**Warehouse Management System (WMS)**

**Warehouse Management System (WMS)**

**Introduction and Project Overview**

**The Warehouse Management System (WMS) is a comprehensive web application designed to streamline the management of inventory, sales, and customer data. Built with PHP for backend functionality, JavaScript for frontend interactivity, and PostgreSQL for database management, this project serves as a demonstration of modern web development practices. It highlights the implementation of CRUD (Create, Read, Update, Delete) operations, database integration, and user-friendly interface design. This system can be extended and customized to meet the needs of various businesses or organizations managing warehouses.**

**Key Features**

* **Inventory and sales management.**
* **Customer management with detailed records.**
* **Multi-language support for English and Russian.**
* **Dynamic filtering, searching, and sorting functionalities.**
* **Responsive design compatible with both desktop and mobile devices.**
* **Secure user authentication and session handling.**

**Target Audience**

**This project is intended for:**

1. **Developers looking to understand practical web application development with PHP and PostgreSQL.**
2. **Businesses in need of a foundational warehouse management system.**
3. **Students and educators exploring web technologies and database integration.**

**Brief Description**

The Warehouse Management System (WMS) is a demo project designed to demonstrate key functionalities for managing inventory, sales, and customer data. This project is developed using PHP for server-side operations, JavaScript for frontend interactivity, and PostgreSQL for backend database management. The main goal is to showcase CRUD (Create, Read, Update, Delete) operations and provide a solid foundation for warehouse management applications.

**Objectives**

1. Simplify the process of inventory and sales tracking.
2. Enable user-friendly customer management.
3. Demonstrate core programming principles in PHP, JavaScript, and PostgreSQL.
4. Showcase efficient database integration and web development techniques.

**Project Structure**

**File Structure Overview:**

* **php/**: Server-side scripts for database interactions (e.g., adding, retrieving, editing, deleting data).
* **css/**: Stylesheets for a responsive and visually appealing user interface.
* **js/**: Client-side scripts for interactivity (e.g., a photo slider, scroll-to-top button).
* **Database**: PostgreSQL with tables for managing products, suppliers, customers, and sales.

**Key Files and Their Functions**

* **db.php**: Connects to the PostgreSQL database using PDO. Configures connection parameters and handles exceptions.
* **styles.css**: Defines UI styles such as fonts, spacing, colors, and element formatting (e.g., buttons, tables).
* **script.js**: Manages dynamic UI elements like the scroll-to-top button and image slider. Tracks scroll events and slide transitions.
* **democonnect.php**: Executes queries using INNER JOIN, LEFT JOIN, and RIGHT JOIN to display related data from tables.
* **connecttables.php**: Similar to democonnect.php, handles different types of JOIN queries for admin users.
* **mass.php**: Displays and deletes records from the contactform table. Access restricted to authenticated sessions.
* **login.php**: Handles user authentication and redirects users to the appropriate dashboard or page.
* **index.php**: Main page with sections like "Features," "Advantages," "Services," and an image slider. Includes links to the login form and contact information.
* **demo.php**: Provides demo warehouse management functionalities such as filtering, searching, sorting, and aggregations.
* **dashboard.php**: Admin panel for managing database records with features like search, sort, add, edit, and delete.
* **contact.php**: Processes user messages and stores them in the contactform table.

**Special Features**

* Multi-language support: Allows users to switch between English and Russian.
* Dynamic filtering, searching, and sorting.
* Responsive design for optimal performance on desktop and mobile devices.

**Instructions for Launching and Testing**

**Prerequisites**

1. Install a local web server (e.g., XAMPP) with PHP and PostgreSQL support.
2. Ensure PostgreSQL is installed and properly configured.

**Setup**

1. Clone or download the project files.
2. Place the project folder in the web server’s root directory (e.g., htdocs for XAMPP).
3. Import the provided PostgreSQL database structure (e.g., database.sql) into your PostgreSQL server.
4. Update php/db.php with your PostgreSQL database credentials.

**Launching the Application**

1. Start the web server and PostgreSQL.
2. Open a web browser and navigate to http://localhost/sklad/php/index.php.

**Testing the Application**

**Login Functionality**

* Test login with valid credentials (e.g., admin / 1256).
* Verify error messages for invalid credentials.

**CRUD Operations**

* **Products**: Add, edit, and delete products; verify changes in the database and UI.
* **Customers**: Add, update, and remove customers; confirm updates in the database and UI.

**Interface and Navigation**

* Test all navigation links for correct page redirects.
* Check responsiveness on different devices (desktop and mobile).

