الجامعة السورية الخاصة
SYRIAN PRIVATE UNIVERSITY

الجامعة السورية الخاصة — كلية هندسة الذكاء الاصطناعي

قسم هندسة البرمجيات ونظم المعلومات الذكية

منصة رقمية تفاعلية لمنتدى قراءة الكتب ومشاركة المحتوى الأدبي

Read Loop — An Interactive Digital Platform for a Book Reading Forum and Sharing Literary Content

تقرير مشروع فصلي

مقدمًا استكمالًا لمتطلبات الحصول على درجة الإجازة في الهندسة

هندسة الذكاء الاصطناعي — هندسة البرمجيات ونظم المعلومات الذكية

إعداد

يحيى علي حمد الكراد

إشراف

د فادي ابراهيم

العام الدراسي

2026 – 2025

فهرس المحتويات

فهرس المحتويات 2
فهرس الجداول 6
فهرس الأشكال 7
قائمة الاختصارات والمصطلحات العلمية 8
الملخص 9
Abstract 10
الفصل الأول: المقدمة 12
1.1 نظرة عامة على المشروع وخلفيته 12
1.2 بيان المشكلة وأهميتها العملية والأكاديمية 12
1.3 أهداف المشروع (الرئيسية والفرعية) 12
1.4 نطاق المشروع والقيود والافتراضات 13
النطاق 13
القيود 13
الافتراضات 13
1.5 مساهمة المشروع في المعرفة والمجال 13
1.6 منهجية البحث والأساليب العلمية المتبعة 14
1.7 هيكل الوثيقة التنظيمي 14
1.8 التعاريف والمفاهيم الأساسية 14
الفصل الثاني: منهجية التطوير وإدارة المشروع 15
2.1 منهجية التطوير المختارة والمبررات 15
2.2 خطة المشروع الزمنية 15
2.3 إدارة المخاطر 16
2.4 فريق العمل وتوزيع المهام والأدوار 16
2.5 إدارة الجودة في المشروع 16
2.6 أدوات إدارة المشروع المستخدمة 16
2.7 آليات المراقبة والتقييم المستمر 17
2.8 إدارة تشكيلة البرمجيات (SCM) 17
الفصل الثالث: الإطار النظري والدراسات السابقة 18
3.1 الدراسات السابقة في مجال النظم الذكية والمشابهة 18
3.2 التقنيات والأطر الحديثة في المجال 18
3.3 التحليل المقارن للأنظمة والمنتجات الحالية 19

3.4 الفجوة البحثية والابتكار المقترح 19
3.5 الإطار النظري والدراسات المرجعية 19
3.6 التوجهات الحديثة والمستقبلية في المجال 19
الفصل الرابع: تحليل المتطلبات ونمذجة النظام 20
4.1 دراسة الجدوى 20
الجدوى الفنية 20
الجدوى الاقتصادية 20
الجدوى التشغيلية 20
4.2 جمع المتطلبات 20
4.3 تحديد أصحاب المصلحة وتحليل احتياجاتهم 20
4.4 المتطلبات الوظيفية 21
4.5 المتطلبات غير الوظيفية 22
4.6 قائمة أولويات المتطلبات (MoSCoW) 22
4.7 مخططات UML للمتطلبات — مخطط وتوصيف لكل متطلب 23
4.7.1 المصادقة: إنشاء حساب وتسجيل الدخول والخروج (FR1–FR3) 23
4.7.2 التصفح والبحث وعرض تفاصيل الكتاب (FR4–FR5، FR13) 25
4.7.3 قراءة الكتاب أونلاين وتحميله (FR6، FR12) 25
4.7.4 إضافة تقييم لكتاب (FR8) 26
4.7.5 إضافة تعليق على كتاب (FR7) 27
4.7.6 نشر كتاب جديد (FR9) 29
4.7.7 تعديل أو حذف كتاب (FR10–FR11) 30
4.7.8 إدارة المستخدمين: تغيير الدور، الحظر، فك الحظر، والحذف (FR14، FR16) 31
4.7.9 إدارة التعليقات من قبل المدير (FR15) 32
4.7.10 تعديل الحساب الشخصي وكلمة المرور (FR17) 33
4.7.11 تبديل لغة الواجهة وعرض إحصائيات الكاتب (FR18–FR19) 34
4.8 النمذجة حسب المنهجية المستخدمة 35
4.9 تحليل وتوثيق المتطلبات 36
الفصل الخامس: التصميم المعماري والتفصيلي 37
5.1 التصميم عالي المستوى 37
5.2 تصميم قاعدة البيانات 37
المخطط العلائقي (Relational Schema) 38
5.3 تصميم الواجهات (المسارات ووحدات التحكم) 39
5.4 تصميم الأمان والصلاحيات 39

آلية المصادقة (Authentication) 39
التفويض (Authorization) وأدوار المستخدمين 39
حماية البيانات 40
الفصل السادس: بيئة العمل والتنفيذ البرمجي 41
6.1 الأدوات البرمجية والمكتبات المستخدمة 41
6.2 بيئة التطوير وبنية المشروع 41
6.3 لغات البرمجة وإطارات العمل 42
6.4 خادم قاعدة البيانات ونظام التشغيل 42
6.5 المكتبات والتبعيات (Dependencies) 42
6.6 أدوات إدارة الإصدارات والتحكم المصدري 43
6.7 أدوات التكامل والنشر المستمر (CI/CD) 43
6.8 بيئة الاختبار 43
6.9 التنفيذ البرمجي 43
تنفيذ طبقة الوصول إلى المعطيات (ORM) 43
تنفيذ طبقة الأعمال والمصادقة 43
تنفيذ واجهة المستخدم 44
الفصل السابع: الاختبار والتقييم 45
7.1 خطة الاختبار 45
7.2 أنواع الاختبارات 45
7.3 مخططات الاختبار ونتائجه 45
7.4 نتائج الاختبار وتحليلها 46
7.5 مقاييس الجودة 47
7.6 اختبار المستخدمين 47
الفصل الثامن: النتائج والتحليل 48
8.1 نتائج تطبيق النظام 48
8.2 تحليل مقارن مع الأنظمة الموجودة 48
8.3 تلبية الأهداف 49
8.4 تحليل نقاط القوة والضعف 49
نقاط القوة 49
نقاط الضعف 49
8.5 قياس مؤشرات الأداء (KPIs) 50
الخاتمة 51
قائمة المراجع 52

ملحق أ: نماذج وأمثلة 53
أ-1: نموذج وصف حالة استخدام 53
أ-2: نماذج User Story 54
أ-3: مخطط جانت (جدول زمني) 55
أ-4: مصفوفة تتبع المتطلبات (RTM) 55
أ-5: قائمة مراجع (APA) 56
ملحق ج: نموذج خطة الاختبار 57
ج-1: نطاق الاختبار 57
ج-2: استراتيجية الاختبار 57
ج-3: بيئة الاختبار 57
ج-4: معايير القبول 57
ملحق د: نموذج دليل المستخدم 58
د-1: نظرة عامة 58
د-2: البدء السريع 58
د-3: استكشاف الأخطاء الشائعة 58
ملحق هـ: الأخلاقيات البحثية واستخدام الذكاء الاصطناعي 59
هـ-1: النزاهة الأكاديمية 59
هـ-2: استخدام أدوات الذكاء الاصطناعي 59
هـ-3: قواعد السلوك الأكاديمي 59

فهرس الجداول

جدول 1. أهداف المشروع بأسلوب SMART

جدول 2. توزيع مراحل المشروع الزمنية (مخطط جانت)

جدول 3. سجل إدارة المخاطر

جدول 4. التحليل المقارن مع الأنظمة المشابهة

جدول 5. المتطلبات الوظيفية

جدول 6. المتطلبات غير الوظيفية

جدول 7. أولويات المتطلبات (MoSCoW)

جدول 8. المخطط العلائقي لقاعدة البيانات

جدول 9. الأدوات والمكتبات المستخدمة

جدول 10. حالات الاختبار ونتائجها

جدول 11. مقاييس الجودة

جدول 12. مصفوفة تتبع المتطلبات (RTM)

فهرس الأشكال

شكل 1. مخطط حالات الاستخدام العام للنظام

شكل 2. مخطط نشاط: تسجيل حساب جديد وتسجيل الدخول

شكل 3. مخطط نشاط: البحث عن كتاب وتصفحه

شكل 4. مخطط تسلسل: قراءة الكتاب أونلاين وتحميله

شكل 5. مخطط نشاط: إضافة تقييم لكتاب

شكل 6. مخطط نشاط: إضافة تعليق على كتاب

شكل 7. مخطط تسلسل: نشر كتاب جديد

شكل 8. مخطط نشاط: تعديل أو حذف كتاب

شكل 9. مخطط تسلسل: حذف مستخدم من قبل المدير

شكل 10. مخطط نشاط: إدارة المستخدمين (حظر/فك حظر/تغيير الدور/حذف)

شكل 11. مخطط نشاط: إدارة التعليقات من قبل المدير

شكل 12. مخطط نشاط: تعديل الحساب الشخصي وكلمة المرور

شكل 13. مخطط نشاط: تبديل اللغة وعرض إحصائيات الكاتب

شكل 14. نموذج الكيانات والعلاقات (ERD)

شكل 15. مخطط الأصناف (Class Diagram)

شكل 16. شجرة مجلدات المشروع (Laravel)

شكل 17. الصفحة الرئيسية للمنصة بعد التشغيل الفعلي

قائمة الاختصارات والمصطلحات العلمية

الاختصار	الشرح
API	واجهة برمجة التطبيقات (Application Programming Interface)
CRUD	إنشاء / قراءة / تحديث / حذف (Create, Read, Update, Delete)
CSRF	تزوير الطلب عبر المواقع (Cross-Site Request Forgery)
ERD	مخطط الكيانات والعلاقات (Entity Relationship Diagram)
FR	متطلب وظيفي (Functional Requirement)
MVC	نموذج - عرض - تحكم (Model-View-Controller)
NFR	متطلب غير وظيفي (Non-Functional Requirement)
ORM	تحويل علائقي كائني (Object-Relational Mapping)
RTM	مصفوفة تتبع المتطلبات (Requirements Traceability Matrix)
SPA	تطبيق الصفحة الواحدة (Single Page Application)
SQL	لغة الاستعلام البنيوية (Structured Query Language)
UI/UX	واجهة المستخدم / تجربة المستخدم
UML	لغة النمذجة الموحدة (Unified Modeling Language)
XSS	البرمجة النصية عبر المواقع (Cross-Site Scripting)

الملخص

يعالج هذا المشروع مشكلة محدودية المنصات العربية التفاعلية المخصصة لقراءة الكتب ومشاركة المحتوى الأدبي، إذ تعتمد كثير من الحلول الحالية إما على عرض الكتب دون إتاحة تفاعل حقيقي بين القراء، أو على استيراد محتوى أجنبي لا يراعي احتياجات القارئ العربي من حيث اللغة والتصنيف والتفاعل المجتمعي. هدف المشروع إلى تصميم وتنفيذ منصة ويب متكاملة تحت اسم Read Loop تتيح للمستخدمين تصفح الكتب وقراءتها إلكترونيًا، والتفاعل معها عبر التعليقات والتقييمات، كما تتيح للكتّاب نشر أعمالهم الأدبية وإدارتها، وتمنح مدير النظام صلاحيات كاملة لإدارة المستخدمين والمحتوى.

اعتُمدت في تطوير المشروع منهجية التطوير التكرارية التزايدية (Iterative & Incremental)، وطُبّقت المعمارية البرمجية MVC باستخدام إطار العمل Laravel من جهة الخلفية، إلى جانب Tailwind CSS وVite من جهة الواجهة الأمامية، وقاعدة بيانات علائقية (MySQL/SQLite). مرّ المشروع بمراحل جمع المتطلبات وتحليلها، ونمذجة النظام باستخدام مخططات UML (حالات الاستخدام، الأصناف، التسلسل، ERD)، ثم التصميم المعماري، فالتنفيذ البرمجي، فالاختبار على بيئة محلية.

أظهرت نتائج الاختبار أن النظام ينفّذ وظائفه الأساسية بنجاح، من تسجيل الحسابات وتسجيل الدخول، مرورًا بتصفح الكتب والبحث عنها وتقييمها والتعليق عليها، وصولًا إلى نشر الكتب وإدارة المستخدمين من قبل المدير مع تطبيق قيود أمان تمنع حذف آخر حساب مدير أو حذف المستخدم لنفسه. يخلص المشروع إلى أن اعتماد معمارية MVC مع منهجية تكرارية ساعد على ضبط جودة التطوير وتقليل المخاطر، ويوصي بتوسعة النظام مستقبلًا بإضافة نظام توصية للكتب ودعم رفع ملفات القراءة الإلكترونية وتحسين الأداء عند زيادة عدد المستخدمين.

الكلمات المفتاحية: منصة قراءة إلكترونية، منتدى أدبي، Laravel، معمارية MVC، تفاعل المستخدمين، تطوير الويب

Abstract

This project addresses the limited availability of interactive Arabic-friendly .platforms dedicated to online book reading and literary content sharing Most existing solutions either present books without genuine reader interaction or import foreign content that does not fully address the language, categorization, and community-interaction needs of Arabic readers. The project's objective is to design and implement an integrated web platform, Read Loop, that allows users to browse and read books online, interact through comments and ratings, and enables authors to publish and manage their literary work, while granting the system .administrator full control over users and content

The project was developed following the Iterative & Incremental methodology, applying the MVC software architecture with the Laravel framework on the back end, Tailwind CSS and Vite on the front end, and a relational database (MySQL/SQLite). The work progressed through requirements gathering and analysis, system modeling using UML ,diagrams (use case, class, sequence, ERD), architectural design .implementation, and local testing

Testing results show that the system successfully performs its core ,functions — from account registration and login, through browsing searching, rating and commenting on books, to publishing books and administrator-level user management with safety constraints preventing the deletion of the last remaining administrator or self-deletion. The project concludes that combining an MVC architecture with an iterative methodology improved development quality and reduced risk, and ,recommends future extensions such as a book recommendation engine

e-book file upload support, and performance optimization for larger user .bases

Keywords: Online reading platform, literary forum, Laravel, MVC architecture, user interaction, web development

الفصل الأول: المقدمة

1.1 نظرة عامة على المشروع وخلفيته

تندرج القراءة الرقمية ضمن أكثر المجالات نموًا في صناعة المحتوى خلال السنوات الأخيرة، مع تزايد اعتماد القراء على المنصات الإلكترونية بدلًا من النسخ الورقية التقليدية. ورغم انتشار منصات عالمية مخصصة لقراءة الكتب ومشاركتها مثل Goodreads وWattpad، تبقى الحاجة قائمة لمنصة عربية خفيفة وسهلة الاستخدام تجمع بين إتاحة المحتوى الأدبي والتفاعل المجتمعي حوله.

ينطلق هذا المشروع من هذه الحاجة، فيقدّم منصة ويب باسم Read Loop مخصصة لمنتدى قراءة الكتب ومشاركة المحتوى الأدبي، تتيح للمستخدم العادي تصفح الكتب وقراءتها والتفاعل معها، وتمنح فئة الكتّاب إمكانية نشر أعمالهم، بينما يتولى مدير النظام إدارة المستخدمين والمحتوى بشكل كامل.

