CHAPTER 1

INTRODUCTION

1.1 Scope of Work

Over the last decade, internet has expanded beyond comparison. This makes the problem of security, that much more difficult to solve. Since decades, since the first mailing service, we have been using only the text based passwords. Only recently, an era of image based passwords along with CAPTCHA based user authentication methods have been established. They serve the people in innumerable ways in day to day life, maintaining security where required.

Security is an important factor in the current scenario of the Web. Until now, we have mostly used only text based passwords. This led to the birth to Captcha, the automated test to differentiate humans and bots. After using Captchas extensively in websites, image based password mechanisms were born. Captcha overcomes many security issues, which are inherent in image based password techniques. We use techniques which overcome other click-based graphical password methods and extend the scope of this subject. Even if it is not applicable or incompatible with some applications, in others, it fits well for applications that need to fend off dictionary attacks and brute force attacks.

1.2 Motivation

With more and more stories of cracking in the news, developers are looking for the best ways of securing their sites. If your site has a member system, it could be at risk from being cracked and your users' data could be compromised. This project will attempt at making a secure login using Image Based Passwords. The code is as good as we can make it, but security and especially encryption are complex subjects that are changing all the time and we can't claim to have the entire field mastered. Therefore we may have missed a few tricks in our code.

Writing a login system is a complex topic and not something to be undertaken by those who are not intimately familiar with a wide variety of security topics. The login system presented here is to be used for educational purposes, or even a production environment.

Few types of Graphical Passwords:

- **Recall Based Techniques**

  A user is asked to reproduce something that he created or selected earlier during the registration stage

- **Recognition Based Techniques**

  A user is presented with a set of images and the user passes the authentication by recognizing and identifying the images he selected during the registration stage.

- **Cued-recall Technique**

  An extra cue is provided to users to remember and target specific locations within a presented image.

CHAPTER 2

LITERATURE SURVEY

Cyber security is an important issue to tackle. Various user authentication methods are used for this purpose. it helps to avoid misuse or illegal use of highly sensitive data. Text and graphical passwords are mainly used for authentication purpose. But due to various flaws, they are not reliable for data security. Text passwords are insecure for reasons and graphical are more secured in comparison but are vulnerable to shoulder surfing attacks. Hence by using graphical password system and captcha technology a new security primitive is proposed. We call it as captcha as graphical password (carp). Carp is a combination of both a captcha and a graphical password scheme. in this paper we conduct a comprehensive survey of existing carp techniques namely clicktext, clickanimal and animalgrid. We discuss the strengths and limitations of each method and point out research direction in this area.

In the current challenges facing technology, the encryption challenge has become a major point of concern. The utilization of text-in- image predicated security checks has availed to reduce fraud, but it’s not in its optimal efficiency. Looking to the future, we can visually perceive that for advanced transactions, mobile SMS verification and text predicated captchas are not enough. Hence we require to come up with an incipient scheme of security, which involves images and security primitives that are predicated on hard mathematical quandaries. Utilizing hard AI quandaries for security is emerging as an exhilarating incipient paradigm, but has been underexplored. The rudimentary task in security is to engender cryptographic primitives predicated on hard mathematical quandaries that are computationally intractable.

Albeit the proposed system may not be a hundred percent efficient approach, we are sure that it will surmount the current drawbacks being faced in internet transactions. The proposed system is an endeavour at simplifying the process of online payments or bank transfers etc. If implemented prosperously, this scheme may provide maximum security. The scope for future video-predicated sanctions withal amplifies.

Security awareness is an important factor in an information security program. While organizations and institutes expand their use of advanced security technology and continuously train their security professionals, fraction of it is used to increase the security awareness among the normal users. As a result, today, organized cyber criminals are trying hard towards research and development of advanced hacking methods that can be used to steal money and secured information from the general public. Password authentication is one of the most common building blocks in implementing access control. Each user has a relatively short sequence of characters commonly referred to as a password. To gain access, providing right password is essential. Common attack for breaking password authenticated systems is dictionary attack [2]. Graphical password is an option for alphanumeric password as text password is slightly hard to remember text password. When any application is provided with user friendly authentication it becomes easy to break and use that application. Cloud security can also be given by alphanumeric password but thing matter is that use of alphanumeric is not that much of secure and easy to remember. Any individual examining the password can memorize it which may lead to its misuse. Graphical password schemes are more reliable and more resilient to dictionary attacks than textual passwords, but more vulnerable to shoulder surfing attacks [3]. Captcha (Completely Automated Public Turing tests to tell Computers and Humans Apart) is a program that generates and grades tests that are human solvable, but current computer programs do not have the ability to solve them. The robustness of Captcha is found in its strength in resisting automatic adversarial attacks, and it has many applications for practical security, including free email services, online polls, search engine bots, preventing dictionary attacks, worms and spam CaRP is a combination of both a Captcha and a graphical password scheme. CaRP overcome a number of security issues, such as relay attacks, online guessing attacks, and, if combined with captcha and graphical password, shoulder-surfing attacks. CaRP is click-based graphical passwords, where order of clicks on an image is used to get a new password. Unlike other click-based graphical passwords, images used in CaRP are used to generate captcha challenges, and for every login attempt a new CaRP image is generated whether the existing user tries authenticating or a new user. In this paper we conduct a comprehensive survey of existing CaRP techniques namely ClickText, ClickAnimal and AnimalGrid. We point out research direction in this area. We also try to answer our CaRP as secured as graphical passwords and text based passwords. Survey will be useful for information security researchers and practitioners who are interested in finding an alternative to graphical authentication methods.

2.1 Existing System

A. **CAPTCHA**

A captcha is a program that can generate and grade tests that: (A) most humans can pass, but (B) current computer programs cannot pass. Such a program can be used to differentiate humans from computer. There are two types of visual captcha: text captcha and Image-Recognition captcha(IRC). Captcha can be circumvented through relay attacks whereby captcha challenges are relayed to human solvers.

B**. GRAPHICAL PASSWORD**

Graphical password schemes have been proposed as a possible alternative to alphanumeric schemes, motivated partially by the fact that humans can remember images easily than text; psychological studies supports such assumption [8]. Images are generally easier to be remembered than text. In addition, if the number of possible images is enough large, the possible password space of a graphical password scheme may exceed that of text-based schemes and thus presumably offer better resistance to dictionary attacks. Because of these (presumed) advantages, there is a increasing interest in graphical password.

Advantages:

1. It offers reasonable security and usability and appears to fit well with some practical applications for improving online security.
2. This threat is widespread and considered as a top cyber security risk. Defense against online dictionary attacks is a more subtle problem than it might appear.

CHAPTER 3

REQUIREMENTS

3.1 Introduction

Requirements analysis or study is an important phase of any system development process. The system is studied to the minute detail and analyzed. The system analyst digs deep into the working of the present system. The system was viewed as a whole and the input of the system are identified. During analysis phase for each problem identified many alternative solutions were evaluated and selected the most feasible one. A feasibility analysis was performed to evaluate possible solutions to recommend the most feasible one. Feasibility study is defined as evaluation or analysis of the potential impact of a proposed project or program. The objective is to determine whether the proposed system is feasible.

Technical feasibility assesses whether the current technical resources are sufficient for the new system. If they are not available, can they be upgraded to provide the level of technology necessary for the new system? It checks whether the proposed system can be implemented in the present system without supporting the existing hardware. Economic feasibility determines whether the time and money are available to develop the system. It also includes the purchase of new equipment, hardware, and software. A software product must be cost effective in the development, on maintenance and in the use. Since the hardware and resources are already available with the organization and the organization can afford to allocate the required resources. Operational feasibility determines if the human resources are available to operate the system once it has been installed. The resources that are required to implement or install are already available with the organization. The persons of the organization need no exposure to computer but have to be trained to use this particular software. A few of them will be trained. Further, training is very less. The management will also be convinced that the project is optimally feasible.

3.2 Functional Requirements

- Selection of first image during registration.
- Selection of an IBP for each file uploaded
- Various modules for processing inputs
- Customizable passwords
- Various modules for processing animations

3.3 Non Functional Requirements

- Performance

  For any system to perform as expected, we need to have performance measures. The IBP mechanism needs to be generated easily and quickly. Even while working with images, we try to load the page faster.

  - **Availability**

    Availability of a system is typically measured as a factor of its reliability - as reliability increases, so does availability (that is, less downtime). Availability of a system may also be increased by the strategy of focusing on increasing testability and maintainability and not on reliability. Improving maintainability is generally easier than reliability. Here it may mean that the file stored on the server must be always available.

  - **Latency**

    Latency is a time delay between the cause and the effect of some physical change in the system being observed. Latency is a result of the limited velocity with which any physical interaction can take place. This velocity is always lower or equal to speed of light. Therefore every physical system that has spatial dimensions different from zero will experience some sort of latency. Here latency applies on every user click, on client-server interactions etc.

- **Response Time**

  Response time is the total amount of time it takes to respond to a request for service. In computing, that service can be any unit of work from a simple disk IO to loading a complex web page. The response time is the sum of three numbers:

  - Service time - How long it takes to do the work requested.
  - Wait time - How long the request has to wait for requests queued ahead of it before it gets to run.
  - Transmission time – How long it takes to move the request to the computer doing the work and the response back to the requestor.

- **Scalability**

  Scalability is the ability of a system, network, or process to handle a growing amount of work in a capable manner or its ability to be enlarged to accommodate that growth. We maintain this factor by having a large database which can house many files. The passwords though unique and unlimited in number, can be stored in the database efficiently. In large scale deployment, we have to use multiple databases and web servers to serve hundreds of users.

- <u>Manageability/Maintainability</u>

  - **Monitoring**

    Monitoring is the way of checking out every part of the system for accuracy and correctness. We monitor the system through the database. There are a wide variety of database tools, one of them being phpmyadmin. This helps us monitor the data in the database.

