VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering
J Component Report

Programme : B.Tech (CSE - AI & ML)

Course Title : Information Security Management

Course Code : CSE3502

Slot : G2

Title: TreSecura - A Three Level Authentication System

Team members :	Jayanand Jayan	- 20BAI1085
	Aayush Balaji	- 20BAI1121
	Rohita Chakraborty	- 20BAI1213

Faculty: M. BRAVEEN
Date: 13.04.2023

ABSTRACT

A security breach may endanger the private records of a company or an individual. Text-based passwords are the most common type of password used for security purposes. However, those passwords can easily be compromised, and one's private information can fall into the wrong hands. Some authentication techniques are based on the user's physical and behavioral characteristics, such as voice recognition, while others are dependent on the user's knowledge, such as textual and graphical passwords.

However, these systems are still insecure and allow attackers to readily grab data. Furthermore, users frequently choose simple passwords that attackers can quickly guess. The three level security system's major goal is to give superior protection to online applications, prevent unwanted access, and make the apps more user friendly.

A three-level authentication system is a security mechanism that employs multiple methods to verify the identity of a user before granting access to a system, device, or application. This system involves three different levels of authentication: something the user knows, something the user has, and something the user is. By combining these three levels of authentication, a system can significantly increase the security of the authentication process and reduce the risk of unauthorized access. The use of multiple factors of authentication makes it much more difficult for hackers or malicious actors to gain access to sensitive information or systems. Overall, the three-level authentication system is an effective security mechanism that provides an additional layer of protection against unauthorized access.

TABLE OF CONTENTS

INTRODUCTION

Authentication is an essential aspect of security in modern computing systems. With the increasing use of technology in various domains, such as finance, healthcare, and e-commerce, the need for robust and reliable authentication mechanisms has become more critical than ever. In recent years, multi-layered authentication systems have gained popularity due to their ability to provide enhanced security and prevent unauthorized access to sensitive information.

This report explores the design and implementation of a three-layer authentication system involving login, OTP, and QR modules. The proposed system aims to provide a secure and convenient authentication mechanism for users, which involves a combination of something they know (login credentials), something they have (OTP), and something they are (QR code).

The first layer of authentication involves the traditional login mechanism, where users enter their username and password. The second layer involves the use of a one-time password (OTP) that is sent to the user's registered email address. The third layer involves the use of a QR code that is displayed on the user's device, and the user is required to scan the code using their mobile device's camera to complete the authentication process.

We thereby present a detailed analysis of the proposed three-layer authentication system, including the design, implementation, and testing. We also compare it with existing authentication mechanisms. The results show that our proposed system provides a higher level of security than traditional login mechanisms and is more convenient than some existing multi-layered authentication systems.

LITERATURE REVIEW

1. "User Authentication: A Three Level Password Authentication Mechanism (2020)" - is based on verification and validation methodology for the user authentication. The three-level security in the proposed system consists of textual password, recognition of bot-attack and detection of color code. From the result analysis it is found that the three-level authentication provides a reliable security level in comparison to the existing mechanisms.

2. "A Secured One Time Password Authentication Technique using (3, 3) Visual Cryptography Scheme (2019)" - proposes a one-time password authentication technique that utilizes visual cryptography to enhance security. Visual cryptography is a technique for encrypting images in such a way that the decrypted image can only be seen when the encrypted images are superimposed. It's possible that the authors propose using this technique to create a secure one-time password that can be used for authentication purposes.

3. "A Three-Level Authentication System Based on Face Recognition and Password for Mobile Devices" by W. Chen and W. Liu (2017) - This research paper proposes a three-level authentication system that combines face recognition and password authentication for mobile devices. The system uses face recognition as the first level of authentication, password as the second level, and SMS verification as the third level. The study shows that this system provides a high level of security and usability for mobile devices.

4. "A Three-Level Authentication Scheme for Cloud Computing Security" by S. Ali, R. H. Abbas, and A. Sher (2019) - This research paper presents a three-level authentication scheme for cloud computing security. The proposed system includes three levels of authentication: password-based, token-based, and biometric-based. The study demonstrates that this system provides an efficient and secure way to authenticate users in cloud computing environments.

5. "A Three-Level Authentication Model for E-Banking System Security" by S. Das and S. Chakraborty (2018) - This research paper proposes a three-level authentication model for e-banking system security. The system uses a combination of password-based, token-based, and biometric-based authentication. The study

shows that this model provides a high level of security and usability for e-banking systems.

6. "A Three-Level Authentication System for Internet of Things (IoT) Devices" by M. R. Khan, M. A. Hossain, and M. A. Islam (2018) - This research paper proposes a three-level authentication system for Internet of Things (IoT) devices. The proposed system includes three levels of authentication: password-based, token-based, and biometric-based. The study demonstrates that this system provides a secure and efficient way to authenticate users in IoT environments.

7. "A Three-Level Authentication System for Online Social Networks" by A. M. ElSisi and H. M. ElNainay (2017) - This research paper proposes a three-level authentication system for online social networks. The proposed system includes three levels of authentication: password-based, token-based, and biometric-based. The study shows that this system provides a high level of security and usability for online social networks. Additionally, the system can detect and prevent various types of attacks, such as phishing attacks and man-in-the-middle attacks.

8. "Enhancing Security of Online Banking Transactions Using Three-Factor Authentication" (2015) by S. Y. Kim et al. - proposes a system that enhances the security of online banking transactions by using three-factor authentication. The system combines a password, a token, and a biometric identifier to verify the user's identity. The authors evaluate the proposed system and show that it is more secure than traditional single-factor authentication systems. The paper concludes that the proposed system can effectively enhance the security of online banking transactions and prevent unauthorized access.

