**TASK-2 (WEB APP SEC)**

**TARGET:**

**1. Find 3 Broken Authentication Vulnerable Websites due to SQL injection.**

**2. Find 2 Broken Authentication Vulnerable Websites due to email verification bypass vulnerability while registration in a website.**

**3. Find 1 Broken Authentication Vulnerable Website due to OTP bypass vulnerability i.e two factor authentication vulnerable.**

**SYNOPSIS:**

**Broken Authentication Vulnerability:**

Broken authentication vulnerability refers to a security weakness in an application’s authentication mechanism that allows unauthorised access to user accounts or system resources. It typically occurs when there are flaws or misconfigurations in the implementation of authentication controls, such as weak password policies, improper session management, or predictable session identifiers.

Here are some common examples of broken authentication vulnerabilities:

1) Weak passwords

2) Insecure session management

3) credential stuffing

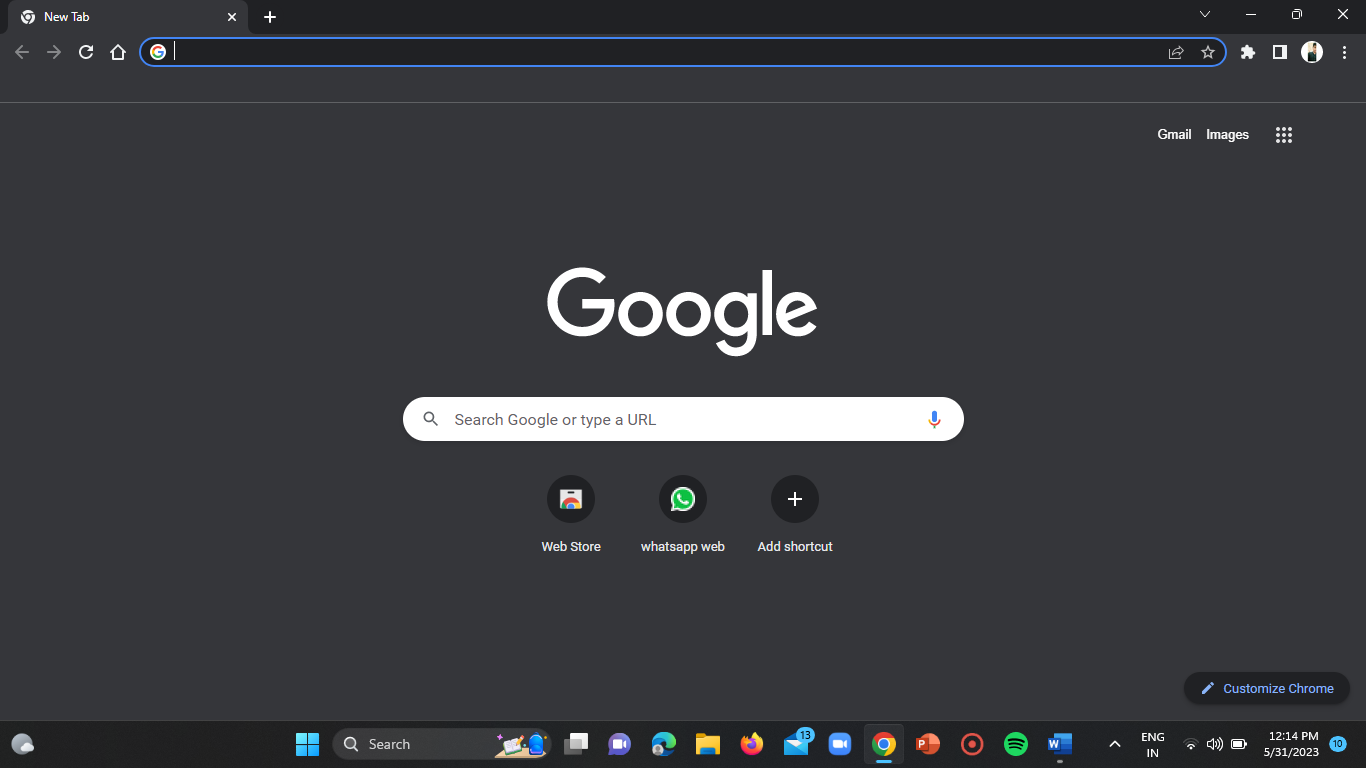
4) Brute-Force Attacks

**PROCEDURE:**

**Broken authentication vulnerable websites:**

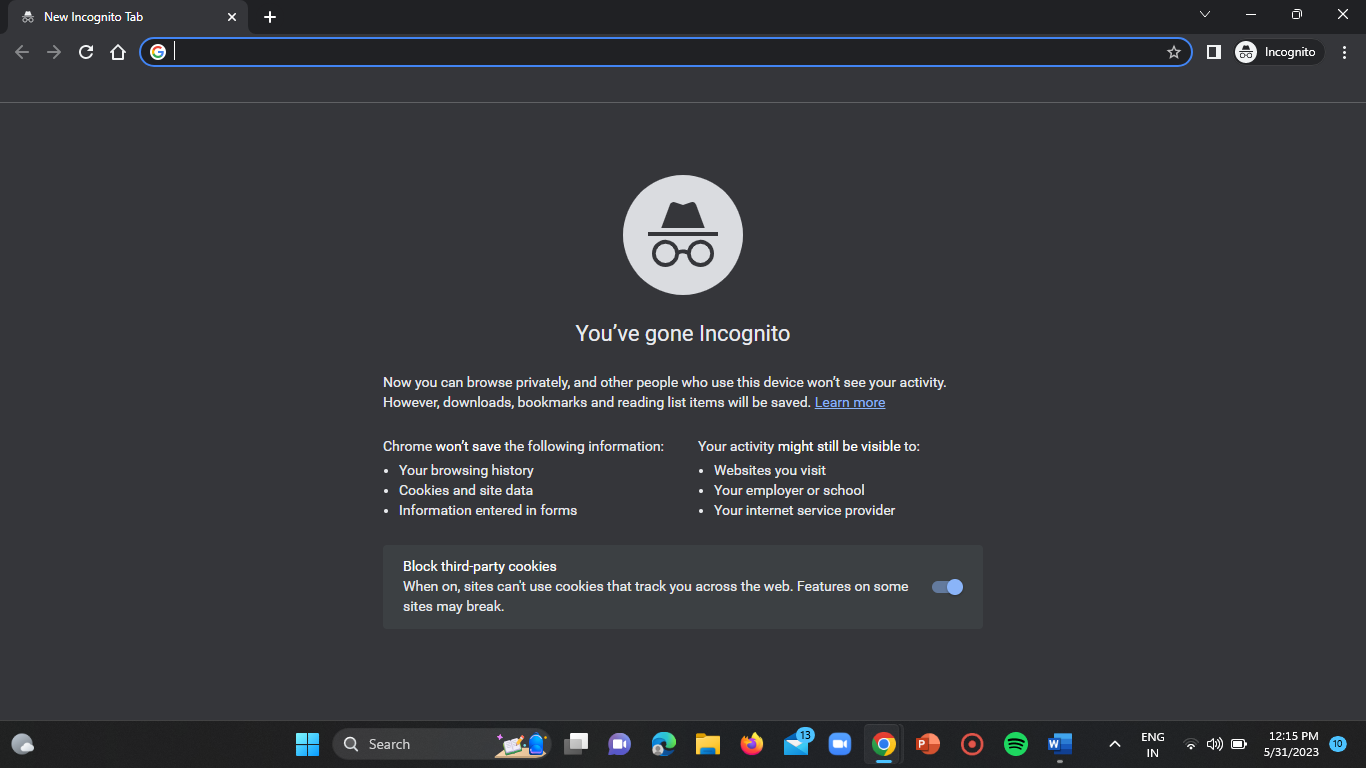
**Step-1:**

Open google chrome.



