**LAB EXPERIMENT – 6**

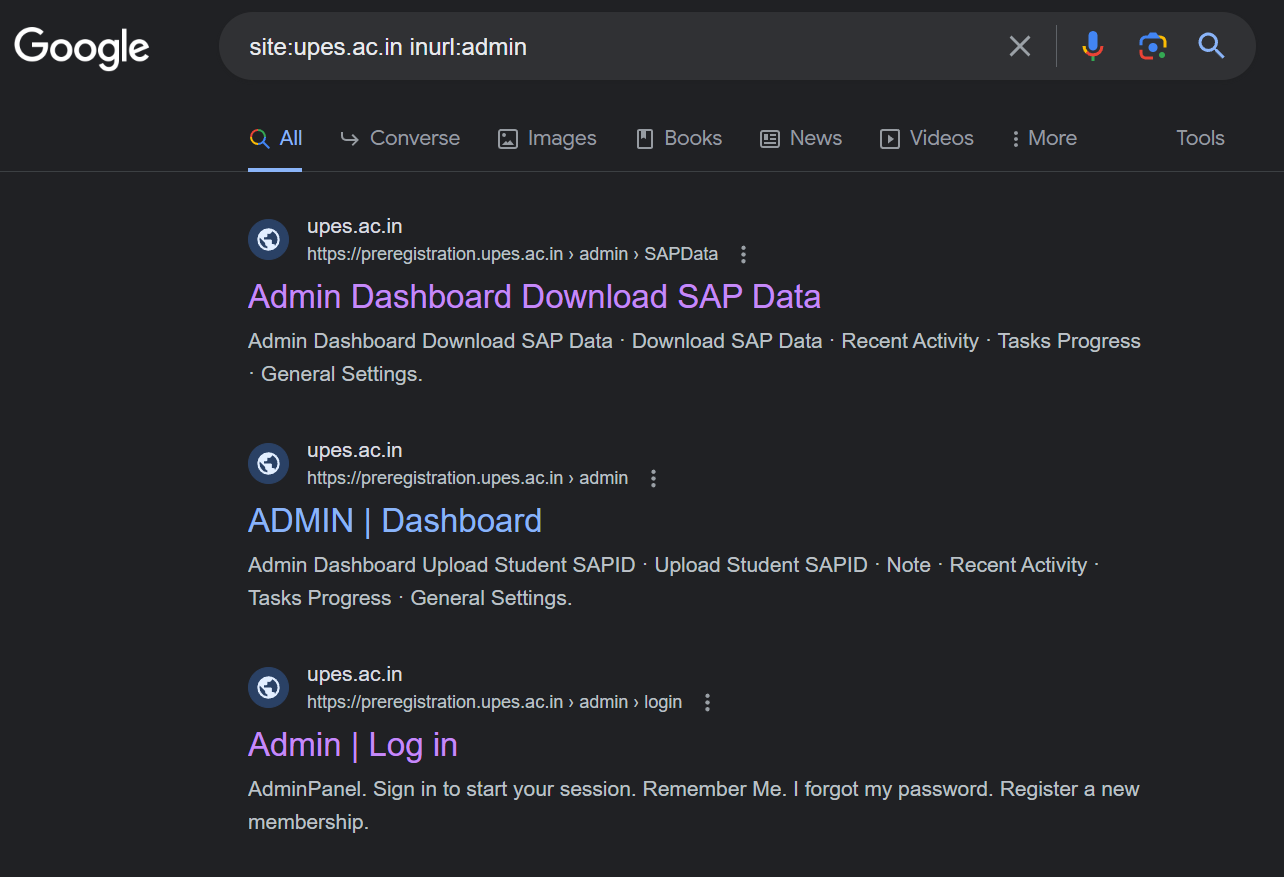
**Hack using Search Engines**

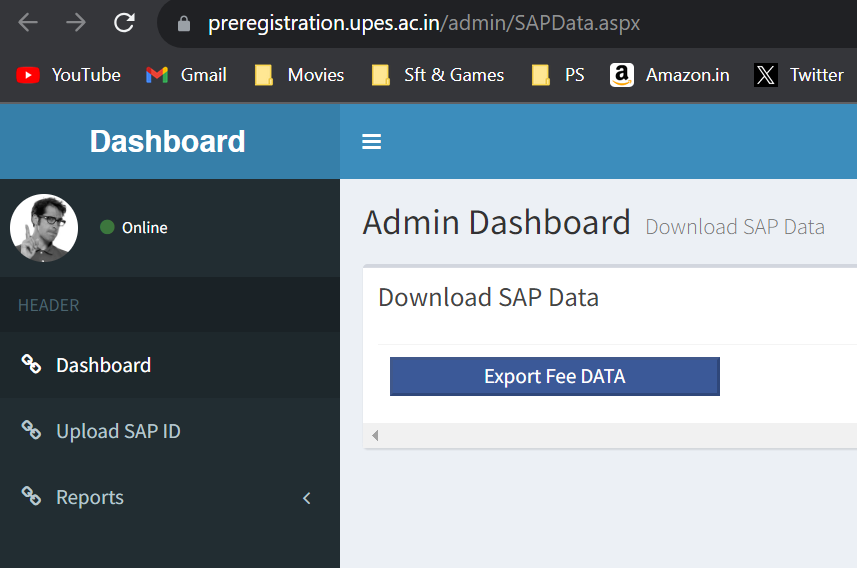
Akshat site:upesac.in intitle:

