# A

**MINI PROJECT REPORT**

# ON

# WEB BASED GRAPHICAL

# PASSWORD AUTHENTICATION SYSTEM

Submitted in partial fulfillment of the requirement for the award of the degree of

## BACHELOR OF TECHNOLOGY

IN

## COMPUTER SCIENCE AND ENGINEERING

Submitted by

**A NIKHIL - 19R91A05D1**

**G SAI PRASAD - 19R91A05E4**

**S SAI KIRAN - 19R91A05J5**

Under the Guidance

Of

## Mrs. M SRIMATHI

## (Assistant Professor)

**Department of CSE**



## Department of Computer Science and Engineering TEEGALA KRISHNA REDDY ENGINEERING COLLEGE

**(An Autonomous Institution)**

**Medbowli, Meerpet, Saroornagar, Hyderabad – 500097**

**(Affiliated to JNTUH, Approved by AICTE, Accredited by NBA & NAAC ‘A’) (2019-2023)**

  **TEEGALA KRISHNA REDDY ENGINEERING COLLEGE**

**(Sponsored by TKR Educational Society)**

**Approved by AICTE, Affiliated by JNTUH, Accredited by NBA & NAAC-A)**

Medbowli, Meerpet, Saroornagar, Hyderabad – 500 097.

Phone: 040-24092838 Fax: +91-040-24092555

E-mail: [tkrec@rediffmail.com](mailto:tkrec@rediffmail.com) Website: [www.tkrec.ac.in](http://www.tkrec.ac.in)

**Department of Computer Science & Engineering**

# College code: R9

# CERTIFICATE

This is to certify that the Mini Project report on “**WEB BASED GRAPHICAL PASSWORD AUTHENTICATION SYSTEM**” is a bonafide work carried out by A Nikhil - 19R91A05D1, G Sai Prasad - 19R91A05E4, S Sai Kiran - 19R91A05J5 in partial fulfillment for the requirement of the award of B.Tech degree in Computer Science and Engineering, Teegala Krishna Reddy Engineering College, Hyderabad, affiliated to Jawaharlal Nehru Technological University, Hyderabad under my guidance and supervision.

The result of investigation enclosed in this report have been verified and found satisfactory. The results embodied in the project work have not been submitted to any other University for the award of any degree.

## INTERNAL GUIDE HEAD OF DEPARTMENT

Mrs. M Srimathi Dr.CH.V.Phani Krishna

(Asst Professor) Professor

………………………………. ….…………………………

## EXTERNAL EXAMINER PRINCIPAL

Dr. K. Venkata Murali Mohan

Professor

………………..……… ………………………

  **TEEGALA KRISHNA REDDY ENGINEERING COLLEGE**

**(Sponsored by TKR Educational Society)**

**Approved by AICTE, Affiliated by JNTUH, Accredited by NBA & NAAC-A)**

Medbowli, Meerpet, Saroornagar, Hyderabad – 500 097.

Phone: 040-24092838 Fax: +91-040-24092555

E-mail: [tkrec@rediffmail.com](mailto:tkrec@rediffmail.com) Website: [www.tkrec.ac.in](http://www.tkrec.ac.in)

**Department of Computer Science & Engineering**

# College code: R9

# DECLARATION

We hereby declare that the Mini Project report entitled **“Web Based Graphical Password Authentication System”** is done under the guidance of **Mrs. M Srimathi**. **,** Department of Computer Science and Engineering, Teegala Krishna Reddy Engineering College, is submitted in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** from **Jawaharlal Nehru Technological University**,Hyderabad.

This is a record of bonafide work carried out by us in **Teegala Krishna Reddy Engineering College** and the results embodied in this project have not been reproduced or copied from any source.

**Submitted b****y**

**A Nikhil (19R91A05D1)**

**G Sai Prasad (19R91A05E4)**

**S Sai Kiran (19R91A05J5)**

# ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance have crowned our efforts with success.

We extend my deep sense of gratitude to **Dr.K.Venkata Murali Mohan**, **Principal** Teegala Krishna Reddy Engineering College, Meerpet, for permitting me to undertake this project.

We are indebted to **Dr.CH.V.Phani Krishna, Professor & Head of the Department**, Computer Science and Engineering, Teegala Krishna Reddy Engineering College, Meerpet for his support and guidance throughout our project.

We are indebted to our guide **M Srimathi**, Computer Science Engineering, Teegala Krishna Reddy Engineering College, Meerpet for his support and guidance throughout our project.

We are indebted to the project coordinators **A.DIVYA SREE,** Assistant professor, and **Y.SHIVA SREE**, Assistant Professor, Computer Science and Engineering, Teegala Krishna Reddy Engineering College, Meerpet for his support and guidance throughout our project.

Finally, we express thanks to one and all that have helped me in successfully completing this technical seminar. Further I would like to thank my family and friends for their moral support and encouragement.

**By**

**A Nikhil (19R91A05D1)**

**G Sai Prasad (19R91A05E4)**

**S Sai Kiran (19R91A05J5)**

**ABSTRACT**

User authentication is a fundamental component in most computer security contexts. It provides the basis for access control and user accountability. While there are various types of user authentication systems, alphanumerical username/passwords are the most common type of user authentication. They are versatile and easy to implement and use . However, It has its disadvantages like easy or short passwords are easy target of dictionary and brute-forced attacks and Difficult passwords are hard to remember. Hence we propose to use graphical passwords, in which graphics (2D images) are used instead of alphanumerical passwords .This can be achieved by asking the user to select regions from an image rather than typing characters as in alphanumeric password approaches. The operation of the purposed scheme is simple and easy to learn for user since they familiar with textual graphical password scheme. In conclusion, this graphical password scheme will make it easier for user to do their authentication process since it is easy to remember and hard to guess by others.

**LIST OF FIGURES**

Figure 2.1 A Sample of PassPoint Method .............................................................................. 5

Figure 2.2 Graphical Password Scheme Suggested by Blonder .............................................. 5

Figure 2.3 Passfaces™ Method……………………………………………………………… 7

Figure 2.4 Déjà vu Method…………………………………………………………………… 8

Figure 4.1 System Architecture….……………………………………………………………11

Figure 4.2 Flowchart ………………………………………………………………………….12

Figure 4.3.1 Use Case Diagram……………………………………………………………….14

Figure 4.3.2 Class Diagram…………………………………………………………………...15

Figure 4.3.3 Activity Diagram ……………….……………………………………………….24

Figure 4.3.4.1 Sequence Diagram for Registration Phase…………………………………….27

Figure 4.3.4.2 Sequence Diagram for Login Phase ………………………………………….28

Figure 5.3 Home Interface for Graphical Password Authentication System………………..

