**TASK – 6**

**TARGET:**

1. 5 websites with “error-based SQL injection” vulnerability.
2. 5 websites with “login bypass using SQL injection” vulnerability.
3. 5 websites with “broken access control” vulnerability.

SYNOPSIS:

* **ERROR BASED SQL INJECTION:**

It throws the error messages given by the database servers. The error messages by the database server are directly visible on client side revealing the information of the database server.

* **LOGIN BYPASS USING SQL INJECTION:**

It is a way to bypass login mechanism of a system with single user i.e admin.

* **BROKEN ACCESS CONTROL:**

It is a security vulnerability where critical and confidential files do not have any security mechanism to protect from unauthorised access. The files can be accessed by any one on the network.

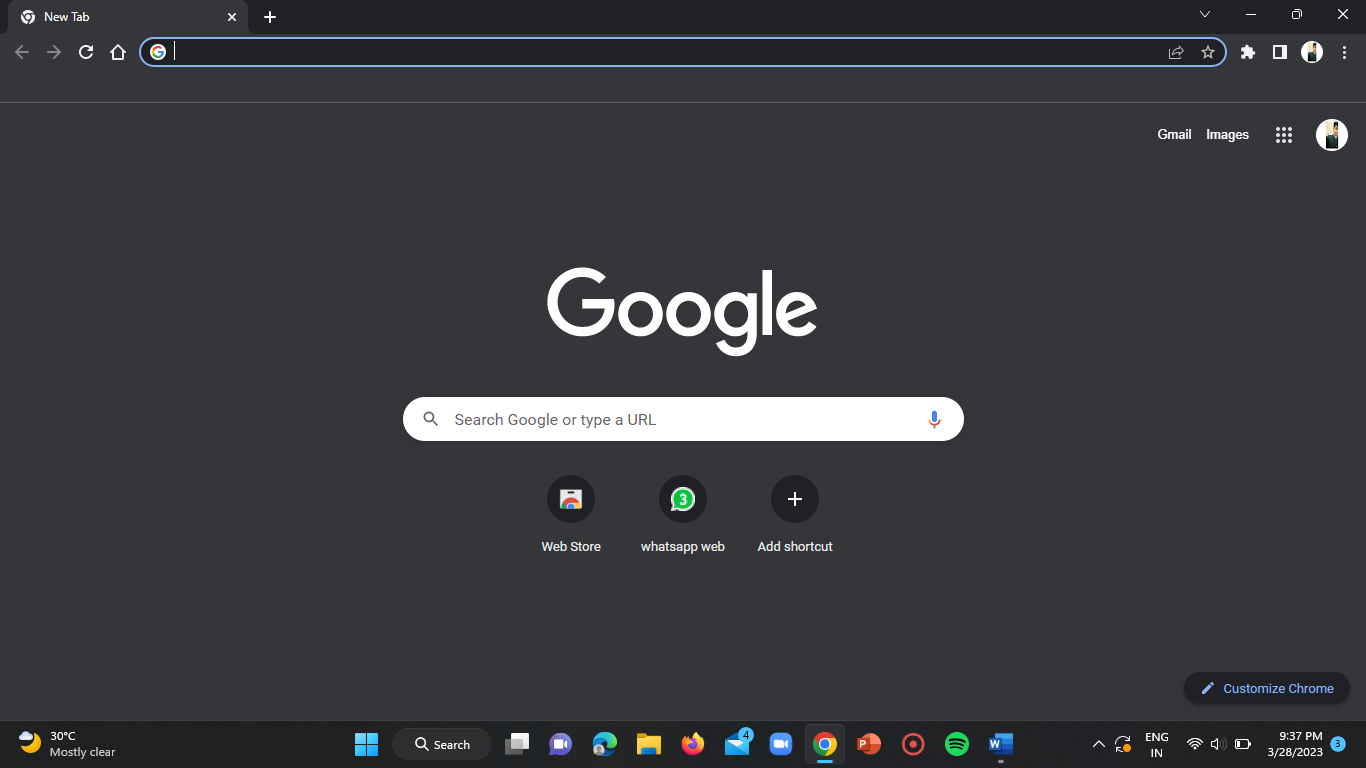
SOLUTION:

**ERROR BASED SQL INJECTION:**

* Here we are going to find out five websites with the “error based SQL injection” vulnerability.

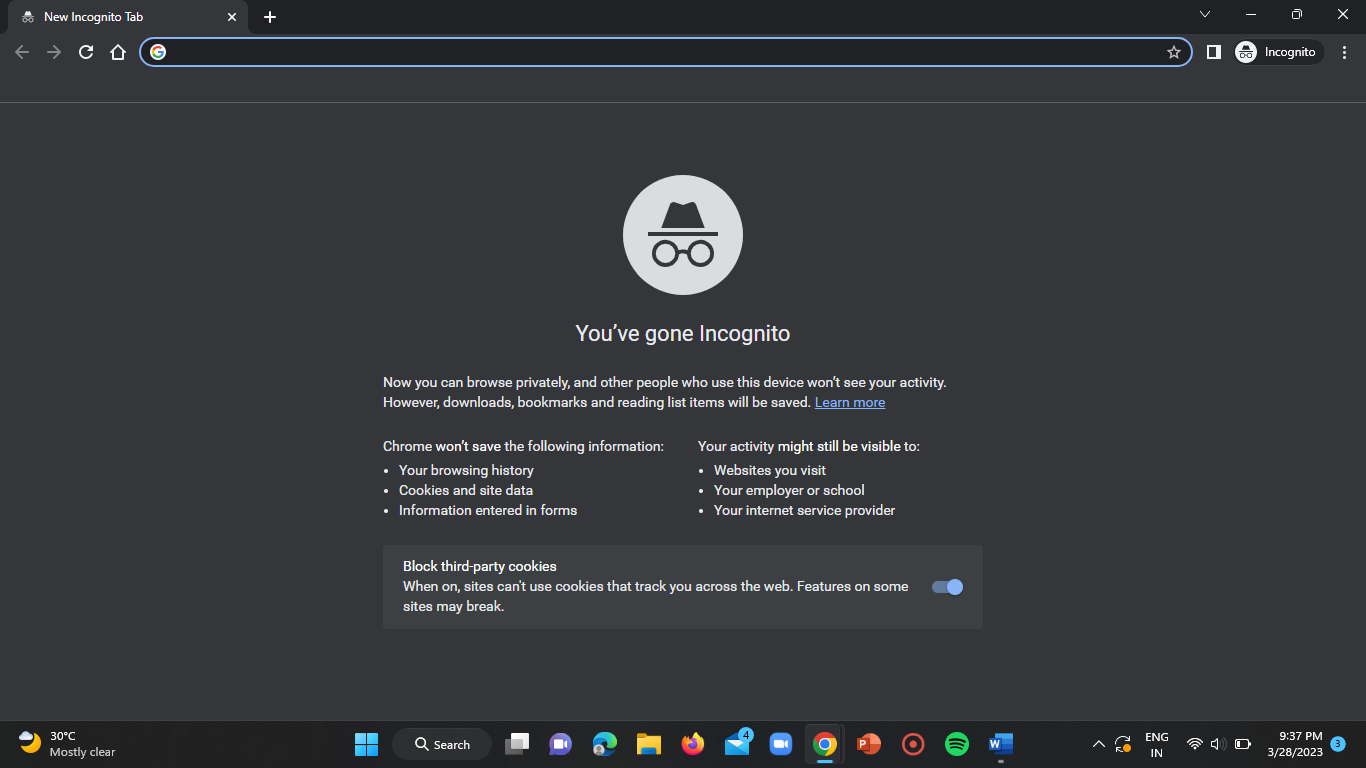
**STEP-1:**

Open google chrome browser.



