

# التوثيق الشامل - (CRM) نظام إدارة علاقات العملاء

---

## 1. نظرة عامة على النظام

### أ. الوصف العام

هو نظام متكامل لإدارة العلاقات مع العملاء وتتبع الطلبات والمعاينات والمخزون. (CRM) نظام إدارة علاقات العملاء يهدف النظام إلى تسهيل عمليات الشركة وتحسين خدمة العملاء وزيادة الكفاءة التشغيلية.

### ب. المكونات الرئيسية

#### 1. إدارة العملاء:

- تسجيل وإدارة بيانات العملاء
- تصنيف العملاء وتتبع تاريخ التعامل
- إدارة ملاحظات العملاء

#### 2. إدارة الطلبات:

- إنشاء وتتبع الطلبات
- إدارة عناصر الطلب
- تتبع حالة الطلب ومراحل التنفيذ

#### 3. إدارة المعاينات:

- جدولة وتنفيذ المعاينات
- تسجيل نتائج المعاينات
- ربط المعاينات بالطلبات

#### 4. إدارة المخزون:

- تسجيل وتتبع المنتجات
- إدارة المخزون والمستودعات
- تنبيهات انخفاض المخزون

#### 5. التقارير والإحصائيات:

- تقارير المبيعات والإيرادات
- تقارير أداء الموظفين
- تقارير نشاط العملاء

#### 6. إدارة المستخدمين والصلاحيات:

- إدارة حسابات المستخدمين
- تحديد الصلاحيات والأدوار
- تتبع نشاط المستخدمين

### ج. التقنيات المستخدمة

### 1. الإطار العام:

- Django 4.2 (Python)
- Bootstrap 5 (واجهة المستخدم)
- PostgreSQL (قاعدة البيانات)

### 2. واجهة برمجة التطبيقات (API):

- Django REST Framework
- JWT للمصادقة

### 3. الاتصال في الوقت الفعلي:

- Django Channels
- WebSockets

### 4. الأمان:

- تشفير البيانات الحساسة
- XSS و CSRF حماية من هجمات
- تسجيل النشاطات والتغييرات

## 2. التحسينات المنفذة

### أ. تبسيط النظام

#### 1. إزالة نظام التسعير الديناميكي:

- تم إزالة نظام التسعير الديناميكي المعقد واستبداله بنظام أبسط
- تم تبسيط حساب أسعار الطلبات
- تم إزالة الجداول والنماذج غير الضرورية

#### 2. إزالة Celery:

- لجدولة المهام Celery تم إزالة نظام
- تم تبسيط إدارة المهام الخلفية
- تم تقليل التبعيات الخارجية

#### 3. تنظيف الكود:

- تم إزالة الكود غير المستخدم
- تم تحسين التعليقات والتوثيق
- تم توحيد أسلوب كتابة الكود

### ب. تحسين الأداء

#### 1. تحسين استعلامات قاعدة البيانات:

- لتقليل عدد الاستعلامات `prefetch_related` و `select_related` تم استخدام
- تم تحسين استعلامات البحث والتصفية
- تم استخدام الفهارس بشكل أفضل

## 2. تحسين حساب الإحصائيات:

- تم استخدام استعلامات تجميعية أكثر كفاءة
- تم تقليل عدد الاستعلامات المتكررة
- تم تحسين أداء لوحات التحكم

## 3. تحسين تحميل الصفحات:

- تم تحسين تحميل البيانات في صفحات العملاء والطلبات
- تم تنفيذ التحميل الكسول للصور والموارد الثقيلة
- تم تحسين استخدام ذاكرة التخزين المؤقت

ج. تحسين واجهة المستخدم

## 1. تبسيط النماذج:

- تم تبسيط نماذج إنشاء وتعديل الطلبات
- تم تحسين التحقق من الصحة في جانب العميل
- تم إضافة تعليقات توضيحية أكثر

## 2. تحسين تجربة المستخدم:

- تم تحسين رسائل الخطأ والنجاح
- تم تحسين التنقل بين الصفحات
- تم تحسين عرض البيانات في الجداول

# 3. الهيكل التقني للنظام

أ. هيكل قاعدة البيانات

## 1. جداول العملاء:

- **Customer**: بيانات العملاء الأساسية
- **CustomerCategory**: تصنيفات العملاء
- **CustomerNote**: ملاحظات العملاء

## 2. جداول الطلبات:

- **Order**: بيانات الطلبات الأساسية
- **OrderItem**: عناصر الطلب
- **Payment**: مدفوعات الطلب
- **ShippingDetails**: تفاصيل الشحن

## 3. جداول المعاينات:

- **Inspection**: بيانات المعاينات
- **InspectionResult**: نتائج المعاينات

## 4. جداول المخزون:

- **Product**: المنتجات

- **Category**: تصنيفات المنتجات
- **Inventory**: المخزون
- **StockMovement**: حركة المخزون