9. "A Three-Factor Authentication System Based on Biometric and Cryptographic Techniques" (2019) by M. Al-Hujaily et al. - proposes a three-factor authentication system that uses biometric and cryptographic techniques to enhance security. The system combines facial recognition, a password, and a one-time token to verify the user's identity. The authors evaluate the proposed system and show that it provides a high level of security compared to traditional authentication systems. The paper concludes that the proposed system can be effectively used in various applications, such as online banking, to prevent unauthorized access.

10. "A Survey of Three-Factor Authentication Schemes" (2014) by S. Jain et al. - is a survey of three-factor authentication schemes used in various applications. The authors provide an overview of the most commonly used three-factor authentication schemes, including password-token, biometric-password-token, and biometric-token-password, among others. The authors also discuss the strengths and weaknesses of each scheme and provide recommendations for their use in various applications. The paper concludes that three-factor authentication can provide a higher level of security compared to traditional single-factor authentication systems.

PROBLEM STATEMENT

The increasing number of security breaches and unauthorized access to sensitive information has become a major concern for organizations of all sizes. To address this issue, a secure and robust authentication system is required that can provide an additional layer of security beyond the traditional username and password method.

The problem that a three-level authentication system aims to solve is the vulnerability of traditional two-factor authentication systems. Two-factor authentication systems, which rely on something the user knows (such as a password) and something the user has (such as a security token), have become widely used in various applications to enhance security. However, these systems still have weaknesses that can be exploited by attackers. For example, passwords can be guessed or stolen, and tokens can be lost or stolen.

To address these limitations, a three-level authentication system adds an additional layer of security by using a third factor. By adding this extra layer of security, a three-level authentication system aims to increase the security of authentication and prevent unauthorized access to sensitive information or systems.

The objective of this project is to design and implement a three-layer authentication system that includes knowledge-based and device-based authentication methods to ensure the highest level of security for the users. It should be easy to use, yet highly secure. It should be able to integrate with existing systems and allow for seamless authentication across multiple devices and platforms. The system should also provide a flexible, scalable, and customizable solution that can be easily adapted to meet the changing needs of the organization.

MODULES AND METHODOLOGY

1. Login layer:

A login authentication system using password verification is a commonly used mechanism to verify the identity of a user attempting to access a system, service, or application.

The Login Layer implements the basic username password-based authentication system. The user will be provided with two input fields (one for username and one for password). The user, after entering the correct information, can click on the login button and if authenticated, will be taken to the next page and otherwise will be informed that the credentials are incorrect and be restricted from further access.

The mechanism of manual input of username and password is supported with salting and hashing. Hashing is the process of transforming any given key or a string of characters into another value. Password salting is a technique to protect passwords stored in databases by adding a string of 32 or more characters and then hashing them. The application also works to prevent SQL Injection attacks using SQLAlchemy library. The system then compares the entered password with the password stored in its database for that particular user. If the entered password matches the stored password, the user is granted access to the system or application.

2. OTP Layer:

OTP stands for One Time Password. After the user's credentials are verified, the system sends a one-time password (OTP) to the user's registered email address. The user then enters the received OTP into the login page to complete the authentication process. The system verifies the entered OTP against the one generated by the system. If the OTPs match, the user is granted access to the protected resources.

The OTP is a unique, temporary code that is generated by the system and is valid for a one-time use. This means that even if hackers intercept the OTP, they will not be able to use it for any unauthorized access since it will become invalid once it is used. By combining something the user knows (their username and password) with something the user has (their phone or email), OTP provides an extra layer of security to the login process, making it more difficult for unauthorized users to access the system. We implement this module primarily using the *random* method provided in python. It is basically a cryptographically strong pseudo-random number generator using which we can generate a six-digit randomized OTP that can be verified through the user’s email.

We implemented this module in PHP using the *PHPMailer* library. This library provides an easy-to-use interface to send mails using PHP. This is used to generate a random OTP using the rand() function and send it to the user's email address. We set the SMTP server settings for Gmail and set the sender's email and password with the credentials of a custom account created by us for this verification purpose. We also set the user’s email address, subject, and body of the email.

3. <u>QR Layer:</u>

QR code authentication is a security process that involves scanning a QR code to verify the identity of a user or device. QR codes are two-dimensional barcodes that contain information that can be read by a QR code reader or smartphone camera.

In QR code authentication, the QR code is generated by the authentication system and contains a unique identifier or a one-time password that is linked to the user's identity. The user then scans the QR code using their smartphone or other compatible device, which validates the code and grants access to the system or service.

QR code authentication can be used in various applications, such as online banking, secure login, and access control systems. It provides an easy and convenient way to authenticate users without the need for physical tokens or passwords. Additionally, QR codes can be generated quickly and easily, making them a popular choice for many businesses and organizations.

In our project, we have generated a QR Code which has a Time-Based OTP (TOTP). This time based OTP will be valid only for 4 minutes, within which the user has to scan the QR Code from their mobile phone. Once they scan the QR Code, a new web page opens on their mobile phone. There, they just have to click on a “Verify” button. If the verification is successful, their laptop screen immediately shows a success message.

