pulls out key terms** `/api/process/document/[id]`	Price, SLA, and terms stored as data
Manager reviews criteria	**AI suggests 5 to 10 weighted criteria, then scores 0 to 10** `/api/process/score/[id]`	A scorecard with a reason for each score
Contracts are checked	**Flags risky clauses as High, Medium, or Low** `/api/process/compliance/[id]`	Flags shown next to the clause
Finance approves or rejects	**Logs the decision and a required comment** `/api/approvals/[id]/decide`	A contract record and renewal alerts

04 Key Features & Capabilities

What the system can actually do

Four user roles

Each role sees only its own data, enforced in the database.

Vendor portal

Invite-only uploads. Vendors see only their own status.

Drag-and-drop pipeline

Seven stages from Invited to Archived.

AI document reading

PDF, Word, and Excel files up to 50MB.

AI scoring

Weights add up to 100%. Managers can override any score.

Risk flags

Spots payment, liability, and renewal risks in contracts.

Comparison reports

Side-by-side scorecard with PDF export and share link.

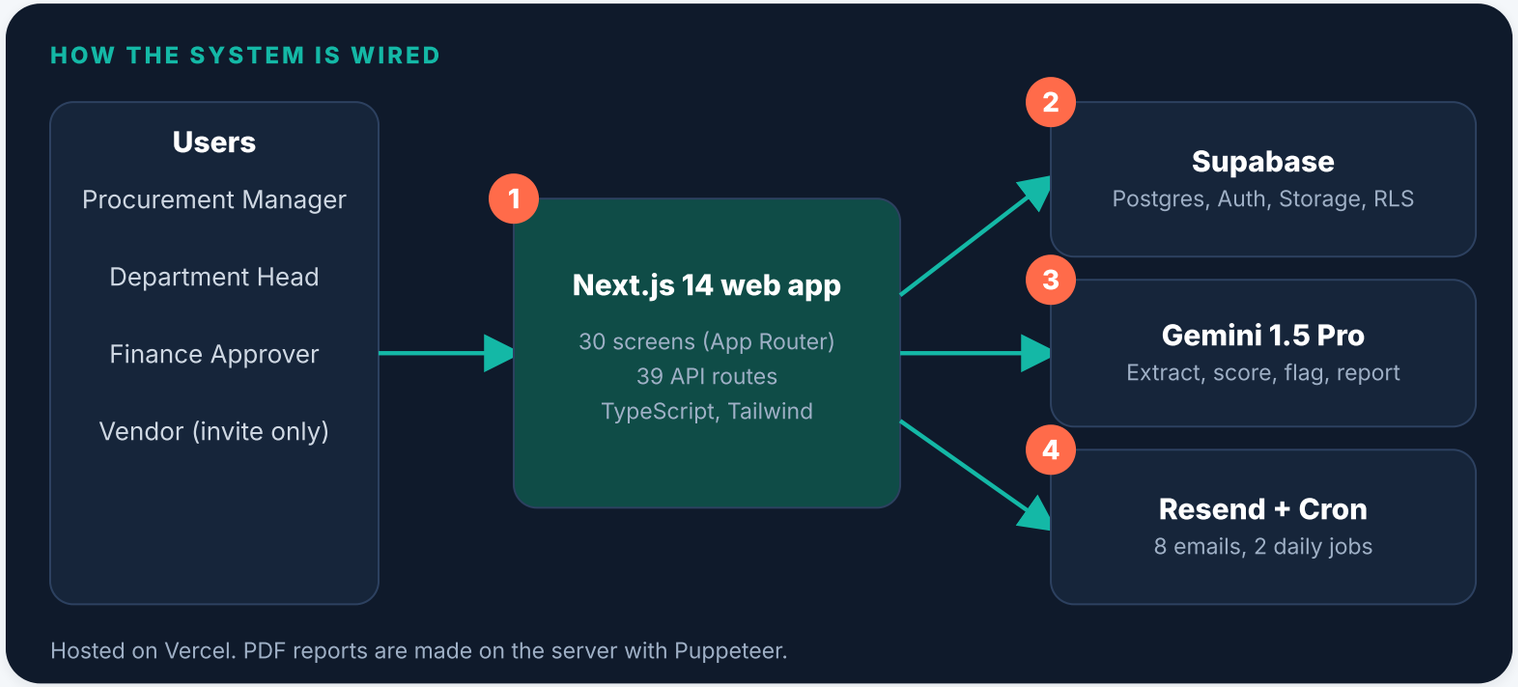
Approvals and renewals

Linear approval chain, then alerts before a contract ends.

AI suggests, people decide. Every AI result is labeled and every approval is logged.

05 Workflow Architecture

How the parts of the system connect



Architecture diagram drawn from the Tech Stack and Backend Schema documents.