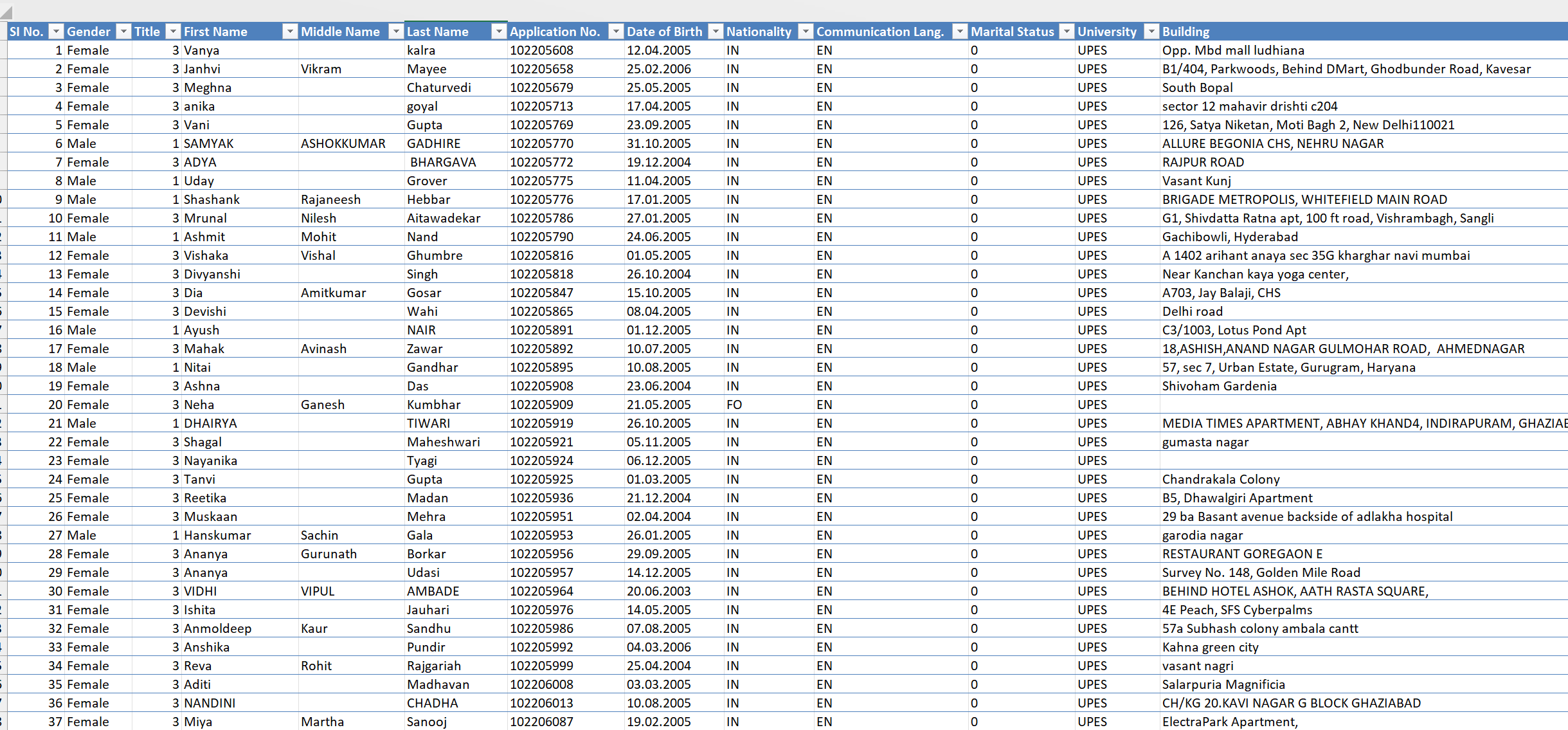
Intitle

Intext

