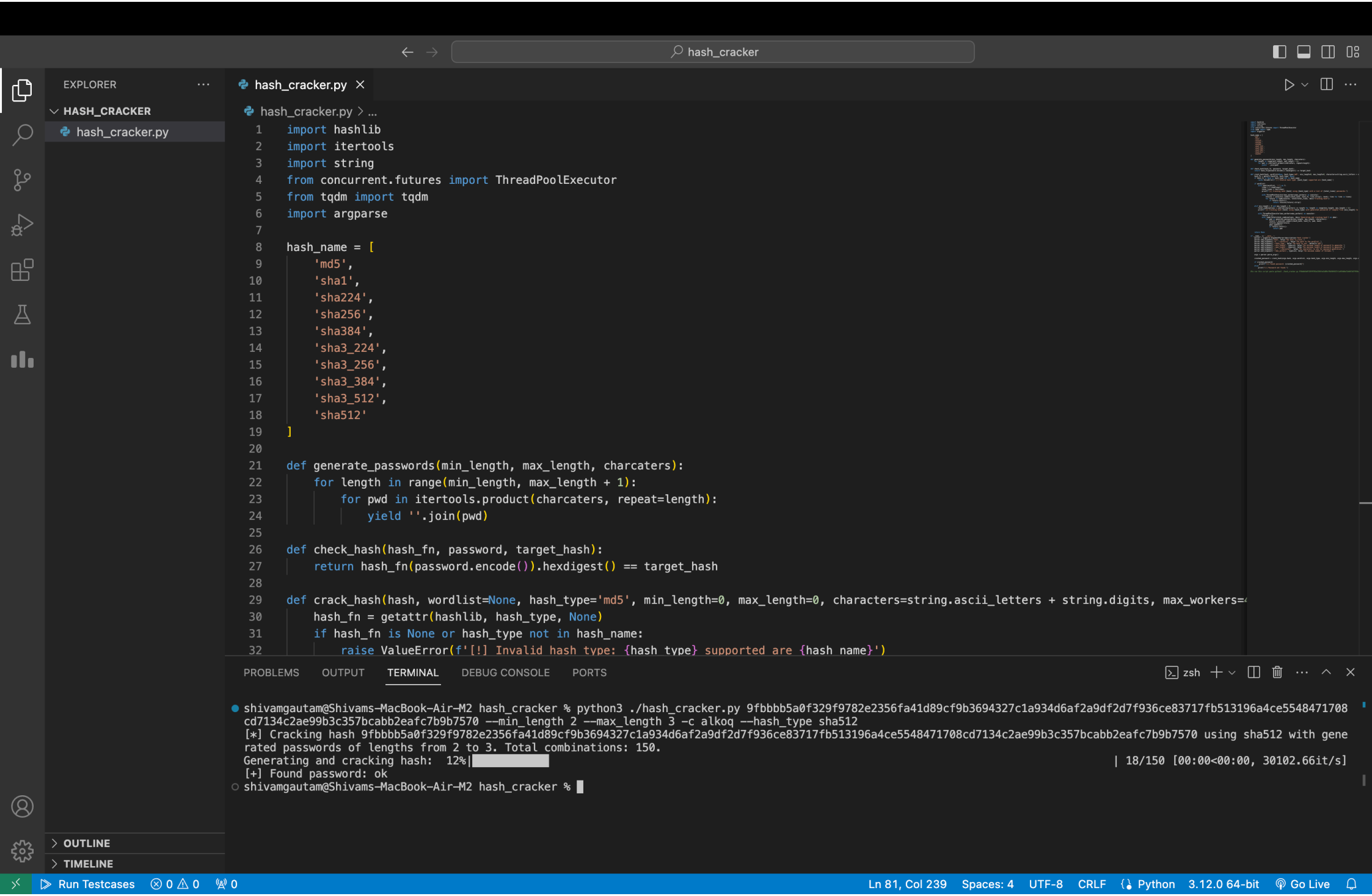
hash_cracker
EXPLORER
HASH_CRACKER
hash_cracker.py
hash_cracker.py
hash_cracker.py > ...
import hashlib
import itertools
import string
from concurrent.futures import ThreadPoolExecutor
from tqdm import tqdm
import argparse
hash_name = [
'md5',
'sha1',
'sha224',
'sha256',
'sha384',
'sha3_224',
'sha3_256',
'sha3_384',
'sha3_512',
'sha512'
]
def generate_passwords(min_length, max_length, charcaters):
for length in range(min_length, max_length + 1):
for pwd in itertools.product(charcaters, repeat=length):
yield ''.join(pwd)
def check_hash(hash_fn, password, target_hash):
return hash_fn(password.encode()).hexdigest() == target_hash
def crack_hash(hash, wordlist=None, hash_type='md5', min_length=0, max_length=0, characters=string.ascii_letters + string.digits, max_workers=
hash_fn = getattr(hashlib, hash_type, None)
if hash_fn is None or hash_type not in hash_name:
raise ValueError(f'[!] Invalid hash type: {hash_type} supported are {hash_name}')
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
PORTS
zsh
shivamgautam@Shivams-MacBook-Air-M2 hash_cracker % python3 ./hash_cracker.py 9fbbbb5a0f329f9782e2356fa41d89cf9b3694327c1a934d6af2a9df2d7f936ce83717fb513196a4ce5548471708
cd7134c2ae99b3c357bcabb2eafc7b9b7570 --min_length 2 --max_length 3 -c alkoq --hash_type sha512
[*] Cracking hash 9fbbbb5a0f329f9782e2356fa41d89cf9b3694327c1a934d6af2a9df2d7f936ce83717fb513196a4ce5548471708cd7134c2ae99b3c357bcabb2eafc7b9b7570 using sha512 with gene
rated passwords of lengths from 2 to 3. Total combinations: 150.
Generating and cracking hash: 12%| | 18/150 [00:00<00:00, 30102.66it/s]
[+] Found password: ok
shivamgautam@Shivams-MacBook-Air-M2 hash_cracker %
OUTLINE
TIMELINE
Run Testcases
0 0
0
Ln 81, Col 239
Spaces: 4
UTF-8
CRLF
Python
3.12.0 64-bit
Go Live