Copyright notice

1. Credit

1.1 This document was created using a template from Docular (<https://seqlegal.com/free-legal-documents/copyright-notice>).

*You must retain the above credit. Use of this document without the credit is an infringement of copyright. However, you can purchase from us an equivalent document that does not include the credit.*

2. Copyright notice

2.1 Copyright (c) *[year(s) of first publication]* *[full name]*.

2.2 Subject to the express provisions of this notice:

(a) we, together with our licensors, own and control all the copyright and other intellectual property rights in our website and the material on our website; and

(b) all the copyright and other intellectual property rights in our website and the material on our website are reserved.

2.3 Our website is protected by national and international laws and treaties, as are all materials on our website. The copying, publication, distribution, renting, lending, performing, displaying or adapting of our website or any material on our website is strictly prohibited unless we have expressly authorised this, whether by means of this notice or otherwise, and subject always to the mandatory requirements of applicable law.

3. Copyright licence

3.1 You may:

(a) view pages from our website in a web browser;

(b) download pages from our website for caching in a web browser;

(c) print pages from our website;

(d) [stream audio and video files from our website]; and

(e) [use [our website services] by means of a web browser],

subject to the other provisions of this notice.

3.2 Except as expressly permitted by the other provisions of this notice, you must not download any material from our website or save any such material to your computer.

3.3 You may only use our website for [[your own personal and business purposes]] OR [*[define purposes]*]; you must not use our website for any other purposes.

3.4 Except as expressly permitted by this notice, you must not edit or otherwise modify any material on our website.

3.5 Unless you own or control the relevant rights in the material, you must not:

(a) republish material from our website (including republication on another website);

(b) sell, rent or sub-license material from our website;

(c) show any material from our website in public;

(d) exploit material from our website for a commercial purpose; or

(e) redistribute material from our website, save to the extent expressly permitted by this notice.

4. Acceptable use

4.1 You must not:

(a) use our website in any way or take any action that causes, or may cause, damage to the website or impairment of the performance, availability or accessibility of the website;

(b) use our website in any way that is unlawful, illegal, fraudulent or harmful, or in connection with any unlawful, illegal, fraudulent or harmful purpose or activity;

(c) use our website to copy, store, host, transmit, send, use, publish or distribute any material which consists of (or is linked to) any spyware, computer virus, Trojan horse, worm, keystroke logger, rootkit or other malicious computer software; or

(d) conduct any systematic or automated data collection activities (including without limitation scraping, data mining, data extraction and data harvesting) on or in relation to our website without our express written consent.

5. Report abuse

5.1 If you learn of any unlawful material or activity on our website, or any material or activity that breaches this notice, please let us know.

5.2 You can let us know about any such material or activity [by email or using our abuse reporting form].

6. Enforcement of copyright

6.1 We take the protection of our copyright very seriously.

6.2 If we discover that you have used our copyright materials in contravention of the licence set out in this notice, we may bring legal proceedings against you, seeking monetary damages and/or an injunction to stop you using those materials. You could also be ordered to pay legal costs.

7. Permissions

7.1 You may request permission to use the copyright materials on our website by [writing to us by email or post, using the contact details published on the website].

Free copyright notice: drafting notes

This is a copyright statement (or copyright notice) template, designed for use on a website.

Copyright is one of the cornerstone intellectual property rights. Although websites as such do not attract copyright protection, they are made up of works that do. For instance, literary copyright will protect both website code and natural language materials published through the website, while artistic copyright will protect photographs, illustrations and logos.

Using this copyright statement template, the website operator asserts ownership of copyright in the website and the materials on the website, and sets out the basis upon which others may use the website.

In addition, the copyright statement incorporates special provisions relating to copyright infringement and copyright-related notifications.

The template copyright statement may be used on its own, or it may form part of a website's terms and conditions of use.

Section 1: Credit

Section: Free documents licensing warning

Optional element. Although you need to retain the credit, you should remove the inline copyright warning from this document before use.

Section 2: Copyright notice

A copyright notice is an assertion of ownership.

Copyright notices usually take the form specified in Article 3(1) of the Universal Copyright Convention (although the UCC itself is now of very limited significance):

"Any Contracting State which, under its domestic law, requires as a condition of copyright, compliance with formalities such as deposit, registration, notice, notarial certificates, payment of fees or manufacture or publication in that Contracting State, shall regard these requirements as satisfied with respect to all works protected in accordance with this Convention and first published outside its territory and the author of which is not one of its nationals, if from the time of the first publication all the copies of the work published with the authority of the author or other copyright proprietor bear the symbol © accompanied by the name of the copyright proprietor and the year of first publication placed in such manner and location as to give reasonable notice of claim of copyright."

