

**University Of Mumbai
Centre for Distance & Online Education**



PRACTICAL JOURNAL SEM IV

**BLOCK CHAIN
DEEP LEARNING
CYBER FORENSICS**

SUBMITTED BY

**MAKRAND PRAFUL GHAG
APPLICATION ID: 93669
SEAT NO: 2900084**

**MASTER OF SCIENCE IN INFORMATION TECHNOLOGY PART-II
SEMESTER IV**

ACADEMIC YEAR 2023-2024

**INSTITUTE OF DISTANCE AND OPEN LEARNING IDOL
BUILDING, VIDYANAGARI,
SANTACRUZ (EAST), MUMBAI-400 098**

**CONDUCTED AT
RIZVI COLLEGE OF ARTS, SCIENCE AND COMMERCE BANDRA (W),
MUMBAI 400050**

University Of Mumbai
Centre for Distance & Online Education



Dr.Shangar Dayal Sharama Bhavan, Kalina, Vidanagari, Santacruz (E),
Mumbai-400 098.

Certificate

This is to certify that

Mr. **MAKRAND GHAG** Application ID: **93669**, Seat No: **2900084** from **Rizvi College of Arts, Science and Commerce Bandra(W)**, Mumbai 400 050 has successfully completed all the practical of Paper I titled **BLOCK CHAIN** for M.sc (IT) Part II in the academic year 2023-2024.

Section I _____

Section II _____

MSc (IT) Co-ordinator, IDOL

External Examiner

MSC.IT PART II

BLOCKCHAIN PRACTICAL JOURNAL

Index

Sr.No	Topic	Remark
1	Practical 1 : Write the following programs for Blockchain in Python : (I). A simple client class that generates the private and public keys by using the built in Python RSA algorithm and test it (II). A transaction class to send and receive money and test it .	
2	Practical 2 : Write the following programs for Blockchain in Python : (I).Create multiple transactions and display them . (II). Create a blockchain, a genesis block and execute it .	
3	Practical 3 : Write the following programs for Blockchain in Python : (I).Create a mining function and test it . (II).Add blocks to the miner and dump the blockchain .	
4	Practical 4 : Implement and demonstrate the use of the following in Solidity : (I).Varaible (II).Operators (III).Loops (IV).Decision Making (V).Strings	
5	Practical 5 : Implement and demonstrate the use of the following in Solidity : (I).Arrays (II).Enums (III).Structs (IV).Mappings (V).Coversations (VI).Ether Units (VII).Special Variables	

Index

Sr.No	Topic	Date	Remark
6	Practical 6: Implement and demonstrate the use of the following in Solidity :		
	(I).Functions		
	(II).View Functions		
	(III).Pure Functions		
	(IV).Fallback Functions		
	(V).Function Overloading		
	(VI).Mathematical Functions		
	(VII).Cryptographic Functions		
7	Practical 7 : Implement and demonstrate the use of the following in Solidity :		
	(I).Contracts		
	(II).Inheritance		
	(III).Constructors		
	(IV).Abstract Class		
	(V).Interfaces		
8	Practical 8 : Implement and demonstrate the use of the following in Solidity :		
	(I).Libraries		
	(II).Assembly		
	(III).Events		
	(IV).Error Handling		

Practical 1

Write the following programs for Blockchain in Python :

(I) A simple client class that generates the private and public keys by using the built in Python RSA algorithm and test it

(II) A transaction class to send and receive money and test it .

→

```
# import libraries

import hashlib
import random
import string
import json
import binascii
import numpy as np
import pandas as pd
import pylab as pl
import logging
import datetime
import collections

pip install pycryptodome

# following imports are required by PKI

import Crypto
import Crypto.Random
from Crypto.Hash import SHA
from Crypto.PublicKey import RSA
from Crypto.Signature import PKCS1_v1_5

import binascii
```

```

class Client:

    def __init__(self):
        random = Crypto.Random.new().read
        self._private_key = RSA.generate(1024, random)
        self._public_key = self._private_key.publickey()
        self._signer = PKCS1_v1_5.new(self._private_key)

    @property
    def identity(self):
        return binascii.hexlify(self._public_key.exportKey(format='DER')).decode('ascii')

class Transaction:

    def __init__(self, sender, recipient, value):
        self.sender = sender
        self.recipient = recipient
        self.value = value
        self.time = datetime.datetime.now()

    def to_dict(self):
        if self.sender == "Genesis":
            identity = "Genesis"
        else:
            identity = self.sender.identity
        return collections.OrderedDict(
            {'sender': identity,
             'recipient': self.recipient,
             'value': self.value,
             'time': self.time})

    def sign_transaction(self):
        private_key = self.sender._private_key

```

```

signer = PKCS1_v1_5.new(private_key)
h = SHA.new(str(self.to_dict()).encode('utf8'))
return binascii.hexlify(signer.sign(h)).decode('ascii')

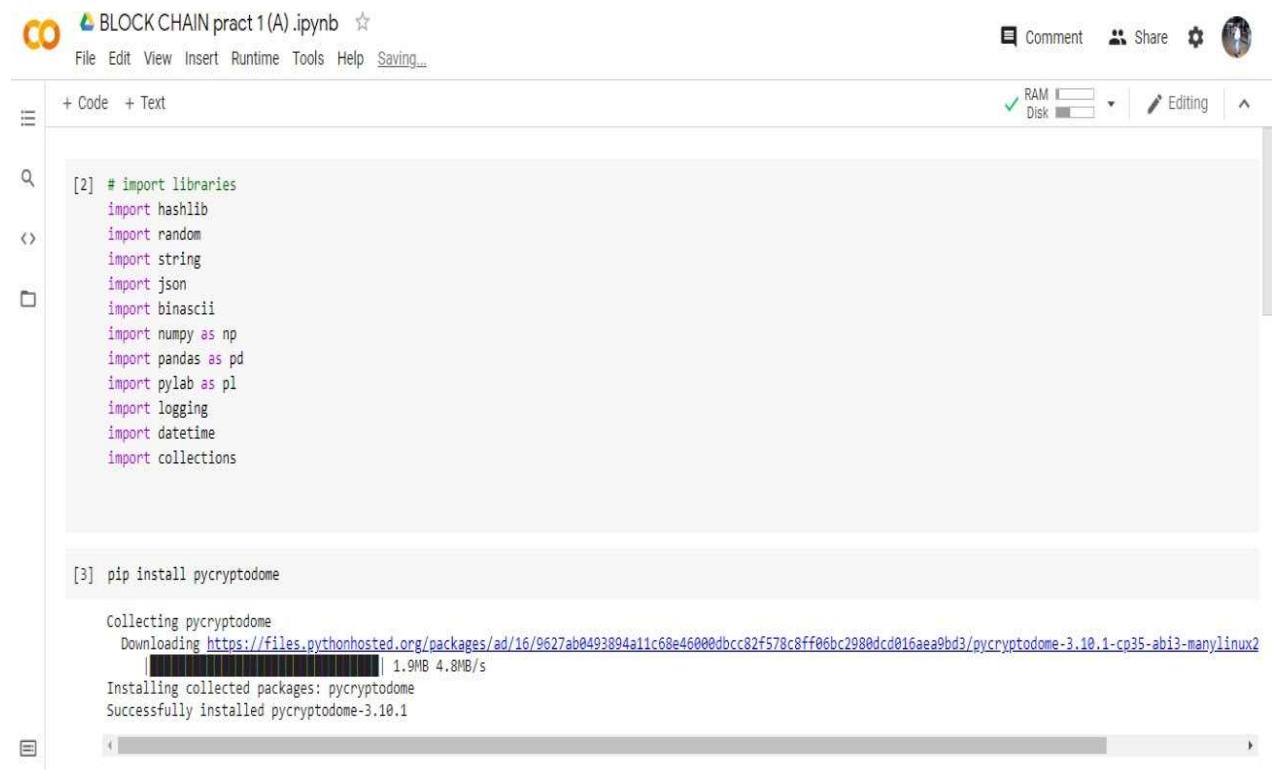
Dinesh = Client()
Ramesh = Client()

t = Transaction(Dinesh,Ramesh.identity,5.0)

signature = t.sign_transaction()
print (signature)

```

Output:



The screenshot shows a Jupyter Notebook interface with the following details:

- Title:** BLOCK CHAIN pract1(A).ipynb
- Toolbar:** File, Edit, View, Insert, Runtime, Tools, Help, Saving...
- Code Cell 2:**

```
[2] # import libraries
import hashlib
import random
import string
import json
import binascii
import numpy as np
import pandas as pd
import pylab as pl
import logging
import datetime
import collections
```
- Code Cell 3:**

```
[3] pip install pycryptodome
Collecting pycryptodome
  Downloading https://files.pythonhosted.org/packages/ad/16/9627ab0493894a11c68e46000dbcc82f578c8ff06bc2980dc016aea9bd3/pycryptodome-3.10.1-cp35-abi3-manylinux2
Installing collected packages: pycryptodome
Successfully installed pycryptodome-3.10.1
```

BLOCK CHAIN pract 1(A).ipynb

```
[5] import Crypto
import Crypto.Random
from Crypto.Hash import SHA
from Crypto.PublicKey import RSA
from Crypto.Signature import PKCS1_v1_5

import binascii

import binascii
class Client:
    def __init__(self):
        random = Crypto.Random.new().read
        self._private_key = RSA.generate(1024, random)
        self._public_key = self._private_key.publickey()
        self._signer = PKCS1_v1_5.new(self._private_key)

    @property
    def identity(self):
        return binascii.hexlify(self._public_key.exportKey(format='DER')).decode('ascii')

class Transaction:
    def __init__(self, sender, recipient, value):
        self.sender = sender
        self.recipient = recipient
        self.value = value
        self.time = datetime.datetime.now()
```

BLOCK CHAIN pract 1(A).ipynb

```
self.value = value
self.time = datetime.datetime.now()
def to_dict(self):
    if self.sender == "Genesis":
        identity = "Genesis"
    else:
        identity = self.sender.identity
    return collections.OrderedDict([
        ('sender': identity,
         'recipient': self.recipient,
         'value': self.value,
         'time': self.time)])
def sign_transaction(self):
    private_key = self.sender._private_key
    signer = PKCS1_v1_5.new(private_key)
    h = SHA.new(str(self.to_dict()).encode('utf8'))
    return binascii.hexlify(signer.sign(h)).decode('ascii')

Dinesh = Client()
Ramesh = Client()

t = Transaction(Dinesh,Ramesh.identity,5.0)

signature = t.sign_transaction()
print (signature)
```

Practical 2

Write the following programs for Blockchain in Python :

- (I).Create multiple transactions and display them .
- (II). Create a blockchain, a genesis block and execute it .

→

```
def display_transaction(transaction):  
    #for transaction in transactions:  
    dict = transaction.to_dict()  
    print ("sender: " + dict['sender'])  
    print (' ----')  
    print ("recipient: " + dict['recipient'])  
    print (' ----')  
    print ("value: " + str(dict['value']))  
    print (' ----')  
    print ("time: " + str(dict['time']))  
    print (' ----')
```

```
transactions = []
```

```
Dinesh = Client()  
Ramesh = Client()  
Seema = Client()  
Vijay = Client()
```

```
t1 = Transaction(  
    Dinesh,  
    Ramesh.identity,  
    15.0)
```

)

```
t1.sign_transaction()  
transactions.append(t1)
```

```
t2 = Transaction(
```

```
    Dinesh,
```

```
    Seema.identity,
```

```
    6.0
```

)

```
t2.sign_transaction()  
transactions.append(t2)
```

```
t3 = Transaction(
```

```
    Ramesh,
```

```
    Vijay.identity,
```

```
    2.0
```

)

```
t3.sign_transaction()  
transactions.append(t3)
```

```
t4 = Transaction(
```

```
    Seema,
```

```
    Ramesh.identity,
```

```
    4.0
```

)

```
t4.sign_transaction()  
transactions.append(t4)
```

```
t5 = Transaction(
```

```
    Vijay,
```

```
    Seema.identity,
```

```
    7.0
)
t5.sign_transaction()
transactions.append(t5)
t6 = Transaction(
    Ramesh,
    Seema.identity,
    3.0
)
t6.sign_transaction()
transactions.append(t6)
t7 = Transaction(
    Seema,
    Dinesh.identity,
    8.0
)
t7.sign_transaction()
transactions.append(t7)
t8 = Transaction(
    Seema,
    Ramesh.identity,
    1.0
)
t8.sign_transaction()
transactions.append(t8)
t9 = Transaction(
    Vijay,
    Dinesh.identity,
    5.0
```

```
)  
t9.sign_transaction()  
transactions.append(t9)  
t10 = Transaction(  
    Vijay,  
    Ramesh.identity,  
    3.0  
)  
t10.sign_transaction()  
transactions.append(t10)
```

for transaction in transactions:

```
    display_transaction (transaction)  
    print ('.....')
```

class Block:

```
    def __init__(self):  
        self.verified_transactions = []  
        self.previous_block_hash = ""  
        self.Nonce = ""
```

last_block_hash = ""

Dinesh = Client()

```
t0 = Transaction (  
    "Genesis",  
    Dinesh.identity,  
    500.0
```

```
)
```

```
block0 = Block()
```

```
block0.previous_block_hash = None
```

```
Nonce = None
```

```
block0.verified_transactions.append(t0)
```

```
digest = hash(block0)
```

```
last_block_hash = digest
```

```
TPCoins = []
```

```
def dump_blockchain(self):
```

```
    print("Number of blocks in the chain: " + str(len(self)))
```

```
    for x in range(len(TPCoins)):
```

```
        block_temp = TPCoins[x]
```

```
        print("block # " + str(x))
```

```
        for transaction in block_temp.verified_transactions:
```

```
            display_transaction(transaction)
```

```
            print('.....')
```

```
            print('=====')
```

```
TPCoins.append(block0)
```

```
dump_blockchain(TPCoins)
```

Output:

```

[10] def display_transaction(transaction):
    #for transaction in transactions:
    dict = transaction.to_dict()
    print ("sender: " + dict['sender'])
    print ('-----')
    print ("recipient: " + dict['recipient'])
    print ('-----')
    print ("value: " + str(dict['value']))
    print ('-----')
    print ("time: " + str(dict['time']))
    print ('-----')

[11] transactions = []

[12] Dinesh = Client()
    Ramesh = Client()
    Seema = Client()
    Vijay = Client()

[13] t1 = Transaction(
        Dinesh,
        Ramesh.identity,
        15.0
    )

[15] t1.sign_transaction()
    transactions.append(t1)

[16] t2 = Transaction(
        Dinesh,
        Seema.identity,
        6.0
    )
    t2.sign_transaction()
    transactions.append(t2)
    t3 = Transaction(
        Ramesh,
        Vijay.identity,
        2.0
    )
    t3.sign_transaction()
    transactions.append(t3)
    t4 = Transaction(
        Seema,
        Ramesh.identity,
        4.0
    )
    t4.sign_transaction()
    transactions.append(t4)
    t5 = Transaction(
        Vijay,
        Seema.identity,
        7.0
    )
    t5.sign_transaction()
    transactions.append(t5)
    t6 = Transaction(
        Ramesh,
        Seema.identity,
        3.0
    )
    t6.sign_transaction()
    transactions.append(t6)
    t7 = Transaction(
        Seema,
        Dinesh.identity,
        8.0
    )
    t7.sign_transaction()
    transactions.append(t7)
    t8 = Transaction(
        Seema,
        Ramesh.identity,
        1.0
    )
    t8.sign_transaction()
    transactions.append(t8)
    t9 = Transaction(
        Vijay,
        Dinesh.identity,
        5.0
    )

```

```

[16] t8.sign_transaction()
transactions.append(t8)
t9 = Transaction(
    Vijay,
    Dinesh.identity,
    5.0
)
t9.sign_transaction()
transactions.append(t9)
t10 = Transaction(
    Vijay,
    Ramesh.identity,
    3.0
)
t10.sign_transaction()
transactions.append(t10)

[17] for transaction in transactions:
    display_transaction(transaction)
    print('-----')

sender: 30819f300d06092a864886f70d010101050003818d0030818
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100b5c36acef717c85079d5
-----
value: 15.0
-----
time: 2021-05-05 07:37:46.751762
-----
sender: 30819f300d06092a864886f70d010101050003818d003081890281810096e3e436b160bf2feecba6d
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100b5c36acef717c85079d5
value: 15.0
time: 2021-05-05 07:37:46.751762
-----
sender: 30819f300d06092a864886f70d010101050003818d003081890281810096e3e436b160bf2feecba6d
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100a3b073014e3d9a1e3353
value: 6.0
time: 2021-05-05 07:38:17.042489
-----
sender: 30819f300d06092a864886f70d010101050003818d0030818902818100b5c36acef717c85079d5eb8
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100967bf6965c7e25f7c711
value: 2.0
time: 2021-05-05 07:38:17.044049
-----
sender: 30819f300d06092a864886f70d010101050003818d0030818902818100a3b073014e3d9a1e335365f
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100b5c36acef717c85079d5
value: 4.0
time: 2021-05-05 07:38:17.045503
-----
sender: 30819f300d06092a864886f70d010101050003818d0030818902818100967bf6965c7e25f7c711a30
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100a3b073014e3d9a1e3353
value: 7.0
-----
time: 2021-05-05 07:38:17.046076

[18] class Block:
    def __init__(self):
        self.verified_transactions = []
        self.previous_block_hash = ""
        self.Nonce = ""

[19] last_block_hash = ""

```



```

+ Code + Text
✓ RAM Disk | Editing | ^

[20] Dinesh = Client()

[21] t0 = Transaction (
    "Genesis",
    Dinesh.identity,
    500.0
)

[22] block0 = Block()

[23] block0.previous_block_hash = None
Nonce = None

[24] block0.verified_transactions.append (t0)

[25] digest = hash (block0)
last_block_hash = digest

[26] def dump_blockchain (self):
    print ("Number of blocks in the chain: " + str(len (self)))
    for x in range (len(TPCoins)):
        block_temp = TPCoins[x]
        print ("block # " + str(x))
        for transaction in block_temp.verified_transactions:
            display_transaction (transaction)
            print ('-----')
        print ('=====') 

[27] TPCoins = []

[28] def dump_blockchain (self):
    print ("Number of blocks in the chain: " + str(len (self)))
    for x in range (len(TPCoins)):
        block_temp = TPCoins[x]
        print ("block # " + str(x))
        for transaction in block_temp.verified_transactions:
            display_transaction (transaction)
            print ('-----')
        print ('=====') 

[29] TPCoins.append (block0)

[30] dump_blockchain(TPCoins)

Number of blocks in the chain: 1
block # 0
sender: Genesis
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100df7d100622cc76eab161
-----
value: 500.0
-----
time: 2021-05-05 07:41:01.767902
-----
=====
=====
```

Practical 3

Write the following programs for Blockchain in Python :

- (I).Create a mining function and test it .
- (II).Add blocks to the miner and dump the blockchain .

→

```
def sha256(message):  
    return hashlib.sha256(message.encode('ascii')).hexdigest()  
  
def mine(message, difficulty=1):  
    assert difficulty >= 1  
    prefix = '1' * difficulty  
    for i in range(1000):  
        digest = sha256(str(hash(message)) + str(i))  
        if digest.startswith(prefix):  
            print ("after " + str(i) + " iterations found nonce: "+ digest)  
            return digest
```

```
last_transaction_index = 0
```

```
block = Block()  
for i in range(3):  
    temp_transaction = transactions[last_transaction_index]  
    # validate transaction  
    # if valid  
    block.verified_transactions.append (temp_transaction)  
    last_transaction_index += 1 mine ("test message", 2)
```

```

block.previous_block_hash = last_block_hash
block.Nonce = mine (block, 2)
digest = hash (block)
TPCoins.append (block)
last_block_hash = digest

# Miner 2 adds a block
block = Block()

for i in range(3):
    temp_transaction = transactions[last_transaction_index]
    # validate transaction
    # if valid
    block.verified_transactions.append (temp_transaction)
    last_transaction_index += 1
    block.previous_block_hash = last_block_hash
    block.Nonce = mine(block, 2)
    digest = hash (block)
    TPCoins.append (block)
    last_block_hash = digest

# Miner 3 adds a block
block = Block()

for i in range(3):
    temp_transaction = transactions[last_transaction_index]
    #display_transaction (temp_transaction)
    # validate transaction
    # if valid

```

```
block.verified_transactions.append (temp_transaction)
last_transaction_index += 1
```

```
block.previous_block_hash = last_block_hash
block.Nonce = mine (block, 2)
digest = hash (block)
```

```
TPCoins.append (block)
last_block_hash = digest
dump_blockchain(TPCoins)
```

Full Output:

```

[ ] # import libraries
import hashlib
import random
import string
import json
import binascii
import numpy as np
import pandas as pd
import pylab as pl
import logging
import datetime
import collections

[ ] pip install pycryptodome

Collecting pycryptodome
  Downloading https://files.pythonhosted.org/packages/ad/16/9627ab0493894a11c68e46000dbcc
    |████████| 1.9MB 5.6MB/s
Installing collected packages: pycryptodome
Successfully installed pycryptodome-3.10.1

[ ] # following imports are required by PKI
import Crypto
import Crypto.Random
from Crypto.Hash import SHA
from Crypto.PublicKey import RSA
from Crypto.Signature import PKCS1_v1_5

[ ] import binascii
class Client:
    def __init__(self):
        random = Crypto.Random.new().read
        self._private_key = RSA.generate(1024, random)
        self._public_key = self._private_key.publickey()
        self._signer = PKCS1_v1_5.new(self._private_key)

    @property
    def identity(self):
        return binascii.hexlify(self._public_key.exportKey(format='DER')).decode('ascii')

class Transaction:
    def __init__(self, sender, recipient, value):
        self.sender = sender
        self.recipient = recipient
        self.value = value
        self.time = datetime.datetime.now()
    def to_dict(self):
        if self.sender == "Genesis":
            identity = "Genesis"
        else:
            identity = self.sender.identity
        return collections.OrderedDict({
            'sender': identity,
            'recipient': self.recipient,
            'value': self.value,
            'time' : self.time})
    def sign_transaction(self):
        private_key = self.sender._private_key
        signer = PKCS1_v1_5.new(private_key)
        h = SHA.new(str(self.to_dict()).encode('utf8'))
        return binascii.hexlify(signer.sign(h)).decode('ascii')

[ ]

[ ] Dinesh = Client()
Ramesh = Client()

[ ] t = Transaction(Dinesh,Ramesh.identity,5.0)

[ ] signature = t.sign_transaction()
print (signature)

251f28f6f523733739979611b404c755210b5b6a70f28d5eb3e3049f7dceea873e472f0df41a55152c71bb6f5

[ ] def display_transaction(transaction):
    #for transaction in transactions:
    dict = transaction.to_dict()
    print ("sender: " + dict['sender'])
    print ('-----')

```

```

[ ] def display_transaction(transaction):
    #for transaction in transactions:
    dict = transaction.to_dict()
    print ("sender: " + dict['sender'])
    print ('-----')
    print ("recipient: " + dict['recipient'])
    print ('-----')
    print ("value: " + str(dict['value']))
    print ('-----')
    print ("time: " + str(dict['time']))
    print ('-----')

[ ] transactions = []

[ ] Dinesh = Client()
Ramesh = Client()
Seema = Client()
Vijay = Client()

[ ] t1 = Transaction(
    Dinesh,
    Ramesh.identity,
    15.0
)

[ ] t1.sign_transaction()
transactions.append(t1)

[ ] t2 = Transaction(
    Dinesh,
    Seema.identity,
    6.0
)
t2.sign_transaction()
transactions.append(t2)
t3 = Transaction(
    Ramesh,
    Vijay.identity,
    2.0
)
t3.sign_transaction()
transactions.append(t3)
t4 = Transaction(
    Seema,
    Ramesh.identity,
    4.0
)
t4.sign_transaction()
transactions.append(t4)
t5 = Transaction(
    Vijay,
    Seema.identity,
    7.0
)
t5.sign_transaction()
transactions.append(t5)
t6 = Transaction(
    Ramesh,
    Seema.identity,
    3.0
)
t6.sign_transaction()
transactions.append(t6)
t7 = Transaction(
    Seema,
    Dinesh.identity,
    8.0
)
t7.sign_transaction()
transactions.append(t7)
t8 = Transaction(
    Seema,
    Ramesh.identity,
    1.0
)
t8.sign_transaction()
transactions.append(t8)
t9 = Transaction(
    Vijay,
    Dinesh.identity,
    5.0
)
t9.sign_transaction()
transactions.append(t9)
t10 = Transaction(
    Vijay,
    Ramesh.identity,
    3.0
)
t10.sign_transaction()
transactions.append(t10)

```

```

+ Code + Text | ⌂ Copy to Drive
Connect ▾ | ✎ Editing | ▾

[ ] for transaction in transactions:
    display_transaction(transaction)
    print('-----')

sender: 30819f300d06092a864886f70d010101050003818d00308189028181008f5e89df98216eb2c1e4713
-----
recipient: 30819f300d06092a864886f70d010101050003818d003081890281810082e40f44f9cef42cac49
-----
value: 15.0
-----
time: 2021-04-08 06:10:52.172517
-----
sender: 30819f300d06092a864886f70d010101050003818d00308189028181008f5e89df98216eb2c1e4713
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100a0b9b51bb0c81cbbad86
-----
value: 6.0
-----
time: 2021-04-08 06:11:40.567785
-----
sender: 30819f300d06092a864886f70d010101050003818d00308189028181008e40f44f9cef42cac492d1
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100d776af73071682cb494e
-----
value: 2.0
-----
time: 2021-04-08 06:11:40.569278
-----
sender: 30819f300d06092a864886f70d010101050003818d0030818902818100a0b9b51bb0c81cbbad86384
-----
recipient: 30819f300d06092a864886f70d010101050003818d003081890281810082e40f44f9cef42cac49
-----
value: 4.0
-----
time: 2021-04-08 06:11:40.570658
-----
sender: 30819f300d06092a864886f70d010101050003818d0030818902818100d776af73071682cb494e752
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100a0b9b51bb0c81cbbad86
-----
value: 7.0
-----
time: 2021-04-08 06:11:40.572173
-----
sender: 30819f300d06092a864886f70d010101050003818d003081890281810082e40f44f9cef42cac492d1
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100a0b9b51bb0c81cbbad86
-----
value: 3.0
-----
2021-04-08 06:11:40.574026

[ ] class Block:
    def __init__(self):
        self.verified_transactions = []
        self.previous_block_hash = ""
        self.Nonce = ""

[ ] last_block_hash = ""

[ ] Dinesh = Client()

[ ] t0 = Transaction (
    "Genesis",
    Dinesh.identity,
    500.0
)

[ ] block0 = Block()

[ ] block0.previous_block_hash = None
Nonce = None

[ ] block0.verified_transactions.append(t0)

```

```

+ Code + Text | ⌂ Copy to Drive
Connect ▾ | 🖊 Editing | ^
[ ] digest = hash(block0)
last_block_hash = digest

[ ] def dump_blockchain(self):
    print("Number of blocks in the chain: " + str(len(self)))
    for x in range(len(TPCoins)):
        block_temp = TPCoins[x]
        print("block # " + str(x))
        for transaction in block_temp.verified_transactions:
            display_transaction(transaction)
            print('-----')
        print('=====') 

[ ] TPCoins = []

[ ] def dump_blockchain(self):
    print("Number of blocks in the chain: " + str(len(self)))
    for x in range(len(TPCoins)):
        block_temp = TPCoins[x]
        print("block # " + str(x))
        for transaction in block_temp.verified_transactions:
            display_transaction(transaction)
            print('-----')
        print('=====') 

[ ] TPCoins.append(block0)

[ ] dump_blockchain(TPCoins)

Number of blocks in the chain: 1
block # 0
sender: Genesis
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100bae5c5a46a1591af94c0
-----
value: 500.0
-----
time: 2021-04-08 06:16:28.464142
-----
-----
=====

[ ] def sha256(message):
    return hashlib.sha256(message.encode('ascii')).hexdigest()

[ ] def mine(message, difficulty=1):
    assert difficulty >= 1
    prefix = '1' * difficulty
    for i in range(1000):
        digest = sha256(str(hash(message)) + str(i))
        if digest.startswith(prefix):
            print("after " + str(i) + " iterations found nonce: " + digest)
            return digest

[ ] mine ("test message", 2)
'938c72d2197fa6c2294df7bc8cea6be98769aad888e3cf15558c78469394ebd'

[ ] last_transaction_index = 0

[ ] block = Block()
for i in range(3):
    temp_transaction = transactions[last_transaction_index]
    # validate transaction
    # if valid
    block.verified_transactions.append(temp_transaction)
    last_transaction_index += 1

block.previous_block_hash = last_block_hash
block.Nonce = mine(block, 2)
digest = hash(block)
TPCoins.append(block)
last_block_hash = digest

```

```

+ Code + Text | Copy to Drive Connect | Editing | ^
[ ] # Miner 2 adds a block
block = Block()

for i in range(3):
    temp_transaction = transactions[last_transaction_index]
    # validate transaction
    # if valid
    block.verified_transactions.append (temp_transaction)
    last_transaction_index += 1
block.previous_block_hash = last_block_hash
block.Nonce = mine(block, 2)
digest = hash (block)
TPCoins.append (block)
last_block_hash = digest
# Miner 3 adds a block
block = Block()

for i in range(3):
    temp_transaction = transactions[last_transaction_index]
    #display_transaction (temp_transaction)
    # validate transaction
    # if valid
    block.verified_transactions.append (temp_transaction)
    last_transaction_index += 1
block.previous_block_hash = last_block_hash
block.Nonce = mine (block, 2)
digest = hash (block)

TPCoins.append (block)
last_block_hash = digest

[ ] dump_blockchain(TPCoins)
time: 2021-04-08 06:11:40.570658
-----
-----
sender: 30819f300d06092a864886f70d010101050003818d0030818902818100d776af73071682cb494e752
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100a0b9b51bb0c81cbbad86
-----
value: 7.0
-----
time: 2021-04-08 06:11:40.572173
-----
-----
sender: 30819f300d06092a864886f70d010101050003818d003081890281810082e40f44f9cef42cac492d1
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100a0b9b51bb0c81cbbad86
-----
value: 3.0
-----
time: 2021-04-08 06:11:40.574036
-----
-----
block # 3
sender: 30819f300d06092a864886f70d010101050003818d0030818902818100a0b9b51bb0c81cbbad86384
-----
recipient: 30819f300d06092a864886f70d010101050003818d00308189028181008f5e89df98216eb2c1e4
-----
value: 8.0
-----
time: 2021-04-08 06:11:40.575310
-----
-----
sender: 30819f300d06092a864886f70d010101050003818d0030818902818100a0b9b51bb0c81cbbad86384
-----
recipient: 30819f300d06092a864886f70d010101050003818d003081890281810082e40f44f9cef42cac49
-----
value: 1.0
-----
time: 2021-04-08 06:11:40.576645
-----
-----
sender: 30819f300d06092a864886f70d010101050003818d0030818902818100d776af73071682cb494e752
-----
recipient: 30819f300d06092a864886f70d010101050003818d00308189028181008f5e89df98216eb2c1e4
-----
value: 5.0
-----
time: 2021-04-08 06:11:40.578007
-----
=====
```

Practical 4

Implement and demonstrate the use of the following in Solidity :

- (I).Varaible
- (II).Operators
- (III).Loops
- (IV).Decision Making
- (V).Strings

(I).Variable

```
pragma solidity ^0.5.0;

contract SolidityTest {
    uint storedData; // State variable

    constructor() public {
        storedData = 10;
    }

    function getResult() public view returns(uint){
        uint a = 1; // local variable
        uint b = 2;
        uint result = a + b;
        return storedData; //access the state variable
    }
}
```

Output:

The screenshot shows the Solidity Compiler interface. On the left, the 'SOLIDITY COMPILER' sidebar includes settings for the compiler version (0.5.17+commit.d196ba13), language (Solidity), EVM version (compiler default), and compiler configuration (Auto compile, Enable optimization at 200, Hide warnings). A prominent blue button at the bottom says 'Compile new one.sol'. Below it, the 'CONTRACT' section shows 'SolidityTest (new one.sol)' with options to 'Publish on Swarm' or 'Publish on IPFS'. The main area displays the Solidity code for the 'SolidityTest' contract:

```

1 pragma solidity >=0.5.0;
2 contract SolidityTest {
3     uint storedData; // State variable
4     constructor() public {
5         storedData = 10;
6     }
7     function getResult() public view returns(uint){
8         uint a = 1; // Local variable
9         uint b = 2;
10        uint result = a + b;
11        return storedData; // Access the state variable
12    }
13 }
14

```

On the right, a transaction details panel shows a transaction being sent from the address 0x563Ba6e701c568545dFc903Fcd875f56bed0C4 to the contract address 0x563Ba6e701c568545dFc903Fcd875f56bed0C4 with 3000000 gas.

The screenshot shows the 'DEPLOY & RUN TRANSACTIONS' interface. It features a 'Deploy' section with fields for value (0 wei) and a 'Deploy' button. Below this is an 'OR' section for deploying at an address or loading a contract from an address. The 'Transactions recorded' section shows a deployed contract named 'SOLIDITYTEST AT 0xD91_3913B (MEM)'. Under 'Low level interactions', there is a 'CALLDATA' section with a 'Transact' button. The main code editor area shows the same Solidity code as the previous screenshot. The transaction details panel on the right shows a transaction being sent from the address 0x563Ba6e701c568545dFc903Fcd875f56bed0C4 to the contract address 0x563Ba6e701c568545dFc903Fcd875f56bed0C4 with 3000000 gas.

```

// Solidity program to demonstrate state variables

pragma solidity ^0.5.0;

// Creating a contract

contract Solidity_var_Test {

    // Declaring a state variable

    uint8 public state_var;

    // Defining a constructor

    constructor() public {
        state_var = 16;
    }
}

```

Output:

The screenshot shows the Solidity Compiler interface with the following details:

- SOLIDITY COMPILER** tab is selected.
- COMPILER**: Version 0.5.17+commit.d19bb13
- LANGUAGE**: Solidity
- EVM VERSION**: compiler default
- COMPILER CONFIGURATION** includes Auto compile, Enable optimization (set to 200), and Hide warnings.
- Compile new one.sol** button is visible.
- CONTRACT**: Solidity_var_Test (new one.sol)
- Publish on Swarm** button is highlighted.
- Deployment Details**:
 - from: 0x5B38Da6a701c568545dCfc803fc8879f5b6ed04
 - to: SolidityTest.(constructor)
 - gas: 3000000 gas

SOLIDITY COMPILER

COMPILER: 0.5.17+commit.d19bba13

LANGUAGE: Solidity

EVM VERSION: compiler default

COMPILER CONFIGURATION: Auto compile, Enable optimization (200), Hide warnings

Contract: Solidity_var_Test (Variable.sol)

Actions: Publish on Swarm, Publish on Ipfs, Compilation Details

ABI **Bytecode**

Code (Variable.sol):

```
// Solidity program to demonstrate state variables
pragma solidity ^0.5.0;

// Creating a contract
contract Solidity_var_Test {
    // Declaring a state variable
    uint8 public state_var;
}

// Defining a constructor
constructor() public {
    state_var = 16;
}
```

DEPLOY & RUN TRANSACTIONS

ENVIRONMENT: JavaScript VM

ACCOUNT: 0x5B3...eddC4 (99.99999999999999)

GAS LIMIT: 3000000

VALUE: 0 wei

CONTRACT: Solidity_var_Test - Variable.sol

Actions: Deploy, Publish to IPFS

OR

At Address: Load contract from Address

Transactions recorded: 1

Deployed Contracts: SOLIDITY_VAR_TEST AT 0xD91...39138

state_var: 0: uint8: 16

Low level interactions: CALLDATA

Code (Variable.sol):

```
// Solidity program to demonstrate state variables
pragma solidity ^0.5.0;

// Creating a contract
contract Solidity_var_Test {
    // Declaring a state variable
    uint8 public state_var;
}

// Defining a constructor
constructor() public {
    state_var = 16;
}
```

```

// Solidity program to show Global variables

pragma solidity ^0.5.0;

// Creating a contract

contract Test {

// Defining a variable

address public admin;

// Creating a constructor to

// use Global variable

constructor() public {

admin = msg.sender;

}

}

```

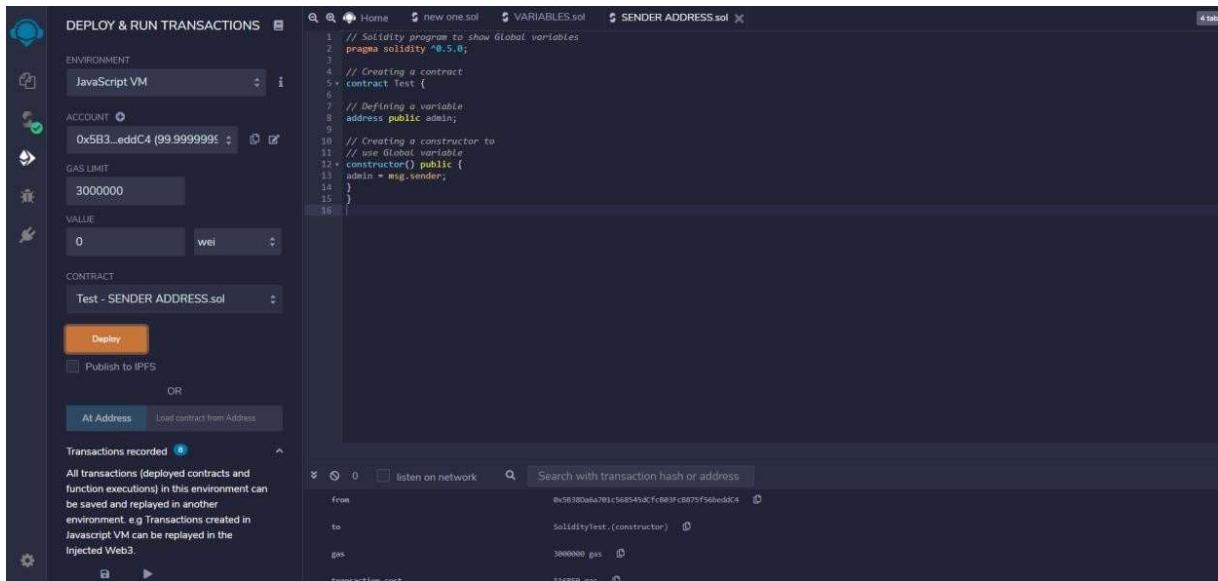
Output:

The screenshot shows the Solidity Compiler interface. On the left, the compiler configuration is set to version 0.5.17+commit.d19bba13, language Solidity, and EVM version compiler default. A button labeled "Compile SENDER ADDRESS.sol" is highlighted in blue. On the right, the Solidity code is displayed in a code editor. Below the code editor, a transaction details panel shows a transaction being sent from address 0x5B38Da6a701c5e0545dC7c883f8875f58bed4 to the contract SolidityTest.(constructor) with a gas limit of 3000000.

```

1 // Solidity program to show Global variables
2 pragma solidity ^0.5.0;
3
4 // Creating a contract
5 contract Test {
6
7 // Defining a variable
8 address public admin;
9
10 // Creating a constructor to
11 // use Global variable
12 constructor() public {
13 admin = msg.sender;
14 }
15
16

```



(II).Operators

// Solidity contract to demonstrate Arithematic Operator

```
pragma solidity ^0.5.0;
```

// Creating a contract

```
contract SolidityTest {
```

// Initializing variables

```
uint16 public a = 20;
```

```
uint16 public b = 10;
```

// Initializing a variable with sum

```
uint public sum = a + b;
```

// Initializing a variable with the difference

```
uint public diff = a - b;
```

```

// Initializing a variable with product
uint public mul = a * b;

// Initializing a variable with quotient
uint public div = a / b;

// Initializing a variable with modulus
uint public mod = a % b;

// Initializing a variable decrement value
uint public dec = --b;

// Initializing a variable with increment value
uint public inc = ++a;
}

```

Output:

The screenshot shows the Solidity Compiler interface with the following details:

- SOLIDITY COMPILER** tab is selected.
- COMPILER**: 0.5.17+commit.d19bba13
- LANGUAGE**: Solidity
- EVM VERSION**: compiler default
- COMPILER CONFIGURATION** includes Auto compile, Enable optimization (set to 200), and Hide warnings.
- Contract**: SolidityTest (arithmetic.sol)
- Buttons**: Publish on Swarm, Publish on Ipfs, Compilation Details.
- Compilation Details** section shows:
 - from: 0x5b380a6a701c568545dCfC083FcB75f56beadC4
 - to: SolidityTest.(constructor)
 - gas: 3000000 gas

DEPLOY & RUN TRANSACTIONS

ENVIRONMENT: JavaScript VM

ACCOUNT: 0x5B3...eddC4 (99.99999999999999)

GAS LIMIT: 3000000

VALUE: 0 wei

CONTRACT: SolidityTest - arithematic.sol

Deploy

Publish to IPFS

OR

At Address Load contract from Address

Transactions recorded 9

```

1 // Solidity contract to demonstrate Arithmetic Operator
2 pragma solidity ^0.5.0;
3
4 // Creating a contract
5 contract SolidityTest {
6
7 // Initializing variables
8 uint16 public a = 20;
9 uint16 public b = 10;
10 // Initializing a variable with sum
11 uint public sum = a + b;
12
13 // Initializing a variable with the difference
14 uint public diff = a - b;
15
16 // Initializing a variable with product
17 uint public mul = a * b;
18
19 // Initializing a variable with quotient
20 uint public div = a / b;
21
22 // Initializing a variable with modulus
23 uint public mod = a % b;
24
25 // Initializing a variable decrement value
26 uint public dec = --b;
27
28 // Initializing a variable with increment value
29 uint public inc = ++a;
30 }
```

listen on network

Search with transaction hash or address

from: 0x5B30a6a701c560545dCfc803fc8875f56beddC4

to: SolidityTest.(constructor)

gas: 3000000 gas

DEPLOY & RUN TRANSACTIONS

SOLIDITYTEST AT 0xD2A...FD005 (MEMORY)

a

b

dec

diff

div

inc

mod

mul

sum

Low level interactions

CALDATA

Transact

transaction cost: 116059 gas

listen on network

Search with transaction hash or address

from: 0x5B30a6a701c560545dCfc803fc8875f56beddC4

to: SolidityTest.(constructor)

gas: 3000000 gas

(IV).Decision Making

// Solidity program to demonstrate the use of 'if statement'

```
pragma solidity ^0.5.0;
```

```
// Creating a contract
```

```
contract Types {
```

```
// Declaring state variable
```

```
uint i = 10;
```

```
// Defining function to demonstrate use of 'if statement'
```

```
function decision_making() public view returns(bool){
```

```
if(i<10){
```

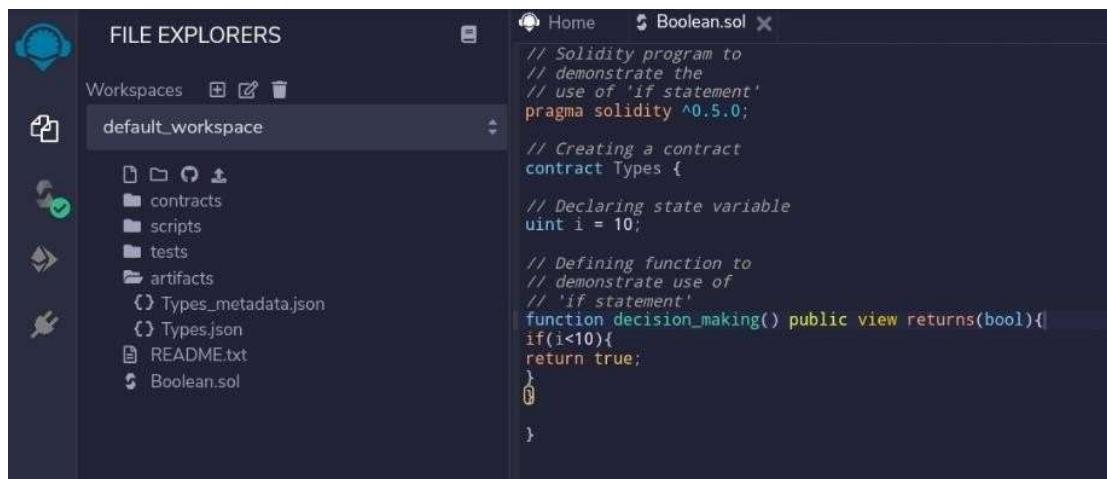
```
return true;
```

```
}
```

```
}
```

```
}
```

Output:



The screenshot shows the Solidity IDE interface. On the left, the 'FILE EXPLORERS' panel displays a 'default_workspace' folder containing 'contracts', 'scripts', 'tests', 'artifacts', and files like 'Types_metadata.json', 'Types.json', 'README.txt', and 'Boolean.sol'. The 'Boolean.sol' file is open in the main editor window. The code is as follows:

```
// Solidity program to
// demonstrate the
// use of 'if statement'
pragma solidity ^0.5.0;

// Creating a contract
contract Types {

    // Declaring state variable
    uint i = 10;

    // Defining function to
    // demonstrate use of
    // 'if statement'
    function decision_making() public view returns(bool){
        if(i<10){
            return true;
        }
    }
}
```

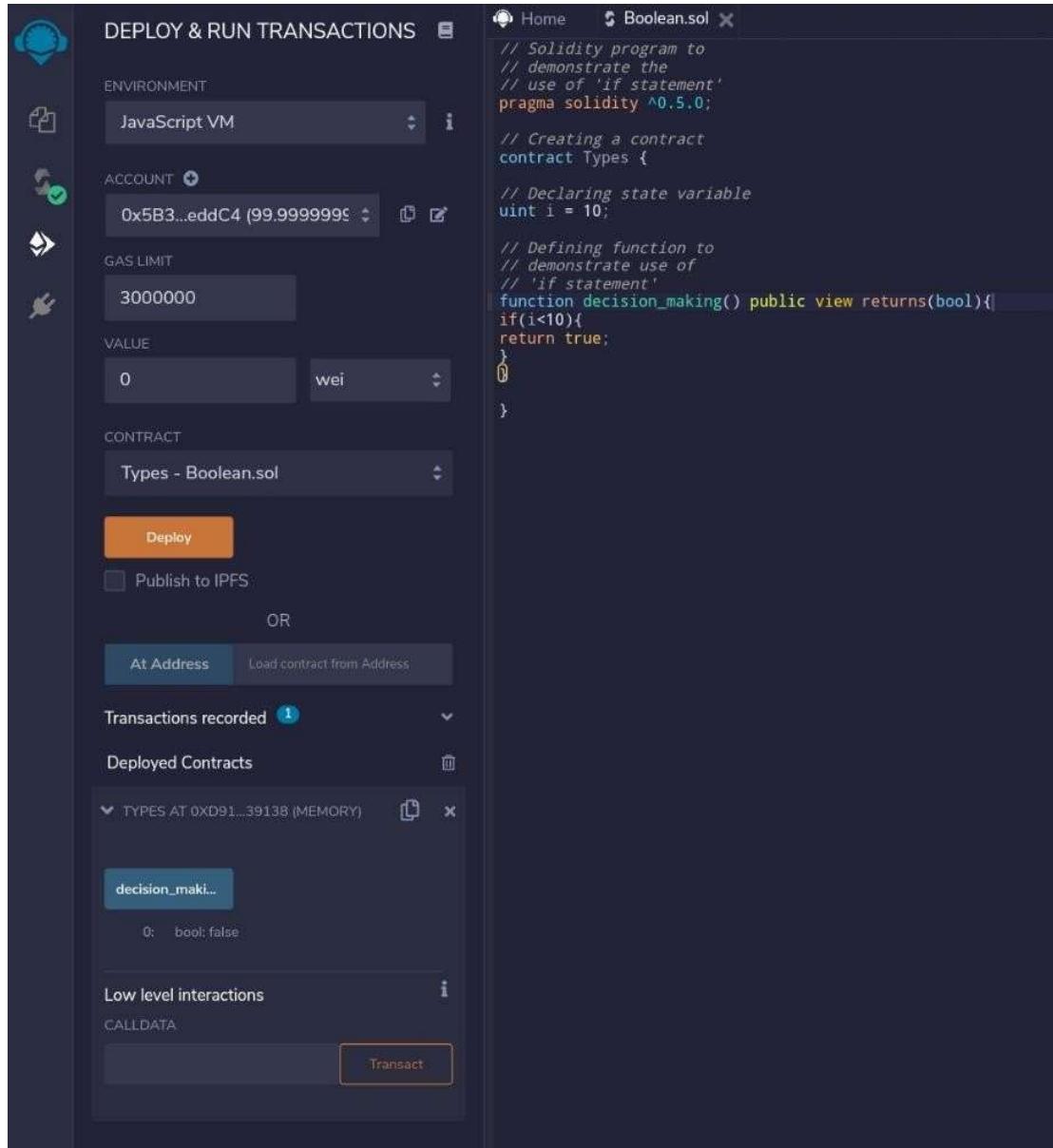
The screenshot shows the Solidity Compiler interface. On the left, there's a sidebar with various icons. The main area is titled "SOLIDITY COMPILER". It includes sections for "COMPILER" (set to 0.5.17+commit.d19bba13), "LANGUAGE" (Solidity), "EVM VERSION" (compiler default), and "COMPILER CONFIGURATION" (Auto compile, Enable optimization set to 200, Hide warnings). A prominent blue button at the bottom says "Compile Boolean.sol". To the right, the file "Boolean.sol" is open, displaying the following Solidity code:

```
// Solidity program to
// demonstrate the
// use of 'if statement'
pragma solidity ^0.5.0;

// Creating a contract
contract Types {

    // Declaring state variable
    uint i = 10;

    // Defining function to
    // demonstrate use of
    // 'if statement'
    function decision_making() public view returns(bool){
        if(i<10){
            return true;
        }
    }
}
```



// Solidity program to demonstrate the use of 'if...else' statement

```
pragma solidity ^0.5.0;
```

// Creating a contract

contract Types {

```
// Declaring state variables
```

```
uint i = 10;  
bool even;  
  
// Defining function to  
// demonstrate the use of  
// 'if...else statement'  
  
function decision_making() public {  
    if(i%2 == 0){  
        even = true;  
    }  
    else{  
        even = false;  
    }  
}  
  
function getresult() public view returns(bool)  
{  
    return even;  
}  
  
}
```

Output:

The screenshot shows the Solidity IDE interface. On the left is the 'FILE EXPLORERS' sidebar, which lists workspaces and files. The 'default_workspace' is selected, containing contracts, scripts, tests, artifacts, and several JSON files. In the center-right is the main workspace area, showing a code editor with the file 'If_else.sol'. The code is a Solidity program demonstrating if-else statements:

```
1 // Solidity program to
2 // demonstrate the use of
3 // 'if...else' statement
4 pragma solidity ^0.5.0;
5
6 // Creating a contract
7 contract Types {
8
9 // Declaring state variables
10 uint i = 10;
11 bool even;
12
13 // Defining function to
14 // demonstrate the use of
15 // 'if...else statement'
16 function decision_making() public {
17 if(i%2 == 0){
18 even = true;
19 }
20 else{
21 even = false;
22 }
23
24 }
25 function getResult() public view returns(bool)
26 {
27 return even;
28 }
29
30 }
```

The screenshot shows the Solidity Compiler interface. On the left is the 'SOLIDITY COMPILER' sidebar, which includes compiler settings (version 0.5.17+commit.d19bba13), language (Solidity), EVM version (compiler default), and compiler configuration (Auto compile, Enable optimization set to 200, Hide warnings). Below this is a large blue button labeled 'Compile If_else.sol'. In the center-right is the main workspace area, showing the same 'If_else.sol' code as the previous screenshot. The code is identical to the one in the File Explorer.

The screenshot shows the Truffle UI interface for deploying and running transactions. On the left, there's a sidebar with various icons. The main area has tabs for 'DEPLOY & RUN TRANSACTIONS' and 'CONTRACTS'.

