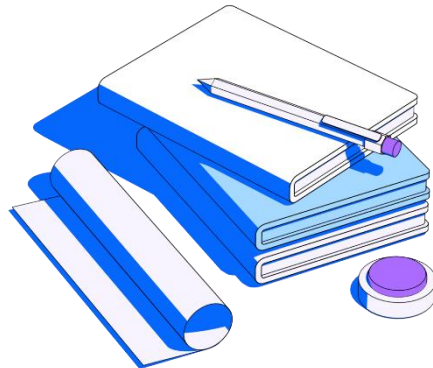




## التقرير

Online Book Inventory System



Student: abubakr\_203648

Tutor: Prof. Bassel Mustaffa alkhatab

Class: ITE\_BWP401\_C3

S23

## المحتويات

2	1. الملخص
3	2. المقدمة
4	3. كيف تم بناء الموقع
4	4. شرح عن الموقع
4	4.1. الصفحة الرئيسية (Homepage)
5	4.2. صفحة تفاصيل الكتاب (Book Details Page)
6	4.3. عمل البحث (Search Functionality)
6	4.4. قائمة قراءة المستخدم (User Reading List)
7	4.5. صفحة المشرف (Admin Section)
8	4.6. تكامل قاعدة البيانات (Database Integration)
9	4.7. المصادقة والتحويل (Authentication and Authorization)
10	4.8. التصميم المتجاوب (Responsive Design)
10	5. خاتمة
12	6. الملاحق
12	6.1. لقطات الشاشة مع صور من الكود
12	6.1.1. الصفحة الرئيسية (Homepage)
13	6.1.2. صفحة تفاصيل الكتاب (Book Details Page)
15	6.1.3. عمل البحث (Search Functionality)
17	6.1.4. قائمة قراءة المستخدم (User Reading List)
19	6.1.5. صفحة المشرف (Admin Section)
24	6.1.6. تكامل قاعدة البيانات (Database Integration)
26	6.1.7. المصادقة والتحويل (Authentication and Authorization)
32	6.1.8. التصميم المتجاوب (Responsive Design)
32	6.2. مراجع

## 1. الملخص

الموقع عبارة عن نظام لجرد الكتب عبر الإنترنت يتيح للمستخدمين تصفح الكتب والبحث فيها وإدارتها من صفحة المشرف.

الأهداف الرئيسية للموقع هي:

- توفير واجهة ويب سهلة الاستخدام وسريعة الاستجابة للوصول إلى مخزون الكتب.
- القيام بتنفيذ ميزات متنوعة مثل وظيفة البحث وقائمة قراءة المستخدم و صفحة المشرف.
- دمج تطبيق الويب مع قاعدة بيانات لتخزين البيانات واسترجاعها.
- تأمين تطبيق الويب بآليات المصادقة والترخيص.

## 2. المقدمة

تتمثل مشكلة هذا الموقع في تصميم وتطوير نظام جرد الكتب عبر الإنترنت الذي يمكنه مواجهة هذه التحديات وتوفير تجربة مستخدم مُرضية. الغرض من هذا الموقع هو تطبيق المعرفة والمهارات الخاصة بتطوير الويب، مثل HTML و CSS و JavaScript و #C و ASP.NET و SQL، لإنشاء تطبيق ويب يمكنه إدارة الكتب وعرضها من معرض كتب عبر الإنترنت. نطاق هذا الموقع هو تنفيذ ثماني ميزات رئيسية، كما هو محدد في متطلبات الموقع، والتي تغطي الوظائف والتصميم والجوانب الأمنية لتطبيق الويب.

وكانت أهداف الموقع هي:

- إنشاء صفحة رئيسية تعرض الكتب المميزة مع الصور والعناوين والأوصاف، وروابط لصفحات تفاصيل الكتاب.
- إنشاء صفحة تفاصيل الكتاب التي تعرض معلومات تفصيلية عن كتاب محدد، وخيار لإضافة الكتاب إلى قائمة القراءة الخاصة بالمستخدم.
- إضافة وظيفة بحث تمكن المستخدمين من إدخال الكلمات الرئيسية ورؤية نتائج البحث الديناميكية على نفس الصفحة.
- تنفيذ مصادقة المستخدم باستخدام مصادقة نماذج ASP.NET، والسماح للمستخدمين بإنشاء حسابات، وتسجيل الدخول، والاطلاع على الكتب الخاصة بهم، وخيار لإزالة الكتب من القائمة.
- تنفيذ صفحة إداري آمن لا يمكن الوصول إليه إلا للمستخدمين المصرح لهم، وإنشاء صفحة إدارية منفصلة مع وظيفة لإضافة كتب جديدة إلى المخزون وتحديث معلومات الكتب الموجودة وحذفها.
- إعداد قاعدة بيانات SQL Server لتخزين واسترجاع المعلومات حول الكتب والمستخدمين وقوائم القراءة، واستخدام ASP.NET مع #C لإنشاء اتصال بقاعدة البيانات.
- استخدام المصادقة المستندة إلى الأدوار للتمييز بين المستخدمين العاديين والمستخدمين الإداريين، وتقييد الوصول إلى صفحات معينة بناءً على أدوار المستخدم.
- تأكد من أن تطبيق الويب جذاب بصريًا وعلمي على الأجهزة والمتصفحات المختلفة، واستخدم تقنيات التصميم سريعة الاستجابة مثل استعلامات الوسائط لتكييف التخطيط بناءً على حجم الشاشة.

### 3. كيف تم بناء الموقع

قضيت حوالي اسبوعين في انشاء الموقع الإلكتروني، بدءًا من التخطيط الأولي وحتى الاختبار النهائي والتقييم. لقد قمت بإدارة الموقع الإلكتروني باستخدام الخطوات التالية:

- لقد حددت الأهداف والنطاق ومؤشرات الأداء الرئيسية للموقع، وحددت الأهداف والتوقعات للتقرير.
- لقد أنشأت إطارًا سلكيًا (wireframe) وبعدها قمت بتحسين التصميم في تطبيق الويب Figma.
- لقد قمت بتطوير تطبيق الويب باستخدام الأدوات والتقنيات المذكورة أعلاه، واتبعت أفضل الممارسات والمعايير لتطوير الويب.
- لقد قمت باختبار وتقييم تطبيق الويب على أجهزة ومتصفحات مختلفة، وحددت وأصلحت أي مشكلات أو أخطاء حدثت.

### 4. شرح عن الموقع

تم وضع الأشياء الأساسية في الموقع مثل الروابط الخارجية ورأس (header) وتذييل (footer) في ملف رئيسي اسمه site.master, وليتم الربط بين الصفحات تم استخدام أحد عناصر التحكم ContentPlaceholder الذي أقوم باستبدال قيمة هذا العنصر مع الصفحة التي يتم التوجه إليها.

#### 4.1. الصفحة الرئيسية (Homepage)

استخدمت نماذج ويب ASP.NET لإنشاء صفحة ASPX باسم Default.aspx. تحتوي هذه الصفحة على ترميز HTML و ASP.NET لتخطيط الصفحة الرئيسية ومحتواها. لقد استخدمت مكتبة Bootstrap لتصميم الصفحة وجعلها سريعة الاستجابة.

لتضمين صفحة للكتب المميزة، استخدمت عنصر تحكم Repeater لعرض قائمة الكتب من قاعدة البيانات ديناميكيًا. يحتوي عنصر التحكم Repeater على خاصية ItemTemplate التي تحدد بنية HTML لكل عنصر في الكتاب. داخل ItemTemplate، استخدمت عنصر التحكم Image لإظهار غلاف الكتاب، ووسم آخر لإظهار باقي البيانات، ولتنفيذ الارتباطات التشعبية على كل كتاب مميز، استخدمت العلامة <a> لتغليف كل بطاقة كتاب، وقمت بتعيين سمة href على عنوان URL لصفحة تفاصيل الكتاب. لقد استخدمت سلسلة استدعاء لتمرير معرف الكتاب كمعلمة إلى صفحة تفاصيل الكتاب، حتى تتمكن من عرض المعلومات الصحيحة للكتاب المحدد.

لملء عنصر التحكم Repeater بالبيانات من قاعدة البيانات، أنشأت ملف اسمه DatabaseAccess.cs الذي يقوم هو فقط في الوصول لقاعدة البيانات وقراءة البيانات، وعند تحميل الصفحة يقوم بقراءة بيانات قاعدة البيانات واسناد قيم كل حقل مع ما يقابله:

[اضغط هنا للانتقال الى صورة الصفحة الرئيسية والكود](#)

#### 4.2. صفحة تفاصيل الكتاب (Book Details Page)

لقد استخدمت لإنشاء ديناميكيًا استنادًا إلى معرف الكتاب الذي تم تمريره كمعلمة سلسلة استدعاء في عنوان URL. ولتصميم لقد استخدمت نفس تخطيط وأسلوب الصفحة الرئيسية، ولكن مع أقسام محتوى مختلفة. أقسام المحتوى هي:

- صورة كبيرة لغلاف الكتاب على الجانب الأيسر، مع وجود شريط عمودي لفصله عن بقية الصفحة.
- سلسلة من الفقرات على الجانب الأيمن، تعرض كل منها معلومات أساسية عن الكتاب، مثل العنوان والمؤلف والنوع والسعر والصفحات والوصف. لقد استخدمت علامة `<span>` لعرض البيانات من قاعدة البيانات، وعلامة `<strong>` للتأكيد على التسميات.
- مجموعة من الأزرار في الأسفل، تتيح للمستخدم إضافة الكتاب أو إزالته من قائمة القراءة الخاصة به، أو تسجيل الدخول للقيام بذلك. لقد استخدمت علامة `<asp:LinkButton>` لإنشاء الأزرار، وعلامة `<asp:UpdatePanel>` لتمكين التحديثات الجزئية للصفحة دون إعادة تحميل الصفحة. لقد استخدمت أيضًا علامة `<asp:Label>` لإظهار رسالة نجاح عندما يقوم المستخدم بإجراء ما.

