

ACTUALIZACIÓN WEB MOTORLAN

Actualización de la programación de compra-venta de motores

- Creación de controladores
- Creación de rutas
- Maquetado de todas las vistas.
- Producción
- Revisiones

Detallado abajo.

Objetivos: Crear un mercado de ventas de motores sin pasarela de pago, se podrá publicar, listar, editar y contactar a las personas.,

Integrar Vuejs en WordPress para obtener datos con rest api, y tener vistas rápidas sin recarga.

1-) Página con todas los motores (paginadas) con filtros personalizados.

vista:

https://www.motorlan.es/es/compra-venta-de-motores-electricos-industriales/?cs-motor-0=&cs-tipo_o_referencia-1=&cs-potencia-3=&cs-velocidad-4=&cs-par_nominal-5=&cs-voltaje-6=&cs-estado_del_articulo-9=&paged=1&posts_per_page=15&orderby=recientes&order=DESC

— Funcionalidades:

- Filtrado por características del motor..
- Mostrar motores filtrados con posibilidad de añadir a favoritos, ver ficha (otra vista/plantilla) y seleccionar
- Poder visualizar promociones, descuentos, garantías.

Necesitamos:

1.1 -) Controlador Entidad que muestre:

- Los motores con sistema de paginación personalizado.
- Filtrado múltiple, con filtros excluyentes.
- Manejo de agregar o quitar entidad favorita de acuerdo al usuario.

1.2 -) Base de datos, tablas para guardar:

- Filtrado por usuario.
- Motores favoritas por los usuarios.

1.3 -) Rutas api rest para .

- Paginación.

- Filtrado.
- Redirección para contactar.
- Añadido/eliminado favorito.

1.4-) Creación de vistas frontend. Vistas con ajax, para evitar recarga (Uso de VUEJS framework)

- Paginación .
- Manejo de filtrado.
- Métodos para motores favoritos
- Array selección.
- Redireccionamiento login.
- Maquetado de acuerdo a diseño.
- Métodos auxiliares.
- Vistas de notificaciones (popup de avisos).

1.5 -) Revisión y cambios.

Horas 15 horas.

2-) Nuevo usuario
vista :

— Funcionalidades:

- Permitir registro de nuevo usuario (crear rol personalizado) .
- Validación de datos del lado del servidor y del cliente.
- Verificación de email para registro exitoso.

Necesitamos:

2.1 -) Controlador Usuario con métodos para:

- Registro de usuario con rol personalizado..
- Validación de datos del lado del servidor y manejo de errores y estados.
- Guardado de datos.
- Envío de email para verificación y bienvenida a la web.

2.2 -) Base de datos, solo añadir campos:

- Guardado de metadatos .

2.3 -) Rutas api rest para .

- 1 ruta, para envío de datos.

2.4-) Creación de vistas frontend. Vistas con ajax, para evitar recarga (Uso de VUEJS framework)

- Formulario de registro .

- Validación de datos del lado del cliente.
- Muestra de errores..
- Envío de datos y manejo de respuesta del servidor.
- link a login.
- Maquetado de acuerdo a diseño.
- Métodos auxiliares.
- Vistas de notificaciones (popup de avisos).

2.5 -) Revisión y cambios.

Horas 10 horas.

3-) Logueo usuario

vista :

— Funcionalidades:

- Permitir logueo de usuario.
- Validación de datos del lado del servidor y del cliente.

Necesitamos:

3.1 -) Controlador Usuario con métodos para:

- Registro de usuario con rol personalizado.
- Validación de datos del lado del servidor y manejo de errores y estados.

3.2 -) Rutas api rest para .

- 1 ruta, para envío de datos.

3.3-) Creación de vistas frontend. Vistas con ajax, para evitar recarga (Uso de VUEJS framework)

- Formulario de logueo .
- Validación de datos del lado del cliente.
- Muestra de errores..
- Envío de datos y manejo de respuesta del servidor.
- Link a recuperar contraseña .
- Maquetado de acuerdo a diseño.
- Métodos auxiliares.
- Vistas de notificaciones (popup de avisos).