ENVIRONMENT: JavaScript VM

ACCOUNT: 0x5B3...eddC4 (99.99999999999999)

GAS LIMIT: 3000000

VALUE: 0 wei

CONTRACT: Types - If_else.sol

Deploy button

Publish to IPFS checkbox

OR

At Address (selected) or **Load contract from Address**

Transactions recorded: 7

Deployed Contracts:

- TYPES AT 0xD91...39138 (MEMORY)
- TEST AT 0xD8B...33FA8 (MEMORY)
- TYPES AT 0xDA0...42B53 (MEMORY)

decision_making button

getresult button

0: bool: true

Low level interactions

CALLDATA: [Input field] **Transact** button

Code Area:

```

1 // Solidity program to
2 // demonstrate the use of
3 // 'if...else' statement
4 pragma solidity ^0.5.0;
5
6 // Creating a contract
7 contract Types {
8
9     // Declaring state variables
10    uint i = 10;
11    bool even;
12
13    // Defining function to
14    // demonstrate the use of
15    // 'if...else' statement
16    function decision_making() public {
17        if(i%2 == 0){
18            even = true;
19        }
20        else{
21            even = false;
22        }
23    }
24
25    function getresult() public view returns(bool)
26    {
27        return even;
28    }
29
30

```

(III) Strings

```
// Solidity program to demonstrate
// how to create a contract
pragma solidity ^0.4.23;

// Creating a contract
contract Test {

    // Declaring variable
    string str;

    // Defining a constructor
    constructor(string str_in) {
        str = str_in;
    }

    // Defining a function to
    // return value of variable 'str'
    function str_out() public view returns(string memory) {
        return str;
    }
}
```

Output

The screenshot shows the Solidity Compiler interface. On the left, there are configuration options for the compiler (version 0.4.26), language (Solidity), EVM version (compiler default), and compiler configuration (Auto compile, Enable optimization). A large text area contains the Solidity code for the 'Test' contract. Below the code, a blue button labeled 'Compile string.sol' is visible. On the right, there's a section for publishing the contract to Swarm or IPFS, and a 'Compilation Details' panel showing transaction details like from, to, and gas.

```
// Solidity program to demonstrate
// how to create a contract
pragma solidity ^0.4.23;

// Creating a contract
contract Test {
    // Declaring variable
    string str;

    // Defining a constructor
    constructor(string str_in) {
        str = str_in;
    }

    // Defining a function to
    // return value of variable 'str'
    function str_out() public view returns(string memory) {
        return str;
    }
}
```

The screenshot shows the Deploy & Run Transactions interface. It displays the same Solidity code as the compiler screen. On the left, there are tabs for different contracts: 'SOLIDITY_VAR_TEST AT 0X358_D5EE3' (selected), 'TEST AT 0X9D7_B5E9 (MEMORY)', and 'TEST AT 0XDDA_54B2D (MEMORY)'. Each tab has a 'Low level interactions' section with a 'CALLDATA' input field. The first tab has a 'state_var' button. The bottom section shows the transaction details for deploying the contract: from, to, and gas. The 'to' field is set to 'SolidityTest.(constructor)'.

```
// Solidity program to demonstrate
// how to create a contract
pragma solidity ^0.4.23;

// Creating a contract
contract Test {
    // Declaring variable
    string str;

    // Defining a constructor
    constructor(string str_in) {
        str = str_in;
    }

    // Defining a function to
    // return value of variable 'str'
    function str_out() public view returns(string memory) {
        return str;
    }
}
```

Practical 5

Implement and demonstrate the use of the following in Solidity :

- (I).Arrays
- (II).Enums
- (III).Structs
- (IV).Mappings
- (V).Coversations
- (VI).Ether Units
- (VII).Special Varaibles

(I).Arrays

```
// Solidity program to demonstrate  
// creating a fixed-size array  
pragma solidity ^0.5.0;  
  
// Creating a contract  
contract Types {  
  
    // Declaring state variables  
    // of type array  
    uint[6] data1;  
    int[5] data;  
  
    // Defining function to add  
    // values to an array  
    function array_example() public returns (int[5] memory, uint[6] memory){  
        data = [int(50), -63, 77, -28, 90];  
    }  
}
```

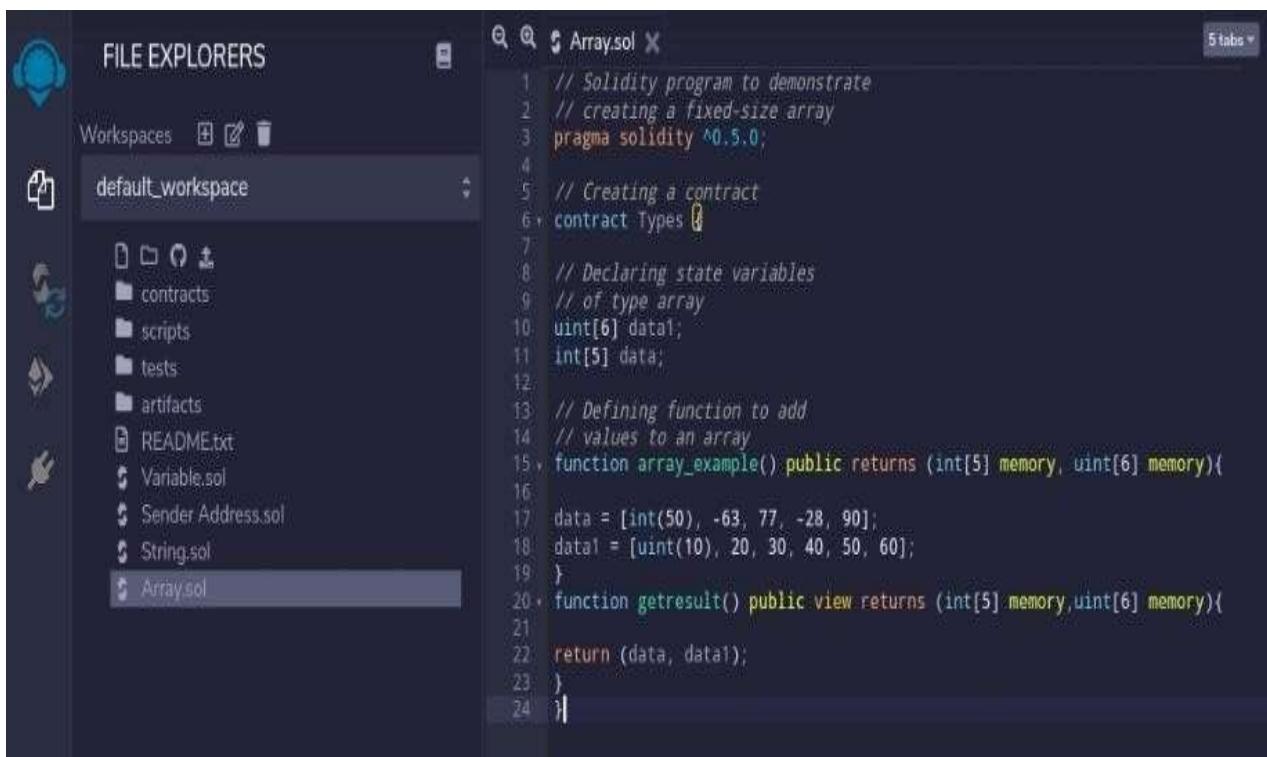
```

data1 = [uint(10), 20, 30, 40, 50, 60];
}

function getresult() public view returns (int[5] memory,uint[6] memory){
    return (data, data1);
}
}

```

Output:



The screenshot shows the Solidity IDE interface. On the left, the FILE EXPLORER panel displays a workspace named "default_workspace" containing several files and folders: contracts, scripts, tests, artifacts, README.txt, Variable.sol, SenderAddress.sol, String.sol, and Array.sol. The Array.sol file is currently selected and highlighted in blue. On the right, the code editor window shows the Solidity code for the Array contract. The code includes comments explaining the creation of a fixed-size array, defining state variables, and implementing functions to add values to an array and retrieve the result.

```

// Solidity program to demonstrate
// creating a fixed-size array
pragma solidity ^0.5.0;

// Creating a contract
contract Types {
    // Declaring state variables
    // of type array
    uint[6] data1;
    int[5] data;

    // Defining function to add
    // values to an array
    function array_example() public returns (int[5] memory, uint[6] memory){
        data = [int(50), -63, 77, -28, 90];
        data1 = [uint(10), 20, 30, 40, 50, 60];
    }
    function getresult() public view returns (int[5] memory,uint[6] memory){
        return (data, data1);
    }
}

```

The image shows the Solidity Compiler interface. On the left, there's a sidebar with various icons. The main area is titled "SOLIDITY COMPILER". It includes settings for "COMPILER" (set to 0.5.17+commit.d19bba13), "LANGUAGE" (Solidity), "EVM VERSION" (compiler default), and "COMPILER CONFIGURATION" (Auto compile, Enable optimization set to 200). Below these are buttons for "Hide warnings" and a large blue "Compile Array.sol" button. To the right, the code editor displays "Array.sol" with the following content:

```
1 // Solidity program to demonstrate
2 // creating a fixed-size array
3 pragma solidity ^0.5.0;
4
5 // Creating a contract
6 contract Types {
7
8 // Declaring state variables
9 // of type array
10 uint[6] data1;
11 int[5] data;
12
13 // Defining function to add
14 // values to an array
15 function array_example() public returns (int[5] memory, uint[6] memory){
16
17 data = [int(50), -63, 77, -28, 90];
18 data1 = [uint(10), 20, 30, 40, 50, 60];
19 }
20 function getresult() public view returns (int[5] memory,uint[6] memory){
21
22 return (data, data1);
23 }
}
```

Below the code editor, under "CONTRACT", it says "Types (Array.sol)". There are three buttons: "Publish on Swarm" (with a Swarm icon), "Publish on Ipfs" (with an IPFS icon), and "Compilation Details". At the bottom, there are links for "ABI" and "Bytecode".

The screenshot shows the Truffle IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel is open, displaying deployment settings: Environment (JavaScript VM), Account (0x5B3...eddC4), Gas Limit (3000000), Value (0 wei), and Contract (Types - Array.sol). A prominent orange 'Deploy' button is at the bottom. Below it, there's a checkbox for 'Publish to IPFS'. The right side of the interface shows the Solidity code for 'Array.sol' and its deployment details.

```

// Solidity program to demonstrate
// creating a fixed-size array
pragma solidity ^0.5.0;

// Creating a contract
contract Types {
    // Declaring state variables
    // of type array
    uint[6] data;
    int[5] data1;

    // Defining function to add
    // values to an array
    function array_example() public returns (int[5] memory, uint[6] memory) {
        data = [int(50), -63, 77, -28, 90];
        data1 = [uint(10), 20, 30, 40, 50, 60];
    }

    function getresult() public view returns (int[5] memory, uint[6] memory) {
        return (data, data1);
    }
}

```

Deployed Contracts:

- array_example**: Returns Int256[5]: 50,-63,77,-28,90 and UInt256[6]: 10,20,30,40,50,60.

Low level interactions:

CALLDATA:

(III).Structs

```

pragma solidity ^0.5.0;

contract test {

    struct Book {
        string title;
        string author;
        uint book_id;
    }

    Book book;
}

```

```

function setBook() public {
    book = Book('Learn Java', 'TP', 1);
}

function getBookId() public view returns (uint) {
    return book.book_id;
}

```

Output:

The screenshot shows the Solidity Compiler interface with the following details:

- Compiler:** 0.5.17+commit.d19bba13
- Language:** Solidity
- EVM Version:** compiler default
- Compiler Configuration:**
 - Auto compile
 - Enable optimization: 200
 - Hide warnings
- Contract:** test (books.sol)
- Buttons:** Publish on Swarm, Publish on Ipfs, Compilation Details
- Contract Details:**
 - From: 0x5B380a6a701c568545dFc883Fc8875f5e0ed04
 - To: SolidityTest.(constructor)
 - Gas: 3000000 gas

The screenshot shows the Truffle IDE interface. On the left, there's a sidebar with icons for deployment, running transactions, and low-level interactions. The main area displays a Solidity code editor with the following code:

```

1 pragma solidity ^0.5.0;
2+ contract test {
3+   struct Book {
4     string title;
5     string author;
6     uint book_id;
7   }
8   Book book;
9+   function setBook() public {
10    book = Book('Learn Java', 'IP', 1);
11  }
12+   function getBookId() public view returns (uint) {
13    return book.book_id;
14  }
15}
16

```

Below the code editor, there are sections for "Low level interactions" and "CALLDATA". Under "Low level interactions", there are buttons for "setBook" and "getBookId". Under "CALLDATA", there is a text input field and a "Transact" button.

On the right side, there is a table titled "transaction cost" with the following data:

	0	1
from	0x5830a6a70c588545dCfc883f8875f56beddCA	
to	SolidityTest.(constructor)	
gas	3000000 gas	
transaction cost	116859 gas	

(IV).Mappings

```

pragma solidity ^0.5.0;

contract LedgerBalance {
  mapping(address => uint) balance;

  function updateBalance() public returns(uint) {
    balance[msg.sender]=20;
    return balance[msg.sender];
  }
}

```

Output:

The image shows the Solidity Compiler interface. On the left, there's a sidebar with icons for file operations (New, Open, Save, Find, Copy, Paste, Delete) and tabs for 'Function_program.sol' and 'Mapping.sol'. The main area is titled 'SOLIDITY COMPILER'.

COMPILER: 0.5.17+commit.d19bba13

Include nightly builds

LANGUAGE: Solidity

EVM VERSION: compiler default

COMPILER CONFIGURATION:

- Auto compile
- Enable optimization 200
- Hide warnings

Compile Mapping.sol (button)

CONTRACT: LedgerBalance (Mapping.sol)

Publish Options:

- Publish on Swarm** (cloud icon)
- Publish on Ipfs** (ipfs icon)
- Compilation Details**

ABI **Bytecode**

```
pragma solidity ^0.5.0;

contract LedgerBalance {
    mapping(address => uint) balance;

    function updateBalance() public returns(uint)
    balance[msg.sender]=20;
    return balance[msg.sender];
}
```

The screenshot shows a web-based Ethereum development environment. On the left, there's a sidebar with icons for account management, gas limit, value, and network selection. The main area is divided into two sections: "DEPLOY & RUN TRANSACTIONS" on the left and "Function_program.sol" code editor on the right.

DEPLOY & RUN TRANSACTIONS

- ENVIRONMENT:** JavaScript VM
- ACCOUNT:** 0x5B3...eddC4 (99.9999995)
- GAS LIMIT:** 3000000
- VALUE:** 0 wei
- CONTRACT:** LedgerBalance - Mapping.sol
- Buttons:** Deploy, Publish to IPFS
- Transactions recorded:** 15
- Deployed Contracts:**
 - > TYPES AT 0XD91...39138 (MEMORY)
 - > TEST AT 0XDBB...33FA8 (MEMORY)
 - > TYPES AT 0XDA0...42B53 (MEMORY)
 - > TEST AT 0X9D7...B5E99 (MEMORY)
 - > LEDGERBALANCE AT 0XD2A...FD005 ()
- Low level interactions:** CALLDATA, Transact

Function_program.sol

```
pragma solidity ^0.5.0;
contract LedgerBalance {
    mapping(address => uint) balance;
    function updateBalance() public returns(uint) {
        balance[msg.sender]=20;
        return balance[msg.sender];
    }
}
```

```

pragma solidity ^0.5.0;

contract LedgerBalance {
    mapping(address => string) name;

    function updateBalance() public returns(string memory){
        name[msg.sender] = "Dip";
        return name[msg.sender];
    }

    function printsender() public view returns(address) {
        return msg.sender;
    }
}

```

Output:

The screenshot shows the Solidity IDE interface. On the left, the 'FILE EXPLORERS' panel displays a workspace named 'default_workspace' containing directories like 'contracts', 'scripts', 'tests', and 'artifacts', along with files such as 'README.txt', 'Boolean.sol', 'Book.sol', 'If_else.sol', 'Function_program.sol', 'Mapping.sol', and 'Mapping_string.sol'. The 'Mapping_string.sol' file is currently selected and highlighted in the list. On the right, the main code editor window shows the Solidity code for the 'LedgerBalance' contract, which includes functions for updating the balance and returning the sender's address.

```

1 pragma solidity ^0.5.0;
2
3
4 contract LedgerBalance {
5     mapping(address => string) name;
6
7
8     function updateBalance() public returns(string memory){
9         name[msg.sender] = "Dip";
10    return name[msg.sender];
11 }
12 function printsender() public view returns(address) {
13     return msg.sender;
14 }
15
16 }

```

The screenshot shows the Solidity Compiler interface. On the left, there's a sidebar with icons for file operations, a search bar, and sections for Compiler (version 0.5.17+commit.d19bba13), Language (Solidity), EVM Version (compiler default), Compiler Configuration (Auto compile, Enable optimization set to 200, Hide warnings), and a prominent blue button labeled "Compile Mapping_string.sol". Below these are buttons for "Publish on Swarm" and "Publish on Ipfs", along with "Compilation Details" and ABI/Bytecode links.

```
pragma solidity ^0.5.0;

contract LedgerBalance {
    mapping(address => string) name;

    function updateBalance() public returns(string memory) {
        name[msg.sender] = "Dip";
        return name[msg.sender];
    }

    function printsender() public view returns(address) {
        return msg.sender;
    }
}
```

This screenshot displays an interactive interface for testing the LedgerBalance contract. It lists several tests: "TEST AT 0XD8B...33FA8 (MEMORY)", "TYPES AT 0XA0...42B53 (MEMORY)", "TEST AT 0XD7...B5E99 (MEMORY)", "LEDGERBALANCE AT 0XD2A...FD005 (MEMORY)", and "LEDGERBALANCE AT 0X332...D4B6D (MEMORY)". The last item is expanded, showing two buttons: "updateBalance" (orange) and "printsender" (teal). Below these buttons, it shows the address: 0x5B38Da6a701c568545dCfcB 03FcB875f56beddC4. At the bottom, there's a section for "Low level interactions" with a "Transact" button.

Practical 6

Implement and demonstrate the use of the following in Solidity :

- (I).Functions
- (II).View Functions
- (III).Pure Functions
- (IV).Fallback Functions
- (V).Function Overloading
- (VI).Mathematical Functions
- (VII).Cryptographic Functions

(I).Functions

```
pragma solidity ^0.5.0;

contract SolidityTest {

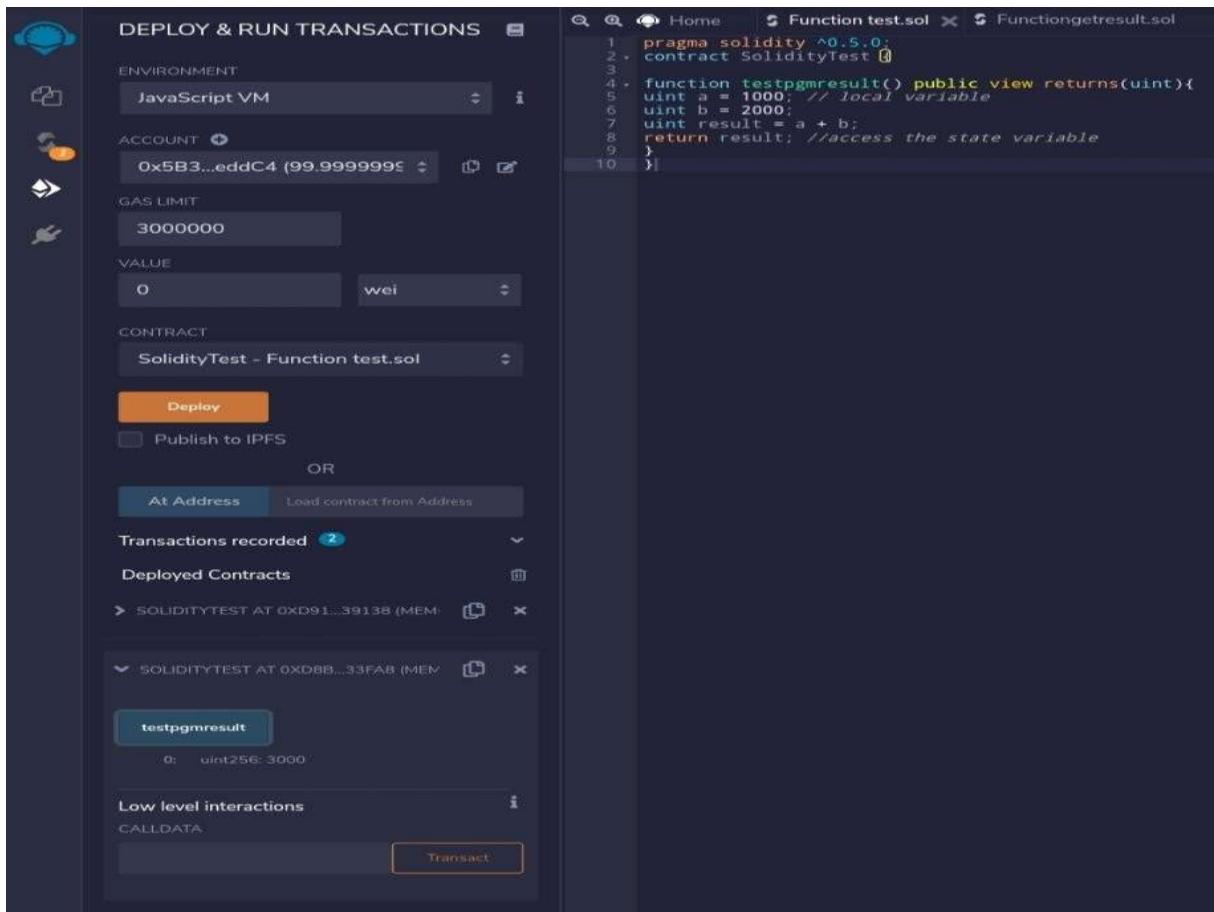
    function testpgmresult() public view returns(uint) {
        uint a = 1000; // local variable
        uint b = 2000;
        uint result = a + b;
        return result; //access the state variable
    }
}
```

Output:

The screenshot shows the Solidity Compiler interface. On the left, there's a sidebar with various icons and settings. The main area displays a Solidity code editor with the following content:

```
pragma solidity ^0.5.0;
contract SolidityTest {
    function testpgmresult() public view returns(uint){
        uint a = 1000; // local variable
        uint b = 2000;
        uint result = a + b;
        return result; //access the state variable
    }
}
```

Below the code editor, there's a large blue button labeled "Compile Function test.sol". Further down, under the heading "CONTRACT", it shows "SolidityTest (Function test.sol)". There are three buttons: "Publish on Swarm" (with a Swarm icon), "Publish on Ipfs" (with an IPFS icon), and "Compilation Details". At the bottom of the sidebar, there are links for "ABI" and "Bytecode".



(II).View Functions

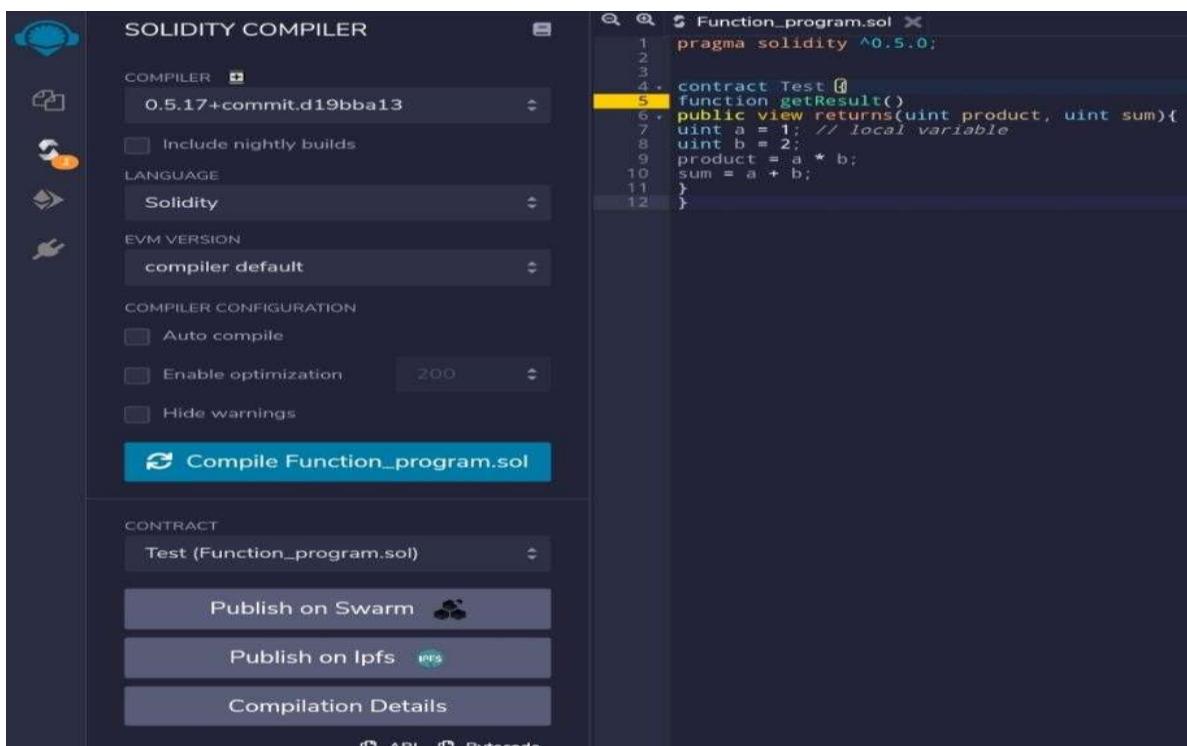
```

pragma solidity ^0.5.0;

contract Test {
    function getResult() public view returns(uint product, uint sum){
        uint a = 1; // local variable
        uint b = 2;
        product = a * b;
        sum = a + b;
    }
}

```

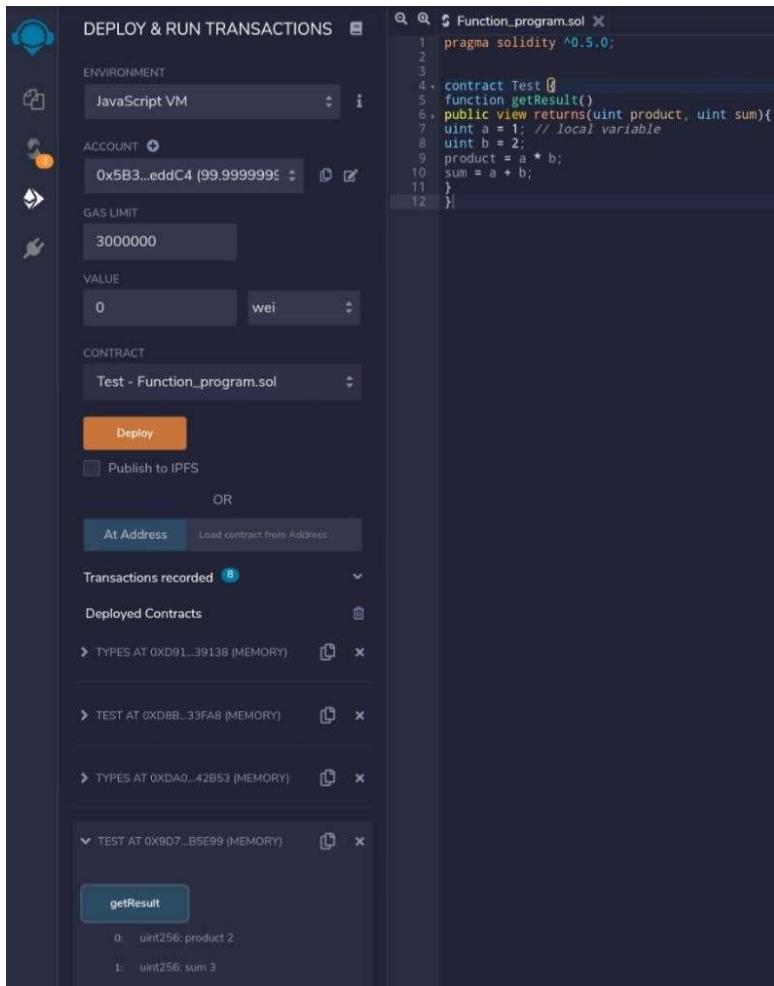
Output:



The screenshot shows the Solidity Compiler interface. On the left, there's a sidebar with icons for Ethereum, Truffle, and Hardhat. The main area is titled "SOLIDITY COMPILER". It has dropdown menus for "COMPILER" (set to "0.5.17+commit.d19bba13"), "LANGUAGE" (set to "Solidity"), and "EVM VERSION" (set to "compiler default"). Under "COMPILER CONFIGURATION", there are checkboxes for "Auto compile", "Enable optimization" (set to 200), and "Hide warnings". A large blue button at the bottom left says "Compile Function_program.sol". On the right, there's a code editor window titled "Function_program.sol" with the following Solidity code:

```
pragma solidity ^0.5.0;

contract Test {
    function getResult() public view returns(uint product, uint sum){
        uint a = 1; // local variable
        uint b = 2;
        product = a * b;
        sum = a + b;
    }
}
```



(III).Pure Functions

```
pragma solidity ^0.5.0;
```

```
contract C {
```

```
//private state variable
```

```
uint private data;
```

```
//public state variable
```

```
uint public info;
```

```
//constructor
```

```

constructor() public {
    info = 10;
}

//private function

function increment(uint a) private pure returns(uint) { return a + 1; }

//public function

function updateData(uint a) public { data = a; }

function getData() public view returns(uint) { return data; }

function compute(uint a, uint b) internal pure returns (uint) { return a + b; }

}

//Derived Contract

contract E is C {
    uint private result;
    C private c;

    constructor() public {
        c = new C();
    }

    function getComputedResult() public {
        result = compute(3, 5);
    }

    function getResult() public view returns(uint) { return result; }

    function getData() public view returns(uint) { return c.info(); }

}

```

Output:

SOLIDITY COMPILER

```

1 pragma solidity ^0.5.0;
2 contract C {
3 //private state variable
4 uint private data;
5
6 //public state variable
7 uint public info;
8
9 //constructor
10 constructor() public {
11 info = 10;
12 }
13 //private function
14 function increment(uint a) private pure returns(uint) { return a + 1; }
15
16 //public function
17 function updateData(uint a) public { data = a; }
18 function getData() public view returns(uint) { return data; }
19 function compute(uint a, uint b) internal pure returns (uint) { return a + b; }
20 }
21
22 //Derived Contract
23 contract E is C {
24 uint private result;
25 C private c;
26
27 constructor() public {
28 c = new C();
29 }
30 function getComputedResult() public {
31 result = compute(3, 5);
32 }
33 function getResult() public view returns(uint) { return result; }
34 function getData() public view returns(uint) { return c.info(); }
35 }

```

CONTRACT
C (calling function external.sol)

Publish on Swarm

Publish on Ipfs

Compilation Details

DEPLOY & RUN TRANSACTIONS

TEST AT 0x0FC_9A836 (MEMORY)

- setBook**
- getBookId**
0: uint256: 1
- Low level interactions**
- CALldata**
- Transact**

CAT 0xAE0_968BB (MEMORY)

- updateData**
25
- transact**
- getData**
0: uint256: 25
- info**
0: uint256: 10

from 0x5830d6a79c568545dcfc803fc8075f56bedd4

to SolidityTest.(constructor)

gas 3000000 gas

Search with transaction hash or address

(V).Function Overloading

```
pragma solidity ^0.5.0;

contract Test {

    function getSum(uint a, uint b) public pure returns(uint){
        return a + b;
    }

    function getSum(uint a, uint b, uint c) public pure returns(uint){
        return a + b + c;
    }

    function callSumWithTwoArguments() public pure returns(uint){
        return getSum(1,2);
    }

    function callSumWithThreeArguments() public pure returns(uint){
        return getSum(1,2,3);
    }
}
```

Output:

The screenshot shows the Solidity Compiler interface with the following details:

- SOLIDITY COMPILER** tab is selected.
- COMPILER**: Version 0.5.17-commit.d19bbal3.
- LANGUAGE**: Solidity.
- EVM VERSION**: compiler default.
- COMPILE CONFIGURATION**: Auto compile, Enable optimization (200), Hide warnings.
- Contract**: Test (overloading.sol).
- Publish on Swarm** and **Publish on IPFS** buttons are present.
- Compilation Details** button.
- Transaction Details**:
 - From: 0x5d380fe6a701c568545dCf883Fc8675F56bedaC4
 - To: SolidityTest.(constructor)
 - Gas: 3000000 gas
- Search bar**: Search with transaction hash or address.

```

1 pragma solidity ^0.5.0;
2 contract Test {
3     function getSum(uint a, uint b) public pure returns(uint){
4         return a + b;
5     }
6     function getSum(uint a, uint b, uint c) public pure returns(uint){
7         return a + b + c;
8     }
9     function callSumWithTwoArguments() public pure returns(uint){
10        return getSum(1,2);
11    }
12    function callSumWithThreeArguments() public pure returns(uint){
13        return getSum(1,2,3);
14    }
15 }
16

```

(VI).Mathematical Functions

```

pragma solidity ^0.5.0;

contract Test {

function callAddMod() public pure returns(uint){
return addmod(4, 5, 3);
}

function callMulMod() public pure returns(uint){
return mulmod(4, 5, 3);
}
}

```

Output:

SOLIDITY COMPILER

COMPILER: 0.5.17+commit.d19bba13

LANGUAGE: Solidity

EVM VERSION: compiler default

COMPILE CONFIGURATION: Auto compile, Enable optimization (200), Hide warnings

Compile Calladdmod.sol

CONTRACT: Test (Calladdmod.sol)

Publish on Swarm, Publish on IPFS, Compilation Details

ABI, Bytecode

```
pragma solidity ^0.5.0;
contract Test {
    function callAddMod() public pure returns(uint){
        return addmod(4, 5, 3);
    }
    function callMulMod() public pure returns(uint){
        return mulmod(4, 5, 3);
    }
}
```

0 wei

CONTRACT: Test - Calladdmod.sol

Deploy

Publish to IPFS

OR

At Address, Load contract from Address

Transactions recorded: 2

Deployed Contracts:

TEST AT 0xD91...39138 (MEMORY)

TEST AT 0xD8B...33FAB (MEMORY)

callAddMod: 0

callMulMod: 2

Low level interactions: CALLDATA

Transact

The screenshot shows the deployment and interaction with the 'Test' contract. It displays two deployed contracts: 'TEST AT 0xD91...39138 (MEMORY)' and 'TEST AT 0xD8B...33FAB (MEMORY)'. Under the first contract, the 'callAddMod' function is called and returns 0. Under the second contract, the 'callMulMod' function is called and returns 2. The 'Low level interactions' section shows a 'CALLDATA' button and a 'Transact' button.

(VII).Cryptographic Functions

```
pragma solidity ^0.5.0;

contract Test {

function callsha256() public pure returns( bytes32 result){
return sha256("ronaldo");

}

function callkeccak256() public pure returns( bytes32 result){
return keccak256("ronaldo");

}
}
```

Output:

The screenshot shows the Truffle UI interface. On the left, the 'SOLIDITY COMPILER' sidebar is open, displaying compiler settings: Compiler version 0.5.17+commit.d19bba13, Language Solidity, EVM Version compiler default, and Compiler Configuration with Auto compile, Enable optimization (set to 200), and Hide warnings checked. A large blue button labeled 'Compile cryptodata.sol' is prominent. Below the sidebar, the 'CONTRACT' section shows 'Test (cryptodata.sol)' selected. Underneath, there are buttons for 'Publish on Swarm' and 'Publish on Infura'. To the right, the main workspace displays two tabs: 'overloading.sol' and 'cryptodata.sol'. The code for 'cryptodata.sol' is identical to the one provided above. Below the tabs, a transaction details panel is visible, showing a transaction from address 0x5d380a6a701c568545dCf8803fc8875f59bedd54 to the contract 'Soliditytest' with a gas limit of 3000000.

The screenshot shows the Remix IDE interface. On the left, there's a sidebar with icons for account, file, settings, and help. The main area has tabs for "overloading.sol" and "cryptodata.sol". The "overloading.sol" tab contains the following Solidity code:

```

1 pragma solidity ^0.5.0;
2 contract Test {
3     function callSha256() public pure returns( bytes32 result){
4         return sha256("ronaldo");
5     }
6     function callKeccak256() public pure returns( bytes32 result){
7         return keccak256("ronaldo");
8     }
9 }

```

On the right, under "TEST AT 0x332...D4B6D (MEMORY)", there are two sections: "callkeccak256" and "callsha256". Both sections show a single output value. Below these, the "Low level interactions" section shows a "TRANSACTION" button.

Practical 7

Implement and demonstrate the use of the following in Solidity :

- (I).Contracts
- (II).Inheritance
- (III).Constructors
- (IV).Abstract Class
- (V).Interfaces

(I).Contracts

```
// Solidity program to
// demonstrate how to
// write a smart contract
pragma solidity >= 0.4.16 < 0.7.0;

// Defining a contract
contract Test
{
    // Declaring state variables
    uint public var1;
    uint public var2;
    uint public sum;

    // Defining public function
    // that sets the value of
    // the state variable
    function set(uint x, uint y) public
    {
        var1 = x;
        var2=y;
        sum=var1+var2;
    }

    // Defining function to
    // print the sum of
    // state variables
    function get(
    ) public view returns (uint) {
```

```
return sum;  
}  
}
```

Output:

The screenshot shows the Solidity Compiler interface. On the left, there's a sidebar with various icons and settings like compiler version (0.5.17+commit.d19bba13), language (Solidity), EVM version (compiler default), and compiler configuration options (Auto compile, Enable optimization set to 200, Hide warnings). The main area displays the Solidity code for `smartcontract.sol`:

```
// Solidity program to  
// demonstrate how to  
// write a smart contract  
pragma solidity >=0.4.16 < 0.7.0;  
// Defining a contract  
contract Test {  
    // Declaring state variables  
    uint public var1;  
    uint public var2;  
    uint public sum;  
    // Defining public function  
    // that sets the value of  
    // the state variable  
    function set(uint x, uint y) public {  
        var1 = x;  
        var2=y;  
        sum=var1+var2;  
    }  
    // Defining function to  
    // print the sum of  
    // state variables  
    function get() public view returns (uint) {  
        return sum;  
    }  
}
```

Below the code, there are buttons for "Compile smartcontract.sol" and "Test (smartcontract.sol)". Further down, there are buttons for "Publish on Swarm" and "Publish on Ipfs". At the bottom, there are fields for "from", "to", and "gas", with "gas" set to 3000000. A search bar at the top right says "Search with transaction hash or address".

```

// Solidity program to demonstrate how to
// write a smart contract
pragma solidity >= 0.4.16 < 0.7.0;

// Defining a contract
contract Test {
    // Declaring state variables
    uint public var1;
    uint public var2;
    uint public sum;

    // Defining public function
    // that sets the value of
    // the state variable
    function set(uint x, uint y) public {
        var1 = x;
        var2=y;
        sum=var1+var2;
    }

    // Defining function to
    // print the sum of
    // state variables
    function get() public view returns (uint) {
        return sum;
    }
}

```

(II).Inheritance

```

// Solidity program to demonstrate Single Inheritance

pragma solidity >=0.4.22 <0.6.0;

// Defining contract

contract parent{

    // Declaring internal state varaiable
    uint internal sum;

    // Defining external function to set value of internal state variable sum
    function setValue() external {
        uint a = 10;
        uint b = 20;
    }
}
```

```

sum = a + b;
}

}

// Defining child contract
contract child is parent{

// Defining external function to return value of internal state variable sum
function getValue() external view returns(uint) {
    return sum;
}
}

// Defining calling contract
contract caller {

// Creating child contract object
child cc = new child();

// Defining function to call setValue and getValue functions
function testInheritance() public {
    cc.setValue();
}

function result() public view returns(uint ){
    return cc.getValue();
}
}

```

Output:

The screenshot shows the Truffle UI interface for deploying and running transactions. On the left, the 'DEPLOY & RUN TRANSACTIONS' sidebar includes fields for ENVIRONMENT (JavaScript VM), ACCOUNT (0x5B3...eddC4), GAS LIMIT (3000000), VALUE (0 wei), and CONTRACT (child - Hierarchical.sol). It features a 'Deploy' button and options to Publish to IPFS or Deploy at a specific address. Below these are sections for Transactions recorded (26) and Deployed Contracts, listing various deployed test contracts like TYPES, TEST, and LEDGERBALANCE. A 'Low level interactions' section at the bottom allows for direct CALLDATA interaction.

Hierarchical.sol

```
// Solidity program to
// demonstrate
// Single Inheritance
pragma solidity >=0.4.22 <0.6.0;

// Defining contract
contract parent{
}

// Declaring internal
// state variable
uint internal sum;

// Defining external function
// to set value of internal
// state variable sum
function setValue() external {
    uint a = 10;
    uint b = 20;
    sum = a + b;
}

// Defining child contract
contract child is parent{
}

// Defining external function
// to return value of
// internal state variable sum
function getValue() external view returns(uint) {
    return sum;
}
```

(III).Constructors

```
// Solidity program to demonstrate  
// creating a constructor  
pragma solidity ^0.5.0;
```

```
// Creating a contract  
contract constructorExample {
```

```
// Declaring state variable  
string str;
```

```
// Creating a constructor  
// to set value of 'str'  
constructor() public {  
    str = "GeeksForGeeks";  
}
```

```
// Defining function to  
// return the value of 'str'  
function getValue()  
) public view returns (  
    string memory) {  
    return str;  
}
```

Output:

The screenshot shows the Solidity Compiler interface. On the left, there's a sidebar with various icons. The main area is titled "SOLIDITY COMPILER". It includes settings for "COMPILER" (set to "0.5.17+commit.d19bba13"), "LANGUAGE" (set to "Solidity"), "EVM VERSION" (set to "compiler default"), and "COMPILER CONFIGURATION" (with options for "Auto compile", "Enable optimization" set to 200, and "Hide warnings"). A large blue button at the bottom left says "Compile Contract constructor.sol". To the right, there are three tabs: "Arithmetic operator.sol", "Smart contract.sol", and "Contract constructor.sol". The "Contract constructor.sol" tab is active, displaying the following Solidity code:

```
// Solidity program to demonstrate
// creating a constructor
pragma solidity ^0.5.0;

// Creating a contract
contract constructorExample {

    // Declaring state variable
    string str;

    // Creating a constructor
    // to set value of 'str'
    constructor() public {
        str = "GeeksForGeeks";
    }

    // Defining function to
    // return the value of 'str'
    function getValue()
    public view returns (
        string memory) {
        return str;
    }
}
```

The screenshot shows the Truffle UI interface for deploying and running transactions. On the left, there's a sidebar with icons for accounts, contracts, and deployment. The main area is divided into sections:

- DEPLOY & RUN TRANSACTIONS**: Includes fields for ENVIRONMENT (JavaScript VM), ACCOUNT (0x5B3...eddC4), GAS LIMIT (3000000), and VALUE (0 wei).
- CONTRACT**: Shows "constructorExample - Contract constructor" and a Deploy button.
- Deploy to IPFS** and **OR** options.
- Transactions recorded**: Shows 6 deployed contracts:
 - SOLIDITYTEST AT 0XD91...39138 (MEMORY)
 - SOLIDITYTEST AT 0XD8B...33FA8 (MEMORY)
 - SOLIDITYTEST AT 0XFBE...9FBE8 (MEMORY)
 - TEST AT 0XD7A...F771B (MEMORY)
 - CONSTRUCTOREXAMPLE AT 0XA0...4 (MEMORY)
- Low level interactions**: Shows a CALLDATA section with a Transact button.

The Solidity code in the editor is:

```
// Solidity program to demonstrate
// creating a constructor
pragma solidity ^0.5.0;

// Creating a contract
contract constructorExample {

    // Declaring state variable
    string str;

    // Creating a constructor
    // to set value of 'str'
    constructor() public {
        str = "GeeksForGeeks";
    }

    // Defining function to
    // return the value of 'str'
    function getValue()
    public view returns (
        string memory) {
        return str;
    }
}
```

Practical 8

Implement and demonstrate the use of the following in Solidity :

(I).Libraries

(II).Assembly

(III).Events

(IV).Error Handling

(IV).Error Handling

```
// Solidity program to demonstrate require statement
```

```
pragma solidity ^0.5.0;
```

```
// Creating a contract
```

```
contract requireStatement {
```

```
// Defining function to check input
```

```
function checkInput(
```

```
uint _input) public view returns(
```

```
string memory){
```

```
require(_input >= 0, "invalid uint8");
```

```
require(_input <= 255, "invalid uint8");
```

```
return "Input is Uint8";
```

```
}
```

```
// Defining function to use require statement
```

```
function Odd(uint _input) public view returns(bool){
```

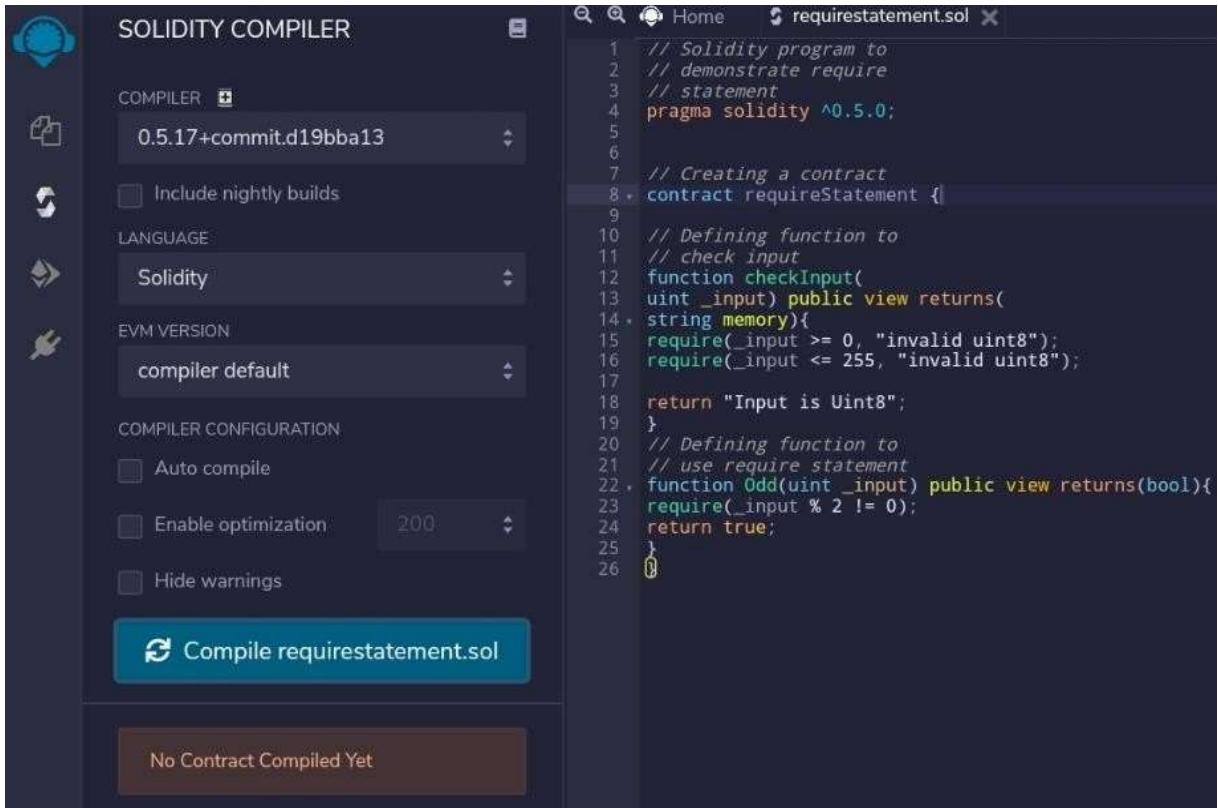
```
require(_input % 2 != 0);
```

```
return true;
```

```
}
```

}

Output:

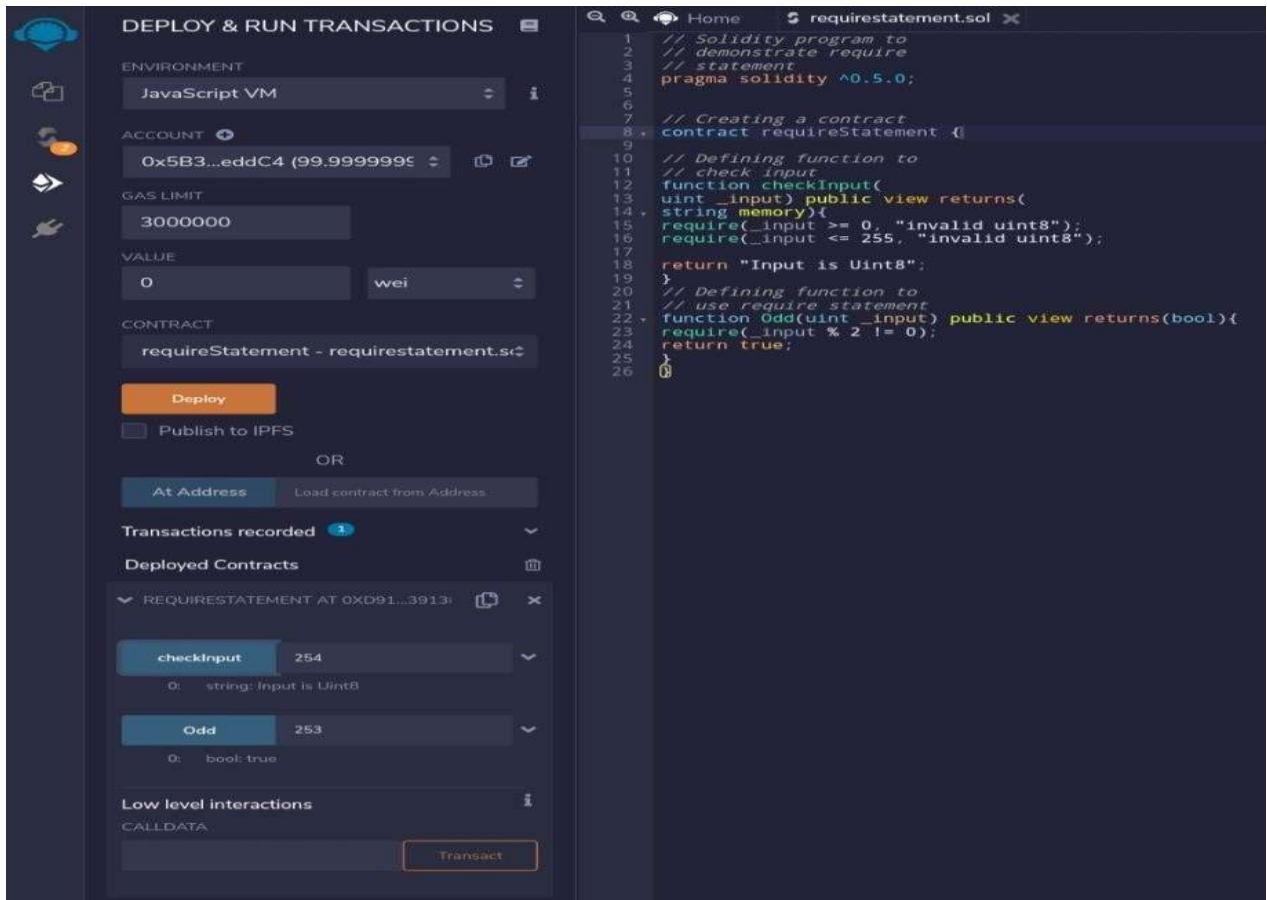


The image shows the Solidity Compiler interface in a web browser. On the left, there's a sidebar with various icons and dropdown menus for compiler settings like language (Solidity), EVM version (compiler default), and compiler configuration (auto compile, enable optimization set to 200). A prominent blue button at the bottom of the sidebar says "Compile requirestatement.sol". To the right of the sidebar is a code editor window titled "requirestatement.sol" containing the following Solidity code:

```
// Solidity program to
// demonstrate require
// statement
pragma solidity ^0.5.0;

// Creating a contract
contract requireStatement {
    // Defining function to
    // check input
    function checkInput(
        uint _input) public view returns(
            string memory){
        require(_input >= 0, "invalid uint8");
        require(_input <= 255, "invalid uint8");
        return "Input is Uint8";
    }
    // Defining function to
    // use require statement
    function Odd(uint _input) public view returns(bool){
        require(_input % 2 != 0);
        return true;
    }
}
```

Below the code editor, a message box says "No Contract Compiled Yet".



// Solidity program to demonstrate assert statement

```
pragma solidity ^0.5.0;
```

// Creating a contract

```
contract assertStatement {
```

```
// Defining a state variable
```

```
bool result;
```

```
// Defining a function to check condition
```

```
function checkOverflow(
```

```

        uint _num1, uint _num2) public {
        uint8 sum = _num1 + _num2;
        assert(sum<=255);
        result = true;
    }
}

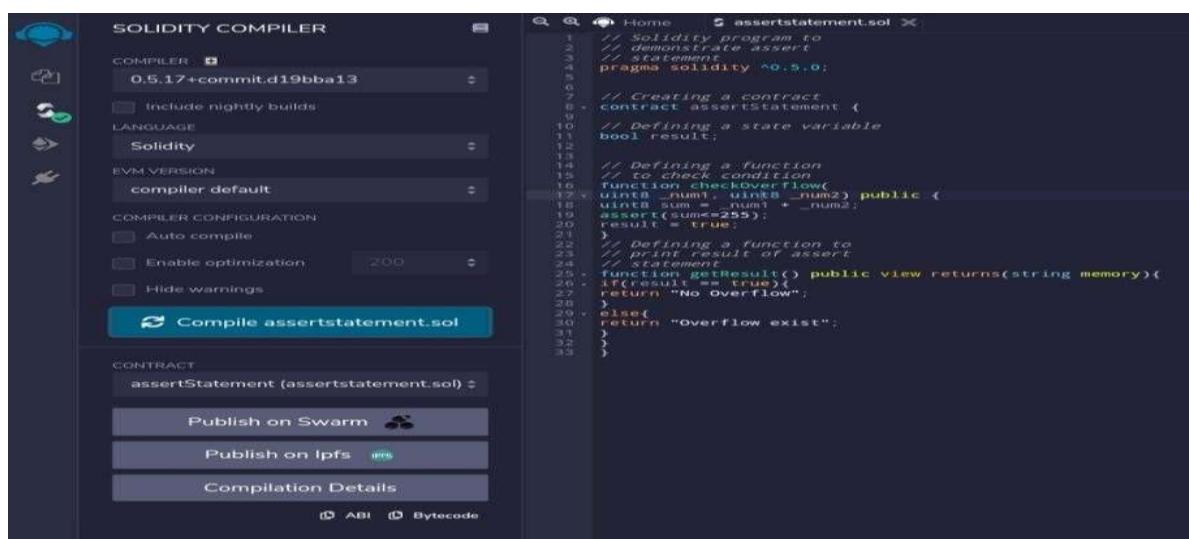
```

```

// Defining a function to print result of assert statement
function getResult() public view returns(string memory){
    if(result == true){
        return "No Overflow";
    }
    else{
        return "Overflow exist";
    }
}

```

Output:



The screenshot shows the Solidity Compiler interface with the following details:

- SOLIDITY COMPILER** section:
 - Compiler: 0.5.17+commit.d19bba13
 - Include nightly builds:
 - LANGUAGE: Solidity
 - EVM VERSION: compiler default
 - COMPILER CONFIGURATION:
 - Auto compile:
 - Enable optimization: 200
 - Hide warnings:
 - Compile button: **Compile assertstatement.sol**
- CONTRACT** section:
 - assertStatement (assertstatement.sol)
 - Publish on Swarm:
 - Publish on Ipfs:
 - Compilation Details: ABI, Bytecode
- assertstatement.sol** file content (right pane):


```

1 // Solidity program to
2 // demonstrate assert
3 pragma solidity ^0.5.0;
4
5 // Creating a contract
6 contract assertStatement {
7     // Defining a state variable
8     bool result;
9
10    // Defining a function
11    // to check condition
12    function checkOverflow()
13    uint8 sum = _num1 + _num2;
14    assert(sum<=255);
15    result = true;
16}
17
18    // Defining a function to
19    // print result of assert
20    // statement
21    function getResult() public view returns(string memory){
22        if(result == true){
23            return "No overflow";
24        }
25        else{
26            return "Overflow exist";
27        }
28    }
29
30
31
32
33

```

The screenshot shows the Truffle IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' sidebar is open, displaying deployment settings: Environment (JavaScript VM), Account (0x5B3...eddC4), Gas Limit (3000000), and Value (0 wei). Below these are options to Deploy or Publish to IPFS, and a section for Deployed Contracts. Under 'Deployed Contracts', there is a list for 'ASSERTSTATEMENT AT 0xD91...39138' which includes two functions: 'checkOverflow' and 'getResult'. The 'getResult' function has a result of '0: string: Overflow exist'. At the bottom, there is a 'Low level interactions' section with a 'CALLDATA' input field and a 'Transact' button.

