Context-Aware Health RAG

1. فكرة المشروع

Context-Aware Health RAG هو نظام ذكاء اصطناعي قائم على تقنية **Retrieval-Augmented Generation (RAG)**. هدفه مساعدة المستخدم في الحصول على المعلومات الصحية المرتبطة **بالموقف أو السياق الحالي** الذي يمر به.

الفكرة ليست بناء Chatbot يستطيع الإجابة عن "ما هو مرض X؟" فقط، وإنما بناء نظام يستطيع فهم:

> **ما حالة المستخدم؟ وما الذي تغير الآن؟ وما المعلومات الطبية المرتبطة بهذا التغيير؟**

المستخدم قد يكون بالفعل على علم بمرضه أو علاجه، لكن تظهر أمامه مواقف جديدة مثل:

- بدأ دواء جديد.
- ظهر عرض جديد.
- تغيرت أعراضه.
- تعرض لظروف بيئية مختلفة.
- تغيرت درجة الحرارة أو الرطوبة.
- يريد معرفة معلومات عن خيارات طبيعية أو تقليدية مرتبطة بحالته.

النظام يقوم بفهم هذا السياق، ثم يحدد مصادر المعرفة المناسبة، ويسترجع منها المعلومات ذات الصلة، ثم يولد إجابة **Grounded** بالمصادر بدل الاعتماد على معرفة النموذج وحدها.

---

2. المشكلة التي نحاول حلها

المعلومات الصحية موجودة بالفعل، لكن المشكلة أن المستخدم يحتاج غالبًا إلى **تجميع وربط المعلومات من عدة مصادر بنفسه.**

مثلاً، شخص لديه حالة جلدية ويستخدم دواءً معينًا، ثم تظهر لديه أعراض جديدة.

لكي يفهم الموقف، قد يحتاج إلى البحث عن:

1. معلومات عن الدواء.
2. المادة الفعالة.
3. الآثار الجانبية.
4. التحذيرات المتعلقة بحالته.
5. المعلومات الطبية المتعلقة بالعرض الجديد.
6. وربما العوامل البيئية المرتبطة بالحالة.

المشكلة أن هذه المعلومات تكون موزعة بين مصادر مختلفة.

المشروع يحاول حل هذه المشكلة من خلال:

> **ربط السياق الحالي للمستخدم بالمعلومة الطبية المناسبة من عدة مصادر، ثم تقديمها في إجابة واحدة موثقة.**

---

3. نطاق النسخة الأولى — MVP

بسبب أن مدة الهاكاثون قصيرة، لن نحاول دعم جميع الأمراض.

سنبدأ بمجال طبي محدد:

Dermatology & Skin-Related Conditions

وسندعم في البداية **3 حالات فقط:**

1. Eczema / Atopic Dermatitis

2. Psoriasis

3. Urticaria / Hives

اختيار هذه الحالات يسمح لنا بإظهار العلاقة بين:

Condition → Symptoms → Triggers → Environment → Context

وفي نفس الوقت يظل حجم الـ Knowledge Base صغيرًا وقابلًا للإدارة.

يمكن تغيير الحالات الثلاث بناءً على جودة البيانات التي نستطيع جمعها، لكن لا نتجاوز 3–5 حالات في الـ MVP.

---

4. ما الذي سيستطيع المستخدم فعله؟

النسخة الأولى ستدعم أربعة أنواع أساسية من الـ Context.

A. Medication Context

المستخدم يسأل عن دوائه في سياق حالته.

مثال:

"أنا باخد الدواء X وعندي Diabetes، إيه التحذير الموجود في المعلومات المتاحة عن الدواء؟"

النظام يستخدم Drug Knowledge Base ويبحث تحديدًا عن المعلومات المتعلقة بالدواء والحالة.

B. Symptom Context

المستخدم بدأ دواءً أو لديه حالة معينة وظهر عرض جديد.

مثال:

"أنا بدأت الدواء X وظهر عندي itching، هل العرض ده مذكور ضمن الآثار الجانبية أو التحذيرات الموجودة عن الدواء؟"

النظام يبحث عن العلاقة بين:

Medication + Symptom

ولا يفترض أن الدواء هو سبب العرض.

C. Environmental Context

المستخدم لديه حالة صحية والظروف المحيطة به تغيرت.

مثال:

> "أنا عندي Eczema والجو النهارده حر جدًا والرطوبة عالية، إيه المعلومات الطبية المرتبطة بالظروف دي؟"

النظام يحصل على بيانات الطقس من Open-Meteo، ثم يستخدم الـ Medical RAG للبحث عن المعلومات المتعلقة بالعوامل البيئية والحالة.

