

Inventory Management System - نظام إدارة المخزون

دليل البدء السريع

الفصل 1: نظرة عامة على المشروع

1.1 وصف المشروع

نظام متكامل لإدارة المخزون يوفر حلاً شاملاً للمشاريع والشركات من جميع الأحجام. يدعم النظام إدارة المنتجات والمخزون والطلبات والموردين والمستودعات مع تتبع دقيق للباركود وتاريخ الصلاحية.

1.2 الميزات الرئيسية

- إدارة المنتجات: إضافة وتحديث وحذف المنتجات مع معلومات كاملة
- نظام الباركود: توليد وطباعة بطاقات باركود لكل منتج
- تتبع الصلاحية: تنبيهات تلقائية للمنتجات التي تنتهي صلاحيتها
- إدارة المخزون: تتبع دقيق للكميات في كل مستودع
- طلبات الشراء: إدارة متكاملة لطلبات الشراء من الموردين
- التقارير والتحليلات: تقارير شاملة عن المخزون والحركات
- سجل التدقيق: تسجيل كامل لجميع العمليات والتغييرات
- نظام التنبيهات: تنبيهات فورية عند حدوث أحداث مهمة

1.3 البنية المعمارية

:مع الطبقات التالية (MVC (Model-View-Controller يتبع المشروع نمط

- View Layer: واجهات المستخدم والعروض
- Controller Layer: معالجة الطلبات والمنطق
- Model Layer: إدارة البيانات والعمليات
- Database Layer: قاعدة البيانات MySQL

الفصل 2: متطلبات النظام

2.1 البرامج والتقنيات المطلوبة

- إصدار 7.4 أو أحدث: PHP
- إصدار 5.7 أو أحدث: MySQL/MariaDB
- خادم ويب: Apache

- **Composer:** مدير الحزم

2.2 المكتبات المقترحة

لإرسال البريد الإلكتروني - phpMailer
محرك القوالب - Twig
للاتصال الآمن بقاعدة البيانات - PDO
توليد الباركود - Barcode Generator
إنشاء ملفات PDF - TCPDF
جداول ديناميكية - DataTables
CSS إطار عمل - Bootstrap

الفصل 3: البيانات الأساسية

3.1 الكيانات الرئيسية (17 جدول)

(المنتجات) Products جدول

الحقول الأساسية:

- معرف المنتج (مفتاح أساسي): ProductID
- اسم المنتج: ProductName
- كود الباركود الفريد: Barcode
- رمز المخزون: SKU
- الوصف: Description
- معرف الفئة: CategoryID
- معرف المورد: SupplierID
- سعر الوحدة: UnitPrice
- تكلفة الشراء: CostPrice
- الحد الأدنى للتخزين: ReorderLevel
- كمية إعادة الطلب: ReorderQuantity
- تاريخ انتهاء الصلاحية: ExpirationDate
- تاريخ التصنيع: ManufacturingDate
- مدة الضمان: WarrantyMonths
- حالة النشاط: IsActive
- تاريخ الإنشاء: CreatedAt
- تاريخ التحديث: UpdatedAt

جدول Stock (المخزون)

يخزن معلومات الكميات في كل مستودع:

- StockID: معرف الجرد
- ProductID: معرف المنتج
- WarehouseID: معرف المستودع
- Quantity: الكمية الكلية
- ReservedQuantity: الكمية المحجوزة
- AvailableQuantity: الكمية المتاحة (محسوبة)
- BatchNumber: رقم الدفعة
- ExpirationDate: تاريخ الصلاحية
- LastCountDate: تاريخ آخر جرد

جدول StockMovements (حركات المخزون)

تسجيل كل حركة مخزون:

- MovementID: معرف الحركة
- ProductID: معرف المنتج
- MovementType: نوع الحركة (IN, OUT, ADJUSTMENT, RETURN, DAMAGED)
- Quantity: الكمية
- Notes: ملاحظات
- MovedBy: معرف المستخدم

جدول PurchaseOrders (طلبات الشراء)

معلومات طلبات الشراء:

- OrderID: معرف الطلب
- OrderNumber: رقم الطلب الفريد
- SupplierID: معرف المورد
- OrderDate: تاريخ الطلب
- ExpectedDeliveryDate: تاريخ التسليم المتوقع
- Status: حالة الطلب
- TotalAmount: المبلغ الإجمالي

جدول GoodsReceipts (استقبال البضائع)

تسجيل استقبال السلع:

- ReceiptID: معرف الاستقبال
- OrderID: معرف الطلب
- WarehouseID: المستودع المستقبل
- ReceiptDate: تاريخ الاستقبال
- Status: حالة الاستقبال

جدول Warehouses (المستودعات)

معلومات المستودعات:

- WarehouseID: معرف المستودع
- WarehouseName: اسم المستودع
- Location: الموقع
- Capacity: الطاقة الاستيعابية
- Manager: مسؤول المستودع

جدول Categories (الفئات)

تصنيف المنتجات:

- CategoryID: معرف الفئة
- CategoryName: اسم الفئة
- ParentCategoryID: الفئة الأب (للتصنيفات الهرمية)

جدول Suppliers (الموردين)

معلومات الموردين:

- SupplierID: معرف المورد
- SupplierName: اسم المورد
- ContactPerson: جهة الاتصال
- Email: البريد الإلكتروني
- PhoneNumber: رقم الهاتف
- Address: العنوان
- PaymentTerms: شروط الدفع

(المستخدمون) Users جدول

معلومات المستخدمين:

- UserID: معرف المستخدم
- Username: اسم المستخدم
- Email: البريد الإلكتروني
- Password: كلمة المرور (مشفرة)
- Role: الدور (ADMIN, MANAGER, WAREHOUSE_STAFF, VIEWER)
- IsActive: حالة النشاط

(سجلات التدقيق) AuditLogs جدول

تسجيل جميع العمليات:

- LogID: معرف السجل
- UserID: معرف المستخدم
- Action: نوع الإجراء
- EntityType: نوع الكيان
- OldValue: القيمة القديمة
- NewValue: القيمة الجديدة
- Timestamp: وقت الإجراء

(التنبيهات) Alerts جدول

إدارة التنبيهات:

- AlertID: معرف التنبيه
- AlertType: نوع التنبيه (إلخ LOW_STOCK, EXPIRATION_WARNING)
- ProductID: معرف المنتج
- Severity: مستوى الخطورة
- Message: رسالة التنبيه
- IsRead: هل تم قراءة التنبيه

3.2 الحقول الإضافية

جداول إضافية للوظائف المتقدمة:

- OrderItems - تفاصيل طلبات الشراء
- GoodsReceiptItems - تفاصيل استقبال البضائع
- SalesOrders - طلبات المبيعات
- SalesOrderItems - تفاصيل طلبات المبيعات

- Stocktakes - الجرد الدوري
- StocktakeItems - تفاصيل الجرد

الفصل 4: العلاقات بين الكيانات

4.1 (1:N) الصلات الرئيسية

الشرح	نوع	إلى	من
كل فئة لها عدة منتجات	1:N	Products	Categories
كل مورد يوفر منتجات متعددة	1:N	Products	Suppliers
كل منتج في مستودعات متعددة	1:N	Stock	Products
كل مستودع يحتوي على منتجات متعددة	1:N	Stock	Warehouses
كل جرد له عدة حركات	1:N	StockMovements	Stock
كل مستخدم يمكنه إجراء حركات متعددة	1:N	StockMovements	Users
كل مورد له طلبات متعددة	1:N	PurchaseOrders	Suppliers
كل طلب يحتوي على عدة أصناف	1:N	OrderItems	PurchaseOrders
كل طلب قد يكون له عدة استقبالات	1:N	GoodsReceipts	PurchaseOrders
كل مستخدم يمكنه إنشاء طلبات متعددة	1:N	PurchaseOrders	Users
كل مستخدم له عدة سجلات	1:N	AuditLogs	Users
كل منتج قد يكون له عدة تنبيهات	1:N	Alerts	Products

