TRADEFLOW OS

Implementation Supplement

Environment · API Spec · Business Rules · AI Prompts · RBAC · Uploads · Email · Seed Data

Companion to: Production Build Plan — Python/FastAPI + React

Version 1.0 · February 2026

SECTIONS IN THIS DOCUMENT

SECTION 1 — Environment Configuration

Complete .env.example covering all services. Generate JWT_SECRET_KEY with: `python -c "import secrets; print(secrets.token_hex(32))"`

Variable Name	Example Value	Used By	Required
Database			
DATABASE_URL	postgresql+asyncpg://user:pass@localhost:5432/tradeflow	Backend	YES
DATABASE_POOL_SIZE	10	Backend	YES
DATABASE_MAX_OVERFLOW	5	Backend	YES
DATABASE_POOL_RECYCLE	3600	Backend	YES
Redis & Celery			
REDIS_URL	redis://localhost:6379/0	Backend + Celery	YES
CELERY_BROKER_URL	redis://localhost:6379/0	Celery	YES
CELERY_RESULT_BACKEND	redis://localhost:6379/1	Celery	YES
Object Storage (MinIO / S3)			
MINIO_ENDPOINT	localhost:9000	Backend	YES
MINIO_ACCESS_KEY	minioadmin	Backend	YES
MINIO_SECRET_KEY	minioadmin	Backend	YES
MINIO_BUCKET	documents	Backend	YES
MINIO_SECURE	false	Backend	YES
Authentication & Security			
JWT_SECRET_KEY	<256-bit random hex>	Backend	YES
JWT_ALGORITHM	HS256	Backend	YES
JWT_ACCESS_TOKEN_EXPIRE_MINUTES	1440	Backend	YES
JWT_REFRESH_TOKEN_EXPIRE_DAYS	30	Backend	YES
BCRYPT_ROUNDS	12	Backend	YES
Anthropic AI			
ANTHROPIC_API_KEY	sk-ant-api03-...	Backend	YES
ANTHROPIC_MODEL	claude-opus-4-5-20251101	Backend	YES
ANTHROPIC_FAST_MODEL	claude-haiku-4-5-20251001	Backend	YES

Variable Name	Example Value	Used By	Required
ANTHROPIC_MAX_TOKENS	4096	Backend	YES
Email (SMTP)			
SMTP_HOST	smtp.gmail.com	Backend	YES
SMTP_PORT	587	Backend	YES
SMTP_USER	notifications@company.com	Backend	YES
SMTP_PASSWORD		Backend	YES
EMAIL_FROM	TradeFlow	Backend	YES
FX Rates			
FX_API_KEY		Backend	YES
FX_API_URL	https://v6.exchangerate-api.com/v6/	Backend	YES
FX_BASE_CURRENCY	AED	Backend	YES
FX_CACHE_TTL_SECONDS	3600	Backend	NO
Application			
APP_ENV	development	Both	YES
APP_DEBUG	true	Backend	YES
CORS_ORIGINS	http://localhost:3000	Backend	YES
FRONTEND_URL	http://localhost:3000	Backend	YES
COMPANY_NAME	Your Trading Co.	Backend	YES
SENTRY_DSN	https://...@sentry.io/...	Backend	NO
Frontend (Next.js)			
NEXT_PUBLIC_API_URL	http://localhost:8000	Frontend	YES
NEXT_PUBLIC_APP_NAME	TradeFlow OS	Frontend	YES

Docker Compose Production Override

```
# docker-compose.prod.yml
services:
  api:
    environment:
      - APP_ENV=production
      - APP_DEBUG=false
      - MINIO_SECURE=true
      - CORS_ORIGINS=https://tradeflow.yourcompany.com
```

SECTION 2 — Complete API Endpoint Specification

All endpoints prefixed with /api. JWT cookie required. Pagination: ?page=1&limit=20.

