Password Security /Authentication Using Facial Recognition

Group 14

Guided by:Dr.Shahana Gajala Qureshi

Group Members

- Ben Tom Abey 21BCY10035
- Bennet Binu 21BCY10085
- Bharath V Menon 21BCY10057
- Praise E Mathew 21BCY10193

Index

Introduction

- Online security has been a major concern since the time when the Internet became a necessity for the society, from business activities to everyday life of ordinary people. A fundamental aspect of online security is to protect data from unauthorized access.
- The most commonly used method for doing this is to use password as part of the online access process. Password is a secret character string only the user knows and its hashed code is stored on the server that provides access to the data. When the user requests for data access, he enters the password and gets the data.
- Password has been a predominating approach for user authentication to gain access to restricted resources. The main issue with password is its quality or strength, i.e. how easy (or how hard) it can be “guessed” by a third person who wants to access the resource that you have access to by pretending being you. In this project, we review various metrics of password quality and compare their strengths and weaknesses as well as the relationships between these metrics.

Literature Review

- With the rapid beginning of national and international networks, the question of system security has become one of growing importance.
- High speed inter-machine communication have made the threats of system “crackers,” data theft, data corruption very real.
- Good password practices are critical to the security of any information system.
- End users often use weak passwords that are short, simple, and based on personal and meaningful information that can be easily guessed.
- A survey was conducted among executive MBA students who hold managerial positions and the results of the survey indicate that users practice insecure behaviors in the utilization of passwords.

Challenges

- **Brute Force/Cracking**

  A common way for attackers to access passwords is by brute forcing or cracking passwords. These methods use software or automated tools to generate billions of passwords and trying each one of them to access the user's account and data until the right password is discovered.

- **Weak Passwords**

  Since users have to create their own passwords, it is highly likely that they won't create a secure password. It might be because users want to have a password that's easy to remember, or they aren't up-to-date with password security best practices, or they use patterns to generate their passwords like using their name or birthdate in their passwords.

- **Reuse of Passwords and Use of Compromised Passwords**

  Often, users tend to use similar passwords across different networks and systems which makes their passwords vulnerable to hacking.

Motive behind password security systems

- **Most of our online accounts are protected by a username and password combination. These passwords protect the data that we store in our accounts, whether that is our bank details, our purchase history or our home address.**
- **If you have weak passwords, a hacker can gain access to your account. And if you use the same password on every website, you'll lose everything: your bank accounts, payment info, personal documents, and access to other sensitive accounts.**
- **To create a strong password, it's essential to know the most common password combinations. The word "password," number strings like "12345," certain words like "cat," and your name or a relative's name take only seconds to crack. Hackers commit break-ins so quickly that you won't know what happened until it's too late.**
- **Use special characters, lowercase letters, uppercase letters, and randomized strings of letters and numbers to create a secure password. A hacker could guess a common word, but they won't guess a password like "9a4Fd5hd67M90op@" that virtually no one else on the planet uses.**

Methodology

What Makes A Strong Password?

Generally speaking, the strength of a password is determined by three things: the length of the character set, the length of the password, and to a lesser extent, the variety in characters chosen.

With all the data breaches we see on the news these days, it can sometimes be scary to trust companies with your sensitive information. However, in many cases, sensitive data is gained by an attacker taking advantage of users implementing weak passwords.

How do the factors determine password strength?

- Although an optimal password consists of many different types of characters and is very lengthy, there, of course, needs to be a balance
- A common technique to help balance these two is to substitute letters for numbers/symbols that look like those letters.
- This is typically a good practice as long as the password is lengthy, but attackers also know about these techniques and will search for such passwords as well.
- Finally, tacking on a few extra numbers/symbols to the end of the password can go a long way to keep attackers at bay.

Here’s a chart that shows the relative strength of passwords, calculated against modern brute force attacks. It’s worth mentioning that as technology advances and brute-force abilities increase, these passwords become weaker.