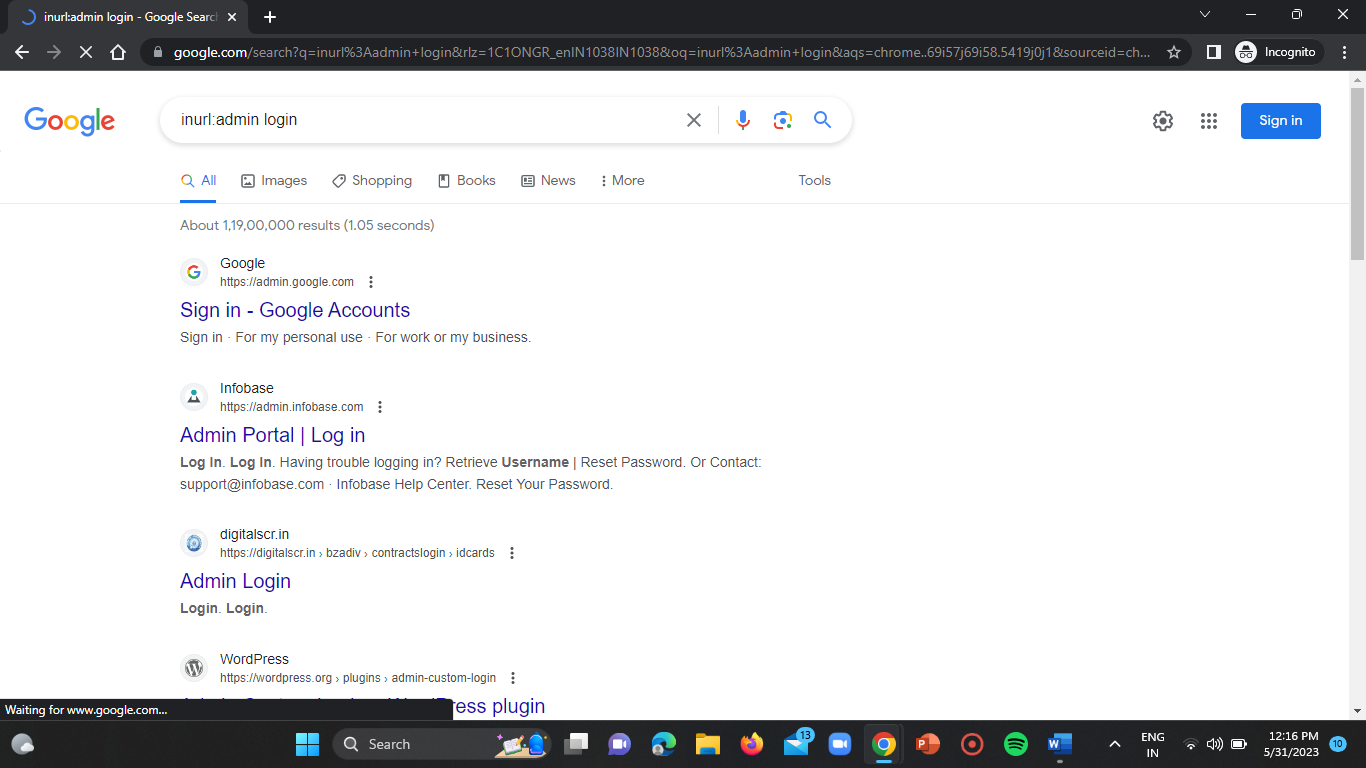
**Step-2:**

Open incognito window using the shortcut “CTRL+SHIFT+N”.



**Step-3:**

Search the google dork “inurl:admin login”.

