

Blackletter Systems – Full Build Plan (Unified)

Tagline: *Old rules. New game.*

1) Overview

Blackletter Systems blends traditional legal expertise with cutting-edge AI automation to deliver end-to-end workflows that save time, reduce cost, and give clients a decisive edge.

2) Stage 1 – Getting Set Up (Beginner-Friendly)

Before launching services, lay down the foundations:

1. **Business Email & Domain** – Domain + pro email (Zoho Mail free tier or Google Workspace).
2. **Website** – No-code site (Google Sites, Carrd, Webflow) with offers, demos, contact forms.
3. **CRM** – HubSpot Free CRM or Zoho CRM for leads, deals, sequences.
4. **File Storage** – Google Drive (15GB free) or OneDrive; create folders per client/matter.
5. **AI & Automation Tools** – OCR (Tesseract/Google Vision), LLMs (OpenAI/Anthropic), e-sign (DocuSign API), vector DB (Weaviate/Pinecone free tiers).
6. **Payments** – Stripe or PayPal (no monthly fee; pay per transaction).

Time to MVP: 3–5 weeks \ **Startup Cost:** £0–£1,500 \ **Ongoing Cost:** £0–£400/month

3) Core Services & Micro-SaaS (with Code Stubs)

3.1 AI Contract Review Micro-SaaS (White-Label + Niche)

Problem: SMEs waste hours reviewing NDAs, leases, and standard contracts. **Solution:** API-based review tailored to a niche (e.g., *AI Lease Checker for Property Lawyers*).

Backend (FastAPI)

```
# backend/routers/contracts.py
from fastapi import APIRouter, UploadFile, File, Form
from models.schemas import ReviewResult
router = APIRouter()

@router.post("/review", response_model=ReviewResult)
async def review_contract(
    file: UploadFile = File(...),
    doc_type: str = Form("Lease"),
    jurisdiction: str = Form("UK"),
    playbook_id: str | None = Form(None),
):
```

```

# TODO: OCR → clause detection → LLM risk scoring → redlines
_ = await file.read()
return ReviewResult(summary="Draft summary placeholder", risks=[],
redlines={}, next_actions=["Upload playbook and re-run for richer scoring"])

```

Frontend (Next.js)

```

// app/upload/page.tsx
import UploadDropzone from '@components/UploadDropzone';
export default function UploadPage(){
  return (
    <section className="space-y-4">
      <h2 className="text-2xl font-semibold">Contract Review</h2>
      <UploadDropzone />
    </section>
  );
}

```

Agent TODOs

- Implement OCR/text extraction (`pypdf`, `pytesseract`).
- Clause heuristics (regex for termination, repair, rent review, assignment, insurance).
- Risk scoring against YAML playbook; generate `python-docx` redlines.
- Export: `review.json`, `summary.md`, `redlined.docx`, `client_pack.zip`.

3.2 AI-Enhanced Compliance Checklist Service

Problem: Legal teams struggle to track scattered regulatory updates. **Solution:** Paid newsletter/web app generating sector-specific checklists.

Backend (FastAPI)

```

# backend/routers/compliance.py
from fastapi import APIRouter
from pydantic import BaseModel
router = APIRouter()

class ChecklistItem(BaseModel):
    title: str; priority: int; due_date: str | None = None
class ChecklistResponse(BaseModel):
    sector: str; items: list[ChecklistItem]

@router.get("/checklist", response_model=ChecklistResponse)
async def generate_checklist(sector: str = "SaaS"):
    # TODO: ingest feeds → summarise → actions
    return ChecklistResponse(sector=sector, items=[
        ChecklistItem(title="Map data flows (PII)", priority=1),

```

```
        ChecklistItem(title="Review DPIA templates", priority=2),
    ])
}
```

Frontend (Next.js)

```
// app/compliance/page.tsx
'use client';
import { useState } from 'react';
import { api } from '@lib/api';
export default function CompliancePage(){
  const [sector, setSector]=useState('SaaS');
  const [res, setRes]=useState<any>(null);
  return (
    <section className="space-y-3">
      <h2 className="text-2xl font-semibold">Compliance Checklist</h2>
      <input className="w-full p-2 text-black" value={sector}
onChange={e=>setSector(e.target.value)} />
      <button className="px-4 py-2 rounded bg-white text-black"
onClick={async()=>{
        const {data}=await api.get('/compliance/checklist',{params:
{sector}});
        setRes(data);
      }}>Generate</button>
      {res && <pre className="bg-neutral-900 p-3 rounded text-xs overflow-
auto">{JSON.stringify(res, null, 2)}</pre>}
    </section>
  );
}
```

Agent TODOs

- Ingest Google Alerts/RSS (ICO, FCA, EU Commission, gov.uk) with APScheduler.
- Deduplicate, tag (jurisdiction, sector), summarise to actions (P1-P3).
- Render weekly PDF (WeasyPrint) + dashboard; email via SendGrid/Resend.