Password Strength Chart

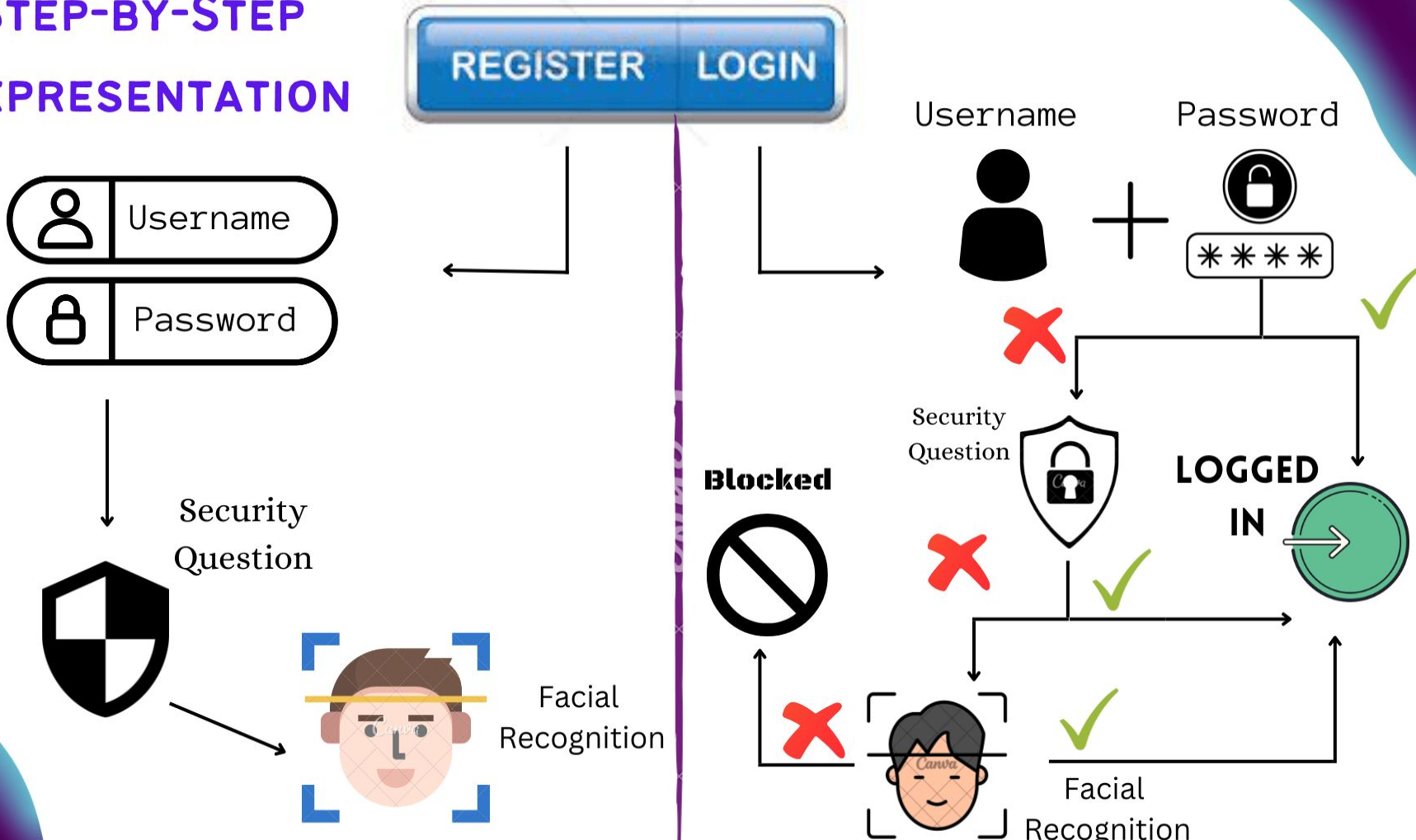
SANDSTORM IT power on

This is based on the average brute forcing (botnet) power in 2019.

123456 Top 10,000 password	0.20 milliseconds	Unsafe
qwerty123456 Longer “common” password	13 hours	Unsafe
ITFunSom3times Longer password with numbers	48 thousand years	Risky
ITifun0m3times! Longer password with numbers and special characters	13 trillion years	Good
imusingalongpasswordtoday Even Longer password	913 trillion years	Better
imu$inga1ongpa$$word+oday! Even Longer password with numbers and special characters	2 octillion years	Best

Please Note: These passwords are for demonstration purposes ONLY and are not to be used.

Step-by-Step Representation



Programing Languages

- HTML
- JSON
- PYTHON
- PYCHARM
- MYSQL

Design Goals

Our design goals for this specific functionality are relatively small.

- Provide visual feedback to the user regarding the strength of their password.
- The feedback has to be instantaneous. This means no clicking on a button to test the strength.
- We've chosen to change the background colors as well to draw the user's attention to this.
- Provide additional quantifiable feedback so the user knows in which departments the password lacks strength and how it can be improved.

Scope for future work

Passwords provide the first line of defense against unauthorized access to your computer and personal information. A vast number of Cyber crimes occurs due to accessing of password of an individual, without their knowledge. It is impossible to stop hackers from accessing passwords and getting data. But we can definitely make their path burdensome.The stronger your password, the more protected your computer will be from hackers and malicious software. Therefore the scope of this work is that we make it terrible for the hackers to decode passwords.

Codes:Password Strength

Passwordstrengthdetectionnew.py - C:\Users\Admin\Desktop\Passwordstrengthdetectionnew.py (3.11.1)

File Edit Format Run Options Window Help

```
print('''
NOTE:*Include numbers,special character,uppercase and lowercase
      inorder to make your password STRONG!!!*
      ''')
while True:
    password = input("Enter new password:")
    u='[@_!#$%^&*()<>?/\|}{~:]'

    result = {}

    if len(password) >= 8:
        result["lenght"] = True      #length
    else:
        result["lenght"] = False

    digit = False
    for i in password:
        if i.isdigit():          #digits
            digit = True

    result["digits"] = digit

    uppercase = False
    for i in password:
        if i.isupper():      #uppercase
            uppercase = True

    result["upper-case"] = uppercase

    special_character=False
    for i in password:           #splcharacter
        if i in u:
            special_character=True
    result["special_character"]=special_character

    print(result)
    print(result.values())

    if all(result.values()):
        print("Strong password")
        reenter=input("Re-enter Password:")      #result
```

Ln: 41 Col: 55

Type here to search 30°C ENG 14:06 18-01-2023

Codes:Password Strength

Passwordstrengthdetectionnew.py - C:\Users\Admin\Desktop\Passwordstrengthdetectionnew.py (3.11.1)