SCREENSHOTS

CODE

main.py

```
import qrcode
import pyotp
import google
import jwt
from fastapi import FastAPI, Request, Response, WebSocket,
WebSocketException, Form, Depends, HTTPException, Body
from fastapi.responses import HTMLResponse, RedirectResponse
from fastapi.templating import Jinja2Templates
from fastapi.staticfiles import StaticFiles
from pydantic import ValidationError
from google.oauth2 import id_token
from google.auth.transport import requests
from sqlalchemy.orm import Session
from typing import Annotated
import crud, models, schemas
from database import SessionLocal, engine
import bcrypt

app = FastAPI()
app.mount("/static", StaticFiles(directory="static"), name="static")

templates = Jinja2Templates(directory="templates")

CLIENT_ID = <our-client-id>
SECRET_KEY = <our-secret-key>
ALGORITHM = <algorithm>

def get_db():
    db = SessionLocal()
    try:
        yield db

    finally:
        db.close()
```

```
@app.get("/login")
async def display_login(request: Request):
    return templates.TemplateResponse("login.html", {"request": request})

@app.get("/signup")
async def display_signup(request: Request):
    return templates.TemplateResponse("signup.html", {"request": request})

@app.post("/register")
def register(request: Request, response: Response, username: str = Form(),
password: str = Form(), mail: str = Form(), phno: str = Form(), db:
Session = Depends(get_db)):
    db_user = crud.get_user_by_username(db, username)
    if db_user is not None:
        raise HTTPException(status_code=403, detail="User already exists")

    new_user = schemas.UserCreate(username=username, password=password,
mail=mail, phno=phno)
    new_user = crud.create_user(db, new_user)

    return {"authorized": True, "name": new_user.username}



@app.post("/verify")
def login(request: Request, response: Response, username: str = Form(),
password: str = Form(), db: Session = Depends(get_db)):
    db_user = crud.get_user_by_username(db, username)
    if db_user is None:
        raise HTTPException(status_code=404, detail="User not found")

    if not bcrypt.checkpw(password.encode('utf-8'),
db_user.hashed_pwd.encode('utf-8')):
        raise HTTPException(status_code=401, detail="Unauthorized user")


    return RedirectResponse("http://localhost/otp-ver/index.php")
```

```
@app.get("/display_qr", response_class=HTMLResponse)
def display_qr(request: Request):
    global totp
    totp = pyotp.TOTP('base32secret3232')
    totp.interval = 240
    global query_arg
    query_arg = str(totp.now())

    return templates.TemplateResponse("qr_display.html", {"request":
request, "totp": query_arg })

connected_client = set()

@app.websocket("/ws")
async def websocket_endpoint(websocket: WebSocket):
    connected_client.add(websocket)
    await websocket.accept()
    try:
        while True:
            data = await websocket.receive_text()
            print("My data is: ", data)
            if data == query_arg:
                for client in connected_client:
                    await client.send_text("Authenticated")
    finally:
        connected_client.remove(websocket)
```

crud.py

```
from sqlalchemy.orm import Session
import models, schemas
import bcrypt

def get_user(db: Session, user_id: int):
    return db.query(models.User).filter(models.User.id == user_id).first()

def get_user_by_username(db: Session, username: str):
    return db.query(models.User).filter(models.User.username ==
username).first()
```

```
def get_user_by_mail(db: Session, mail: str):
    return db.query(models.User).filter(models.User.mail==mail).first()

def create_user(db: Session, user: schemas.UserCreate):
    hashed_password = bcrypt.hashpw(user.password.encode('utf-8'),
bcrypt.gensalt())
    db_user = models.User(username=user.username,
hashed_pwd=hashed_password, phno=user.phno, mail=user.mail)
    db.add(db_user)
    db.commit()
    db.refresh(db_user)
    return db_user
```

database.py

```
from sqlalchemy import create_engine
from sqlalchemy.ext.declarative import declarative_base
from sqlalchemy.orm import sessionmaker

SQLALCHEMY_DATABASE_URL = "mysql://root:<password>@localhost/tresecura"

engine = create_engine(SQLALCHEMY_DATABASE_URL)

SessionLocal = sessionmaker(autocommit=False, autoflush=False,
bind=engine)

Base = declarative_base()
```

models.py

```
from sqlalchemy import Boolean, Column, Integer, String
from database import Base

class User(Base):
    __tablename__ = "users"
    id = Column(Integer, primary_key=True, index=True)
    username = Column(String(255), index=True)
    hashed_pwd = Column(String(1024))
    mail = Column(String(255))
    phno = Column(Integer)
    otp = Column(String(10))
```

schemas.py

```
from pydantic import BaseModel

class UserBase(BaseModel):
    username: str

class UserCreate(UserBase):
    mail: str
    password: str
    phno: int
    otp: str

class User(UserBase):
    id: int

    class Config:
        orm_mode = True
```

login.html

```
<html>

<head>
    <title>TreSecura</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="{{ url_for('static', path='/styles/login.css') }}"
rel="stylesheet">
    <link rel="icon" type="image/x-icon" href="{{ url_for('static',
path='/images/favicon.ico') }}">
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link
href="https://fonts.googleapis.com/css2?family=Montserrat:wght@600&family=
Poppins&display=swap" rel="stylesheet">
    <script src="https://accounts.google.com/gsi/client" async
defer></script>
```

```
</head>

<body>
    <div class="container">
        <img class="logo" src="{{ url_for('static',
path='/images/Logo.png') }}" alt="TreSecura">
        <form class="login_form" action="/verify" method="POST">
            <label for="uname">Username<span style="color:
red;">*</span></label><br>
            <input id="uname" name="username" type="text" required>
            <label for="pass">Password<span style="color:
red;">*</span></label><br>
            <input id="pass" name="password" type="password" required>
            <button class="log_butt">
                Sign in
            </button>
            <br>
            <a class="signup" href="/signup">Sign up now!</a>
        </form>
    </div>
</body>

</html>
```