M0 — Auth & Users

Method	Path	Description	Roles	Code
POST	**/auth/login**	Login → JWT cookie + refresh token	Public	200/201
POST	**/auth/refresh**	Refresh access token	Public	200/201
POST	**/auth/logout**	Clear cookies	All	200/201
GET	**/auth/me**	Current user profile	All	200/201
GET	**/users**	List users	admin	200/201
POST	**/users**	Create user	admin	200/201
PATCH	**/users/{id}**	Update user / role / password	admin	200/201
DELETE	**/users/{id}**	Deactivate user (soft delete)	admin	200/201

M1 — Deals

Method	Path	Description	Roles	Code
GET	**/deals**	List deals (?status=&customer;_id=)	All	200/201
POST	**/deals**	Create deal	sales,admin	200/201
GET	**/deals/{id}**	Deal detail (with nested entities)	All	200/201
PATCH	**/deals/{id}**	Update deal fields	sales,admin	200/201
PATCH	**/deals/{id}/status**	Transition deal status	sales,procurement,admin	200/201
DELETE	**/deals/{id}**	Soft-delete deal	admin	200/201
GET	**/deals/{id}/activities**	Activity log (paginated)	All	200/201
GET	**/deals/kpis**	Aggregate KPIs for dashboard	All	200/201

M2 — CRM & Sales

Method	Path	Description	Roles	Code
GET	**/customers**	List customers (?search=&on;_hold=)	sales,finance,admin	200/201
POST	**/customers**	Create customer	sales,admin	200/201
GET	**/customers/{id}**	Customer profile + deal history	sales,finance,admin	200/201

Method	Path	Description	Roles	Code
PATCH	/customers/{id}	Update customer	sales,admin	200/201
PATCH	/customers/{id}/hold	Toggle credit hold	finance,admin	200/201
GET	/quotes	List all quotes (?deal_id=&status;=)	sales,finance,admin	200/201
POST	/quotes	Create quote from deal	sales,admin	200/201
PATCH	/quotes/{id}	Update quote (increments version)	sales,admin	200/201
PATCH	/quotes/{id}/status	Send / accept / reject quote	sales,admin	200/201
POST	/customer-pos	Record customer PO	sales,admin	200/201
PATCH	/customer-pos/{id}/accept	Accept PO → advances deal status	sales,admin	200/201

M3 — Procurement

Method	Path	Description	Roles	Code
GET	/vendors	List vendors (?search=&min;_score=)	procurement,admin	200/201
POST	/vendors	Create vendor	procurement,admin	200/201
GET	/vendors/{id}	Vendor detail + history	procurement,admin	200/201
PATCH	/vendors/{id}	Update vendor	procurement,admin	200/201
GET	/vendors/search	Semantic vendor search (?q=)	procurement,admin	200/201
GET	/proposals	List proposals (?deal_id=&status;=)	procurement,admin	200/201
POST	/proposals	Create proposal record	procurement,admin	200/201
PATCH	/proposals/{id}	Update proposal pricing/terms	procurement,admin	200/201
PATCH	/proposals/{id}/select	Select vendor → reject others	procurement,admin	200/201
GET	/proposals/compare/{deal_id}	Side-by-side comparison	procurement,admin	200/201
POST	/vendor-pos	Create vendor PO from proposal	procurement,admin	200/201
PATCH	/vendor-pos/{id}/status	Update production status	procurement,admin	200/201

M4 — AI Engine & Documents

Method	Path	Description	Roles	Code
POST	/documents/upload	Upload document → triggers parse task	All	200/201
GET	/documents	List docs (?deal_id=&entity;_type=)	All	200/201
GET	/documents/{id}	Document metadata + parsed_data	All	200/201
GET	/documents/{id}/download	Signed URL for file download	All	200/201
POST	/ai/parse-rfq	Manually trigger RFQ re-parse	sales,admin	200/201
POST	/ai/parse-proposal	Manually trigger proposal re-parse	procurement,admin	200/201
POST	/ai/compare-po	Re-run PO vs Quote comparison	sales,admin	200/201
GET	/ai/vendor-search	Semantic vendor search endpoint	procurement,admin	200/201

