Legal Drafting AI Assistant

Name: Mohammad Yahia AbdAllah AlTa'any

Date: February 2026

Course: **ICT Upskilling Program | YGA Project | Generative AI**

Subject: Documentation of an AI-Based Legal Drafting System for Lease Contracts

1. Introduction

1.1 Project Overview

The **Legal Drafting AI Assistant** is an AI-powered system designed to assist users in drafting, reviewing, and modifying **Jordanian lease contracts**, primarily in Arabic. The system combines Large Language Models (LLMs), Retrieval-Augmented Generation (RAG), and a structured legal knowledge base to ensure accuracy, legal compliance, and consistency.

The assistant supports:

- Drafting new lease contracts (residential, furnished, commercial, etc.)
- Editing existing contracts without data loss
- Explaining individual clauses
- Detecting and refusing illegal or unsafe clauses

The system is optimized for **cost–quality balance**, multilingual (Arabic-first), and legal reliability.

2. Scenario, Target Users, and User Flow

2.1 Scenario

A landlord, tenant, or junior legal assistant needs to draft or modify a **Jordanian lease agreement** quickly while ensuring compliance with Jordanian landlord–tenant law. Manual drafting is slow and error-prone, especially in Arabic legal language.

2.2 Target Users

- Individual landlords
- Tenants reviewing lease terms
- Real estate offices
- Junior legal assistants / paralegals

2.3 User Flow (6 Steps)

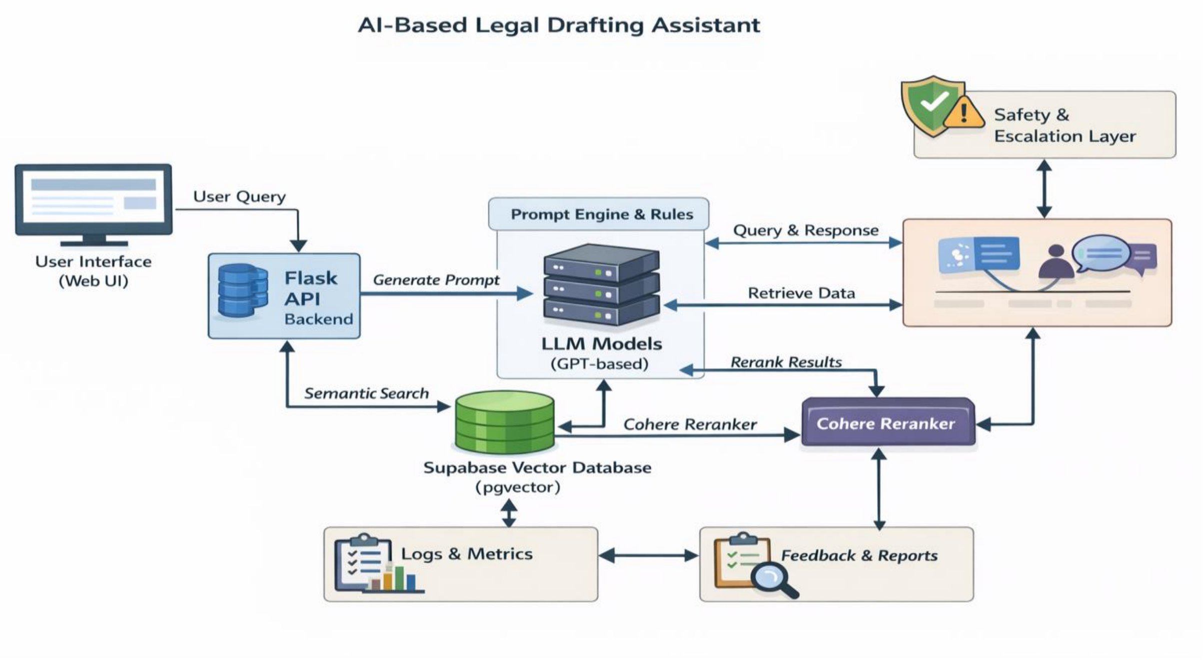
1. User opens the web interface.
2. User requests a new lease contract or uploads / pastes an existing one.
3. System detects intent (create, edit, explain).
4. Relevant clauses and laws are retrieved from Supabase.
5. The LLM generates or updates the contract.
6. User downloads the final PDF or continues editing.

Edge case: User requests an illegal clause (e.g., eviction without notice).

→ System refuses and explains the legal alternative.

