Problem 1

1. Role-Based Access Control (RBAC)

Users are assigned different roles (Admin, Salesperson, Manager), and each role is allowed access to specific pages only. This is implemented using session variables and redirection logic to prevent unauthorized access to admin/sales/manager pages.

2. Input Validation

In forms, numeric fields are validated using TryParse to ensure only correct data types are submitted. Required field checks are also present to avoid empty submissions.

3. Server-Side Access Control

Even if a user manually changes the URL, server-side checks (in Page_Load events) prevent unauthorized access.

4. Parameterized Queries (SQL Injection Prevention)

All SQL queries use parameters (@param) with SqlCommand.Parameters.AddWithValue, preventing SQL injection attacks.

5. Authentication and Session Management

After login, user roles and IDs are stored in session variables. Access to pages is restricted based on whether the session is active or not. This prevents access to protected pages without logging in.

6. Feedback and Error Handling

The system provides safe error messages like "Product ID already exists" without exposing internal SQL errors. This reduces the risk of information leakage to potential attackers.

7. Form Resubmission Prevention

Use of IsPostBack ensures that dropdowns and form components are only populated once, preventing repeated DB hits and resubmissions that could lead to duplicate entries or logic bugs.