M5 — Finance

Method	Path	Description	Roles	Code
GET	/payments	All payments (?direction=&status;=)	finance,admin	200/201
POST	/payments	Create manual payment record	finance,admin	200/201
PATCH	/payments/{id}/approve	Approve payment	finance,admin	200/201
PATCH	/payments/{id}/paid	Mark paid (date + ref + fx_rate)	finance,admin	200/201
GET	/invoices	List invoices (?status=&customer;_id=)	finance,sales,admin	200/201
POST	/invoices	Create invoice from deal	finance,admin	200/201
GET	/invoices/{id}/pdf	Download generated PDF	finance,admin	200/201
POST	/invoices/{id}/payment	Record partial/full payment	finance,admin	200/201
GET	/finance/aging	AR aging buckets	finance,admin	200/201
GET	/finance/pnl/{deal_id}	Deal P&L; breakdown	finance,admin	200/201

M6 — Dashboard

Method	Path	Description	Roles	Code
GET	/dashboard/kpis	KPI cards (active deals, AR, margin)	All	200/201
GET	/dashboard/pipeline	Pipeline funnel by status + values	All	200/201
GET	/dashboard/revenue	Monthly revenue chart data	finance,admin	200/201
GET	/dashboard/activity	Last 20 activity items (RBAC filtered)	All	200/201
GET	/dashboard/notifications	Unread notifications	All	200/201

SECTION 3 — Business Rules & Validation Logic

3.1 Deal Status Transitions

From	To	Condition / Note
rfq_received	**sourcing**	Auto or manual after vendors identified
sourcing	**quoted**	Quote created and sent to customer
quoted	**sourcing**	Customer requests revision — quote back to draft
quoted	**po_received**	Customer PO uploaded and accepted
po_received	**ordered**	Vendor PO placed
ordered	**in_production**	Vendor confirms production started
in_production	**shipped**	Shipping docs uploaded
shipped	**delivered**	GRN created or customer confirms delivery
delivered	**invoiced**	Final invoice sent to customer
invoiced	**paid**	Full payment received from customer
paid	**closed**	Finance closes the deal
ANY (pre-delivery)	**cancelled**	Admin only; cascades to linked POs/payments

3.2 Margin Calculation

```
# Step 1: Normalise to base currency (AED)
customer_total_aed = customer_po.total_value * fx_rate_at_po_time
vendor_total_aed   = vendor_po.total_cost    * fx_rate_at_po_time

# Step 2: Estimated margin (at quote time)
estimated_margin_pct = ((quote.total - proposal.total) / quote.total) * 100

# Step 3: Actual margin (on deal close)
total_costs = vendor_po.total_cost_aed + logistics_cost_aed + other_costs_aed
actual_margin_pct = ((customer_total_aed - total_costs) / customer_total_aed) * 100

# Thresholds
# < 5%   RED flag — manager approval required before sending quote
# 5-10%  AMBER warning shown in quote builder
# > 10%  GREEN — no action required
```

3.3 Credit Limit Rules

Rule	Detail
Credit used = ?	SUM of all unpaid invoices (AR) + value of deals in status po_received → invoiced

Rule	Detail
Check trigger	On quote creation. Warn if projected exposure > 85%, block if > 100%
Utilisation warning	Show amber banner at 85%, red block at 100%. Admin can override with written note.
Credit hold	finance role only. Blocks new quote creation. Existing deals continue unaffected.
Recalculation	Real-time: every time a payment is recorded or a new deal advances status.