السفر هنا مجرد مثال.

يمكن للمستخدم أن يكون في منزله ويقول:

> "الجو عندي النهارده شديد الحرارة."

D. Natural / Traditional Knowledge

المستخدم يريد معرفة ما إذا كانت هناك معلومات عن أعشاب أو ممارسات تقليدية مرتبطة بحالته.

مثال:

> "هل توجد أعشاب مرتبطة تقليديًا بالحالة دي؟"

النظام يعرض المعلومات من Natural Medicine Knowledge Base، مع توضيح:

- الاستخدام التقليدي.
- مستوى الدليل إن وجد.
- معلومات السلامة.
- التداخلات المحتملة مع الأدوية.
- المصدر.

ولا يقدم العشبة باعتبارها بديلًا للدواء الموصوف.

---

5. مصادر البيانات

النسخة الأولى ستعتمد على أربعة مصادر أساسية.

Drug Knowledge Base 5.1

لدينا بالفعل بيانات الأدوية.

البيانات تتضمن مثلًا:

- Commercial name
- Arabic name
- Active ingredient
- Manufacturer
- Drug class
- Route
- Uses
- Warnings
- Pregnancy warnings
- Lactation warnings
- Kidney warnings
- Liver warnings
- Heart warnings
- Diabetes warnings
- Hypertension warnings

هذه البيانات ستكون أساس **Drug RAG**.

Medical / Condition Knowledge Base 5.2

سننشئ Knowledge Base صغيرة للحالات الثلاث المختارة.

لكل Condition نحتاج:

```
Condition
├── Description
├── Symptoms
├── Causes / Associated Factors
├── Common Triggers
├── Environmental Factors
├── Precautions
└── Source
```

لن نحاول جمع كل المعلومات الطبية عن المرض.

سنركز فقط على المعلومات التي تخدم الـ Context-Aware Use Cases.

مثلاً:

```
Eczema

Symptoms:
- Itching
- Dry skin
- Rash

Triggers:
- Irritants
- Stress
- Environmental changes

Environmental Factors:
- Temperature
- Humidity
- Dryness

Precautions:
- ...

Source:
- Medical source
```

Weather / Environmental Data 5.3

سنستخدم Open-Meteo كمصدر Dynamic Data.

في الـ MVP سنركز على:

- Temperature
- Humidity
- UV
- Precipitation
- Wind

هذه البيانات لن تكون جزءًا أساسيًا من الـ Vector Database.

هي **Dynamic Context** يدخل إلى الـ RAG وقت السؤال.

مثلاً:

```
Location = Cairo

Temperature = 38°C
```

```
Humidity = 25%
UV = High
```

ثم النظام يبحث في الـ Medical Knowledge Base:

ما المعلومات الطبية المتعلقة بهذه الظروف بالنسبة للحالة التي ذكرها المستخدم؟

5.4 Natural / Traditional Medicine Knowledge Base

هذه ستكون صغيرة في الـ MVP.

نستهدف تقريبًا:

10–20 herbs فقط.

لكل Herb:

```
Herb
├── Common Name
├── Scientific Name
├── Traditional Uses
├── Evidence Level
├── Safety Information
├── Potential Interactions
└── Source
```

الهدف ليس تقديم "بدائل طبيعية للأدوية"، وإنما تقديم **معلومات موثوقة عن الاستخدامات الطبيعية/التقليدية المرتبطة بالسياق.**

---

6. كيف يعمل الـ RAG؟

المشروع لا يستخدم RAG بالطريقة البسيطة:

```
Question
↓
Vector Search
↓
LLM
```

بل:

```
User Query
↓
Context Extraction
↓
Intent Detection
↓
Query Routing
↓
Relevant Knowledge Retrieval
↓
Reranking
↓
Context Fusion
↓
LLM
↓
Grounded Answer + Sources
```

---

7. Context Extraction

النظام يحاول استخراج العناصر المهمة من سؤال المستخدم.

مثلاً:

"أنا عندي Eczema وباخد X ومن يومين ظهرت عندي حكة."

النظام يفهم:

```
Condition = Eczema

Medication = X

Symptom = Itching

Time Context = 2 days

Intent = Medication + Symptom
```

وهذا يساعد في توجيه عملية الـ Retrieval.

---

8. Query Routing

بعد فهم السؤال، النظام يقرر أي Knowledge Base يحتاجها.

