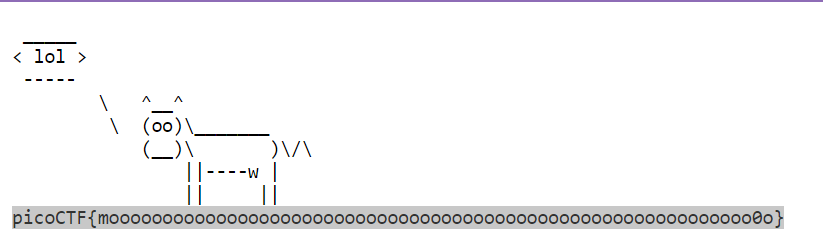
Keygenme  
First the code asks for the user’s license key and stores it in the user\_key variable, it goes through the check key function with another variable called bUsername\_trial whose value is “FRASER”. If the function returns tru and it then decrypts the key.

After that 4,5,3,6… and some other characters of the decrypted key are going through the sha256 function and the part of the flag can be determined by just those few letters, they have already provided the initial part of the flag in the code.

Caas-

Caas stands for cowsay as a service

We can type the command in the URL itself to get our required flag  


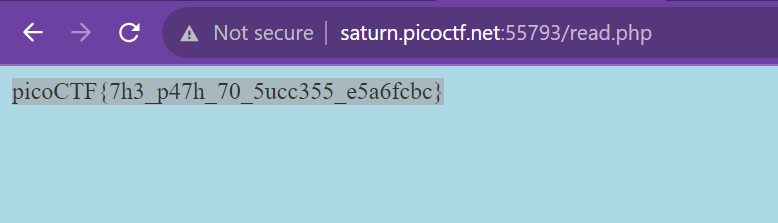
Flag - picoCTF{moooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo0o}

Forbidden paths-

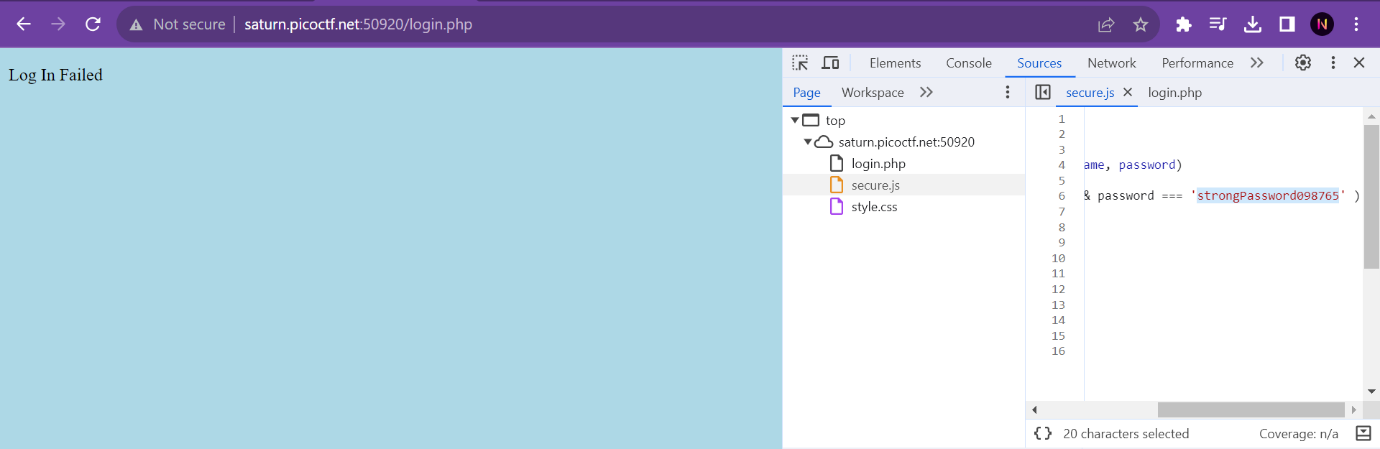
We have to move up the directories from our current one so we use /../../../../flag.txt, we’ve moved up 4 times as they have specified 4 different ones

