

تقرير المشروع: نظام لإدارة طلبات الدعم

المقدمة:

يهدف هذا المشروع إلى تطوير نظام ويب بسيط لإدارة طلبات الصيانة داخل المؤسسة بشكل منظم وفعال. يُتيح النظام للموظفين إمكانية رفع طلبات الصيانة المتعلقة بالمشاكل الفنية أو الأعطال التي تواجههم أثناء العمل، بحيث يتم إرسال هذه الطلبات إلى موظفي الصيانة لمراجعتها واتخاذ الإجراءات المناسبة. كما يوفر النظام وسيلة لمتابعة حالة الطلبات وتحديثها من قبل موظفي الصيانة، مما يساعد على تسريع عملية المعالجة وتقليل التأخير. يساهم النظام في تحسين التواصل بين الأقسام الفنية والمستخدمين، ويرفع من كفاءة العمل وجودة الخدمة داخل المؤسسة.

المتطلبات:

المتطلبات الوظيفية:

- تمكين الموظفين من تسجيل الدخول باستخدام اسم المستخدم وكلمة المرور.
- السماح للموظفين برفع طلبات صيانة تحتوي على وصف المشكلة.
- عرض قائمة بجميع الطلبات التي قام الموظف بتقديمها، مع حالة كل طلب.
- تمكين موظفي الصيانة من عرض الطلبات المرسلة إليهم وتحديث حالتها (معلق - قيد التنفيذ - مكتمل).
- تمكين المشرف من إدارة حسابات المستخدمين.

المتطلبات غير الوظيفية:

- يجب أن يكون النظام سهل الاستخدام لجميع أنواع المستخدمين.
- يجب أن يعمل النظام بسرعة واستجابة مناسبة على مختلف المتصفحات.
- يجب أن يُخزّن النظام البيانات بشكل آمن في قاعدة البيانات.
- يجب أن يكون الكود قابلاً للتطوير والتحسين في المستقبل.

وصف المشروع:

تم تطوير هذا المشروع بهدف إنشاء نظام إلكتروني يساعد في إدارة طلبات الصيانة داخل بيئة العمل. يقوم النظام بتوفير واجهة سهلة للمستخدمين (الموظفين) لرفع طلبات صيانة عند حدوث أعطال أو مشاكل فنية، بحيث يتم تحويل هذه الطلبات تلقائيًا إلى موظفي الصيانة لمتابعتها واتخاذ الإجراءات المناسبة.

يتكوّن النظام من واجهات مختلفة مخصصة حسب نوع المستخدم، حيث يمكن للموظف العادي رفع الطلبات ومتابعة حالتها، بينما يمتلك موظف الصيانة القدرة على معالجة الطلبات وتحديث حالتها. كما يتيح النظام للمشرف إدارة الحسابات ومراقبة الطلبات بشكل عام. يهدف هذا المشروع إلى تقليل الاعتماد على الطرق التقليدية في تقديم طلبات الصيانة، وتحسين سرعة وكفاءة المعالجة داخل المؤسسة.

الوظائف الرئيسية للنظام:

يقدم النظام مجموعة من الوظائف الأساسية التي تضمن سير عملية الصيانة بشكل منظم وسهل، وتشمل:

تسجيل الدخول:

يتيح للمستخدمين الدخول إلى النظام باستخدام بياناتهم الخاصة، مع صلاحيات مختلفة حسب نوع المستخدم (موظف – موظف صيانة – مشرف).

رفع طلب صيانة:

يمكن للموظف العادي إرسال طلب صيانة يحتوي على وصف للمشكلة، ليتم استلامه من قبل فريق الصيانة.

عرض الطلبات:

يمكن لكل مستخدم عرض الطلبات التي أنشأها، ومتابعة حالتها (معلق، قيد التنفيذ، مكتمل).

تحديث حالة الطلب:

يتيح لموظفي الصيانة تعديل حالة الطلب حسب تقدم العمل على المشكلة.