**Database Integration**

* Verify all data updates in the PostgreSQL database.
* Test error handling during CRUD operations.

**Special Features**

* Test the photo slider for smooth transitions.
* Ensure the scroll-to-top button functions as expected.
* Switch between English and Russian and confirm accurate translations.

**Checklist for Testing Functionality**

1. **Login Functionality**
   * Test valid and invalid credentials.
   * Confirm redirection to the appropriate pages.
2. **CRUD Operations**
   * Products: Add, edit, delete.
   * Customers: Add, update, remove.
3. **Interface Navigation**
   * Verify links and buttons.
   * Check layout on different screen sizes.
4. **Database Integration**
   * Verify CRUD updates in the database.
   * Check for errors during data operations.
5. **Special Features**
   * Photo slider.
   * Scroll-to-top button.
   * Language translation.

**Additional Notes**

1. This project demonstrates basic and advanced SQL queries with JOIN operations.
2. The application implements secure session handling and error reporting.
3. Ensure php.ini settings support error logging and debugging during development.

**Database**

CREATE TABLE ContactForm (

name VARCHAR(255) NOT NULL,

company VARCHAR(255),

email VARCHAR(255) NOT NULL,

message TEXT NOT NULL

)

ALTER TABLE "contactform"

ALTER COLUMN "id" SET DEFAULT nextval('ContactForm\_id\_seq');

CREATE TABLE Suppliers (

SupplierID INT PRIMARY KEY,

SupplierName VARCHAR(100) NOT NULL,

ContactInfo VARCHAR(255),

Address VARCHAR(255)

);

CREATE TABLE Products (

ProductID INT PRIMARY KEY,

ProductName VARCHAR(100) NOT NULL,

Category VARCHAR(50),

Price DECIMAL(10, 2) NOT NULL CHECK (Price >= 0),

StockQuantity INT NOT NULL CHECK (StockQuantity >= 0)

);

CREATE TABLE Customers (

CustomerID INT PRIMARY KEY,

CustomerName VARCHAR(100) NOT NULL,

ContactInfo VARCHAR(255),

Address VARCHAR(255)

);

CREATE TABLE Sales (

SaleID INT PRIMARY KEY,

SaleDate DATE NOT NULL,

CustomerID INT,

TotalAmount DECIMAL(10, 2) NOT NULL CHECK (TotalAmount >= 0),

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);

CREATE TABLE SaleDetails (

SaleDetailID INT PRIMARY KEY,

SaleID INT,

ProductID INT,

Quantity INT NOT NULL CHECK (Quantity > 0),

Price DECIMAL(10, 2) NOT NULL CHECK (Price >= 0),

FOREIGN KEY (SaleID) REFERENCES Sales(SaleID),

FOREIGN KEY (ProductID) REFERENCES Products(ProductID)

);

CREATE TABLE Payments (

PaymentID INT PRIMARY KEY,

SaleID INT,

PaymentDate DATE NOT NULL,

Amount DECIMAL(10, 2) NOT NULL CHECK (Amount >= 0),

PaymentMethod VARCHAR(50),

FOREIGN KEY (SaleID) REFERENCES Sales(SaleID)

);

CREATE TABLE suppliers\_products (

suppliers\_supplierid INT,

products\_productid INT,

FOREIGN KEY (suppliers\_supplierid) REFERENCES Suppliers(SupplierID),

FOREIGN KEY (products\_productid) REFERENCES Products(ProductID)

);

INSERT INTO Suppliers (SupplierID, SupplierName, ContactInfo, Address) VALUES

(1, 'ТОО КазТоргСервис', '77 7012345678', 'Алматы, ул. Абая, 123'),

(2, 'ТОО АстанаСтрой', '77 7023456789', 'Нур-Султан, пр. Туран, 34'),

(3, 'ТОО Алматинская Техника', '77 7034567890', 'Алматы, ул. Наурызбай батыра, 45'),

(4, 'ТОО Мегаполис Логистик', '77 7045678901', 'Шымкент, ул. Байтурсынова, 23'),

(5, 'ТОО АвтоГрупп', '77 7056789012', 'Атырау, пр. Абая, 12'),

(6, 'ТОО ТехноПроект', '77 7067890123', 'Костанай, ул. Гагарина, 15'),

(7, 'ТОО АгроТехСнаб', '77 7078901234', 'Талдыкорган, ул. Сатпаева, 67'),

(8, 'ТОО СтройПартнер', '77 7089012345', 'Павлодар, ул. Гоголя, 8'),

(9, 'ТОО Альфа Трейд', '77 7090123456', 'Караганда, ул. Маметова, 5'),

(10, 'ТОО Бизнес Плюс', '77 7101234567', 'Актобе, пр. Кенесары, 11'),

(11, 'ТОО ВостокТех', '77 7112345678', 'Усть-Каменогорск, ул. Сейфуллина, 32'),

(12, 'ТОО Казахстан Строй', '77 7123456789', 'Тараз, ул. Бауыржана Момышулы, 29'),

