

## Graphical Password Authentication

N.Sudha Laxmaiah, Akhila Pagidimarri, Deekshita Ryaka, Hrithika Reddy Boyapally

<sup>1</sup> Associate Professor, Department Of Cse, Bhoj Reddy Engineering College For Women, India.

<sup>2,3,4</sup> B. Tech Students, Department Of Cse, Bhoj Reddy Engineering College For Women, India.

### ABSTRACT

Graphical passwords present a compelling alternative to traditional alphanumeric passwords, capitalizing on the human cognitive ability to recall visual information more effectively than textual data. This extended abstract introduces a novel graphical password authentication system designed to enhance security while maintaining user-friendliness. The proposed system allows users to select a personal image and define multiple points-of-interest (POIs) within it, each associated with a specific word or phrase. Users can further strengthen their password by enforcing the order of POI selection, adding an additional layer of security. User authentication is a critical pillar of computer security, ensuring access control and user accountability. However, conventional alphanumeric username/password systems face significant challenges: users often choose short, predictable passwords that are susceptible to dictionary and brute-force attacks, while stringent password policies may lead to users writing down complex passwords, exposing them to theft. To address these issues, alternative approaches such as passphrases—longer, memorable phrases—and graphical passwords, where users interact with images via a graphical user interface, have been proposed. Graphical passwords leverage the infinite search space of images to increase resistance to brute-force attacks and enhance memorability by allowing users to click on specific image regions rather than typing characters. The proposed system, implemented using Visual Basic .NET 2005 and Python, integrates graphical elements, text, POI

order, and number to deliver multi-factor authentication within an intuitive framework. By combining the strengths of both graphical and text-based approaches, the system achieves a balance between robust security and ease of use, offering a large password space that is resistant to common attack vectors. This paper details the system's operation through illustrative examples, such as a user selecting a family photo and associating names with specific regions, and highlights key features, including its flexibility in allowing users to customize images and POIs, its multi-factor authentication capabilities, and its potential to mitigate the vulnerabilities of traditional passwords. The system's implementation, feasibility, and alignment with user needs are also discussed, underscoring its promise as an innovative and effective authentication solution for modern security contexts.

### 1-INTRODUCTION

User authentication is a fundamental component in most computer security contexts. It provides the basis for access control and user accountability [1]. While there are various types of user authentication systems, alphanumeric username/passwords are the most common type of user authentication. They are versatile and easy to implement and use. Alphanumeric passwords are required to satisfy two contradictory requirements. They have to be easily remembered by a user, while they have to be hard to guess by an impostor [2]. Users are known to choose easily guessable and/or short text passwords, which are an easy target of dictionary and brute-force

attacks [3, 4, 5]. Enforcing a strong password policy sometimes leads to an opposite effect, as a user may resort to writing his or her difficult-to-remember passwords on sticky notes exposing them to direct theft.

In the literature, several techniques have been proposed to reduce the limitations of alphanumeric passwords. One proposed solution is to use an easy to remember long phrases (passphrase) rather than a single word [6]. Another proposed solution is to use graphical passwords, in which graphics (images) are used instead of alphanumeric passwords [7]. This can be achieved by asking the user to select regions from an image rather than typing characters as in alphanumeric password approaches.

In this extended abstract, we propose a graphical password authentication system. The system combines graphical and text-based passwords trying to achieve the best of both worlds. In section 2, we provide a brief review of graphical passwords. Then, the proposed system is described in section 3. In section 4, we briefly discuss implementation and highlight some aspects about the proposed system.

### Existing System

Graphical passwords utilize images or drawings as an authentication mechanism, capitalizing on the human tendency to recall visuals better than text. They are theoretically more resistant to brute-force attacks due to their vast password space. Graphical password techniques are divided into recognition-based, where users identify pre-selected images, and recall-based, where users reproduce specific actions, such as clicking predetermined locations on an image. An early recall-based approach by Greg Blonder (1996) involved users clicking specific image locations to create and authenticate passwords. PassPoints later refined this concept. However, existing systems face challenges: users may struggle to remember precise points or

sequences on complex images, increasing cognitive load and error rates, which can lead to frustration. Security risks like shoulder surfing persist, and predictable point selections weaken defenses. Additionally, these systems often lack flexible recovery options and may overwhelm users with complex images, limiting usability.