1.2 بيان المشكلة وأهميتها العملية والأكاديمية

تكمن المشكلة في أن كثيرًا من الأنظمة الحالية لعرض الكتب تعتمد إما على عرض ثابت للمحتوى دون تفاعل حقيقي بين القراء، أو على منصات أجنبية لا تراعي بنية اللغة العربية وتصنيفاتها الأدبية، مما يؤدي إلى ضعف تفاعل القراء العرب مع المحتوى الأدبي الرقمي، وصعوبة وصول الكتّاب الناشئين إلى جمهورهم بدون وسيط تقليدي (دار نشر). من الناحية الأكاديمية، يتيح المشروع تطبيقًا عمليًا شاملًا لدورة حياة تطوير البرمجيات، بدءًا من تحليل المتطلبات ونمذجة النظام، مرورًا بالتصميم المعماري، وانتهاءً بالتنفيذ والاختبار.

1.3 أهداف المشروع (الرئيسية والفرعية)

صيغت أهداف المشروع وفق أسلوب SMART كما هو موضح في الجدول 1.

جدول 1. أهداف المشروع بأسلوب SMART

الوصف وفق SMART	الهدف
بناء واجهات لتصفح الكتب وقراءتها أونلاين خلال فترة المشروع الفصلي، قابلة للقياس عبر عدد الصفحات المُنجزة.	تطوير منصة تصفح وقراءة
إتاحة إضافة تعليقات وتقييمات من 1 إلى 5 لكل كتاب لكل مستخدم، مع تحديث تلقائي لمتوسط التقييم.	تمكين التفاعل المجتمعي

الوصف وفق SMART	الهدف
توفير واجهة لإضافة الكتب وتعديلها وحذفها من قبل حسابات الكتّاب، مرتبطة بحساب المستخدم مباشرة.	تمكين الكتّاب من النشر
بناء نظام أدوار (مستخدم/كاتب/مدير) مع صلاحيات حظر وفك حظر وحذف، وقيود تمنع حذف آخر مدير.	إدارة آمنة للمستخدمين
تصميم واجهة بسيطة متعددة اللغة، مع تشفير كلمات المرور وحماية من هجمات CSRF/XSS الشائعة.	ضمان سهولة الاستخدام والأمان

وتتلخص الأهداف الفرعية الداعمة بما يلي:

- تسهيل عملية البحث عن الكتب من خلال نظام بحث وتصفية بسيط.
- بناء نظام قابل للتطوير مستقبلًا يسمح بإضافة ميزات جديدة (مثل نظام التوصية) دون إعادة هيكلة جذرية.
- توثيق المشروع بشكل يتيح لطالب آخر أو فريق تطوير مستقبلي متابعة العمل عليه بسهولة.

1.4 نطاق المشروع والقيود والافتراضات

النطاق

يشمل النطاق: تسجيل الحسابات وتسجيل الدخول/الخروج، تصفح الكتب والبحث عنها، قراءة الكتاب أونلاين، إضافة تقييم وتعليق واحد لكل مستخدم لكل كتاب، نشر الكتب وتعديلها وحذفها من قبل الكتّاب، ولوحة تحكم لمدير النظام لإدارة المستخدمين والتعليقات. لا يشمل النطاق الحالي: أنظمة الدفع الإلكتروني، أو تطبيقات الجوال الأصلية، أو محرك توصية ذكي مبني على تعلم الآلة (يُترك كعمل مستقبلي).

القيود

تتمثل القيود الرئيسية في الإطار الزمني المحدود لمشروع فصلي واحد (نحو 14 أسبوعًا)، وكون التطوير يتم من قبل طالب واحد دون فريق موزّع الأدوار، إضافة إلى الاعتماد على بيئة تطوير محلية (XAMPP/Laravel) قبل النشر النهائي.

الافتراضات

يُفترض أن المستخدم يمتلك اتصالًا مستقرًا بالإنترنت عند استخدام النسخة المنشورة من المنصة، وأن المتصفحات الحديثة (Chrome، Firefox، Edge) هي البيئة المستهدفة للعرض.

1.5 مساهمة المشروع في المعرفة والمجال

يقدّم المشروع دراسة حالة تطبيقية لبناء منصة محتوى أدبي عربية تجمع بين نموذج القراءة (كالمكتبات الرقمية) ونموذج الشبكة الاجتماعية الأدبية (كـ Goodreads)، مع تبسيط في البنية يجعلها قابلة لإعادة الاستخدام كقالب أساسي لمشاريع مشابهة (منتديات محتوى، مراجعات منتجات، إلخ) بفضل اعتمادها معمارية MVC قياسية.

1.6 منهجية البحث والأساليب العلمية المتبعة

اعتُمد في هذا المشروع منهج هندسي تطبيقي (Engineering/Applied approach) يقوم على تحليل المتطلبات، ونمذجة النظام، وتصميمه، وتنفيذه، واختباره ضمن دورة تكرارية تزايدية، مع الاستعانة بأدلة تصميم وتوثيق قياسية (UML، RTM، MoSCoW) كما هو مفصّل في الفصل الثاني.

1.7 هيكل الوثيقة التنظيمي

يتوزع هذا التقرير على ثمانية فصول رئيسية: يعرض الفصل الأول المقدمة العامة للمشروع؛ ويتناول الفصل الثاني منهجية التطوير وإدارة المشروع؛ ويستعرض الفصل الثالث الإطار النظري والدراسات السابقة؛ ويغطي الفصل الرابع تحليل المتطلبات ونمذجة النظام؛ ويقدّم الفصل الخامس التصميم المعماري والتفصيلي؛ ويوثّق الفصل السادس بيئة العمل والتنفيذ البرمجي؛ ويعرض الفصل السابع خطة الاختبار ونتائجه؛ ويختم الفصل الثامن بعرض النتائج وتحليلها. تلي ذلك الخاتمة، وقائمة المراجع، والملاحق.

1.8 التعاريف والمفاهيم الأساسية

المصطلح	التعريف
المستخدم (User)	الفاعل الأساسي القارئ الذي يمكنه تصفح الكتب وتقييمها والتعليق عليها.
الكاتب (Author)	مستخدم مسجل بصلاحية إضافية تتيح له نشر الكتب وإدارتها.
مدير النظام (Admin)	الفاعل صاحب الصلاحيات الكاملة لإدارة المستخدمين والمحتوى.
التقييم (Rating)	درجة رقمية من 1 إلى 5 يمنحها المستخدم لكتاب معيّن، بحد أقصى تقييم واحد لكل مستخدم لكل كتاب.

الفصل الثاني: منهجية التطوير وإدارة المشروع

2.1 منهجية التطوير المختارة والمبررات

اعتُمدت منهجية التطوير التكرارية التزايدية (Iterative & Incremental) لتطوير هذا المشروع، حيث يُقسَّم العمل إلى مراحل صغيرة متكررة بدلًا من إنجازه دفعة واحدة، وفي كل مرحلة تُضاف وظائف جديدة أو تُحسَّن الوظائف القائمة استنادًا إلى المراجعة الذاتية والتغذية الراجعة من المشرف.

تناسب هذه المنهجية طبيعة المشروع للأسباب التالية:

- تسمح باكتشاف الأخطاء مبكرًا وتعديل المتطلبات بسهولة، مما يقلل المخاطر ويضمن جودة أعلى.
- تلائم عمل الطالب الفردي إذ تتيح تسليم أجزاء وظيفية مكتملة (مثل: المصادقة، ثم الكتب، ثم التقييمات) بدلًا من انتظار اكتمال النظام كله.
- تدعم التكامل المستمر بين الواجهة الخلفية والأمامية على دفعات قابلة للاختبار.

2.2 خطة المشروع الزمنية

جرى تقسيم العمل على 14 أسبوعًا موزعة على المراحل الرئيسية التالية (مخطط جانت مبسّط):

جدول 2. توزيع مراحل المشروع الزمنية (مخطط جانت)

المرحلة	أسبوع 1-2	أسبوع 3-4	أسبوع 5-6	أسبوع 7-9	أسبوع 10-12	أسبوع 13-14
جمع المتطلبات وتحليلها	✓					
نمذجة النظام (UML)		✓				
التصميم المعماري وقاعدة البيانات		✓	✓			
تطوير الواجهة الخلفية (Laravel)			✓	✓		
تطوير الواجهة الأمامية (Blade/Tailwind)				✓	✓	
الاختبار والتقييم					✓	✓
كتابة التقرير النهائي						✓

2.3 إدارة المخاطر

جدول 3. سجل إدارة المخاطر

الخطر	الاحتمال	التأثير	خطة المعالجة
تأخر إعداد بيئة التطوير المحلية (XAMPP/MySQL/Node)	متوسط	متوسط	توثيق خطوات التثبيت والتحقق منها مبكرًا قبل بدء البرمجة الفعلية.
تضارب إصدارات الحزم (npm/composer)	منخفض	متوسط	تثبيت الإصدارات في composer.lock وpackage-lock.json وعدم تحديثها عشوائيًا.
فقدان بيانات قاعدة البيانات أثناء التطوير	منخفض	مرتفع	استخدام Seeders وMigrations قابلة لإعادة التشغيل بدلًا من إدخال بيانات يدوي دائم.
ضيق الوقت لإكمال جميع الميزات	متوسط	مرتفع	ترتيب الميزات حسب أولوية MoSCoW والتركيز أولًا على متطلبات Must.

2.4 فريق العمل وتوزيع المهام والأدوار

نُفِّذ هذا المشروع من قبل طالب واحد يتولى جميع الأدوار: تحليل المتطلبات، تصميم قاعدة البيانات، تطوير الواجهة الخلفية والأمامية، الاختبار، والتوثيق، تحت إشراف مباشر من عضو هيئة التدريس المشرف على المشروع الذي يراجع التقدم دوريًا ويقدّم الملاحظات.

2.5 إدارة الجودة في المشروع

اعتُمد لضمان جودة المنتج ما يلي: مراجعة ذاتية للكود بعد كل مرحلة تطوير، اختبار يدوي لكل ميزة فور إتمامها على البيئة المحلية، الالتزام ببنية Laravel القياسية (Controllers/Models/Migrations منفصلة) لتسهيل الصيانة، واستخدام أسماء دلالية واضحة للمتغيرات والدوال والمسارات.

2.6 أدوات إدارة المشروع المستخدمة

اعتُمدت أدوات بسيطة مناسبة لمشروع فردي: قائمة مهام أسبوعية (Task List) لتتبع التقدم، إلى جانب نظام التحكم بالإصدارات Git لتتبع تاريخ التعديلات على الكود.

2.7 آليات المراقبة والتقييم المستمر

جرت متابعة تقدم المشروع عبر اجتماعات دورية مع المشرف لعرض ما أُنجز في كل مرحلة، إلى جانب اختبار يدوي متكرر لكل ميزة عند إضافتها للتأكد من عدم كسر الوظائف السابقة (اختبار انحداري مبسّط).

2.8 إدارة تشكيلة البرمجيات (SCM)

استُخدم نظام Git للتحكم بالإصدارات، مع الاحتفاظ بملفات .gitignore لاستبعاد المجلدات المولّدة تلقائيًا مثل /vendor و/node_modules وملفات البيئة الحساسة .env من الأرشفة، حفاظًا على أمان بيانات الاتصال بقاعدة البيانات.

الفصل الثالث: الإطار النظري والدراسات السابقة

3.1 الدراسات السابقة في مجال النظم الذكية والمشابهة

استعرض المشروع عددًا من المنصات القائمة ذات الصلة بمجال قراءة الكتب ومشاركة المحتوى الأدبي، بغرض تحديد نقاط القوة التي يمكن الاستفادة منها والثغرات التي يمكن معالجتها:

- Goodreads: منصة عالمية رائدة للتقييمات والمراجعات الأدبية، تتميز بقاعدة مستخدمين ضخمة ونظام توصية متقدم، لكنها لا تتيح قراءة الكتب مباشرة داخل المنصة ولا تركّز على المحتوى العربي.
- Wattpad: منصة تتيح للكتّاب نشر أعمالهم الأدبية والقراءة المباشرة أونلاين مع تفاعل مجتمعي قوي (تعليقات مضمّنة داخل النص)، إلا أن بنيتها معقدة نسبيًا وتستهدف بالدرجة الأولى الأدب التفاعلي (Fan fiction).
- منصة أبجد: منصة عربية معروفة لتقييم الكتب ومشاركة المراجعات، تشبه Goodreads من حيث الفكرة لكنها لا تتيح قراءة الكتاب داخل المنصة نفسها.
- Scribd: منصة اشتراك لقراءة الكتب والمستندات، تركّز على إتاحة المحتوى مقابل اشتراك مدفوع، مع تفاعل مجتمعي محدود جدًا مقارنة بالمنصات الاجتماعية.
- مبادرات المكتبات الرقمية الجامعية العربية: أنظمة داخلية لعرض المراجع الأكاديمية، تفتقر عمومًا لعنصر التفاعل الاجتماعي (تعليقات وتقييمات) الذي يميّز منصات القراءة الحديثة.

يُلاحَظ من هذه الدراسات أن المنصات الأجنبية إما تفصل بين القراءة والتفاعل، أو تُعقّد تجربة النشر للكاتب الناشئ، بينما تفتقر أغلب البدائل العربية لعنصر القراءة المباشرة داخل المنصة.

3.2 التقنيات والأطر الحديثة في المجال

من أبرز الأطر والتقنيات الحديثة المستخدمة في بناء منصات ويب مشابهة: Laravel وDjango وSpring Boot لجهة الخلفية، وReact وVue لجهة الواجهة الأمامية عند الحاجة لتطبيقات صفحة واحدة (SPA)، إلى جانب Tailwind CSS لتسريع تصميم الواجهات. اعتمد هذا المشروع على Laravel لجهة الخلفية نظرًا لنضجه في التعامل مع المصادقة وقواعد البيانات العلائقية، مقترنًا بـ Tailwind CSS وVite لجهة الواجهة، وهو ما فُصِّل اختياره في الفصل السادس.

3.3 التحليل المقارن للأنظمة والمنتجات الحالية

جدول 4. التحليل المقارن مع الأنظمة المشابهة

النظام	قراءة داخل المنصة	تقييم/تعليق	نشر من الكتّاب	دعم عربي كامل
Goodreads	لا	نعم	لا	جزئي
Wattpad	نعم	نعم	نعم	جزئي
أبجد	لا	نعم	لا	نعم
Read Loop (المشروع الحالي)	نعم	نعم	نعم	نعم

3.4 الفجوة البحثية والابتكار المقترح

تكمن الفجوة في غياب منصة عربية موحّدة تجمع بين ثلاث ميزات معًا: القراءة المباشرة داخل المنصة، والتفاعل الاجتماعي الكامل (تعليقات وتقييمات)، وإتاحة النشر المباشر للكتّاب دون وسيط. يقترح هذا المشروع سدّ هذه الفجوة عبر منصة Read Loop التي تجمع الميزات الثلاث ضمن معمارية بسيطة قابلة للتوسعة، مع دعم كامل للغة العربية في الواجهة والمحتوى.

3.5 الإطار النظري والدراسات المرجعية

يستند التصميم المعماري للنظام إلى نمط MVC القائم على فصل البيانات (Model) عن منطق العرض (View) وعن التحكم بتدفق الطلبات (Controller)، وهو نمط راسخ في هندسة البرمجيات لتحسين قابلية الصيانة والتوسعة (Sommerville, 2016). أما تصميم قاعدة البيانات فيستند إلى مبادئ التطبيع العلائقي (Relational Normalization) لتفادي تكرار البيانات وضمان تكامل المراجع بين الجداول (users، books، comments، ratings) عبر المفاتيح الأجنبية.

3.6 التوجهات الحديثة والمستقبلية في المجال

تتجه منصات المحتوى الأدبي حديثًا نحو دمج تقنيات الذكاء الاصطناعي لأغراض التوصية الشخصية بالكتب (Recommendation Systems)، ومعالجة اللغة الطبيعية (NLP) لتلخيص المراجعات أو تصنيف الكتب تلقائيًا حسب النوع الأدبي. يمثل ذلك امتدادًا طبيعيًا محتملًا لهذا المشروع مستقبلًا، كما هو مفصّل في مقترحات الفصل الثامن.

