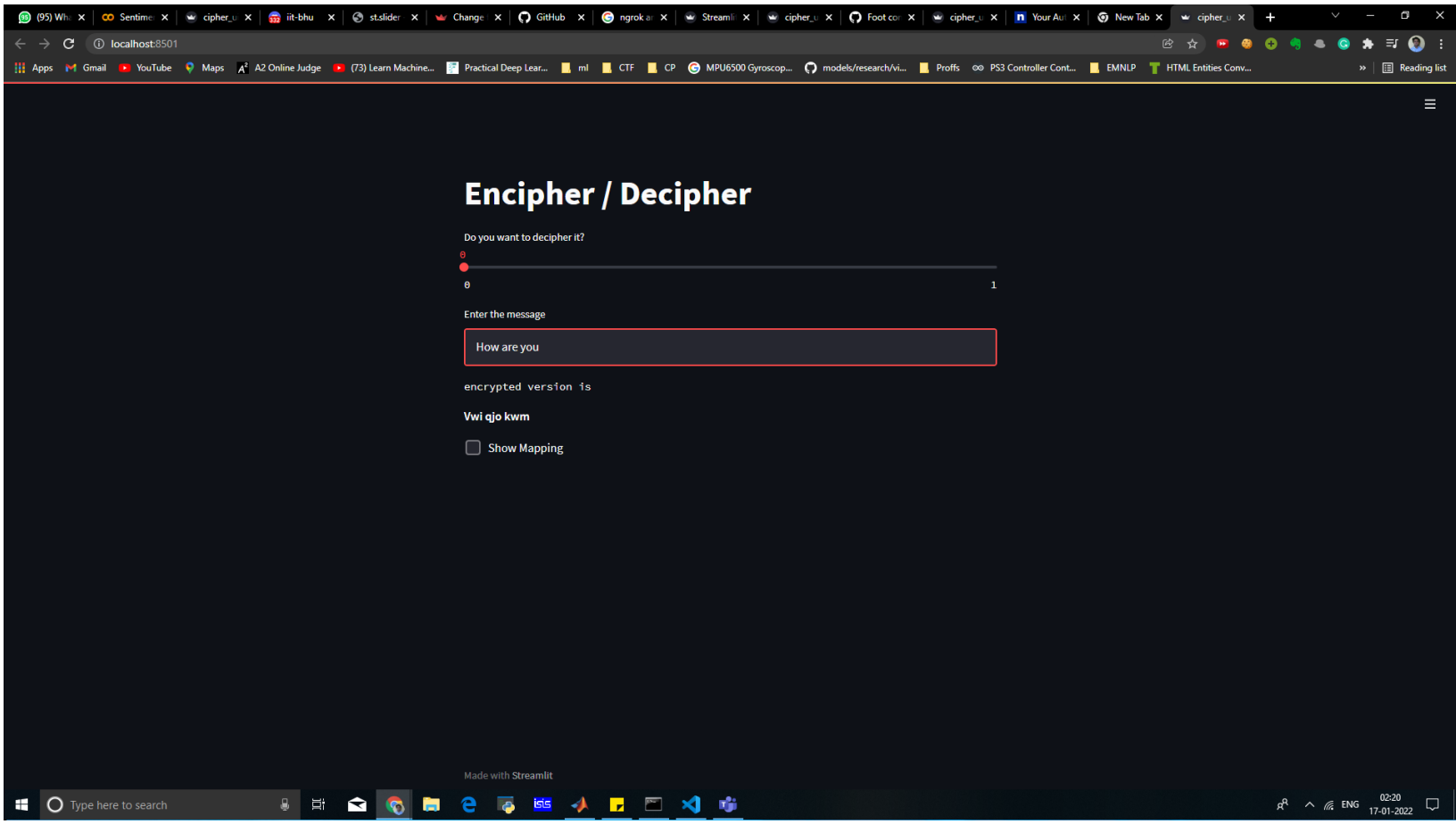
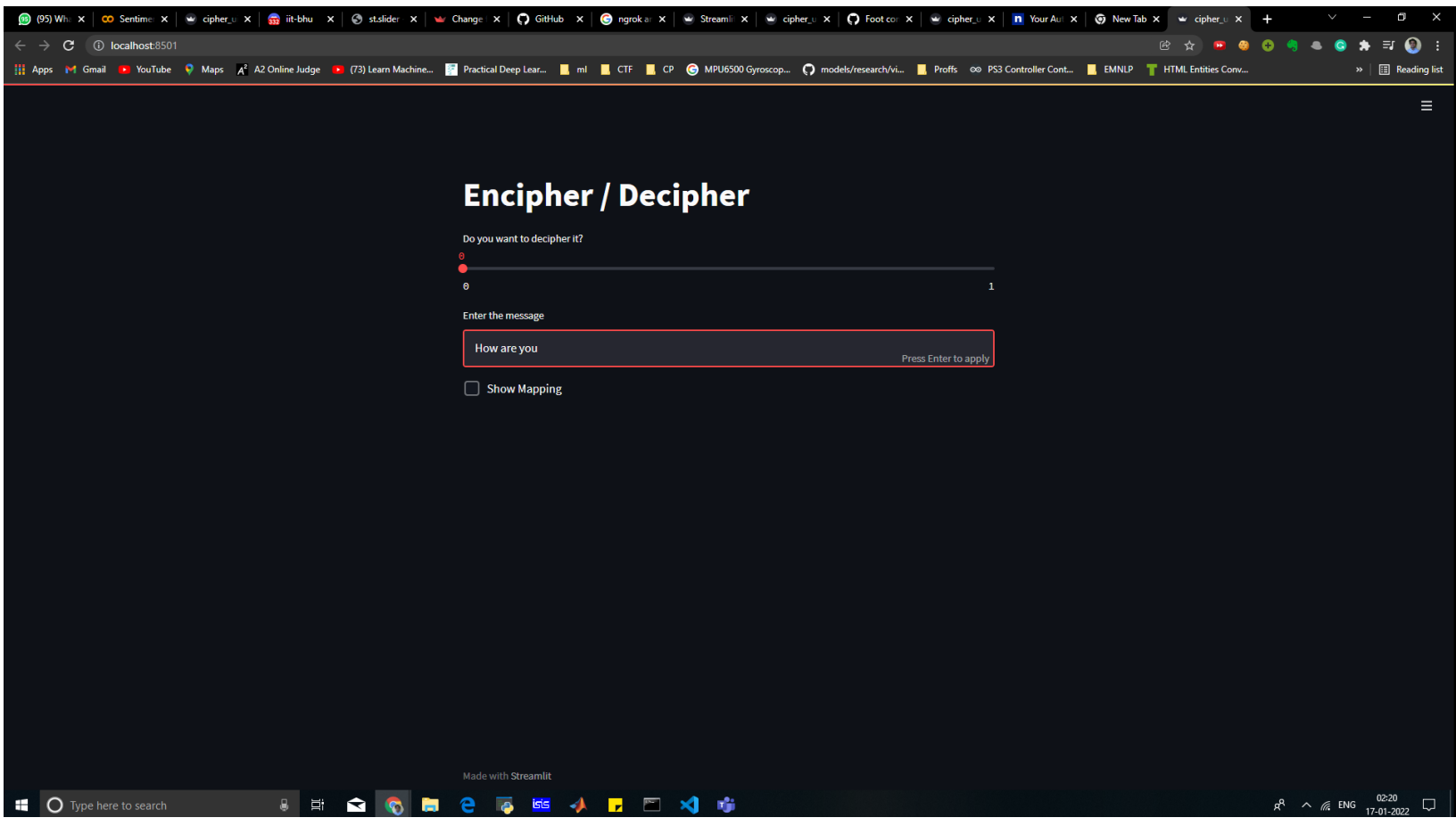