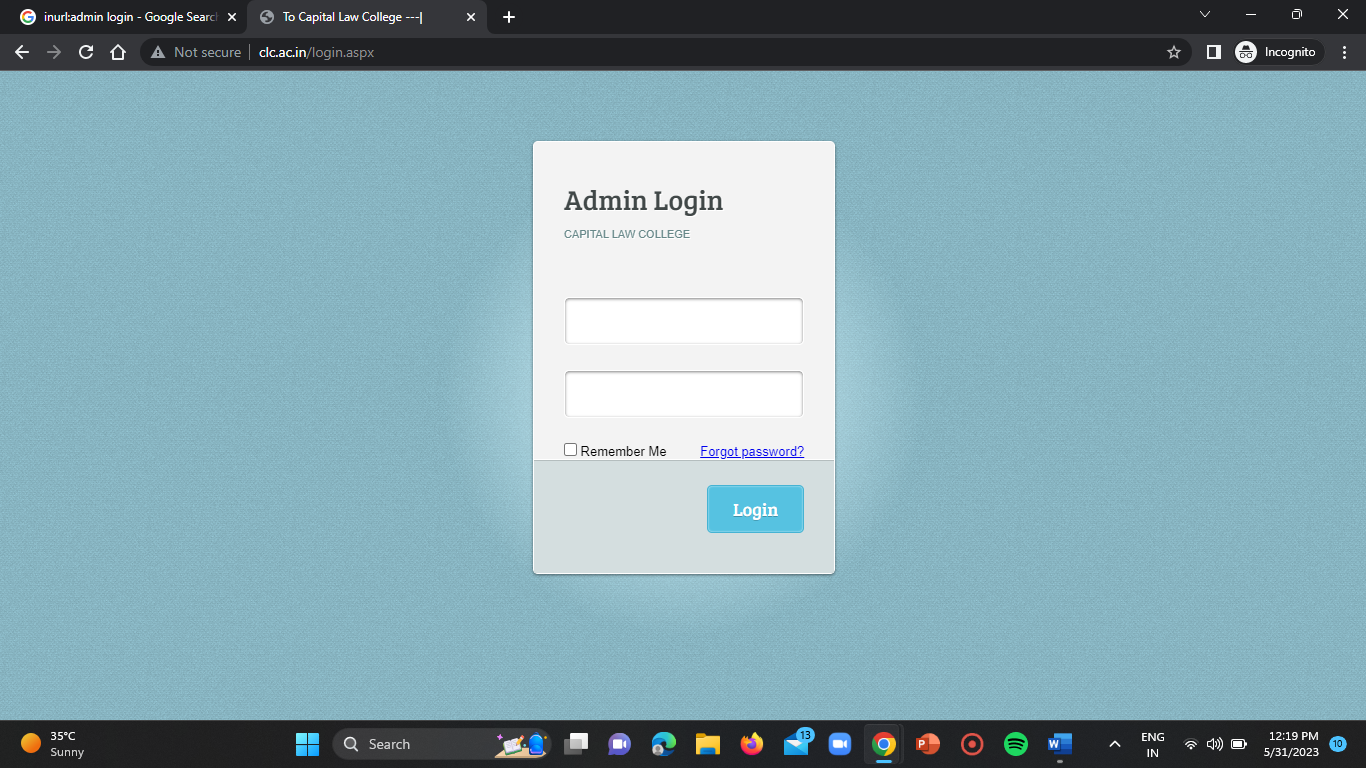


**Step-4:**

Now test the exploited authentication vulnerable websites by using the provided admin login pages.

**WEBSITE-1:**

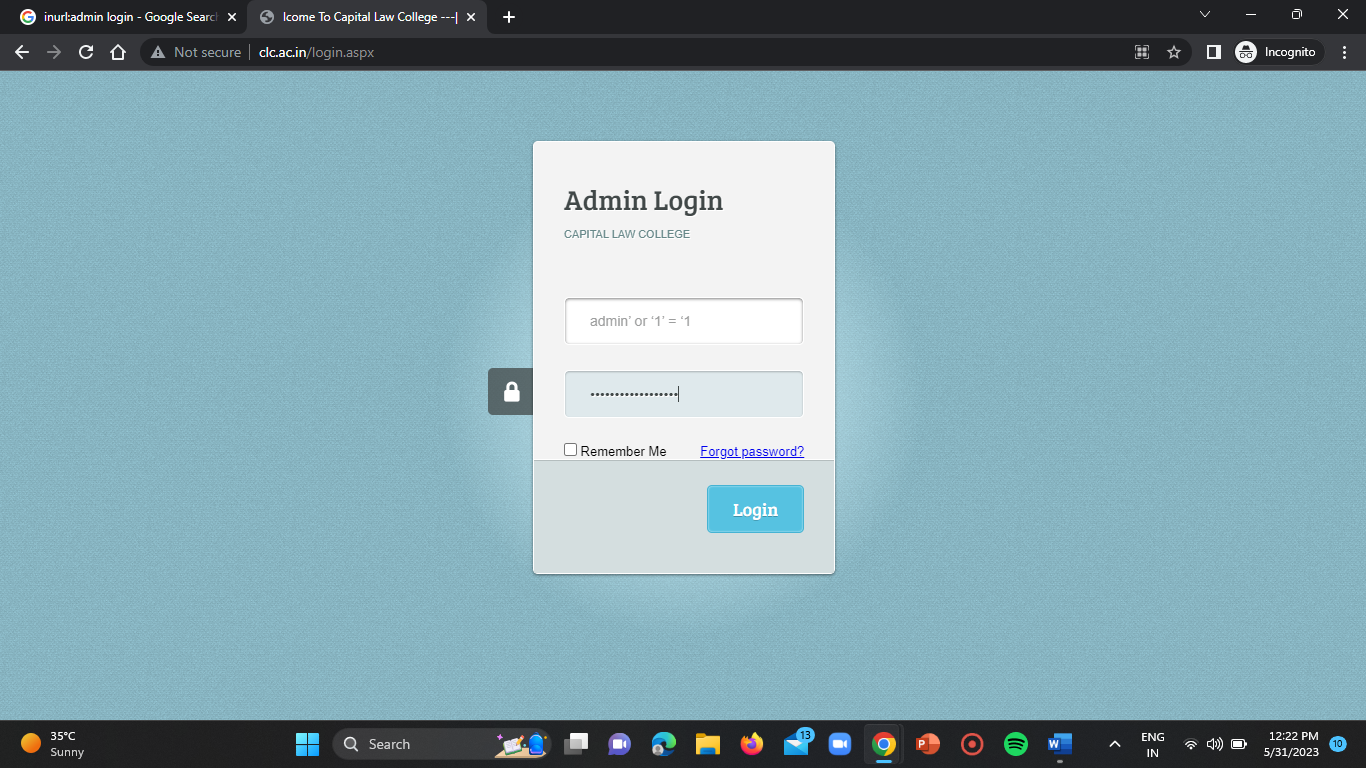
**URL:** <http://clc.ac.in/login.aspx>



**Step-5:**

Now enter the below mentioned payload in the username and the password input bars for testing the vulnerability.