3.4 Vendor Credibility Score Algorithm

```
def calculate_credibility(vendor_id: UUID) -> int:
    deals = get_completed_deals(vendor_id)          # closed/delivered only
    if len(deals) == 0: return 50                   # neutral for new vendors

    on_time     = len([d for d in deals if d.delivered_on_time]) / len(deals)
    quality     = mean([d.quality_rating for d in deals]) / 5.0   # 1-5 -> 0-1
    prices      = [d.unit_price_aed for d in deals]
    consistency = 1 - (stdev(prices)/mean(prices)) if len(prices)>1 else 0.8
    deal_score  = min(len(deals) / 50.0, 1.0)       # caps at 50 deals
    days_since  = (today - max(d.closed_at for d in deals)).days
    recency     = math.exp(-days_since / 180.0)     # 180-day half-life

    score = (on_time*0.30 + quality*0.30 +
             consistency*0.20 + deal_score*0.10 + recency*0.10) * 100
    return round(clamp(score, 0, 100))
```

3.5 PO Discrepancy Severity Matrix

Field	Threshold	Severity	Action Required
Price variance	> 2%	CRITICAL	Block PO acceptance until resolved
Price variance	0.5-2%	WARNING	Highlight; procurement manager notified
Quantity variance	> 5%	CRITICAL	Block PO acceptance
Quantity variance	1-5%	WARNING	Procurement to confirm
Delivery date	> 7d earlier	CRITICAL	Check vendor PO; notify logistics
Delivery date	> 7d later	WARNING	Update vendor PO if accepted
Payment terms	Any difference	CRITICAL	Finance approval required
Incoterms	Any difference	CRITICAL	Finance + legal must review
Currency	Different currency	CRITICAL	FX risk assessment required
Missing line item	Item absent in PO	WARNING	Confirm omission with customer

SECTION 4 — AI Document Parsing Prompts

Use **claude-opus-4-5-20251101** for parsing (accuracy critical). Use **claude-haiku-4-5-20251001** for discrepancy detection (cost optimisation). Temperature = 0 for all parsing tasks.

4.1 RFQ Parsing System Prompt

```
RFQ_SYSTEM_PROMPT = """
You are an expert extraction agent for oil & gas RFQ documents.
Industry context:
  Standards: API 5L, API 5CT, ASTM A106, ASTM A53, ASME B16, ISO 3183
  Pipe grades: X42, X52, X60, X65, X70, X80
  End finishes: PE (plain end), BE (bevelled end), T&C; (threaded & coupled)
  Materials: Carbon steel, Alloy steel, Stainless 316L/304
  Units: MT, PCS, M, FT, KG
  Incoterms 2020: EXW, FCA, FOB, CFR, CIF, CPT, CIP, DAP, DDP
  Certs: MTC, CoC, TPI, NACE, IBR, CE

Extraction rules:
  1. Normalise material specs to UPPERCASE (e.g. "API 5L X65 PSL2")
  2. Quantity = number only. Unit = separate field.
  3. Dates = ISO format YYYY-MM-DD only.
  4. If a field is absent return null — never guess.
  5. Multiple line items → array. Preserve original order.
"""
```

4.2 ParsedRFQ Pydantic Schema

```
class RFQLineItem(BaseModel):
    line_number:          int | None
    description:          str
    material_spec:        str | None   # e.g. "API 5L X65 PSL2 SMLS"
    quantity:             float | None
    unit:                 str | None   # MT / PCS / M / FT / KG
    required_delivery:    str | None   # YYYY-MM-DD
    special_requirements: str | None

class ParsedRFQ(BaseModel):
    document_type:   Literal["rfq","po","enquiry"] = "rfq"
    customer_name:   str | None
    rfq_reference:   str | None
    issue_date:      str | None
    validity_date:   str | None
    delivery_terms:  str | None
    payment_terms:   str | None
    required_certs:  list[str]
    line_items:      list[RFQLineItem]
    confidence:      float          # 0.0-1.0
```