CS537: Network Security

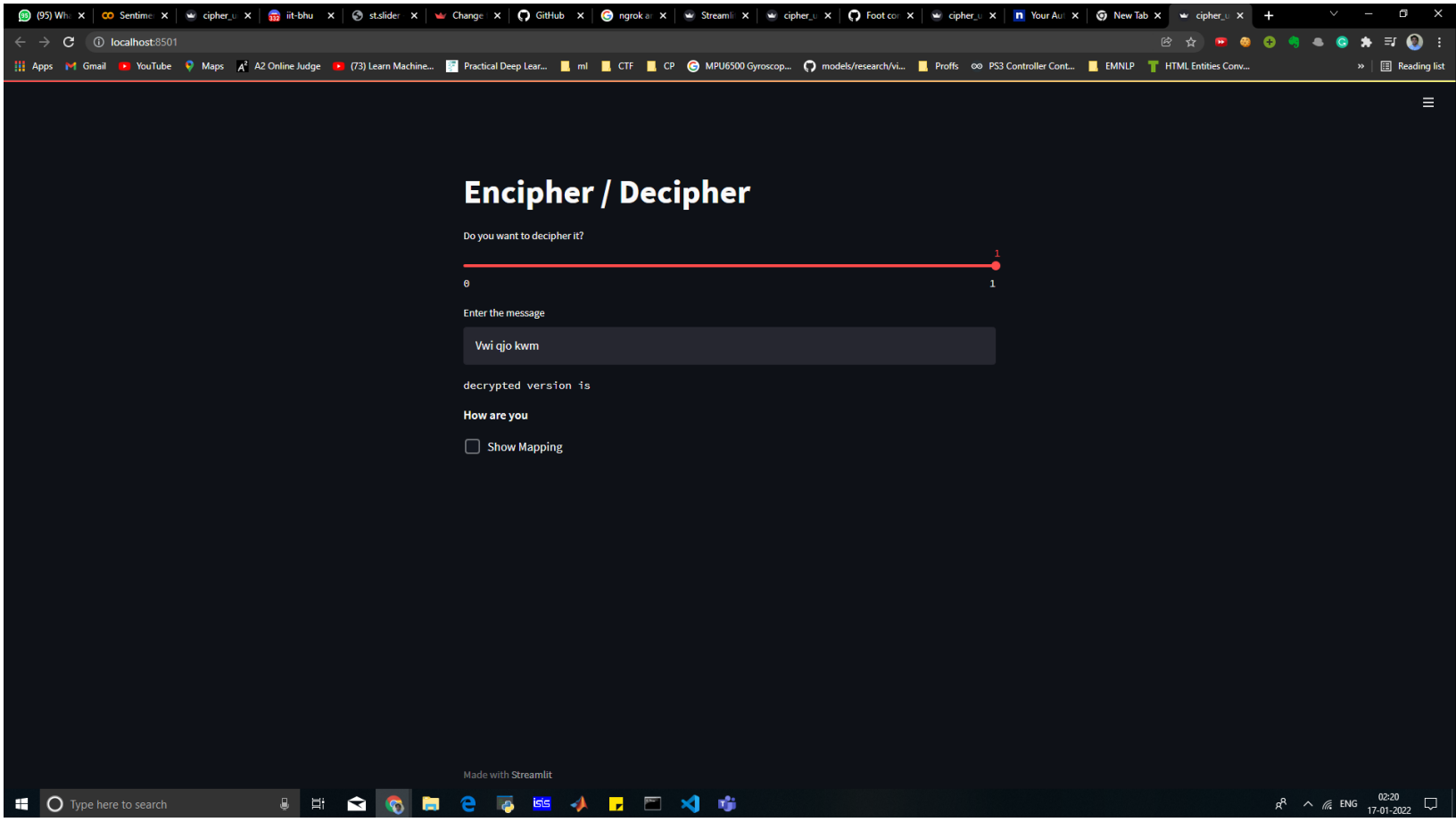