  - **Maintenance**

    The fact that each type of password is independent of each other gives us the advantages to update the system as and when required. We need to isolate defects or their cause, correct defects or their cause, prevent unexpected breakdowns etc.

- <u>Security</u>

  - **Protection**

    Credentials must be encrypted. They must be transferred over a secure connection to the server. Here we use php functions to encrypt and decrypt the data.

  - **Authorization and Authentication**

To access any feature of IBP, the user has to be logged in. While uploading files, he must set a password which he must use to download the same file. The password must be sharable among users so that they can access the file also.

3.4 Hardware Requirement

Processor type	Intel Core i3/i5/i7 x86
Speed	3.0 GHz
Ram	1024 MB
Hard Disk	500 GB
Keyboard	Standard

Table 3.1 – Hardware Requirements

3.5 Software Requirement

Operating System	Windows 7/8/8.1, Ubuntu, Mac OSX
Browser	Mozilla, Chrome, Opera
Application Server	Apache 2.4.9
Front End	HTML5,CSS/CSS3, JavaScript
Scripts	JavaScript
Server side Script	PHP 5.5.12
Database	MYSQL 5.6.17
Database Connectivity	mysqli_connect

Table 3.2 – Software Requirements

3.6 Key Features

- Recognizing the pattern selected during sign up and storing the credentials in database and matching the user credentials along with graphical password during login. Session created after successful login.
- Uploading a file to the server. Choose a type of graphical password required while downloading a file
- Good Animations. Highlight image on click.
- Users can register their account by providing a username, password, email and graphical password.
- Database connection is established to store the credentials in a database table.
- If the user click on a particular uploaded file they will be asked to login.
- Users can login into their account by submitting their credentials that they have used while registering their account.
- The credentials entered will be matched with the credentials that are stored in the database and the user will be alerted the status of his login.
- If successful users can download files available that the user has uploaded or other users have uploaded by entering the custom password.
- Users while uploading a file need to specify a pattern to create a graphic password that sr4any user will be asked while downloading the file.
- Circle pattern helps in generating a graphical password that has values of the coordinates that the user chooses by clicking at a point in a given image
- The data entered in the upload form will be stored in the database.
- If the user wants to download a file they need to be logged in.
- If the credentials submitted match with the credentials stored in the database based on the file chosen. A link will be generated for the user to download the file.
- If the image credentials don’t match with those stored in the database based on the file chosen alert will be provided to user that the credentials don’t match and they will be asked to re-submit the credentials again

CHAPTER 4

SOFTWARE DESCRIPTION

4.1 HTML, CSS, JavaScript

HTML, Hypertext Markup Language is the building blocks of Web Applications. Hyper Text Markup Language, commonly referred to as **HTML**, is the standard markup language used to create web pages. It is written in the form of **HTML** elements consisting of tags enclosed in angle brackets (like <**html**>). Web browsers can read HTML files and render them into visible or audible web pages. Browsers do not display the HTML tags and scripts, but use them to interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language, rather than a programming language. HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages. Web browsers can also refer to Cascading Style Sheets (CSS) to define the look and layout of text and other material. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1999.

Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to change the style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging WebPages, user interfaces for web applications, and user interfaces for many mobile applications.

JavaScript also known as ECMA Script (the un-trademarked name used for the standard), is a dynamic programming language. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. It is also used in server-side network programming with runtime environments such as Node.js, game development and the creation of desktop and mobile applications. JavaScript is classified as a prototype-based scripting language with dynamic typing and first-class functions. This mix of features makes it a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles. Despite some naming, syntactic, and standard library similarities, JavaScript and Java are otherwise unrelated and have very different semantics. The syntax of JavaScript is actually derived from C, while the semantics and design are influenced by the Self and Scheme programming languages. JavaScript is also used in environments that aren't web-based, such as PDF documents, site-specific browsers, and desktop widgets. Newer and faster JavaScript virtual machines (VMs) and platforms built upon them have also increased the popularity of JavaScript for server-side web applications. On the client side, JavaScript has been traditionally implemented as an interpreted language, but more recent browsers perform just-in-time compilation.

4.2 PHP/MySQL

PHP is an open source, server-side scripting language designed for web development which is also used as a general-purpose programming language. PHP stands for Hypertext Preprocessor. A PHP file is one that has a default extension .php extension that follows its file name. It can include text, HTML, JavaScript code and PHP code. PHP code executes on the server, the result is then returned to the browser as plain HTML. There are several tasks PHP does, some of which are, collect form data, send and receive cookies, generate dynamic page content, encrypt data. The advantages of using PHP which makes it widely-used are, it runs on various platforms like Windows, Linux, UNIX, etc. it is compatible with numerous servers used today like Apache,with PHP you are not limited to output HTML. You can output images, PDF files, and even flash movies. You can also output any text, such as XHTML and XML. PHP has built in support for Mysql and comes with MYSQL alongside with it.

4.3 JQuery

JQuery is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML. JQuery is the most popular JavaScript library in use today JQuery is free, open-source software licensed under the MIT License. JQuery’s syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications. JQuery also provides capabilities for developers to create plug-ins on top of the JavaScript library. This enables developers to create abstractions for low-level interaction and animation, advanced effects and high-level, theme-able widgets. The modular approach to the JQuery library allows the creation of powerful dynamic web pages and web applications. The set of JQuery core features—DOM element selections, traversal and manipulation—enabled by its selector engine (named "Sizzle" from v1.3), created a new "programming style", fusing algorithms and DOM data structures. This style influenced the architecture of other JavaScript frameworks like YUI v3 and Dojo, later stimulating the creation of the standard Selectors API. Microsoft and Nokia bundle JQuery on their platforms. Microsoft includes it with Visual Studio for use within Microsoft's ASP.NET AJAX framework and ASP.NET MVC Framework while Nokia has integrated it into the Web Run-Time widget development platform. JQuery has also been used in Media Wiki since version 1.16.

4.4 WAMP Server

WampServer refers to a software stack for the Microsoft Windows operating system, created by Romain Bourdon and consisting of the Apache web server, OpenSSL for SSL support, MySQL database and PHP programming language. The latest stable release is v2.5. It is secured under the GNU GPL License.

4.5 JCrop

JCrop is really a Javascript object. When we refer to the "JCrop API" we are really talking about the actual underlying object. In the source code, this is referenced as core, because it forms the core of a Jcrop instance. The JCrop API controls every facet of the JCrop instance. Most integration will require little, if any, use of the API. Sometimes you'll be using the API without even really knowing it. Some developers will feel more comfortable directly interacting with JCrop via Javascript methods. Others may require more direct access for reasons of developing or debugging Filters or other JCrop components.

The JCrop core performs the following:

- Creates a new JCrop instance by attaching to the container

- Initializes and manages all components used by JCrop
- Stores option settings
- Manages Selections
- Manages and assigns filters to new Selections
- Provides a basic instance-wide set of methods

CHAPTER 5

SYSTEM DESIGN

5.1 Proposed Model

The system proposed here is a brainchild of the CaRP scheme created initially as a security primitive. We take that scheme on a different path, using what we need and leaving out the unnecessary. In Carp, they use a captcha and a graphical scheme. Here we make the user select various images and gestures related to images, as credentials.

The proposed system consists of a basic login where the user must set his identity by selecting an IBP. He must the use the same IBP to login each time. Then the system includes an upload and download region, where the user selects any file to upload. While selecting he sets up a password for the file using one of the methods, such as Circle Selection, Grid Based or Number keypad. IBP is both a Captcha and a graphical image password scheme. IBP addresses a number of security problems altogether, such as online guessing attacks, relay attacks, and, if combined with dual-view technologies, shoulder-surfing attacks.

We implement a set of password mechanisms in a single package. These passwords are of the type basic, circle selection and grid image, each used in its appropriate place, for specific purposes. Under this paradigm, the most notable primitive invented is Captcha, which distinguishes human users from computers by presenting a challenge.

<u>We propose this model</u>

- <u>Basic Login/Signup</u>

  The user clicks login tab. He must enter his username, text password. Then he must enter the graphical password. This system tests his authenticity and also whether he is a human. He is required to select three images as his password, from a set of images. The images selected must conform to the pattern in which he had initially selected it.

  If the user is not registered, then he must signup first. He selects the signup tab, and then enters the required credentials such as username, password, email id and then the graphical password. The IBP is of type basic, where the user only selects 3 images.

- <u>Circle Selection</u>

  The user selects a circle selection IBP. Then he is shown an image and on that image he will have to select a variable size circle, which encloses part of the image. This is to be done while uploading the file to server. While downloading, he just selects a smaller fixed size circle inside the previously chosen image area.

- <u>Grid Based</u>

  In this type, we plot a grid over an image where the user has to select a pattern of grid cells while uploading the file. He must select a total of 8 cells. Later, while downloading he must click and select the same 8 cells in the same pattern that they were chosen.

- <u>Number Based</u>

  In this type, the user must select a total of 8 numbers from a keyboard layout. Later, while downloading he must click and select the same 8 numbers in the same pattern that they were chosen, from the same keypad. The keypad will be generated automatically.

The system framework revolves around a homepage and demo-page. The demo-page consists of the login signup, upload, and download page as the front page. We build our application on HTML which defines the document structure and composition. We use CSS to define and describe various styling elements. They are further sculpted with JavaScript which allows transitions and animations to be possible.