وفي code-behind للتعامل مع تحميل الصفحة وأحداث النقر على الزر. لقد استخدمت خاصية Request.QueryString للحصول على معرف الكتاب من عنوان URL، وأسلوب DatabaseAccess.Books.GetBookFromID للحصول على كائن الكتاب من قاعدة البيانات. ثم استخدمت كائن الكتاب لملء عناصر HTML ببيانات الكتاب، مثل مصدر الصورة، والنص الممتد، ورؤية الزر. لقد استخدمت أيضًا خاصية SessionManager.IsUserLoggedIn للتحقق مما إذا كان المستخدم قد قام بتسجيل الدخول، وطريقة RedirectManager.ToHomePage لإعادة توجيه المستخدم إلى الصفحة الرئيسية إذا كان معرف الكتاب غير صالح. لقد استخدمت أيضًا أساليب

DatabaseAccess.Books.AddToReadingList

و DatabaseAccess.Books.RemoveFromReadingList لتحديث قائمة القراءة الخاصة بالمستخدم في قاعدة البيانات، وخاصية lblMessage.Text لإظهار رسالة نجاح للمستخدم.

[اضغط هنا للانتقال الى صورة الصفحة تفاصيل الكتاب والكود](#)

#### 4.3. عمل البحث (Search Functionality)

لقد استخدمت العلامة <asp:TextBox> لإنشاء عنصر إدخال نص بنص نائب "البحث عن الكتب...". لقد استخدمت CSS لتصميم شريط البحث بحدود مستديرة ومساحة حشو. لقد استخدمت أيضًا عنصر <div> بمعرف "searchResults" لإنشاء حاوية لنتائج البحث، ووضعت أسفل شريط البحث باستخدام CSS.

لاستخدام JavaScript لعرض نتائج البحث ديناميكيًا على نفس الصفحة دون إعادة التحميل، استخدمت jQuery لإرفاق معالج حدث الإدخال بشريط البحث، والذي يتم تنشيطه عندما يكتب المستخدم شيئًا ما. داخل معالج الأحداث، استخدمت الأسلوب \$.post لإرسال طلب AJAX إلى صفحة SearchResults.aspx، وتميرير الكلمة الأساسية للبحث كمعلمة سلسلة استدعاء.

للتأكد من أن نتائج البحث تتضمن عناوين الكتب والتفاصيل ذات الصلة، استخدمت رمز #C الخلفي للتعامل مع حدث تحميل الصفحة لصفحة SearchResults.aspx. لقد استخدمت الخاصية Request.QueryString للحصول على الكلمة الأساسية للبحث من عنوان URL، والأسلوب DatabaseAccess.Books.GetBooks للحصول على قائمة بكائنات الكتب من قاعدة البيانات التي تطابق الكلمة الأساسية للبحث. ثم استخدمت خاصية Response.ContentType لتعيين نوع الاستجابة إلى "text/xml"، وفئة XmlWriter لإنشاء مستند XML يحتوي على بيانات الكتاب.

[اضغط هنا للانتقال الى صورة عملية البحث والكود](#)

#### 4.4. قائمة قراءة المستخدم (User Reading List)

لإنشاء قائمة قراءة خاصة بالمستخدم حيث يمكن للمستخدمين إضافة كتب ورؤية اختياراتهم الخاصة، استخدمت عنصر التحكم <asp:Repeater> لعرض شبكة من بطاقات الكتب في صفحة قائمة القراءة، والتي لا يمكن الوصول إليها إلا للمستخدمين الذين قاموا بتسجيل الدخول. لقد استخدمت أسلوب DatabaseAccess.Books.GetBooksFromUserID للحصول على قائمة بكائنات الكتب من قاعدة البيانات التي تنتمي إلى قائمة القراءة الخاصة بالمستخدم الحالي، وربطها بعنصر تحكم المكرر. تعرض كل

بطاقة كتاب صورة الكتاب وعنوانه ومؤلفه ووصفه ورابطاً إلى صفحة تفاصيل الكتاب. لقد استخدمت أيضاً عنصر التحكم `<asp:LinkButton>` لإنشاء زر إزالة لكل بطاقة كتاب، مما يسمح للمستخدم بإزالة الكتاب من قائمة القراءة الخاصة به.

لتنفيذ مصادقة المستخدم باستخدام الملف المنشأ `SessionManager.cs` لتخزين واسترجاع معرف المستخدم وحالة تسجيل الدخول في حالة الجلسة.

للسماح للمستخدمين بإنشاء حسابات وتسجيل الدخول، قمت بتوفير روابط لصفحة تسجيل الدخول وصفحة تسجيل المستخدم على الصفحة الرئيسية وصفحة تفاصيل الكتاب. في صفحة تسجيل الدخول، يمكن للمستخدمين إدخال اسم المستخدم وكلمة المرور الخاصة بهم، والنقر فوق زر تسجيل الدخول لتسجيل الدخول. في صفحة تسجيل المستخدم، يمكن للمستخدمين إدخال اسم المستخدم وكلمة المرور والبريد الإلكتروني وكلمة المرور، ثم النقر فوق إنشاء مستخدم زر لإنشاء حساب. يمكن للمستخدمين أيضاً تسجيل الخروج من خلال النقر على زر تسجيل الخروج في الصفحة الرئيسية أو صفحة تفاصيل الكتاب.

[اضغط هنا للانتقال الى قائمة قراءة المستخدم والكود](#)

#### 4.5. صفحة المشرف (Admin Section)

لقد استخدمت عنصر التحكم `<asp:HyperLink>` لإنشاء رابط لصفحة المشرف على الصفحة الرئيسية، والذي يكون مرئياً فقط لدور المشرف. في صفحة الإدارة، استخدمت عنصر التحكم `<asp:Repeater>` لعرض شبكة من بطاقات الكتب، تشبه الصفحة الرئيسية، ولكن مع أزرار إضافية لحذف الكتب وتحريرها. لقد استخدمت أيضاً عنصر تحكم `<asp:HyperLink>` آخر لإنشاء رابط إلى صفحة إضافة كتاب، حيث يمكنني إدخال معلومات الكتاب وتحميل صورة الكتاب. لقد استخدمت أساليب `DatabaseAccess.Books.AddBook` و `DatabaseAccess.Books.UpdateBook` لإدراج بيانات الكتاب وتحديثها في قاعدة البيانات.

لقد استخدمت صف `SessionManager.cs` الذي يحتوي فيه دور المستخدم. لقد قمت بإنشاء دورين: المشرف والمستخدم. لقد قمت بتعيين دور المشرف لنفسه ودور المستخدم للمستخدمين الآخرين. ويمكن فقط لدور المشرف بالوصول إلى صفحة المشرف ورفض دور المستخدم والمستخدمين المجهولين. لقد استخدمت



أيضاً الصف RedirectManager لإعادة توجيه المستخدمين غير المصرح لهم إلى الصفحة الرئيسية أو صفحة تسجيل الدخول إذا حاولوا الوصول إلى صفحة الإدارة.

للتأكد من أن المستخدمين الذين لديهم أدوار إدارية هم فقط من يمكنهم الوصول إلى هذه الصفحة واستخدامه، استخدمت طريقة Roles.IsUserAdmin للتحقق من دور المستخدم في حدث تحميل الصفحة لصفحة المشرف و صفحة إضافة كتاب. لقد استخدمت طريقة نفسها للحصول على دور المستخدم على الصفحة الرئيسية، واستخدمت الخاصية Visible لإظهار أو إخفاء الرابط إلى صفحة المشرف بناءً على دور المستخدم.

[اضغط هنا للانتقال الى صورة صفحة المشرف وتحكمته والكود](#)

#### 4.6. تكامل قاعدة البيانات (Database Integration)

قمت بأعداد قاعدة بيانات SQL Server لتخزين معلومات حول الكتب والمستخدمين وقوائم القراءة، استخدمت SQL Server Management Studio لإنشاء قاعدة بيانات باسم OBIS وثلاثة جداول باسم الكتب والمستخدمين وقائمة القراءة. لقد حددت الأعمدة وأنواع البيانات لكل جدول، وقمت بتعيين المفاتيح الأساسية والمفاتيح الاجنبي حسب الحاجة. لقد قمت أيضاً بملء الجداول ببعض بيانات العينة لأغراض الاختبار. لقد استخدمت صف ConfigurationManager لتخزين سلسلة الاتصال بقاعدة البيانات في ملف Web.config.

لاتصال بقاعدة البيانات، قمت بإنشاء صف باسم DatabaseAccess، والتي تحتوي على ثلاث فئات متداخلة تسمى Books، Users، وReadingList. تحتوي كل فئة متداخلة على أساليب ساكن تؤدي عمليات قاعدة البيانات المختلفة، مثل الحصول على البيانات وإضافتها وحذفها وتحديثها. لقد استخدمت أيضاً صف Db لتحديد كائنات الكتاب والمستخدم وقائمة القراءة وخصائصها.

[اضغط هنا للانتقال الى صورة تكامل قاعدة البيانات والكود، ويمكن رؤية كامل الكود في داخل مجلد](#)

[App code في مجلد publish](#)

#### 4.7. المصادقة والتحويل (Authentication and Authorization)

لتنفيذ صفحة تسجيل مستخدم لإنشاء حسابات جديدة، قمت بإنشاء صفحة Signup.aspx والذي يسمح للمستخدمين بإدخال اسم المستخدم وكلمة المرور والبريد الإلكتروني والمعلومات الاختيارية الأخرى. لقد استخدمت أسلوب DatabaseAccess.Users.AddUser لإنشاء حساب مستخدم جديد في قاعدة البيانات، ويكون دور المستخدم بشكل افتراضي User. لقد استخدمت أيضًا صف RedirectManager لإعادة توجيه المستخدم إلى الصفحة الرئيسية أو صفحة تسجيل الدخول بعد إنشاء الحساب، ويتم التأكد من مدخلات المستخدم في حال استخدام كانت المدخلات فارغة و اسم المستخدم مستخدم أو البريد الإلكتروني مستخدم أيضا و بعض التحقيقات الأخيرة.

لتطوير صفحة تسجيل دخول بمنطق المصادقة للتحقق من بيانات اعتماد المستخدم، قمت بإنشاء صفحة Signup.aspx والذي يسمح للمستخدمين بإدخال اسم المستخدم أو البريد الإلكتروني وكلمة المرور الخاصة بهم. لقد استخدمت طريقة DatabaseAccess.Users.GetUser للتحقق مما إذا كانت بيانات اعتماد المستخدم صالحة، وأسلوب SessionManager.LoginUser لتخزين المستخدم في الجلسة الحالية وإعادة توجيه المستخدم إلى الصفحة المناسبة. لقد استخدمت أيضًا صف SessionManager لتخزين واسترجاع معرف المستخدم واسم المستخدم والدور في حالة الجلسة، وصف RedirectManager لإعادة توجيه المستخدم إلى الصفحة الرئيسية أو صفحة المشرف حسب دوره، ويتم التحقق من مدخلات المستخدم في حال استخدام كانت المدخلات فارغة و اسم المستخدم أو بريد الإلكتروني لا يطابق كلمة المرور.