The right side of the interface shows the Solidity code for the 'assertStatement' contract:

```

1 // Solidity program to
2 // demonstrate assert
3 // statement
4 pragma solidity ^0.5.0;
5
6
7 // Creating a contract
8 contract assertStatement {
9
10 // Defining a state variable
11 bool result;
12
13 // Defining a function
14 // to check condition
15 function checkOverflow(
16     uint8 num1, uint8 num2) public {
17     uint8 sum = num1 + num2;
18     assert(sum<=255);
19     result = true;
20 }
21
22 // Defining a function to
23 // print result of assert
24 // statement
25 function getResult() public view returns(string memory){
26     if(result == true){
27         return "No Overflow";
28     }
29     else{
30         return "Overflow exist";
31     }
32 }
33

```

// Solidity program to demonstrate assert statement

```
pragma solidity ^0.5.0;
```

// Creating a contract

```
contract assertStatement {
```

// Defining a state variable

```
bool result;
```

// Defining a function

// to check condition

```
function checkOverflow(uint8 sum) public {
```

```

assert(sum<=255);

result = true;

}

// Defining a function to print result of assert statement

function getResult() public view returns(string memory){

if(result == true){

return "No Overflow";

}

else{

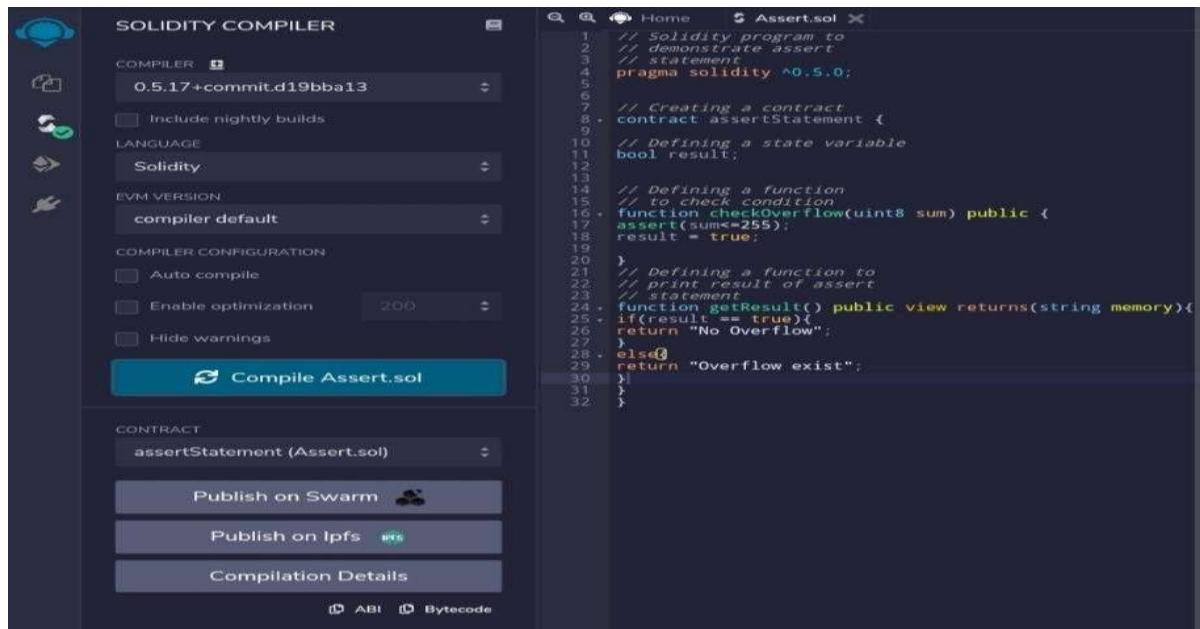
return "Overflow exist";

}

}

```

Output:

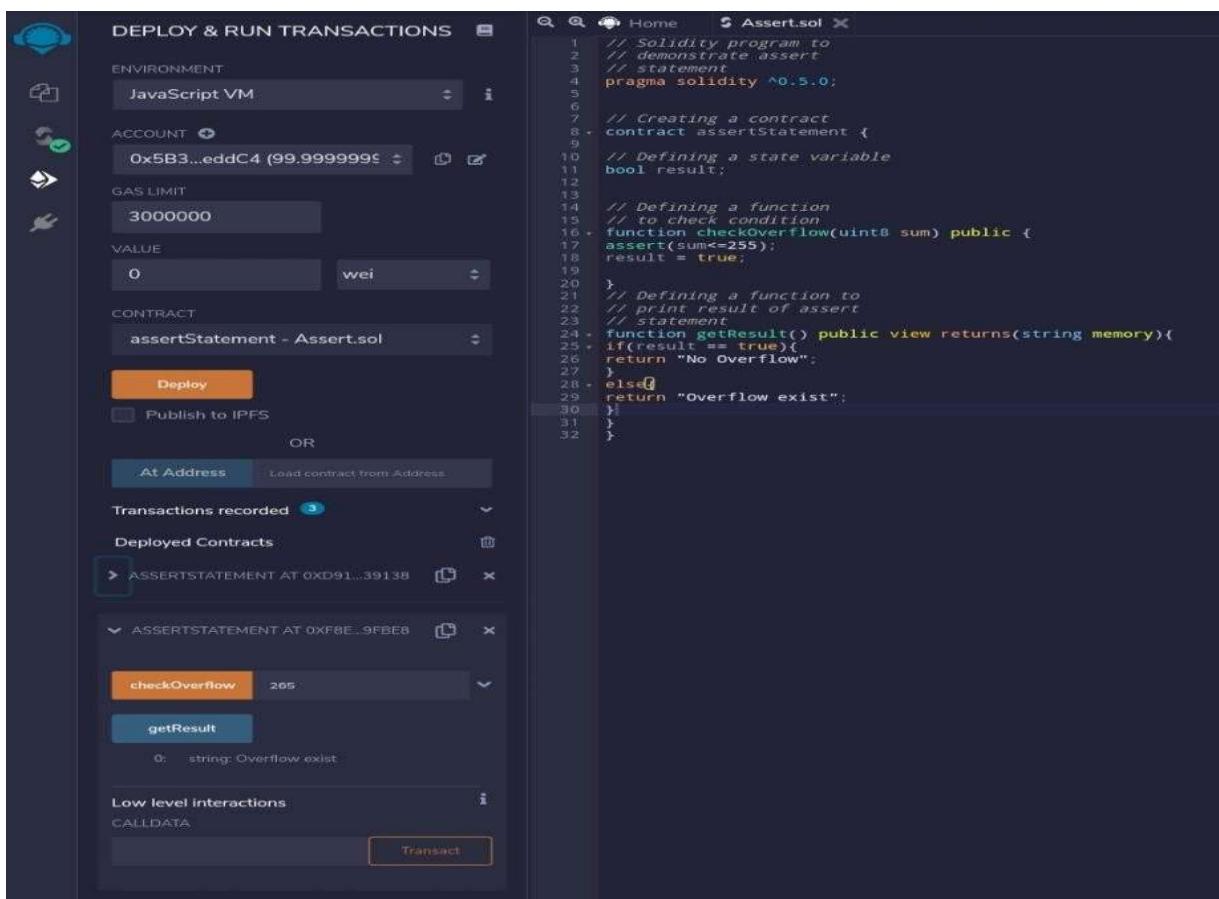


The screenshot shows the Solidity Compiler interface with the following details:

- SOLIDITY COMPILER** tab is selected.
- COMPILER**: 0.5.17+commit.d19bba13
- LANGUAGE**: Solidity
- EVM VERSION**: compiler default
- COMPILER CONFIGURATION**:
 - Auto compile
 - Enable optimization: 200
 - Hide warnings
- Compile Assert.sol** button is highlighted in blue.
- CONTRACT**: assertStatement (Assert.sol)
- Publish on Swarm** and **Publish on Ipfs** buttons are visible.
- Compilation Details** section at the bottom.
- Code Area** (Assert.sol):


```

1 // Solidity program to
2 // demonstrate assert
3 // statement
4 pragma solidity ^0.5.0;
5
6
7 // Creating a contract
8 contract assertStatement {
9
10 // Defining a state variable
11 bool result;
12
13
14 // Defining a function
15 // to check condition
16 function checkOverflow(uint8 sum) public {
17 assert(sum<=255);
18 result = true;
19 }
20
21 // Defining a function to
22 // print result of assert
23 // statement
24 function getResult() public view returns(string memory){
25 if(result == true){
26 return "No Overflow";
27 }
28 else{
29 return "Overflow exist";
30 }
31 }
32 }
```



```
// Solidity program to demonstrate revert statement
pragma solidity ^0.5.0;

// Creating a contract
contract revertStatement {

    // Defining a function to check condition
    function checkOverflow(
        uint _num1, uint _num2) public view returns(
        string memory, uint) {

        uint sum = _num1 + _num2;
        if (sum >= 255) {
            revert("Overflow exist");
        }
        return ("No Overflow", sum);
    }

    // Defining a function to print result of assert
    function getResult() public view returns(string memory) {
        if(result == true){
            return "No Overflow";
        } else{
            return "Overflow exist";
        }
    }
}
```

```

if(sum < 0 || sum > 255){

revert(" Overflow Exist");

}

else{

return ("No Overflow", sum);

}

}

}

}

}

```

Output:

The screenshot shows the Solidity Compiler interface with the following details:

- COMPILER:** 0.5.17+commit.d19bba13
- LANGUAGE:** Solidity
- EVM VERSION:** compiler default
- COMPILER CONFIGURATION:** Auto compile, Enable optimization (200), Hide warnings
- CONTRACT:** revertStatement (revertstatement.sol)
- Buttons:** Publish on Swarm, Publish on Ipfs, Compilation Details
- Output:**
 - Contract address: 0x5fca5de9819161eb66c12b59aab8e9c427b740d75c05a17394c2854b1143c6
 - Status: true Transaction mined and execution succeed
 - Transaction hash: 0x85f...143c6

The screenshot shows the Truffle UI interface for deploying and running transactions. On the left, there's a sidebar titled "DEPLOY & RUN TRANSACTIONS" with sections for "var1" and "var2". Below that is a "Low level interactions" section with "CALLDATA" and a "Transact" button. A dropdown menu shows "REVERTSTATEMENT AT 0x1C9...2B48C".

The main area displays a Solidity code editor with the following code:

```
// Solidity program to demonstrate revert statement
pragma solidity ^0.5.0;

// Creating a contract
contract revertStatement {
    // Defining a function to check condition
    function checkOverflow(
        uint _num1, uint _num2) public view returns(
            string memory, uint) {
        uint sum = _num1 + _num2;
        if(sum < 0 || sum > 255){
            revert("Overflow Exist");
        } else{
            return ("No Overflow", sum);
        }
    }
}
```

Below the code editor, there's a "checkOverflow" section with input fields for "_num1" (52) and "_num2" (35), and a "call" button.

At the bottom, there's a "Low level interactions" section with "CALLDATA" and a "Transact" button. A status bar at the bottom right shows "[vm] from: 0x583...eddC4 to: Test.set(uint256,uint256) 0x5e1...4Efff value: 0 wei data: 0x1ab...0000f logs: 0 hash: 0x85f...143c6".

University Of Mumbai
Centre for Distance & Online Education



Dr.Shangar Dayal Sharama Bhavan, Kalina, Vidanagari, Santacruz (E),
Mumbai-400 098.

Certificate

This is to certify that

Mr. **MAKRAND GHAG** Application ID: **93669**, Seat No: **2900084** from **Rizvi College of Arts, Science and Commerce Bandra(W)**, Mumbai 400 050 has successfully completed all the practical of Paper II titled **DEEP LEARNING** for M.sc (IT) Part II in the academic year 2023-2024.

Section I _____

Section II _____

MSc (IT) Co-ordinator, IDOL

External Examiner

INDEX

Sr. No.	Topic	Remarks
1	Performing matrix multiplication and finding Eigen vectors and Eigen values using Tensor Flow.	
2	Solving XOR problem using deep feed forward network.	
3	Implementing deep neural network for performing classification task.	
4	Using deep feed forward network with two hidden layers for performing classification and predicting the class.	
5	Evaluating feed forward deep network for regression using K Fold cross validation	
6	Implementing regularization to avoid overfitting in binary classification.	
7	Demonstrate recurrent neural network that learns to perform sequence analysis for stock price.	
8	Performing encoding and decoding of images using deep autoencoder.	
9	Implementation of convolutional neural network to predict numbers from number images	
10	Denoising of images using auto encoder.	

PRACTICAL 1

AIM: Performing matrix multiplication and finding Eigen vectors and Eigen values using Tensor Flow.

CODE:

The screenshot shows a Jupyter Notebook interface with the title "PRAC1". The code cell contains the following Python code:

```
In [2]: import tensorflow as tf
print("Matrix Multiplication Demo")
x=tf.constant([1,2,3,4,5,6],shape=[2,3])
print(x)
y=tf.constant([7,8,9,10,11,12],shape=[3,2])
y=tf.constant([7,8,9,10,11,12],shape=[3,2])
print(y)
z=tf.matmul(x,y)
print("Product:",z)
e_matrix_A=tf.random.uniform([2,2],minval=3,maxval=10,dtype=tf.float32,name="matrixA")
print("Matrix A:\n{}\n".format(e_matrix_A))
eigen_values_A,eigen_vectors_A=tf.linalg.eigh(e_matrix_A)
print("Eigen Vectors:\n{}\nEigen Values:\n{}".format(eigen_vectors_A,eigen_values_A))
```

OUTPUT:

The output cell displays the results of the executed code:

```
Matrix Multiplication Demo
tf.Tensor(
[[1 2 3]
 [4 5 6]], shape=(2, 3), dtype=int32)
tf.Tensor(
[[ 7  8]
 [ 9 10]
 [11 12]], shape=(3, 2), dtype=int32)
Product: tf.Tensor(
[[ 58  64]
 [139 154]], shape=(2, 2), dtype=int32)
Matrix A:
[[7.1561775 9.106806 ]
 [6.0239253 9.841141 ]]

Eigen Vectors:
[[-0.7802314 -0.625491 ]
 [ 0.625491  -0.7802314]]

Eigen Values:
[ 2.326955 14.670364]
```

PRACTICAL 2

AIM: Solving XOR problem using deep feed forward network.

CODE:

The screenshot shows a Jupyter Notebook interface with the title "PRAC 2" and a status bar indicating "Last Checkpoint: an hour ago (unsaved changes)". The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. Below the menu is a toolbar with various icons for file operations like new, open, save, and run. The code cell (In [3]) contains Python code for creating a neural network and training it on the XOR dataset.

```
In [3]: import numpy as np
from keras.layers import Dense
from keras.models import Sequential
model=Sequential()
model.add(Dense(units=2,activation='relu',input_dim=2))
model.add(Dense(units=1,activation='sigmoid'))
model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])
print(model.summary())
print(model.get_weights())
X=np.array([[0.,0.],[0.,1.],[1.,0.],[1.,1.]])
Y=np.array([0.,1.,1.,0.])
model.fit(X,Y,epochs=1000,batch_size=4)
print(model.get_weights())
print(model.predict(X,batch_size=4))
```

OUTPUT:

The output cell displays the model summary and training progress. It shows a sequential model with two layers: a dense layer of 2 units and a dense_1 layer of 1 unit. The total number of parameters is 9, all of which are trainable and non-trainable. The training progress shows Epoch 1/1000 with a loss of 0.7073 and an accuracy of 0.5000.

```
Model: "sequential"
-----  
Layer (type)      Output Shape     Param #
-----  
dense (Dense)    (None, 2)        6  
dense_1 (Dense)  (None, 1)        3  
-----  
Total params: 9  
Trainable params: 9  
Non-trainable params: 0  
-----  
None  
[array([-1.0452268 ,  0.38884377],  
      [-1.0456628 , -1.0601878 ], dtype=float32), array([0., 0.], dtype=float32), array([[ 1.1397151 ],  
      [-0.28245735]], dtype=float32), array([0.], dtype=float32)]  
Epoch 1/1000  
1/1 [=====] - 3s 3s/step - loss: 0.7073 - accuracy: 0.5000
```

```
None
[array([[-1.0452268 ,  0.38884377],
       [-1.0456628 , -1.0601878 ]], dtype=float32), array([0., 0.], dtype=float32), array([[ 1.1397151 ],
       [-0.28245735]], dtype=float32), array([0.], dtype=float32)]
Epoch 1/1000
1/1 [=====] - 3s 3s/step - loss: 0.7073 - accuracy: 0.5000
Epoch 2/1000
1/1 [=====] - 0s 7ms/step - loss: 0.7071 - accuracy: 0.2500
Epoch 3/1000
1/1 [=====] - 0s 9ms/step - loss: 0.7070 - accuracy: 0.2500
Epoch 4/1000
1/1 [=====] - 0s 14ms/step - loss: 0.7069 - accuracy: 0.2500
Epoch 5/1000
1/1 [=====] - 0s 14ms/step - loss: 0.7067 - accuracy: 0.2500
Epoch 6/1000
1/1 [=====] - 0s 15ms/step - loss: 0.7066 - accuracy: 0.2500
Epoch 7/1000
1/1 [=====] - 0s 16ms/step - loss: 0.7065 - accuracy: 0.2500
Epoch 8/1000
1/1 [=====] - 0s 11ms/step - loss: 0.7063 - accuracy: 0.2500
```

```
Epoch 503/1000
1/1 [=====] - 0s 11ms/step - loss: 0.6931 - accuracy: 0.5000
Epoch 504/1000
1/1 [=====] - 0s 9ms/step - loss: 0.6931 - accuracy: 0.5000
Epoch 505/1000
1/1 [=====] - 0s 10ms/step - loss: 0.6931 - accuracy: 0.5000
Epoch 506/1000
1/1 [=====] - 0s 11ms/step - loss: 0.6931 - accuracy: 0.5000
Epoch 507/1000
1/1 [=====] - 0s 11ms/step - loss: 0.6931 - accuracy: 0.5000
Epoch 508/1000
1/1 [=====] - 0s 10ms/step - loss: 0.6931 - accuracy: 0.5000
Epoch 509/1000
1/1 [=====] - 0s 8ms/step - loss: 0.6931 - accuracy: 0.5000
Epoch 510/1000
1/1 [=====] - 0s 9ms/step - loss: 0.6931 - accuracy: 0.5000
Epoch 511/1000
1/1 [=====] - 0s 15ms/step - loss: 0.6931 - accuracy: 0.5000
Epoch 512/1000
1/1 [=====] - 0s 10ms/step - loss: 0.6931 - accuracy: 0.5000
```

```
1/1 [=====] - 0s 10ms/step - loss: 0.6931 - accuracy: 0.5000
Epoch 996/1000
1/1 [=====] - 0s 14ms/step - loss: 0.6931 - accuracy: 0.5000
Epoch 997/1000
1/1 [=====] - 0s 11ms/step - loss: 0.6931 - accuracy: 0.5000
Epoch 998/1000
1/1 [=====] - 0s 10ms/step - loss: 0.6931 - accuracy: 0.5000
Epoch 999/1000
1/1 [=====] - 0s 10ms/step - loss: 0.6931 - accuracy: 0.5000
Epoch 1000/1000
1/1 [=====] - 0s 11ms/step - loss: 0.6931 - accuracy: 0.5000
[array([[-1.0452268,  0.1875426],
       [-1.0456628, -1.0601878]], dtype=float32), array([ 0.          , -0.20130123], dtype=float32), array([[ 1.1397151 ],
       [-0.13495907]], dtype=float32), array([6.0260376e-08], dtype=float32)]
1/1 [=====] - 0s 107ms/step
[[0.5000006]
 [0.5000006]
 [0.5000006]
 [0.5000006]]
```

PRACTICAL 3

AIM: Implementing deep neural network for performing classification task.

PROBLEM STATEMENT: The given dataset comprises of health information about diabetic women patient. We need to create deep feed forward network that will classify women suffering from diabetes mellitus as 1.

CODE & OUTPUT:

In [1]:

```
from numpy import loadtxt
from keras.models import Sequential
from keras.layers import Dense
dataset=loadtxt('diabetes.csv',delimiter=',')
dataset
```

Out[1]:

```
array([[ 6. , 148. , 72. , ..., 0.627, 50. , 1. ],
       [ 1. , 85. , 66. , ..., 0.351, 31. , 0. ],
       [ 8. , 183. , 64. , ..., 0.672, 32. , 1. ],
       ...,
       [ 5. , 121. , 72. , ..., 0.245, 30. , 0. ],
       [ 1. , 126. , 60. , ..., 0.349, 47. , 1. ],
       [ 1. , 93. , 70. , ..., 0.315, 23. , 0. ]])
```

```
In [3]: model=Sequential()

In [4]: model.add(Dense(12, input_dim=8,activation='relu' ))
model.add(Dense(8,activation='relu' ))
model.add(Dense(1,activation='sigmoid' ))
model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])
model.fit(X,Y,epochs=150,batch_size=4)
```

```
Epoch 1/150
192/192 [=====] - 1s 2ms/step - loss: 6.7649 - accuracy: 0.5260
Epoch 2/150
192/192 [=====] - 0s 2ms/step - loss: 1.8355 - accuracy: 0.5690
Epoch 3/150
192/192 [=====] - 0s 2ms/step - loss: 1.0783 - accuracy: 0.6185
Epoch 4/150
192/192 [=====] - 0s 2ms/step - loss: 0.8862 - accuracy: 0.6185
Epoch 5/150
192/192 [=====] - 0s 2ms/step - loss: 0.8770 - accuracy: 0.6211
Epoch 6/150
192/192 [=====] - 0s 2ms/step - loss: 0.8434 - accuracy: 0.6289
Epoch 7/150
192/192 [=====] - 1s 3ms/step - loss: 0.7703 - accuracy: 0.6628
Epoch 8/150
192/192 [=====] - 0s 2ms/step - loss: 0.7334 - accuracy: 0.6589
Epoch 9/150
192/192 [=====] - 0s 2ms/step - loss: 0.7118 - accuracy: 0.6445
Epoch 10/150
```

```
In [5]: _,Accuracy=model.evaluate(X,Y)

24/24 [=====] - 0s 3ms/step - loss: 0.4209 - accuracy: 0.8073
```

```
In [6]: print("Accuracy of Model", (Accuracy*100))

Accuracy of Model 80.72916865348816
```

```
In [7]: prediction=model.predict(X)

24/24 [=====] - 0s 3ms/step
```

```
In [8]: exec("for i in range(5):print(X[i].tolist,prediction[i], Y[i])" )

<built-in method tolist of numpy.ndarray object at 0x000001AF6AF8DDB0> [0.69462395] 1.0
<built-in method tolist of numpy.ndarray object at 0x000001AF6AF8DDB0> [0.06601566] 0.0
<built-in method tolist of numpy.ndarray object at 0x000001AF6AF8DDB0> [0.8118147] 1.0
<built-in method tolist of numpy.ndarray object at 0x000001AF6AF8DDB0> [0.10350533] 0.0
<built-in method tolist of numpy.ndarray object at 0x000001AF6AF8DDB0> [0.73300654] 1.0
```

PRACTICAL 4

A. AIM: Using deep feed forward network with two hidden layers for performing classification and predicting the class.

CODE:

The screenshot shows a Jupyter Notebook interface with the title "jupyter prac4A Last Checkpoint: Yesterday at 3:20 PM (autosaved)". The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. Below the menu is a toolbar with icons for file operations like Open, Save, Run, and Cell type selection. The code cell (In [1]) contains the following Python script:

```
from keras.models import Sequential
from keras.layers import Dense
from sklearn.datasets import make_blobs
from sklearn.preprocessing import MinMaxScaler
X,Y=make_blobs(n_samples=100,centers=2,n_features=2,random_state=1)
scalar=MinMaxScaler()
scalar.fit(X)
X=scalar.transform(X)
model=Sequential()
model.add(Dense(4,input_dim=2,activation='relu'))
model.add(Dense(4,activation='relu'))
model.add(Dense(1,activation='sigmoid'))
model.compile(loss='binary_crossentropy',optimizer='adam')
model.fit(X,Y,epochs=500)
Xnew,Yreal=make_blobs(n_samples=3,centers=2,n_features=2,random_state=1)
Xnewscalar=transform(Xnew)
Ynew=model.predict(Xnew)
for i in range(len(Xnew)):
    print("X=%s, Predicted=%s, Desired=%s" % (Xnew[i], Ynew[i], Yreal[i]))
```

OUTPUT:

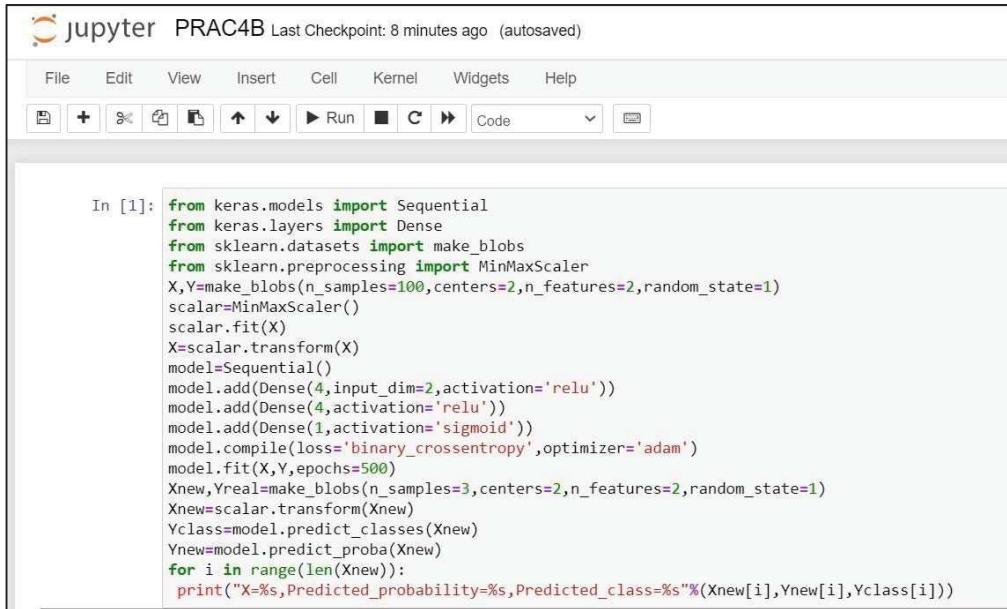
The screenshot shows the output of the Jupyter Notebook code. It displays the training progress of a neural network over 500 epochs. The first part of the output shows epochs from 1 to 10, with loss values decreasing from approximately 0.7221 to 0.7101. The second part of the output shows epochs from 494 to 500, with loss values decreasing from approximately 0.0033 to 0.0032. The final row shows the prediction for three new samples (Xnew) along with their actual values (Yreal) and predicted values (Ynew).

```
Epoch 1/500
4/4 [=====] - 2s 15ms/step - loss: 0.7221
Epoch 2/500
4/4 [=====] - 0s 3ms/step - loss: 0.7202
Epoch 3/500
4/4 [=====] - 0s 8ms/step - loss: 0.7188
Epoch 4/500
4/4 [=====] - 0s 5ms/step - loss: 0.7173
Epoch 5/500
4/4 [=====] - 0s 5ms/step - loss: 0.7158
Epoch 6/500
4/4 [=====] - 0s 5ms/step - loss: 0.7145
Epoch 7/500
4/4 [=====] - 0s 7ms/step - loss: 0.7129
Epoch 8/500
4/4 [=====] - 0s 5ms/step - loss: 0.7114
Epoch 9/500
4/4 [=====] - 0s 8ms/step - loss: 0.7101
Epoch 10/500
4/4 [=====] - 0s 5ms/step - loss: 0.7085

4/4 [=====] - 0s 5ms/step - loss: 0.0033
Epoch 494/500
4/4 [=====] - 0s 4ms/step - loss: 0.0033
Epoch 495/500
4/4 [=====] - 0s 5ms/step - loss: 0.0033
Epoch 496/500
4/4 [=====] - 0s 5ms/step - loss: 0.0032
Epoch 497/500
4/4 [=====] - 0s 5ms/step - loss: 0.0032
Epoch 498/500
4/4 [=====] - 0s 3ms/step - loss: 0.0032
Epoch 499/500
4/4 [=====] - 0s 5ms/step - loss: 0.0032
Epoch 500/500
4/4 [=====] - 0s 3ms/step - loss: 0.0032
1/1 [=====] - 0s 149ms/step
X=[0.89337759 0.65864154],Predicted=[0.00576355],Desired=0
X=[0.29097707 0.12978982],Predicted=[0.9970082],Desired=1
X=[0.78082614 0.75391697],Predicted=[0.00585788],Desired=0
```

4B. AIM: Using a deep feed forward network with two hidden layers for performing classification and predicting the probability of class.

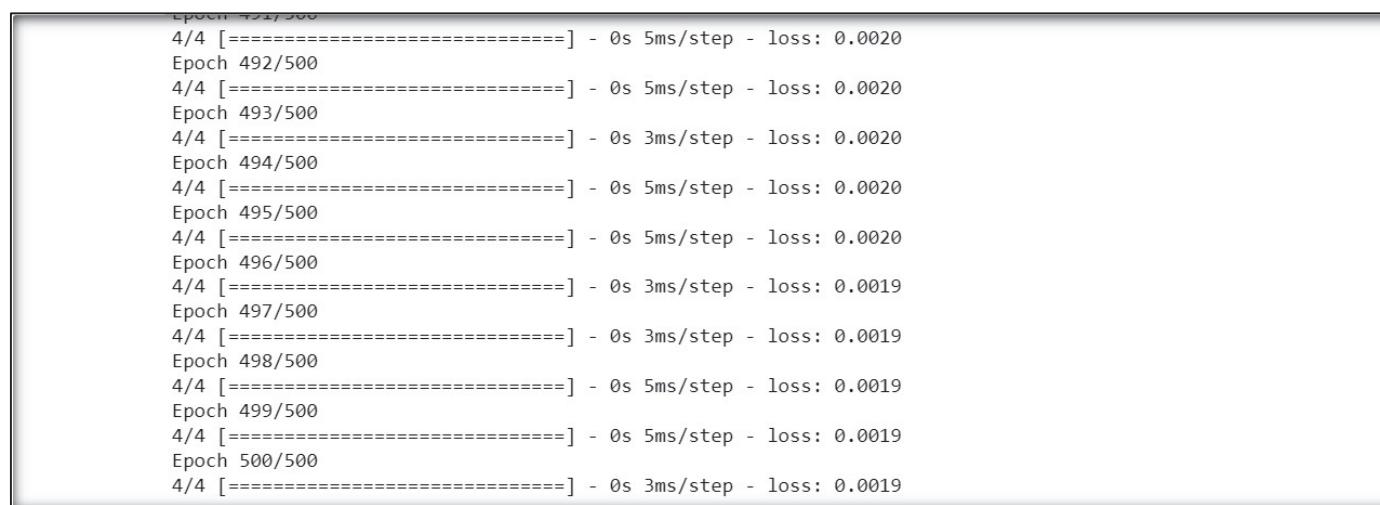
CODE:



The screenshot shows a Jupyter Notebook window titled "jupyter PRAC4B Last Checkpoint: 8 minutes ago (autosaved)". The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. Below the menu is a toolbar with icons for file operations and cell execution. The code cell contains the following Python script:

```
In [1]: from keras.models import Sequential
from keras.layers import Dense
from sklearn.datasets import make_blobs
from sklearn.preprocessing import MinMaxScaler
X,Y=make_blobs(n_samples=100,centers=2,n_features=2,random_state=1)
scalar=MinMaxScaler()
scalar.fit(X)
X=scalar.transform(X)
model=Sequential()
model.add(Dense(4,input_dim=2,activation='relu'))
model.add(Dense(4,activation='relu'))
model.add(Dense(1,activation='sigmoid'))
model.compile(loss='binary_crossentropy',optimizer='adam')
model.fit(X,Y,epochs=500)
Xnew,Yreal=make_blobs(n_samples=3,centers=2,n_features=2,random_state=1)
Xnew=scalar.transform(Xnew)
Yclass=model.predict_classes(Xnew)
Ynew=model.predict_proba(Xnew)
for i in range(len(Xnew)):
    print("X=%s, Predicted_probability=%s, Predicted_class=%s" % (Xnew[i], Ynew[i], Yclass[i]))
```

OUTPUT:



The screenshot shows a terminal window displaying the output of a neural network training session. The output shows the progress of 500 epochs, with each epoch taking approximately 5ms per step. The loss value remains relatively stable around 0.0020 to 0.0019 across all epochs.

```
Epoch 451/500
4/4 [=====] - 0s 5ms/step - loss: 0.0020
Epoch 492/500
4/4 [=====] - 0s 5ms/step - loss: 0.0020
Epoch 493/500
4/4 [=====] - 0s 3ms/step - loss: 0.0020
Epoch 494/500
4/4 [=====] - 0s 5ms/step - loss: 0.0020
Epoch 495/500
4/4 [=====] - 0s 5ms/step - loss: 0.0020
Epoch 496/500
4/4 [=====] - 0s 3ms/step - loss: 0.0019
Epoch 497/500
4/4 [=====] - 0s 3ms/step - loss: 0.0019
Epoch 498/500
4/4 [=====] - 0s 5ms/step - loss: 0.0019
Epoch 499/500
4/4 [=====] - 0s 5ms/step - loss: 0.0019
Epoch 500/500
4/4 [=====] - 0s 3ms/step - loss: 0.0019
```

4C. AIM: Using a deep field forward network with two hidden layers for performing linear regression and predicting values.

CODE:

The screenshot shows a Jupyter Notebook interface with a toolbar at the top and two code cells labeled In [1] and In [2].

```
In [1]: from keras.models import Sequential
from keras.layers import Dense
from sklearn.datasets import make_regression
from sklearn.preprocessing import MinMaxScaler

In [2]: X,Y=make_regression(n_samples=100,n_features=2,noise=0.1,random_state=1)
scalarX,scalarY=MinMaxScaler(),MinMaxScaler()
scalarX.fit(X)
scalarY.fit(Y.reshape(100,1))
X=scalarX.transform(X)
Y=scalarY.transform(Y.reshape(100,1))
model=Sequential()
model.add(Dense(4,input_dim=2,activation='relu'))
model.add(Dense(4,activation='relu'))
model.add(Dense(1,activation='sigmoid'))
model.compile(loss='mse',optimizer='adam')
model.fit(X,Y,epochs=1000,verbose=0)
Xnew,a=make_regression(n_samples=3,n_features=2,noise=0.1,random_state=1)
Xnew=scalarX.transform(Xnew)
Ynew=model.predict(Xnew)
for i in range(len(Xnew)):
    print("X=%s,Predicted=%s"%(Xnew[i],Ynew[i]))
```

OUTPUT:

```
1/1 [=====] - 0s 125ms/step
X=[0.29466096 0.30317302],Predicted=[0.18164389]
X=[0.39445118 0.79390858],Predicted=[0.76110995]
X=[0.02884127 0.6208843 ],Predicted=[0.39497763]
```

PRACTICAL 5

A. AIM: Evaluating feed forward deep network for regression using K Fold cross validation

CODE AND OUTPUT:

```
jupyter PRAC5 Last Checkpoint: 3 minutes ago (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help
In [ ]: import pandas as pd
from keras.models import Sequential
from keras.layers import Dense
#from keras.wrappers.scikit_learn import KerasRegressor
# from scikeras.wrappers import KerasClassifier, KerasRegressor
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline

In [ ]: dataframe=pd.read_csv("housing (1).csv",delim_whitespace=True,header=None)
dataset=dataframe.values

In [ ]: X=dataset[:,0:13]
Y=dataset[:,13]
```

```
In [2]: def wider_model(my_param):
    model=Sequential()
    model.add(Dense(15,input_dim=13,kernel_initializer='normal',activation='relu'))
    model.add(Dense(13,kernel_initializer='normal',activation='relu'))
    model.add(Dense(1,kernel_initializer='normal'))
    model.compile(loss='mean_squared_error',optimizer='adam')
    return model

In [3]: estimators=[]
estimators.append(('standardize',StandardScaler()))
estimators.append(('mlp',KerasClassifier(model=wider_model,my_param=123)))
pipeline=Pipeline(estimators)
kfold=KFold(n_splits=10)
results=cross_val_score(pipeline,X,Y,cv=kfold)
print("Wider: %.2f (%.2f) MSE" % (results.mean(), results.std()))
```

(After changing neuron)

```
model.add(Dense(20, input_dim=13,kernel_initializer='normal',activation='relu'))
```

5B. AIM: Evaluating feed forward deep network for multiclass Classification using K Fold cross-validation.

CODE AND OUTPUT:

```
In [6]: import pandas
from keras.models import Sequential
from keras.layers import Dense
from keras.wrappers.scikit_learn import KerasClassifier
from keras.utils import np_utils
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
from sklearn.preprocessing import LabelEncoder

In [7]: dataset=pandas.read_csv('flower.csv',sep=',',header=1)
print(dataset)

      1   0   1.1  1.2   4   3   15   25   15.1
0    2   1   0     0   2   1   3   150   50
1    3   0   1     0   0   3   3   1   150   50
2    4   0   0     1   4   2   16   125   50
3    5   0   1     0   5   2   2   20   15
4    6   0   1     0   0   4   3   12   50   40
5    7   0   0     0   0   4   3   13   40   20
6    8   0   0     1   2   2   7   100   15
7    9   1   1     0   3   1   4   25   15
8   10   1   1     0   5   2   14   100   60
9   11   1   1     1   5   3   8   45   10
10  12   1   1     1   1   2   9   90   25
11  13   1   1     0   1   2   6   20   10
12  14   1   1     1   4   2   11   80   30
13  15   1   0     0   3   2   10   40   20
14  16   1   0     0   4   2   18   200   60
15  17   1   0     0   2   2   17   150   60
16  18   0   0     1   2   1   5   25   10
```

```
In [8]: dataset1=dataset.values
X=dataset1[:,0:4].astype(float)
Y=dataset1[:,4]
print(Y)
encoder=LabelEncoder()
encoder.fit(Y)
encoder_Y=encoder.transform(Y)
print(encoder_Y)
dummy_Y=np_utils.to_categorical(encoder_Y)

[2 3 4 5 4 4 2 3 5 5 1 1 4 3 4 2 2]
[1 2 3 4 3 3 1 2 4 4 0 0 3 2 3 1 1]

In [9]: print(dummy_Y)
def baseline_model():
    model=Sequential()
    model.add(Dense(8,input_dim=4,activation='relu'))
    model.add(Dense(3,activation='softmax'))
    model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
    return model
estimator=KerasClassifier(build_fn=baseline_model,epochs=100,batch_size=5)
kfold = KFold(n_splits=10, shuffle=True)
results = cross_val_score(estimator, X, dummy_Y, cv=kfold)
print("Baseline: %.2f%% (%.2f%%)" % (results.mean()*100, results.std()*100))
#(Changing neuron)
model=Sequential()
model.add(Dense(10,input_dim=4,activation='relu'))
```

```
[0. 1. 0. 0. 0.]
[0. 0. 1. 0. 0.]
[0. 0. 0. 1. 0.]
[0. 0. 0. 0. 1.]
[0. 0. 0. 1. 0.]
[0. 0. 0. 1. 0.]
[0. 0. 0. 1. 0.]
[0. 1. 0. 0. 0.]
[0. 0. 1. 0. 0.]
[0. 0. 0. 0. 1.]
[0. 0. 0. 0. 1.]
[0. 0. 0. 0. 1.]
[1. 0. 0. 0. 0.]
[0. 0. 0. 1. 0.]
[0. 0. 1. 0. 0.]
[0. 0. 0. 1. 0.]
[0. 1. 0. 0. 0.]
[0. 1. 0. 0. 0.]]
Epoch 1/100
c:\users\srinivas\appdata\local\temp\ipython-9_4ord70ha27kadv-2: DeprecationWarning: KerasClassifier is deprecated. Use KerasClassifier from https://github.com/fchollet/keras
```

```
In [10]: #(Changing neuron)
model.add(Dense(10,input_dim=4,activation='relu'))
```

PRACTICAL 6

AIM: Implementing regularization to avoid overfitting in binary classification.

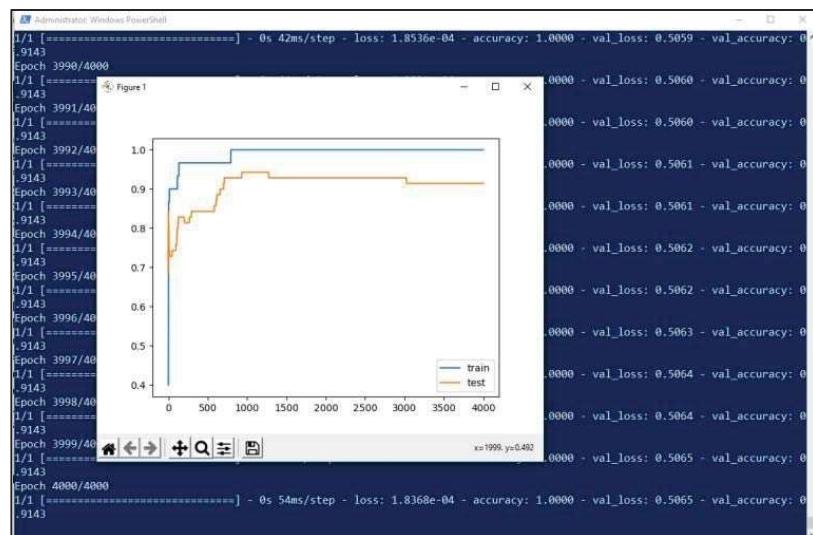
CODE & OUTPUT:

```
jupyter Untitled6 Last Checkpoint: 06/08/2022 (autosaved)

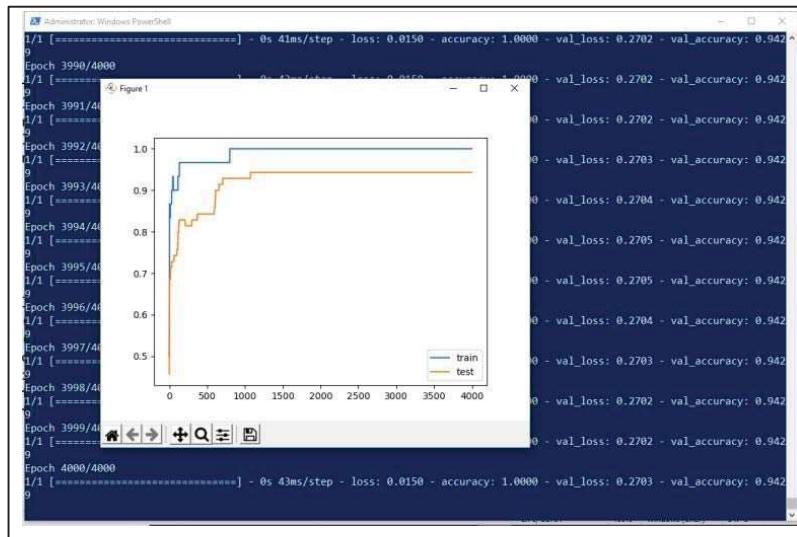
File Edit View Insert Cell Kernel Widgets Help
+ 9% Run Cell Code

In [ ]: from matplotlib import pyplot
from sklearn.datasets import make_moons
from keras.models import Sequential
from keras.layers import Dense
X,Y=make_moons(n_samples=100,noise=0.2,random_state=1)
n_train=30
trainX,testX=X[:n_train,:],X[n_train:]
trainY,testY=Y[:n_train],Y[n_train:]
#print(trainX)
#print(trainY)
#print(testX)
#print(testY)
model=Sequential()
model.add(Dense(500,input_dim=2,activation='relu'))
model.add(Dense(1,activation='sigmoid'))
model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])
history=model.fit(trainX,trainY,validation_data=(testX,testY),epochs=4000)
pyplot.plot(history.history['accuracy'],label='train')
pyplot.plot(history.history['val_accuracy'],label='test')
pyplot.legend()
pyplot.show()
```

```
Epoch 1/4000
1/1 [=====] - 1s 900ms/step - loss: 0.6947 - accuracy: 0.4667 - val_loss: 0.6843 - val_accuracy: 0.6857
Epoch 2/4000
1/1 [=====] - 0s 47ms/step - loss: 0.6777 - accuracy: 0.8000 - val_loss: 0.6735 - val_accuracy: 0.6857
Epoch 3/4000
1/1 [=====] - 0s 57ms/step - loss: 0.6612 - accuracy: 0.8333 - val_loss: 0.6631 - val_accuracy: 0.6857
Epoch 4/4000
1/1 [=====] - 0s 80ms/step - loss: 0.6452 - accuracy: 0.8333 - val_loss: 0.6531 - val_accuracy: 0.6857
Epoch 5/4000
1/1 [=====] - 0s 44ms/step - loss: 0.6296 - accuracy: 0.8667 - val_loss: 0.6434 - val_accuracy: 0.7143
Epoch 6/4000
1/1 [=====] - 0s 47ms/step - loss: 0.6146 - accuracy: 0.8667 - val_loss: 0.6341 - val_accuracy: 0.7143
Epoch 7/4000
```



```
In [*]: from matplotlib import pyplot
from sklearn.datasets import make_moons
from keras.models import Sequential
from keras.layers import Dense
from keras.regularizers import l2
X,Y=make_moons(n_samples=100,noise=0.2,random_state=1)
n_train=30
trainX,testX=X[:n_train,:],X[n_train:]
trainY,testY=Y[:n_train],Y[n_train:]
#print(trainX)
#print(trainY)
#print(testX)
#print(testY)
model=Sequential()
model.add(Dense(500,input_dim=2,activation='relu',kernel_regularizer=l2(0.001)))
model.add(Dense(1,activation='sigmoid'))
model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])
history=model.fit(trainX,trainY,validation_data=(testX,testY),epochs=4000)
pyplot.plot(history.history['accuracy'],label='train')
pyplot.plot(history.history['val_accuracy'],label='test')
pyplot.legend()
pyplot.show()
```



Practical No: 7

Aim: Demonstrate recurrent neural network that learns to perform sequence analysis for stock price.

CODE & OUTPUT:

The screenshot shows a Jupyter Notebook interface with two code cells and their corresponding outputs.

Code Cell 1 (In [2]):

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Dropout
from sklearn.preprocessing import MinMaxScaler
dataset_train=pd.read_csv('Google_stock_price.csv')
#print(dataset_train)
training_set=dataset_train.iloc[:,1:2].values
```

Code Cell 2 (In [3]):

```
#print(training_set)
sc=MinMaxScaler(feature_range=(0,1))
training_set_scaled=sc.fit_transform(training_set)
#print(training_set_scaled)
X_train=[]
Y_train=[]
for i in range(60,1258):
    X_train.append(training_set_scaled[i-60:i,0])
    Y_train.append(training_set_scaled[i,0])
X_train,Y_train=np.array(X_train),np.array(Y_train)
print(X_train)
print("*****")
print(Y_train)
X_train=np.reshape(X_train,(X_train.shape[0],X_train.shape[1],1))
print("*****")
print(X_train)
regressor=Sequential()
regressor.add(LSTM(units=50,return_sequences=True,input_shape=(X_train.shape[1],1)))
regressor.add(Dropout(0.2))
regressor.add(LSTM(units=50,return_sequences=True))
regressor.add(Dropout(0.2))
```

Output:

```
[0.08581368 0.09701243 0.09433366 ... 0.07846566 0.08034452 0.08497656]
[0.09701243 0.09433366 0.09156187 ... 0.08034452 0.08497656 0.08627874]
[0.09433366 0.09156187 0.07984225 ... 0.08497656 0.08627874 0.08471612]
...
[[0.92106928 0.92438053 0.93048218 ... 0.95475854 0.95204256 0.95163331]
[0.92438053 0.93048218 0.9299055 ... 0.95204256 0.95163331 0.95725128]
[0.93048218 0.9299055 0.93113327 ... 0.95163331 0.95725128 0.93796041]]
*****
[0.08627874 0.08471612 0.07454052 ... 0.95725128 0.93796041 0.93688146]
*****
[[[0.08581368]
[0.09701243]
[0.09433366]
...
[0.07846566]
[0.08034452]
[0.08497656]]]

[[0.09701243]
[0.09433366]
[0.09156187]]]
```

```

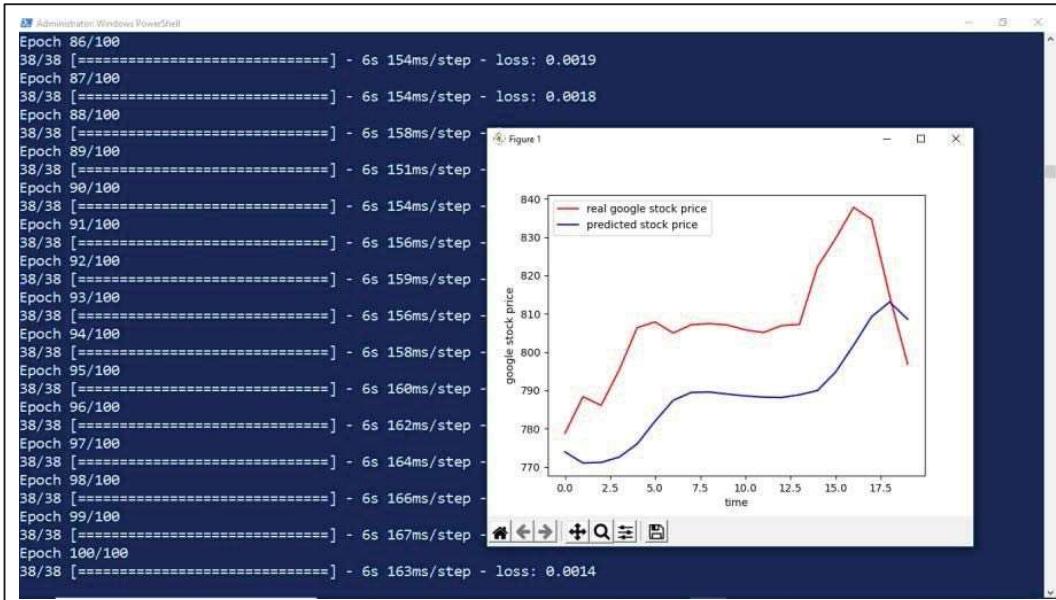
[[0.09433366]
[0.09156187]
[0.07984225]
...
[0.08497656]
[0.08627874]
[0.08471612]]
...
[[0.92106928]
[0.92438053]
[0.93048218]
...
[0.95475854]
[0.95204256]
[0.95163331]]
[[0.92438053]
[0.93048218]
[0.9299055 ]
...
[0.95204256]
[0.95163331]
[0.95725128]]
[[0.93048218]
[0.9299055 ]
[0.93113327]
...
[0.95163331]
[0.95725128]
[0.93796041]]]

```

```

In [ ]: regressor.compile(optimizer='adam',loss='mean_squared_error')
regressor.fit(X_train,Y_train,epochs=100,batch_size=32)
dataset_test=pd.read_csv('Google_stock_price.csv')
real_stock_price=dataset_test.iloc[:,1:2].values
dataset_total=pd.concat((dataset_train['Open'],dataset_test['Open']),axis=0)
inputs=dataset_total[len(dataset_total)-len(dataset_test)-60: ].values
inputs=inputs.reshape(-1,1)
inputs=sc.transform(inputs)
X_test=[]
for i in range(60,80):
    X_test.append(inputs[i-60:i,0])
X_test=np.array(X_test)
X_test=np.reshape(X_test,(X_test.shape[0],X_test.shape[1],1))
predicted_stock_price=regressor.predict(X_test)
predicted_stock_price=sc.inverse_transform(predicted_stock_price)
plt.plot(real_stock_price,color='red',label='real google stock price')
plt.plot(predicted_stock_price,color='blue',label='predicted stock price')
plt.xlabel('time')
plt.ylabel('google stock price')
plt.legend()
plt.show()

```



PRACTICAL 8

AIM: Performing encoding and decoding of images using deep autoencoder.

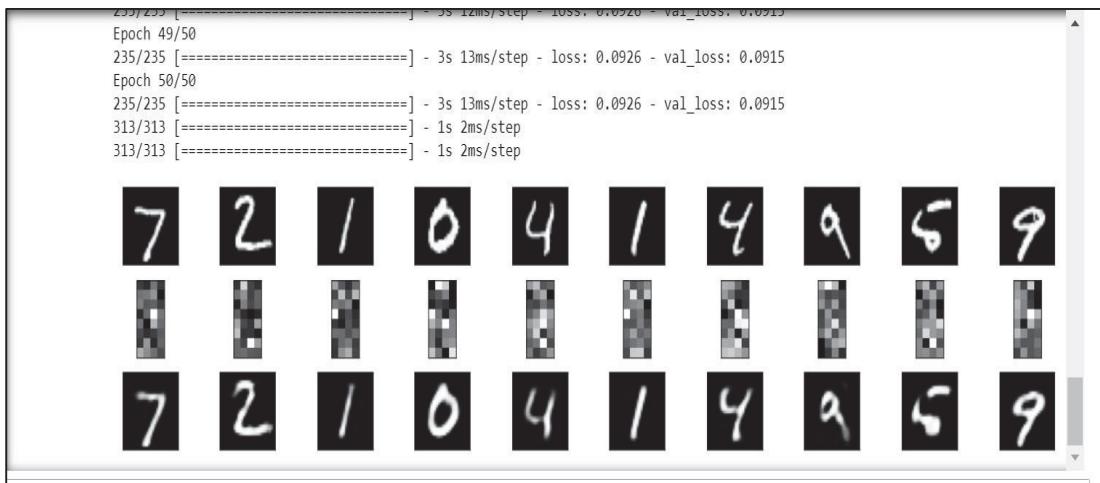
CODE:

```
In [1]: import keras
from keras import layers
from keras.datasets import mnist
import numpy as np
encoding_dim=32
#this is our input image
input_img=keras.Input(shape=(784,))
#"encoded" is the encoded representation of the input
encoded=layers.Dense(encoding_dim, activation='relu')(input_img)
#"decoded" is the lossy reconstruction of the input
decoded=layers.Dense(784, activation='sigmoid')(encoded)
#creating autoencoder model
autoencoder=keras.Model(input_img,decoded)
#create the encoder model
encoder=keras.Model(input_img,encoded)
encoded_input=keras.Input(shape=(encoding_dim,))
#Retrieve the last layer of the autoencoder model
decoder_layer=autoencoder.layers[-1]
#create the decoder model
decoder=keras.Model(encoded_input,decoder_layer(encoded_input))
autoencoder.compile(optimizer='adam',loss='binary_crossentropy')
#scale and make train and test dataset
(X_train,_) ,(X_test,_) =mnist.load_data()
X_train=X_train.astype('float32')/255.
X_test=X_test.astype('float32')/255.
X_train=X_train.reshape((len(X_train),np.prod(X_train.shape[1:])))
X_test=X_test.reshape((len(X_test),np.prod(X_test.shape[1:])))
print(X_train.shape)
print(X_test.shape)
#train autoencoder with training dataset
autoencoder.fit(X_train,X_train,
 epochs=50,
 batch_size=256,
```

```
print(X_train.shape)
print(X_test.shape)
#train autoencoder with training dataset
autoencoder.fit(X_train,X_train,
 epochs=50,
 batch_size=256,
 shuffle=True,
 validation_data=(X_test,X_test))
encoded_imgs=encoder.predict(X_test)
decoded_imgs=decoder.predict(encoded_imgs)
import matplotlib.pyplot as plt
n = 10 # How many digits we will display
plt.figure(figsize=(10, 4))
for i in range(10):
    # display original
    ax = plt.subplot(3, 20, i + 1)
    plt.imshow(X_test[i].reshape(28, 28))
    plt.gray()
    ax.get_xaxis().set_visible(False)
    ax.get_yaxis().set_visible(False)
    # display encoded image
    ax = plt.subplot(3, 20, i + 1 + 20)
    plt.imshow(encoded_imgs[i].reshape(8,4))
    plt.gray()
    ax.get_xaxis().set_visible(False)
    ax.get_yaxis().set_visible(False)
    # display reconstruction
    ax = plt.subplot(3, 20, 2*20 +i+ 1)
    plt.imshow(decoded_imgs[i].reshape(28, 28))
    plt.gray()
    ax.get_xaxis().set_visible(False)
    ax.get_yaxis().set_visible(False)
plt.show()
```

OUTPUT:

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 [=====] - 43s 4us/step
(60000, 784)
(10000, 784)
Epoch 1/50
235/235 [=====] - 4s 12ms/step - loss: 0.2756 - val_loss: 0.1899
Epoch 2/50
235/235 [=====] - 4s 16ms/step - loss: 0.1716 - val_loss: 0.1545
Epoch 3/50
235/235 [=====] - 3s 14ms/step - loss: 0.1447 - val_loss: 0.1326
Epoch 4/50
235/235 [=====] - 3s 11ms/step - loss: 0.1274 - val_loss: 0.1202
Epoch 5/50
235/235 [=====] - 3s 12ms/step - loss: 0.1173 - val_loss: 0.1121
Epoch 6/50
235/235 [=====] - 3s 12ms/step - loss: 0.1104 - val_loss: 0.1065
Epoch 7/50
235/235 [=====] - 3s 15ms/step - loss: 0.1054 - val_loss: 0.1021
Epoch 8/50
```



PRACTICAL 9

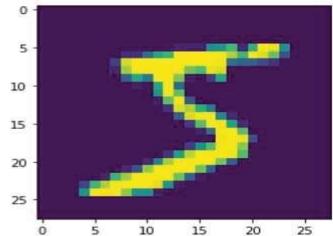
AIM: Implementation of convolutional neural network to predict numbers from number images

CODE & OUTPUT:

jupyter PRAC9 Last Checkpoint: 3 minutes ago (unsaved changes)

In [1]:

```
from keras.datasets import mnist
from keras.utils import to_categorical
from keras.models import Sequential
from keras.layers import Dense,Conv2D,Flatten
import matplotlib.pyplot as plt
#download mnist data and split into train and test sets
(X_train,Y_train),(X_test,Y_test)=mnist.load_data()
#plot the first image in the dataset
plt.imshow(X_train[0])
plt.show()
```



In [2]:

```
print(X_train[0].shape)
X_train=X_train.reshape(60000,28,28,1)
X_test=X_test.reshape(10000,28,28,1)
Y_train=to_categorical(Y_train)
Y_test=to_categorical(Y_test)
Y_train[0]
print(Y_train[0])
```

(28, 28)
[0. 0. 0. 0. 0. 1. 0. 0. 0. 0.]

```
In [3]: model=Sequential()
#add model layers
#learn image features
model.add(Conv2D(64,kernel_size=3,activation='relu',input_shape=(28,28,1)))
model.add(Conv2D(32,kernel_size=3,activation='relu'))
model.add(Flatten())
model.add(Dense(10,activation='softmax'))
model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])
#train
model.fit(X_train,Y_train,validation_data=(X_test,Y_test),epochs=3)
print(model.predict(X_test[:4]))
#actual results for 1st 4 images in the test set
print(Y_test[:4])

Epoch 1/3
1875/1875 [=====] - 201s 107ms/step - loss: 0.2541 - accuracy: 0.9520 - val_loss: 0.0963 - val_accuracy: 0.9714
Epoch 2/3
1875/1875 [=====] - 173s 92ms/step - loss: 0.0684 - accuracy: 0.9796 - val_loss: 0.0816 - val_accuracy: 0.9753
Epoch 3/3
1875/1875 [=====] - 180s 96ms/step - loss: 0.0479 - accuracy: 0.9849 - val_loss: 0.1011 - val_accuracy: 0.9743
```

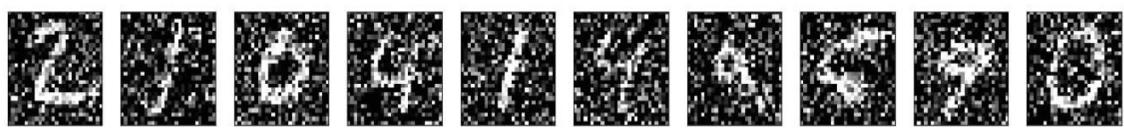
```
Epoch 1/3
1875/1875 [=====] - 201s 107ms/step - loss: 0.2541 - accuracy: 0.9520 - val_loss: 0.0963 - val_accuracy: 0.9714
Epoch 2/3
1875/1875 [=====] - 173s 92ms/step - loss: 0.0684 - accuracy: 0.9796 - val_loss: 0.0816 - val_accuracy: 0.9753
Epoch 3/3
1875/1875 [=====] - 180s 96ms/step - loss: 0.0479 - accuracy: 0.9849 - val_loss: 0.1011 - val_accuracy: 0.9743
1/1 [=====] - 0s 187ms/step
[[1.76193229e-08 5.17589769e-13 1.27019305e-07 2.36613255e-06
 4.52036629e-13 1.80279767e-11 4.82169312e-15 9.99997497e-01
 2.48748737e-08 9.88452098e-10]
[2.94664765e-10 6.09432573e-05 9.99938965e-01 9.68984781e-10
 4.67801145e-12 2.16221369e-13 9.06896886e-08 4.22226781e-15
 3.89350374e-09 6.33098090e-15]
[1.30127512e-06 9.99911308e-01 7.55180736e-07 6.27240269e-08
 1.10290584e-05 1.53752826e-05 8.20892467e-07 6.14862665e-06
 5.28885648e-05 2.15647262e-07]
[9.99999404e-01 3.44014366e-11 2.22936958e-08 4.20287589e-12
 1.45322955e-11 4.09832479e-09 5.80229084e-07 4.25992158e-11
 4.5494961e-10 4.17157553e-09]]
[[0. 0. 0. 0. 0. 1. 0. 0.]
 [0. 0. 1. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0. 0.]
 [1. 0. 0. 0. 0. 0. 0. 0.]]
```

PRACTICAL 10

AIM: Denoising of images using auto encoder.

