

(1 هدف المشروع) (Scope)

Hospital System – Desktop GUI :يغطي:

تسجيل المرضى (CRUD)

إدارة المواعيد والأطباء

الدخول الداخلي (Admissions) وغرف المستشفى

الملف الطبي (Diagnoses, Vitals, Lab tests)

الروشتات والصيدلية (Prescriptions & Inventory)

الفواتير والمدفوعات (Billing)

المستخدمين والصلاحيات (Admin/Doctor/Nurse/Reception/Pharmacist/Lab)

السجلات (Audit Log)

(2 معمارية نظيفة)

Presentation (GUI Layer): ، Validation ، جداول ، فورمات ،

Service/Domain Layer: Business rules (حجز موعد ، حساب فاتورة. ...)

Data Access Layer: Repositories (باستخدام SQLAlchemy لا كويريز في الـ GUI)

(3 تصميم قاعدة البيانات ERD) مختصر)

الكيانات والعلاقات الأساسية:

users(user_id, username, password_hash, role_id, active)

roles(role_id, name) (Admin/Doctor/Nurse/Reception/Lab/Pharmacy)

patients(patient_id, national_id, first_name, last_name, dob, gender, phone, address, emergency_contact)

staff(staff_id, user_id, first_name, last_name, phone, department, job_title)

doctors(doctor_id, staff_id, specialization)

```

rooms(room_id, ward, room_number, bed_count, status)

admissions(admission_id, patient_id, room_id, admitted_at, discharged_at, notes)

appointments(appointment_id, patient_id, doctor_id, scheduled_at, status, reason)

medical_records(record_id, patient_id, doctor_id, visit_date, diagnosis, notes)

vitals(vital_id, record_id, temp, bp_sys, bp_dia, pulse, spo2)

lab_tests(test_id, record_id, test_type, requested_by, result, requested_at, resulted_at,
status)

prescriptions(rx_id, record_id, prescribed_by, created_at, notes)

rx_items(item_id, rx_id, drug_id, dose, frequency, duration_days)

drugs(drug_id, name, form, strength, stock_qty, reorder_level, unit_price)

invoices(invoice_id, patient_id, admission_id?, total, paid, created_at, status)

invoice_items(item_id, invoice_id, description, qty, unit_price)

audit_logs(log_id, user_id, action, entity, entity_id, timestamp, details)

```

علاقات مهمة:

```

patients 1—* appointments, admissions, medical_records, invoices
medical_records 1—1..* vitals, lab_tests, prescriptions
prescriptions 1—* rx_items —> drugs
invoices 1—* invoice_items
اختياري users —* audit_logs | users 1—1 staff —> doctors (
rooms 1—* admissions

```

—

(4 شاشات الـ GUI الحد الأدنى الرابع)

A) الدخول والصلاحيات

Login + Role-based menu. أمان وتخصيص واجبات كل دور.

B) لوحة رئيسية (Dashboard)

Cards: عدد المرضى اليوم، المواعيد القادمة، تنبيهات مخزون الأدوية تحت reorder_level فواتير غير مدفوعة. ليه؟ نظرة فورية للحالة التشغيلية.

C) المرضى

بحث سريع، إضافة/تعديل/حذف، تفصيل Patient Profile (زيارات، فواتير، ملفات). ليه؟ أسرع نقطة تعامل عند الاستقبال.

D) المواعيد

Calendar/جدول، فلترة بالـ Doctor/Date تغيير حالة (Scheduled/Completed/No-Show). ليه؟ تقليل تضارب وتنظيم وقت الطبيب.

E) الدخول الداخلي (Admissions & Rooms)

تخصيص سرير، نقل غرفة. discharge، ليه؟ تشغيل غرف المستشفى.

F) الملف الطبي (EMR)

زيارة جديدة: تشخيص + Vitals + طلب تحاليل + روشتة.

عرض تاريخ المريض الطبي. ليه؟ لب النظام الطبي.

G) المعمل (Lab)

Queue بالتحاليل المطلوبة، إدخال النتائج، وسم الحالة. ليه؟ دورة كاملة لطلبات التحاليل.