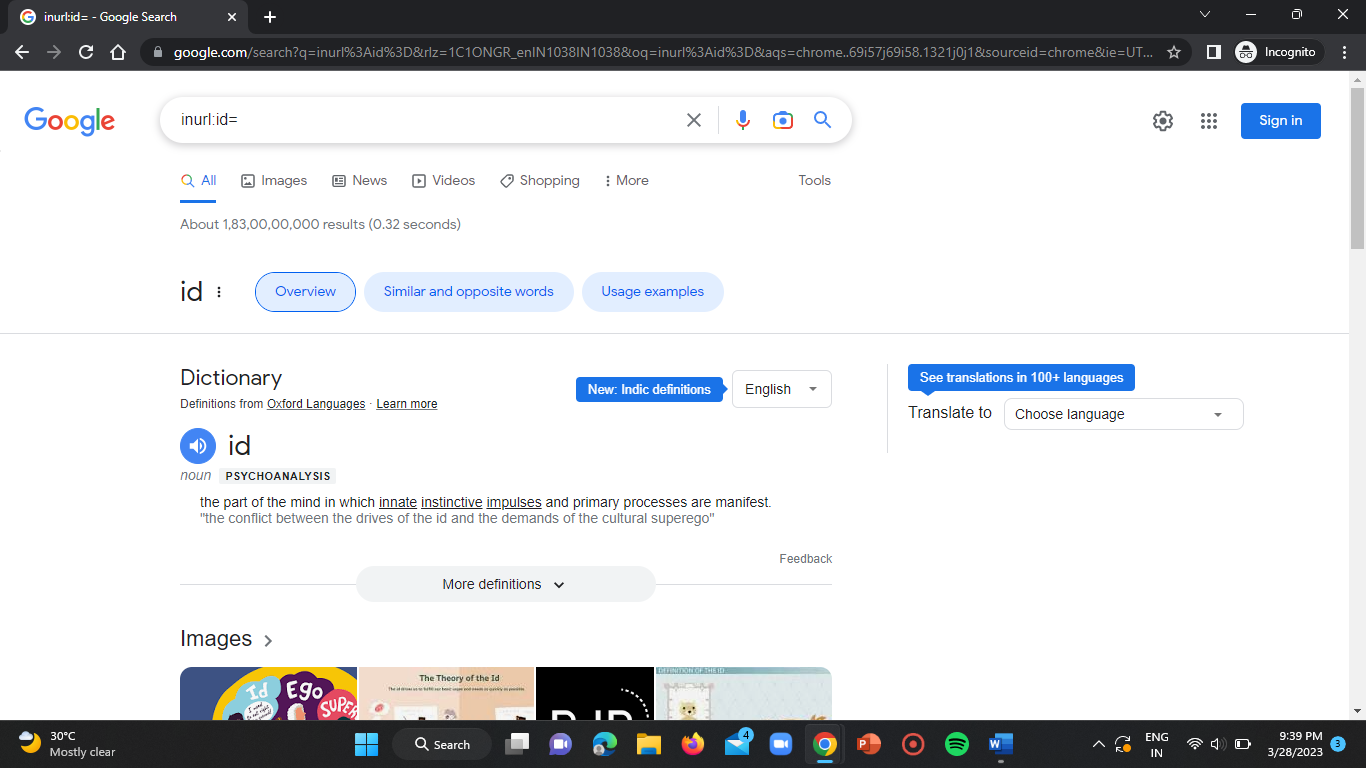
**STEP-2:**

Open an incognito tab in your google browser.



**STEP-3:**

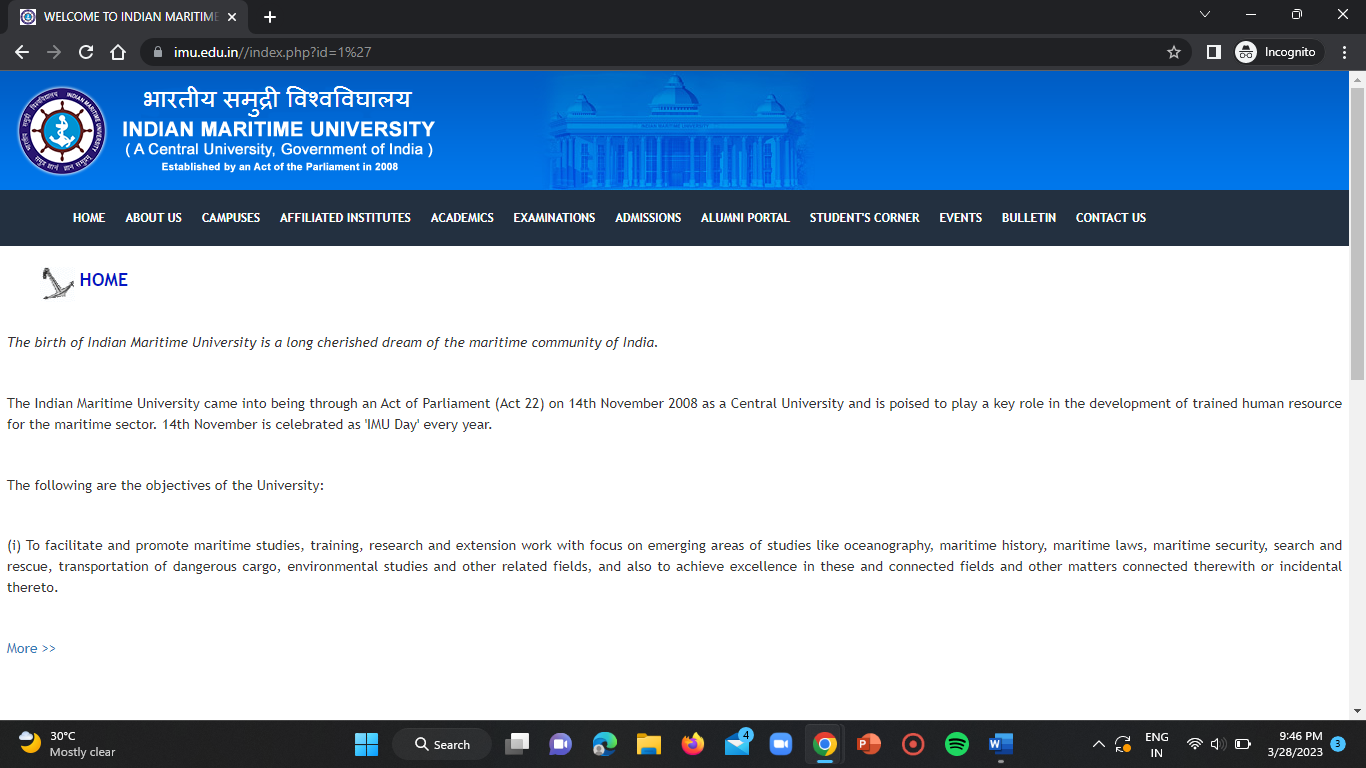
Firstly, I’ve used the google dork “inurl:id=” to find the websites that use database to store the data with ID parameter.



**STEP-4:**

I’ve visited every website one by one and added a single quote (‘) at the end of the URL.





Here the %27 after id=1 in the second screenshot indicates the single quote(‘).