(13, 'ТОО ПромТехСнаб', '77 7134567890', 'Кызылорда, ул. Курмангалиева, 50'),

(14, 'ТОО ГарантСервис', '77 7145678901', 'Жезказган, ул. Мира, 3'),

(15, 'ТОО ТрансЛогистик', '77 7156789012', 'Темиртау, ул. Кирова, 18'),

(16, 'ТОО СапаСервис', '77 7167890123', 'Актау, ул. Кенесары, 75'),

(17, 'ТОО СтройГрупп', '77 7178901234', 'Семей, ул. Бектурова, 22'),

(18, 'ТОО ПродуктСнаб', '77 7189012345', 'Костанай, ул. Курмангалиева, 10'),

(19, 'ТОО Евразия Логистик', '77 7190123456', 'Атырау, ул. Абая, 50'),

(20, 'ТОО КазахстанТорг', '77 7201234567', 'Алматы, ул. Жибек Жолы, 58');

INSERT INTO Products (ProductID, ProductName, Category, Price, StockQuantity) VALUES

(1, 'Болт М16', 'Инструменты', 1000.99, 50),

(2, 'Шайба круглая', 'Запчасти', 150.49, 60),

(3, 'Гайка М20', 'Инструменты', 200.79, 70),

(4, 'Дрель Makita', 'Электроника', 15000.99, 80),

(5, 'Пила дисковая', 'Инструменты', 25000.99, 90),

(6, 'Перфоратор Bosch', 'Электроника', 30000.99, 100),

(7, 'Гаечный ключ', 'Инструменты', 700.99, 110),

(8, 'Молоток', 'Инструменты', 1800.99, 120),

(9, 'Лебедка электрическая', 'Запчасти', 22000.99, 130),

(10, 'Отвертка электрическая', 'Инструменты', 1100.99, 140),

(11, 'Ремень приводной', 'Запчасти', 999.99, 150),

(12, 'Шлифмашина', 'Электроника', 13000.99, 160),

(13, 'Трос стальной', 'Запчасти', 1499.99, 170),

(14, 'Генератор электрический', 'Электроника', 17000.99, 180),

(15, 'Шиномонтажный станок', 'Оборудование', 150000.99, 190),

(16, 'Ключ торцевой', 'Инструменты', 1200.99, 200),

(17, 'Двигатель электродрели', 'Запчасти', 21900.99, 210),

(18, 'Аккумулятор для дрели', 'Запчасти', 23900.99, 220),

(19, 'Генератор бензиновый', 'Оборудование', 25000.99, 230),

(20, 'Комплект инструментов', 'Инструменты', 2300.99, 240);