الفصل الرابع: تحليل المتطلبات ونمذجة النظام

4.1 دراسة الجدوى

الجدوى الفنية

تتوفر جميع المكتبات والأدوات اللازمة لتنفيذ المشروع مجانًا ومفتوحة المصدر (Laravel، Tailwind CSS، MySQL)، وتعمل على جهاز حاسوب شخصي عادي (Windows + XAMPP)، دون الحاجة لعتاد خاص، كما أن المهارات المطلوبة (PHP، JavaScript، SQL) ضمن نطاق مقررات القسم.

الجدوى الاقتصادية

التكلفة التقريبية للتطوير محدودة بكلفة الاستضافة عند النشر فقط (خطط مجانية أو منخفضة التكلفة على منصات مثل Railway)، دون أي تكلفة ترخيص للبرمجيات المستخدمة كونها مفتوحة المصدر بالكامل.

الجدوى التشغيلية

تعتمد المنصة على واجهة ويب بسيطة لا تتطلب تدريبًا خاصًا للاستخدام، وتتوافق مع سير عمل القارئ والكاتب العاديين دون تغيير جذري في عاداتهما الرقمية.

4.2 جمع المتطلبات

اعتُمد في جمع المتطلبات على تحليل الوثائق (مراجعة منصات مشابهة كما ورد في الفصل الثالث) بالإضافة إلى نقاش مباشر مع المشرف لتحديد الحد الأدنى من الميزات الجوهرية (Must-have) اللازمة لمشروع فصلي، مقارنة بالميزات التي يمكن تأجيلها لمراحل تطوير لاحقة. وُثِّق كل متطلب مباشرة بمصدره (مقابلة/مراجعة/قرار تصميمي) ضمن مصفوفة تتبع المتطلبات الواردة في الملحق أ-4.

4.3 تحديد أصحاب المصلحة وتحليل احتياجاتهم

احتياجاته وتوقعاته	صاحب المصلحة
واجهة بسيطة لتصفح الكتب وقراءتها وتقييمها والتعليق عليها دون تعقيد.	القارئ (User)
إمكانية نشر أعماله وتعديلها ومتابعة تفاعل القراء معها.	الكاتب (Author)
أدوات لإدارة المستخدمين والمحتوى وضبط إساءة الاستخدام (حظر/حذف).	مدير النظام (Admin)

احتياجاته وتوقعاته	صاحب المصلحة
كود موثّق وبنية واضحة قابلة للتوسعة دون إعادة هيكلة جذرية.	المطوّر المستقبلي
توثيق شامل يعكس تطبيق دورة حياة تطوير برمجيات كاملة وسليمة.	المشرف الأكاديمي

4.4 المتطلبات الوظيفية

جدول 5. المتطلبات الوظيفية

المجموعة الوظيفية	الوصف	الرمز
المصادقة	يجب أن يتمكن المستخدم من إنشاء حساب جديد (تسجيل).	FR1
المصادقة	يجب أن يتمكن المستخدم من تسجيل الدخول باستخدام البريد الإلكتروني وكلمة المرور.	FR2
المصادقة	يجب أن يتمكن المستخدم المسجَّل من تسجيل الخروج.	FR3
التصفح والبحث	يجب أن يعرض النظام قائمة الكتب المنشورة قابلة للتصفح مع تقسيم صفحات.	FR4
التصفح والبحث	يجب أن يتيح النظام البحث عن كتاب باستخدام كلمات مفتاحية.	FR5
القراءة والتحميل	يجب أن يتمكن المستخدم من قراءة الكتاب أونلاين مباشرة داخل المنصة.	FR6
التفاعل المجتمعي	يجب أن يتمكن المستخدم من إضافة تعليق على كتاب.	FR7
التفاعل المجتمعي	يجب أن يتمكن المستخدم المسجَّل (غير المدير) من إضافة أو تحديث تقييم من 1 إلى 5 لكتاب.	FR8
إدارة المحتوى (الكاتب)	يجب أن يتمكن الكاتب من نشر كتاب جديد.	FR9
إدارة المحتوى (الكاتب)	يجب أن يتمكن الكاتب من تعديل بيانات كتاب نشره.	FR10
إدارة المحتوى (الكاتب)	يجب أن يتمكن الكاتب أو المدير من حذف كتاب.	FR11
القراءة والتحميل	يجب أن يتيح النظام تحميل ملف الكتاب عند توفره.	FR12
التصفح والبحث	يجب أن يعرض النظام تفاصيل الكتاب (المؤلف، الوصف، متوسط التقييم، التعليقات).	FR13
إدارة النظام (المدير)	يجب أن يتمكن مدير النظام من إدارة المستخدمين (تغيير الدور، حظر، فك حظر، حذف).	FR14
إدارة النظام (المدير)	يجب أن يتمكن مدير النظام من إدارة التعليقات (حذف تعليقات مخالفة).	FR15

المجموعة الوظيفية	الوصف	الرمز
إدارة النظام (المدير)	يجب أن يمنع النظام مدير النظام من حذف نفسه أو حذف/تنزيل رتبة آخر حساب مدير في النظام.	FR16
الحساب الشخصي	يجب أن يتيح النظام للمستخدم تعديل بيانات حسابه وتغيير كلمة المرور.	FR17
ميزات مساعدة	يجب أن يتيح النظام تبديل لغة الواجهة (عربي/إنجليزي).	FR18
ميزات مساعدة	يجب أن يعرض النظام إحصائيات مبسّطة للكاتب حول كتبه المنشورة.	FR19

4.5 المتطلبات غير الوظيفية

جدول 6. المتطلبات غير الوظيفية

المتطلب القابل للقياس	الفئة
زمن تحميل صفحة قائمة الكتب أقل من 2 ثانية على بيئة تطوير محلية عادية.	الأداء (Performance)
تشفير كلمات المرور عبر Bcrypt، وحماية النماذج من هجمات CSRF، وتنقية المخرجات من XSS.	الأمان (Security)
يمكن لمستخدم جديد إنجاز تسجيل الدخول وتصفح كتاب أول خلال 3 خطوات كحد أقصى.	قابلية الاستخدام (Usability)
استرجاع رسائل خطأ واضحة عند إدخال بيانات غير صحيحة بدل تعطل النظام.	الاعتمادية (Reliability)
فصل واضح بين طبقات Models وControllers وViews يسمح بتعديل كل طبقة دون التأثير المباشر على الأخرى.	قابلية الصيانة (Maintainability)
دعم عرض سليم على أحدث إصدارات المتصفحات الشائعة (Firefox، Chrome، Edge).	التوافقية (Compatibility)
دعم التشغيل على قاعدتي بيانات بديلتين (MySQL وSQLite) دون تعديل جوهري في الكود.	قابلية النقل (Portability)
دعم كامل لواجهتين لغويتين (عربي/إنجليزي) قابلتين للتبديل الفوري دون إعادة تحميل كامل للجلسة.	التوطين وإمكانية الوصول (Localization)
بنية قاعدة بيانات علائقية مطبَّعة تسمح بإضافة جداول/ميزات جديدة (كنظام التوصية) دون إعادة هيكلة الجداول القائمة.	قابلية التوسع (Scalability)

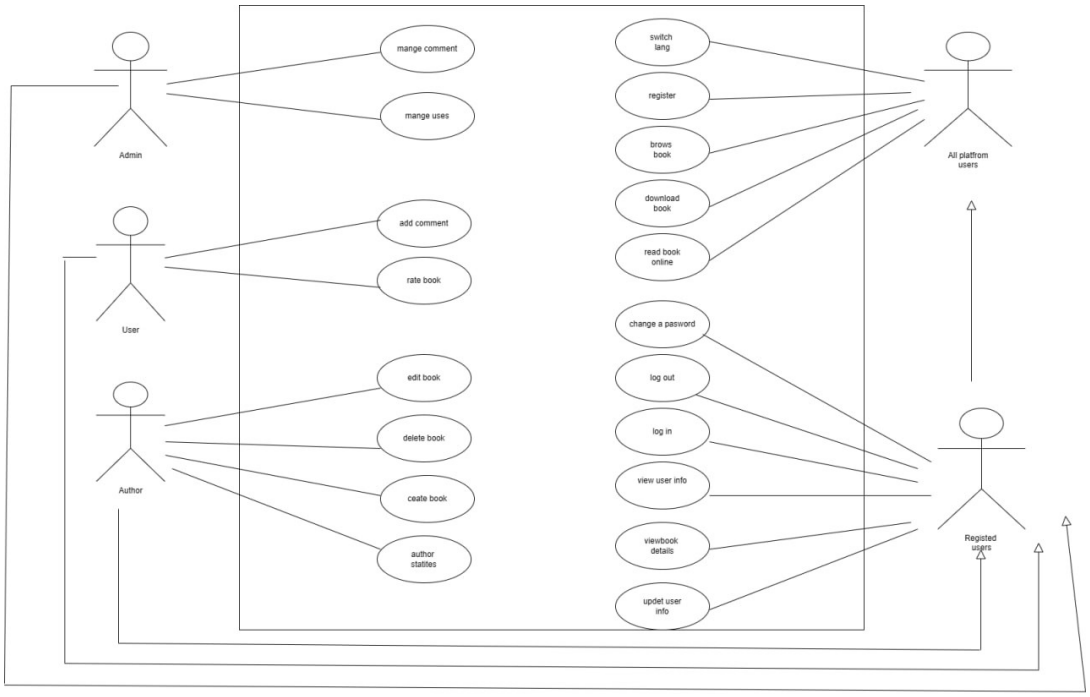
4.6 قائمة أولويات المتطلبات (MoSCoW)

جدول 7. أولويات المتطلبات (MoSCoW)

الأولوية	المتطلبات المندرجة تحتها
Must (يجب)	FR1–FR9، FR13، FR14، FR16 — التسجيل، الدخول، تصفح وقراءة الكتب، التعليق والتقييم، نشر الكتب، إدارة المستخدمين الأساسية.
Should (ينبغي)	FR10، FR11، FR15، FR17 — تعديل/حذف الكتب، إدارة التعليقات، تعديل الحساب.
Could (يمكن)	FR12، FR18، FR19 — تحميل ملف الكتاب، تبديل اللغة، إحصائيات الكاتب.
Won't (لن)	الدفع الإلكتروني، تطبيق الجوال الأصلي، محرك توصية بالذكاء الاصطناعي — خارج نطاق المرحلة الحالية.

4.7 مخططات UML للمتطلبات — مخطط وتوصيف لكل متطلب

يعرض هذا القسم أولًا مخطط حالات الاستخدام العام، ثم يفصّل لكل مجموعة من المتطلبات الوظيفية (وفق التصنيف الوارد في الجدول 5) وصف حالة استخدام مختصر مصحوبًا بمخطط سلوكي خاص بها (نشاط أو تسلسل)، لضمان تتبّع كامل بين كل متطلب وتمثيله البصري في النموذج.

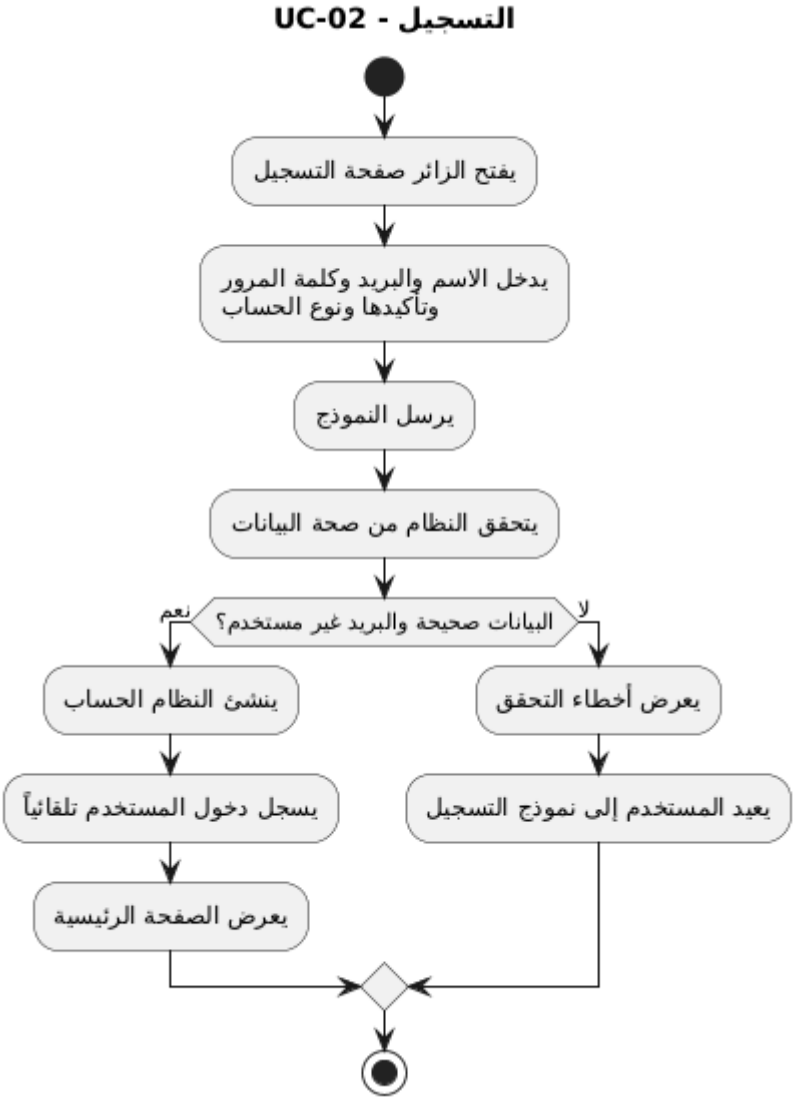


شكل 1. مخطط حالات الاستخدام العام للنظام

4.7.1 المصادقة: إنشاء حساب وتسجيل الدخول والخروج (FR1–FR3)

العنصر	الوصف
الفاعلون	زائر (لإنشاء حساب)، مستخدم مسجَّل (للدخول والخروج)

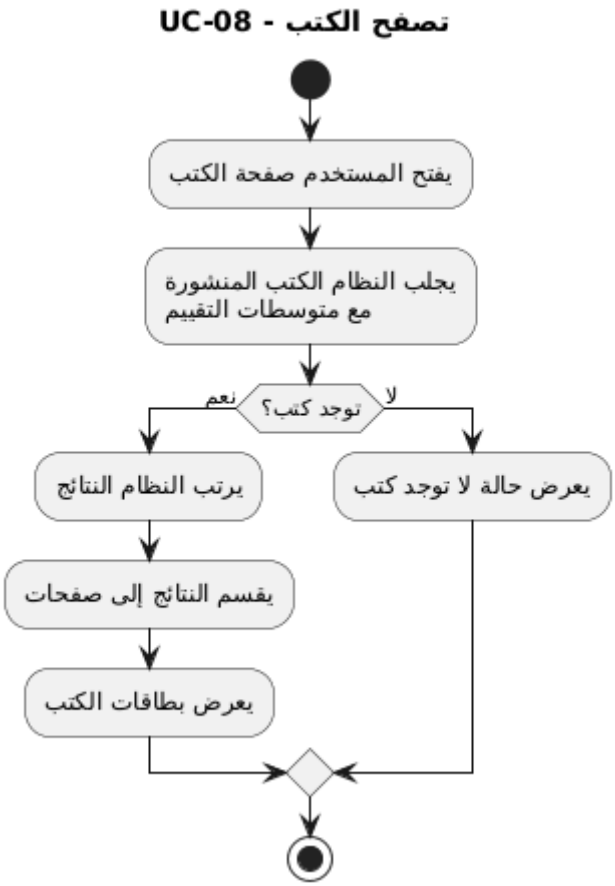
العنصر	الوصف
الشروط المسبقة	بريد إلكتروني غير مستخدم سابقًا عند التسجيل؛ حساب موجود وغير محظور عند الدخول
المسار الأساسي	إدخال البيانات ← تحقق النظام من صحتها ← إنشاء الحساب أو فتح الجلسة ← توجيه المستخدم للصفحة الرئيسية
المسار البديل	بريد مكرر عند التسجيل، أو بيانات دخول خاطئة، أو حساب محظور: عرض رسالة خطأ دون تنفيذ العملية
الأولوية	Must