Figure 5.3.1 Registration Page….. …………………………………………………………….28

Figure 5.3.2 Login Interface …………………………………………………………………..33

Figure 4.11 Login Password ………………………………………………………………….33

Figure 4.12 First Click in Login Phase ……………………………………………………….34

Figure 4.13 Second Click in Login Phase …………………………………………………….34

Figure 4.14 Third Click in Login Phase ………………………………………………………35

Figure 4.15 Fourth Click in Login Phase ……………………………………………………..35

Figure 4.16 Fifth Click in Login Phase ………………………………………………………..36

Figure 4.17 Success Login Interface ………………………………………………………….36

Figure 4.18 Failed Login Interface ……………………………………………………………36

**LIST OF TABLES**

**CONTENTS**

TITLE PAGE NO

|  |
| --- |
| **CHAPTER 1 INTRODUCTION** |
| 1.1 Background 1 |
| 1.2 Problem Statement 1 |
| 1.3 Objectives 2 |
| 1.4 Project Scope 2 |
| 1.5 Existing 3 |
| 1.6 Proposed 3 |
| **CHAPTER 2 LITERATURE SURVEY** |
| 2.1 Introduction 4 |
| 2.2. PassPoint Method 4 |
| 2.3 Blonder Graphical Password Scheme 5 |
| 2.4 Security in Graphical Password Authentication 5 |
| 2.5 Multiple-image schemes 6 |
| 2.5.1 Passfaces Method 6 |
| 2.6 Déjà Vu 7 |
| 2.7 Summary 7 |
| **CHAPTER 3 SYSTEM ANALYSIS** |
| 3.1 Hardware Requirements |
| 3.2 Software Requirements |
| **CHAPTER 4 SYSTEM DESIGN** |
| 4.1 System Architecture 17 |
| 4.2 Flow Chart 17 |
| 4.3 UML Diagrams 18 |
| 4.3.1 Use Case Diagram 21 |
| 4.3.2 Class Diagram 23 |
| 4.3.3 Activity Diagram |
| 4.3.4 Sequence Diagram |
| 4.3.4.1 Sequence Diagram for Registration Phase |
| 4.3.4.1 Sequence Diagram for Login Phase |
| **CHAPTER 5 IMPLEMENTATION**  26 |
| 5.1 Introduction 26 |
| 5.2 Environmental Setup 26 |
| 5.3 User Interface of the System |
| 5.3.1 Registration Phase 27 |
| 5.3.2 Login Phase 32 |
| 5.4 Summary |
| **CHAPTER 6 SYSTEM TEST** |
| 6.1 Unit Testing 38 |
| 6.2 Integration Testing 38 |
| 6.3 Functional Testing 39 |
| 6.4 System Testing |
| 6.5 Whitebox Testing |
| 6.6 Blackbox Testing |
| 6.7 Acceptance Testing |
| **CHAPTER 7 CODING** |
| **CHAPTER 8 OUTPUT SCREENS** |
| **CHAPTER 9 CONCLUSION** 41 |
| 9.1 Introduction 41 |
| 9.2 System Contribution 41 |
| 9.3 Future Work 41 |
| 9.4 Summary |
| **CHAPTER 10 FUTURE ENHANCEMENT** |
| **CHAPTER 11 REFERENCES** 43 |

**CHAPTER 1**

**INTRODUCTION**

**1.1 Background**

Authentication is the process of determining that the person requesting a resource

is the one who it claims to be. Most of the authentication system nowadays uses an

integration of username and password. The problem with the password is that it

requires user to remember it and it should be kept secret. Each authentication system

has its own guidelines and limitations like password length, password must contain

alphanumeric and special characters. These passwords are mostly text-based passwords.

Either user use passwords that are easy to remember like license plate number, parent

name, phone number sometimes their own name which are very much predictable or

complex passwords which they overlook so they might be use the same password for

different accounts or they jot down their password somewhere. Moreover, user is

vulnerable to various attacks. Text-based passwords faces from security and usability

matters.

To overcome these shortcomings of alphanumeric passwords, graphical password

schemes have been proposed. In graphical password authentication application by using

passpoints scheme a password contains an image where user can input password with

the help of mouse events like click and drag. Picture Superiority Effect Theory reveals

that pictures can be recognized and recalled easily by human brain, enhancing the ability

to. Strong passwords can be invented which are resistant to guessing, dictionary

attack and social engineering.

**1.2 Problem Statement**

The problem statement that can be describe in this project are user have problem

to remember their complicated password over time due to the limitation of human

brain, user tend to forget about their password. Next, user tent to use the same

password for all type of account. So, if one account is hacked, the possibility for other

account to be hack is high. Therefore, choosing the simple textual passwords may

increase its vulnerability for attacks or intrusions.

**1.3 Objectives**

The first objective of the research is

1. To design a Graphical Password Authentication implemented in mobile application.
2. To implement the Graphical Password Authentication application using PassPoint technique.
3. To test the effectiveness of Graphical Password Authentication system using PassPoint technique to authenticate user by using web-based system.

**1.4 Project Scope**

The scope for this project is identified which to make the web system process easier.

This project concentrates more on the security of the system.

i) Scope of User

- Enter username, password, email during registration and login phase.

- Select an image during registration phase and login phase.

- Click five points during registration phase and login phase.

ii) Scope of System

- Sign up – the authentication system let the user select picture and click points

in a correct number of clicks.

- Log in – check either the user username, password, image and clicked points

are valid and exist in the data store.

**1.5 EXISTING SYSTEM:**

Graphical passwords refer to using pictures (also drawings) as passwords. In theory, graphical passwords are easier to remember, since humans remember pictures better than words. Also, they should be more resistant to brute force attacks, since the search space is practically infinite.In general, graphical passwords techniques are classified into two main categories: recognition-based and recall based graphical techniques. In recognition-based techniques, a user is authenticated by challenging him/her to identify one or more images he or she chooses during the registration stage. In recall-based techniques, a user is asked to reproduce something that he or she created or selected earlier during the registration stage.

Pass faces is a recognition-based technique, where a user is authenticated by challenging him/her into recognizing human faces. An early recall-based graphical password approach was introduced by Greg Blonder in 1996. In this approach, a user create a password by clicking on several locations on an image. During authentication, the user must click on those locations. Pass Points builds on Blonders idea, and overcomes some of the limitations of his scheme. Several other approaches have been surveyed in the following paper.

**1.6 PROPOSED SYSTEM:**

The proposed authentication system works as follows. At the time of registration, a user creates a graphical password by first entering a picture he or she chooses. The user then chooses several point-of-interest (POI) regions in the picture. Each POI is described by a circle (center and radius). For every POI, the user types a word or phrase that would be associated with that POI. The user can choose either to enforce the order of selecting POIs (stronger password), or to make the order insignificant.

For authentication, the user first enters his or her username. The system, then, displays the registered picture. The user, then, has to correctly pick the POIs and type the associated words. At any time, typed words are either shown as asterisks (\*) or hidden.

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 Introduction**

This chapter discussed about the related research that is review for Graphical Password Authentication which are being proposed. Generally, this including a few article and journal that related directly and indirectly to the secure graphical password system. All this research was described, summarized, evaluated and clarified. It is a regulation in order to establish the credibility for a better project.

**2.2. PassPoint Method**

In this paper it is an extended Blonder’s idea by eliminating the predefined boundaries and allowing arbitrary images to be used. The image could be any natural picture or painting then it contains several possible clicks points. As a result, a user can click on any place on an image (as opposed to some pre- dined areas) to create a password. A tolerance around each chosen pixel is calculated. In order to be authenticated, the user must click within the tolerance of their chosen pixels and also in the correct sequence as in Figure 2.1. When using this method user might easily able to quickly create a valid password.

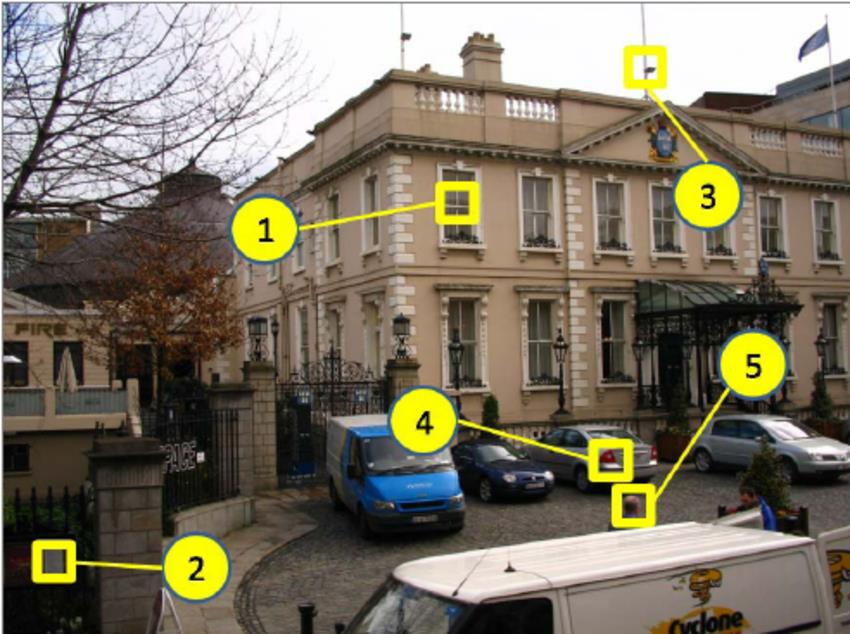


Figure 2.1 A Sample of PassPoint Method

**2.3 Blonder Graphical Password Scheme**

Single-image based schemes use one single image as a background, and require a user to repeat several actions with an input device, such as clicking or dragging in the same manner as in the registration stage.

