**SE EXTC (2020 - 2021) Python project report**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project member details**   |  |  | | --- | --- | | Name: Priti Pore | Roll number: 19104A0028 | | Name: Tabassum Shaikh | Roll number: 19104A0070 | | Name: Preshit Pawar | Roll number: 19104A0043 | |
| **Project title:** File Locker System |
|  |
| **Abstract:** File locker system is provision to lock some of your personal or confidential files. It secures your personal data stored in files and disables any nth person to access without the password. This project enables the user to convert his files to password protected zip files. It also captures a picture using the webcam if incorrect password is being entered for n number of times. The program requires some libraries to be imported, namely pyminizip, zipfile, and cv2.  For future scope, we are planning to send the captured screenshot to a particular email address mentioned in program and also save the picture in some particular folder as per the user’s convenience instead of saving in the same folder. |
|  |
| **Introduction:** This Project is designed to lock your files using password which is to be generated by the using the library pyminizip. The maximum attempts of incorrect password that any user and make is limited to 3. On the 3rd incorrect attempt, the program captures a picture and saves it into the folder. For this project, we need to first install the required libraries from pip. |
|  |
| **Working:**  Code:  import pyminizip  inputt = "SBL\_MP.txt"  output = "save.zip"  password = "1234"  com\_lvl = 5  pyminizip.compress(inputt, None, output, password, com\_lvl)  from zipfile import ZipFile import cv2  file\_name = "save.zip"  with ZipFile(file\_name, 'r') as zip:   count = 0  c = None  while count < 3:  try:  string = input("Enter your password: ")  zip.extractall(pwd=bytes(string, 'utf-8'))  c = "successful"  if c == "successful":  print(c)  break   except:  a = print("You have entered a wrong password")  print(a)  if count == 2:  camera = cv2.VideoCapture(0)  return\_value, image = camera.read()  cv2.imwrite('FileLockImage' + '.png', image)  del camera   count += 1  Explanation:  The code initially imports the library pyminizip. For this project, a sample text file has been created for demonstration named as “SBL\_MP.txt”. This file is taken as input into the program.  The compression level is set to 5. The password created is to be mentioned. Using the function pyminizip.compress(), the text file will be convert to a zip file with the new name that is specified in the variable ‘output’, i.e. “save.zip”. This function takes in five arguments as follows: -  1. source file path (string) 2. source file prefix path (string) or None (path to prepend to file) 3. destination file path (string) 4. password (string) or None (to create no-password zip) 5. compress\_level(int) between 1 to 9, 1 (more fast) <---> 9 (more compress) or 0 (default)  This function always returns None.  Furthermore, two more libraries namely, zipfile and cv2 have been imported. To set the maximum incorrect attempts to 3, a counter variable named “count” have been initialised with 0. Another variable ‘c’ is declared with default value ‘None’ to specify the status of the program. While count is less than 3, the program asks the user to enter a password to open the file. It extracts the zip file using the function zip.extractall() and passes the password into it. If the password is correct, it prints “successful” and breaks out of the loop. If not, it prints “You have entered a wrong password”; the counter in incremented and the loop continues till c==2. As soon as the user makes 3 incorrect attempts to the password, the value of ‘count’ becomes 2 and the program executes the function cv2.VideoCapture() which turns the camera on. Then the function cv2.imwrite() is executed by which the webcam captures picture of the user or whatever is in front of camera. This picture is saved into the save folder as that of the program file and the sample text file. |
|  |
| **Output:**  For correct password:    For 3 incorrect passwords:    Zip file created and Image saved in folder (for wrong password):    File becomes password protected: |
|  |
| **Conclusion:**  The project has successfully protected our personal files with password and can capture the picture when incorrect password is entered. |