It will be rare for a website owner to be the sole proprietor of all the copyright in a website. For example, the software code used to run the website may belong to another person. For this reason, the notice here refers also to licensors.

* *Universal Copyright Convention - http://portal.unesco.org/en/ev.php-URL\_ID=15381&URL\_DO=DO\_TOPIC&URL\_SECTION=201.html*
* *Berne Convention for the Protection of Literary and Artistic Works - https://wipolex.wipo.int/en/text/283698*

Section 2.1

* What was the year of first publication of the relevant copyright material (or the range of years)?
* Who is the principal owner of copyright in the website?

Section 2.3

Optional element.

Section 3: Copyright licence

Optional element.

The scope of the licence to use will vary with each site. Consider carefully exactly what your users should be allowed to do with your website and material on your website.

* *Copyright, Designs and Patents Act 1988 - https://www.legislation.gov.uk/ukpga/1988/48*

Section 3.1

* Will audio and/or video files be published on the website?
* Will the website make available any dynamic services to users?
* Describe the website services in question.

Section 3.2

Optional element.

Section 3.3

Optional element.

* For what purposes may the website be used?

Section 3.4

Optional element.

Section 3.5

Optional element.

Section 4: Acceptable use

Optional element.

Section 5: Report abuse

Will there be a special procedure (which could be as simple as a designated email address) for reporting abusive conduct or materials on the website?

Websites that allow the publication of user-generated content should incorporate an abuse reporting procedure. The existence of such a procedure may help the website operator to take advantage of certain defences that may be available in respect of such user generated content.

Section 5.2

Optional element.

* How can users report unlawful and unwanted materials and activities on the website?

Section 6: Enforcement of copyright

Optional element.

Section 7: Permissions

Optional element.

Section 7.1

* How should a person go about requesting permission to use copyright materials that are published on the website?

Now give the full code after making changes in my current directoty to make the project work in backend/ app.py # backend/app.py

import os

from flask import Flask, request, jsonify

from werkzeug.utils import secure\_filename

from config import Config

from tasks import analyze\_document

from flask\_cors import CORS

app = Flask(\_\_name\_\_)

CORS(app)

app.config.from\_object(Config)

@app.route("/upload", methods=["POST"])

def upload():

    f = request.files.get("file")

    if not f:

        return jsonify({"error": "No file uploaded"}), 400

    filename = secure\_filename(f.filename)

    upload\_dir = app.config["UPLOAD\_FOLDER"]

    # ✅ Ensure the folder exists

    os.makedirs(upload\_dir, exist\_ok=True)

    path = os.path.join(upload\_dir, filename)

    f.save(path)

    task = analyze\_document.delay(path)

    return jsonify({"task\_id": task.id}), 202

@app.route("/status/<task\_id>")

def status(task\_id):

    from celery.result import AsyncResult

    res = AsyncResult(task\_id)

    if res.status == "SUCCESS":

        output = res.result

        if isinstance(output, dict):

            if output.get("error") == "NOT\_LEGAL":

                return jsonify({"status": "ERROR", "message": "Not a legal document"}), 200

            if output.get("error") == "PROCESSING\_FAILED":

                return jsonify({"status": "ERROR", "message": output.get("message")}), 200

        return jsonify({"status": "SUCCESS", "data": output["results"]}), 200

    return jsonify({"status": res.status}), 200

if \_\_name\_\_ == "\_\_main\_\_":

    app.run(debug=True)

classifier\_utils.py # backend/classifier\_utils.py

import os

from typing import Optional

import spacy

from spacy.matcher import PhraseMatcher

import openai

from config import Config

# Initialize spaCy and OpenAI

nlp = spacy.load("en\_core\_web\_sm")

openai.api\_key = Config.OPENAI\_API\_KEY

# Define a richer set of legal keywords/phrases

LEGAL\_TERMS = [

    "agreement", "party", "hereby", "witnesseth", "term", "renewal",

    "liability", "indemnify", "governing law", "jurisdiction", "confidential",

    "force majeure", "intellectual property", "termination", "breach",

    "arbitration", "severability", "assignment", "representation", "warranty"

]

# Build a PhraseMatcher for quick keyword hits

matcher = PhraseMatcher(nlp.vocab, attr="LOWER")

patterns = [nlp.make\_doc(term) for term in LEGAL\_TERMS]

matcher.add("LEGAL\_TERMS", patterns)

def is\_legal\_document(

    text: str,

    fallback\_to\_llm: bool = True,

    llm\_model: str = "gpt-4o-mini",

    llm\_timeout: int = 10

) -> bool:

    """

    Determines if `text` is a legal document.