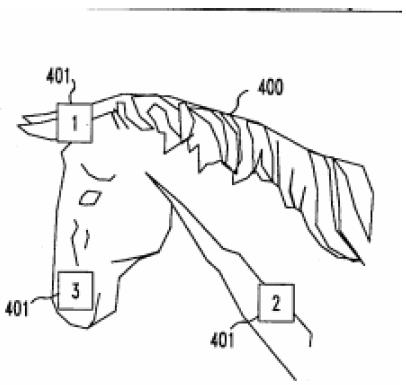


Figure 2.2 Graphical password scheme suggested by Blonder

Blonder gave the initial idea of graphical password. In his scheme, a user is presented with one predetermined image on a visual display and required to select one or more predetermined positions on the displayed image in a particular order to access the restricted resource. The major drawback of this scheme is that user cannot click arbitrarily on the background. The memorable password space was not studied by the author either.

**2.4 Security in Graphical Password Authentication**

According to the paper, the first defence for computer system is authentication. Graphical authentication may offer greater resistance to guessing and capture attacks but there are other attacks against graphical authentication including social engineering, brute force attacks, shoulder surfing, intercepted communication and spyware which those attacks might be threats to the security breach. Authentication mechanism that often being used is the combination of usernames and passwords which is based on textual-based password. Nevertheless, this traditional approach had shown some disadvantages. The significant consequences of the approach are the user might choose simple password for authentication process or the user can create a strong password however it is hard to be remembered by the user itself. This paper mentioned about three categories of the graphical authentication scheme which are Drawmetric schemes, Searchmetric schemes and Locimetric system. There are also CAPTCHA, but it is not based on recognition or re-creation password like the other graphical password but it is relied on human (as opposed to computer) abilities to recognize obfuscated text displayed in form of image. There is also hybrid scheme which is made up of combination of two or more schemes.By using graphical password scheme, it can provide highly secure authentication process by enable the user to remember the complex password easily. And it also can be used as defence to the shoulder surfing, Spybot and similar compromises of user systems. The highly secure authentication system can be achieved by adding some security features in graphical user authentication.

**2.5 Multiple-image schemes**

In multiple-image schemes, on the other hand, multiple images are presented and a user is required to recognize and identify one or more of it, which are previously seen and selected by the user. Psychological studies suggest that people are much better at imprecise recall, particularly in recognition of previously experienced stimuli. This class of passwords was shown to be remembered by user for a long period after short perception.

**2.5.1 Passfaces Method**

Passfaces is a commercial product by Passfaces Corporation , requires a user to select previously seen human face pictures as a password, as shown in Figure 2.3. One problem with Passfaces is that some faces displayed might not be welcomed by certain user. In other words, if a user has to look at some faces, he/she does not like or even dislike., the login process will become unpleasant. Another drawback of Passfaces is that it cannot be used by people who are face-blind (a disease which affects a person’s ability to tell faces apart).



Figure 2.3 Passfaces Method

**2.6 Déjà Vu**

Graphical authentication mechanism based on hash visualization technique is proposed by [5]. In the proposed scheme, the user will have to choose a few pictures from a group of random pictures generated by a program. Then, the user has to identify the selected images during registration before in order to be authenticated. In this research paper, by using graphical authentication the outcome shows that 90% of all the participants success the authentication session while only 70% succeeded using textbased password and PINS. However, this proposed technique uses more time than the traditional approach in terms of average log in time. They also mention the disadvantages of this technique is there are needs to store the details of the images of each user in plaintext in the server. Considering the fact that the password space of textual passwords is much larger than that of Déjà vu is easier to remember.

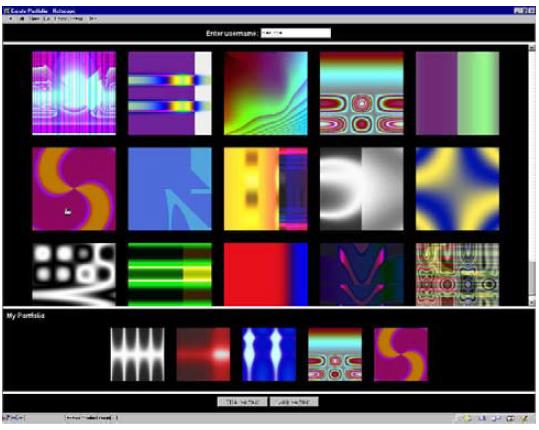


Figure 2.4: Déjà vu Method

**2.7 Summary**

This chapter is discussed about the literature review that use a reference for

development process of this proposed system. The analysis is done to find suitable

technique and method for this system.

**CHAPTER 3**

**SYSTEM ANALYSIS**

**3.1 HARD REQUIRMENTS :**

* System    :   Pentium IV 2.4 GHz.
* Hard Disk  :   40 GB.
* Floppy Drive :   1.44 Mb.
* Monitor   :   15 VGA Colour.
* Mouse    :   Logitech.
* Ram    :   512 MB.

**3.2 SOFTWARE REQUIRMENTS :**

* Operating system   : Windows 8Professional.
* Coding Language  : python

**CHAPTER 4**

**SYSTEM DESIGN**

## 4.1 SYSTEM ARCHITECTURE:

## System Architecture is a sketch of following process that allows how the system works and happen. Figure 4.1 shows that user can register to the system by enter username, email and phone number and then user is required to select a picture displayed. At this point, user need to click any five points in the picture that had been chosen before. After that, registration information will be saves in database. During login phase, user need to insert the username that has been registered during registration phase. Then, user is required to verify the picture displayed in the application that they had choose during 18 registration phase. After that, user is required to click five points that they clicked during the registration phase respectively. The system will make a comparison by checking the information with database. The database server will send result whether user have registered or not to the user. Finally, user will be authenticated if the information entered and given by user are all correct.

## 

## Figure 4.1 System Architecture

## 4.2 FLOW CHART

## 

## 4.3 UML DIAGRAMS:

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: A Metamodel and a notation. In the future, some form of method or process may also be added to; or associated with, UML. The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing object oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