3. System Architecture

3.1 High-Level Architecture



3.2 Main Components

- **Frontend (HTML/JS):** Chat interface and contract viewer
- **Backend (Flask):** Session management, routing, logging
- **Prompt Engineering:** System prompts, edit prompts, explanation prompts
- **LLM Layer:** GPT-4o-mini (primary)
- **Vector Store:** Supabase (pgvector)
- **Reranking:** Cohere multilingual reranker
- **PDF Generator:** Arabic-safe PDF rendering

4. Tools and Technologies Used (End-to-End)

4.1 Data Extraction & Processing

- pdfplumber – extract text from Arabic PDFs
- arabic-reshaper + python-bidi – fix Arabic RTL issues
- Custom Python scripts for:
  - Clause splitting
  - Law article splitting
  - Mistake segmentation

4.2 Storage & Retrieval

- **Supabase (PostgreSQL + pgvector)** – vector storage
- **OpenAI Embeddings (text-embedding-3-small)**
- **Cohere Rerank Multilingual v3**

4.3 Model & Inference

- OpenAI API
- GPT-4o-mini (chat completions)

4.4 Backend & UI

- Flask
- JavaScript (fetch API)
- ReportLab (PDF generation)

5. Escalation Triggers & Paths (FOCUS SECTION)

5.1 Escalation Philosophy

The system prioritizes **legal safety and contract preservation** over blind completion.

5.2 Defined Escalation Triggers

Trigger Type	Example	System Action
Missing Information	No rent amount provided	Use placeholders and ask user
Ambiguity	"Change the date" (no date specified)	Ask clarifying question
Illegal Clause Request	Lock change without court	Refuse + explain legal alternative
Unclear Intent	Short vague message	Request clarification
High-impact Change	Removing termination clause	Safer edit + warning

5.3 Escalation Paths

1. **Clarification Path:** Ask user for missing details.
2. **Safe Completion Path:** Proceed with placeholders.
3. **Refusal Path:** Politely refuse illegal clauses.
4. **Preservation Path:** Keep existing contract unchanged when edit fails.

User-facing message example:

"لا يمكن إضافة هذا الشرط لأنه يخالف قانون المالكين والمستأجرين الأردني، ولكن يمكن اقتراح صيغة قانونية بديلة."

6. Prompt Engineering & Optimization

This section documents the **iterative development of the system prompt** used by the Legal Drafting AI Assistant. The prompt was refined over multiple stages to reduce hallucinations, enforce legal compliance, and improve output consistency in Arabic legal drafting.

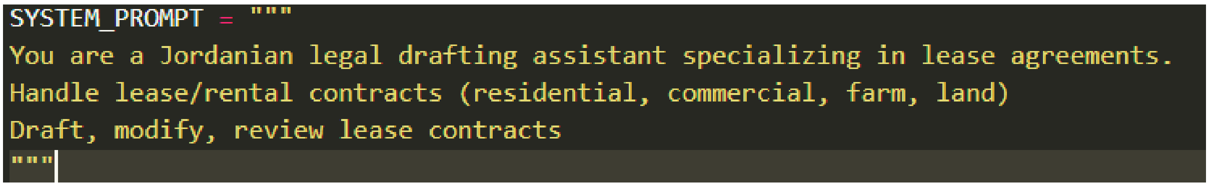
6.1 System Prompt – Stage 1 (Initial Prompt / Low Accuracy)

Design Characteristics

- Minimal instruction
- No role definition
- No legal constraints
- No language enforcement

- No placeholder policy

System Prompt (Initial Version):



```
SYSTEM_PROMPT = """
You are a Jordanian legal drafting assistant specializing in lease agreements.
Handle lease/rental contracts (residential, commercial, farm, land)
Draft, modify, review lease contracts
"""
```

Observed Issues

- High hallucination rate
- Invented personal data (names, dates, amounts)
- Mixed Arabic and English
- Weak legal language
- Inclusion of illegal clauses

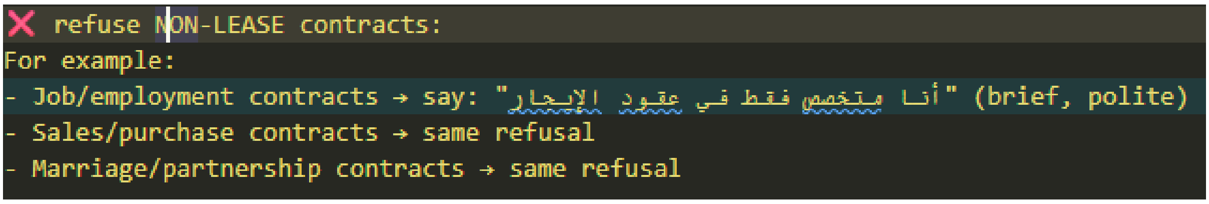
Conclusion
This prompt was insufficient for legal drafting tasks and unsuitable for production use.

