



**VIT<sup>®</sup>**  
**Vellore Institute of Technology**  
(Deemed to be University under section 3 of UGC Act, 1956)

**CSE3501-Information Security Analysis and Audit**

**Topic:- Create a simple secure API and demonstrate it**

**Assignment-6**

**Name: Adrija Mukhopadhyay**

**Register Number: 19BDS0159**

**Submitted Date: 21.11.2021**

**Course Instructor: Prof. SARAVANAGURU RA. K**

**FULL CODE :-****API WITH AUTHENTICATION**

```

from flask import Flask, jsonify, request, make_response
#import jwt

import datetime
from functools import wraps
app = Flask(__name__)

app.config['SECRET_KEY'] = 'thisisthesecretkey'
def token_required(f):
    @wraps(f)
    def decorated(*args, **kwargs):
        token = request.args.get('token')
        if not token:
            return jsonify({'message': 'Token is missing!'}),403
        try:
            data = jwt.decode(token, app.config['SECRET_KEY'])
        except:
            return jsonify({'message' : 'Token is missing or invalid!'}),403
        return f(*args,**kwargs)

    return decorated

@app.route('/unprotected')
def unprotected():
    return jsonify({'message': 'Anyone can view this!'})

@app.route('/protected/<int:n>')
@token_required
def protected(n):
    return jsonify({'message' : 'This is only available for people with token ids.'})

if __name__=="__main__":
    app.run(debug=True)

@app.route('/login/')
def login():
    auth = request.authorization

    if auth and auth.password == 'password':
        token = jwt.encode({'user' : auth.username,'exp':
datetime.datetime.utcnow() + datetime.timedelta(minutes=30)},
app.config['SECRET_KEY'])
        return jsonify({'token': token.decode('UTF-8')})

```

```

    return make_response('Could not verify!', 401, {'WWW-Authenticate' : 'Basic
realm="Login Required"'})

@app.route("/")
def hello_world():
    return "<p>Hello, World!</p>"

@app.route("/armstrong/<int:n>")
def armstrong(n):
    sum = 0
    order = len(str(n))
    copy_n = n

    while (n > 0):
        digit = n % 10
        sum += digit ** order
        n = n // 10
    if(sum == copy_n):
        print(f"{copy_n}is an Armstrong number")
        result = {
            "Number" : copy_n,
            "Armstrong": True,
            "Server IP": "122.234.213.53 "
        }
    else:
        print(f"{copy_n}is not an Armstrong number")
        result = {
            "Number" : copy_n,
            "Armstrong": False,
            "Server IP": "122.234.213.53 "
        }
    return jsonify(result)

if __name__=="__main__":
    app.run(debug=True)

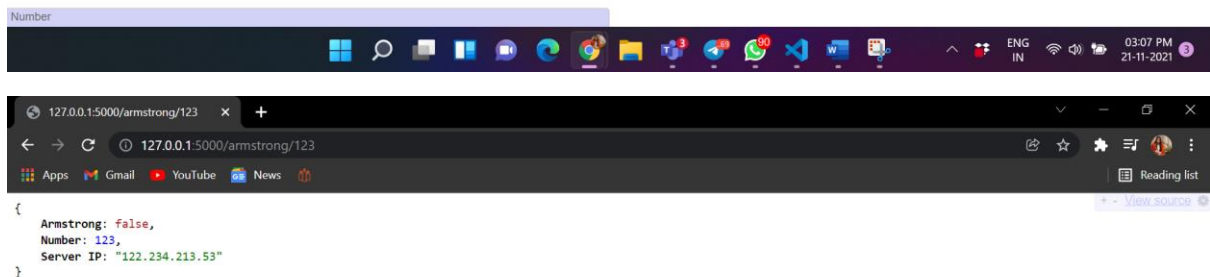
```

