



INVENTARIO CON VUE.JS

Documentación

Miguel Brown
browncoloma@gmail.com

índice

Introducción	3
Estructura de Carpetas.....	4
Archivos Principales	5
Funcionalidades Clave	6
Conclusión	8
Bibliografía.....	9

Introducción

En el mundo moderno de las aplicaciones web, la gestión eficiente de inventarios es fundamental para el éxito de cualquier negocio. Es aquí donde la aplicación "Inventario con Vue.js" desempeña un papel crucial, brindando una solución robusta y fácil de usar para el control y administración de productos.

Esta documentación detallada tiene como objetivo guiar a los desarrolladores y profesionales de software a través de la estructura, funcionalidades y códigos clave de la aplicación "Inventario con Vue.js". Diseñada con un enfoque modular y organizado, esta aplicación aprovecha el poder de Vue.js, un popular framework de JavaScript, para ofrecer una experiencia de usuario fluida y reactiva.

A lo largo de esta documentación, se explorará en profundidad la estructura de carpetas y archivos de la aplicación, brindando una comprensión clara de cómo se organiza y distribuye el código fuente. Además, se analizarán detalladamente las funcionalidades clave, como listar, agregar, editar y eliminar productos, así como el cálculo del stock total disponible.

Cada sección de la documentación estará respaldada por fragmentos de código relevantes, lo que permitirá a los lectores comprender mejor la implementación y el flujo de la aplicación. Estos ejemplos de código estarán acompañados de explicaciones claras y concisas, facilitando la comprensión de los conceptos subyacentes.

Ya sea que se trate de un desarrollador experimentado o de un profesional de software en formación, esta documentación servirá como una guía invaluable para comprender a fondo la aplicación "Inventario con Vue.js". Al final, los lectores habrán adquirido un conocimiento sólido de cómo se construye y funciona esta aplicación, lo que les permitirá adaptarla, mantenerla y mejorarla según sus necesidades específicas.

Estructura de Carpetas

- Bbdd: Esta carpeta contiene los archivos relacionados con la base de datos.
 - conexión.php: Archivo encargado de establecer la conexión con la base de datos.
 - crud.php: Archivo que maneja las operaciones CRUD (Crear, Leer, Actualizar y Eliminar) en la base de datos.
- Bootstrap: Carpeta que alberga los archivos CSS y JavaScript de Bootstrap, un framework CSS utilizado para el diseño responsive y la aplicación de estilos.
- js: Carpeta que contiene los archivos JavaScript de Bootstrap.
- jquery: Carpeta que contiene la biblioteca jQuery, utilizada para simplificar la manipulación del DOM y el manejo de eventos.
- plugins: Carpeta que contiene plugins adicionales.
 - sweetalert: Carpeta que contiene los archivos del plugin SweetAlert, utilizado para mostrar cuadros de diálogo modales atractivos.
 - sweetalert2.all.min.js: Archivo JavaScript principal del plugin SweetAlert.
 - sweetalert2.min.css: Archivo CSS del plugin SweetAlert.

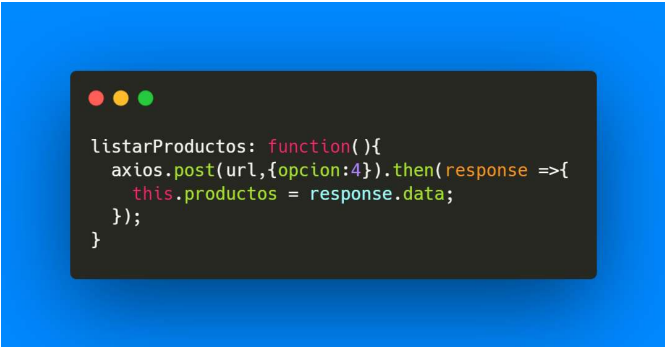
Archivos Principales

- `index.php`: Este archivo contiene la estructura HTML de la aplicación y enlaza los archivos CSS, JavaScript y bibliotecas necesarias.
- `main.css`: Archivo CSS que contiene los estilos personalizados de la aplicación.
- `main.js`: Archivo JavaScript principal que contiene la lógica de la aplicación Vue.js.

Funcionalidades Clave

1. Listar Productos

La aplicación utiliza Vue.js y Axios para obtener los datos de los productos desde el servidor mediante una petición POST. La función `listarProductos()` en `main.js` realiza esta tarea y almacena los datos en la variable `productos`.



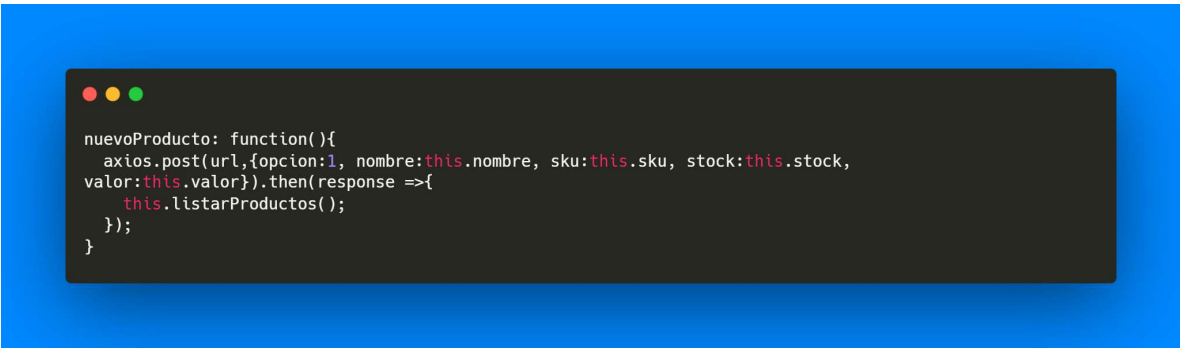
```
listarProductos: function(){
  axios.post(url,{opcion:4}).then(response =>{
    this.productos = response.data;
  });
}
```