CODE & OUTPUT:

```
import keras
from keras.datasets import mnist
from keras import layers
import numpy as np
from keras.callbacks import TensorBoard
import matplotlib.pyplot as plt
(X_train,_),(X_test,_)=mnist.load_data()
X_train=X_train.astype('float32')/255.
X_test=X_test.astype('float32')/255.
X_train=np.reshape(X_train,(len(X_train),28,28,1))
X_test=np.reshape(X_test,(len(X_test),28,28,1))
noise_factor=0.5
X_train_noisy=X_train+noise_factor*np.random.normal(loc=0.0,scale=1.0,size=X_train.shape)
X_test_noisy=X_test+noise_factor*np.random.normal(loc=0.0,scale=1.0,size=X_test.shape)
X_train_noisy=np.clip(X_train_noisy,0.,1.)
X_test_noisy=np.clip(X_test_noisy,0.,1.)
n=10
plt.figure(figsize=(20,2))
for i in range(1,n+1):
    ax=plt.subplot(1,n,i)
    plt.imshow(X_test_noisy[i].reshape(28,28))
    plt.gray()
    ax.get_xaxis().set_visible(False)
    ax.get_yaxis().set_visible(False)
plt.show()
input_img=keras.Input(shape=(28,28,1))
x=layers.Conv2D(32,(3,3),activation='relu',padding='same')(input_img)
x=layers.MaxPooling2D((2,2),padding='same')(x)
x=layers.Conv2D(32,(3,3),activation='relu',padding='same')(x)
encoded=layers.MaxPooling2D((2,2),padding='same')(x)
x=layers.Conv2D(32,(3,3),activation='relu',padding='same')(encoded)
x=layers.UpSampling2D((2,2))(x)
x=layers.Conv2D(32,(3,3),activation='relu',padding='same')(x)
x=layers.UpSampling2D((2,2))(x)
x=layers.Conv2D(1,(3,3),activation='sigmoid',padding='same')(x)
decoded=layers.Conv2D(1,(3,3),activation='sigmoid',padding='same')(x)
autoencoder=keras.Model(input_img,decoded)
autoencoder.compile(optimizer='adam',loss='binary_crossentropy')
autoencoder.fit(X_train_noisy,X_train,
                epochs=3,
                batch_size=128,
                shuffle=True,
                validation_data=(X_test_noisy,X_test),
                callbacks=[TensorBoard(log_dir='/tmp/tb',histogram_freq=0,write_graph=False)])
predictions=autoencoder.predict(X_test_noisy)
m=10
plt.figure(figsize=(20,2))
for i in range(1,m+1):
    ax=plt.subplot(1,m,i)
    plt.imshow(predictions[i].reshape(28,28))
    plt.gray()
    ax.get_xaxis().set_visible(False)
    ax.get_yaxis().set_visible(False)
plt.show()
```



Epoch 1/3

469/469 [=====] - 155s 328ms/step - loss: 0.1734 - val_loss: 0.1182

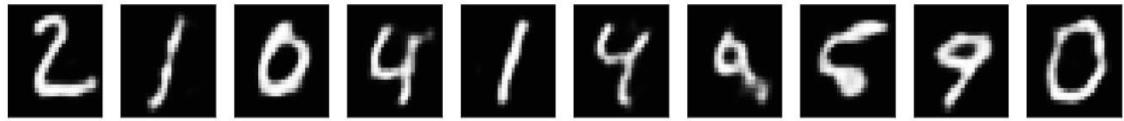
Epoch 2/3

469/469 [=====] - 149s 318ms/step - loss: 0.1149 - val_loss: 0.1100

Epoch 3/3

469/469 [=====] - 146s 311ms/step - loss: 0.1088 - val_loss: 0.1056

313/313 [=====] - 7s 23ms/step



University Of Mumbai
Centre for Distance & Online Education



Dr.Shangar Dayal Sharama Bhavan, Kalina, Vidanagari, Santacruz (E),
Mumbai-400 098.

Certificate

This is to certify that

Mr. **MAKRAND GHAG** Application ID: **93669**, Seat No: **2900084** from **Rizvi College of Arts, Science and Commerce Bandra(W)**, Mumbai 400 050 has successfully completed all the practical of Paper III titled **CYBER FORENSICS** for M.Sc (IT) Part II in the academic year 2023-2024.

Section I _____

Section II _____

MSc (IT) Co-ordinator, IDOL

External Examiner

INDEX

PRACTICAL NO	TITLE
1	Creating Image of an Evidence(Logical / physical drive)
2	Creating an Image of content of folder from system
3	Analyse and investigate the image file provided in Autopsy
4	Capture and Analyse the network Packets
5	Investigate the packets provided in Wireshark
6	Perform Network packet analyse and process monitoring
7	Investigate the Mobile Device
8	Investigate Email File Given
9	Browser Forensic
10	Perform Stegnography

PRACTICAL 1

Aim: Creating a Forensic Image using FTK Imager/Encase Imager:

- Creating Forensic Image

- Check Integrity of Data

- Analyse Forensic Image

-
FTK Imager allows you to make several different types of forensic images. In addition, drive content and hash lists can be exported.

a) Creating Forensic Image

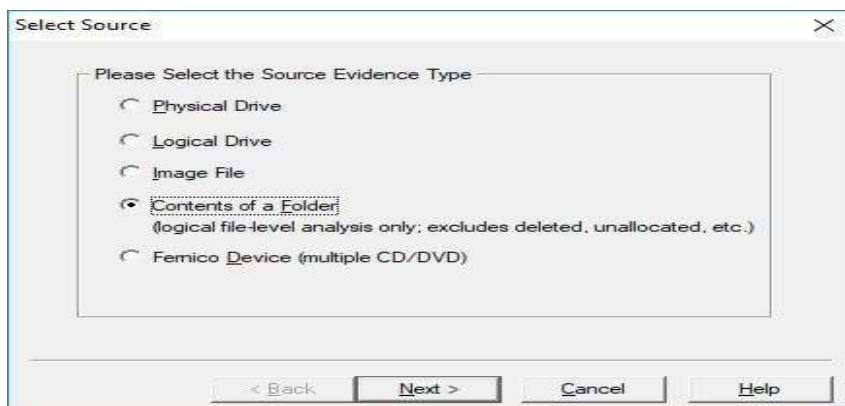
FTK Imager allows you to write an image file to a single destination or to simultaneously write multiple image

files to multiple destinations using the same source data or drive.

To create a forensic image

1. Click File > Create Disk Image.

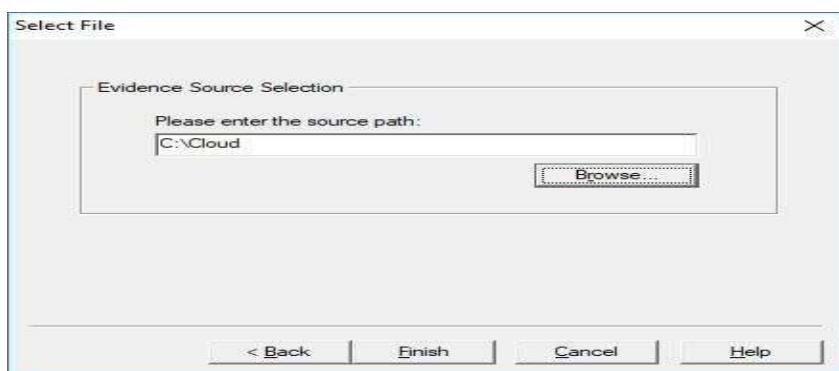
2. In the Select Source dialog box, select the source you want to make an image of.



3. Click Next.

4. Click Yes

5. Browse to the source of the image you want, and then click Finish.

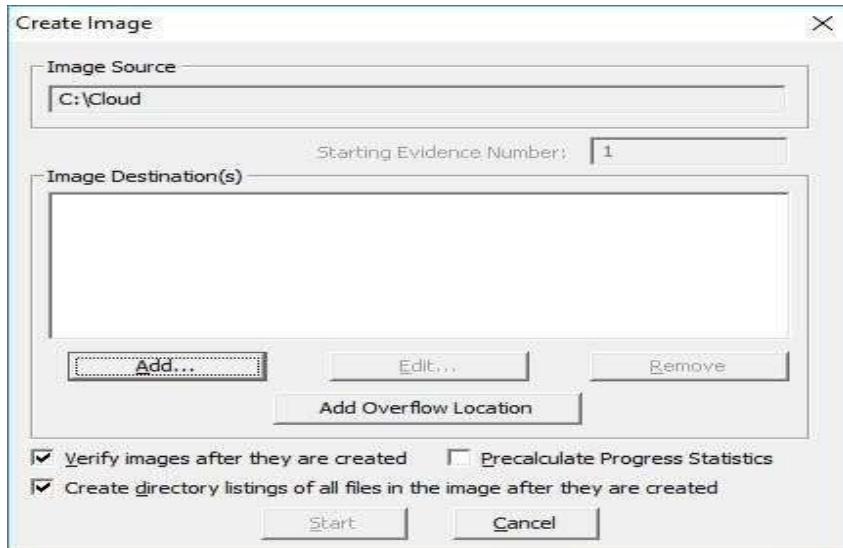


6. In the Create Image dialog,

6.a. Compare the stored hashes of your image content by checking the Verify images after they are created box. If a file doesn't have a hash, this option will generate one.

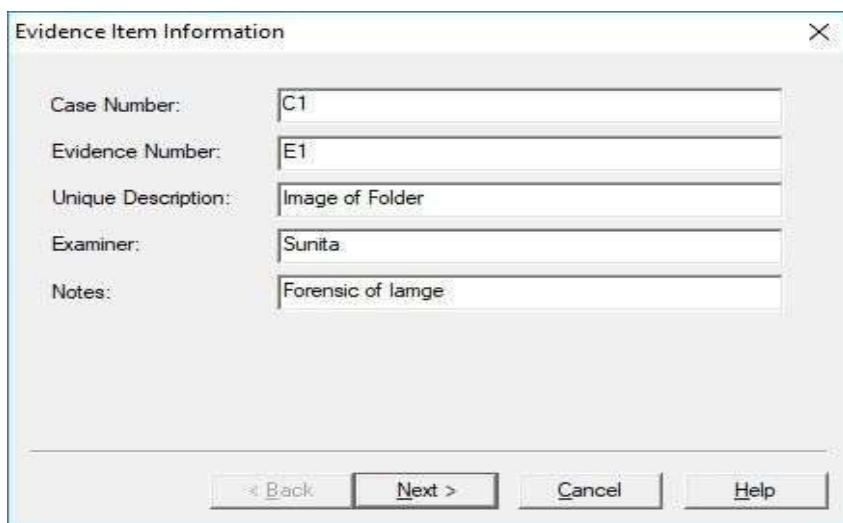
6.b. List the entire contents of your images with path, creation dates, whether files were deleted, and other metadata. The list is saved in tab-separated value (.CSV) format.

6.c. click Add.



7. Click Next.

8. Specify Evidence Item Information. All Evidence Item Information is optional, but it is helpful to have the information easily accessible in case it is called into question at any time after creation



9. Click Next.

10. In the Image Destination Folder field Click Browse to find and select the desired location.

11. In the Image Filename field, specify a name for the image file but do not specify a file extension

12.a.Specify the Image fragment Size:

Default Image Fragment Size = 1500 MB

The S01 format is limited by design to sizes between 1 MB and 2047 MB (2 GB). Compressed block pointers are 31-bit numbers (the high bit is a compressed flag), which limits the size of any one segment to two gigabytes.

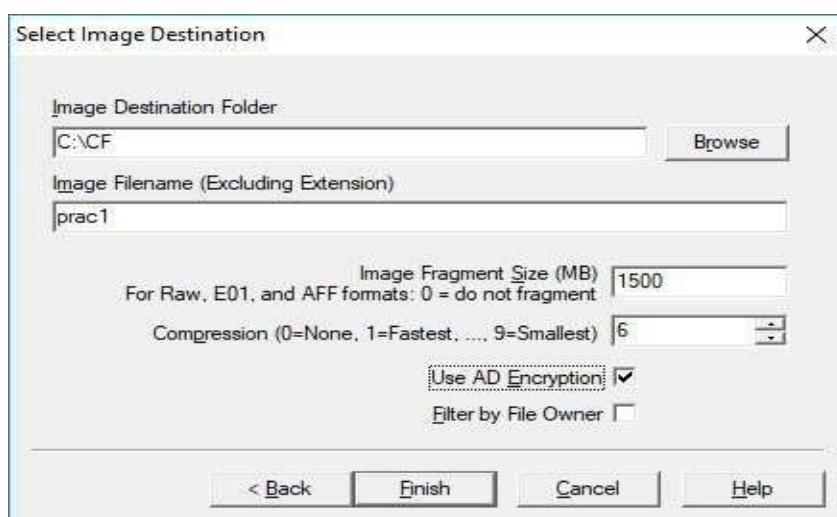
12.b. Select the compression level to use.

0=No Compression

1=Fastest, Least Compression (faster, and also slightly smaller than a 0-compression file)

9=Slowest, Most Compression (smallest file, slowest to create). Numbers between 1 and 9 produce an image with varying levels of compression to speed ratio.

13. To encrypt the new image with AD Encryption, mark the Use AD Encryption box.



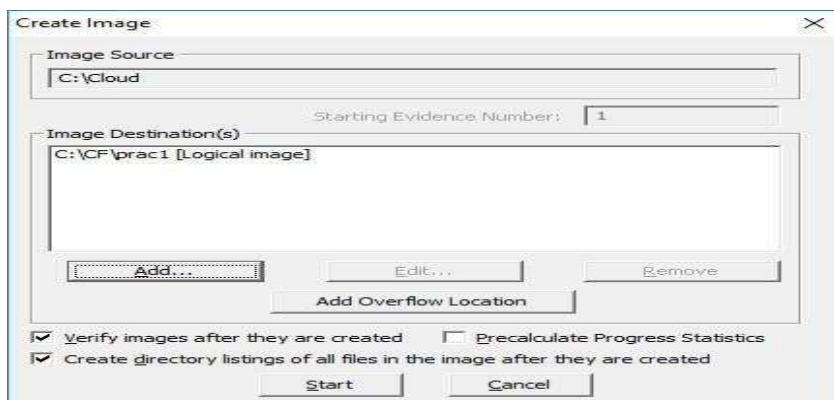
14. When AD Encryption is selected, you can choose between encrypting with a password, or encrypting with a certificate.



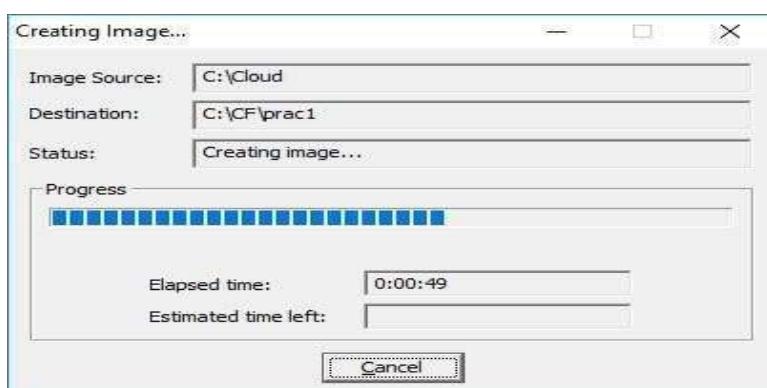
15. When encryption selections are made, click OK to save selections and return to the Create Image dialog.

16. To add another image destination (i.e., a different saved location or image file type), click Add, and repeat steps

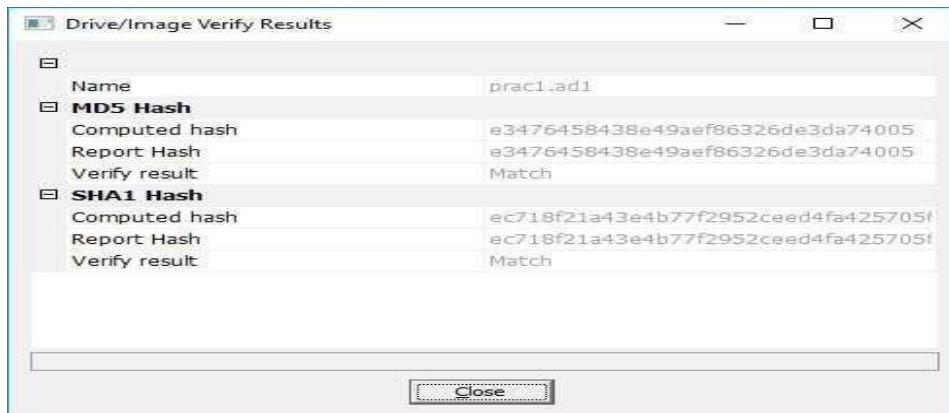
17. Click Start to begin the imaging process.



18. A progress dialog appears that shows the following

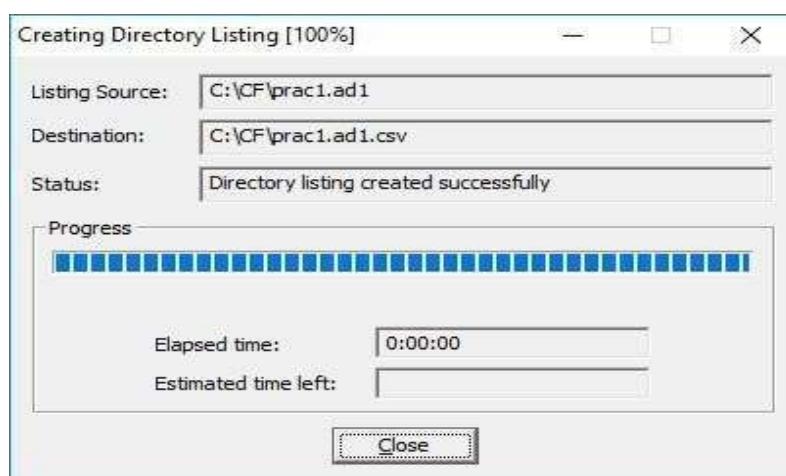


19. After the images are successfully created, the Drive/Image Verify Results box shows detailed image information, including MD5 and SHA1 check sums, and bad sectors.

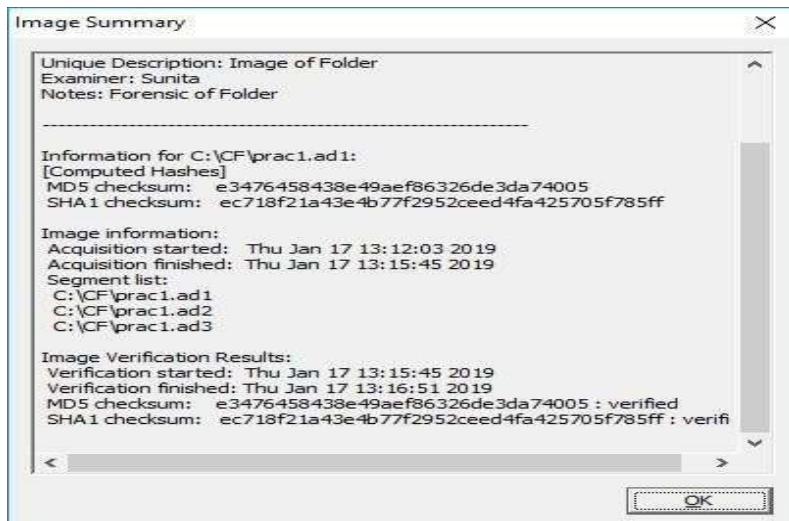


20. Click on Close to close the Drive/Image Verify Results box

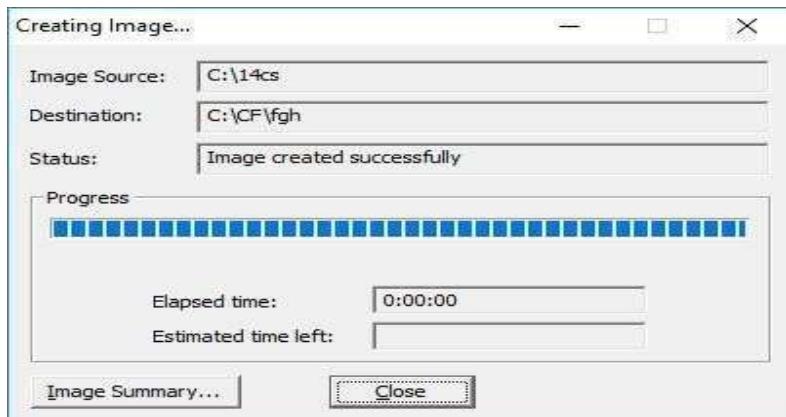
21. Click on Close to close the Creating Directory Listing dialog



22. Click Image Summary to close the Image Summary. The Image Summary also includes the data you entered in the Evidence Item Information dialog.



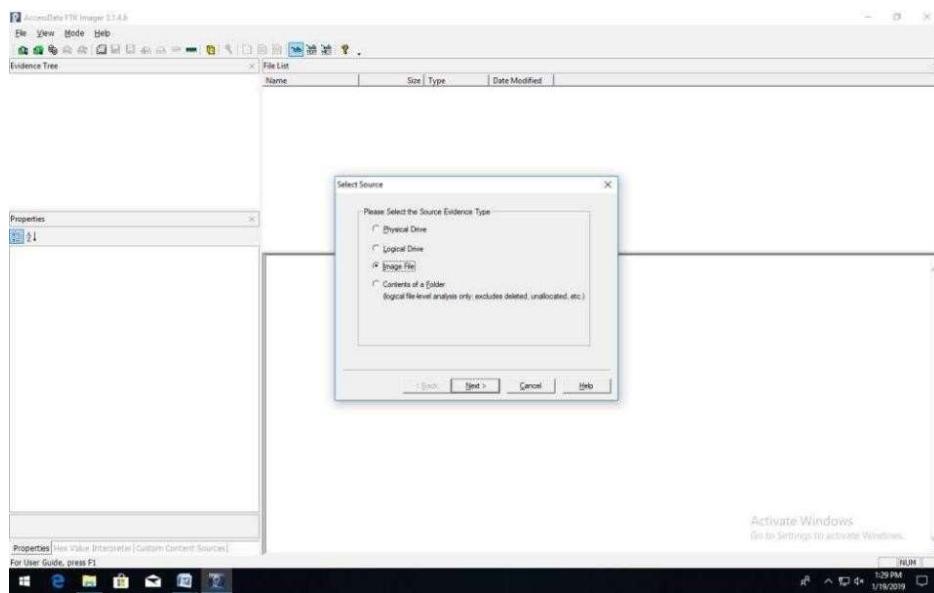
23. Click Close to exit back to Imager.



b) Check Integrity of Data

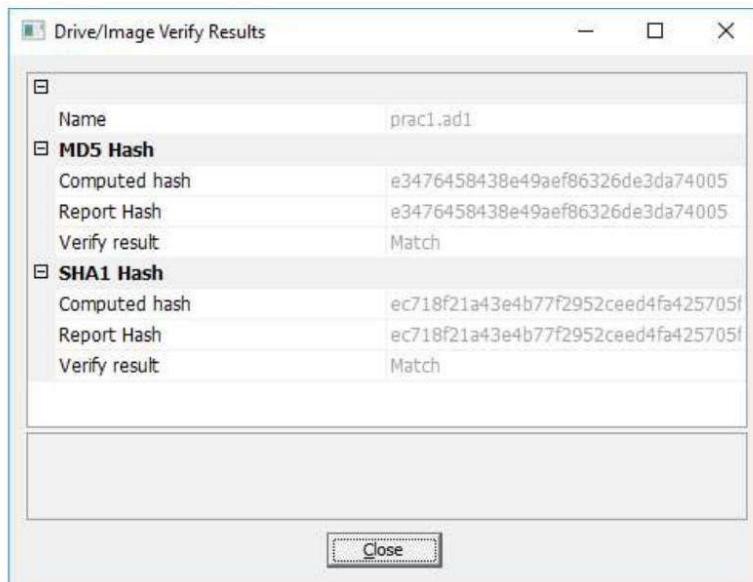
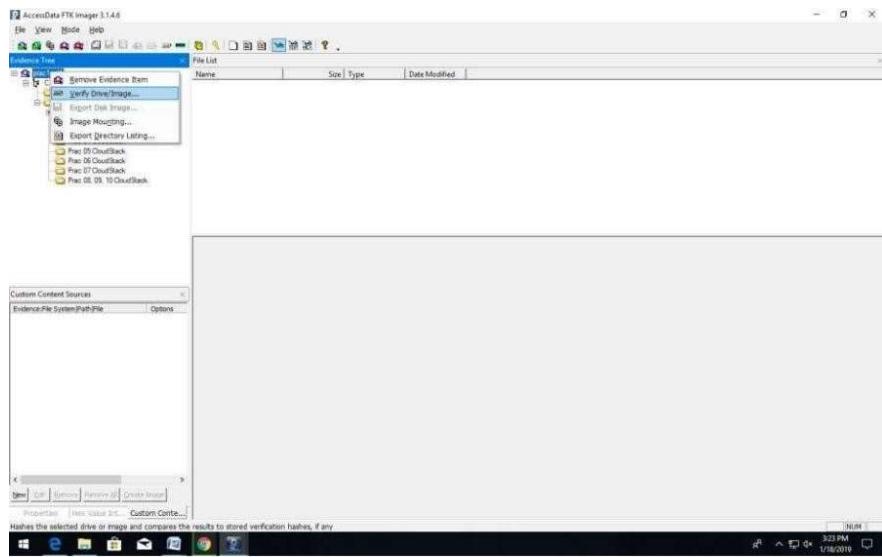
Hashing is the process of generating a unique value based on a file's contents. This value can then be used to prove that a copy of a file has not been altered in any way from the original file. It is computationally infeasible for an altered file to generate the same hash number as the original version of that file. The Export File Hash List feature in FTK Imager uses the MD5 and SHA1 hash algorithms to generate hash numbers for files.

Go to File->Add Evidence Item..->Select Option as Image File



Browse your image file->Click on Finish button-> enter password and Click on Ok button



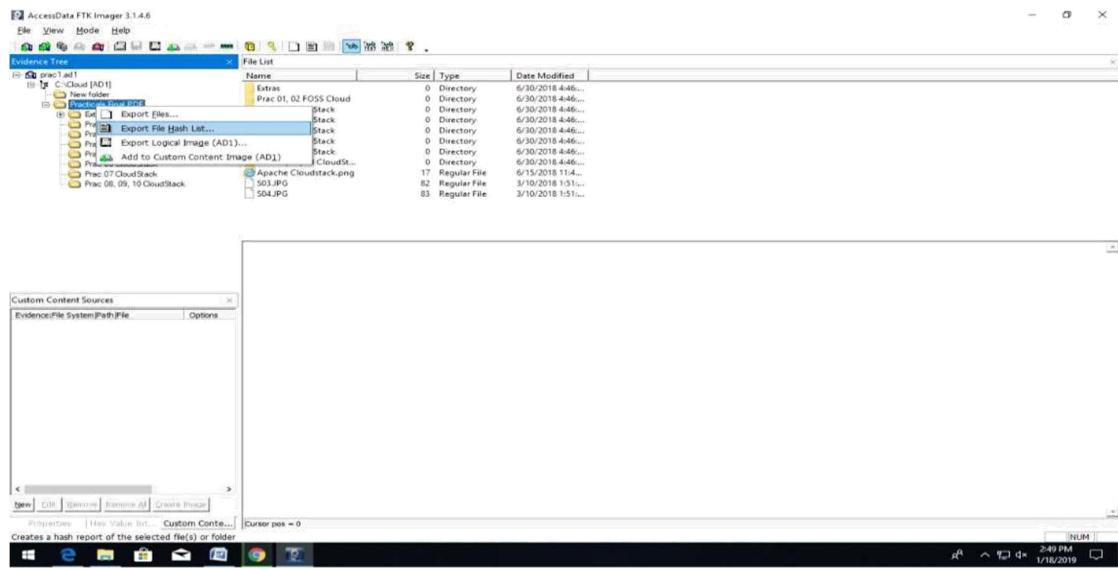


To generate and export hash values to a list

1. In the Evidence Tree, select the folder that contains the objects you want to hash. The object's contents are displayed in the File List.

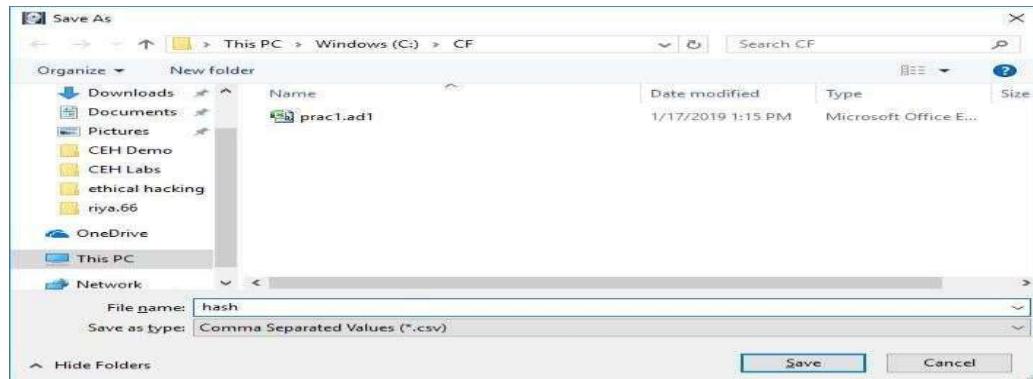
2. In the File List, select the folders or files you want to hash. If you select a folder, all the files contained in the folder and its sub folders are hashed.

3. Click File > Export File Hash List



4. In the Save As dialog, type a name for the file hash list in the File Name field.

5. Click Save.



The hash list is saved as a file of comma-separated values (*.CSV).

	A	B	C
1	M05	SHA1	FileName
2	53108117dbbd0e08c79ec329625c5e5	prac1.ad1\Cloud [AD1]\Practicals Final PDF\Apache Cloudstack.png	prac1.ad1\Cloud [AD1]\Practicals Final PDF\Extract\01 CloudStack Documentation.pdf
3	bfd13ea75f9a469c017b5121afe7b05	521d4e4ea4e0809907318a07e7af411e74a969777c	prac1.ad1\Cloud [AD1]\Practicals Final PDF\Extract\02 CloudStack Installation.pdf
4	ed794a747af5a52ba8e05491402416	0166310d0f2aae2c153cc2b12ed777dd9e019141	prac1.ad1\Cloud [AD1]\Practicals Final PDF\Extract\03 CloudStack Administration.pdf
5	5545e4580580e121fae0de1209902f26	9702e2b137b44013bc59777044e011ae180e	prac1.ad1\Cloud [AD1]\Practicals Final PDF\Extract\04 CloudStack Releases.pdf
6	63014654474a21897fa3d10e6664728	791c1715e72d48f1ce62d818ddc7a77702dc	prac1.ad1\Cloud [AD1]\Practicals Final PDF\Extract\05 DevCloud - Apache Cloudstack - Apache Software Foundation.pdf
7	4b1599dadeab059740f1c911e4da3aa3f	6982c74270a3d58f80d725ab79f1294d0f99e3	prac1.ad1\Cloud [AD1]\Practicals Final PDF\Extract\06 Setting up a CloudStack dev environment on Windows - Apache Cloudstack - Apache Software Foundation.pdf
8	daaf34e508833e00f32a2d103539	e4e4f17a33915b7911787079309aca6111e3c136	prac1.ad1\Cloud [AD1]\Practicals Final PDF\Extract\07 Quick Installation Guide for CentOS 6 [U+2014] Apache CloudStack Installation Doc
9	194-23747452845c5dca7600b1a	b105288e712097883e7663b101085cb3a4408	prac1.ad1\Cloud [AD1]\Practicals Final PDF\Extract\08 Quick Installation Guide for CentOS 6 [U+2014] Apache CloudStack Installation Doc
10	13e9c93301e0042525b15615e769e3	ea7b810bb824174ff5fbcbad1bda20069e3	prac1.ad1\Cloud [AD1]\Practicals Final PDF\Extract\09 PRAC 4 Working with Virtual Machines [U+2014] Apache CloudStack Admin
11	53aa765861ad004001620a0a82e20ff94a0d4	d8317d5861ad004001620a0a82e20ff94a0d4	prac1.ad1\Cloud [AD1]\Practicals Final PDF\Extract\10 PRAC 5 A.pdf
12	6c772e2166293778329490578b7ef	arc0404424234104068170fa384ee1a08c	prac1.ad1\Cloud [AD1]\Practicals Final PDF\Extract\11 PRAC 5 B How To Configure SSH Key-Based Authentication on a Linux System.pdf
13	f3b168985032a3959e99c1ac2079d	7979f34ecc14d661a999e9283a358fc5d64bb	prac1.ad1\Cloud [AD1]\Practicals Final PDF\Extract\12 PRAC 6 A Managing the Cloud [U+2014] Apache CloudStack Administration.pdf
14	347031611598be26a5000396f9a6	9175273452b75327ja0e079e1114ca5a654ab	prac1.ad1\Cloud [AD1]\Practicals Final PDF\Extract\13 PRAC 6 B Report CPU sockets - Apache Cloudstack - Apache Software Foundation.pdf
15	4ec12fcd773e35945d4b0d0026273e0d	cbebedcfcc2a0b7a011116571b9a968bc27730e0d	prac1.ad1\Cloud [AD1]\Practicals Final PDF\Extract\14 PRAC 8, 9, 10Managing the Cloud [U+2014] Apache CloudStack Administration.pdf
16	9368e79cb772558e40067a3b0a786227	f19616eb1b157860b505951296f330007719d	prac1.ad1\Cloud [AD1]\Practicals Final PDF\Extract\15 PRAC 10 Managing the Cloud [U+2014] Apache CloudStack Administration.pdf
17	9506e835029060601600a0da2a7e0f6a	91cbe0fe73495a6a017509794655e2a020310e	prac1.ad1\Cloud [AD1]\Practicals Final PDF\Extract\16 Recipes for Apache CloudStack [Book].pdf
18	4d0b2346a4ec2bf0f1d7f2b117e7e1f65	433a07136d6751f16038374a39f6cfa5358f8e0a	prac1.ad1\Cloud [AD1]\Practicals Final PDF\Extract\17 How to test Apache CloudStack 4.2 on your local machine - LeaseWeb failover.pdf
19	12f883770d013300e0e1f046567b627	c8e62f132d0b35e59a46a7672372b6ff2e1e0f	prac1.ad1\Cloud [AD1]\Practicals Final PDF\Extract\18 PL_P2.pdf
20	ea0d4ea7a045127fe0f212a98d9f195	de657282959c712a6762e7fb9b0fe0f3fc1c1b49	prac1.ad1\Cloud [AD1]\Practicals Final PDF\Extract\19 PL_P3.pdf
21	0623e80708a8a453e59e270a0f486b6	49e288371c083368112d31100eee01338c394705e2	prac1.ad1\Cloud [AD1]\Practicals Final PDF\Extract\20 PRAC 4, P4.pdf
22	419f98474593730454e206057170c5	3b06170a187b03d5f5761e3180a0d9ed5687e0a	prac1.ad1\Cloud [AD1]\Practicals Final PDF\Extract\21 Rohit Yadav - DevCloud for CloudStack Development.pdf
23	2d00cc2224a101342d41594657171c	9e012230402217b6330729177f7253867968	prac1.ad1\Cloud [AD1]\Practicals Final PDF\Extract\22 FOSS Cloud@ Installation Demo System - FOSS-Cloud.pdf
24	ea0522e1a3c0a9060614adfd107a	13c761821294821446c88f774fafe5a51e669	prac1.ad1\Cloud [AD1]\Practicals Final PDF\Extract\23 FOSS Cloud@ After Demo Installation.pdf
25	4d45150d153027079f7a48f6d0529	06297439472563144e7808336367536224843	prac1.ad1\Cloud [AD1]\Practicals Final PDF\Extract\24 PRAC 3.pdf
26	6010d1e069446f93d948c39f19b	01b52a860800800a0e12b0f72095e199d5a0b	prac1.ad1\Cloud [AD1]\Practicals Final PDF\Extract\25 PRAC 4.pdf
27	03045c3e0d7e6858c8b0f4d04742b	250b010bb4504745081b500772851996582	prac1.ad1\Cloud [AD1]\Practicals Final PDF\Extract\26 PRAC 5.pdf
28	37e113cd0852643c8207634a59b	0e1b81a1e969ce04a687095a6b5421a076d9	prac1.ad1\Cloud [AD1]\Practicals Final PDF\Extract\27 PRAC 5B.pdf
29	ed6eaaf11e7ef0805fcdf9055af435e	040401bc1f552c19b180e9aa8583e66756	prac1.ad1\Cloud [AD1]\Practicals Final PDF\Extract\28 PRAC 6.pdf
30	9ea6345c3140713739029a8802a2f	2c38f782d4e0f3f47b125d61003119c729e1	prac1.ad1\Cloud [AD1]\Practicals Final PDF\Extract\29 PRAC 7.pdf
31	942c48e11ed62c9add1a120770489	cefd5e3a8787559036979b0c352e9f1d8f4b	prac1.ad1\Cloud [AD1]\Practicals Final PDF\Extract\30 PRAC 8, 9, 10 CloudStack [Book].pdf
32	33050000000000000000000000000000	94a7a056119411401000794137455784376	prac1.ad1\Cloud [AD1]\Practicals Final PDF\Extract\31 PRAC 9.pdf

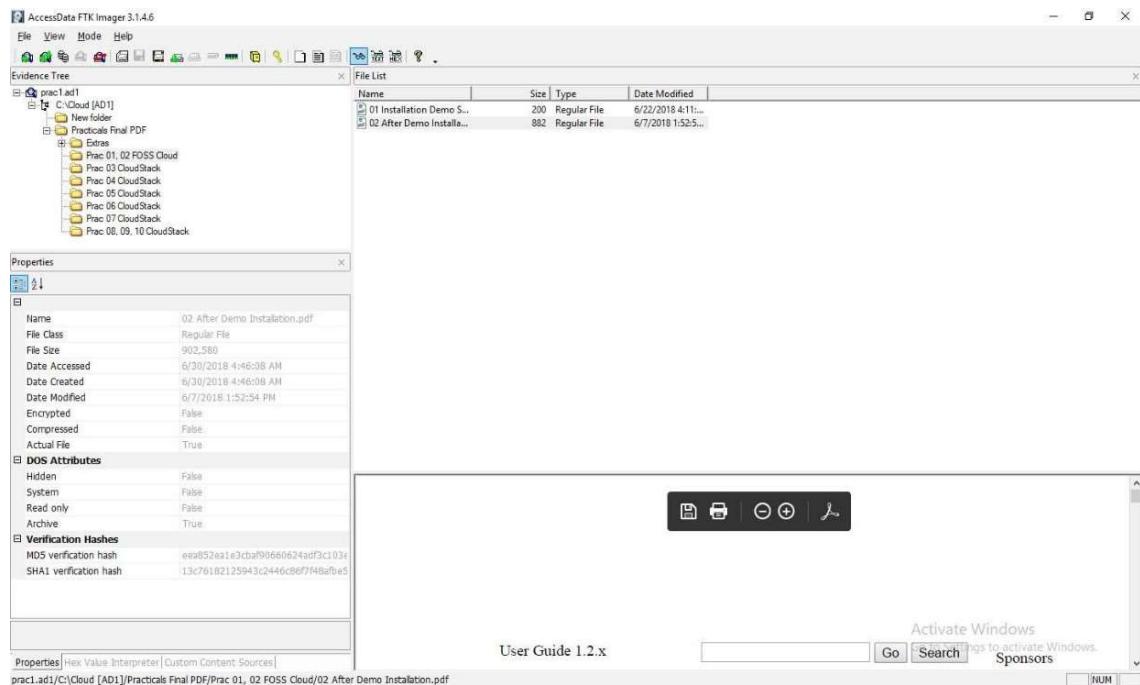
c) Analyze Forensic Image

a. Open Image using Add Evidence Item Option.

b. Select any folder from Evidence Tree.

c. Files will be displayed in File List

d. Select any File and you can view Properties of file in Properties Box



PRACTICAL 2

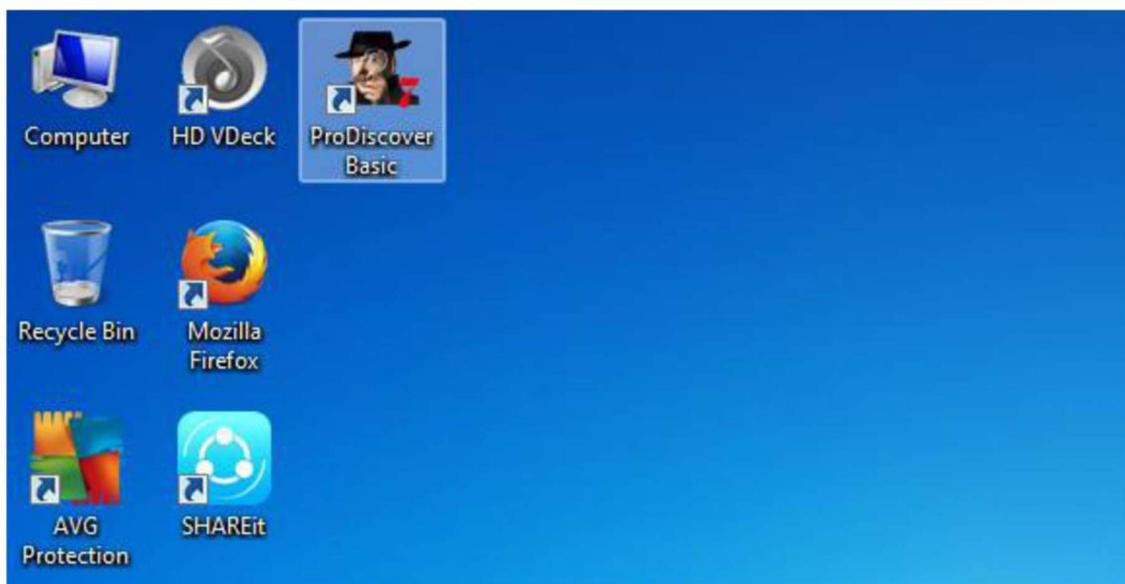
Aim: Creating an Image of content of folder from systemData

Acquisition:

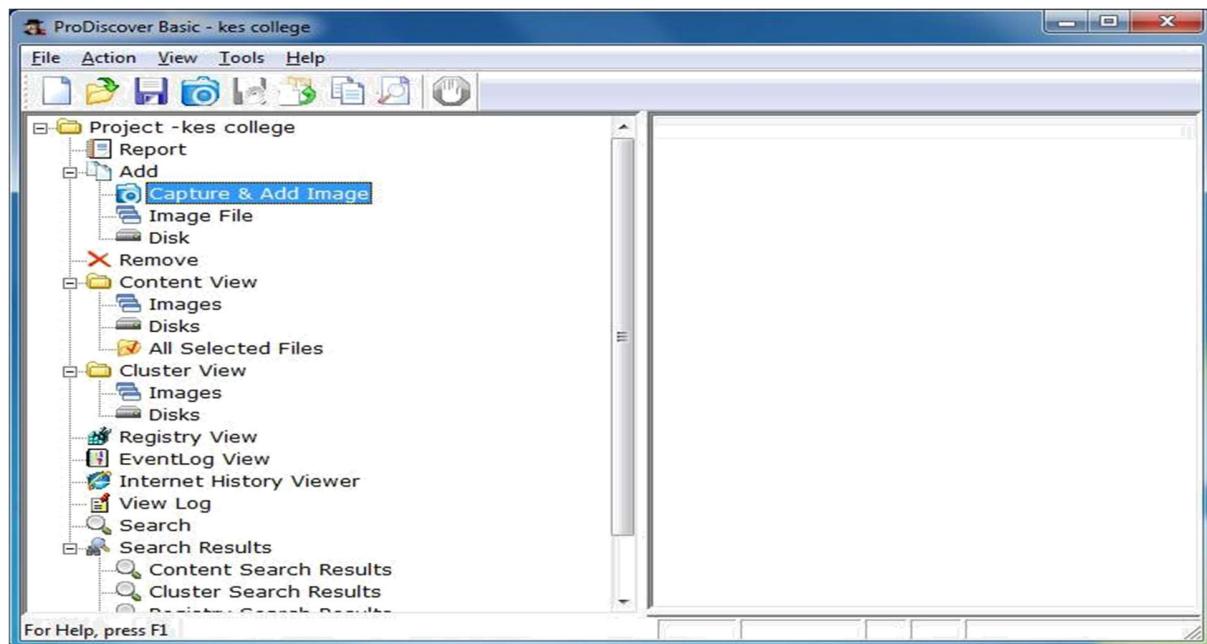
Perform data acquisition using:
USB Write Blocker + FTK Imager

Steps:

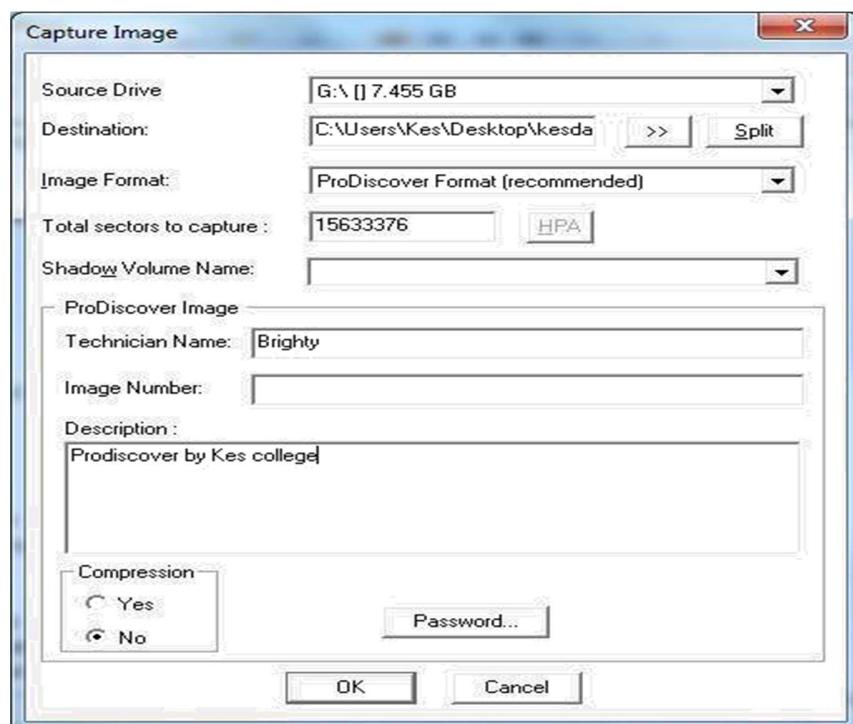
Step 1: First Open Prodiscover Basic and start with new case.



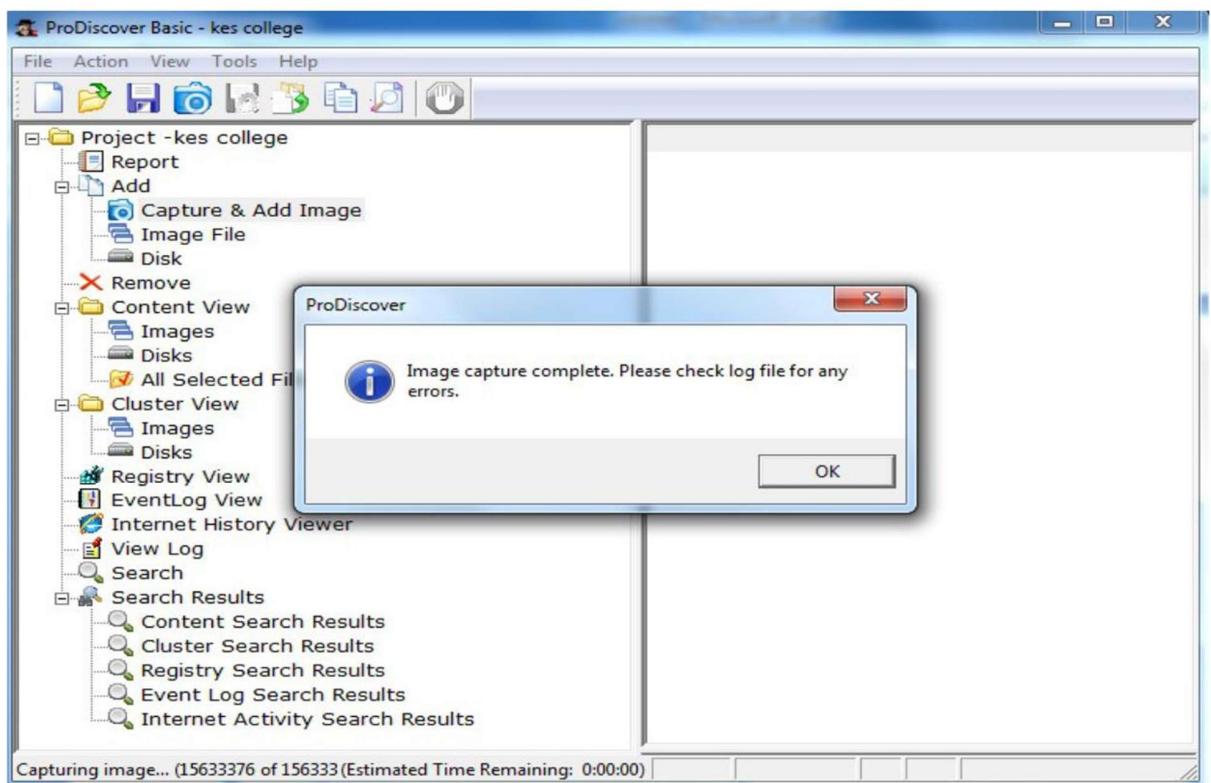
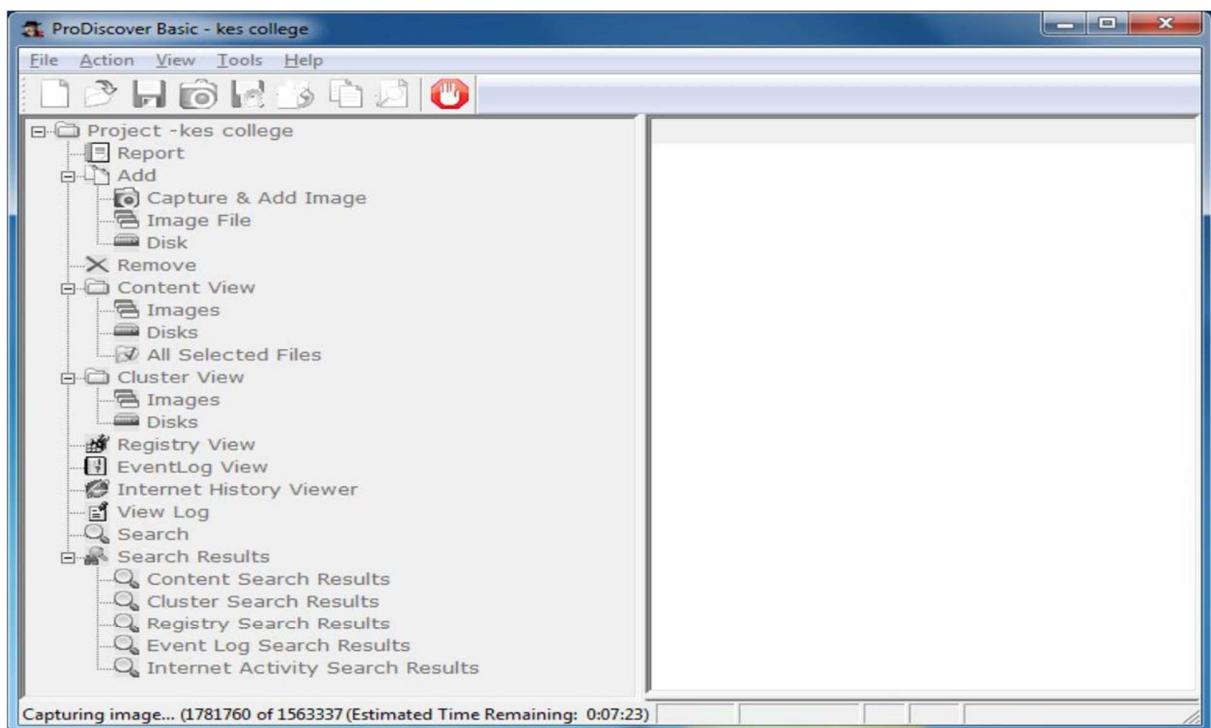
Step 2: The created project appears in left pane and select add>capture & add image.