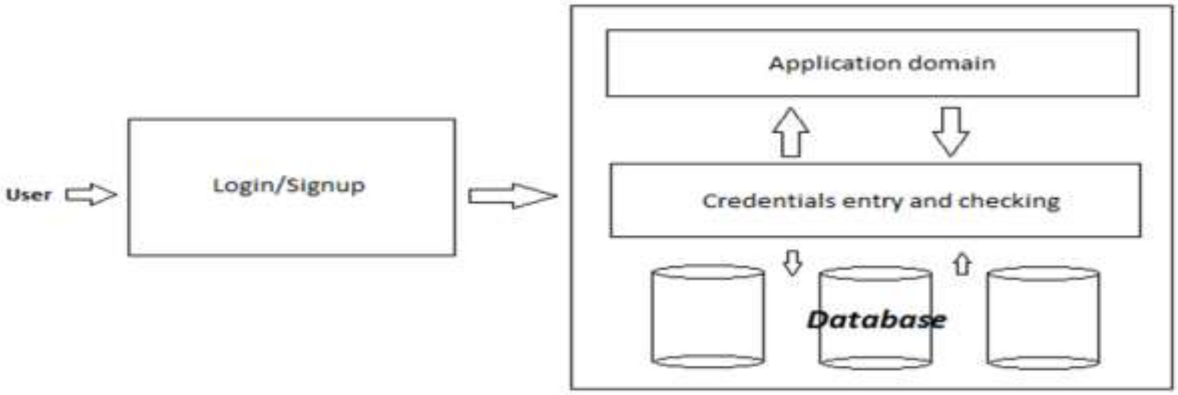


Fig 5.1 – Basic System Design

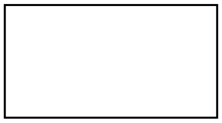
5.2 High Level Design

5.2.1 Data Flow Diagrams:

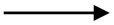
Data flow diagrams (DFD) was first developed by Larry Constantine as way representing system requirements in a graphical form; this lead to modular design. A DFD describes what data flow (logical) rather than how they are processed, so it does not depend on hardware, software, data structure or file organization. It is also known as ‘bubble chart’.

A Data Flow Diagrams is a structured analysis and design tool that can be used for flowcharting in place of, or in association with, information-oriented and process-oriented systems flowcharts. A DFD is a network that describes the flow of data and the processes that change, or transform, data throughout a system. This network is constructed by using a set of symbols that do not imply a physical implementation. It has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So it is the starting point of the design phase that functionality decomposes the requirement specifications down to the lowest level of detail.

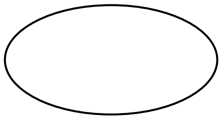
The symbols used to prepare DFD do not imply a physical implementation, a DFD can be considered to an abstract of the logic of an information-oriented or a process-oriented system flow-chart. For these reasons DFDs are often referred to as logical data flow diagrams. The four basic symbols used to construct data flow diagrams are shown below:



A rectangle represents a data source or destination.



A directed edge represents the flow of data that is data stream.



An enclosed figure, usually a circle or an oval bubble, represents a process that transforms data Streams.

These are symbols that represent data flows, data sources, data transformations and data storage. The points at which data are transformed are represented by enclosed figures, usually circles, which are called nodes. The principle processes that take place at nodes are:

1. combining data streams
2. splitting data streams
3. Modifying data streams.

5.2.2 Basic Structures

1. Level 1 data flow

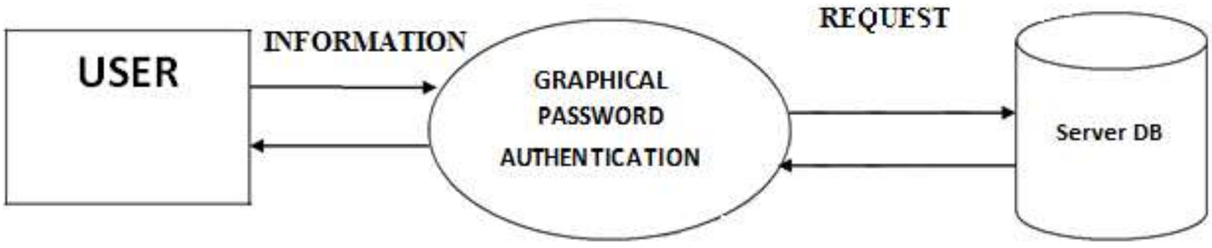


Fig 5.2 – System Data Flow

2. Register /Login /Upload /Update: Level 2 data flow

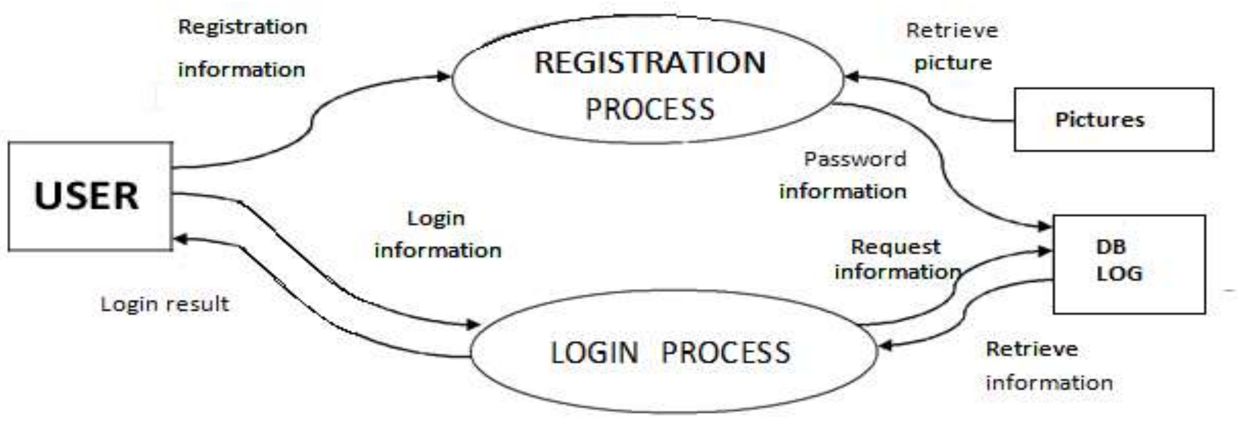


Fig 5.3 – Level 2 Internal Process

5.2.3 Basic Login Process:

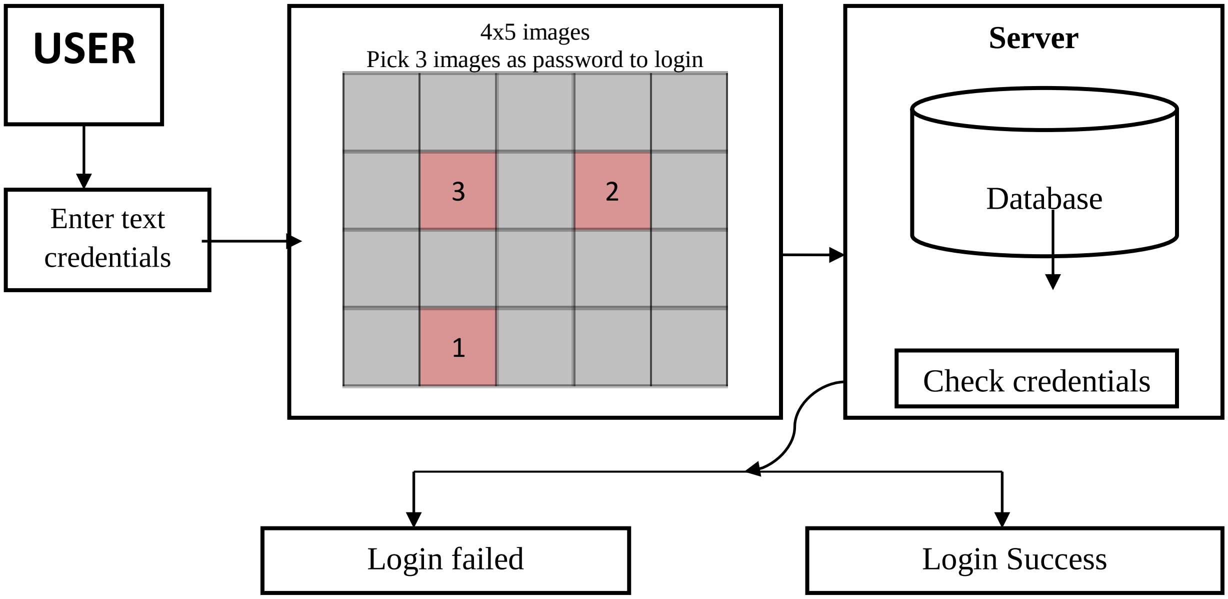
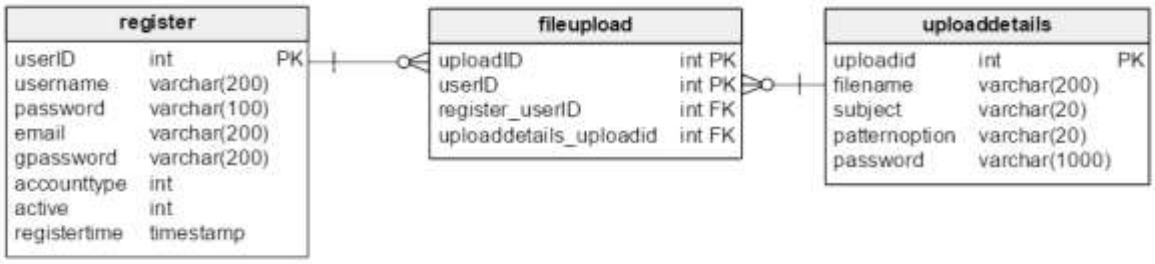


Fig 5.4 – Login Implementation

5.2.4 Database Design



register		
userID	int	PK
username	varchar(200)	
password	varchar(100)	
email	varchar(200)	
gpassword	varchar(200)	
accounttype	int	
active	int	
registertime	timestamp	

fileupload	
uploadID	int PK
userID	int PK
register_userID	int FK
uploaddetails_uploadid	int FK

uploaddetails		
uploadid	int	PK
filename	varchar(200)	
subject	varchar(20)	
patternoption	varchar(20)	
password	varchar(1000)	