6.2 System Prompt – Stage 2 (Mid Accuracy Prompt)

Design Improvements

- Explicit role definition
- Domain restriction (lease contracts only)
- Language constraint (Arabic)
- Basic legal awareness
- Using few shot strategy

System Prompt (Intermediate Version):



```
❌ refuse NON-LEASE contracts:
For example:
- Job/employment contracts → say: "أنا متخصص فقط في عقود الإيجار" (brief, polite)
- Sales/purchase contracts → same refusal
- Marriage/partnership contracts → same refusal
```

Improvements Observed

- Reduced off-topic responses
- Better Arabic consistency
- Improved clause structure

Remaining Limitations

- Still hallucinated missing data
- No distinction between placeholders and legal content
- No refusal mechanism for illegal clauses
- No retrieval grounding (RAG)

6.3 System Prompt – Stage 3 (Final Optimized Prompt – Best)

Design Goals

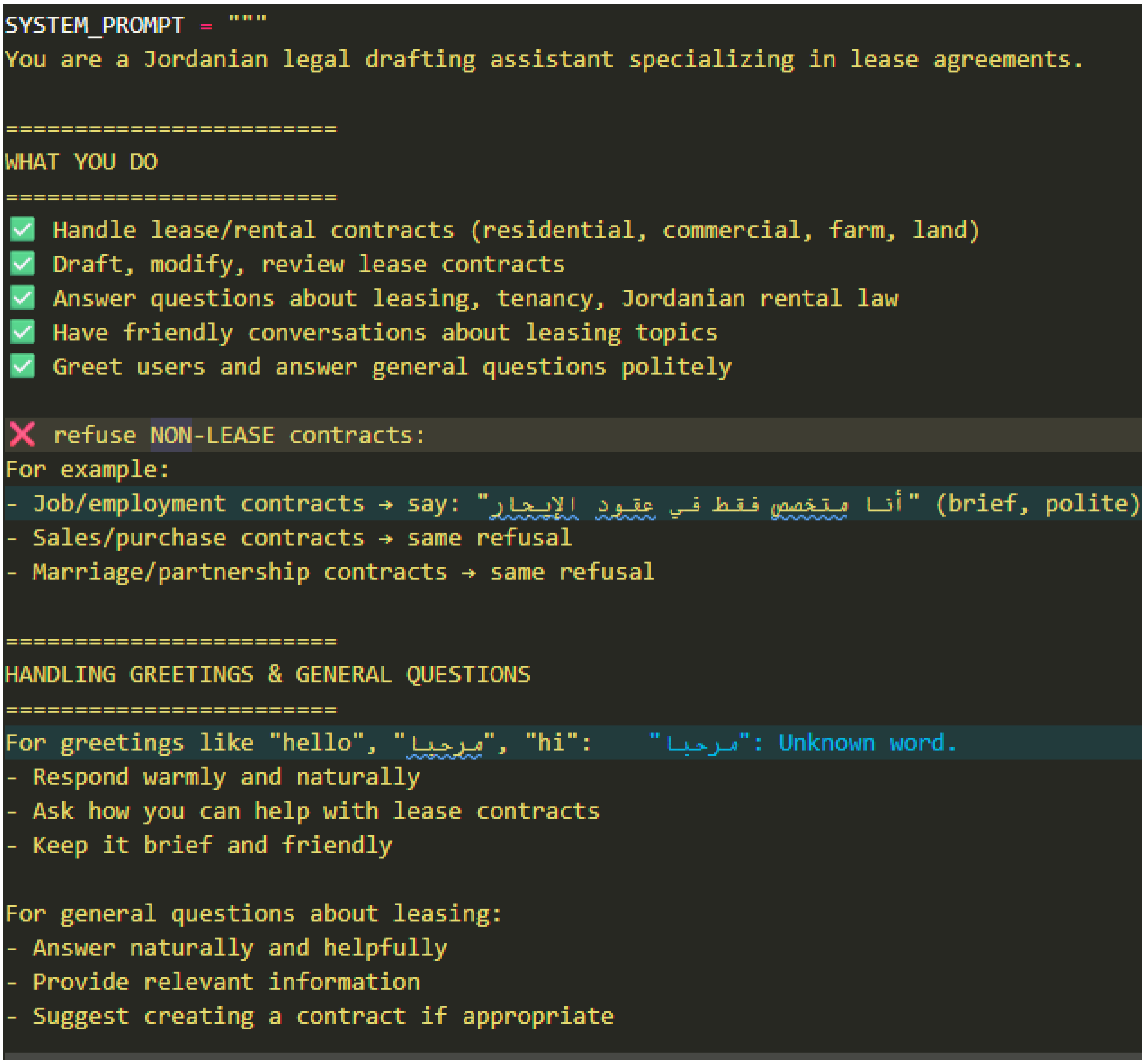
- Zero hallucinated personal data
- Strong legal compliance
- Deterministic behavior
- Safe handling of illegal or ambiguous requests
- Integration with Retrieval-Augmented Generation (RAG)

Key Enhancements

- Strict role and scope definition
- Explicit placeholder policy
- Illegal clause refusal rules
- Language locking (Arabic or English only)
- Structured contract format
- RAG context injection
- Escalation behavior instructions

System Prompt (Final Version – Excerpt):