signup.html

```
<html>

<head>
    <title>TreSecura</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="{{ url_for('static', path='/styles/signup.css') }}"
rel="stylesheet">
    <link rel="icon" type="image/x-icon" href="{{ url_for('static',
path='/images/favicon.ico') }}">
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link
href="https://fonts.googleapis.com/css2?family=Montserrat:wght@600&family=
Poppins&display=swap" rel="stylesheet">
```

```
    <script src="https://accounts.google.com/gsi/client" async
defer></script>
</head>

<body>
    <div class="container">
        <img class="logo" src="{{ url_for('static',
path='/images/Logo.png') }}" alt="TreSecura">
        <form class="signup_form" action="/register" method="POST">
            <label for="uname">Username<span style="color:
red;">*</span></label><br>
            <input id="uname" name="username" type="text" required>
            <label for="pass">Password<span style="color:
red;">*</span></label><br>
            <input id="pass" name="password" type="password" required>
            <label for="mail">Email ID<span style="color:
red;">*</span></label><br>
            <input id="mail" name="mail" type="email" required>
            <label id="phno">Phone Number<span style="color:
red;">*</span></label><br>
            <input id="phno" name="phno" type="tel" required>
            <button class="signup_butt">
                Sign Up
            </button>
            <a class="signin" href="/login">Sign in instead</a>
        </form>
    </div>
</body>

</html>
```

index.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">
```

```
        <title>TresSecura - OTP Verification</title>
        <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.cs
s">
        <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></
script>
        <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js">
</script>
        <style type="text/css">
            .login-form {
                width: 340px;
                margin: 50px auto;
            }
            .login-form form {
                margin-bottom: 15px;
                background: #f7f7f7;
                box-shadow: 0px 2px 2px rgba(0, 0, 0, 0.3);
                padding: 30px;
            }
            .login-form h2 {
                margin: 0 0 15px;
            }
            .form-control, .btn {
                min-height: 38px;
                border-radius: 2px;
            }
            .btn {
                font-size: 15px;
                font-weight: bold;
            }
        </style>
    </head>
<body>
    <div class="login-form">
        <img src="Logo.png" style="width: 400px; height=250px; align-
items: left;">
        <form  method="post">
            <h2 class="text-center">Log in</h2>
            <div class="form-group first_box">
                <input type="text" id="email" class="form-control"
placeholder="Email" required="required">
                <span id="email_error" class="field_error"></span>
            </div>
            <div class="form-group first_box">
                <button type="button" class="btn btn-primary btn-block"
onclick="send_otp()">Acquire OTP</button>
            </div>
            <div class="form-group second_box">
                <input type="text" id="otp" class="form-control"
placeholder="OTP" required="required">
                <span id="otp_error" class="field_error"></span>
            </div>
            <div class="form-group second_box">
                <button type="button" class="btn btn-primary btn-block"
onclick="submit_otp()">Submit OTP</button>
            </div>
        </form>
    </div>
    <script>
        function send_otp(){
            var email=jQuery('#email').val();
            jQuery.ajax({
                url:'send_otp.php',
                type:'post',
                data:'email='+ email,
                success:function(result){
                    // alert(result);
                    if(result=='yes'){
                        jQuery('.second_box').show();
                        jQuery('.first_box').hide();
                    }
                    if(result=='not_exist'){
                        jQuery('#email_error').html('Please enter a valid
email');
                    }
                }
            });
        }
        function submit_otp(){
            var otp=jQuery('#otp').val();
            jQuery.ajax({
                url:'check_otp.php',
                type:'post',
                data:'otp='+otp,
                success:function(result){
                    if(result=='yes'){
                        window.location='http://localhost:8000/display_qr'
                    }
                    if(result=='not_exist'){
                        jQuery('#otp_error').html('Please enter a valid
otp');
                    }
                }
            });
        }
    </script>
    <style>
        .second_box {
            display:none;
        }
        .field_error {
            color:red;
        }
    </style>
</body>
</html>
```

send_otp.php

```
<?php

require 'requires/PHPMailer.php';
require 'requires/Exception.php';
require 'requires/SMTP.php';

use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;
```

```php
use PHPMailer\PHPMailer\SMTP;

session_start();
$con=mysqli_connect('localhost','root','Password','tresecura');
$email=$_POST['email'];
$res=mysqli_query($con, "select * from users where mail =  '$email'");
$count=mysqli_num_rows($res);
if($count>0){
    $otp=rand(100000, 999999);
    mysqli_query($con,"update users set otp='$otp' where mail='$email'");
    $html="Hi there!<br>Your OTP verification code is: <h2>".$otp
."</h2>";
    $_SESSION['EMAIL']=$email;
    smtp_mailer($email,'OTP Verification',$html);
    echo "yes";
}else{
    echo "not_exist";
}