Fig 5.5 – Database Design

5.2.5 After login

1. Upload

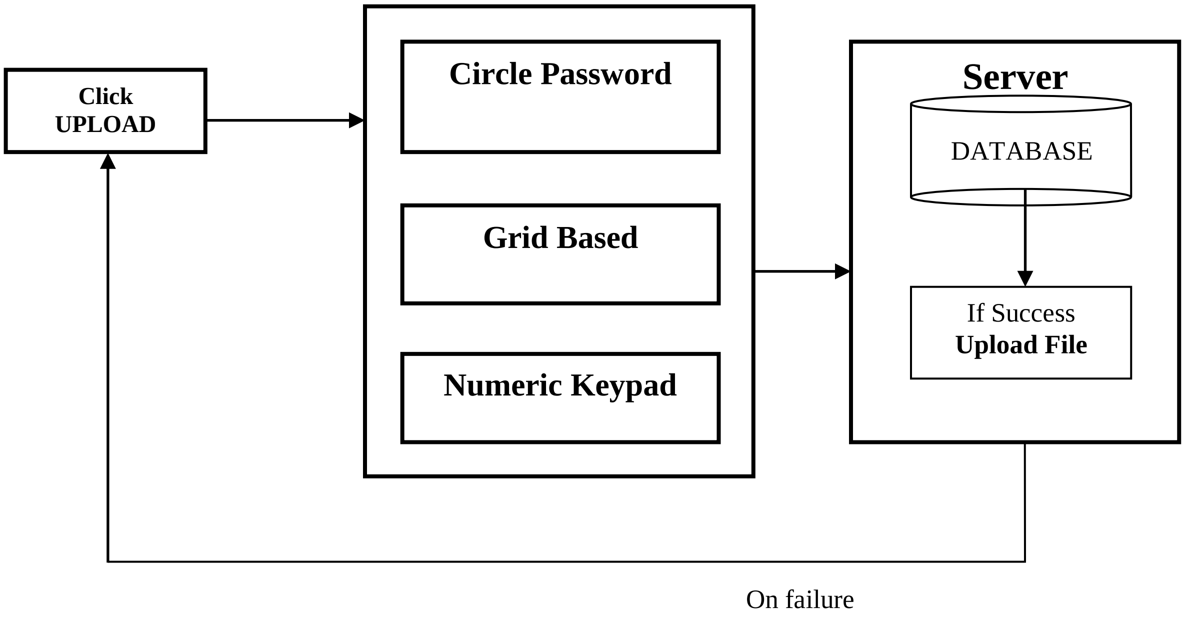


Fig 5.6 – File Upload Process

2. Download

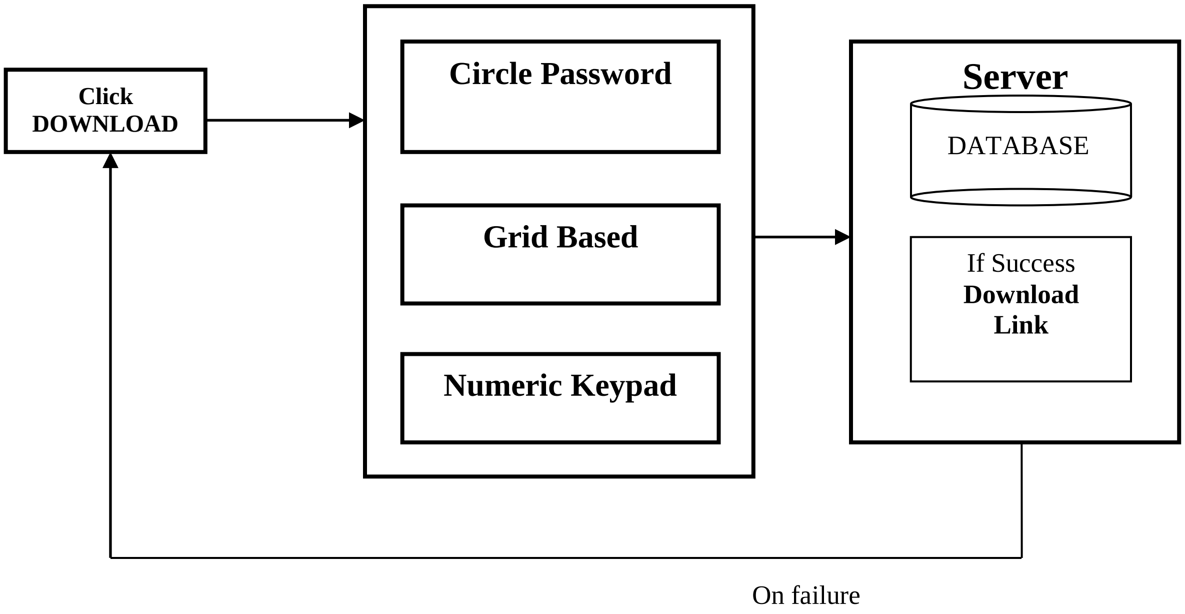


Fig 5.7 - File Download Process

5.2.5 Use Case Diagram

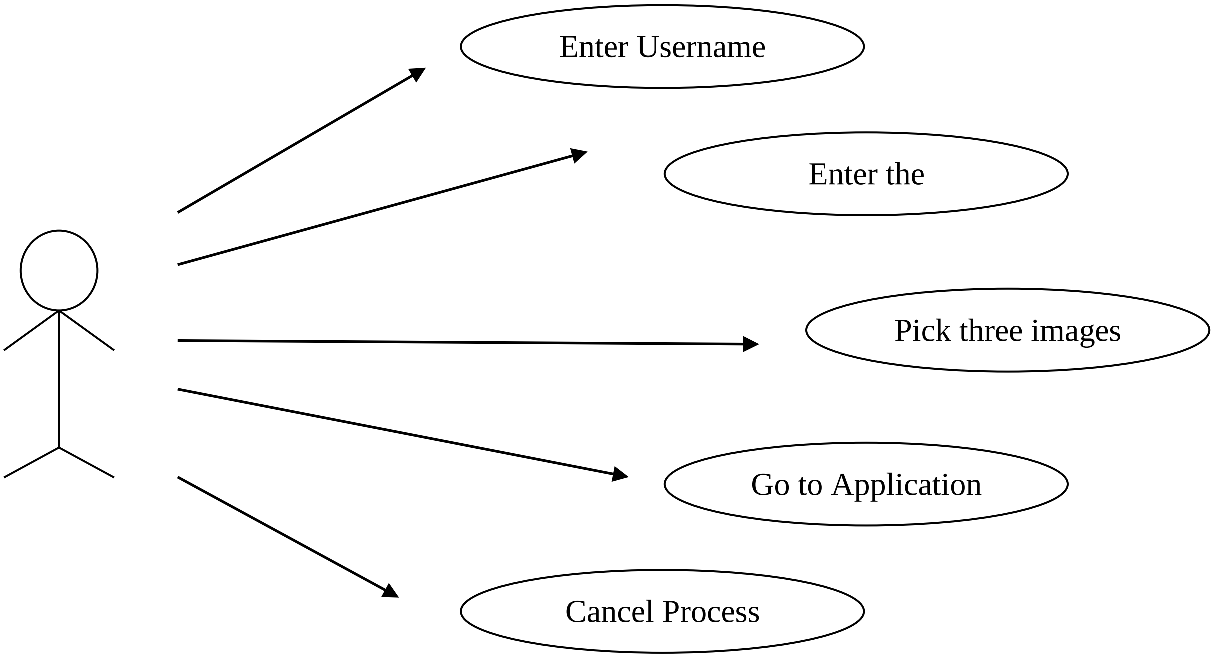


Fig 5.8 – Login Use Case

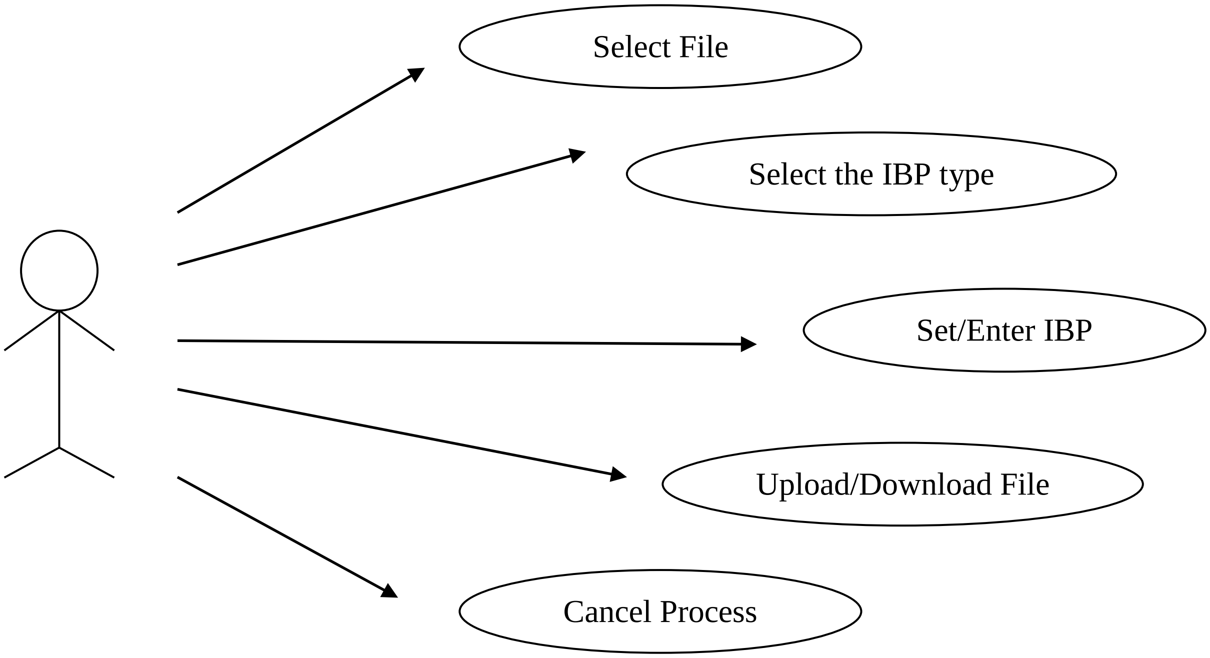


Fig 5.9 – Upload/Download Use Case

CHAPTER 6

IMPLEMENTATION

Most web hosting services will have PHP and MySQL already installed. You will just have to check they have the most recent versions of PHP and MySQL for this guide to work. If they don't have at least PHP5.3 and MySQL5 you might like to ask a few questions about their commitment to security. Keeping your software up-to-date is a part of the security process.