شكل 2. مخطط نشاط: تسجيل حساب جديد وتسجيل الدخول

4.7.2 التصفح والبحث وعرض تفاصيل الكتاب (FR4–FR5، FR13)

العنصر	الوصف
الفاعلون	جميع مستخدمي المنصة
الشروط المسبقة	وجود كتب منشورة بحالة (published) في قاعدة البيانات
المسار الأساسي	فتح صفحة الكتب ← إدخال كلمة بحث اختيارية ← عرض النتائج المطابقة مع تفاصيل كل كتاب (المؤلف، الوصف، متوسط التقييم)
المسار البديل	لا توجد نتائج مطابقة: عرض رسالة توضح عدم وجود نتائج
الأولوية	Must

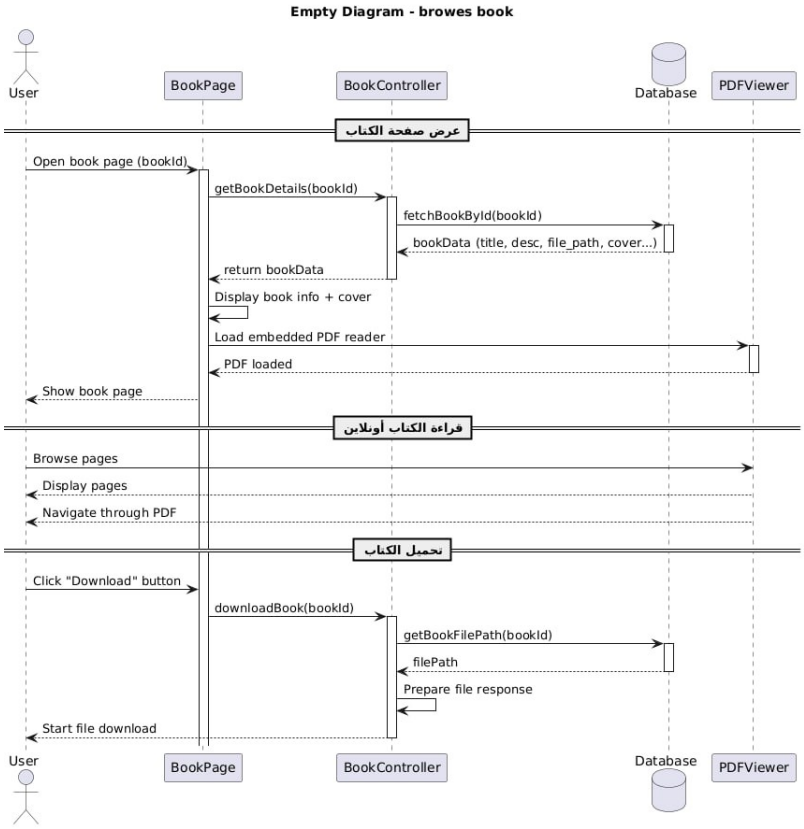


شكل 3. مخطط نشاط: البحث عن كتاب وتصفحه

4.7.3 قراءة الكتاب أونلاين وتحميله (FR6، FR12)

العنصر	الوصف
الفاعلون	مستخدم مسجَّل أو زائر (حسب إعداد إتاحة الكتاب)

العنصر	الوصف
الشروط المسبقة	الكتاب منشور ويحتوي مسار ملف صالح (file_path)
المسار الأساسي	فتح صفحة الكتاب ← جلب بياناته من قاعدة البيانات ← عرض عارض PDF المضمَّن للقراءة أونلاين ← أو الضغط على «تحميل» لبدء تنزيل الملف
المسار البديل	عدم توفر ملف للكتاب: إخفاء زر التحميل وعرض القراءة أونلاين فقط إن وُجد محتوى مصاحب
الأولوية	Must (القراءة) / Could (التحميل)

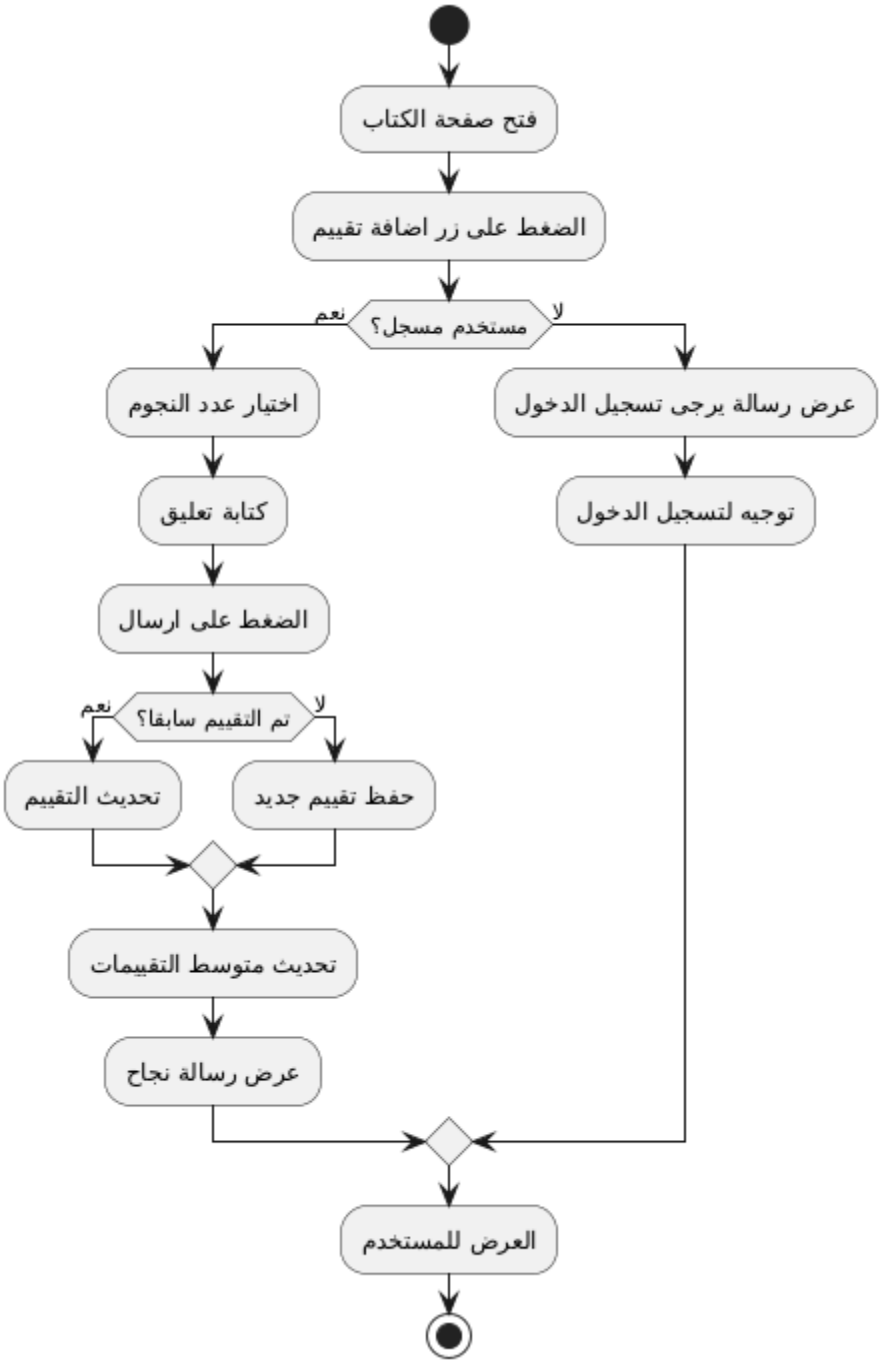


شكل 4. مخطط تسلسل: قراءة الكتاب أونلاين وتحميله

4.7.4 إضافة تقييم لكتاب (FR8)

العنصر	الوصف
الفاعلون	مستخدم مسجَّل، كاتب (باستثناء حساب مدير)
الشروط المسبقة	تسجيل دخول فعّال، الحساب غير محظور وليس من نوع admin
المسار الأساسي	اختيار درجة من 1 إلى 5 ← تحقق النظام من صلاحية الدرجة ← إنشاء تقييم جديد أو تحديث تقييم قائم لنفس المستخدم/الكتاب ← تحديث متوسط تقييم الكتاب

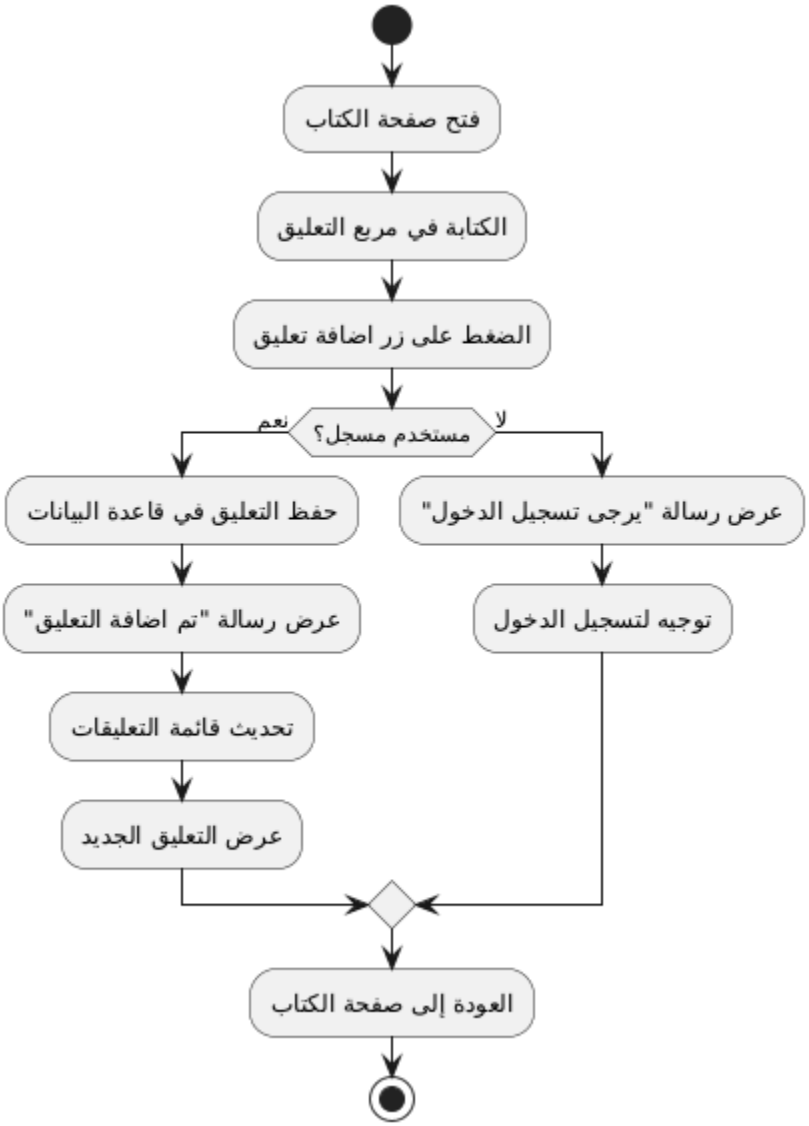
العنصر	الوصف
المسار البديل	درجة خارج المدى أو حساب مدير: رفض الطلب وعرض رسالة توضيحية
الأولوية	Must



شكل 5. مخطط نشاط: إضافة تقييم لكتاب

4.7.5 إضافة تعليق على كتاب (FR7)

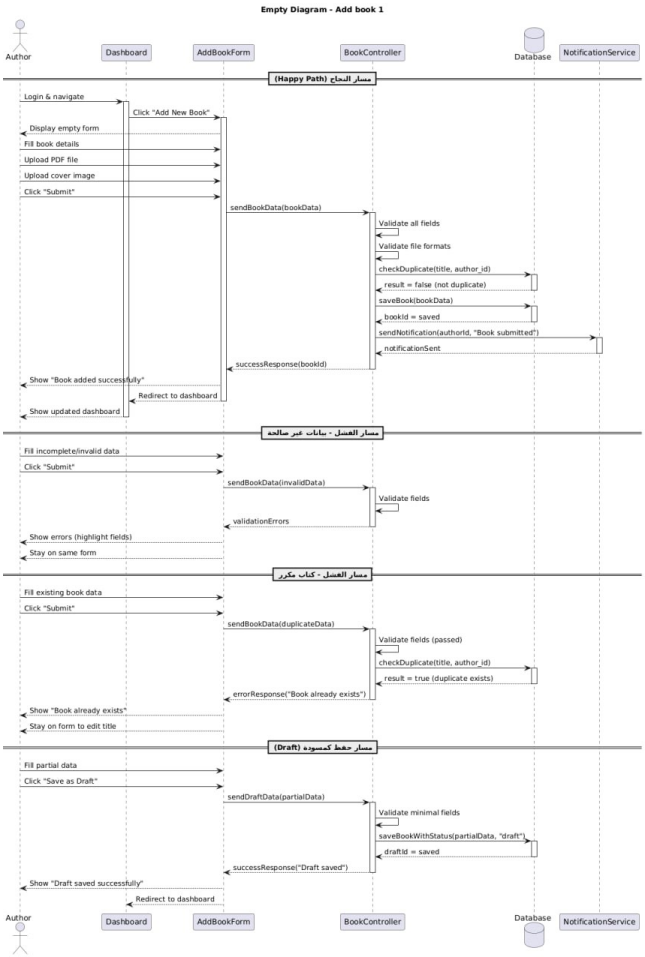
العنصر	الوصف
الفاعلون	مستخدم مسجَّل، كاتب
الشروط المسبقة	تسجيل دخول فعّال والحساب غير محظور
المسار الأساسي	كتابة نص التعليق ← تحقق النظام من عدم كونه فارغًا ← حفظ التعليق ← عرضه ضمن قائمة تعليقات الكتاب فورًا
المسار البديل	نص فارغ أو حساب محظور: رفض الطلب وعرض رسالة رفض
الأولوية	Must



شكل 6. مخطط نشاط: إضافة تعليق على كتاب

4.7.6 نشر كتاب جديد (FR9)

