Activity 6: Login System with Token-Based Authentication & String Exercises

Objective:

The objective is to create a **login system** using a **2D array** to store usernames and passwords. The program will:

- Allow users **three login attempts** before locking them out.
- Generate a **token** upon successful login (concatenation of the reversed password + username).
- Implement a **menu-driven system** with authentication-based access.
- Include **interactive string manipulation exercises** to reinforce key string methods.

Implementation Steps:

1. Create a C# Console Application

- Open Visual Studio or any C# IDE.
- Create a new **C# Console Application**.

2. Define User Storage & Login System

- Use a **2D array** to store usernames and passwords.
- Implement a **case-insensitive login check** for usernames.
- Allow **three login attempts** before exiting.
- Generate a **token** for logged-in users.

3. Implement the Main Menu

The program will display the following options:

1️⃣ **Login** – Allows users to log in and generate a token.
2️⃣ **Global Check (String Exercises)** – Accessible without login.
3️⃣ **Authorization Check** – Requires a token to access.
4️⃣ **Logout** – Clears the token and returns to the main menu.

---

4. Implement Option 1: Login

- Prompt the user for a **username** (case-insensitive) and **password**.
- Validate credentials from the **2D array**.
- On success, generate a **token** (reversed password + username).
- On failure, decrement attempts and lock the user out after 3 failed tries.

5. Implement Option 2: Global Check (String Exercises)

This option includes **interactive string challenges**:

Exercise 1: Reverse a Sentence

- The user inputs a sentence.
- The program **reverses the words** and displays the result.

Exercise 2: Word Search in a Paragraph

- A predefined paragraph is displayed.
- The user enters a word to search.
- The program **checks if the word exists** using string.Contains().

6. Implement Option 3: Authorization Check (Restricted Access)

- If the **token is present**, display:
  "You're currently viewing this page as an authorized user."
- If **no token is found**, display:
  "Unauthorized view. Please log in to access this page."

7. Implement Option 4: Logout

- Clears the stored token and logs the user out.
- Returns to the main menu.

8. Keep the Program Running

- The **menu loops continuously** until the user selects **Logout**.
- After logout, the program **resets** for another user session.

secured-login-system

```
Welcome! Choose an option:
1. Login
2. Global Check (String Exercises)
3. Authorization Check
4. Logout
Enter your choice: 1

Enter Username: Admin
Enter Password: admin123
Login Successful!
Generated Token: 321nimdaadmin

Welcome! Choose an option:
1. Login
2. Global Check (String Exercises)
3. Authorization Check
4. Logout
Enter your choice: 2

Welcome to the Global Check!
Choose an activity:
1. Reverse a sentence
2. Search for a word in a paragraph
3. Go back to the main menu

Enter your choice: 1
Enter a sentence: Learning C# is fun!
Reversed sentence: fun! is C# Learning

Choose an activity:
1. Reverse a sentence
2. Search for a word in a paragraph
3. Go back to the main menu

Enter your choice: 3
Returning to the main menu...

Welcome! Choose an option:
1. Login
2. Global Check (String Exercises)
3. Authorization Check
4. Logout
Enter your choice: 3

Unauthorized view. Please log in to access this page.

Welcome! Choose an option:
1. Login
2. Global Check (String Exercises)
3. Authorization Check
4. Logout
Enter your choice: 4

Logging out...
Token cleared.
```