Para probar este test, se debe de verificar previamente los siguientes pasos:

- Tener instalado Node.js en el equipo.
- Tener instalado Yarn (de preferencia por que así se ejecuta el proyecto)
- Tener alguna instancia de mysql corriendo (ya que en ese se tiene que crear la base de datos para el backend).
- Crear una base de datos con el nombre que se deja de ejemplo en el archivo .env.example
- Copiar el archivo .env.example y pegarlo, después reemplazar esa copia por .env
- El proyecto corre a partir de Vite 4 usando el stack de React.js, en la configuración de las variables de entorno, configurar el APP_URL con el nombre del host que genere el frontend, si se usa un virtualhost como los que generan herramientas como Laragon o Laravel Valet definirlo en la variable correspondiente.
- Configurar los accesos a base de datos correctos en el archivo de variables.

En este proyecto, se utilizó un stack con las siguientes tecnologías:

- PHP
- Laravel
- Typescript (por mejor manejo de tipado estricto)
- React.js

Dentro de este Stack, en el tema del front (usando React.js) contemplaron los siguientes paquetes más importantes:

- Mantine (librería de componentes).
- React Hook Form (manejador de formularios).
- Redux Toolkit (manejo de estados, solo aplico en autenticación).
- RTK Query (manejador de peticiones interno de Redux Toolkit).

Por último, en el tema del back (usando laravel), se contemplaron el uso de algunos conceptos, por ejemplo:

- Uso de query scopes dentro de los modelos de Product y ProductTranslation.
- Uso de relaciones de eloquent.
- Validaciones de formularios en la $request.
- Abstracción personalizada del manejador de errores para mantener un estándar de respuestas en base a ciertos errores.

Nota Importante: Para el paso 10 del aplicativo, dejo los ejemplos de las rutas que se deben consumir con sus ejemplos y los parámetros que requiere :

Lista de productos: /api/products

Parámetro	Opciones Válidas
language	es, en
search	Cualquier texto (busca sobre el sku y nombre de producto en el idioma)
order_by_field	products.dollar_price, products.peso_price, name
order_by_field_direction	asc, desc

Detalle de producto: /api/products/{url}

Parámetro	Opciones Validas
url	Cualquier slug de producto en su idioma