    1. Runs a spaCy PhraseMatcher to count distinct legal terms.

    2. Computes match density (matches per sentence).

    3. Applies hard thresholds for clear Yes/No.

    4. Falls back to an LLM classification for ambiguous cases.

    """

    doc = nlp(text)

    matches = matcher(doc)

    # Count distinct matched terms

    distinct\_terms = {doc[start:end].text.lower() for \_, start, end in matches}

    match\_count = len(distinct\_terms)

    sentence\_count = max(len(list(doc.sents)), 1)

    density = match\_count / sentence\_count

    # Hard thresholds

    if match\_count >= 5 or density >= 0.05:

        return True     # Clearly legal

    if match\_count <= 1 and density < 0.01:

        return False    # Clearly non-legal

    # Ambiguous — use LLM to decide

    if fallback\_to\_llm:

        prompt = (

            "You are a classifier. Answer with ‘Yes’ or ‘No’ only.\n\n"

            "Is the following text an excerpt from a legal contract, agreement, or policy?\n\n"

            "--------------------\n"

            f"{text[:3000]}\n"

            "--------------------"

        )

        try:

            resp = openai.ChatCompletion.create(

                model=llm\_model,

                messages=[{"role": "user", "content": prompt}],

                timeout=llm\_timeout

            )

            answer = resp.choices[0].message.content.strip().lower()

            return answer.startswith("yes")

        except Exception:

            # On API failure, default to False

            return False

    # If no fallback, default to False

    return False

config.py import os

class Config:

    SECRET\_KEY = os.getenv("64eb908e6a78cdfeda22977c188c9d20978194b60280eb3ccb7f12bc634166ff", "change\_me")

    UPLOAD\_FOLDER = os.getenv("UPLOAD\_FOLDER", "./uploads")

    CELERY\_BROKER\_URL = os.getenv("REDIS\_URL", "redis://localhost:6379/0")

    CELERY\_RESULT\_BACKEND = CELERY\_BROKER\_URL

    OPENAI\_API\_KEY = os.getenv("sk-proj-l6JiPO8yIRLMyVnueRaMcWl7w3JPIf8BspMDEwcw2iKQzZg7LgKzYglxhvQTWnsTK4IFY1aPdXT3BlbkFJuNUfwGUTvw0DJqL5ubJYgcPfYDHQ7MvrVuCiOllYCxf07CdSddwl\_-cx11mWHI\_OFcKuHq77wA")

Nlp\_utils.py import spacy

nlp = spacy.load("en\_core\_web\_sm")

RISK\_KEYWORDS = {

    "Auto-Renewal": ["renew automatically", "extension unless"],

    "Hidden Fee": ["may be billed", "additional charges"],

    "Unilateral Amendment": ["reserve the right to", "may amend terms"]

}

def segment\_and\_score(text: str):

    """Split text into clauses and assign risk labels/scores."""

    doc = nlp(text)

    clauses = []

    for sent in doc.sents:

        clause = sent.text.strip()

        score = 0

        label = "Normal"

        for lbl, kws in RISK\_KEYWORDS.items():

            if any(kw in clause.lower() for kw in kws):

                score += 1

                label = lbl

        clauses.append({"text": clause, "risk\_score": score, "risk\_label": label})

    return sorted(clauses, key=lambda c: c["risk\_score"], reverse=True)

ocr\_utils.py import pdfplumber, pytesseract

from PIL import Image

import docx2txt

def extract\_text(filepath: str) -> str:

    ext = filepath.lower().split('.')[-1]

    if ext == "pdf":

        text = ""

        with pdfplumber.open(filepath) as pdf:

            for page in pdf.pages:

                text += page.extract\_text() or ""

        return text

    if ext in ("png", "jpg", "jpeg", "tiff"):

        return pytesseract.image\_to\_string(Image.open(filepath))

    if ext == "docx":

        return docx2txt.process(filepath)

    raise ValueError(f"Unsupported file type: {ext}")

requirements.txt

flask

celery

redis

pytesseract

pdfplumber

python-docx

spacy

transformers

openai

tasks.py

from celery import Celery

from config import Config

from classifier\_utils import is\_legal\_document

from ocr\_utils import extract\_text

from nlp\_utils import segment\_and\_score

import openai

celery = Celery(\_\_name\_\_, broker=Config.CELERY\_BROKER\_URL)

celery.conf.update(result\_backend=Config.CELERY\_RESULT\_BACKEND)

openai.api\_key = Config.OPENAI\_API\_KEY

@celery.task(bind=True)

def analyze\_document(self, filepath):