4.3 Vendor Proposal Parsing System Prompt

```
PROPOSAL_SYSTEM_PROMPT = """
You are a pricing extraction agent for oil & gas vendor proposals.

Rules:
  1. Extract ALL price components — unit, total, freight, insurance separately.
  2. Currency: 3-letter ISO code (AED/USD/EUR/GBP/CNY/INR).
  3. Lead time in DAYS only (convert "6 weeks" -> 42, "2 months" -> 60).
  4. Payment terms: preserve exact text.
  5. Validity: YYYY-MM-DD.
  6. Flag spec deviations — do not silently accept substitutions.
"""

class ParsedProposal(BaseModel):
    vendor_name:     str | None
    quote_reference: str | None
    validity_date:   str | None
    currency:        str | None
    lead_time_days:  int | None
    payment_terms:   str | None
    incoterms:       str | None
    grand_total:     float | None
    offered_certs:   list[str]
    line_items:      list[ProposalLineItem]
    confidence:      float
```

4.4 PO vs Quote Discrepancy Prompt

```
PO_COMPARISON_PROMPT = """
Compare a customer PO against the accepted quote.
For each discrepancy return:
  field:         exact field name
  quote_value:  value in the accepted quote
  po_value:     value in the customer PO
  severity:     "critical" | "warning" | "info"
  description:  plain English explanation

Severity:
  critical = affects payment, margin >2%, or changes legal terms
  warning  = minor variance, delivery +-7 days, or missing info
  info     = reference number / formatting differences only

Return [] if no discrepancies found.
"""
```

4.5 Semantic Search — Embedding Strategy

```
# Text to embed for VENDOR (on vendor create/update)
def vendor_embed_text(vendor: Vendor) -> str:
    return f"""
    Vendor: {vendor.name}. Country: {vendor.country}.
    Products: {", ".join(vendor.product_categories)}.
    Certifications: {", ".join(vendor.certifications)}.
    """

# Text to embed for DEAL SEARCH QUERY
```

```
def deal_query_text(deal: Deal) -> str:
    items = "; ".join([f'{li["description"]} {li.get("material_spec","")}'
                       for li in deal.line_items])
    return f"Oil & gas procurement: {deal.description}. Items: {items}."

# Ranking formula
final_score = (cosine_similarity * 0.60 +
               vendor.credibility_score/100 * 0.30 +
               recency_score * 0.10)
# Only return vendors with cosine_similarity > 0.55
```

SECTION 5 — RBAC Permission Matrix

Chinese wall enforced in API response serialisation: sales endpoints strip vendor cost fields; procurement endpoints strip customer quote pricing.

Permission	admin	sales	proc.	finance	quality	logist.	whouse
DEALS							
Create deal	YES	YES	–	–	–	–	–
View all deals	YES	YES	YES	YES	YES	YES	YES
Edit deal fields	YES	YES	–	–	–	–	–
Change deal status	YES	YES	YES	–	–	–	–
Cancel deal	YES	–	–	–	–	–	–
CUSTOMERS & SALES							
Create / edit customer	YES	YES	–	–	–	–	–
View customer financials	YES	–	–	YES	–	–	–
Toggle credit hold	YES	–	–	YES	–	–	–
Create / send quote	YES	YES	–	–	–	–	–
View quote pricing	YES	YES	–	YES	–	–	–
Accept customer PO	YES	YES	–	–	–	–	–
PROCUREMENT							
Create / edit vendor	YES	–	YES	–	–	–	–
View vendor costs	YES	–	YES	YES	–	–	–
Request / record proposals	YES	–	YES	–	–	–	–
Select vendor	YES	–	YES	–	–	–	–
Create / manage vendor PO	YES	–	YES	–	–	–	–
FINANCE							
Approve payment	YES	–	–	YES	–	–	–
Mark payment paid	YES	–	–	YES	–	–	–
Create invoice	YES	–	–	YES	–	–	–
View deal P&L;	YES	–	–	YES	–	–	–
DOCUMENTS & AI							
Upload documents	YES	YES	YES	YES	YES	YES	–
View documents	YES	YES	YES	YES	YES	YES	YES
Trigger AI parse	YES	YES	YES	–	–	–	–
ADMINISTRATION							
Create / manage users	YES	–	–	–	–	–	–

