

TASK as of 12 SEPTEMBER 2024

I believe this solid project will really test your PHP skills. It is a Stock Management System. This project will give you a comprehensive understanding of PHP and how it interacts with a database. Here's a rundown of the functionalities you'll need to implement:

1. **User Authentication**

- Register: Users should be able to create an account with a username, email, and password. Passwords should be hashed before storage.
- Login: Users should be able to log in with their credentials.
- Logout: Implement a session-based logout mechanism.
- Access Control: Differentiate between regular users and admin users. Admins should have additional privileges.

2. **Dashboard**

- Overview: After logging in, users should see an overview of stock status, including current stock levels, recent transactions, and any alerts or notifications.
- Statistics: Provide basic statistics such as total stock value, recent trends, etc.

3. **Stock Management**

- Add New Stock Items: Allow users to add new stock items, including details like item name, category, quantity, purchase price, selling price, and supplier.
- Update Stock Items: Users should be able to update stock details.
- Delete Stock Items: Implement functionality to remove stock items, with appropriate confirmations to prevent accidental deletions.
- Search and Filter Stock: Users should be able to search for items and filter by various criteria (e.g., category, supplier).

4. **Inventory Tracking**

- Current Stock Levels: Display current stock levels and highlight items that are low in stock.
- Stock Movement History: Maintain a log of all stock movements (additions, removals, adjustments) with timestamps and user information.

5. **Sales and Purchases**

- Record Sales Transactions: Implement functionality to record sales transactions, including item details, quantities sold, and total sales amount.
- Record Purchase Orders: Allow users to create purchase orders, track received items, and update stock levels accordingly.
- Generate Invoices: Generate and print invoices for sales transactions.

6. **Alerts and Notifications**

- Low Stock Alerts: Notify users when stock levels fall below a certain threshold.
- Transaction Notifications: Send notifications for significant transactions or updates.

7. **Reporting and Analytics**

- Sales Reports: Create reports to show sales performance over time, by category, or by user.
- Stock Reports: Generate reports on stock levels, including items that are low in stock or near expiry.
- Profit and Loss Analysis: Calculate and display profit margins for different stock items.

8. **Search and Filter**

- Search: Implement a search functionality to find specific stock items or transactions.
- Filter: Provide options to filter stock and transactions based on various criteria.

9. Responsive Design

- User Interface: Ensure the system is user-friendly and responsive. It should work well on both desktop and mobile devices.

10. Database Design

- Schema: Design a relational database schema to store all the necessary data. Use tables for users, stock items, transactions, etc.

- Normalization: Ensure the database is normalized to reduce redundancy and improve efficiency.

11. Data Import and Export

- Import/Export Stock Data: Allow users to import stock data from CSV files and export stock reports and transaction logs.

12. Security

- Input Validation: Validate and sanitize user input to prevent SQL injection and other vulnerabilities.

- Data Protection: Protect sensitive data, such as passwords and transaction details, with appropriate encryption and hashing.

This project will cover a broad range of PHP functionalities, from basic CRUD operations to more advanced features like user authentication and reporting. Make sure to test thoroughly and document your work as you go.