**4.3.1 USE CASE DIAGRAM:**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

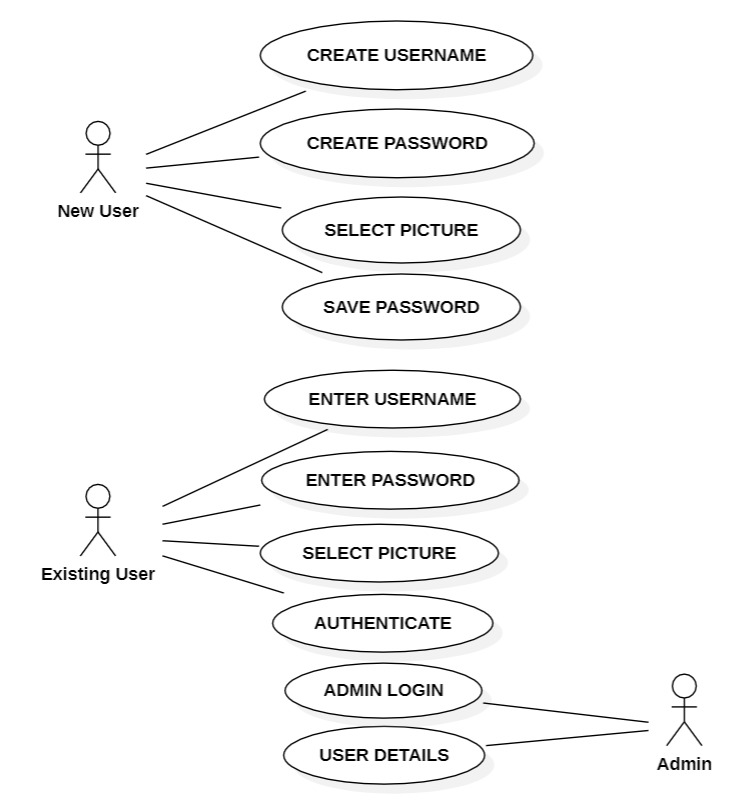


Fig 4.3.1 USECASE DIAGRAM

# 4.3.2 CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

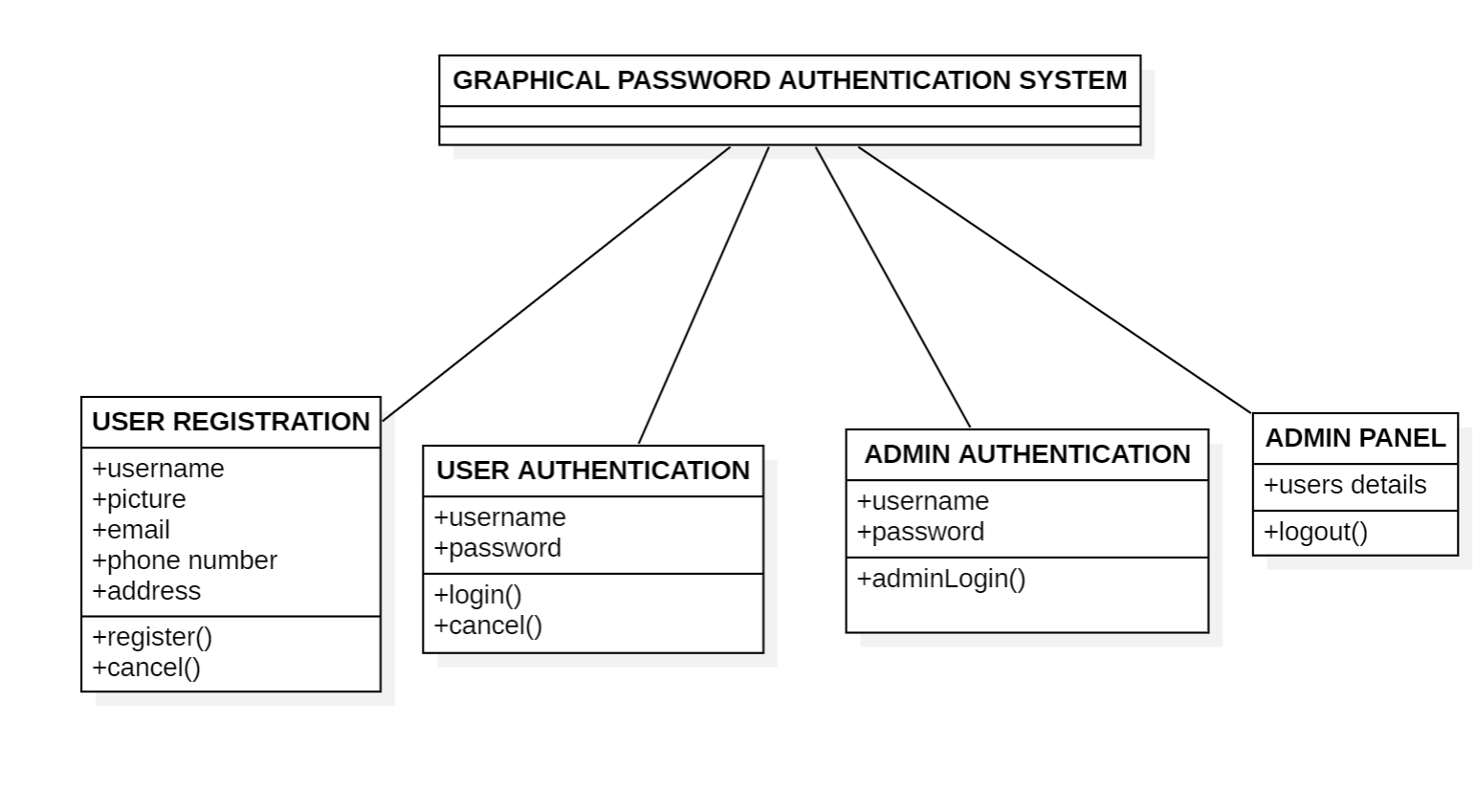


Fig 4.3.2 CLASS DIAGRAM

**4.3.3 ACTIVITY DIAGRAM**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-bystep workflows of components in a system. An activity diagram shows the overall flow of control.

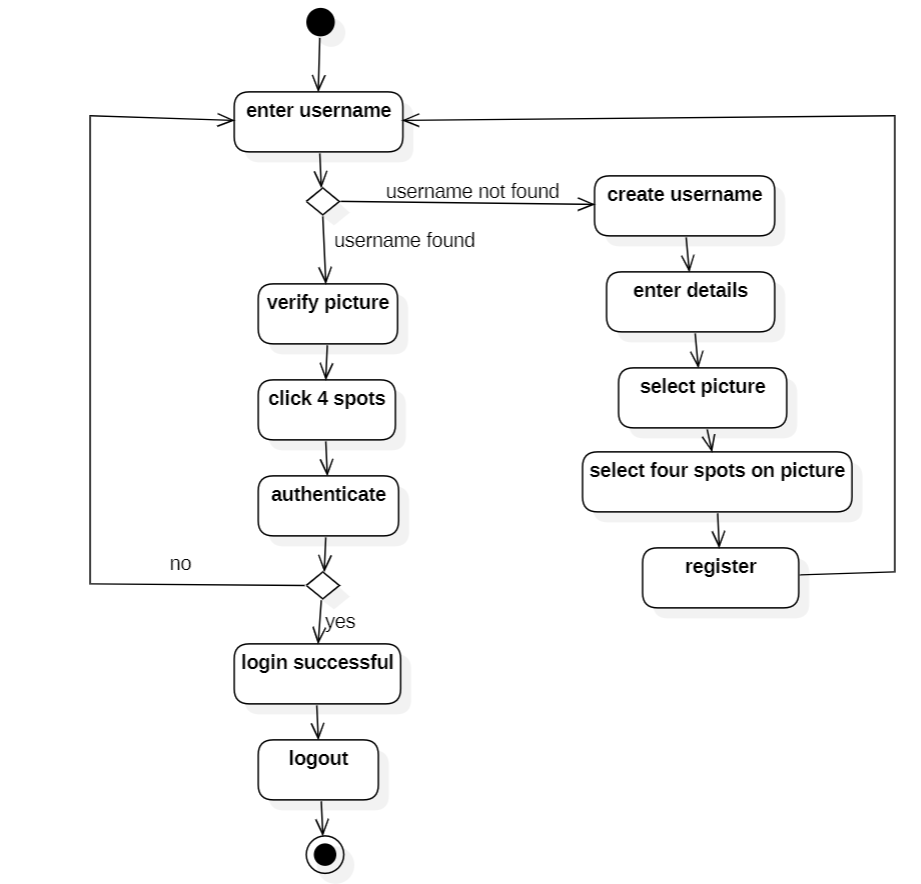
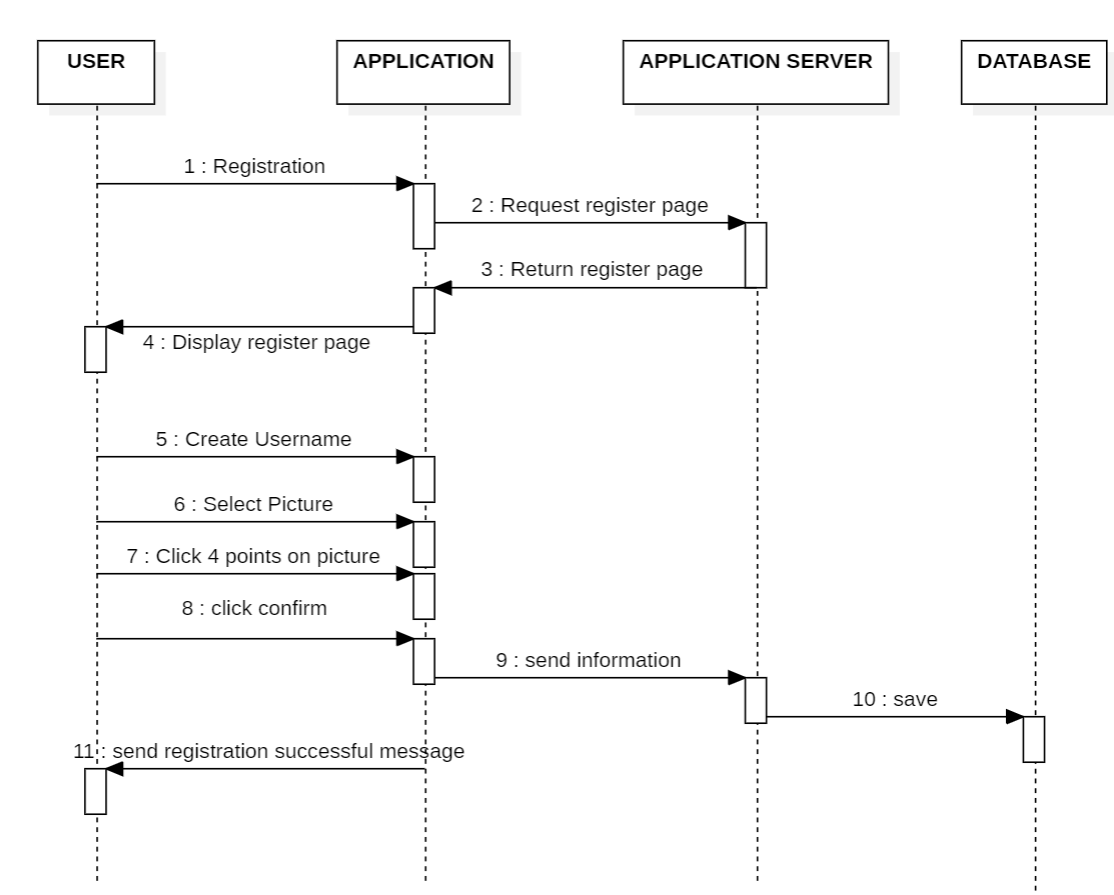
****