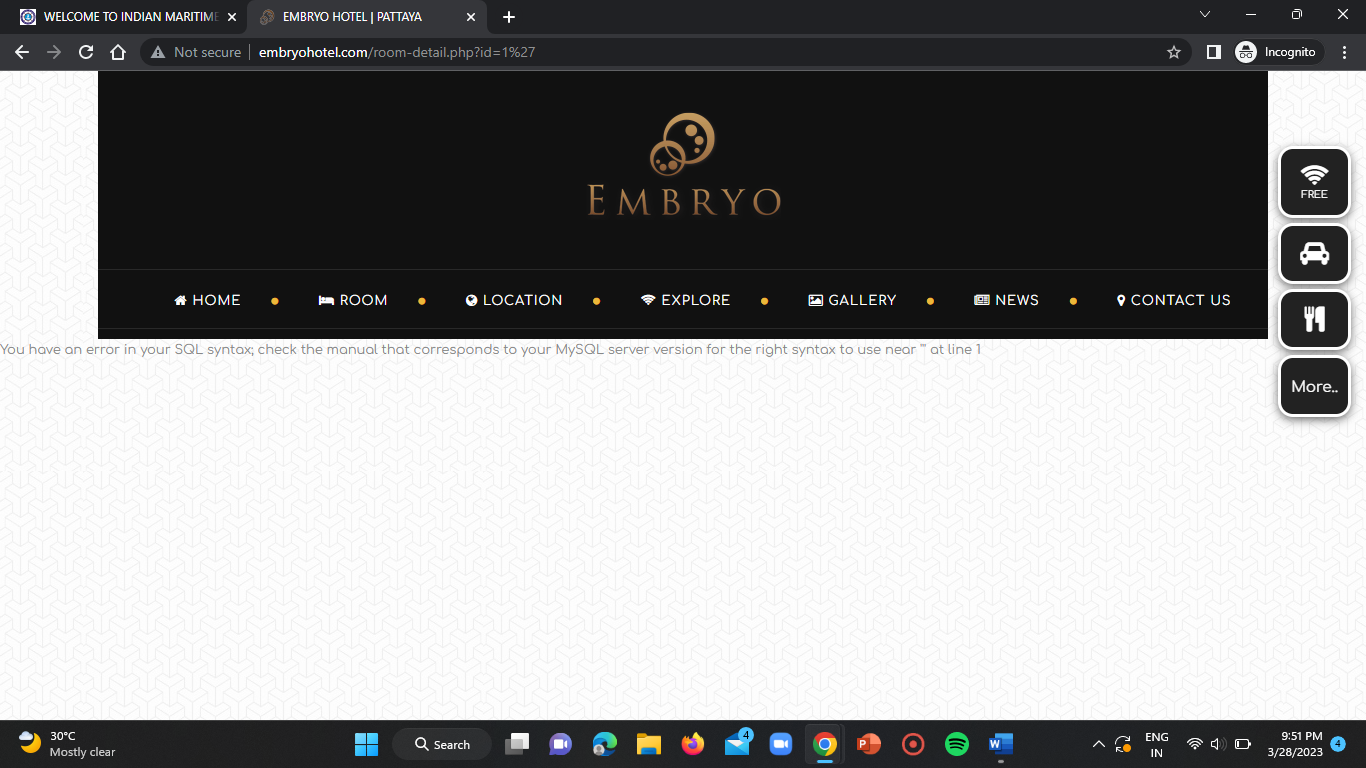
After visiting the manipulated URL, there was no error regarding database. So this website doesn’t have error based SQL injection vulnerability.

**STEP-5:**

I’ve tried the above steps for many websites and found the bellow mentioned websites with such vulnerability.

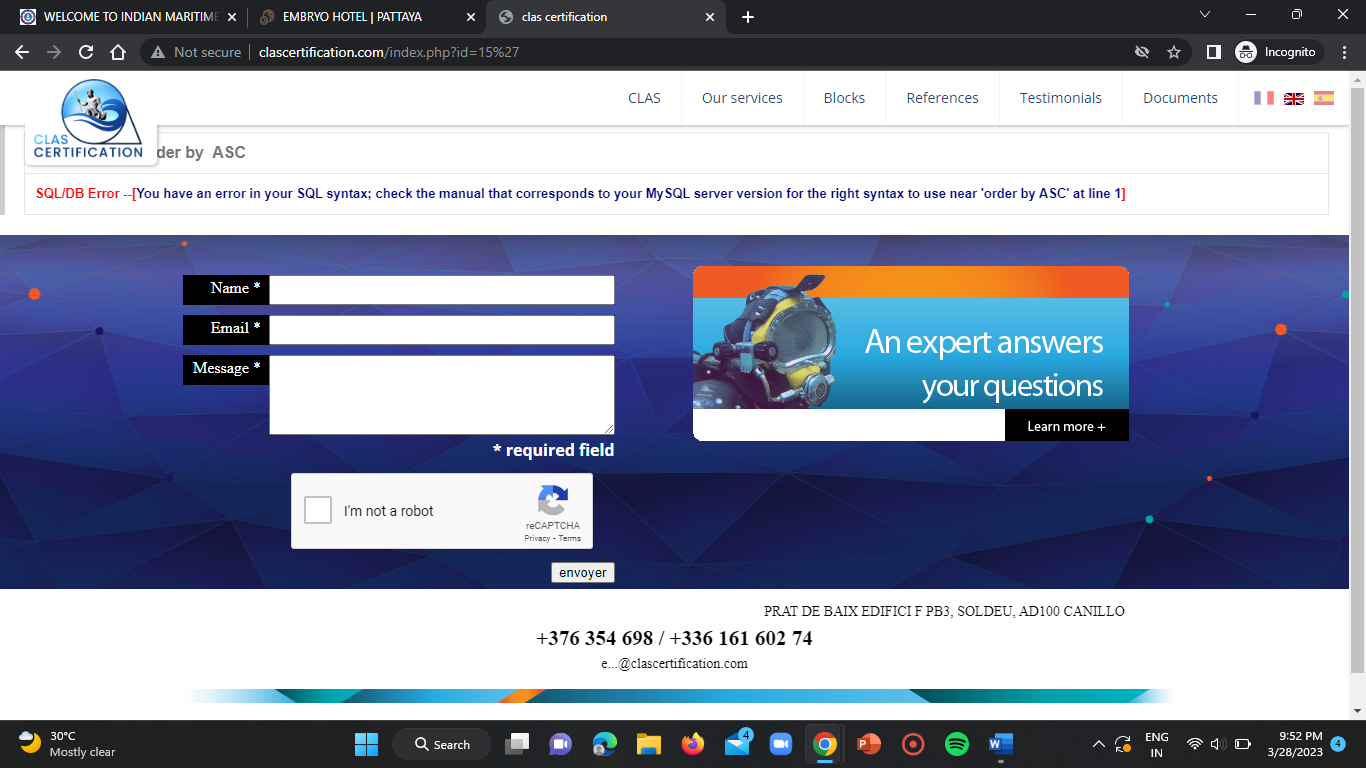
WEBSITE-1:

<http://www.embryohotel.com/room-detail.php?id=1%27>



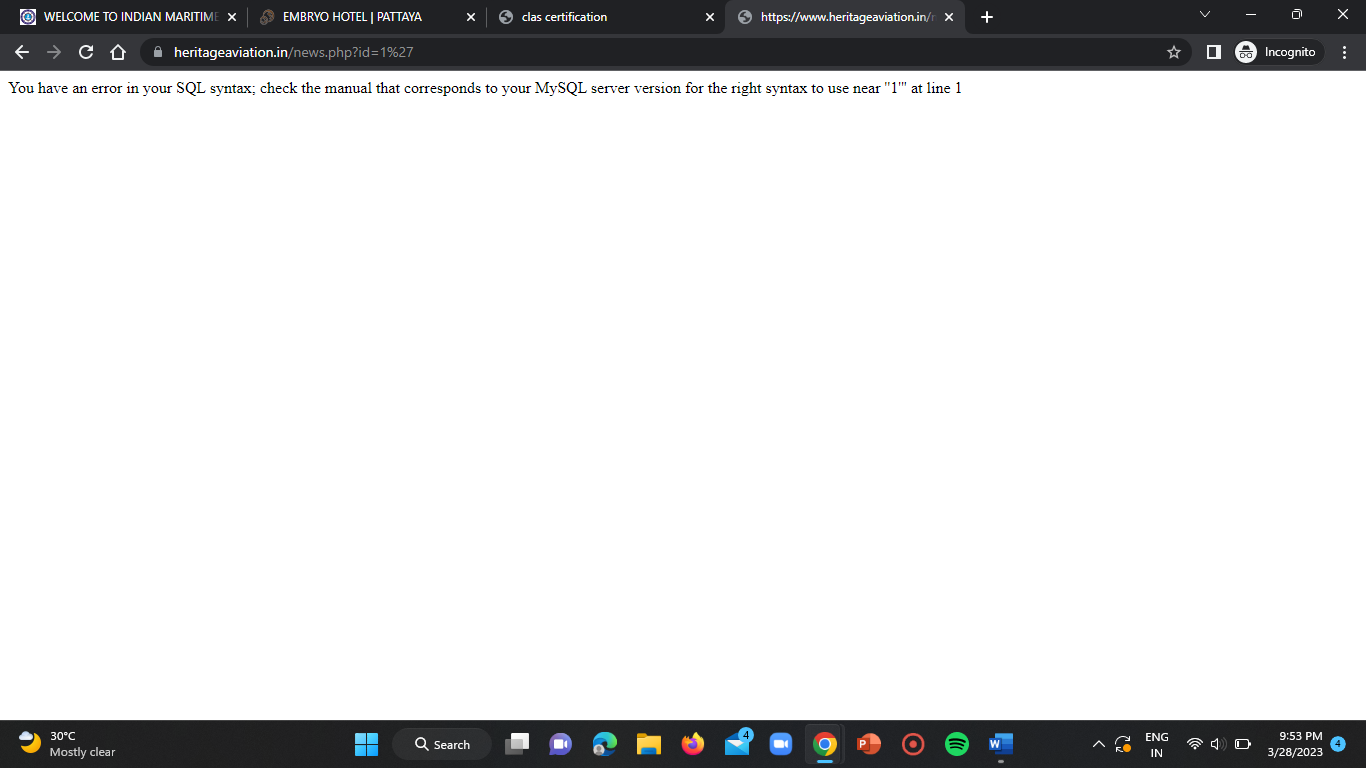
WEBSITE-2:

<http://www.clascertification.com/index.php?id=15%27>



WEBSITE-3:

<https://www.heritageaviation.in/news.php?id=1%27>



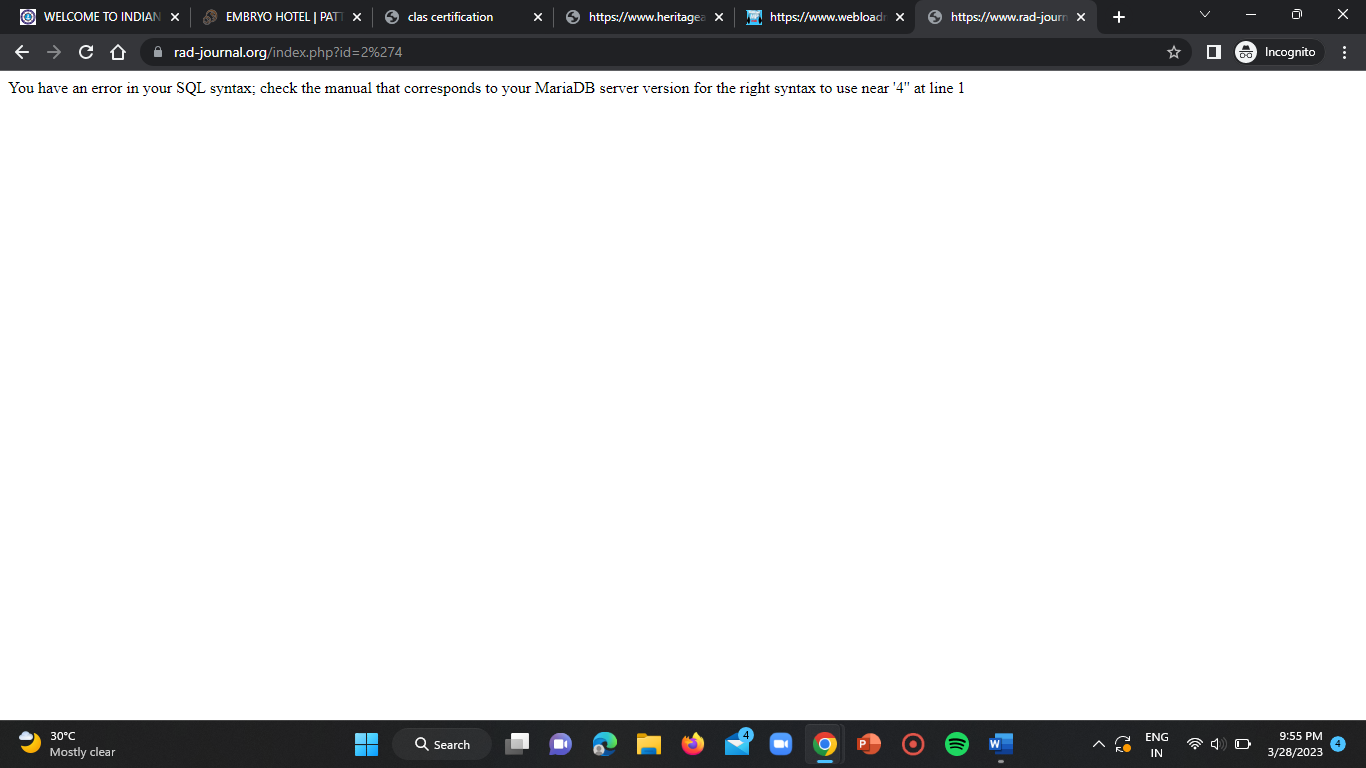
WEBSITE-4:

<https://www.webloadmpstore.com/product.php?id=3%27>



WEBSITE-5:

<https://www.rad-journal.org/index.php?id=2%274>



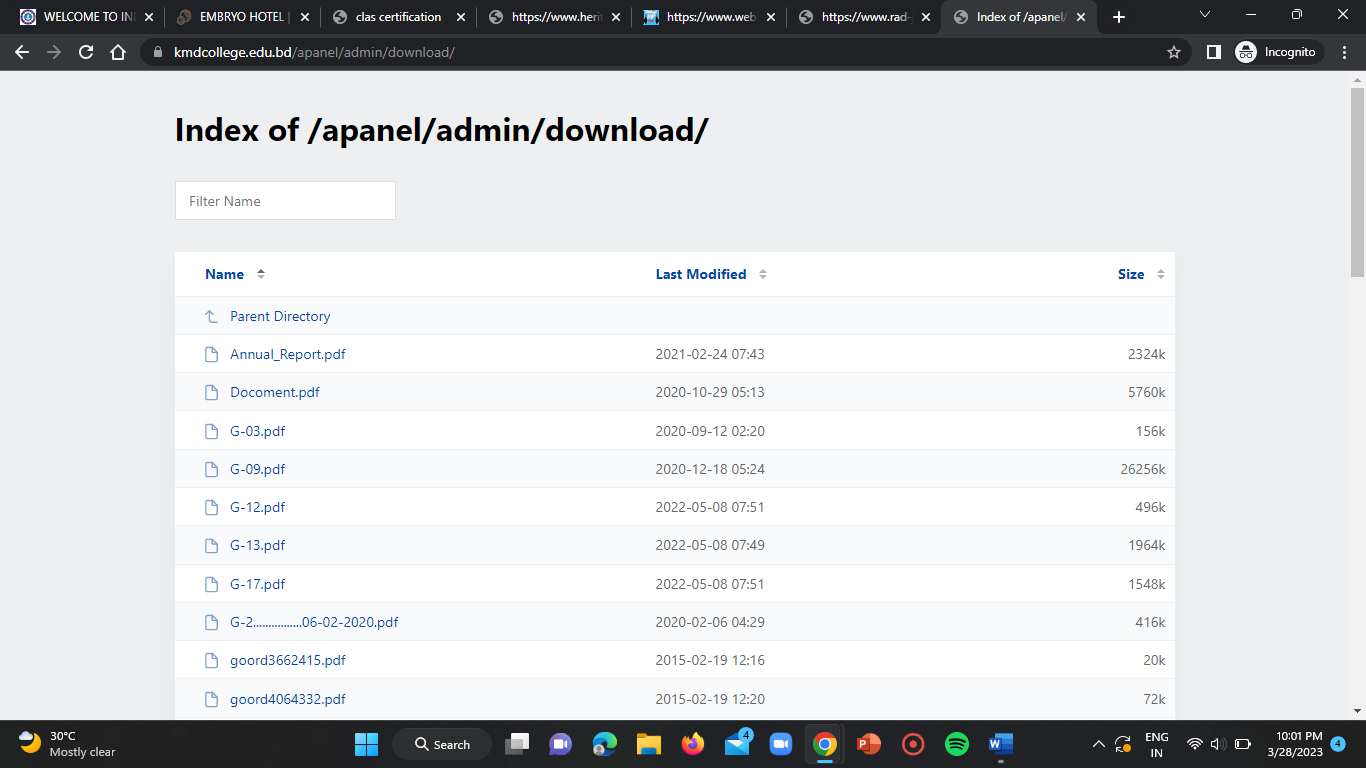
**LOGIN BYPASS USING SQL INJECTION:**

* Here we are going to find out the websites which have the “login bypass using SQL injection” vulnerability.