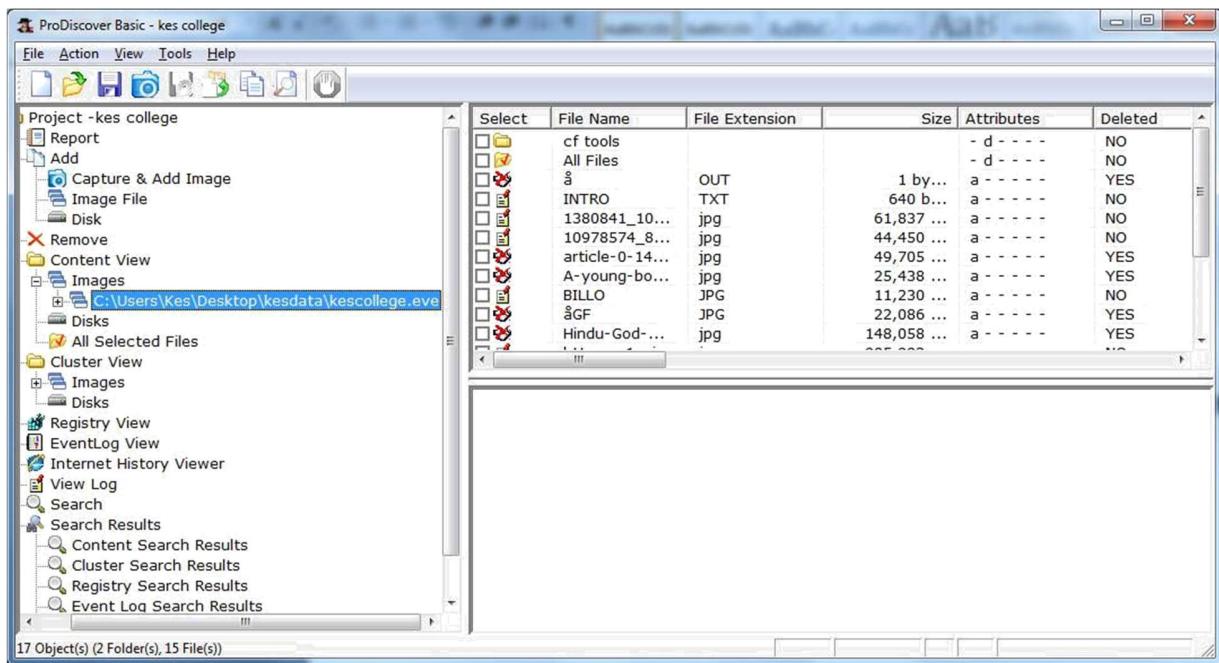
Step 3: fill the details as below. And click ok.



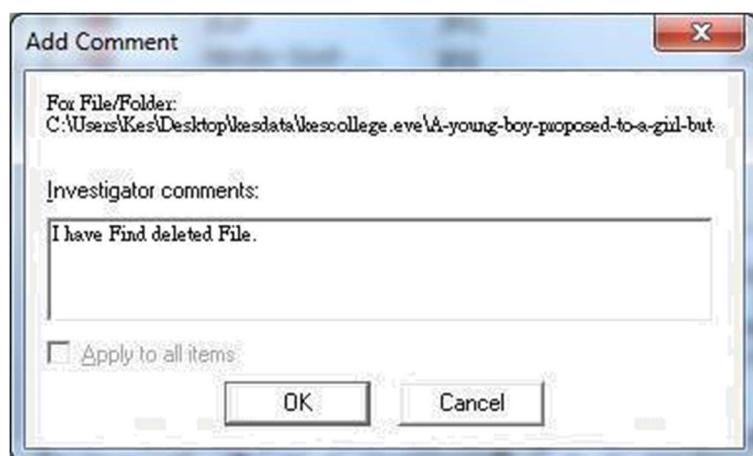
Step 4: capturing of image starts.



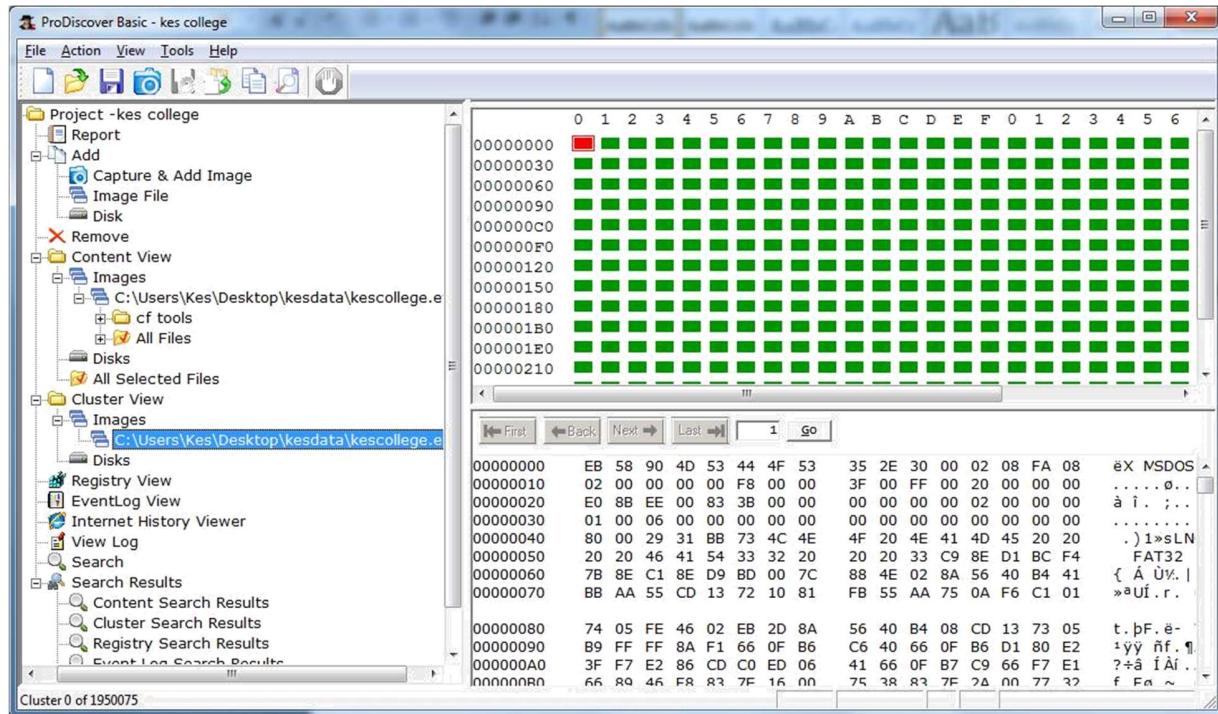
Step 5: Open the image created, go to Add > Images in left pane.



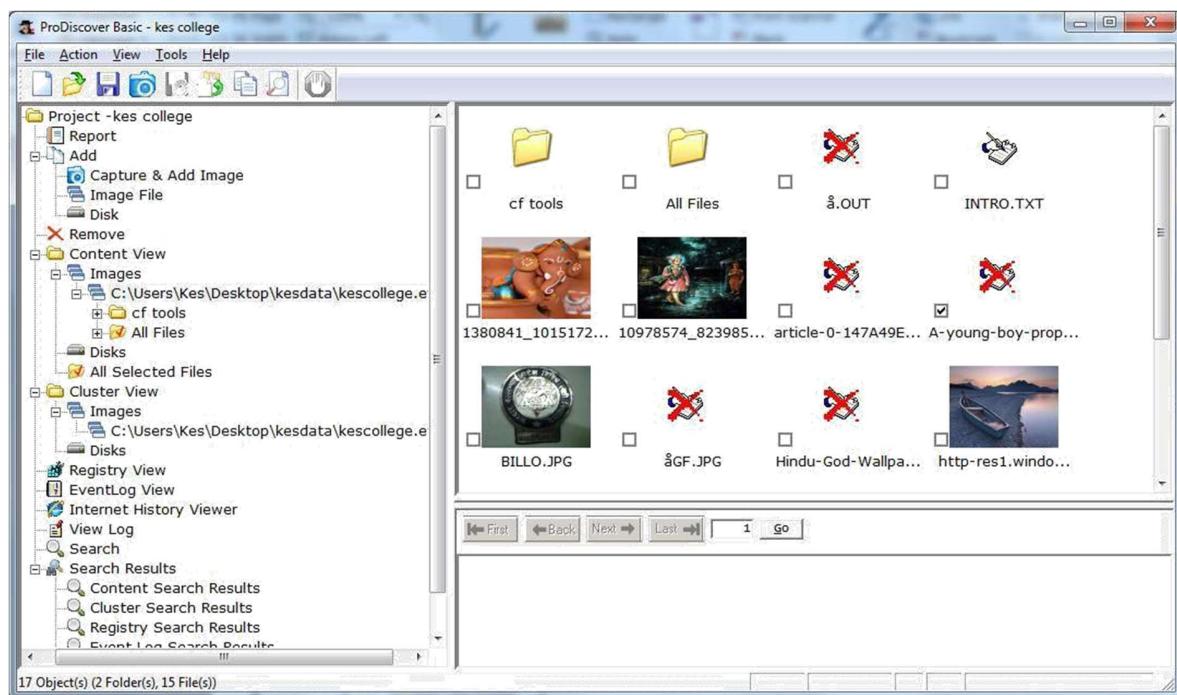
Step 6: Click on any File and type a comment.



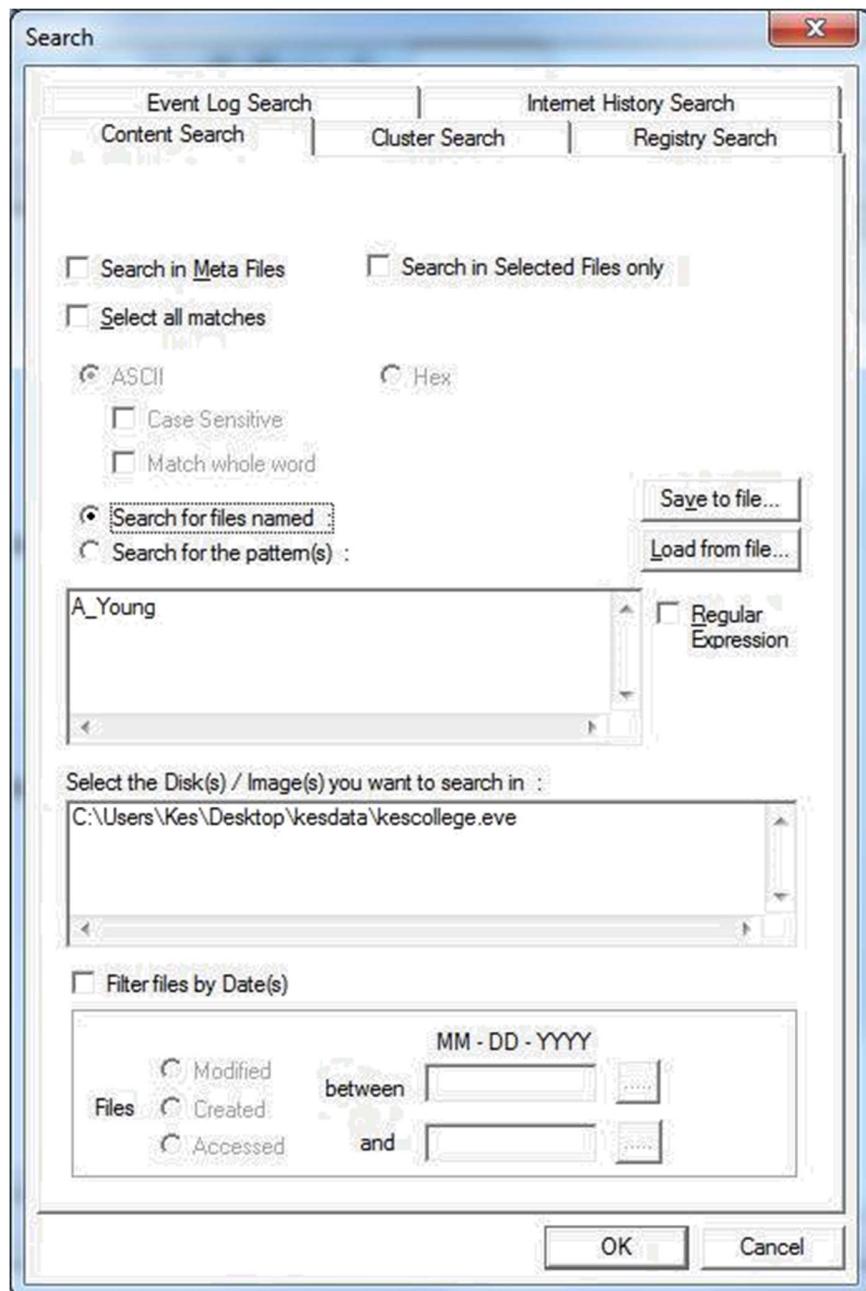
Step 7 : the cluster view is seen from the cluster view in left panel.



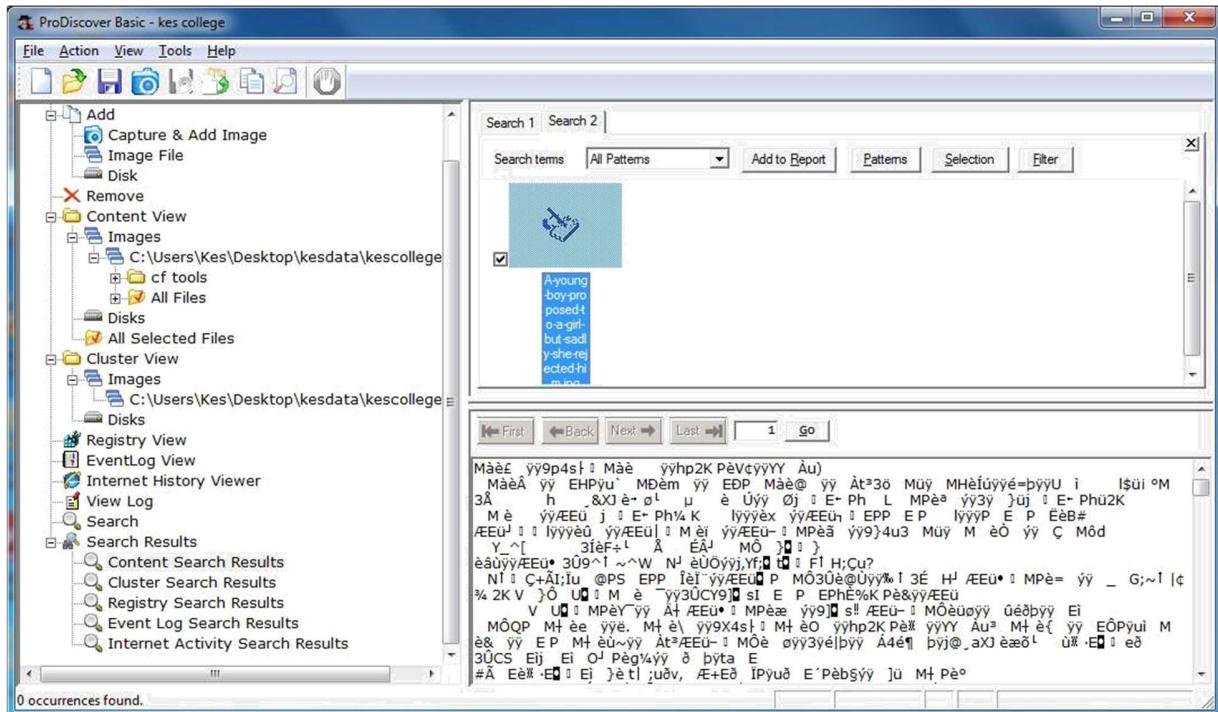
Step 8 : We can also view gallery view by Right Click.



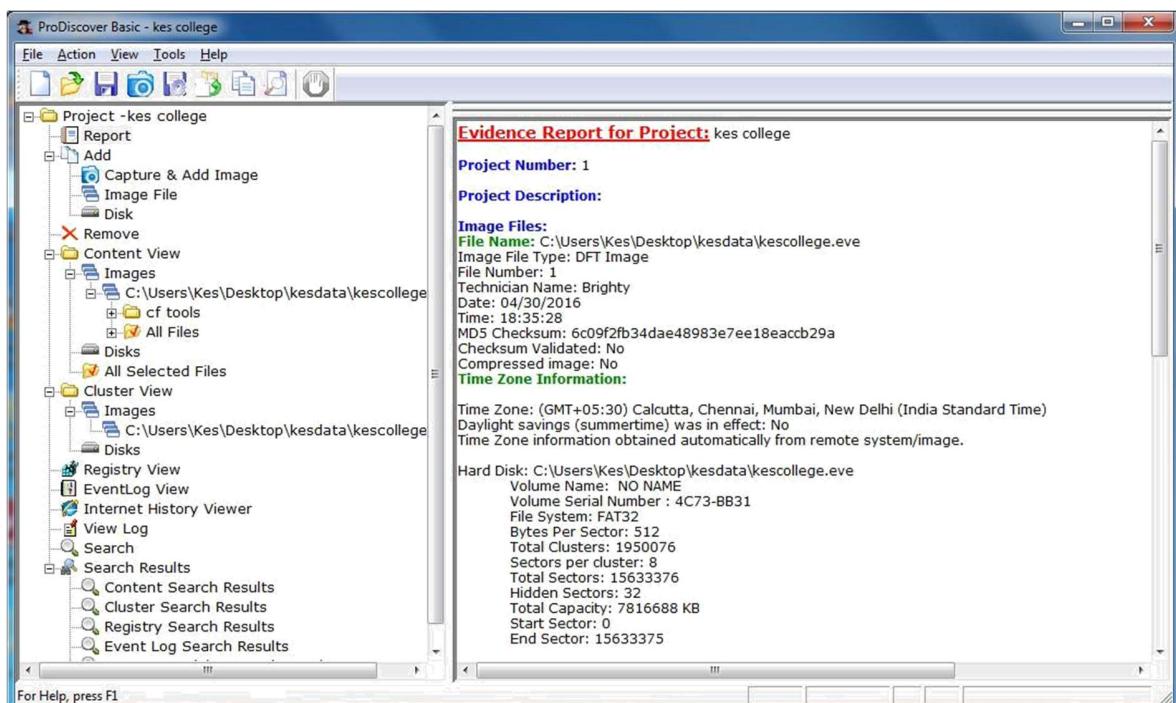
Step 9: Keyword search. Click on Search in left pane and Enter the filename to be searched in the image created.

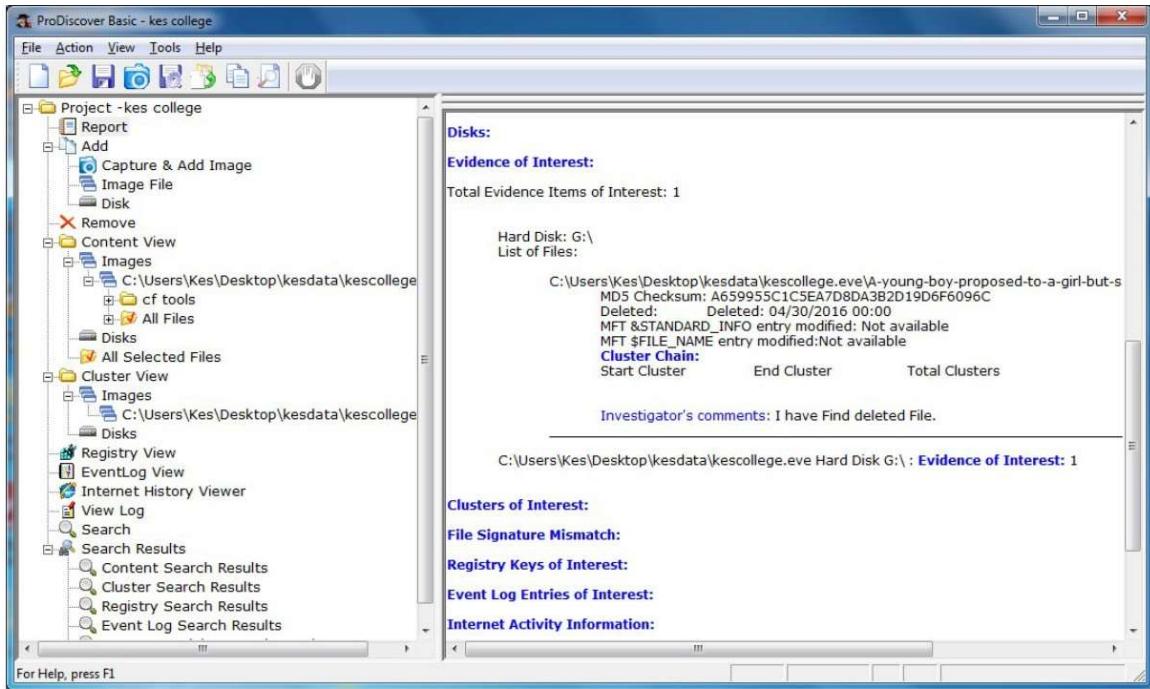


Step 10 : Output of Keyword search.



Step 11 : Click on View>Report.





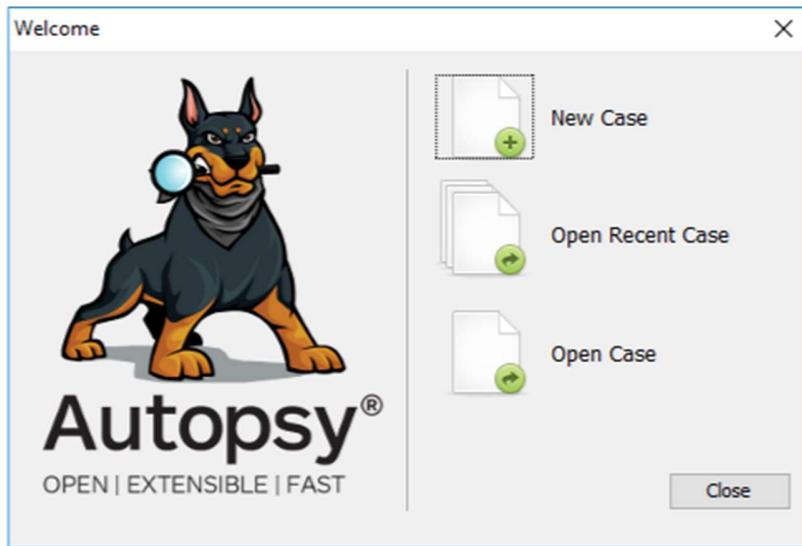
PRACTICAL 3

AIM: - Solve the Case study (image file) provide in lab using EncaseInvestigator or Autopsy.

Step 1: Open Autopsy



Step 2 : Click on new case



Step 3 : Enter details regarding the case and click on next button.

New Case Information

Steps

Case Information

1. Case Information
2. Optional Information

Case Name:

Base Directory:

Case Type: Single-user Multi-user

Case data will be stored in the following directory:

< Back Finish Cancel Help

Step 4 : Enter further details and click on next button

New Case Information

Steps

Optional Information

1. Case Information
2. Optional Information

Case

Number:

Examiner

Name:

Phone:

Email:

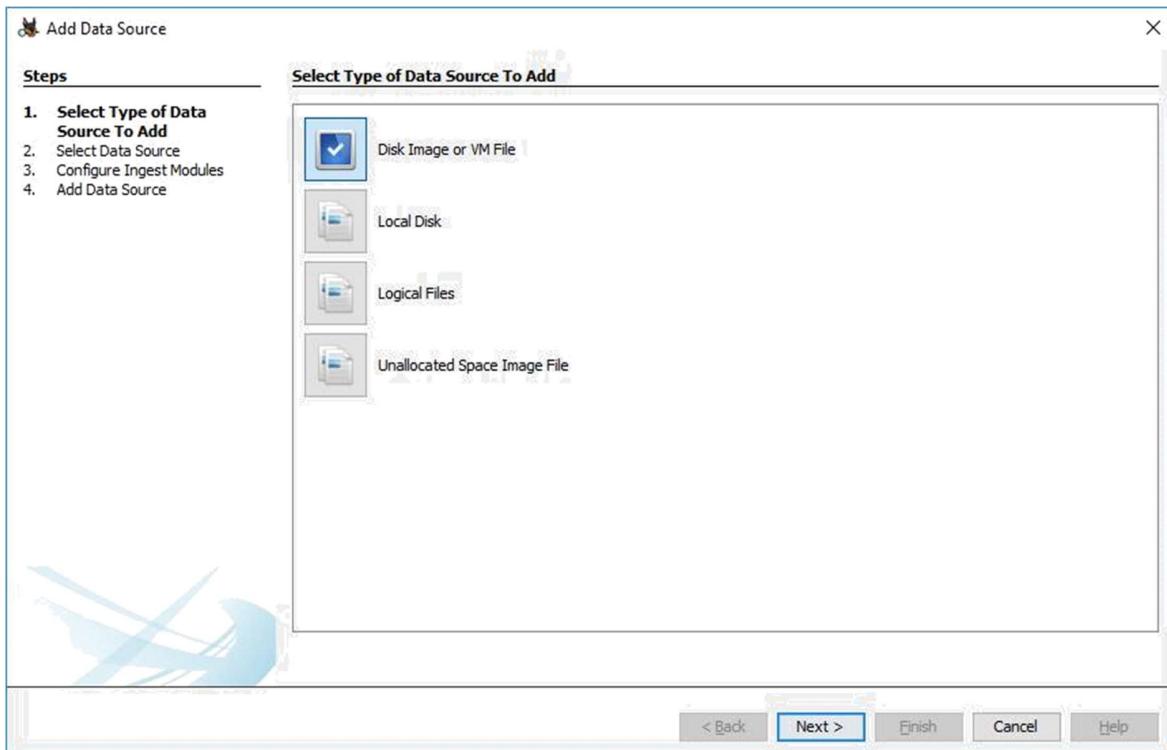
Notes:

Organization

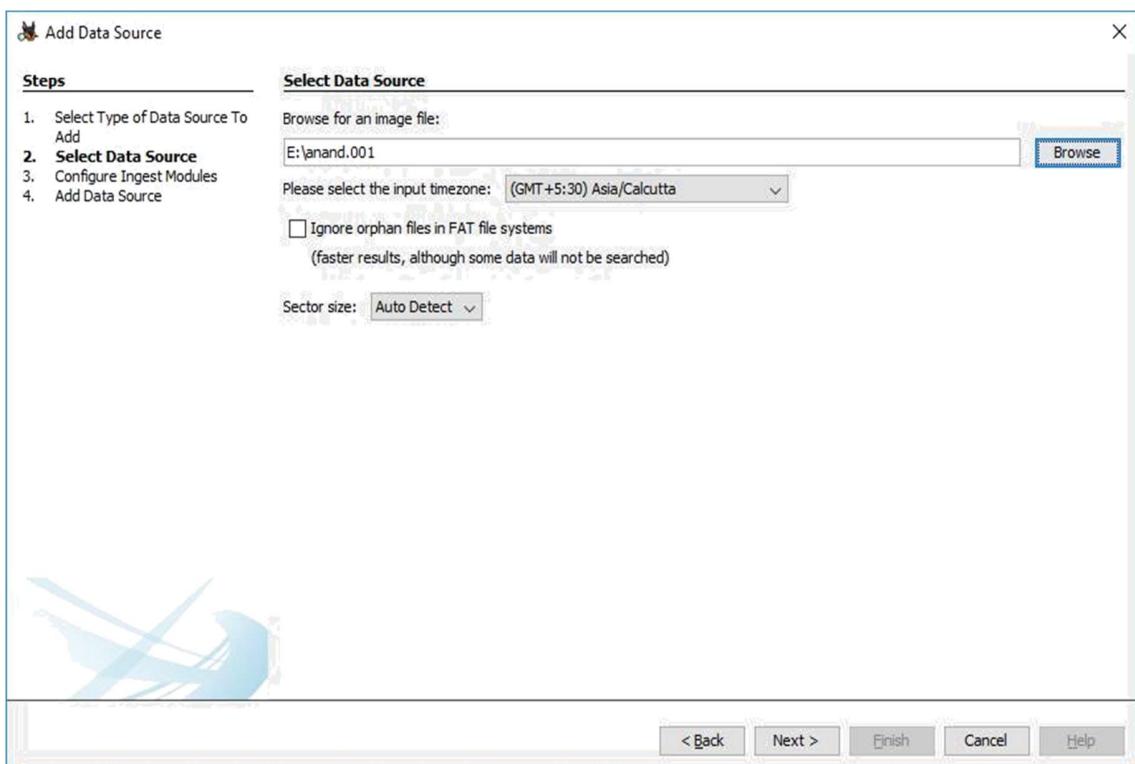
Organization analysis is being done for:

< Back Finish Cancel Help

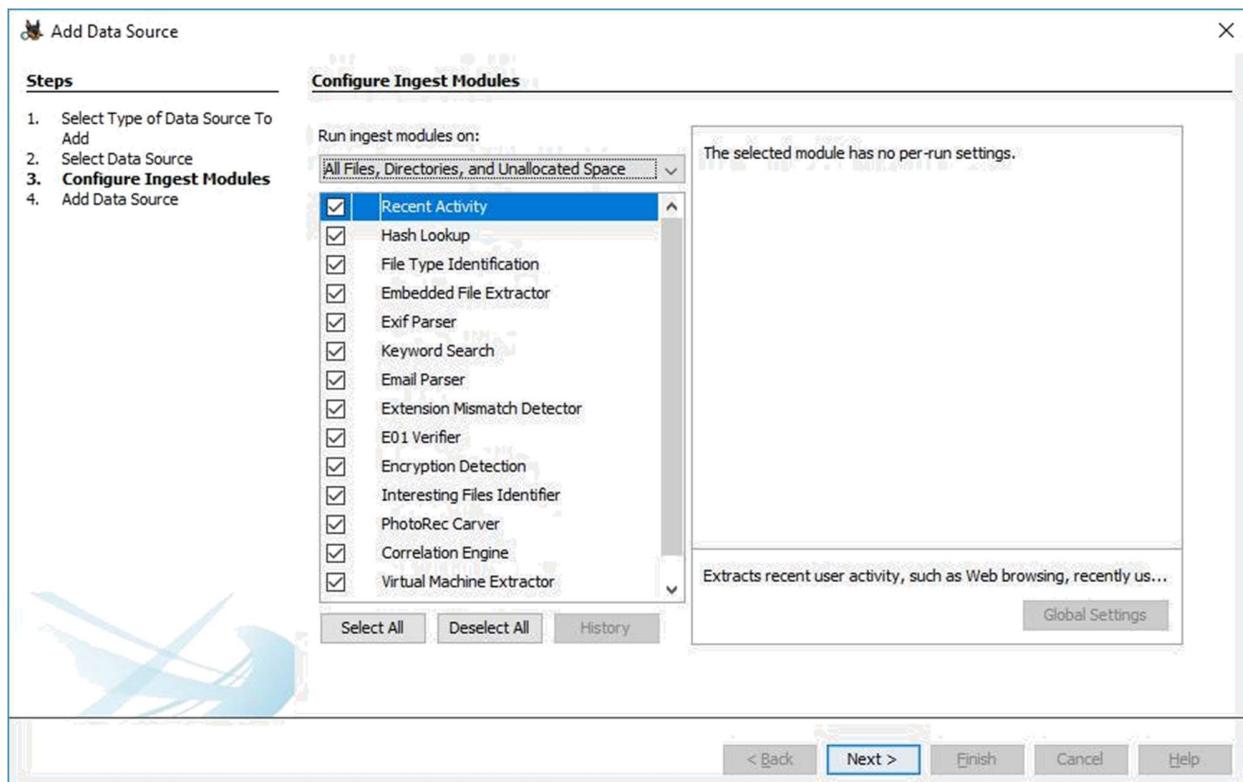
Step 5 : Now here we have to select Type of data source to add , in our case diskimage or VM file and click on next



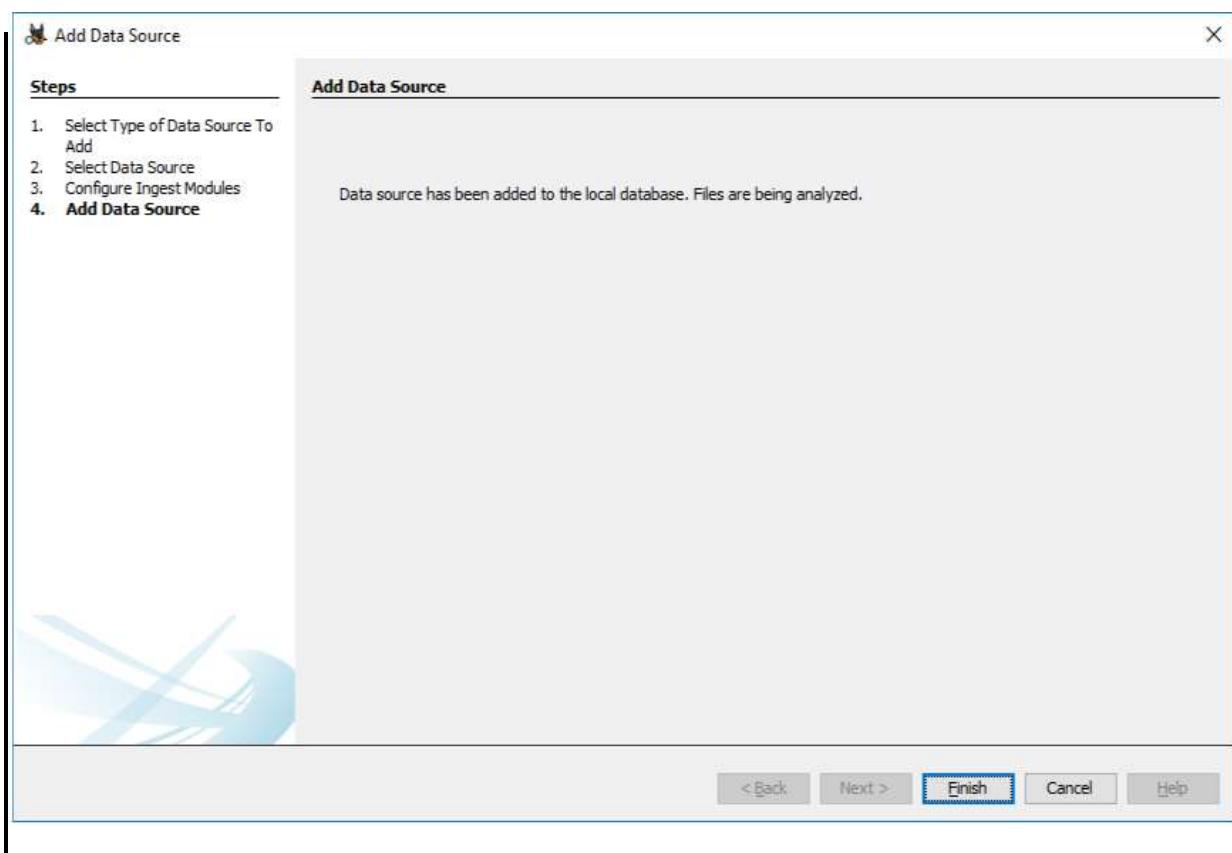
Step 6 : Now we have to select image file and click on next button



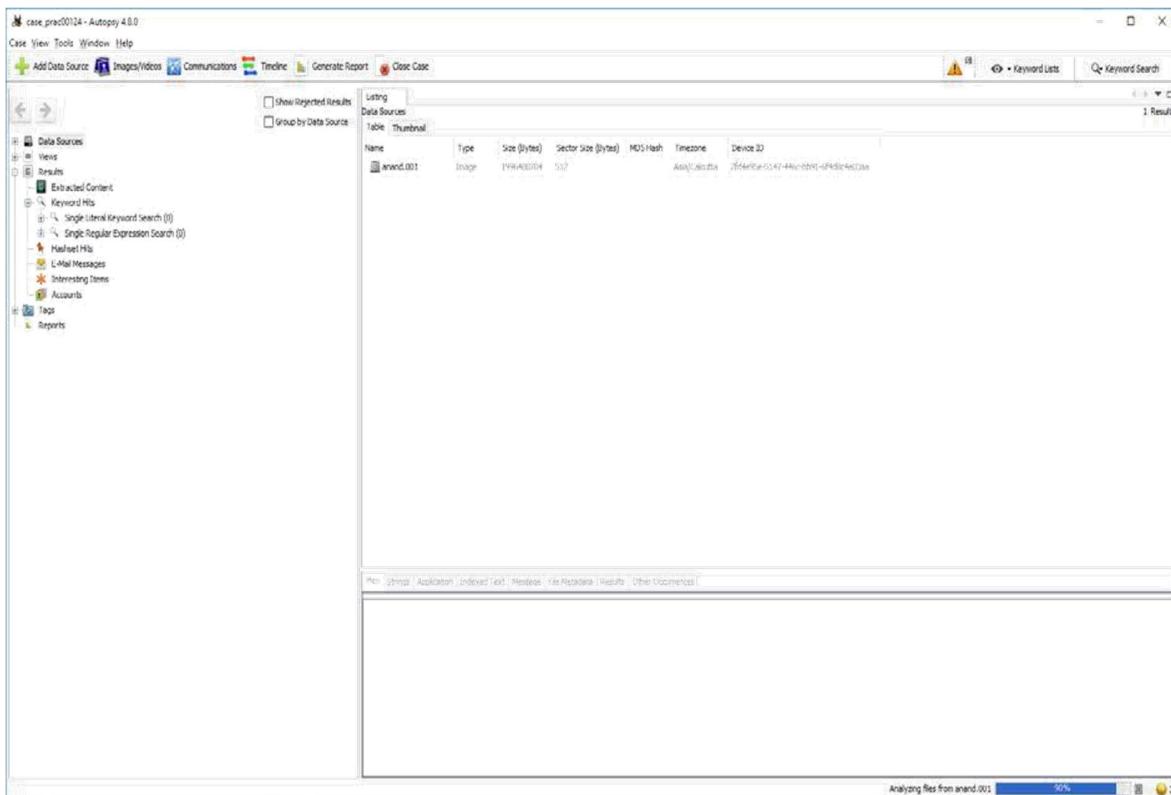
Step 7 : Now click on select all in order to Run ingest modules on: and click onnext.



Step 8 : Now click on finish



Step 9 : Now Autopsy window will appear and it will analyse the disk that we have selected .



Step 10 : All image files appears in the Table tab. Select any file to see the data

case_prac00124 - Autopsy 4.8.0

Case View Tools Window Help

Add Data Source Images/Videos Communications Timeline Generate Report Close Case Keyword Lists Keyword Search

Show Rejected Results Group by Data Source

Listing /Img_anand.001/datasience 8 Results

Table Thumbnail

Name	Modified Time	Change Time	Access
[current folder]	2018-12-11 11:12:52 IST	0000-00-00 00:00:00	2018
[parent folder]	0000-00-00 00:00:00	0000-00-00 00:00:00	0000
Getting Started with RStudio.pdf	2018-12-09 15:24:06 IST	0000-00-00 00:00:00	2019
high level pgmg lang.docx	2018-12-09 18:11:48 IST	0000-00-00 00:00:00	2019
lec1.R	2018-12-09 22:05:18 IST	0000-00-00 00:00:00	2019
R-3.5.1-win.exe	2018-12-09 15:27:54 IST	0000-00-00 00:00:00	2019
RStudio-1.1.463.exe	2018-12-09 15:33:26 IST	0000-00-00 00:00:00	2019
RStudio101.pdf	2018-12-09 14:36:08 IST	0000-00-00 00:00:00	2019

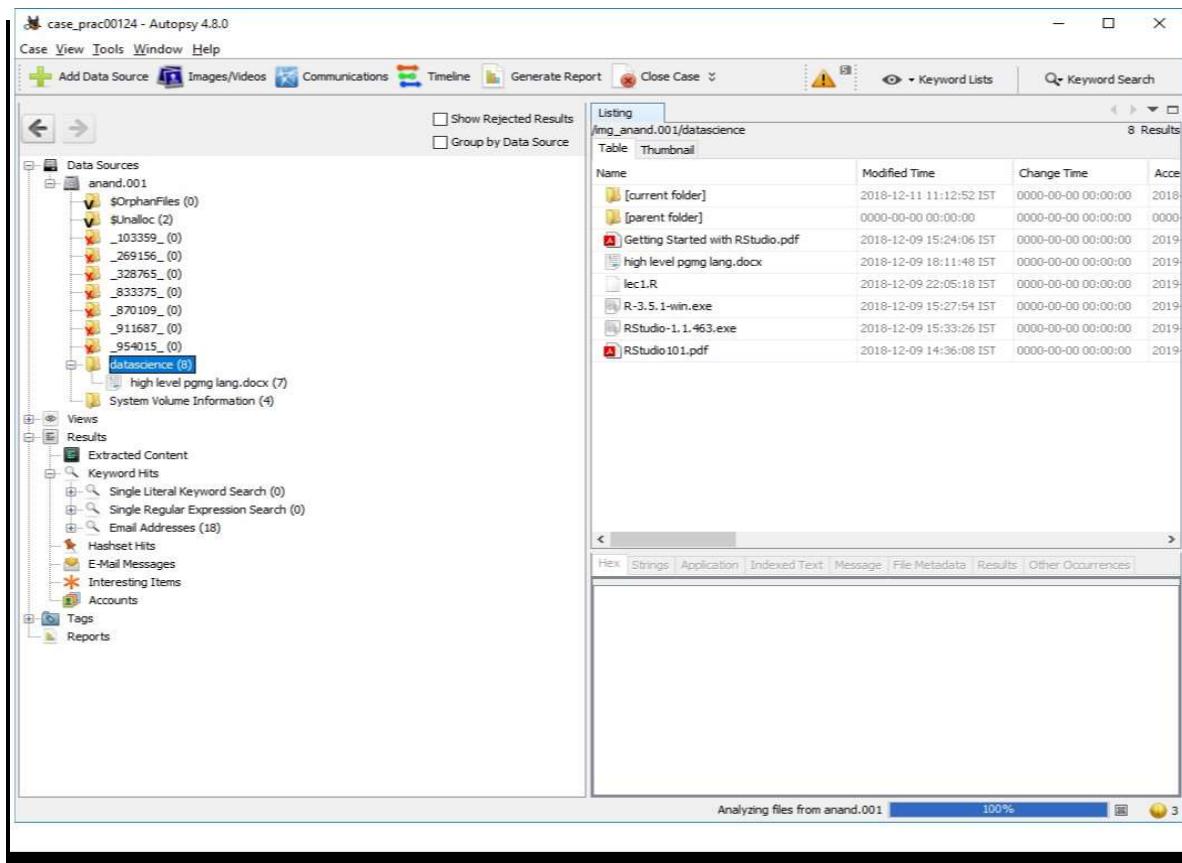
Views Results

- Extracted Content
- Keyword Hits
- Single Literal Keyword Search (0)
- Single Regular Expression Search (0)
- Email Addresses (18)
- Hashset Hits
- E-Mail Messages
- Interesting Items
- Accounts

Tags Reports

Hex Strings Application Indexed Text Message File Metadata Results Other Occurrences

Analyzing files from anand.001 100% 3



Step 11 : Expand the tree from left side panel to view the document files.

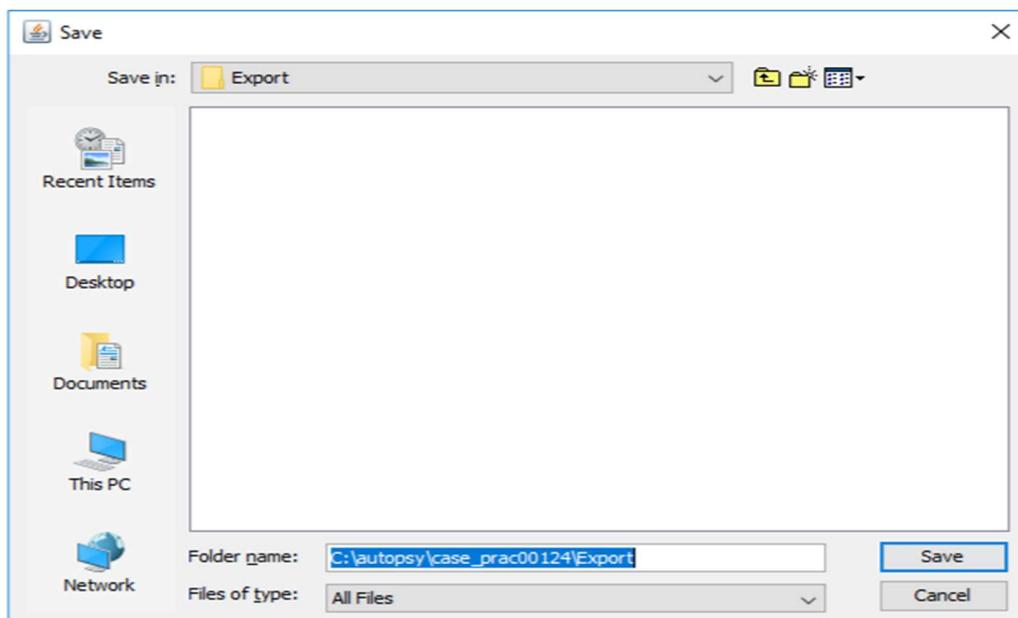
The screenshot shows the Autopsy 4.8.0 interface. The left sidebar displays a hierarchical tree of data sources and analysis results. Under 'Data Sources', the 'anand.001' source is expanded, showing various file types like '\$OrphanFiles', '\$Unalloc', and several deleted files represented by red X icons. Other nodes include 'datascience' and 'System Volume Information'. The 'Views' section contains 'File Types', 'Deleted Files' (which is currently selected), 'File System', and 'All (34)'. The 'MB File Size' and 'Results' sections also contain various items like 'Extracted Content', 'Keyword Hits', and 'Interesting Items'. The main pane shows a table of recovered files under the 'Listing' tab. The table has columns for 'Name' and 'Location'. Several files are listed, including '_269156_', '_328765_', '_954015_', '_833375_', '_870109_ (selected)', '_911687_', '_103359_', 'mongodb-win32-x86_64-2012plus-latest-signed.msi', 'f1597948.exe', 'f1682120.txt', 'f1682256.java', 'f1682304.pcx', 'f1682332.deb', 'f1682380.deb', 'f1682512.deb', and 'f1683808.deb'. The 'Indexed Text' tab is active at the bottom, showing the message 'No indexed text for this file.'

Step 12 : To recover the files , go to view code
any file and right click on it then select Extract files option

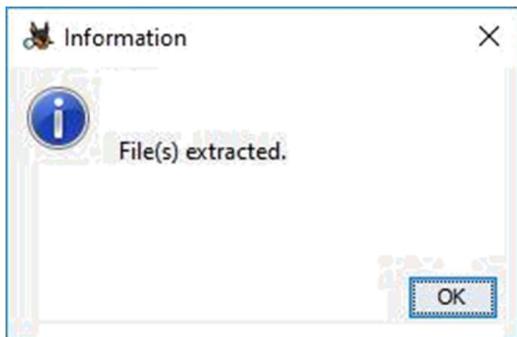
→ Deleted files node , hereselect

The screenshot shows the Autopsy 4.8.0 interface. The left sidebar contains a tree view of the case structure, including Data Sources (anand.001), Views, MB File Size, and Results (Extracted Content, Keyword Hits, Tags, Reports). The main pane displays a table of files under the 'Listing' tab. The table has columns for Name and Location. A context menu is open over the file '_870109_'. The menu includes options such as Properties, View File in Directory, Extract File(s), Add File Tag, Remove File Tag, and Add file to hash set.

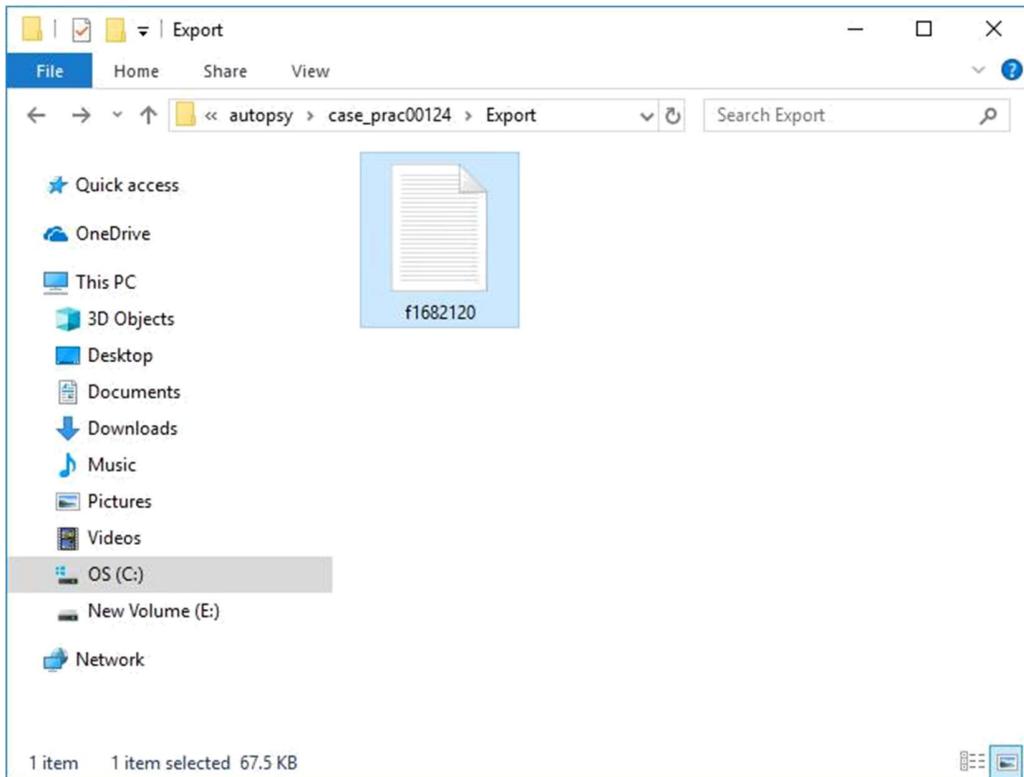
Step 12: Select Path where you want to save extracted file and click on save .



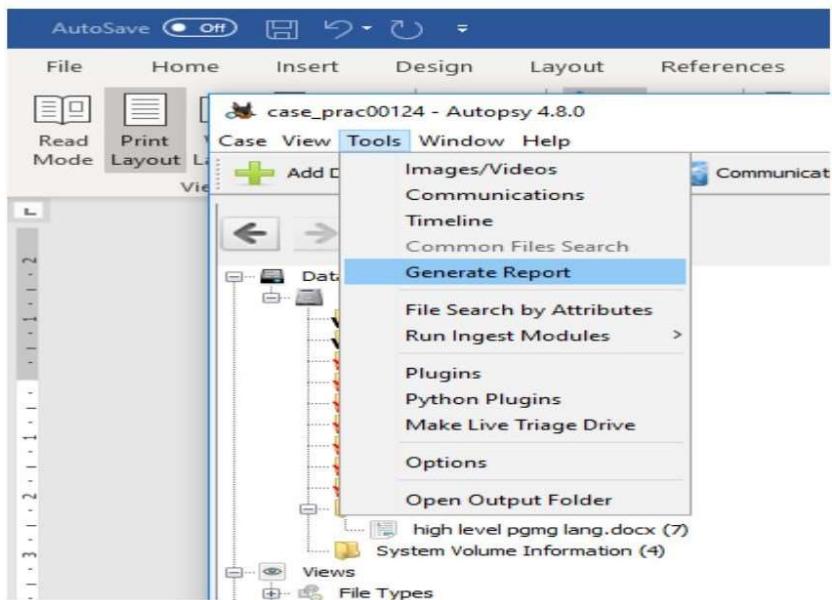
Step 13 : Now click on OK



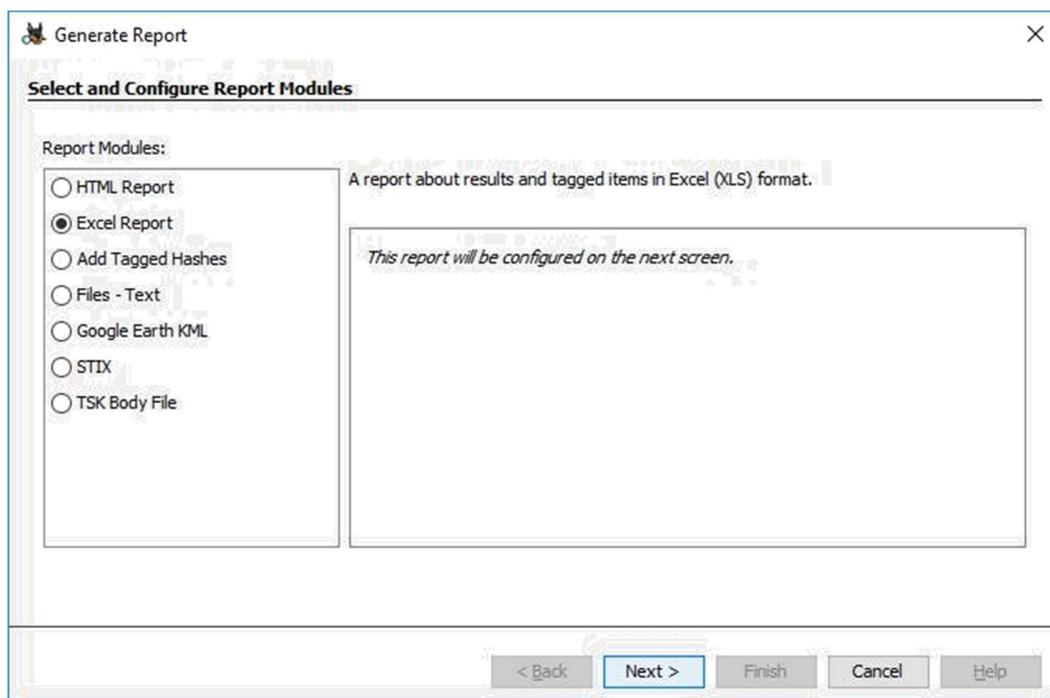
Step 14 : Now go to C:\autopsy\case_prac00124\Export folder to see recoverfile



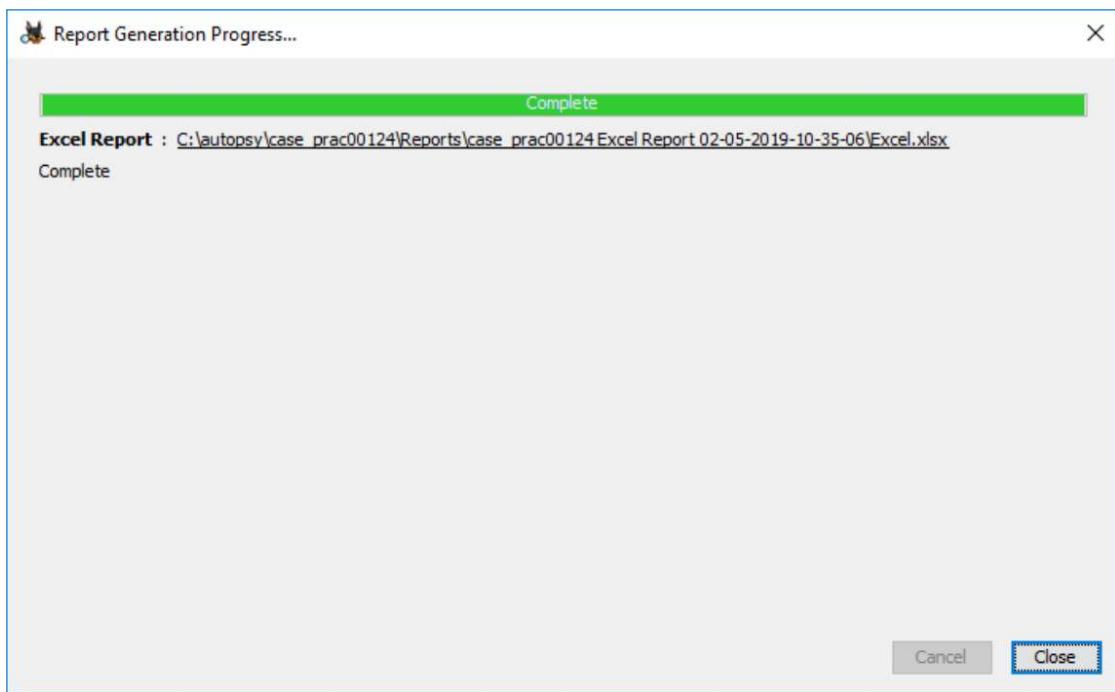
Step 15 : Click on generate report from Autopsy window and select the Excelformat and click on next



Step 16 : This window will appear



Step 17 : Now report is generated so click on close button. We can see the Report on Report Node



Step 18 : Click on report

Case View Tools Window Help

Add Data Source Images/Videos Communications Timeline Generate Report Close Case

Data Sources

- anand.001
 - \$OrphanFiles (0)
 - SUnalloc (2)
 - _103359_(0)
 - _269156_(0)
 - _328765_(0)
 - _833375_(0)
 - _870109_(0)
 - _911687_(0)
 - _954015_(0)
 - datascience (8)
 - high level pgmg lang.docx (7)
 - System Volume Information (4)

Views

- File Types
- Deleted Files
 - File System (8)
 - All (34)

MB File Size

Results

- Extracted Content
- Keyword Hits
 - Single Literal Keyword Search (0)
 - Single Regular Expression Search (0)
 - Email Addresses (18)
 - Hashset Hits
- E-Mail Messages
- Interesting Items
- Accounts

Tags

Reports

Listing

Table Thumbnail

Source Module Name	Report Name	Created Time
Excel Report		2019-02-05 10:35

Hex Strings Application Indexed Text Message File Met

PRACTICAL 4

AIM : Capturing and analyzing network packets using Wireshark(Fundamentals) :

Identification the live network Capture

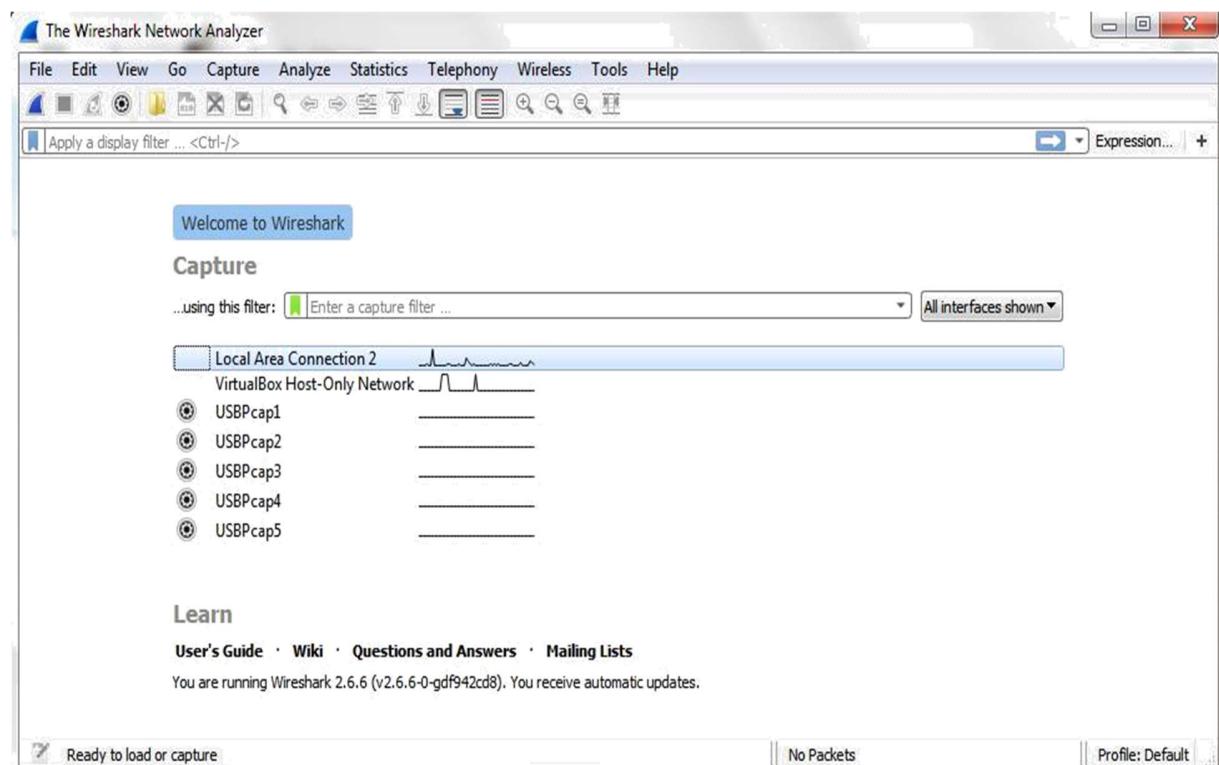
Packets

Analyze the captured packets

Capturing Packets

Capture traffic on your wireless network, click your wireless interface.