H) الصيدلية (Pharmacy & Inventory)

صرف روشتة، خصم من المخزون، تنبيه المخزون المنخفض.

إدارة الأدوية. (CRUD) ليه؟ ربط سريري-مالي-تشغيلي واضح.

I) الفواتير (Billing)

إنشاء فاتورة تلقائيًا من الخدمات/الروشتة.

دفع كلي/جزئي، إيصال PDF. ليه؟ فلوس = تقييم عالي في المشاريع.

J) الإدارة (Admin)

مستخدمين وصلاحيات، باك أب/ريستور Logs،ليه؟ جاهزية إنتاجية وأمان.

(5)الربط بين الـ GUI والداتا بيز

ملف db.py: إنشاء (Session + Models (SQLAlchemy).

repositories/: classes Entity (CRUD + استعلامات مركبة).

services/: منطق الأعمال (...), discharge_patient(), book_appointment()

ui: فورمات PySide6. كل فورم ينادي Services فقط (مش SQL مباشر).

events & validation: قبل الحفظ، تحقق من الحقول (رقم قومي، تليفون، تواريخ).

transactions: استخدام session.begin() للعمليات المركبة (صرف روشتة + خصم مخزون + إنشاء فاتورة).

migrations: Alembic لو وقتكم يسمح؛ وإلا SQL سكربت أولي. seed +

(6)جودة الكود والتسليم

Structure:

```
/hospital
  /ui
  /services
  /repositories
  /models
  /tests
  /assets
  config.py
  main.py
  requirements.txt
  README.md
docs/ (ERD, Screens, SQL schema, Demo script)
```

Logging: إلى file + audit_logs في DB.

Config: .env للاتصال بالداتا بيز.

Testing: وحدات لخدمات أساسية (حجز موعد، إصدار فاتورة...).

Seed Data: 15 مريض، 6 دكاترة، 30 موعد، 10 غرف، 20 دواء، 12 تحليل، 8 فواتير. فيه؟ ديمو سلس Marker ، يشوف سيناريوهات حقيقية.

(7 تقسيم الفريق (6 أفراد)

عضو 1 – قائد المشروع Backend Lead/

إعداد المشروع. audit، users/roles، Alembic/SQL schema، models، config،

يكتب services الأساسية ويعمل Code Review. فيه؟ يقلل المخاطر ويوحد المعايير.

عضو 2 Patients & Appointments –

شاشات المرضى + المواعيد + البحث والفلتر.

Service: booking rules (تعارض الوقت، مدة الكشف). فيه؟ مدخل النظام.

عضو 3 EMR & Vitals –

شاشة الملف الطبي. + history + vitals

Service: إنشاء زيارة، ربط طبيب/مريض. فيه؟ قلب الاستخدام السريري.

عضو 4 Lab Module –

طلب/تسجيل نتائج التحاليل + حالة الطلب.

Reports بسيطة (تحاليل اليوم/المعلقة). فيه؟ يظهر دورة عمل متكاملة.

عضو 5 Pharmacy & Inventory –

إدارة الأدوية + صرف الروشتات + خصم المخزون + تنبيهات إعادة الطلب. فيه؟ تكامل سريري-تشغيلي.

عضو 6 Billing & Admin & Packaging –

إنشاء/سداد الفواتير + طباعة إيصال PDF.

إدارة المستخدمين + cfg/باك أب/ريستور + إعداد README و Setup/Demo. فيه؟ فلوس + تسليم محترف.

>ملاحظة: الكل يكتب Tests بسيطة لموديوله + يلتزم Naming/Style واحد.

—

(8تايملاين تنفيذ (مثالي لـ 5 أيام مكثفة)

اليوم 1

ERD نهائي + SQL schema + إنشاء المشروع وهيكل المجلدات.

Setup DB + Models + Seed مبدئي.

سكيتش سريع لكل شاشة (بالقلم) لتحديد الحقول والتدفق.ليه؟ تثبيت الأساس قبل GUI.

اليوم 2

بناء Repositories و Services الأساسية.(Patients/Appointments/Users)

بدء GUI: Login + Dashboard + Patients List.ليه؟ توصيل أول شاشة بالـ DB مبكرًا.