لتقييد الوصول إلى صفحات معينة (على سبيل المثال، صفحة المشرف) بناءً على أدوار المستخدم، استخدمت فئة SessionManager للتحقق من حالة تسجيل دخول المستخدم ودوره في حدث تحميل الصفحة لكل صفحة. لقد استخدمت طريقة SessionManager.RedirectToLoginIfNotLoggedIn لإعادة توجيه المستخدم إلى صفحة تسجيل الدخول إذا لم يقوم بتسجيل الدخول، وطريقة SessionManager.RedirectToLoginIfNotAdmin لإعادة توجيه المستخدم إلى صفحة تسجيل الدخول إذا لم يكن مسؤولاً. لقد استخدمت أيضًا فئة RedirectManager لإعادة توجيه المستخدم إلى الصفحة الرئيسية أو صفحة الإدارة حسب الحاجة.

[اضغط هنا للانتقال الى صورة صفحة التسجيل دخول والاشتراك والكود](#)

#### 4.8. التصميم المتجاوب (Responsive Design)

للتأكد من أن موقع الويب جذاب بصريًا وعملي على الأجهزة المختلفة (أجهزة الكمبيوتر المكتبية والأجهزة اللوحية والهواتف المحمولة)، استخدمت إطار عمل Bootstrap لإنشاء تخطيط ونمط سريع الاستجابة ومتوافق مع الأجهزة المحمولة لصفحات الويب. لقد استخدمت نظام شبكة Bootstrap لتنظيم المحتوى في صفوف وأعمدة، واستخدمت صفوف Bootstrap لضبط حجم العناصر ومحاذاتها وتباعدها ومظهرها. لقد استخدمت أيضًا مكونات Bootstrap لإنشاء عناصر متسقة وتفاعلية، مثل أشرطة التنقل والأزرار والبطاقات والنماذج.

#### اضغط هنا للانتقال الى صورة الموقع في حجم الموبايل والكود

#### 5. خاتمة

كانت أهداف الموقع هو إنشاء موقع ويب سهل الاستخدام وآمن وسريع الاستجابة يمكنه التعامل مع مختلف العمليات والبيانات المتعلقة بالكتب، مثل عرض الكتب المميزة، وإظهار تفاصيل الكتاب، والبحث عن الكتب، وإضافة الكتب إلى قوائم القراءة، وإدارة الكتب المخزون، والمصادقة وترخيص المستخدمين.

وكانت الدروس المستفادة من إنشاء الموقع هي:

- كيفية استخدام HTML و CSS و JavaScript و ASP.NET و #C و SQL Server لتصميم وتطوير تطبيق ويب يمكنه التعامل مع مختلف العمليات والبيانات المتعلقة بالكتب.
- كيفية استخدام Bootstrap و jQuery , وإضافة التفاعل والوظائف، والعمل مع قاعدة البيانات باستخدام كائنات #C ، واختبار تطبيق الويب على متصفحات مختلفة.
- كيفية استخدام فئات العضوية والأدوار لإدارة حسابات المستخدمين وكلمات المرور، وتعيين جلسات للمصادقة وإعادة توجيه المستخدمين، وإنشاء أدوار المستخدم وقواعد التفويض وإدارتها.
- كيفية استخدام صفوف SqlConnection و SqlCommand و SqlDataReader و SqlParameter لإنشاء اتصال بقاعدة البيانات، وإنشاء أمر باستخدام استعلام SQL والمعلومات، وتنفيذ الأمر لعمليات مختلفة، وقراءة البيانات من قاعدة البيانات، وإنشاء المعلومات للأمر.
- كيفية استخدام فئة XmlWriter لإنشاء مستند XML يحتوي على بيانات الكتاب.

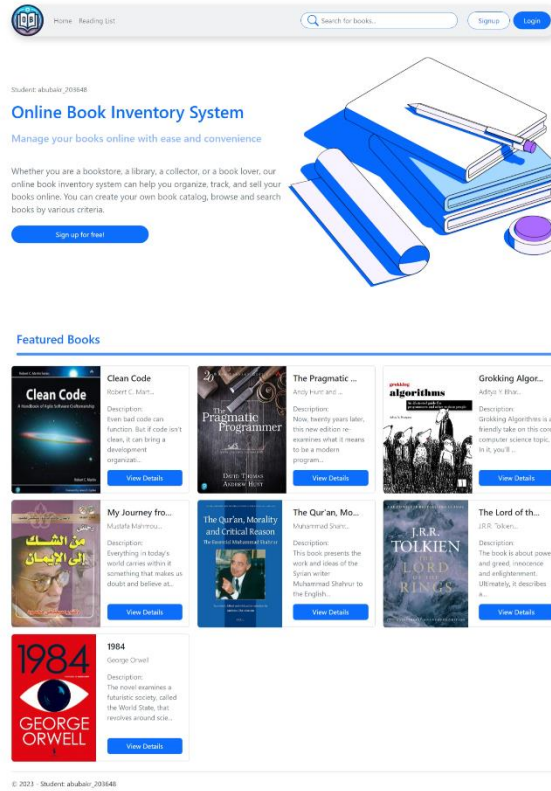
الأعمال المستقبلية والتحسينات للموقع هي:

- تنفيذ تقنيات التشفير والتحقق والتحسين لتطبيق الويب، مما قد يؤدي إلى تحسين أمان وأداء تطبيق الويب.
- لإضافة ميزات مثل التقييمات والمراجعات والتوصيات وخيارات الدفع عبر الإنترنت إلى تطبيق الويب، مما قد يعزز تجربة المستخدم والقيمة التجارية لمتجر الكتب عبر الإنترنت.
- تطبيق مبادئ التصميم التي تركز على المستخدم وإجراء اختبارات المستخدم والتعليقات على تطبيق الويب، مما قد يؤدي إلى تحسين تصميم واجهة المستخدم وسهولة استخدام تطبيق الويب.

## 6. الملاحق

### 6.1.1 لقطات الشاشة مع صور من الكود

#### 6.1.1.1 الصفحة الرئيسية (Homepage)



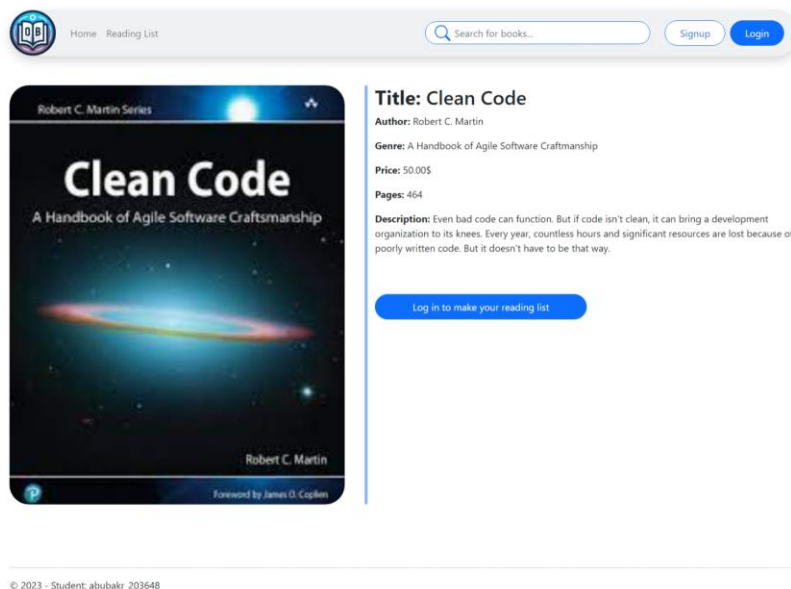
```
asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">
    <div class="min-vh-75 d-flex flex-row justify-content-between align-items-center flex-md-row flex-column-reverse">
        <div class="col-md-6 col-sm-12">
            <p class="text-secondary">Student: ahubakr_203648</p>
            <h1 class="text-primary mb-4 fs-1">Online Book Inventory System</h1>
            <h2 class="text-primary text-opacity-50 mb-5 fs-4">Manage your books online with ease and convenience</h2>
            <p class="text-secondary fs-5">
                Whether you are a bookstore, a library, a collector, or a book lover,
                our online book inventory system can help you organize, track, and sell your books online. You can create your own book catalog,
                browse and search books by various criteria.
            </p>
            <a class="nav-link btn btn-primary w-25 w-50 text-white p-2 mt-4 rounded-pill mb-5" runat="server" href="@RedirectManager.PagesURL.SignupURL">Sign up for free!</a>
        </div>
        <div class="col-md-6 col-sm-12 mv-50 flex-fill">
            
        </div>
    </div>
    <div class="container mt-5">
        <h3 class="text-primary">Featured Books</h3>
        <hr class="bg-primary border-primary rounded-pill opacity-75" style="height:6px" />
        <div class="book-section row justify-content-between align-items-center">
            <asp:Repeater ID="BooksSection" runat="server">
                <itemTemplate>
                    <div class="col-md-12 col-lg-4 card my-2 p-8" style="height:300px;width:420px">
                        <div class="d-flex p-0 h-100">
                            <div class="col-6 h-100">
                                <asp:Image ImageUrl="@Eval("Image")" ID="BookImage" runat="server" CssClass="img-fluid rounded-start mv-100" Height="300" />
                            </div>
                            <div class="col-6 p-0 flex-fill m-100">
                                <div class="card-body d-flex justify-content-between flex-column h-100 w-100">
                                    <div>
                                        <h5 class="card-title">@Eval("Name").ToString().Count() < 14?Eval("Name"):Eval("Name").ToString().Substring(0,14)+"...</h5>
                                        <p class="card-text fs-6 fw-light">@Eval("Author").ToString().Count() < 14? Eval("Author"): Eval("Author").ToString().Substring(0,14)+"...</p>
                                        <p class="card-text text-secondary">Description: <br />@Eval("Description").ToString().Count() < 90? Eval("Description"): Eval("Description").ToString().Substring(0,90)+"...</p>
                                    </div>
                                    <p class="card-text"><a href="@RedirectManager.PagesURL.BookDetailsURL"/BookID="@Eval("Id")" class="btn btn-primary w-100">View Details</a></p>
                                </div>
                            </div>
                        </div>
                    </div>
                </itemTemplate>
            </asp:Repeater>
        </div>
    </div>
```

```

namespace Online_Book_Inventory_System
{
    2 references
    public partial class _Default : Page
    {
        0 references
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                Page.DataBind();
                BooksSection.DataSource = DatabaseAccess.Books.GetBooks();
                BooksSection.DataBind();
            }
        }
    }
}