function smtp_mailer($to, $subject, $msg) {
    $mail = new PHPMailer(true);
    $mail->IsSMTP();
    $mail->Host = "smtp.gmail.com";
    $mail->SMTPAuth = true;
    $mail->Username = "ism.tressecura@gmail.com";
    $mail->Password = "ptubxasoyrugxgvt";
    $mail->SMTPSecure = 'ssl';
    $mail->Port = 465;
    $mail->SetFrom("tressecura@gmail.com", "TreSecura");
    $mail->AddAddress($to);
    $mail->IsHTML(true);
    $mail->CharSet = 'UTF-8';
    $mail->Subject = $subject;
    $mail->Body =$msg;
    if(!$mail->Send()){
        return 0;
    }else{
        return 1;
    }
}
```

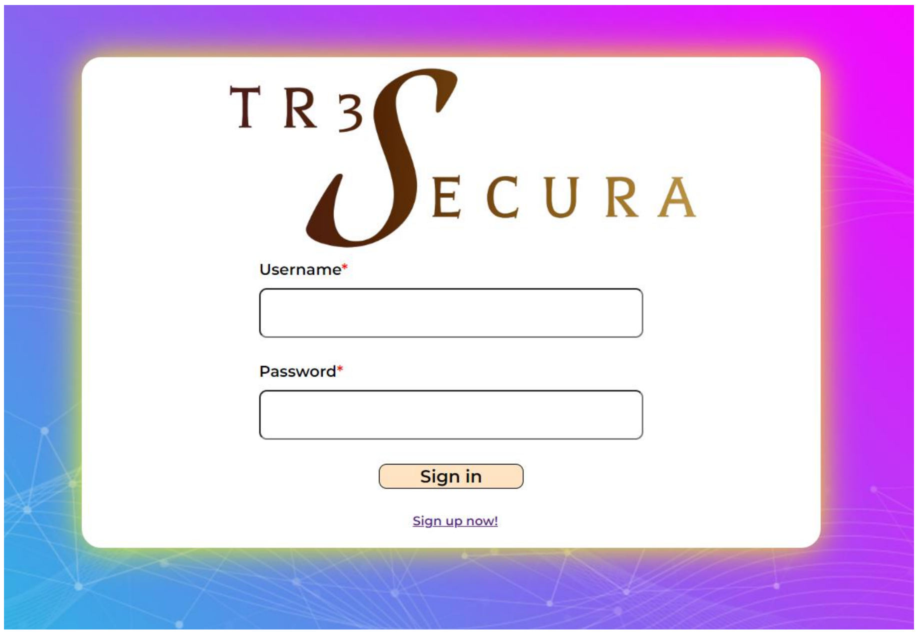
```
?>
```

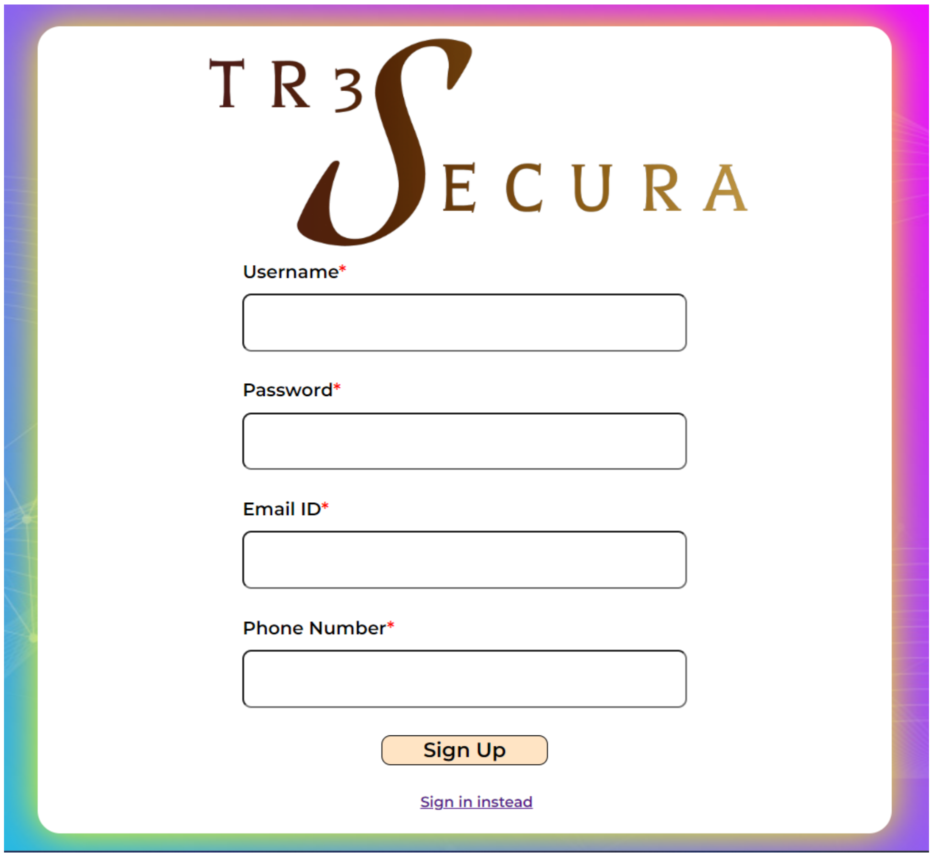
qr_display.html

```
<html>
    <head>
        <title>TresSecura</title>
        <meta name="viewport" content="width=device-width, initial-
scale=1.0">
        <link href="{{ url_for('static', path='/styles/qr.css') }}"
rel="stylesheet">
        <link rel="icon" type="image/x-icon" href="{{ url_for('static',
path='/images/favicon.ico') }}">
        <script
src="https://cdnjs.cloudflare.com/ajax/libs/qrcodejs/1.0.0/qrcode.min.js">
</script>
    </head>