Practical Assignment 0

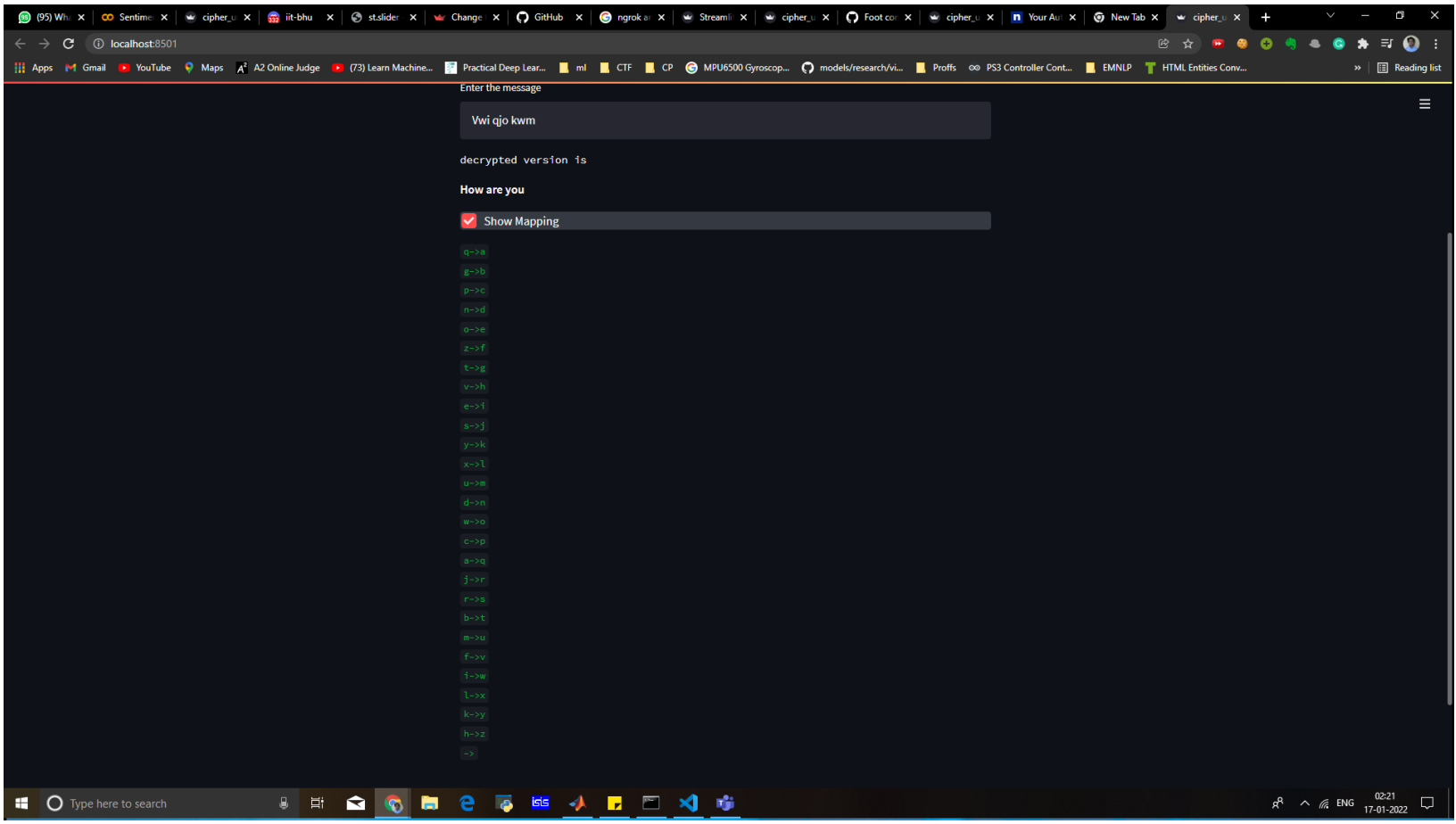