```

## 6.1.2. صفحة تفاصيل الكتاب (Book Details Page)



```

asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">
    <div class="row min-vh-100 mt-5 align-items-start">
        <div class="col-lg-6 h-100 d-flex position-relative align-items-center" style="max-width:600px">
            <img src="" id="BookImage" runat="server" class="img-fluid h-100 w-100 rounded-5" style="max-width:580px">
            <div class="vr position-absolute end-0 h-100 bg-primary rounded-pill d-lg-block d-none opacity-50" style="width:5px; right:-5px;></div>
        </div>
        <div class="col-lg-6 d-flex flex-fill justify-content-between flex-column">
            <div class="h-100">
                <h2><strong>Title:</strong> <span id="BookName" runat="server"></span></h2>
                <p><strong>Author:</strong> <span id="BookAuthor" runat="server"></span></p>
                <p><strong>Genre:</strong> <span id="BookGenre" runat="server"></span></p>
                <p><strong>Price:</strong> <span id="BookPrice" runat="server"></span></p>
                <p><strong>Pages:</strong> <span id="BookPages" runat="server"></span></p>
                <p><strong>Description:</strong> <span id="BookDescription" runat="server"></span></p>
            </div>
            <div>
                <asp:UpdatePanel ID="UpdatePanel1" runat="server">
                    <ContentTemplate>
                        <asp:LinkButton CssClass="nav-link btn btn-primary w-25 w-50 text-white p-2 mt-5 rounded-pill mb-5" ID="addToReadingList" runat="server" OnClick="AddToReadingList" Text="Add to Reading List" />
                        <asp:LinkButton CssClass="nav-link btn btn-primary w-25 w-50 text-white p-2 mt-5 rounded-pill mb-5" ID="removeFromReadingList" runat="server" OnClick="RemoveFromReadingList" Text="Remove from Reading List" />
                        <asp:Label CssClass="alert alert-success invisible w-100" ID="lblMessage" runat="server" Text="" />
                    </ContentTemplate>
                </asp:UpdatePanel>
                <asp:UpdatePanel>
                    <ContentTemplate>
                        <asp:LinkButton ID="LoginButton" runat="server" href="" class="nav-link btn btn-primary w-25 w-50 text-white p-2 mt-4 rounded-pill mb-5">Log in to make your reading list</asp:LinkButton>
                    </ContentTemplate>
                </asp:UpdatePanel>
            </div>
        </div>
    </div>
</asp:Content>

```

```

protected void Page_Load(object sender, EventArgs e)
{
    if (Request.QueryString["BookID"] == null)
    {
        RedirectManager.ToHomePage();
        return;
    }
    if (!IsPostBack)
    {
        int bookId = int.Parse(Request.QueryString["BookID"]);
        Book book = DatabaseAccess.Books.GetBookFromID(bookId);

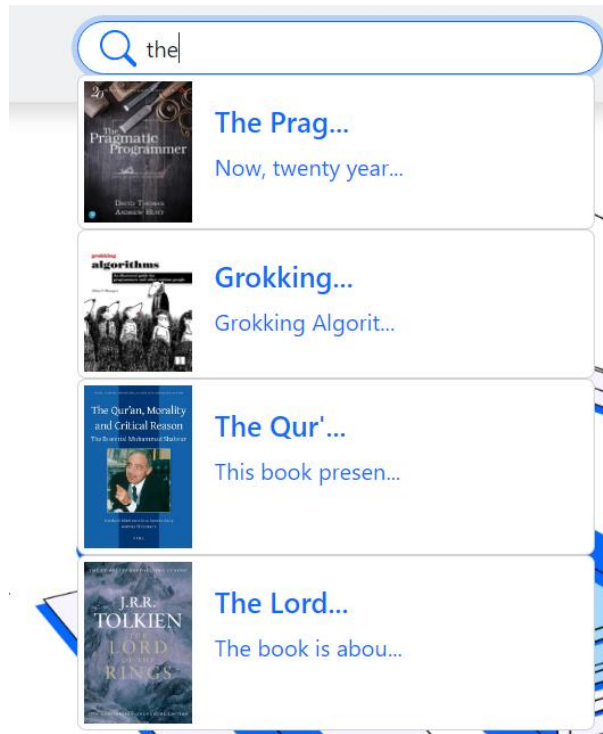
        UpdateBookDetails(book);
        UpdateReadingListButtons(bookId);
    }
}
2 references
private void UpdateReadingList(bool addToReadingList, string message)
{
    SessionManager.RedirectToLoginIfNotLoggedIn();

    int bookId = int.Parse(Request.QueryString["BookID"]);
    int userId = int.Parse(Session["UserID"].ToString());

    if (addToReadingList)
    {
        DatabaseAccess.ReadingLists.AddToReadingList(bookId, userId);
    }
    else
    {
        DatabaseAccess.ReadingLists.RemoveFromReadingList(bookId, userId);
    }
    lblMessage.CssClass = "alert alert-success visible";
    lblMessage.Text = message;
    UpdateReadingListButtons(bookId);
}
0 references
protected void AddToReadingList(object sender, EventArgs e)
{
    UpdateReadingList(true, "Book added to reading list!");
}
0 references
protected void RemoveFromReadingList(object sender, EventArgs e)
{
    UpdateReadingList(false, "Book removed from reading list!");
}

```

### 6.1.3. عمل البحث (Search Functionality)



```
<div class="position-absolute text-primary top-50 translate-middle-y " style="left: 1rem">
  <svg xmlns="http://www.w3.org/2000/svg" width="26" height="26" fill="currentColor" class="bi bi-search" viewBox="0 0 16 16">
    <path d="M11.742 10.344a6.5 6.5 0 1 0-1.397 1.398h-.001c.034.062.098.115.151.138a1 1 0 0 0 1.415-1.414l-3.85-3.85a1.007 1.007 0 0 0-.115-.115z">
  </svg>
</div>
<asp:TextBox ID="searchBar" runat="server" class="form-control mw-100 rounded-pill ps-5 border-primary" placeholder="Search for books..."></asp:TextBox>
<div id="searchResults" class="position-absolute"></div>
```



```

<script type="text/javascript">
    $(document).ready(function () {
        $('#searchBar').on('input', function () {
            $('#searchResults').css("top", $('#searchBar').position().top + $('#searchBar').outerHeight());
            $('#searchResults').css("left", $('#searchBar').position().left);
            $('#searchResults').css("width", $('#searchBar').width() + 56);
            var searchKeyword = $(this).val();
            $.post('/SearchResults.aspx?keywords=' + searchKeyword)
                .done(function (data) {
                    console.log(data);
                    $('#searchResults').empty();
                    $(data).find('Book').each(function () {
                        var id = $(this).find('Id').text();
                        var title = $(this).find('Title').text();
                        var details = $(this).find('Details').text();
                        var image = $(this).find('Image').text();
                        var body = `<div class="card w-100" style="z-index:1000">
                            <a href="/BookDetails?BookID=${id}" class="d-flex flex-row p-1 text-decoration-none">
                                
                                <div class="card-body d-block w-100">
                                    <h5 class="card-title">${title}</h5>
                                    <p class="card-text">${details}</p>
                                </div>
                            </a>
                        </div>`;
                        $('#searchResults').append(body);
                    });
                });
        });
        $(document).on("click", function () {
            $('#searchResults').empty();
        })
    });
</script>

```

```

1 reference
public partial class SearchResults : Page
{
    0 references
    protected void Page_Load(object sender, EventArgs e)
    {
        string keywords = Request.QueryString["keywords"];
        List<Db.Book> books = DatabaseAccess.Books.GetBooks(keywords);
        Response.ContentType = "text/xml";
        using (XmlWriter writer = XmlWriter.Create(Response.Output))
        {
            writer.WriteStartDocument();
            writer.WriteStartElement("Books");
            foreach (Db.Book book in books)
            {
                writer.WriteStartElement("Book");
                writer.WriteElementString("Id", book.Id.ToString());
                writer.WriteElementString("Title", (book.Name.Length < 8)? book.Name : book.Name.Substring(0,8)+"...");
                writer.WriteElementString("Image", book.Image);
                writer.WriteElementString("Details", (book.Description.Length < 16) ? book.Description : book.Description.Substring(0, 16) + "...");
                writer.WriteEndElement();
            }
            writer.WriteEndElement();
            writer.WriteEndDocument();
        }
        Response.End();
    }
}

```

## 6.1.4. قائمة قراءة المستخدم (User Reading List)




Home Reading List Admin

Search for books...

Logout

Welcome asp4ever

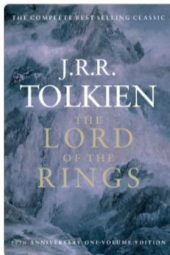
### Your book list



**The Pragmatic ...**  
Andy Hunt and ...

Description:  
Now, twenty years later, this new edition re-examines what it means to be a modern program...

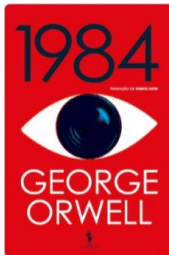
[Details](#) [Trash](#)



**The Lord of th...**  
J.R.R. Tolkien...

Description:  
The book is about power and greed, innocence and enlightenment. Ultimately, it describes a...

[Details](#) [Trash](#)



**1984**  
George Orwell

Description:  
The novel examines a futuristic society, called the World State, that revolves around scie...