    <body>
        <div class="flexContainer" id="mainContainer">
        </div>
        <div id="hiddenStuff" style="visibility: hidden;">{{ totp }}</div>
        <script>
            var ws = new WebSocket("ws://192.168.65.54:8000/ws");
            ws.onmessage = function(event) {
                if (event.data == "Authenticated") {
                    if (!(navigator.userAgent.match(/Android/i)
                        || navigator.userAgent.match(/webOS/i)
                        || navigator.userAgent.match(/iPhone/i)
                        || navigator.userAgent.match(/iPad/i)
                        || navigator.userAgent.match(/iPod/i)
                        || navigator.userAgent.match(/BlackBerry/i)
                        || navigator.userAgent.match(/Windows Phone/i))) {

document.getElementById("mainContainer").innerHTML = "";
                            const succImg = document.createElement("img");
                            succImg.src = "{{ url_for('static',
path='/images/Tick.gif') }}"
                            succImg.setAttribute("class", succImg)
```

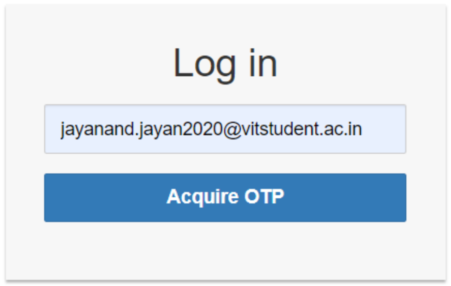
```
document.getElementById("mainContainer").appendChild(succImg);
                            const succMsg = document.createElement("h1");
                            succMsg.setAttribute("class", "success");
                            succMsg.innerHTML = "Successfully
Authenticated!";

document.getElementById("mainContainer").appendChild(succMsg);
                        }
                }
                else {
                    if (!(navigator.userAgent.match(/Android/i)
                        || navigator.userAgent.match(/webOS/i)
                        || navigator.userAgent.match(/iPhone/i)
                        || navigator.userAgent.match(/iPad/i)
                        || navigator.userAgent.match(/iPod/i)
                        || navigator.userAgent.match(/BlackBerry/i)
                        || navigator.userAgent.match(/Windows Phone/i))) {

document.getElementById("mainContainer").innerHTML = "";
                            const failImg = document.createElement("img");
                            failImg.src = "{{ url_for('static',
path='/images/Cross.gif') }}"
                            failImg.setAttribute("class", failImg)

document.getElementById("mainContainer").appendChild(failImg);
                            const failMsg = document.createElement("h1");
                            failMsg.setAttribute("class", "failure");
                            failMsg.innerHTML = "Authentication Failed!";

document.getElementById("mainContainer").appendChild(failMsg);
                        }
                }
            }
            function waitForSocketConnection(socket, callback) {
                setTimeout(
                    function() {
                        if (socket.readyState === WebSocket.OPEN) {
                            console.log("Connection is made");
                            if (callback != null) {
                                callback();
                            }
                            else {
                                console.log("Waiting for connection...");
                                waitForSocketConnection(socket, callback);
                            }
                        }
                    }, 100);
            }


            function handleClick() {
                    waitForSocketConnection(ws, function() {
                    var data =
document.getElementById("hiddenStuff").innerHTML;
                    ws.send(data)
                    console.log("Message sent");
                });
            }
            if (navigator.userAgent.match(/Android/i)
                || navigator.userAgent.match(/webOS/i)
                || navigator.userAgent.match(/iPhone/i)
                || navigator.userAgent.match(/iPad/i)
                || navigator.userAgent.match(/iPod/i)
                || navigator.userAgent.match(/BlackBerry/i)
                || navigator.userAgent.match(/Windows Phone/i)) {
            const container = document.getElementById("mainContainer");
            const plswait = document.createElement("h1");
            plswait.setAttribute("class", "success");
            plswait.innerHTML = "Please check your laptop screen for
authentication status";
            container.appendChild(plswait);
            const verifyButton = document.createElement("button");
            verifyButton.setAttribute("class", "verifyBtn");
            verifyButton.onclick = handleClick;
            verifyButton.innerHTML = "Verify"
            container.appendChild(verifyButton);
         }