Name: Somnath Sendhil Kumar **Roll no:** 19085089

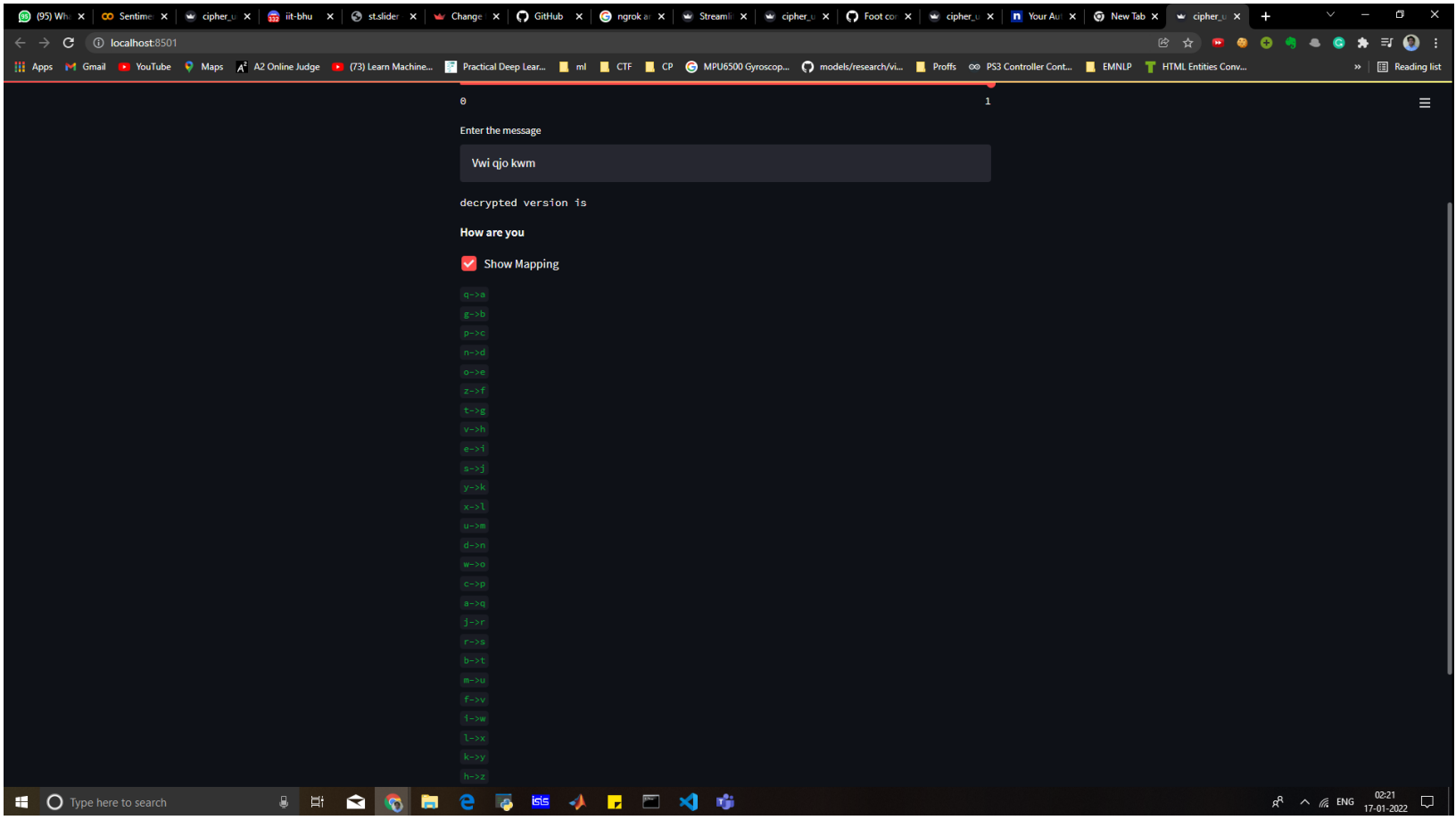