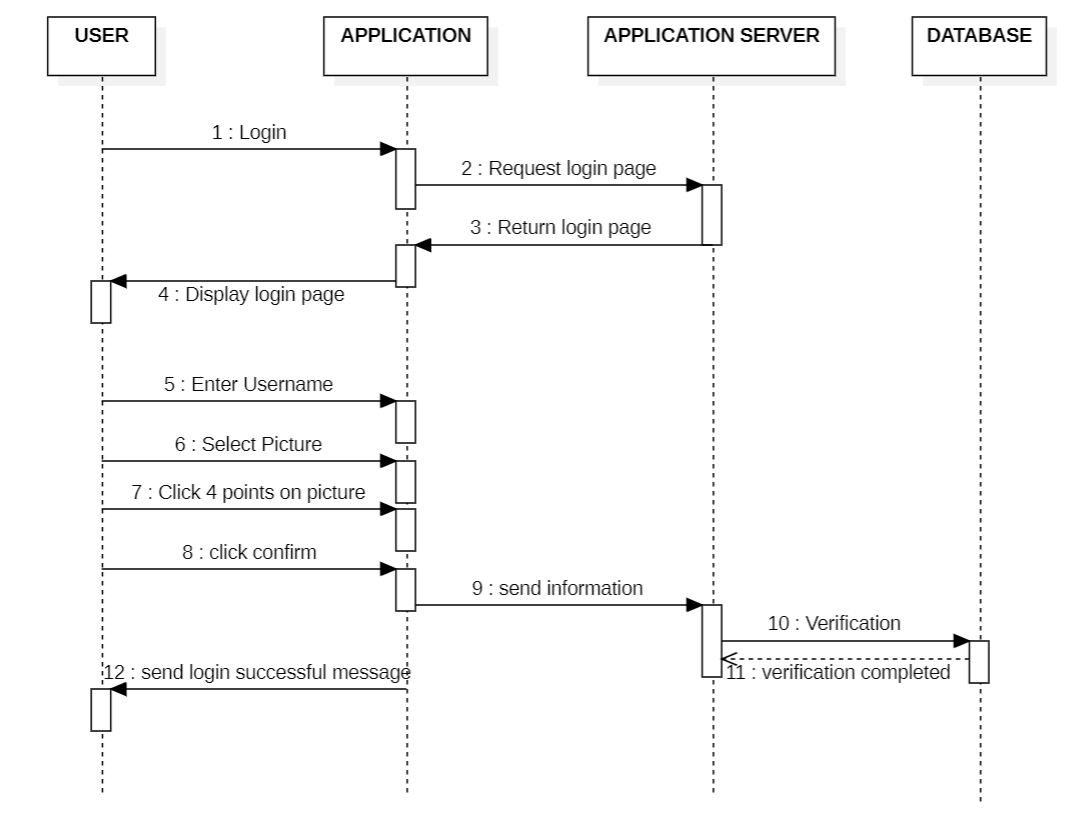
Fig 4.3.3 ACTIVITY DIAGRAM

**4.3.4 SEQUENCE DIAGRAM:**

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



4.3.4.1 SEQUENCE DIAGRAM FOR REGISTRATION PHASE



4.3.4.2 SEQUENCE DIAGRAM FOR LOGIN PHASE

**CHAPTER 5**

**IMPLEMENTATION**

**5.1 Introduction**

This chapter discuss about the implementation and testing of graphical password authentication in web system. The implementation is the writing of code line and run the code in localhost. Meanwhile, testing phase are being used to find the bug in the system by the test with dummy input data.

**5.2 Environmental Setup**

Python a versatile programming language doesn’t come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace. The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

## How to Install Python on Windows and Mac :

Before you start with the installation process of Python. First, you need to know about your **System Requirements**. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a **Windows 64-bit operating system**. So the steps below are to install python version 3.7.4 on Windows 10 device or to install Python 3.

### Download the Correct version into the system

**Step 1:** Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: [https://www.python.org](https://www.python.org/)



Now, check for the latest and the correct version for your operating system.

**Step 2:** Click on the Download Tab.

****

**Step 3:** You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4

****

**Step 4:** Scroll down the page until you find the Files option.

**Step 5:** Here you see a different version of python along with the operating system.



* To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.
* To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer. Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

**Note:** To know the changes or updates that are made in the version you can click on the Release Note Option.

### Installation of Python

**Step 1:** Go to Download and Open the downloaded python version to carry out the installation process.



**Step 2:** Before you click on Install Now, Make sure to put a tick on Add Python 3.7 to PATH.



**Step 3:** Click on Install NOW After the installation is successful. Click on Close.



With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

**Note:** The installation process might take a couple of minutes.

### Verify the Python Installation

**Step 1:** Click on Start

**Step 2:** In the Windows Run Command, type “cmd”.

**Step 3:** Open the Command prompt option.

**Step 4:** Let us test whether the python is correctly installed. Type **python –V** and press Enter.



**Step 5:** You will get the answer as 3.7.4

**Note:** If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

**5.3 User Interface of the System**

There are two types of user interface (UI) which are command line and

graphical use interface (GUI). In this research, graphical user interface (GUI) is

implemented which means user can interact with system or software through graphical

image.