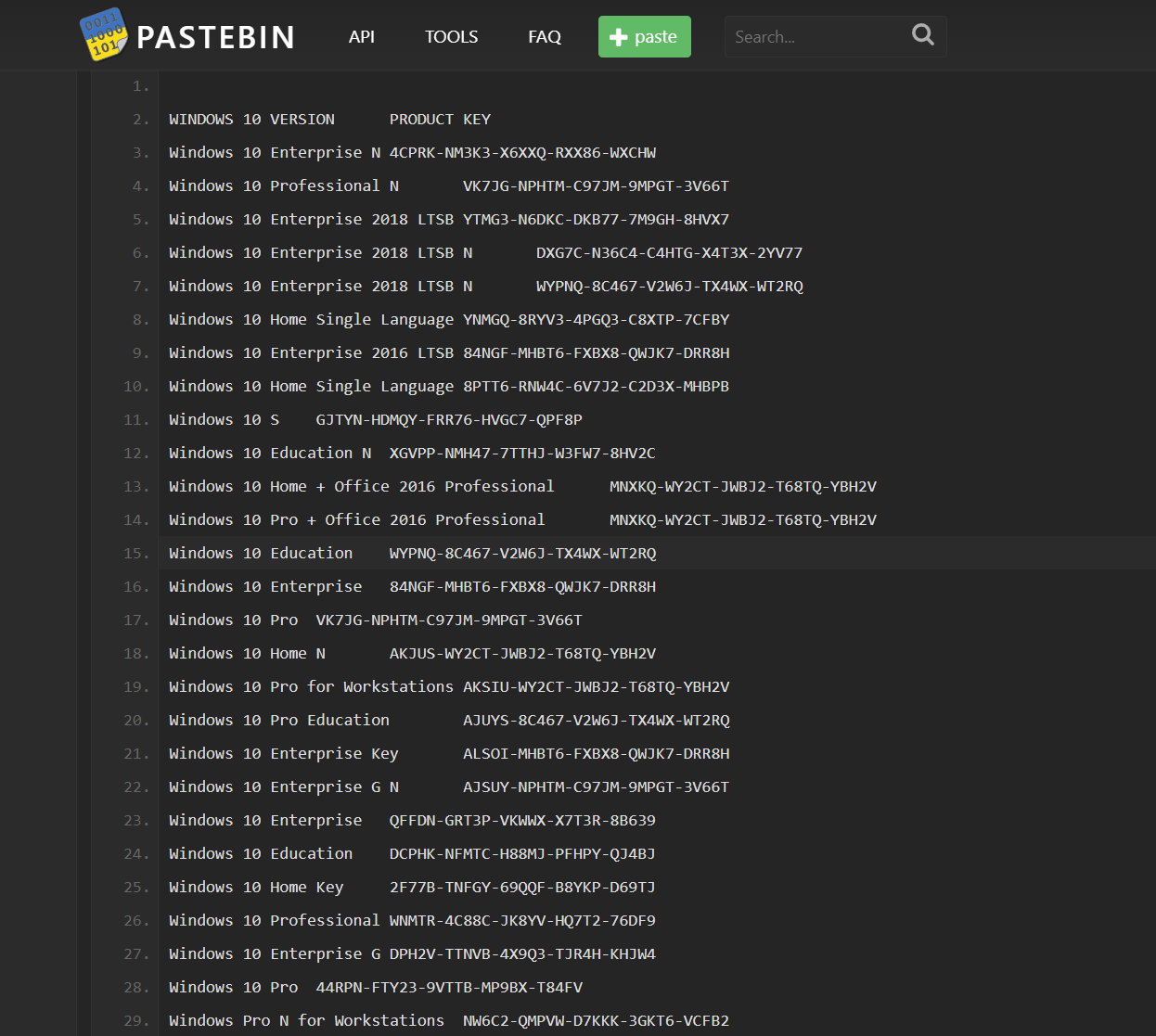
Inurl



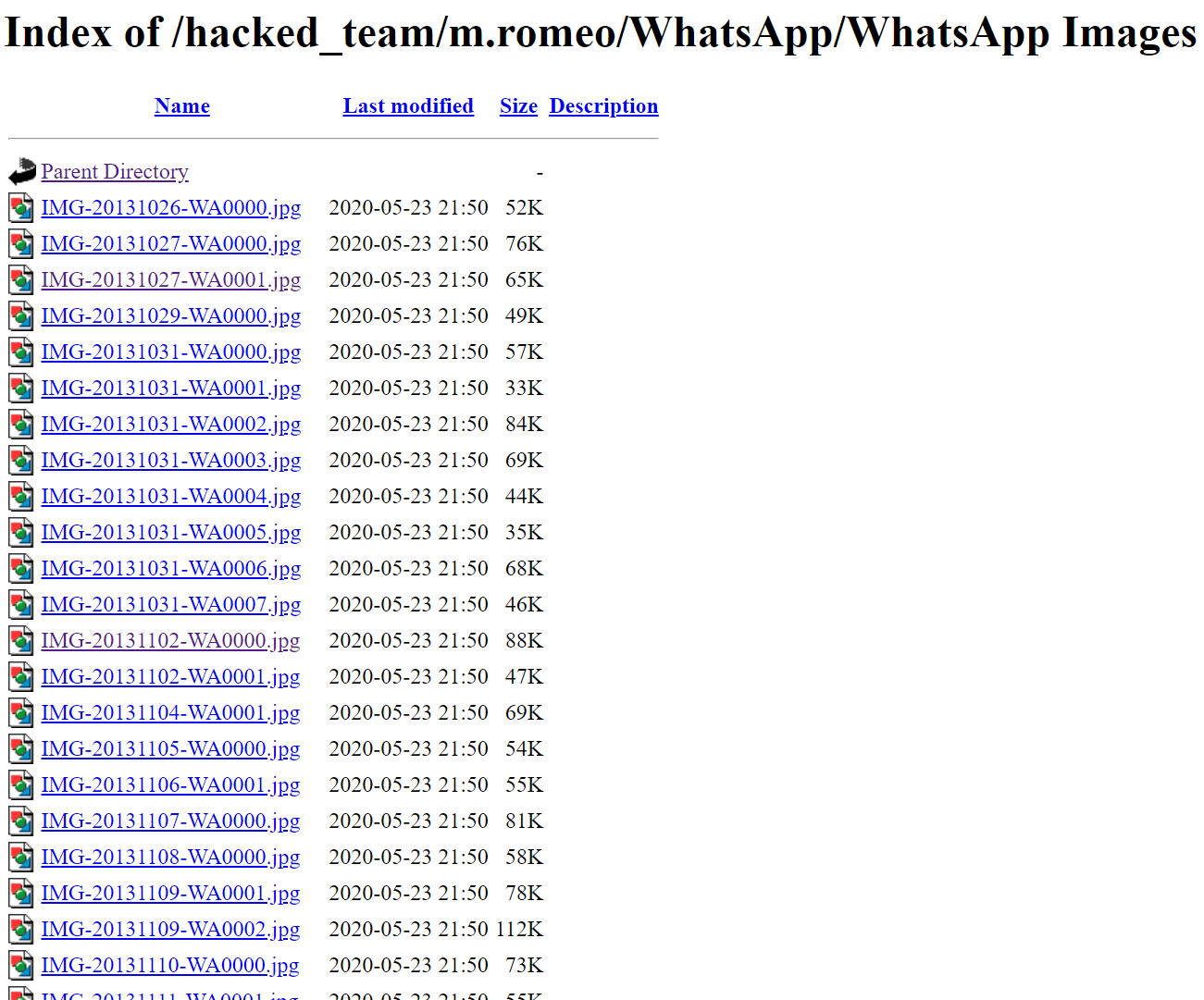