Screenshots:

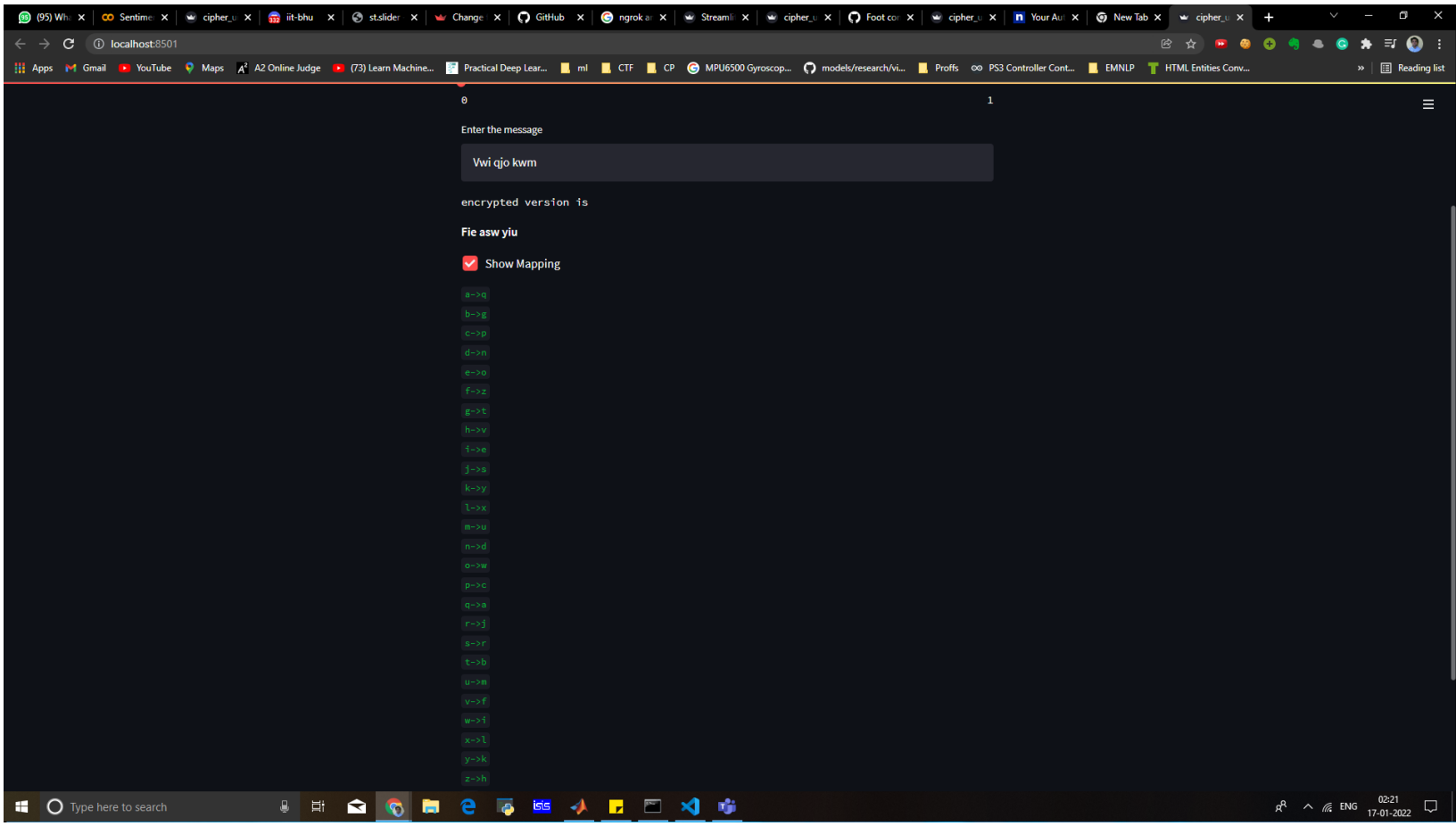


