|  |
| --- |
|  |
| Ministry of High Education,  Culture and Science City at Oct 6,  The High Institute of Computer Science & Information Systems  new image552.bmp  المعهد العالي لعلوم الحاسب ونظم المعلومات  **Graduation Project**  **Three Level Password System** |
| **Supervised by**  **Assoc. Prof. Dr. Salah Elden Shaban Essa Abdulrahman**  Assistant:  **Eng. Dalia Tourky**  **Project No: 12 Academic Year: 2020/2021**  Ministry of high education,  Culture and science city at Oct 6,  High Institute of Computer Science & Information Systems  new image552.bmp  المعهد العالي لعلوم الحاسب ونظم المعلومات  **Graduation Project**  **Three Level Password System**  Prepared by  41006- احمد محمد عبداللطيف حسن 41048- كريم عبدالتواب جمعه ابوالقاسم  41033- عبدالرحمن ماجد عبدالحميد محمد 41050- ماجد عبدالعزيز امين قطب  41043- عمرو احمد ابراهيم فتحى جبر 41059- محمد عبدالكريم عبدالمجيد سنجاب  **Supervised by**  **Assoc. Prof. Dr. Salah Elden Shaban Essa Abdulrahman**  **Assistant**  **Eng. Dalia Tourky**  **Project No: 12 Academic Year: 2020/2021** |
|  |

# Acknowledgment

First and foremost, all praises to Allah for giving us **the opportunity, blessing, and** strength to accomplish this project

We take this opportunity to express our profound gratitude and deep regards to our Supervisor, **prof. Salah Elden Shaban**, for his exemplary guidance, monitoring, and constant encouragement all over the project against all problems that faced us.

Moreover, we take this opportunity to thank **Eng. Dalia Tourky** for her guidance and valuable information that made which helped us through various stages.

we are thankful to all faculty members at the Higher Institute of Computer Science and Information Systems for the information provided by them throughout the years.

Finally, we must express our very profound gratitude to our families for providing us with unfailing support and continuous encouragement to accomplish this graduation project.

**xTeam**

**July 2021**

# Abstract

Back in the 1960s, **passwords** were **the first method** used to authenticate into computers more than fifty years ago, they have existed for a long time in the analog world but they immediately showed the industry some **problems** they’d need to solve to make them **more secure**. As more digital systems **leveraged** passwords for security, more **researchers and hackers** found ways to **abuse** **them** as well. And if the user used a strong long password for example he might be **at risk of forgetting it**. This led to the industry constantly looking for **increasingly** **secure** methods of authentication. One of the biggest risks with a normal, persistent password system is that if an attacker can guess, steal, or intercept your password, they could **replay** **it**.

To combat this, what if a user **another type** of password? What if the user can use **images** instead? That’s when **graphical passwords** came in handy, where a user could **authenticate by clicking photos** **in a preset order**, the idea seems great, where the user **can’t use easy to guess or weak passwords** anymore. But hackers would **still be able to guess** the password by trying all the **possible companions** (brute force attack). Again, we had to find a **more secure**, **robust**, **strong**, and usable way.

That’s when we thought of using **drawing** as a way of authentication, where the user has to **draw on a photo** of his choice, **free-handed** or **any kind** of drawing or **pattern**, and dose the same when logging in. This way the attacker has to try an **almost** **infinite number of patterns** before he could break in, and by making the drawing **invisible** on the **login page** it becomes **nearly immune** against **shoulder**-**surfing** attacks as well. **Of cores**, it is still possible to crack if the attacker got a very **lucky** guess, and the module **isn’t as easy to use as the first two**, but it defiantly provides **much higher security** than any other **knowledge-based** authentication form.

With that said, this project aims at **implanting all three modules** companiestogether in one **web-based** system, it also includes **storing passwords** with **salted** **hash**, a **user friendly** and easy to use interface, a function to **help users generate** **easy to remember** and **strong** text passwords, and most importantly **testing (hacking) module** for each of the three levels to **ensure the strength** of each against **most common** attacks.

# Table of Contents

[Acknowledgement 3](#_Toc75855221)

[Abstract 4](#_Toc75855222)

[Table of Contents 5](#_Toc75855223)

[List of Abbreviations 6](#_Toc75855224)

[List of Figures 7](#_Toc75855225)

[List of Tables 9](#_Toc75855226)

[1 Chapter One: Introduction 10](#_Toc75855227)

[1.1 Overview 10](#_Toc75855228)

[1.2 Project Motivation 10](#_Toc75855229)

[1.3 Project Contribution 10](#_Toc75855230)

[1.4 Project Scope 10](#_Toc75855231)

[1.5 Project Objectives 10](#_Toc75855232)

[1.6 Technologies Used 10](#_Toc75855233)

[1.7 Project Documentation Structure 10](#_Toc75855234)

[2 Chapter Two: Background Literature 10](#_Toc75855235)

[2.1 Introduction to Authentication. 10](#_Toc75855236)

[2.2 Authentication Strength. 11](#_Toc75855237)

[2.3 Single- Factor Authentication. 1](#_Toc75855238)2

[2.4 Multi-Factor Authentication. 14](#_Toc75855239)

[2.5 Authentication methods and protocols. 17](#_Toc75855240)

[2.6 Types of Security attack 21](#_Toc75855241)

[3 Chapter The Proposed System Architecture 25](#_Toc75855242)

[3.1 Proposed Architecture. 25](#_Toc75855243)

[3.2 System Components. 26](#_Toc75855245)

[3.3 System steps. 27](#_Toc75855245)

[3.4 System Operations. 29](#_Toc75855246)

[4 Chapter Four Security attack and Analysis 31](#_Toc75855247)

[4.1 Proposed Attacking Scenario. 31](#_Toc75855248)

[4.2 Components of the Attack. 32](#_Toc75855249)

[4.3 Attack steps. 34](#_Toc75855250)

[4.4 Attack operations. 35](#_Toc75855251)

[5 Chapter Five Implementation 39](#_Toc75855252)

[5.1 Front-end Implementation 39](#_Toc75855253)

[5.2 Back-end Implementation 43](#_Toc75855254)

[6 Chapter Six: Conclusion and future work: 53](#_Toc75855256)

[6.1 Conclusion: 53](#_Toc75855257)

[6.2 Future works: 53](#_Toc75855258)

[References 54](#_Toc75855259)

[Appendices 54](#_Toc75855260)

[Appendix A: 54](#_Toc75855261)

[Appendix B: 54](#_Toc75855262)

[ملخص المشروع باللغة العربية 55](#_Toc75855263)

# List of Abbreviations

|  |  |
| --- | --- |
| **Symbol** | **Term** |
| **SFA** | Single Factor Authentication |
| **2FA** | Two-Factor Authentication |
| **MFA** | Multi-Factor authentication |
| **PIL** | Pillow(python imaging library ) |
| **GUI** | Graphical User Interface |
| **HTML** | HyperText Markup Language |
| **CSS** | Cascading Style Sheets |
| **SQL** | Structured Query Language |

# List of Figures

[Figure ‎2.4‑1 Authentications factores 15](#_Toc47805851)

[Figure ‎2.6‑1 Credential stuffing attack 21](#_Toc47805851)

[Figure 2.6‑2 Man-In-The\_middle attack 22](#_Toc47805851)

[Figure 2.6‑3 Social Engineering life cycle 23](#_Toc47805851)

[Figure 3.1-1 The steps of Regestration 25](#_Toc47805851)­

[Figure 3.1‑2 The Steps of log-in 26](#_Toc47805851)

[Figure 3.2‑1 Steps of sign-up 27](#_Toc47805851)

[Figure 3.6‑3 Steps of log-in 28](#_Toc47805851)

[Figure 3.6‑3 steps of authentication 29](#_Toc47805851)

[Figure 4.4-1 Attack by word list 35](#_Toc47805851)

[Figure‎ 4.4-2 Generating password attack 36](#_Toc47805851)

[Figure ‎4.4‑3 Level Two Attack 37](#_Toc47805851)

[Figure ‎4.4‑4 Level Three Attack 38](#_Toc47805851)

[Figure ‎5.2‑1 Regetration interface 43](#_Toc47805851)

[Figure ‎5.2‑2 Password Generator interface 44](#_Toc47805851)

[Figure ‎5.2‑3 Log-in Interface level one 45](#_Toc47805851)

[Figure ‎5.2‑4 Sign-up Level Two interface a 46](#_Toc47805851)

[Figure ‎5.2‑5 level 2 upolding page 47](#_Toc47805851)

[Figure ‎5.2‑6 Sign-up level two interface b 48](#_Toc47805851)

[Figure ‎5.2‑7 log-in level two interface 49](#_Toc47805851)

[Figure ‎5.2‑8 uploding page level three 50](#_Toc47805851)

[Figure ‎5.2‑9 sign-up interface level three 51](#_Toc47805851)

[Figure ‎5.2‑10 log-in interface level three 52](#_Toc47805851)

# Chapter One: Introduction

## Overview

This project is a **web application** with **three levels** of password-based authentication; each of them is **a knowledge-based** module where the uses need to know the password of each level to prudes to the next.

This project also includes **testing** all the modules against **the** **most** **common attacks** and calculating the **average time needed to crack** into each level.

## Project Motivation

Since the early day of authentication, **passwords** were the **easiest** and **most commonly used** option; because it was **fast**, **simple** and **very flexible**.

When people started to have more accounts with the growth of technology and automation problems like when users can’t remember hard passwords for each account so they started using either **very easy to remember** or **short words** (like password1, 12345678), or they will simply **reuse the same password** in all of their accounts.

Which makes hacking using **a word lest attack** or **brute force** short passwords **very easy** and **fast,** especially nowadays with the growth of processing power.

## Project Contribution

This project gives **more secure models** rather than the old **classic text password**, and it’s all knowledge-based as well.

As if one of the ways might **replace** the text-based normal password for good.

## Project Scope

This project aims to make a **secure and user-friendly authentication** **system** only using **knowledge**-**based** factors and **finds more optimal and better ways** to authenticate a user than a text password.

## Project Objectives

1. Build the system with a **useable and friendly GUI.**
2. **Test and try** all three models of knowledge base passwords and see if any of them could be better and more secure than a good text password.
3. **Test the module's security** against some of the most common attacks tools and methods.
4. Make authentication more secure but also not very hard and complicated for the average user.

## Technologies Used

1. **Flask**: a web framework in python used to develop the back-end.
2. **HTML**: used to build the templates for the front-end.
3. **CSS** and **bootstrap**: used to decorate and make the UI more appealing for the user.
4. **Python**: used in the backend as the main programing language and testing as well.
5. **SQLite**: to make and manage the database.
6. **Git** and **Github**: to keep a good synchronization as we all work on the same project and as a backup in case of any problem.

## Project Documentation Structure

1. **Chapter one**: introduction
2. **Chapter two:** Background Literature
3. **Chapter three**: The Proposed System Architecture
4. **chapter four**: Security attack and Analysis
5. **chapter five**: Implementation
6. **chapter six**: Conclusion and future work

# Chapter Two: Background Literature

## Introduction to Authentication.

**Authentication** is the act of [**proving**](https://en.wikipedia.org/wiki/Proof_(truth)) **an**[**assertion**](https://en.wikipedia.org/wiki/Logical_assertion), such as the [**identity**](https://en.wikipedia.org/wiki/Digital_identity) of a computer system user. In contrast with [**identification**](https://en.wikipedia.org/wiki/Identity_(philosophy)), the act of indicating a person or thing's identity, authentication is the process of verifying that identity.[1] It might involve **validating** **personal** [**identity** documents](https://en.wikipedia.org/wiki/Identity_document), verifying the authenticity of a [**website**](https://en.wikipedia.org/wiki/Website) with a [**digital** **certificate**](https://en.wikipedia.org/wiki/Public_key_certificate),[2] determining the age of an artifact by [carbon dating](https://en.wikipedia.org/wiki/Carbon_dating), or ensuring that a product or document is not [counterfeit](https://en.wikipedia.org/wiki/Counterfeit).

Authentication is relevant to multiple fields. In art, antiques, and anthropology, a common problem is verifying that a given artifact was produced by a certain person or in a certain place or period of history. In computer science, verifying a user's identity is often required to allow access to confidential data or systems. [3]

Authentication can be considered to be of three types:

1. **Knowledge-based authentication**: is accepting proof of identity given by a credible person who has first-hand evidence that the identity is genuine. Like a password (pica of data).
2. **Ownership-based authentication**: is comparing the attributes of the object itself to what is known about objects of that origin. For example, an art expert might look for similarities in the style of painting, check the location and form of a signature, or compare the object to an old photograph.
3. **Biometric-based authentication:** is a security process that relies on the unique biological characteristics of an individual. Biometric authentication technologies are used by consumers, governments, and private corporations including airports, military bases, and national borders. Common biometric authentication methods include:
   1. **Facial recognition**
   2. **Fingerprint scanners**
   3. **Voice identification**
   4. **Eye scanners**

## Authentication Strength.

Security questions: Security questions have traditionally been used for password resets. They are simple to set up, but they can be hacked or stolen very easily.

One-time passwords (OTPs): They are more secure than security questions as they use a secondary authentication category. The user has a device and their password. Verification codes or OTPs sent via SMS are also convenient.

App-generated codes: A software-based OTP uses the time-based one-time password algorithm (TOTP) presented via a third-party app.

Specialized authentication apps: Rather than providing the user with an OTP, this requires users to verify their identity by interacting with the app on their smartphone. The authentication token is then sent to the service directly, strengthening security by eliminating the need for a user-entered OTP.

Physical authentication keys: The authentication process is secured by an asymmetric encryption algorithm where the private key never leaves the device. USBs that are plugged in when prompted and smart cards that users swipe are examples.

Biometrics: This is tough to hack, but no method is perfect, and biometrics comes with challenges and privacy concerns.

Like passwords, biometric data must be stored in some form of database, which could be compromised.

Cryptographic challenge-response protocol: A database sends a challenge to another, and the recipient must respond with the appropriate answer. All the communication is encrypted during transmission, so it can't be hacked or manipulated.

**Recommendations For Strong Authentication ( basic security measures must be covered:**

-Enforce a dictionary check to ensure that users cannot choose common words for their password.

-Require a strong username that includes a numeric character.

-Limit the number of failed login attempts to three and temporarily suspend account access.

-Use SSL to create an encrypted link between your server and the user’s web browser during account enrolment.

## 

## Single-Factor Authentication.

It is a method of logging users into resources by having them present only one way of verifying their identity. Username and password is the dominant form of SFA. Also, it is a process for securing access to a given system, such as a network or website, that identifies the party requesting access through only one category of credentials.

The most common ones include a password-based authentication system. The user who sets his account needs to create a unique and robust password to ensure no one can access the data behind it.