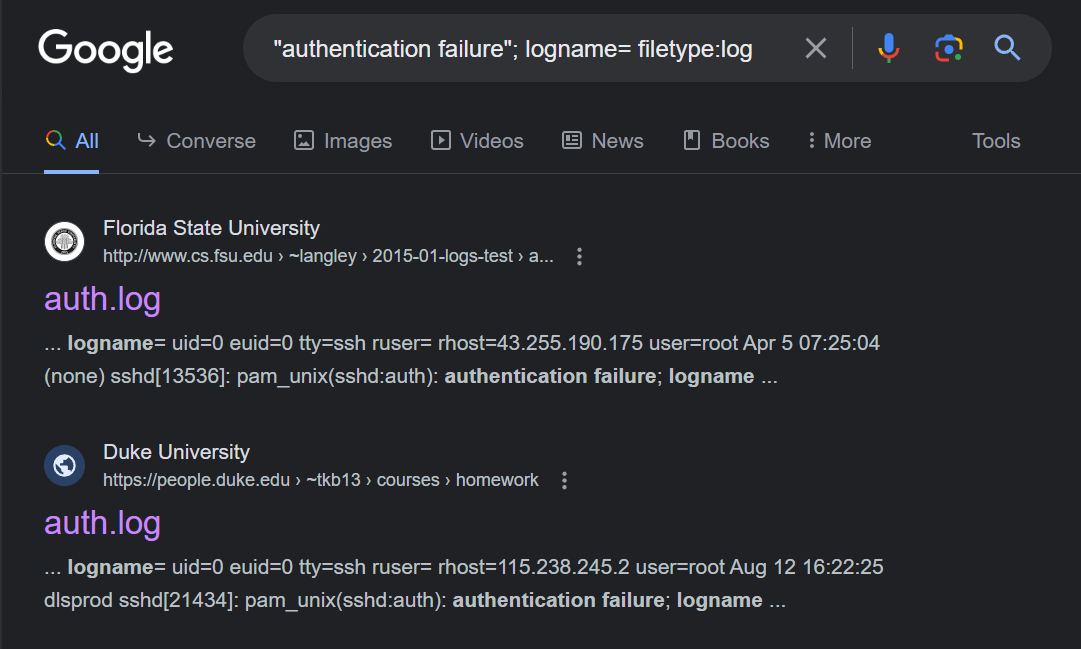
[inurl:pastebin "Windows 10 Product Keys\*"](https://www.exploit-db.com/ghdb/7754)