1 Web app

Next.js shows each role its own screens. The same app also hosts the API routes.

2 Data and security

Supabase stores 18 tables. Row-level security keeps each company and vendor in its own lane.

3 AI pipeline

Server code pulls text from PDF, Word, and Excel files. Gemini returns structured data, scores, and risk flags.

4 Email and jobs

Resend sends 8 email types. Daily cron jobs check renewals and approval deadlines.

SIMPLE VIEW

Upload proposal → **Extract** text → **Gemini** analyzes → **Human** review → **Contract** + alerts

06 How It Works

From the first request to a signed contract

1. **Need raised**
   Department head fills out a form.
2. **Vendors invited**
   Each gets a secure link.
3. **Proposals in**
   Files upload and are read.
4. **AI evaluates**
   Scores and risk flags appear.
5. **Finance decides**
   Approve or reject with notes.
6. **Contract tracked**
   Renewal alerts go out.

Example walkthrough (illustrative)

Department head: We need a cloud storage vendor.

Manager created the RFP and invited three vendors.

AI: Read all proposals. Pricing, SLA, and termination terms extracted. One High Risk flag found in a liability clause.

Manager: Shortlist Globex and send for approval.

Finance approved with a comment. Contract created, renewal alert set for 90 days.

BEHIND THE SCENES

1. Each file moves through `Queued, Processing, Extracted`.
2. `score` and `compliance` routes call Gemini and save results.
3. Manager can override any AI score with a note.
4. Finance decision is saved with user, time, and comment.
5. A database trigger creates the contract on approval.
6. A daily 8am UTC cron job sends renewal alerts.

07 Business Benefits & Expected Outcomes

The value, the time saved, and the operational gains

Faster evaluations

AI reads and scores proposals in minutes, not days.

Consistent scoring

Every vendor is judged on the same weighted criteria.

Fewer missed risks

Risky clauses are flagged on every document.

Clear approvals

One linear path with a saved record of each decision.

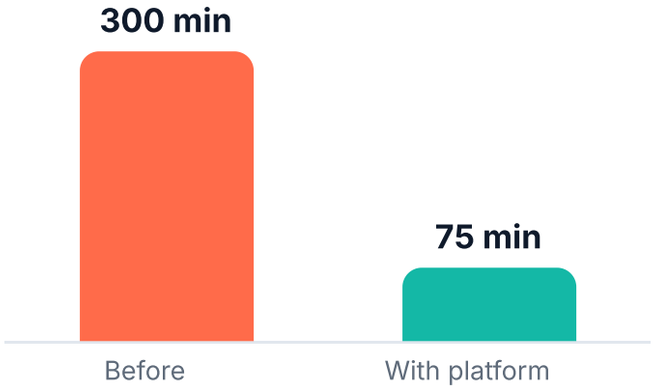
No missed renewals

Alerts go out before each contract ends.

Full visibility

One dashboard shows every RFP, vendor, and contract.

Example time savings



5 proposals x 60 minutes of first-pass review = 300 minutes. If AI extraction and scoring cut that to 15 minutes each, 75 remain. That frees about **225 minutes** (3.75 hours) per RFP.

Example only. Real results depend on proposal length and team process.

08 Technology Stack & Implementation

The tools, what you need, and how it can grow

TECHNOLOGY STACK

Next.js 14 + TypeScript	App Router web app and API routes, strict typing
Tailwind + shadcn/ui	Consistent styling and ready-made components
Supabase	Postgres database, login, file storage, row-level security
Google Gemini 1.5 Pro	Reads long contracts and returns structured JSON
Resend + React Email	Invites, approvals, and renewal alerts
Puppeteer	Branded PDF comparison reports
Vercel + Cron	Hosting and two daily background jobs
Playwright	End-to-end tests for the key flows

WHAT YOU NEED

- Node.js 20 or higher and Git
- Supabase project (database, auth, storage)
- Gemini API key
- Resend account for email
- Vercel account for hosting

CUSTOMIZATION IDEAS

- ERP links such as SAP or Oracle
- Slack or Teams notifications
- OCR for scanned documents
- Flexible, role-based approval routing
- Mobile app

How it gets built

1 Plan

Six spec docs: product, app flow, schema, design, stack, build plan.

2 Foundation

Setup, 27 database migrations, login, role-based app shell.

3 Core features

Pipeline, vendor portal, AI processing, approvals, contracts.

4 Polish and ship

Error states, Playwright tests, then Vercel deploy.

Have a web app idea to build?

Start with clear written specs, build in small phases, and test each phase before moving on. That keeps AI-assisted development fast and under control.

Write the spec

Build in phases

Test and launch

Built with Next.js, TypeScript, Supabase, Gemini, Resend, and Vercel.