تطوير الواجهة الخلفية (Backend Developer)

هدف المهمة: تطوير الواجهة الخلفية لنظام إدارة العقارات (PMS)، بما في ذلك إنشاء الخوادم، قواعد البيانات، واجهات البرمجة (APIs)، وضمان التكامل السلس مع واجهة المستخدم.

المهام الرئيسية:

1. تحليل المتطلبات التقنية:

مراجعة متطلبات النظام التي تم جمعها من قبل محلل الأعمال.

تحديد الميزات الأساسية التي يجب تطويرها في الجزء الخلفي مثل إدارة العقارات، تحصيل الإيجارات، إدارة الصيانة، والتقارير.

تحديد التقنيات التي سيتم استخدامها (مثل Node.js، Python/Django، Ruby on Rails، PHP).

2. إعداد بيئة التطوير:

تحديد الأدوات اللازمة لبناء الخادم مثل قاعدة البيانات، الخوادم (مثل Express.js، Flask، أو Django)، وأنظمة التحكم في الإصدارات (مثل Git).

إعداد بيئة محلية للتطوير واختبار الأكواد.

تحديد أطر العمل (Frameworks) التي ستُستخدم في تطوير واجهة البرمجة (APIs).

3. تصميم قاعدة البيانات:

تصميم قاعدة البيانات التي ستحتفظ بكافة البيانات مثل تفاصيل العقارات، المستأجرين، المدفوعات، وطلبات الصيانة.

اختيار نوع قاعدة البيانات المناسبة (SQL مثل MySQL أو PostgreSQL، أو NoSQL مثل MongoDB).

إعداد الجداول والعلاقات بين البيانات لضمان تكامل النظام.

تحسين قاعدة البيانات لأداء سريع واستعلامات فعّالة.

4. إنشاء واجهات البرمجة (APIs):

تصميم وبناء APIs للتفاعل بين واجهة المستخدم والبيانات المخزنة في قاعدة البيانات.

تطوير واجهات APIs RESTful أو GraphQL لتمكين الوصول إلى البيانات بشكل آمن وفعّال.

ضمان أن كل API يعيد البيانات المطلوبة بشكل دقيق وضمن الوقت المحدد.

5. التكامل مع الأنظمة الخارجية:

بوابات الدفع: تكامل النظام مع بوابات الدفع عبر الإنترنت مثل PayPal أو Stripe لتسهيل عمليات الدفع والإيصالات.

إرسال إشعارات: تكامل النظام مع خدمات الإشعارات مثل البريد الإلكتروني أو الرسائل النصية لإرسال إشعارات الدفع، تواريخ الاستحقاق، أو تحديثات حالة الصيانة.

6. إدارة الأمان:

التحقق من هوية المستخدم (Authentication): استخدام تقنيات مثل JWT (JSON Web Tokens) أو OAuth لضمان أمان حسابات المستخدمين.

إذن الوصول (Authorization): ضمان أن كل مستخدم يمكنه الوصول فقط إلى البيانات التي له الحق في رؤيتها.

حماية البيانات: تطبيق تقنيات مثل التشفير (Encryption) لحماية البيانات الحساسة مثل معلومات الدفع.

7. اختبار النظام:

إجراء اختبارات وحدات (Unit Testing) للتحقق من أن كل جزء من الكود يعمل كما هو متوقع.

اختبار التكامل (Integration Testing) للتحقق من التفاعل بين مكونات النظام المختلفة.

إجراء اختبارات الأداء لضمان أن النظام يمكنه التعامل مع العديد من الطلبات في وقت واحد دون التأثير على الأداء.

8. إعداد المراقبة والصيانة:

إعداد أدوات مراقبة الأداء مثل New Relic أو Prometheus لمراقبة صحة النظام وأداء الخوادم.

إعداد نسخ احتياطية منتظمة للبيانات لضمان استعادتها في حالة حدوث أي فشل في النظام.

مراقبة أخطاء النظام وتقديم إصلاحات عند الحاجة.

النتائج المتوقعة:

واجهة خلفية متكاملة: بناء خادم قادر على التعامل مع جميع الطلبات من الواجهة الأمامية بشكل فعّال.

API جاهز للاستخدام: واجهات API تسمح بالوصول إلى البيانات بشكل آمن وسريع.

قاعدة بيانات قوية: قاعدة بيانات مرنة تدير كل من البيانات المالية، العقارية، والمستخدمين بطريقة منظمة وآمنة.

تكامل مع الأنظمة الأخرى: مثل بوابات الدفع وإرسال الإشعارات.

المدة الزمنية المتوقعة:

5-7 أسابيع لتطوير الواجهة الخلفية، بما في ذلك الاختبارات والتكامل مع الأنظمة الأخرى.

توجيهات إضافية:

تأكد من أن النظام قابل للتوسيع والتعديل بسهولة في المستقبل.

التعاون المستمر مع المطورين الأماميين لضمان أن الواجهة الخلفية تدعم جميع وظائف واجهة المستخدم.

مراجعة الأمان بشكل دوري لضمان حماية البيانات الحساسة.