1.)File System Explorer: Extend the file system explorer project by incorporating advanced features such as file compression, encryption, and permissions. Discuss the significance of these features in real-world file systems and how they contribute to data security and integrity.

import os

import gzip

from cryptography.fernet import Fernet

import stat

def explore\_directory(path):

try:

# List all files and directories in the given path

contents = os.listdir(path)

print("Contents of", path, ":")

for item in contents:

print(item)

except OSError as e:

print("Error:", e)

def compress\_file(source\_file, compressed\_file):

try:

with open(source\_file, 'rb') as f\_in:

with gzip.open(compressed\_file, 'wb') as f\_out:

f\_out.writelines(f\_in)

print("File compressed successfully.")

except FileNotFoundError:

print("Error: Source file not found.")

def encrypt\_file(input\_file, encrypted\_file):

try:

key = Fernet.generate\_key()

cipher = Fernet(key)

with open(input\_file, 'rb') as f\_in:

data = f\_in.read()

encrypted\_data = cipher.encrypt(data)

with open(encrypted\_file, 'wb') as f\_out:

f\_out.write(encrypted\_data)

print("File encrypted successfully.")

except FileNotFoundError:

print("Error: Source file not found.")

def set\_file\_permissions(file\_path, permission):

try:

os.chmod(file\_path, permission)

print("File permissions set successfully.")

except FileNotFoundError:

print("Error: File not found.")

if \_name\_ == "\_main\_":

directory\_path = "C:\\Program Files (x86)"

source\_file = "example.txt"

compressed\_file = "example.txt.gz"

encrypted\_file = "example\_encrypted.txt"

permission = stat.S\_IRUSR | stat.S\_IWUSR | stat.S\_IRGRP | stat.S\_IROTH # Example permission (read-only for owner, read-only for group, read-only for others)

explore\_directory(directory\_path)

compress\_file(source\_file, compressed\_file)

encrypt\_file(source\_file, encrypted\_file)

set\_file\_permissions(source\_file, permission)