Permission	admin	sales	proc.	finance	quality	logist.	whouse
View audit log (all)	YES	–	–	–	–	–	–
Override system rules	YES	–	–	–	–	–	–

FastAPI Implementation Pattern

```
# backend/app/middleware/rbac.py
PERMISSIONS = {
    "admin":       ["*"],
    "sales":       ["deal.*","customer.*","quote.*","customer_po.*","document.*"],
    "procurement": ["deal.read","vendor.*","proposal.*","vendor_po.*","document.*"],
    "finance":     ["deal.read","customer.read","payment.*","invoice.*","finance.*"],
    "quality":     ["deal.read","vendor.read","document.*"],
    "logistics":   ["deal.read","vendor_po.read","document.*"],
    "warehouse":   ["deal.read","vendor_po.read","document.read"],
}

def require_permission(permission: str):
    def dependency(user: User = Depends(get_current_user)):
        perms = PERMISSIONS.get(user.role, [])
        if "*" in perms: return user
        ns = permission.split(".")[0]
        if any(p == permission or p == f"{ns}.*" for p in perms): return user
        raise HTTPException(403, f"Insufficient permissions: {permission}")
    return dependency
```

SECTION 6 — File Upload & Processing Pipeline

6.1 Upload Endpoint Spec

```
POST /api/documents/upload
Content-Type: multipart/form-data

Form fields:
  file:         binary (required)
  deal_id:      UUID (optional)
  entity_type:  rfq|proposal|customer_po|vendor_po|invoice|certificate|other
  entity_id:    UUID (optional)

Validation:
  MIME type: application/pdf, image/jpeg, image/png, image/tiff (python-magic)
  Max size: 50 MB
  Filename: sanitise with pathlib.Path(name).name

Storage: {BUCKET}/uploads/{deal_id or "general"}/{uuid4}_{original_name}

Response 202:
  {"document_id":"uuid","status":"processing","file_name":"...","file_size":12345}
```

6.2 Celery Parse Task

```
@celery.task(bind=True, max_retries=3, default_retry_delay=30)
def parse_document_task(self, document_id: str):
    try:
        doc = sync_db.get(Document, document_id)
        file_bytes = minio_client.get_object(BUCKET, doc.storage_key).read()

        # Extract text
        text = extract_with_pdfplumber(file_bytes)
        if is_scanned(text):   # heuristic: <50 chars extracted
            text = pytesseract_ocr(file_bytes)
        doc.raw_text = text[:100_000]

        # AI parse
        parser_map = {"rfq": parse_rfq, "proposal": parse_proposal,
                      "customer_po": parse_customer_po}
        if doc.entity_type in parser_map:
            doc.parsed_data = parser_map[doc.entity_type](text)

        # Embedding for semantic search
        doc.embedding = generate_embedding(f"{doc.entity_type}: {text[:8000]}")
        doc.parsed = True
        sync_db.commit()
        send_notification(doc.uploaded_by_id, "document_parsed", {})

    except Exception as exc:
        doc.parse_error = str(exc)
        sync_db.commit()
        raise self.retry(exc=exc)
```

6.3 Text Extraction Functions

```
def extract_with_pdfplumber(file_bytes: bytes) -> str:
    with pdfplumber.open(io.BytesIO(file_bytes)) as pdf:
        pages = [page.extract_text() or "" for page in pdf.pages]
    return "\n".join(pages)

def is_scanned(text: str, threshold: int = 50) -> bool:
    return len(text.strip()) < threshold

def pytesseract_ocr(file_bytes: bytes) -> str:
    from pdf2image import convert_from_bytes
    images = convert_from_bytes(file_bytes, dpi=300)
    return "\n".join(pytesseract.image_to_string(img) for img in images)
```

6.4 Signed Download URL

```
GET /api/documents/{id}/download

def get_download_url(document: Document) -> str:
    return minio_client.presigned_get_object(
        BUCKET, document.storage_key,
        expires=timedelta(minutes=15)   # Short-lived signed URL
    )

# Frontend opens in new tab — never proxy binary through FastAPI
```