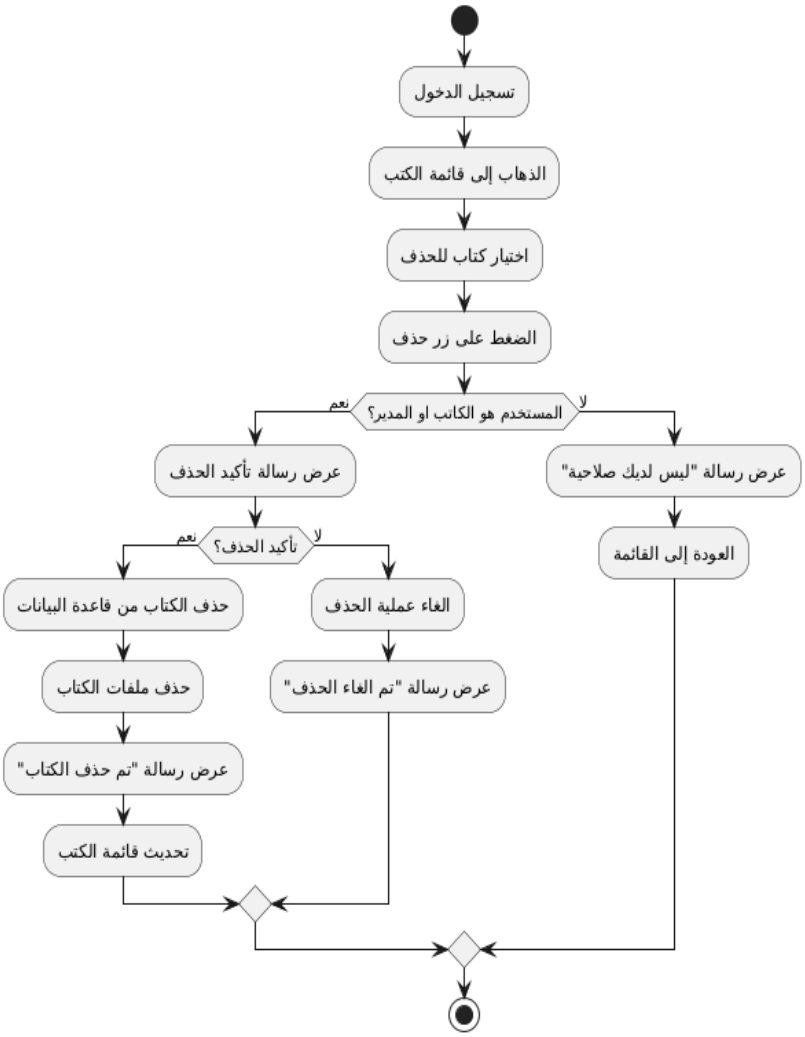
العنصر	الوصف
الفاعلون	الكاتب (Author)
الشروط المسبقة	تسجيل دخول بحساب من نوع كاتب وغير محظور
المسار الأساسي	فتح نموذج «إضافة كتاب» ← إدخال العنوان والوصف ورفع الملف/الغلاف ← تحقق النظام من الحقول ومن عدم وجود كتاب مكرر لنفس الكاتب ← إنشاء سجل الكتاب
المسار البديل	حقول ناقصة، أو ملف بصيغة غير مدعومة، أو كتاب مكرر: رفض الطلب وعرض رسالة الخطأ المناسبة، مع إمكانية الحفظ كمسوّدة (Draft)
الأولوية	Must



شكل 7. مخطط تسلسل: نشر كتاب جديد

4.7.7 تعديل أو حذف كتاب (FR10–FR11)

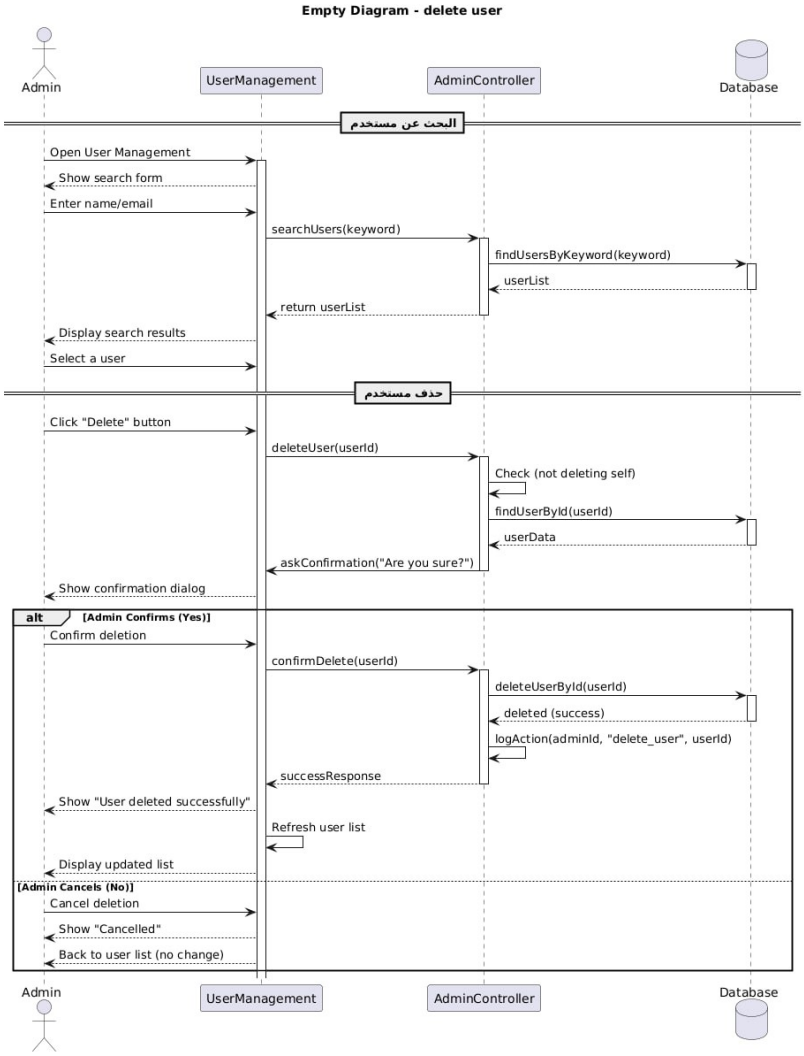
العنصر	الوصف
الفاعلون	الكاتب (لكتبه فقط)، مدير النظام (لأي كتاب)
الشروط المسبقة	الكتاب يخص الكاتب الحالي، أو أن الحساب من نوع مدير
المسار الأساسي	اختيار كتاب من لوحة الكاتب ← اختيار تعديل أو حذف ← تحقق النظام من ملكية الكتاب أو صلاحية المدير ← تنفيذ العملية وتحديث/حذف السجل
المسار البديل	محاولة تعديل أو حذف كتاب لا يخص الكاتب الحالي (وليس مديرًا): رفض العملية
الأولوية	Should



شكل 8. مخطط نشاط: تعديل أو حذف كتاب

4.7.8 إدارة المستخدمين: تغيير الدور، الحظر، فك الحظر، والحذف (FR14، FR16)

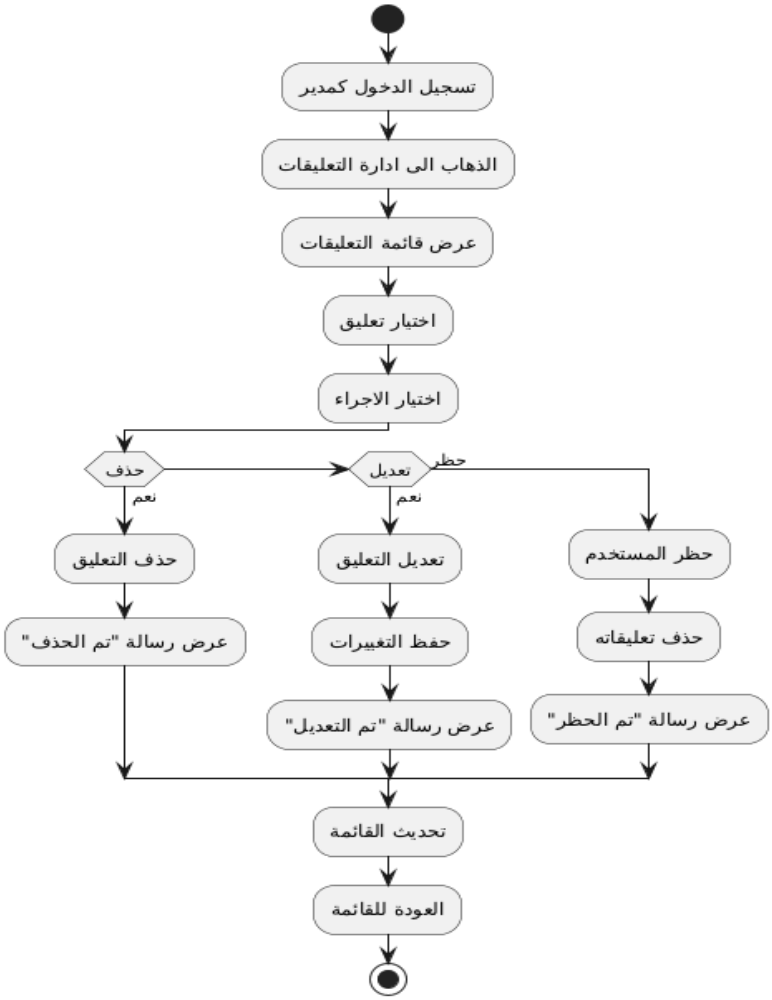
العنصر	الوصف
الفاعلون	مدير النظام (Admin)
الشروط المسبقة	تسجيل دخول بحساب مدير
المسار الأساسي	فتح لوحة إدارة المستخدمين ← البحث عن مستخدم ← اختيار إجراء (حظر/فك حظر/تغيير دور/حذف) ← تحقق النظام من كون الهدف ليس المدير نفسه وليس آخر مدير متبقٍ ← تنفيذ الإجراء
المسار البديل	الإجراء = حذف الحساب ذاته أو حذف آخر مدير متبقٍ: رفض العملية وعرض تنبيه صريح
الأولوية	Must



شكل 9. مخطط تسلسل: حذف مستخدم من قبل المدير

4.7.9 إدارة التعليقات من قبل المدير (FR15)

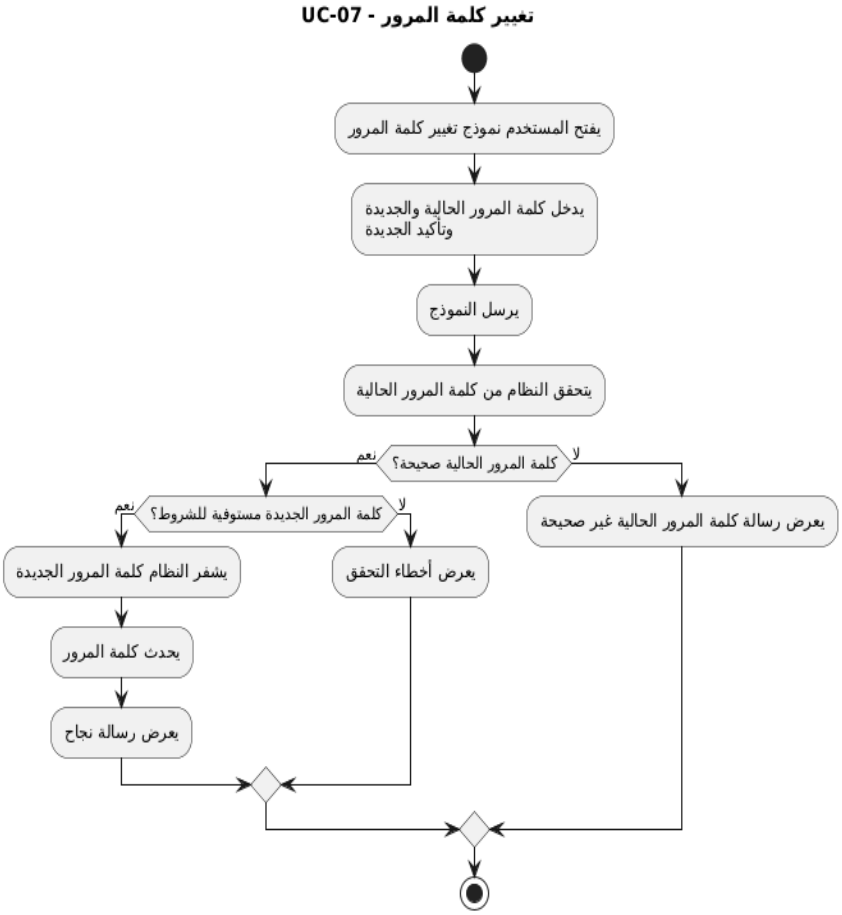
العنصر	الوصف
الفاعلون	مدير النظام (Admin)
الشروط المسبقة	تسجيل دخول بحساب مدير
المسار الأساسي	فتح لوحة إدارة التعليقات ← مراجعة التعليقات المبلّغ عنها أو المشتبه بمخالفتها ← حذف التعليق المخالف أو الإبقاء عليه
المسار البديل	—
الأولوية	Should



شكل 11. مخطط نشاط: إدارة التعليقات من قبل المدير

4.7.10 تعديل الحساب الشخصي وكلمة المرور (FR17)

العنصر	الوصف
الفاعلون	أي مستخدم مسجَّل (قارئ، كاتب، مدير)
الشروط المسبقة	تسجيل دخول فعّال
المسار الأساسي	فتح صفحة الحساب الشخصي ← تعديل الاسم أو كلمة المرور ← تحقق النظام من صحة كلمة المرور الحالية عند تغييرها ← تحديث بيانات الحساب
المسار البديل	كلمة المرور الحالية غير صحيحة: رفض التحديث وعرض رسالة خطأ
الأولوية	Should

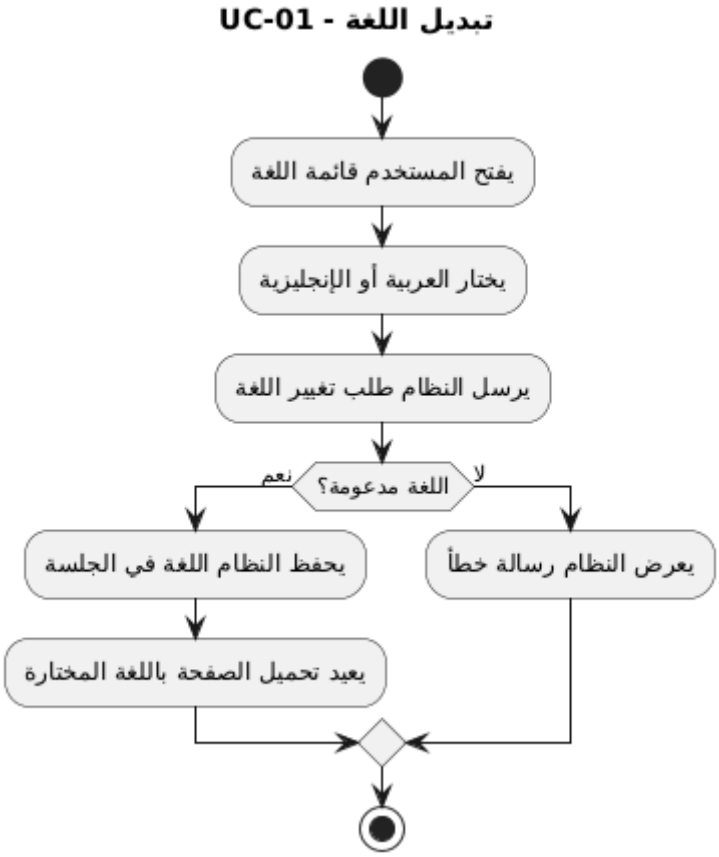


شكل 12. مخطط نشاط: تعديل الحساب الشخصي وكلمة المرور

4.7.11 تبديل لغة الواجهة وعرض إحصائيات الكاتب (FR18–FR19)

الوصف	العنصر
جميع المستخدمين (تبديل اللغة)، الكاتب (الإحصائيات)	الفاعلون
لا يوجد شرط خاص لتبديل اللغة؛ تسجيل دخول بحساب كاتب لعرض الإحصائيات	الشروط المسبقة

العنصر	الوصف
المسار الأساسي	اختيار مبدّل اللغة ← إعادة عرض الواجهة فورًا باللغة المختارة / فتح لوحة الكاتب ← حساب عدد الكتب والتقييمات والتعليقات لكل كتاب ← عرضها
المسار البديل	—
الأولوية	Could



شكل 13. مخطط نشاط: تبديل اللغة وعرض إحصائيات الكاتب

4.8 النمذجة حسب المنهجية المستخدمة

بما أن المنهجية المتبعة تكرارية تزايدية، جرى تنظيم المتطلبات على شكل قائمة ميزات (Feature List) مجمّعة حسب كل تكرار تطويري (Iteration): التكرار الأول شمل المصادقة والتسجيل (FR1–FR3)؛ التكرار الثاني شمل تصفح الكتب والبحث والقراءة (FR4–FR6، FR12–FR13)؛ التكرار الثالث شمل التقييمات والتعليقات (FR7–FR8)؛ التكرار الرابع شمل نشر الكتب وإدارتها من قبل الكتّاب (FR9–FR11، FR19)؛ والتكرار الخامس شمل لوحة إدارة المستخدمين والتعليقات من قبل المدير (FR14–FR16) وتبديل اللغة (FR18).