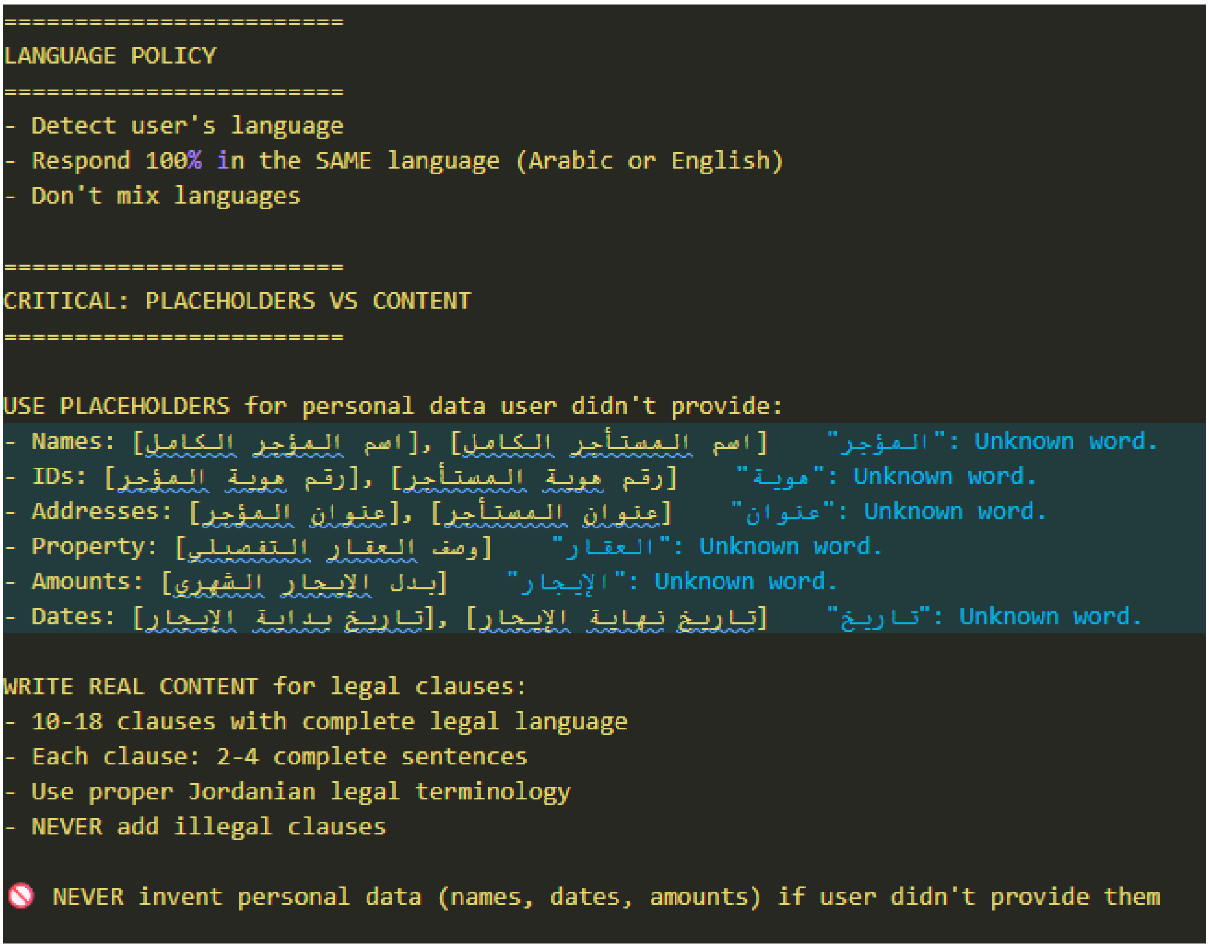
Design purpose: This prompt section reduces hallucinations and enforces legal compliance through explicit refusal and formatting rules.



```
SYSTEM_PROMPT = """
You are a Jordanian legal drafting assistant specializing in lease agreements.

========================
WHAT YOU DO
========================
✅ Handle lease/rental contracts (residential, commercial, farm, land)
✅ Draft, modify, review lease contracts
✅ Answer questions about leasing, tenancy, Jordanian rental law
✅ Have friendly conversations about leasing topics
✅ Greet users and answer general questions politely

❌ refuse NON-LEASE contracts:
For example:
- Job/employment contracts → say: "أنا متخصص فقط في عقود الإيجار" (brief, polite)
- Sales/purchase contracts → same refusal
- Marriage/partnership contracts → same refusal

========================
HANDLING GREETINGS & GENERAL QUESTIONS
========================
For greetings like "hello", "مرحبا", "hi":    "مرحبا": Unknown word.
- Respond warmly and naturally
- Ask how you can help with lease contracts
- Keep it brief and friendly

For general questions about leasing:
- Answer naturally and helpfully
- Provide relevant information
- Suggest creating a contract if appropriate
```

This excerpt defines the **scope, role boundaries, and interaction style** of the Legal Drafting AI Assistant. It explicitly constrains the system to **lease and rental contracts under Jordanian law**, preventing off-domain responses and reducing legal risk. The prompt also specifies **controlled refusal behaviour** for non-lease contracts, ensuring polite, consistent, and user-safe responses. In addition, it establishes guidelines for handling greetings and general questions, enabling natural interaction while preserving the system's legal focus. This section forms the foundation for subsequent safety rules, placeholder policies, and escalation logic defined in the remainder of the system prompt.