If you have your own server or computer, you should install the required software in the normal way for your system. Generally speaking, if you are not going to use the setup for production purposes and you are developing under Windows or OS/X, installing an XAMPP stack is the way to go. Get the appropriate version for your operating system at: http://www.apachefriends.org/en/xampp.html

When we login or signup, we need to select 3 images as a graphical password. The pattern will be recoded along with the images.

6.1 Basic Image Password

This is a basic type of Image based passwords which is derived from **CaRP** which is a combination of both an Image and a password. Nowadays, this is becoming obsolete. Here we can see that the user selects a specified number of images as a password, which will be used while logging in.

SignUp

The user has to select an ID. Then the system checks the Database and determines if the ID is already registered. If not, it allows the user to register, else it prompts for new ID. Then after receiving a valid username, it checks if the user has entered the graphical password on submit. If not then it prompts to enter the password. The user has to select upto 3 images in a particular order. Upon entering the graphic password, the system contacts the server with the credentials for storage on the server-end.

Algorithm for SignUp

```
registerBTN.onclick = validate(){
        res1 = valUserName();
        res2 = valEmail();
        res3 = valPWD();
        res4 = promptImageSelect();
        if(res1 && res2 && res3 && res4)
                register();
        else
                prompt(“Invalid Details”);
}
register(){
        data = getData();
        initializeSQL();
        storeInDatabase();
}
```

Login

The user has to enter his ID and textual password. If it receives a valid username, it checks if the user has entered the graphical password on submit. If not then it prompts to enter the password. The user has to select up to 3 images in a particular order. Upon entering the graphic password, the system contacts the server with the credentials for validation on the server-end. If the credentials are correct, it responds with the appropriate message and logs in. A session is created under the username.

Algorithm for Login

```
signInBTN.onclick = checkCredentials(){
    UN = getUserNameFromDB();
PW = getPasswordFromDB();
    If(username == UN){
        If(password == PW){
            GPW = getUserGraphicPassword();
            If(GraphicPass == GPW)
                loginSuccess();
            Else
                loginFailed(WRONG_IBP);
        Else
            loginFailed(WRONG_PW);
    Else
loginFailed(WRONG_UN);
}
loginSuccess(){
    createsession();
    showIndePage();
}
```

6.2 Login Form

```
<form id="login_form" action="login.php"  method="post">
  <ul><li><label>Username</label><br/>
  <input  type="text" required class="tstb" id="tstb1" name="username" placeholder="Enter Username"
maxlength="20" /></li>
  <li><label>Password</label><br/>
  <input type="password"  required class="tstb" id="tstb2" name="password" placeholder="Enter desired
Password" /></li></ul>
 <p><input id="tstb2" class="tsbut"  type="submit" name="signin" value="Sign in" /></p>
 <p><label>Select your choosen graphical password</label><br/></p>
 <div class="divimage_select">
             LoadImages();
 </div>
  </form>
  </div>
```

```
<?php
session_start();
if(isset($_SESSION['username'],$_SESSION['login_string']))
{?>
    <li><?php echo $_SESSION['username']; ?></li>
    <li><a class="tsbut" href="logout.php">Logout </a></li>
 <?php }else{?>
             <li><a id="loginsignupa" class="tsbut">Login / Signup</a></li>
<?php }?>
```

```
<?php
if(isset($_SESSION['username'],$_SESSION['login_string'],$_SESSION['accountype']))
{
             echo showUsers($dbc);
}
else if(isset($_SESSION['username'],$_SESSION['login_string']))
{
             echo showOptionlist('csea');
             echo seminarTopics('csea',$dbc);
  }
else
{
             echo showseminartopics('csea',$dbc);
}?>
```

6.3 Function Show Files

```
function showOptionlist($subject){
          echo "<div class=option_list>";
          echo "<ul>";
          echo "<li>";
          echo "<select class=seminaroptions id=".$subject."_seminaroptions >";
          echo "<option value=null>Select a option</option>";
   echo "<option value=mine>Show only my files</option>";
   echo "<option value=others>Show other files</option>";
   echo "<option selected  value=all>Show all files</option>";
   echo "</select>";
          echo "</li>";
   echo "<li>";
          echo "<button class=upload_seminar>Upload </button>";
          echo "</li>";
          echo "</ul>";
   echo "</div>";
          echo "<br>";
          echo "<div class=login_seminartopics id=".$subject."_topicdiv>";
          echo "</div>";
}
```

```
function showseminartopics($subject,$dbc)
{

        $t="SELECT `uploadid`, `filename` FROM `uploaddetails` WHERE subject='$subject' ORDER BY 
filename";
        $y=@mysqli_query($dbc,$t);
        if(@mysqli_num_rows($y) >0)
        {
                echo "<h3 id=seminarheader>Files Uploaded<hr/></h3>";
                echo "<div id=seminarSel>";
                $currentLetter=null;
                while($row_fetch=@mysqli_fetch_array($y))
                {
                        foreach(explode(",",$row_fetch[1]) as $list )
                        {
                                ucfirst($list);
                                $firstletter=substr($list,0,1);

                                if($firstletter!== $currentLetter)
                                {
                                        echo "<br>".ucfirst($firstletter)."<br>--<br>";
                                        $currentLetter=$firstletter;
                                }
                                $seminar=$row_fetch[1];
                                $seminar=ucwords(strtolower($seminar));

                                echo "<span class=’nologseminar_click’ >";  echo $seminar;  echo 
"</span>";
                        }
                        echo "<br>";

                }
                echo "</div>";
        }
        else
        {
                echo "<h3 align=center>No files have been uploaded</h3>";
        }
        @mysqli_close($dbc);
}
```

This function lists the uploaded files available on server on the front page.

6.4 PsuedoCode

Login

```
$("#login_form").on("submit", function(e) {
e.preventDefault();
    var username=$('#tstb1').val();
    var password=$('#tstb2').val();
    if(loginimage.length < 3)
    {
        alert('Please select your graphical password');
    }
    else
    {
        var gp=loginimage.toString();
        $.post("login.php", {username:username,password:password,gp:gp},
        function(result){
            loginimage=[];
lresponse(result);
        });

    }
    });
$('#login_div .divimage_select ul li img,#login_div.divimage_select ul li img.border-highlight').click(function(){
        var fileName = $(this).attr('src');
        var selectid=$(this).attr('id');
        if(loginimage.length != 3){
            if(!($(this).hasClass('border-highlight'))){
                loginimage.push(fileName);
            }
            else{
                loginimage.pop();

            }
            $(this).toggleClass('border-highlight');
        }
        else{
            loginimage.length = 0;
            $('.regImgs').removeClass('border-highlight');
        }
    });
```