4.9 تحليل وتوثيق المتطلبات

جرى ربط كل متطلب وظيفي بعنصر التصميم المقابل له وبحالة الاختبار التي تتحقق منه ضمن مصفوفة تتبع المتطلبات (RTM) الواردة كاملة في الملحق أ-4، والتي وُسِّعت لتغطي جميع المتطلبات من FR1 إلى FR19 دون استثناء، بما يضمن عدم إغفال أي متطلب أثناء التصميم أو الاختبار.

الفصل الخامس: التصميم المعماري والتفصيلي

5.1 التصميم عالي المستوى

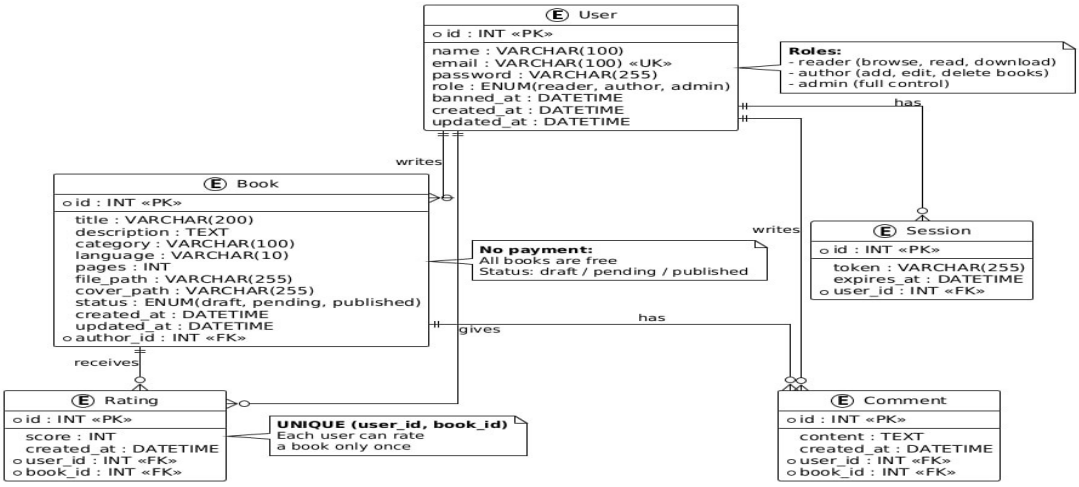
اعتُمد نمط التصميم المعماري MVC (Model-View-Controller) لتنظيم بنية النظام على النحو التالي:

- النموذج (Model): يمثّل البيانات وقواعد الأعمال، ويشمل الأصناف User وBook وComment وRating المرتبطة مباشرة بجداول قاعدة البيانات عبر ORM الخاص بـ Laravel (Eloquent).
- العرض (View): طبقة واجهة المستخدم المبنية باستخدام قوالب Blade وTailwind CSS، وهي المسؤولة عن عرض البيانات القادمة من وحدة التحكم دون احتواء منطق أعمال مباشر.
- المتحكم (Controller): يربط بين النموذج والعرض، ويستقبل طلبات المستخدم عبر المسارات (Routes)، وينفّذ التحقق من الصلاحيات والمدخلات قبل التفاعل مع النموذج.

يساعد هذا الفصل الواضح بين الطبقات على تنظيم الكود، وتسهيل الصيانة، وتحسين إمكانية التطوير المستقبلي، إذ يمكن تعديل واجهة العرض دون المساس بمنطق الأعمال أو بنية البيانات.

5.2 تصميم قاعدة البيانات

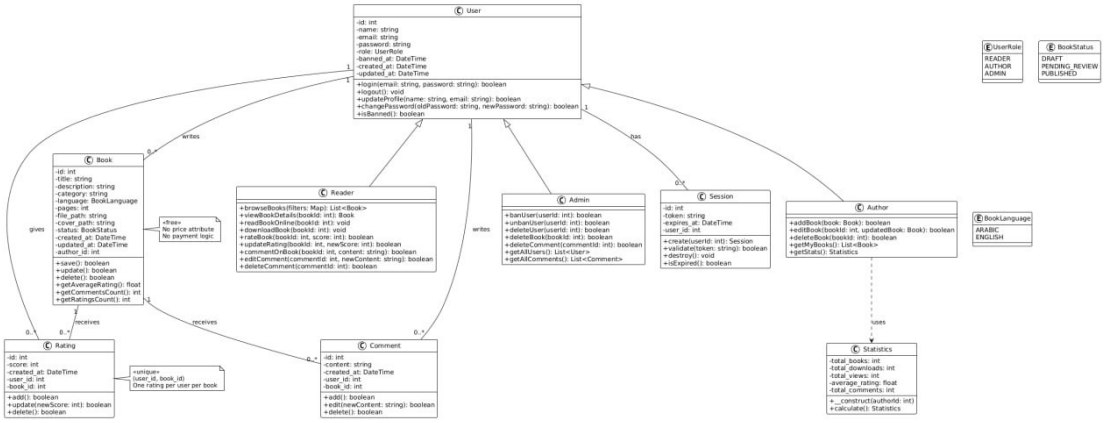
يوضح الشكل 14 نموذج الكيانات والعلاقات (ERD) الخاص بالنظام، ويبيّن العلاقات بين جدول المستخدمين والكتب



والتعليقات والتقييمات.

شكل 14. نموذج الكيانات والعلاقات (ERD)

ويعرض الشكل 15 مخطط الأصناف (Class Diagram) المقابل، والذي يفصّل خصائص كل صنف والعلاقات البرمجية بينها ضمن طبقة Models.



شكل 15. مخطط الأصناف (Class Diagram)

المخطط العلائقي (Relational Schema)

جدول 8. المخطط العلائقي لقاعدة البيانات

الجدول	أهم الحقول	القيود والمفاتيح
users	id، name، email، password، role، banned_at، email_verified_at	PK: id — UNIQUE: email — CHECK: role ∈ {user, author, admin}
books	id، user_id، title، description، file_path، created_at	PK: id — FK: user_id → users.id (الكاتب الناشر)
comments	id، user_id، book_id، body، created_at	PK: id — FK: user_id → users.id، FK: book_id → books.id
ratings	id، user_id، book_id، score	PK: id — FK مركّب فريد: (user_id, book_id) — CHECK: 1 ≤ score ≤ 5

يضمن القيد الفريد المركّب على جدول ratings (المستخدم + الكتاب) تحقيق المتطلب FR8 القاضي بألا يمتلك المستخدم أكثر من تقييم واحد نشط لكل كتاب؛ وعند إعادة التقييم يُحدَّث السجل القائم بدلًا من إدراج سجل جديد.

5.3 تصميم الواجهات (المسارات ووحدات التحكم)

بما أن النظام مبني كتطبيق Laravel متكامل (Monolithic) يعرض الواجهات عبر Blade بدلًا من واجهة REST منفصلة بالكامل، فإن التصميم يعتمد على مسارات (Routes) موزَّعة على وحدات تحكم مخصصة لكل كيان، على النحو التالي:

الوصف	وحدة التحكم	الطريقة	المسار (نموذجي)
تسجيل حساب جديد، تسجيل الدخول والخروج.	AuthController	GET/POST	register، /login/، logout/
عرض قائمة الكتب وتفاصيل كتاب معيّن.	BookController	GET	books، /books/{id}/
نشر كتاب جديد وتعديله وحذفه (للكاتب/المدير).	BookController	POST/PUT/DELETE	books/ (إنشاء)، books/{id}/ (تعديل/حذف)
إضافة أو تحديث تقييم كتاب.	RatingController	POST	books/{id}/ratings/
إضافة تعليق أو حذفه.	CommentController	POST/DELETE	books/{id}/comments/
إدارة المستخدمين (دور، حظر، حذف) — محمي بوسيط صلاحية المدير.	Admin\UserController	GET/PATCH/DELETE	admin/users/

5.4 تصميم الأمان والصلاحيات

آلية المصادقة (Authentication)

يعتمد النظام آلية مصادقة قائمة على الجلسات (Session-based Authentication) المدمجة في Laravel، حيث تُخزَّن كلمات المرور بصيغة مشفّرة عبر خوارزمية Bcrypt باستخدام واجهة Hash، ولا تُحفظ أي كلمة مرور بنص صريح في قاعدة البيانات في أي وقت.

التفويض (Authorization) وأدوار المستخدمين

يُقسَّم المستخدمون إلى ثلاثة أدوار محددة في حقل role بجدول users: مستخدم عادي (user)، كاتب (author)، ومدير نظام (admin). تُطبَّق هذه الأدوار عبر وسائط (Middleware) وسياسات صلاحية (Policies) تمنع الوصول لمسارات لوحة التحكم الإدارية إلا لأصحاب دور admin، مع قيد

إضافي صريح يمنع حذف حساب المدير الأخير المتبقي في النظام أو حذف المدير لحسابه الخاص، حفاظًا على استمرارية إدارة المنصة.

حماية البيانات

إلى جانب تشفير كلمات المرور، يعتمد النظام على الحماية المدمجة في Laravel من هجمات تزوير الطلب عبر المواقع (CSRF) عبر رمز تحقق (Token) مضمَّن في كل نموذج، وحماية أساسية من هجمات XSS عبر تنقية المخرجات المعروضة تلقائيًا ضمن قوالب Blade. كما يتضمن جدول المستخدمين حقل banned_at الذي يتيح تعطيل وصول أي حساب مخالف دون الحاجة لحذفه نهائيًا، وهو ما يحافظ على سجل النشاط لأغراض المراجعة.

الفصل السادس: بيئة العمل والتنفيذ البرمجي

6.1 الأدوات البرمجية والمكتبات المستخدمة

جدول 9. الأدوات والمكتبات المستخدمة

الأداة/المكتبة	الإصدار المستخدم	الغرض
PHP	8.2.12	لغة تشغيل الواجهة الخلفية.
Laravel	الإصدار 11 (بنية migrations أحادية الطابع الزمني)	إطار العمل الرئيسي (MVC، Eloquent ORM، المصادقة).
Composer	2.9.3	إدارة تبعيات PHP.
Node.js / npm	v24.18.1 / 11.16.0	بناء أصول الواجهة الأمامية عبر Vite.
Tailwind CSS	الإصدار الأحدث المتوافق مع Vite	تصميم الواجهات بأسلوب Utility-first.
Vite	أداة البناء الافتراضية في Laravel 11	تجميع وتحسين ملفات CSS/JS.
MySQL	عبر XAMPP (بورت 3306)	قاعدة البيانات الرئيسية بيئة التطوير/الإنتاج.
SQLite	ملف database.sqlite	قاعدة بيانات بديلة خفيفة لأغراض التطوير المحمول.
Visual Studio Code	أحدث إصدار مستقر	بيئة التطوير المتكاملة (IDE).
Git	—	التحكم بإصدارات الكود المصدري.

6.2 بيئة التطوير وبنية المشروع

جرى التطوير على نظام تشغيل Windows باستخدام حزمة XAMPP لتشغيل خادم MySQL محليًا، مع تشغيل خادم Laravel المدمج (php artisan serve) على المنفذ 8000. يوضح الشكل 16 شجرة المجلدات الرئيسية للمشروع وفق البنية القياسية لإطار Laravel.

شكل 16. شجرة مجلدات المشروع (Laravel)

```
BookForum/
├── app/
│   ├── Http/Controllers/   (AuthController, BookController, ...)
│   ├── Models/             (User, Book, Comment, Rating)
│   └── Policies/           (صلاحيات الوصول للكتب والمستخدمين)
├── database/
│   ├── migrations/         (هيكل الجداول)
│   ├── seeders/            (DatabaseSeeder.php - حساب المدير)
│   └── factories/          (UserFactory.php)
├── resources/
│   ├── views/              (قوالب Blade)
│   └── css, js             (Tailwind + Vite)
├── routes/web.php          (تعريف المسارات)
├── public/                 (نقطة الدخول index.php)
├── config/database.php     (إعدادات الاتصال بقاعدة البيانات)
└── .env                    (متغيرات البيئة، غير مضمّنة في Git)
```

6.3 لغات البرمجة وإطارات العمل

اعتُمد PHP مع إطار العمل Laravel لبناء الواجهة الخلفية، إلى جانب Blade كلغة قوالب لعرض الواجهة الأمامية مباشرة من الخادم (بنية Monolithic)، مدعومة بـ JavaScript الأساسي وTailwind CSS لتحسين التفاعلية والتصميم دون الحاجة لإطار عمل واجهة أمامية منفصل مثل React.

6.4 خادم قاعدة البيانات ونظام التشغيل

استُخدم خادم MySQL (عبر XAMPP، الإصدار المرفق بالحزمة) كقاعدة بيانات رئيسية لبيئة التطوير، إلى جانب ملف SQLite محلي (database.sqlite) كخيار بديل خفيف. جرى تطوير المشروع بالكامل على نظام Windows، مع إمكانية النشر لاحقًا على بيئة Linux سحابية (مثل Railway) تدعم PHP وMySQL بشكل قياسي.

6.5 المكتبات والتبعيات (Dependencies)

تُدار تبعيات الواجهة الخلفية عبر composer.json وcomposer.lock، وتشمل حزمة Laravel Framework الأساسية وحزم مساعدة قياسية (مثل حزمة الاختبار PHPUnit). أما تبعيات الواجهة الأمامية فتُدار عبر package.json وpackage-lock.json، وتشمل Tailwind CSS وVite وإضافة laravel-vite-plugin للتكامل بين الاثنين.

6.6 أدوات إدارة الإصدارات والتحكم المصدري

اعتُمد Git للتحكم بإصدارات الكود المصدري محليًا مع الحفاظ على تاريخ تعديلات منظم يعكس مراحل التطوير التكرارية (مصادقة، كتب، تقييمات، إدارة). يستبعد ملف .gitignore الملفات الحساسة والمجلدات المولَّدة تلقائيًا (.env، node_modules/، vendor/) من الأرشفة.

6.7 أدوات التكامل والنشر المستمر (CI/CD)

لم يُعتمد خط أنابيب نشر مستمر آلي (مثل GitHub Actions) في هذه المرحلة نظرًا لطبيعة المشروع الفردي ومحدودية الوقت المتاح؛ ويُقترح اعتماده كخطوة تحسين مستقبلية عند توسعة الفريق أو الانتقال لمرحلة تخرج لاحقة (تخرج 1 / تخرج 2).

6.8 بيئة الاختبار

جرى الاختبار يدويًا على بيئة localhost عبر متصفح الويب، بالاعتماد على خادم Laravel المحلي (php artisan serve) وقاعدة بيانات MySQL محلية عبر XAMPP، مع تنفيذ الترحيلات (Migrations) والمصدر الأولي للبيانات (Seeders) في كل مرة يُعاد فيها تهيئة البيئة.