الفصل 5: خطوات التثبيت والإعداد

5.1 الخطوة الأولى: إنشاء قاعدة البيانات

```
-- ثم قم بتنفيذ MySQL افتح --
mysql -u root -p < inventory_database_schema.sql
```

5.2 الخطوة الثانية: إعداد الملفات

```
# انسخ ملفات المشروع
cp -r inventory-management-system /var/www/html/

# دخول المجلد
cd /var/www/html/inventory-management-system

# تثبيت المكتبات
composer install
```

الخطوة الثالثة: إعداد الإعدادات 5.3

config/config.php: عدّل ملف

```
define('DB_HOST', 'localhost');
define('DB_USER', 'root');
define('DB_PASS', 'كلمة_المُرور_الخاصة_بك');
define('DB_NAME', 'inventory_management');
```

الخطوة الرابعة: اختبار التثبيت 5.4

:افتح المتصفح وانتقل إلى

<http://localhost/inventory-management-system>

الفصل 6: شرح هيكل الملفات

```
inventory-management-system/
├── config/
│   ├── config.php           # الإعدادات الرئيسية
│   └── database.php         # إعدادات قاعدة البيانات
├── models/
│   ├── ProductModel.php     # نموذج المنتجات
│   ├── InventoryModel.php   # نموذج المخزون
│   ├── OrderModel.php       # نموذج الطلبات
│   ├── SupplierModel.php    # نموذج الموردين
│   ├── UserModel.php        # نموذج المستخدمين
│   └── ReportModel.php      # نموذج التقارير
├── controllers/
│   ├── ProductController.php # تحكم المنتجات
│   ├── InventoryController.php # تحكم المخزون
│   ├── OrderController.php   # تحكم الطلبات
│   └── ...
├── views/
│   ├── layouts/             # القوالب الأساسية
│   ├── dashboard/           # عروض لوحة التحكم
│   ├── products/            # عروض المنتجات
│   ├── inventory/           # عروض المخزون
│   └── ...
├── public/
│   ├── css/                 # ملفات CSS
│   ├── js/                  # ملفات JavaScript
│   └── index.php             # نقطة الدخول الرئيسية
├── helpers/
│   ├── Database.php         # فئة قاعدة البيانات
│   ├── Validation.php       # التحقق من البيانات
│   └── ...
└── database/
    ├── schema.sql           # مخطط قاعدة البيانات
    └── seed_data.sql        # البيانات الأولية
```

الفصل 7: مثال عملي - إضافة منتج

7.1 ProductModel.php الكود في

```
public function addProduct($data) {
    $stmt = $this->db->prepare("
        INSERT INTO Products
        (ProductName, Barcode, SKU, Description, CategoryID,
        SupplierID, UnitPrice, CostPrice, ReorderLevel,
        ReorderQuantity, ExpirationDate, ManufacturingDate,
        WarrantyMonths, IsActive)
        VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)
    ");

    $stmt->bind_param(
        'ssssiiddiissi',
        $data['ProductName'],
        $data['Barcode'],
        $data['SKU'],
        $data['Description'],
        $data['CategoryID'],
        $data['SupplierID'],
        $data['UnitPrice'],
        $data['CostPrice'],
        $data['ReorderLevel'],
        $data['ReorderQuantity'],
        $data['ExpirationDate'],
        $data['ManufacturingDate'],
        $data['WarrantyMonths'],
        $data['IsActive']
    );

    if ($stmt->execute()) {
        return $this->db->getLastInsertId();
    }

    return false;
}
```