3.4 -) Revisión y cambios.

Horas 10 horas.

4-) Recuperar contraseña

vista : Sin vista

— Funcionalidades:

- Permitir recuperar la contraseña del usuario .
- Validación de datos del lado del servidor y del cliente.
- Envío de código con fecha de vencimiento al email del usuario para verificar que es propietario de la cuenta.

Necesitamos:

4.1 -) Controlador Usuario con métodos para:

- Verificar existencia del email.
- Validación de datos del lado del servidor y manejo de errores y estados.
- Envío de email con código para verificar la autenticidad de la cuenta.
- Cambio de contraseña .

4.2 -) Base de datos, solo añadir campos:

- Guardado de metadatos .

4.3 -) Rutas api rest para .

- 1 ruta, para envío de email .
- 1 ruta para manejar código de verificación y nueva contraseña

4.4-) Creación de vistas frontend. Vistas con ajax, para evitar recarga (Uso de VUEJS framework) . Serían 2 vistas, esto en un proceso de 2 pasos.

- Formulario.
- Validación de datos del lado del cliente.
- Muestra de errores..
- Envío de datos y manejo de respuesta del servidor.
- link a login.
- Métodos auxiliares.
- Vistas de notificaciones (popup de avisos).

4.5 -) Revisión y cambios.

Horas 5 horas.

5-) Mi cuenta / página de inicio.

vista:

— Funcionalidades:

- Mostrar las últimas 5 búsquedas.
-
- Mostrar los últimos motores como favoritos.
- Sidebar con enlaces a otras secciones

Necesitamos:

5.1 -) Controlador Usuario, con siguientes métodos:

- Retorno de datos del usuario.
- Retorno búsquedas guardadas.
- Retorno últimos motores .
- Retorno de favoritos.

5.2 -) Base de datos, nada que hacer solo consultas.

5.3 -) Rutas api rest para .

- Retorno de datos del controlador.

5.4-) Creación de vistas frontend. Vistas con ajax, para evitar recarga (Uso de VUEJS framework)

- Métodos para obtener datos con axios .
- Rutas internas para el sidebar.
- Maquetado de acuerdo a diseño.
- Métodos auxiliares.
- Vistas de notificaciones (popup de avisos).

5.5 -) Revisión y cambios.

Horas 10 horas.

6-) Mi cuenta / página de perfil.

vista:

— Funcionalidades:

- Mostrar formulario de perfil.
- Guardar datos.
- Validación de datos del lado del cliente y del servidor.

Necesitamos:

6.1 -) Controlador Usuario, con siguientes métodos:

- Validación de datos del lado del servidor.
- Guardado de datos.

6.2 -) Base de datos. Nada, solo guardar datos

6.3 -) Rutas api rest para .

- Envío de datos al servidor y retorno de la respuesta.

6.4-) Creación de vistas frontend. Vistas con ajax, para evitar recarga (Uso de VUEJS framework)

- Métodos para obtener/enviar datos con axios .

- Validación de datos del lado del cliente.
- Maquetado de acuerdo a diseño.
- Métodos auxiliares.
- Vistas de notificaciones (popup de avisos).

6.5 -) Revisión y cambios.

Horas 10 horas.

7-) Mi cuenta / mis motores

— Funcionalidades:

- Mostrar motores creados paginados.
- Permitir borrar motores .
- Redireccionar a edición de motores.

Necesitamos:

7.1 -) Controlador motores, con siguientes métodos:

- Permitir consultar motores creados.
- Manejar paginación..
- Permitir borrar motores

7.2 -) Base de datos. Nada, solo guardar datos y consultas

7.3 -) Rutas api rest para .

- Solicitar borrar motores.
- Solicitar todos los motores con paginación .
- Manejar paginación .