**Single Factor Authentication Types**

It has three factors on which it sets up authentication, these factors are:

**1-Something You Know**

is the most used factor of SFA. It includes a PIN and password with a username. One should set a strong and unique PIN/password. Strong Password is a mixture of lowercase, uppercase, and special characters.

**2-Something You Have**

It involves any smart card, token, credit card, and identity card. This factor is applicable where there are smart card readers or embedded identity certificates in other cards.

**3-Something You Are**

The most common method of this factor is a biometric method. The strongest method of authentication. It does not get leaked, hacked, or stolen. But this may not work if the system/scanner does not perform well.

**Single-factor authentication examples or applications include:**

-Signing in to two different apps using a password

-WhatsApp PIN verification

-Biometric systems

-Smart cards

-Token generation system

**Single Factor Authentication advantages and disadvantages :**

Advantages :

**-Simplicity**: It is the simplest authentication process.

-Easy to Use: SFA is easy as you only have to clear one process

**-Less Time:** Single-factor reduces the time consumption.

**-Independent**: It does not need user cooperation.

**-Strong**: This is a sound method of authentication

**-Non-troublesome**: It makes the process hassle-free.

RDP: it is defensive against RDP.

Disadvantages :

-Data Breaching: Research shows that data breaching is high in SFA.

-Leak: It can leak passwords. Someone can steal the password using a keylogger or screen capture.

-Hack Weak password results in hacking. They then used these passwords for phishing.

-Blackmailing: Once someone stole the data, he/she can blackmail you.

-Security: It is less secure.

-Access to Different Apps: Reusing passwords is dangerous, as a hacker can quickly access data of other applications as well

## Multi-Factor Authentication.

It is a security technology that requires multiple methods of authentication from independent categories of credentials to verify a user's identity for a login or other transaction.

The goal of MFA is to create a layered defense that makes it more difficult for an unauthorized person to access a target, such as a physical location, computing device, network, or database. If one factor is compromised or broken, the attacker still has at least one or more barriers to breach before successfully breaking into the target.



Figure ‎2‑1 multi-factor authentication

How it works :

MFA works by requiring additional verification information (factors). One of the most common MFA factors that users encounter is one-time passwords (OTP). OTPs are those 4-8 digit codes that you often receive via email, SMS, or some sort of mobile app. With OTPs a new code is generated periodically or each time an authentication request is submitted. The code is generated based upon a seed value that is assigned to the user when they first register and some other factor which could simply be a counter that is incremented or a time value.

The different types of multi-factor authentication technologies :

**Biometric authentication** :

They are a form of authentication that accurately and securely authenticate users through their mobile devices. The most common biometric modalities are fingerprint scans and face recognition. Biometric authentication also includes behavioral biometrics, which provides an invisible layer of security by continuously authenticating an individual based on the unique ways they interact with their computer or mobile device: keystrokes, swipe pattern, and mouse movements.

**Hardware tokens**:

They are small, easy-to-use devices that an owner carries to authorize access to a network service. By supporting strong authentication with one-time passcodes (OTPs), the physical tokens provide a possession factor for multi-factor authentication while enabling enhanced security for banks and application providers that need to secure multiple applications with a single device.

**Mobile authentication**:

It is the process of verifying a user via their Android or iOS device or verifying the device itself. This technology allows users to log in to secure locations and access resources from anywhere with enhanced security.

**Out-of-band authentication**:

This authentication type requires a secondary verification method through a separate communication channel, typically the person’s Internet connection and the wireless network on which their mobile phone operates.

**Advantages :**

-Improved security

Multi-factor authentication provides increased security over static passwords and single-factor authentication processes.

-Regulatory compliance

Multi-factor authentication can help organizations comply with their industry regulations. For example, MFA is necessary to satisfy the strong authentication requirement of PSD2 for Strong Customer Authentication (SCA).

-Improved user experience

Breaking the reliance on passwords can improve the customer experience. By focusing on low-friction authentication challenges, organizations can increase security and improve the user experience.

**Disadvantages :**

-a phone is needed to get a text message code. they can get lost or stolen;

-hardware tokens can get lost or stolen;

-the biometric data calculated by MFA algorithms for personal IDs, such as thumbprints, are not always accurate.

-MFA verification can fail if there is a network or internet.

## Authentication methods and protocols.

Cybercriminals always improve their attacks. This is why companies are starting to implement more incident response strategies, some common authentication methods used to secure modern systems:

1**. Password-based authentication**

Passwords are the most common method of authentication. Passwords can be in the form of a string of letters, numbers, or special characters. To protect yourself you need to create strong passwords that include a combination of all possible options.

The bottom line is that passwords have a lot of weaknesses and are not sufficient in protecting online information. Hackers can easily guess user credentials by running through all possible combinations until they find a match.

2. **Multi-factor authentication**

Multi-Factor Authentication (MFA) is an authentication method that requires two or more independent ways to identify a user. Examples include codes generated from the user’s smartphone, Captcha tests, fingerprints, or facial recognition.

It increases the confidence of users by adding multiple layers of security. MFA may be a good defense against most account hacks, but it has its pitfalls. People may lose their phones or SIM cards and not be able to generate an authentication code.

3. **Certificate-based authentication**

Certificate-based authentication technologies identify users, machines, or devices by using digital certificates. A digital certificate is an electronic document based on the idea of a driver’s license or a passport.

The certificate contains the digital identity of a user including a public key, and the digital signature of a certification authority. Digital certificates prove the ownership of a public key and are issued only by a certification authority.

4. **Biometric authentication**

Biometrics authentication is a security process that relies on the unique biological characteristics of an individual.

Common biometric authentication methods include:

-Facial recognition—matches the different face characteristics of an individual trying to gain access to an approved face stored in a database.

-Fingerprint scanners—match the unique patterns on an individual’s fingerprints. They are currently the most popular biometric technology for everyday consumers, despite their frequent inaccuracies.

-Voice identification—examines a speaker’s speech patterns for the formation of specific shapes and sound qualities.

-Eye scanners—include technologies like iris recognition and retina scanners. Iris scanners project a bright light towards the eye and search for unique patterns. The patterns are then compared to approved information stored in a database.

-Token-based authentication

Token-based authentication technologies enable users to enter their credentials once and receive a unique encrypted string of random characters in exchange. You can then use the token to access protected systems instead of entering your credentials all over again. The digital token proves that you already have access permission.

The most common authentication protocols :

Authentication protocols are the designated rules for interaction and verification that endpoints for systems use to communicate. as many different applications that users need access to, there are just as many standards and protocols.

Password Authentication Protocol (PAP) :

PAP is the least secure protocol for validating users, due mostly to its lack of encryption. It is essentially a routine log-in process that requires a username and password combination to access a given system, which validates the provided credentials.

Challenge Handshake Authentication Protocol (CHAP):

CHAP is an identity verification protocol that verifies a user to a given network with a higher standard of encryption using a three-way exchange of a “secret.” First, the local router sends a “challenge” to the remote host, which then sends a response with an MD5 hash function. The router matches against its expected response (hash value) and depending on whether the router determines a match, it establishes an authenticated connection—the “handshake”—or denies access.

Extensible Authentication Protocol (EAP) :

This protocol supports many types of authentication, from one-time passwords to smart cards. When used for wireless communications, EAP is the highest level of security as it allows a given access point and remote device to perform mutual authentication with built-in encryption. It connects users to the access point that requests credentials, confirms identity via an authentication server, and then makes another request for an additional form of user identification to again confirm via the server—completing the process with all messages transmitted, encrypted.

## 

## Types of Security attack

A security attack is an unauthorized attempt to steal, damage, or expose data from an information system such as your website. Malicious hackers can go about this in a variety of ways:

**Credential Stuffing:**

It is a cyber-attack method in which attackers use lists of compromised user credentials to breach a system. The attack uses bots for automation and scale and is based on the assumption that many users reuse usernames and passwords across multiple services.

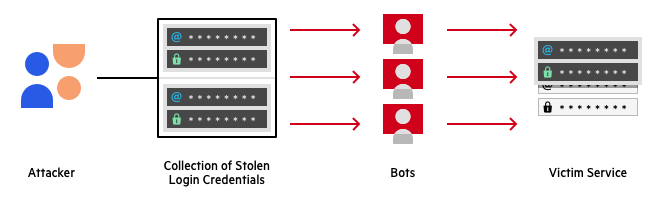


Figure ‎2‑2 shows Credential stuffing

**Drive-by downloads:**

A drive-by download is a method of distributing malware and occurs when a malicious script is inserted into a page’s PHP or HTTP. When a person visits the infected site, the malware is downloaded onto and silently infects, the device.

These threats can be tricky because they are not attributed to human error. You could visit a seemingly legitimate site, unaware it’s been compromised.

**Man-In-The-Middle (MITM) attacks:**

With MITM attacks, the criminal positions themselves between your device and the server. They eavesdrop on, intercept, and manipulate communication between two parties – this often happens on unsecured wireless networks such as public WiFi.

Detection of these attacks is difficult, but prevention is possible. Always use secure WiFi connections, and consider investing in a Virtual Private Network (VPN).

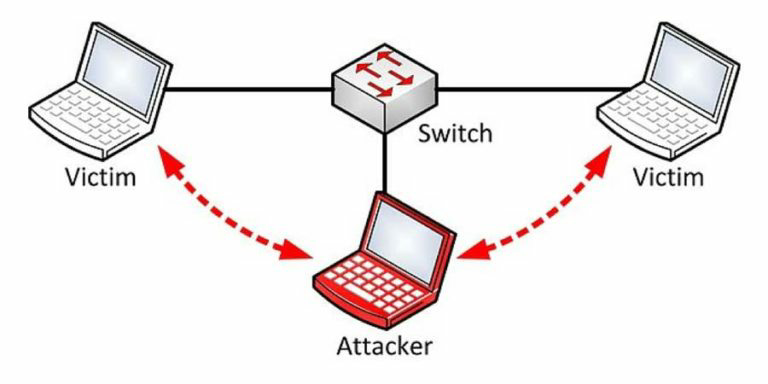


Figure ‎2.6‑2 shows Man in the middle attack

**An eavesdropping attack:**

It occurs when a hacker intercepts, deletes, or modifies data that is transmitted between two devices. Eavesdropping, also known as sniffing or snooping, relies on unsecured network communications to access data in transit between devices.

How it works :

cybercriminals must take advantage of a weak network connection that will allow them to transfer network signals to themselves. This is done by installing network monitoring software (sniffers) either on a PC or server, that handles the attack and catches the data being transferred. Wi-fi hotspots and websites that do not run over HTTPS are the most common examples of weak or insecure networks that are vulnerable to eavesdropping.

**Social Engineering:**

It is the term used for a broad range of malicious activities accomplished through human interactions. It uses psychological manipulation to trick users into making security mistakes or giving away sensitive information.

Social engineering attacks happen in one or more steps. A perpetrator first investigates the intended victim to gather necessary background information, such as potential points of entry and weak security protocols, needed to proceed with the attack. Then, the attacker moves to gain the victim’s trust and provide stimuli for subsequent actions that break security practices, such as revealing sensitive information or granting access to critical resources.

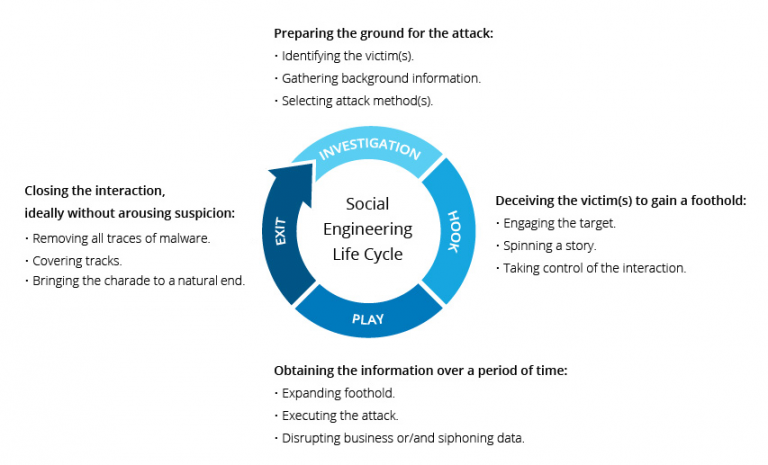


Figure ‎2.6‑3 shows Soical engineering

Types of social engineering attacks

Baiting: An attacker leaves a malware-infected physical device. The target then picks up the device and inserts it into their computer, unintentionally installing the malware.

Phishing: When a malicious party sends a fraudulent email disguised as a legitimate email. The message is meant to trick the recipient into sharing financial or personal information or clicking on a link that installs malware.

Spear phishing: the attack is tailored for a specific individual or organization.

Vishing: it involves the use of social engineering over the phone to gather financial or personal information from the target.

Whaling: it targets high-profile employees, such as the chief financial officer to trick the targeted employee into disclosing sensitive information.

# Chapter three: The Proposed System Architecture

## Proposed Architecture

The system made of three main components:

1. The front-end templets
2. The back-end framework
3. The Dara Base

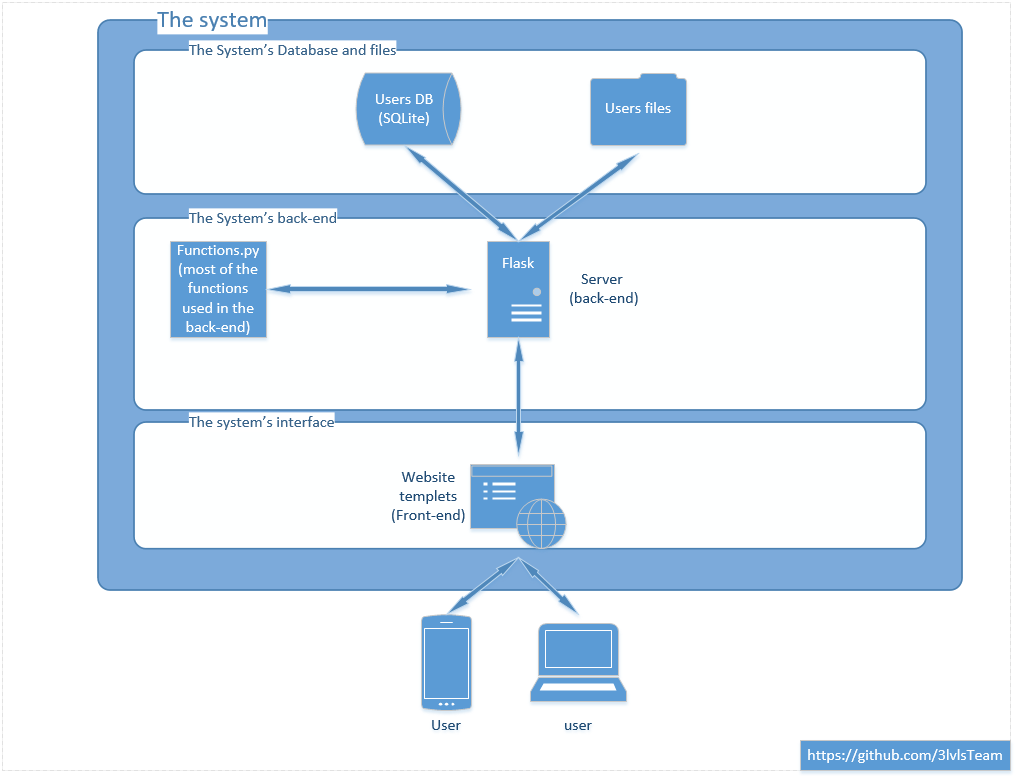


Figure 3.1-1 System’s architecture diagram

In figure 3.1-1, it shows the general architecture of the system and how its component connected.