SECTION 7 — Email Templates & Notification Rules

7.1 Notification Trigger Matrix

Trigger	Subject Line	Recipient	Schedule
Document parsed	Document ready	Uploader	Immediate
Quote expiring (7d)	Quote expiring soon	Deal owner (sales)	Daily 08:00
Quote expired	Quote expired	Deal owner (sales)	Day of expiry
PO discrepancy critical	Action required on PO	Deal owner + manager	Immediate
Payment due in 3d	Payment reminder	Finance team	Daily 08:00
Payment overdue 1d	Invoice overdue	Customer (AR)	Daily 08:00
Payment overdue 15d	Second reminder	Customer + finance mgr	Daily 08:00
Payment overdue 30d	Urgent: overdue invoice	Customer + director	Daily 08:00
Vendor PO confirmed	PO confirmed by vendor	Procurement + deal owner	Immediate
Deal closed	Deal closed — P&L; ready	Finance + deal owner	Immediate

7.2 Vendor Proposal Request Template

```
Subject: RFQ {deal.deal_number} — Request for Quotation: {deal.description}

Dear {vendor.primary_contact_name},

We seek quotations for the following oil & gas materials.
Please provide your best offer by {deadline}.

REQUIREMENTS:
{for each line_item}
  {idx}. {item.description}
     Spec:     {item.material_spec}
     Qty:      {item.quantity} {item.unit}
     Delivery: {item.required_delivery_date}
{end for}

Please include: unit price, total, lead time, payment terms, Incoterms,
validity (min 30 days), and available certifications (MTC, CoC, TPI).

Best regards,
{user.full_name} | {COMPANY_NAME} Procurement
```

7.3 Payment Reminder Templates (Tiered)

```
# Tier 1: 1-7 days overdue — polite
Subject: Friendly Reminder — Invoice {inv.invoice_number} Due {inv.due_date}
```

```

This is a courtesy reminder that invoice {inv.invoice_number} for
{inv.total} {inv.currency} was due on {inv.due_date}.

# Tier 2: 8-30 days overdue — firm
Subject: Payment Overdue — Invoice {inv.invoice_number} ({days} Days)

Invoice {inv.invoice_number} is now {days} days overdue. Please arrange
payment to avoid impacting your credit facility.

# Tier 3: 31+ days overdue — urgent
Subject: URGENT: Invoice {inv.invoice_number} Requires Immediate Attention

Despite previous reminders, this invoice remains unpaid {days} days past
due. We are escalating to our finance director. Contact us within 48 hours.
```

7.4 Celery Beat Schedule

```
app.conf.beat_schedule = {
    "daily-payment-reminders": {
        "task": "workers.payment_reminders.send_reminders",
        "schedule": crontab(hour=8, minute=0),
    },
    "daily-quote-expiry-check": {
        "task": "workers.quote_monitor.check_expiring",
        "schedule": crontab(hour=8, minute=15),
    },
    "hourly-fx-rate-refresh": {
        "task": "workers.fx_service.refresh_rates",
        "schedule": crontab(minute=0),
    },
    "weekly-credibility-recalc": {
        "task": "workers.vendor_score.recalculate_all",
        "schedule": crontab(day_of_week="sun", hour=2),
    },
}
```

SECTION 8 — Seed Data Specification

Run: `make seed` (executes backend/seeds/main.py). Creates realistic demo data for a 10-person oil & gas trading company in the UAE.