إدارة المستخدمين (للمشرف):

يمكن للمشرف إضافة أو حذف أو تعديل بيانات المستخدمين الموجودين في النظام.

البيئة والأدوات المستخدمة:

البرمجيات المستخدمة

تم استخدام مجموعة من البرمجيات والأدوات لتطوير النظام، منها:

لغة PHP: لبناء الجانب الخلفي من النظام والتعامل مع الخادم وقاعدة البيانات.

MySQL: لإدارة قاعدة البيانات وتخزين البيانات.

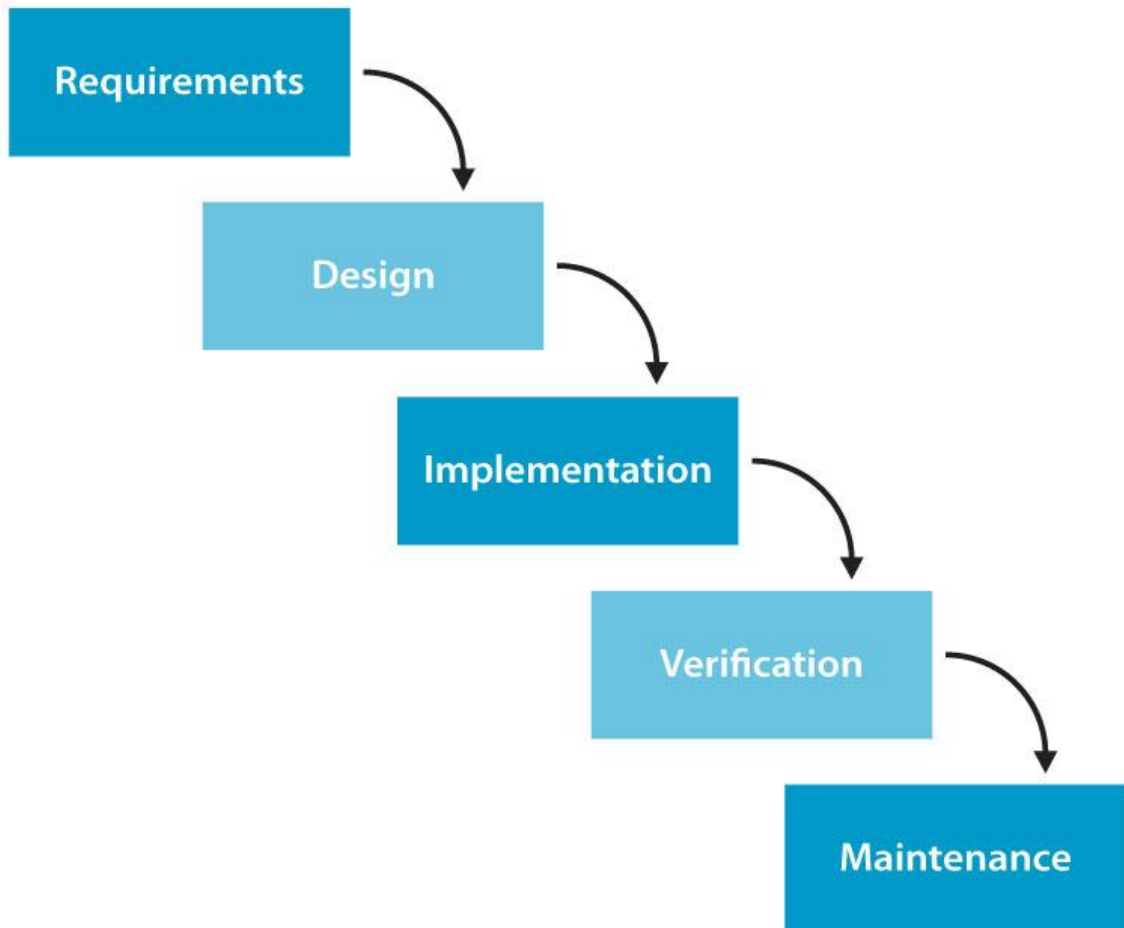
HTML / CSS / JavaScript: لتصميم الواجهات الأمامية للمستخدم والتفاعل معها.