## System Components

|  |  |
| --- | --- |
| **Flask** | is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries.[5] |
| **RE** | A regular expression (or RE) specifies a set of strings that matches it; the functions in this module let you check if a particular string matches a given regular expression (or if a given regular expression matches a particular string, which comes down to the same thing). |
| **Bcrypt** | is a password-hashing function designed by Niels Provos and David Mazières, based on the Blowfish cipher and presented at USENIX in 1999.[6] Besides incorporating a salt to protect against rainbow table attacks, bcrypt is an adaptive function: over time, the iteration count can be increased to make it slower, so it remains resistant to brute-force search attacks even with increasing computation power. |
| **Datetime** | is a python library used to calculate the age of the users dinamctly and to store the user's action’s time. |
| **SQLAlchemy** | is an open-source SQL toolkit and object-relational mapper (ORM) for the Python programming language released under the MIT License.[7] SQLAlchemy's philosophy is that relational databases behave less like object collections as the scale gets larger and performance starts being a concern, while object collections behave less like tables and rows as more abstraction is designed into them. |
| **PIL** | Python Imaging Library is a free and open-source additional library for the Python programming language that adds support for opening, manipulating, and saving many different image file formats. It is available for Windows, Mac OS X , and Linux. The latest version of PIL is 1.1.7, which was released in September 2009.[8] |
| **Imagehash** | Image hashes tell whether two images look nearly identical. This is different from cryptographic hashing algorithms (like MD5, SHA-1) where tiny changes in the image give completely different hashes. In image fingerprinting, we want our similar inputs to have similar output hashes as well.[9] |
| **Ping3** | Ping3 is a pure python3 version of ICMP ping implementation using raw socket.[10] |
| **random** | is a built-in module to generate the pseudo-random variables. It can be used to perform some action randomly such as getting a random number, selecting random elements from a list, shuffle elements randomly, etc. |
| **OS** | The OS module in Python provides functions for interacting with the operating system. OS comes under Python’s standard utility modules. This module provides a portable way of using operating system-dependent functionality. The \*os\* and \*os.path\* modules include many functions to interact with the file system. |
| **Urlib.request** | is a Python module for fetching URLs (Uniform Resource Locators). It offers a very simple interface, in the form of the urlopen function. This is capable of fetching URLs using a variety of different protocols. It also offers a slightly more complex interface for handling common situations - like basic authentication, cookies, proxies, and so on. These are provided by objects called handlers and openers. |
| **Bootstrap** | is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components. |
| **SQLite** | is a relational database management system (RDBMS) contained in a C library. In contrast to many other database management systems, SQLite is not a client-server database engine. Rather, it is embedded into the end program. |
| **HTML** | is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. |
| **CSS** | **Cascading Style Sheets** is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.[11] CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript |
| **Javascript** | is a programming language that conforms to the ECMAScript specification.[12] JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object orientation, and first-class functions. |

## System steps

#### 3.3.1 Sign-up :

3.3.1.1 sign-up level one:

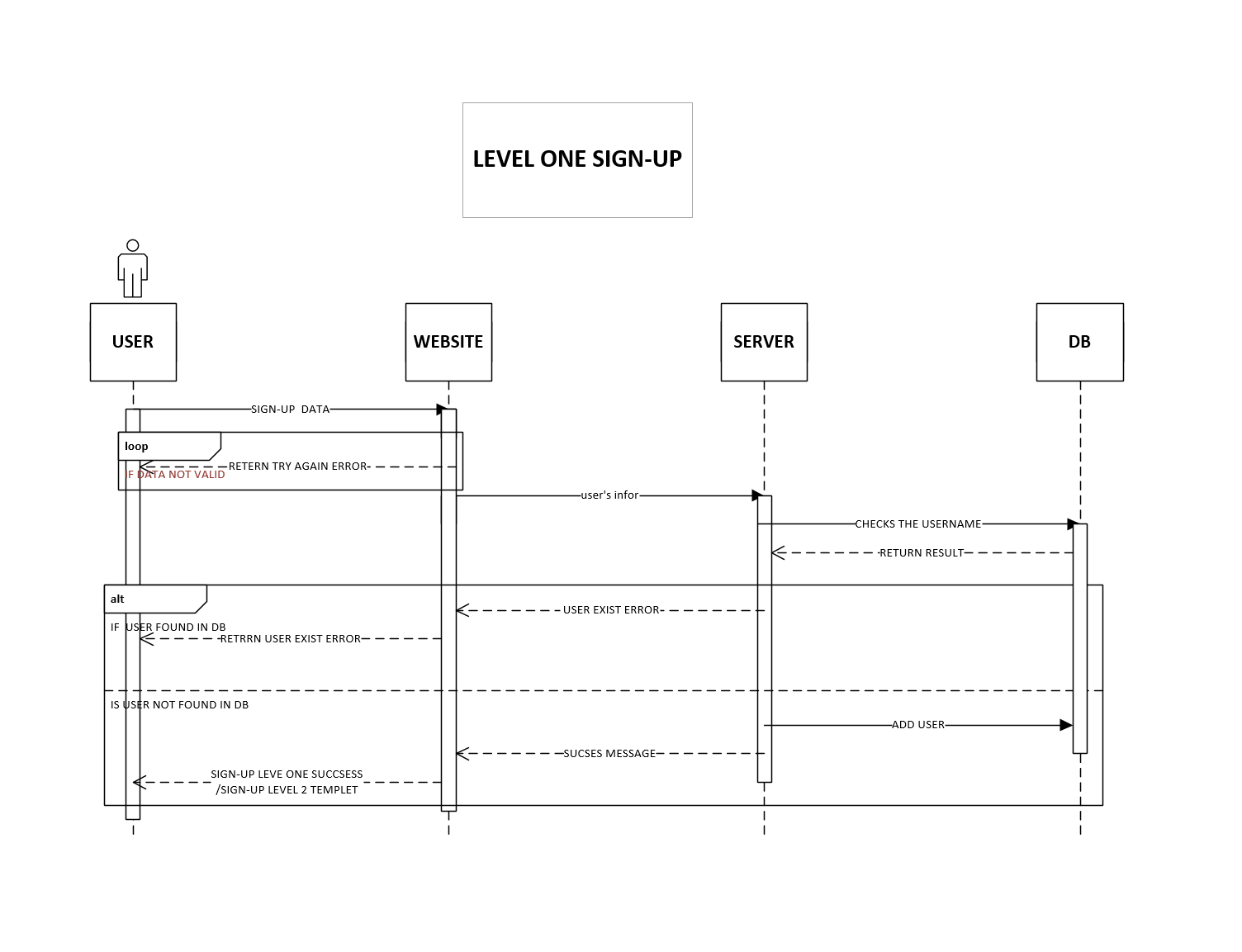


Figure 3.3-1 system steps

-First the user starts at the sign up page at level one, then the user is asked to enter some personal data like name, age, email, username, and then picks a password; then the website **checks** if the data he entered had any **syntax errors**, if it found any it ask the user to change it.

-Then the server takes the data and checks first if the username or the email were **used before** in the database, then if the password is strong or not using the **Information Technology Services (ITS) policy** then hash it using the **bcrypt** tool.

-After that the server finally **stores** the user’s data in the database and direct the user to the next page (sign-up level two).

3.3.1.2 sign-up level two:

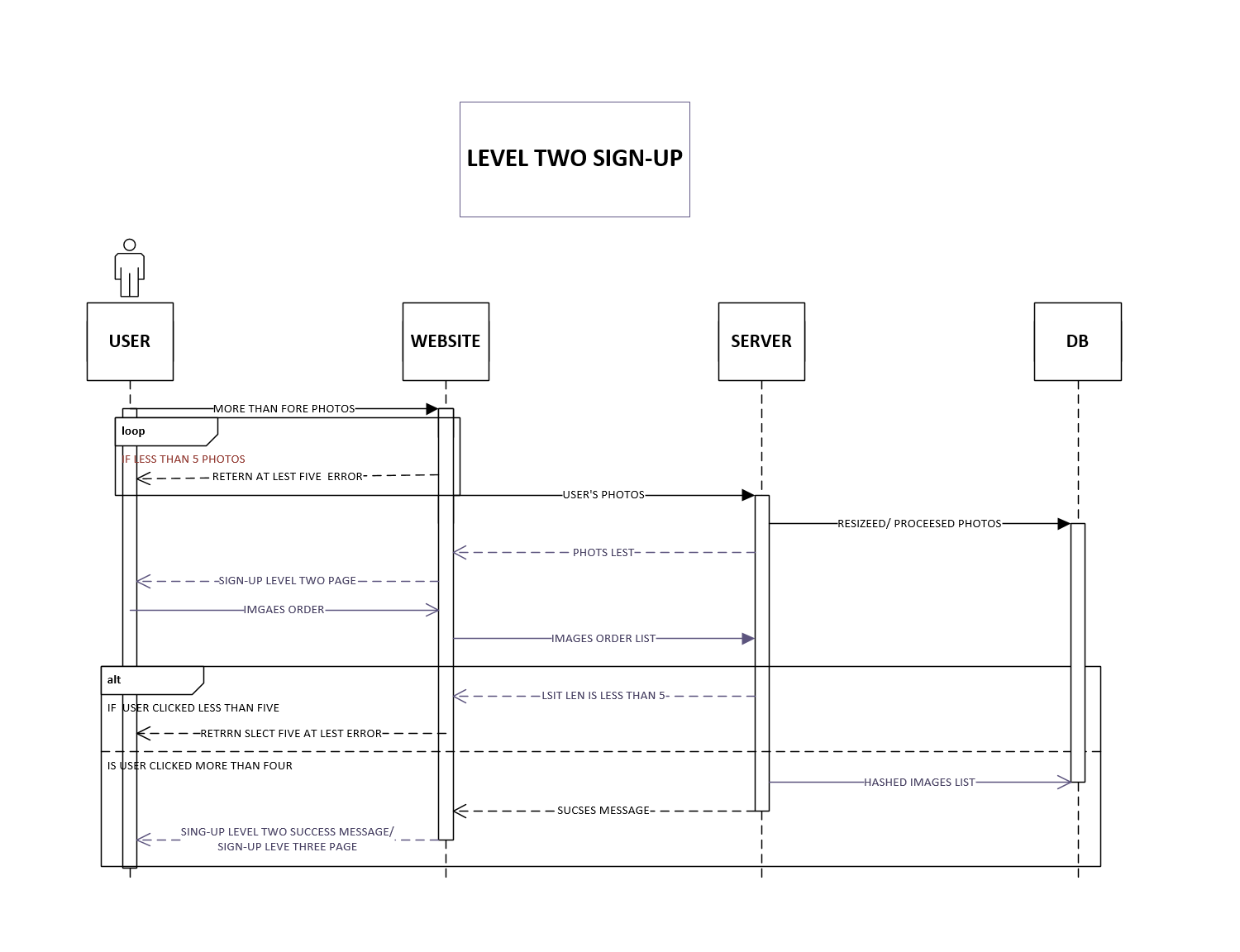


Figure 3.3-2 system steps

-After user is finished with sign-up level one, level two is opened where he **upholds** **at least five images** and then order them in any way he likes.

-If the number of images is **less than five**, the server will return an **error**.

-Then the website sends the photo where the server will **resize**, **presses**, **rename** then **store** the images in the **user-id directory**.

-After that the user will **click** on the images in **any oared** he like and the site will **create** the password from **images names and their order** then send it to the back-end, where it will be **hashed and stored** in the databased as a **text**.

-If he clicked on **less** **than five** images he gets an **error** as well from the server.

3.3.1.3 sign-up level three:

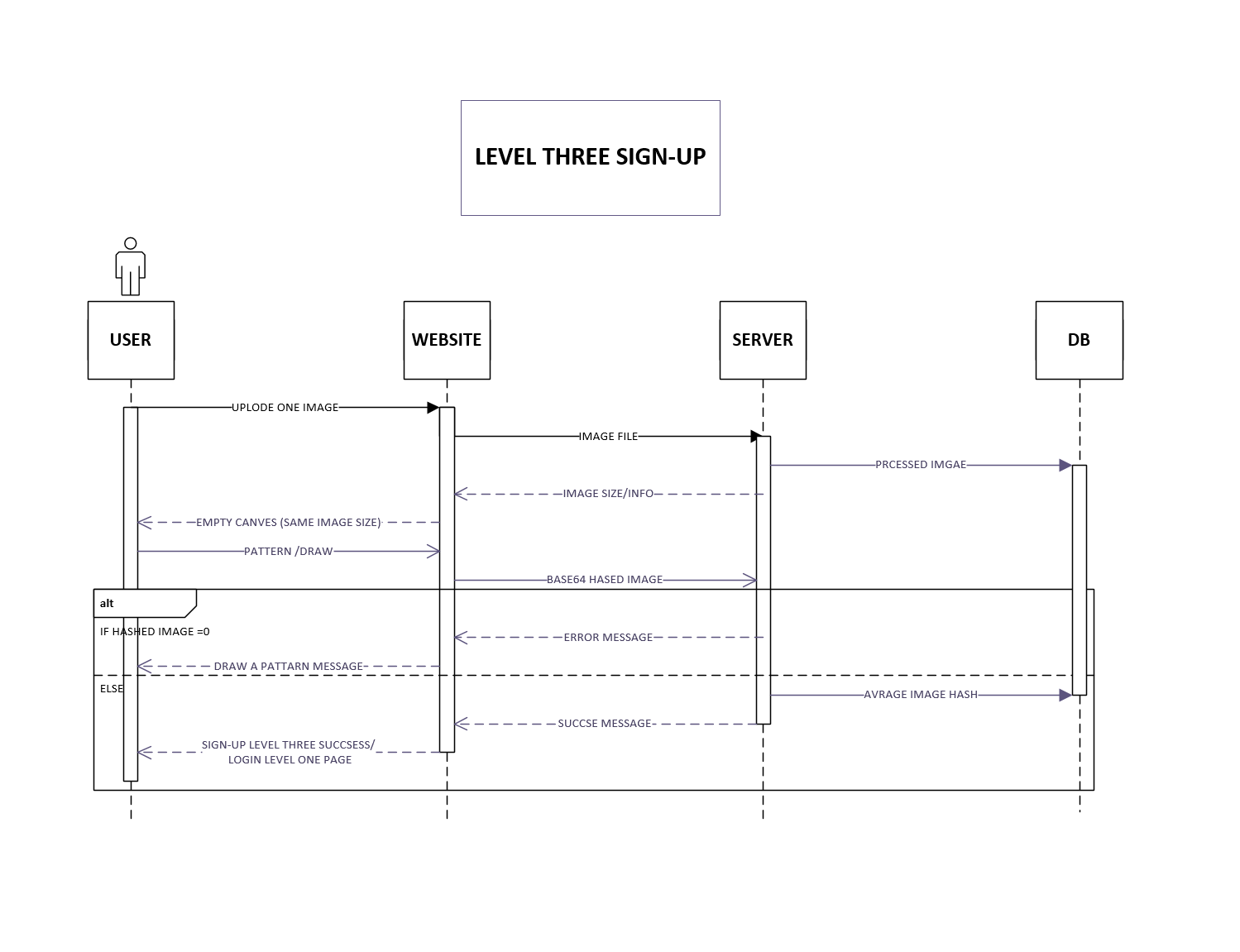


Figure 3.3-3 system steps

-After the user done with level one and two, he gets redirect to page three where he will upload another image.