Upload File

```
$('.upload_seminar').click(function(){
        var sem_sub=$('#seminar_subject').val();
        $('#blanket').show(500);
        $('#upload_div').show();
    });
$("#uploadseminar").on('submit',(function(e) {
        e.preventDefault();

    if(imgcoordinates.length < 8)
```

```
        {
                var pattern=$('#upload_imagepass').val();
                if(pattern == 'pattern1')
                {
                    alert('Please draw a circle on the image.');
                }
                else if(pattern == 'pattern2')
                {
                    alert('Please click on 8-length graphical  numerical password');
                }
                else if(pattern == 'pattern3')
                {
                    alert('Please select 8 cells in a grid as your graphical password');

                }
        }
        else
        {
                var sem_subject=$('#seminar_subject').val();
                var patternpass=$('#pass_pattern').val();

        $('#sem_subject').val(sem_subject);


        $("#message").empty();
        $('#loading').show();

        $.ajax({
        url: "ajax_file_upload.php", // Url to which the request is send
        type: "POST",            // Type of request to be send, called as method
        data: new FormData(this), // Data sent to server, a set of key/value pairs (i.e. form fields and
        values)
        contentType: false,      // The content stype used when sending data to the server.
        cache: false,            // To unable request pages to be cached
        processData:false,       // To send DOMDocument or non processed data file it is set to false
        success: function(data)   // A function to be called if request succeeds
        {
                $('#loading').hide();
                alert(data);
                window.location.reload();

        }
        });

        }


}));

// Function to preview image after validation
$(function() {
$("#file").change(function() {
$("#message").empty(); // To remove the previous error message
var file = this.files[0];
var imagefile = file.type;
//var match= ["application/vnd.openxmlformats-
officedocument.presentationml.presentation","image/png","image/jpg"];
//if(!((imagefile==match[0]) || (imagefile==match[1]) || (imagefile==match[2])))
```

```
var matched= ["application/vnd.openxmlformats-
officedocument.presentationml.presentation","application/pdf","application/vnd.openxmlformats-
officedocument.spreadsheetml.sheet","application/vnd.openxmlformats-
officedocument.wordprocessingml.document"];
if(!((imagefile==matched[0]) || (imagefile==matched[1]) || (imagefile==matched[2] ||
(imagefile==matched[1]) || (imagefile==matched[3]))))
{
        $("#message").html("<p id='error'>Please select a valid document file
        </p>"+"<h4>Note</h4>"+"<span id='error_message'>Only MS office 2007 version of
        WORD,EXCEL,and POWERPOINT as well as  PDF files of any size will be allowed</span>");
        $('#file').val('');
return false;

}
else
{
var reader = new FileReader();
reader.onload = imageIsLoaded;
reader.readAsDataURL(this.files[0]);
}
});
});
function imageIsLoaded(e) {
$("#file").css("color","green");
};
$('#upload_imagepass').change(function(){
                  $('.password_patterns').hide();
                  var select_patt=$(this).val();
                  $('#u'+select_patt).show();
                  if(select_patt == 'pattern3')
                  {
                        var x = 10;
                        var y = 5;
                        var t = '<table cellspacing="0" border="1" cellpadding="0" class="grxd">';
                        for (var i = 1; i <= (x * y); i++) {
                        t += (i == 1 ? '<tr>' : '');
                        t += '<td style="cursor:pointer;"></td>';
                  if (i == (x * y)) {
        t += '</tr>';
                  } else {
        t += (i % 10 === 0 ? '</tr><tr>' : '');
                        }
                        }
                        t += '</table>';
                        $("#drawTable").html(t);
                        $(".grxd  td").click(function(){
                        var col = $(this).css('background-color');
                        var column = $(this).parent().children().index(this);
                  var row = $(this).parent().parent().children().index(this.parentNode);
                        var cellclicked="column-"+column+"row-"+row;
                        if(col == "rgba(0, 0, 0, 0)" || col == 'transparent')
                        {

                                if(imgcoordinates.length < 8)
                                {
                                          imgcoordinates.push("column-"+column+"row"+row);
                                          $( this ).css("background-color","#EEEEEE");
                                          $('#pass_pattern').val(imgcoordinates);
                                }
                                else
                                {
                                          alert('You have selected the maximum no of cells  as your
                                          graphical password');
                                }
                        }
                        else
                        {
                                imgcoordinates=[];
                                $('.grxd td').css("background-color","transparent");
                                }

                        });
                  }
                  else
                  {
                        $('#uploadsubmit').show();
                  }
                  imgcoordinates=[];
                  $('#pass_pattern').val('');
        });

var jcrop_api;

$('#uimage_pattern1').Jcrop({
                  maxSize:[180,180],
//onChange:   showCoords,
onSelect:   showCoords,
onRelease:  clearCoords
},function(){
jcrop_api = this;
});

$('#coords').on('change','input',function(e){
var x1 = $('#x1').val(),
x2 = $('#x2').val(),
y1 = $('#y1').val(),
y2 = $('#y2').val();
jcrop_api.setSelect([x1,y1,x2,y2]);
});

function showCoords(c)
{

$('#x1').val(c.x);
$('#y1').val(c.y);
$('#x2').val(c.x2);
$('#y2').val(c.y2);
$('#w').val(c.w);
$('#h').val(c.h);

        imgcoordinates.splice(0,2,c.x,c.y);
        imgcoordinates.splice(2,4,c.x2,c.y2);
        imgcoordinates.splice(4,6,c.w,c.h);
        imgcoordinates.splice(6,8,'0','0');
```

```
    $('#pass_pattern').val(imgcoordinates);
};

function clearCoords()
{
$('#coords input').val('');
};

    $("#upattern2 #pass img").on("click",function(){
if($(' #upattern2 #password ul').children().length == 8)
    {
        alert('Numerical graphical password maximum length has been reached ');
    }
    else
    {
        var imga = $(this).attr('src');
        $(" #upattern2 #password ul").append('<li><img src="' + imga + '" /><span
        class=close>&times;</span></li>');
        imgcoordinates.push(imga);
        $('#pass_pattern').val(imgcoordinates);


$("#upattern2 #password ul li span").click(function(){

    $(this).parents('li').remove();
    imgcoordinates=[];
    $('#upattern2 #password ul li img').each(function(i)
    {
        var numi=$(this).attr('src');
        imgcoordinates.push(numi);
        $('#pass_pattern').val(imgcoordinates);
    });
    });
}
});
```

Download file

```
$("#downloadseminar").on('submit',(function(e) {
    e.preventDefault();
    $("#message").empty();
    $('#loading').show();
    var pattern=$('#download_imagepass').val();
    $.ajax({
    url: "downloadfile.php", // Url to which the request is send
    type: "POST",            // Type of request to be send, called as method
    data: new FormData(this), // Data sent to server, a set of key/value pairs (i.e. form fields and
    values)
```

```
        contentType: false,       // The content stype used when sending data to the server.
        cache: false,             // To unable request pages to be cached
        processData:false,        // To send DOMDocument or non processed data file it is set to false
        success: function(data)   // A function to be called if request succeeds
        {
                $('#loading').hide();
                if(data == '0')
                {
                        alert('Invalid Pattern - Graphical Password Combination');
                        $('#downloadseminar')[0].reset();
                        imgcoordinates=[];

                        $("#dpattern2 #password ul").empty();
                        $('.password_patterns').hide();
                }
                else
                {

                        $('#downloadhref').html(data);
                        $('#downloadseminar')[0].reset();
                        $('.password_patterns').hide();
                }
        }
        });
}));
$('#download_imagepass').change(function(){
                $('.password_patterns').hide();
                var select_patt=$(this).val();
                $('#downloadhref').html('');
                $('#d'+select_patt).show();
                if(select_patt == 'pattern3')
                {
                        var x = 10;
                        var y = 5;
                        var t = '<table cellspacing="0" border="1" cellpadding="0" class="grxd">';
                        for (var i = 1; i <= (x * y); i++) {
                        t += (i == 1 ? '<tr>' : '');
                        t += '<td style="cursor:pointer;"></td>';
                if (i == (x * y)) {
        t += '</tr>';
                } else {
        t += (i % 10 === 0 ? '</tr><tr>' : '');
                        }
                        }
                        t += '</table>';
                        $("#downdrawTable").html(t);
                        $(".grxd  td").click(function(){
                        var col = $(this).css('background-color');
                        var column = $(this).parent().children().index(this);
                var row = $(this).parent().parent().children().index(this.parentNode);
                        var cellclicked="column-"+column+"row-"+row;
                        if(col == "rgba(0, 0, 0, 0)" || col == 'transparent')
                        {

                                        imgcoordinates.push("column-"+column+"row"+row);
                                        $( this ).css("background-color", "white");
                                        $('#down_pattern').val(imgcoordinates);
```

```
            }
            else
            {
                imgcoordinates=[];
                $('.grxd  td').css("background-color","transparent");
                /*$('.grxd  td').each(function(i){
                var col=$(this).css('background-color');
                if(col == 'rgba(0, 0, 0, 0)' || col == 'transparent' )
                {


                }
                else
                {
                    var column = $(this).parent().children().index(this);
                    var row =
                    $(this).parent().parent().children().index(this.parentNode);
                    imgcoordinates.push("column-"+column+"row"+row);
                    $('#down_pattern').val(imgcoordinates);
                }
                });*/

            }
            });
        }
        });
    $('#dimage_pattern1').Jcrop({
        maxSize:[20,20],
//onChange:   showCoords,
onSelect:   showCoord,
onRelease:  clearCoord
},function(){
jcrop_api = this;
});

// Simple event handler, called from onChange and onSelect
// event handlers, as per the Jcrop invocation above
function showCoord(c)
{

    imgcoordinates.splice(0,2,c.x,c.y);
    imgcoordinates.splice(2,4,c.x2,c.y2);
    imgcoordinates.splice(4,6,c.w,c.h);
    imgcoordinates.splice(6,8,'0','0');

    $('#down_pattern').val(imgcoordinates);

};

function clearCoord()
{
$('#coords input').val('');
};

        /**************Pattern 2**********/
    $("#dpattern2 #pass img").on("click",function(){
if($('#dpattern2 #password ul').children().length == 8)
    {
```

```
                        alert('Numerical graphical password maximum length has been reached ');
        }
        else
        {
                var imga = $(this).attr('src');
                $("#dpattern2 #password ul").append('<li><img src="' + imga + '" /><span
                class=close>&times;</span></li>');
                imgcoordinates.push(imga);

$("#dpattern2 #password ul li span").click(function(){

        $(this).parents('li').remove();
        imgcoordinates=[];
        $('#dpattern2 #password ul li img').each(function(i){
                var numi=$(this).attr('src');
                imgcoordinates.push(numi);
                });
        });
        imgcoordinates.toString();
$('#down_pattern').val(imgcoordinates);

}



});
```

CHAPTER 7

TESTING

Software testing is the process of checking whether the developed system is working according to the original objectives and requirements. Software testing process commences once the program is created and the documentation and related data structures are designed. Software testing is essential for correcting errors. Otherwise the project is not said to be complete.

The system should be tested experimentally with test data so as to ensure that the system works according to the required specification. When the system is found working, test it with actual data and check performance. Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding.

Need for Testing

Testing was essential for the following reasons:-

- Existence of program defects of inadequacies
- The software behavior as intended by its designer
- Conformance with requirement specification/user needs.
- Assess the operational reliability of the system.
- Reflect the frequency of actual user inputs.
- Find the fault, which caused the output anomaly.
- Checks for detect flaws and deficiencies in the requirements.
- Check whether the software is operationally useful.
- Exercise the program using data like the real data processed by the program.

7.1 System Testing

The primary goal of testing is to take the smallest piece of testable software in the application, isolate it from the remainder of the code, and determine whether it behaves exactly as you expect. Each unit is tested separately before integrating them into modules to test the interfaces between modules. Unit wise testing has proven its value in that a large percentage of defects are identified during its use.

7.1.1 Unit Testing:

The primary goal of unit testing is to take the smallest piece of testable software in the application, isolate it from the remainder of the code, and determine whether it behaves exactly as you expect. Each unit is tested separately before integrating them into modules to test the interfaces between modules. Unit testing has proven its value in that a large percentage of defects are identified during its use.

Unit testing is a software verification and validation method where the programmer gains confidence that individual units of source code are fit for use. A unit is the smallest testable part of an application. In procedural programming a unit may be an individual program, function, procedure, etc., while in object-oriented programming, the smallest unit is a class, which may belong to a base/super class, abstract class or derived/child class.Ideally, each test case is independent from the others: substitutes like method stubs, mock objects, fakes and test harnesses can be used to assist testing a module in isolation. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended. Its implementation can vary from being very manual (pencil and paper) to being formalized as part of build automation.