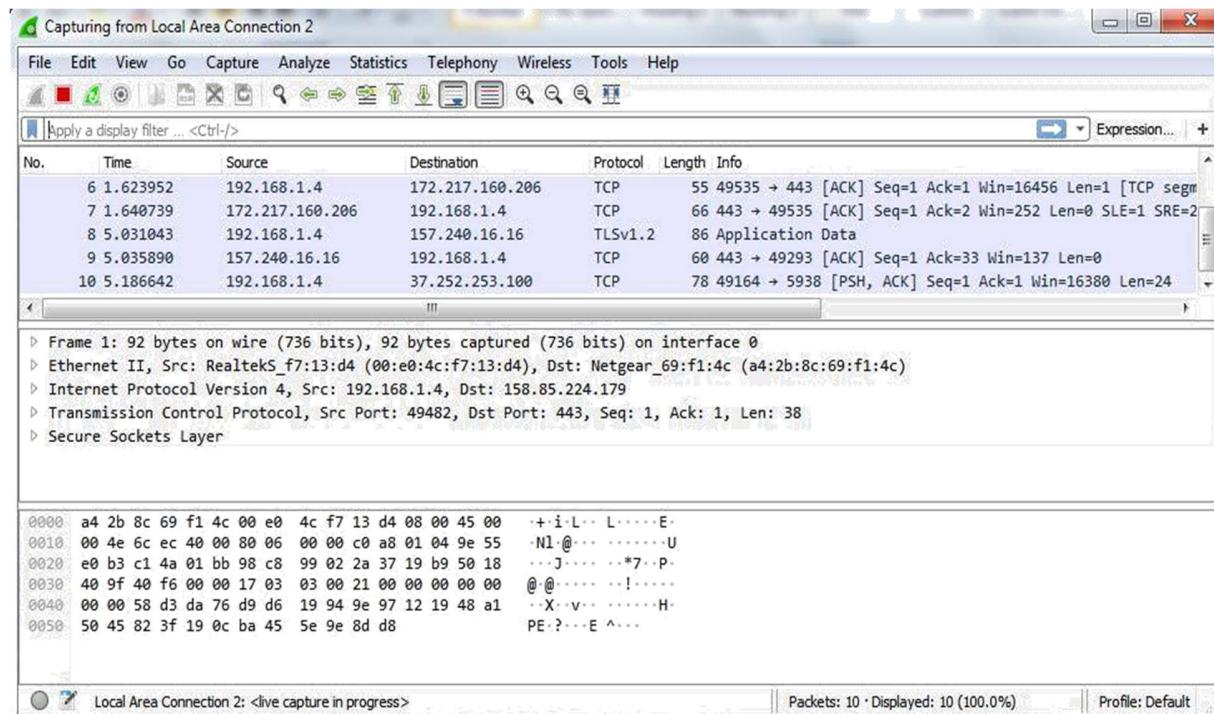
You can configure advanced features by clicking Capture > Options, but this isn't necessary for now.



As soon as you single-click on your network interface's name, you can see how the packets are working in real time. Wireshark will capture all the packets going in and out of our systems.

Promiscuous mode is the mode in which you can see all the packets from other systems on the network and not only the packets send or received from your network adapter. Promiscuous mode is enabled by default. To check if this mode is enabled, go to Capture and Select Options. Under this window check, if the checkbox is selected and activated at the bottom of the window. The checkbox says “Enable promiscuous mode on all interfaces”.

The red box button “STOP” on the top left side of the window can be clicked to stop the capturing of traffic on the network.



Color Coding

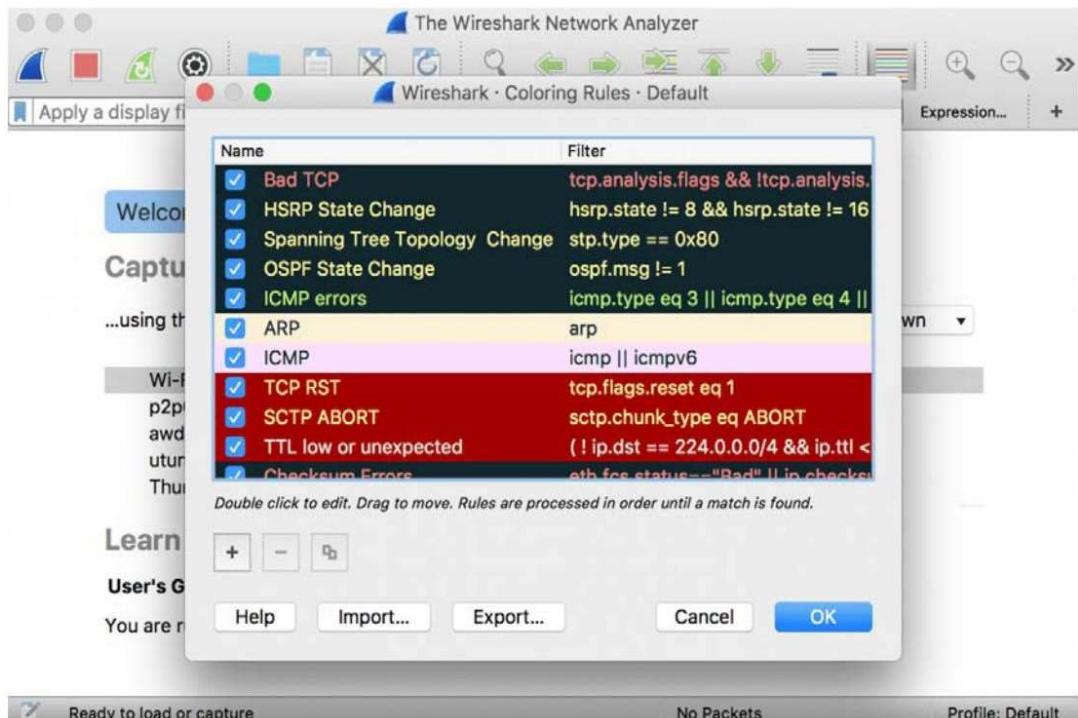
Different packets are seen highlighted in various different colors. This is Wireshark’s way of displaying traffic to help you easily identify the types of it. Default colors are:

Light Purple color for TCP traffic

Blue color for UDP traffic

Black color identifies packets with errors – example these packets are delivered in an unordered manner.

To check the color coding rules click on View and select Coloring Rules. These color coding rules can be customized and modified to fit your needs.



Analyze the captured Packets:

First of all, click on a packet and select it. Now, you can scroll down to view all its details.

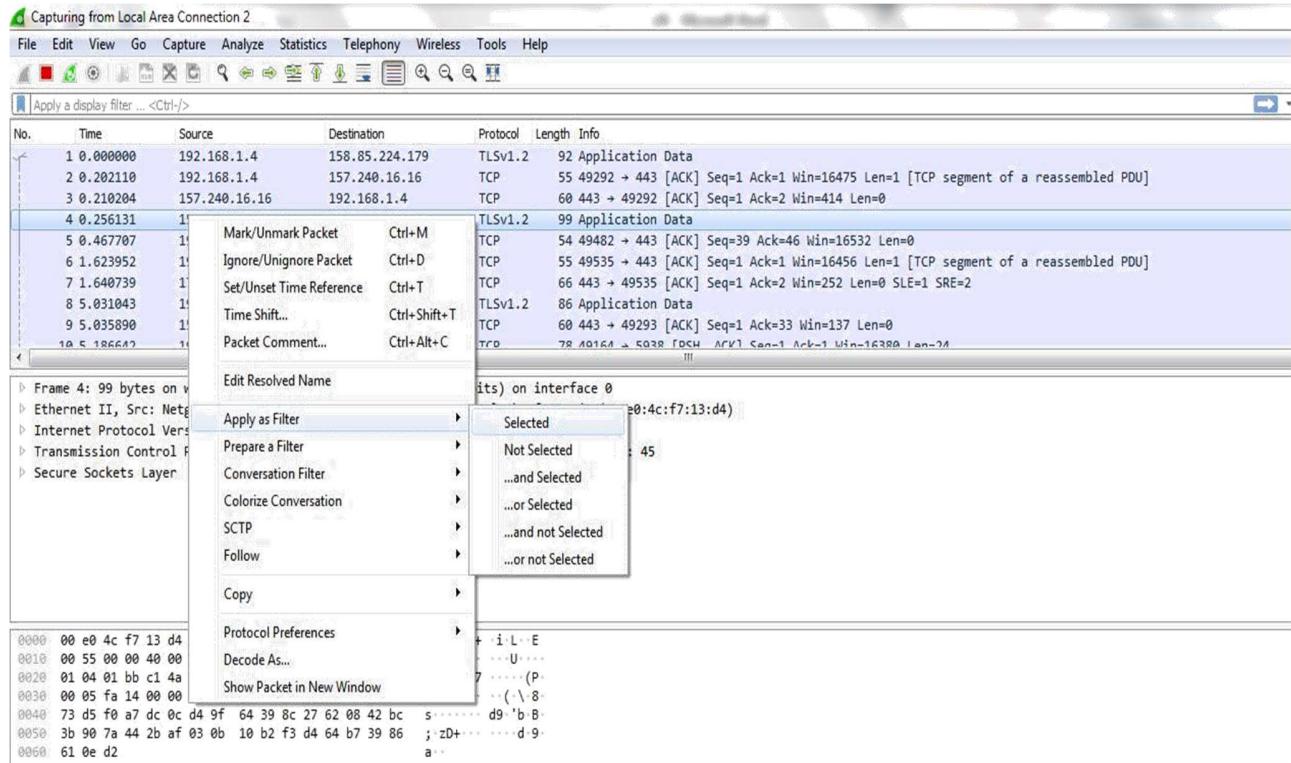
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.4	158.85.224.179	TLSv1.2	92	Application Data
2	0.202110	192.168.1.4	157.240.16.16	TCP	55	49292 → 443 [ACK] Seq=1 Ack=1 Win=16475 Len=1 [TCP segm
3	0.210204	157.240.16.16	192.168.1.4	TCP	60	443 → 49292 [ACK] Seq=1 Ack=2 Win=414 Len=0
4	0.256131	158.85.224.179	192.168.1.4	TLSv1.2	99	Application Data
5	0.467707	192.168.1.4	158.85.224.179	TCP	54	49482 → 443 [ACK] Seq=39 Ack=46 Win=16532 Len=0

Frame 4: 99 bytes on wire (792 bits), 99 bytes captured (792 bits) on interface 0
D Ethernet II, Src: Netgear_69:f1:4c (4e:2b:8c:69:f1:4c), Dst: Realtek5_f7:13:d4 (00:e0:4c:f7:13:d4)
D Internet Protocol Version 4, Src: 158.85.224.179, Dst: 192.168.1.4
D Transmission Control Protocol, Src Port: 443, Dst Port: 49482, Seq: 1, Ack: 39, Len: 45
D Secure Sockets Layer

```

0000  00 e0 4c f7 13 d4 a4 2b  8c 69 f1 4c 08 00 45 20  .L...+ .i.L-E
0010  00 55 00 00 40 00 2e 06  0b ce 9e 55 e0 b3 c0 a8  .U@. ....U...
0020  01 04 01 b2 c1 4a 2a 37  19 b9 98 c8 99 28 50 18  ....J#7 ....(P
0030  00 05 fa 14 00 00 17 03  03 00 28 1b 5c 89 38 8f  ....(.'8.
0040  73 d5 f0 a7 dc 0c d4 9f  64 39 8c 27 62 08 42 bc  s..... d9.'b.8.
0050  3b 90 7a 44 2b af 03 0b  10 b2 f3 d4 64 b7 39 86  ;.zD+.... ....d.9.
0060  61 0e d2  a..
```

Filters can also be created from here. Right-click on one of any details. From the menu select Apply as Filter drop-down menu so filter based on it can be created.



Display filter command –

1. Display packets based on specific IP-address

`ip.addr == 192.0.2.1`

No.	Time	Source	Destination	Protocol	Length	Info
49176	632.590744	192.168.1.4	216.58.219.227	TCP	55	[TCP Keep-Alive] 49231 → 443 [ACK] Seq=4349 Ack=5923 Win=65408 Len=1
49177	632.915897	216.58.219.227	192.168.1.4	TCP	66	[TCP Keep-Alive ACK] 443 → 49231 [ACK] Seq=5923 Ack=4350 Win=69632 Len=0 SLE=4349 SRE=4350
49178	633.207727	0.0.0.0	224.0.0.1	IGMPv2	60	Membership Query, general
49179	633.415028	192.168.1.4	239.255.255.250	IGMPv2	46	Membership Report group 239.255.255.250
49180	633.876818	192.168.1.4	172.217.167.163	TCP	55	[TCP Keep-Alive] 49185 → 443 [ACK] Seq=19248 Ack=947960 Win=84176 Len=1
49181	633.901488	172.217.167.163	192.168.1.4	TCP	66	[TCP Keep-Alive ACK] 443 → 49185 [ACK] Seq=947960 Ack=19249 Win=75776 Len=0 SLE=19248 SRE=19249
49182	634.414944	192.168.1.4	224.0.0.252	IGMPv2	46	Membership Report group 224.0.0.252
49183	640.313942	192.168.1.3	239.255.255.250	SSDP	167	M-SEARCH * HTTP/1.1
49184	640.604029	192.168.1.3	239.255.255.250	SSDP	167	M-SEARCH * HTTP/1.1
49185	640.904021	192.168.1.3	239.255.255.250	SSDP	167	M-SEARCH * HTTP/1.1

2. Display packets which are coming from specific IP-address

ip.src == 192.168.1.3

ip.src == 192.168.1.3						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.3	239.255.255.250	SSDP	167	M-SEARCH * HTTP/1.1
2	0.293839	192.168.1.3	239.255.255.250	SSDP	167	M-SEARCH * HTTP/1.1
3	0.591360	192.168.1.3	239.255.255.250	SSDP	167	M-SEARCH * HTTP/1.1
12	10.037574	192.168.1.3	239.255.255.250	SSDP	167	M-SEARCH * HTTP/1.1
13	10.333930	192.168.1.3	239.255.255.250	SSDP	167	M-SEARCH * HTTP/1.1
14	10.633876	192.168.1.3	239.255.255.250	SSDP	167	M-SEARCH * HTTP/1.1
16	12.458395	192.168.1.3	224.0.0.251	MDNS	103	Standard query 0x0059 PTR _233637DE._sub._googlecast._tcp.local, "QM" question PTR _googlecast._tcp.lo
19	20.010644	192.168.1.3	239.255.255.250	SSDP	167	M-SEARCH * HTTP/1.1
20	20.301273	192.168.1.3	239.255.255.250	SSDP	167	M-SEARCH * HTTP/1.1
21	20.602551	192.168.1.3	239.255.255.250	SSDP	167	M-SEARCH * HTTP/1.1
22	20.618275	10.168.1.2	239.255.255.250	SSDP	167	M-SEARCH * HTTP/1.1

3. Display packets which are having specific IP-address destination

ip.dst == 192.168.1.1

ip.dst==192.168.1.1						
No.	Time	Source	Destination	Protocol	Length	Info
4	4.037895	192.168.1.4	192.168.1.1	DNS	85	Standard query 0xc7f4 A teredo.ipv6.microsoft.com
6	5.032826	192.168.1.4	192.168.1.1	DNS	85	Standard query 0xc7f4 A teredo.ipv6.microsoft.com
7	6.032784	192.168.1.4	192.168.1.1	DNS	85	Standard query 0xc7f4 A teredo.ipv6.microsoft.com
11	8.032694	192.168.1.4	192.168.1.1	DNS	85	Standard query 0xc7f4 A teredo.ipv6.microsoft.com
15	12.033085	192.168.1.4	192.168.1.1	DNS	85	Standard query 0xc7f4 A teredo.ipv6.microsoft.com
55	74.984400	192.168.1.4	192.168.1.1	TCP	66	49173 + 56688 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
57	74.984875	192.168.1.4	192.168.1.1	TCP	54	49173 + 56688 [ACK] Seq=1 Ack=1 Win=65700 Len=0
58	74.985092	192.168.1.4	192.168.1.1	HTTP	250	GET /rootDesc.xml HTTP/1.1
64	74.987818	192.168.1.4	192.168.1.1	TCP	54	49173 + 56688 [ACK] Seq=197 Ack=4102 Win=65700 Len=0
65	74.989866	192.168.1.4	192.168.1.1	TCP	54	49173 + 56688 [FIN, ACK] Seq=197 Ack=4102 Win=65700 Len=0
90	05.721021	10.168.1.4	10.168.1.1	DNS	85	Standard query 0xc7f4 A teredo.ipv6.microsoft.com

4. Display packets which are using http protocol http

No.	Time	Source	Destination	Protocol	Length	Info
58	74.985092	192.168.1.4	192.168.1.1	HTTP	250	GET /rootDesc.xml HTTP/1.1
62	74.987756	192.168.1.1	192.168.1.4	HTTP/X...	1234	HTTP/1.1 200 OK
972	129.457310	192.168.1.4	172.217.166.174	HTTP	1000	GET / HTTP/1.1
975	129.542230	172.217.166.174	192.168.1.4	HTTP	594	HTTP/1.1 301 Moved Permanently (text/html)
39156	277.292187	192.168.1.4	117.18.237.29	OCSP	137	Request
39157	277.314544	117.18.237.29	192.168.1.4	OCSP	842	Response
39168	277.419340	192.168.1.4	117.18.237.29	OCSP	137	Request
39169	277.463638	117.18.237.29	192.168.1.4	OCSP	842	Response
39204	279.409683	192.168.1.4	23.57.219.27	OCSP	137	Request
39206	279.420870	23.57.219.27	192.168.1.4	OCSP	712	Response
20019	270.492459	10.168.1.4	12.57.210.27	OCSP	127	Request

5. Display packets which are using http request http.request

http.request						
No.	Time	Source	Destination	Protocol	Length	Info
40	50.307358	192.168.1.3	239.255.255.250	SSDP	167	M-SEARCH * HTTP/1.1
41	50.607228	192.168.1.3	239.255.255.250	SSDP	167	M-SEARCH * HTTP/1.1
46	60.815835	192.168.1.3	239.255.255.250	SSDP	167	M-SEARCH * HTTP/1.1
47	60.306194	192.168.1.3	239.255.255.250	SSDP	167	M-SEARCH * HTTP/1.1
48	60.605851	192.168.1.3	239.255.255.250	SSDP	167	M-SEARCH * HTTP/1.1
49	70.031605	192.168.1.3	239.255.255.250	SSDP	167	M-SEARCH * HTTP/1.1
50	70.321279	192.168.1.3	239.255.255.250	SSDP	167	M-SEARCH * HTTP/1.1
51	70.626289	192.168.1.3	239.255.255.250	SSDP	167	M-SEARCH * HTTP/1.1
53	73.874454	192.168.1.4	239.255.255.250	SSDP	175	M-SEARCH * HTTP/1.1
58	74.985092	192.168.1.4	192.168.1.1	HTTP	250	GET /rootDesc.xml HTTP/1.1
59	74.985092	192.168.1.4	239.255.255.250	SSDP	175	M-SEARCH * HTTP/1.1

6. Display packets which are using TCP protocol tcp

tcp						
No.	Time	Source	Destination	Protocol	Length	Info
31	41.077583	192.168.1.4	188.65.76.135	TCP	54	49163 + 5938 [ACK] Seq=25 Ack=25 Win=16592 Len=0
32	41.184892	188.65.76.135	192.168.1.4	TCP	78	[TCP Spurious Retransmission] 5938 + 49163 [PSH, ACK] Seq=1 Ack=25 Win=1022 Len=24
33	41.184946	192.168.1.4	188.65.76.135	TCP	66	[TCP Dup ACK 31#1] 49163 + 5938 [ACK] Seq=25 Ack=25 Win=16592 Len=0 SLE=1 SRE=25
37	45.4958801	192.168.1.4	188.65.76.135	TCP	78	49163 + 5938 [PSH, ACK] Seq=25 Ack=25 Win=16592 Len=24
38	46.087275	188.65.76.135	192.168.1.4	TCP	60	5938 + 49163 [ACK] Seq=25 Ack=49 Win=1022 Len=0
45	54.788090	192.168.1.4	184.25.218.21	TCP	54	49171 + 443 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
55	74.984400	192.168.1.4	192.168.1.1	TCP	66	49173 + 56688 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
56	74.984790	192.168.1.1	192.168.1.4	TCP	66	56688 + 49173 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM=1 WS=2
57	74.984875	192.168.1.4	192.168.1.1	TCP	54	49173 + 56688 [ACK] Seq=1 Ack=1 Win=65700 Len=0
58	74.985092	192.168.1.4	192.168.1.1	HTTP	250	GET /rootDesc.xml HTTP/1.1
59	74.985276	192.168.1.4	192.168.1.1	TCP	60	56688 + 49173 [ACK] Seq=1 Ack=1 Win=65700 Len=0

7. Display packets having no error connecting to server

http.response.code==200

http.response.code==200						
No.	Time	Source	Destination	Protocol	Length	Info
40241	315.834863	27.106.94.17	192.168.1.4	TCP	455	HTTP/1.1 200 OK [TCP segment of a reassembled PDU]
40251	315.941483	192.168.1.1	192.168.1.4	HTTP/X...	315	HTTP/1.1 200 OK
40261	315.967166	192.168.1.1	192.168.1.4	HTTP	250	HTTP/1.1 200 OK
40270	315.968680	192.168.1.4	192.168.1.1	HTTP	191	HTTP/1.1 200 OK
40282	315.977822	192.168.1.1	192.168.1.4	HTTP/X...	539	HTTP/1.1 200 OK
40294	315.982033	192.168.1.1	192.168.1.4	HTTP/X...	557	HTTP/1.1 200 OK
40308	315.999143	192.168.1.1	192.168.1.4	HTTP/X...	315	HTTP/1.1 200 OK
40318	316.005125	192.168.1.1	192.168.1.4	HTTP	250	HTTP/1.1 200 OK
40327	316.007892	192.168.1.4	192.168.1.1	HTTP	191	HTTP/1.1 200 OK
40339	316.015485	192.168.1.1	192.168.1.4	HTTP/X...	539	HTTP/1.1 200 OK
40351	316.019180	192.168.1.1	192.168.1.4	HTTP/X...	557	HTTP/1.1 200 OK

8. Display packets having port number 80

tcp.port==80 || udp.port==80

tcp.port==80 udp.port==80						
No.	Time	Source	Destination	Protocol	Length	Info
40216	315.186100	192.168.1.4	172.217.160.206	TCP	54	49295 + 80 [ACK] Seq=1 Ack=1 Win=66240 Len=0
40217	315.186313	192.168.1.4	172.217.160.206	HTTP	293	HEAD /edged1/release2/chrome_component/HPO7sha1VDw_4916/4916_all_crl-set-13576662708261436161.data.crx3 HTTP/1.1
40218	315.208973	172.217.160.206	192.168.1.4	TCP	60	80 + 49295 [ACK] Seq=1 Ack=240 Win=61952 Len=0
40225	315.497872	172.217.160.206	192.168.1.4	HTTP	608	HTTP/1.1 302 Found
40228	315.512340	192.168.1.4	27.106.94.17	TCP	66	49296 + 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
40231	315.693760	192.168.1.4	172.217.160.206	TCP	54	49295 + 80 [ACK] Seq=240 Ack=555 Win=65684 Len=0
40237	315.823271	27.106.94.17	192.168.1.4	TCP	66	80 + 49296 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1452 SACK_PERM=1 WS=256
40238	315.823365	192.168.1.4	27.106.94.17	TCP	54	49296 + 80 [ACK] Seq=1 Ack=1 Win=66792 Len=0
40239	315.823558	192.168.1.4	27.106.94.17	HTTP	404	HEAD /edged1/release2/chrome_component/HPO7sha1VDw_4916/4916_all_crl-set-13576662708261436161.data.crx3?cms_red...
40241	315.834863	27.106.94.17	192.168.1.4	HTTP	455	HTTP/1.1 200 OK
40244	315.906000	192.168.1.4	27.106.94.17	TCP	54	49296 + 80 [ACK] Seq=251 Ack=602 Win=66288 Len=0

9.Display packets which contains keyword facebook tcp contains facebook

tcp contains facebook						
No.	Time	Source	Destination	Protocol	Length	Info
7711	32.085504	192.168.1.4	31.13.79.35	TLSv1.3	571	Client Hello
8160	32.867205	192.168.1.4	31.13.79.35	TLSv1.3	571	Client Hello
9739	35.561576	192.168.1.4	157.240.16.35	TLSv1.3	571	Client Hello
29814	162.425666	192.168.1.4	157.240.16.35	TLSv1.3	571	Client Hello
37226	273.164934	192.168.1.4	157.240.16.16	TLSv1.2	571	Client Hello
37388	274.375759	192.168.1.4	157.240.16.16	TLSv1.3	571	Client Hello
43811	381.014078	192.168.1.4	157.240.16.35	TLSv1.3	571	Client Hello
47765	569.305448	192.168.1.4	157.240.16.35	TLSv1.3	571	Client Hello

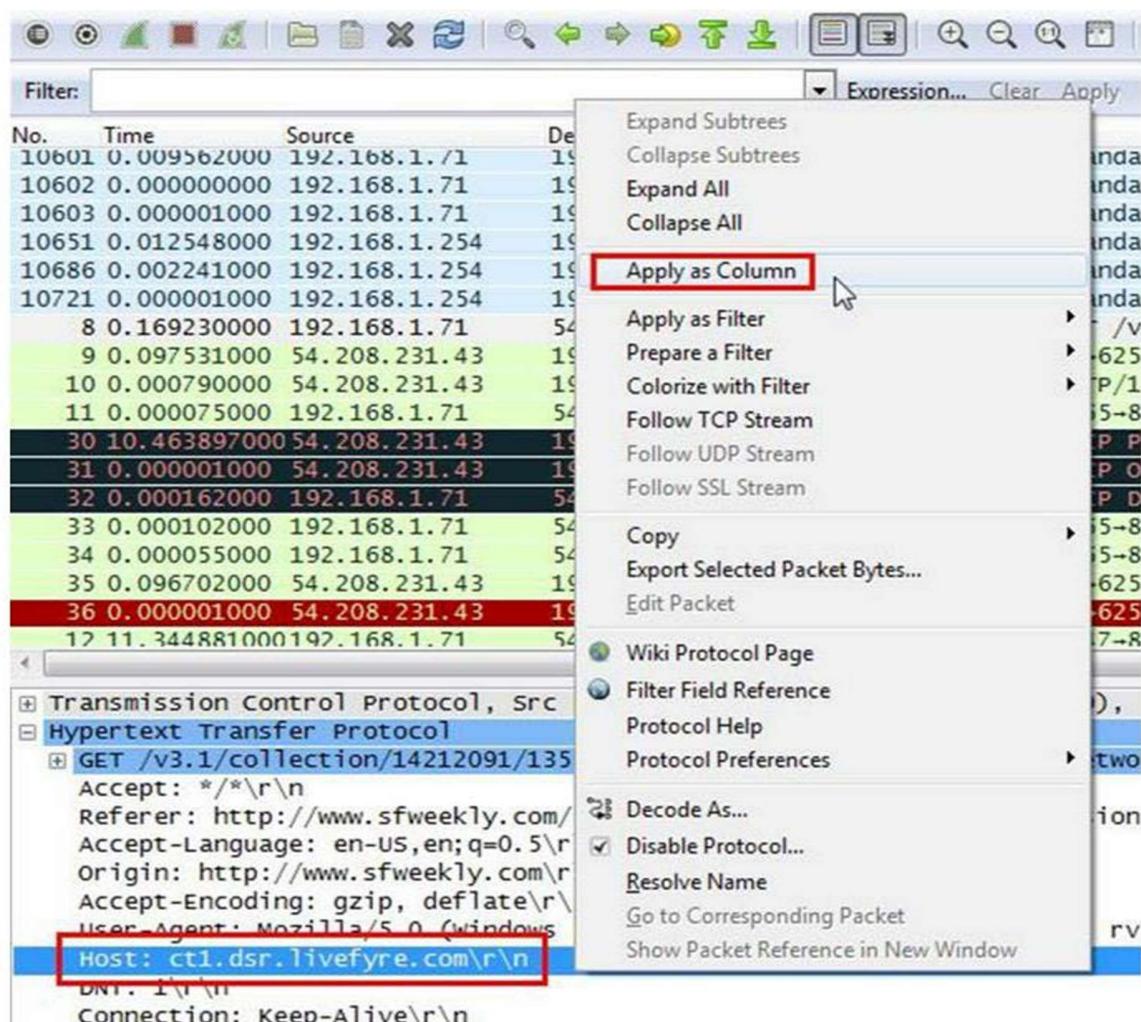
PRACTICAL 5

Aim :- Analyze the packets provided in lab and solve the questions using Wireshark :

What web server software is used by www.snopes.com? About what cell phone problem is the client concerned? According to Zillow, what instrument will Ryan learn to play? How many web servers are running Apache?

What web server software issued by www.snopes.com?

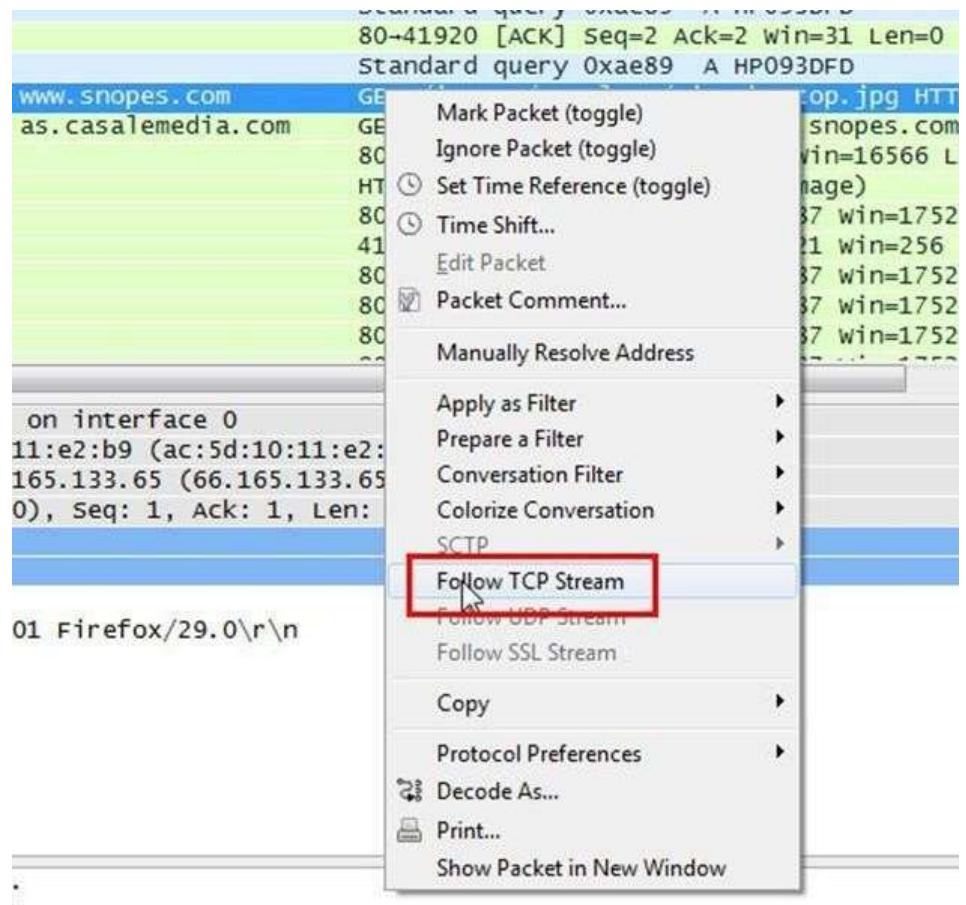
Analysis – The domain name be found from host header so we will set host header column where we will see all domain name. Select any HTTP request and expand the Hypertext Transfer Protocol then right click on Host header and then Apply as Column.



Now we can see our host www.snopes.com in host column.

Time	Source	Destination	Protocol	Length	Host
11 0.055571000	192.168.1.254	192.168.1.71	DNS	222	
12 0.073696000	64.49.225.166	192.168.1.71	TCP	60	
13 0.000150000	192.168.1.71	64.49.225.166	TCP	54	
14 0.000056000	192.168.1.71	64.49.225.166	TCP	54	
15 0.036217000	fe80::856e:7b6d:6ff02::1:3		LLMNR	88	
16 0.001465000	192.168.1.68	224.0.0.252	LLMNR	68	
17 0.041273000	64.49.225.166	192.168.1.71	TCP	60	
18 0.057682000	192.168.1.68	224.0.0.252	LLMNR	68	
19 0.244659000	192.168.1.71	66.165.133.65	HTTP	440	www.snopes.com
20 0.018898000	192.168.1.71	207.109.230.161	HTTP	1037	as.casalemedia.com
21 0.025753000	207.109.230.161	192.168.1.71	TCP	60	
22 0.053733000	66.165.133.65	192.168.1.71	HTTP	1514	
23 0.000839000	66.165.133.65	192.168.1.71	TCP	1514	
24 0.000057000	192.168.1.71	66.165.133.65	TCP	54	
25 0.000751000	66.165.133.65	192.168.1.71	TCP	1514	
26 0.000775000	66.165.133.65	192.168.1.71	TCP	1514	
27 0.000002000	66.165.133.65	192.168.1.71	TCP	1514	

Right click on the selected packet and then select Follow TCP stream.



Now we can see the webserver name in server header it is Microsoft IIS 5.0

```
Stream Content
GET /images/template/site-bg-top.jpg HTTP/1.1
Host: www.snopes.com
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; rv:29.0) Gecko/2
Accept: image/png,image/*;q=0.8,*/*;q=0.5
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.snopes.com/style.css
Cookie: ASPSESSIONIDQQQDSBBA=OJMBNHECFCAKIJJGBBMBLDO
Connection: keep-alive

HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Thu, 22 May 2014 01:49:06 GMT
Content-Type: image/jpeg
Accept-Ranges: bytes
Last-Modified: Mon, 03 Nov 2008 04:34:19 GMT
ETag: "98242b706d3dc91:b5f"
Content-Length: 32173

.....JFIF.....d.d.....Ducky.....U.....Adobe.
d.....
```

2. About what cell phone problem is the client concerned?

Analysis – Client talking about cell so we search for cell keyword in whole packets. We will use regular express for searching the cell keyword. Apply framematches “(?! cell”

Filter: frame matches "(?i)cell"

Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Host
20	0.018898000	192.168.1.71	207.109.230.161	HTTP	1037	as.casalemedia.com
70	0.000001000	207.109.230.161	192.168.1.71	TCP	408	
94	0.039888000	192.168.1.71	74.125.196.139	HTTP	1192	www.google-analytics.com
102	0.017700000	192.168.1.71	50.19.115.152	HTTP	418	stat.komoona.com
106	0.019119000	192.168.1.71	107.20.177.71	HTTP	462	a.komoona.com
126	0.330874000	192.168.1.71	50.19.115.152	HTTP	540	stat.komoona.com
128	0.050275000	192.168.1.71	64.12.239.201	HTTP	510	adserver.adtechus.com
152	0.109725000	192.168.1.71	176.32.99.164	HTTP	436	s.komoona.com
156	0.039271000	192.168.1.71	54.85.82.173	HTTP	439	x.bidswitch.net
157	0.020117000	192.168.1.71	74.209.219.38	HTTP	500	aol-match.dotomi.com
176	0.429894000	192.168.1.71	23.210.219.85	HTTP	989	ads.rubiconproject.com
194	0.014825000	192.168.1.71	54.84.236.238	HTTP	508	pool.adizio.com
200	0.188424000	192.168.1.71	69.25.24.23	HTTP	1091	optimized-by.rubicon
229	0.337378000	192.168.1.71	23.210.231.153	HTTP	1514	ads.pubmatic.com
259	0.000134000	192.168.1.71	54.241.183.234	HTTP	528	x.skimresources.com
268	0.590522000	192.168.1.71	162.248.19.142	HTTP	1514	showads.pubmatic.com
269	0.000010000	192.168.1.71	162.248.19.142	TCP	1514	
610	0.000165000	107.169.1.71	66.165.122.65	HTTP	807	www.ebay.com

After applying the filter now, we will start to check every HTTP request. We noticed in the first HTTP request cell keyword is in URL and it was about cellphone charging issue.

Filter: frame matches "(?i)cell"						
Time	Source	Destination	Protocol	Length	Info	
20 0.018898000	192.168.1.71	207.109.230.161	HTTP	1037	GET /?s=81847&u=http%3A//www.snopes.com/horrors/techno/cellcharge.asp&f=1&id=4240355892,9460	
70 0.000001000	207.109.230.161	192.168.1.71	TCP	408	80-41934 [PSH, ACK] Seq=7316 Ack=984 Win=16566 Len=354	
94 0.039888000	192.168.1.71	74.125.196.139	HTTP	1192	GET /__utm.gif?utmwv=5.5.1&utmsr=624349962&utmhn=www.snopes.com&utmcn=windows-1252&utmsc=windows-1252&utmp=192.168.1.71	
102 0.017700000	192.168.1.71	50.19.115.152	HTTP	418	GET /?s?tagid=cad674db7f73589c9a110884ce73bb7_728_90v=2,16&c=b=516430883&s=2 HTTP/1.1	
106 0.019119000	192.168.1.71	107.20.177.71	HTTP	462	GET /tag/cad674db7f73589c9a110884ce73bb7_728_90,js?l=http%3A%2Fwww.snopes.com%2Fhorrors%2	
126 0.330874000	192.168.1.71	50.19.115.152	HTTP	340	GET /?s?tagid=cad674db7f73589c9a110884ce73bb72&v=2,16&c=b=516430883&ts=-1&p=cad674db7f73589c9a1	
128 0.050275000	192.168.1.71	64.12.239.201	HTTP	510	GET /addyn/3.0/9423.1/3142865/0/225/ADTECH;loc=100;target=_blank;misc=&SBTIMESTAMP%5B;rdclick	
152 0.109725000	192.168.1.71	176.32.99.164	HTTP	436	GET /passback/npcad674db7f73589c9a110884ce73bb72,js HTTP/1.1	
156 0.039271000	192.168.1.71	54.85.82.173	HTTP	439	GET /sync?ssp=aol HTTP/1.1	
157 0.020117000	192.168.1.71	74.209.219.38	HTTP	500	GET /ao1/match?cb=https://ums.adtechus.com/mapuser?providerid=1013;userid=\$UID HTTP/1.1	
176 0.429898400	192.168.1.71	23.210.219.85	HTTP	989	GET /ad/9192.js HTTP/1.1	
194 0.014825000	192.168.1.71	54.84.236.238	HTTP	508	GET /sync?ssp=bidswitch&bidswitch_ssp_id=aol HTTP/1.1	
200 0.188420000	192.168.1.71	69.25.24.23	HTTP	1091	GET /a/9192/19861/64229-2.js?&c=b=0,1877159557158202&k_st=l&rp_s=c&p_exp=1&p_pos=atf&p_scre	
229 0.337378000	192.168.1.71	23.210.231.153	HTTP	1514	GET /AdServer/js/showad.js?rn=516430883 HTTP/1.1	
259 0.000134000	192.168.1.71	54.241.183.234	HTTP	528	GET /?provider=adiziomode=check&uid=1039da81-f78e-44cc-a317-d4139ca80c0c HTTP/1.1	
268 0.590522000	192.168.1.71	162.248.19.142	HTTP	1514	GET /AdServer/AdvertiserServlet?pubId=32702&siteId=46838&adId=80732&kadwidth=728&kadheight=90&s	
269 0.000010000	192.168.1.71	162.248.19.142	TCP	1514	41950-80 [ACK] Seq=1461 Ack=1 Win=16445440 Len=1460	
270 0.000165000	192.168.1.71	66.165.123.64	HTTP	0	GET /?s?tagid=cad674db7f73589c9a110884ce73bb72,js HTTP/1.1	
					ce 0 Sd:10;11:e2:b9) 74.125.196.139) ck: 1, Len: 1138	
					utmcn=windows-1252&utmsr=1920x1080&utmvp=1920x953&utmsc=24-bit&utmul=en-us&utmje=l&utmfl=13.0%20r0&utmdt=snopes.com%3A%20ce11%20Phone%20recharging%20Techno	
					,0;r\n	

3. According to Zillow, what instrument will Ryan learn to play?

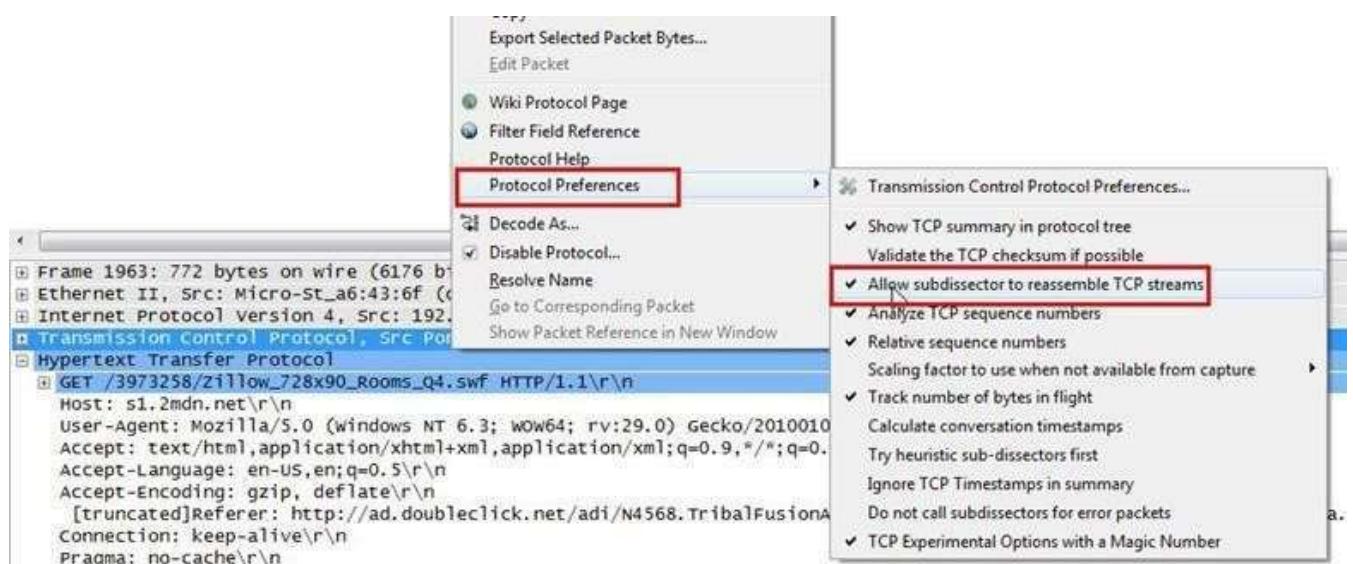
Analysis – As we did in the last challenge, we will apply a regular express filter for the Zillow keyword. Apply frame matched “(?i) zillow”

Filter: frame matches "(?i)zillow"						
Time	Source	Destination	Protocol	Length	Info	
94 0.039888000	192.168.1.71	74.125.196.139	HTTP	1192	GET /__utm.gif	
95 0.004442000	199.189.107.4	192.168.1.71	TCP	60	80-41929 [ACK]	
96 0.000769000	199.189.107.4	192.168.1.71	TCP	60	[TCP Dup ACK 9]	
97 0.060923000	199.189.107.4	192.168.1.71	TCP	60	80-41930 [FIN,	
98 0.000136000	192.168.1.71	199.189.107.4	TCP	54	41930-80 [ACK]	
99 0.000052000	192.168.1.71	199.189.107.4	TCP	54	41930-80 [FIN,	
100 0.015401000	74.125.196.139	192.168.1.71	TCP	60	80-41931 [ACK]	
101 0.000796000	74.125.196.139	192.168.1.71	HTTP	458	HTTP/1.1 200 OK	
102 0.017700000	192.168.1.71	50.19.115.152	HTTP	418	GET /?s?tagid=c	
103 0.011551000	192.168.1.71	74.125.196.139	TCP	54	41931-80 [ACK]	
104 0.029132000	199.189.107.4	192.168.1.71	TCP	60	80-41930 [ACK]	
105 0.000000000	199.189.107.4	192.168.1.71	TCP	60	[TCP Dup ACK 10]	
106 0.019119000	192.168.1.71	107.20.177.71	HTTP	462	GET /tag/cad674	
107 0.034965000	50.19.115.152	192.168.1.71	TCP	60	80-41934 [ACK]	
108 0.001555000	50.19.115.152	192.168.1.71	HTTP	338	HTTP/1.1 200 OK	
109 0.023341000	192.168.1.71	199.189.107.4	TCP	54	[TCP Retransmis	
110 0.016019000	192.168.1.71	50.19.115.152	TCP	54	41934-80 [ACK]	
111 0.010773000	107.20.177.71	107.169.1.71	TCP	60	80-41935 [ACK]	

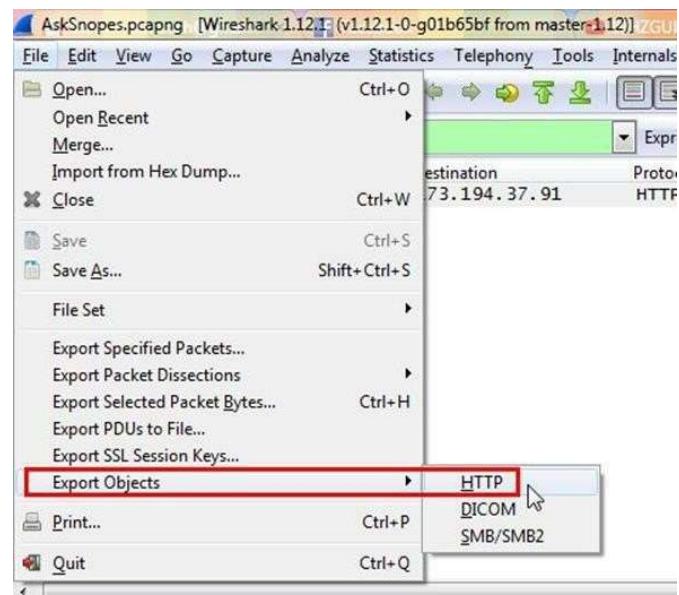
After applying the filter, we found only one packet with the Zillow keyword

lo.	Time	Source	Destination	Protocol	Length	Info
1963	0.604769000	192.168.1.71	173.194.37.91	HTTP	772	GET /3973258/zillow_728x90_Rooms_Q4.swf HTTP/1.1

Select the packet and expand the Hypertext Transfer Protocol tab right click on it go to Protocol Preferences and check Allow subdissector to reassemble TCP stream.



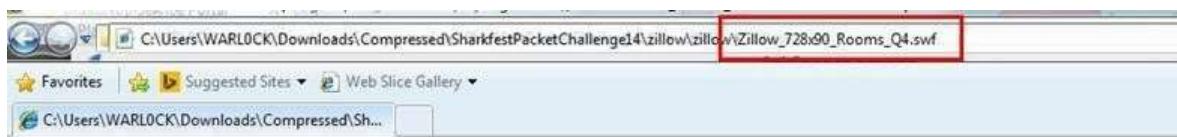
Now go to file and select Export Objects > HTTP. It will save all objects from the packet.



Click on save all.

Packet num	Hostname	Content Type	Size	Filename
52	www.snopes.com	image/jpeg	32 kB	site-bg-top.jpg
54		text/plain	15 bytes	
70	as.casalemedia.com	text/javascript	6735 bytes	cellcharge.asp&f=1&id=4240355892.9460454
101	www.google-analytics.com	image/gif	35 bytes	_utm.gif?utmwv=5.5.1&utms=1&utmn=624
108	stat.komoona.com	application/x-javascript	4 bytes	s?tagid=cad674db7f73589c9a110884ce73bb7.
112	a.komoona.com	application/x-javascript	815 bytes	cad674db7f73589c9a110884ce73bb72_728_90
129	stat.komoona.com	application/x-javascript	4 bytes	s?tagid=cad674db7f73589c9a110884ce73bb7.
133	adserver.adtechus.com	application/x-javascript	431 bytes	ADTECH;loc=100;target=_blank;misc=%5BTI
154	s.komoona.com	application/x-javascript	5603 bytes	cad674db7f73589c9a110884ce73bb72.js
182	ads.rubiconproject.com	text/javascript	18 kB	9192.js
205	optimized-by.rubiconproject.com	text/javascript	1852 bytes	64229-2.js?&cb=0.18771559557158202&tk_st:
212	ocsp.thawte.com	application/ocsp-request	115 bytes	\
215	ocsp.thawte.com	application/ocsp-response	1421 bytes	\
223	ocsp.thawte.com	application/ocsp-request	115 bytes	\
225	ocsp.thawte.com	application/ocsp-response	1421 bytes	\
251	ads.pubmatic.com	text/html	54 kB	showad.js?rn=516430883
261	x.skimresources.com	application/json	79 bytes	?provider=adizio&mode=check&uid=1039d;
330	pr.ybp.yahoo.com	image/gif	43 bytes	E6EF997B-80FE-4373-AB1F-500144B03A7B
334	rt.legolas-media.com	image/gif	6 bytes	lgrt?ci=12&ti=64523&pbi=11057
346	um.eqads.com	text/html	196 bytes	pub.aspx?
353	ads.pubmatic.com	text/html	454 bytes	ro_914.html

After saving all files in a directory and we found a swf file with name Zillow. After opening the flash file, we saw that Zillow was trying to learn saxophone.