## Normal Api calling without authentication



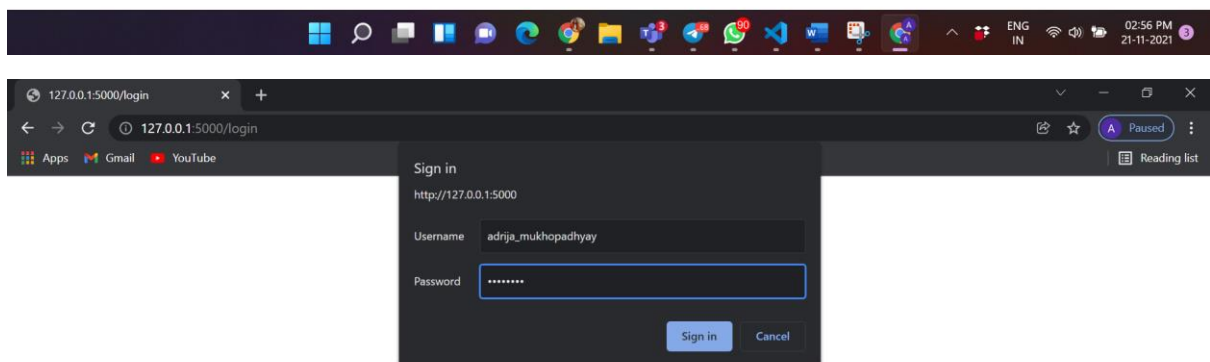
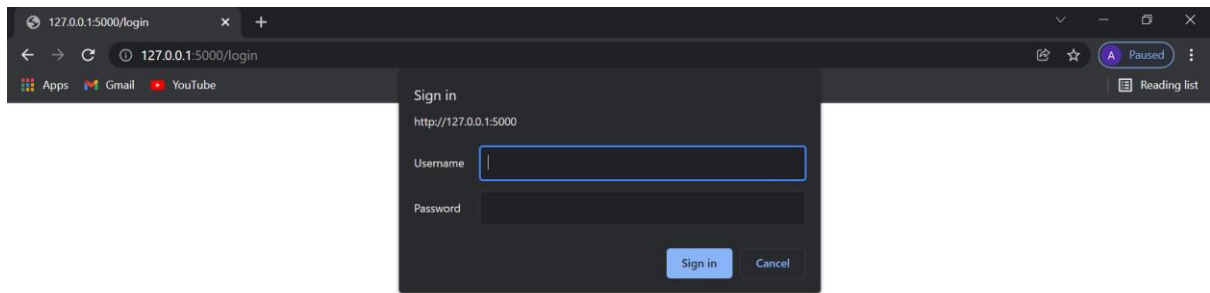
```
{
  Armstrong: true,
  Number: 153,
  Server IP: "122.234.213.53"
}
```

Number

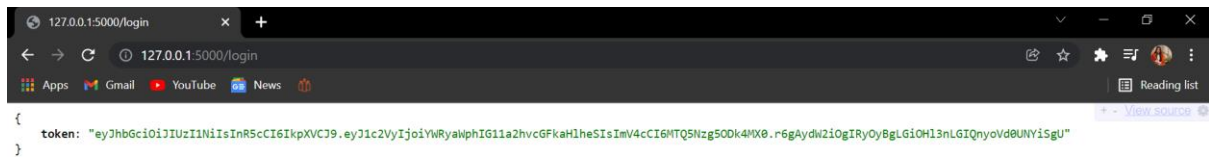


```
{
  Armstrong: false,
  Number: 123,
  Server IP: "122.234.213.53"
}
```