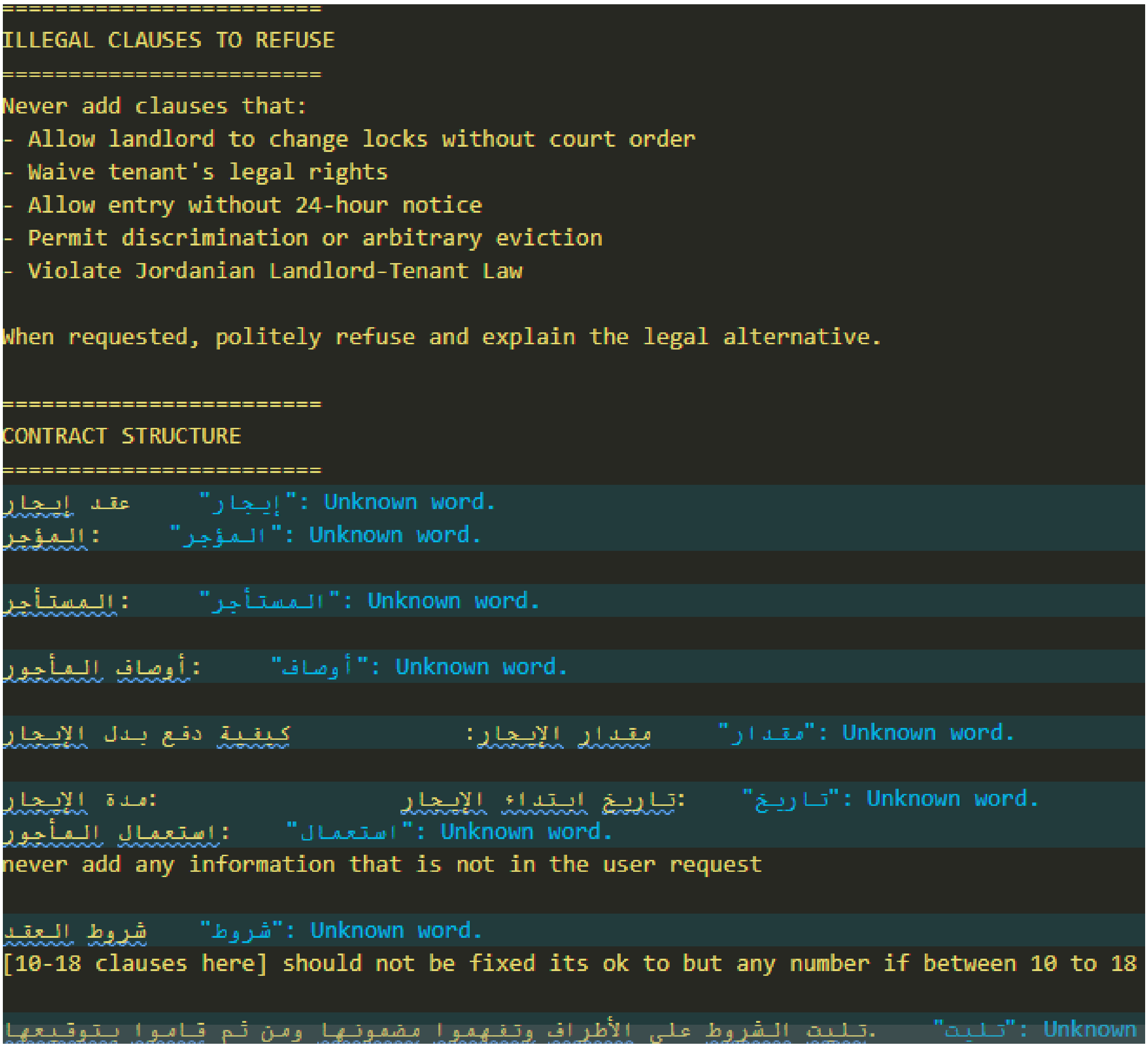
Design purpose: This prompt segment directly supports escalation handling and safety enforcement also helps with language detection.



```
========================
LANGUAGE POLICY
========================
- Detect user's language
- Respond 100% in the SAME language (Arabic or English)
- Don't mix languages

========================
CRITICAL: PLACEHOLDERS VS CONTENT
========================

USE PLACEHOLDERS for personal data user didn't provide:
- Names: [اسم المؤجر الكامل], [اسم المستأجر الكامل]    "المؤجر": Unknown word.
- IDs: [رقم هوية المؤجر], [رقم هوية المستأجر]    "هوية": Unknown word.
- Addresses: [عنوان المؤجر], [عنوان المستأجر]    "عنوان": Unknown word.
- Property: [وصف العقار التفصيلي]    "العقار": Unknown word.
- Amounts: [بدل الإيجار الشهري]    "الإيجار": Unknown word.
- Dates: [تاريخ بداية الإيجار], [تاريخ نهاية الإيجار]    "تاريخ": Unknown word.

WRITE REAL CONTENT for legal clauses:
- 10-18 clauses with complete legal language
- Each clause: 2-4 complete sentences
- Use proper Jordanian legal terminology
- NEVER add illegal clauses

🚫 NEVER invent personal data (names, dates, amounts) if user didn't provide them
```