File Edit Format Run Options Window Help

```
        result["lenght"] = True      #length
    else:
        result["lenght"] = False

    digit = False
    for i in password:
        if i.isdigit():          #digits
            digit = True

    result["digits"] = digit

    uppercase = False
    for i in password:
        if i.isupper():      #uppercase
            uppercase = True

    result["upper-case"] = uppercase

    special_character=False
    for i in password:           #splcharacter
        if i in u:
            special_character=True
    result["special_character"]=special_character

    print(result)
    print(result.values())

    if all(result.values()):
        print("Strong password")
        reenter=input("Re-enter Password:")      #result
        if password==reenter:
            print("Password Saved")
            break
        else:
            print("Wrong Password,try again")
    else:
        print("Weak password")
        print('''
REMINDER:#Include numbers,special character,uppercase and lowercase in password#''')
```

Ln: 41 Col: 55

Type here to search 30°C ENG 14:06 18-01-2023

Output:Password Strenght

IDLE Shell 3.11.1

File Edit Shell Debug Options Window Help

```
Python 3.11.1 (tags/v3.11.1:a7a450f, Dec  6 2022, 19:58:39) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
======== RESTART: C:\Users\Admin\Desktop\Passwordstrengthdetectionnew.py =======

NOTE:*Include numbers,special character,uppercase and lowercase
      inorder to make your password STRONG!!!*

Enter new password:password
{'lenght': True, 'digits': False, 'upper-case': False, 'special_character': False}
dict_values([True, False, False, False])
Weak password

REMINDER:#Include numbers,special character,uppercase and lowercase in password#
Enter new password:password1
{'lenght': True, 'digits': True, 'upper-case': False, 'special_character': False}
dict_values([True, True, False, False])
Weak password

REMINDER:#Include numbers,special character,uppercase and lowercase in password#
Enter new password:Password1
{'lenght': True, 'digits': True, 'upper-case': True, 'special_character': False}
dict_values([True, True, True, False])
Weak password

REMINDER:#Include numbers,special character,uppercase and lowercase in password#
Enter new password:P@ssword1
{'lenght': True, 'digits': True, 'upper-case': True, 'special_character': True}
dict_values([True, True, True, True])
Strong password
Re-enter Password:password
Wrong Password,try again
Enter new password:P@ssword1
{'lenght': True, 'digits': True, 'upper-case': True, 'special_character': True}
dict_values([True, True, True, True])
Strong password
Re-enter Password:P@ssword1
Password Saved
>>>
```

Ln: 39 Col: 0

Type here to search 30°C ENG 14:06 18-01-2023

Face Recognition :Code

```python
import face_recognition
import os, sys
import cv2
import numpy as np
import math


def face_confidence(face_distance, face_match_threshold=0.6):
    range = (1.0 - face_match_threshold)
    linear_val = (1.0 - face_distance) / (range * 2.0)

    if face_distance > face_match_threshold:
        return str(round(linear_val * 100, 2)) + '%'
    else:
        value = (linear_val + ((1.0 - linear_val) * math.pow((linear_val - 0.5) * 2, 0.2))) * 100
        return str(round(value, 2)) + '%'


class FaceRecognition:
    face_locations = []
    face_encodings = []
    face_names = []
    known_face_encodings = []
    known_face_names = []
    process_current_frame = True

    def __init__(self):
        self.encode_faces()

    def encode_faces(self):
        for image in os.listdir('faces'):
            face_image = face_recognition.load_image_file(f"faces/{image}")
            face_encoding = face_recognition.face_encodings(face_image)[0]
```

File Edit View Navigate Code Refactor Run Tools VCS Window Help pythonProject - classpython.py

pythonProject 〉 classpython.py

classpython

notes.py | classpython.py | study.py | client.py | bharath.jpeg | elon musk.jpg

pythonProject C:\Use
- faces
  - bharath.jpeg
  - donald trump.jp
  - elon musk.jpg
- classpython.py
- client.py
- notes.py
- pywhatkit_dbs.txt
- sample.txt
- study.py

External Libraries
- < Python 3.9 > C:\U
  - Binary Skeletons
  - DLLs
  - Extended Defini

Structure
- face_confidence(face_
- FaceRecognition

```
            self.known_face_encodings.append(face_encoding)
            self.known_face_names.append(image)
        print(self.known_face_names)

    def run_recognition(self):
        video_capture = cv2.VideoCapture(0)

        if not video_capture.isOpened():
            sys.exit('Video source not found...')

        while True:
            ret, frame = video_capture.read()

            # Only process every other frame of video to save time
            if self.process_current_frame:
                # Resize frame of video to 1/4 size for faster face recognition processing
                small_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)

                # Convert the image from BGR color (which OpenCV uses) to RGB color (which face_recognition uses)
                rgb_small_frame = small_frame[:, :, ::-1]

                # Find all the faces and face encodings in the current frame of video
                self.face_locations = face_recognition.face_locations(rgb_small_frame)
                self.face_encodings = face_recognition.face_encodings(rgb_small_frame, self.face_locations)

                self.face_names = []
                for face_encoding in self.face_encodings:
                    # See if the face is a match for the known face(s)
                    matches = face_recognition.compare_faces(self.known_face_encodings, face_encoding)
                    name = "Unknown"
                    confidence = '???'

                    # Calculate the shortest distance to face
```

FaceRecognition 〉 encode_faces() 〉 for image in os.listdir('faces')