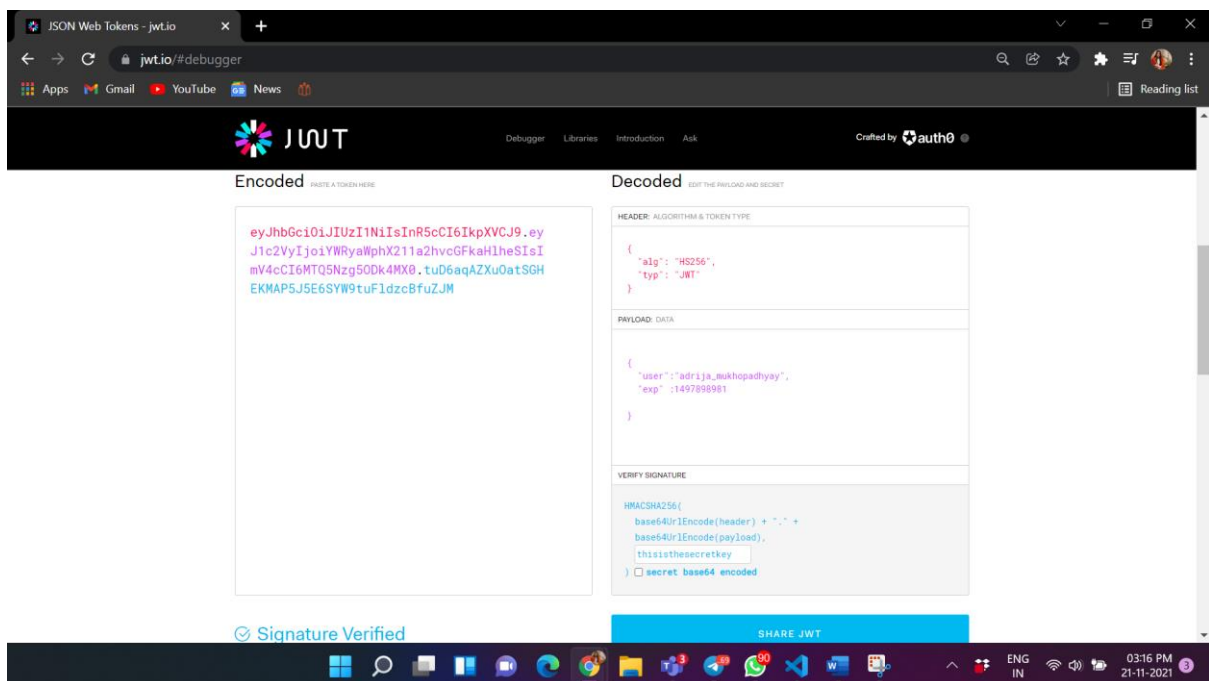
## After putting authentication



**This page gives the token for entering the protected area**



**Secret key for the token was: **thisistheseckey****

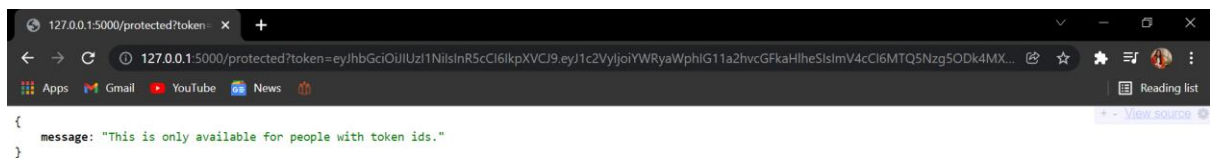


**By this we can even verify that our token is valid and working properly**

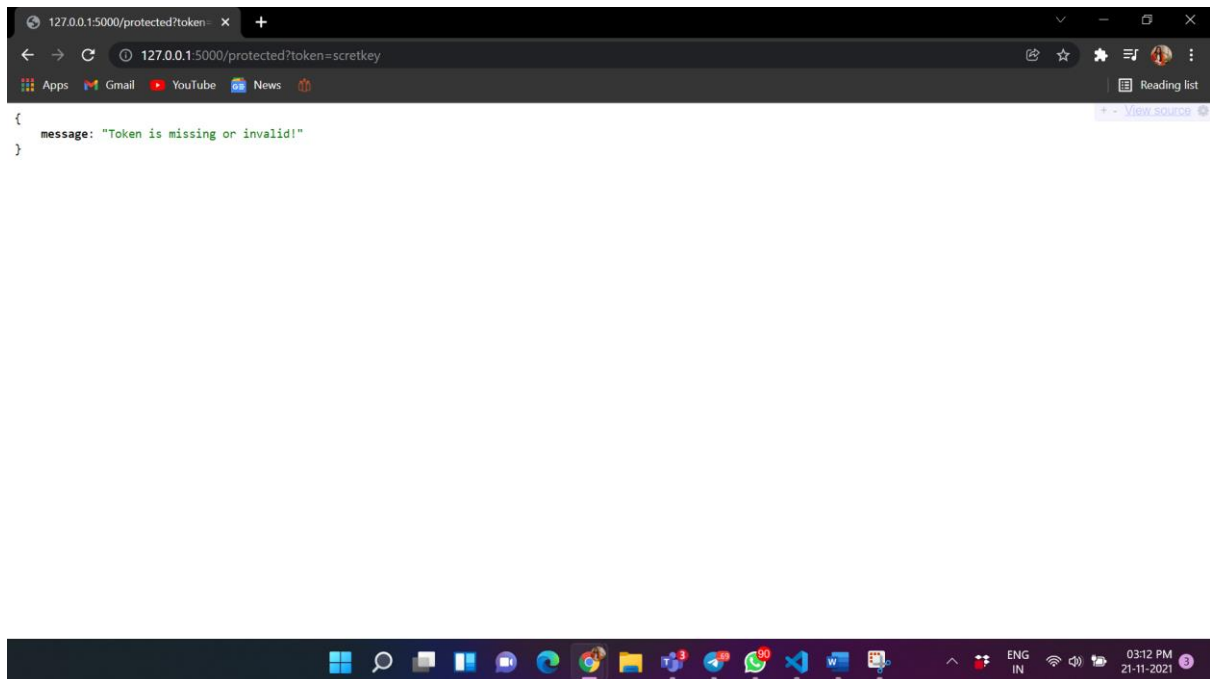
**If we check the unprotected part of the api we can simply see the message without any verification**



**If I give the token from the login page then we can view the protected part**



## If you give an invalid key it prints token is invalid



## If we don't give the token then it shows token is missing