-The image then gets resized, processed and renamed as well as level two.

-The site makes transparent canvas on top of the image where the user will draw a pattern he will use to log-in.

-The site then sends the canvas transparent image to the back-end using **base64** encryption where the server will use **imagehash** to get the **average hash** of the image and the store is in the database as string. (NOTE: **only string could be sorted in the database not hash values, so we convert the average hash value into string then store it**.)

-If the canvas were sent completely empty the server will return an error to the site

3.3.2.1 Log-in level one:

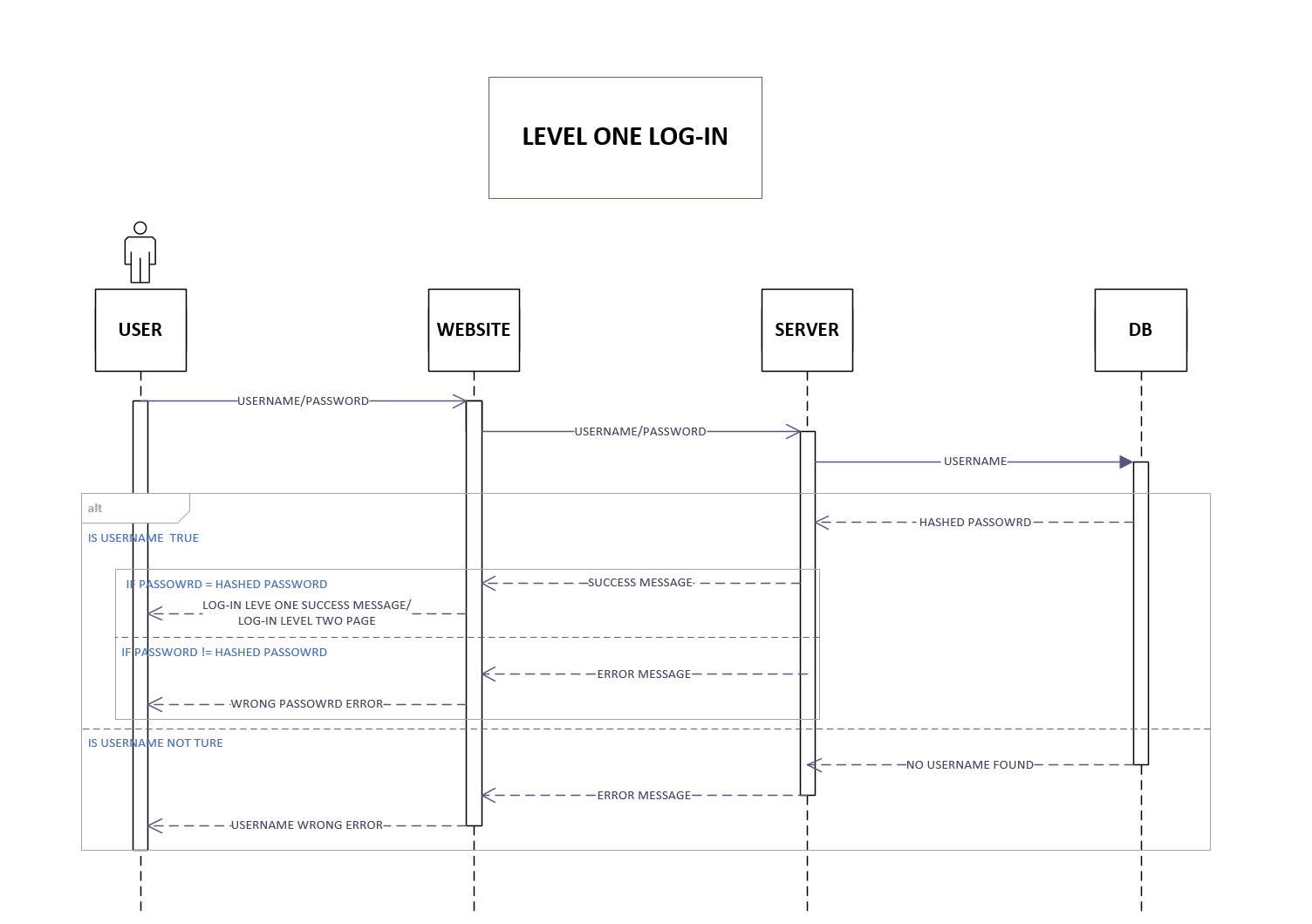


Figure 3.3-4 system steps

-After the user done with all three levels in the sign-up process, it’s time to **log in**.

-First user has to enter a **username** and a **password**, the site sends them to the backend and the server cheeks of the **username exists** in the database and if not it reruns an error to the site.

-After than the server calls the **hashed password** from the database then **compare it** using **checkpw()**  from **bcrypt.**

-If the user name and the password matches, the server redirect the user to the next level.

-If not, then he will stay in this level and he won’t be able to go anywhere else

3.3.2.2 Log-in level two:

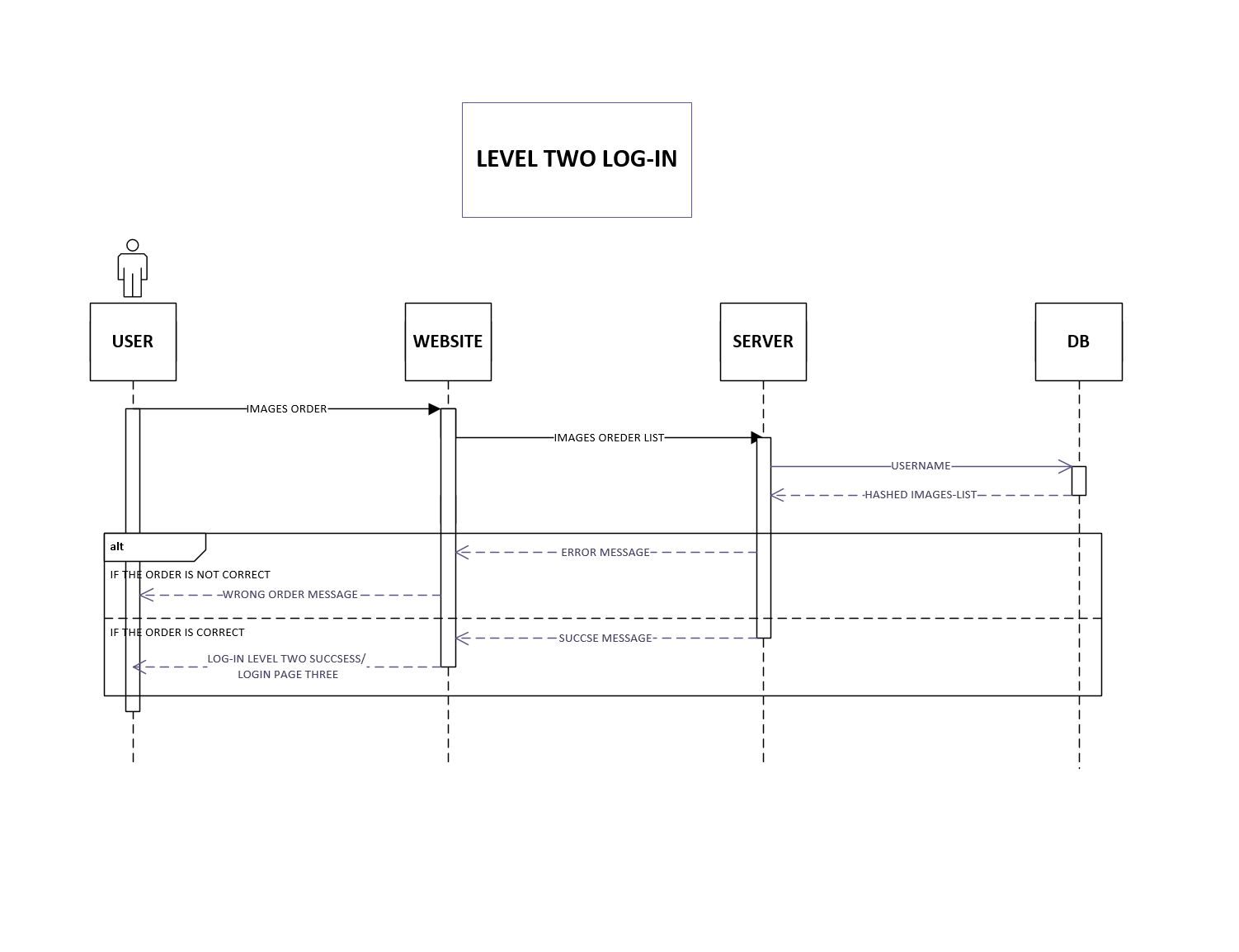


Figure 3.3-5 system steps

-After done logging in level one, the server redirects the user to level two, where he gets the images from the **user**-**id** directory, and he has to click them in the same order as he did in sign-up level two.

-If the order is wrong the server returns an error.

-If the order is correct the server redirect the user to level three.

3.3.2.3 Log-in level three:

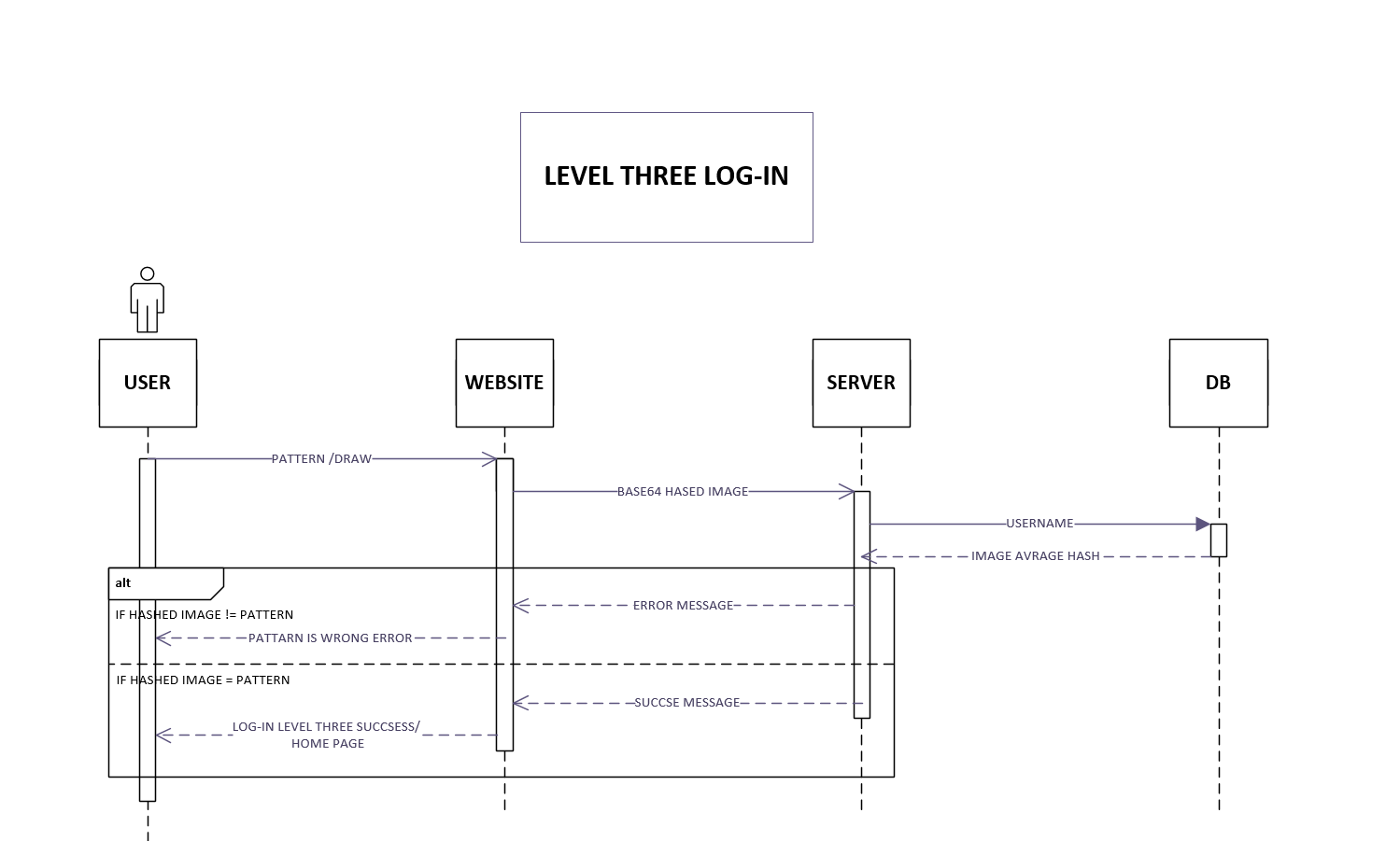


Figure 3.3-6 system steps

-After level one and two where passed, the server will redirect the user to level three.

-The user has to **draw** the same **pattern** as in sign-up.

-Then site sends the **base64** image to the server where the **average** **hash** will be calculated.

-Then the server calls the average hash from the database and compare the similarity by **subtracting** both hash values after **decode** it from string to hash again suing **hex\_to\_hash()** from **imagehash.**

-if the outcome is **more** than 1 then the photos are **not** **similar** and the server sends error message to the site.