XAMPP: كبيئة محلية لتشغيل Apache و MySQL على الجهاز أثناء التطوير.

phpMyAdmin: لإدارة قواعد البيانات بطريقة مرئية وسهلة.

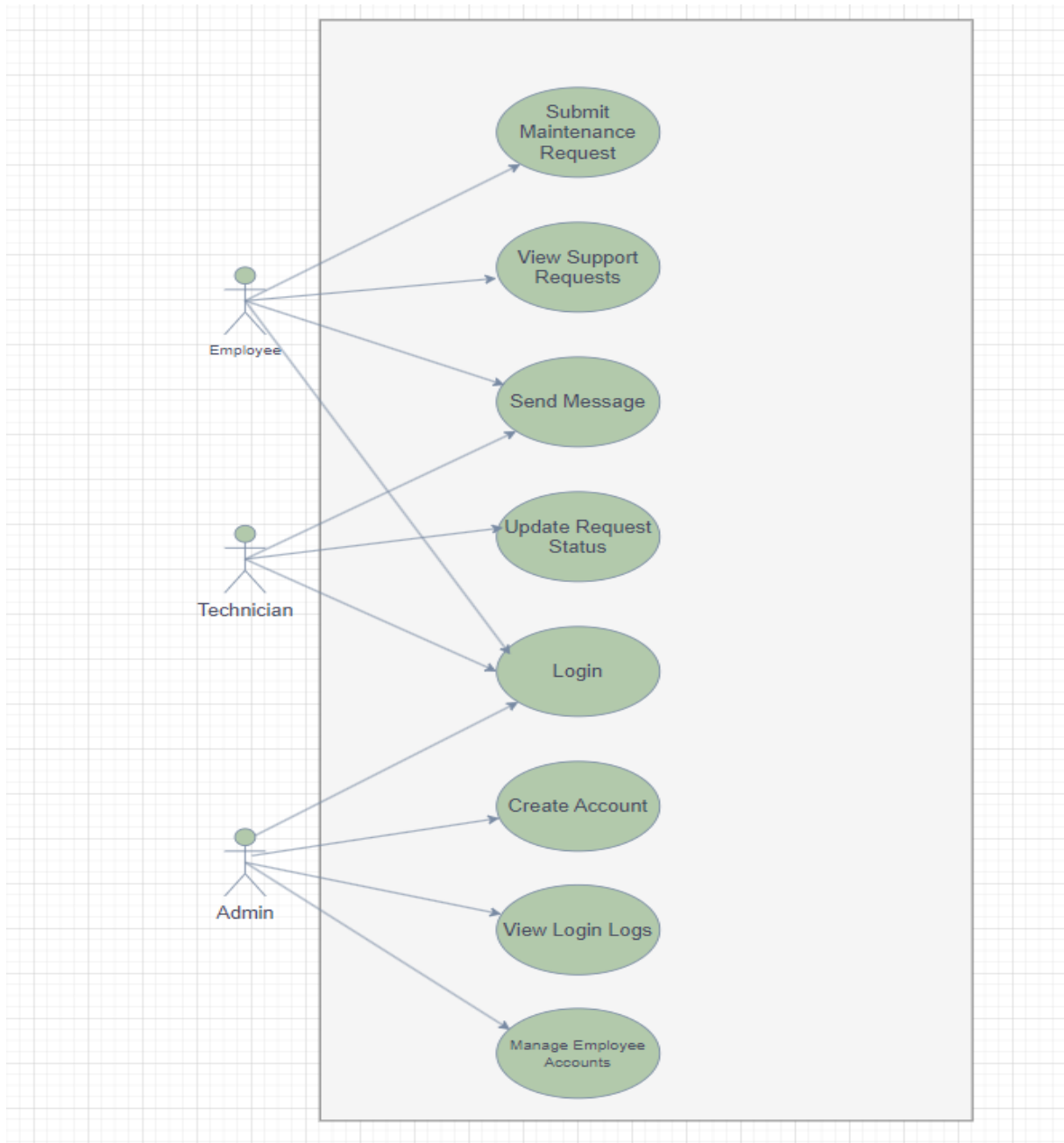
المنهجية المستخدمة:

Waterfall :



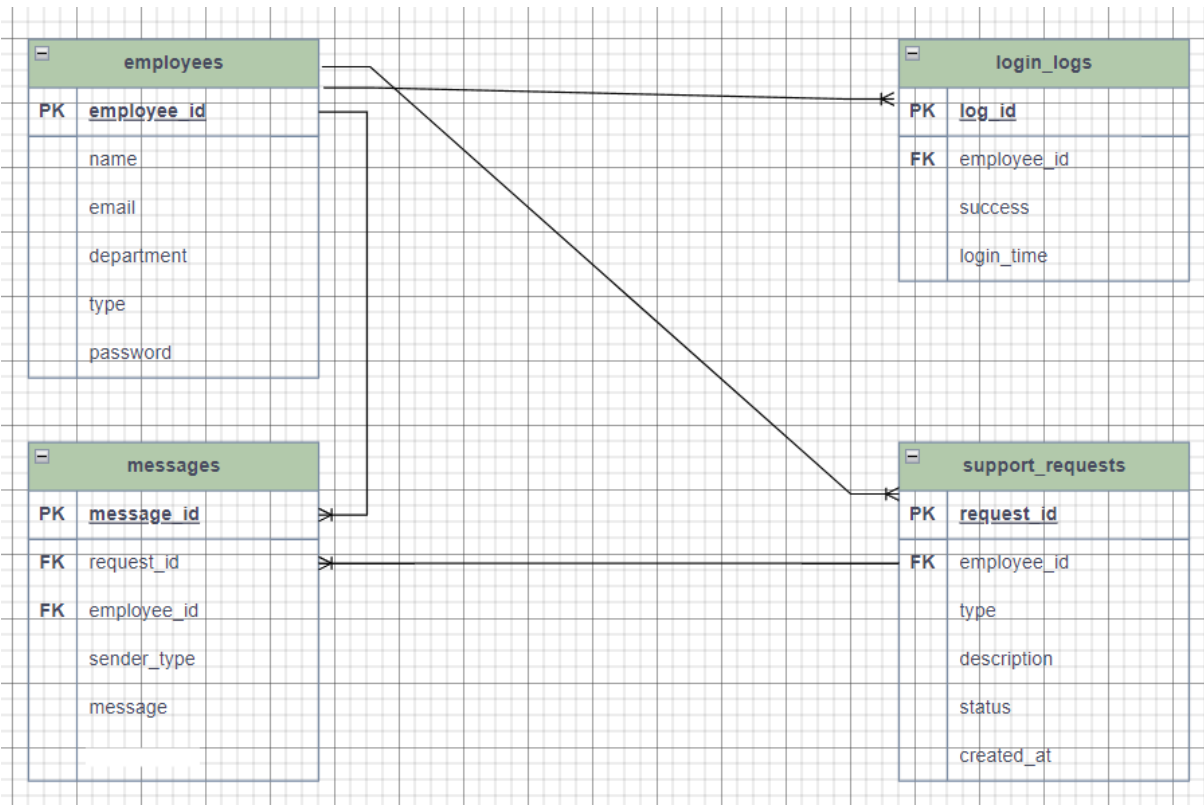
تم اختيار منهجية الشلال (Waterfall Model) لتطوير هذا المشروع نظرًا لطبيعة المشروع المحددة والواضحة من البداية، حيث لا توجد تغييرات متوقعة في المتطلبات أثناء التنفيذ. منهجية الشلال تُعد مناسبة للمشاريع التي تتبع مراحل متسلسلة، تبدأ بتحليل المتطلبات، ثم التصميم، يليها التنفيذ، ثم الاختبار، وأخيرًا التسليم.

