Plantilla de Documentación de Prompts

Campo	Contenido
ID del Prompt	P004-Fase 4
Fecha de Uso	25/03/2025
Modelo de IA	ChatGPT o3 mini-high
Categoría de la Tarea	Frontend/Backend/Autenticación
Tecnología	HTML/CSS/JavaScript/PHP/SQL
Texto Completo del Prompt	V.1 Prompt: proporciona una implementación detallada para la Fase 4 de la Tienda Virtual del CMS del Mercado Local San Mateo. En esta fase, se deben abordar los siguientes aspectos: Carrito de Compra: Definir la estructura de datos necesaria en base a la base de datos proporcionada. Implementar un controlador que gestione la adición, eliminación y actualización de productos en el carrito. Utilizar sesiones para almacenar temporalmente el carrito antes de confirmar la compra. Proporcionar una vista para mostrar el carrito con opciones de modificación. Proceso de Checkout: Implementar un flujo de compra que incluya validación de datos de envío y método de pago. Garantizar la verificación de stock antes de la confirmación del pedido. Manejar múltiples vendedores si aplica, asegurando la correcta asociación de productos a cada pedido. Proponer una estructura para gestionar cambios en el estado de los pedidos. Integración de Pasarela de Pago: Determinar opciones viables de pasarela de pago considerando el mercado español. Proporcionar un controlador para gestionar pagos y callbacks de confirmación. Garantizar la seguridad en la transmisión de datos sensibles. Proponer una estrategia de gestión de errores en pagos y posibles reembolsos. Consideraciones adicionales: Cumplimiento de normativas españolas de comercio electrónico (IVA, derecho de desistimiento, protección de datos). Notificaciones para clientes y vendedores sobre cambios en pedidos. Implementación de un historial de pedidos accesible para los clientes. Recomendaciones para la optimización de la experiencia de usuario y mejoras futuras. Documentos adjuntos: Documentación de la especificación y DB del proyecto 1º Iteracion: Dame el codigo para: Normativas Españolas: Asegúrate de incluir IVA en los cálculos y mostrar la información requerida en facturas. Implementa políticas de desistimiento y protección de datos en las vistas y controladores. Notificaciones: Utiliza eventos y listeners (o notificaciones por correo/SMS) para avisar a clientes y vendedores de cambios en el estado de los pedidos. Historial de Pedidos: Crear una vista y controladores necesarios para que el cliente pueda ver sus pedidos anteriores. El vendedor pueda ver los pedidos hechos al proveedor y pueda editar los que tiene con los clientes. El proveedor pueda ver los pedidos y editar los que tiene con los vendedores. Crear una vista y controladores necesarios para apartados de productos y puestos -Se mostraran los productos y puestos según el rol del usuario(vendedor ve los productos de los proveedores y vendedores, los clientes y usuarios no registrados solo de vendedores, los proveedores solo pueden ver productos de otros proveedores y vendedores) -Los clientes y usuarios invitados solo podrán añadir al carrito los productos de vendedores, los vendedores solo añadir al carrito los productos de proveedores y los proveedores no podrán añadir ningún producto al carrito(no tienen carrito) o comprarlo 2º Iteracion: Dame los enlaces para ponerlos en app.blade.php 3º Iteracion: tengo este error: SQLSTATE[42S22]: Column not found: 1054 Unknown column 'orders.deleted_at' in 'where clause' 4º Iteracion: Dame la codigo para order.show.blade.php 5º Iteracion: dame la ruta para la vista 6º Iteracion: Dame el codigo de OrderControlle 7º Iteracion: no me deja entrar al enlace a pesar de estar registrado: Route::get('/order/{order}', [OrderController::class, 'show']) ->name('order.show') ->middleware('auth'); 8º Iteracion: Dame la vista, ruta y función del controlador para order.edit