INSERT INTO Customers (CustomerID, CustomerName, ContactInfo, Address) VALUES

(1, 'ИП Ибраев', '77 7012345678', 'Алматы, ул. Абая, 15'),

(2, 'ИП Сулейменов', '77 7023456789', 'Нур-Султан, ул. Туран, 6'),

(3, 'ИП Токтаров', '77 7034567890', 'Шымкент, ул. Байтурсынова, 45'),

(4, 'ИП Абдувалиев', '77 7045678901', 'Атырау, ул. Абая, 29'),

(5, 'ИП Жарқынов', '77 7056789012', 'Алматы, ул. Гоголя, 3'),

(6, 'ИП Кусаинов', '77 7067890123', 'Костанай, ул. Сейфуллина, 8'),

(7, 'ИП Жумалиев', '77 7078901234', 'Павлодар, ул. Курмангалиева, 2'),

(8, 'ИП Мауленов', '77 7089012345', 'Караганда, ул. Байтурсынова, 9'),

(9, 'ИП Муканов', '77 7090123456', 'Талдыкорган, ул. Махатова, 4'),

(10, 'ИП Дариев', '77 7101234567', 'Усть-Каменогорск, ул. Ленина, 13'),

(11, 'ИП Сагимбаев', '77 7112345678', 'Актобе, ул. Жибек Жолы, 7'),

(12, 'ИП Нургалиев', '77 7123456789', 'Жезказган, ул. Октябрьская, 25'),

(13, 'ИП Кенжебаев', '77 7134567890', 'Тараз, ул. Толе Би, 5'),

(14, 'ИП Шарипов', '77 7145678901', 'Кызылорда, ул. Гагарина, 4'),

(15, 'ИП Аманбаев', '77 7156789012', 'Семей, ул. Чкалова, 14'),

(16, 'ИП Тлеубаев', '77 7167890123', 'Темиртау, ул. Свердлова, 3'),

(17, 'ИП Умиров', '77 7178901234', 'Актау, ул. Абая, 5'),

(18, 'ИП Кенешов', '77 7189012345', 'Талдыкорган, ул. Пушкина, 12'),

(19, 'ИП Болатбаев', '77 7190123456', 'Павлодар, ул. Гоголя, 33'),

(20, 'ИП Муханов', '77 7201234567', 'Алматы, ул. Панфилова, 20');

INSERT INTO Sales (SaleID, SaleDate, CustomerID, TotalAmount) VALUES

(1, '2024-01-01', 1, 1000.99),

(2, '2024-01-02', 2, 1500.49),

(3, '2024-01-03', 3, 2000.79),

(4, '2024-01-04', 4, 750.49),

(5, '2024-01-05', 5, 1200.69),

(6, '2024-01-06', 6, 890.99),

(7, '2024-01-07', 7, 1100.89),

(8, '2024-01-08', 8, 1300.69),

(9, '2024-01-09', 9, 1600.99),

(10, '2024-01-10', 10, 1453.39),

(11, '2024-01-11', 11, 1750.79),

(12, '2024-01-12', 12, 2050.59),

(13, '2024-01-13', 13, 2300.49),

(14, '2024-01-14', 14, 1800.89),

(15, '2024-01-15', 15, 1400.99),

(16, '2024-01-16', 16, 1200.59),

(17, '2024-01-17', 17, 1350.49),

(18, '2024-01-18', 18, 1250.99),

(19, '2024-01-19', 19, 1500.39),

(20, '2024-01-20', 20, 1450.69);

INSERT INTO Payments (PaymentID, SaleID, PaymentDate, Amount, PaymentMethod) VALUES

(1, 1, '2024-01-02', 1000.99, 'Кредитная карта'),

(2, 2, '2024-01-03', 1500.49, 'Наличные'),

(3, 3, '2024-01-04', 2000.79, 'Кредитная карта'),

(4, 4, '2024-01-05', 750.49, 'Карта'),

(5, 5, '2024-01-06', 1200.69, 'Перевод'),

(6, 6, '2024-01-07', 890.99, 'Наличные'),

(7, 7, '2024-01-08', 1100.89, 'Кредитная карта'),

(8, 8, '2024-01-09', 1300.69, 'Перевод'),

(9, 9, '2024-01-10', 1600.99, 'Карта'),

(10, 10, '2024-01-11', 1453.39, 'Наличные'),

(11, 11, '2024-01-12', 1750.79, 'Кредитная карта'),

(12, 12, '2024-01-13', 2050.59, 'Наличные'),

(13, 13, '2024-01-14', 2300.49, 'Кредитная карта'),

(14, 14, '2024-01-15', 1800.89, 'Перевод'),

(15, 15, '2024-01-16', 1400.99, 'Карта'),

(16, 16, '2024-01-17', 1200.59, 'Наличные'),

(17, 17, '2024-01-18', 1350.49, 'Кредитная карта'),

(18, 18, '2024-01-19', 1250.99, 'Перевод'),

(19, 19, '2024-01-20', 1500.39, 'Карта'),

(20, 20, '2024-01-21', 1450.69, 'Наличные');

INSERT INTO SaleDetails (SaleDetailID, SaleID, ProductID, Quantity, Price) VALUES

(1, 1, 1, 2, 1000.99),

(2, 2, 2, 3, 150.49),

(3, 3, 3, 5, 200.79),

(4, 4, 4, 1, 15000.99),

(5, 5, 5, 1, 25000.99),

(6, 6, 6, 2, 30000.99),

(7, 7, 7, 3, 700.99),

(8, 8, 8, 4, 1800.99),

(9, 9, 9, 1, 22000.99),

(10, 10, 10, 2, 1100.99),

(11, 11, 11, 1, 999.99),

(12, 12, 12, 1, 13000.99),

(13, 13, 13, 3, 1499.99),

(14, 14, 14, 2, 17000.99),

(15, 15, 15, 1, 150000.99),

(16, 16, 16, 5, 1200.99),

(17, 17, 17, 3, 21900.99),

(18, 18, 18, 2, 23900.99),

(19, 19, 19, 1, 25000.99),

(20, 20, 20, 2, 2300.99);