مقترح هندسة وتطوير "منظومة Service Coin"

أضع بين يديكم هذا الشرح الفني وخطة العمل التفصيلية لتنفيذ "منظومة Service Coin".
المشروع ليس مجرد تطبيق عادي، بل هو بنية تحتية متكاملة تدمج بين تقنيات الويب، تطبيقات الهواتف الذكية، وتكنولوجيا البلوك تشين (Blockchain).

أولاً: الفكرة العامة للمشروع (ببساطة)

تهدف المنظومة إلى إنشاء منصة مؤسسية لتبادل الخدمات تعتمد على تقنية البلوك تشين، وتتوافق بالكامل مع أحكام الشريعة الإسلامية. الفكرة الأساسية هي أننا نبتعد تماماً عن فكرة "العملات المشفرة للمضاربة"، ونقوم بتحويل "الخدمة الملموسة" إلى "عملة أو تذكرة رقمية" بحيث تُستمد القيمة الاقتصادية من الخدمات الحقيقية وليس من الفوائد الربوية.

مثال عملي مبسط لكيفية عمل النظام:

لنفترض أن مستشفى معتمداً في المنصة لديه 100 موعد متاح لإجراء فحوصات

- **الإضافة في النظام:** يقوم المستشفى بإضافة هذه المواعيد من خلال لوحة التحكم الخاصة به.
- **الإصدار :** يقوم نظامنا الخلفي بالتواصل مع شبكة البلوك تشين برمجياً لإصدار 100 عملة (Service Coin) فقط. كل عملة تمثل موعداً واحداً، ولا يمكن للنظام طباعة أي عملة إضافية دون وجود خدمة حقيقية تقابلها.
- **الاستهلاك والحرق :** عندما يحجز المريض الموعد ويذهب للمستشفى للحصول على الخدمة، يقوم النظام تلقائياً بـ "حرق" أو تدمير هذه العملة وإزالتها من التداول نهائياً.

ثانياً: لماذا البلوك تشين؟ وما هي التكلفة التشغيلية (Gas Fees)؟

قد يتساءل البعض: لماذا لا نستخدم قواعد بيانات عادية فقط؟ الجواب هو الشفافية المطلقة. في قواعد البيانات العادية، يمكن لأي مدير نظام تعديل الأرقام. أما البلوك تشين، فهو دفتر سجلات يضمن عدم قدرة

أي مؤسسة على تعديل خدماتها أو التلاعب بالأرصدة بعد توثيقها.

هل ندفع أموالاً حقيقية مقابل كل عملة نصدرها؟

الإجابة هي: لا. نحن لا نشتري العملات من الشبكة. النظام يعمل وفق مبدأ "رسوم التشغيل" (Gas Fees). عندما نرسل أمراً برمجياً لشبكة البلوك تشين لإصدار 1000 عملة، فإننا ندفع أجزاء من السنتات فقط (رسوم زهيدة جداً) كأجرة مقابل الطاقة الحاسوبية التي استهلكتها الشبكة لتنفيذ هذا الكود المبرمج، تماماً كما ندفع تكاليف استضافة الخوادم السحابية للموقع.

ثالثاً: خطة العمل الهندسية والتنفيذية

لضمان جودة النظام، سيتم تصميم المشروع برمجياً من الصفر (End-to-End) بدون الاعتماد على أي قوالب جاهزة، وفق المراحل التالية:

المرحلة الأولى: التحليل وهندسة قواعد البيانات

- **التصميم المخصص (Custom Schema):** تخطيط وبناء جداول قواعد البيانات المركزية (MySQL) يدوياً لتشمل إدارة المؤسسات، المستخدمين، وسجلات المعاملات.
- **الامتثال:** التنسيق التنظيمي والشرعي لضمان مطابقة المنظومة للأحكام.

المرحلة الثانية: بناء العقود الذكية والبلوك تشين

- **برمجة العقد الذكي:** كتابة كود البلوك تشين لإصدار عملة Service Coin
- **البرمجة الوظيفية:** برمجة دوال الإصدار المرتبطة بإضافة الخدمات، ودوال الحرق المرتبطة باستهلاكها.
- **شبكة الاختبار:** الإطلاق على شبكة الاختبار (Testnet) لتجربة جميع العمليات المالية والمزامنة بشكل مجاني بالكامل وآمن.

المرحلة الثالثة: برمجة النظام الخلفي ولوحة التحكم (Backend & Admin Panel)

- **الخوادم والنظام الخلفي:** تأسيس بنية الخوادم وتطوير واجهات برمجة التطبيقات (APIs) باستخدام إطار عمل Laravel وبيئة تشغيل PHP 8.3 لضمان أعلى سرعة استجابة.
- **الربط اللامركزي :** كتابة أكواد الربط الآلي بين قاعدة البيانات (MySQL) وشبكة البلوك تشين، بحيث تتم عمليات الإصدار والحرق برمجياً دون تدخل يدوي.
- **لوحة التحكم:** بناء واجهة إدارية شاملة ومتقدمة لإدارة النظام بالكامل للتحكم الدقيق في صلاحيات المؤسسات والمستخدمين وكل شيئ داخل المنصة.

المرحلة الرابعة: تطوير تطبيقات الهواتف الذكية (iOS & Android)

- **برمجة التطبيقات:** بناء كود موحد عالي الأداء باستخدام Flutter و Dart ليعمل بسلاسة على أنظمة آبل وأندرويد معاً.
- **المحافظ الرقمية:** تكامل المحافظ الرقمية داخل التطبيق ليتمكن المستخدم من استعراض رصيده من العملات الخدمية واستخدامها.
- **الاتصال اللحظي (Real-time):** هندسة خوادم WebSockets (عبر أدوات مثل Laravel Reverb) لربط التطبيق بالخوادم وتحديث حالات الحجوزات والعملات في نفس اللحظة (Live Data Streaming).
- **الإشعارات السحابية:** بناء معمارية الإشعارات الفورية (Push Notifications) عبر Firebase لضمان وصول التنبيهات للمستخدمين فور توفر خدمات جديدة أو تأكيد استهلاك الخدمة.