### Proposed System

The proposed graphical password authentication system offers a hybrid approach combining graphical and text-based elements for enhanced security and usability. During registration, users upload a personal image and select multiple points of interest (POIs), each defined by a circular region (center and radius). For each POI, users can associate a word or phrase, or leave it as an empty string for simplicity. Users can choose whether the order of POI selection is enforced, allowing for stronger or more flexible passwords. For authentication, users enter their username, view the registered image, select the correct POIs, and input the associated text. The system masks typed text for security. This approach enhances security by resisting brute-force and dictionary attacks, reduces shoulder surfing risks through image-based interaction, and improves usability by leveraging memorable visuals and intuitive interfaces. It also supports personalization, as users can choose meaningful images, and is accessible to those with cognitive challenges.

## 2-REQUIREMENT ANALYSIS

### Functional Requirements

These are the requirements that refer to the specific actions, behaviors, or tasks a system or application is designed to perform. Functional requirements describe what the system must do to achieve its objectives and typically outline features, inputs, outputs, and interactions.

**Admin Module:**

- Admin Login
- View Users
- Manage Users
- Logout

**Non-Functional requirements**

These are the requirements that refers to the quality attributes or characteristics of a system that do not directly relate to its specific tasks but focus on how the system performs under certain conditions. These requirements address performance, usability, reliability, scalability, and other operational aspects.

- Scalability : Ability to handle a growing number of users.
- Usability : User-friendly interface that is easy to navigate.
- Reliability : Maintain high system availability to prevent critical periods.
- Security : Securely stored and to restrict unauthorized actions.
- Compatibility : It should be compatible with modern web browsers.
- Portability : software to be transferred from one system to another system

**Computational Resource Requirements****Hardware Requirements:**

- Processor : Intel i7
- RAM : 16.0 GB
- Hard Disk : 512 GB

**Software Requirements:**

- Operating System : Windows 10 Pro
- IDE : Visual Basic .net 2005 (VB.net)
- Web server : MySQL
- Code Behind : Python, React, Node.js.

**2.4 Life Cycle Model**

The Spiral Model is an iterative software development approach that combines elements of both design and prototyping. Based on the image provided, I can explain the four key phases shown in the diagram.

The Spiral Model works through repeated cycles (iterations) of these four phases, with each iteration representing a "spiral" that moves outward from the center as the project progresses. This approach allows for incremental development where each cycle builds upon the previous one.

**3-DESIGN**

Project architecture represents the number of components we are using as a part of our project and the flow of request processing i.e. what components in processing the request and in which order. An architecture description is a formal description and representation of a system organized in a way that supports reasoning about the structure of the system. Architecture is of two types. They are:

- Software Architecture
- Technical Architecture

**Software Architecture**

Software architecture design tools help to build software that does not have security issues. This is key because there are software risks in all areas of the software development process. When teams avoid software flaws or bugs, they can move forward with confidence. However, since this is not always possible, software architecture design tools also need to have the ability to find flaws during the creation of software and correct them efficiently. When using software architecture design tools that can identify flaws, you will have the ability to analyse the fundamental software design, assess the chance of an attack, figure out potential threat elements, and identify any weaknesses or gaps in existing security.