8.1 Demo Users

```
# All demo passwords: TradeFlow2026!
SEED_USERS = [
    {"email":"admin@tradeflow.demo",        "role":"admin",        "name":"Alex Admin"},
    {"email":"sarah.sales@tradeflow.demo",  "role":"sales",        "name":"Sarah Al-Rashid"},
    {"email":"mike.sales@tradeflow.demo",   "role":"sales",        "name":"Mike Johnson"},
    {"email":"omar.proc@tradeflow.demo",    "role":"procurement",  "name":"Omar Hassan"},
    {"email":"priya.proc@tradeflow.demo",   "role":"procurement",  "name":"Priya Sharma"},
    {"email":"fatima.fin@tradeflow.demo",   "role":"finance",      "name":"Fatima Al-Zaabi"},
    {"email":"chen.fin@tradeflow.demo",     "role":"finance",      "name":"Chen Wei"},
    {"email":"james.qc@tradeflow.demo",     "role":"quality",      "name":"James Okonkwo"},
    {"email":"luis.log@tradeflow.demo",     "role":"logistics",    "name":"Luis Garcia"},
    {"email":"anya.wh@tradeflow.demo",      "role":"warehouse",    "name":"Anya Petrov"},
]
```

8.2 Demo Customers (UAE Oil & Gas)

```
SEED_CUSTOMERS = [
    {"name":"ADNOC Onshore",       "type":"operator",   "credit_limit":5_000_000,  "currency":"AED"},
    {"name":"Shell Abu Dhabi B.V.","type":"operator",   "credit_limit":10_000_000, "currency":"USD"},
    {"name":"TechnipFMC UAE",      "type":"epc",        "credit_limit":3_000_000,  "currency":"USD"},
    {"name":"Petrofac Emirates",   "type":"epc",        "credit_limit":2_500_000,  "currency":"USD"},
    {"name":"Dragon Oil Limited",  "type":"operator",   "credit_limit":4_000_000,  "currency":"AED"},
    {"name":"McDermott ME",        "type":"contractor", "credit_limit":2_000_000,  "currency":"USD"},
    {"name":"Baker Hughes OFSE",   "type":"contractor", "credit_limit":6_000_000,  "currency":"USD"},
]
```

8.3 Demo Vendors

```
SEED_VENDORS = [
    {"name":"Tenaris Global",    "country":"Italy",   "credibility":90,
     "categories":["OCTG","Line Pipe","Casing"]},
    {"name":"Vallourec",         "country":"France",  "credibility":85,
     "categories":["Premium Tubes","Line Pipe","Fittings"]},
    {"name":"Sumitomo Corp",     "country":"Japan",   "credibility":82,
     "categories":["Line Pipe","Fittings","Valves"]},
    {"name":"JSW Steel",         "country":"India",   "credibility":68,
     "categories":["Line Pipe","Plates","Structurals"]},
    {"name":"Tianjin Pipe Corp", "country":"China",   "credibility":72,
     "categories":["ERW Pipe","SSAW Pipe","Fittings"]},
    {"name":"EUROPIPE GmbH",     "country":"Germany", "credibility":88,
     "categories":["Large Diameter Pipe","Offshore Pipeline"]},
    {"name":"Gulf Valves LLC",   "country":"UAE",     "credibility":75,
     "categories":["Valves","Actuators","Instrumentation"]},
]
```

8.4 Deal Scenario Distribution

Status	Count	Description
rfq_received	4	New RFQs awaiting sourcing
sourcing	3	Vendor search in progress
quoted	4	Quotes sent, awaiting customer response
po_received	3	Customer PO received, ordering from vendor
ordered	2	Vendor PO placed, tracking production
in_production	2	Production underway at vendor factory
shipped	1	Goods in transit to destination
invoiced	2	Awaiting final payment from customer
paid	2	Fully paid, pending formal close
closed	1	Closed with final P&L; calculated
cancelled	1	Cancelled at sourcing stage

8.5 Deal Value Ranges

```
DEAL_VALUE_RANGES = {
    "small":  (50_000,   200_000),   # AED — 40% of deals
    "medium": (200_001, 1_000_000),  # AED — 45% of deals
    "large":  (1_000_001,5_000_000), # AED — 15% of deals
}

# Seed total: 25 deals across all statuses
# Each deal has: 2-5 line items, 1-3 vendor proposals, 1 quote
# Closed deals have: customer PO, vendor PO, invoice, payments, P&L;
```