

Plantilla de Documentación de Prompts

Campo	Contenido
ID del Prompt	P002-Fase 2
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	<p>V.1 Necesito una implementación completa de autenticación y gestión de perfiles en Laravel X.X para un CMS de Mercado Local. El sistema debe permitir el registro e inicio de sesión de usuarios con diferentes roles (cliente, vendedor y proveedor) y la gestión de perfiles según el rol. También se debe incluir middleware de autorización para restringir accesos.</p> <p>Requisitos detallados:</p> <p>Autenticación</p> <p>Implementar el sistema de autenticación con Laravel UI.</p> <p>Implementación de validaciones seguras en el RegisterController y LoginController.</p> <p>Middleware de Autorización</p> <p>Crear un middleware CheckRole para restringir acceso según el rol.</p> <p>Registrar el middleware en Kernel.php.</p> <p>Definir rutas protegidas para cada tipo de usuario en routes/web.php.</p> <p>Gestión de Perfiles por Rol</p> <p>Crear controladores ClientController, VendorController, ProviderController con sus respectivas funcionalidades:</p> <p>Cliente: ver y actualizar perfil.</p> <p>Vendedor: ver y actualizar perfil, gestionar su puesto (stand), gestionar productos y pedidos.</p> <p>Proveedor: ver y actualizar perfil, gestionar su puesto (stand), gestionar productos y pedidos.</p> <p>Implementar vistas en Blade para la edición de perfil de cada usuario, siguiendo un diseño basado en Bootstrap.</p> <p>Flujo de autenticación y redirecciones</p> <p>Modificar LoginController para redirigir según el rol.</p> <p>Implementar logout y protección contra ataques CSRF.</p> <p>Corrección de errores comunes</p> <p>Garantizar que las rutas están correctamente definidas (App\Http\Controllers...).</p> <p>Incluir comandos de Laravel (php artisan migrate, db:seed) para inicializar correctamente el sistema.</p> <p>Configuraciones y datos de prueba para facilitar la ejecución</p> <p>Incluir ejemplos de usuarios en UsersTableSeeder.</p> <p>Explicar cómo probar la funcionalidad (rutas relevantes, credenciales de prueba).</p> <p>Incluir el layout base (app.blade.php) con navegación adecuada según el rol del usuario.</p> <p>Documentos adjuntos: Se incluye una especificación del sistema y estructura de base de datos</p> <p>1º Iteración: Dame las vistas y las rutas necesarias para el home y el login</p> <p>Respuesta: A continuación se muestra un ejemplo completo de las rutas y vistas necesarias para el home y el login en un proyecto Laravel que utiliza Laravel UI.</p> <p>2º Iteración: Dame todas las rutas públicas (home, login, register, etc.) se generan con Laravel UI</p> <p>Respuesta: A continuación se muestra un ejemplo completo de las rutas públicas que se generan al instalar Laravel UI con el scaffold de autenticación. Estas rutas incluyen el home, login, registro y las rutas de recuperación de contraseña.</p>

3º Iteracion: Como registro el seeder UserSeeder en DatabaseSeeder.php
Resultado: Para registrar el seeder en **DatabaseSeeder.php**, abre el archivo ubicado en **database/seeder/DatabaseSeeder.php** y dentro del método **run()** llama al seeder que deseas ejecutar, por ejemplo:

```
<?php
```

```
namespace Database\Seeders;
```

```
use Illuminate\Database\Seeder;
```

```
class DatabaseSeeder extends Seeder
```

```
{
```

```
    public function run()
```

```
    {
```

```
        // Registra el seeder de usuarios
```

```
        $this->call(UserSeeder::class);
```

```
    }
```

```
}
```