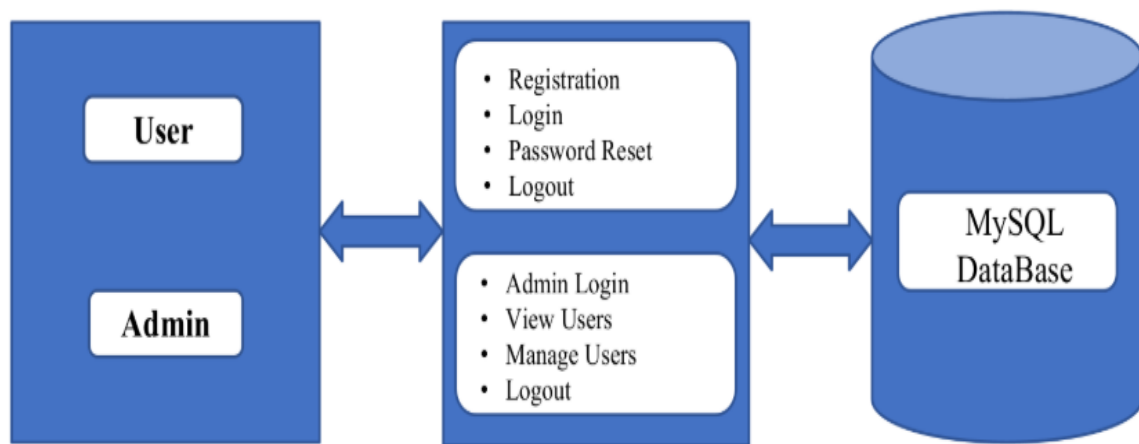


Fig. 3.1 Software Architecture

### Technical Architecture

Technical Architecture is a form of IT architecture that is used to design computer systems. It involves

the development of a technical blueprint regarding the arrangement, interaction, and interdependence of all elements so that system-relevant requirements are met.

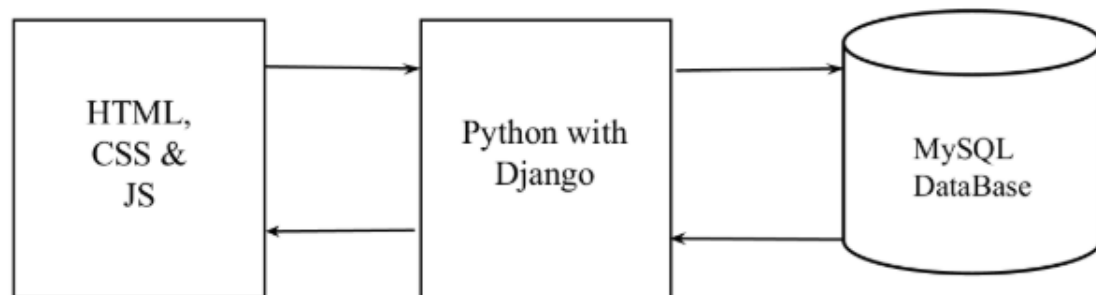


Fig.3.1.2 Technical Architecture

### Implementation

This system is developed using python programming language.

### Python

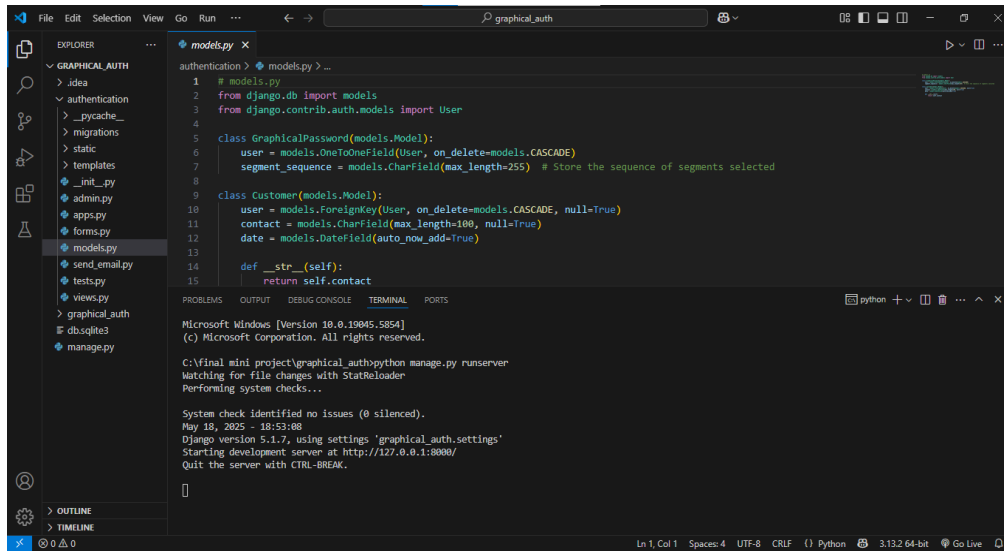
Python is one of the most popular programming languages now existing. The main reason for the creation of a programming language like python was to enhance the features to a large extent that were available in the present existing languages. The other reason was to invent a language which can be

used easily for the developers who work a lot on media other than texts like speech, images and videos. The other important reason was to increase the built-in functions so as to reduce the number of lines in the codes and implement simplicity. Python is basically created in such a way that the garbage is involuntarily and automatically collected. The Python language can be called as a mixture of all the languages with more features added to it. It is a structured language yet it does not support the use of

the semicolons at the end of each operation. Python consists of a very large standard library which consists of a huge number of built-in functions

which reduce the developer's load of writing hundreds of lines to perform a single and simple task.

