**Trabajo Práctico Integrador**

**Programación III**

**Objetivo:**  
Diseñar y desarrollar, como trabajo práctico universitario, un sistema web de comercio electrónico para una empresa de comidas que permita gestionar ventas, gastos y finanzas, integrando medios de pago digitales y criptomonedas. El proyecto deberá incluir un chatbot para atender consultas dentro de la plataforma y una solución de inteligencia artificial para responder correos electrónicos. La implementación se realizará aplicando tecnologías de front-end como HTML, CSS, JavaScript y React, fomentando el uso de buenas prácticas de programación, trabajo en equipo y resolución de problemas reales del entorno empresarial.

**Requisitos del Proyecto:**

**Parte 1: Estructura y Diseño en HTML**

**1.1 Formularios de Registro**

* **Formulario de Ventas:**
  + **Campos:** Nombre del cliente, Producto(s) vendido(s), Cantidad, Método de pago (tarjeta, criptomonedas, efectivo, billeteras virtuales), Total de la venta (en moneda fiduciaria), Fecha de la transacción.
  + **Funcionalidad:** Este formulario permitirá registrar una venta y automáticamente generar un asiento contable en el sistema.
* **Formulario de Gastos:**
  + **Campos:** Descripción del gasto, Monto (en moneda fiduciaria o criptomonedas), Método de pago, Fecha del gasto.
  + **Funcionalidad:** El registro de un gasto también generará un asiento contable, reflejando el impacto en la contabilidad de la empresa.

**1.2 Grillas para Visualización de Datos**

* **Grilla de Ventas:**
  + Mostrar todas las ventas realizadas, incluyendo los detalles del cliente, productos vendidos, método de pago, y montos en moneda fiduciaria y criptomonedas.
* **Grilla de Gastos:**
  + Visualizar todos los gastos operativos registrados con detalles como fecha, descripción, monto, y método de pago.

**Parte 2: Estilos en CSS**

**2.1 Diseño Responsivo con Media Queries**

* **Media Queries:** Asegurar que el sistema sea completamente responsivo, adaptándose a diferentes tamaños de pantalla (móviles, tablets, desktops).
* **Estilos para Formularios y Grillas:** Diseñar los formularios y grillas para que sean fáciles de usar y legibles en cualquier dispositivo.

**2.2 Layout con Flexbox y Grid**

* **Flexbox:** Organizar el layout general de la página utilizando Flexbox, incluyendo encabezado, cuerpo principal, y pie de página.
* **CSS Grid:** Utilizar CSS Grid para estructurar los elementos dentro de las grillas y formularios, asegurando un diseño limpio y funcional.

**Parte 3: Programación en JavaScript**

**3.1 Gestión de Ventas, y Gastos**

* **Arrays:** Usar arrays para almacenar los datos de ventas y gastos.
* **Local Storage:** Implementar el almacenamiento de estos arrays en localStorage, para mantener la persistencia de los datos entre sesiones del usuario.

**3.2 Integración con Criptomonedas**

* **Conversión de Criptomonedas:** Implementar una función en JavaScript que consulte una API de criptomonedas usando fetch para obtener el valor actual de una criptomoneda en moneda fiduciaria. Usar esta información para convertir los pagos en cripto a su equivalente en dólares/euros.
* **Registro de Pagos:** Integrar la conversión de criptomonedas en el proceso de registro de ventas y gastos, para que las transacciones se almacenen correctamente en los arrays y localStorage.

**Parte 4: Desarrollo de Interfaz con React**

**4.1 Gestión de Estados con Hooks**

* **useState:** Utilizar useState para manejar el estado de los datos de ventas y gastos, dentro de los componentes de React.
* **useEffect:** Usar useEffect para realizar la consulta a la API de criptomonedas cada vez que se actualice el formulario de ventas o gastos, actualizando el estado con la tasa de cambio más reciente.

**4.2 Fetch para Datos en Tiempo Real**

* **Fetch API:** Implementar fetch para obtener datos en tiempo real de una API de criptomonedas. Utilizar los datos para actualizar los montos de ventas y gastos en la interfaz.

**4.3 Interactividad y Actualización Dinámica**

* **Renderización Condicional:** Aplicar renderización condicional para mostrar diferentes vistas según los datos ingresados, como una confirmación al registrar una venta o gasto.
* **Actualización en Tiempo Real:** Asegurar que la interfaz responda de manera fluida a las acciones del usuario, mostrando actualizaciones inmediatas al registrar nuevas transacciones.

**Parte 5: Documentación y Presentación**

**5.1 Documentación Técnica**

* **Manual de Usuario:** Explicar cómo utilizar el sistema para registrar ventas, gastos y cómo gestionar criptomonedas.
* **Documentación del Código:** Incluir comentarios detallados en el código HTML, CSS, JavaScript, y React, explicando la lógica detrás de cada sección.

**5.2 Presentación del Proyecto**

* **Demo del Sistema:** Mostrar cómo se registran ventas y gastos, y cómo se integran los pagos en criptomonedas.

**Entrega:**

El proyecto debe ser entregado en un repositorio (GitHub) que incluya todo el código fuente, la base de datos en localStorage, y la documentación. Se espera que los estudiantes presenten su sistema en clase, explicando cómo se cumplieron los requisitos del trabajo práctico.