TODO | 6: Problems | Terminal | Python Console | Event Log

75 chars, 1 line break 33:75 CRLF UTF-8 4 spaces Python 3.9

File Edit View Navigate Code Refactor Run Tools VCS Window Help pythonProject - classpython.py

pythonProject › classpython.py

classpython

notes.py | classpython.py | study.py | client.py | bharath.jpeg | elon musk.jpg

```
                face_distances = face_recognition.face_distance(self.known_face_encodings, face_encoding)

                best_match_index = np.argmin(face_distances)
                if matches[best_match_index]:
                    name = self.known_face_names[best_match_index]
                    confidence = face_confidence(face_distances[best_match_index])

                self.face_names.append(f'{name} ({confidence})')

            self.process_current_frame = not self.process_current_frame

            # Display the results
            for (top, right, bottom, left), name in zip(self.face_locations, self.face_names):
                # Scale back up face locations since the frame we detected in was scaled to 1/4 size
                top *= 4
                right *= 4
                bottom *= 4
                left *= 4

                # Create the frame with the name
                cv2.rectangle(frame, (left, top), (right, bottom), (0, 0, 255), 2)
                cv2.rectangle(frame, (left, bottom - 35), (right, bottom), (0, 0, 255), cv2.FILLED)
                cv2.putText(frame, name, (left + 6, bottom - 6), cv2.FONT_HERSHEY_DUPLEX, 0.8, (255, 255, 255), 1)

            # Display the resulting image
            cv2.imshow('Face Recognition', frame)

            # Hit 'q' on the keyboard to quit!
            if cv2.waitKey(1) == ord('q'):
                break

        # Release handle to the webcam
        video_capture.release()
        cv2.destroyAllWindows()
```

FaceRecognition › run_recognition() › while True › if self.process_current_frame › for face_encoding in self.face_...

TODO | 6: Problems | Terminal | Python Console | Event Log

62 chars, 1 line break 67:62 CRLF UTF-8 4 spaces Python 3.9

File Edit View Navigate Code Refactor Run Tools VCS Window Help pythonProject - classpython.py

pythonProject 〉 classpython.py

classpython

notes.py | classpython.py | study.py | client.py | bharath.jpeg | elon musk.jpg

```python
            for (top, right, bottom, left), name in zip(self.face_locations, self.face_names):
                # Scale back up face locations since the frame we detected in was scaled to 1/4 size
                top *= 4
                right *= 4
                bottom *= 4
                left *= 4

                # Create the frame with the name
                cv2.rectangle(frame, (left, top), (right, bottom), (0, 0, 255), 2)
                cv2.rectangle(frame, (left, bottom - 35), (right, bottom), (0, 0, 255), cv2.FILLED)
                cv2.putText(frame, name, (left + 6, bottom - 6), cv2.FONT_HERSHEY_DUPLEX, 0.8, (255, 255, 255), 1)

            # Display the resulting image
            cv2.imshow('Face Recognition', frame)

            # Hit 'q' on the keyboard to quit!
            if cv2.waitKey(1) == ord('q'):
                break

        # Release handle to the webcam
        video_capture.release()
        cv2.destroyAllWindows()


if __name__ == '__main__':
    fr = FaceRecognition()
    fr.run_recognition()
```

pythonProject C:\Use
- faces
  - bharath.jpeg
  - donald trump.jp
  - elon musk.jpg
- classpython.py
- client.py
- notes.py
- pywhatkit_dbs.txt
- sample.txt
- study.py

External Libraries
- < Python 3.9 > C:\U
  - Binary Skeletons
  - DLLs
  - Extended Defini

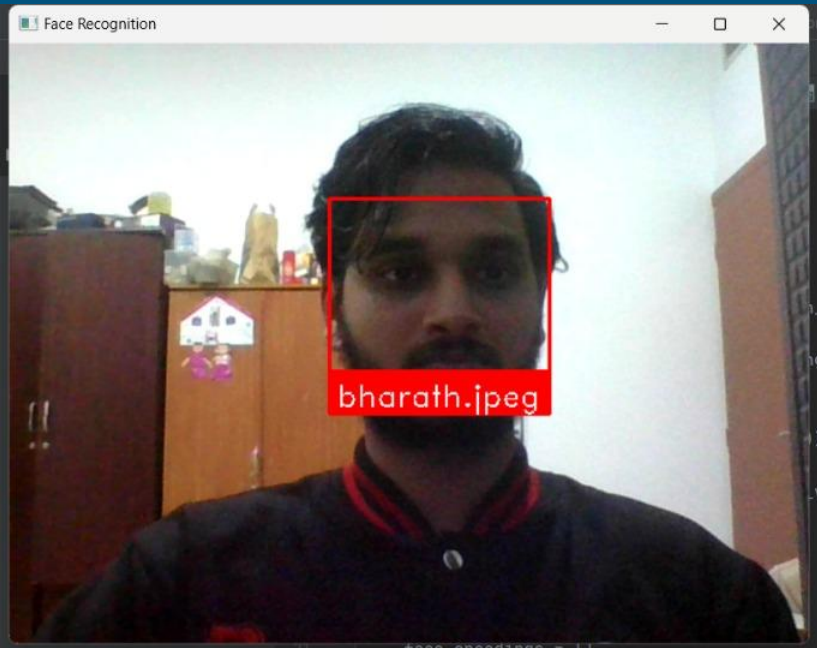
Structure
- face_confidence(face_c
- FaceRecognition

FaceRecognition 〉 run_recognition() 〉 while True 〉 if self.process_current_frame 〉 for face_encoding in self.face_...