#### 4-SCREENSHOTS



```

authentication > models.py > ...
1 # models.py
2 from django.db import models
3 from django.contrib.auth.models import User
4
5 class GraphicalPassword(models.Model):
6     user = models.OneToOneField(User, on_delete=models.CASCADE)
7     segment_sequence = models.CharField(max_length=255) # Store the sequence of segments selected
8
9 class Customer(models.Model):
10     user = models.ForeignKey(User, on_delete=models.CASCADE, null=True)
11     contact = models.CharField(max_length=100, null=True)
12     date = models.DateField(auto_now_add=True)
13
14 def __str__(self):
15     return self.contact
  
```

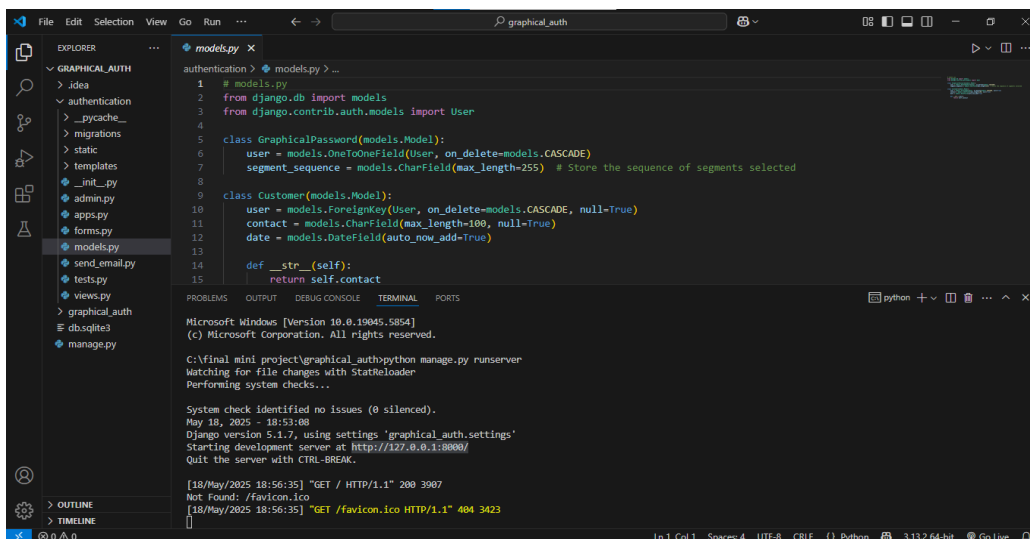
```

Microsoft Windows [Version 10.0.19045.5854]
(c) Microsoft Corporation. All rights reserved.

C:\final mini project\graphical_auth>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
May 18, 2025 - 18:53:08
Django version 5.1.7, using settings 'graphical_auth.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
  
```

Screenshot 1 Run the web app from the command prompt



```

authentication > models.py > ...
1 # models.py
2 from django.db import models
3 from django.contrib.auth.models import User
4
5 class GraphicalPassword(models.Model):
6     user = models.OneToOneField(User, on_delete=models.CASCADE)
7     segment_sequence = models.CharField(max_length=255) # Store the sequence of segments selected
8
9 class Customer(models.Model):
10     user = models.ForeignKey(User, on_delete=models.CASCADE, null=True)
11     contact = models.CharField(max_length=100, null=True)
12     date = models.DateField(auto_now_add=True)
13
14 def __str__(self):
15     return self.contact
  