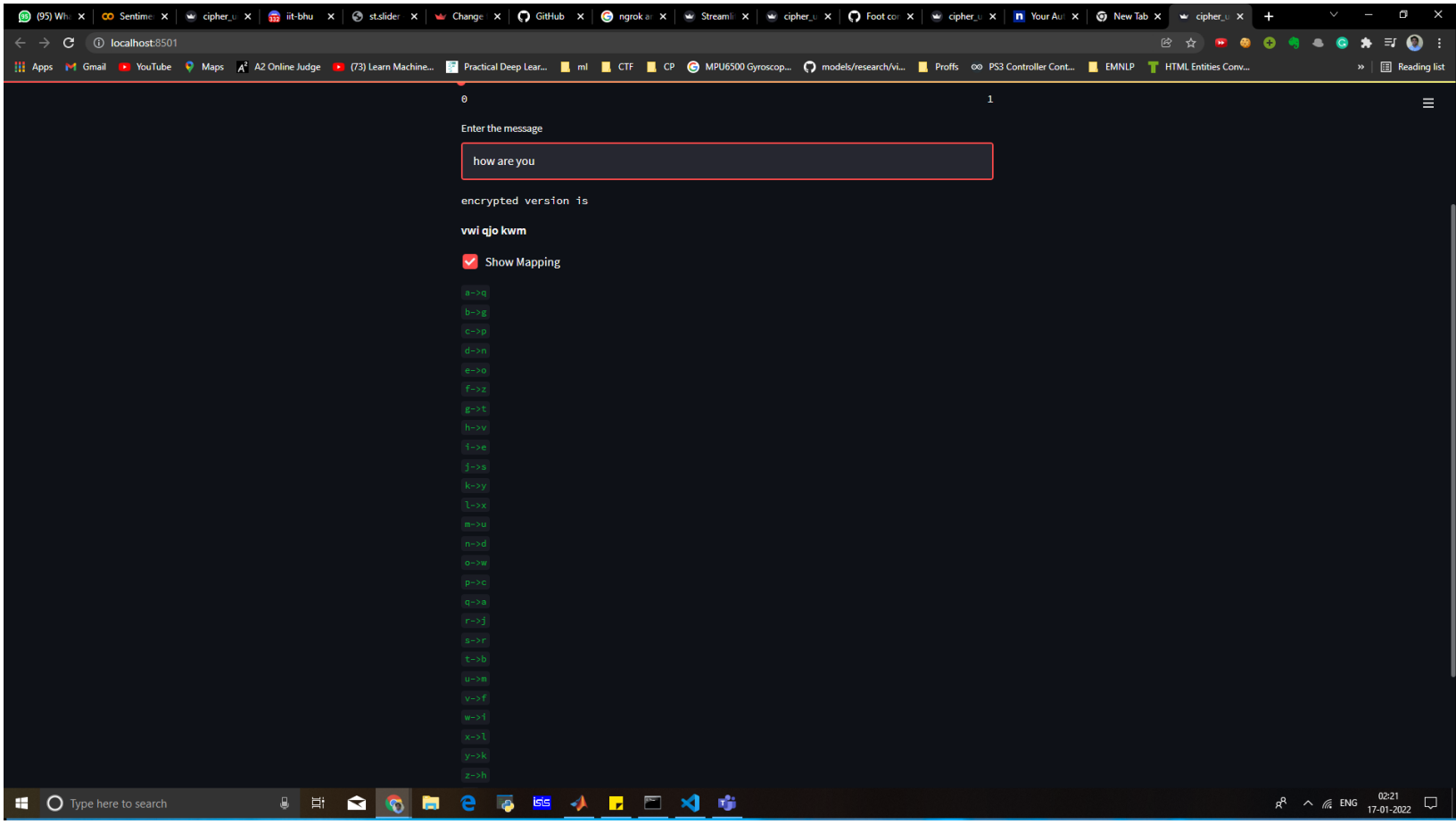


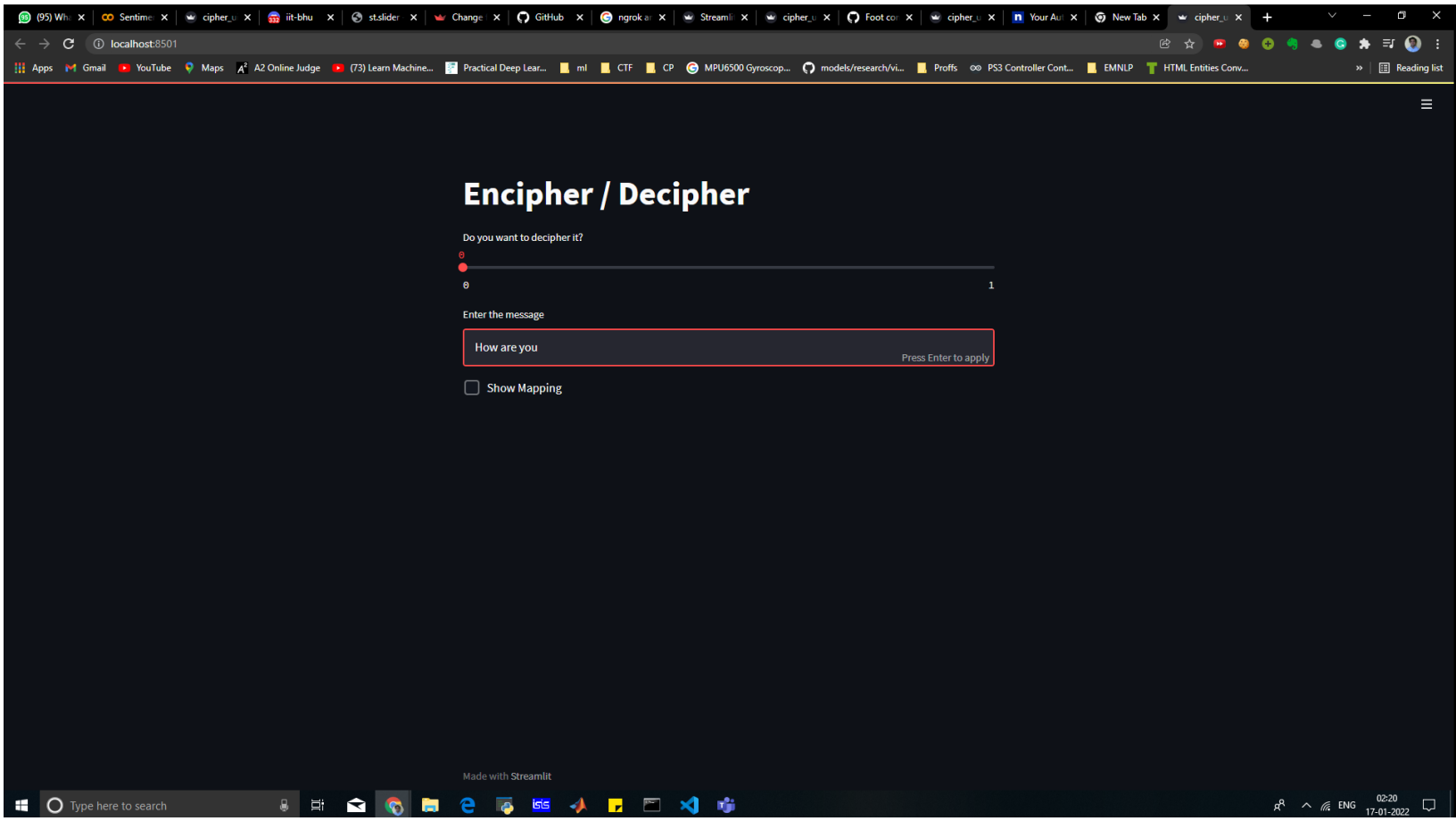

localhost:8501
Encipher / Decipher
Do you want to decipher it?
