**Iteration 2**

**Plain Text Password Storage**: Storing passwords as plain text in the database makes it easy for anyone with access to the database to read them. This is highly insecure.

**SQL Injection**: The code uses parameterized queries, which reduces the risk of SQL injection, but it's always important to validate and sanitize user inputs thoroughly.

**Lack of Encryption**: There is no use of encryption for sensitive data both in transit and at rest. This can expose the data to interception and unauthorized access.

**Weak Authentication Mechanism**: The authentication mechanism is quite basic and lacks features like account lockout after multiple failed attempts, which could help mitigate brute force attacks.

**No Password Complexity Requirement**: The code doesn't enforce any password complexity rules, making it easier for attackers to guess simple passwords.

**Hardcoded Admin Credentials**: Having hardcoded credentials in the source code is a poor practice, as it can be easily discovered and misused.

**No User Session Management**: The application does not have a proper session management system. It simply switches views based on a login status, which can be easily bypassed.

**No Input Validation**: The code doesn't validate the inputs (e.g., ensuring usernames and passwords meet certain criteria), which could lead to unexpected behavior or vulnerabilities.