7.4-) Creación de vistas frontend. Vistas con ajax, para evitar recarga (Uso de VUEJS framework)

- Métodos para obtener/enviar datos con axios .
- Paginación
- Ver motor
- Maquetado de acuerdo a diseño.
- Métodos auxiliares.
- Link a editar motor
- Vistas de notificaciones (popup de avisos).

7.5 -) Revisión y cambios.

Horas 10 horas.

8-) Mi cuenta / Editar motor

vista:

— Funcionalidades:

- Editar nombre y título del motor .
- Permitir otras configuraciones .

Necesitamos:

8.1 -) Controlador motores, con siguientes métodos:

- Configurar nombre , título y opciones de motor.
- Guardar/actualizar datos.
- Retornar motores seleccionadas de acuerdo a filtro.

8.2 -) Base de datos.

- Creación de tablas y de campos necesarios, guardar datos y consultas

8.3 -) Rutas api rest para .

- Retornar y enviar datos de configuración.
- Modificación de orden de las motores en el motores .
- Retornar motores seleccionadas.

8.4-) Creación de vistas frontend. Vistas con ajax, para evitar recarga (Uso de VUEJS framework)

- Métodos para obtener/enviar datos con axios .
- Maquetación de pdf en html junto con todas sus variaciones.
- Maquetado de acuerdo a diseño.
- Métodos auxiliares.
- Link a edita rmotores
- Vistas de notificaciones (popup de avisos).

8.5 -) Revisión y cambios.

Horas 15 horas.

9-) Mi cuenta / favoritos

vista:

— Funcionalidades:

- Visualizar favoritos .
- Eliminar favoritos .
- Ver ficha del motor .
- Permitir contactar a alguien para comprar a través de favoritos

Necesitamos:

9.1 -) Controlador motor y entidad, con siguientes métodos:

- Retornar motores favoritas, permitir de-seleccionarlas como favoritas. .
- Retornar a las motores seleccionadas favoritas.

9.2 -) Base de datos.

- Guardar datos y consultas

9.3 -) Rutas api rest para .

- Retornar y enviar datos de configuración.
- Retornar a las motores favoritos .
- Eliminar motores de favoritos

9.4-) Creación de vistas frontend. Vistas con ajax, para evitar recarga (Uso de VUEJS framework)

- Métodos para obtener/enviar datos con axios ..
- Maquetado de acuerdo a diseño.
- Métodos auxiliares.
- Link a editar motor
- Vistas de notificaciones (popup de avisos).

9.5 -) Revisión y cambios.

Horas 10 horas.

10-) Motor / Ficha motor

vista:

— Funcionalidades:

- Visualizar datos del motor .
- Mostrar motores relacionados y del mismo vendedor.

Necesitamos:

10.1 -) Controlador motores con siguientes métodos:

- Retornar entidad y datos.
- Retornar motores relacionados
- Mostrar si el motor tiene garantías

10.2 -) Base de datos.

- consultas

10.3 -) Rutas api rest para .

- Retornar y enviar datos del motor.

10.4-) Creación de vistas frontend. Vistas con ajax, para evitar recarga (Uso de VUEJS framework)

- Métodos para obtener/enviar datos con axios ..
- Maquetado de acuerdo a diseño.
- Métodos auxiliares.
- Vistas de notificaciones (popup de avisos).

10.5 -) Revisión y cambios.

Horas 10 horas.

11-) Mensajería/contacto

Se implementará un sistema de comunicación entre vendedor y cliente. Para que puedan gestionar la compra a través de la web. Se considerará notificaciones en la web además de notificaciones vía email.

Horas: 7 horas

12-) Instalación del Wordpress, configuración, traslado y envío a producción.

Añadimos también la exportación de los motores antiguos. Se crearía un script para trasladar los motores ya agregados a la nueva implementación.

Horas 23 horas.

TOTAL: 135 horas