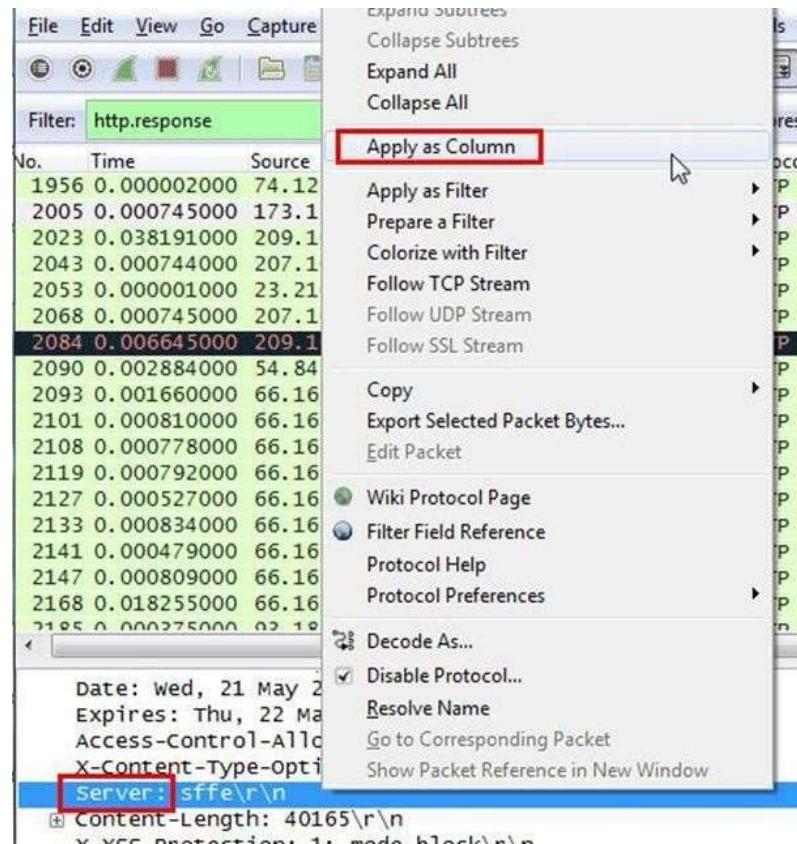


4. How many web servers are running Apache?

Analysis – The web server name can be retrieved from HTTP response header. So will apply filter http.response and we can see all http response packets.

No.	Time	Source	Destination	Protocol	Length	Info
1956	0.000002000	74.125.21.154	192.168.1.71	HTTP	432	HTTP/1.1 200 OK (text/javascript)
2005	0.000745000	173.194.37.91	192.168.1.71	HTTP	580	HTTP/1.1 200 OK (application/javascript)
2023	0.038191000	209.107.194.81	192.168.1.71	HTTP	1478	HTTP/1.1 200 OK (application/javascript)
2043	0.000744000	207.109.230.154	192.168.1.71	HTTP	1054	HTTP/1.1 200 OK (text/html)
2053	0.000001000	23.210.231.153	192.168.1.71	HTTP	178	HTTP/1.1 200 OK
2068	0.000745000	207.109.230.154	192.168.1.71	HTTP	1054	HTTP/1.1 200 OK (text/html)
2084	0.006645000	209.107.194.81	192.168.1.71	HTTP	1478	[TCP Retransmission] HTTP/1.1 200 OK (text/html)
2090	0.002884000	54.84.148.104	192.168.1.71	HTTP	626	HTTP/1.1 200 OK (GIF89a)
2093	0.001660000	66.165.133.65	192.168.1.71	HTTP	1201	HTTP/1.1 200 OK (GIF89a)
2101	0.000810000	66.165.133.65	192.168.1.71	HTTP	673	HTTP/1.1 200 OK (GIF89a)
2108	0.000778000	66.165.133.65	192.168.1.71	HTTP	324	HTTP/1.1 200 OK (GIF89a)
2119	0.000792000	66.165.133.65	192.168.1.71	HTTP	176	HTTP/1.1 200 OK (GIF89a)
2127	0.000527000	66.165.133.65	192.168.1.71	HTTP	591	HTTP/1.1 200 OK (GIF89a)
2133	0.000834000	66.165.133.65	192.168.1.71	HTTP	482	HTTP/1.1 200 OK (GIF89a)
2141	0.000479000	66.165.133.65	192.168.1.71	HTTP	592	HTTP/1.1 200 OK (GIF89a)
2147	0.000809000	66.165.133.65	192.168.1.71	HTTP	1414	HTTP/1.1 200 OK (GIF89a)

Now we will set the server header as column select any packet and right click on it then select Apply as Column.



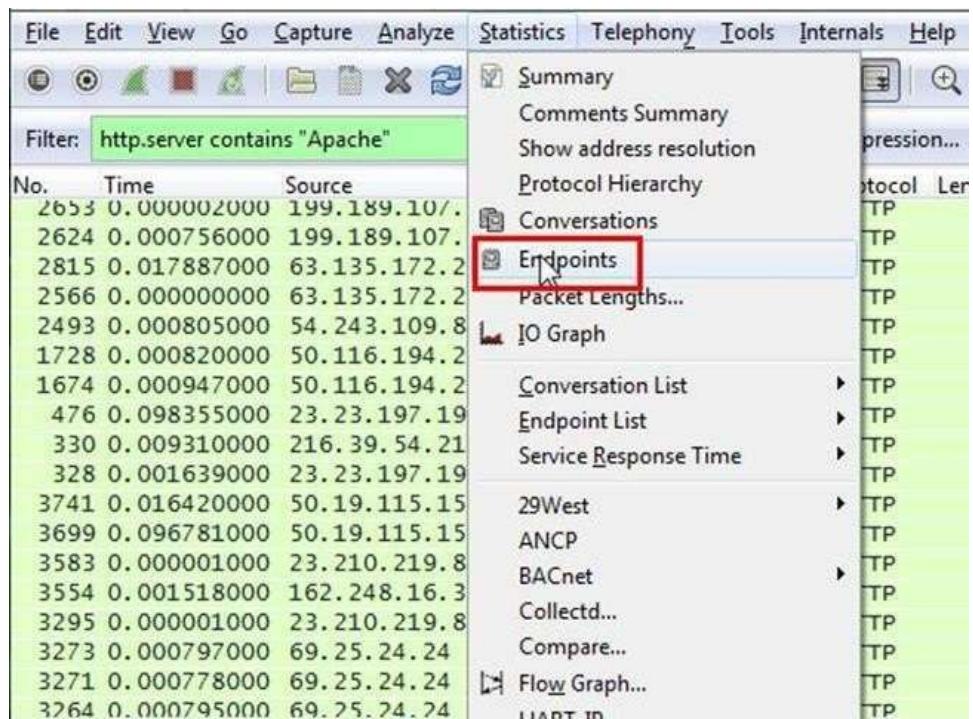
Now can see the server column where all server name is showing.

Destination	Protocol	Length	Server	Info
192.168.1.71	HTTP	828	sffe	HTTP/1.1 200 OK (JPEG JFIF image)
192.168.1.71	HTTP	580	sffe	HTTP/1.1 200 OK (application/x-shockwave-flash)
192.168.1.71	HTTP	807	sffe	HTTP/1.1 200 OK (text/javascript)
192.168.1.71	HTTP	463	sffe	HTTP/1.1 200 OK (text/javascript)
192.168.1.71	HTTP	959	radiumone/1.2	HTTP/1.1 200 OK (GIF89a)
192.168.1.71	HTTP	525	radiumone/1.2	HTTP/1.1 200 OK (text/html)
192.168.1.71	HTTP	875	post/2.0	HTTP/1.1 200 OK (application/x-javascript)
192.168.1.71	OCSP	829	ocsp_responder	response
192.168.1.71	HTTP	1159	nginx/1.5.3	HTTP/1.1 302 Found
192.168.1.71	HTTP	1092	nginx/1.5.3	HTTP/1.1 302 Found
192.168.1.71	HTTP	626	nginx/1.4.7	HTTP/1.1 200 OK (GIF89a)
192.168.1.71	HTTP	685	nginx/1.4.7	HTTP/1.1 302 Moved Temporarily
192.168.1.71	HTTP	626	nginx/1.4.7	HTTP/1.1 200 OK (GIF89a)
192.168.1.71	HTTP	626	nginx/1.4.7	HTTP/1.1 200 OK (GIF89a)
192.168.1.71	HTTP	681	nginx/1.4.7	HTTP/1.1 302 Moved Temporarily
192.168.1.71	HTTP	323	nginx/1.4.3	TCP out-Of-Order HTTP/1.1 302 Found
192.168.1.71	HTTP	303	nginx/1.4.3	HTTP/1.1 302 Found
192.168.1.71	HTTP	225	nginx/1.2.0	HTTP/1.1 200 OK (application/x-javascript)

Now we have to check how many Apache packets are there we can't count manually for each packet so we will apply another filter http.server contains "Apache"

No.	Time	Source	Destination	Protocol	Length	Server
1811	0.051151000	50.19.115.152	192.168.1.71	HTTP	338	Apache
1609	0.003943000	50.19.115.152	192.168.1.71	HTTP	338	Apache
1483	0.000002000	23.210.219.85	192.168.1.71	HTTP	1078	Apache
1344	0.000747000	23.210.219.85	192.168.1.71	HTTP	1078	Apache
1317	0.016574000	50.19.115.152	192.168.1.71	HTTP	338	Apache
1295	0.000774000	107.20.177.71	192.168.1.71	HTTP	515	Apache
1287	0.001961000	50.19.115.152	192.168.1.71	HTTP	338	Apache
1222	0.015700000	207.109.230.161	192.168.1.71	HTTP	765	Apache
1173	0.001648000	69.25.24.24	192.168.1.71	HTTP	1171	Apache
1165	0.001172000	69.25.24.24	192.168.1.71	HTTP	1160	Apache
1139	0.001222000	69.25.24.24	192.168.1.71	HTTP	1121	Apache
669	0.001691000	69.25.24.24	192.168.1.71	HTTP	1128	Apache
182	0.000744000	23.210.219.85	192.168.1.71	HTTP	1078	Apache
129	0.038194000	50.19.115.152	192.168.1.71	HTTP	338	Apache
112	0.002082000	107.20.177.71	192.168.1.71	HTTP	955	Apache
108	0.001555000	50.19.115.152	192.168.1.71	HTTP	338	Apache
70	0.000001000	207.109.230.161	192.168.1.71	HTTP	408	Apache

After applying filter go to Statistics > Endpoints



It will show all connections

Ethernet: 7	Fibre Channel	FDDI	IPv4: 107	IPv6: 4	IPX	JXTA	NCP	RSVP	SCTP	TCP: 361	Token
IPv4 Endpoints											
Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes	Latitude	Lc			
192.168.1.71	3 987	1 814 693	1 976	413 339	2 011	1 401 354	-	-			
192.168.1.254	409	50 248	187	32 761	222	17 487	-	-			
74.125.196.139	10	2 118	4	644	6	1 474	-	-			
207.109.230.161	30	12 164	15	9 252	15	2 912	-	-			
64.49.225.166	20	6 963	11	6 018	9	945	-	-			
192.168.1.68	16	1 088	16	1 088	0	0	-	-			
224.0.0.252	36	2 432	0	0	36	2 432	-	-			
66.165.133.65	535	289 649	264	243 481	271	46 168	-	-			
108.160.167.165	45	4 923	20	2 083	25	2 840	-	-			
50.19.115.152	50	13 256	18	4 706	32	8 550	-	-			
107.20.177.71	29	6 905	13	4 011	16	2 894	-	-			
199.189.107.4	209	160 954	133	154 206	76	6 748	-	-			
192.168.1.66	16	1 088	16	1 088	0	0	-	-			
64.12.239.201	74	10 457	38	5 410	36	5 047	-	-			
176.32.99.164	55	36 111	29	30 476	26	5 635	-	-			
54.85.82.173	21	3 224	9	1 739	12	1 485	-	-			
74.209.219.38	22	2 796	11	1 168	11	1 628	-	-			
23.210.219.85	56	43 884	31	34 152	25	9 732	-	-			
54.84.236.238	10	1 733	4	943	6	790	-	-			
69.25.24.23	88	34 477	39	22 618	49	11 859	-	-			
23.7.139.27	15	5 288	7	3 912	8	1 376	-	-			
23.210.231.153	314	237 690	179	173 883	135	63 807	-	-			

Name resolution Limit to display filter
[Help](#) [Copy](#) Limit the list to endpoints matching the current display filter.

Check the limit to display filter then it will show the actual Apache connections. Now there are showing 22 connections but will exclude 192.168.1.71 because it is client's IP not a server IP so there are actual 21Apache servers.

IPv4 Endpoints - Filter: http.sen

Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes	Latitude
207.109.230.161	2	1 173	2	1 173	0	0	
192.168.1.71	80	60 911	0	0	80	60 911	
50.19.115.152	13	4 394	13	4 394	0	0	
107.20.177.71	4	3 143	4	3 143	0	0	
23.210.219.85	6	6 468	6	6 468	0	0	
23.210.231.153	12	6 163	12	6 163	0	0	
23.23.197.19	2	1 179	2	1 179	0	0	
216.39.54.212	1	225	1	225	0	0	
162.248.19.136	3	2 363	3	2 363	0	0	
162.248.16.24	2	1 692	2	1 692	0	0	
69.25.24.24	13	15 024	13	15 024	0	0	
207.109.230.154	3	3 162	3	3 162	0	0	
50.97.236.98	2	1 753	2	1 753	0	0	
69.25.24.26	3	3 087	3	3 087	0	0	
50.116.194.21	1	1 045	1	1 045	0	0	
50.116.194.28	1	527	1	527	0	0	
54.243.109.84	1	609	1	609	0	0	
63.135.172.251	2	837	2	837	0	0	
199.189.107.4	4	3 950	4	3 950	0	0	
50.63.243.230	1	1 007	1	1 007	0	0	
207.109.230.187	3	3 036	3	3 036	0	0	
162.248.16.37	1	74	1	74	0	0	

Name resolution Limit to display filter

CONCLUSION: We have successfully analyzed the packets provided and solved the questions using wireshark.

PRACTICAL - 6

AIM: Using Sysinternals tools for Network Tracking and Process Monitoring

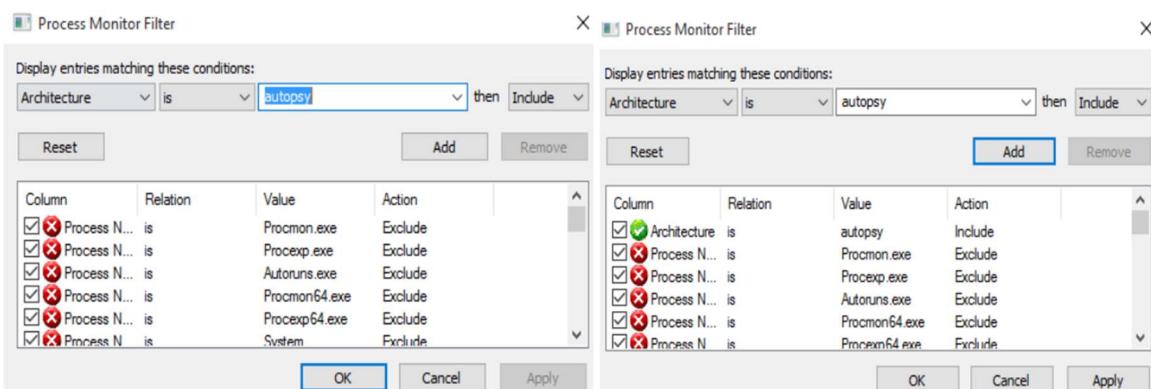
1. Check Sysinternals tools

→
Google sysinternal tools

2. Monitor Live Processes

Process Monitor is an advanced monitoring tool for Windows that shows real-time file system, Registry and process/thread activity. It combines the features of two legacy Sysinternals utilities, *Filemon* and *Regmon*, and adds an extensive list of enhancements including rich and non-destructive filtering, comprehensive event properties such session IDs and user names, reliable process information, full thread stacks with integrated symbol support for each operation, simultaneous logging to a file, and much more. Its uniquely powerful features will make *Process Monitor* a core utility in your system troubleshooting and malware hunting toolkit.

Sysinternal procmn



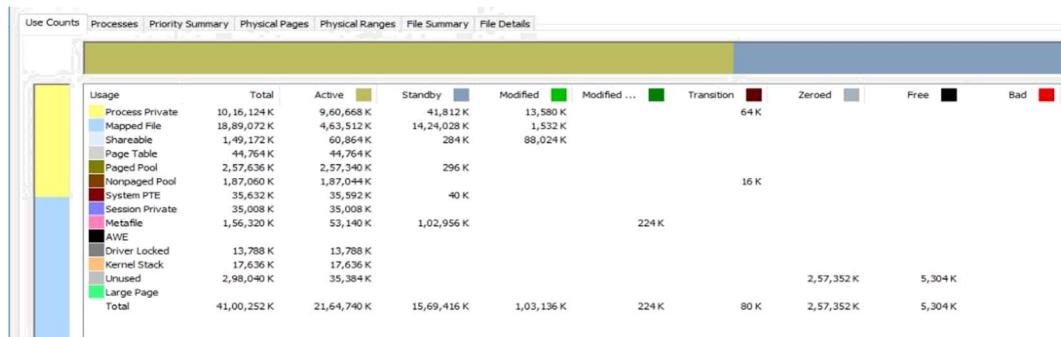
3. Capture RAM

RAMMap is an advanced physical memory usage analysis utility for Windows Vista and higher. It presents usage information in different ways on its several different tabs:

- Use Counts*: usage summary by type and paging list
- Processes*: process working set sizes
- Priority Summary*: prioritized standby list sizes
- Physical Pages*: per-page use for all physical memory
- Physical Ranges*: physical memory addresses
- File Summary*: file data in RAM by file
- File Details*: individual physical pages by file

Use RAMMap to gain understanding of the way Windows manages memory, to analyze application memory usage, or to answer specific questions about how RAM is being allocated. RAMMap's refresh feature enables you to update the display and it includes support for saving and loading memory snapshots.

STEPS sysinternals → RAMMap



4.Capture TCP/UDP packets

TCPView is a Windows program that will show you detailed listings of all TCP and UDP endpoints on your system, including the local and remote addresses and state of TCP connections. On Windows Server 2008, Vista, and XP, TCPView also reports the name of the process that owns the endpoint. TCPView provides a more informative and conveniently presented subset of the Netstat program that ships with Windows. The TCPView download includes Tcpcvcon, a command-line version with the same functionality.

Using TCPView

When you start TCPView it will enumerate all active TCP and UDP endpoints, resolving all IP addresses to their domain name versions. You can use a toolbar button or menu item to toggle the display of resolved names. On Windows XP systems, TCPView shows the name of the process that owns each endpoint. By default, TCPView updates every second, but you can use the **Options|Refresh Rate** menu item to change the rate. Endpoints that change state from one update to the next are highlighted in yellow; those that are deleted are shown in red, and new endpoints are shown in green. You can close established TCP/IP connections (those labeled with a state of ESTABLISHED) by selecting **File|Close Connections**, or by right-clicking on a connection and choosing **Close Connections** from the resulting context menu. You can save TCPView's output window to a file using the **Save** menu item.

Using Tcpcvcon

Tcpcvcon usage is similar to that of the built-in Windows netstat utility:

Usage: tcpcvcon [-a] [-c] [-n] [process name or PID]

Parameter Description

-a Show all endpoints (default is to show established TCP connections).

-c Print output as CSV.

-n Don't resolve addresses.

STEPS

Download TCPView

Process /	PID	Protocol	Local Address	Local Port	Remote Address	Remote Port	State	Sent Packets	Sent Bytes	Rcvd Packets	Rcvd Bytes
chrome.exe	5808	TCP	desktop-vgrdu	16233	ss-in[198.1.100.net]	5229	ESTABLISHED				
chrome.exe	5808	TCP	desktop-vgrdu	16475	111.221.23.254	https	ESTABLISHED				
chrome.exe	5808	TCP	desktop-vgrdu	16477	172.217.194.157	https	ESTABLISHED				
chrome.exe	5808	TCP	desktop-vgrdu	16495	bom0511-in[14.1..]	https	ESTABLISHED				
chrome.exe	5808	TCP	desktop-vgrdu	16501	bom0715-in[10.1..]	https	ESTABLISHED				
chrome.exe	5808	TCP	desktop-vgrdu	16520	40.77.226.250	https	ESTABLISHED	1	3,090	1	436
chrome.exe	5808	TCP	desktop-vgrdu	16525	151.101.36.133	https	ESTABLISHED				
chrome.exe	5808	TCP	desktop-vgrdu	16528	52.162.216.193	https	ESTABLISHED				
chrome.exe	5808	TCP	desktop-vgrdu	16529	52.162.216.193	https	ESTABLISHED				
chrome.exe	5808	TCP	desktop-vgrdu	16530	52.162.216.193	https	ESTABLISHED				
chrome.exe	5808	UDP	DESKTOP-VGRDL...	5383	*	*				33	695
chrome.exe	5808	UDP	DESKTOP-VGRDL...	5383	*	*					
chrome.exe	5808	UDP	DESKTOP-VGRDL...	5383	*	*					
chrome.exe	5808	UDPV6	[0.0.0.0.0.0]	5383	*	*					
chrome.exe	5808	UDPV6	[0.0.0.0.0.0]	5383	*	*					
explorer.exe	3248	TCP	desk-top-vgrdu	16249	52.230.84.0	Https	ESTABLISHED				
lsass.exe	756	TCP	DESKTOP-VGRDL...	1540	DESKTOP-VGRDL...	0	LISTENING				
lsass.exe	756	TCPV6	[0.0.0.0.0.0]	1540	[0.0.0.0.0.0]	0	LISTENING				
msrnd.exe	2280	TCP	DESKTOP-VGRDL...	3306	DESKTOP-VGRDL...	0	LISTENING				
oracle.exe	2296	TCP	DESKTOP-VGRDL...	1544	DESKTOP-VGRDL...	0	LISTENING				
oracle.exe	2296	TCPV6	[0.0.0.0.0.0]	1544	[0.0.0.0.0.0]	0	LISTENING				
oracle.exe	2296	TCPV6	[fe80::0:68b5::1..]	2239	[fe80::0:68b5::1..]	1521	ESTABLISHED	1	414	1	202
services.exe	772	TCP	DESKTOP-VGRDL...	1545	DESKTOP-VGRDL...	0	LISTENING				
services.exe	772	TCPV6	[0.0.0.0.0.0]	1545	[0.0.0.0.0.0]	0	LISTENING				
spoolsv.exe	1772	TCP	DESKTOP-VGRDL...	1539	DESKTOP-VGRDL...	0	LISTENING				
spoolsv.exe	1772	TCPV6	[0.0.0.0.0.0]	1539	[0.0.0.0.0.0]	0	LISTENING				
svchost.exe	940	TCP	DESKTOP-VGRDL...	epmap	DESKTOP-VGRDL...	0	LISTENING				
svchost.exe	1144	TCP	DESKTOP-VGRDL...	1537	DESKTOP-VGRDL...	0	LISTENING				
svchost.exe	336	TCP	DESKTOP-VGRDL...	1538	DESKTOP-VGRDL...	0	LISTENING				
svchost.exe	1136	UDP	DESKTOP-VGRDL...	nlp	*	*					
svchost.exe	1152	UDP	DESKTOP-VGRDL...	sdcp	*	*					
svchost.exe	1152	UDP	desk-top-vgrdu	sdcp	*	*					
svchost.exe	1152	UDP	DESKTOP-VGRDL...	ws-discovery	*	*					
svchost.exe	1152	UDP	DESKTOP-VGRDL...	ws-discovery	*	*					
svchost.exe	1238	UDP	DESKTOP-VGRDL...	5383	*	*				11	132
svchost.exe	1238	UDP	DESKTOP-VGRDL...	limtr	*	*				33	695
svchost.exe	1152	UDP	DESKTOP-VGRDL...	53517	*	*					
svchost.exe	1152	UDP	desktop-vgrdu	57748	*	*					
svchost.exe	1152	UDP	DESKTOP-VGRDL...	57749	*	*					
svchost.exe	940	TCPV6	[0.0.0.0.0.0]	epmap	[0.0.0.0.0.0]	0	LISTENING				
svchost.exe	1144	TCPV6	[0.0.0.0.0.0]	1537	[0.0.0.0.0.0]	0	LISTENING				
svchost.exe	336	TCPV6	[0.0.0.0.0.0]	1538	[0.0.0.0.0.0]	0	LISTENING				
svchost.exe	1136	UDPV6	[0.0.0.0.0.0]	123	*	*					
svchost.exe	1152	UDPV6	[0.0.0.0.0.0]	1000	*	*					

Activate Windows
Go to Settings to activate Windc

5. Monitor Hard Disk

DiskMon is an application that logs and displays all hard disk activity on a Windows system. You can also minimize *DiskMon* to your system tray where it acts as a disk light, presenting a green icon when there is disk-read activity and a red icon when there is disk-write activity.

→ Download DiskMon Run as administrator

#	Time	Duration (s)	Disk	Request	Sector	Length
7072	73.892145	0.000000000	0	Write	121125280	32
7073	73.892748	0.000000000	0	Write	121125280	32
7074	73.893353	0.000000000	0	Write	121125280	32
7075	73.894042	0.000000000	0	Write	121125280	32
7076	73.894725	0.000000000	0	Write	7168600	16

6.Monitor Virtual Memory

VMMMap is a process virtual and physical memory analysis utility. It shows a breakdown of a process's committed virtual memory types as well as the amount of physical memory (working set) assigned by the operating system to those types. Besides graphical representations of memory usage, VMMMap also shows summary information and a detailed process memory map. Powerful filtering and refresh capabilities allow you to identify the sources of process memory usage and the memory cost of application features

sysinternals VMMMap



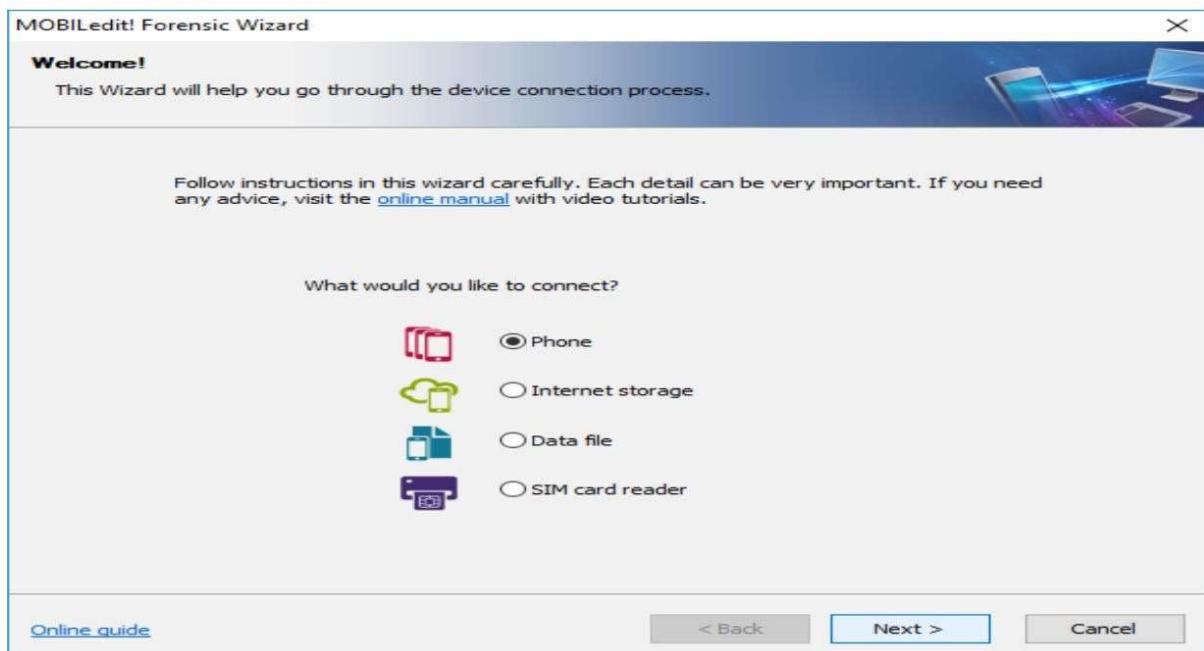
7.Monitor Cache Memory

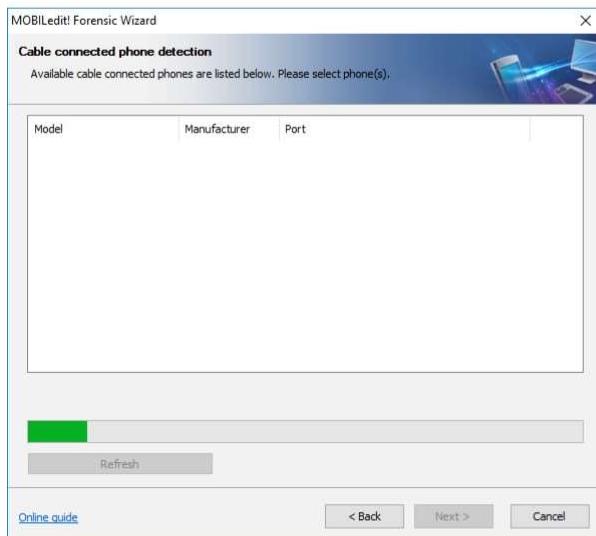
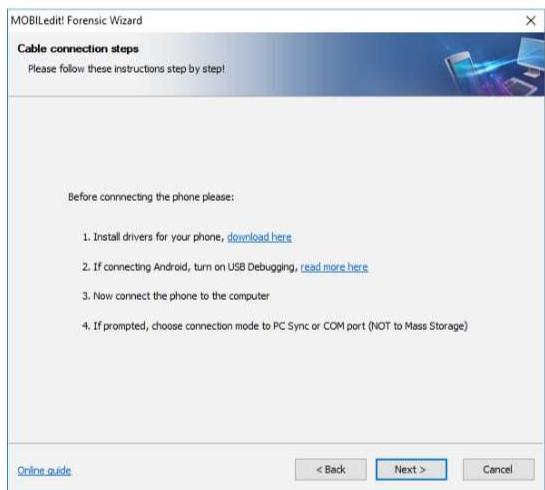
CacheSet is an applet that allows you to manipulate the working-set parameters of the system file cache. Unlike CacheMan, *CacheSet* runs on all versions of NT and will work without modifications on new Service Pack releases. In addition to providing you the ability to control the minimum and maximum working set sizes, it also allows you to reset the Cache's working set, forcing it to grow as necessary from a minimal starting point. Also unlike CacheMan, changes made with *CacheSet* have an immediate effect on the size of the Cache.

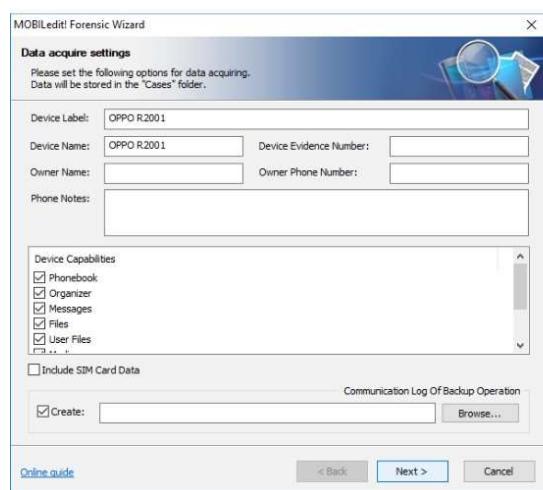
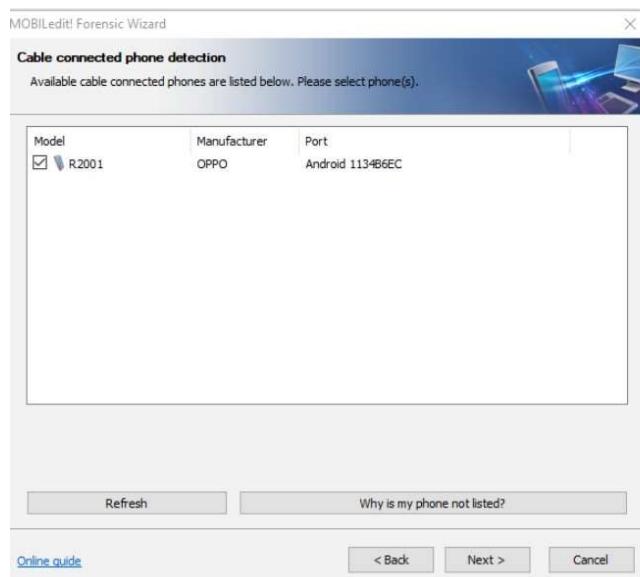
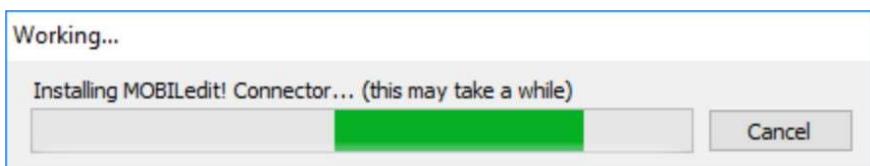
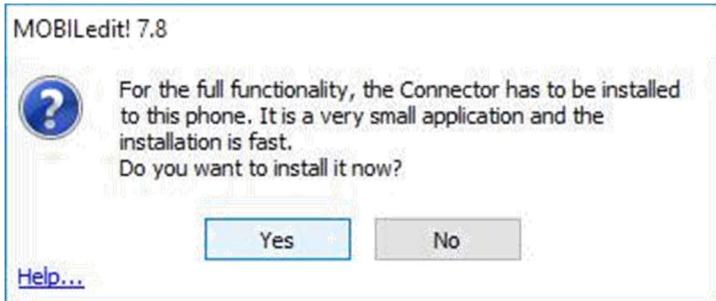
The figure shows the CacheSet applet interface. It includes sections for Cache Information (Current size: 136260 KB, Peak size: 280556 KB), Adjust Cache Settings (Working set minimum: 1024 KB, Working set maximum: -4 KB), and buttons for Apply, Clear, Reset, Cancel, and a SysInternals logo.

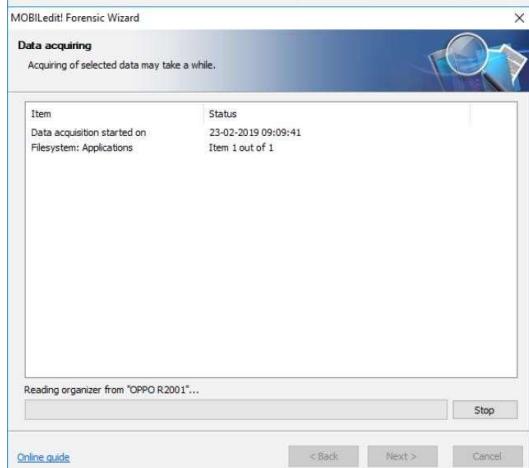
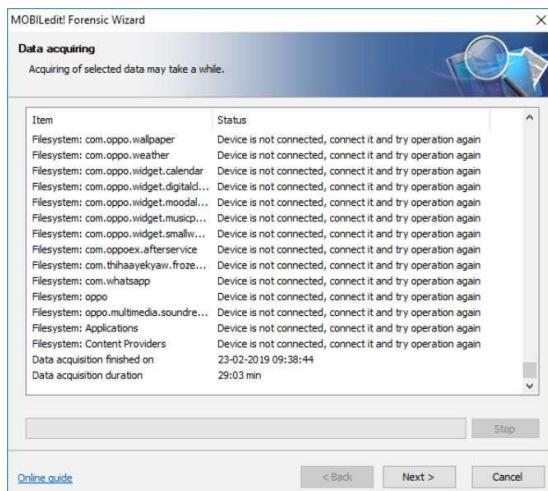
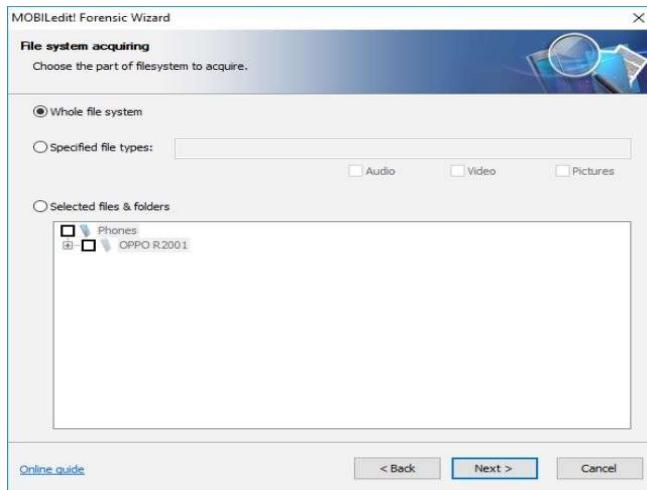
PRACTICAL - 7

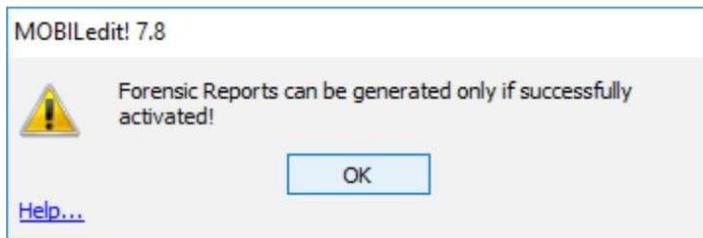
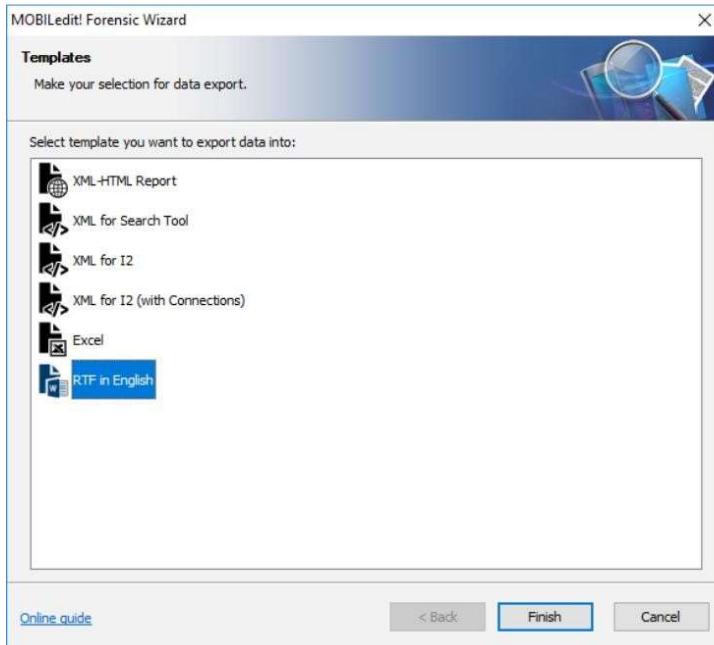
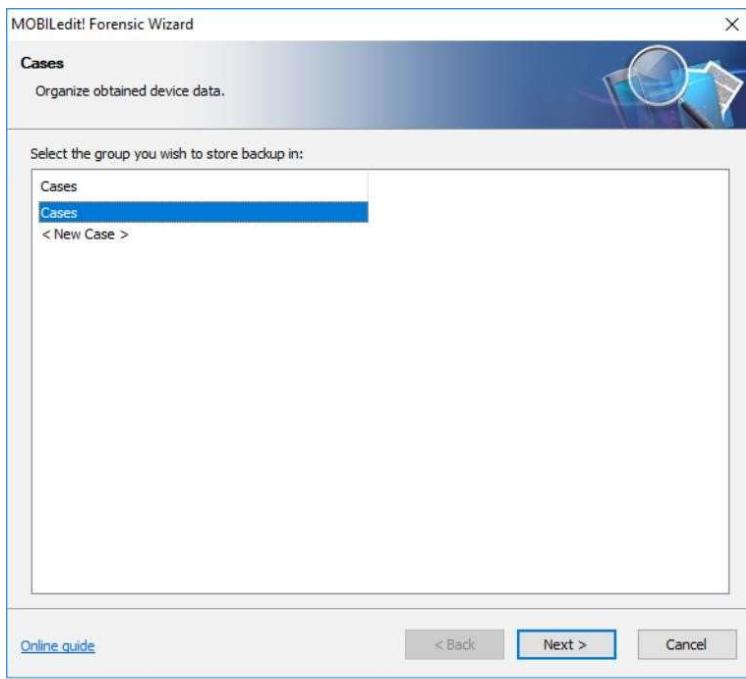
AIM: Investigate the Mobile Device.

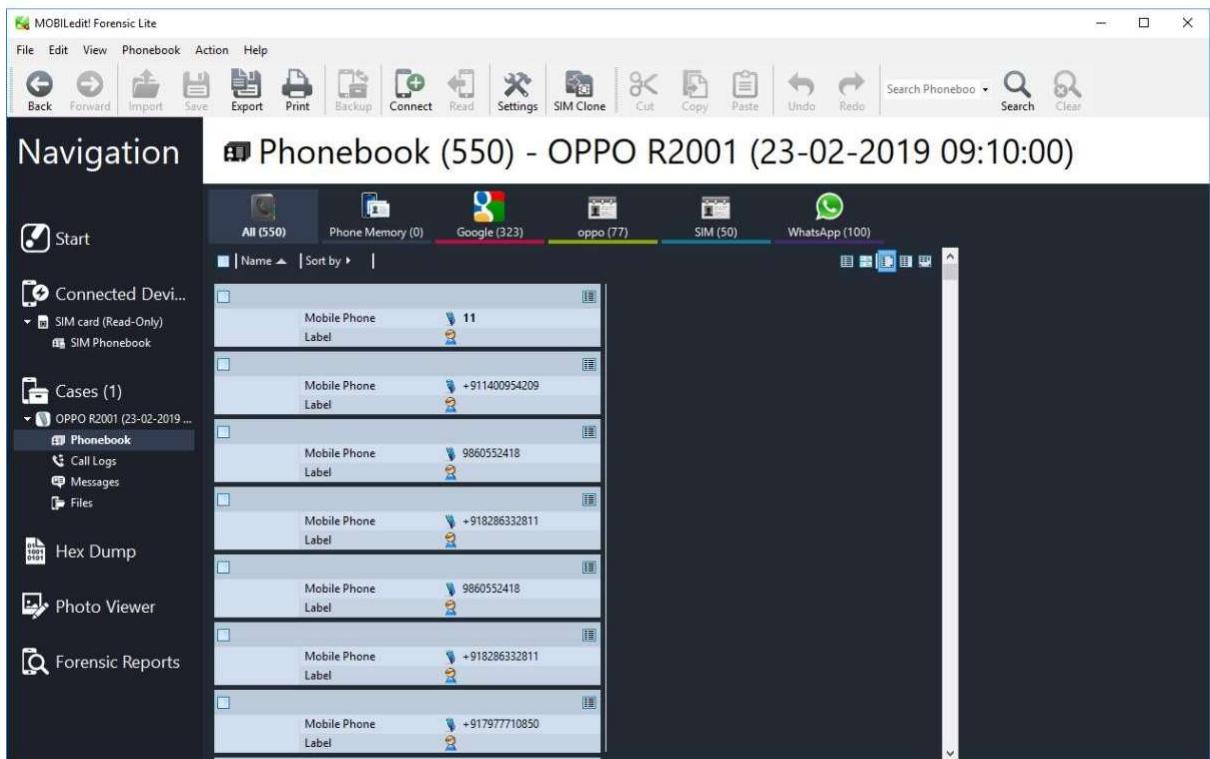
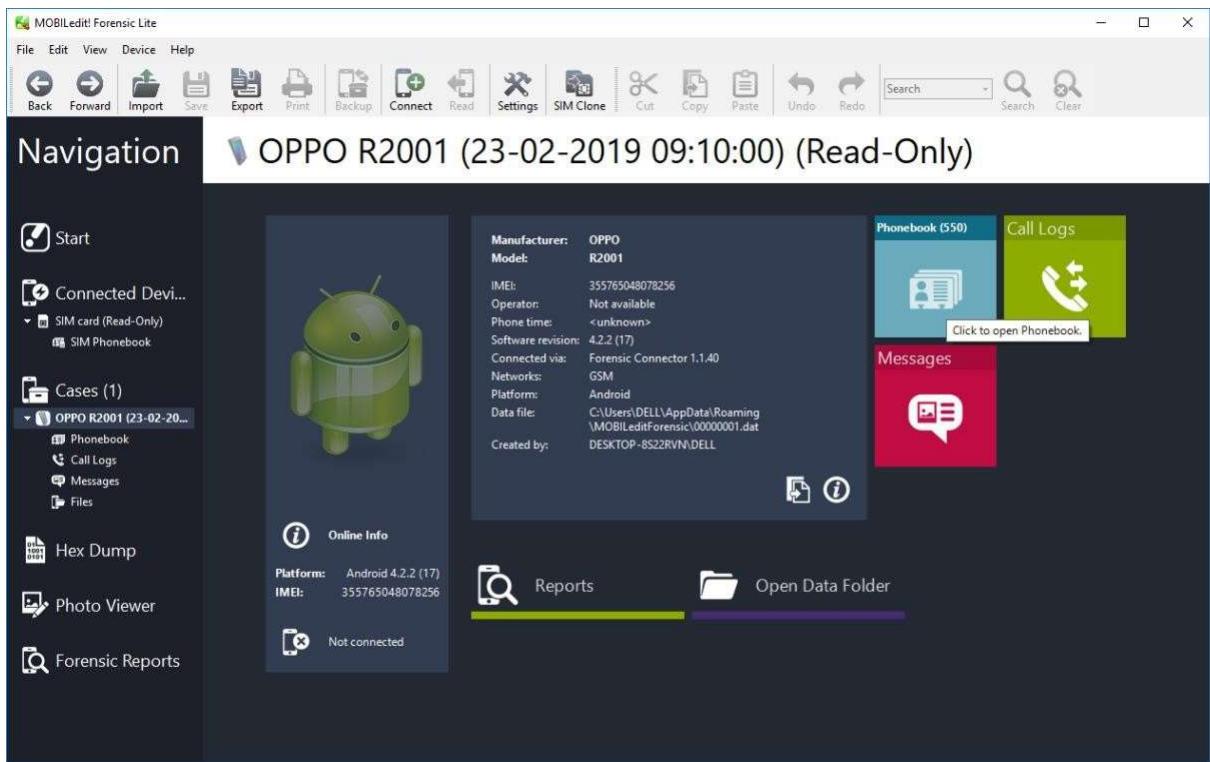












Navigation

Call Logs (97) - OPPO R2001 (23-02-2019 09:10:00)

Name	Number	Time
	+911400954501	22-02-2019 20:10:12
	+911400954448	22-02-2019 16:23:14
	+911400954496	22-02-2019 14:37:00
	+911400954490	21-02-2019 15:44:20
Sainat	+919930547554	20-02-2019 11:38:44
Sainat	+919930547554	20-02-2019 11:29:22
Sainat	+919930547554	20-02-2019 10:16:51
	+911400954496	19-02-2019 16:30:13
	+912239502000	19-02-2019 10:04:24
	+917977438836	18-02-2019 21:26:18
	+917977438836	18-02-2019 21:19:07
Papaa	+919004480339	18-02-2019 20:25:20
Santosh Bhai	+919702346277	18-02-2019 20:17:29
	+911400954437	18-02-2019 19:43:53
Aanad IY	+918779088436	17-02-2019 21:44:42
Aanad IY	+918779088436	17-02-2019 21:29:40

Navigation

Messages - OPPO R2001 (23-02-2019 09:10:00)

Sender / Recipient	Time
55256	23-02-2019 08:25:27
IZ-IDEA	22-02-2019 20:55:25
Aaaaa	22-02-2019 17:19:31
IM-655456	22-02-2019 14:46:43
IM-612345	22-02-2019 14:15:48
IM-6554563	22-02-2019 10:34:12
+919987501727	21-02-2019 16:52:49
MD-KOTAKB	21-02-2019 16:44:12
AX-IYCGOV	21-02-2019 12:05:36
IM-657886	
55256	23-02-2019 08:25:27
Don't Wait! Start playing now to Win GOLD voucher worth Rs 25000. CALL 55256 Tollfree. TnC	23-02-2019 08:25:27
55256	22-02-2019 14:03:05
ओर बैंक करें Re 35 और 65 को रदीचार्ज जीतें। ठायल करें 55256 मुफ्त से।	22-02-2019 14:03:05
55256	22-02-2019 08:27:34
Chance! Win Rs 65 & Rs 35 worth FREE Recharge Everyday. Dial *777# (Tollfree). TnC	22-02-2019 08:27:34
55256	21-02-2019 14:03:26
Rs 5000 का नवक कार्ड जीतें और आनंद लें। ठायल करें 55256 मुफ्त से।	21-02-2019 14:03:26
55256	21-02-2019 08:19:34
Limited Chance! Call 55256 Tollfree and Win Rs 65 worth Recharge Everyday. TnC	21-02-2019 08:19:34
55256	20-02-2019 12:31:37
मुफ्त ऑफर। रु 35 रोचार्ज जीतो हर रोज़, कठोर 55256 टॉलफ्रीटो	20-02-2019 12:31:37
55256	26-01-2019 09:09:37
Special OFFER! Aaj khelo aur jeet sakte ho Rs.1000 CashWho is Known as MAHATMATA () Gandhiji() NehruReply now A or B to 55256TnC & Aaj Jeeto	26-01-2019 09:09:37

PRACTICAL - 8

AIM: Investigate Email File Given.

FTK can filter or find files specific to e-mail clients and servers. You can configure these filters when you enter search parameters.

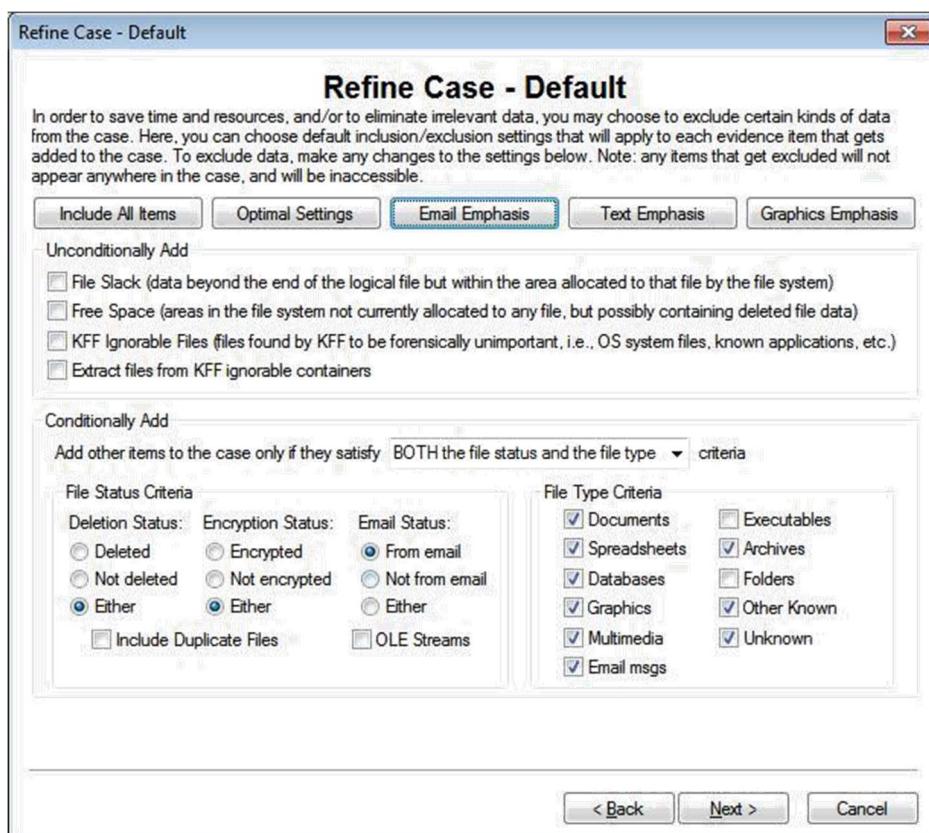
Because of Jim's responses to a poor performance review, the CEO of SuperiorBicycles, Martha Dax, suspects he might have obtained sensitive information about the company's business model that he's leaking to a competitor.

Martha asked her CIO, to have an IT employee copy the Outlook .pst file from Jim Shu's old computer to a USB drive.

To process this investigation, we need to examine the Jim_shu's.pst file, locate the message, and export it for further analysis of its header to see how Jim might have received it.

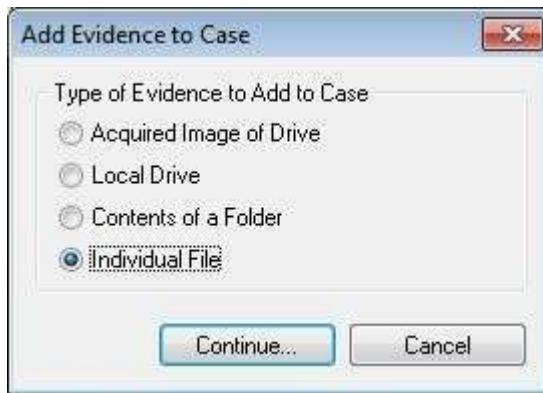
Recovering Email

Start AccessData FTK and click **Start a new case**, then click **OK**. Click **Next** until you reach the **Refine Case - Default** dialog box. Click the **Email Emphasis** button, and then click **Next**.

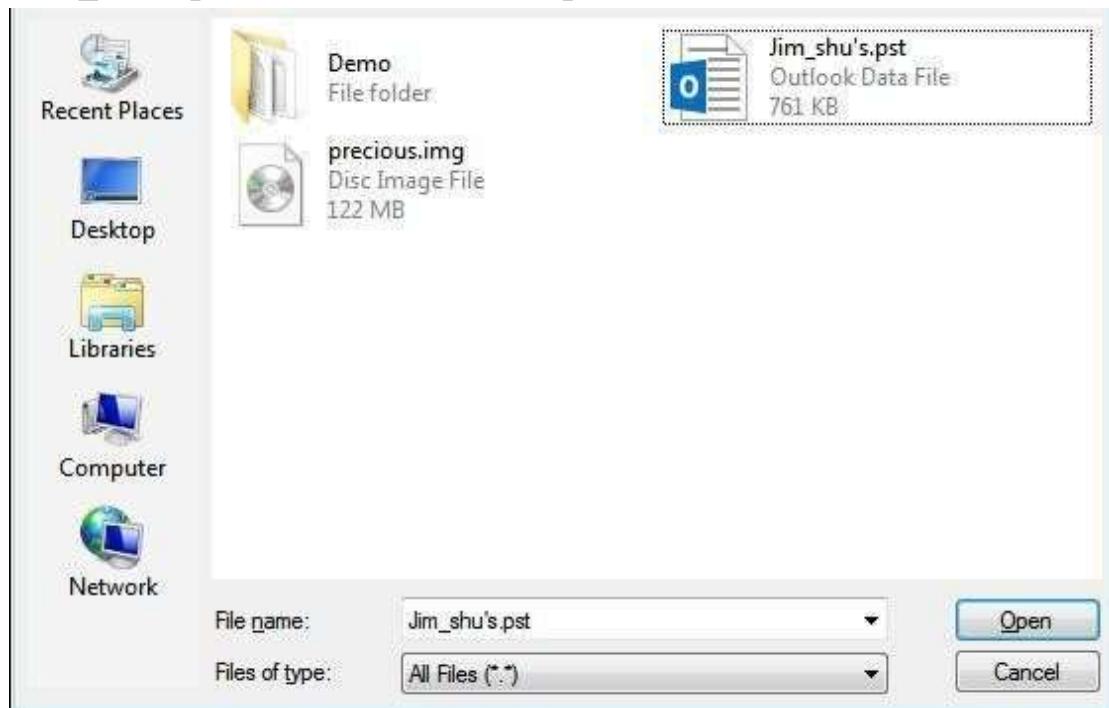


Click **Next** until you reach the **Add Evidence to Case** dialog box, and then click the **Add Evidence** button.

In the Add Evidence to Case dialog box, click the **Individual File** option button, and then click **Continue**.



In the **Select File** dialog box, navigate to your work folder, click the **Jim_shu's.pst** file, and then click **Open**.



When the **Add Evidence to Case** dialog box opens, click **Next**. In the **Case summary** dialog box, click **Finish**.

When FTK finishes processing the file, in the main FTK window, click the **E-mail Messages** button, and then click the **Full Path** column header to sort the records.