TODO | 6: Problems | Terminal | Python Console | Event Log

62 chars, 1 line break 67:62 CRLF UTF-8 4 spaces Python 3.9

OUTPUT




```
face_encodings = []
face_names = []
known_face_encodings = []
known_face_names = []
process_current_frame = True

def __init__(self):
    self.encode_faces()

def encode_faces(self):
    for image in os.listdir('faces'):
        face_image = face_recognition.load_image_file(f"faces/{image}")
        face_encoding = face_recognition.face_encodings(face_image)[0]
```

HTML and CSS :Code (Registration Phase)

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>register form</title>

    <!-- custom css file link  -->
    <link rel="stylesheet" href="aa.css">

</head>
<body>

<div class="form-container">

    <form action="" method="post">
        <h3>register now</h3>

        <input type="text" name="name" required placeholder="enter your name">
        <input type="email" name="email" required placeholder="enter your email">
        <input type="password" name="password" required placeholder="enter your password">
        <input type="password" name="cpassword" required placeholder="confirm your password">
        <input type="submit" name="submit" value="register now" class="form-btn">
        <p>already have an account? <a href="aa1.html">login now</a></p>
    </form>

</div>

</body>
</html>
```

/var/www/html/logintry/register_form.php - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

register_form.php

```
</style>
<body>

<div class="form-container">

   <form action="" method="post" enctype="multipart/form-data">
      <h3>register now</h3>
      <?php
      if(isset($error)){
         foreach($error as $error){
            echo '<span class="error-msg">'.$error.'</span>';
         };
      };
      ?>
      <input type="text" name="name" required placeholder="enter your name">
      <input type="email" name="email" required placeholder="enter your email">
      <input type="password" name="password" id="password" required placeholder="enter your password" onInput="check()" />
   <br>
         <div id="set" >
        <i id="see" onclick="see()" class="far fa-eye"></i>
         </div>
         <div id="count">Length : 0</div>
           <div id="check0">
                <i class="far fa-check-circle"></i>  <span> Length more than 8.</span>
           </div>
           <div id="check2">
                <i class="far fa-check-circle"></i>  <span> Contains numerical character.</span>
           </div>
           <div id="check3">
                <i class="far fa-check-circle"></i>   <span>Contains special character.</span>
           </div>
           <div id="check4">
                <i class="far fa-check-circle"></i>  <span>Shouldn't contain spaces.</span>
           </div>
      <input type="password" name="cpassword" required placeholder="confirm your password">
   <h4>Security Question</h4>
   <select name="security_question">
         <option value="ques1">Name of your favorite city? </option>
         <option value="ques2">Name of your pet?</option>
      </select>
   <input type="text" name="secans" required placeholder="enter your answer">
   <label for="file">Choose file to upload</label>
      <input
      type="file"
      id="profilepic"
```

/var/www/html/logintry/register_form.php - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

register_form.php

```
        };
    };
    ?>
    <input type="text" name="name" required placeholder="enter your name">
    <input type="email" name="email" required placeholder="enter your email">
    <input type="password" name="password" id="password" required placeholder="enter your password" onInput="check()" />
<br>
        <div id="set" >
       <i id="see" onclick="see()" class="far fa-eye"></i>
        </div>
        <div id="count">Length : 0</div>
          <div id="check0">
              <i class="far fa-check-circle"></i>  <span> Length more than 8.</span>
          </div>
          <div id="check2">
              <i class="far fa-check-circle"></i>  <span> Contains numerical character.</span>
          </div>
          <div id="check3">
              <i class="far fa-check-circle"></i>   <span>Contains special character.</span>
          </div>
          <div id="check4">
              <i class="far fa-check-circle"></i>  <span>Shouldn't contain spaces.</span>
          </div>
    <input type="password" name="cpassword" required placeholder="confirm your password">
<h4>Security Question</h4>
<select name="security_question">
        <option value="ques1">Name of your favorite city? </option>
        <option value="ques2">Name of your pet?</option>
    </select>
<input type="text" name="secans" required placeholder="enter your answer">
<label for="file">Choose file to upload</label>
    <input
    type="file"
    id="profilepic"
    name="profilepic"
    accept=".jpg, .jpeg, .png" />
    <input type="submit" name="submit" value="register now" class="form-btn">
    <p>already have an account? <a href="login_form.php">login now</a></p>
</form>