[Details](#) [Trash](#)

© 2023 - Student: abubakr\_203648

```
asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">
    <h2 class="text-primary text-opacity-50 my-5 fs-4">Welcome @ SessionManager.GetUsername @ </h2>
    <asp:UpdatePanel ID="UpdatePanel1" runat="server" UpdateMode="Conditional">
        <ContentTemplate>
            <div class="container mt-5">
                <h3 class="text-primary">Your book list</h3>
                <hr class="bg-primary border-primary rounded-pill opacity-75" style="height: 6px;" />
                <div class="book-section row d-flex justify-content-between">
                    <asp:Repeater ID="ReadingListBooks" runat="server">
                        <ItemTemplate>
                            <div class="col-md-12 col-lg-4 card my-2 p-0" style="height: 300px; width: 420px">
                                <div class="d-flex p-0 h-100">
                                    <asp:Image DataField="Image" ImageUrl="@ Eval("Image") @ ID="BookImage" runat="server" CssClass="img-fluid rounded-start mw-100" Height="300" />
                                </div>
                                <div class="col-6 p-0 flex-fill w-100">
                                    <div class="card-body d-flex justify-content-between flex-column h-100 w-100">
                                        <div>
                                            <h5 class="card-title">@ (Eval("Name").ToString().Count() < 14)? Eval("Name"): Eval("Name").ToString().Substring(0,14)+"... " @ </h5>
                                            <p class="card-text fs-6 text-light">@ (Eval("Author").ToString().Count() < 14)? Eval("Author"): Eval("Author").ToString().Substring(0,14)+"... " @ </p>
                                            <p class="card-text text-secondary">Description:<br />@ (Eval("Description").ToString().Count() < 90)? Eval("Description"): Eval("Description").ToString().Substring(0,90)+"... " @ </p>
                                        </div>
                                        <div>
                                            <p class="card-text p-0 w-100">
                                                <a href="@ RedirectManager.PagesURL.BookDetailsURL @ BookID @ Eval("Id") @ class="btn btn-primary col-sm-8 col-12" Details</a>
                                                <asp:LinkButton ID="RemoveButton" runat="server" OnClick="RemoveBook" CommandArgument="@ Bind("Id") @ CssClass="btn btn-danger text-white col-sm-3 col-12">
                                                    <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor" class="bi bi-trash" viewBox="0 0 16 16">
                                                        <path d="M6.5 1h3.5 5 0 0 1 .5 5v1h6v-1a.5.5 0 0 1 -.5-.5M11 2.5v1A1.5 1.5 0 0 0 9.5 0h-3A1.5 1.5 0 0 0 8 2.5v1A1.5 1.5 0 0 0 6.5 0h-3A1.5 1.5 0 0 0 5 2.5v1A1.5 1.5 0 0 0 3.5 0h-3A1.5 1.5 0 0 0 2 2.5v1A1.5 1.5 0 0 0 0.5 2.5v1A1.5 1.5 0 0 0 0 4h3.5 5 0 0 1 .5 5v1h6v-1a.5.5 0 0 1 -.5-.5M11 5.5 5 0 0 1 5.5 5v1h6v-1a.5.5 0 0 1 -.5-.5M11 8.5 5 0 0 1 5.5 8.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 11.5 5 0 0 1 5.5 11.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 14.5 5 0 0 1 5.5 14.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 17.5 5 0 0 1 5.5 17.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 20.5 5 0 0 1 5.5 20.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 23.5 5 0 0 1 5.5 23.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 26.5 5 0 0 1 5.5 26.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 29.5 5 0 0 1 5.5 29.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 32.5 5 0 0 1 5.5 32.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 35.5 5 0 0 1 5.5 35.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 38.5 5 0 0 1 5.5 38.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 41.5 5 0 0 1 5.5 41.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 44.5 5 0 0 1 5.5 44.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 47.5 5 0 0 1 5.5 47.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 50.5 5 0 0 1 5.5 50.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 53.5 5 0 0 1 5.5 53.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 56.5 5 0 0 1 5.5 56.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 59.5 5 0 0 1 5.5 59.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 62.5 5 0 0 1 5.5 62.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 65.5 5 0 0 1 5.5 65.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 68.5 5 0 0 1 5.5 68.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 71.5 5 0 0 1 5.5 71.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 74.5 5 0 0 1 5.5 74.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 77.5 5 0 0 1 5.5 77.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 80.5 5 0 0 1 5.5 80.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 83.5 5 0 0 1 5.5 83.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 86.5 5 0 0 1 5.5 86.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 89.5 5 0 0 1 5.5 89.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 92.5 5 0 0 1 5.5 92.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 95.5 5 0 0 1 5.5 95.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 98.5 5 0 0 1 5.5 98.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 101.5 5 0 0 1 5.5 101.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 104.5 5 0 0 1 5.5 104.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 107.5 5 0 0 1 5.5 107.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 110.5 5 0 0 1 5.5 110.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 113.5 5 0 0 1 5.5 113.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 116.5 5 0 0 1 5.5 116.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 119.5 5 0 0 1 5.5 119.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 122.5 5 0 0 1 5.5 122.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 125.5 5 0 0 1 5.5 125.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 128.5 5 0 0 1 5.5 128.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 131.5 5 0 0 1 5.5 131.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 134.5 5 0 0 1 5.5 134.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 137.5 5 0 0 1 5.5 137.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 140.5 5 0 0 1 5.5 140.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 143.5 5 0 0 1 5.5 143.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 146.5 5 0 0 1 5.5 146.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 149.5 5 0 0 1 5.5 149.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 152.5 5 0 0 1 5.5 152.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 155.5 5 0 0 1 5.5 155.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 158.5 5 0 0 1 5.5 158.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 161.5 5 0 0 1 5.5 161.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 164.5 5 0 0 1 5.5 164.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 167.5 5 0 0 1 5.5 167.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 170.5 5 0 0 1 5.5 170.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 173.5 5 0 0 1 5.5 173.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 176.5 5 0 0 1 5.5 176.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 179.5 5 0 0 1 5.5 179.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 182.5 5 0 0 1 5.5 182.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 185.5 5 0 0 1 5.5 185.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 188.5 5 0 0 1 5.5 188.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 191.5 5 0 0 1 5.5 191.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 194.5 5 0 0 1 5.5 194.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 197.5 5 0 0 1 5.5 197.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 200.5 5 0 0 1 5.5 200.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 203.5 5 0 0 1 5.5 203.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 206.5 5 0 0 1 5.5 206.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 209.5 5 0 0 1 5.5 209.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 212.5 5 0 0 1 5.5 212.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 215.5 5 0 0 1 5.5 215.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 218.5 5 0 0 1 5.5 218.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 221.5 5 0 0 1 5.5 221.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 224.5 5 0 0 1 5.5 224.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 227.5 5 0 0 1 5.5 227.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 230.5 5 0 0 1 5.5 230.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 233.5 5 0 0 1 5.5 233.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 236.5 5 0 0 1 5.5 236.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 239.5 5 0 0 1 5.5 239.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 242.5 5 0 0 1 5.5 242.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 245.5 5 0 0 1 5.5 245.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 248.5 5 0 0 1 5.5 248.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 251.5 5 0 0 1 5.5 251.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 254.5 5 0 0 1 5.5 254.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 257.5 5 0 0 1 5.5 257.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 260.5 5 0 0 1 5.5 260.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 263.5 5 0 0 1 5.5 263.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 266.5 5 0 0 1 5.5 266.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 269.5 5 0 0 1 5.5 269.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 272.5 5 0 0 1 5.5 272.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 275.5 5 0 0 1 5.5 275.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 278.5 5 0 0 1 5.5 278.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 281.5 5 0 0 1 5.5 281.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 284.5 5 0 0 1 5.5 284.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 287.5 5 0 0 1 5.5 287.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 290.5 5 0 0 1 5.5 290.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 293.5 5 0 0 1 5.5 293.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 296.5 5 0 0 1 5.5 296.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 299.5 5 0 0 1 5.5 299.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 302.5 5 0 0 1 5.5 302.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 305.5 5 0 0 1 5.5 305.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 308.5 5 0 0 1 5.5 308.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 311.5 5 0 0 1 5.5 311.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 314.5 5 0 0 1 5.5 314.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 317.5 5 0 0 1 5.5 317.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 320.5 5 0 0 1 5.5 320.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 323.5 5 0 0 1 5.5 323.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 326.5 5 0 0 1 5.5 326.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 329.5 5 0 0 1 5.5 329.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 332.5 5 0 0 1 5.5 332.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 335.5 5 0 0 1 5.5 335.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 338.5 5 0 0 1 5.5 338.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 341.5 5 0 0 1 5.5 341.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 344.5 5 0 0 1 5.5 344.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 347.5 5 0 0 1 5.5 347.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 350.5 5 0 0 1 5.5 350.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 353.5 5 0 0 1 5.5 353.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 356.5 5 0 0 1 5.5 356.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 359.5 5 0 0 1 5.5 359.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 362.5 5 0 0 1 5.5 362.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 365.5 5 0 0 1 5.5 365.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 368.5 5 0 0 1 5.5 368.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 371.5 5 0 0 1 5.5 371.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 374.5 5 0 0 1 5.5 374.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 377.5 5 0 0 1 5.5 377.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 380.5 5 0 0 1 5.5 380.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 383.5 5 0 0 1 5.5 383.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 386.5 5 0 0 1 5.5 386.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 389.5 5 0 0 1 5.5 389.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 392.5 5 0 0 1 5.5 392.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 395.5 5 0 0 1 5.5 395.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 398.5 5 0 0 1 5.5 398.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 401.5 5 0 0 1 5.5 401.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 404.5 5 0 0 1 5.5 404.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 407.5 5 0 0 1 5.5 407.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 410.5 5 0 0 1 5.5 410.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 413.5 5 0 0 1 5.5 413.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 416.5 5 0 0 1 5.5 416.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 419.5 5 0 0 1 5.5 419.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 422.5 5 0 0 1 5.5 422.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 425.5 5 0 0 1 5.5 425.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 428.5 5 0 0 1 5.5 428.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 431.5 5 0 0 1 5.5 431.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 434.5 5 0 0 1 5.5 434.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 437.5 5 0 0 1 5.5 437.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 440.5 5 0 0 1 5.5 440.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 443.5 5 0 0 1 5.5 443.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 446.5 5 0 0 1 5.5 446.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 449.5 5 0 0 1 5.5 449.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 452.5 5 0 0 1 5.5 452.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 455.5 5 0 0 1 5.5 455.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 458.5 5 0 0 1 5.5 458.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 461.5 5 0 0 1 5.5 461.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 464.5 5 0 0 1 5.5 464.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 467.5 5 0 0 1 5.5 467.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 470.5 5 0 0 1 5.5 470.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 473.5 5 0 0 1 5.5 473.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 476.5 5 0 0 1 5.5 476.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 479.5 5 0 0 1 5.5 479.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 482.5 5 0 0 1 5.5 482.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 485.5 5 0 0 1 5.5 485.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 488.5 5 0 0 1 5.5 488.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 491.5 5 0 0 1 5.5 491.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 494.5 5 0 0 1 5.5 494.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 497.5 5 0 0 1 5.5 497.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 500.5 5 0 0 1 5.5 500.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 503.5 5 0 0 1 5.5 503.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 506.5 5 0 0 1 5.5 506.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 509.5 5 0 0 1 5.5 509.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 512.5 5 0 0 1 5.5 512.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 515.5 5 0 0 1 5.5 515.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 518.5 5 0 0 1 5.5 518.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 521.5 5 0 0 1 5.5 521.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 524.5 5 0 0 1 5.5 524.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 527.5 5 0 0 1 5.5 527.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 530.5 5 0 0 1 5.5 530.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 533.5 5 0 0 1 5.5 533.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 536.5 5 0 0 1 5.5 536.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 539.5 5 0 0 1 5.5 539.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 542.5 5 0 0 1 5.5 542.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 545.5 5 0 0 1 5.5 545.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 548.5 5 0 0 1 5.5 548.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 551.5 5 0 0 1 5.5 551.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 554.5 5 0 0 1 5.5 554.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 557.5 5 0 0 1 5.5 557.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 560.5 5 0 0 1 5.5 560.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 563.5 5 0 0 1 5.5 563.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 566.5 5 0 0 1 5.5 566.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 569.5 5 0 0 1 5.5 569.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 572.5 5 0 0 1 5.5 572.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 575.5 5 0 0 1 5.5 575.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 578.5 5 0 0 1 5.5 578.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 581.5 5 0 0 1 5.5 581.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 584.5 5 0 0 1 5.5 584.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 587.5 5 0 0 1 5.5 587.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 590.5 5 0 0 1 5.5 590.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 593.5 5 0 0 1 5.5 593.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 596.5 5 0 0 1 5.5 596.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 599.5 5 0 0 1 5.5 599.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 602.5 5 0 0 1 5.5 602.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 605.5 5 0 0 1 5.5 605.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 608.5 5 0 0 1 5.5 608.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 611.5 5 0 0 1 5.5 611.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 614.5 5 0 0 1 5.5 614.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 617.5 5 0 0 1 5.5 617.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 620.5 5 0 0 1 5.5 620.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 623.5 5 0 0 1 5.5 623.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 626.5 5 0 0 1 5.5 626.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 629.5 5 0 0 1 5.5 629.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 632.5 5 0 0 1 5.5 632.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 635.5 5 0 0 1 5.5 635.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 638.5 5 0 0 1 5.5 638.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 641.5 5 0 0 1 5.5 641.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 644.5 5 0 0 1 5.5 644.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 647.5 5 0 0 1 5.5 647.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 650.5 5 0 0 1 5.5 650.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 653.5 5 0 0 1 5.5 653.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 656.5 5 0 0 1 5.5 656.5v1h6v-1a.5.5 0 0 1 -.5-.5M11 659.5 
```