2. Agregar Nuevo Producto

La función `btnNuevo()` en `main.js` maneja la lógica para agregar un nuevo producto. Utiliza `SweetAlert` para mostrar un cuadro de diálogo modal donde el usuario puede ingresar los detalles del nuevo producto. Luego, la función `nuevoProducto()` envía una petición POST al servidor para insertar el nuevo registro en la base de datos.

3. Editar Producto

La función `btnEditar()` en `main.js` se encarga de manejar la edición de un producto existente. Muestra un cuadro de diálogo modal con `SweetAlert` donde el usuario puede modificar los detalles del producto seleccionado. Luego, la función `editarProducto()` envía una petición POST al servidor para actualizar el registro correspondiente en la base de datos.



```
nuevoProducto: function(){
  axios.post(url,{opcion:1, nombre:this.nombre, sku:this.sku, stock:this.stock,
  valor:this.valor}).then(response =>{
    this.listarProductos();
  });
}
```

4. Eliminar Producto

La función `btnBorrar()` en `main.js` maneja la eliminación de un producto. Primero, muestra un cuadro de diálogo de confirmación con `SweetAlert`. Si el usuario confirma, la función `borrarProducto()` envía una petición POST al servidor para eliminar el registro correspondiente de la base de datos.

```
editarProducto:function(id, nombre, sku, stock, valor){
  axios.post(url,{opcion:2,id:id, nombre:nombre, sku:sku, stock:stock, valor:valor}).then(response =>
  {  this.listarProductos();
    });
}
```

5. Cálculo del Stock Total

La aplicación calcula el stock total de todos los productos mediante una propiedad computada `totalStock` en `main.js`. Esta propiedad recorre la lista de productos y suma los valores de stock individuales.

```
totalStock() {
  this.total = 0;
  for (let producto of this.productos) {
    this.total += parseInt(producto.stock);
  }
  return this.total;
}
```

En resumen, la aplicación "Inventario con Vue.js" sigue un enfoque modular y organizado, separando las responsabilidades en diferentes archivos y carpetas. Utiliza `Vue.js` como framework frontend, `Axios` para las peticiones HTTP, `SweetAlert` para los cuadros de diálogo modales y `Bootstrap` para el diseño responsive. La lógica principal se encuentra en `main.js`, donde se implementan las funciones CRUD y el cálculo del stock total. El archivo `crud.php` maneja las operaciones de base de datos, mientras que `conexión.php` se encarga de establecer la conexión con la base de datos.

Conclusión

La aplicación "Inventario con Vue.js" es un excelente ejemplo de cómo aprovechar el poder de las tecnologías web modernas para crear soluciones eficientes y escalables. A lo largo de esta documentación, se han explorado en detalle su estructura, funcionalidades y código fuente, brindando a los desarrolladores y profesionales de software una comprensión profunda de su funcionamiento.

Al utilizar Vue.js, un framework de JavaScript de vanguardia, la aplicación logra una experiencia de usuario fluida y reactiva, facilitando la gestión y administración de inventarios. Además, se complementa con otras herramientas y bibliotecas populares, como Bootstrap, jQuery, SweetAlert y Font Awesome, que enriquecen su diseño, interactividad y funcionalidad.

Bootstrap, un marco de trabajo CSS y JavaScript, proporciona a la aplicación un diseño responsive y atractivo, asegurando una experiencia consistente en diferentes dispositivos y tamaños de pantalla. jQuery, por su parte, simplifica la manipulación del DOM y el manejo de eventos, optimizando el rendimiento y la eficiencia del código. SweetAlert, un plugin de cuadros de diálogo modales, agrega un toque de elegancia y accesibilidad a las interacciones del usuario. Finalmente, Font Awesome, una biblioteca de iconos vectoriales escalables, enriquece la interfaz de usuario con iconos atractivos y fácilmente personalizables.

Al combinar todas estas tecnologías, la aplicación "Inventario con Vue.js" se convierte en una solución robusta y escalable para la gestión de inventarios, adaptable a las necesidades cambiantes de los negocios modernos.

Bibliografía:

1. "ChatGPT". Anthropic. Accedido el 8 de marzo de 2024. <https://www.anthropic.com/>
2. "jsDelivr - A free, fast, and reliable CDN for open source." jsDelivr. Accedido el 8 de marzo de 2024. <https://www.jsdelivr.com/>
3. "Bootstrap · The most popular HTML, CSS, and JS library in the world." Bootstrap. Accedido el 8 de marzo de 2024. <https://getbootstrap.com/>
4. "Font Awesome · The internet's icon library and toolset." Font Awesome. Accedido el 8 de marzo de 2024. <https://fontawesome.com/>

Esta documentación, junto con las referencias bibliográficas proporcionadas, pretende ser un recurso valioso para aquellos que deseen comprender, mantener y mejorar la aplicación "Inventario con Vue.js", aprovechando al máximo las capacidades de Vue.js y las tecnologías complementarias utilizadas.