../../../../flag.txt

picoCTF{7h3\_p47h\_70\_5ucc355\_e5a6fcbc}

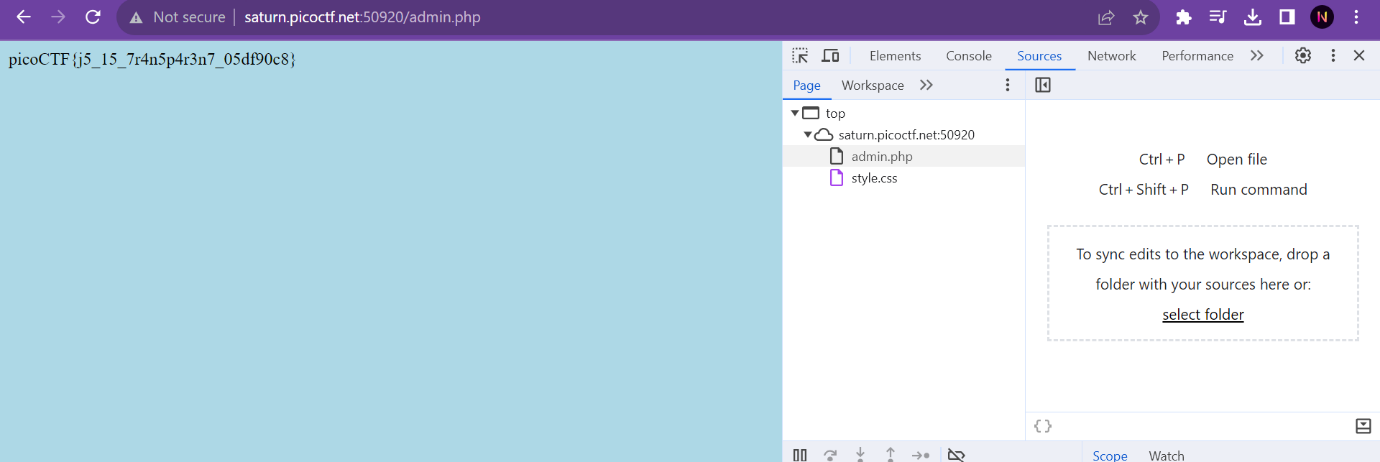


Local authority -  
when we look at the source code of the login page, nothing seems odd. But however when we check the source code of the login failed webpage, there is an extra file which contains our required information



Id – admin

Password - strongPassword098765

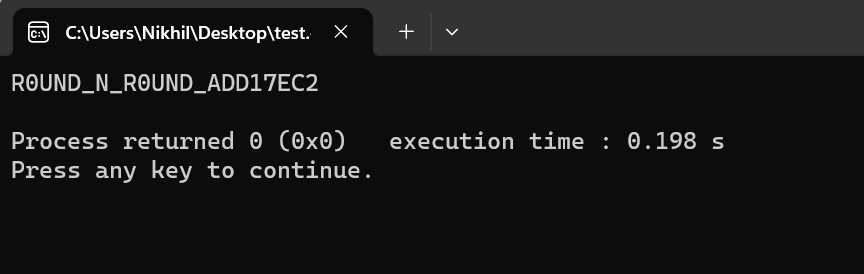


Flag - picoCTF{j5\_15\_7r4n5p4r3n7\_05df90c8}

The public key consists of the modulus **N** and the public exponent **e**. In RSA, encryption is performed using the formula c≡m^e(modN)*c*≡*me*(mod*N*), where *c* is the ciphertext, *m* is the plaintext message, *e* is the public exponent, and *N* is the modulus.

Mod 1-

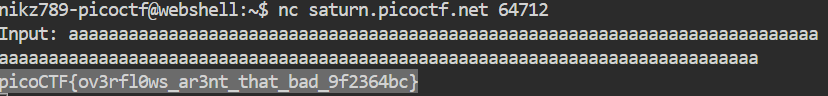
I asked ChatGPT to write me a program to perform everything



Overflow 1-

As the title was overflow, I thought of just spamming some letters and that actually worked

picoCTF{ov3rfl0ws\_ar3nt\_that\_bad\_9f2364bc}



ARMssembly 0-

The given code is written using assembly.

Asking GPT what the code does, the code returns the larger number in hexadecimal form

picoCTF{5EE79C2B}