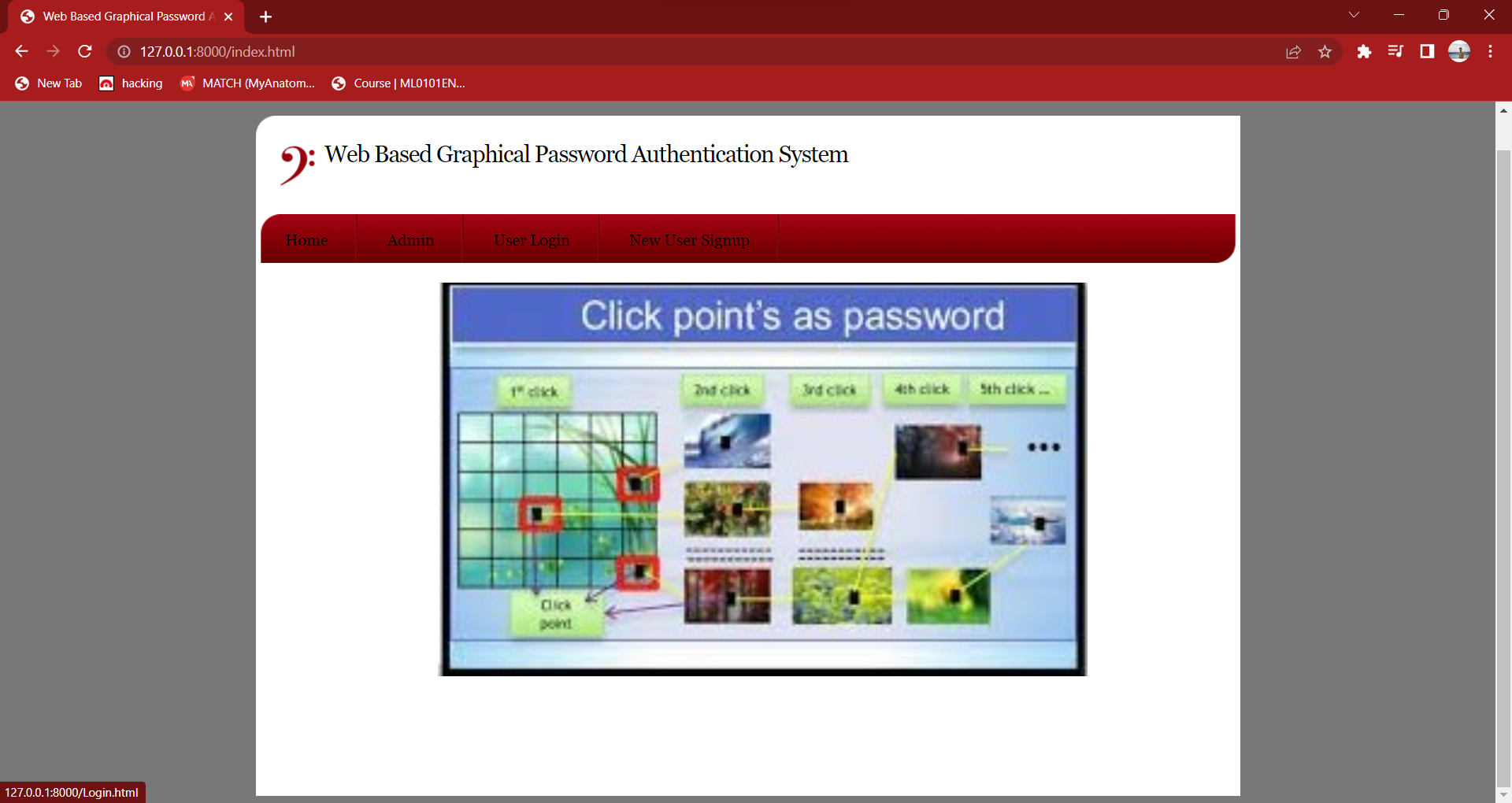
****

Figure 5.3: Home Interface for Graphical Password Authentication (GPA) System

**5.3.1 Registration Phase**

Figure 5.3 shows the home interface of Graphical Password Authentication System that contains ‘Login’ and ‘Register’ link. When a user clicks on ‘Register’ link, it will redirect the user to register page which will show in Figure 5.3.1. In this page, a user will be asked to fill the details such as username, email and phone number. After user had fill in the details, he/she need to click on ‘Register’ button that will bring to the next page which is registration pass. Next, user will be requested to choose a picture that represent their password out of all pictures from database. Then, user have to choose four clicks in the picture that had been chose which have no secret in every click than to make user remember every click as it is their password. Each of these click point will be shown as coordinate\_1, coordinate\_2, coordinate\_3 and coordinate\_4 respectively. Lastly, user will be directed to registration success page which means the user are successfully registered to the system.

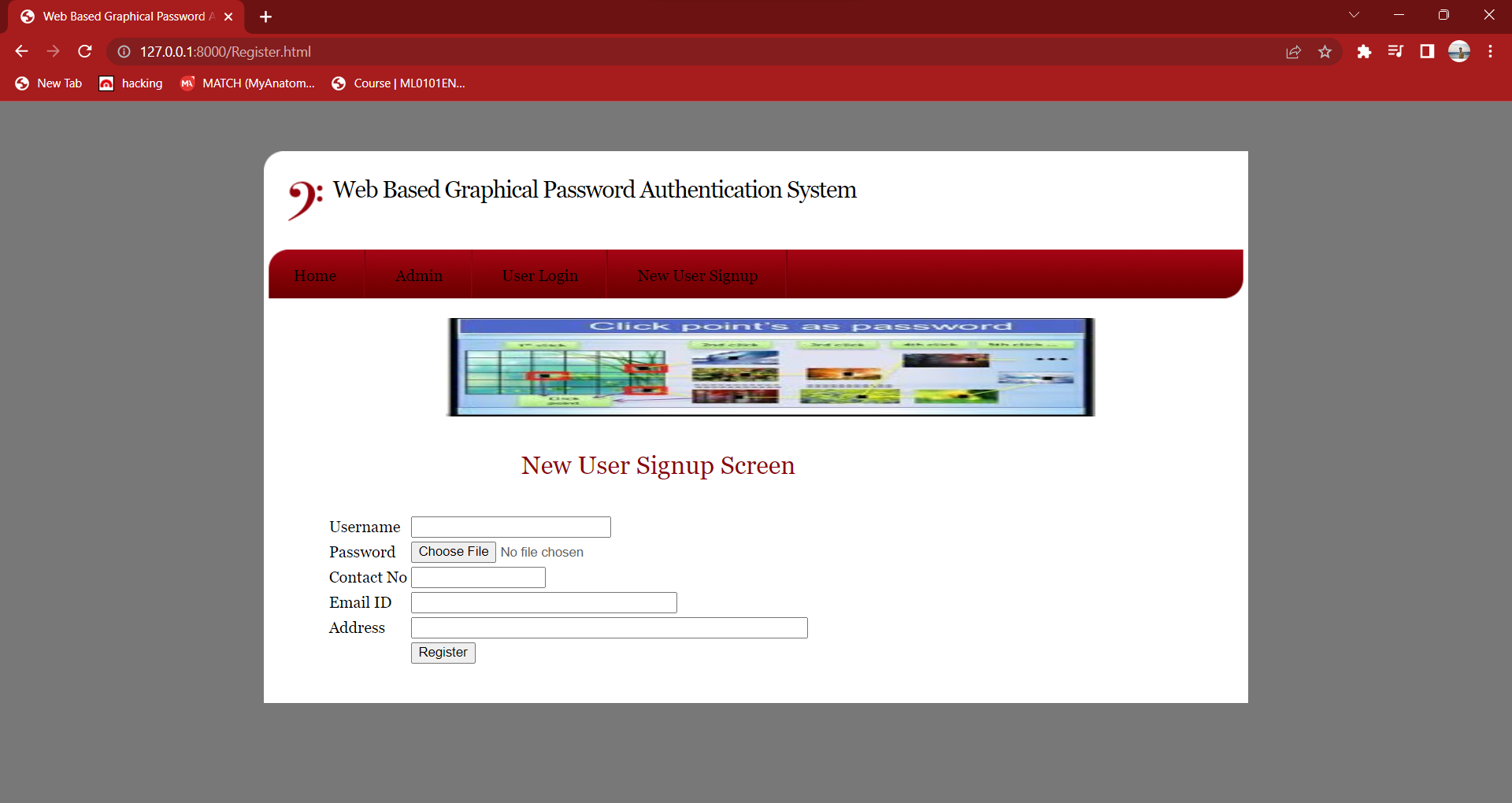


Figure 5.3.1 Registration Page

**5.3.2 Login Phase**

In login phase, user need to sign in a valid username that had registered before in registration phase. After user enter the username, he/she will be link to choose password page that will be shown. Same with the username,Then, the user needs to click on four points . Finally, the user will be authenticated by the system and successful login interface will be displayed. As a reminder, the selection of click points must be click in the same order with registration phase before. If the order of click points are wrong, the system will not authenticate current user to enter the system and the interface will display that the user is failed to login.