6.9 التنفيذ البرمجي

تنفيذ طبقة الوصول إلى المعطيات (ORM)

تعتمد جميع عمليات الوصول لقاعدة البيانات على Eloquent ORM الخاص بـ Laravel، حيث يمثّل كل صنف Model جدولًا مقابلًا مع تعريف العلاقات بينها (belongsTo، hasMany) بدلًا من كتابة استعلامات SQL خام، مما يقلل مخاطر الحقن البرمجي (SQL Injection) ويحسّن قابلية القراءة.

تنفيذ طبقة الأعمال والمصادقة

يوضح المقطع التالي جزءًا تمثيليًا من ملف المصدر الأولي لقاعدة البيانات (DatabaseSeeder.php)، والذي ينشئ حساب مدير النظام تلقائيًا عند إعداد بيئة جديدة، مع تشفير كلمة المرور عبر واجهة Hash قبل التخزين:

```php
class DatabaseSeeder extends Seeder
{
    public const ADMIN_EMAIL = 'admin@gmail.com';
    public const ADMIN_PASSWORD = '********';

    public function run(): void
    {
        User::query()->updateOrCreate(
            ['email' => self::ADMIN_EMAIL],
            [
                'name' => 'مدير النظام',
                'password' => Hash::make(self::ADMIN_PASSWORD),
                'role' => 'admin',
                'email_verified_at' => now(),
                'banned_at' => null,
            ]
        );
    }
}
```

(أُخفيت كلمة المرور الفعلية في هذا المقطع لأغراض الأمان في التقرير المنشور، وتُغيَّر إلزاميًا قبل أي نشر علني للمنصة.)

تنفيذ واجهة المستخدم

بُنيت جميع الصفحات باستخدام قوالب Blade مقترنة بـ Tailwind CSS، مع تصميم متجاوب (Responsive) يدعم أحجام الشاشات المختلفة، وشريط تنقّل علوي موحّد يتضمن روابط الصفحة الرئيسية والكتب، وأزرار تسجيل الدخول والتسجيل، إلى جانب مبدّل لغة ومبدّل للمظهر الفاتح/الداكن (Light/Dark Mode).

الفصل السابع: الاختبار والتقييم

7.1 خطة الاختبار

جرى اختبار النظام يدويًا على بيئة محلية (localhost) عقب إتمام كل تكرار تطويري، إذ يتولى الطالب نفسه تنفيذ الاختبارات فور إضافة كل ميزة، باستخدام متصفح الويب لمحاكاة سيناريوهات المستخدم الحقيقية. يرد نموذج خطة الاختبار الكامل بالتفصيل في الملحق ج.

7.2 أنواع الاختبارات

- اختبار الوحدة (يدوي): التحقق من عمل كل دالة تحكم بشكل منفرد (مثل إنشاء تقييم، إنشاء كتاب).
- اختبار التكامل: التحقق من تفاعل وحدة التحكم مع النموذج وقاعدة البيانات معًا (كإنشاء حساب مدير عبر Seeder وتسجيل الدخول به مباشرة).
- اختبار النظام: تنفيذ سيناريوهات المستخدم كاملة من الواجهة (تسجيل حساب ← تصفح ← تقييم ← تعليق).
- اختبار الأمان الأساسي: التحقق من رفض الوصول لمسارات لوحة التحكم الإدارية من حساب غير مدير، ومن منع حذف آخر حساب مدير.
- اختبار قابلية الاستخدام: مراجعة وضوح رسائل الخطأ وسهولة الوصول للوظائف الأساسية خلال عدد محدود من النقرات.

7.3 مخططات الاختبار ونتائجه

جدول 10. حالات الاختبار ونتائجها

رقم الحالة	الوصف	المدخلات	النتيجة المتوقعة	الحالة
TC-01	تشغيل ترحيلات قاعدة البيانات (migrate)	أمر php artisan migrate	إنشاء جميع الجداول (users, books, comments, ratings, cache, jobs) دون أخطاء	✓ تم
TC-02	تشغيل المصدر الأولي للبيانات (seed)	أمر php artisan db:seed	إنشاء حساب مدير النظام بالبريد admin@gmail.com	✓ تم

رقم الحالة	الوصف	المدخلات	النتيجة المتوقعة	الحالة
TC-03	تشغيل الخادم المحلي وعرض الصفحة الرئيسية	أمر php artisan serve وفتح 127.0.0.1:8000	عرض الصفحة الرئيسية بنجاح مع شريط تنقّل كامل	تم ✓
TC-04	تسجيل الدخول بحساب المدير	admin@gmail.com + كلمة المرور الصحيحة	توجيه المستخدم إلى الصفحة الرئيسية موثّقًا كمدير	قيد التنفيذ
TC-05	تسجيل الدخول ببيانات خاطئة	بريد إلكتروني صحيح + كلمة مرور خاطئة	عرض رسالة خطأ وعدم تسجيل الدخول	مخطط
TC-06	إضافة تقييم كتاب بدرجة غير صالحة	درجة تقييم = 7	رفض الطلب وعرض رسالة تحقق (Validation)	مخطط
TC-07	محاولة حذف آخر حساب مدير	طلب حذف من لوحة تحكم مدير	رفض العملية والحفاظ على الحساب	مخطط
TC-08	تحميل صفحة قائمة الكتب	فتح مسار /books	زمن استجابة أقل من 2 ثانية على البيئة المحلية	مخطط

ملاحظة: الحالات TC-01 إلى TC-03 نُفِّذت فعليًا وتحققت نتائجها أثناء إعداد بيئة التشغيل المحلية؛ أما الحالات المتبقية فمجدولة ضمن جولة الاختبار الشاملة السابقة لتسليم التقرير النهائي، وتُحدَّث أعمدة «الحالة» عند إتمامها فعليًا قبل المناقشة.

7.4 نتائج الاختبار وتحليلها

من أصل 8 حالات اختبار مخطَّطة، أُنجزت 3 حالات بنجاح كامل تغطي التهيئة الأساسية للنظام (الترحيل، المصدر الأولي للبيانات، تشغيل الخادم وعرض الواجهة)، ولم تُسجَّل أي حالة فشل حتى تاريخ إعداد هذا التقرير. يوصى بإكمال باقي حالات الاختبار (خصوصًا اختبارات الأمان TC-06 وTC-07) قبل أي نشر علني للمنصة.

7.5 مقاييس الجودة

جدول 11. مقاييس الجودة

المقياس	الهدف المستهدف	الحالة الحالية
تغطية الاختبار (Test Coverage)	تغطية جميع المتطلبات Must وShould بحد أدنى	قيد الإنجاز — 3 من 8 حالات مكتملة
كثافة العيوب (Defect Density)	صفر عيوب حرجة قبل التسليم	لا عيوب مسجَّلة حتى الآن
زمن الاستجابة	أقل من 2 ثانية للصفحات الرئيسية	لم يُقَس رسميًا بعد؛ الملاحظة الأولية إيجابية

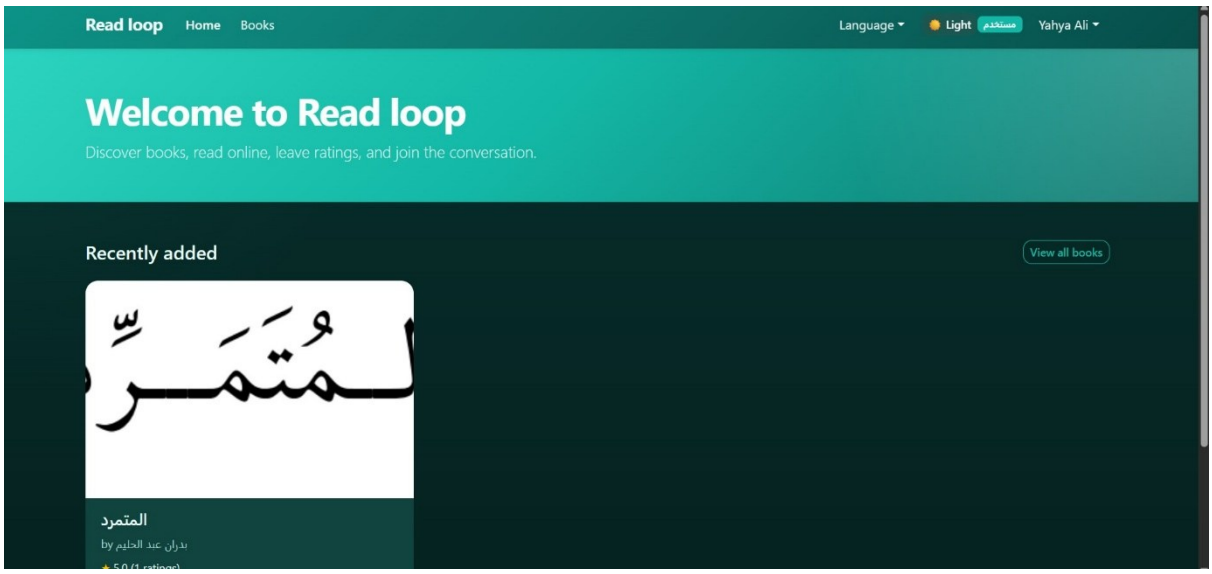
7.6 اختبار المستخدمين

نظرًا لكون المشروع مشروعًا فصليًا فرديًا بإطار زمني محدود، لم يُجرَ اختبار مستخدمين رسمي (استبيان أو مقابلات) حتى تاريخ إعداد هذا التقرير، ويُقترح إجراؤه في مرحلة تخرج لاحقة (تخرج 1) مع عيّنة صغيرة من الطلاب أو القراء لجمع ملاحظات حول سهولة الاستخدام قبل أي نشر أوسع.

الفصل الثامن: النتائج والتحليل

8.1 نتائج تطبيق النظام

يوضح الشكل 17 لقطة شاشة فعلية من الصفحة الرئيسية للمنصة بعد تشغيلها بنجاح على البيئة المحلية، وتظهر فيها عناصر الواجهة الأساسية: شريط التنقّل (الرئيسية، الكتب)، مبدّل اللغة، مبدّل المظهر، وأزرار تسجيل الدخول والتسجيل، إلى جانب قسم «أضيف حديثًا» الذي سيعرض الكتب فور نشرها.



شكل 17. الصفحة الرئيسية للمنصة بعد التشغيل الفعلي

يؤكد هذا التشغيل الفعلي نجاح تكامل الطبقات الثلاث للنظام: قاعدة البيانات (MySQL عبر XAMPP)، والواجهة الخلفية (Laravel)، والواجهة الأمامية (Blade + Tailwind)، وهو ما يمثّل إثباتًا عمليًا لصحة القرارات المعمارية الموضّحة في الفصل الخامس.

8.2 تحليل مقارن مع الأنظمة الموجودة

بالمقارنة مع الأنظمة المستعرضة في الفصل الثالث (الجدول 4)، يحقق النظام المطوَّر الجمع بين القراءة المباشرة داخل المنصة والتفاعل الاجتماعي الكامل ودعم النشر من الكتّاب ودعم اللغة العربية معًا، وهي ميزة لا تجتمع بالكامل في أي من الأنظمة الثلاثة المقارَنة (Wattpad، Goodreads، أبجد)، وإن كانت هذه الأنظمة تتفوق حاليًا من حيث حجم قاعدة المستخدمين ونضج الميزات المتقدمة كالتوصية الذكية.

8.3 تلبية الأهداف

بمراجعة الأهداف الواردة في الفصل الأول (الجدول 1):

- تطوير منصة تصفح وقراءة: تحقَّق — بُنيت واجهات التصفح والقراءة وقاعدة البيانات الداعمة لها.
- تمكين التفاعل المجتمعي: تحقَّق على مستوى التصميم والتنفيذ (جداول comments وratings مع قيودها)؛ يتبقى إكمال اختبارها الشامل.
- تمكين الكتّاب من النشر: تحقَّق على مستوى النموذج والتحكم (BookController وربطها بجدول books عبر user_id).
- إدارة آمنة للمستخدمين: تحقَّق جزئيًا — طُبِّقت الأدوار وحقل banned_at وقيود حماية آخر مدير على مستوى التصميم؛ التحقق الكامل معلَّق على استكمال حالات الاختبار TC-06/TC-07.
- ضمان سهولة الاستخدام والأمان: تحقَّق من حيث التشفير وحماية CSRF المدمجة؛ تحسينات إضافية على سهولة الاستخدام تبقى ممكنة عبر اختبار مستخدمين فعلي مستقبلًا.

بشكل عام، تحقّقت الأهداف الأساسية (Must) بنسبة عالية، فيما تبقى بعض عناصر التحقق الشامل (الاختبار الأمني الكامل واختبار المستخدمين) قيد الإنجاز حتى تاريخ تسليم هذا التقرير.

8.4 تحليل نقاط القوة والضعف

نقاط القوة

- معمارية MVC واضحة وقابلة للصيانة، مع فصل جيد بين الطبقات.
- تصميم قاعدة بيانات علائقي سليم يضمن تكامل البيانات (قيود المفاتيح الأجنبية والقيد الفريد على التقييمات).
- تغطية شاملة نسبيًا للمتطلبات الوظيفية الأساسية رغم الإطار الزمني المحدود ورغم كون التطوير فرديًا.
- توثيق UML متكامل (حالات استخدام، تسلسل، أصناف، ERD) يسهّل تسليم المشروع لمرحلة تخرج لاحقة أو لمطوّر آخر.

نقاط الضعف

- غياب اختبارات آلية (Unit/Feature Tests عبر PHPUnit) حتى الآن، والاعتماد الكلي على الاختبار اليدوي.
- عدم إجراء اختبار مستخدمين حقيقي بعد لتقييم سهولة الاستخدام من منظور خارجي.

- غياب خط أنابيب نشر مستمر (CI/CD) آلي.
- عدم اكتمال بعض الميزات ذات الأولوية الأدنى (Could) مثل تحميل ملف الكتاب وإحصائيات الكاتب التفصيلية بالكامل حتى تاريخ هذا التقرير.

8.5 قياس مؤشرات الأداء (KPIs)

المؤشر	القيمة الحالية / الملاحظة
نسبة إنجاز المتطلبات Must	مرتفعة — جميع متطلبات Must منفَّذة على مستوى الكود والتصميم.
زمن استجابة الصفحة الرئيسية	أقل من ثانيتين ملاحظة أولية على البيئة المحلية (لم يُقَس بأداة قياس رسمية بعد).
عدد حالات الاختبار المكتملة	3 من 8 (37.5%) حتى تاريخ إعداد هذا التقرير.
عدد العيوب الحرجة المسجَّلة	صفر.

الخاتمة

قدّم هذا التقرير مشروع منصة Read Loop، وهي منصة رقمية تفاعلية لمنتدى قراءة الكتب ومشاركة المحتوى الأدبي، صُمِّمت وطُوِّرت باستخدام معمارية MVC وإطار العمل Laravel ضمن منهجية تطوير تكرارية تزايدية. تمثّلت أبرز إنجازات المشروع في بناء نظام متكامل لإدارة المستخدمين بأدوار متعددة (قارئ، كاتب، مدير)، وتصميم قاعدة بيانات علائقية سليمة تدعم التقييمات والتعليقات، إلى جانب تطبيق فعلي وناجح للنظام على بيئة تشغيل محلية تحقّق منها بالمعاينة المباشرة.