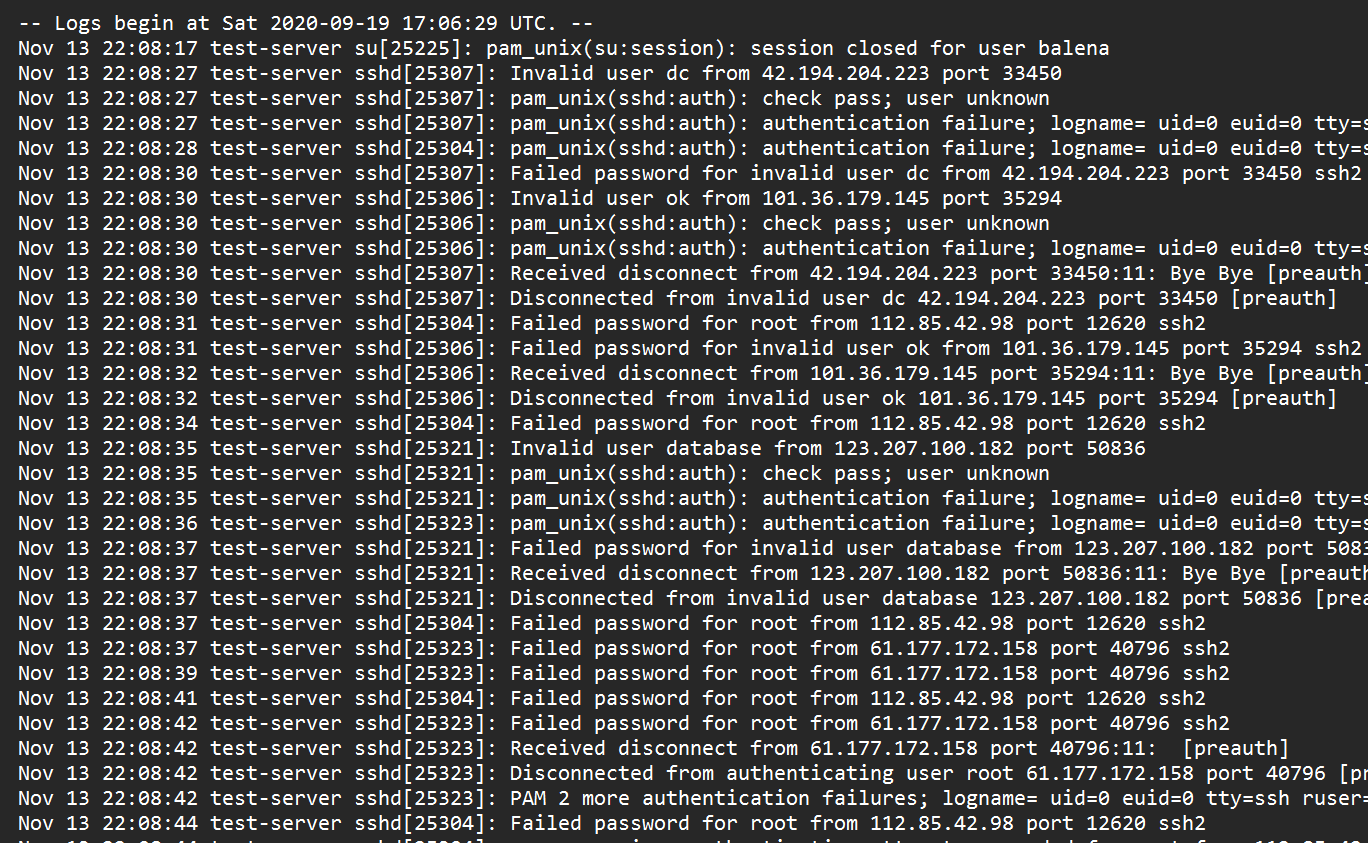


intitle:"Index of" "WhatsApp Images"