4º Iteracion: Dame un app.css para el CMS

5º Iteración: Mejora la vista: <!DOCTYPE html> <html lang="es"> <head>
<meta charset="UTF-8"> <meta name="csrf-token" content="{{ csrf_token() }}">
<title>CMS Mercado Local</title> <link href="{{ asset('css/app.css') }}"
rel="stylesheet"> </head> <body> <nav class="navbar navbar-expand-lg
navbar-light bg-light"> Mercado
Local <div class="collapse navbar-collapse"> <ul class="navbar-nav
ml-auto"> @guest <li class="nav-item"><a class="nav-link" href="{{
route('login') }}">Login <li class="nav-item"><a class="nav-link"
href="{{ route('register') }}">Registro @else <li class="nav-item">
Hola, {{ Auth::user()->name }}
@if(Auth::user()->role == 'cliente') <li class="nav-item"><a class="nav-link"
href="{{ route('client.profile') }}">Mi Perfil @elseif(Auth::user()->role ==
'vendedor') <li class="nav-item"><a class="nav-link" href="{{
route('vendor.profile') }}">Mi Perfil @elseif(Auth::user()->role ==
'proveedor') <li class="nav-item"><a class="nav-link" href="{{
route('provider.profile') }}">Mi Perfil @endif <li class="nav-item"> <a
class="nav-link" href="{{ route('logout') }}"
onclick="event.preventDefault();document.getElementById('logout-form').submit
();"> Logout <form id="logout-form" action="{{ route('logout') }}"
method="POST" style="display: none;"> @csrf </form> @endguest
</div> </nav> <div class="container mt-4"> @if(session('success')) <div
class="alert alert-success">{{ session('success') }}</div> @endif
@yield('content') </div> <script src="{{ asset('js/app.js') }}"></script> </body>
</html>

6º Iteracion: Si el css lo tengo en resources/css como puedo cargarlo en
app.blade.php?

V.1.1 Objetivo:

Desarrollar un sistema completo de autenticación y gestión de perfiles para un CMS de un *Mercado Local* utilizando **Laravel versión X.X**. El sistema debe gestionar distintos roles de usuario (**cliente, vendedor y proveedor**), protegiendo el acceso con middleware de autorización y permitiendo funcionalidades específicas según el rol.



Requisitos Técnicos Detallados



Autenticación y Seguridad

- Usar **Laravel UI** con Bootstrap para generar vistas de autenticación.
- Personalizar **RegisterController** y **LoginController** con validaciones seguras:
 - Email único, contraseña mínima de 8 caracteres, validación del campo **role** con **Rule::in([...])**.
- Implementar redirección tras login según el rol del usuario.
- Activar protección CSRF en todos los formularios.



Middleware de Autorización

- Crear middleware **CheckRole** para restringir el acceso por rol (**cliente, vendedor, proveedor**).
- Registrar el middleware en **app/Http/Kernel.php**.



Rutas

- Definir rutas públicas: home (/), login, registro, recuperación de contraseña (Laravel UI).
- Definir rutas protegidas para cada rol en **routes/web.php**, incluyendo:
 - **/cliente/perfil, /vendedor/perfil, /proveedor/perfil**.
 - CRUD de productos, pedidos y puesto para vendedor y proveedor.



Gestión de Perfiles

- Crear los controladores:
 - **ClientController**: ver y actualizar perfil.
 - **VendorController** y **ProviderController**: ver y actualizar perfil, gestionar productos, pedidos y puestos.
- Validar datos del perfil (**name, phone**, etc.).
- Utilizar **Auth::user()** para recuperar datos del usuario.






Vistas Blade

- Crear un layout base **resources/views/layouts/app.blade.php** con navegación dinámica según rol.
- Crear vistas de perfil con formularios en Bootstrap: **client.profile, vendor.profile, provider.profile**.