7.1.1 Integration Testing

Integration testing, also known as integration and testing (I&T), is a software development process which program units are combined and tested as groups in multiple ways. In this context, a unit is defined as the smallest testable part of an application. Integration testing can expose problems with the interfaces among program components before trouble occurs in real-world program execution. Integration testing is a component of Extreme Programming (XP), a pragmatic method of software development that takes a meticulous approach to building a product by means of continual testing and revision.

There are two major ways of carrying out an integration test, called the bottom-up method and the top-down method. Bottom-up integration testing begins with unit testing, followed by tests of progressively higher-level combinations of units called modules or builds. In top-down integration testing, the highest-level modules are tested first and progressively lower-level modules are tested after that. In a comprehensive software development environment, bottom-up testing is usually done first, followed by top-down testing.

7.1.3 Validation testing

At the validation level, testing focuses on user visible actions and user recognizable output from the system. Validations testing is said to be successful when software functions in a manner that can be reasonably expected by the customer.

7.2 Test Cases

SERIAL NUMBER	DESCRIPTION	INPUT	EXPECTED RESULT
1	Running the project	Enter url in the browser	A web application with home page
2	On Click Demo	Click the demo option	Demo page
3	Login popup	Submit without password field	Prompt enter password
4	Login popup	Submit invalid username or password	Prompt invalid username or password
5	Login popup	Submit without graphical password	Prompt select 3 images as your graphical password
6	Login popup	Submit invalid username	Prompt username doesnot appear in database
7	Login popup	Submit invalid username, password and graphical password	Prompt invalid username and password

SERIAL NUMBER	DESCRIPTION	INPUT	EXPECTED RESULT
8	Register	Click signup button	Prompt register popup opens
9	Register	Submit without username	Prompt enter username
11	Register	Submit without password	Prompt enter password
12	Register	Submit without graphical password	Prompt select any three image as your graphical password
13	Register	Submit with duplicate username	Prompt username already present
14	Register	Submit valid username, password and graphical password	Prompt register successful
15	Uploading file	Click upload	Prompt upload popup window opens
16	Uploading file	Submit without file	Prompt select a file to upload
17	Uploading file	Submit without choosing graphical pattern	Prompt select a pattern and create a password
18	Uploading file	Submit invalid file type	Prompt select only .xls, .doc, .ppt, file types only
19	Uploading file	Same file name	Prompt files already present
20	Uploading file	Submit valid file and password	Prompt file upload

SERIAL NUMBER	DESCRIPTION	INPUT	EXPECTED RESULT
21	Download file	Click on any filename after successful login	Prompt download popup opens
22	Download file	Submit with created a graphical password	Prompt your chosen pattern and enter your chosen password
23	Download file	Submit valid file and valid graphical password	Prompt file download Link is generated

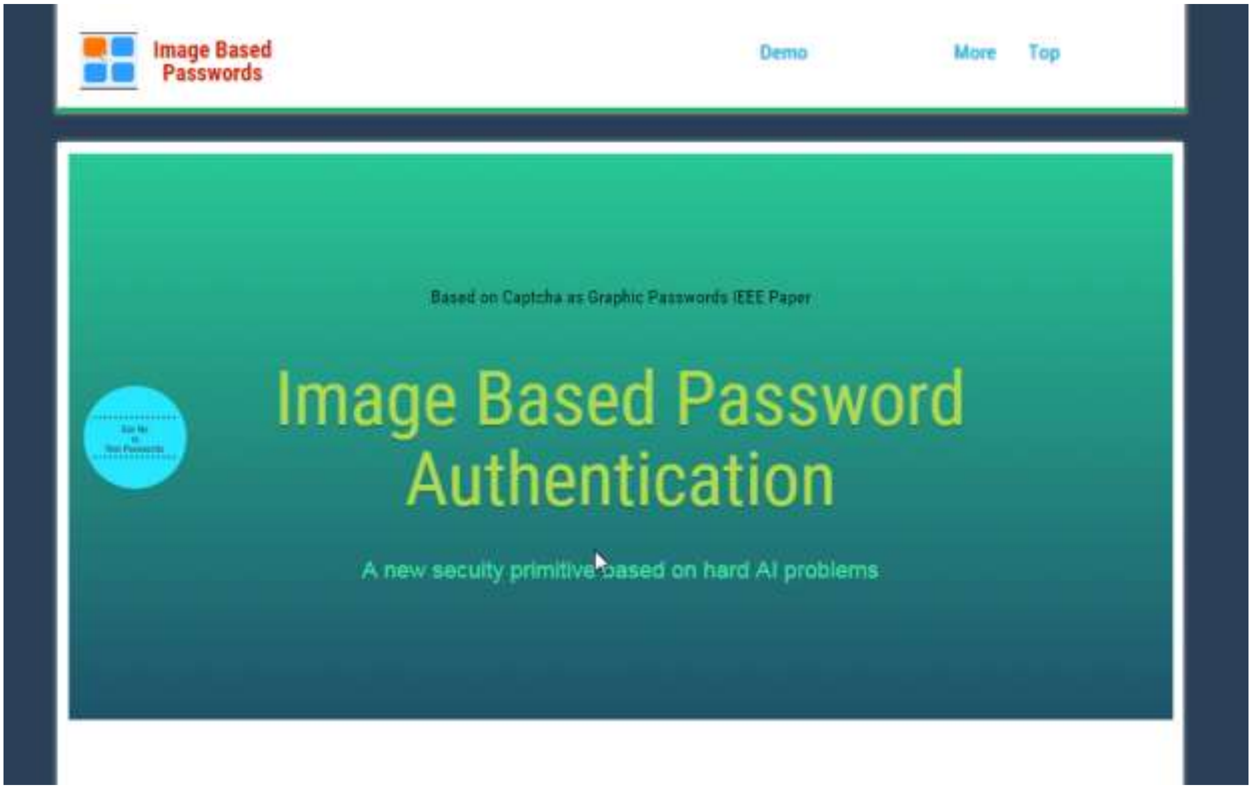
Table 7.1 – Test Case Description

CHAPTER 8

SNAPSHOTS

8.1 Front Page

Front page the user has to see options in the menu button.



Based on Captcha as Graphic Passwords IEEE Paper

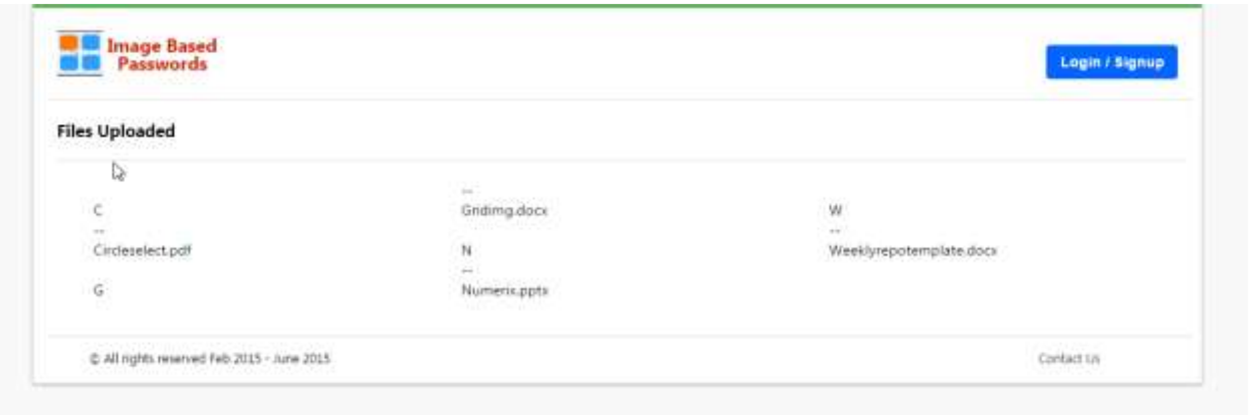
Image Based Password Authentication

A new secuity primitive based on hard AI problems

Fig 8.1 Front page module

8.2 Demo

The user is to enter the login details to download or upload the file.



Files Uploaded

C
--
Circleselect.pdf

G

--
Gridimg.docx

N
--
Numerix.pptx

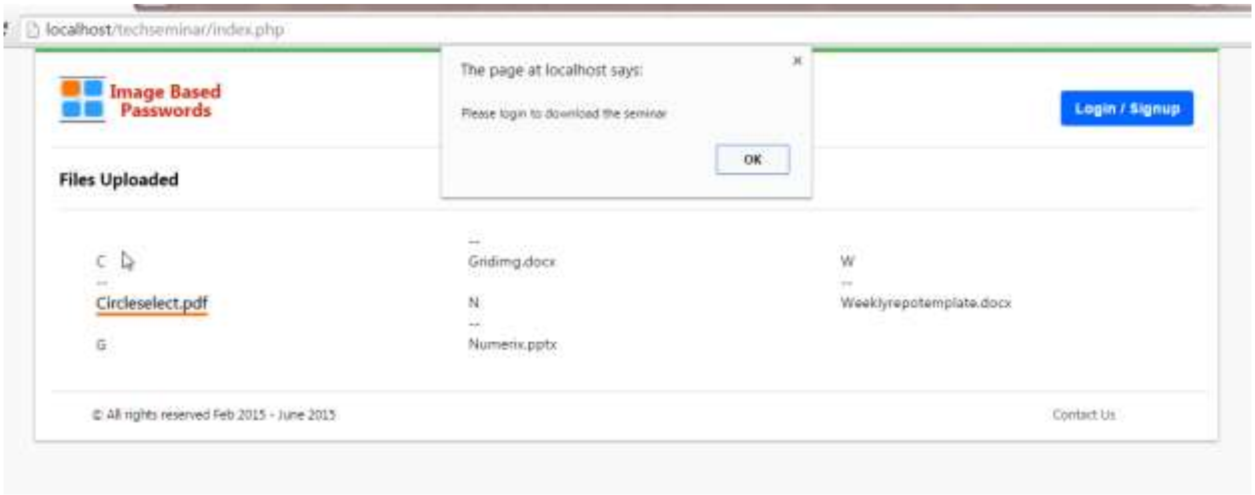
W
--
Weeklyrepotemplate.docx

© All rights reserved Feb 2015 - June 2015

Fig 8.2 Demo Front Page Module

Without Login Details

The user may not enter the login details in such case appropriate prompt will appear to login the page.