Use Case Diagram:



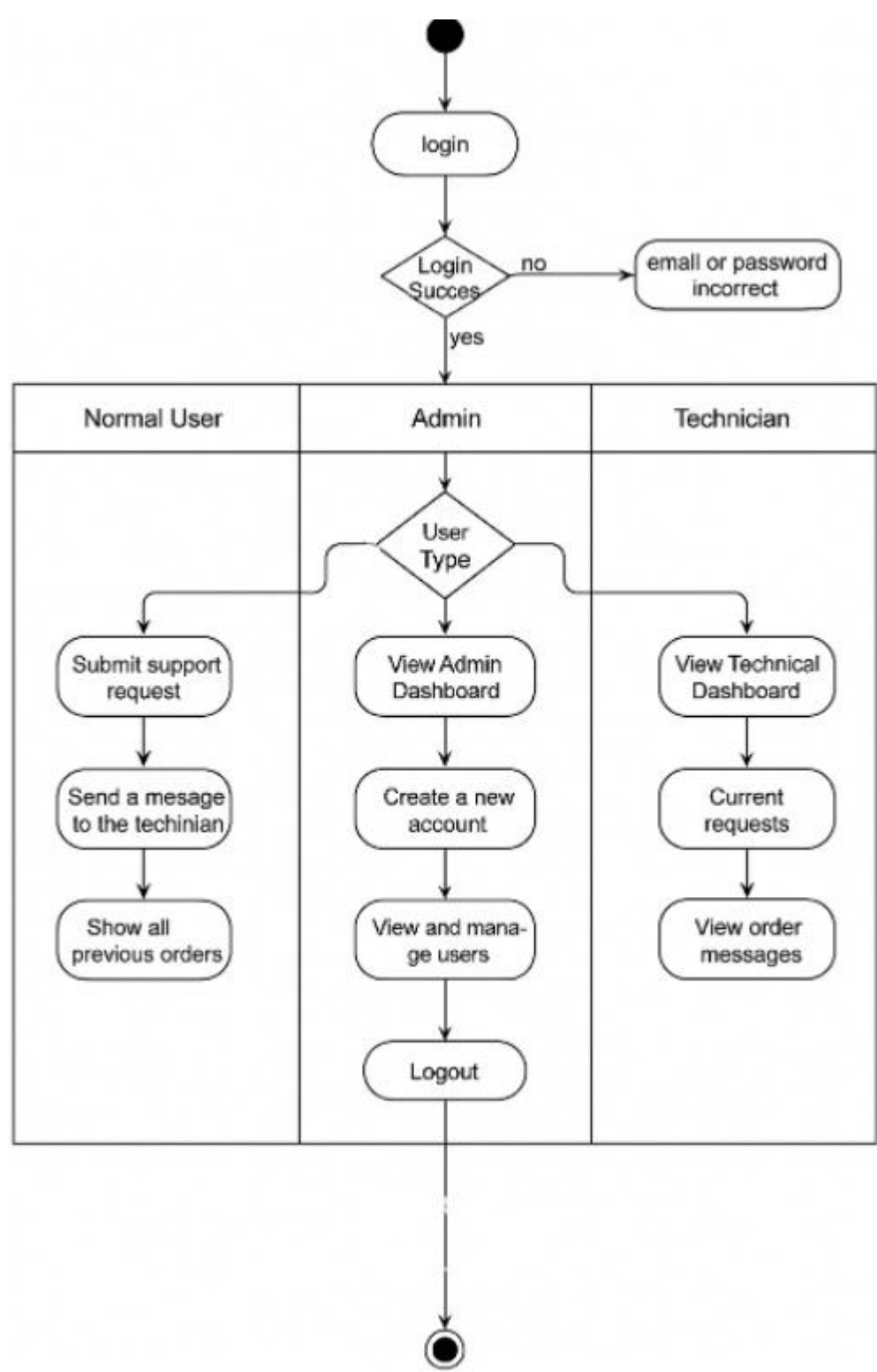
يُظهر مخطط حالات الاستخدام (Use Case Diagram) العلاقة بين المستخدمين الرئيسيين للنظام (الموظف – موظف الصيانة – المشرف) والوظائف التي يمكن لكل منهم القيام بها، وذلك لتوضيح كيفية تفاعلهم مع النظام.