File Name	Full Path	Recycle Bi...	Ext	File Type	Category	Subject	Cr Date	Mod Date	Acc Date	L-Size	P-Size	Chi
Message0001	C:\Users\Administrator\Desktop\All Docs of Des...			E-mail Messa...	E-mail	'RE: Bike ...	12/3/2006 10:05:51...	12/3/2006 10:05:51...	N/A	3,016	3,016	
Message0001	C:\Users\Administrator\Desktop\All Docs of Des...			E-mail Messa...	E-mail	'Request!'	12/3/2006 9:06:44 PM	12/7/2006 6:39:39 PM	N/A	3,488	3,488	
Message0001	C:\Users\Administrator\Desktop\All Docs of Des...			E-mail Messa...	E-mail	'RE: Bicyc...	12/3/2006 9:09:12 PM	12/3/2006 9:09:12 PM	N/A	1,472	1,472	
Message0001	C:\Users\Administrator\Desktop\All Docs of Des...			E-mail Messa...	E-mail	'problem'	12/3/2006 9:06:45 PM	12/7/2006 6:39:27 PM	N/A	3,552	3,552	
Message0002	C:\Users\Administrator\Desktop\All Docs of Des...			E-mail Messa...	E-mail	'FW: probl...	12/7/2006 6:39:22 PM	12/7/2006 6:39:22 PM	N/A	1,854	1,854	
Message0002	C:\Users\Administrator\Desktop\All Docs of Des...			E-mail Messa...	E-mail	'Bike spec...	12/3/2006 9:06:40 PM	12/7/2006 6:39:57 PM	N/A	2,836	2,836	
Message0002	C:\Users\Administrator\Desktop\All Docs of Des...			E-mail Messa...	E-mail	'E: Bike ...	12/3/2006 9:08:27 PM	12/3/2006 9:08:27 PM	N/A	1,588	1,588	
Message0002	C:\Users\Administrator\Desktop\All Docs of Des...			E-mail Messa...	E-mail	'Bicycle of...	12/3/2006 9:06:43 PM	12/7/2006 6:39:47 PM	N/A	3,456	3,456	
Message0003	C:\Users\Administrator\Desktop\All Docs of Des...			E-mail Messa...	E-mail	'FW: anot...	12/7/2006 6:39:59 PM	12/7/2006 6:38:58 PM	N/A	1,990	1,990	
Message0003	C:\Users\Administrator\Desktop\All Docs of Des...			E-mail Messa...	E-mail	'RE: Bike ...	12/3/2006 9:16:48 PM	12/7/2006 6:39:12 PM	N/A	4,026	4,026	
Message0003	C:\Users\Administrator\Desktop\All Docs of Des...			E-mail Messa...	E-mail	'FW: Bike ...	12/7/2006 6:39:51 PM	12/7/2006 6:39:51 PM	N/A	1,458	1,458	
Message0004	C:\Users\Administrator\Desktop\All Docs of Des...			E-mail Messa...	E-mail	'Re: Bicycl...	12/3/2006 9:16:46 PM	12/7/2006 6:39:19 PM	N/A	4,626	4,626	
Message0004	C:\Users\Administrator\Desktop\All Docs of Des...			E-mail Messa...	E-mail	'FW: Bicyc...	12/7/2006 6:39:43 PM	12/7/2006 6:39:43 PM	N/A	1,370	1,370	
Message0005	C:\Users\Administrator\Desktop\All Docs of Des...			E-mail Messa...	E-mail	'Re: Bicycl...	12/3/2006 10:04:32...	12/7/2006 6:38:35 PM	N/A	4,636	4,636	
Message0005	C:\Users\Administrator\Desktop\All Docs of Des...			E-mail Messa...	E-mail	'FW: Req...	12/7/2006 6:39:32 PM	12/7/2006 6:39:32 PM	N/A	1,802	1,802	
Message0006	C:\Users\Administrator\Desktop\All Docs of Des...			E-mail Messa...	E-mail	'RE: Bike ...	12/3/2006 10:04:33...	12/7/2006 6:38:25 PM	N/A	4,094	4,094	
Message0006	C:\Users\Administrator\Desktop\All Docs of Des...			E-mail Messa...	E-mail	'FW: Bike ...	12/7/2006 6:39:06 PM	12/7/2006 6:39:06 PM	N/A	2,662	2,662	
Message0007	C:\Users\Administrator\Desktop\All Docs of Des...			E-mail Messa...	E-mail	'Re: Bicycl...	12/4/2006 9:38:44 AM	12/7/2006 6:38:17 PM	N/A	5,898	5,898	
Message0007	C:\Users\Administrator\Desktop\All Docs of Des...			E-mail Messa...	E-mail	'FW: Activ...	12/7/2006 6:38:44 PM	12/7/2006 6:38:44 PM	N/A	3,928	3,928	

For email recovery follow following steps:

Click the **E-Mail** tab. In the tree view, click to expand all folders, and then click the **Deleted Items** folder.

File Name	Full Path	Recycle Bi...	Ext	File Type	Category	Subject	Cr Date	Mod Date
Message0001	C:\Users\Administrator\Desktop\All Docs of Des...			E-mail Messa...	E-mail	'RE: Bike ...	12/3/2006 10:05:51...	12/3/2006
Message0002	C:\Users\Administrator\Desktop\All Docs of Des...			E-mail Messa...	E-mail	'FW: probl...	12/7/2006 6:39:22 PM	12/7/2006
Message0003	C:\Users\Administrator\Desktop\All Docs of Des...			E-mail Messa...	E-mail	'FW: anot...	12/7/2006 6:38:58 PM	12/7/2006

Select any message say Message0001 right click and select option LaunchDetached Viewer and you can see detail of deleted message.

The screenshot shows the AccessData FTK interface. The left pane displays a tree view of email folders under 'PRAC07 Jim_shu's.pst'. The 'Deleted Items' folder is expanded, showing three messages: Message0001, Message0002, and Message0003. The message details for Message0001 are shown in the center pane, including the subject 'RE: Bike spec's', from 'Jim Shu', date '12/3/2006 10:07:00 PM', and to '5amspade@myway.com'. The message body contains text about changing file extensions and needing money. The right pane shows a preview of the message body and its attachments, including 'AC19.gpj'. The status bar at the bottom indicates '3 Listed' and '4 Checked Total'.

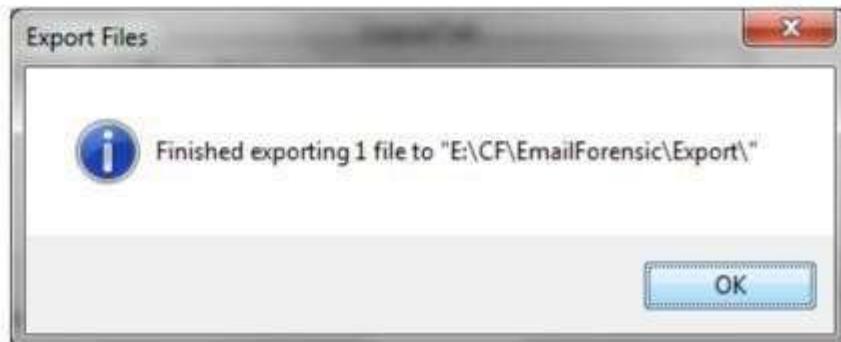
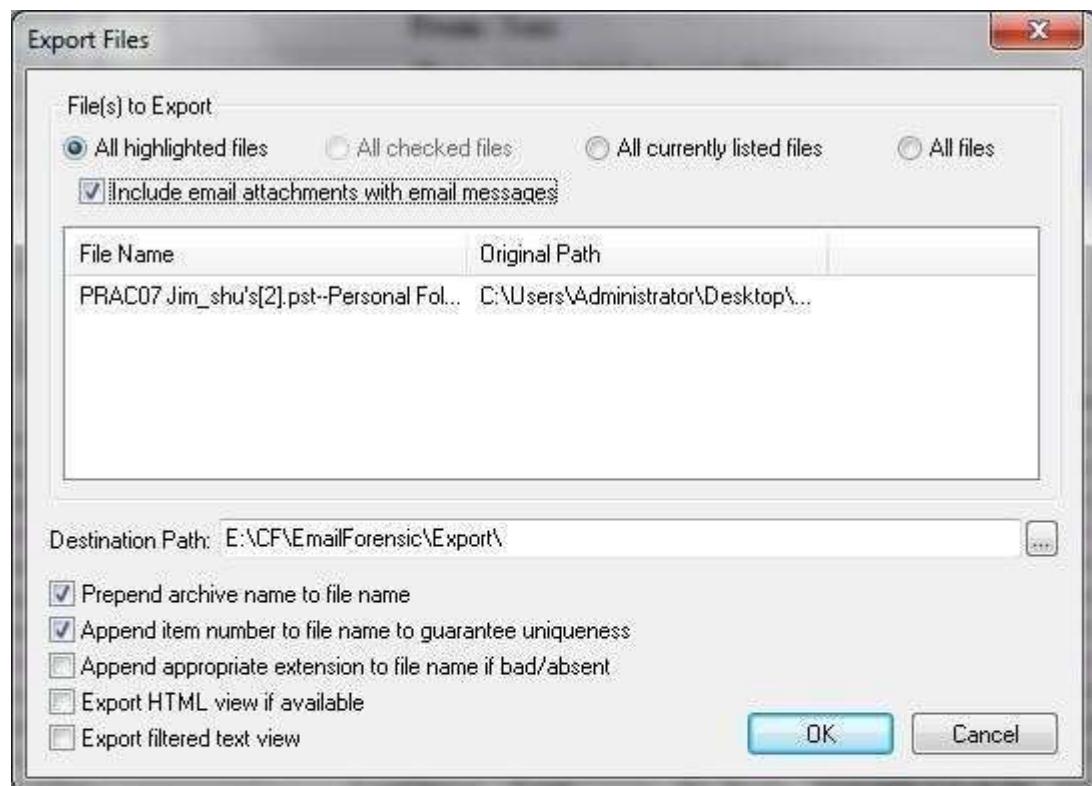
For analyzing header follow following steps:

Click the **E-Mail** tab. In the tree view, click to expand all folders, andthen click the **Inbox** folder.

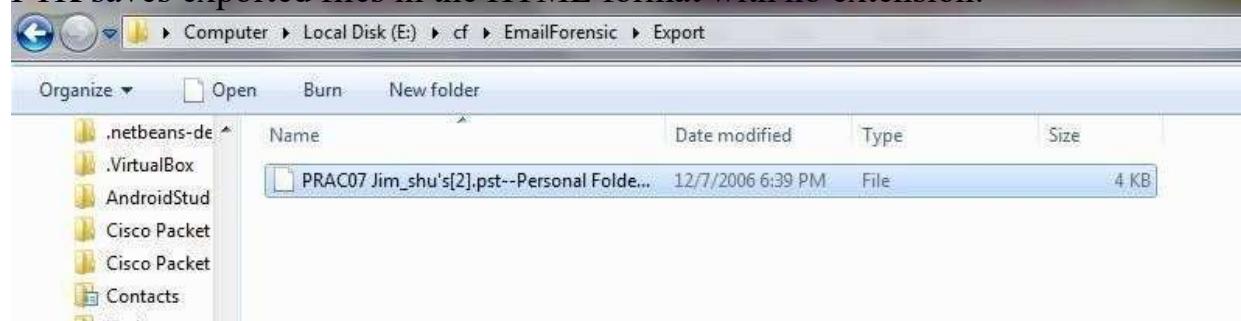
In the File List pane at the upper right, click Message0003; as shown inthe pane at the bottom, it's from **Sam** and is addressed to**Jim_shu@comcast.net**.

The screenshot shows the AccessData FTK interface with the 'E-Mail' tab selected. The left pane displays a tree view of email folders under 'PRAC07 Jim_shu's.pst'. The 'Inbox' folder is expanded, showing ten messages. The message details for Message0003 are shown in the center pane, including the subject 'RE: Bike spec's', from 'Sam', date '12/3/2006 9:14:02 PM', and to 'Jim_shu@comcast.net'. The message body contains text about potential investors. The right pane shows a preview of the message body. The status bar at the bottom indicates '10 Listed' and '5 Checked Total'.

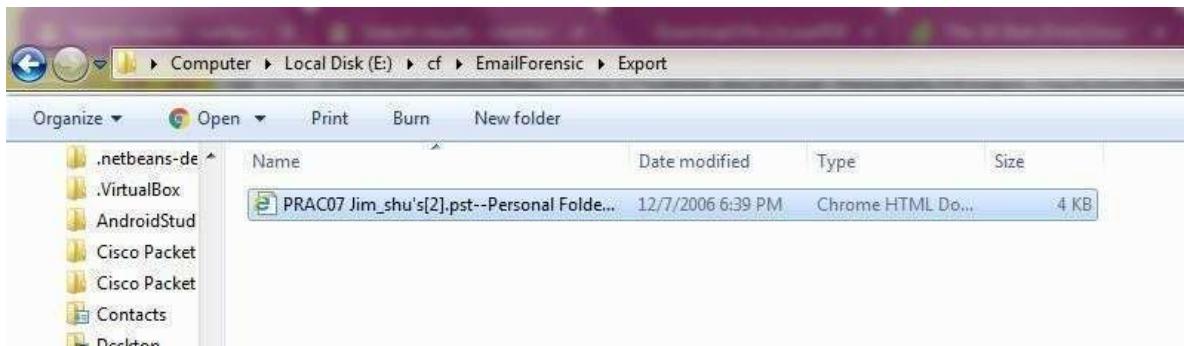
Right-click on any message say Message0003 in the File List pane and click Export File. In the Export Files dialog box, click OK.



FTK saves exported files in the HTML format with no extension.



Right-click the Message0003 file and click Rename. TypeMessage0003.html and press Enter.



Double-click Message0003.html to view it in a Web browser.



Conversation Topic: Bike spec's Sender Name: Sam Received By: Jim Shu Delivery Time: 12/3/2006 9:14:02 PM Creation Time: 12/3/2006 9:16:48 PM Modification Time: 12/7/2006 6:39:12 PM Submit Time: 12/3/2006 9:14:14 PM Flags: 1=Read Size: 6456 Received: from myway.com (ml.exciternetwork.com[207.159.120.55](untrusted sender)) by alnrmxc23.comcast.net (alnrmxc23) with ESMTP id <20061204021402a230090t3e>; Mon, 4 Dec 2006 02:14:02 +0000 X-Originating-IP: [207.159.120.55] Received: by mprdmixin.myway.com (Postfix, from userid 110) id 63B6067669; Sun, 3 Dec 2006 21:14:14 -0500 (EST) To: Jim_shu@comcast.net Subject: RE: Bike spec's Received: from [24.18.24.250] by mprdmailfe3.nwk.myway.com via HTTP; Sun, 03 Dec 2006 21:14:14 EST X-AntiAbuse: This header was added to track abuse, please include it with any abuse report X-AntiAbuse: ID = f869dfbea97fe07b9eab2f865d19b540 Reply-to: Samspade@myway.com From: "Sam" <Samspade@myway.com> MIME-Version: 1.0 X-Sender: Samspade@myway.com X-Mailer: PHP Content-Type: text/plain; charset="US-ASCII" Content-Transfer-Encoding: 7bit Message-Id: <2006120402141463B6067669@mprdmixin.myway.com> Date: Sun, 3 Dec 2006 21:14:14 -0500 (EST) We might be able to go \$4000 if it is good. Is it? Sam --- On Sun 12/03, Jim Shu <Jim_shu@comcast.net> wrote: From: Jim Shu [mailto: Jim_shu@comcast.net] To: Samspade@myway.com Date: Sun, 3 Dec 2006 18:09:06 -0800 Subject: RE: Bike spec's How much are you willing to pay me to get these plans to you? Jim-----Original Message-----From: Sam [mailto: Samspade@myway.com] Sent: Sunday, December 03, 2006 5:40 PM To: Jim_shu@comcast.net Subject: Bike spec's Do you have them yet? I've got people in Asia ready to duplicate them? Sam _____ No banners. No pop-ups. No kidding. Make My Way your home on the Web - http://www.myway.com _____ No banners. No pop-ups. No kidding. Make My Way your home on the Web - http://www.myway.com

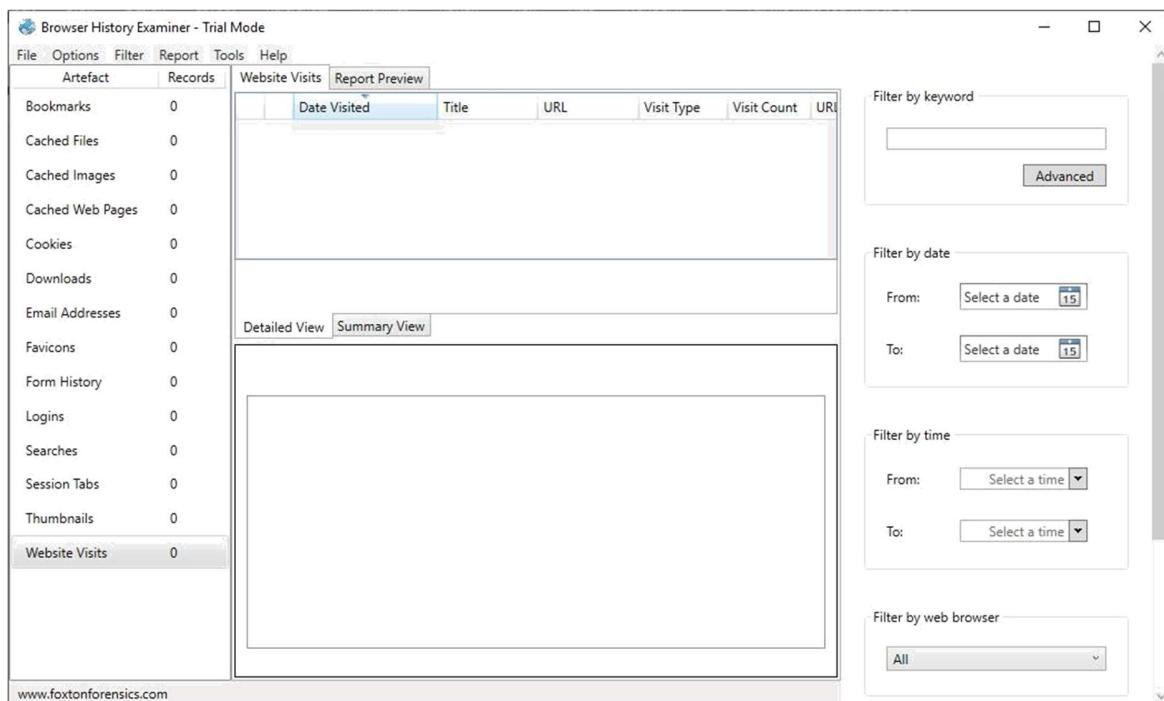
PRACTICAL - 9

Aim: Web Browser Forensics .

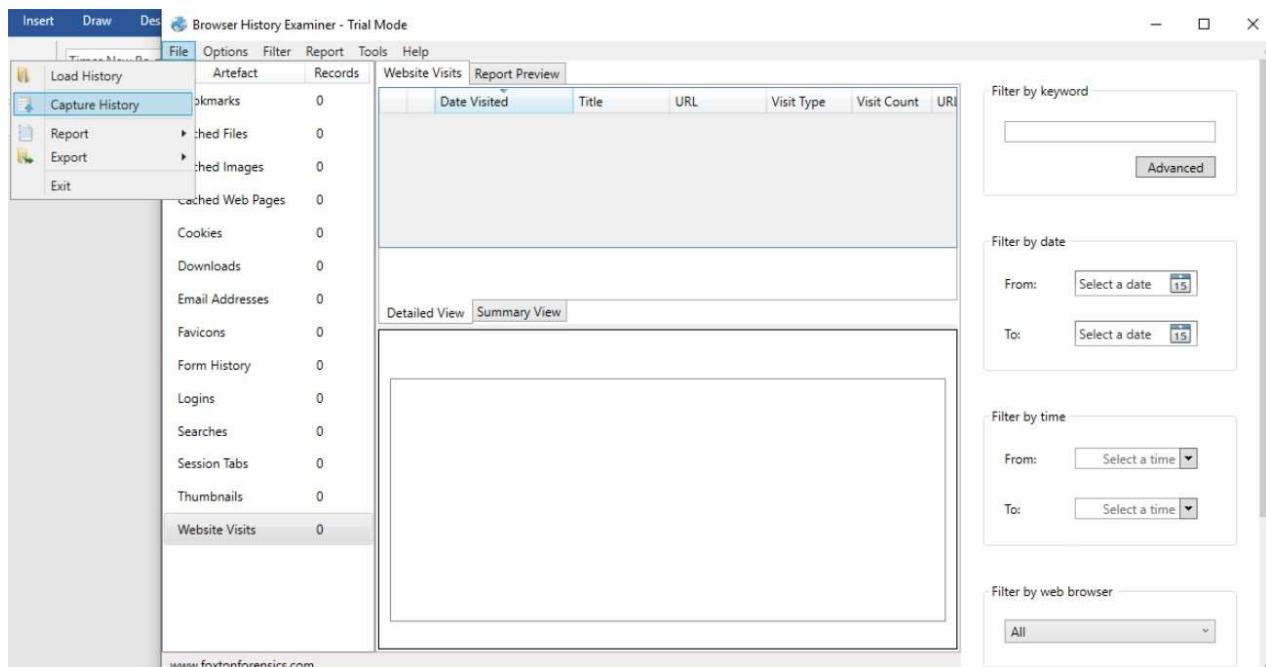
- Web Browser working
- Forensics activities on browser
- Cache / Cookies analysis
- Last Internet activity

Steps:

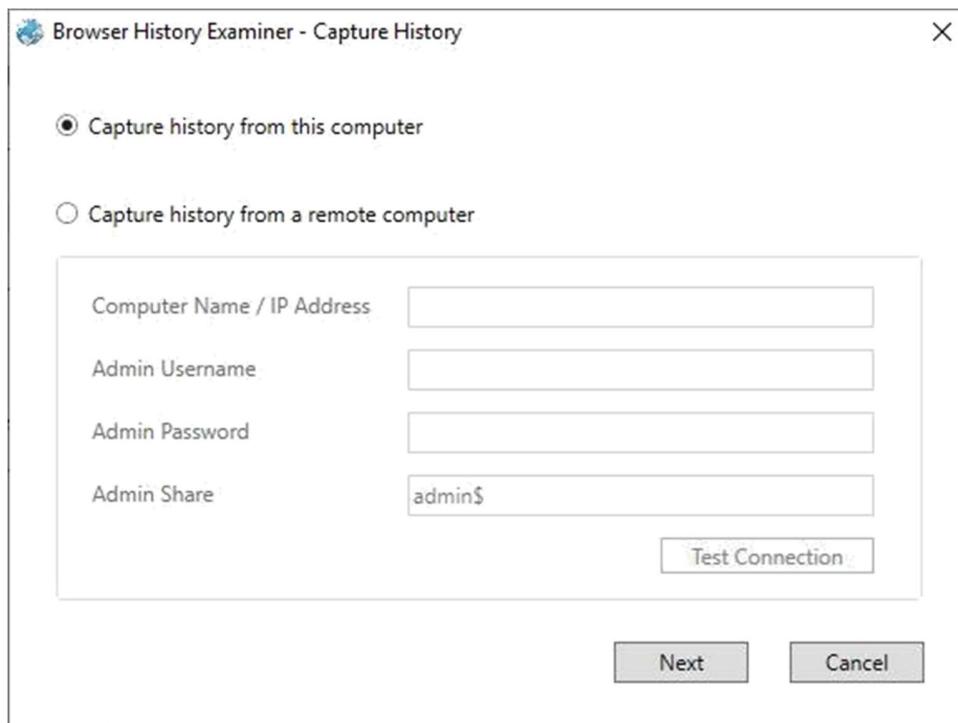
1. Open BrowserHistoryExaminer.



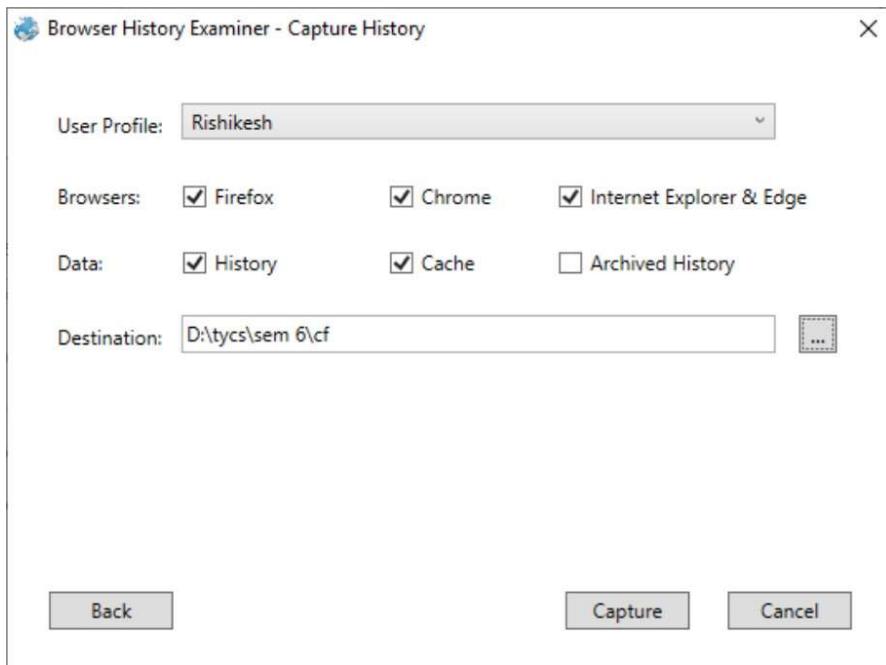
2. Click on file > Capture History



3. Select the capture folder and click on next.



4. Enter the destination to capture the data.



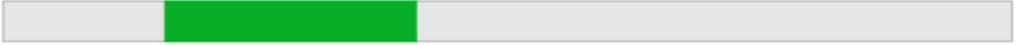
5. The History is been extracting.

Browser History Examiner - Capturing Data X

Capturing data from the local machine...

Browser History Examiner - Extracting Data X

Initialising...
Extracting Firefox website history...
Extracting Chrome website history...
Extracting Internet Explorer website history...

A horizontal progress bar consisting of three segments: a light grey segment on the left, a dark green segment in the middle, and a light grey segment on the right.

6. The data has been retrieved.

The screenshot shows the 'Browser History Examiner - Trial Mode' application. The left panel lists various artifacts with their counts: Bookmarks (8), Cached Files (4615), Cached Images (177), Cached Web Pages (36), Cookies (1566), Downloads (80), Email Addresses (30), Favicons (1790), Form History (31), Logins (3), Searches (1184), Session Tabs (62), Thumbnails (12), and Website Visits (2688). The 'Website Visits' tab is selected, displaying a table with columns: Date Visited, Title, URL, Visit Type, and Visit Count. The table shows two entries for March 18, 2019. Below the table is a message: 'Viewing 25/25 records'. To the right of the table is a bar chart titled 'Website Visit Count - 17-03-2019 to 18-03-2019'. The chart has a single blue bar extending from the 12 mark on the y-axis to the 13 mark. The x-axis shows dates from 17-03-19 to 18-03-19. At the bottom of the window, it says 'Time zone: UTC, DST Enabled' and 'Date format: dd/mm/yyyy'.

7. On the left panel click on bookmarks.

The screenshot shows the 'Browser History Examiner - Trial Mode' application. The left panel lists various artifacts with their counts, identical to the previous screenshot. The 'Bookmarks' tab is selected, displaying a table with columns: Date Added, Last Modified, Title, URL, and Web Browser. The table lists eight bookmark entries. Below the table is a message: 'Viewing 8/8 records'. To the right of the table are four filter panels: 'Filter by keyword', 'Filter by date', 'Filter by time', and 'Filter by web browser'. The 'Filter by keyword' panel contains a text input field and an 'Advanced' button. The 'Filter by date' panel has 'From' and 'To' date selection fields, both set to 'Select a date 15'. The 'Filter by time' panel has 'From' and 'To' time selection fields, both set to 'Select a time'. The 'Filter by web browser' panel has a dropdown menu set to 'All'. At the bottom of the window, it says 'Time zone: UTC, DST Enabled' and 'Date format: dd/mm/yyyy'.

8. On the left panel click on cached files.

The screenshot shows the 'Browser History Examiner - Trial Mode' application interface. The menu bar includes File, Options, Filter, Report, Tools, and Help. The main window has tabs for Artefact, Records, Cached Files, and Report Preview, with 'Cached Files' selected. On the left, a sidebar lists various history types with their counts: Bookmarks (8), Cached Files (4615), Cached Images (177), Cached Web Pages (36), Cookies (1566), Downloads (80), Email Addresses (30), Favicons (1790), Form History (31), Logins (3), Searches (1184), Session Tabs (62), Thumbnails (12), and Website Visits (2688). The main pane displays a table of cached files with columns: Last Fetched, Content Type, UI, Fetch Count, and File Size (Bytes). The table contains 25 records, with the last few rows showing file sizes of 18820976, 3523651, 2097152, 2097152, 2097152, 2097152, 2097152, 2097152, 2097152, 2097152, 2097152, 2097152, 2097152, 2097152, 2097152, 2097152, 2097152, 2097152, 2097152, 2082608, 2070260, 2060551, 2051858, and 2044048. To the right, there are four filter panels: 'Filter by keyword' with a text input and 'Advanced' button; 'Filter by date' with 'From' and 'To' date pickers; 'Filter by time' with 'From' and 'To' time pickers; and 'Filter by web browser' with a dropdown menu set to 'All'. At the bottom, it says 'Viewing 25/25 records' with page navigation buttons, a 'Page size' dropdown set to 50, and status bars for 'Time zone: UTC, DST Enabled' and 'Date format: dd/mm/yyyy'.

9. On the left panel click on cached images.

Browser History Examiner - Trial Mode

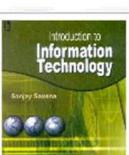
File Options Filter Report Tools Help

Artefact	Records
Bookmarks	8
Cached Files	4615
Cached Images	177
Cached Web Pages	36
Cookies	1566
Downloads	80
Email Addresses	30
Favicons	1790
Form History	31
Logins	3
Searches	1184
Session Tabs	62
Thumbnails	12
Website Visits	2688

Cached Images Report Preview

Last Fetched	Content Type	UI	Fetch Count	File Size (Bytes)	Web
image/jpeg	htl			1150328	Chrc ^
image/jpeg	htl			899593	Chrc
image/jpeg	htl	1		491093	Inter
image/png	htl			292601	Chrc
image/jpeg	htl			237697	Chrc
image/png	htl			165665	Chrc
image/jpeg	htl			129410	Chrc
image/jpeg	htl	2		128642	Firef
image/png	htl			103776	Chrc ^

Viewing 25/25 records < 1 > Page size 50





Filter by keyword Advanced

Filter by date From: To:

Filter by time From: To:

Filter by web browser All Date format dd/mm/yyyy

Time zone: UTC, DST Enabled

www.foxtonforensics.com

10. On the left panel click on cookies.

Browser History Examiner - Trial Mode

Artefact	Records
Bookmarks	8
Cached Files	4615
Cached Images	177
Cached Web Pages	36
Cookies	1566
Downloads	80
Email Addresses	30
Favicons	1790
Form History	31
Logins	3
Searches	1184
Session Tabs	62
thumbnails	12
Website Visits	2688

Report Preview

Date Created	UI	Last Accessed	Date Expires	N	C	W
17-03-2019 20:32:32	ac	17-03-2019 20:32:32	17-03-2019 20:32:42	Ch	Ch	^
17-03-2019 20:32:31	ac	17-03-2019 20:32:31	16-03-2021 20:32:31	GA	Ch	
17-03-2019 20:32:30	mi	17-03-2019 20:32:30	18-03-2019 20:32:30	GM	Ch	
17-03-2019 20:32:30	.gc	17-03-2019 20:32:30	16-09-2019 21:32:30	NI	Ch	
17-03-2019 20:05:48	mi	17-03-2019 20:05:48	27-03-2019 21:05:49	CC	Ch	
17-03-2019 20:05:26	mi	17-03-2019 20:17:27	27-03-2019 21:05:26	CC	Ch	
17-03-2019 20:05:24	.gc	17-03-2019 20:32:29	16-04-2019 21:05:24	1P	Ch	
17-03-2019 20:05:22	mi	17-03-2019 20:32:29	27-03-2019 21:05:23	CC	Ch	
17-03-2019 20:05:22	mi	17-03-2019 20:32:26		GM	Ch	
17-03-2019 20:05:21	.gc	17-03-2019 20:05:21	16-03-2021 20:05:21	AF	Ch	
17-03-2019 20:05:21	.gc	17-03-2019 20:05:21	16-03-2021 20:05:21	HS	Ch	
17-03-2019 20:05:21	.gc	17-03-2019 20:05:21	16-09-2019 21:05:21	NI	Ch	
17-03-2019 20:05:21	.gc	17-03-2019 20:05:21	16-03-2021 20:05:21	SA	Ch	
17-03-2019 20:05:21	.gc	17-03-2019 20:05:21	16-03-2021 20:05:21	SIT	Ch	
17-03-2019 20:05:21	.gc	17-03-2019 20:05:21	16-03-2021 20:05:21	SS	Ch	
17-03-2019 20:04:46	wv	17-03-2019 20:04:46		AS	Ch	
17-03-2019 20:04:46	.fo	17-03-2019 20:04:46	17-03-2019 20:05:46	_g	Ch	
17-03-2019 20:04:46	.fo	17-03-2019 20:04:53	16-03-2021 20:04:53	_g	Ch	
17-03-2019 20:04:46	.fo	17-03-2019 20:04:53	18-03-2019 20:04:53	_g	Ch	
17-03-2019 20:04:42	.gc	17-03-2019 20:04:42	16-09-2019 21:04:42	SH	Ch	
17-03-2019 09:04:09	wv	17-03-2019 09:04:09	17-03-2019 09:14:09	DV	US	Fir
17-03-2019 09:03:37	.ac	17-03-2019 09:16:27	10-04-2020 10:03:37	ou	5c	Fir
17-03-2019 09:03:37	.ac	17-03-2019 09:16:27	10-04-2020 10:03:37	di	aU	Fir

Viewing 25/25 records < | 1 | > of 1 pages Page size: 50

Time zone: UTC, DST Enabled Date format: dd/mm/yyyy

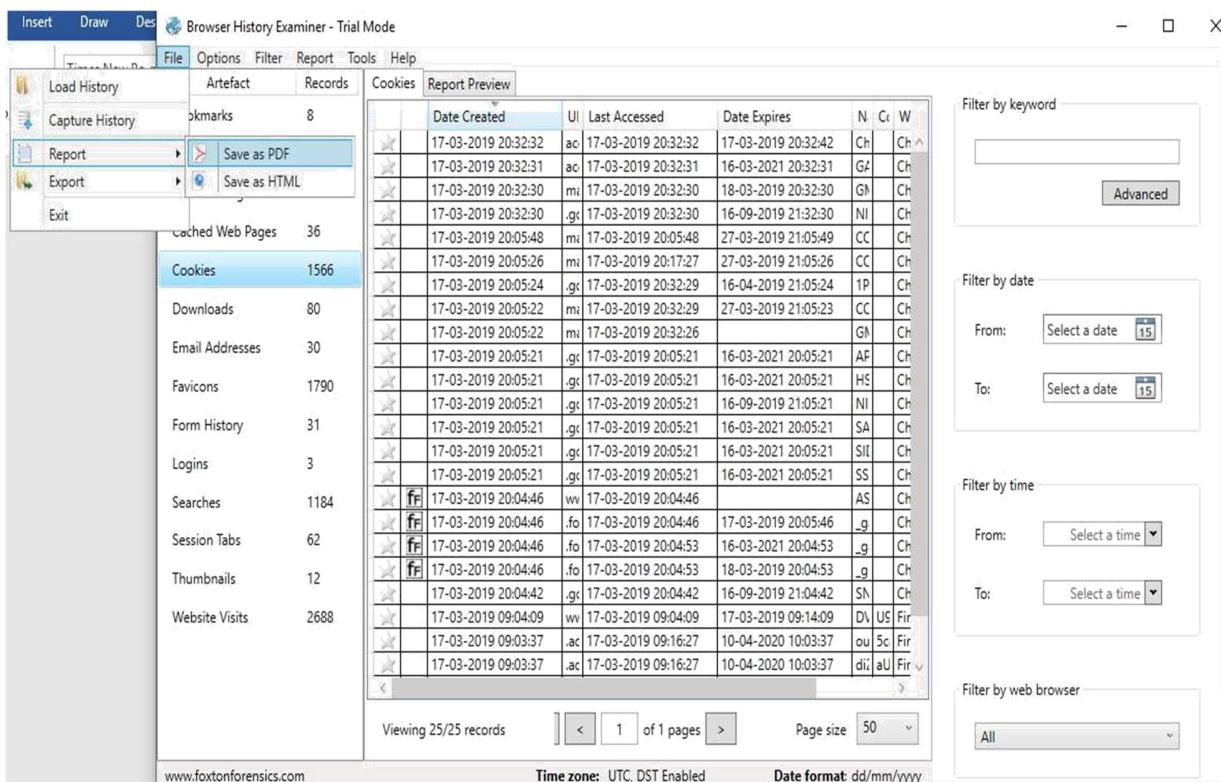
Filter by keyword: _____ Advanced

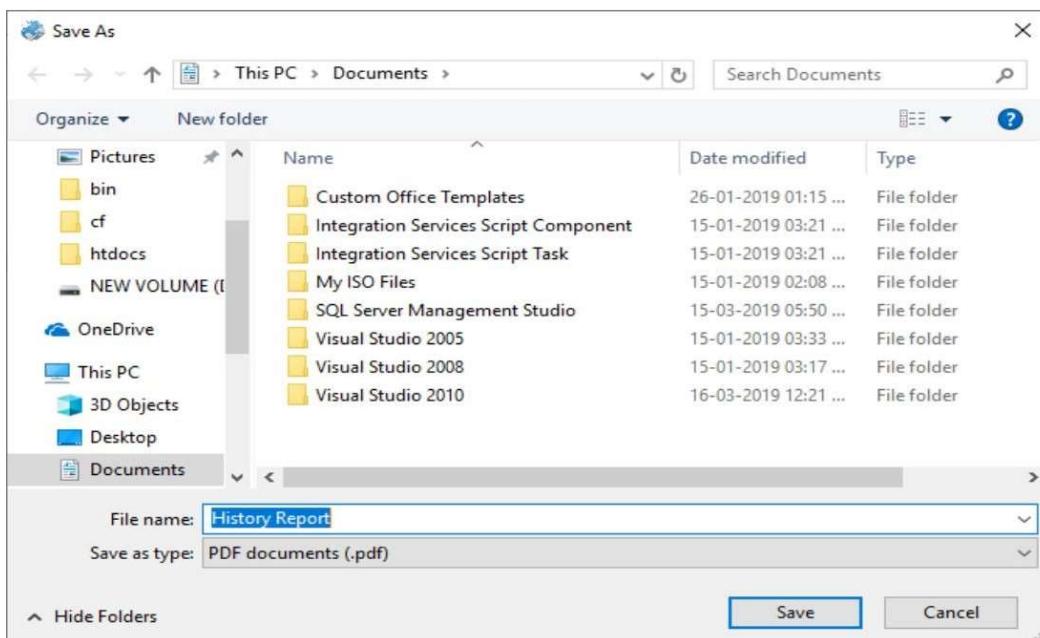
Filter by date: From: Select a date [15] To: Select a date [15]

Filter by time: From: Select a time [] To: Select a time []

Filter by web browser: All

11. To Create Reports. Click on file > Report and save the report as pdf or html page.





Web Browser History Report

Created: 18-03-2019 09:36
 Created using: Browser History Examiner v1.9
 Time zone: UTC, DST Enabled
 Date format: dd/mm/yyyy

Bookmarks

Date Added	Last Modified	Title	URL	Web Browser
17-03-2019 09:03:01	17-03-2019 09:03:01	Getting Started	https://www.mozilla.org/en-US/firefox/central/	Firefox
17-03-2019 09:03:01	17-03-2019 09:03:01	Help and Tutorials	https://support.mozilla.org/en-US/products/firefox	Firefox
17-03-2019 09:03:01	17-03-2019 09:03:01	Customize Firefox	https://support.mozilla.org/en-US/kb/customize-firefox-controls-buttons-and-toolbars?utm_source=fire...	Firefox
17-03-2019 09:03:01	17-03-2019 09:03:01	Get Involved	https://www.mozilla.org/en-US/contribute/	Firefox
17-03-2019 09:03:01	17-03-2019 09:03:01	About Us	https://www.mozilla.org/en-US/about/	Firefox
14-03-2019 05:01:05		New Tab	chrome://newtab/	Chrome
22-01-2019 06:40:50		Download Microsoft® SQL Server® 2012 Express from Official Microsoft Download Center	https://www.microsoft.com/en-us/download/confirmation.aspx?id=29062	Chrome
		Bing	http://go.microsoft.com/fwlink/p/?LinkId=255142	Internet Explorer

Cached Files

Last Fetched	Content Type	URL	Fetch Count	File Size (Bytes)	Web Browser
		https://mail.attachment.googleusercontent.com/attachment/u/0/?ui=2&ik=5c151dfa36&attid=0.1&a...	18820976		Chrome
	application/zip	https://r4-sn-4p8xoxu-cvhe.gvt1.com/edged/widenvine-cdm/4.10.1146.0-win-x64.zip?cmr_redirect=yes&a...	1	3523651	Firefox
		https://r3-sn-4p8xoxu-cvhe.googlevideo.com/videoplayback?ei=uYLNNeQ4ei1Abh7K4Cg&dur=152.733...		2097152	Chrome
		https://r3-sn-4p8xoxu-cvhe.googlevideo.com/videoplayback?ei=uYLNNeQ4ei1Abh7K4Cg&dur=152.733...		2097152	Chrome
		https://r3-sn-4p8xoxu-cvhe.googlevideo.com/videoplayback?ei=uYLNNeQ4ei1Abh7K4Cg&dur=152.733...		2097152	Chrome

Browser History Examiner - Trial Mode

File Options Filter Report Tools Help

Artefact	Records	Bookmarks	Report Preview
Load History	Bookmarks 8		
Capture History		Date Added	Last Modified
Report	Attached Files 4615	17-03-2019 09:03:01	17-03-2019 09:03:01
Export	Export to Excel	3-2019 09:03:01	17-03-2019 09:03:01
	Export to HTML	3-2019 09:03:01	17-03-2019 09:03:01
	Export to CSV	3-2019 09:03:01	17-03-2019 09:03:01
	Export to XML	3-2019 05:01:05	
	Export to Concordance Load File	1-2019 06:40:50	Download Microsoft® SQL Server® 2012 Express from Official Microsoft Download Center
Downloads	80		Bing
Email Addresses	30		
Favicons	1790		
Form History	31		
Logins	3		
Searches	1184		
Session Tabs	62		
Thumbnails	12		
Website Visits	2688		

Viewing 8/8 records | < | 1 of 1 pages | > | Page size: 50 |

Filter by keyword: Advanced

Filter by date: From: Select a date [15] To: Select a date [15]

Filter by time: From: Select a time To: Select a time

Filter by web browser: All

Filter | Undo | Clear

86 of 86 2652 words | www.foxtonforensics.com | Time zone: UTC, DST Enabled | Date format: dd/mm/yyyy

Web Browser History Report

Created: 18-03-2019 09:40
 Created using: Browser History Examiner v1.9
 Time zone: UTC, DST Enabled
 Date format: dd/mm/yyyy

Bookmarks

Date Added	Last Modified	Title	URL	Web Browser
17-03-2019 09:03:01	17-03-2019 09:03:01	Getting Started	https://www.mozilla.org/en-US/firefox/central/	Firefox
17-03-2019 09:03:01	17-03-2019 09:03:01	Help and Tutorials	https://support.mozilla.org/en-US/products/firefox	Firefox
17-03-2019 09:03:01	17-03-2019 09:03:01	Customize Firefox	https://support.mozilla.org/en-US/kb/customize-firefox-controls-buttons-and-toolbars?utm_source=fire...	Firefox
17-03-2019 09:03:01	17-03-2019 09:03:01	Get Involved	https://www.mozilla.org/en-US/contribute/	Firefox
17-03-2019 09:03:01	17-03-2019 09:03:01	About Us	https://www.mozilla.org/en-US/about/	Firefox
14-03-2019 05:01:05		New Tab	chrome://newtab/	Chrome
22-01-2019 06:40:50		Download Microsoft® SQL Server® 2012 Express from Official Microsoft Download Center	https://www.microsoft.com/en-us/download/confirmation.aspx?id=29062	Chrome
		Bing	http://go.microsoft.com/fwlink/p/?LinkId=255142	Internet Explorer

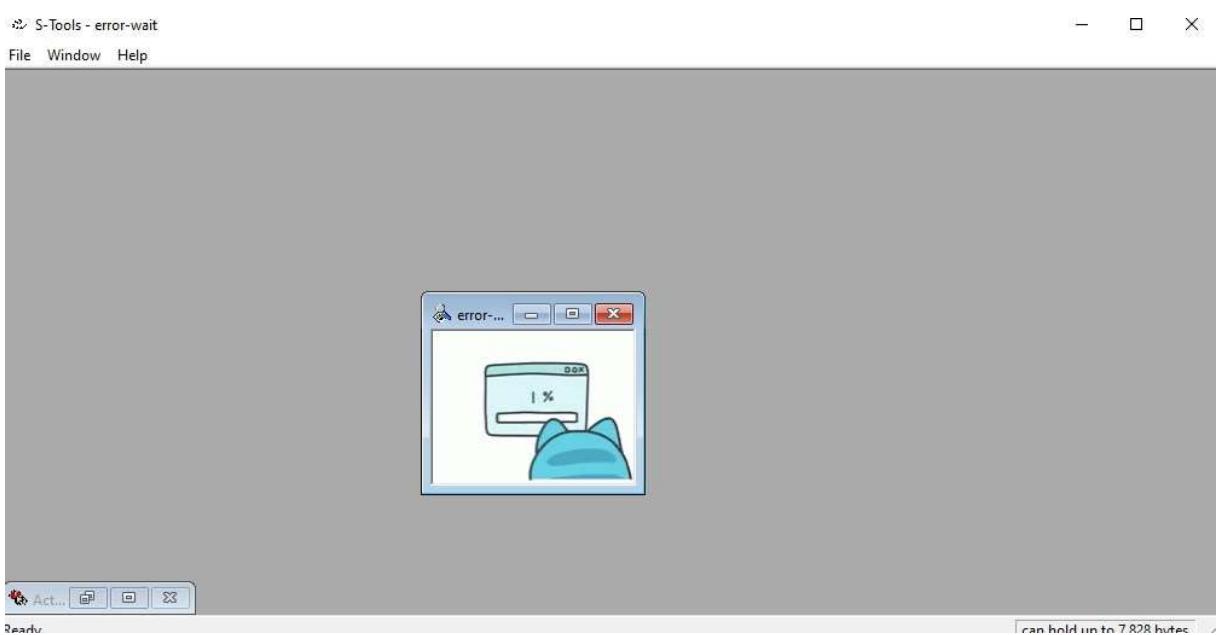
PRACTICAL - 10

A. Using Steganography Tools [S-Tools]

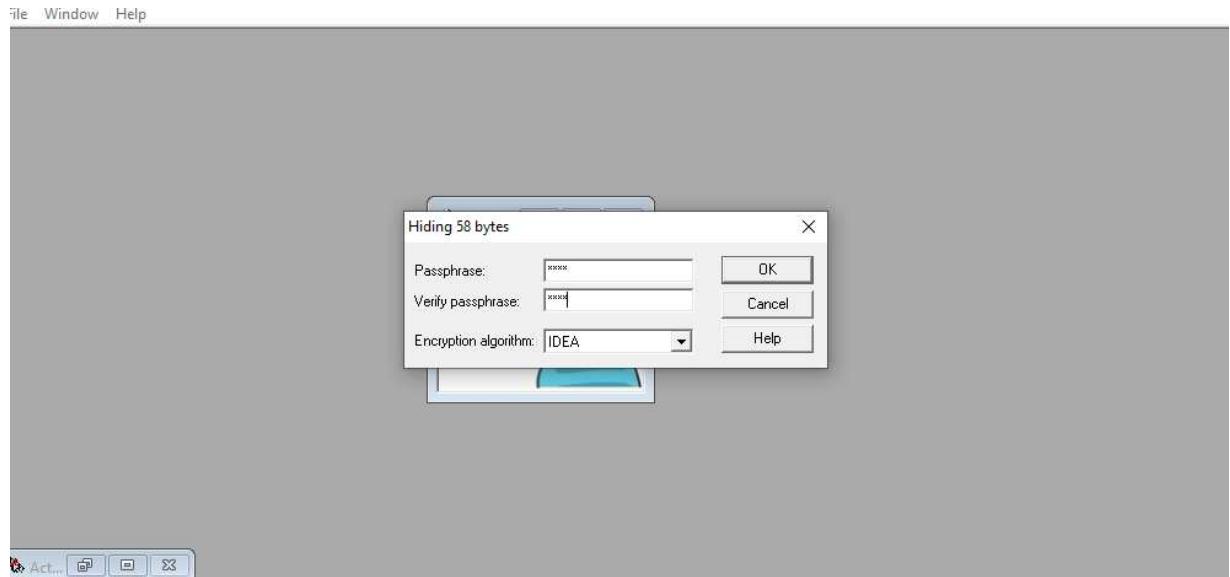
Step 1: The dark gray background area is used for dropping the image or sound files into.



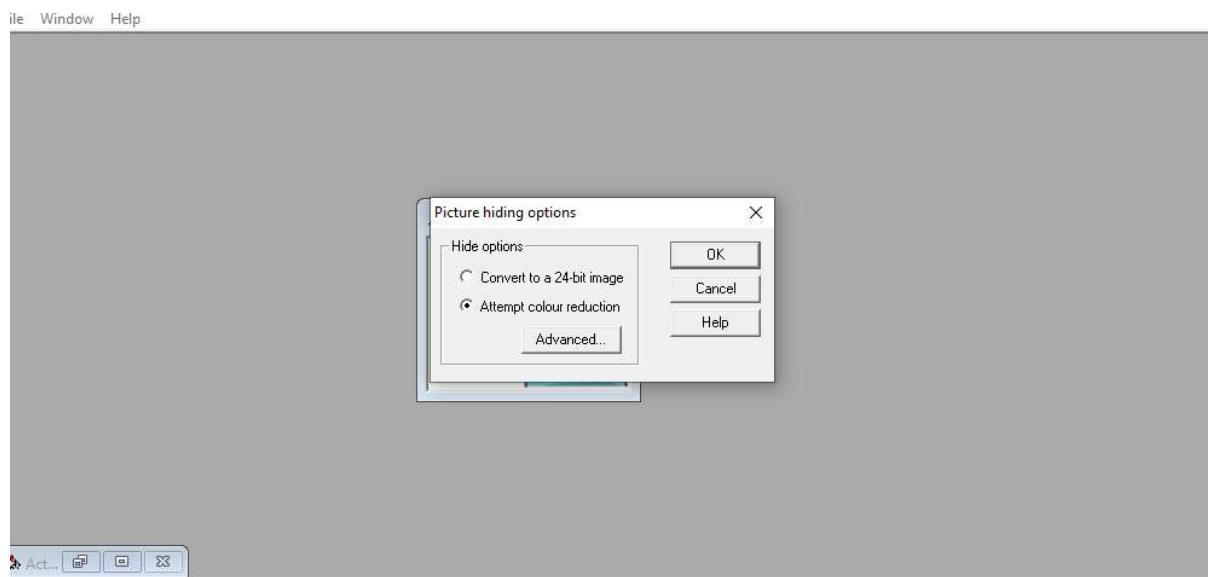
Step 2: After opening S-Tools (S-Tools.exe) and Windows Explorer, drag Cover File.gif into the main working area of S-Tools



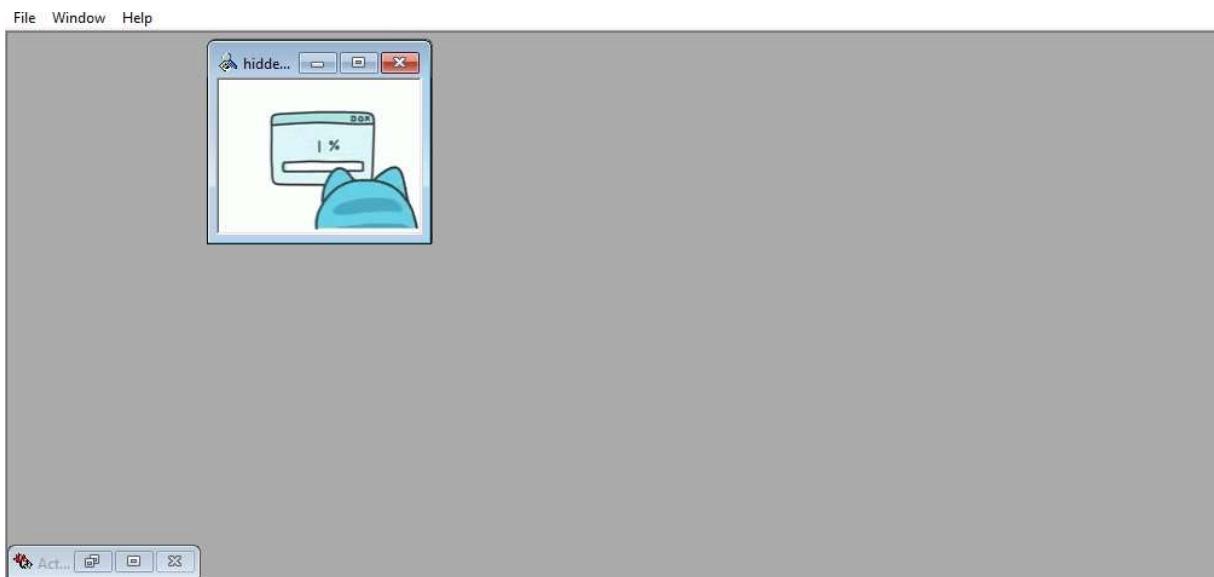
Step 3: The pass phrase is used in generating the pseudorandom number that is used to insert the bits into the cover file. S-Tools gives a choice between IDEA, DES, TripleDES, and MDC encryption algorithms



Step 4: To have S-Tools process the GIF file, a dialog box prompts for choices to be made



Step 5 : When S-Tools finishes inserting the data, the output image will display with the marker "hidden data" at the top



B. Using Whitespace Stegnography tool SNOW

A screenshot of a Notepad window titled 'cbd - Notepad'. The menu bar includes 'File', 'Edit', 'Format', 'View', and 'Help'. The text 'Practical no. 5' is typed into the main area. The status bar at the bottom shows the path 'C:\Users\DELL\Downloads\drive-download-20220707T100455Z-00' and the drive letter 'C:\'. The right side of the window is heavily redacted with black.

```
snow -C -m "MessageToHide" -p "Password" "InputTextFile" "OutputTextFile"
```

```
C:\Users\DELL\Downloads\drive-download-20220707T100455Z-001\SNOW.EXE
C:\CF Pracs\practical 5>snow -C -m "Practical number 5 " -p "kaushal" cbd.txt efg.txt
Compressed by 35.67%
Message exceeded available space by approximately 788.54%
An extra 4 lines were added.
```

practical 5				
	Share	View		
	This PC	> Local Disk (C:)	> CF Pracs	> practical 5
Name	Date modified	Type	Size	
cbd	7/19/2022 3:47 PM	Text Document	1 KB	
efg	7/19/2022 3:59 PM	Text Document	1 KB	

```
C:\Users\DELL\Downloads\drive-download-20220707T100455Z-001\SNOW.EXE
C:\CF Pracs\practical 5>snow -C -p "kaushal" efg.txt
Practical number 5
C:\CF Pracs\practical>
C:\CF Pracs\practical>
```