WEBSITE-1:

**STEP-1:**

While I was finding for broken access control vulnerability, luckily I found a website that has login bypass vulnerability.



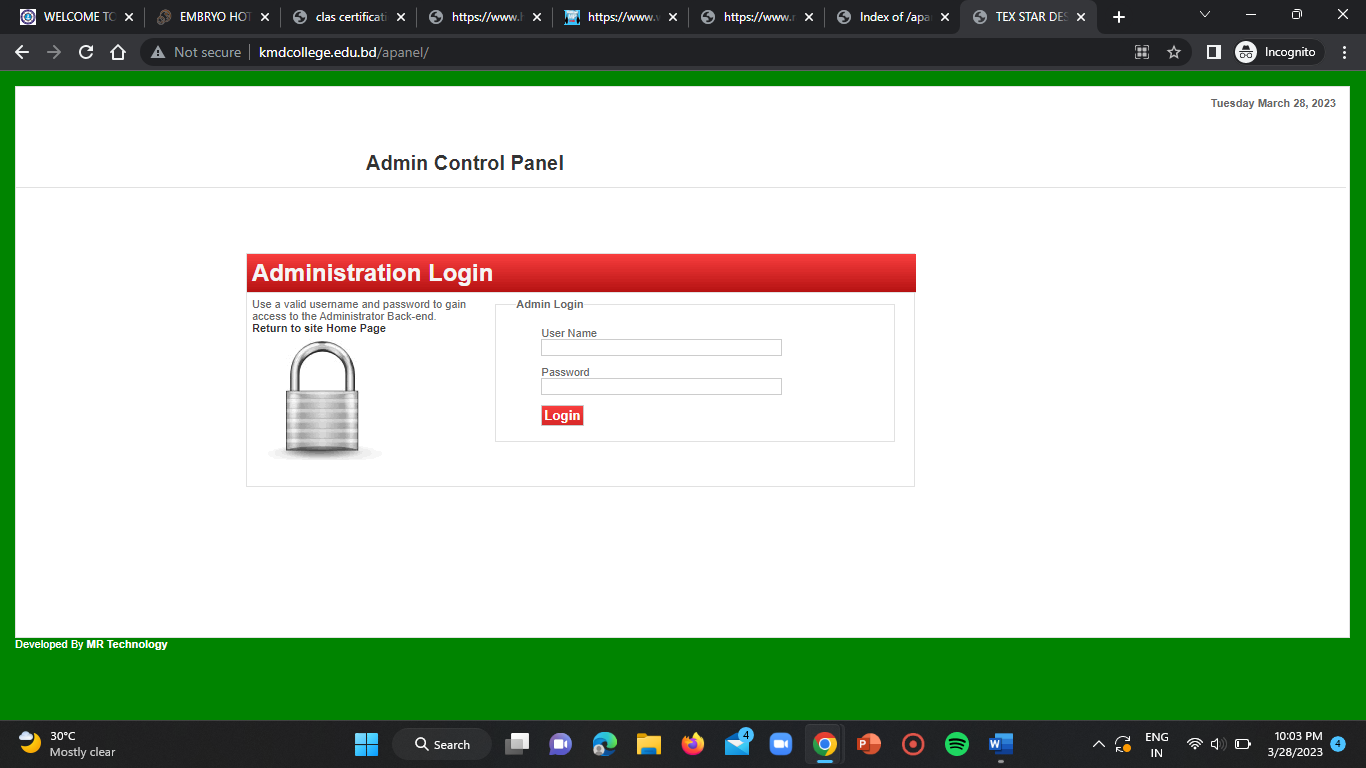
I just came two directories back i.e, from