1
0
1
Enter the message
Vwi qjo kwm
decrypted version is
How are you
Show Mapping
Made with Streamlit


localhost:8501
Enter the message
Vwi qjo kwm
decrypted version is
How are you
Show Mapping
q->a
g->b
p->c
n->d
o->e
z->f
t->g
v->h
e->i
a->j
y->k
x->l
u->m
d->n
w->o
c->p
a->q
j->r
r->s
b->t
m->u
f->v
i->w
l->x
k->y
h->z
->
Type here to search
ENG
02:21
17-01-2022


0
1
Enter the message
Vwi qjo kwm
decrypted version is
How are you
Show Mapping
q->a
g->b
p->c
n->d
o->e
z->f
t->g
v->h
e->i
s->j
y->k
x->l
u->m
d->n
w->o
c->p
a->q
j->r
r->s
b->t
m->u
f->v
i->w
l->x
k->y
h->z


0
1
Enter the message
Vwi qjo kwm
encrypted version is
Fie asw yiu
Show Mapping
a->q
b->g
c->p
d->n
e->o
f->z
g->t
h->v
i->e
j->s
k->y
l->x
m->u
n->d
o->w
p->c
q->a
r->j
s->r
t->b
u->m
v->f
w->i
x->l
y->k
z->h





Source Code: https://github.com/hex-plex/CS537-Assignments/blob/master/PA0/cipher_ui.py

Deployment link: http://a0f8-103-151-208-4.ngrok.io/

```
import streamlit as st

map_e_to_d = {
    'a':'q',
    'b':'g',
    'c':'p',
    'd':'n',
    'e':'o',
    'f':'z',
    'g':'t',
    'h':'v',
    'i':'e',
    'j':'s',
    'k':'y',
    'l':'x',
    'm':'u',
    'n':'d',
    'o':'w',
    'p':'c',
    'q':'a',
    'r':'j',
    's':'r',
    't':'b',
    'u':'m',
    'v':'f',
    'w':'i',
    'x':'l',
    'y':'k',
    'z':'h',
    ' ':' '
}
## Random Manual mapping
map_d_to_e = {

}
## Inverse Mapping

def gen_inv():
    '''
    Inverting the encryption map
    '''
    if len(map_d_to_e)==0:
        for e, d in map_e_to_d.items():
            map_d_to_e[d]=e

def encrypt(x):
    '''
    Encrypts the input based on table lookup
```

```
    '''
    out = ''
    try:
        for c in x:
            sub = map_e_to_d[c.lower()]
            out += sub.lower() if c.islower() else sub.upper()
        return out
    except KeyError:
        st.warning("Input character out of encoding dictionary")
        return x

def decrypt(x):
    '''
    Decrypts the input based on table lookup
    '''
    gen_inv()

    out = ''
    try:
        for c in x:
            sub = map_d_to_e[c.lower()]
            out += sub.lower() if c.islower() else sub.upper()
        return out
    except KeyError:
        st.warning("Input character out of decoding dictionary")
        return x

st.title('Encipher / Decipher')
level = st.slider("Do you want to decipher it?", 0, 1)

input_string = st.text_input('Enter the message')
# crypt = st.text_input('Enter the encrypted message')

result = ''
if input_string!='':
    if level == 0:
        result='encrypted version is \n','**' + encrypt(str(input_string)) + '**'
    elif level == 1:
        result='decrypted version is \n', '**' + decrypt(str(input_string)) + '**'
# if crypt!='':
#     result='decrypted version is ' + str(crypt)

if result!='':
    st.text(result[0])
    st.markdown(result[1])

if st.checkbox("Show Mapping"):
    if level == 0:
```

```
        st.markdown("<br/>".join([ "`"+k+"->"+v+"`" for k, v in map_e_to_d.items()]),
unsafe_allow_html=True)
    elif level == 1:
        gen_inv()
        st.markdown("<br/>".join([ "`"+k+"->"+v+"`" for k, v in map_d_to_e.items()]),
unsafe_allow_html=True)
```