Project Report – Food Store Application

By Abu Bakar

1. Overview

The **Food Store Application** is a feature-rich online shopping platform that allows users to browse products, add items to a shopping cart, and complete purchases through a secure checkout process. It supports user registration and authentication, dynamic navigation based on login status, and robust order management.

A core focus of the application is comprehensive **activity logging**, with logs stored either in the file system or a dedicated SQL database table, enabling reliable monitoring and auditing.

2. Architecture and Design

The application follows a **layered architecture**, ensuring a clean separation of concerns:

- **Presentation Layer**: Developed using **ASP.NET Web Forms**, it employs dynamic controls such as LoginView to personalize content based on the user's authentication status.
- **Business Logic Layer**: Implemented in **VB.NET code-behind files**, it handles product listings, cart operations, authentication, and order processing.
- **Data Access Layer**: Built with **ADO.NET**, it uses **parameterized SQL queries** to ensure secure, efficient database interaction.
- **Logging Mechanism**: A centralized and modular logging system captures user actions and system events. Logs include details like timestamp, URL, HTTP method, IP address, and user identity. They can be stored in text files (App_Data) or the ActivityLogs database table.

3. Key Features and Technical Highlights

Dynamic Navigation and Authentication

- Utilizes the LoginView control for context-aware navigation.
- Implements Forms Authentication for secure session management.
- Logged-in users receive a personalized interface and experience (Pending/I'm able to do it on local but not on somee).

Shopping Cart and Order Management

- Supports **session-based cart management**, allowing even anonymous users to browse and add items.
- Checkout process requires login, ensuring user identification before transactions.
- Features include quantity updates, product removal, and client/server-side validation for checkout.

Database Connectivity and Security

• Uses secure ADO.NET components: connection objects, commands, and data readers.

- Employs parameterized SQL queries to defend against SQL injection attacks.
- Customer and order data are maintained in normalized, well-structured tables.

Logging and Monitoring

- Logs every request with key metadata (timestamp, HTTP method, user ID, etc.).
- Offers dual log storage options: plain-text logs or SQL table entries.
- Designed with **fail-safe mechanisms**—if one logging method fails, alternate paths ensure continuity.
- Facilitates **real-time monitoring**, troubleshooting, and security auditing.

Robust Error Handling

- Global error handlers maintain application stability even in failure scenarios.
- Logging components are isolated with fallback logic to capture issues without interrupting the user experience.

4. Security and Best Practices

- Secure authentication via **Forms Authentication**.
- Sensitive session data is used conservatively to balance personalization with privacy.
- All database interactions are protected by **parameterized queries** to maintain **data integrity and security**.
- Emphasizes separation of concerns and modular design to ensure long-term maintainability.

5. Conclusion

The **Food Store Application** is a well-architected, secure, and scalable e-commerce solution that exemplifies:

- Tried to use Best practices in web application design,
- Effective user experience through dynamic personalization,
- And powerful, modular logging for auditing and diagnostics.

It provides a solid foundation for future enhancements.