	9º Iteracion: Modifica el código de la tabla para que haya dos tablas: una con los pedidos encargados por el usuario (orders->buyer_id) y otro con los pedidos encargados al usuario (seller_id) si tiene el rol de vendedor o proveedor 10º Iteracion: Define más categorías para los productos de vendedores y proveedores: <select name="category" class="form-control"> <option value="">Todas las categorías</option> <option value="Frutas" {{ request('category') == 'Frutas' ? 'selected' : '' }}>Frutas</option> <option value="Artesanía" {{ request('category') == 'Artesanía' ? 'selected' : '' }}>Artesanía</option> <!-- Agrega más categorías según sea necesario --> </select> 11º Iteracion: Conviertelo en un select con las categorias que definiste: <label for="category">Categoría</label> <input type="text" name="category" class="form-control" value="{{ old('category', $product->category) }}"> 12º Iteracion: Añade filtrado y categorización a la vista: @section('title', 'Productos') @section('content') <h2 class="mb-4">Productos Disponibles</h2> @foreach($products as $product) @if($product->image) <img src="{{ asset('storage/' . $product->image) }}" class="card-img-top" alt="{{ $product->name }}"> @endif <h5 class="card-title">{{ $product->name }}</h5> {{ Str::limit($product->description, 80) }} Precio: ${{ number_format($product->price, 2) }} @if((!$user || $user->role === 'cliente') && $product->user->role === 'vendedor') <!-- Clientes e invitados pueden añadir productos de vendedores --> <form action="{{ route('cart.add', $product->id) }}" method="POST"> @csrf <button type="submit" class="btn btn-primary btn-block">Añadir al carrito</button> </form> @elseif($user && $user->role === 'vendedor' && $product->user->role === 'proveedor') <!-- Vendedores pueden añadir productos de proveedores --> <form action="{{ route('cart.add', $product->id) }}" method="POST"> @csrf <button type="submit" class="btn btn-primary btn-block">Añadir al carrito</button> </form> @elseif($user && $user->role === 'proveedor') <!-- Los proveedores no pueden comprar --> No disponible para compra @endif @endforeach {{ $products->appends(request()->query())->links() }} @endsection 13º Iteracion: El filtrado por categoria funciona pero el resto de funciones no 14º Iteracion: modifica la vista para que el usuario pueda modificar la foto de perfil:@section('content') <!-- Card para el perfil del cliente --> <h3 class="mb-0">Perfil del Cliente</h3> <form action="{{ route('client.profile.update') }}" method="POST"> @csrf <label for="name">Nombre</label> <input type="text" name="name" value="{{ old('name', $user->name) }}" class="form-control" placeholder="Ingresa tu nombre" required> <label for="phone">Teléfono</label> <input type="text" name="phone" value="{{ old('phone', $user->phone) }}" class="form-control" placeholder="Ingresa tu número de teléfono"> <!-- Puedes agregar más campos aquí según tus requerimientos --> <button type="submit" class="btn btn-primary mt-4">Actualizar Perfil</button> </form> @endsection 15º Iteracion: ¿Si quiero poner una pasarela de pago como puedo hacerlo? 16º Iteracion: no tengo Stripe.js, dame el codigo y donde ponerlo 17º Iteracion: añade a la vista el pago con stripe: @extends('layouts.app') @section('title', 'Checkout') @section('content') <h2 class="mb-4">Proceso de Checkout</h2> <form action="{{ route('checkout.process') }}" method="POST"> @csrf <label for="shipping_address">Dirección de Envío</label> <textarea name="shipping_address" class="form-control" rows="3" placeholder="Ingresa tu dirección de envío" required>{{ old('shipping_address') }}</textarea> <label for="payment_method">Método de Pago</label> <select name="payment_method" class="form-control" required> <option value="">Selecciona un método de pago</option> <option value="tarjeta">Tarjeta de Crédito/Débito</option> <option value="transferencia">Transferencia Bancaria</option>

	<!-- Se pueden agregar más opciones --> </select> <h4 class="mt-4">Resumen del Carrito</h4> <table class="table table-bordered mt-2"> <thead class="thead-light"> <tr> <th>Producto</th> <th>Cantidad</th> <th>Precio Unitario</th> <th>Subtotal</th> </tr> </thead> <tbody> @php $total = 0; @endphp @foreach($cart as $item) @php $subtotal = $item['price'] * $item['quantity']; $total += $subtotal; @endphp <tr> <td>{{ $item['name'] }}</td> <td>{{ $item['quantity'] }}</td> <td>${{ number_format($item['price'], 2) }}</td> <td>${{ number_format($subtotal, 2) }}</td> </tr> @endforeach <tr> <td colspan="3" class="text-right">Total:</td> <td><strong>${{ number_format($total, 2) }}</td> </tr> </tbody> </table> <button type="submit" class="btn btn-primary mt-3">Confirmar Compra</button> </form> @endsection
	V.1.1 **Instrucción:** Desarrolla una implementación completa y modular de la **Fase 4 del CMS para la Tienda Virtual del Mercado Local San Mateo**. Esta fase debe estar completamente integrada con las fases anteriores y contemplar tanto funcionalidades de usuario como cumplimiento legal y experiencia de uso. Asegúrate de incluir código funcional en Laravel (controladores, modelos, vistas, rutas y migraciones si es necesario) y aplicar buenas prácticas de desarrollo. 📌 **Requisitos obligatorios:** **1. Carrito de Compra** ● Definir estructura de datos basada en sesiones y la base de datos proporcionada. ● Crear un controlador que permita agregar, actualizar y eliminar productos del carrito. ● Desarrollar una vista Blade para visualizar y modificar el contenido del carrito. ● Asegurar que solo usuarios con roles permitidos (cliente, vendedor) puedan agregar productos, según estas reglas: ○ **Clientes e invitados:** solo pueden agregar productos de *vendedores*. ○ **Vendedores:** solo pueden agregar productos de *proveedores*. ○ **Proveedores:** no pueden agregar productos al carrito. **2. Proceso de Checkout** ● Validar datos de envío y método de pago. ● Verificar el stock de productos antes de la confirmación. ● Manejar múltiples vendedores si aplica (crear pedidos agrupados por vendedor). ● Almacenar los pedidos con IVA incluido (21%). ● Implementar estructura para los estados del pedido (pendiente, procesando, enviado, entregado, cancelado). **3. Integración de Pasarela de Pago** ● Implementar integración con una pasarela real (Stripe, Redsys o PayPal). ● Simular callbacks de confirmación de pago. ● Asegurar transmisión segura de datos sensibles. ● Gestionar errores de pago y soporte para reembolsos. **4. Cumplimiento Legal (España)** ● Calcular y mostrar el **IVA (21%)** en vistas de factura. ● Incluir aceptación de políticas de **protección de datos** y **derecho de desistimiento** en formularios de registro y compra. ● Validar dicha aceptación en el backend. **5. Notificaciones Automáticas** ● Implementar eventos y listeners para notificar a clientes y vendedores sobre cambios en el estado del pedido (vía correo electrónico). ● Utilizar el sistema de notificaciones de Laravel.