ERD:

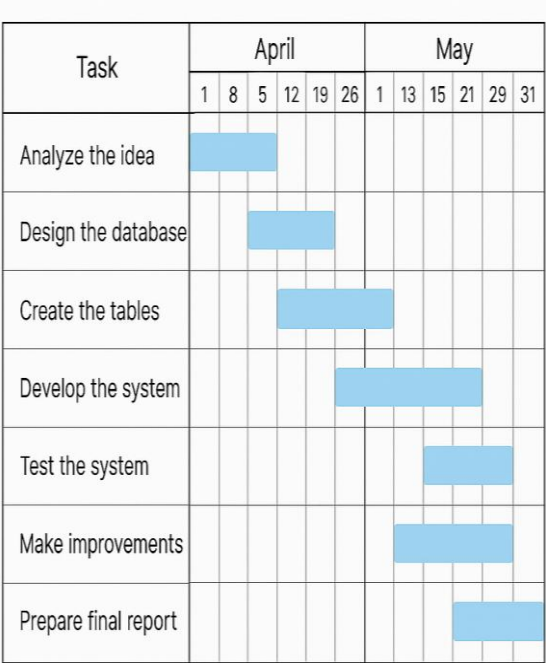


يعرض مخطط الكيان-العلاقة (Entity Relationship Diagram) الهيكل العام لقاعدة البيانات المستخدمة في النظام، ويوضح العلاقات بين الجداول الأساسية مثل جدول الموظفين، طلبات الصيانة، وطلبات الدعم، وغيرها. يساهم هذا المخطط في توضيح كيفية تخزين البيانات وكيفية ارتباط الكيانات ببعضها.

Activity Diagram:



Gantt Chart:



يتم تنفيذ المشروع عبر مجموعة من المهام المتتابعة لضمان سير العمل بشكل منظم وفعال، وهذه المهام تشمل:

تحليل الفكرة (Analyze the Idea):

دراسة فكرة المشروع، تحديد الأهداف والمتطلبات الأساسية للنظام.

تصميم قاعدة البيانات (Design the Database):

وضع هيكل قاعدة البيانات وتنظيم الجداول والعلاقات بينها.

إنشاء الجداول (Create the Tables):

تنفيذ تصميم قاعدة البيانات فعليًا عبر إنشاء الجداول في نظام إدارة قواعد البيانات.

تطوير النظام (Develop the System):

برمجة الوظائف الأساسية للنظام باستخدام لغات البرمجة المناسبة.

اختبار النظام (Test the System):

التحقق من صحة عمل النظام واكتشاف وإصلاح الأخطاء المحتملة.

إجراء التحسينات (Make Improvements):

تعديل وتطوير النظام بناءً على نتائج الاختبار والملاحظات.

إعداد التقرير النهائي (Prepare Final Report):

توثيق جميع مراحل المشروع ونتائجه في تقرير شامل ومنظم.

الخلاصة والتوصيات:

يُعد هذا المشروع خطوة مهمة لتطوير نظام بسيط وفعال لإدارة طلبات الصيانة داخل المؤسسة. ساعد المشروع على تطبيق المهارات المكتسبة في تحليل وتصميم وتطوير الأنظمة، كما وفر تجربة عملية في التعامل مع قواعد البيانات وتطوير واجهات المستخدم.

ينصح بتطوير المشروع مستقبلاً بإضافة مزايا جديدة مثل:

دعم استعادة كلمة المرور لتعزيز أمان الحسابات.

إضافة نظام إشعارات لتبليغ المستخدمين بحالة طلباتهم.

تحسين تصميم الواجهة لزيادة سهولة الاستخدام.

اعتماد منهجيات تطوير أكثر مرونة.