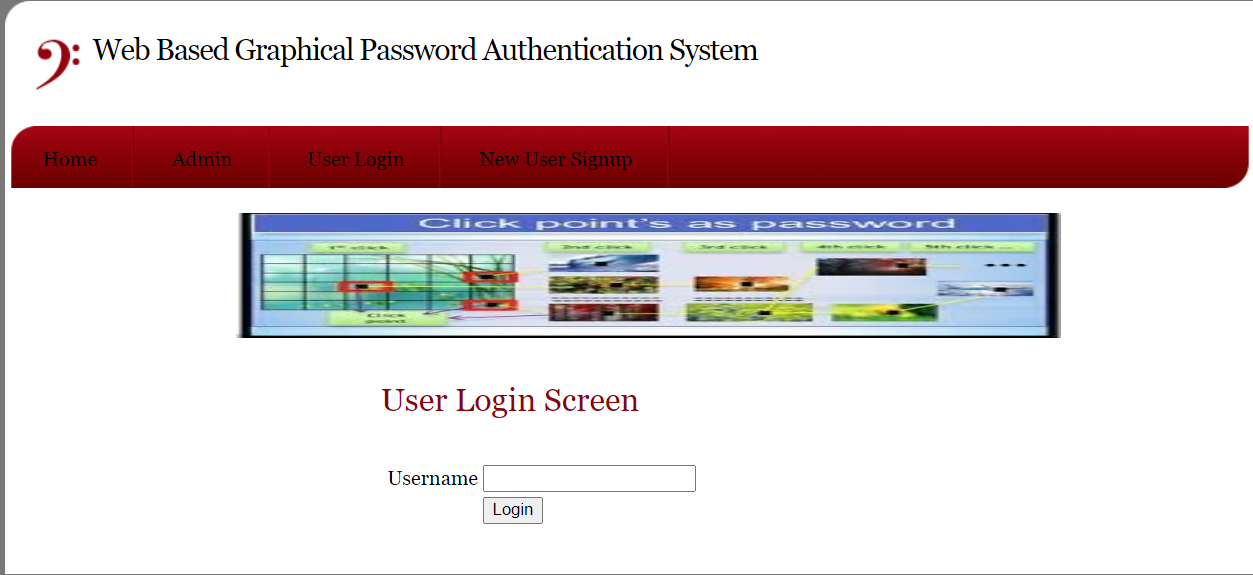


Figure 5.3.2 Login Interface

**5.4 Summary**

This shows the implementation of graphical password authentication system by using passpoints scheme. The users are given a guide step-by-step on how to register and login the system.

**CHAPTER 6**

**SYSTEM TEST**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

### **TYPES OF TESTS**

**6.1 Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**6.2 Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

**6.3 Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**6.4 System Test**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**6.5 White Box Testing**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**6.6 Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**Unit Testing**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

**Test strategy and approach**

Field testing will be performed manually and functional tests will be written in detail.

**Test objectives**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

**Features to be tested**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

# Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**6.7 Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**CHAPTER 7**

**CODING**

from django.shortcuts import render

from django.template import RequestContext

from django.contrib import messages

from django.http import HttpResponse

from django.conf import settings

import os

import pymysql

from django.core.files.storage import FileSystemStorage

global username, password, contact, email, address

def UserLogin(request):

global username

if request.method == 'POST':

username = request.POST.get('t1', False)

password = "none"

con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'GraphPassword',charset='utf8')

with con:

cur = con.cursor()

cur.execute("select username,password FROM register")

rows = cur.fetchall()

for row in rows:

if row[0] == username:

password = row[1]

break

if password != 'none':

output = '<center><img src="/static/password/'+password+'" alt="" width="400" height="300" id="myImgId" onmousemove="getPos(event)"/></center>'

context= {'data':output}

return render(request, 'ShowAuthenticateImage.html', context)

if password == 'none':

context= {'data':'Invalid username'}

return render(request, 'Login.html', context)

def Register(request):

if request.method == 'GET':

return render(request, 'Register.html', {})

def Reset(request):

if request.method == 'GET':

return render(request, 'Reset.html', {})

def index(request):

if request.method == 'GET':

return render(request, 'index.html', {})

def Login(request):

if request.method == 'GET':

return render(request, 'Login.html', {})

def AdminLogin(request):

if request.method == 'GET':

return render(request, 'AdminLogin.html', {})

def getSpotValue(spot):

arr = spot.split(",")

values = []

values.append(int(arr[0].strip()))

values.append(int(arr[1].strip()))

return values

def authspots(old\_spot, new\_spot):

auth = False

old\_x = old\_spot[0]

old\_y = old\_spot[1]

previous\_x = old\_x - 10

forward\_x = old\_x + 10

previous\_y = old\_y - 10

forward\_y = old\_y + 10

if new\_spot[0] >= previous\_x and new\_spot[0] <= forward\_x and new\_spot[1] >= previous\_y and new\_spot[1] <= forward\_y:

auth = True

return auth

def PasswordAuthAction(request):

if request.method == 'POST':

global username

spot1 = getSpotValue(request.POST.get('t1', False))

spot2 = getSpotValue(request.POST.get('t2', False))

spot3 = getSpotValue(request.POST.get('t3', False))

spot4 = getSpotValue(request.POST.get('t4', False))

password = "none"

con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'GraphPassword',charset='utf8')

with con:

cur = con.cursor()

cur.execute("select username,spot1,spot2,spot3,spot4 FROM register")

rows = cur.fetchall()

for row in rows:

if row[0] == username:

old\_spot1 = getSpotValue(row[1])

old\_spot2 = getSpotValue(row[2])

old\_spot3 = getSpotValue(row[3])

old\_spot4 = getSpotValue(row[4])

if authspots(old\_spot1, spot1) and authspots(old\_spot2, spot2) and authspots(old\_spot3, spot3) and authspots(old\_spot4, spot4):

password = "success"

break

if password == "success":

context= {'data':'Welcome '+username+"<br/>Login successfull"}

return render(request, 'UserScreen.html', context)

else:

context= {'data':'Invalid spot selection'}

return render(request, 'Login.html', context)

def PasswordAction(request):

if request.method == 'POST':

global username, password, contact, email, address

spot1 = request.POST.get('t1', False)

spot2 = request.POST.get('t2', False)

spot3 = request.POST.get('t3', False)

spot4 = request.POST.get('t4', False)

db\_connection = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'GraphPassword',charset='utf8')

db\_cursor = db\_connection.cursor()

student\_sql\_query="INSERT INTO

register(username,password,contact,email,address,spot1,spot2,spot3,spot4) VALUES('"+username+"','"+password+"','"+contact+"','"+email+"','"+address+"','"+spot1+"','"+spot2+"','"+spot3+"','"+spot4+"')"

db\_cursor.execute(student\_sql\_query)

db\_connection.commit()

print(db\_cursor.rowcount, "Record Inserted")

if db\_cursor.rowcount == 1:

context= {'data':'Signup Process Completed'}

return render(request, 'Register.html', context)

else:

context= {'data':'Error in signup process'}

return render(request, 'Register.html', context)

def RegisterAction(request):

if request.method == 'POST':

global username, password, contact, email, address

username = request.POST.get('t1', False)

contact = request.POST.get('t3', False)

email = request.POST.get('t4', False)

address = request.POST.get('t5', False)

password = request.FILES['t2'].name

myfile = request.FILES['t2']

fs = FileSystemStorage()

output = "none"

con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'GraphPassword',charset='utf8')

with con:

cur = con.cursor()

cur.execute("select username,email FROM register")

rows = cur.fetchall()

for row in rows:

if row[0] == username:

output = username+" Username already exists"

break

if row[1] == email:

output = email+" Email id already exists"

break

if output == "none":

fs.save('GraphPasswordApp/static/password/'+password, myfile)

output = '<center><img src="/static/password/'+password+'" alt="" width="400" height="300" id="myImgId" onmousemove="getPos(event)"/></center>'