**Payload:** admin’ or ‘1’ = ‘1



**Step-6:**

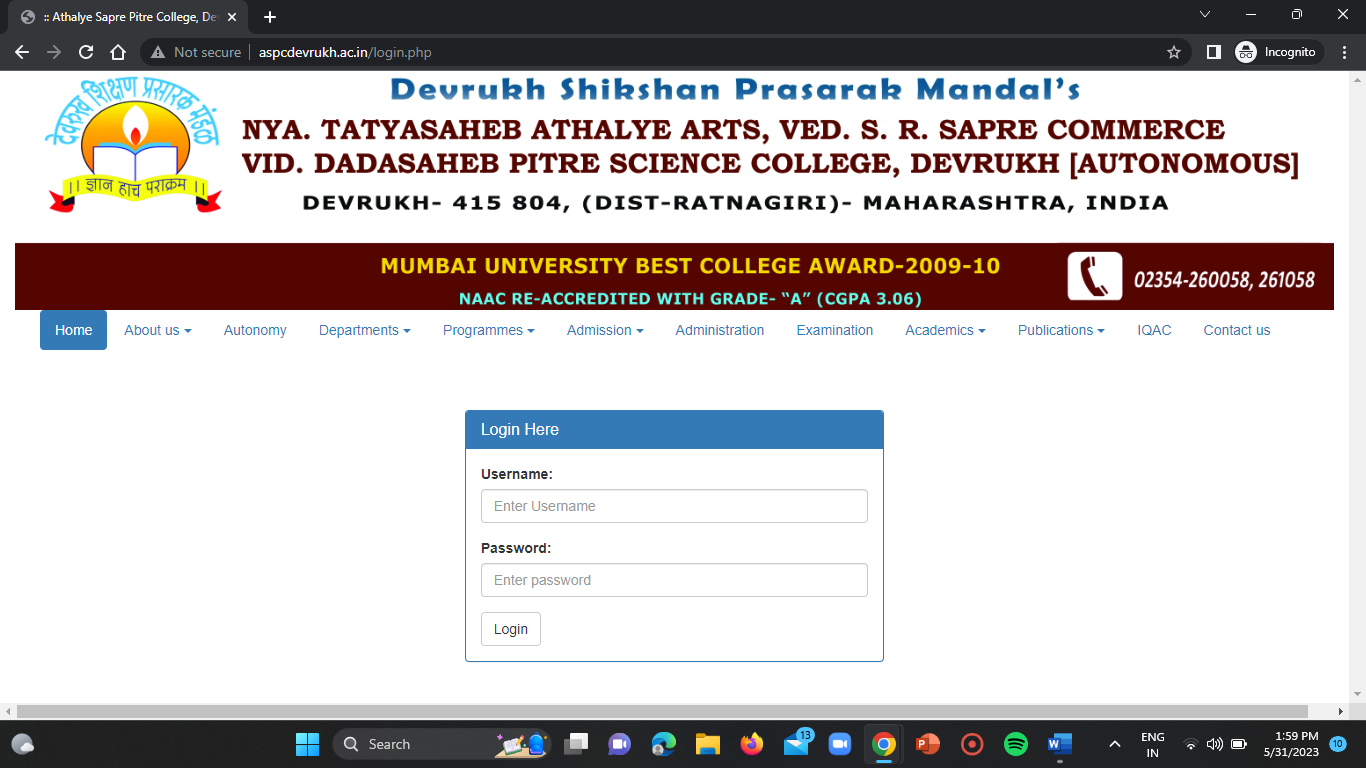
Now click on login.



Login was successful and the information got displayed successfully. So, the domain is vulnerable.