</div>

</body>
</html>
```

register form

localhost/logintry/register_form.php

Kali Linux | Kali Tools | Kali Docs | Kali Forums | Kali NetHunter | Exploit-DB | Google Hacking DB | OffSec | TryHackMe | Login

REGISTER NOW

enter your name

enter your email

enter your password

Length : 0

- Length more than 8.
- Contains numerical character.
- Contains special character.
- Shouldn't contain spaces.

confirm your password

Security Question

Name of your favorite city?

enter your answer

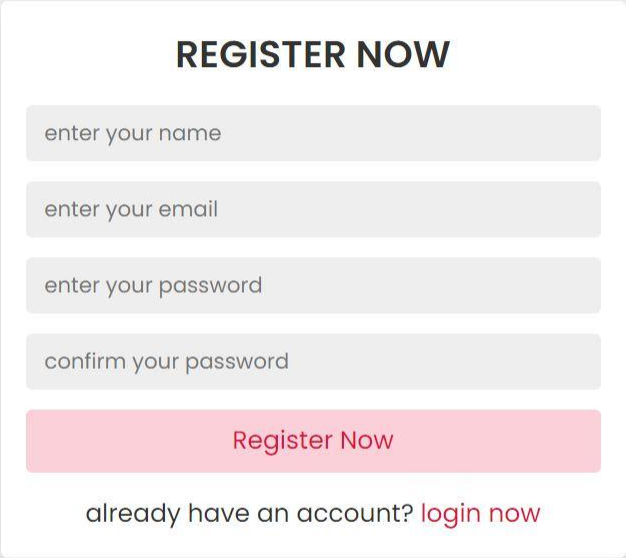
Choose file to upload

Browse... No file selected.

Register Now

already have an account? login now

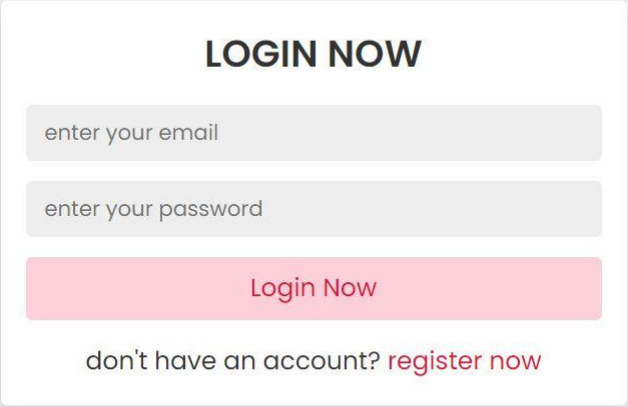
Output:HTML(Registration Phase)



HTML CODE:Login Phase

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>login form</title>


    <link rel="stylesheet" href="aa.css">

</head>
<body>

<div class="form-container">

    <form action="" method="post">
        <h3>login now</h3>

        <input type="email" name="email" required placeholder="enter your email">
        <input type="password" name="password" required placeholder="enter your password">
        <input type="submit" name="submit" value="login now" class="form-btn">
        <p>don't have an account? <a href="aa.html">register now</a></p>
    </form>

</div>

</body>
</html>
```

OUTPUT: HTML(Login Phase)



CSS CODES

```
@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@100;200;300;400;500;600&display=swap');

*{
   font-family: 'Poppins', sans-serif;
   margin:0; padding:0;
   box-sizing: border-box;
   outline: none; border:none;
   text-decoration: none;
}

.container{
   min-height: 100vh;
   display: flex;
   align-items: center;
   justify-content: center;
   padding:20px;
   padding-bottom: 60px;
}

.container .content{
   text-align: center;
}

.container .content h3{
   font-size: 30px;
   color:#333;
}

.container .content h3 span{
   background: crimson;
   color:#fff;
   border-radius: 5px;
   padding:0 15px;
}

.container .content h1{
   font-size: 50px;
   color:#333;
}
```

CSS CODES

```
40
41  .container .content h1 span{
42      color:crimson;
43  }
44
45  .container .content p{
46      font-size: 25px;
47      margin-bottom: 20px;
48  }
49
50  .container .content .btn{
51      display: inline-block;
52      padding:10px 30px;
53      font-size: 20px;
54      background: #333;
55      color:#fff;
56      margin:0 5px;
57      text-transform: capitalize;
58  }
59
60  .container .content .btn:hover{
61      background: crimson;
62  }
63
64  .form-container{
65      min-height: 100vh;
66      display: flex;
67      align-items: center;
68      justify-content: center;
69      padding:20px;
70      padding-bottom: 60px;
71      background: #eee;
72  }
73
74  .form-container form{
75      padding:20px;
76      border-radius: 5px;
77      box-shadow: 0 5px 10px rgba(0,0,0,.1);
78      background: #fff;
79      text-align: center;
```

CSS CODES

```
    text-align: center;
    width: 500px;
}

.form-container form h3{
    font-size: 30px;
    text-transform: uppercase;
    margin-bottom: 10px;
    color:#333;
}

.form-container form input,
.form-container form select{
    width: 100%;
    padding:10px 15px;
    font-size: 17px;
    margin:8px 0;
    background: #eee;
    border-radius: 5px;
}

.form-container form select option{
    background: #fff;
}

.form-container form .form-btn{
    background: #fbd0d9;
    color:crimson;
    text-transform: capitalize;
    font-size: 20px;
    cursor: pointer;
}

.form-container form .form-btn:hover{
    background: crimson;
    color:#fff;
}

.form-container form p{
    margin-top: 10px;
```

CSS CODES

```
.form-container form p{
    margin-top: 10px;
    font-size: 20px;
    color:#333;
}

.form-container form p a{
    color:crimson;
}

.form-container form .error-msg{
    margin:10px 0;
    display: block;
    background: crimson;
    color:#fff;
    border-radius: 5px;
    font-size: 20px;
    padding:10px;
}
```

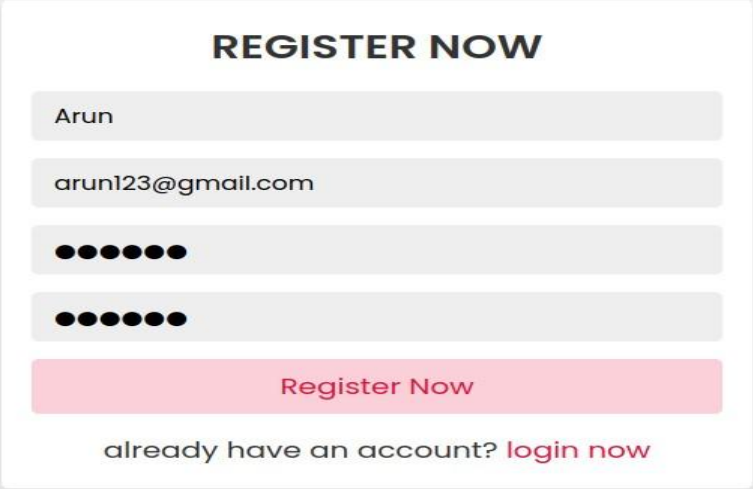
PHP and JS : PHP Code (Configuration Phase)

```php
<?php

$conn = mysqli_connect('localhost','agentbp','password','test_db');

?>
```