#### 5. جداول المستخدمين:

- **User**: المستخدمين
- **Branch**: الفروع
- **Department**: الأقسام
- **Notification**: الإشعارات

### ب. هيكل التطبيقات

#### 1. تطبيق العملاء (**customers**):

- إدارة بيانات العملاء
- تصنيف العملاء
- إدارة ملاحظات العملاء

#### 2. تطبيق الطلبات (**orders**):

- إدارة الطلبات
- إدارة عناصر الطلب
- إدارة المدفوعات

#### 3. تطبيق المعاينات (**inspections**):

- إدارة المعاينات
- جدولة المعاينات
- تسجيل نتائج المعاينات

#### 4. تطبيق المخزون (**inventory**):

- إدارة المنتجات
- إدارة المخزون
- تتبع حركة المخزون

#### 5. تطبيق الحسابات (**accounts**):

- إدارة المستخدمين
- إدارة الصلاحيات
- إدارة الفروع والأقسام

#### 6. تطبيق التقارير (**reports**):

- تقارير المبيعات
- تقارير العملاء
- تقارير المخزون

### 4. المقترحات المستقبلية

## أ. تحسينات تقنية

### 1. تحسين البنية التحتية:

- تنفيذ هيكل خدمات أكثر مرونة
- تحسين فصل المسؤوليات
- تنفيذ نمط الوحدات القابلة للتوسيع

### 2. تحسين إدارة البيانات:

- تنفيذ استراتيجيات أرشفة البيانات
- تحسين التعامل مع البيانات الكبيرة
- تنفيذ تقسيم البيانات

### 3. تحسين المراقبة والتشخيص:

- إضافة تسجيل أفضل للأحداث
- تنفيذ مراقبة الأداء في الوقت الفعلي
- إضافة تنبيهات للمشكلات المحتملة

## ب. تحسينات وظيفية

### 1. تحسين نظام التقارير:

- إضافة تقارير تحليلية متقدمة
- إضافة رسوم بيانية تفاعلية
- إضافة إمكانية تصدير التقارير بتنسيقات مختلفة

### 2. تحسين إدارة العملاء:

- إضافة نظام ولاء العملاء
- تحسين تحليل سلوك العملاء
- إضافة إمكانية التواصل التلقائي مع العملاء

### 3. تحسين إدارة المخزون:

- إضافة نظام التنبؤ بالطلب
- تحسين إدارة المشتريات
- إضافة إمكانية تتبع المنتجات

## ج. خطة التنفيذ المقترحة

### 1. (المرحلة الأولى (1-2 أشهر):

- تحسين أداء النظام الحالي
- إكمال تنظيف الكود
- تحسين التوثيق

### 2. (المرحلة الثانية (2-3 أشهر):

- تحسين نظام التقارير

- تحسين واجهة المستخدم
- إضافة ميزات جديدة لإدارة العملاء

### 3. (المرحلة الثالثة (3-4 أشهر):

- تحسين إدارة المخزون
- تنفيذ نظام التنبؤ بالطلب
- تحسين إدارة المشتريات

## 5. دليل التشغيل السريع.

### أ. متطلبات النظام

#### 1. متطلبات الخادم:

- Python 3.8+
- PostgreSQL 13+
- Redis 7.0+

#### 2. متطلبات التطوير:

- Git
- أو أي محرر نصوص آخر Visual Studio Code
- Python 3.8+

### ب. تثبيت وإعداد النظام

#### 1. تثبيت التبعيات:

```
pip install -r requirements.txt
```

#### 2. إعداد قاعدة البيانات:

```
python manage.py migrate
```

#### 3. إنشاء مستخدم مدير:

```
python manage.py createsuperuser
```

#### 4. تشغيل الخادم:

```
python manage.py runserver
```

## ج. الصيانة الدورية

### 1. النسخ الاحتياطي:

```
python manage.py dbbackup
```

### 2. استعادة النسخ الاحتياطي:

```
python manage.py dbrestore
```

### 3. تنظيف قاعدة البيانات:

```
python manage.py clearsessions
```

## 6. الخاتمة

بشكل كبير من خلال تبسيط البنية وتحسين الأداء وتحسين واجهة (CRM) تم تحسين نظام إدارة علاقات العملاء وتم تحسين استعلامات Celery، تم إزالة الأنظمة المعقدة غير الضرورية مثل نظام التسعير الديناميكي و قاعدة البيانات لتحسين الأداء.

المقترحات المستقبلية تركز على تحسين البنية التحتية وإدارة البيانات وإضافة ميزات جديدة لتحسين تجربة المستخدم وزيادة كفاءة النظام.

يمكن تنفيذ هذه التحسينات على مراحل لضمان استمرارية عمل النظام وتقليل المخاطر.