2 references

```
public partial class ReadingList : System.Web.UI.Page
```

```
{
```

0 references

```
protected void Page_Load(object sender, EventArgs e)
```

```
{
```

```
    SessionManager.RedirectToLoginIfNotLoggedIn();
```

```
    if (!IsPostBack)
```

```
    {
```

```
        BindGrid();
```

```
    }
```

```
}
```

2 references

```
private void BindGrid()
```

```
{
```

```
    int userId = Convert.ToInt32(SessionManager.GetUserID);
```

```
    ReadingListBooks.DataSource = DatabaseAccess.Books.GetBooksFromUserID(userId);
```

```
    ReadingListBooks.DataBind();
```

```
}
```

0 references

```
protected void RemoveBook(object sender, EventArgs e)
```

```
{
```

```
    SessionManager.RedirectToLoginIfNotLoggedIn();
```

```
    int bookId = Convert.ToInt32(((LinkButton)sender).CommandArgument);
```

```
    int userId = Convert.ToInt32(SessionManager.GetUserID);
```


```
    DatabaseAccess.ReadingLists.RemoveFromReadingList(bookId, userId);
```

```
    BindGrid();
```

```
}
```

```
}
```

## 6.1.5 . صفحة المشرف (Admin Section)

[Home](#) [Reading List](#) [Admin](#)


Search for books...

Logout

### Welcome asp4ever

#### Leatest Books


Add Book



**Clean Code**  
Robert C. Mart...

Description:  
Even bad code can function. But if code ...

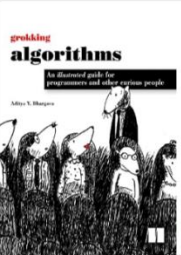
Details



**The Pragmatic ...**  
Andy Hunt and ...

Description:  
Now, twenty years later, this new editio...

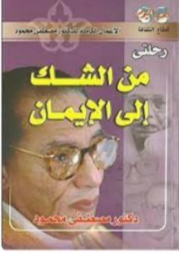
Details



**Grokking Algor...**  
Aditya Y. Bhar...

Description:  
Grokking Algorithms is a friendly take o...

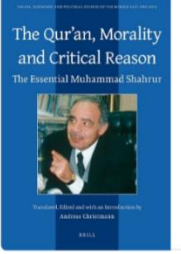
Details



**My Journey fro...**  
Mustafa Mahmou...

Description:  
Everything in today's world carries with...

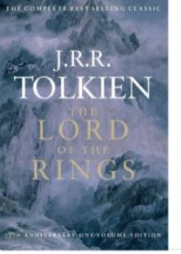
Details



**The Qur'an, Morality and Critical Reason**  
The Essential Muhammad Shahrur

Description:  
This book presents the work and ideas of...

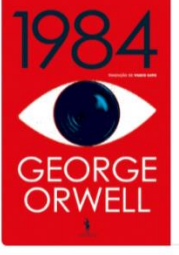
Details



**The Lord of th...**  
J.R.R. Tolkien...

Description:  
The book is about power and greed, innoc...

Details



**1984**  
George Orwell

Description:  
The novel examines a futuristic society,...

Details

© 2023 - Student: abubakr\_203648





## Add Book

Title

e.g. The lord of the rings...

Author

e.g. J. R. R. Tolkien...

Description

the saga of a group of sometimes reluctant heroes who set forth to save their world from consummate evil...

Image

Choose File

No file chosen

Genre

e.g. Fantasy

Price

e.g. 100\$

Pages

e.g. 300

Add

```
2 references
public partial class AddBook : Page
{
    0 references
    protected void Page_Load(object sender, EventArgs e)
    {
        SessionManager.RedirectToLoginIfNotAdmin();
    }
    0 references
    protected void AddButton_Click(object sender, EventArgs e)
    {
        Page.Validate();
        if (Page.IsValid)
        {
            var name = NameTextBox.Text;
            var description = DescriptionTextBox.Text;
            var imagePath = ImageFileUpload.FileName;

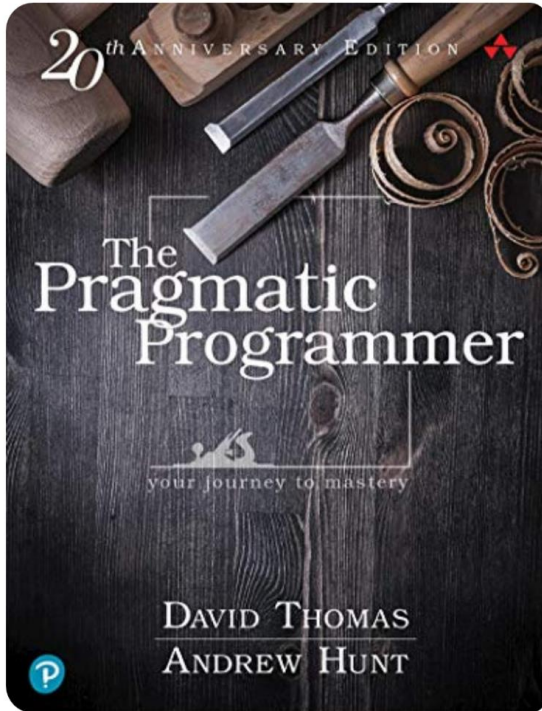
            if (ImageFileUpload.HasFile)
            {
                string fileName = Path.GetFileName(ImageFileUpload.PostedFile.FileName);
                ImageFileUpload.PostedFile.SaveAs(Server.MapPath("~/images/") + fileName);
                imagePath = "~/images/" + fileName;
            }

            var author = AuthorTextBox.Text;
            var genre = GenreTextBox.Text;
            var price = Convert.ToDecimal(PriceTextBox.Text);
            var pages = Convert.ToInt32(PagesTextBox.Text);

            var book = new Book(0, name, description, imagePath, author, genre, price, pages);
            DatabaseAccess.Books.AddBook(book);
            RedirectManager.ToAdminPage();
        }
        else
        {
            // Input is not valid, show error message
            ErrorMessageLabel.Text = "Please enter valid input.";
        }
    }
}
```



## Edit Book



Title

The Pragmatic Programmer: Your Journey to Mastery, 20th Anniversary Edition

Author

Andy Hunt and Dave Thomas

Description

Now, twenty years later, this new edition re-examines what it means to be a modern programmer. Topics range from personal responsibility and career development to architectural techniques for keeping your code flexible and easy to adapt and reuse. Read this book, and you'll learn how to: Fight software rot.