<https://kmdcollege.edu.bd/apanel/admin/download/>

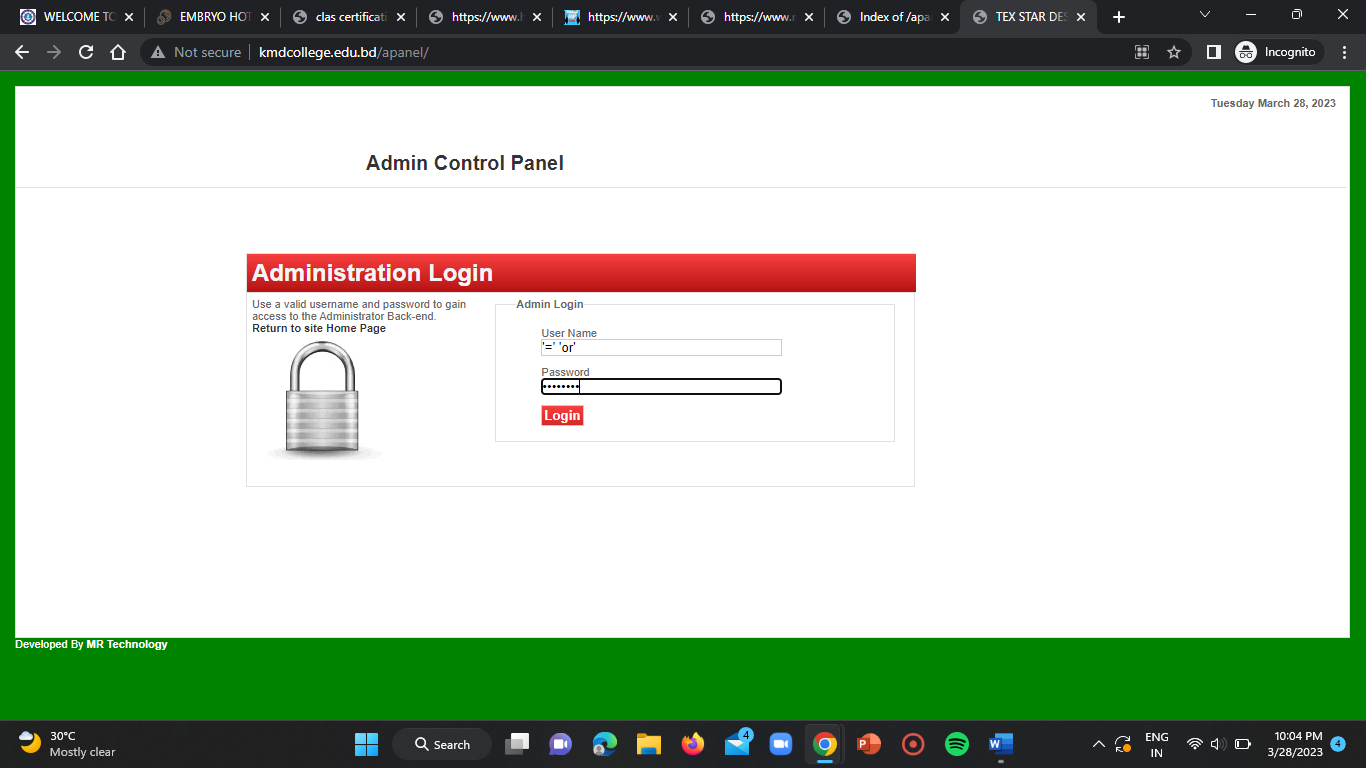
to,

<http://kmdcollege.edu.bd/apanel/>

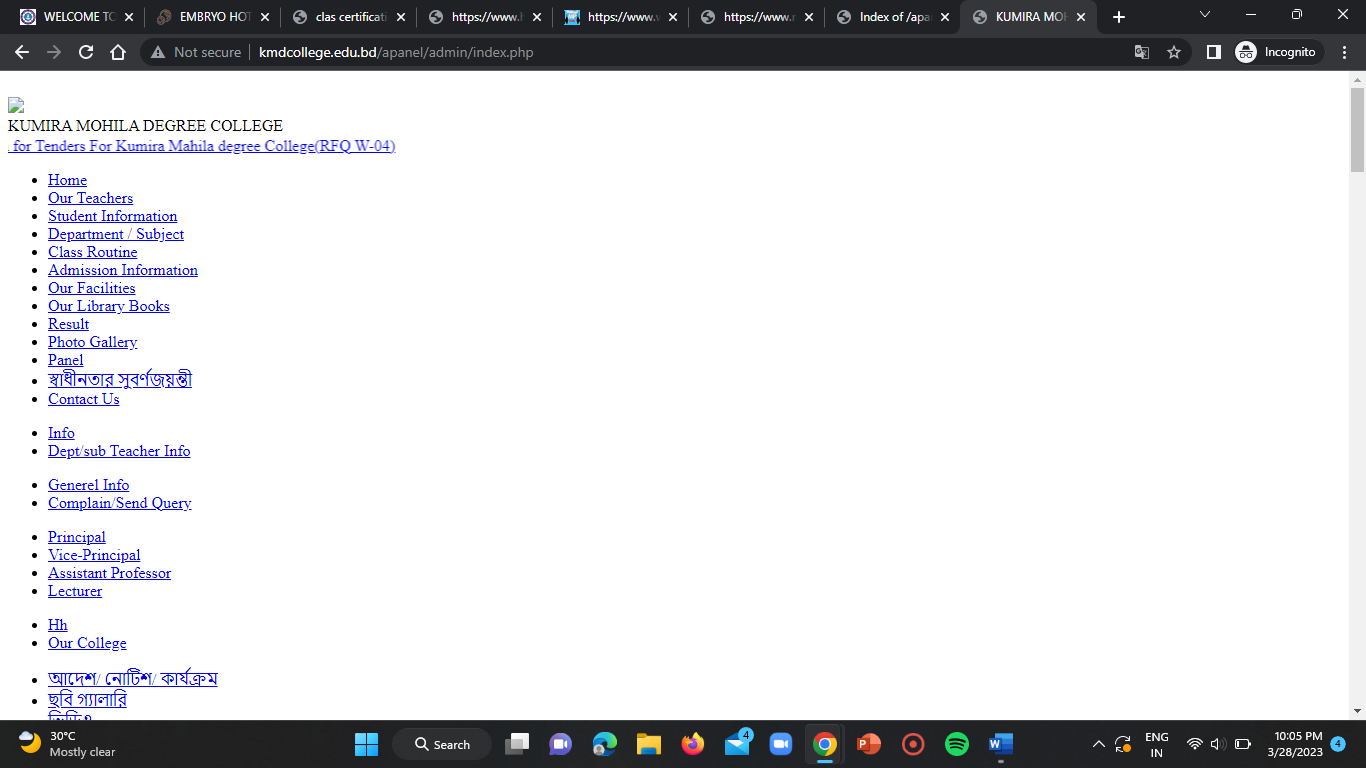
then the webpage asked me for admin login details.



So, I entered ‘=’ ‘or’ in both the username and password input fields and clicked on login.



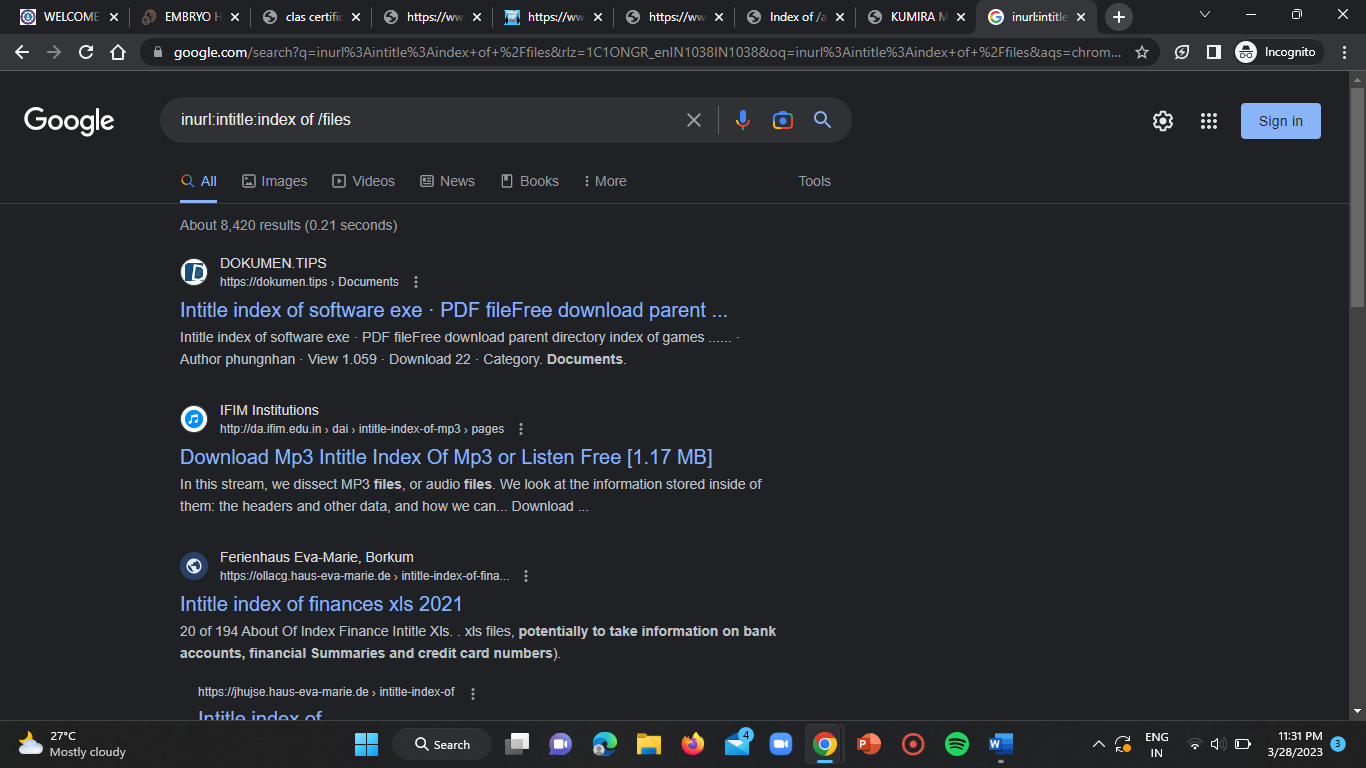
The login attempt made by me was successful and it got navigated to the admin portal’s index page.



**BROKEN ACCESS CONTROL:**

**STEP-1:**

In this case I’ve used the google dork “inurl:intitle:index of /files” to find the URLs of webserver file system.



Here, I got redirected by listing many number of websites.