**WEBSITE-2:**

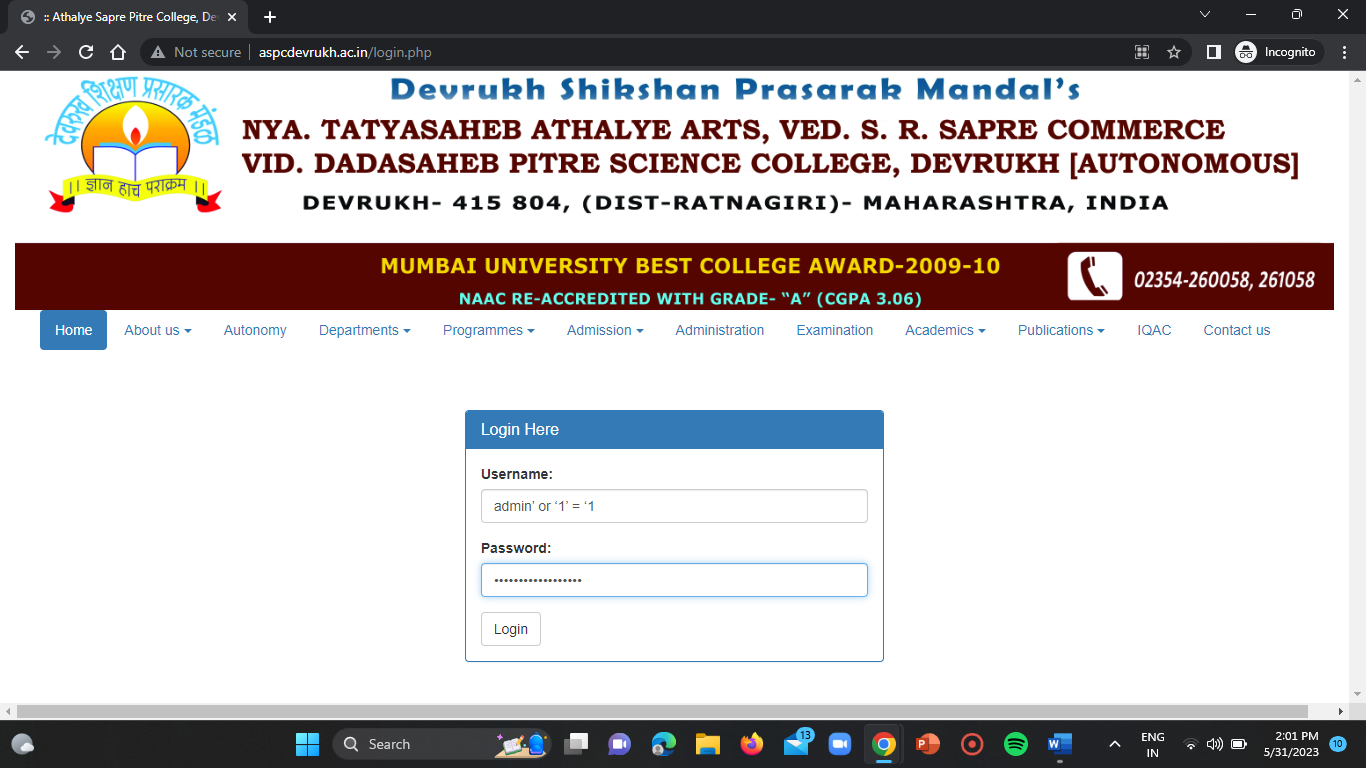
**URL =** <http://aspcdevrukh.ac.in/login.php>



Now repeat the above steps using the given payload.

**Payload:** admin’ or ‘1’ = ‘1

Enter the payload in the login input fields.



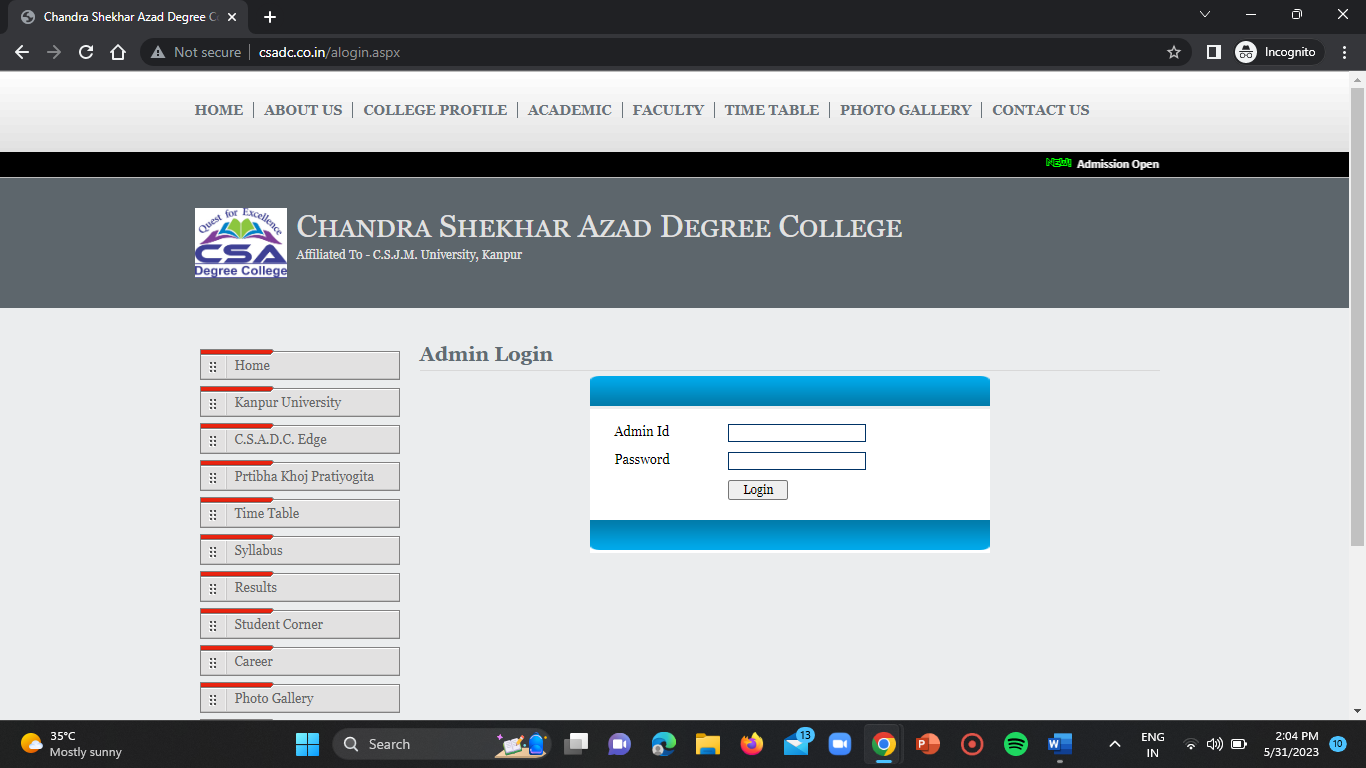
Now click on login.



