UNIVERSITY OF THE PUNJAB
Session: BS (IT) Fall 2021 Morning
Course: Web Technologies
Homework 02

Technical Concepts:

ADO.NET (ActiveX Data Object), Parameterized Queries, SQL Injection

Objective:

Dear students, we will be engaging in hands-on practice with the technical concepts mentioned above. These concepts are crucial for anyone looking **to earn money** online. Querying databases is essential skill for becoming a proficient web developer, and we will be using ADO.NET for this purpose. Parameterized queries will prove to be particularly useful in safeguarding against security vulnerabilities like SQL injection. These techniques are essential for developing secure and reliable web applications. **Let's practice** these concepts and add some more tools and techniques to our C# Web Developer Toolkit!

Happy Learning!

Scenario:

Let's assume you are hired as C# Web Developer at *QueenShop*, a clothing brand that sells the latest fashions designed especially to boost confidence. So you are required to make a local DB to store order information. This will enable *QueenShop* to efficiently track and manage orders. Typically, companies store order details in their databases before forwarding them to relevant courier partners via API integration. So, for now, you are not required to integrate any Courier Partner because QueenShop's Procurement Management team is still looking for

better Courier Services Provider. And UI/UX team is busy in designing order taking form. So, you can relax after implementing the basic database infrastructure!

Excited..?
Let's Go!

Task-1: (10 marks)

1. Create a new Project. Name it as QueenLocalDataHandling.
2. Create a new Database File. Name it as QueensDB.
3. Create a Table in QueensDB. Name the table as Orders.

It should have following attributes to save order details

a) Order ID
b) Customer's CNIC
c) Customer's Name
d) Customer's Phone
e) Customer's Address
f) Product ID
g) Price
h) Size of Product (Large, Medium, Small etc.,)

5. Create the class Order in the Project with necessary attributes.
6. Create a CRUD operations class for querying OueensDB database. Name it as OrderCRUD
7. Implement a InsertOrder method in OrderCRUD to insert new Order in QueensDB by taking Order object as parameter.
8. Add another method to get all records/rows from table Orders and display on screen. Name it as GetAllOrders.
9. Create a new member function 'UpdateAddress' to update Customer's Address by using Customer's Phone No
10. Insert a new method 'DeleteOrder' to delete order on the basis of order ID.

If you have done task-1. Great!!

Task-2: (20 Marks)

Now, let's understand the usage & importance of Parameterized Queries!
Consider an example of a SQL injection attack and how parameterized queries can prevent it:

1. SQL Injection Attack without Parameterized Queries:

Let's say we have a vulnerable login form on a website with the following SQL query to authenticate users:

```
SELECT * FROM users WHERE username = '$username' AND password = '$password';
```

The hacker enters the following credentials into the login form:
Username: admin'--
Password: any_password

The SQL query becomes:

```
SELECT * FROM users WHERE username = 'admin'--' AND password = 'any_password';
```

The -- in SQL comments out the rest of the query, effectively bypassing the password check. So the hacker logs in as the admin without knowing the correct password.

2. SQL Injection Attack Prevention with Parameterized Queries:

Now, let's see how using parameterized queries prevents such attacks. The same login form with a parameterized query would look like this:

```
SELECT * FROM users WHERE username = ? AND password = ?;
```

When the hacker tries the same payload:

Username: admin'—

Password: any_password

Even if the hacker tries to inject malicious code, the parameterized query treats the input as data, not as part of the SQL statement. So the hacker's attempt to bypass the password check fails, and the user is not authenticated.

Now, we have understood the usage of Parameterized queries. Let's update our OrderCRUD class with improved method. QueenStore don't want that some hacker can change their customers' order address. So, let's update with a new one!

11. Create another method, UpdateOrderAddress, that will take phone number and updated address and set them in DB. But use parameterized query to make it secure!

Don't delete the old UpdateOrder method.

Task-3 (5 marks)

Now, the founder of QueenStore wants to see your good job. Let's create a menu for her!

Simply create a menu and demonstrate the working of all methods!

Congratulations, All Done!