Image

No file chosen

Genre

Reference work

Price

35.00

Pages

334

```

2 references
public partial class EditBook : Page
{
    0 references
    protected void Page_Load(object sender, EventArgs e)
    {
        SessionManager.RedirectToLoginIfNotAdmin();
        if (!IsPostBack)
        {
            int id = Convert.ToInt32(Request.QueryString["id"]);

            Book book = DatabaseAccess.Books.GetBookFromID(id);

            IdLabel.Text = book.Id.ToString();
            NameTextBox.Text = book.Name;
            DescriptionTextBox.Text = book.Description;
            BookImage.ImageUrl = book.Image;
            AuthorTextBox.Text = book.Author;
            GenreTextBox.Text = book.Genre;
            PriceTextBox.Text = book.Price.ToString();
            PagesTextBox.Text = book.Pages.ToString();
        }
    }

    0 references
    protected void UpdateButton_Click(object sender, EventArgs e)
    {
        Page.Validate();
        if (Page.IsValid)
        {
            int id = Convert.ToInt32(IdLabel.Text);
            string imagePath = BookImage.ImageUrl.ToString();

            if (ImageFileUpload.HasFile)
            {
                string fileName = Path.GetFileName(ImageFileUpload.PostedFile.FileName);
                ImageFileUpload.PostedFile.SaveAs(Server.MapPath("~/images/") + fileName);
                imagePath = "~/images/" + fileName;
            }

            var name = NameTextBox.Text;
            var description = DescriptionTextBox.Text;
            var author = AuthorTextBox.Text;
            var genre = GenreTextBox.Text;
            var price = Convert.ToDecimal(PriceTextBox.Text);
            var pages = Convert.ToInt32(PagesTextBox.Text);

            var book = new Book(id, name, description, imagePath, author, genre, price, pages);
            DatabaseAccess.Books.UpdateBook(book);
            RedirectManager.ToAdminPage();
        }
        else
        {
            ErrorMessageLabel.Text = "Please enter valid input.";
        }
    }
}

```



[Home](#) [Reading List](#) [Admin](#)

[Logout](#)

## Delete Book

Are you sure you want to delete the book: The Pragmatic Programmer: Your Journey to Mastery, 20th Anniversary Edition?

[Confirm](#)

[Cancel](#)



2 references

```
public partial class DeleteBook : Page
```

```
{
```

0 references

```
protected void Page_Load(object sender, EventArgs e)
```

```
{
```

```
    SessionManager.RedirectToLoginIfNotAdmin();
```

```
    if (!IsPostBack)
```

```
    {
```

```
        int id = Convert.ToInt32(Request.QueryString["id"]);
```

```
        Db.Book book = DatabaseAccess.Books.GetBookFromID(id);
```

```
        NameLabel.Text = "Are you sure you want to delete the book: " + book.Name + "?";
```

```
    }
```

```
}
```

0 references

```
protected void ConfirmButton_Click(object sender, EventArgs e)
```

```
{
```

```
    SessionManager.RedirectToLoginIfNotLoggedIn();
```

```
    int id = Convert.ToInt32(Request.QueryString["id"]);
```

```
    DatabaseAccess.Books.DeleteBookByID(id);
```

```
    RedirectManager.ToAdminPage();
```

```
}
```

0 references

```
protected void CancelButton_Click(object sender, EventArgs e)
```

```
{
```

```
    RedirectManager.ToAdminPage();
```

```
}
```

```
}
```

## 6.1.6. تكامل قاعدة البيانات (Database Integration)

ABUBAKRALSHAIKH_OBIS - dbo.Users			
Column Name	Data Type	Allow Nulls	
id	int	<input type="checkbox"/>	
username	nvarchar(255)	<input type="checkbox"/>	
email	nvarchar(255)	<input type="checkbox"/>	
password	nvarchar(255)	<input type="checkbox"/>	
role	nvarchar(255)	<input type="checkbox"/>	

ABUBAKRALSHAIKH_OBIS - dbo.ReadingList			
Column Name	Data Type	Allow Nulls	
id	int	<input type="checkbox"/>	
book_id	int	<input type="checkbox"/>	
user_id	int	<input type="checkbox"/>	

ABUBAKRALSHAIKH_OBIS - dbo.Books			
Column Name	Data Type	Allow Nulls	
id	int	<input type="checkbox"/>	
name	nvarchar(255)	<input type="checkbox"/>	
description	nvarchar(MAX)	<input type="checkbox"/>	
image	nvarchar(255)	<input type="checkbox"/>	
author	nvarchar(255)	<input type="checkbox"/>	
genre	nvarchar(255)	<input type="checkbox"/>	
price	decimal(10, 2)	<input type="checkbox"/>	
pages	int	<input type="checkbox"/>	

```

18 references
public class DatabaseAccess
{
    public readonly static string _connectionString = ConfigurationManager.ConnectionStrings["OBIS"].ConnectionString;
    4 references
    public static class ErrorMessages[...]
    4 references
    private static void ExecuteNonQuery(string query, SqlParameter[] parameters)[...]
    10 references
    public static class Books
    {
        4 references
        private static List<Book> ExecuteBookQuery(string query, SqlParameter[] parameters = null)[...]
        2 references
        public static List<Book> GetBooks()[...]
        1 reference
        public static List<Book> GetBooks(string search)[...]
        1 reference
        public static List<Book> GetBooksFromUserID(int userId)[...]
        3 references
        public static Book GetBookFromID(int id)[...]
        1 reference
        public static void DeleteBookByID(int id)[...]
        1 reference
        public static void UpdateBook(Book book)[...]
        1 reference
        public static void AddBook(Book book)[...]
    }
    4 references
    public class Users
    {
        2 references
        private static User ExecuteUserQuery(string query, SqlParameter[] parameters = null)[...]
        1 reference
        public static User GetUser(string usernameOrEmail, string password)[...]
        1 reference
        public static User AddUser(string username, string email, string password, string role)[...]
        1 reference
        public static bool IsUsernameTaken(string username)[...]
        1 reference
        public static bool IsEmailTaken(string email)[...]
        2 references
        private static bool IsFieldTaken(string field, string value)[...]
        0 references
        private static void DeleteAllUsers()[...]
    }
    5 references
    public static class ReadingLists
    {
        4 references
        private static List<ReadingList> ExecuteReadingListQuery(string query, SqlParameter[] parameters = null)[...]
        1 reference
        public static List<ReadingList> GetReadingLists(int userId)[...]
        1 reference
        public static void AddToReadingList(int bookId, int userId)[...]
        2 references
        public static void RemoveFromReadingList(int bookId, int userId)[...]
        1 reference
        public static bool IsInReadingList(int bookId, int userId)[...]
    }
}


```

```

public static class Db
{
    23 references
    public class Book
    {
        4 references
        public int Id { get; private set; }
        9 references
        public string Name { get; private set; }
        8 references
        public string Description { get; private set; }
        6 references
        public string Image { get; private set; }
        5 references
        public string Author { get; private set; }
        5 references
        public string Genre { get; private set; }
        5 references
        public decimal Price { get; private set; }
        5 references
        public int Pages { get; private set; }
        3 references
        public Book(int id, string name, string description, string image, string author, string genre, decimal price, int pages) ...
    }
    7 references
    public class User
    {
        3 references
        public int Id { get; set; }
        3 references
        public string Username { get; set; }
        1 reference
        public string Email { get; set; }
        1 reference
        public string Password { get; set; }
        3 references
        public string Role { get; set; }
        1 reference
        public User(int id, string username, string email, string password, string role) ...
    }
    8 references
    public class ReadingList
    {
        1 reference
        public int Id { get; private set; }
        2 references
        public int BookId { get; private set; }
        1 reference
        public int UserId { get; private set; }
        1 reference
        public ReadingList(int id, int bookId, int userId) ...
    }
}

```

## 6.1.7 المصادقة والتخويل (Authentication and Authorization)


[Home](#)
[Reading List](#)

[Signup](#)
[Login](#)

### Welcom Back

Enter your email and password

Email / Username:

Email or Username is required.

Password:

Password is required.

Login

```

<asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">
    <div class="row justify-content-center align-items-center my-5 h-100">
        <div class="form w-50 w-sm-75">
            <div class="text-center">
                <h2>Welcome Back</h2>
                <p class="text-secondary fs-5 mb-5">Enter your email and password</p>
            </div>
            <div class="form-group text-secondary">
                <asp:Label ID="usernameEmailLabel" runat="server" Text="Email / Username:" CssClass="control-label"></asp:Label>
                <asp:TextBox AutoCompleteType="Email" ID="usernameEmailTextBox" runat="server" CssClass="form-control border-primary rounded-pill" placeholder="Example@gmail.com..."></asp:TextBox>
                <asp:RequiredFieldValidator ControlToValidate="usernameEmailTextBox" ErrorMessage="Email or Username is required." runat="server" CssClass="text-danger" />
            </div>
            <br />
            <div class="form-group text-secondary">
                <asp:Label ID="passwordLabel" runat="server" Text="Password:" CssClass="control-label"></asp:Label>
                <asp:TextBox ID="passwordTextBox" runat="server" TextMode="Password" CssClass="form-control border-primary rounded-pill" placeholder="password..."></asp:TextBox>
                <asp:RequiredFieldValidator ControlToValidate="passwordTextBox" ErrorMessage="Password is required." runat="server" CssClass="text-danger" />
            </div>
            <br />
            <asp:Button ID="loginButton" runat="server" Text="Login" OnClick="LoginButton_Click" CssClass="btn btn-primary rounded-pill px-4 w-25 mb-5" />
            <br />
            <asp:Label ID="ErrorMessageLabel" runat="server" ForeColor="Red" CssClass="invisible" />
        </div>
    </div>
</asp:Content>

```

```

2 references
public partial class Login : System.Web.UI.Page
{
    0 references
    protected void Page_Load(object sender, EventArgs e)
    {
        SessionManager.RedirectToHomeIfLoggedIn();
    }
    0 references
    protected void LoginButton_Click(object sender, EventArgs e)
    {
        Page.Validate();
        if (Page.IsValid)
        {
            var usernameOrEmail = usernameEmailTextBox.Text;
            var password = passwordTextBox.Text;

            Db.User user = DatabaseAccess.Users.GetUser(usernameOrEmail, password);

            if (user != null)
            {
                SessionManager.LoginUser(user.Id.ToString(), user.Username, user.Role);
                if (user.Role == "admin")
                {
                    RedirectManager.ToAdminPage();
                }
                else
                {
                    RedirectManager.ToHomePage();
                }
            }
            else
            {
                ErrorMessageLabel.CssClass = "alert alert-danger visible";
                ErrorMessageLabel.Text = "Invalid username/email or password. Please try again.";
            }
        }
        else
        {
            ErrorMessageLabel.Text = "Please enter valid input.";
        }
    }
}

```



## Sign up

Enter your information to continue with us

Username:

Enter your name...

Email:

Example@gmail.com...

Password:

password...