```

```

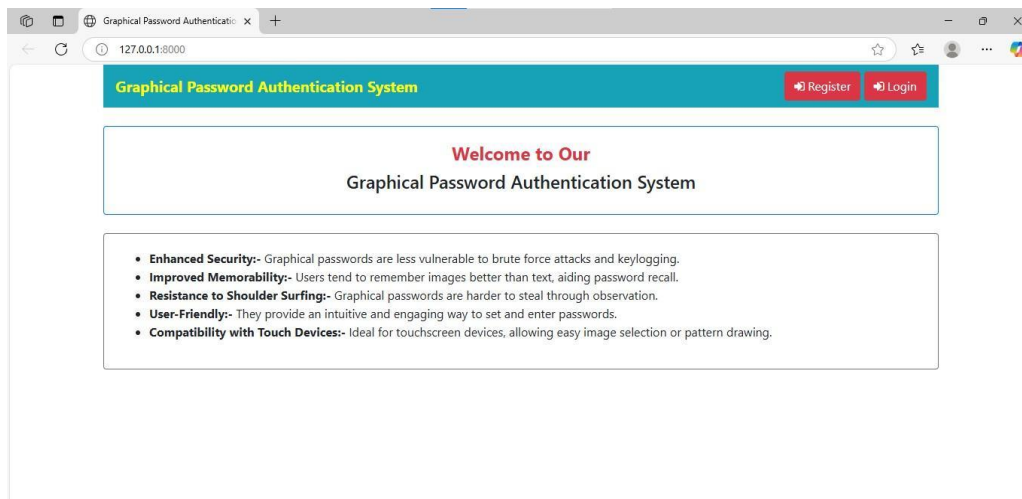
Microsoft Windows [Version 10.0.19045.5854]
(c) Microsoft Corporation. All rights reserved.

C:\final mini project\graphical_auth>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

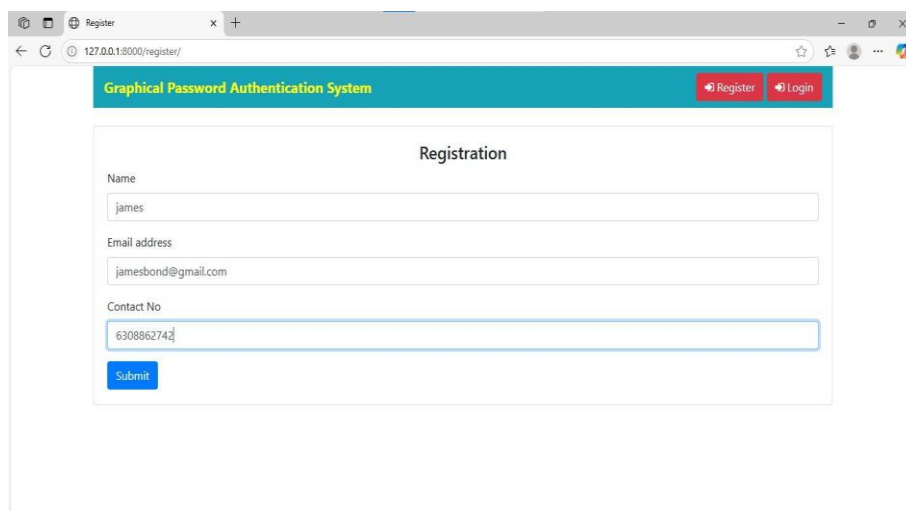
System check identified no issues (0 silenced).
May 18, 2025 - 18:53:08
Django version 5.1.7, using settings 'graphical_auth.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

[18/May/2025 18:56:35] "GET / HTTP/1.1" 200 3907
Not Found: /favicon.ico
[18/May/2025 18:56:35] "GET /favicon.ico HTTP/1.1" 404 3423
  