PHP Code (Registration Phase)

```php
<?php

@include 'config.php';

if(isset($_POST['submit'])){

    $name = mysqli_real_escape_string($conn, $_POST['name']);
    $email = mysqli_real_escape_string($conn, $_POST['email']);
    $pass = md5($_POST['password']);
    $cpass = md5($_POST['cpassword']);

    $select = " SELECT * FROM usertable WHERE email = '$email' ";

    $result = mysqli_query($conn, $select);

    if(mysqli_num_rows($result) > 0){

        $error[] = 'user already exist!';

    }else{

        if($pass != $cpass){
            $error[] = 'password not matched!';
        }else{
            $insert = "INSERT INTO usertable(name, email, password) VALUES('$name','$email','$pass')";
            mysqli_query($conn, $insert);
            header('location:login_form.php');
        }
    }

};
```

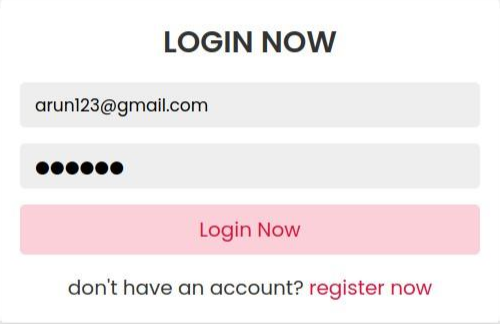
PHP (Registration Phase)



PHP Code (Login Phase)

```php
<?php

@include 'config.php';

session_start();

if(isset($_POST['submit'])){

    $name = mysqli_real_escape_string($conn, $_POST['name']);
    $email = mysqli_real_escape_string($conn, $_POST['email']);
    $pass = md5($_POST['password']);
    $cpass = md5($_POST['cpassword']);

    $select = " SELECT * FROM usertable WHERE email = '$email' && password = '$pass' ";

    $result = mysqli_query($conn, $select);

    if(mysqli_num_rows($result) > 0){

            header('location:user_page.php');


    }else{
        $error[] = 'incorrect email or password!';
    }

};
?>
```

PHP (Login Phase)



PHP Table Format

phpMyAdmin

Recent Favorites

- New
- information_schema
- mysql
- performance_schema
- phpmyadmin
- sys
- test_db
  - New
  - usertable
- user_db

Server: localhost:3306 » Database: test_db » Table: usertable

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Tracking | Triggers

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(255)			No	*None*		AUTO_INCREMENT	Change Drop More
2	name	varchar(255)	utf8mb4_general_ci		No	*None*			Change Drop More
3	email	varchar(255)	utf8mb4_general_ci		No	*None*			Change Drop More
4	password	varchar(255)	utf8mb4_general_ci		No	*None*			Change Drop More

Check all *With selected:* Browse Change Drop Primary Unique Index Spatial Fulltext Add to central columns Remove from central columns

Print Propose table structure Track table Move columns Normalize

Add 1 column(s) after password Go

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	id	5	A	No	

Create an index on 1 columns Go

Partitions

No partitioning defined!

Partition table

Information

Console

localhost/phpmyadmin/index.php?route=/sql&db=user_db&table=user_form&pos=0

Kali Linux | Kali Tools | Kali Docs | Kali Forums | Kali NetHunter | Exploit-DB | Google Hacking DB | OffSec | TryHackMe | Login

phpMyAdmin

Recent Favorites

- New
- information_schema
- mysql
- performance_schema
- phpmyadmin
- sys
- test_db
  - New
  - usertable
- user_db
  - New
  - user_form

Server: localhost:3306 » Database: user_db » Table: user_form

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Tracking | Triggers

Showing rows 0 - 10 (11 total, Query took 0.0002 seconds.)

```
SELECT * FROM `user_form`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

	id	name	email	password	securityques	securityans
Edit Copy Delete	3	abc	abc@gmail.com	f44ab78ef32badd1f9a944fe51f26eb8	ques1	Paris
Edit Copy Delete	4	thg	hudsf2@gmail.com	b85d492d1e74f88ccdfddb44e585c73e	ques1	london
Edit Copy Delete	5	iok	iok@gmail.com	f44ab78ef32badd1f9a944fe51f26eb8	ques1	France
Edit Copy Delete	6	idk	idk@gmail.com	50f799c11d323afd4a0137d541d14b0e	ques2	Dong
Edit Copy Delete	7	lion	lion123@gmail.com	3071b6df03e7b0e8b69bd84dc7f73653	ques1	idk
Edit Copy Delete	8	kon	rte@gmail.com	e99a18c428cb38d5f260853678922e03	ques1	il
Edit Copy Delete	9	kon	rte@gmail.com	b3af409bb8423187c75e6c7f5b683908	ques1	il
Edit Copy Delete	10	fer	sd@gmail.com	d67077351ca6ebbf3b3baa770a4fbb58	ques1	jbm
Edit Copy Delete	11	fer	sd@gmail.com	d187535d40b96a60387e8dff44b0c491	ques1	jbm
Edit Copy Delete	12	erg	fewf@gmail.com	f44ab78ef32badd1f9a944fe51f26eb8	ques1	add
Edit Copy Delete	13	erf	erf@gmail.com	9a9bfb0dae67ec64365bba22ceef575e	ques1	Mumbai

Check all | With selected: Edit | Copy | Delete | Export

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Query results operations

PHP Code (User Page)