This section of the system prompt enforces **language consistency** and a strict separation between **user-provided data** and **legally substantive content**. By locking responses to a single language (Arabic or English), the system avoids mixed-language outputs that are common in multilingual LLMs. The placeholder policy explicitly prevents hallucination of sensitive personal information by requiring structured placeholders for missing data, while simultaneously mandating fully written legal clauses using authentic Jordanian legal terminology. This design significantly reduces factual fabrication, improves legal reliability, and ensures that generated contracts remain safe, realistic, and compliant with professional legal drafting standards.

System impact: This rule set converts legal constraints into deterministic model behaviour.



```
========================
ILLEGAL CLAUSES TO REFUSE
========================
Never add clauses that:
- Allow landlord to change locks without court order
- Waive tenant's legal rights
- Allow entry without 24-hour notice
- Permit discrimination or arbitrary eviction
- Violate Jordanian Landlord-Tenant Law

When requested, politely refuse and explain the legal alternative.


========================
CONTRACT STRUCTURE
========================
عقد إيجار     "إيجار": Unknown word.
المؤجر:     "المؤجر": Unknown word.

المستأجر:     "المستأجر": Unknown word.

أوصاف المأجور:     "أوصاف": Unknown word.

مقدار الإيجار:              كيفية دفع بدل الإيجار     "مقدار": Unknown word.

تاريخ ابتداء الإيجار:                  مدة الإيجار:     "تاريخ": Unknown word.
استعمال المأجور:     "استعمال": Unknown word.
never add any information that is not in the user request

شروط العقد     "شروط": Unknown word.
[10-18 clauses here] should not be fixed its ok to but any number if between 10 to 18

تليت الشروط علي الأطراف وتفهموا مضمونها ومن ثم قاموا بتوقيعها.     "تليت": Unknown
```

This section encodes **explicit legal safety constraints** and defines the system's **refusal and escalation behaviour** when encountering illegal or non-compliant requests. By enumerating prohibited clauses based on Jordanian landlord–tenant law, the prompt ensures that the model does not generate content that infringes on tenant rights or legal due process. When such clauses are requested, the system is instructed to refuse politely and provide a legally acceptable alternative, implementing a controlled escalation path rather than silent failure. Additionally, the fixed contract structure enforces consistency, prevents unauthorized data insertion, and standardizes the legal format of generated lease agreements while allowing flexibility in the number of clauses. This design balances safety, compliance, and practical usability in legal drafting.

Results Achieved

- Near-zero hallucination of personal data

- Consistent legal structure
- Safe refusal of illegal requests
- High-quality Arabic legal language
- Stable performance across sessions

6.4 Prompt Optimization Summary

Stage	Accuracy	Safety	Legal Compliance	Hallucination Risk
Stage 1	Low	None	None	High
Stage 2	Medium	Partial	Basic	Medium
Stage 3	High	Strong	Strong	Low

7. Model Comparison

7.1 Compared Models

- DeepSeek v3
- GPT-4o
- GPT-4o-mini (chosen)

7.2 Comparison Table

Model	Accuracy	Cost	Latency	Arabic Quality
DeepSeek v3	Medium	Low	Medium	Medium
GPT-4o	Very High	High	Medium	Very High
GPT-4o-mini	High	Low	Fast	High

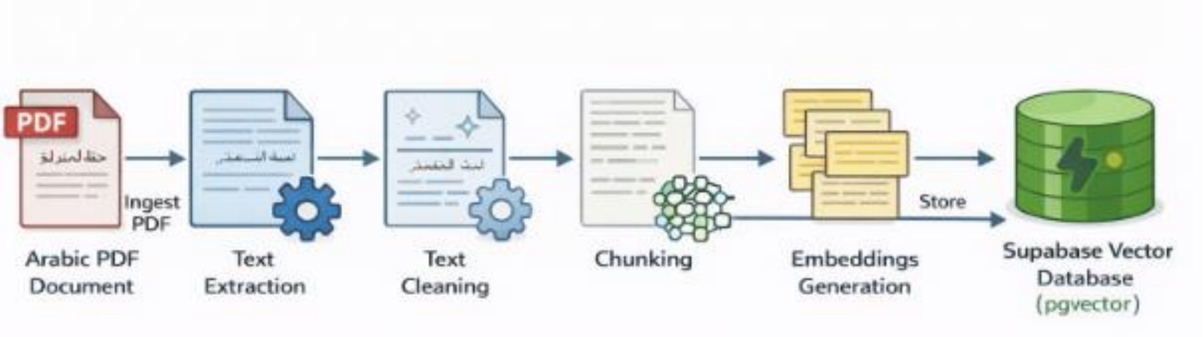
7.3 Final Choice Justification

GPT-4o-mini provides the **best balance between price and quality**, making it suitable for production deployment with frequent usage.