مثلاً:

سؤال عن دواء:

```
Drug RAG
```

دواء + عرض:

```
Drug RAG
+
Medical / Symptom Retrieval
```

حالة + طقس:

```
Medical RAG
+
Weather API
```

حالة + عشبة:

```
Medical RAG
+
Natural Medicine RAG
```

دواء + عشبة:

```
Drug RAG
+
Natural Medicine RAG
+
Interaction Retrieval
```

وهذه من أهم نقاط تميز المشروع.

---

9. مثال كامل

User Query

"أنا عندي Eczema والجو النهارده حر جدًا والرطوبة قليلة، هل ده ممكن يكون مرتبط بحالتي؟"

النظام يقوم بالآتي:

Step 1

يستخرج:

```
Condition = Eczema
Context = Heat + Low Humidity
```

Step 2

يحصل على الطقس الحالي من Open-Meteo.

Step 3

يبحث في Medical Knowledge Base عن:

```
Eczema
+
Heat
+
Dryness
+
Humidity
+
Environmental factors
```

Step 4

يقوم بعمل Reranking للنتائج.

Step 5

يجمع:

```
Current Weather
+
Relevant Medical Evidence
```

Step 6

الـ LLM يولد إجابة مبنية على المعلومات المسترجعة.

النتيجة

الإجابة توضح:

- الظروف الحالية.
- ما إذا كانت هذه العوامل مرتبطة بالحالة وفقًا للمصادر.

- المعلومات ذات الصلة.
- المصدر.

ولا تقول:

> "الطقس هو سبب الأعراض."

---

10. مثال آخر

User Query

> "أنا بدأت الدواء X من يومين وظهر عندي itching."

النظام:

```
Medication = X
Symptom = Itching
```

ثم يبحث في Drug Knowledge Base عن:

```
Drug X
↓
Side Effects / Warnings
↓
Itching
```

ثم يعطي إجابة grounded.

إذا لم يجد دليلًا في البيانات:

> "لم أجد في المصادر المتاحة لدي معلومات كافية تربط هذا العرض بالدواء."

وهذا أفضل من أن يخمن الـ LLM.

---

11. لماذا RAG مهم هنا؟

لأن النظام لا يفترض أن الـ LLM يعرف المعلومة الصحيحة.

نحن نريد:

```
User Context
      ↓
Retrieve Evidence
```

```
        ↓
Answer from Evidence
```

وبالتالي نستطيع إظهار:

Source

المعلومة جاءت من أين؟

Evidence

ما الجزء الذي بُنيت عليه الإجابة؟

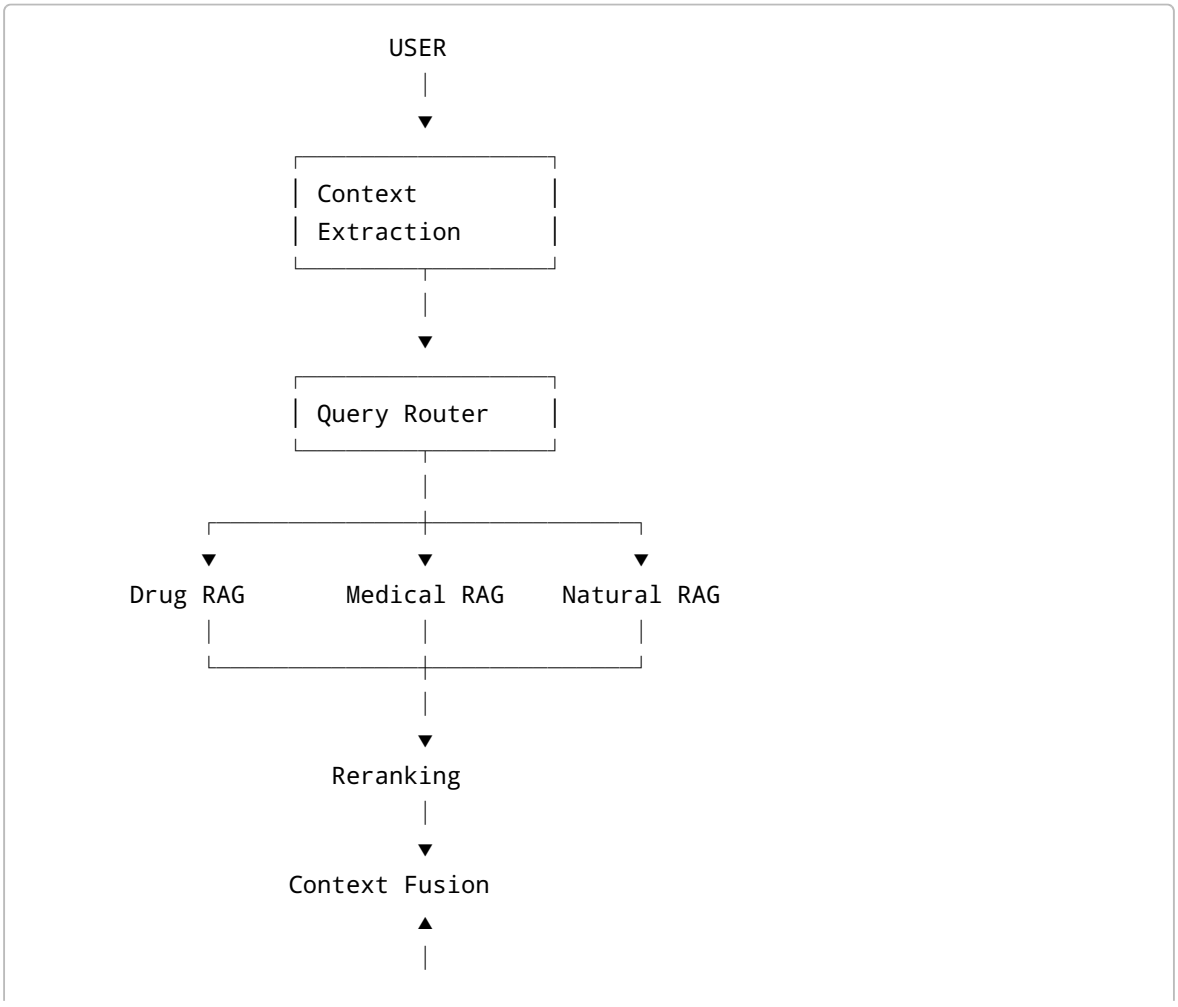
Grounding

هل الإجابة موجودة بالفعل في الـ Retrieved Context؟

وهذه نقاط مهمة جدًا في المشروع.

---

12. الـ Architecture المقترحة



```
Weather API
      |
      ▼
     LLM
      |
      ▼
Grounded Answer
      +
   Sources
```

---

13. ما الذي لن ندخله في الـ MVP؟

للحفاظ على إمكانية إنهاء المشروع في يومين، لن ندخل:

- جميع الأمراض.
- تشخيص المرض.
- اقتراح جرعات.
- تغيير أو إيقاف دواء.
- استبدال دواء بعشبة.
- تحليل صور جلدية.
- Patient history معقد.
- Air Quality في البداية.
- آلاف الأعشاب.
- Knowledge Graph معقد.
- Agentic workflow معقد.

هذه كلها مراحل مستقبلية.

---

14. حجم الـ MVP المقترح

Component	حجم الـ MVP
Medical Conditions	3
Drug Dataset	البيانات الموجودة
Herbs	20–10
Weather Variables	5 تقريبًا
Medical Sources	2–1 مصادر أساسية
RAG	Hybrid + Semantic
Reranker	1
Main Use Cases	4
UI	Chat + Sources + Context

Component	حجم الـ MVP
Evaluation Questions	15–20

---

15. لماذا هذا الـ Scope مناسب؟

لأننا نريد إثبات **الفكرة الأساسية** وليس كمية البيانات.

لو اللجنة سألت:

> "هل النظام يستطيع دعم أمراض أخرى؟"

الإجابة:

> نعم. الـ Architecture ليست مرتبطة بالأمراض الثلاثة. نحن فقط قمنا بتحديد نطاق الـ Knowledge Base في الـ MVP لضمان جودة البيانات والـ Retrieval خلال وقت الهاكاثون. إضافة مرض جديد لاحقًا تعني إضافة Knowledge جديدة إلى النظام وليس إعادة تصميم الـ RAG.

---

16. الـ MVP في جملة واحدة

> **نظام RAG طبي يركز مبدئيًا على 3 حالات جلدية، ويستطيع فهم سياق المستخدم مثل دواء جديد، عرض جديد، أو ظروف بيئية متغيرة، ثم يسترجع المعلومات الطبية ذات الصلة من مصادر موثوقة ويقدم إجابة grounded بالمصادر، مع إمكانية دمج معلومات طبيعية/تقليدية بشكل آمن.**

الهدف في اليومين: لا نحاول بناء "طبيب AI"، بل نبني **نواة RAG قوية وقابلة للتوسع** تثبت أن النظام يستطيع ربط **Health Condition + Medication + Symptoms + Environment + Knowledge Sources** بطريقة ذكية ومبنية على الأدلة.