         else {
            const container = document.getElementById("mainContainer");
            const logoImg = document.createElement("img");
            logoImg.src = "{{ url_for('static', path='/images/Logo.png')
}}";
            logoImg.alt = "TreSecura";
            logoImg.setAttribute("class", "logo");
            container.appendChild(logoImg);
            container.appendChild(document.createElement("br"));
            container.appendChild(document.createElement("br"));
            container.appendChild(document.createElement("br"));
            const scanInstr = document.createElement("p");
            scanInstr.innerHTML = "Scan the below QR code using your
mobile phone";
            container.appendChild(scanInstr);
            const qrImg = document.createElement("div");
            qrImg.setAttribute("id", "myqrcode");
            qrImg.alt = "QR Code";
            qrImg.setAttribute("class", "qr");
            container.appendChild(qrImg);
            var qrcode = new QRCode("myqrcode", {
                text: "http://192.168.65.54:8000/display_qr",
                colorLight: "#4A171E",
                colorDark: "#FFFFFF",
                correctLevel: QRCode.CorrectLevel.H
            });
         }
        </script>
    </body>
</html>
```

OUTPUT


TR3 SECURA
Username*
Password*
Sign in
Sign up now!


TR3 SECURA
Username*
Password*
Email ID*
Phone Number*
Sign Up
Sign in instead

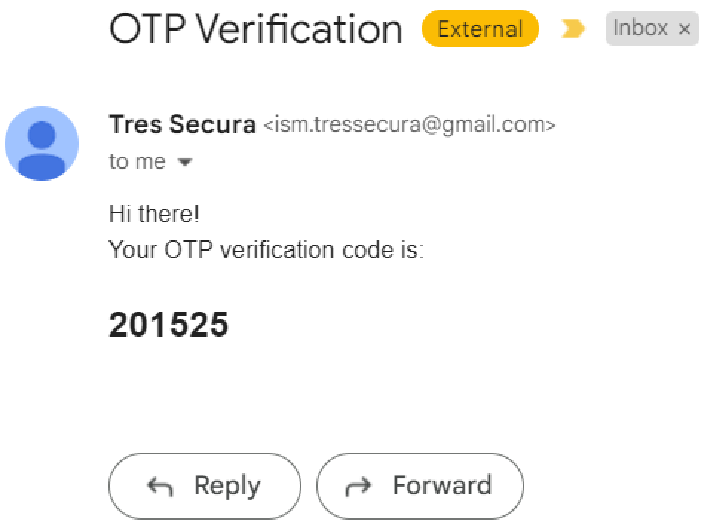
TR3 SECURA



Log in

jayanand.jayan2020@vitstudent.ac.in

Acquire OTP

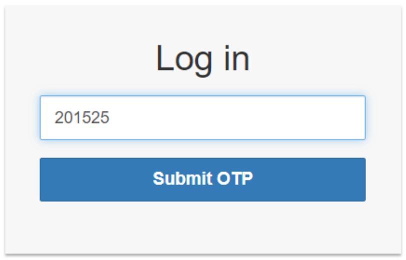


OTP Verification External Inbox ×

Tres Secura <ism.tressecura@gmail.com>
to me

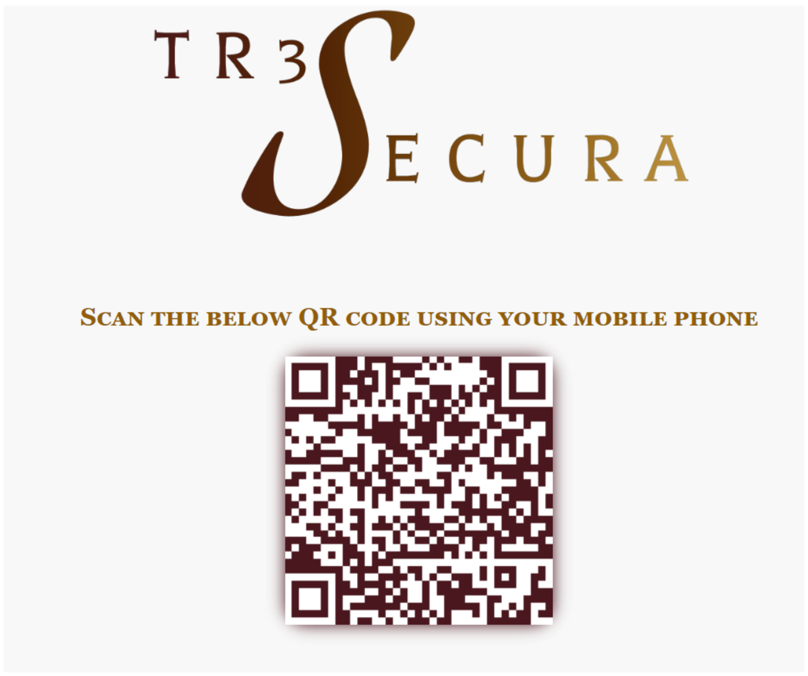
Hi there!
Your OTP verification code is:

201525

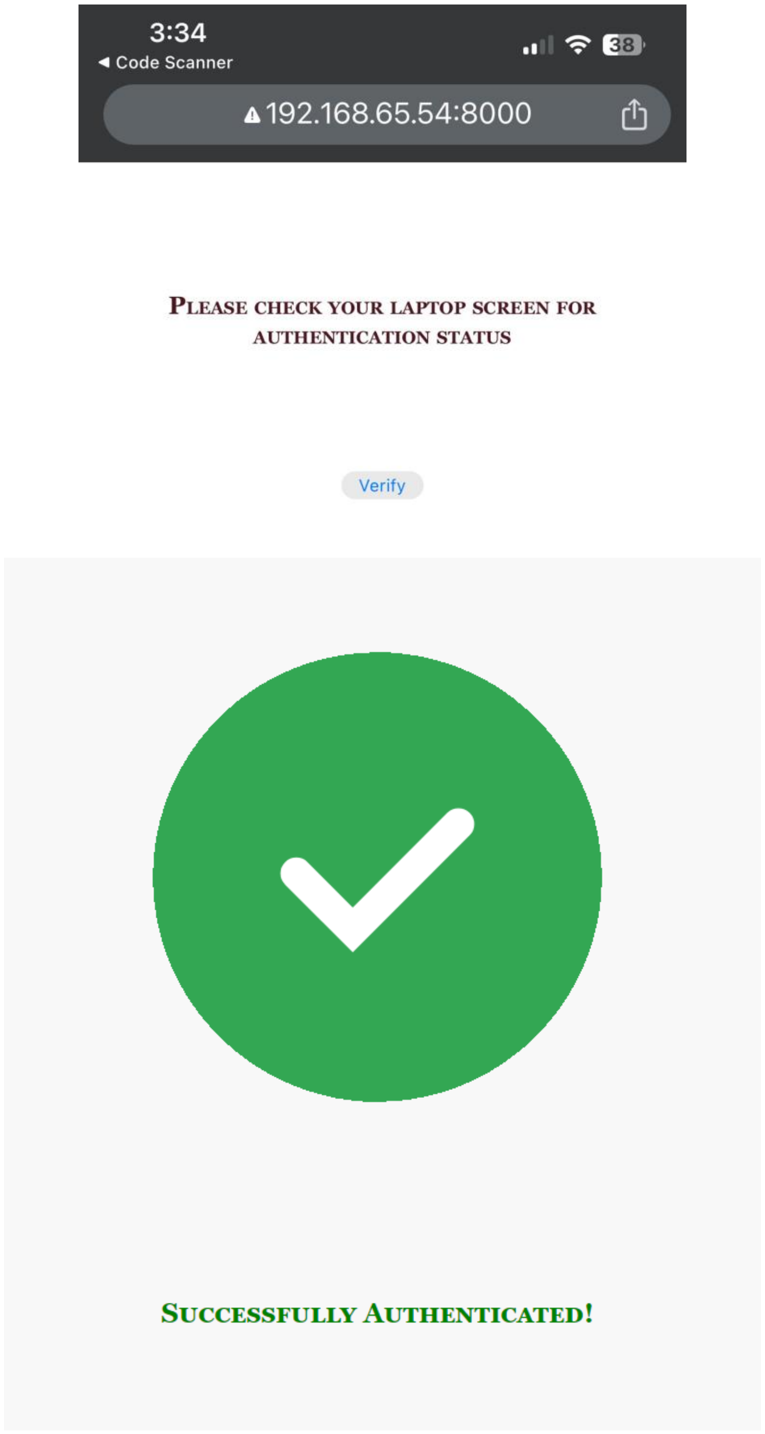


Log in

201525



SCAN THE BELOW QR CODE USING YOUR MOBILE PHONE



PLEASE CHECK YOUR LAPTOP SCREEN FOR AUTHENTICATION STATUS

SUCCESSFULLY AUTHENTICATED!

RESULTS AND CONCLUSION

The proposed three-layer authentication system is much more secure than the traditional login module, which can be supported with results analysis. Here's why:

- The OTP is one-timed and sent only to the mail id registered.
- Transmission of the mail uses TLS/SSL encryption and is very secure.
- The overall system design is relatively more complex, and would hence make it tough for a hacker to bypass.
- The QR module with the implementation of a time-based OTP ensures the absence of bots controlling the verification process.

We have thereby discussed the design, implementation, and testing of the proposed system, and evaluated its effectiveness in terms of security and usability. The results show that the proposed system provides a higher level of security than traditional login mechanisms and is more convenient than some existing multi-layered authentication systems.