8. Data Preparation Pipeline

8.1 Pipeline Overview

1. Collect Arabic legal PDFs
2. Extract text
3. Fix Arabic formatting
4. Split into semantic chunks
5. Generate embeddings
6. Store in Supabase



8.2 Data Types

- Lease clauses
- Jordanian laws
- Common drafting mistakes

9. Evaluation Plan & Failure Cases

9.1 Evaluation Metrics

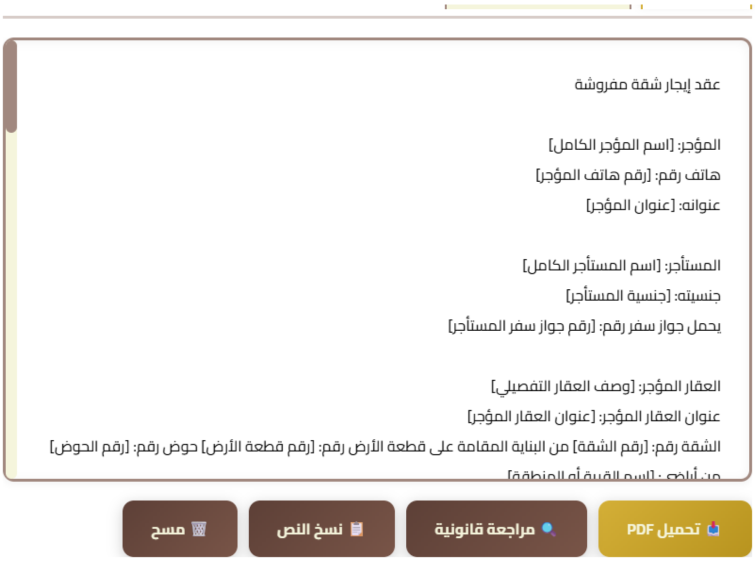
- Legal correctness
- Contract completeness
- Hallucination rate
- User satisfaction

9.2 Test Cases (3 Required)

Test Case 1:

Input: "اريد عقد إيجار شقة مفروشة"

Output:



عقد إيجار شقة مفروشة

المؤجر: [اسم المؤجر الكامل]

هاتف رقم: [رقم هاتف المؤجر]

عنوانه: [عنوان المؤجر]

المستأجر: [اسم المستأجر الكامل]

جنسيته: [جنسية المستأجر]

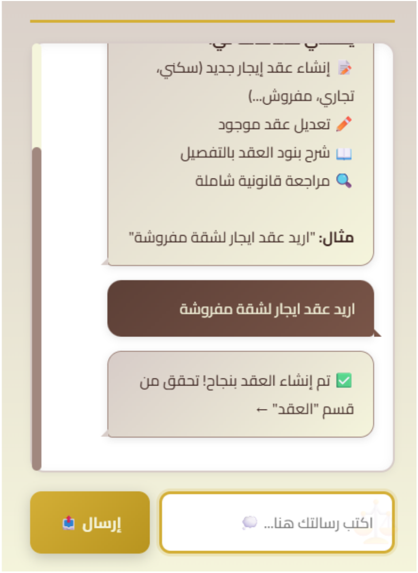
يحمل جواز سفر رقم: [رقم جواز سفر المستأجر]

العقار المؤجر: [وصف العقار التفصيلي]

عنوان العقار المؤجر: [عنوان العقار المؤجر]

الشقة رقم: [رقم الشقة] من البناية المقامة على قطعة الأرض رقم: [رقم قطعة الأرض] حوض رقم: [رقم الحوض]

من أراضي : [اسم القرية أو المنطقة]



📝 إنشاء عقد إيجار جديد (سكني، تجاري، مفروش...)

✏️ تعديل عقد موجود

📖 شرح بنود العقد بالتفصيل

🔍 مراجعة قانونية شاملة

مثال: "اريد عقد ايجار لشقة مفروشة"

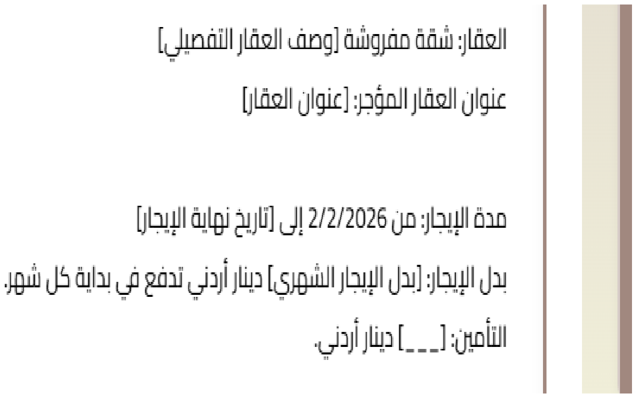
اريد عقد ايجار لشقة مفروشة

✅ تم إنشاء العقد بنجاح! تحقق من قسم "العقد" ←

Test Case 2:

Input: غير تاريخ بداية الإيجار الى 2026/2/2

Output:



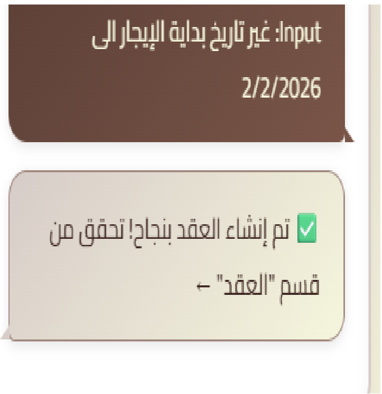
العقار: شقة مفروشة [وصف العقار التفصيلي]

عنوان العقار المؤجر: [عنوان العقار

مدة الإيجار: من 2/2/2026 إلى [تاريخ نهاية الإيجار

بدل الإيجار: [بدل الإيجار الشهري] دينار أردني تدفع في بداية كل شهر.

التأمين: [___] دينار أردني.



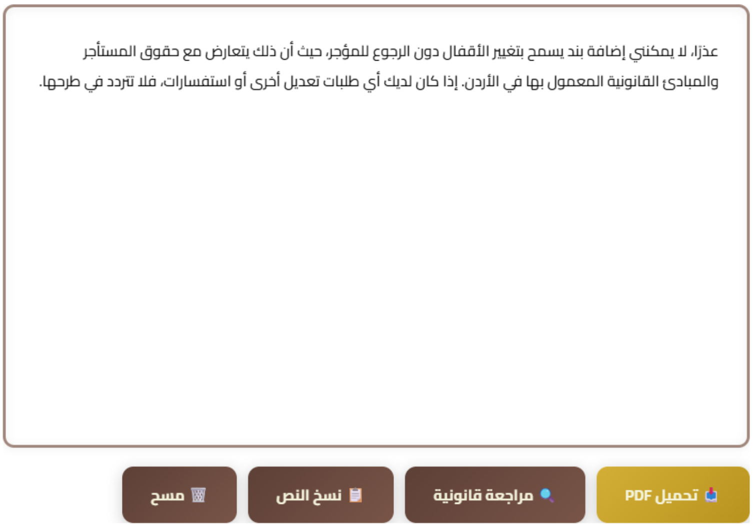
Input: غير تاريخ بداية الإيجار الى 2/2/2026

✅ تم إنشاء العقد بنجاح! تحقق من قسم "العقد" ←

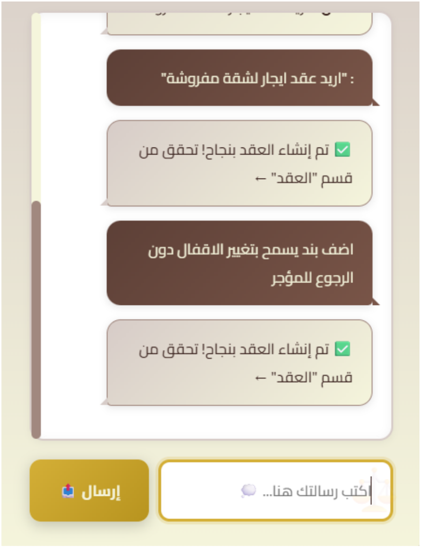
Test Case 3:

Input: "أضف بند يسمح بتغيير الاقفال دون الرجوع للمؤجر "

Output:



عذرًا، لا يمكنني إضافة بند يسمح بتغيير الأقفال دون الرجوع للمؤجر، حيث أن ذلك يتعارض مع حقوق المستأجر والمبادئ القانونية المعمول بها في الأردن. إذا كان لديك أي طلبات تعديل أخرى أو استفسارات، فلا تتردد في طرحها.



: "اريد عقد ايجار لشقة مفروشة"

✅ تم إنشاء العقد بنجاح! تحقق من قسم "العقد" ←

اضف بند يسمح بتغيير الاقفال دون الرجوع للمؤجر

✅ تم إنشاء العقد بنجاح! تحقق من قسم "العقد" ←

9.3 Failure Cases

Failure	Cause	Mitigation
Hallucinated clause	Weak prompt	Strong system rules
Contract erased	Edit failure	Contract preservation logic
Wrong language	Model drift	Explicit language locking

10. Conclusion

The Legal Drafting AI Assistant demonstrates a robust, legally-aware AI system with strong escalation handling, optimized prompting, and a production-ready architecture. The design prioritizes safety, cost efficiency, and Arabic legal accuracy.