Security Features of ASP.NET Food Delivery System Project

Authentication (User Login System)

Only registered users (Customers and Suppliers) should be able to access specific parts of the system. This is implemented using the Users_T table which stores hashed passwords and user roles. ASP.NET handles authentication to verify identity before granting access.

Authorization (Data-Driven Access Control)

Each page has authorization rules. These rules define which pages can be accessed by which user roles. Unauthorized access attempts are blocked during page_load event, adding an additional layer of security before code execution.

Password Hashing and Secure Storage

Passwords are stored in the Users table using hashed values (not plain text). The hashing algorithm SHA-256 guarantees that even if the database is compromised, passwords cannot be easily retrieved.

SQL Injection Prevention

All SQL queries in the code use parameterized queries (e.g., cmd.Parameters.AddWithValue(...)), which prevent attackers from injecting malicious SQL commands through form inputs. This is a core defense against one of the most common web vulnerabilities.

Input Validation and Field Validators

ASP.NET RequiredFieldValidator, RegularExpressionValidator, and RangeValidator ensure that user input is valid (e.g., numeric quantities, non-empty product names). This helps prevent attacks like script injection (XSS) and logic errors in the app.

Session Management

User sessions are securely maintained after login using ASP.NET sessions (e.g., Session("Customer_ID")). Sessions are automatically timed out after inactivity to reduce the risk of session hijacking.

Error Handling and Exception Logging

Try-catch blocks are used around database insertions. This prevents sensitive error messages from being shown to users and instead logs errors securely. It helps avoid information disclosure and improves system reliability.

Security Features Testing:

1. **Password Hashing**

   Ensure password is hashed before storage in the database

   Register user with password "test123"

   The password will be stored as a Hash

2. **Input Validation**

   Entering incorrect values will result in incorrect input

3. **Role Based Access Control**

   Only appropriate User can access database

   A supplier tries to access Customer page

   Access will be denied

4. **Session Management**

   Pages require session authentication

   Try to access customer.aspx without login

   Redirect to login page

5. **Secure Login**

   Login only works with valid username and password

6. **Error Handling**

   Valid error messages are shown in appropriate places

7. SQL Injection Prevention

A user attempts to log in, and the application uses parametrized queries to prevent SQL injection.