The report highlights the importance of multi-layered authentication systems in enhancing security in modern computing systems. It is evident that with the increasing use of technology in various domains, the need for robust and reliable authentication mechanisms has become more critical than ever. Therefore, the proposed system can be implemented in various domains to prevent unauthorized access to sensitive information.

In conclusion, the three-layer authentication system involving login, OTP, and QR modules is a practical solution that provides enhanced security and convenience to users. Future research could focus on further improvements to the proposed system or the development of new authentication mechanisms that leverage emerging technologies such as biometrics, machine learning, and blockchain.

REFERENCES

[1]. Mishra, Gouri Sankar, et al. "User Authentication: A Three Level Password Authentication Mechanism." Journal of Physics: Conference Series. Vol. 1712. No. 1. IOP Publishing, 2020.

[2]. Christiana, Abikoye Oluwakemi, et al. "A secured one time password authentication technique using (3, 3) visual cryptography scheme." journal of physics: conference series. Vol. 1299. No. 1. IOP Publishing, 2019.

[3]. Chen, W., & Liu, W. (2017). A Three-Level Authentication System Based on Face Recognition and Password for Mobile Devices. International Journal of Smart Home, 11(3), 1-8.

[4]. Ali, S., Abbas, R. H., & Sher, A. (2019). A Three-Level Authentication Scheme for Cloud Computing Security. International Journal of Advanced Computer Science and Applications, 10(7), 258-262.

[5]. Das, S., & Chakraborty, S. (2018). A Three-Level Authentication Model for E-Banking System Security. International Journal of Engineering and Technology(UAE), 7(2.5), 183-189.

[6]. Khan, M. R., Hossain, M. A., & Islam, M. A. (2018). A Three-Level Authentication System for Internet of Things (IoT) Devices. International Journal of Computer Science and Information Security, 16(4), 37-44.

[7]. ElSisi, A. M., & ElNainay, H. M. (2017). A Three-Level Authentication System for Online Social Networks. International Journal of Computer Applications, 177(21), 25-30.

[8]. Kim, S. Y., Park, J. H., Lee, J. H., & Yoon, Y. J. (2015). Enhancing Security of Online Banking Transactions Using Three-Factor Authentication. Security and Communication Networks, 8(10), 1792-1801.

[9]. Al-Hujaily, M., Al-Mansour, S., Al-Jarallah, R., & Al-Muhtadi, J. (2019). A Three-Factor Authentication System Based on Biometric and Cryptographic Techniques. International Journal of Advanced Computer Science and Applications, 10(7), 110-116.

[10]. Jain, S., Singh, N., & Kumar, A. (2014). A Survey of Three-Factor Authentication Schemes. International Journal of Computer Applications, 93(9), 1-6.