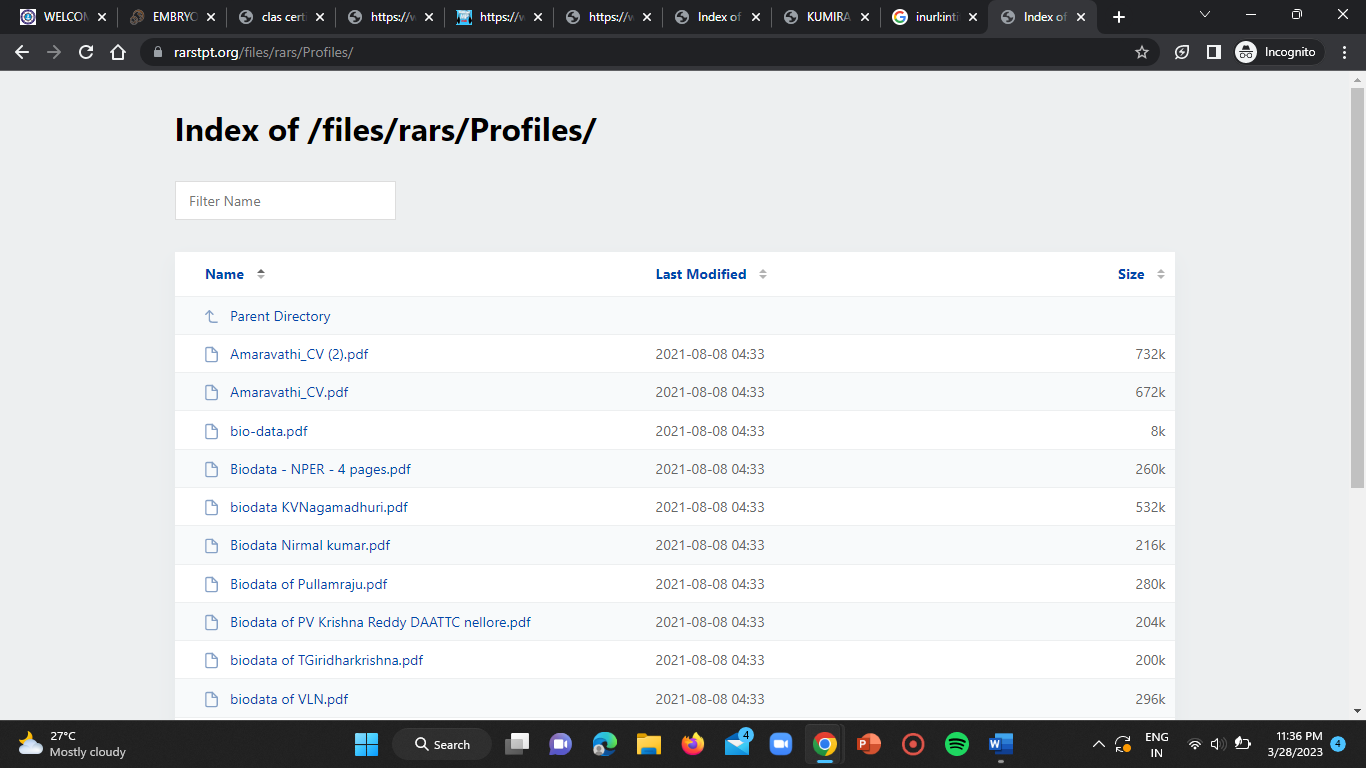
**STEP-2:**

Now, I’ve visited each and every website to find out files that are confidential, critical and that are not to be available publicly.

So, I found the below websites with broken access control vulnerability.

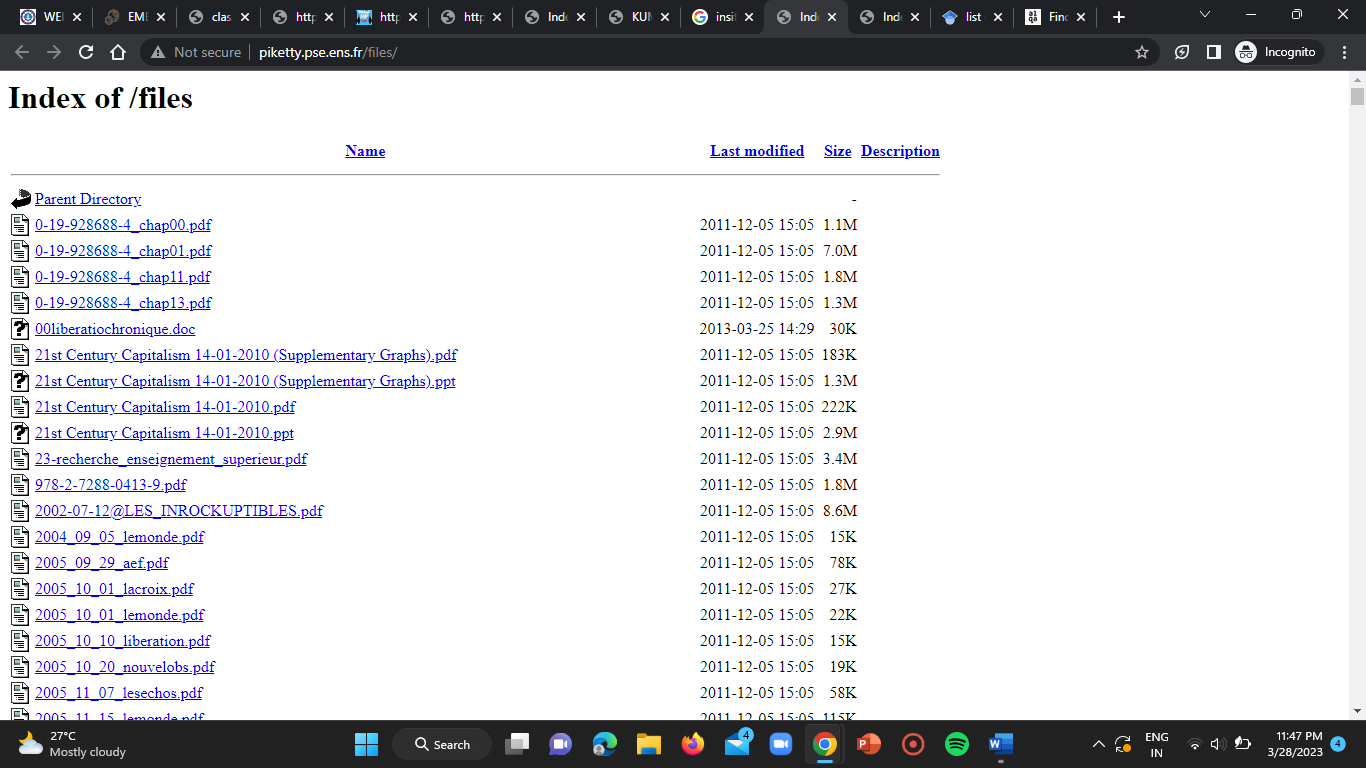
WEBSITE-1:

<https://rarstpt.org/files/rars/Profiles/>



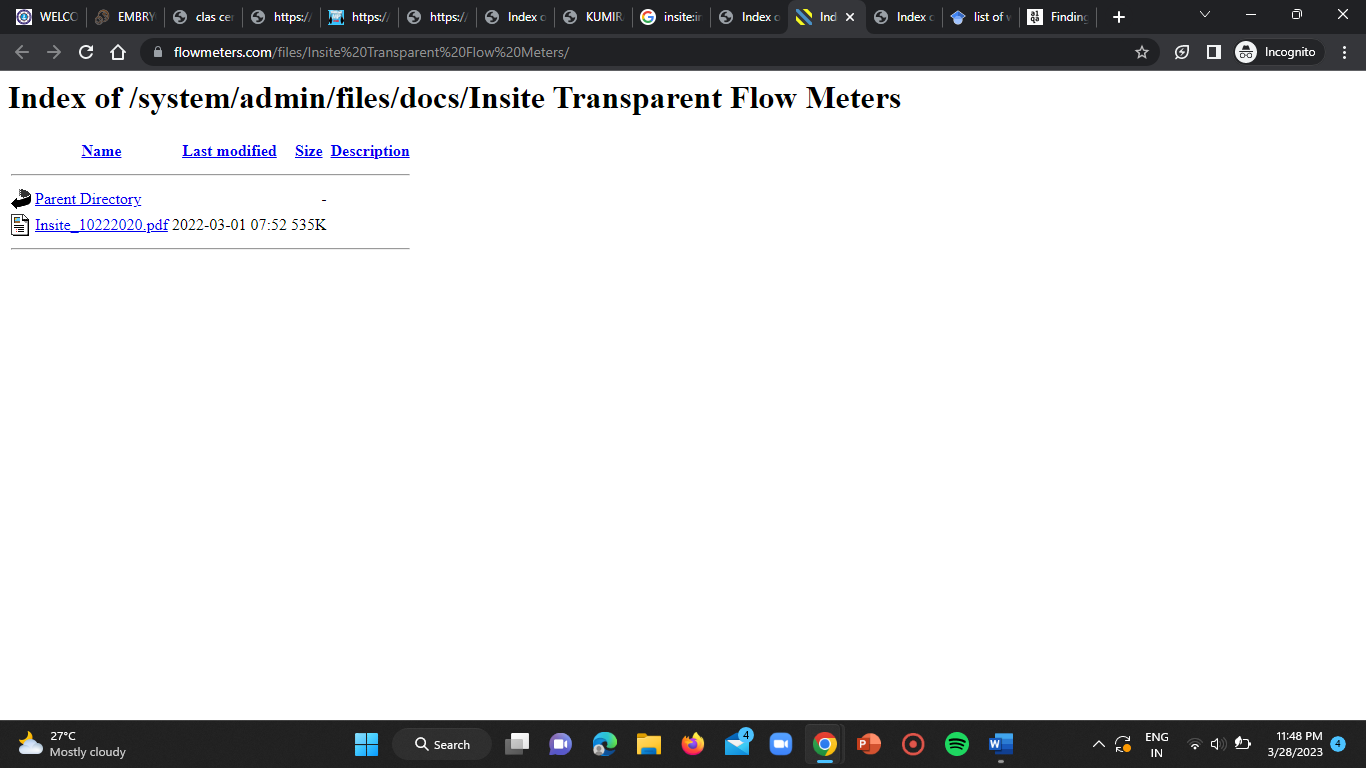
WEBSITE-2:

<http://piketty.pse.ens.fr/files/>



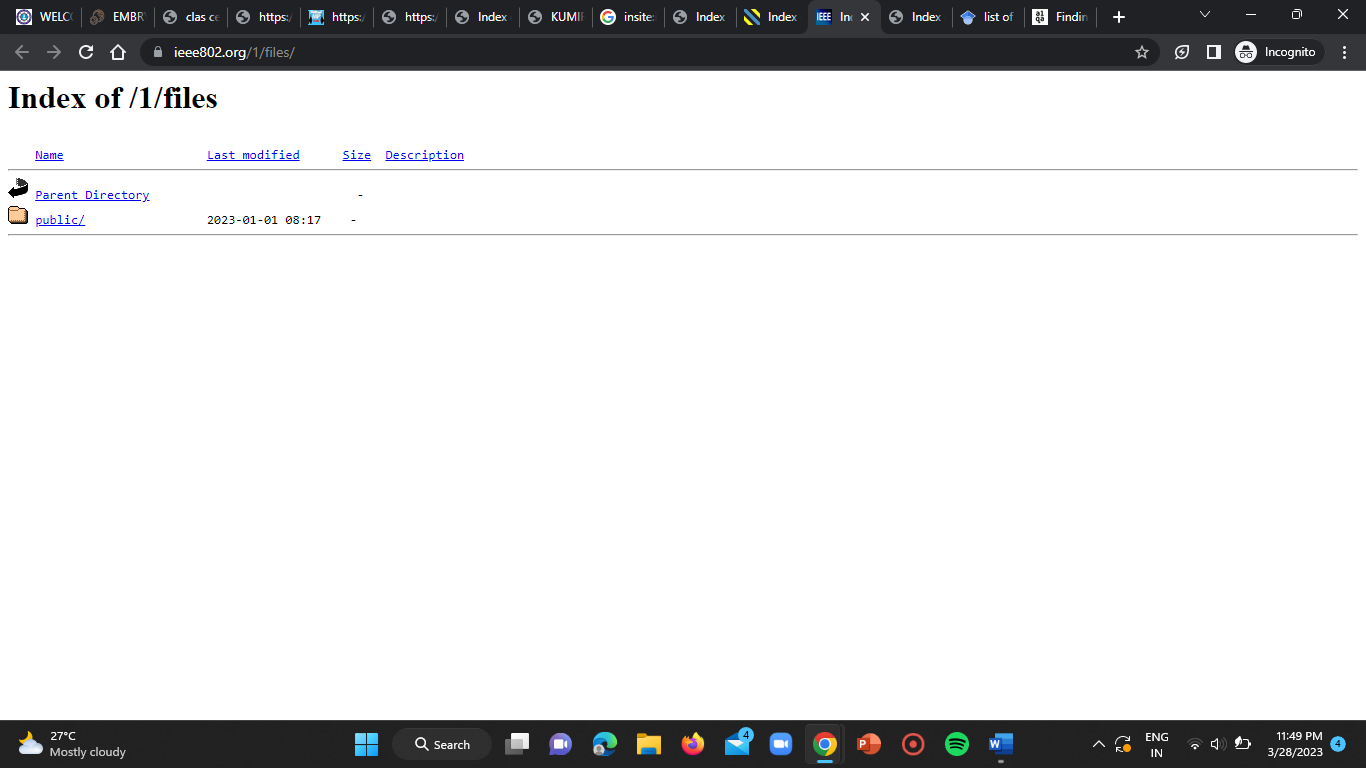
WEBSITE-3:

<https://www.flowmeters.com/files/Insite%20Transparent%20Flow%20Meters/>



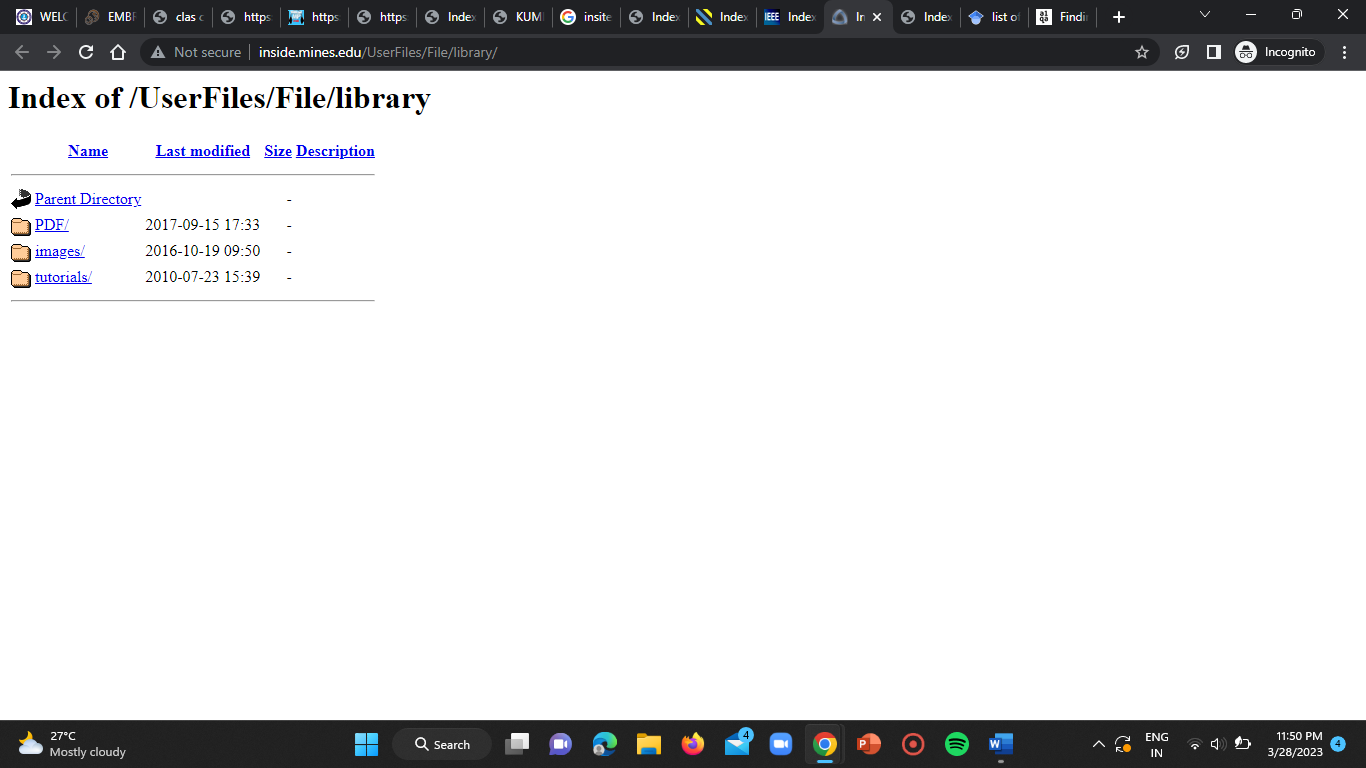
WEBSITE-4:

<https://www.ieee802.org/1/files/>



WEBSITE-5:

<http://inside.mines.edu/UserFiles/File/library/>



**CONCLUSION:**

By this task I’ve learn’t about the SQL injection based vulnerabilities and also got practised of using google dorks inorder to find the vulnerable websites.