```

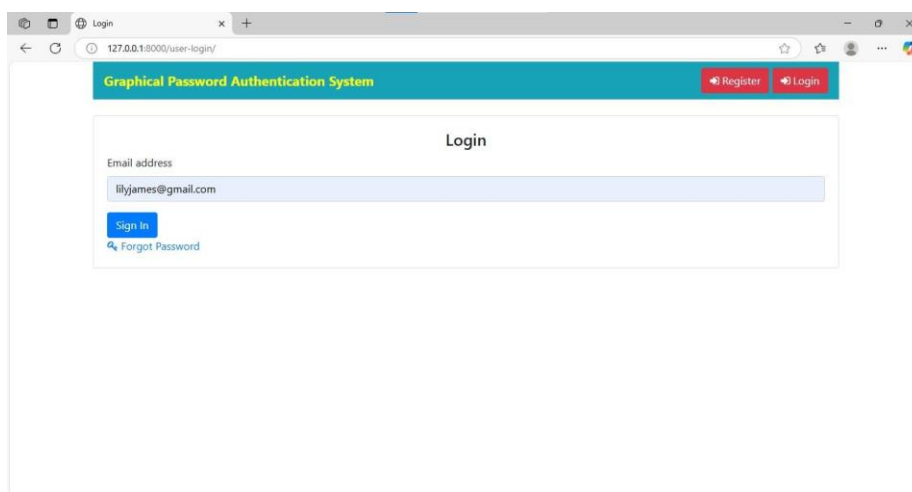
Screenshot 2 URL is generated



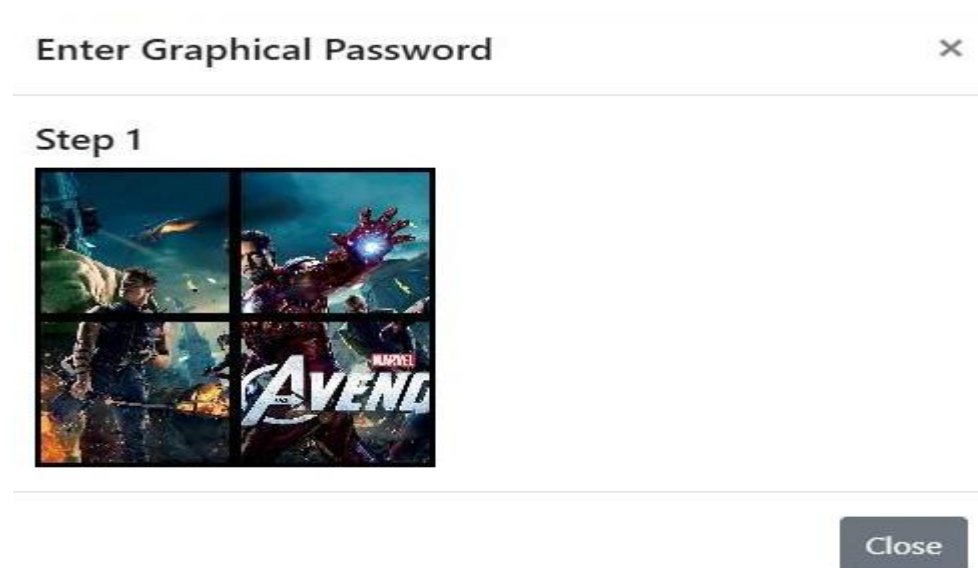
Screenshot 3 Web page



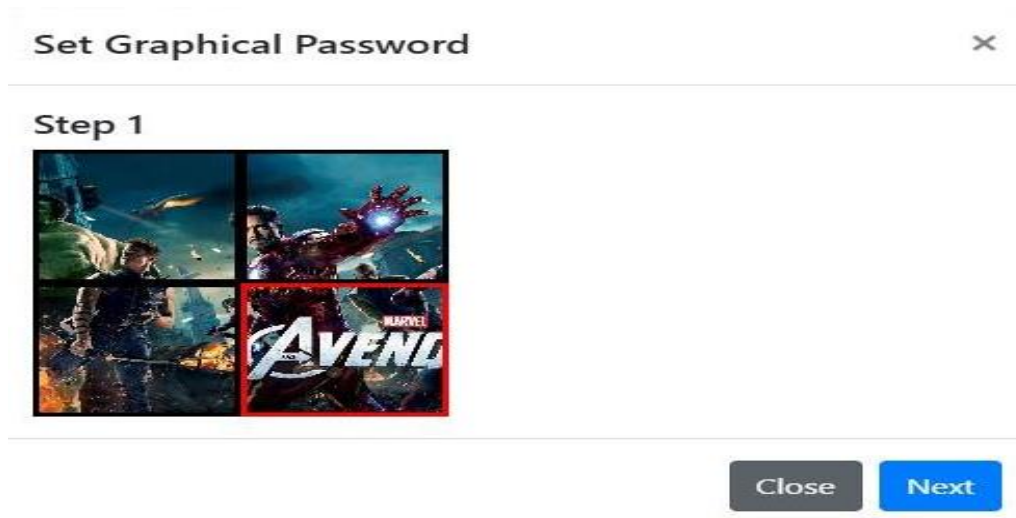
Screenshot 4 Registration page



Screenshot 5 Login page



Screenshot 6 Setting the password

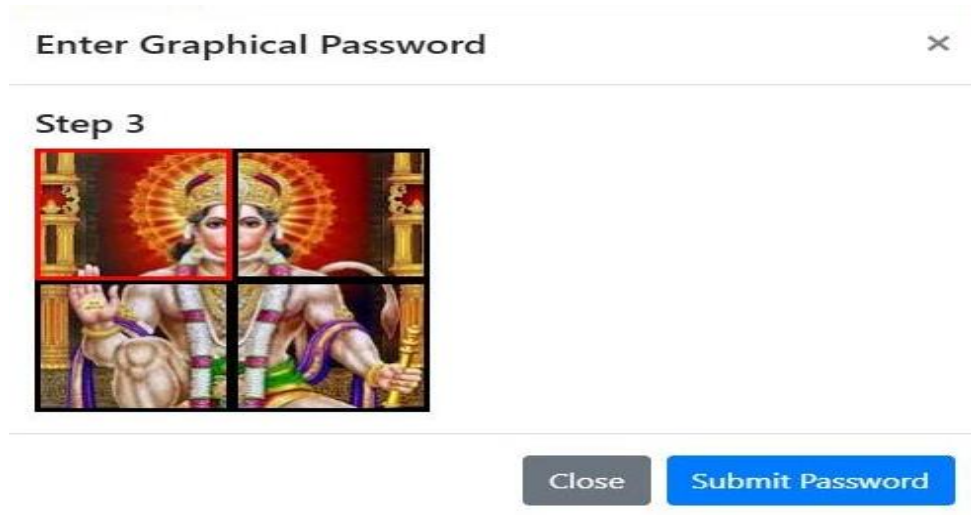


Screenshot 7 Selecting the POI's

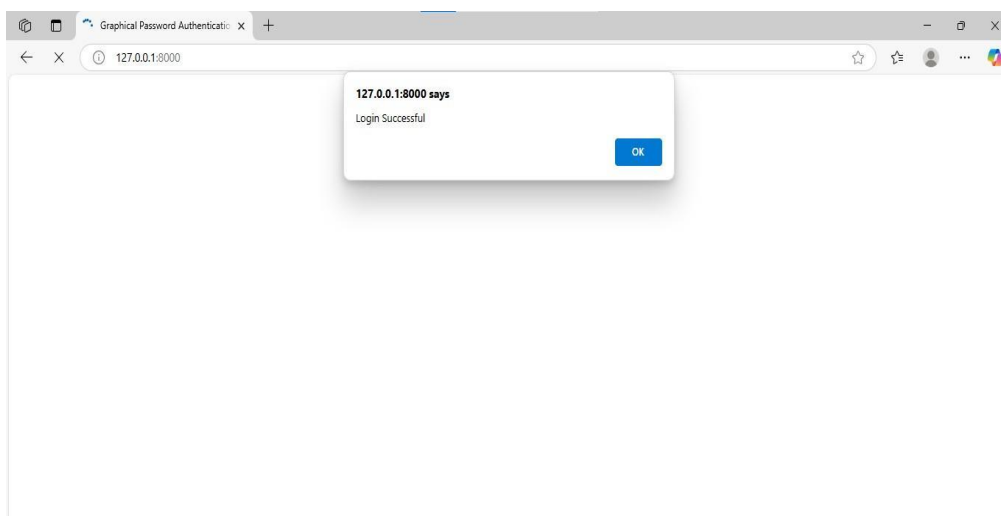




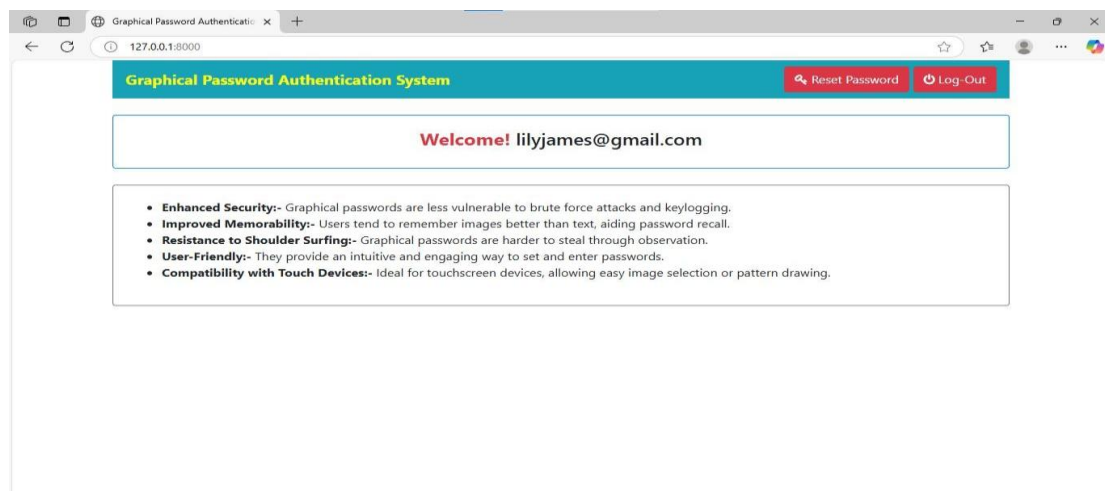
Screenshot 8 Selecting the POI's



Screenshot 9 Selecting the POI's



Screenshot 10 Login Successful



Screenshot 11 Welcome Page



## 5-CONCLUSION AND FUTURE SCOPE

### Conclusion

User authentication is a fundamental component in most computer security contexts. In this extended abstract, we proposed a simple graphical password authentication system. The system combines graphical and text-based passwords trying to achieve the best of both worlds. It also provides multi-factor authentication in a friendly intuitive system. We described the system operation with some examples, and highlighted important aspects of the system. In conclusion, A graphical password authentication system enhances security by utilizing visual memory, making it harder to predict or replicate compared to text-based passwords. It also provides a more engaging user experience, aligning well with modern web design trends and improving overall usability.