-If the outcome is **less** than 1 then the photos **are** **similar** and the user is **authenticated** and redirected to the home page.

## System operations

**3.4.1 sign-up activity:**

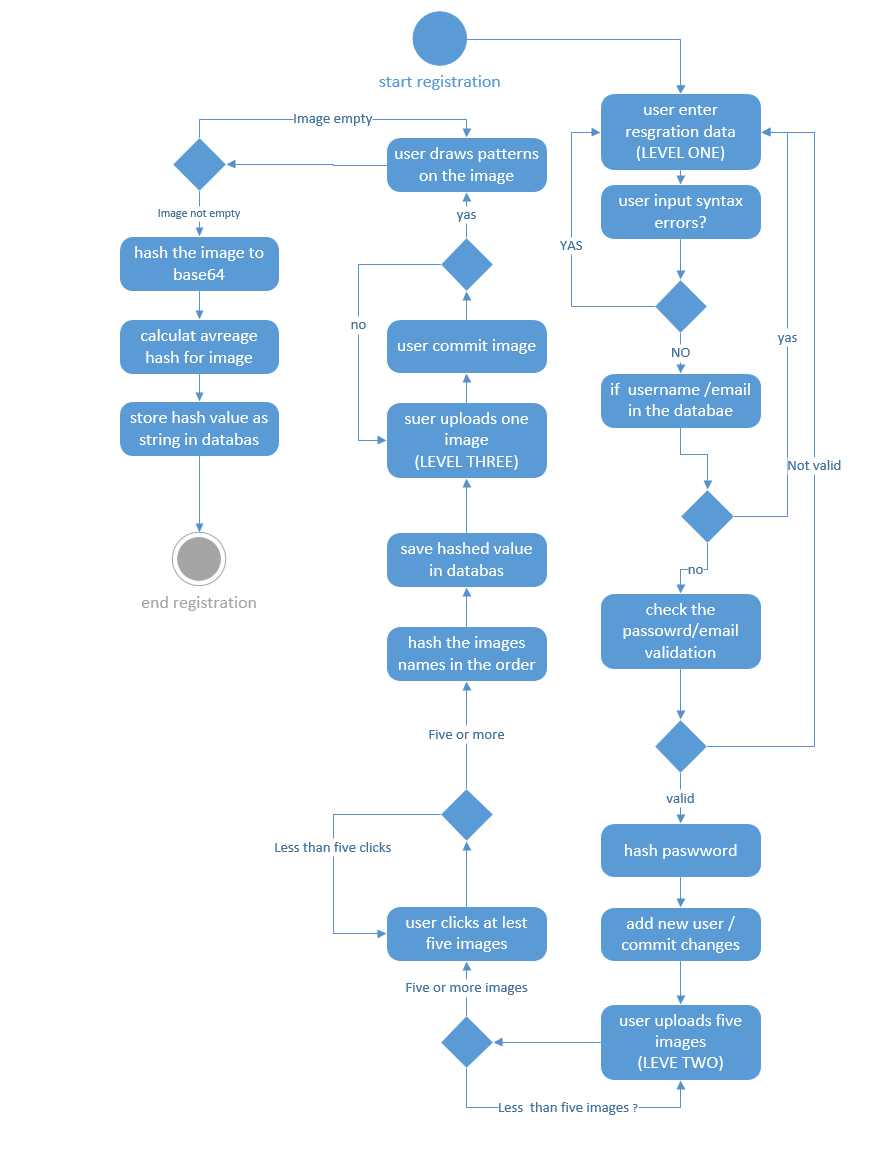
****

Figure 3.4-1 system operations in sign-up

**3.4.2 log-in activity:**

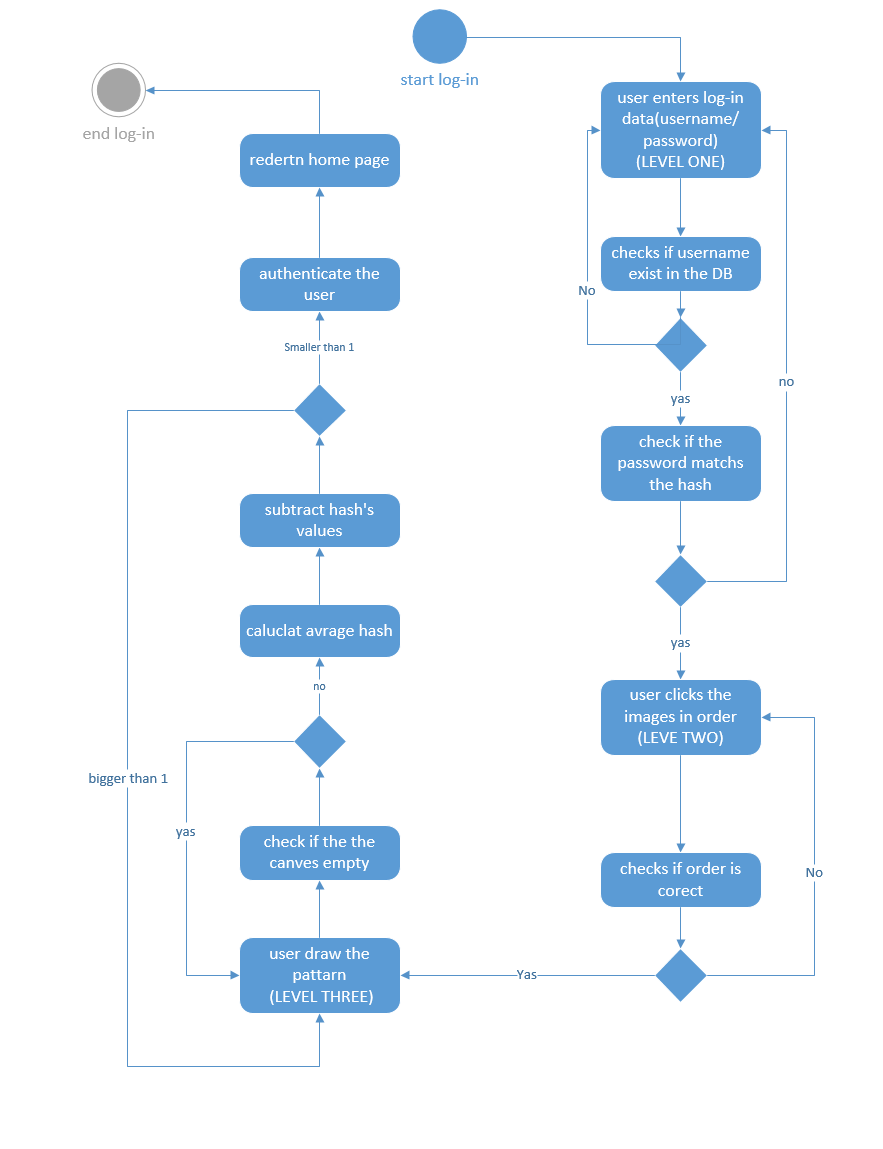


Figure 3.4-2 system operations in log-in

# Chapter Four: Security attacks and Analysis

## Proposed Attacking Scenario.

Level one attacking scenario:

The written password could be breach throw multiple ways:

1. Wordlist attack
2. Brute force attack
3. Dictionary attack
4. Social engineering

**1.** **Wordlist attack:**

When the attacker knows the user id or email and tries different password using libraries contain a list of password in it (RockYou library, specialist library) it is goon take a lot of time to find if the password in the list because his goon try every password in the list and that depends on the device specification that is used.

**2. Brute force password:**

When the attacker knows the length of the user password and start generating all possible password that could work and try every one of them till he found it, this way goon takes a lot of time and use all the hardware capability of the attacker device and the cost to it is so high.

**3. Dictionary attack:**

The attacker attempt to use an English dictionary combined with some knowledge about the target like his favorite animal, food, etc. And attempt to use some AI to try related words from the dictionary.

**4. Social engineering:**

Employee behavior can have a big impact on information security in organizations. Cultural concepts can help different segments of the organization work effectively or work against effectiveness towards information security within an organization. "Exploring the Relationship between Organizational Culture and Information Security Culture" provides the following definition of information security culture: "ISC is the totality of patterns of behavior in an organization that contributes to the protection of information of all kinds."[4]

**Level two attacking scenario:**

**Brute force attack:**

This level standing on the user to upload images of his/her choice from his/her dives then confirm them to the site and arrange the image in the order that pleas him and save it, the attacker have to know the right order for the image to go the next level of singing process, the attacker will use brute force attack to get all the possible combination to the image order and try them all.

**Level three attacking scenario:**

This level is all about the user making watermarking on an image of his choice that he upload it to the site then make a signature on it that he’s the only one that now it, the attacker have to take the same image if he manages to go throw the past level and make an algorithm to try guessing the watermark signature this process goon take a lot of time, efforts, and high cost to make a throw.

## Components of the Attack.

1. **Request library:**

**Requests** allow you to send HTTP/1.1 requests extremely easily. There’s no need to manually add query strings to your URLs or to form-encode your POST data. Keep-alive and HTTP connection pooling are 100% automatic, thanks to [urllib3](https://github.com/urllib3/urllib3).Time library.it used her to send a new password every time to the site to test it.

**Used in levels one, two, three**

1. **Relibrary**:  
   A regular expression (Re library )can be used to check if a string contains the specified search pattern. Throw Rockyou list that contains 14,341,564 unique passwords, used in 32,603,388 accounts

**Used in level one only.**

1. **Time library** :

The time() function returns the number of seconds passed since epoch,

to calculate the time to find the correct password.

**Used in levels one, two, three**

1. **urlopen library**:

Urllib module is the URL handling module for python. It is used to fetch URLs (Uniform Resource Locators). It uses the urlopen function and can fetch URLs using a variety of different protocols.

Urllib is a package that collects several modules for working with URLs, such as:

* urllib.request for opening and reading.
* urllib.parse for parsing URLs
* urllib.error for the exceptions raised
* urllib.robotparser for parsing robot.txt files

This library is used to open the site and take the HTML file and save it in **file.txt** to get dimensional of the image.

**Used in level three only.**

1. **PIL image library** :

PIL is the Python Imaging Library which provides the python interpreter with image editing capabilities. The Image module provides a class with the same name which is used to represent a PIL image. The module also provides several factory functions, including functions to load images from files, and to create new images.

To get the image from the site to start drawing on it.

**Used in level three only.**

1. **Base64 library** :

To decode the images the attacker wants to test them to see if one of them is what the user draws.

**Used in level three only.**

## Attack steps:

**4.3.1 Level one attack**

* + Frist attack
    - is using dictionary attack & Credential Stuffing
    - Run code (readlist.py) to choose every password got the conditions from the given word list (RockYou) and save it in a text file
    - Open word list (txt file )
    - Run code(creacker.py) to try to attack the website
      * Open the website use the request library
      * Use the user name that already knows by the creaker
      * Try every password in the list to login
  + Seanad attack
    - is Password Guessing
    - Run code to generate all possible password that has system conditions and save the password in
    - Run code(creacker.py) to try to attack the website
      * Open the website using the request library
      * Use the user name that already knows by the creaker
      * Try every password in the list to login

4.3.2 Level two attack

* Login to level one by username &password that already know
* Read the HTML page to get the image names
* By using Heap’s Algorithm get all permutations for these images
* Try every permutation to login

4.3.3Level three attack

* Is Brute-force attack in grad Algorithm
* Login to level one & two by username &password that already know
* Read the HTML page to get image Length & width
* generate all possible images that have the same length & width as the original image ( every image with a different pattern )
* hash every image after that send it to the website to try • login

## Attack operations.

1. Level one:

Wordlist attack

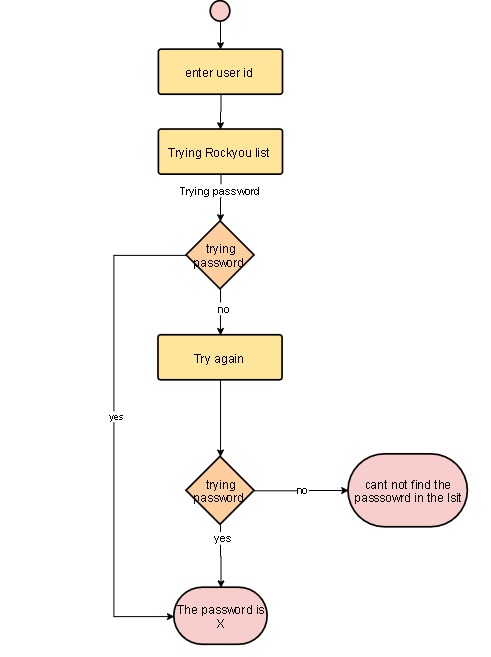


Figure 4.4-1 attack by word list

figure4.1 shows the word list attack first

the attacker enter the user name that already knows after that open the word list (RockYou list) try every password to log in if login successful show the user name & password in the terminal

if not login try again with the next password in the list until the end of the list

Generating password

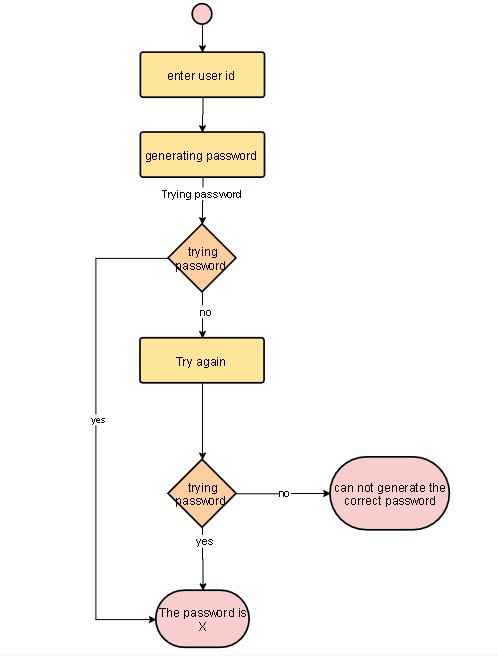


figure 4.4-2 Generating password attack

figure 4.2 show 2 Generating password attack: the attacker enters the user name that already knows after that Generating all possible password that matches the policies of the website save the password in a wordlist

try every password that has been generated to log in if login successful show the user name & password in the terminal. if not login try again with the next password

1. Level two:

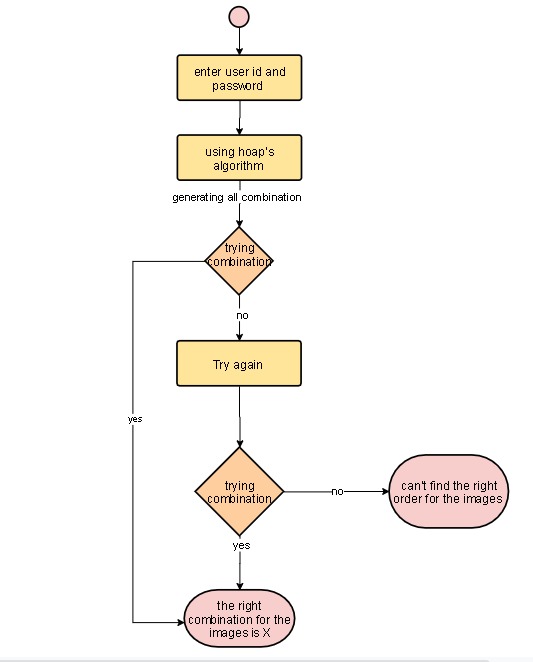
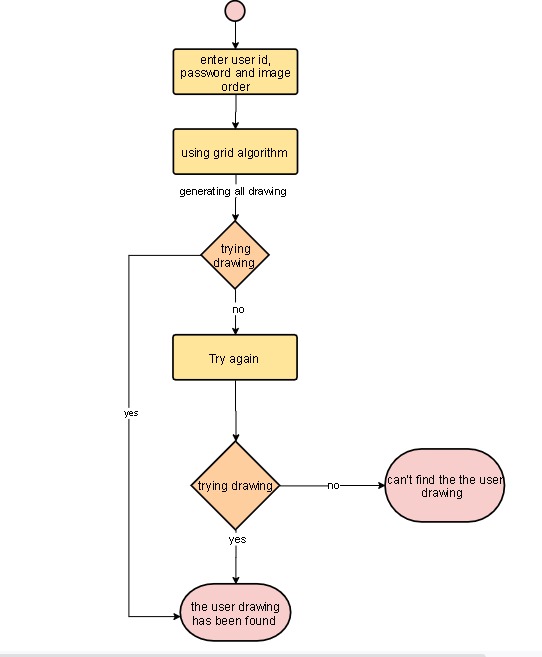


figure 4.4-3 level two attacks

figure4.3 shows the level two attack first the attacker login to the first level with a username and password that already know. Read HTML file of the login page to get images names after that using hopes algorithm to get all permutations, send every permutation to the web site to try login

1. Level three:



fegure4.4-4 level three attack

fegure4.4 level three attack: attacker login to level one and two by username and password that already know The code reads the HTML page of level three to got image Length and width generate all possible combinations of a password(images that have the same length and width with a different pattern ) by drow grades in the original image .after that hash every image that the creak generated it .send every hash to website as a password to try login.

# Chapter Five Implementation

**5.1 Front-end implementation**

The website has been designed three different times

The first try was not that successful the particular reason for the circumstances is that we couldn’t attach The javascript, CSS, and HTML to the back.

The design for our first attempt didn’t come out the way we wanted, the more we tried the more there was to fix and other problems seemed to appear.

the second attempt More problems seemed to appear As for the third attempt and final attempt things seemed to come our way, the following reasons is why. We have used bootstrap because it has helped in supporting responsive design, saves a lot of development in time, consistency and helped the work be more customizable. The first two attempts were a challenge, but at the third attempt we had gotten the hang of it and it made us reach the final design we had in mind.

**Why bootstrap**

-Bootstrap is a giant collection of handy, reusable bits of code written in HTML, CSS, and JavaScript. It’s also a frontend development framework that enables developers and designers to quickly build fully responsive websites.

-Essentially, Bootstrap saves you from writing lots of CSS code, giving you more time to spend on designing web pages.

**Why do we use Bootstrap?**

**-** It is easy to set up and master, it has a lot of components, a good grid system, styling for many HTML elements ranging from typography to buttons, as well as support of JavaScript plugins, making it even more flexible.

-Bootstrap is great for creating layouts, as its responsive CSS is designed to conform to different devices. It can be employed to ensure consistency, and eliminate cross-browser issues. Also, it helps in :

**The following are the advantages of Bootstrap Framework in development:**

**-Bootstrap is Easy to Use**: Bootstrap is very easy and quite simple to use for designing and development. It can be used with CSS, LESS, or SaaS.

**-An Alliance is Quite Easy**: Bootstrap is a framework that is easily integrated with bulk frameworks uninterrupted with existing sites or the new ones.

**-Fast and Time-Saving Framework**: it is an agile framework that is quite faster than other frameworks. It saves time due to its standard ready-made coding blocks, responsiveness, and cross-browser capabilities.

**-Reinforcement of Grids**: One more recommendable feature is that bootstrap holds 12 column grid styles and supports responsiveness, counterbalance, and embedded elements.

**-Adhere to Basics**: Bootstrap is a framework that holds base styling HTML elements like, tables, typography, buttons, forms, images, lists, and icons, etc.

**-A Pack of JavaScript Fundamentals**: Bootstrap supports Javascript Components so one does not require knowledge of scripting. It comprises JavaScript components like tooltips, modal windows, alerts, etc. to add functionalities easily.

**-Support**: Bootstrap is well-known for its big support community and agile response or online solution for the problems or queries. It also offers great documentation along with examples and demo.

**-Customizable**: It can be customized as per the project requirements. Common CSS, utilities, Javascript Components, and components can compose the appearance to create ease for developers.

**-Agile Responsive**: Bootstrap is a fully responsive platform supporting fluid grid layout. It crafts a mobile-ready site smoothly and easily and sets web design as per the dimensions of the particular device.

**-Bootstrap's Consistency**: Websites or mobile applications made with bootstrap technology can make you feel the awesome pairing of developers and designers ensuring consistency.

**-Frequent Updates**: It provides regular updates and makes the whole eco-system hefty and robust according to the time.

**Disadvantages of Bootstrap**

**1-Templates**

Sites developed with Bootstrap are similar to each other: the same structure, navigation, buttons. Each new site is similar to the many already created ones, and this does not seem professional so trying not to use ready-made solutions will help create unique websites. That being said, it is important to understand all the tools offered by Bootstrap to maximize their use in development.

**2-Lack of flexibility**

Despite all the advantages, Bootstrap is a tool that has its limitations ( all sites on it are similar to each other). Therefore, Bootstrap may not be suitable for the implementation of some projects.

**3-Older browsers**

Since Bootstrap tries to keep up with the times and is constantly updated and as a result, the sites on Bootstrap may not display correctly in older browsers.

WordPress website development seems the obvious and cost-effective solution for many types of websites. For corporate website development, it is better to ask web professionals for advice.

**5.2 Back-end implementation:**

### sign up level one

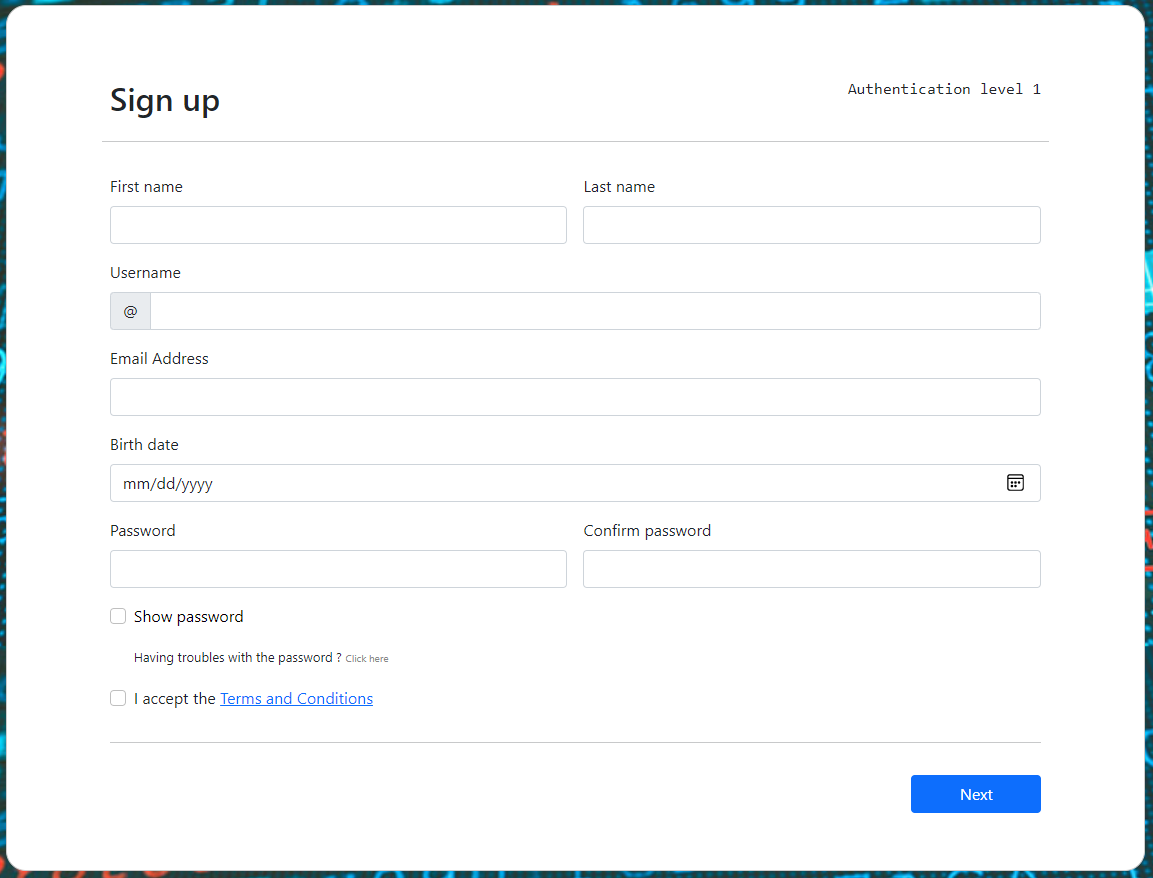


Figure ‎5.2‑1 shows the steps of sign up

* **Screen ID:** 1.
* **Screen Precondition:** the user must fill this form with some data like first name, last name, username, email address, birth date, password, confirm password.
* **Screen Description:** Check the first name, last name, username, email address, birth date, password, confirm password if the first name is not letters only or last name is not letters only or username is not letters and numbers only and is existed or email address is not valid and is existed or birth date is less than 18 years or password is not strong or password and confirm password does not match, the user redirects to sign up level one as shown in Figure **(5.2-1)**
* **Screen Post-condition:** the user redirects to sign up level two page as shown in Figure **(5.2-6)** if the all data in sign up level one is correct, otherwise redirect to sign up level one as shown in Figure **(5.2-1)** if any data in sign up level one is not correct.

### 5.2.2 password generator

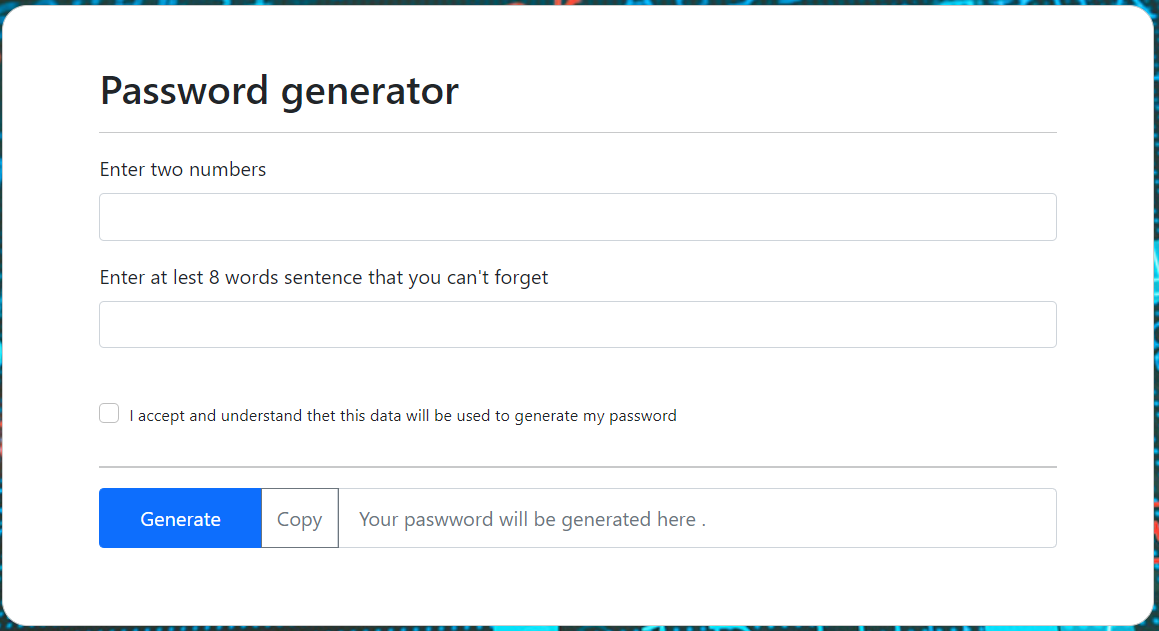


Figure ‎5.2‑2 shows password generator

1

* **Screen ID:** 2.
* **Screen Precondition:** the user must fill the form with two numbers. Then, writing a set of words or a sentence (eight words or up) as shown in Figure **(5.2-2)**.
* **Screen Description:** After the user clicks on generate button, the system makes the output password collects random numbers and some random symbols and takes the first letter of each word, and puts them together.
* **Screen Post-condition:** the user redirects to sign up level one page as shown in Figure **(5.2-1)** to complete sign up if the password is correct, otherwise redirect to password generator as shown in Figure **(5.2-2)**

### 5.2.3 login level one

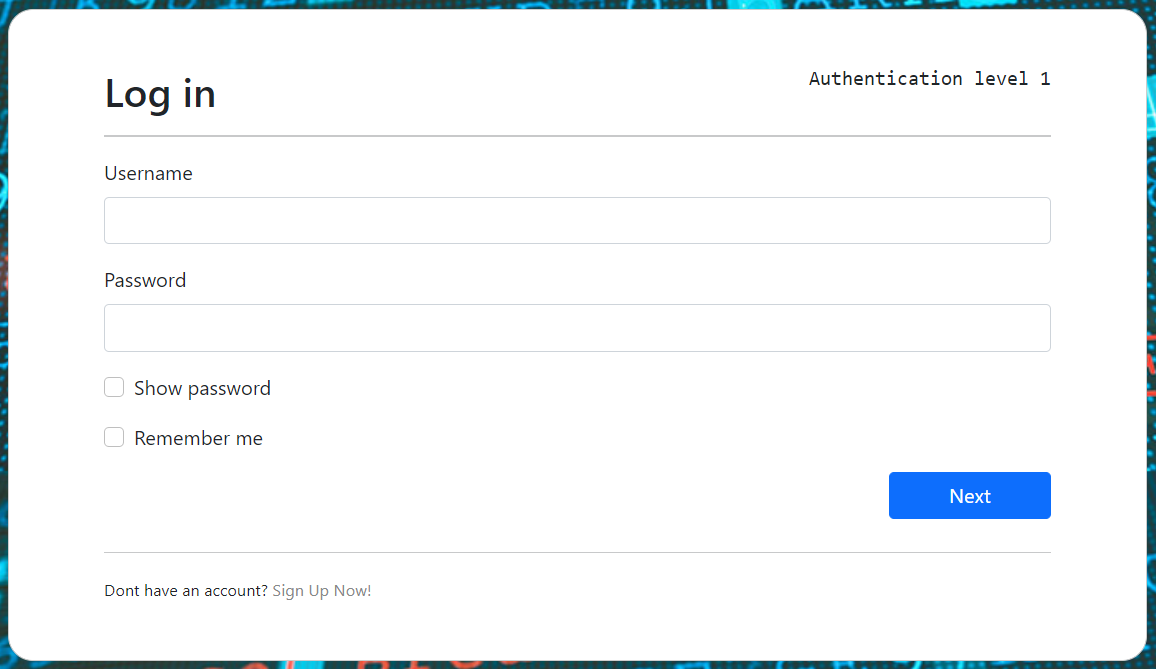


Figure ‎5.2‑3 shows login of first level 1

* **Screen ID:** 3.
* **Screen Precondition:** it must be a first-time login.
* **Screen Description:** the user enters the username and password correctly as shown in Figure **(5.2-3).**
* **Screen Post-condition:** the user is directed to login level two if enters the username and password correctly page as shown in Figure **(5.2-7),** otherwise redirects to login level one as shown in Figure **(5.2-3).**

### 5.2.4 uploader level two

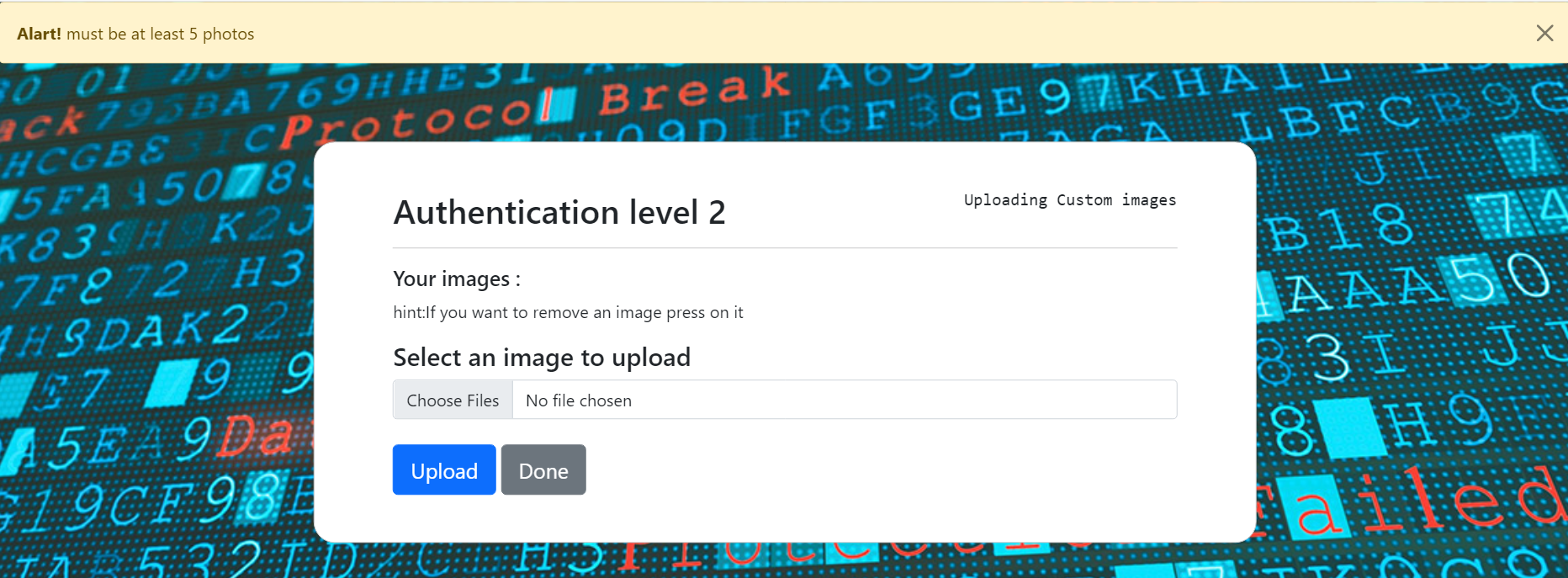


Figure 5.2-4 shows level 2

* Screen ID: 4
* Screen precondition: the user has to upload multiple images at the same time as shown in figure (5.2-4)
* Screen description: check if the user uploaded at least 5 photos and check the type of the photos if they are {jpg, jpeg, png, gif}
* Screen post-condition: the user redirects to the uploader

Page as shown in figure (5.2-4) if the images are 5 at least and the right type.

### 5.2.5 offered the selected images

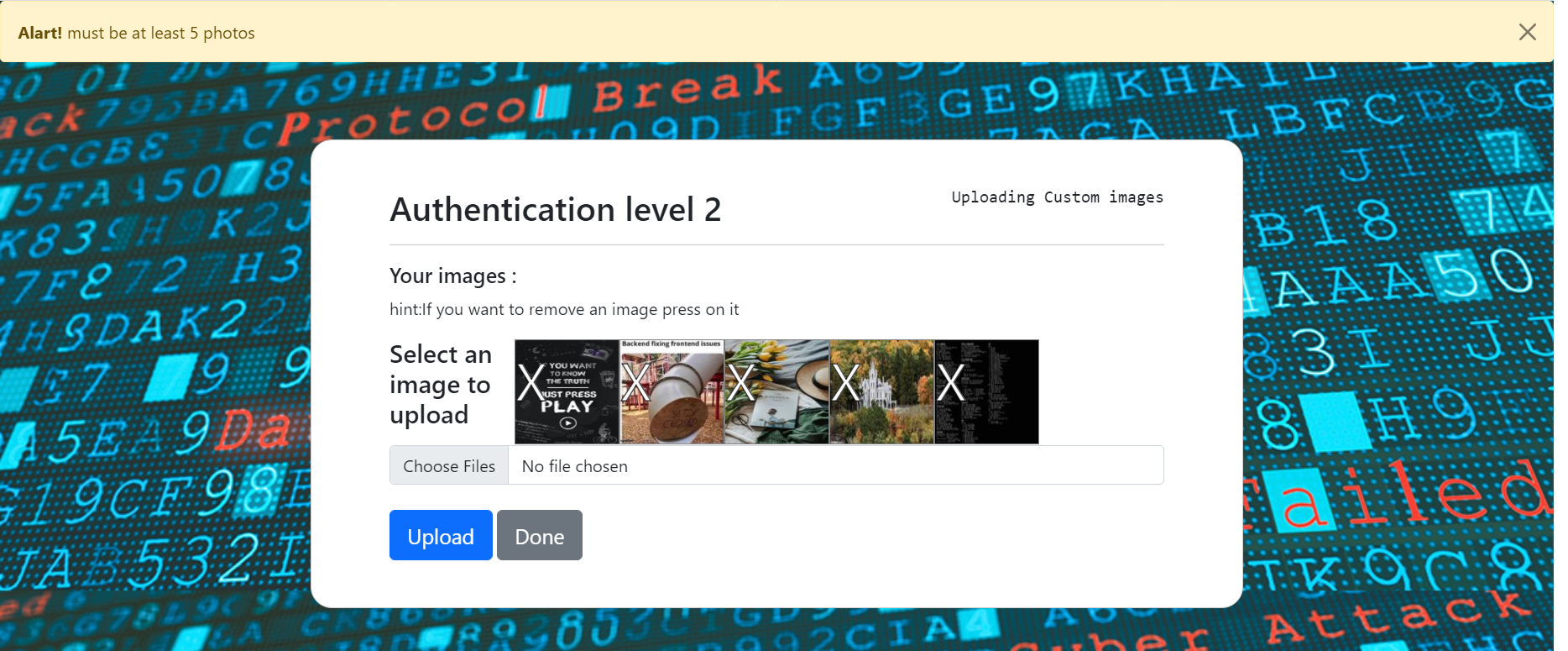


Figure ‎5.2‑5 shows selected images

* Screen ID: 5
* Screen precondition: user ensure that the images are uploaded and if he wanted to remove an image click on images and remove it
* Screen description: check if user removed images are still in the minimum 5
* Screen post-condition: the user after click done redirects to signupf2 as shown in figure (5.2-6) if the images are 5 at least.

### 5.2.6 sign up level two

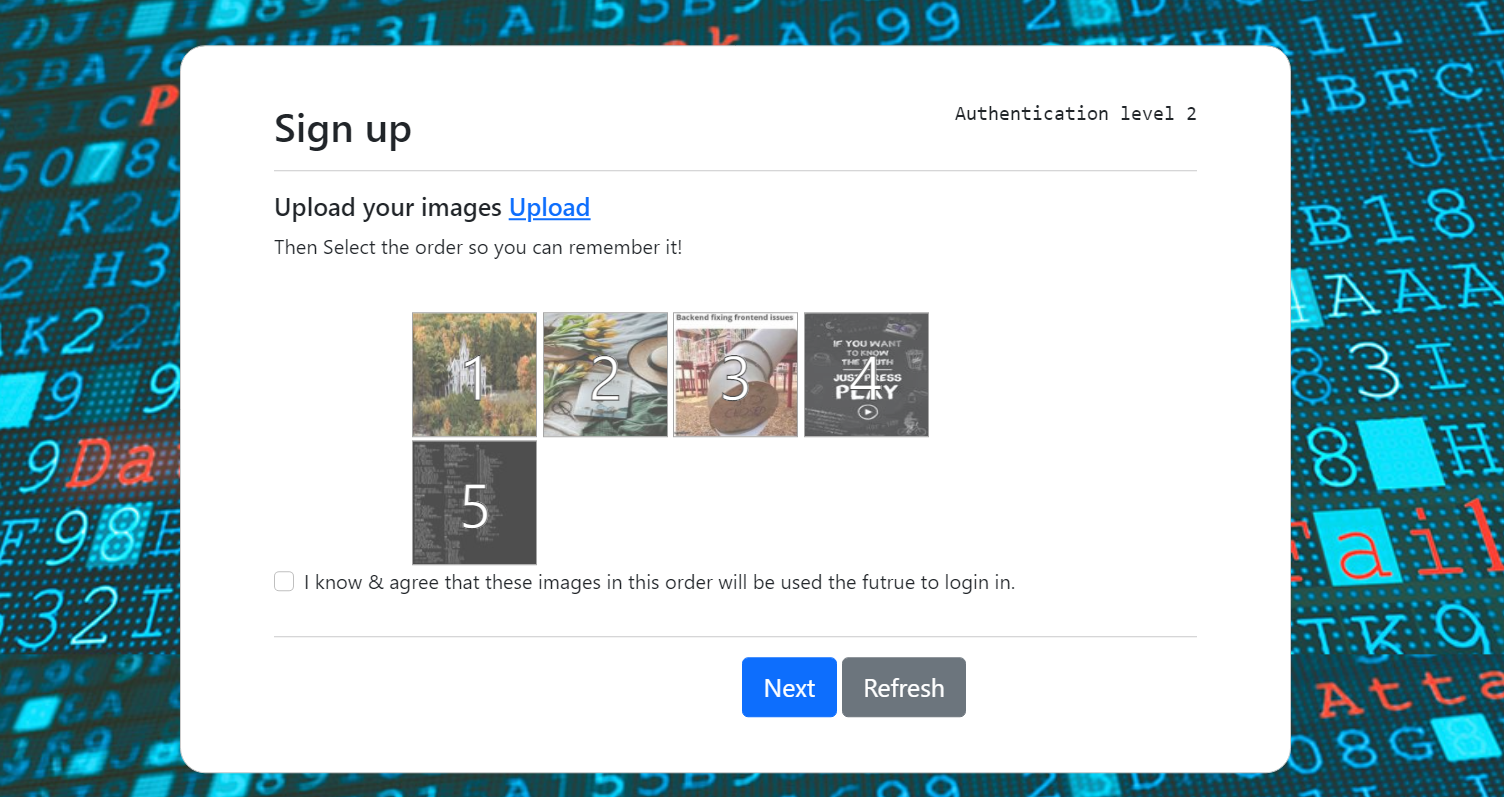


Figure ‎5.2‑6 shows sign up level 2

* Screen ID: 6
* Screen precondition: user order the images and remember them very good cause when back to sign in
* Screen description: check if the user ordered the images and if the user clicks on that knowledge and agree that these images in this order will be used in the future to login in as shown in figure (5.2-7)
* Screen post-condition: the user after ordered the images redirects to log in as shown in figure (5.2-7).

### 5.2.7 login level two

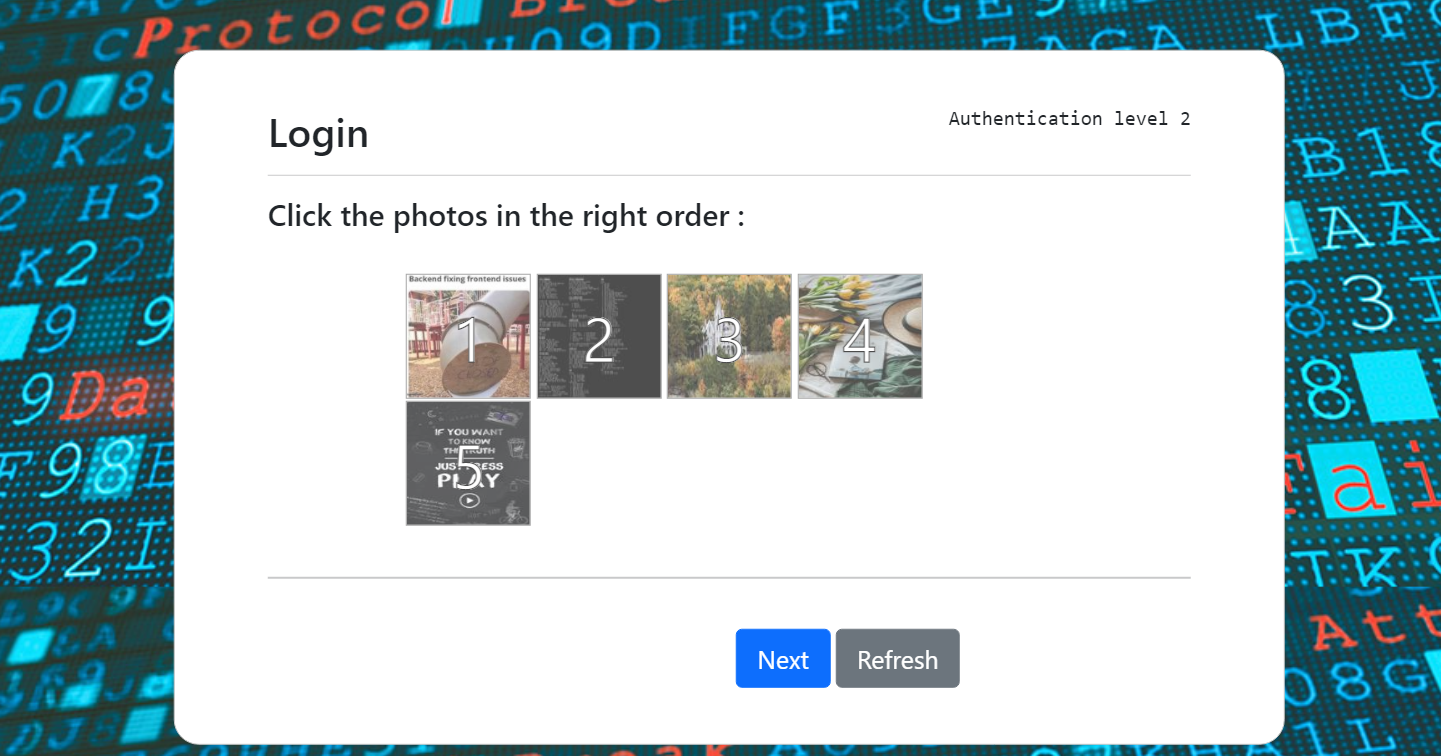


Figure ‎5.2‑7 shows login of level 2

* Screen ID: 7
* Screen precondition: the user orders images from his/her memory
* Screen description: check if the user orders the images correct or not it depends on comparing the photos of login and the photos that stored in a database
* Screen post-condition: the user directed to signupf3 as shown in figure (5.2-9) if the order is correct.

### 5.2.8 uploader level three

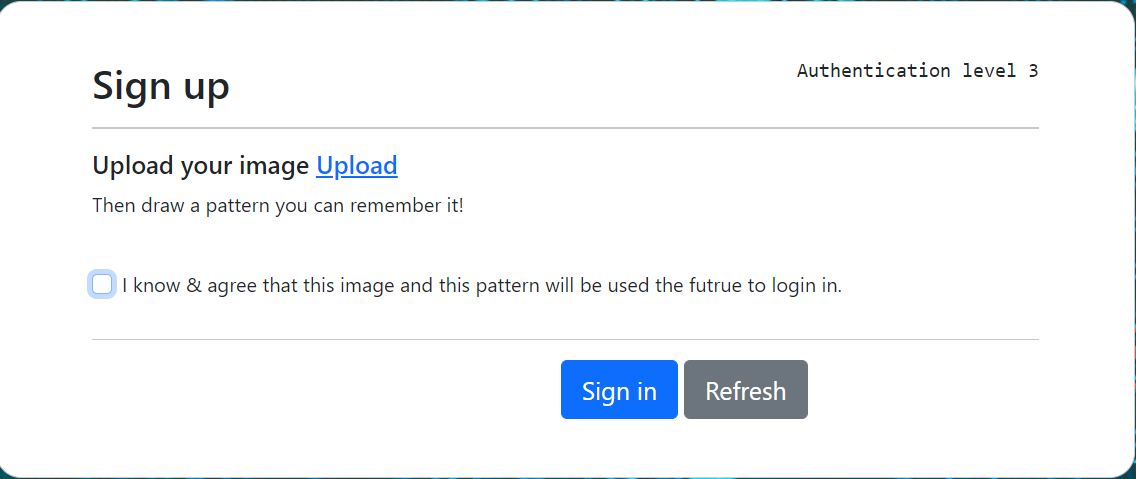


Figure ‎5.2‑8 shows uploader level 3

* **Screen ID:** 8.
* **Screen Precondition:** the user must upload an image. The size of the image should be less or equal (1000\*1000) and the Allowed image types are png, jpg, jpeg, gif.
* **Screen Description:** If the image larger than this size, print the alert for the user they resize the image into this size as shown in Figure **(5.2-8)** then, the user redirects to sign up level three as shown in Figure **(5.2-9).** If the image types are not: png, jpg, jpeg, gif**,** theuser redirects to uploader level three as shown in Figure **(5.2-8).** If the image types are: png, jpg, jpeg, gif**,** theuser redirects to sign up level three as shown in Figure **(5.2-9).**
* **Screen Post-condition:** After the user confirms an image, it offered it to the user can able to draw his pattern in this image as shown in Figure **(5.2-9).**

### 5.2.9 sign up level three

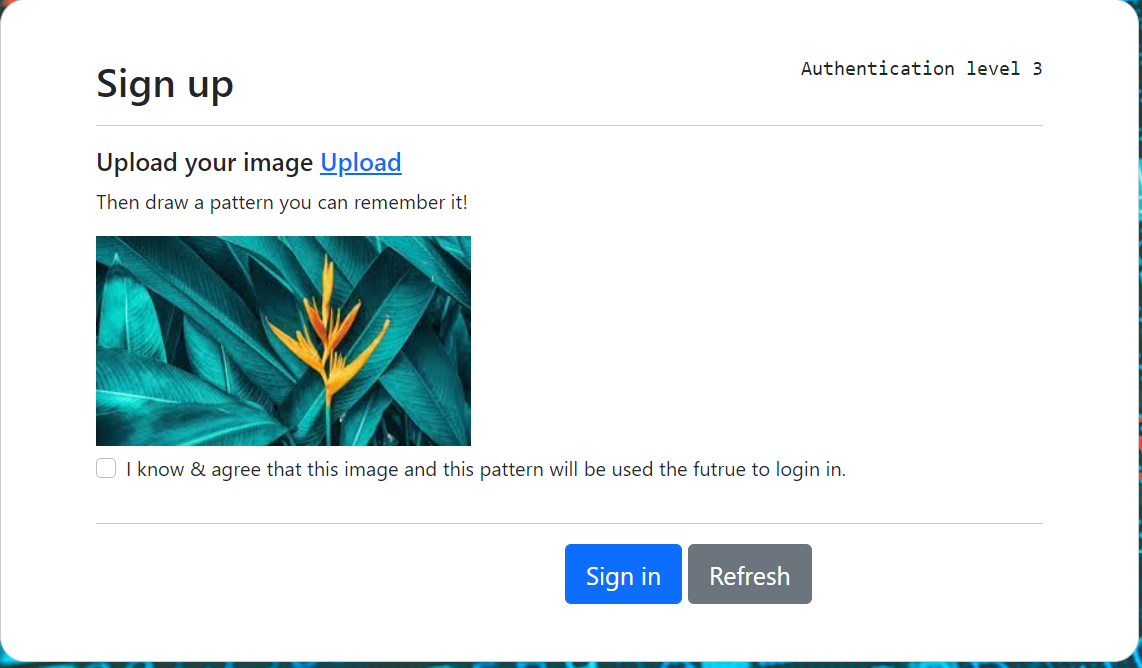
****

Figure ‎5.2‑9 shows sign up level 3

* **Screen ID:** 9.
* **Screen Precondition:** the user must draw his pattern on the selected image which can remember it in the login stage as shown in Figure **(5.2-10).** Then, click on the check box. Then, click on the sign-up button to ensure this pattern.
* **Screen Description:** the user can change the pattern by clicking on the refresh button if he did not complete the sign-up level three as shown in Figure **(5.2-10).**
* **Screen Post-condition:** if the user draws his pattern and click on the sign-up button, the user redirects to login level one as shown in Figure **(5.2-3).**

### 5.2.10 login level three



Figure ‎5.2‑10 shows login level 3

* **Screen ID:** 10.
* **Screen Precondition:** the user should draw the correct pattern which draws it into the sign-up stage to make correctly access to the app.
* **Screen Description:** The drawn pattern was invisible to the user. Then, the user should the login button to ensure if he draws the correct pattern or not.
* **Screen Post-condition:** if the user draws the correct pattern directs to the home page, otherwise the user redirects to login level three as shown in Figure **(5.2-10).**

# Chapter Six: Conclusion and future work:

## Conclusion:

We all use to and loved the old classic text-based password. but, time shows it **can’t stay forever**, and with the growth of processing power day by day, it becomes much easier to **brute force** text passwords. And one day, text passwords will be **removed** and **replaced** like any other old technology. Right now **MFA** is considered to be the **best** option to replace it, whether by adding **more factors** or **more levels** of the same factor to the systems.

We tried using **images**, **graphics**, and **patterns** as a replacement for the text-based password and so far it gave us much **better results** but it also had some **downsides** to it.

Till we overcome these disadvantages or the user gets used to it we might stay on using test passwords and **TFA** since it’s much user-friendly and more secure than just a text.

In the end, we **proved** that the **knowledge-based authentication factor** could be more **secure** than just a normal text, and with the right **policies** we would still be able to keep going with only text-based passwords.

But no matter how far we go with the **length** or **complexity** of a text, they would be **always breakable**.

## Future works:

In the future, level three is mostly to be the ideal module with much **more usability** and much **greater security**.

There are **greater chances** to be done and more work to do, and it would be great if the module made it out to the **market** and the **average user**.

# References

1. [*"What is Authentication? Definition of Authentication, Authentication Meaning"*](https://economictimes.indiatimes.com/definition/authentication). The Economic Times*. Retrieved 2020-11-15*.
2. Turner, Dawn M. [*"Digital Authentication: The Basics"*](http://www.cryptomathic.com/news-events/blog/digital-authentication-the-basics). Cryptomathic. [*Archived*](https://web.archive.org/web/20160814214552/http:/www.cryptomathic.com/news-events/blog/digital-authentication-the-basics) from the original on 14 August 2016*. Retrieved 9 August 2016*.
3. McTigue, E.; Thornton, E.; Wiese, P. (2013). ["Authentication Projects for Historical Fiction: Do you believe it?"](http://eric.ed.gov/?id=EJ1012775). *The Reading Teacher*. **66** (6): 495–505. [doi](https://en.wikipedia.org/wiki/Doi_(identifier)):[10.1002/trtr.1132](https://doi.org/10.1002%2Ftrtr.1132). [Archived](https://web.archive.org/web/20150707211148/http:/eric.ed.gov/?id=EJ1012775) from the original on 2015-07-07.
4. Lim, Joo S., et al. "[Exploring the Relationship between Organizational Culture and Information Security Culture](https://ro.ecu.edu.au/cgi/viewcontent.cgi?article=1011&context=ism)." Australian Information Security Management Conference.
5. [*"Flask Foreword"*](https://web.archive.org/web/20171117015927/http:/flask.pocoo.org/docs/0.10/foreword). Archived from [*the original*](http://flask.pocoo.org/docs/0.10/foreword) on 2017-11-17.
6. Provos, Niels; Mazières, David; Talan Jason Sutton 2012 (1999). [*"A Future-Adaptable Password Scheme"*](http://www.usenix.org/events/usenix99/provos/provos_html/node1.html). Proceedings of 1999 USENIX Annual Technical Conference: 81–92.
7. [*"zzzeek / sqlalchemy / source / LICENSE"*](https://bitbucket.org/zzzeek/sqlalchemy/src/775ff3419fa03ddbb1b4b6d7350344b69df461ef/LICENSE). BitBucket*. Retrieved 21 February 2015*.
8. [*"Python Imaging Library"*](http://www.pythonware.com/products/pil/). Secret Labs AB*. Retrieved December 8, 2013*.
9. https://pypi.org/project/ImageHash/
10. https://pypi.org/project/ping3/
11. [*"CSS developer guide"*](https://developer.mozilla.org/en-US/docs/Web/Guide/CSS). Mozilla Developer Network. [*Archived*](https://web.archive.org/web/20150925133829/https:/developer.mozilla.org/en-US/docs/Web/Guide/CSS) from the original on 2015-09-25*. Retrieved 2015-09-24*.
12. [*"ECMAScript® 2020 Language Specification"*](https://tc39.es/ecma262/#sec-overview).

# **Appendices**

## **Appendix A**:

## This is the Github link where all the score code is stored for the project itself or the testing codes.

## <https://github.com/3lvlsTeam>

# ملخص المشروع باللغة العربية

**"نظام ثلاثة مستويات لكلمة السر"**

**في أيامنا هذه ، حيث أصبحت أنظمة المعلومات أكثر انفتاحًا على عالم الإنترنت ، تزداد أهمية أمن الشبكات. الأمن هو وسيلة الحماية لحماية الأمة أو الأشخاص أو الأشخاص من الخطر والضرر والخسارة والجريمة. لذلك فإن كلمات المرور النصية ليست كافية لمواجهة مثل هذه المشاكل. تتطلب هذه الحاجة إلى شيء مؤمن بشكل جيد إلى جانب سهولة الاستخدام ، لذلك فإن الأمان ذي المستويات الثلاثة هو برنامج سهل الاستخدام. حيث تم زيادة الأمان باستخدام 3 مستويات, في هذا المشروع "نظام أمان ثلاثي المستوى باستخدام المصادقة المستندة إلى الصور". تم اقتراح نظام مصادقة من النوع الجديد ، وهو آمن للغاية في هذا المشروع. هذا النظام ، هو أيضا أكثر سهولة في الاستخدام. على الرغم من أن نظام الأمان ثلاثي المستويات يعد نهجًا مستهلكًا للوقت ، إلا أنه سيوفر أمانًا قويًا حيث نحتاج إلى تخزين البيانات المهمة والسرية والحفاظ عليها. توفر هذه الأنظمة قناة اتصال آمنة بين الكيانات المتصلة. كما تدعم سهولة استخدام الصور وتذكرها ككلمة مروريدعم نطاق جميع الانظمة .**

**1.المستوى الاول يعتمج على كلمة مرور نصية تحت شروط معينة من اجل جعل كلمة المرور قوية.**

**كلمة المرور تعمد على ان تحتور على احرف كير واحرف صغيرة تتكون من 12 حرف على الاقل وتحتوي على الارقام والرموز رقم ورمز واحد على الاقل.**

**2.المسنوى الثاني يتم على الصور من ناحية رفع صور من المستخدم وترتيبها بشكل معين.**

**حيث يقوم المستخدم برفع خمس صور على الاقل ثم ترتيبها حسب ما يريدز**

**3.المستوى الثالث يعتمد على ان المستخدم يقوم بوضع توقيع او رسم شكل على صورة يقوم برفعها.**

**يقوم المستخدم برفع صورة ثم يقوم برسم شكل سهل التذكر له او امصاء على الصورة بحيث لا احد غيره يستطيع ان يقوم به.**