7.2 ProductController.php الكود في

```
public function create() {
    if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
        return ['success' => false, 'message' => 'طريقة الطلب غير صحيحة'];
    }

    // التحقق من البيانات المدخلة
    $required = ['ProductName', 'Barcode', 'SKU', 'CategoryID',
        'SupplierID', 'UnitPrice', 'CostPrice'];

    foreach ($required as $field) {
        if (!isset($_POST[$field]) || empty($_POST[$field])) {
```



```

        return ['success' => false,
                'message' => "مطلوب الحقل $field"];
    }
}

$data = [
    'ProductName' => $_POST['ProductName'],
    'Barcode' => $_POST['Barcode'],
    'SKU' => $_POST['SKU'],
    'Description' => $_POST['Description'] ?? '',
    'CategoryID' => intval($_POST['CategoryID']),
    'SupplierID' => intval($_POST['SupplierID']),
    'UnitPrice' => floatval($_POST['UnitPrice']),
    'CostPrice' => floatval($_POST['CostPrice']),
    'ReorderLevel' => intval($_POST['ReorderLevel'] ?? 10),
    'ReorderQuantity' => intval($_POST['ReorderQuantity'] ?? 50),
    'ExpirationDate' => $_POST['ExpirationDate'] ?? null,
    'ManufacturingDate' => $_POST['ManufacturingDate'] ?? null,
    'WarrantyMonths' => intval($_POST['WarrantyMonths'] ?? 0),
    'IsActive' => 1
];

$productId = $this->model->addProduct($data);

if ($productId) {
    return [
        'success' => true,
        'message' => 'تم إضافة المنتج بنجاح',
        'data' => ['ProductID' => $productId]
    ];
}

return ['success' => false, 'message' => 'فشل إضافة المنتج'];
}

```

الفصل 8: معايير وممارسات التطوير

8.1 معايير التسمية

الجدول والأعمدة:

- أسماء الجداول: بصيغة المفرد (Product, Category)
- أسماء الأعمدة: CamelCase (ProductName, SupplierID)
- المفاتيح الأساسية: [TableName]ID
- المفاتيح الأجنبية: [ReferencedTable]ID

الدوال والمتغيرات:

- الدوال والمتغيرات: camelCase
- الثوابت: UPPER_CASE
- الفئات: PascalCase

8.2 الأمان

```
// 1. استخدام Prepared Statements
$stmt = $db->prepare("SELECT * FROM Products WHERE ProductID = ?");
$stmt->bind_param("i", $productId);

// 2. تشفير كلمات المرور
$hashedPassword = password_hash($password, PASSWORD_DEFAULT);

// 3. التحقق من الجلسة
if (!isset($_SESSION['user_id'])) {
    header('Location: /login');
    exit;
}

// 4. تسجيل العمليات الحساسة
logAudit($userId, 'DELETE_PRODUCT', 'Products', $productId);
```

8.3 الأداء

- استخدام الفهارس على الحقول الشائعة
- Pagination للـ LIMIT و OFFSET استخدام
- N+1 تجنب استعلامات

الفصل 9: التقارير والتحليلات

9.1 التقارير المتاحة

1. تقرير المخزون: عرض الكميات والقيم
2. تقرير حركة المخزون: تحليل الإضافة والسحب
3. تقرير الصلاحية: المنتجات القريبة من انتهاء الصلاحية
4. تقرير المخزون المنخفض: المنتجات تحت الحد الأدنى
5. تقرير الموردين: أداء وعدد الطلبات
6. تقرير المبيعات: حجم وقيمة المبيعات

الفصل 10: الخطوات التالية والتطوير

10.1 المرحلة الأولى (الأسابيع 1-2)

- إعداد قاعدة البيانات ✓
- إنشاء البنية الأساسية ✓
- إنشاء نظام المصادقة

المرحلة الثانية (الأسبوع 3) 10.2

- إدارة المنتجات (CRUD)
- نظام الباركود
- إدارة الفئات والموردين

المرحلة الثالثة (الأسبوع 4) 10.3

- إدارة المخزون
- حركات المخزون
- الجرد الدوري

المرحلة الرابعة (الأسبوع 5) 10.4

- طلبات الشراء
- استقبال البضائع
- الطلبات والمبيعات

المرحلة الخامسة (الأسابيع 6-8) 10.5

- التقارير والتحليلات
- التنبيهات
- الاختبارات والتحسينات

المراجع والموارد

- [MySQL Documentation](#)
- [PHP Official Documentation](#)
- [MVC Pattern Guide](#)

الإصدار: 1.0

تاريخ التحديث: 13-11-2025

المؤلف: فريق تطوير النظام

✱