### Future Scope

This project involves developing a graphical password authentication system that combines graphical (image-based) and text-based passwords for user authentication. Users will select specific areas of an image and input text-based components to create a secure password. The system combines graphical password selection (such as choosing areas from an image) with traditional text-based passwords to improve both security and usability. This project aims to enhance security by combining graphical and text-based passwords for user authentication in web applications, making it harder for attackers to guess passwords. It also strives to improve user experience by offering an intuitive, interactive, and secure authentication process.

## REFERENCES

- [1] William Stallings and Lawrie Brown. Computer Security: Principles and Practices. Pearson Education, 2022.
- [2] Susan Wiedenbeck, Jim Waters, Jean-Camille Birget, Alex Brodskiy, and Nasir Memon. Passpoints: Design and Longitudinal Evaluation of a Graphical Password System. *International Journal of Human-Computer Studies*, 63:102–127, July 2023.
- [3] Robert Morris and Ken Thompson. Password Security: A Case History. *Communications of the ACM*, 22:594–597, November 2020.
- [4] Daniel V. Klein. Foiling the Cracker: A Survey of, and Improvements to, Password Security. In *Proceedings of the 2nd USENIX UNIX Security Workshop*, 2021.
- [5] Eugene H. Spafford. Observing Reusable Password Choices. In *Proceedings of the 3rd Security Symposium*. Usenix, pages 299–312, 2022.
- [6] Sigmund N. Porter. A Password Extension for Improved Human Factors. *Computers & Security*, 1(1):54–56, 2020.
- [7] Xiaoyuan Suo, Ying Zhu, and G. Scott Owen. Graphical Passwords: A Survey. In *Proceedings of Annual Computer Security Applications Conference*, pages 463–472, 2023.
- [8] Antonella De Angeli, Lynne Coventry, Graham Johnson, and Karen Renaud. Is a Picture Really Worth a Thousand Words? Exploring the Feasibility of Graphical Authentication Systems. *International Journal of Human-Computer Studies*, 63:128–152, July 2024.
- [9] Real User Corporation. The Science Behind Passfaces, June 2022.
- [10] G. E. Blonder. Graphical Password. U.S. Patent 5559961, Lucent Technologies, Inc. (Murray Hill, NJ), August 2021.