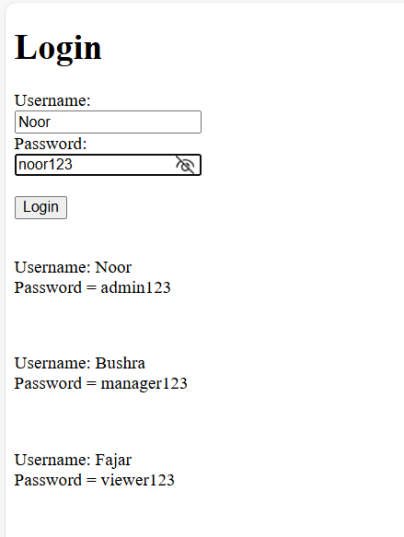
Problem 3

1. Password Hashing

In the login page of the Inventory Management System, user-entered passwords are hashed using the SHA-256 algorithm. The hash is then compared with the stored hashed password in the database. This ensures that even if the database is compromised, raw passwords are never exposed

Testcase:



2. Session Management

After login, session variables track the authenticated user. These sessions ensure pages like Dashboard, Products, etc., are only accessible to logged-in users.

Test case:

You can check session management for example, after a successful login, a page becomes accessible only after verifying session variables like Username and Role. If the session expires or the user logs out, and the session variables are missing, the user is redirected to the login page to prevent unauthorized access.

3. Session Timeout

If the user remains inactive for a set duration (e.g., 20 minutes), the session expires automatically. This prevents unauthorized access if the user leaves the system open.

Test case:

After any inactivity of about 20 minutes you can check that it redirects to login page

4. Role-Based Access Control (RBAC)

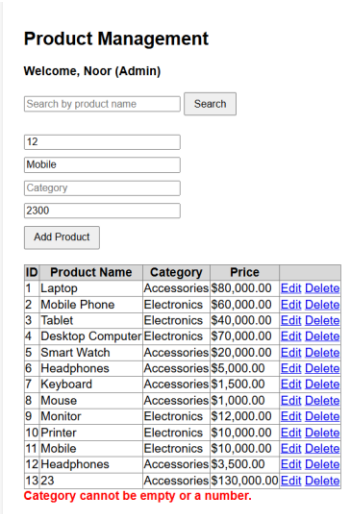
The system defines three roles: Admin, Manager, and Viewer. Each role has restricted access to specific pages. For example, only Admin can add or delete products, while Viewers can only view the inventory data.

Test case:

You can see from the demonstration that role-based access control has been implemented. For example, a *Viewer* can only view the products, an *Admin* has full access, and a *Manager* can update and add products but cannot delete them from the product table.

5. Input Validation

All form inputs like product name, category , price, etc., are validated to ensure correct formats and prevent invalid or malicious data from being submitted.



6. SQL Injection Prevention

Parameterized SQL queries are used throughout the system, preventing direct user input from modifying SQL commands and protecting against SQL injection attacks.

Testcase:

7. Authentication Redirects

Pages like Dashboard.aspx and Products.aspx check if a session exists. If not, users are automatically redirected to the login page, blocking unauthenticated access.

Testcase:

Authentication redirects are implemented to ensure that users are directed to the appropriate page based on their login status—unauthenticated users are redirected to the login page, while authenticated users are taken to the dashboard.