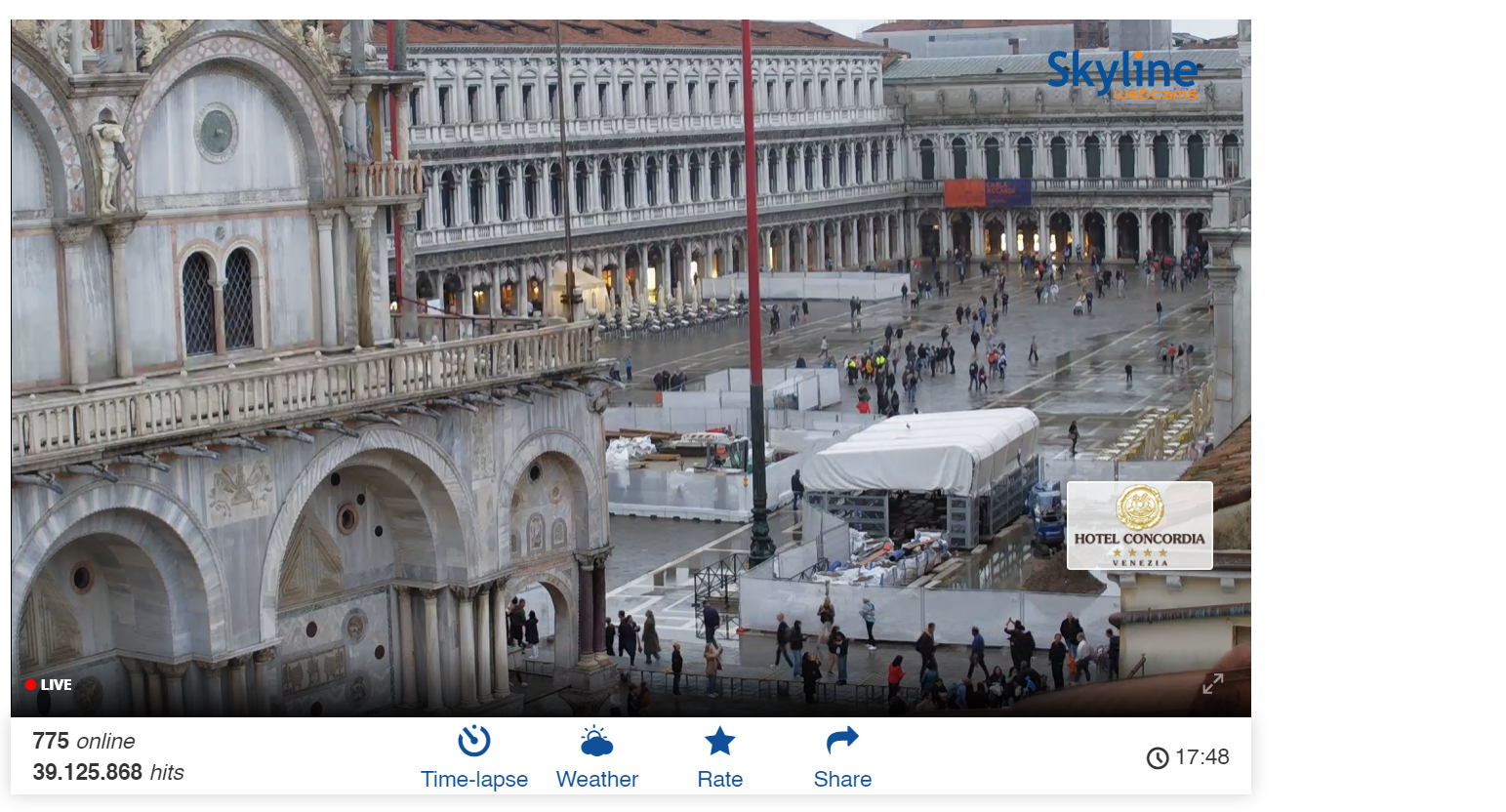


"authentication failure"; logname= filetype:log

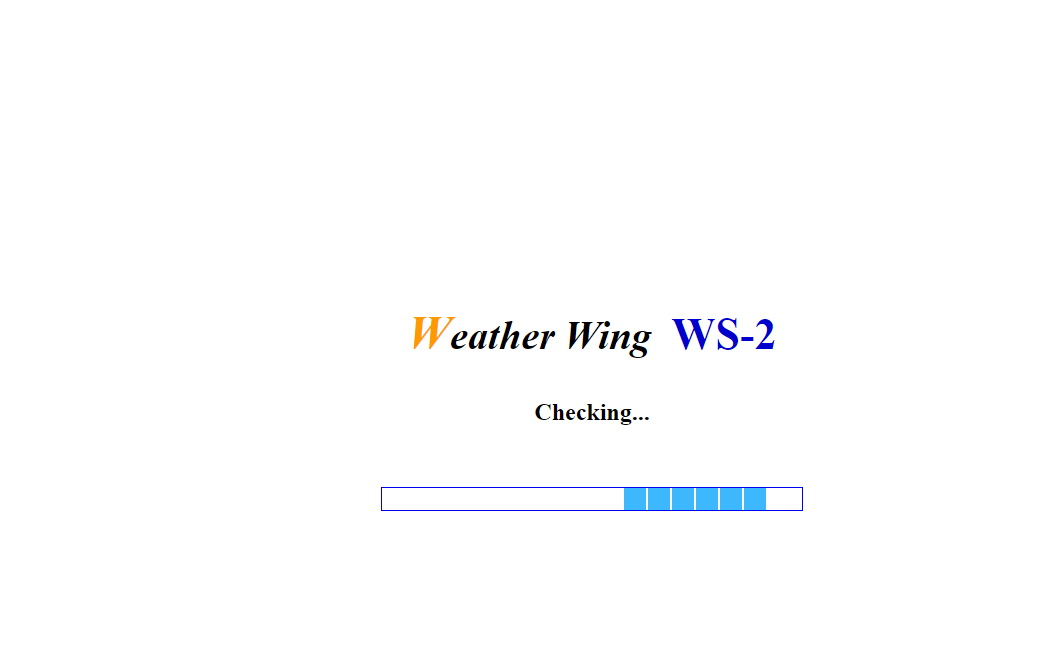




<https://www.skylinewebcams.com/en/webcam/italia/veneto/venezia/piazza-san-marco.html>



intitle:"Weather Wing WS-2"





**Ways to Prevent Yourself from Google Dork Method**

1. Securely configure web applications:

Ensure that your web applications are properly secured, following industry best practices. This includes implementing strong authentication mechanisms, input validation, and access controls to prevent unauthorized access or leakage of sensitive information.

2. Limit search engine indexing:

Use robots.txt or meta tags to instruct search engines not to index certain parts of your website or specific files. This can help prevent sensitive information from being exposed through search engine results.

3. Implement access controls:

Control access to sensitive information by implementing proper authorization mechanisms. Ensure that only authorized users can access sensitive data or perform sensitive actions.

4. Regularly review and update security configurations:

Regularly review and update security configurations for your web applications, content management systems (CMS), and any other platforms you use. This includes keeping software up to date, applying security patches, and following vendor recommendations.

5. Avoid storing sensitive information in plain text:

Do not store sensitive information, such as passwords, API keys, or database credentials, in plain text within your web applications or configuration files. Utilize encryption and secure storage mechanisms to protect sensitive data.

6. Use secure coding practices:

Adhere to secure coding practices, such as input validation, output encoding, and proper handling of user-supplied data. These practices can help prevent common vulnerabilities like SQL injection, cross-site scripting (XSS), and command injection.

7. Implement web application firewalls (WAF):

Consider implementing a WAF to provide an additional layer of protection against attacks targeting your web applications. WAFs can help detect and block suspicious or malicious requests, including those generated by Google dorks.

8. Regularly monitor and audit your web applications:

Implement monitoring and auditing mechanisms to detect and respond to potential security incidents or unauthorized access attempts. This can include log monitoring, intrusion detection systems (IDS), or security information and event management (SIEM) solutions.

9. Educate users and developers:

Provide security awareness training to users and developers to help them understand the risks associated with Google dorks and the importance of following secure coding practices. Encourage them to report any suspicious activities or potential vulnerabilities they encounter.

10. Stay informed about security vulnerabilities:

Stay up to date with security news and vulnerabilities related to the software and platforms you use. Subscribe to security mailing lists, follow security blogs, and apply security patches promptly to mitigate known vulnerabilities.