3.3 AI Discovery & Research Assistant for Case Law

Problem: Associates lose days finding relevant precedents. **Solution:** Semantic search + synthesis over curated corpus.

Backend (FastAPI)

```
# backend/routers/research.py
from fastapi import APIRouter
from pydantic import BaseModel
router = APIRouter()
```

```

class QueryRequest(BaseModel):
    question: str; area: str = "employment"; jurisdiction: str = "England & Wales"
class Citation(BaseModel):
    case_id: str; para: str | None = None
class QueryResponse(BaseModel):
    answer: str; citations: list[Citation]

@router.post("/query", response_model=QueryResponse)
async def query_law(req: QueryRequest):
    # TODO: RAG over vector DB
    return QueryResponse(answer="Placeholder answer; connect to vectors",
        citations=[Citation(case_id="[2014] UKSC 13", para="27")])

```

Frontend (Next.js)

```

// app/research/page.tsx
'use client';
import { useState } from 'react';
import { api } from '@lib/api';
export default function ResearchPage(){
    const [q,setQ] = useState('What is the approach to implied term of trust and confidence?');
    const [res,setRes] = useState<any>(null);
    return (
        <section className="space-y-3">
            <h2 className="text-2xl font-semibold">Research Assistant</h2>
            <input className="w-full p-2 text-black" value={q}
onChange={e=>setQ(e.target.value)} />
            <button className="px-4 py-2 rounded bg-white text-black"
onClick={async()=>{
                const {data}=await api.post('/research/query',{question:q,
area:'employment', jurisdiction:'England & Wales'});
                setRes(data);
            }}>Ask</button>
            {res && <pre className="bg-neutral-900 p-3 rounded text-xs overflow-auto">{JSON.stringify(res, null, 2)}</pre>}
        </section>
    );
}

```

Agent TODOs

- Build corpus (BAILII + legislation.gov.uk), chunk 1–2k tokens, embed to Weaviate/Pinecone.
- Retrieve-then-answer; enforce paragraph-level citations; memo export (.docx).

4) Global Technical Standards

- **Langs:** Python 3.11 (backend), TypeScript/Next.js 14 (frontend).
 - **Infra:** Windows 11 dev; Docker Desktop optional.
 - **Auth:** Clerk/Auth0 or Supabase magic link.
 - **Storage:** PostgreSQL (Supabase) or SQLite for MVP; Supabase Storage for files.
 - **LLMs:** OpenAI + Anthropic fallback.
 - **Vectors:** Weaviate/Pinecone free tier.
 - **Security:** `.env` for secrets, GDPR compliance, audit logs.
-

5) Run Locally (Win11)

Backend

```
cd blackletter\backend
python -m venv ..\.venv
. ..\.venv\Scripts\Activate.ps1
pip install -r requirements.txt
uvicorn main:app --reload --port 8000
```

Frontend

```
cd blackletter-frontend
$env:NEXT_PUBLIC_API_URL="http://localhost:8000"
npm run dev
```

You now have:

- `http://localhost:8000/docs` (OpenAPI)
 - `http://localhost:3000/upload` (contract demo)
 - `http://localhost:3000/research` (RAG stub UI)
 - `http://localhost:3000/compliance` (checklist stub UI)
-

6) Monetisation

- SaaS subscriptions (per seat or per firm).
 - Setup/playbooks (one-off fees).
 - Done-for-you consulting and SLAs.
 - Licensing workflows to firms.
-

7) Acceptance, Security & Next Steps

Acceptance Tests

- Contract: flags key clause types; generates coherent redlines.
- Research: answers include ≥ 2 primary authorities with paragraph-level cites.
- Compliance: ≥ 5 actionable tasks produced per update batch with links + dates.

Security

- File scanning, least-privilege access, encryption at rest, 30-day retention default.

Next Steps

- Fill in OCR/LLM logic; connect vectors; automate feeds; add audit logs & tests.

Blackletter Systems turns complex legal work into fast, precise, and profitable automated solutions—now fully mapped from idea to code and launch.