context= {'data':output}

return render(request, 'ShowImage.html', context)

else:

context= {'data':output}

return render(request, 'Register.html', context)

def ViewUsers(request):

if request.method == 'GET':

output = '<table border=1 align=center width=100%>'

font = '<font size="" color="black">'

arr = ['Username','Password Image','Contact No','Email ID','Address','Spot1','Spot2','Spot3','Spot4']

output += "<tr>"

for i in range(len(arr)):

output += "<th>"+font+arr[i]+"</th>"

con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'GraphPassword',charset='utf8')

with con:

cur = con.cursor()

cur.execute("select \* FROM register")

rows = cur.fetchall()

for row in rows:

output += "<tr><td>"+font+row[0]+"</td>"

output+='<td><img src=/static/password/'+row[1]+' height=100 width=100/></td>'

output += "<td>"+font+row[2]+"</td>"

output += "<td>"+font+row[3]+"</td>"

output += "<td>"+font+row[4]+"</td>"

output += "<td>"+font+row[5]+"</td>"

output += "<td>"+font+row[6]+"</td>"

output += "<td>"+font+row[7]+"</td>"

output += "<td>"+font+row[8]+"</td>"

context= {'data':output}

return render(request, 'ViewUsers.html', context)

def UpdatePassword(request):

if request.method == 'POST':

global username, password

spot1 = request.POST.get('t1', False)

spot2 = request.POST.get('t2', False)

spot3 = request.POST.get('t3', False)

spot4 = request.POST.get('t4', False)

db\_connection = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'GraphPassword',charset='utf8')

db\_cursor = db\_connection.cursor()

student\_sql\_query = "update register set password='"+password+"',spot1='"+spot1+"',spot2='"+spot2+"',spot3='"+spot3+"',spot4='"+spot4+"' where username='"+username+"'"

db\_cursor.execute(student\_sql\_query)

db\_connection.commit()

print(db\_cursor.rowcount, "Record Inserted")

if db\_cursor.rowcount == 1:

context= {'data':'Password Reset Process Completed'}

return render(request, 'Register.html', context)

else:

context= {'data':'Error in reset password'}

return render(request, 'Reset.html', context)

def ResetAction(request):

if request.method == 'POST':

global username, password

password = request.FILES['t1'].name

myfile = request.FILES['t1']

fs = FileSystemStorage()

fs.save('GraphPasswordApp/static/password/'+password, myfile)

output = '<center><img src="/static/password/'+password+'" alt="" width="400" height="300" id="myImgId" onmousemove="getPos(event)"/></center>'

context= {'data':output}

return render(request, 'ShowResetImage.html', context)

def AdminLoginAction(request):

if request.method == 'POST':

user = request.POST.get('t1', False)

password = request.POST.get('t2', False)

if user == 'admin' and password == 'admin':

context= {'data':'Welcome '+user}

return render(request, 'AdminScreen.html', context)

else:

context= {'data':'Invalid login'}

return render(request, 'AdminLogin.html', context)

**CHAPTER 8**

**OUTPUT SCREENS**

**CHAPTER 9**

**CONCLUSION**

**9.1 Introduction**

This chapter will discuss about writing a conclusion which is the final part of the research paper. Conclusion usually allow researcher to have some last words of subject which include this research project. It also allows can synthesize our thoughts and to discuss about future work.

**9.2 System Contribution**

Graphical password authentication implemented in web-based system is an alternative password that can replace the standard textual-based password. The objective of the system that have achieved were to design a graphical password authentication implemented in web-based system. Then to implement the graphical password authentication using passpoints scheme. Finally, to test the effectiveness of

graphical password authentication system using passpoints scheme to authenticate users by using web system.

**9.3 Future Work**

In the future, hopefully this system can be applied in real life because it may help users that have secret or privacy account want to keep their account private and protect their data privacy. To make this more secure, the selection of every click points might be good if they have their own character or password.

**9.4 Summary**

In conclusion, it is important to know what kind of algorithm are suitable for a system and the way to implement the algorithm in a system. In this proposed project, graphical password authentication by using passpoints scheme can give many benefits to users in many aspects. It will secure the users to make an authentication process in spite of the fact it takes users longer time to access into a system.

**CHAPTER 10**

**FUTURE ENHANCEMENT**

**CHAPTER 11**

**REFERENCES**

1. Wiedenbeck, S., Waters, J., Birget, J.-C., Brodskiy, A., & Memon, N. (2005). PassPoints: Design and longitudinal evaluation of a graphical password system. p. 26.
2. Aakansha Gokhale, & Vijaya Waghmare. (2013). Graphical Password Authentication Techniques: A Review. 7.
3. Ahmet Emir Dirik, Nasir Memon, & Jean-Camille Birget. (2007). Modeling user choice in the PassPoints graphical password scheme. 8.
4. Nelson, D. L., Reed, V. S., & Walling, J. R. (1976). Pictorial superiority effect. Journal of experimental psychology. Human learning and memory, 2(5), 523–528.
5. Dhamija, R. (n.d.). Hash Visualization in User Authentication . 2.
6. Khan , W. Z., & Aalsalem, M. Y. (19 December, 2013). A Graphical Password Based System for Small Mobile Devices. p. 11.
7. Manjunath G, Satheesh K, Saranyadevi C, & Nithya M. (2014). Text-Based Shoulder Surfing Resistant Graphical Password Scheme. 4.
8. N.Asokan. (16 May, 2014). A Closer Look at Recognition-based Graphical Passwords. p. 13.
9. Tao, H. (2006). Pass-Go, a New Graphical Password Scheme. 11.
10. Towseef Akram , Vakeel Ahmad, Israrul Haq, & Monisa Nazir. (2017). Graphical Password Authentication. 7.
11. Vishal Kolhe, Vipul Gunjal, Sayali Kalasakar, & Pranjal Rathod. (2013). Secure Authentication with 3D Password. 7.
12. Zheng, Z., Xiyu Liu , Lizi Yin , & Zhaocheng Liu. (2010). A Hybrid Password Authentication Scheme Based on Shape and Text. 8.
13. Awais, A., Muhammad , A., M., K. H., & Talib, R. (2016). Secure Graphical Password Techniques agaist Shoulder Surfing and Camera based Attacks. 9.
14. Krishnan, S., Watkins, K.E. & Bishop, D.V. (2017). The effect of recall, reproduction, and restudy on word learning: a pre-registered study. BMC Psychol 5, 28.44
15. Borkar, V. S., & Golar, P. C. (2015). Click Based Graphical Passward with Text Password Authentication. International Journal of Computer Science and Network Security, 15(11), 76-79.
16. Chiasson, S., Stobert, E., Forget, A., Biddle, R., & Van Oorschot, P. C. (2012).Persuasive cued click-points: Design, implementation, and evaluation of a knowledge-based authentication mechanism. IEEE Transactions on Dependable and Secure Computing, 9(2), 222–235. https://doi.org/10.1109/TDSC.2011.55
17. Rupavathy, N., Carmel Mary Belinda, M. J., & Nivedhitha, G. (2018). A shoulder surfing resistance using graphical authentication system. International Journal of Engineering and Technology (UAE), 7(1.7 Special Issue 7), 169– 174. https://doi.org/10.14419/ijet.v7i1.7.10644
18. Lashkari, A. H., Gani, A., Sabet, L. G., & Farmand, S. (2010). A new algorithm on Graphical User Authentication (GUA) based on multi-line grids. Scientific Research and Essays, 5(24), 3865–3875.
19. Science, C., & Security, I. (2008). STUDY AND DEVELOP A NEW GRAPHICAL PASSWORD SYSTEM Dedicated to my beloved parents , my lovely wife , my children , brothers and sisters With thanks for all the years of caring , love , and support . November.
20. Zuo, M., Zeng, G., & Tu, X. (2010). Research and improvement of face detection algorithm based on the OpenCV. 2nd International Conference on Information Science and Engineering, ICISE2010 - Proceedings, 1413–1416 https://doi.org/10.1109/ICISE.2010.5691414