Sign Up

© 2023 - Student: abubaki\_203648

```
<asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">
    <div class="row justify-content-center align-items-center my-5 h-100">
        <div class="form w-50 w-sm-75">
            <div class="text-center">
                <h2>Sign up</h2>
                <p class="text-secondary fs-5 mb-5">Enter your information to continue with us</p>
            </div>
            <div class="form-group">
                <label for="UsernameTextBox">Username:</label>
                <asp:TextBox ID="UsernameTextBox" runat="server" Placeholder="Enter your name..." CssClass="form-control border-primary rounded-pill" />
                <asp:RequiredFieldValidator ControlToValidate="UsernameTextBox" ErrorMessage="Username is required." runat="server" CssClass="text-danger" />
            </div>
            <div class="form-group">
                <label for="EmailTextBox">Email:</label>
                <asp:TextBox ID="EmailTextBox" runat="server" Placeholder="Example@gmail.com..." CssClass="form-control border-primary rounded-pill" />
                <asp:RequiredFieldValidator ControlToValidate="EmailTextBox" ErrorMessage="Email is required." runat="server" CssClass="text-danger" />
                <asp:RegularExpressionValidator ControlToValidate="EmailTextBox" ValidationExpression="\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*" ErrorMessage="Email is not valid." runat="server" CssClass="text-danger" />
            </div>
            <div class="form-group">
                <label for="PasswordTextBox">Password:</label>
                <asp:TextBox ID="PasswordTextBox" runat="server" TextMode="Password" Placeholder="password..." CssClass="form-control border-primary rounded-pill" />
                <asp:RequiredFieldValidator ControlToValidate="PasswordTextBox" ErrorMessage="Password is required." runat="server" CssClass="text-danger" />
                <asp:RegularExpressionValidator ControlToValidate="PasswordTextBox" ValidationExpression="^(?=.*[0-9])(?=.*[!@#$%^&*~`{|}~\.\-_\+])" ErrorMessage="Password must be at least 8 characters long." runat="server" CssClass="text-danger" />
            </div>
            <div class="form-group">
                <label for="RoleDropDownList">Role(it's for testing):</label>
                <asp:DropDownList ID="RoleDropDownList" runat="server" CssClass="form-control border-primary rounded-pill">
                    <asp:ListItem Value="user">user</asp:ListItem>
                    <asp:ListItem Value="admin">admin</asp:ListItem>
                </asp:DropDownList>
            </div>
            <asp:Button ID="SignupButton" runat="server" Text="Sign Up" OnClick="SignupButton_Click" CssClass="btn btn-primary rounded-pill px-4 w-25 my-5" />
            <asp:Label ID="ErrorMessageLabel" runat="server" ForeColor="Red" CssClass="invisible" />
        </div>
    </div>
</asp:Content>
```

```

2 references
public partial class Signup : System.Web.UI.Page
{
    0 references
    protected void Page_Load(object sender, EventArgs e)
    {
        SessionManager.RedirectToHomeIfLoggedIn();
    }

    0 references
    protected void SignupButton_Click(object sender, EventArgs e)
    {
        Page.Validate();
        if (Page.IsValid)
        {
            string username = UsernameTextBox.Text;
            string email = EmailTextBox.Text;
            string password = PasswordTextBox.Text;

            if (username.Length > 50)
            {
                ShowErrorLable();
                ErrorMessageLabel.Text = "Your name is so long.";
            }
            if (password.Length < 8)
            {
                ShowErrorLable();
                ErrorMessageLabel.Text = "Password must be at least 8 characters long.";
            }
            else if (DatabaseAccess.Users.IsUsernameTaken(username))
            {
                ShowErrorLable();
                ErrorMessageLabel.Text = "Username is already in use. Please try again.";
            }
            else if (DatabaseAccess.Users.IsEmailTaken(email))
            {
                ShowErrorLable();
                ErrorMessageLabel.Text = "Email is already in use. Please try again.";
            }
            else
            {
                // Username and email are not in use, proceed with storing in the database
                string role = RoleDropDownList.Text;
                DatabaseAccess.Users.AddUser(username, email, password, role);

                RedirectManager.ToHomePage();
            }
        }
        else
        {
            ShowErrorLable();
            ErrorMessageLabel.Text = "Please enter valid input.";
        }
    }
}
5 references

```

16 references

```
public class RedirectManager
```

```
{
```

8 references

```
public static class PagesURL...
```

5 references

```
public static void ToHomePage()
```

```
{
```

```
    HttpContext.Current.Response.Redirect(PagesURL.HomeURL);
```

```
}
```

2 references

```
public static void ToLoginForm()
```

```
{
```

```
    HttpContext.Current.Response.Redirect(PagesURL.LoginURL);
```

```
}
```

0 references

```
public static void ToSignupForm()
```

```
{
```

```
    HttpContext.Current.Response.Redirect(PagesURL.SignupURL);
```

```
}
```

5 references

```
public static void ToAdminPage()
```

```
{
```

```
    HttpContext.Current.Response.Redirect(PagesURL.AdminURL);
```

```
}
```

```
}
```

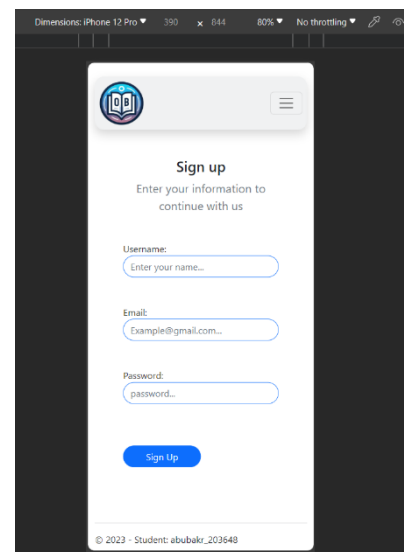
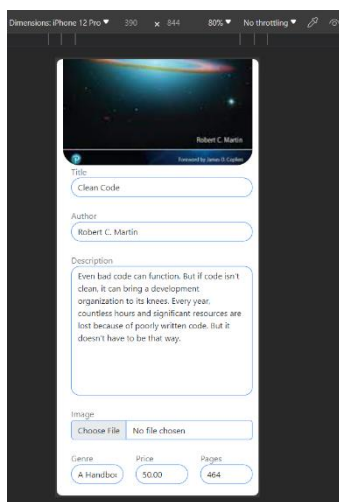
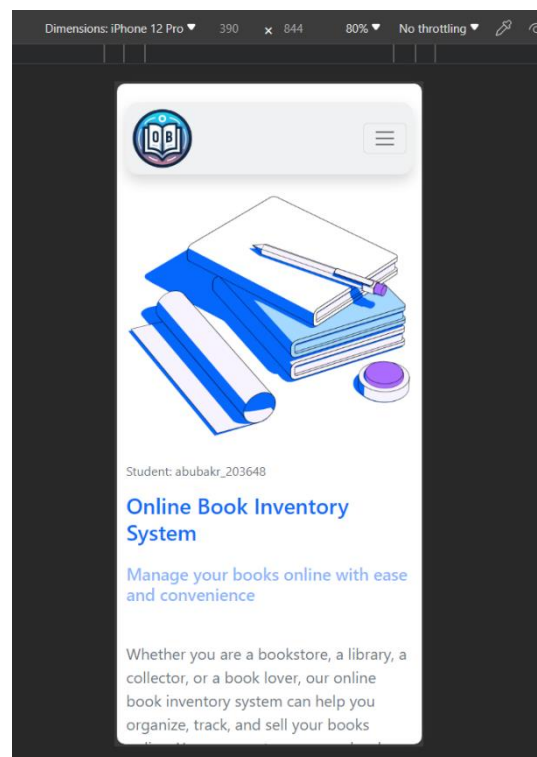
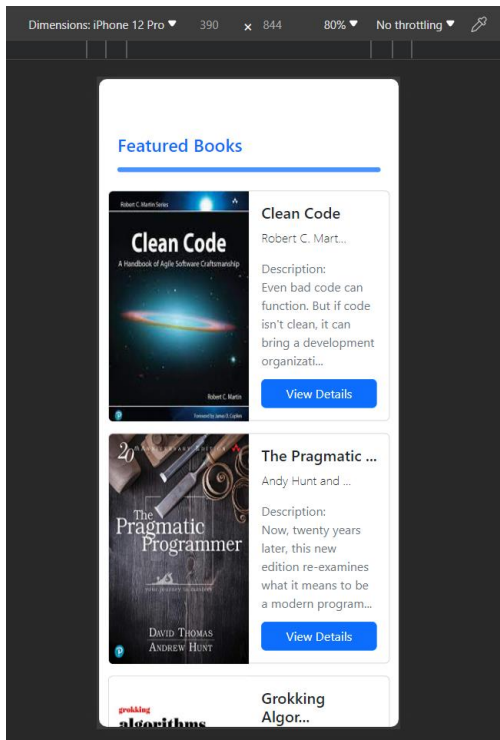
```

18 references
public static class SessionManager
{
    4 references
    public static bool IsUserLoggedIn ...
    1 reference
    public static string GetUserRole ...
    1 reference
    public static bool IsUserAdmin ...
    3 references
    public static int GetUserID ...
    1 reference
    public static string GetUsername ...
    2 references
    public static void LoginUser(string userId, string username, string role)
    {
        HttpContext.Current.Session["UserID"] = userId;
        HttpContext.Current.Session["username"] = username;
        HttpContext.Current.Session["role"] = role;
    }
    1 reference
    public static void LogoutUser()
    {
        HttpContext.Current.Session.Clear();
        RedirectManager.ToHomePage();
    }
    4 references
    public static void RedirectToLoginIfNotLoggedIn()
    {
        if (!IsUserLoggedIn)
        {
            RedirectManager.ToLoginForm();
        }
    }
    2 references
    public static void RedirectToHomeIfLoggedIn()
    {
        if (IsUserLoggedIn)
        {
            RedirectManager.ToHomePage();
        }
    }
    4 references
    public static void RedirectToLoginIfNotAdmin()
    {
        if (!IsUserAdmin)
        {
            RedirectManager.ToLoginForm();
        }
    }
}

```



## 6.1.8. التصميم المتجاوب (Responsive Design)



## 6.2. مراجع

- 1- رابط الموقع: [Home Page \(b-site.net\)](https://b-site.net)
- 2- رابط تصميم الموقع في Figma: [Online Book Inventory System – Figma](#)
- 3- مقطع في اليوتيوب ساعدني في رفع الموقع على الانترنت: [How to Upload Your Asp.Net Project For Free At FreeAspHosting | Tutorial4You \(youtube.com\)](#)