اليوم 3

استكمال GUI: Appointments + EMR + Vitals.

بدء Lab + Pharmacy Services.

إضافة Audit logging.ليه؟ تغطية المسار السريري بالكامل.

اليوم 4

إنهاء Lab + Pharmacy GUI.

Billing + PDF + Inventory Alerts.

Roles/Permissions على القوائم والأزرار.ليه؟ اكتمال التشغيل + أمان.

اليوم 5

اختبارات وظيفية شاملة + إصلاح Bugs.

Seed نهائي جميل للديمو Script + عرض.

كتابة + README لقطات شاشة + فيديو قصير (إن أمكن). ليه؟ التسليم أهم من كماليات لاحقة.

9) قواعد GUI عملية (من غير "UI/UX" تقيل)

نمط واحد ثابت Font: موحد، تباعد منتظم، عناوين واضحة.

اختصارات: Enter=Save، Esc=Cancel، Ctrl+F=Search.

Validation فوري: مثل رقم قومي 14 رقم، تاريخ حجز بعد الآن، مخزون لا يصبح سالب.

جداول قابلة للبحث/الفرز: Top filter + pagination.

حالة النظام Loading spinner، Toast: رسائل نجاح/فشل. ليه؟ سرعة ووضوح وتقليل أخطاء.

10) اختبارات سيناريو (جاهزة للديمو)

1. استقبال: إضافة مريض → حجز موعد مع دكتور كاردبيولوجي الساعة → 10:30 يظهر في Calendar.

2. كشف: فتح الموعد → إنشاء زيارة → تشخيص. Vitals +

3. تحليل: طلب CBC من المعمل Lab → يسجل النتيجة → تظهر في ملف المريض.

4. روثنة: الطبيب يكتب روثنة → الصيدلية تصرف → ينقص المخزون تلقائي.

5. فاتورة: إنشاء فاتورة للخدمات (كشف + تحليل + أدوية → (سداد جزئي ثم كامل → طباعة إيصال. PDF

6. دخول داخلي: حجز غرفة → نقل → discharge → فاتورة إقامة.

7. أمان: تسجيل دخول User بصلاحيات Reception لا يقدر يفتح Billing أو Admin.

8. Audit: إظهار سجل عملية حذف/تعديل معينة.

(11) تسليم احترافي (Checklist)

Repo ✓ منظم + README فيه: المتطلبات، خطوات التشغيل، بيانات دخول لكل Role، صور للشاشات.

ERD (✓ صورة SQL schema + Draw.io/مرفق).

Seed ✓ جاهز لتشغيل سيناريو الديمو.

Video ✓ ديمو 2-4 دقائق (اختياري لكنه قوي).

Backup/Restore Script (✓ اختياري).
ليه؟ Marker يلاقي كل حاجة بسرعة ويفهم قيمة شغلهم.

—

(12) متطلبات نجاح وتميز (Rubric) داخلي

تكامل الوحدات بدون كراش.

سرعة البحث والتنقل.

صلاحيات فعالة (تجربة دخول بأدوار مختلفة).

عدم فقدان بيانات. (Transactions + Validation)

توثيق واضح Seed + غني. ليه؟ دي غالبًا نقاط التصحيح.

—

(13) مخاطر وكيف نتفادها

تأخير GUI: ابدأوا بأبسط فورمات (CRUD) وبعدين حسّنوا.

تعقيد Billing: ابّنوا abstraction: Invoice + Items + Calculator service.

تعارض مواعيد: تحقق على مستوى (Service + Unique constraint (doctor_id, scheduled_at).

مخزون سالب Trigger/Service: يمنع. $qty < 0$.
ليه؟ تقليل Bugs فائتة في الديمو.

—

Bonus (14) يفرّق في الدرجات

تقرير بسيط (PDF/Excel) بعدد المرضى/الزيارات خلال فترة.

Export/Import CSV للأدوية والمرضى.

إشعار تنبيهي داخل النظام. (Low stock / overdue bills)
ليه؟ value ملموسة بدون مجهود ضخم.