	6. Historial de Pedidos ● Crear vistas y controladores para visualizar pedidos anteriores. ● Mostrar dos secciones: ○ Pedidos realizados por el usuario (como comprador). ○ Pedidos recibidos por el usuario (como vendedor o proveedor). ● Permitir edición de pedidos únicamente a vendedores/proveedores en función del rol. **7. Apartado de Productos y Puestos** ● Mostrar productos y puestos filtrados por rol: ○ **Clientes e invitados:** solo productos de vendedores. ○ **Vendedores:** productos de proveedores y vendedores. ○ **Proveedores:** productos de proveedores y vendedores (solo visualización). ● Incluir filtrado por categoría y buscador textual. **8. Opcional: Optimización UX** ● Aplicar AJAX en acciones del carrito (agregar, actualizar, eliminar). ● Incluir alertas sobre bajo stock y confirmaciones visuales de acciones del usuario.	
Long. del Prompt	V.1 9027 caracteres	V.1.1 2982 caracteres
Documentos incluidos	Sí, Documentación de la especificación y DB del proyecto	
Ejemplos Incluidos	Sí, código que requería de modificaciones	
Calidad de la Respuesta	V.1 5	V.1.1 5
P. Ponderada	V.1 5	V.1.1 5
Prob. de Código	Implementacion de codigo no previamente generado o especificado, Problemas de compatibilidad y manejo de dependencias Javascript	
Iteraciones	17	
Mejoras Recomendadas	Mejor estructuración del prompt, aplicar las especificaciones necesarias para evitar futuras iteraciones.	
Resultados de aprendizaje vinculados	V1 R1, R3, R4: El estudiante comprende y aplica estructuras de software web (carrito, checkout, pasarela). R5, R11, R12, R14, R15: Trabajo autónomo, planificación y aplicación práctica. R6, R7, R10: Presentación escrita de soluciones y documentación de código.	
	V1.1 R1, R2, R3, R4, R5, R6, R12, R14, R15	
Competencias relacionadas	V1 CETM-1 / CETM-6 / CETM-7: Desarrollo e integración de servicios y sistemas web complejos. CG-5: Programación y uso de herramientas en telecomunicaciones/web. CB-2 / CB-4 / CT-1 / CT-2: Aplicación profesional, comunicación y trabajo colaborativo.	
	V1.1 CETM-1, CETM-2, CETM-6, CG-5, CT-1, CT-2, CB-2, CB-4	
Objetivos de la asignatura	V1 OBJ-1, OBJ-2, OBJ-3: Uso de tecnologías web, desarrollo de sistema completo y herramientas de e-commerce.	
	V1.1 OBJ-1, OBJ-2, OBJ-3	

Tabla de Comparación de Versiones de Prompt

Versión	Camb. Clave	Cla.	Esp.	Prec. Téc.	Ctx.	Estruc.	Cal. Cód.	Punt. Pond.	Mej.
1.0	Inicial	5	5	5	5	5	5	5	-
1.1	Se agregaron especificaciones técnicas	5	5	5	5	5	5	5	0

Leyenda de Abreviaturas

- **Camb. Clave** = Cambios Clave
- **Cla.** = Claridad
- **Esp.** = Especificidad

- **Prec. Téc.** = Precisión Técnica
- **Ctx.** = Contexto
- **Estruc.** = Estructura
- **Cal. Cód.** = Calidad del Código
- **Punt. Pond.** = Puntuación Ponderada
- **Mej.** = Mejora