Login was successful. Hence, the domain is vulnerable.

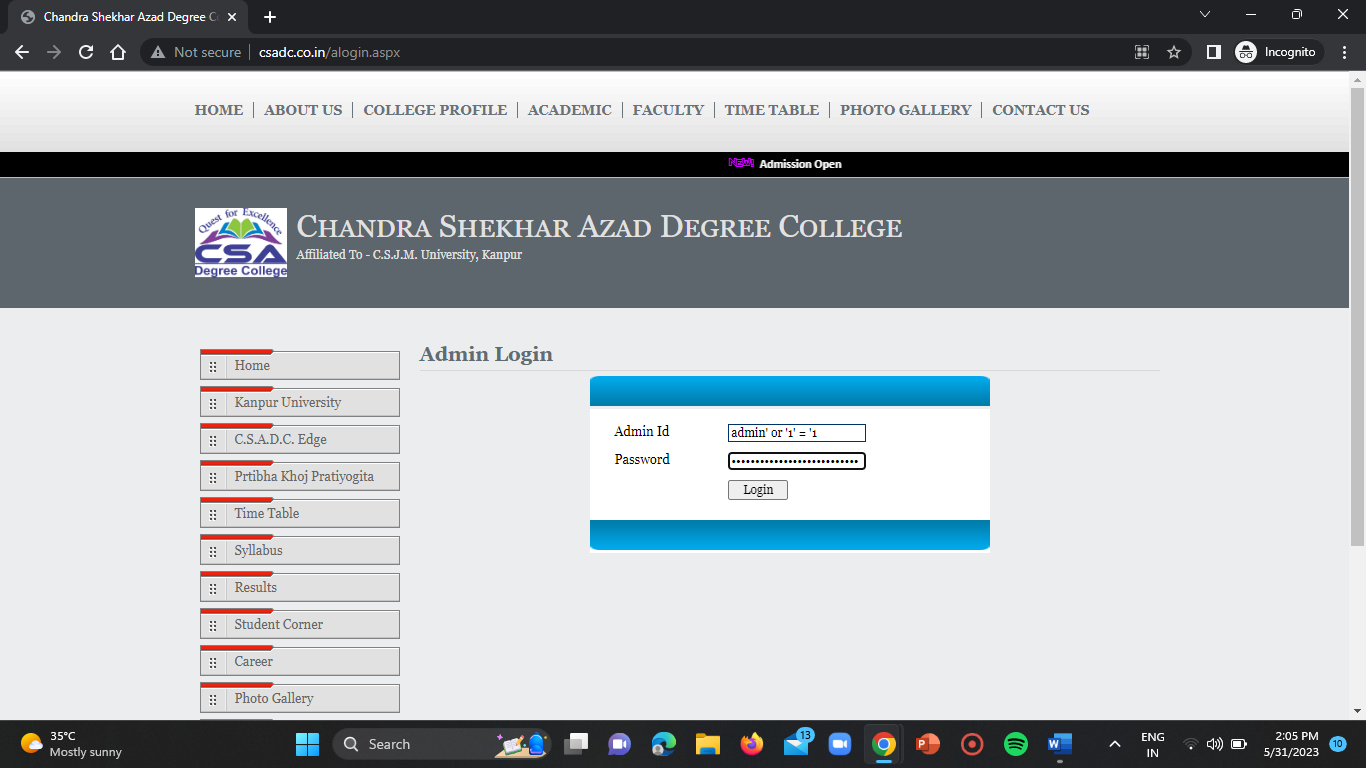
**WEBSITE-3:**

**URL =** <http://csadc.co.in/alogin.aspx>

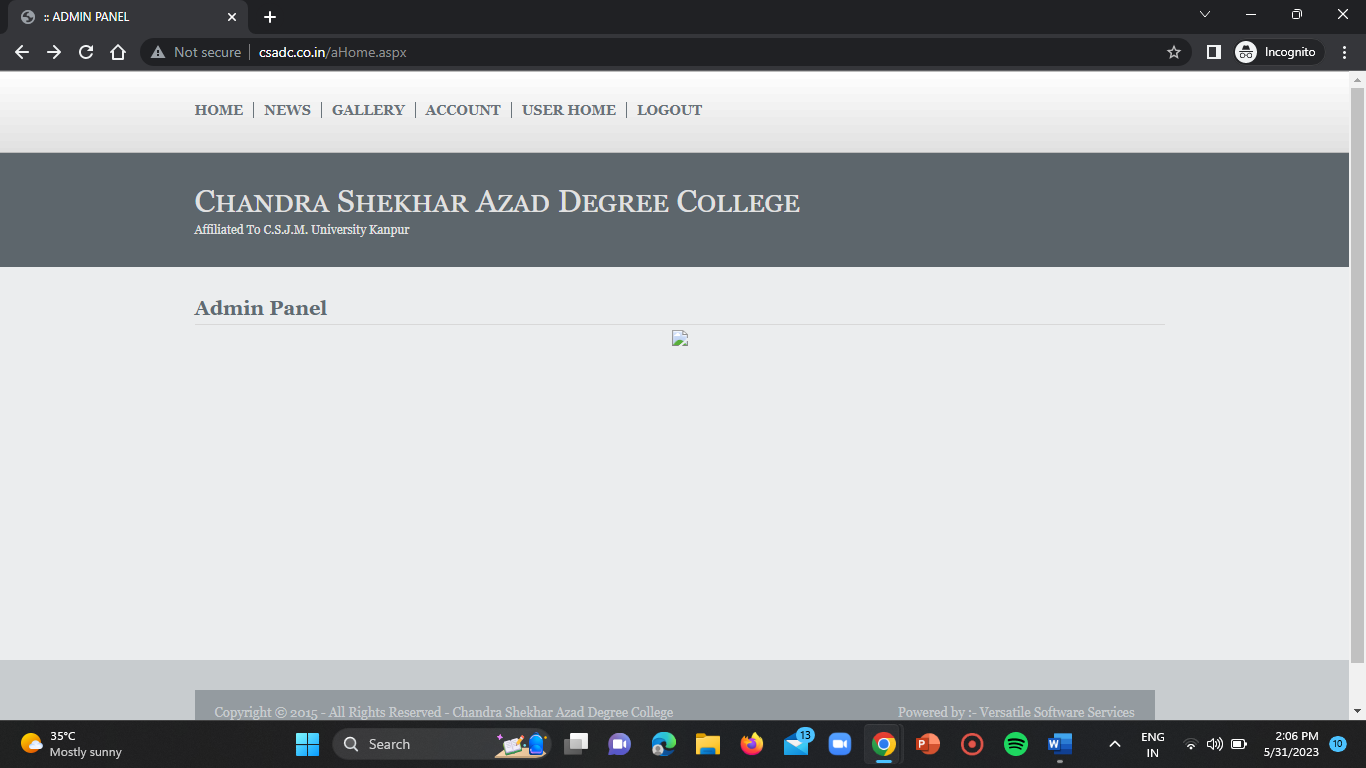


Now enter the payload in both the login input fields.

**Payload:** admin’ or ‘1’ = ‘1



Now click on login button.



The login was successful. So, the page is vulnerable.

**IMPACT:**

1) Data access is not acknowledged.

2) Data can be modified or removed.

3) Datasets can be spied.

**REMEDIATION ACTION:**

1) Cleaning up and assessing the input.

2) Substituting raw SQL queries with parameterised ones.

**CONCLUSION:**

By this task I conclude that the SQL injection is a method by which malicious actors alter the SQL queries and gain unauthorised access, data modification and theft, or even system compromise and utilise the admin privileges and personal information for their own benefits. There are many impacts like data modification and access to unacknowledged, spying the datasets.