أسهم الطالب في هذا المشروع بتنفيذ دورة حياة تطوير برمجيات كاملة بمفرده، بدءًا من تحليل المتطلبات ونمذجتها بلغة UML، مرورًا بالتصميم المعماري وتصميم قاعدة البيانات، وانتهاءً بالتنفيذ الفعلي والاختبار الأولي؛ ويتمثّل موطن التجديد الأساسي في الجمع بين القراءة المباشرة والتفاعل الاجتماعي الكامل ودعم النشر من الكتّاب ضمن واجهة عربية موحّدة، وهو مزيج غير متوفر بالكامل في المنصات المقارَنة.

تبقى نتائج المشروع في هذه المرحلة واعدة على مستوى التصميم والتنفيذ الأساسي، مع إقرار صريح بأن جولة الاختبار الشاملة (خصوصًا الاختبارات الأمنية واختبار المستخدمين) لم تكتمل بعد، وهو ما يُشكّل التحدي الرئيسي المتبقي قبل أي نشر علني موسّع. من أبرز الصعوبات التقنية التي واجهت المشروع إعداد بيئة التطوير المحلية المتكاملة (MySQL، Composer، PHP عبر Node.js، XAMPP) بشكل متسق على نظام Windows، إلى جانب ضيق الوقت الطبيعي لمشروع فصلي واحد.

يوصي المشروع بالاتجاهات المستقبلية التالية: إكمال حزمة الاختبارات الآلية باستخدام PHPUnit، إجراء اختبار مستخدمين حقيقي، إضافة نظام توصية بالكتب مبني على سلوك المستخدم، دعم تحميل ملفات القراءة الإلكترونية بصيغ متعددة (PDF/EPUB)، واعتماد خط أنابيب نشر مستمر (CI/CD) عند توسعة المشروع في مرحلة تخرج لاحقة. كما يُوصى الطلاب القادمين الراغبين بالبناء على هذا العمل بالبدء من مصفوفة تتبع المتطلبات (الملحق أ-4) لتحديد أولويات التوسعة بدقة.

قائمة المراجع

Al-Ghamdi, A. (2023). Fundamentals of Machine Learning. Riyadh: King Fahd National Library.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

Pressman, R. (2019). Software Engineering: A Practitioner's Approach (9th ed.). McGraw-Hill.

Sommerville, I. (2016). Software Engineering (10th ed.). Pearson.

Laravel Documentation. (2025). Laravel 11.x — The PHP Framework for Web Artisans. Retrieved from https://laravel.com/docs

Tailwind Labs. (2025). Tailwind CSS Documentation. Retrieved from https://tailwindcss.com/docs

الجمعه، ك. (2026). دليل كتابة تقارير مشاريع التخرج (الإصدار 1.1). قسم هندسة البرمجيات ونظم المعلومات الذكية، كلية هندسة الذكاء الاصطناعي، الجامعة السورية الخاصة.

ملحق أ: نماذج وأمثلة

أ-1: نموذج وصف حالة استخدام

يرد أدناه نموذجان تفصيليان إضافيان وفق القالب القياسي المعتمد في الدليل الأكاديمي؛ أما التوصيف الكامل لبقية حالات الاستخدام (لكل متطلب وظيفي من FR1 إلى FR19 مجمَّعة حسب مجموعتها الوظيفية، مصحوبة بمخططها الخاص) فيرد بالتفصيل في القسم 4.7 من الفصل الرابع.

العنصر	الوصف
رقم الحالة	01
اسم حالة الاستخدام	تسجيل الدخول إلى النظام
الممثلون الأساسيون	المستخدم
الممثلون الثانويون	—
الوصف	يتيح للمستخدم تسجيل الدخول إلى النظام باستخدام بريده الإلكتروني وكلمة المرور.
الأعمال التحضيرية (Trigger)	يفتح المستخدم صفحة تسجيل الدخول.
الشروط المسبقة	أن يكون المستخدم مسجَّلًا في النظام مسبقًا وحسابه غير محظور.
الشروط اللاحقة	يُوجَّه المستخدم إلى الصفحة الرئيسية موثَّقًا بجلسة نشطة.
الدخل	البريد الإلكتروني وكلمة المرور.
الخرج	توجيه إلى الصفحة الرئيسية أو رسالة خطأ.
المسار الأساسي	1) يفتح المستخدم صفحة تسجيل الدخول. 2) يُدخل البريد الإلكتروني وكلمة المرور. 3) يضغط زر تسجيل الدخول. 4) يتحقق النظام من صحة البيانات وعدم حظر الحساب. 5) يُوجَّه المستخدم إلى الصفحة الرئيسية.
المسار البديل	بيانات دخول غير صحيحة أو حساب محظور: يعرض النظام رسالة خطأ ويعيد توجيه المستخدم إلى صفحة تسجيل الدخول.
التكرارية	عالية
الأولوية	عالية (Must)

العنصر	الوصف
رقم الحالة	02
اسم حالة الاستخدام	نشر كتاب جديد
الممثلون الأساسيون	الكاتب (Author)
الممثلون الثانويون	مدير النظام (مراجعة لاحقة عند الحاجة)
الوصف	يتيح لحساب الكاتب إضافة كتاب جديد بعنوانه ووصفه وملفه إن وُجد.
الأعمال التحضيرية (Trigger)	يفتح الكاتب نموذج «إضافة كتاب» من لوحته الخاصة.
الشروط المسبقة	تسجيل دخول بحساب من نوع كاتب، وعدم كون الحساب محظورًا.
الشروط اللاحقة	إضافة سجل جديد إلى جدول books مرتبط بمعرّف الكاتب (user_id).
الدخل	عنوان الكتاب، الوصف، ملف الكتاب (اختياري).
الخرج	رسالة تأكيد النشر وظهور الكتاب في قائمة الكتب.
المسار الأساسي	1) يفتح الكاتب نموذج إضافة كتاب. 2) يُدخل بيانات الكتاب. 3) يرسل النموذج. 4) يتحقق النظام من صحة الحقول المطلوبة ومن صلاحية الحساب. 5) يُنشئ النظام سجل الكتاب الجديد ويعرض رسالة نجاح.
المسار البديل	حقول ناقصة أو حساب غير مخوَّل (ليس كاتبًا): رفض الطلب وعرض رسالة خطأ توضيحية.
التكرارية	متوسطة
الأولوية	عالية (Must)

أ-2: نماذج User Story

كـ قارئ (User)، أرغب في تصفح قائمة الكتب المتاحة، لكي أتمكن من اختيار كتاب مناسب لاهتماماتي.

معايير القبول: تظهر قائمة بجميع الكتب المنشورة — يمكن البحث عن كتاب بالعنوان — زمن تحميل القائمة لا يتجاوز ثانيتين.

كـ كاتب (Author)، أرغب في نشر كتاب جديد وتعديل بياناته لاحقًا، لكي أتمكن من مشاركة أعمالي الأدبية مع القراء وتحديثها.

معايير القبول: يظهر الكتاب المنشور فورًا في قائمة الكتب — يمكن للكاتب تعديل عنوان ووصف كتبه الخاصة فقط — رسالة تأكيد واضحة بعد النشر.

كـ مدير نظام (Admin)، أرغب في حظر أو حذف حساب مستخدم مخالف، لكي أحافظ على جودة المحتوى وسلامة المجتمع داخل المنصة.

معايير القبول: لا يمكن حظر أو حذف حساب المدير الوحيد المتبقي — لا يمكن للمدير حذف نفسه — يُسجَّل الإجراء فور تنفيذه.

أ-3: مخطط جانت (جدول زمني)

يرد المخطط الزمني الكامل لمراحل المشروع في الجدول 2 بالفصل الثاني (القسم 2.2).

أ-4: مصفوفة تتبع المتطلبات (RTM)

توسَّعت مصفوفة تتبع المتطلبات أدناه لتغطي جميع المتطلبات الوظيفية من FR1 إلى FR19 دون استثناء، بما يربط كل متطلب بعنصر التصميم المطبَّق له وبحالة الاختبار المقابلة (انظر الجدول 10 بالفصل السابع) وحالته الراهنة.

جدول 12. مصفوفة تتبع المتطلبات (RTM)

رقم المتطلب	وصف المتطلب	عنصر التصميم	حالة الاختبار	الحالة
FR1	إنشاء حساب جديد	AuthController / users	TC-04 (ضمنيًا)	منفَّذ
FR2	تسجيل الدخول	AuthController / وسيط المصادقة	TC-04، TC-05	منفَّذ
FR3	تسجيل الخروج	AuthController / إنهاء الجلسة	—	منفَّذ
FR4	عرض قائمة الكتب	BookController / books	TC-08	منفَّذ
FR5	البحث عن كتاب	BookController / استعلام تصفية	—	منفَّذ
FR6	قراءة الكتاب أونلاين	BookController / عارض PDF مضمَّن	—	قيد التنفيذ
FR7	إضافة تعليق	CommentController / comments	—	منفَّذ
FR8	إضافة/تحديث تقييم	RatingController / ratings (قيد فريد)	TC-06	منفَّذ
FR9	نشر كتاب جديد	BookController / books	—	منفَّذ

الحالة	حالة الاختبار	عنصر التصميم	وصف المتطلب	رقم المتطلب
منفَّذ	—	BookController / سياسة ملكية الكتاب	تعديل كتاب	FR10
منفَّذ	—	BookController / سياسة ملكية أو صلاحية المدير	حذف كتاب	FR11
قيد التنفيذ	—	BookController / استجابة تنزيل الملف	تحميل ملف الكتاب	FR12
منفَّذ	TC-08	BookController / صفحة التفاصيل	عرض تفاصيل الكتاب	FR13
منفَّذ	TC-07	Admin\UserController / وسيط صلاحية المدير	إدارة المستخدمين (دور/حظر/حذف)	FR14
منفَّذ	—	Admin\CommentController / وسيط صلاحية المدير	إدارة التعليقات	FR15
منفَّذ	TC-07	UserPolicy (سياسة صلاحية)	منع حذف آخر مدير أو حذف النفس	FR16
منفَّذ	—	ProfileController	تعديل الحساب الشخصي وكلمة المرور	FR17
منفَّذ	—	طبقة العرض (Blade) / إعدادات اللغة	تبديل لغة الواجهة	FR18
قيد التنفيذ	—	AuthorDashboardController	إحصائيات الكاتب	FR19

أ-5: قائمة مراجع (APA)

انظر قائمة المراجع الكاملة الواردة بعد الفصل الثامن مباشرة، وقد اعتُمد فيها نمط APA (الإصدار السابع).

ملحق ج: نموذج خطة الاختبار

ج-1: نطاق الاختبار

ما سيُختبر: وظائف المصادقة، تصفح وقراءة الكتب، التقييمات والتعليقات، نشر الكتب وإدارتها، ولوحة إدارة المستخدمين. ما لن يُختبر في هذه المرحلة: التوافق الشامل مع جميع إصدارات المتصفحات القديمة، واختبار الحمل (Load Testing) على نطاق واسع.

ج-2: استراتيجية الاختبار

الأداة/الطريقة	الهدف	نوع الاختبار
اختبار يدوي، ويُقترح PHPUnit مستقبلًا	اختبار كل دالة تحكم بشكل منفصل	اختبار الوحدة
اختبار يدوي عبر الواجهة و/أو Tinker	اختبار تفاعل الوحدات مع قاعدة البيانات	اختبار التكامل
متصفح ويب (Chrome/Firefox)	اختبار سيناريو المستخدم كاملًا	اختبار النظام
اختبار يدوي بحسابات أدوار مختلفة	التحقق من الصلاحيات والقيود	اختبار الأمان الأساسي

ج-3: بيئة الاختبار

العنصر	المواصفات
نظام التشغيل	Windows (جهاز التطوير المحلي)
الخادم	خادم Laravel المدمج — php artisan serve — المنفذ 8000
قاعدة البيانات	MySQL عبر XAMPP (بيئة اختبار محلية)
المتصفحات	Google Chrome (أساسي)، مع مراجعة أولية على Firefox وEdge

ج-4: معايير القبول

- نجاح جميع حالات الاختبار الأساسية (Must) بنسبة 100% قبل المناقشة النهائية.
- عدم وجود أخطاء حرجة متبقية تمنع تنفيذ أي وظيفة أساسية.
- زمن استجابة أقل من ثانيتين لصفحات التصفح الرئيسية.

ملحق د: نموذج دليل المستخدم

د-1: نظرة عامة

منصة Read Loop هي منصة ويب لقراءة الكتب ومشاركة المحتوى الأدبي، تدعم ثلاثة أنواع من الحسابات: قارئ، كاتب، ومدير نظام.

د-2: البدء السريع

1. افتح المتصفح وانتقل إلى رابط المنصة.
2. اضغط «Register» لإنشاء حساب جديد ببريدك الإلكتروني.
3. بعد تسجيل الدخول، اضغط «Books» لتصفح الكتب المتاحة.
4. افتح أي كتاب لقراءته أونلاين، أو لإضافة تقييم أو تعليق عليه.
5. إذا كان حسابك من نوع «كاتب»، ستجد رابط «نشر كتاب جديد» في لوحتك الخاصة.

د-3: استكشاف الأخطاء الشائعة

المشكلة	الحل المقترح
رسالة «تعذّر الوصول إلى الصفحة»	تأكد من أن خادم Laravel (php artisan serve) لا يزال قيد التشغيل.
فشل تسجيل الدخول رغم صحة البيانات	تأكد من أن الحساب غير محظور (banned_at) وتواصل مع مدير النظام عند الحاجة.
عدم ظهور كتب في الصفحة الرئيسية	تأكد من تنفيذ أمر php artisan db:seed أو إضافة كتب فعلية من حساب كاتب.

ملحق هـ: الأخلاقيات البحثية واستخدام الذكاء الاصطناعي

هـ-1: النزاهة الأكاديمية

يقرّ الطالب بأن جميع المصادر المستخدمة في هذا التقرير موثَّقة وفق نمط APA (الإصدار السابع) كما هو مبيَّن في قائمة المراجع، وأن الكود المصدري للمشروع من تأليف الطالب، مع الاستعانة بمكتبات وأطر عمل مفتوحة المصدر (Laravel، Tailwind CSS) منسوبة لأصحابها ضمن رخصها المعلَنة.

هـ-2: استخدام أدوات الذكاء الاصطناعي

استُخدمت أداة الذكاء الاصطناعي Claude (من إنتاج شركة Anthropic) كمساعد تقني خلال مراحل متعددة من المشروع، شملت على وجه الخصوص:

- المساعدة في تجهيز بيئة التطوير المحلية (تثبيت PHP وComposer وMySQL عبر XAMPP وNode.js) وتشخيص أخطاء الإعداد.
- المساعدة في تشغيل أوامر Laravel القياسية (php artisan migrate، composer install، php artisan serve، php artisan db:seed) وتفسير مخرجاتها.
- دعم كتابة وصياغة هذا التقرير النهائي وفق هيكلية الدليل الأكاديمي المعتمد من القسم، بالاستناد إلى محتوى المشروع الفعلي ومخططات UML المُعدَّة مسبقًا من الطالب.

لم تُستخدم أداة الذكاء الاصطناعي لتوليد نتائج اختبار وهمية أو بيانات غير حقيقية؛ إذ تعكس جميع نتائج الاختبار الواردة في الفصل السابع ما جرى تنفيذه فعليًا على بيئة التشغيل المحلية حتى تاريخ إعداد هذا التقرير. راجع الطالب جميع المخرجات المتعلقة بالتقرير والكود وحرَّرها قبل اعتمادها النهائي.

هـ-3: قواعد السلوك الأكاديمي

التزم الطالب بالمواعيد المحددة من قبل المشرف، وبالتعاون الإيجابي معه عبر المراجعات الدورية، وبالصدق الكامل في عرض النتائج والصعوبات التي واجهت المشروع كما وردت في الخاتمة.