Seeders y Configuración Inicial

	<ul style="list-style-type: none"> • Crear seeder <code>UsersTableSeeder</code> con 3 usuarios de prueba (uno por rol) y contraseñas conocidas. <p>Registrar seeder en <code>DatabaseSeeder.php</code> y ejecutar:</p> <pre>bash CopiarEditar php artisan migrate --seed</pre> <p> Flujo de Pruebas</p> <ul style="list-style-type: none"> • Iniciar servidor (<code>php artisan serve</code>). • Probar login con usuarios semilla y verificar: <ul style="list-style-type: none"> ◦ Redirección correcta por rol. ◦ Accesos autorizados y error 403 en accesos no autorizados. ◦ Visualización dinámica del menú. • Incluir instrucciones claras para probar rutas, credenciales y funcionalidades. <hr/> <p> Archivos Esperados</p> <ol style="list-style-type: none"> 1. <code>RegisterController.php</code>, <code>LoginController.php</code> (validaciones + redirección) 2. <code>CheckRole.php</code> middleware 3. <code>routes/web.php</code> con rutas públicas y protegidas 4. <code>ClientController.php</code>, <code>VendorController.php</code>, <code>ProviderController.php</code> 5. Vistas Blade: layout base y formularios de perfil 6. <code>UsersTableSeeder.php</code> y <code>DatabaseSeeder.php</code> 7. CSS base (<code>resources/css/app.css</code>) si se requiere personalización adicional <hr/> <p> Notas Adicionales</p> <ul style="list-style-type: none"> • Asegúrate de compilar CSS si está en <code>resources/css</code> usando Vite o Laravel Mix. • El diseño debe ser responsive, accesible y coherente con Bootstrap. 	
Long. del Prompt	V.1 1952	V.1.1 2860
Documentos incluidos	Sí, especificación del sistema y estructura de base de datos	
Ejemplos Incluidos	Sí, en las iteraciones siguientes se incluyen ejemplos para un mejor resultado	
Calidad de la Respuesta	V.1 5	V.1.1 5
P. Ponderada	V.1 5	V.1.1 5
Prob. de Código	No incluir las vistas y rutas al principio No incluir detalles de cómo implementar el código generado Implementacion de codigo ubicado en otros directorios dentro del proyecto	
Iteraciones	6	
Mejoras Recomendadas	Sugerencias específicas para mejorar el prompt	
Resultados de aprendizaje vinculados	V.1 R1, R2, R3, R4: Diseño e implementación de servicios web completos (cumplido). R6, R8, R10, R12, R14, R15: Redacción de documentación técnica, uso de recursos en inglés, ejecución autónoma (cumplido). R5, R11, R13: Trabajo colaborativo, gestión de tareas, planificación (implícito si se desarrolla en equipo).	

	<p>V1.1 R1 – R4: Diseño, implementación, validación y entrega de servicios web completos (cumplido al 100%).</p> <p>R5 – R15: El prompt fomenta la planificación, trabajo autónomo, búsqueda de documentación, documentación de código y validación funcional, todos cubiertos con precisión.</p>
Competencias relacionadas	<p>V1 CETM-1, CETM-4, CETM-6, CETM-7: Construcción y gestión de aplicaciones y servicios telemáticos.</p> <p>CG-5: Diseño y programación de aplicaciones.</p> <p>CB-2, CB-3, CB-4, CB-5: Aplicación de conocimientos, análisis de información, comunicación clara y aprendizaje autónomo.</p> <p>CT-1, CT-2: Comunicación efectiva y colaboración.</p> <p>V1.1 CETM-1, CETM-4, CETM-6, CETM-7: Implementación de sistemas y servicios telemáticos usando herramientas analíticas y programación.</p> <p>CG-5: Diseño y programación de aplicaciones.</p> <p>CB-2, CB-3, CB-4, CB-5: Aplicación práctica, reflexión crítica, comunicación técnica y autonomía en el aprendizaje.</p> <p>CT-1, CT-2, CT-3: Comunicación eficaz, cooperación profesional y mejora continua.</p>
Objetivos de la asignatura	<p>V1 OBJ-1: Tecnologías web cliente/servidor.</p> <p>OBJ-2: Desarrollo de un sistema web completo.</p> <p>OBJ-3: Herramientas para e-commerce.</p> <p>OBJ-4: Conceptos y herramientas de redes sociales (implícito en la gestión de roles y perfiles).</p> <p>V1.1 OBJ-1: Tecnologías web cliente y servidor.</p> <p>OBJ-2: Implementación de un sistema completo con base de datos.</p> <p>OBJ-3: Herramientas para el desarrollo de e-commerce (roles y productos).</p> <p>OBJ-4: Interacciones sociales en el sistema (gestión de perfiles, permisos).</p>

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	5
1.1	Se agregaron especificaciones técnicas	5	5	5	5	5	5	5	50

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