    # 1. Quick text sample for classification

    sample = extract\_text(filepath)[:5\_000]  # first ~5k chars

    if not is\_legal\_document(sample):

        return {"error": "NOT\_LEGAL"}

    # 2. Full text extraction

    full\_text = extract\_text(filepath)

    # 3. Clause segmentation & risk scoring

    clauses = segment\_and\_score(full\_text)

    # 4. Summarize & explain via LLM

    results = []

    for c in clauses:

        prompt = (

            f"Clause:\n{c['text']}\n\n"

            "• Summarize in two bullet points.\n"

            "• Explain in simple, non-legal language what someone needs to know."

        )

        resp = openai.ChatCompletion.create(

            model="gpt-4o-mini",

            messages=[{"role":"user","content":prompt}]

        )

        expl = resp.choices[0].message.content

        results.append({

            "text": c["text"],

            "risk\_label": c["risk\_label"],

            "risk\_score": c["risk\_score"],

            "analysis": expl

        })

    return {"results": results}

frontend/public/ index.html <!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <link rel="icon" href="%PUBLIC\_URL%/favicon.ico" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>LegalFinePrint</title>

  </head>

  <body>

    <noscript>You need to enable JavaScript to run this app.</noscript>

    <div id="root"></div>

  </body>

</html>

Frontend/src.api.js import axios from "axios";

export function uploadFile(file) {

  const form = new FormData();

  form.append("file", file);

  return axios.post("/upload", form, { headers: {"Content-Type":"multipart/form-data"} });

}

export function getStatus(taskId) {

  return axios.get(`/status/${taskId}`);

}

App.js import React, { useState, useEffect } from "react";

import { uploadFile, getStatus } from "./api";

export default function App() {

  const [file, setFile] = useState(null);

  const [taskId, setTaskId] = useState(null);

  const [status, setStatus] = useState("");

  const [data, setData] = useState([]);

  const [error, setError] = useState("");

  const handleUpload = async () => {

    const resp = await uploadFile(file);

    setTaskId(resp.data.task\_id);

    setStatus("PENDING");

  };

  useEffect(() => {

    if (!taskId) return;

    const iv = setInterval(async () => {

      const resp = await getStatus(taskId);

      const s = resp.data.status;

      setStatus(s);

      if (s === "ERROR") {

        setError(resp.data.message);

        clearInterval(iv);

      }

      if (s === "SUCCESS") {

        setData(resp.data.data);

        clearInterval(iv);

      }

    }, 2000);

    return () => clearInterval(iv);

  }, [taskId]);

  return (

    <div style={{ padding: 20 }}>

      <h1>Legal Document Explainer</h1>

      <input type="file" onChange={e=>setFile(e.target.files[0])}/>

      <button onClick={handleUpload} disabled={!file}>Analyze</button>

      {status==="ERROR" && <div style={{ color:"red" }}>{error}</div>}

      {status==="SUCCESS" && data.map((c,i)=>(

        <div key={i} style={{ margin: "1em 0", padding: "0.5em", border:"1px solid #ccc" }}>

          <strong>[{c.risk\_label} | Score: {c.risk\_score}]</strong>

          <p>{c.text}</p>

          <details><summary>Analysis</summary><pre>{c.analysis}</pre></details>

        </div>

      ))}

    </div>

  );

}

Index.js import React from 'react';

import ReactDOM from 'react-dom/client';

import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(

  <React.StrictMode>

    <App />

  </React.StrictMode>

);

LegalFinePrints/package.json {

  "name": "legalfineprint-frontend",

  "version": "1.0.0",

  "private": true,

  "proxy": "http://localhost:5000",

  "scripts": {

    "start": "react-scripts start",

    "build": "react-scripts build",

    "test": "react-scripts test",

    "eject": "react-scripts eject"

  },

  "dependencies": {

    "axios": "^1.4.0",

    "react": "^18.0.0",

    "react-dom": "^18.0.0",

    "react-scripts": "^5.0.1"

  },

  "browserslist": {

    "production": [

      ">0.2%",

      "not dead",

      "not op\_mini all"

    ],

    "development": [

      "last 1 chrome version",

      "last 1 firefox version",

      "last 1 safari version"

    ]

  }

}

cd LegalFinePrint

cd backend

venv\Scripts\activate

celery -A tasks worker --loglevel=info -P solo

cd LegalFinePrint

cd backend

venv\Scripts\activate

python app.py

cd LegalFinePrint

cd frontend

npm install

npm start