

قالب موحد مقترن للتذكرة (Ticket Template)

لضمان أن الموديل (GPT-5.1 Codex Max High) يفهم المهام وينفذها بوضوح وبدون أخطاء، من المهم صياغة التذكرة بشكل منهجي وواضح. فيما يلي قالب موحد مقترن لكتابه التذكرة/المهمة التي سترسل إلى النموذج عبر Cursor. ينبغي اتباع هذا القالب قدر الإمكان لكل مهمة، مع تعديل المحتوى حسب الحاجة طبعاً:

- **عنوان المهمة (Task Title):** وصف مختصر ودقيق للمهمة المطلوبة.
- مثال: إضافة خاصية تصدير البيانات بصيغة CSV لقارير البيعات
- **السياق والخلفية (Context):** فقرة قصيرة تشرح خلفية المهمة ولماذا تحتاجها. تضمين أي معلومات من التذكرة الأصلية أو وضع النظام الحالي.
- مثال: حالياً نظام التقارير يدعم عرض البيانات على الشاشة فقط. تحتاج إمكانية تصدير التقرير إلى ملف CSV ليتمكن المستخدم من تنزيل البيانات. هناك موديل ReportData يمثل التقرير حالياً.
- **المتطلبات التفصيلية (Requirements):** قائمة نقطية بالمتطلبات المحددة التي يجب على الحل تحقيقها. كن صريحاً واضحاً قدر الإمكان.
 - وصف ما الذي يجب أن يفعله النموذج بالضبط (وظيفة).
 - ذكر أي قيود أو شروط (مثلاً: "استخدام مكتبة X المتوفرة"، أو "عدم تعديل واجهة المستخدم الحالية").
 - تضمين متطلبات غير وظيفية مهمة (مثل الأداء أو الأمان) إذا وجدت.
- **اعتبارات التصميم والمعمارية (Design & Architecture Considerations):** نقاط توضح أي إرشادات خاصة بالهيكلية يجب مراعاتها أثناء تنفيذ المهمة، لضمان الحل ضمن معمارية Clean Architecture الخاصة بالمشروع.
 - أشر بوضوح إلى الطبقات أو الوحدات التي ينبغي أن يتم التغيير فيها. مثال: يجب إضافة منطق التصدير في الطبقة Domain/Application وعدم ارتباطه مباشرة بطبقة العرض.
 - إذا كان هناك أجزاء موجودة يمكن إعادة استخدامها بدل كتابة كود جديد، اذكر ذلك (يوجد بالفعل خدمة CsvExporterHelper يمكن استخدامها لإعداد الملف).
 - أي أمور يجب تجنبها (تجنب تكرار أي كود متوفّر في ReportService، واعتمد مبدأ عدم تكرار الذات DRY).
- **خطة التنفيذ المقترنة (Implementation Plan):** اختياري ولكن مفيد، خاصة في المهام المعقدة. يمكن أن تكون قائمة خطوات عالية المستوى لكيفية الشروع في حل المشكلة. هذا يساعد النموذج على تنظيم الإجابة.
 - الخطوة الأولى: الملفات/الوحدات التي يجب تعديليها أو إنشاؤها. مثال: إنشاء فئة جديدة باسم CsvReportExporter في الطبقة Infrastructure تتولى إنشاء ملف CSV من كائن ReportData.
 - الخطوة الثانية: التعديلات اللازمة لربط هذه الفئة بالنظام. مثال: تعديل ReportService لاستدعاء CsvReportExporter وتوفير البيانات المطلوبة.
 - الخطوات التالية حسب الحاجة ... (تعديل واجهة المستخدم لإضافة زر "تصدير CSV" يستدعي المنطق الجديد عبر ViewModel).
- **معايير القبول (Acceptance Criteria):** قائمة بنود توضح كيف تتحقق من أن المهمة أُنجزت بنجاح وبشكل صحيح. هذه المعايير بمثابة اختبارات يجب أن يمر بها الحل.
 - مثال: ✓ عند الضغط على "تصدير CSV" في شاشة التقارير، يتم تنزيل ملف CSV يحتوي على البيانات الظاهرة حالياً بدقة.
 - مثال: ✓ ملف CSV يتضمن ترويسات الأعمدة الصحيحة وبالتالي المطلوب (UTF-8 encoding).
 - مثال: ✓ في حال حدوث خطأ أثناء التصدير (مثلاً فشل في الكتابة على القرص)، يتم عرض رسالة خطأ مناسبة للمستخدم دون تعطل التطبيق.
 - مثال: ✓ يعطي الاختبار الآلي ReportServiceTests سيناريو تصدير CSV ويتحقق من إنشاء الملف بالبيانات الصحيحة.

كيفية استخدام هذا القالب:

عند فتح تذكرة جديدة للعمل عليها مع نموذج GPT في Cursor، قم بملء الأقسام أعلاه بمعلومات المهمة. تأكد من أن العنوان موجز وواضح، المتطلبات محددة وقابلة للفهم (يمكن ترقيمها)، واعتبارات التصميم تذكرة التذكرة بمعايير المشروع (كأنك تذكرة زميل مطور بال نقاط المهمة قبل أن يبدأ العمل). أيضاً صياغة معايير القبول بشكل ✓ قائمة تحقق يساعد النموذج على تبيين الحل ليمرر هذه الاختبارات، بل وربما يقوم بكتابة اختبارات وحدات إذا طلبنا ذلك ضمناً. باستخدام هذا القالب بشكل موحد لكل مهمة، سنحقق عدة فوائد:

- **وضوح الاتصالات:** النموذج سيحصل دائمًا على الصورة الكاملة: ماذا نريد، لماذا، وكيف نقيس النجاح. هذا يقلل التخمين أو التفسير الخاطئ من جانبه.

- منع الأخطاء منذ البداية: ذكر المحاذير (مثل أماكن التعديل وعدم التكرار) فإننا نوجه النموذج لتقادي الأخطاء الشائعة (كأن يعرّف دالة موجودة مسبقاً أو يخرب الفصل المنطقى بين الطبقات).
- احترافية وثبات في المخرجات: عندما يتبع Cursor (الموديل + المطور) نفس النمط لكل تذكره، تصبح المخرجات أكثر اتساقاً ويسهل على الفريق قرائتها ومراجعةتها. سند أن كل حل مقدم يحتوي على شرح وتنظيم مشابه مما يجعل المراجعة التقنية (Code Review) أكثر سلاسة.

وأخيراً، لا بد من التأكيد أن دورنا لا ينتهي هنا. رغم قوة النموذج المختار وال قالب الموضوع، مراجعة الكود ومراقبة مخرجات النموذج أمر أساسى. لكن مع مرور الوقت، إذا التزمنا بهذه المنهجية، سنحصل على تعاون فعال جداً بين الفريق البشري والنماذج الذكى يؤدي إلى تسريع العمل مع الحفاظ على جودة عالية في مشروع DW_workspace. المصادر: