

## Child Friendly Authentication (CFA)

Nada Hamdan Mohammed AL-Risi<sup>1\*</sup> and Fatma Khalfan Mohammed AL-Badi<sup>1</sup>

<sup>1</sup>Information Technology Department, University of Technology and Applied Sciences (UTAS) Suhar – Sultanate of Oman

\*Corresponding author

### Abstract

A child-friendly authentication system that uses graphic images is an excellent solution to avoid potential electronic risks to children from attackers. This report provides a summary of building a child-friendly authentication system by replacing the traditional text-based authentication method with an image-based authentication method. Where some studies and authors' research have been presented on the benefits of using images as passwords and that it is considered the best compared to the text password, on the grounds that the human mind has more ability to remember images faster than text. In addition, this report included some of the applications and examples of using this technology in advance for adults. This report covered many important and critical points related to building a graphical authentication system for children. The results of previous research have added valuable information to this report. Finally, this report contains the challenges we may encounter while applying this technique, and how to avoid them.

**Keywords:** Password; Authentication; Child friendly; Multifactor authentication; Graphical authentication.

### 1. PROBLEM STATEMENT

Recently, after the Covid-19 pandemic, children (between 5 and 11 years old) are increasingly using electronic devices and internet services. Among the examples of systems that children had a major role in using are educational systems due to the existence of this epidemic, education has become online, so children's use of information technology systems has increased, and there are many electronic systems that children use and there is likely to be a point A futuristic transformation and officials resort to using various electronic systems with the category of children, given that the world is facing its transfer to the world of technology. This has left them vulnerable as they have little or no knowledge of using passwords and how to authenticate their accounts. Given the importance of authentication in information technology systems, and the children's tendency to electronic systems, it is necessary to try to find a way to protect their electronic activity. To protect children from being hacked, an authentication mechanism based on the use of graphical passwords is designed for them given the fact that humans can remember

images easier than text. Graphical authentication use mechanism includes special characters and letters, but they are represented graphically. With graphical authentication, younger users click pictures instead of typing characters to verify their identity. In this project, a child-friendly authentication system is designed, a system that allows children from (5-11) years old to create a password using graphics. The child enters a password with a certain length required by the system and then the system matches the entered password by requesting confirmation from the Child.

## 2 LITERATURES REVIEW

### 2.1 Authentication definition

Authentication is a function in which the user provides some credentials to the system. The system recognizes the credentials or matches the credentials to a particular group on the system, in this case it is said that the user is authorized to enter the system, meaning that the process of matching the data was correct and the authentication has been done; If the match does not occur, the user is not authorized to enter the system. The authentication process is critically required to allow the system to perform some tasks the user (Michael et al., 1990). A typical authentication process consists of an appendix, notarized, and Authentication server. Defines a petitioner as “an entity that is attested by a file.” Authenticator”, while Authenticator is defined as “an entity that requires authentication from Supplicant (Lunde et al., 2006) [1]. In addition, Akula et al. (2004) The user needs a permit in order for the system to provide services to him. It is necessary that before a user can be authenticated on a particular system, they must be registered in system for the first time. This step is called registration. That is, for the new user, he must first register in the system and then certify it before he can request system services.

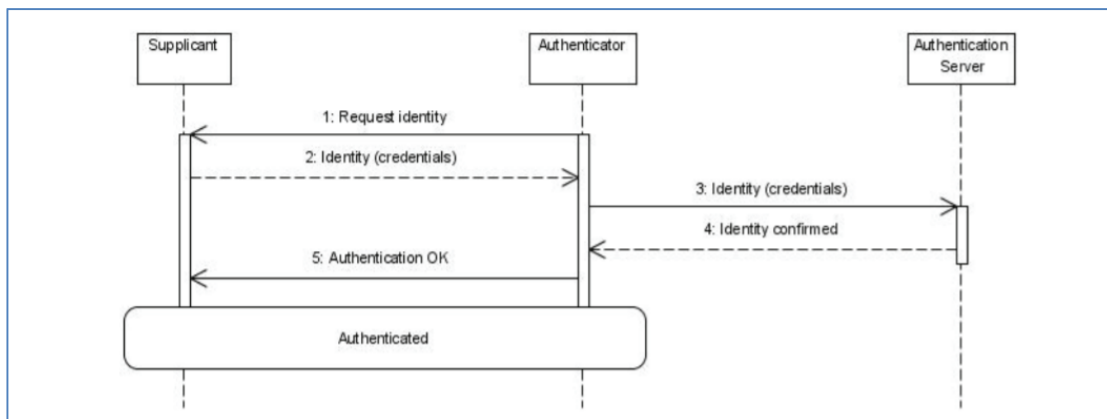


Figure 1 General Authentication Scheme (Lunde et al., 2006)

### 2.2 Types of authentications

At the beginning of the authentication app, only one factor was used for authentication for example using a password (or PIN) to confirm ownership, but this is the weakest level of authentication. By sharing the password, a hacker can easily hack accounts. As an obvious step forward, two-factor authentication (2FA) was proposed that coupled representative data with a proprietary factor, but also faced many security threats. To provide a higher level of security and facilitate the ongoing protection of computers, Multi-Factor Authentication (MFA) is proposed (Ometov et al, 2018). Sabzevar & Stavrou (2008) indicate that the MFA relies on the availability of three types of combinations of factors to associate the individual with the created credentials: the knowledge factor as a password, the ownership factor as something the user owns, such as cards, smartphones, or other tokens, and the biometric factor,

i.e., what the user is biometric data or behavior pattern. The MFA relies more on biometrics, which is the automatic identification of individuals based on their behavior.

A password is a very good and strong authentication method that is still used today. But because of the tremendous progress in the uses of computers in many applications such as data sharing, data transfer and logging in to the Internet or emails, there are some disadvantages to the traditional password interface such as: forgetting the password, weak password, or password theft. So, there was a great need to find a strong authentication method to secure all user requests as much as possible. From this need, the researchers came up with an advanced level password called the graphic password.

### 2.3 Graphical password authentication

Blonder (1996) indicate that, several graphic password systems have been proposed as an alternative to text passwords, and research has shown that text passwords have drawbacks such as simplicity of use and security challenges, making them unsuitable for solving data and information security issues. According to the research, the human brain is better at identifying and retrieving images from texts, and graphic passwords are trying to take advantage of this human trait so that we can reduce the memory burden on users by making use of the vast space of images for passwords, enabling the production of more secure passwords and reduces risks for users. Based on these studies, it is known that the subconscious mind of children tends to images and shapes in the process of remembering and responding faster and better than remembering texts than adults. According to Yang et al. (2012), a graphic password is a type of password in which images or graphics are used to complete the authentication process and is considered an alternative to the traditional authentication method in which numbers, letters, or what is known as written text are entered. A graphic password is not a new idea to the world of technology, it was introduced by researcher Blonder in 1996, which provides for authentication through a graphic password, which is an image that appears on the screen and the user must click on the image area to choose the correct areas (Blonder, 1996).

Graphical password authentication technology is not a new technology in login systems, and can be divided into two types:

1 Recall based graphical password technique: It is also divided into two types:

1.1 *Pure recall-based technique:*

The user must reproduce the password without any assistance from the system. For instance, DAS, Grid selection, and so on.

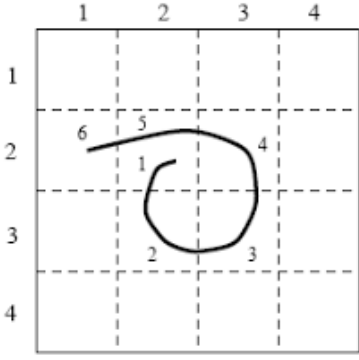
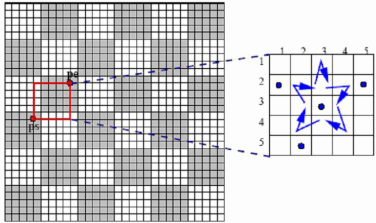
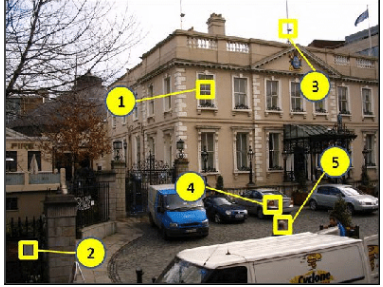
1.2 *Cued recall-based technique*


In this strategy, the user is given a suggestion that will assist him to remember the passwords he chose during the registration phase. For instance, the PassPoint.

2 Recognition based graphical password technique:

In this method, the user is given a series of pictures from which he must choose the correct image that he chose during the registration phase. Passface and Déjàvu are a few examples.

**Table 1: shows several techniques of Graphical password authentication.**

Technique	Description	Example Picture
<p><b>DAS</b> <b>(Draw-a Secret)</b></p>	<p>It is a pure retrieval-based technique in which the user must redraw a predefined image pattern on a 2D grid. User authentication is successful when the graphic touches the same networks in the same sequence. This is a technique commonly used in mobile devices (graphic pattern lock system applications).</p>	 <p><b>Figure 2 Draw-a-Secret technique</b></p>
<p><b>Grid selection</b></p>	<p>The Grid selection algorithm is also a pure recall-based authentication technology. Where the user is asked to select a small area from a large rectangular grid. This area is then enlarged upon selection and then he is asked to draw a password pattern.</p>	 <p><b>Figure 3 Grid selection</b></p>
<p><b>PassPoint</b></p>	<p>It is a recall-based technique where the system allows any natural image to be used enough to get as many clicks points as possible. The task of the image is only to provide a hint to the user that helps him remember the click points. While logging in, points are clicked and must be selected in the same order as at the registration stage.</p>	 <p><b>Figure 4 PassPoint</b></p>

<p><b>Passface</b></p>	<p>It is an authentication technology based on face recognition. The set of faces is assigned to the user and then the user recognizes the faces and clicks anywhere on the known face. This procedure is repeated for several attempts until the user is authenticated if he has correctly identified the four faces.</p>	 <p><b>Figure 5 Passface technique</b></p>
------------------------	--	--

## 2.4 Benefits of using graphical authentication

The most important motivation behind using a graphical authentication mechanism is that its memorability trumps text passwords. According to a study that compared the types of human memory, it was found that words are processed by our short-term memory and that is why we have 7-digit phone numbers. On the other hand, images are stored directly to long-term memory where they are usually engraved. He also noted that long-term memory is not easily destroyed over time and an unlimited amount of information can be stored in it, and this supports the approach that human memory can identify and remember pictures better than text (Jebriel & Poet, 2014).

As for the security of the graphic password system compared to text-based passwords, it is more secure. Rittenhouse and Chaudry & Lee (2013) indicate that the main defense against brute force search is to have a large enough password space. Text-based passwords have a larger password space as  $94^N$ , N is the password length, and 94 is the number of printable characters excluding SPACE. As for graphic passwords, some techniques have been demonstrated to provide a password space similar to or larger than that of text passwords (Hu et al., 2010). In practice, recognition-based graphic passwords tend to have smaller password spaces than recall-based methods. Moreover, A brute force attack on graphic passwords is also more difficult to implement than text passwords. Where attack programs need to automatically generate accurate mouse movement to mimic human input, it is more difficult than intercepting keyboard input. So, Hu et al. (2010) believes that graphical passwords are more resistant to brute force attack than text-based passwords.

## 2.5 Authentication for child

Child Friendly Authentication is a proposed system on an authentication interface that meets the needs of children to increase the child's interests in adopting graphical passwords (Yang et al., 2012). Moreover, they suggested before setting this system that the child's abilities in graphical password application should be studied:

- a. **Visual ability:** The graphic password authentication display screen should be clear to the child because the problem of color blindness in children is undeniable. Color-blind patients suffer from not recognizing a particular color. Where a study was conducted in 2001 and found that 3.9% of children with color blindness, the average age tested was 10 years (Shrestha, 2010; Shrestha 2016). Therefore, the author Gao (2009) pointed out A graphical password authentication system should be designed based on no color recognition to ensure that a color-blind individual is not restricted from using the system.
- b. **Memory capacity:** Children tend to have difficulty remembering and recognizing too many graphic images in accordance with their current memory capabilities and caused them to waste time

retrieving their password (Imran, 2016). Furthermore, Adults have their own strategy for remembering their chosen password. Whereas kids are only able to choose the graphic password based on their interests (Pacchioli, 2005). The author Stobert (2017) advises that a child can be given 2 graphic images during validation without a graphic sequence to be able to remember them.

### 3. METHODS AND METHODOLOGY

#### 3.1 Design of database

Database design is the organization of data according to the database model used in the system. The following design explains what CFA data should be stored and how the data elements are interconnected. By using this information, we ensure the accuracy of the data entering the system as we ensure the consistency of the information, eliminate redundant data, efficiently execute queries, and improve database performance. In our system we need the user database, the administrator database, the image database used to implement the graphical authentication, and finally a database of login tries which are used to store the data of the user who tried to enter the site

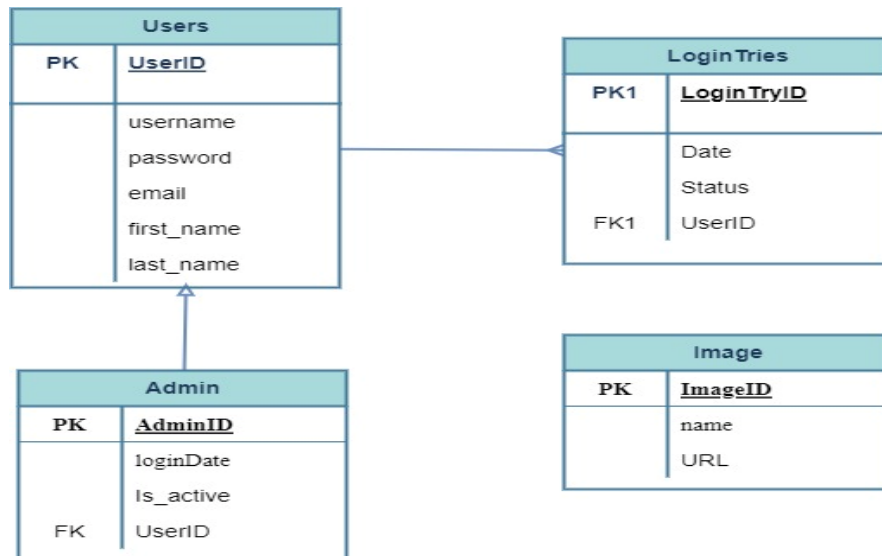
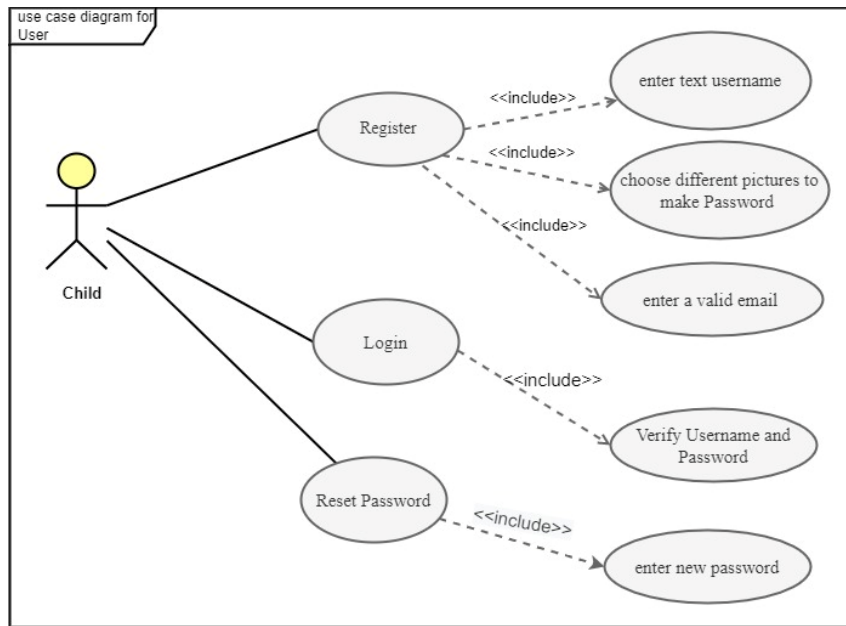


Figure 6: Design of database for CFA

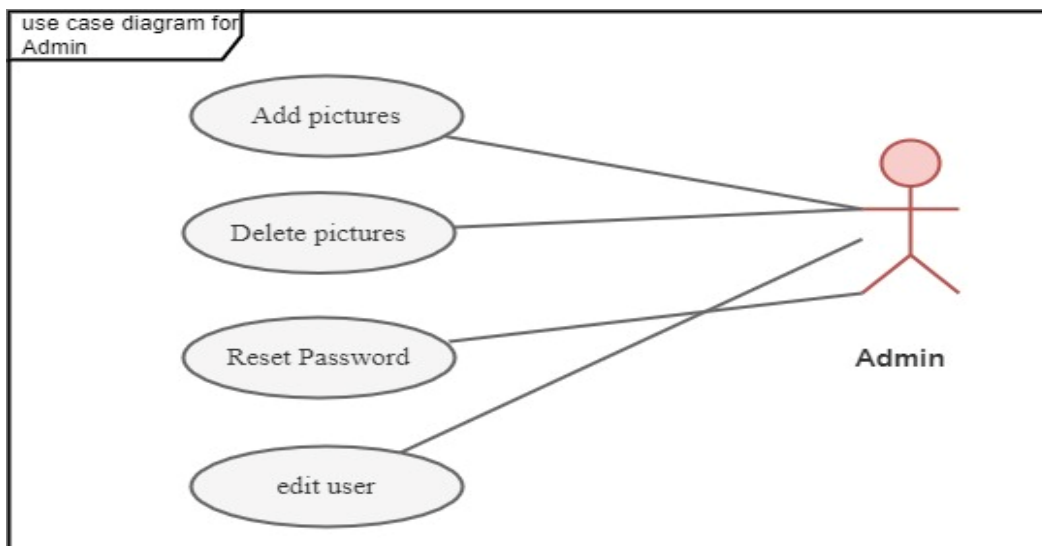
#### 3.2 Use case diagram

A use case diagram is a graphic depiction of external user interactions with the proposed system. A child-friendly authentication system contains representatives. The first actor is the child who has 3 use cases which are Register, Login and Reset password. Firstly, in the registration process, he should enter his username and email as text and then choose different pictures to make a password for him. Secondly, in the login use case, the user can easily signIn to the system by entering the username and password that he made in the registration process. Finally, the user can reset his password if he forgets it.



**Figure 7 :Use case diagram for Child**

The second actor is the admin who is responsible for the system and editing the pictures for example he can add more pictures, delete pictures, or reset the password if the user forgot his password.



**Figure 8:Use case diagram for Admin**

### 3.3 Class diagram

The following class diagram illustrates the proposed system by visualizing the different types of objects within the system, the types of fixed relationships that exist between them, and the operations and properties of the classes. In a child-friendly authentication system, we need 4 classes. A class for users and a class for the one responsible for modifying the users and images used in the system also a class for images and finally a class for login try.

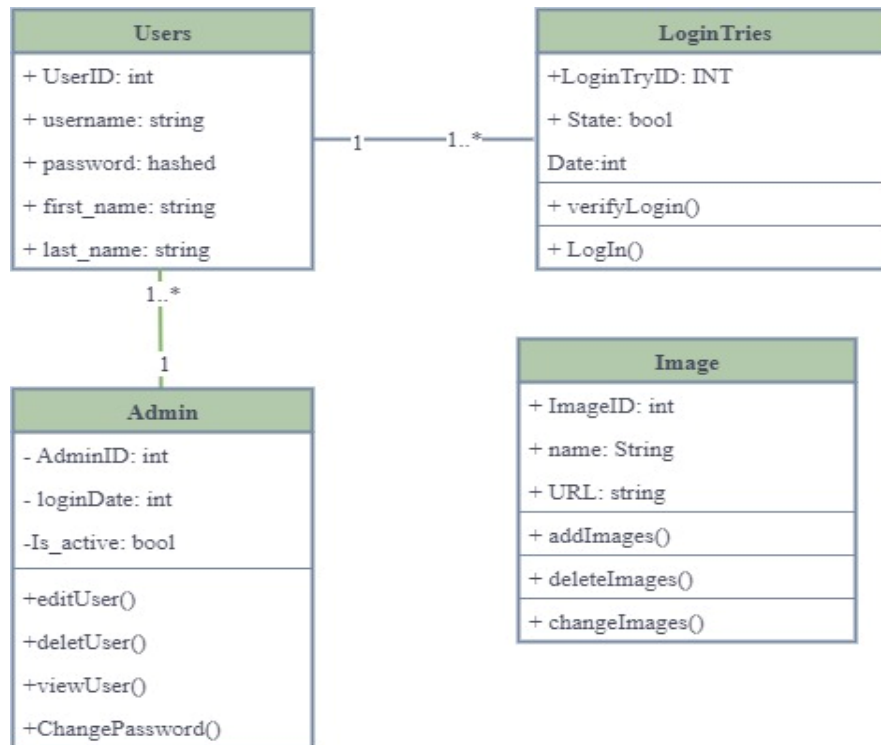


Figure 9 : Class diagram for CFA

### 3.4 Sequence diagram

The sequence diagram shows the interactions of objects in the system arranged in a time sequence. It depicts the objects involved in the way the system operates and the sequence of messages exchanged between the objects necessary to carry out the functions of the system. The child-friendly authentication system contains 3 objects as shown in the following diagram. The first object is the user (the child) exchanging direct messages with the system to register first. Then the system communicates with the database to verify whether the registration process is correct or not. Then the system returns to notify the user to log in to the system, and if the user enters his data correctly, the system asks the database to authenticate the user's password. If the authentication is done, the user can enter the system and if it is not valid, the system will give the user 3 attempts to re-login and if he fails in one of them, the system will ban the user account.



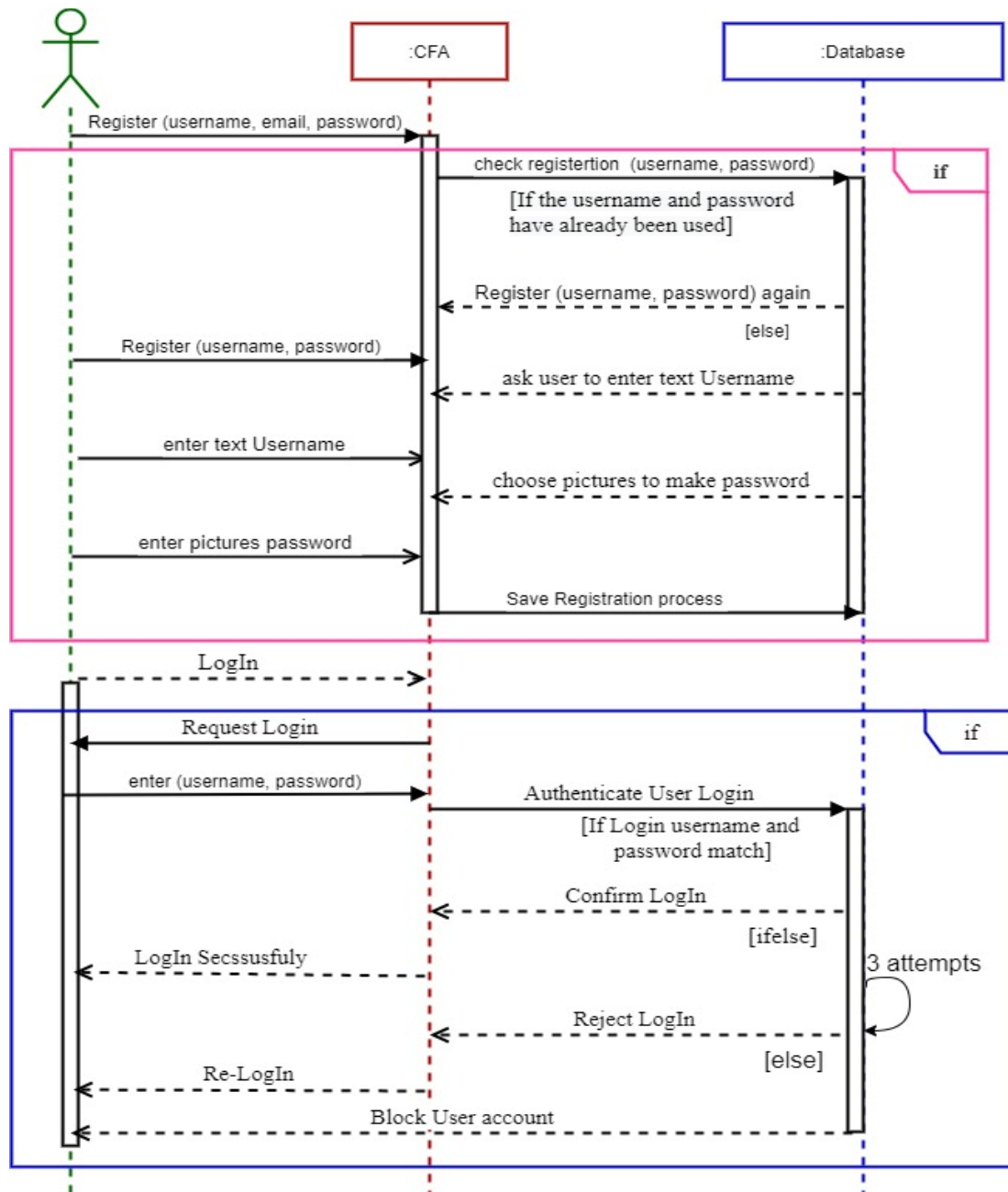


Figure10 :Sequence diagram for CEA

The following figure shows the activity diagram of the process of forgetting the password. When the system fetches the user's account due to the expiry of its attempts, the user will be able to request a password reset, thus the system will ask the user to confirm their pre-registered email. If the e-mail is correct, the system will send an e-mail to the user containing a link to reset the password, then the user will be able to create a new graphical password through the graphic images that will appear in front of him.

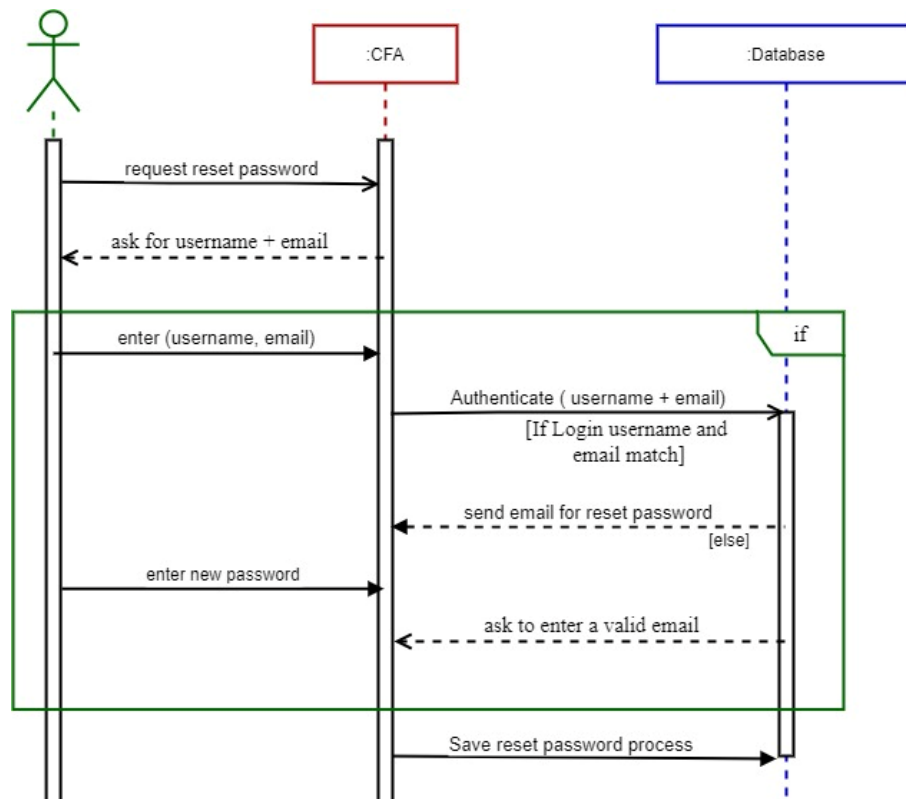
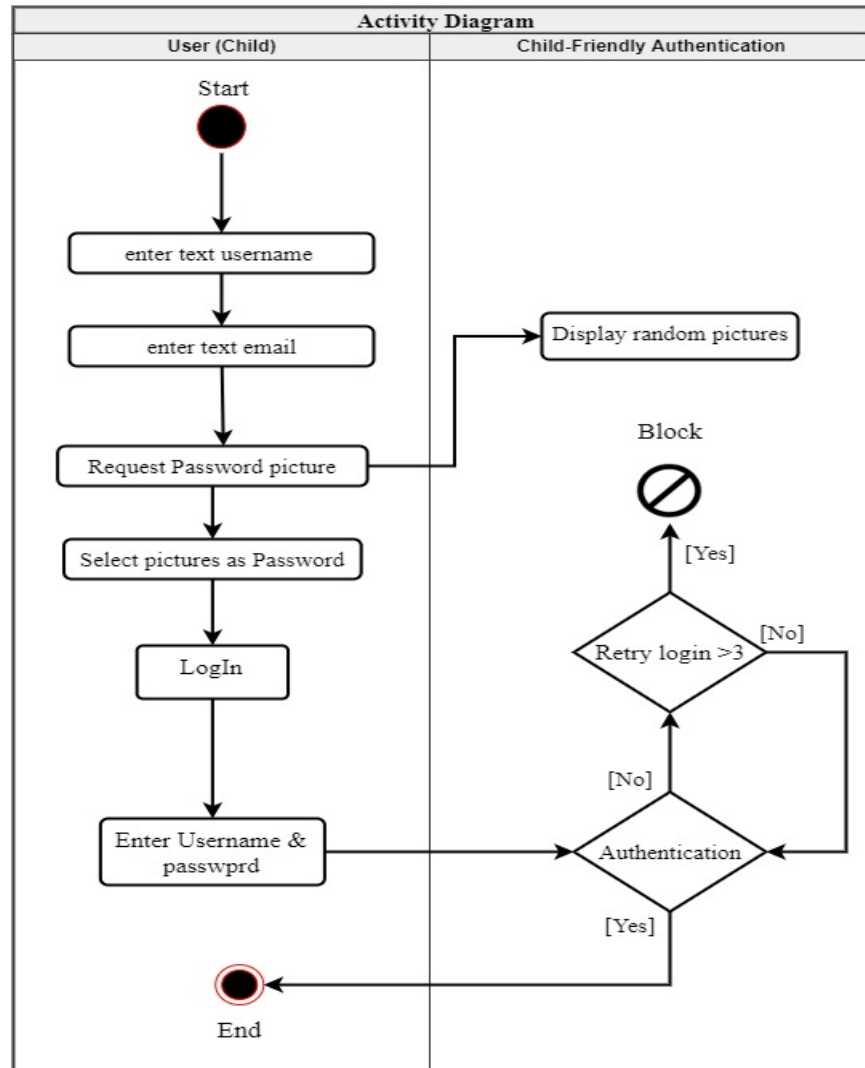


Figure 11 : Sequence diagram for Reset Password

### 3.5 Activity diagram

The activity diagram below for a child-friendly authentication system is a graphic representation that shows the workflow of activities in the system and explains step-by-step procedures with support for selection, repetition, and synchronization. The workflow in this system begins when the user (the child) enters the username and email as a text and then enters the password consisting of different images that the system displays to him and then logs in. At the login stage, the user enters the username and password again, and then the system performs authentication. If the password and username match the password and username registered by the user at the registration stage, the authentication currency is considered valid and the work ends. However, in case it does not match, the system gives 3 attempts to the user to re-login, and if all of them are consumed, the system bans his account.



**Figure 12 Activity diagram for user**

The following figure (13) shows the activity diagram of the administrator, where when he logs into the Django database using his username and password, he will be shown the site's own tables. If he chooses the images table, for example, the administrator will be able to modify the images, and similarly if he chooses the users table, the administrator will be able to modify the users by deleting, adding, and others.

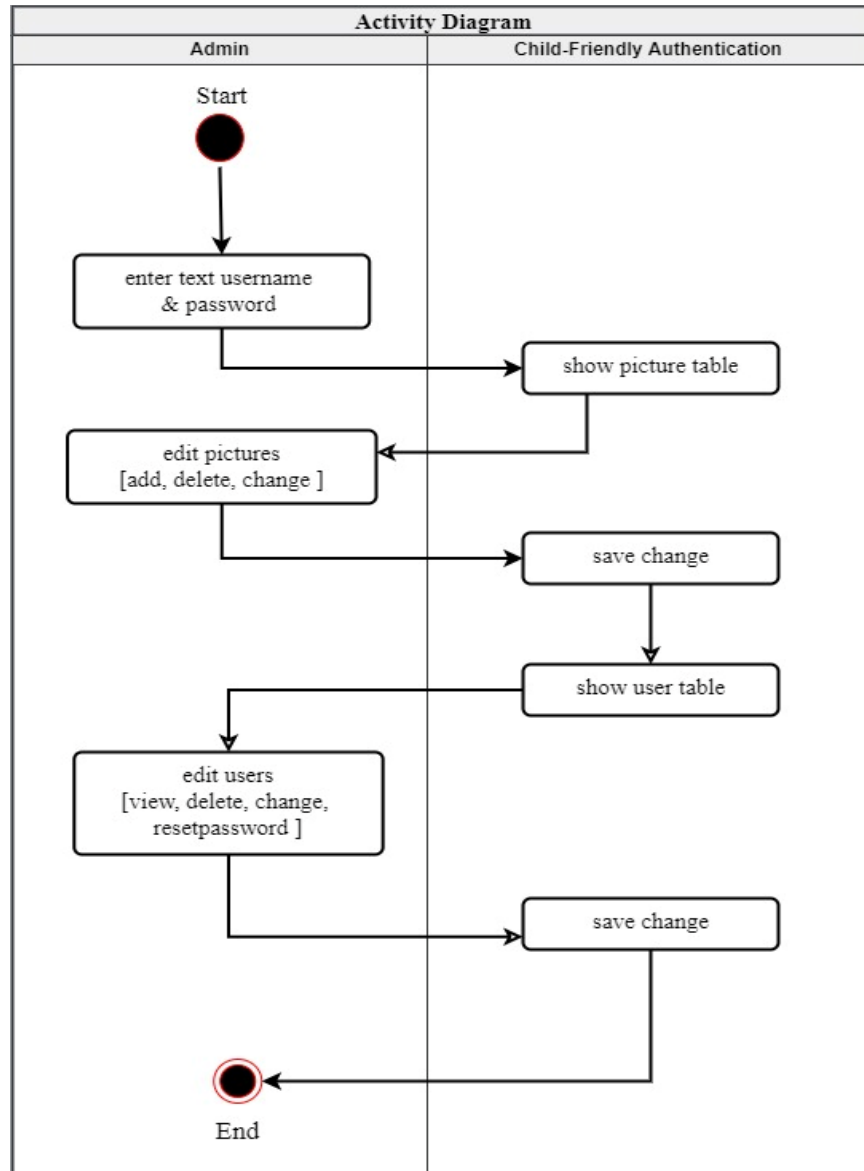


Figure 13 Activity diagram for admin

### 3.6 DFD diagram

Data flow diagrams are used to show the flow of data in a corporate information system graphically. DFD refers to the procedures used in a system to send information from the input to file storage and report production. CFA has 2 external entities users (child), and admin they provide data to the system or receive output from the system. A user's entity they have 2 process (register and login) that receives input data and produces output. Moreover, an admin's entity has 2 processes one for the image and the other for the user.

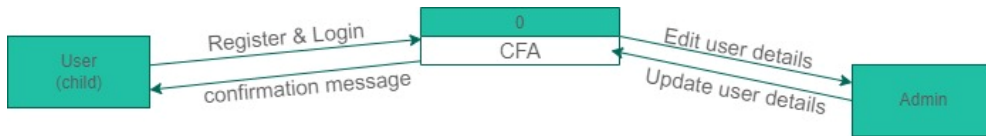


Figure 14: DFD diagram for CFA

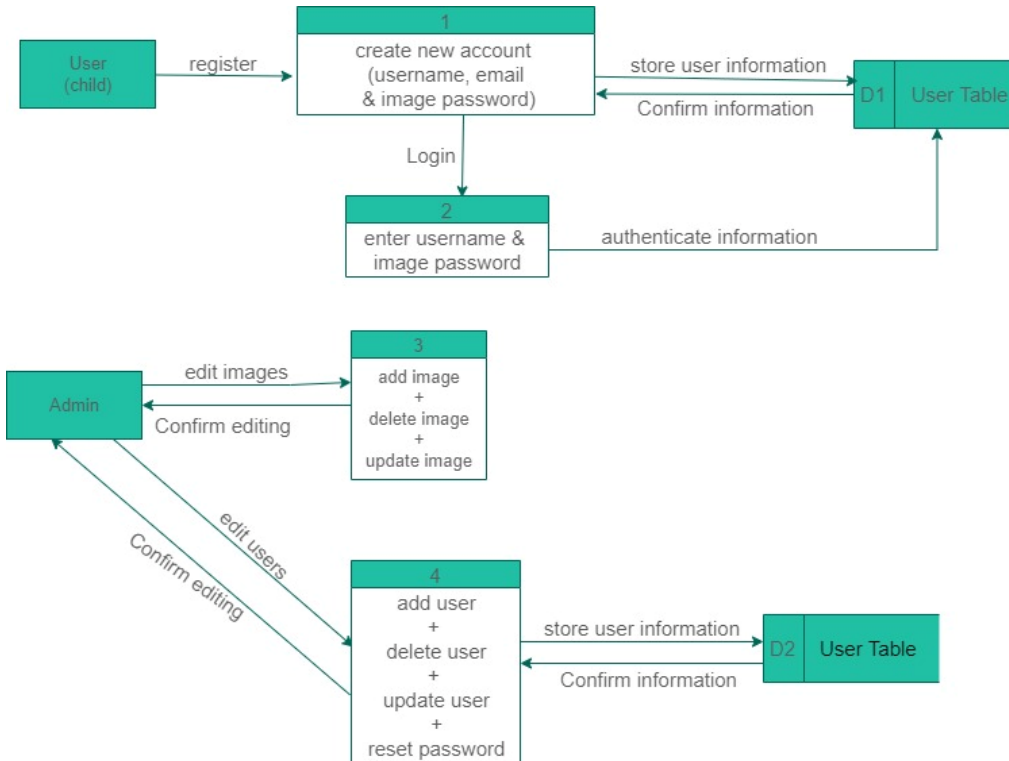


Figure 15: DFD diagram details

## 4. IMPLEMENTATION

### 4.1 Introduction

In this chapter, we will discuss our initial implementation of the system in terms of installing the required programs, languages, and packages, also start writing the required code.

### 4.2 Install programs, languages, and packages

We need PyCharm as an integrated development environment to implement our system on it and to install it visit this link:

<https://www.google.com/search?q=pycharm&oq=puchar&aqs=chrome.1.69i57j0i1014j46i512i2j0i512i3.8325j0j7&sourceid=chrome&ie=UTF-8>

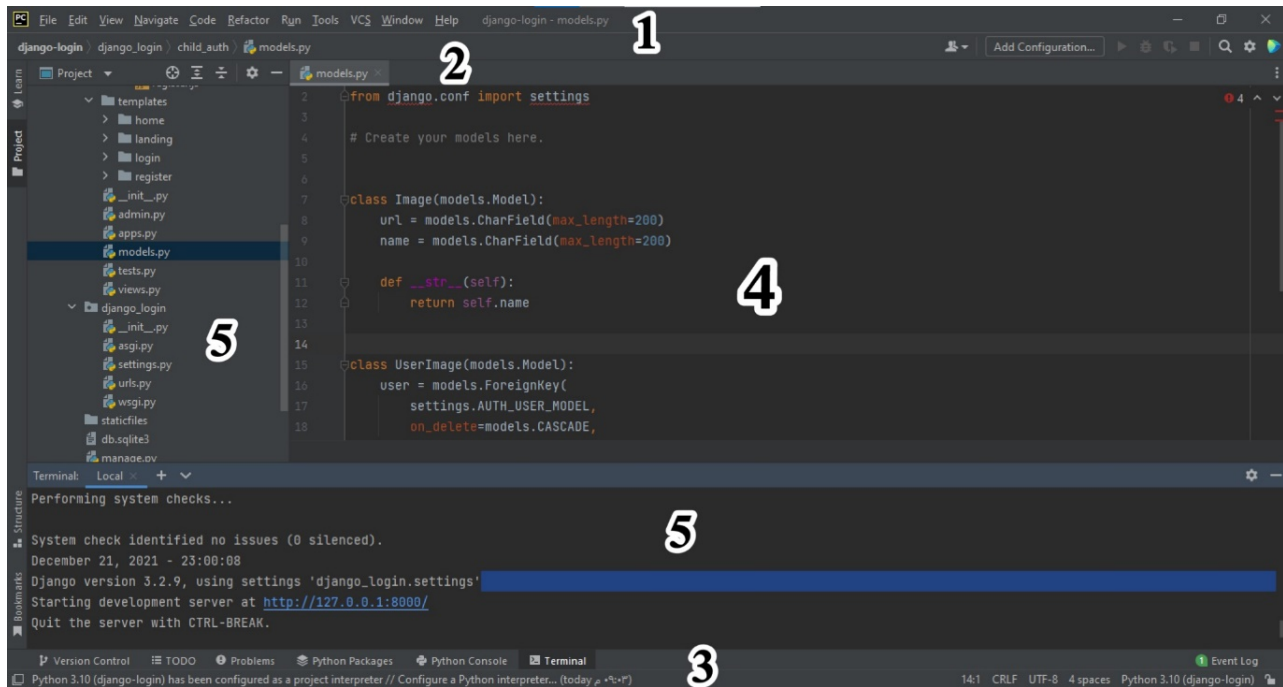


Figure 16: PyCharm user interface

- 1- **Main menu and toolbar:** Contain commands that affect the entire project or parts of the project, such as opening, creating projects, refactoring code, running, and debugging applications, and keeping files under version control.
- 2- **Navigation bar:** Quick replacement of the Project tool window. It is used to browse the entire project and open the file for editing.
- 3- **Status bar:** Indicates the state of the project, and the entire IDE, and displays various warnings and information messages.
- 4- **Editor:** Here, the code can be read, created, and modified.
- 5- **Tool windows:** A help window, providing access to project management, search, playback, debugging, integration with the version control system, and other tasks.

We will use Python as a programming language that lets us work quickly and integrate systems more effectively also it has several libraries and it's great for prototypes.

To install it: <https://www.python.org/>

Finally, we used Django as a framework in the PyCharm program since it provides almost the whole things we need to create authentication pages to handle login, log out, and password management. To install it, first must install it in pip in command prompt and then install it as a framework in PyCharm.

Command to install Django: `$ python -m pip install Django`

### 4.3 Creating a CFA project

Firstly, click on the New Project button in the QuickStart area of the Welcome screen in the PyCharm and select the desired project type (Django) and specify the project name and the location. Then, PyCharm automatically creates a new virtual environment and you don't need to configure anything.

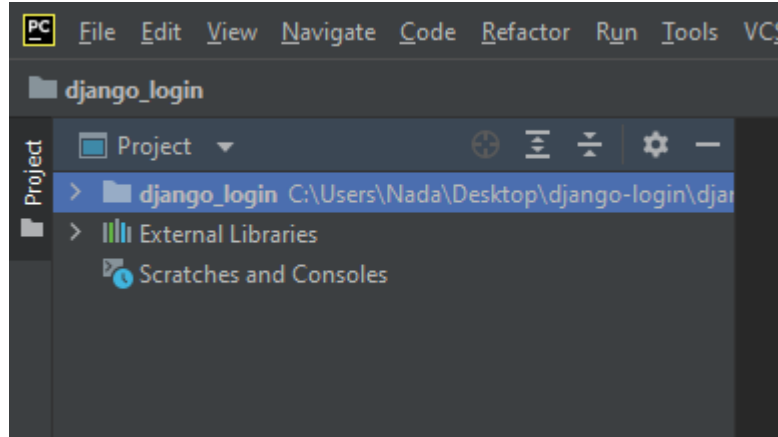


Figure17:Creating CFA project

#### 4.4 Create an admin page

The Django framework provides an administration page ready-made with every project that is created, and it has many benefits for managing the models used in the project as well as managing user groups and their ability to make modifications to the website. To be able to log in to the admin, we need a username and password. To create a user account, open the command prompt and run the following command:

```
python manage.py createsuperuser
```

Then you will be asked to enter your desired username for example fatma:

```
Username: fatma
```

Then the CMD will ask you to enter your email address:

```
Email address: fatma@example.com
```

Finally, enter the password twice:

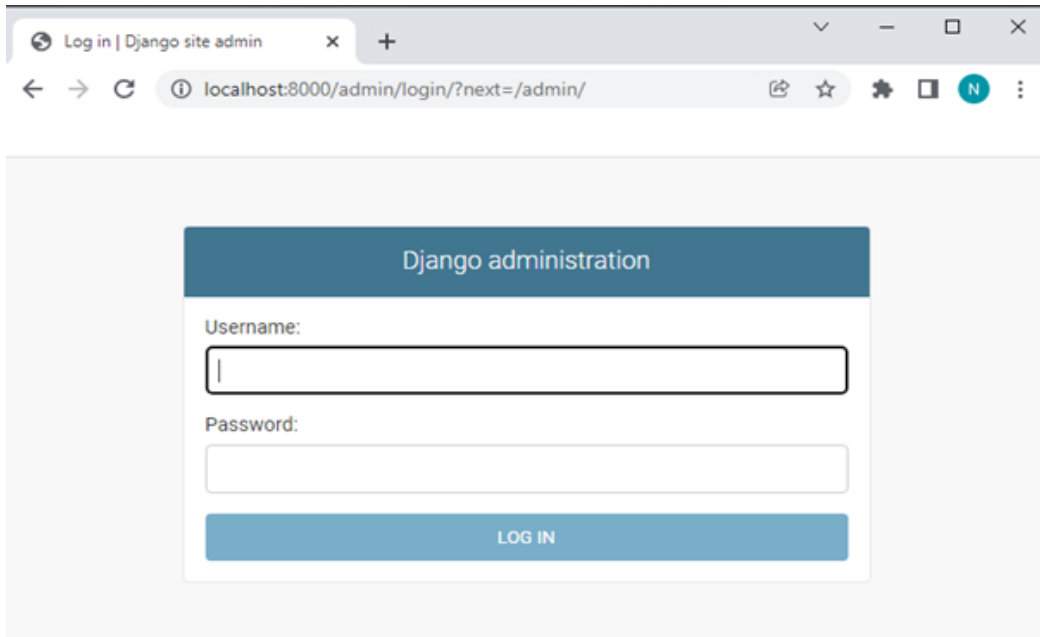
```
Password: *****
```

```
Password (again): *****
```

```
Superuser created successfully.
```

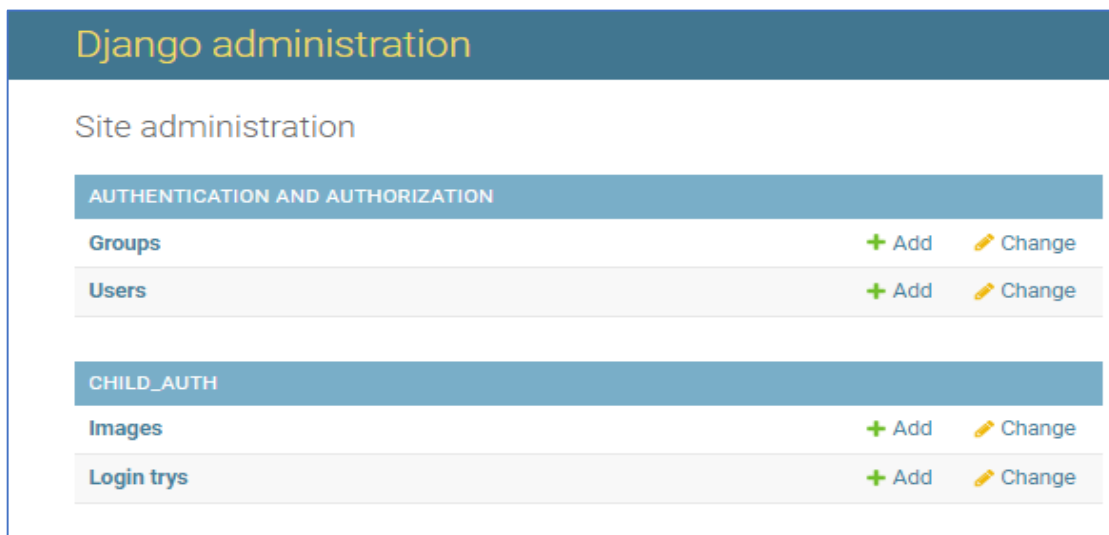
Now, open the browser on admin Django, where it will ask to use a path containing the word admin /, to open it start the server of Django and then head in the browser to the following address:

```
http://127.0.0.1:8000/admin
```



**Figure 18: Create admin page**

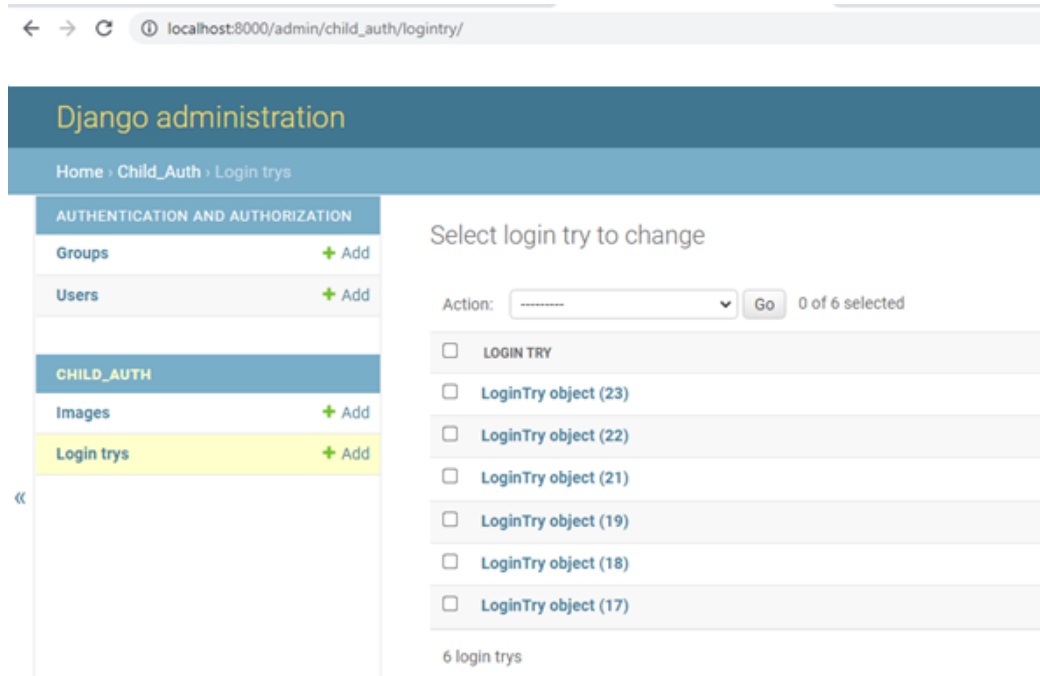
Then enter the username and password that was created a while ago, and you will now be able to access the administration page, which will appear as follows:



**Figure 19: Site administration**

On this page (Figure19), the administrator will be able to modify some things related to users and user groups and add other control interfaces to the forms in the project. As you can see, a CHILD\_AUTH model has been added, which includes two objects: Images and login try. In the Image object (Figure17), you can see all the images that will display for the user (child) as a password with their URL and name. In addition, in the Login Try table, we will see the login attempts that combine the username with the image name to apply the authentication process (Figure 20).

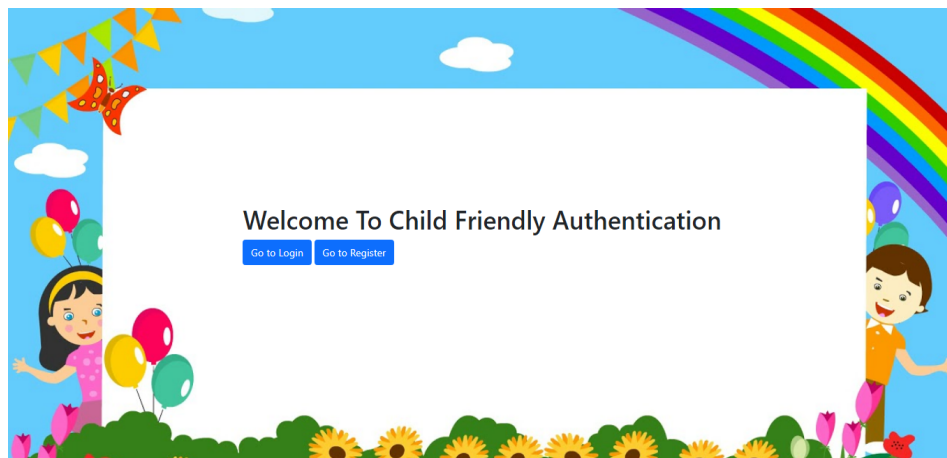




**Figure 20: Login tries in the administration page**

#### 4.5 Create an index page

In the index page, it will be the first page that appears to the user when running the project, which contains a welcome for the child and two options for him to either go to log in if the child is already registered or go to register as a new user.



**Figure 21: index page**

#### 4.6 Create Register page

If the child chooses to go to register, the registration page will appear, which contains the username, email, and the graphic password. To register successfully, the child must write the username and email in a text form and then choose from the images in front of him to form his password, user can move between the image field to configure his password then press the “Register” button to show him a notification that the registration process has been completed successfully.

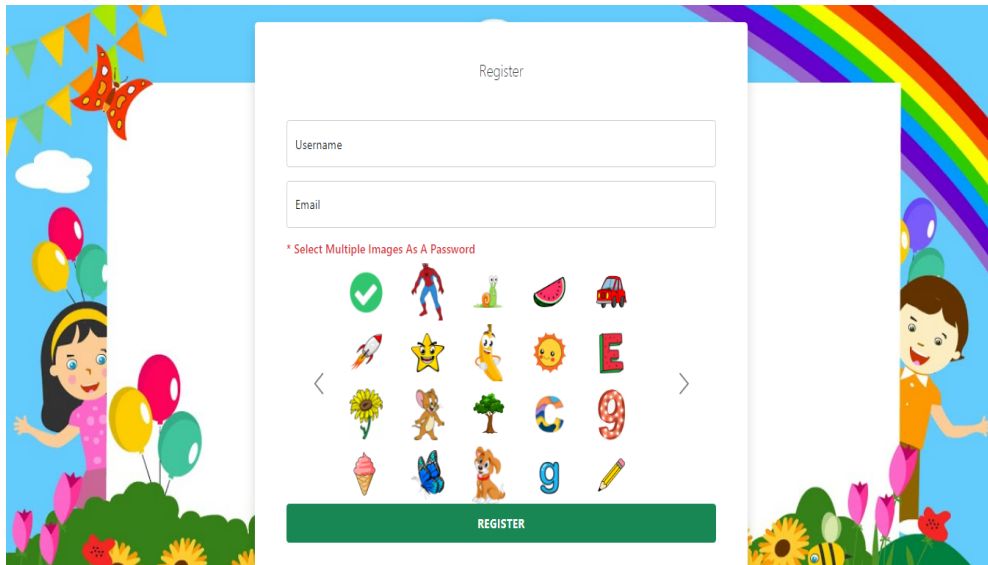


Figure 22: Register page

#### 4.7 Create Sign In page

Upon completion of the registration on the site, the login page will appear for the child, and he must re-enter his username and choose the same images he chose in the registration process as a password (Figure23).

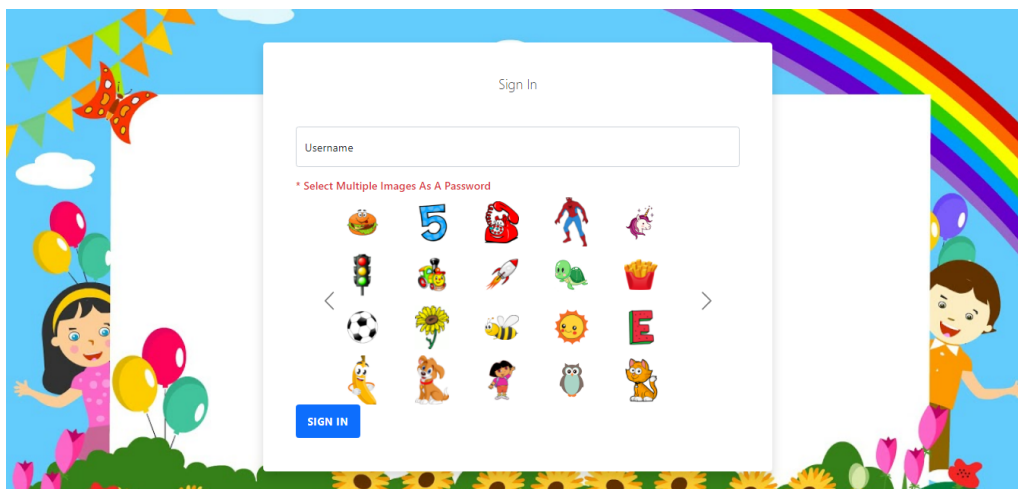


Figure 23: sign-in page

Then the system will authenticate the login process by matching the user data when registering and when logging in. If the authentication is correct, the home page will open when the user clicks the login button. And if it is not correct, the system will alert the user that the process failed and he has 3 attempts to re-login, and if he does not pass it, his account will be banned, and the reset password button will appear to him.

#### 4.8 Create a Home page

If the child's home page appears, it means that the login and authentication process is correct.



Figure 24: Home page

#### 4.9 Reset password page

If the user forgets his password and his account is blocked by thy system, he can reset his password easily through the reset password button (Figure 25) and then the system will ask him to enter his username and email which he enters in the registration (Figure 26). After that, an email will be sent to the user to reset his password (Figure 27).

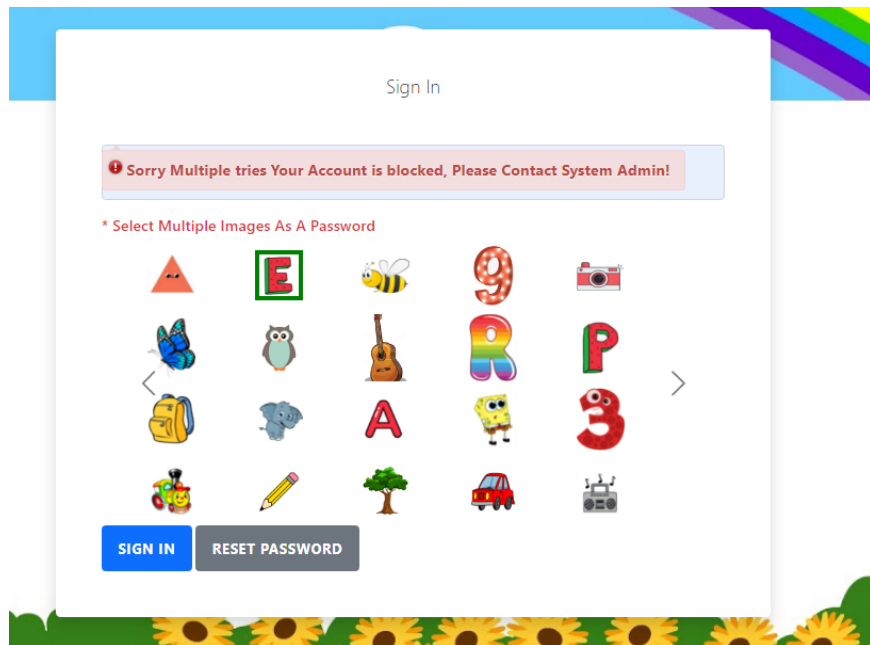


Figure 25: Reset password

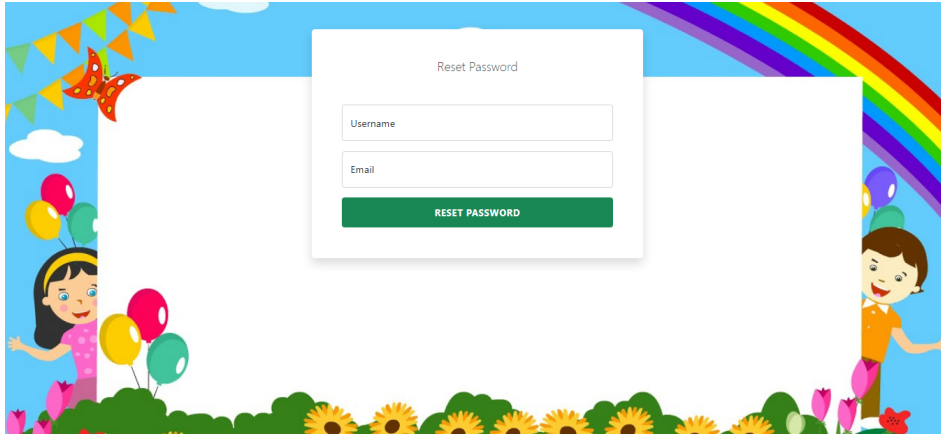


Figure 26: Email for Reset password page

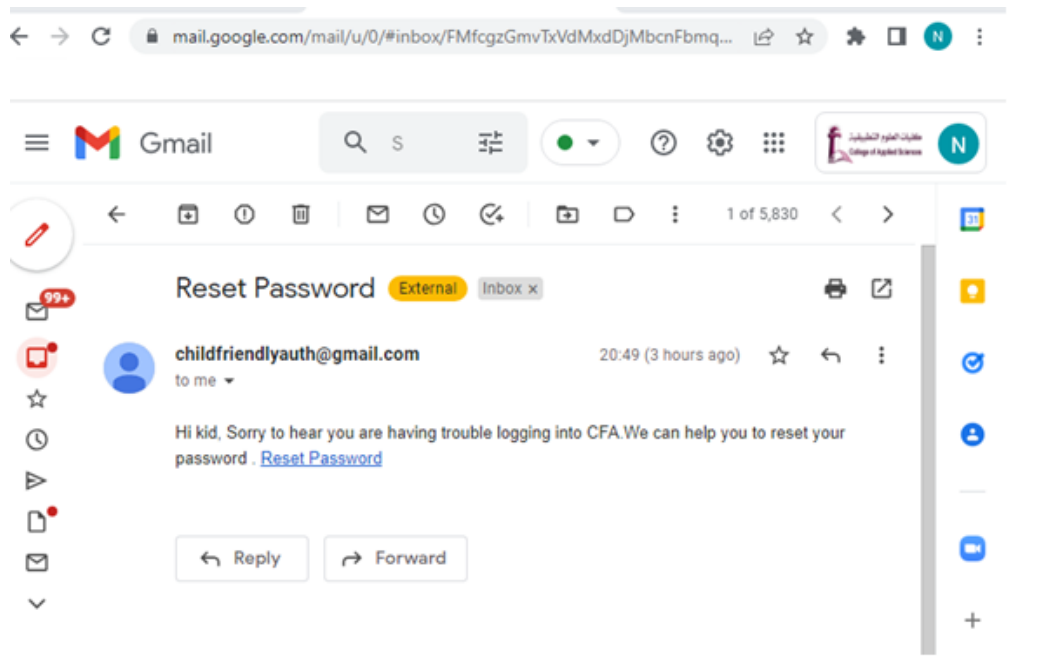
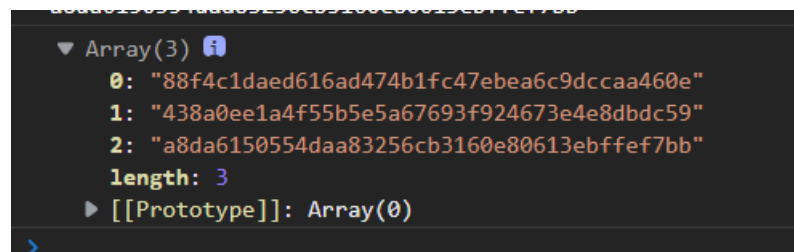
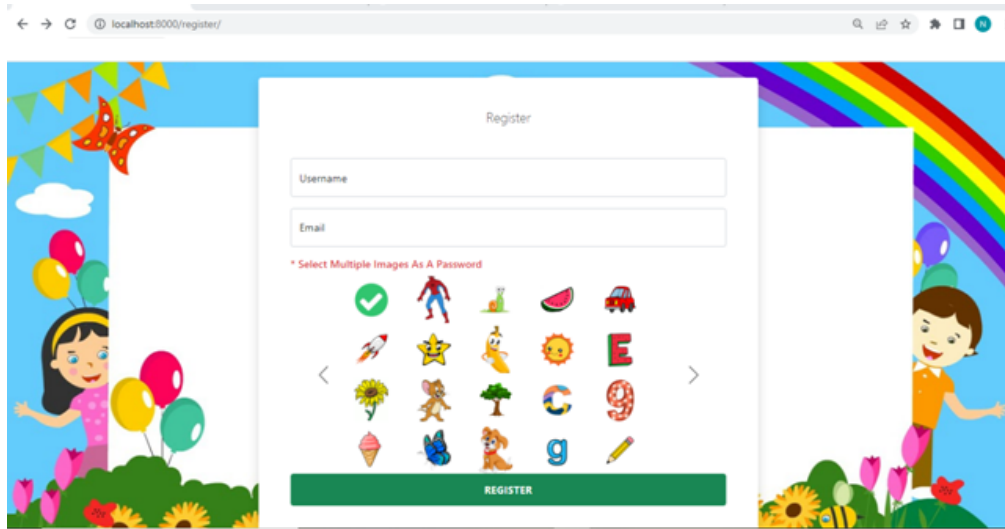


Figure 27: Reset password email

#### 4.10 Registration process

What happens when the user selects multiple images as a password,



First of all we read the image to base64 “`toDataURL()`” then we hash the base64 text to sha1 “`base64ToHash()`” and add the selected images to array ,

```
var list_selected_images = [];
```

```
const base64ToHash = function (string) {
  let hash = CryptoJS.SHA1(string)
  return CryptoJS.enc.Hex.stringify(hash)
}
```

```
const toDataURL = url => fetch(url)
  .then(response => response.blob())
  .then(blob => new Promise((resolve, reject) => {
    const reader = new FileReader()
    reader.onloadend = () => resolve(reader.result)
    reader.onerror = reject
    reader.readAsDataURL(blob)
  })))
```

```
const selectImage = function (element) {
  let imageUrl = $(element).attr('src')
  toDataURL(imageUrl)
  .then(dataUrl => {
    let hash = base64ToHash(dataUrl);
    let index = list_selected_images.indexOf(hash);
    console.log(hash)
  })
}
```

```

    if (index !== -1) {
      $(element).removeClass('img-selected')
      list_selected_images.splice(index, 1)
    }
    else {
      $(element).addClass('img-selected')
      list_selected_images.push(hash)
    }
    console.log(list_selected_images)
  })
}

```

Then we join the array to one big string, and we hash again for good security and we have the hashed password

```

//convert the array to one hash
let hashImagePasswd = base64ToHash(list_selected_images.join())
console.log(hashImagePasswd);

```

How images show in deferent position every time  
shuffle array to randomize the position of elements in the array

```

function shuffle(array) {
  let currentIndex = array.length, randomIndex;

  // While there remain elements to shuffle...
  while (currentIndex !== 0) {

    // Pick a remaining element...
    randomIndex = Math.floor(Math.random() * currentIndex);
    currentIndex--;

    // And swap it with the current element.
    [array[currentIndex], array[randomIndex]] = [
      array[randomIndex], array[currentIndex]];
  }

  return array;
}

```

Create images array then shuffle it

```

for (let i = 1; i <= 25; i++) {
  images_grid[i] = 'image (' + i + ').png'
}
shuffle(images_grid)

```

create the images grid from the array

```

images_grid.forEach(element => {
  if (element) {
    let tmp = image_template.cloneNode(true)
    let imageName = element
    tmp.firstElementChild.src = '/static/images/gallery/' + imageName;
    tmp.firstElementChild.id = imageName;
    tmp.firstElementChild.alt = imageName;
    auth_gallery.appendChild(tmp)
  }
});

```

Login gets the username and password from a post request and then checks in the **User** table if exists it returns a success message if the user tries > 3 and all false then returns an error then the account is blocked.

Register only takes username and password and stores them in the **User** table:

```

@csrf_exempt
def login(request):
    is_ajax = request.META.get('HTTP_X_REQUESTED_WITH') == 'XMLHttpRequest'
    if is_ajax and request.method == "POST":
        # request.raw_post_data w/ Django < 1.4
        json_data = json.loads(request.body)
        username = json_data['username']
        password = json_data['password']
        #print('Raw Data:', json_data)
        if User.objects.filter(username=username).exists():
            tryuser = User.objects.get(username=username)
            # get how many tries he tried yet
            tries_count = LoginTry.objects.filter(
                user=tryuser, status=False, date__lt=datetime.today()).count()
            # print(tries_count)
            # if tries > 3 and all false then return error than account is blocked
            if tries_count >= 3:
                return JsonResponse({"code": -1, "message": "Sorry Multiple tries Your Account is blocked, Please Contact System Admin!"}, status=200)
            user = authenticate(username=username, password=password)
            if user is None:
                # log the action in the login then return with error
                # get the user who tryed the login
                LoginTry.objects.create(
                    user=tryuser, status=False, date=datetime.today())
                return JsonResponse({"code": -1, "message": "wrong Credentials, Remaining Tries is ({}).format(3 - tries_count)}, status=200)
            else:
                LoginTry.objects.create(
                    user=tryuser, status=True, date=datetime.today())
                return JsonResponse({"code": 0, "message": "success"}, status=200)
        else:

```

```

        return JsonResponse({"code": -1, "message": "wrong Credentials"}, status=200)
    else:
        return render(request, 'login/login.html')

@csrf_exempt
def register(request):
    is_ajax = request.META.get('HTTP_X_REQUESTED_WITH') == 'XMLHttpRequest'
    if is_ajax and request.method == "POST":
        # request.raw_post_data w/ Django < 1.4
        json_data = json.loads(request.body)
        username = json_data['username']
        password = json_data['password']
        try:
            user = User.objects.create_user(username, "", password)
            return JsonResponse({"code": 0, "message": "user created successfully"}, status=200)
        except:
            return JsonResponse({"code": -1, "message": "sorry user is not created!"}, status=200)
    else:
        return render(request, 'register/register.html')

```

### Another way of generating a password:

user selects images as an array -> we send the array to python code (backend) and save selected images IDs as passwords in another table. However, its complicated and causes heavy load on server-side code and we don't want to do that.

## 5. TESTING

### 5.1 Unit Testing

Unit testing is a software development technique in which the smallest testable pieces of a program, referred to as units are examined separately and independently for correct functioning. Unit testing's major goal is to separate written code in order to test and verify if it works as intended. In addition, unit tests save time and money by identifying faults early in the development cycle. It also assists developers in knowing the testing code base and allowing them to make changes rapidly. Good unit tests benefit from code reuse and act as documentation for projects (Jamil et al.,2016).

Figure (28) CFA Application Login shows that when the user tries to register to the site by forgetting to type one of the required data, a message will appear to notify the user, please fill in the required fields.



Figure 28: Unit testing 1

In Figure (29) when the user forgets to enter the password composed of images. A message will appear to the user, you must choose at least one image.

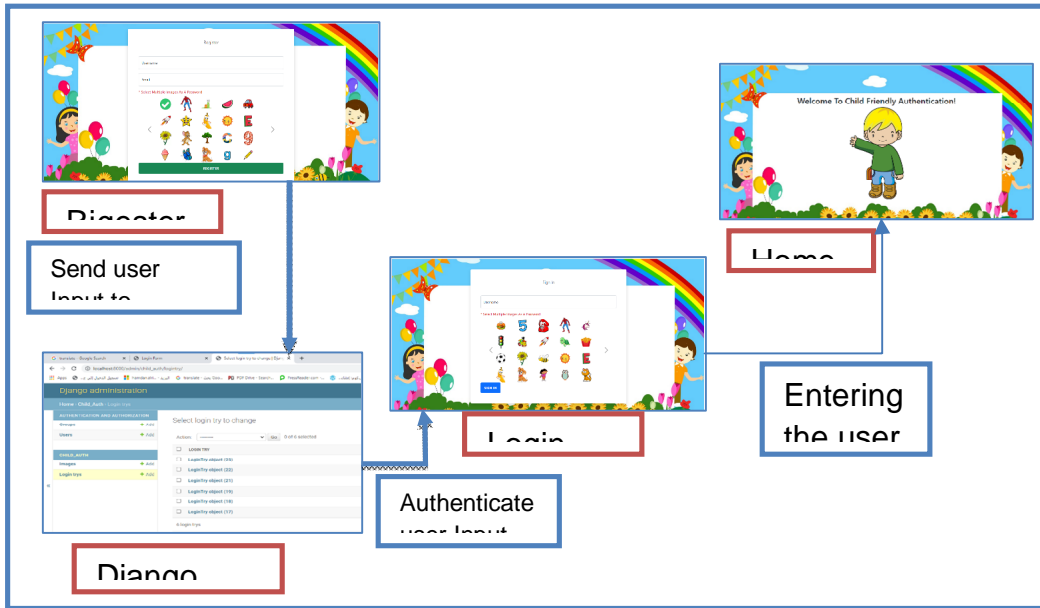
Figure 29: Unit testing 2

## 5.2 Integrated Testing

Integration testing is a type of testing in which software modules are conceptually connected and tested as a unit. A typical software project is made up of several software modules written by several programmers. The goal of this level of testing is to find faults in how these software components interact

when they're put together. Integration testing is concerned with ensuring that data is communicated between various units (Jamil et al.,2016).

First, the user creates an account for him on the registration page, then the system stores the user's data in the database. When the user enters his data on the login page, the system authenticates the user data he has entered and opens the site for him. If the authentication fails, the system sends a notification to the user that the data he entered is incorrect (Figure 30).



### 5.3 Additional Testing

The figure below shows a message that appears to the user when he tries to log in. The message states the number of attempts left for the user to close his account.

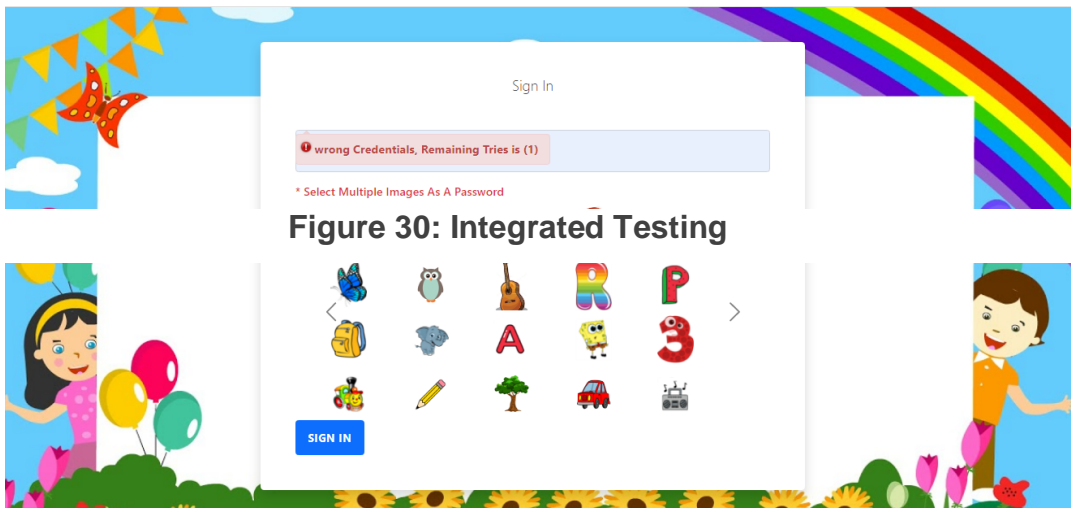
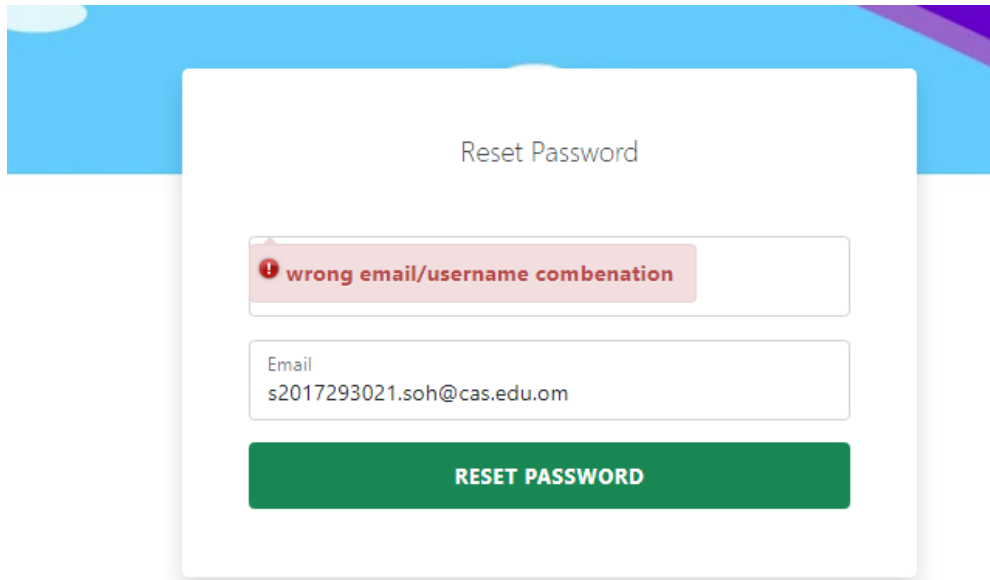


Figure 31: Test login tries

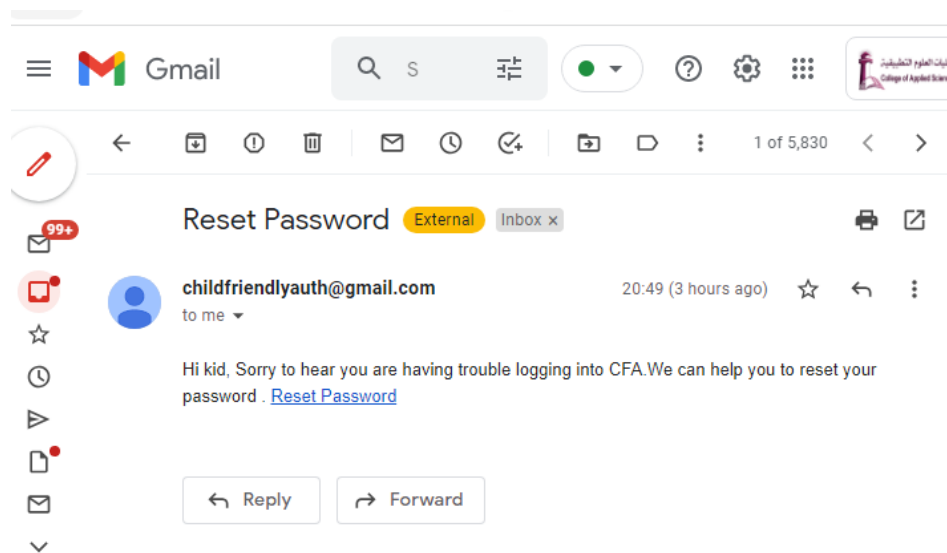
When the user requests to reset the password, a page appears for him to match the username with the email previously registered in the system. If authentication is done, the user will be able to continue

with the password reset process. And if the email does not match the username, a message will appear to the user that does not match, as shown in the picture (32).



**Figure 32: Test reset password**

As shown in Figure (33), the site sends a link in the user's email to reset the password when his password is forgotten. So, when the user clicks on the link, it will open a new page to set the new password.



**Figure 33 test reset password email**

## 6. RESULTS AND DISCUSSION

### 6.1 Expected Result

Our project aims to build an authentication system that matches their age, making it easier for them to use. It also aims to protect children from exposure to Internet threats resulting from not protecting their devices and accounts using traditional passwords. The system procedures include the following :Register

for the system by entering a username and email and choosing a password from the image field. Moreover, login to the system by authenticating the username and password used for registration.

## 6.2 Actual Results

The system functions as intended, and the prototype demonstrated that everything in the system is in good working order.

Please scan the QR code to try the CFA:



## 7. CONCLUSION

In conclusion, the past decade has seen a growing interest in using graphical passwords as an alternative to traditional text passwords, arguing that people are better at memorizing graphic passwords than text-based passwords, and through research and studies that it is difficult to crack graphic passwords using traditional attack methods. Child-friendly authentication is a suggested method for an authentication interface that caters to the needs of children to them to use graphic passwords. In general, the importance of child-friendly authentication lies in establishing a technological encourage generation, given the transitional phase the world is witnessing to the world of digital technology.

## 8. FUTURE WORK

Some future work can be done to improve the system, such as adding the family supervision feature, which is to limit the duration of use and block or restrict certain applications and features, such as restricting purchases and downloads. Also, we will work on making this system one of the sign in options found in the operating systems of the devices.

## REFERNCES

- [1] Yang, T. Y., Shamala, P., Chinniah, M., & Foozy, C. F. M. (2021, February). Graphical Password Authentication For Child Personal Storage Application. In *Journal of Physics: Conference Series* (Vol. 1793, No. 1, p. 012065). IOP Publishing.
- [2] Jebriel, S., & Poet, R. (2014, March). Automatic registration of user drawn graphical passwords. In *2014 6th International Conference on Computer Science and Information Technology (CSIT)* (pp. 172-177). IEEE.

- [3] Sabzevar, A. P., & Stavrou, A. (2008, November). Universal multi-factor authentication using graphical passwords. In 2008 IEEE international conference on signal image technology and internet based systems (pp. 625-632). IEEE.
- [4] Izadeen, G. Y., & Ameen, S. Y. (2021). Smart android graphical password strategy: A review. *Asian Journal of Research in Computer Science*, 59-69.
- [5] Masrom, F. T. (2009). A Survey on Recognition-Based Graphical User Authentication Algorithms. *International Journal of Computer Science and Information Security*, Vol. 6, No. 2, pp. 119–127.
- [6] Blonder, G. E. (1996). Graphical password. U.S. Patent 5559961. Murray Hill: Lucent echnologies, Inc.
- [7] Ratakonda, D. K. (2019, June). Children's Authentication: Understanding and Usage. In *Proceedings of the 18th ACM International Conference on Interaction Design and Children* (pp. 743-746).
- [8] Cole, J., Walsh, G., & Pease, Z. (2017, June). Click to enter: Comparing graphical and textual passwords for children. In *Proceedings of the 2017 Conference on Interaction Design and Children* (pp. 472-477).
- [9] Assal, H., Imran, A., & Chiasson, S. (2018). An exploration of graphical password authentication for children. *International Journal of Child-Computer Interaction*, 18, 37-46.
- [10] Gao, H. (2009). Design and Analysis of a Graphical Password Scheme. *Innovative Computing, Information and Control (ICICIC)*.
- [11] Shrestha, R. (2010). Color Vision Defects in School Going Children. *Journal of the Nepal Medical Association*, 264-266.
- [12] Shrestha, S. (2016). A Universally Designed and Usable Data Visualization for A Mobile Application in the Context of Rheumatoid Arthritis. *International Journal of Advanced Computer Science and Applications*, 7.
- [13] Pacchioli, D. (2005). Probing Question: Do children have better memories than adults do? Penn State: Infant Brain Development and Cognition Laboratory .
- [14] Stobert, E. (2017). Teaching Authentication in High Schools:Challenges and Lessons Learned.
- [15] Imran, A. (2016). An Exploration of Graphical Password. Canada: School of Computer Science, Carleton University.
- [16] Michael Burrows, Martin Abadi, Roger Needham, "A logic of authentication", *ACM Transactions on Computer Systems (TOCS)*, Volume 8, Issue 1 (February 1990), Pages: 18 – 36.
- [17] Akula, S., & Devisetty, V. (2004, April). Image based registration and authentication system. In *Proceedings of midwest instruction and computing symposium* (Vol. 4, p. 5).
- [18] Ometov, A., Bezzateev, S., Mäkitalo, N., Andreev, S., Mikkonen, T., & Koucheryavy, Y. (2018). Multi-factor authentication: A survey. *Cryptography*, 2(1), 1.
- [19] Lunde, L., & Wangensteen, A. (2006). *Using SIM for strong end-to-end Application Authentication* (Master's thesis, NTNU).
- [20] Rittenhouse, R. G., Chaudry, J. A., & Lee, M. (2013). Security in graphical authentication. *International Journal of Security and Its Applications*, 7(3), 347-356.
- [21] Hu, W., Wu, X., & Wei, G. (2010, October). The security analysis of graphical passwords. In *2010 International Conference on Communications and Intelligence Information Security* (pp. 200-203). IEEE.
- [22] Forcier, J., Bissex, P., & Chun, W. J. (2008). *Python web development with Django*. Addison-Wesley Professional.
- [23] Jamil, M. A., Arif, M., Abubakar, N. S. A., & Ahmad, A. (2016, November). Software testing techniques: A literature review. In *2016 6th international conference on information and communication technology for the Muslim world (ICT4M)* (pp. 177-182). IEEE.