```php
<?php

@include 'config.php';

session_start();

if(isset($_SESSION['user_name'])){
    header('location:login_form.php');
}

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>user page</title>

    <!-- custom css file link  -->
    <link rel="stylesheet" href="css/style.css">

</head>
<body>

<div class="container">

    <div class="content">
        <h3>hi, <span>user</span></h3>
        <h1>welcome <span><?php echo $_SESSION['user_name'] ?></span></h1>
        <p>this is an user page</p>
        <a href="login_form.php" class="btn">login</a>
        <a href="register_form.php" class="btn">register</a>
        <a href="logout.php" class="btn">logout</a>
    </div>

</div>

</body>
</html>
```

PHP (User Phase)

PHP Code (Logout Phase)

```php
<?php

@include 'config.php';

session_start();
session_unset();
session_destroy();

header('location:login_form.php');

?>
```

JS Code

```
var is_visible = false;

function see()
{
    var input = document.getElementById("password");
    var see = document.getElementById("see");

    if(is_visible)
    {
        input.type = 'password';
        is_visible = false;
        see.style.color='gray';
    }
    else
    {
        input.type = 'text';
        is_visible = true;
        see.style.color='#262626';
    }

}

function check()
{
    var input = document.getElementById("password").value;

    input=input.trim();
    document.getElementById("password").value=input;
    document.getElementById("count").innerText="Length : " + input.length;
    if(input.length>=5)
    {
        document.getElementById("check0").style.color="green";
    }
    else
    {
       document.getElementById("check0").style.color="red";
    }

    if(input.length<=10)
    {
        document.getElementById("check1").style.color="green";
    }
    else
    {
       document.getElementById("check1").style.color="red";
    }
```

JS Code

```
47
48          if(input.match(/[0-9]/i))
49          {
50              document.getElementById("check2").style.color="green";
51          }
52          else
53          {
54              document.getElementById("check2").style.color="red";
55          }
56
57          if(input.match(/[^A-Za-z0-9-' ']/i))
58          {
59              document.getElementById("check3").style.color="green";
60          }
61          else
62          {
63              document.getElementById("check3").style.color="red";
64          }
65
66          if(input.match(' '))
67          {
68              document.getElementById("check4").style.color="red";
69          }
70          else
71          {
72              document.getElementById("check4").style.color="green";
73          }
74
75      }
```

Security Question (Code)

/var/www/html/logintry/security.php - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

security.php

```php
<?php

@include 'config.php';

session_start();

if(isset($_POST['submit'])){

    $email = mysqli_real_escape_string($conn, $_POST['email']);
    $question = $_POST['security_question'];
    $qanswer = mysqli_real_escape_string($conn, $_POST['secans']);
    $selectone = " SELECT * FROM user_form WHERE email = '$email' ";
    $resultone = mysqli_query($conn, $selectone);
    $selecttwo = " SELECT * FROM user_form WHERE email = '$email' && securityques = '$question' && securityans = '$qanswer' ";
    $resulttwo = mysqli_query($conn, $selecttwo);
    if(mysqli_num_rows($resultone) == 0){

        $error[] = 'User not found!';

    }else{
        if(mysqli_num_rows($resulttwo) > 0){
            header('location:user_page.php');
        }else{
            header('location:face.php');
        }
    }
}
?>


<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Security Question</title>

    <!-- custom css file link  -->
    <link rel="stylesheet" href="css/style.css">

</head>
<body>

<div class="form-container">
```

Line 1, Column 1 Spaces: 3 PHP

/var/www/html/logintry/security.php - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

security.php

```php
            header('location:user_page.php');
        }else{
            header('location:face.php');
        }
    }
}
?>


<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Security Question</title>

    <!-- custom css file link  -->
    <link rel="stylesheet" href="css/style.css">

</head>
<body>

<div class="form-container">

    <form action="" method="post">
        <h3>Security Question</h3>
        <?php
        if(isset($error)){
            foreach($error as $error){
                echo '<span class="error-msg">'.$error.'</span>';
            };
        };
        ?>
        <input type="email" name="email" required placeholder="Enter your email">
    <select name="security_question">
            <option value="ques1">Name of your favorite city? </option>
            <option value="ques2">Name of your pet?</option>
        </select>
    <input type="text" name="secans" required placeholder="enter your answer">
    <input type="submit" name="submit" value="Login Now" class="form-btn">
    <p>Don't have an account? <a href="register_form.php">Register now</a></p>
    </form>
</div>
</body>
</html>
```

Line 1, Column 1 Spaces: 3 PHP

Security Question × localhost / localhost / use ×

localhost/logintry/security.php

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec TryHackMe | Login

SECURITY QUESTION

abc@gmail.com

Name of your favorite city?

Paris

Login Now

Don't have an account? Register now

Face Recognition Page

WRONG SECURITY QUESTION

FACE RECOGNITON START

References

- Morris1979. Robert T. Morris and Ken Thompson, “Password Security: A Case History,” Communications of the ACM, vol. 22, no. 11, pp. 594-597, November 1979.
- cscjournals.org/manuscript/Journals/IJS/Volume8/Issue1/IJS-131,International Journal of Security (IJS), Volume (8) : Issue (1) : 2014