The page at localhost says:

Please login to download the seminar

Files Uploaded

C
--
Circleselect.pdf

G

--
Gridimg.docx

N
--
Numerix.pptx

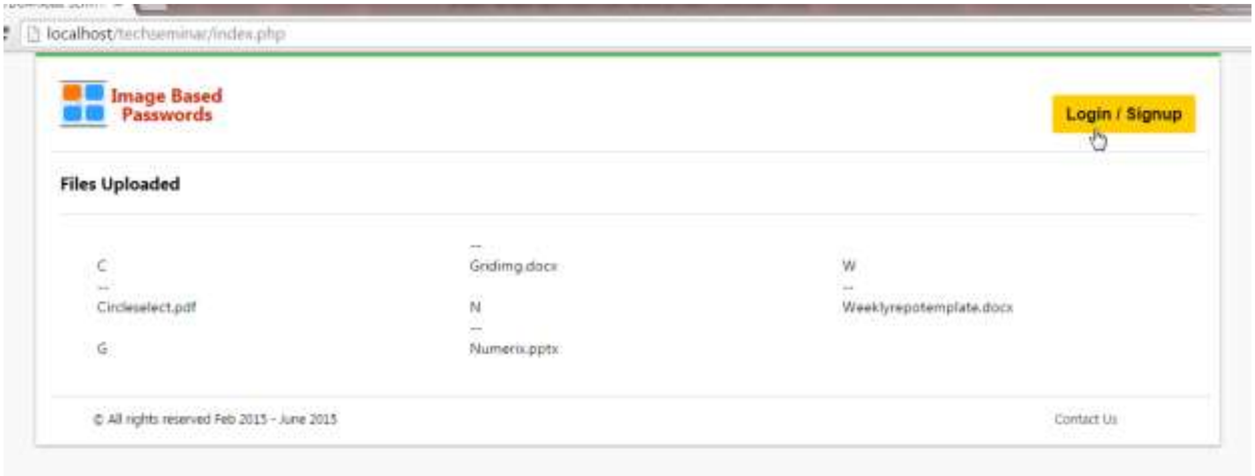
W
--
Weeklyrepotemplate.docx

© All rights reserved Feb 2015 - June 2015

Fig 8.3 User tries to login without login details

Login/Signup click

The user has to click the signup/login button.



localhost/techseminar/index.php

Files Uploaded

C
--
Circleselect.pdf

G
--
Gridimg.docx

N
--
Numerix.pptx

W
--
Weeklyrepotemplate.docx

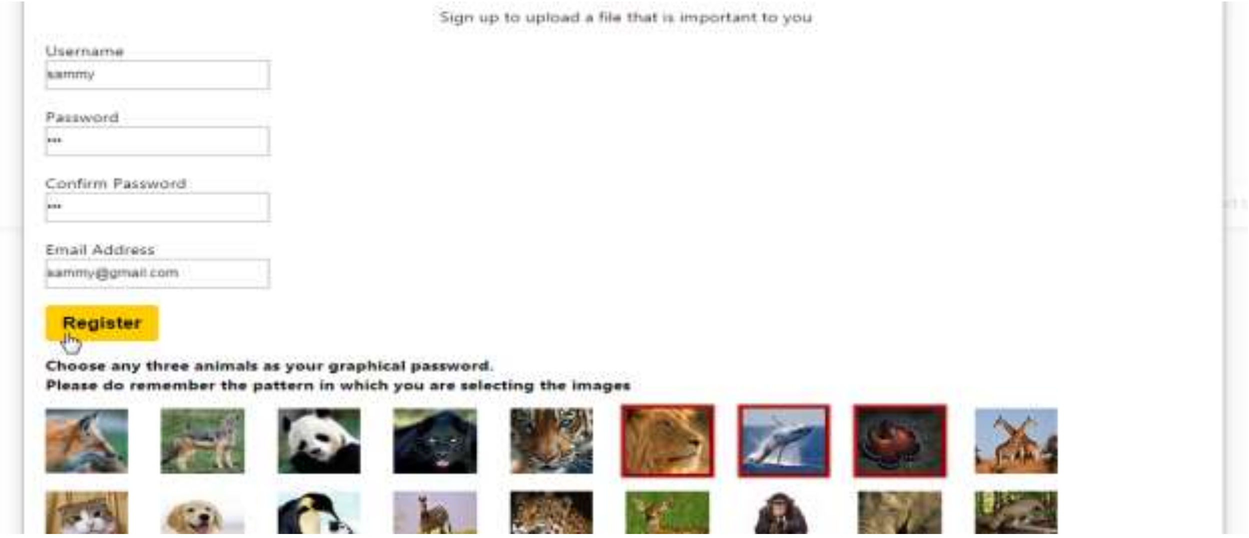
© All rights reserved Feb 2015 - June 2015

Contact Us

Fig 8.4 Click the Login/Signup

Signup Page

Signup page to enter the username and password



Sign up to upload a file that is important to you

Username: sammy

Password: ***

Confirm Password: ***

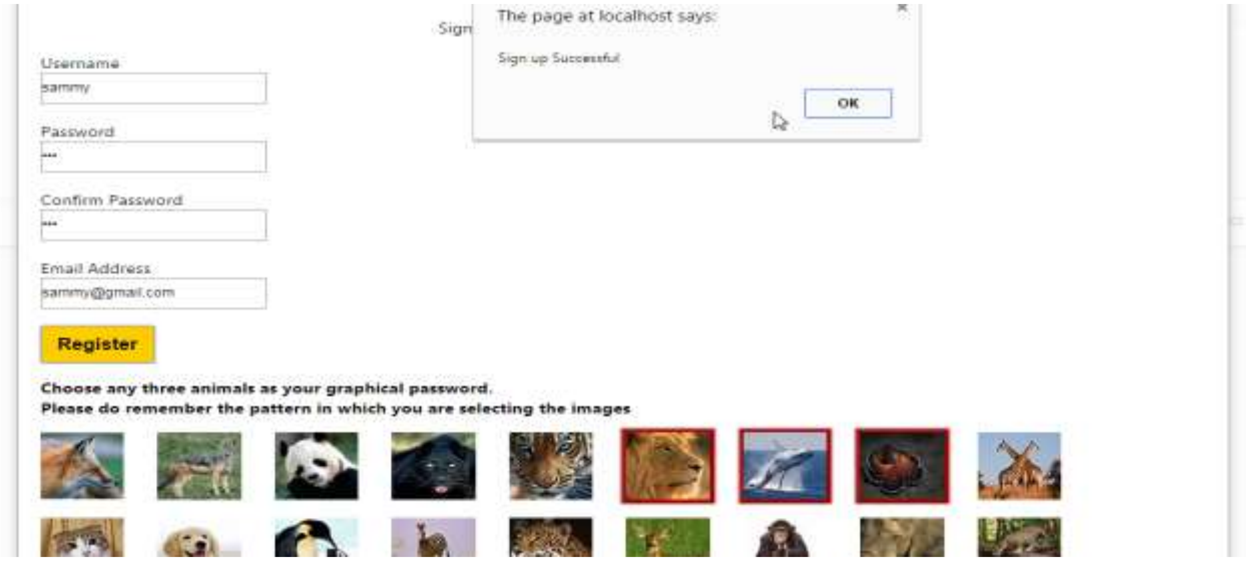
Email Address: sammy@gmail.com

Choose any three animals as your graphical password.
Please do remember the pattern in which you are selecting the images

Fig 8.5 Signup Page Module

Signup Successful

After entering the username, password and selecting 3 images prompt message appear as login is successful.



The page at localhost says:

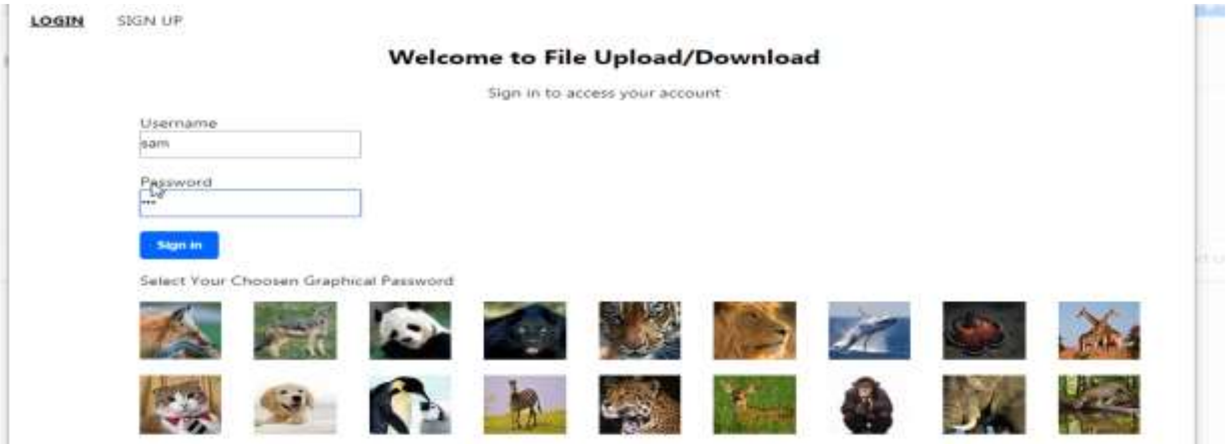
Sign up Successful

Choose any three animals as your graphical password.
Please do remember the pattern in which you are selecting the images

Fig 8.6 Signup Successful Module

Login page

The user is existing user then go to login page to enter the login details.



Welcome to File Upload/Download

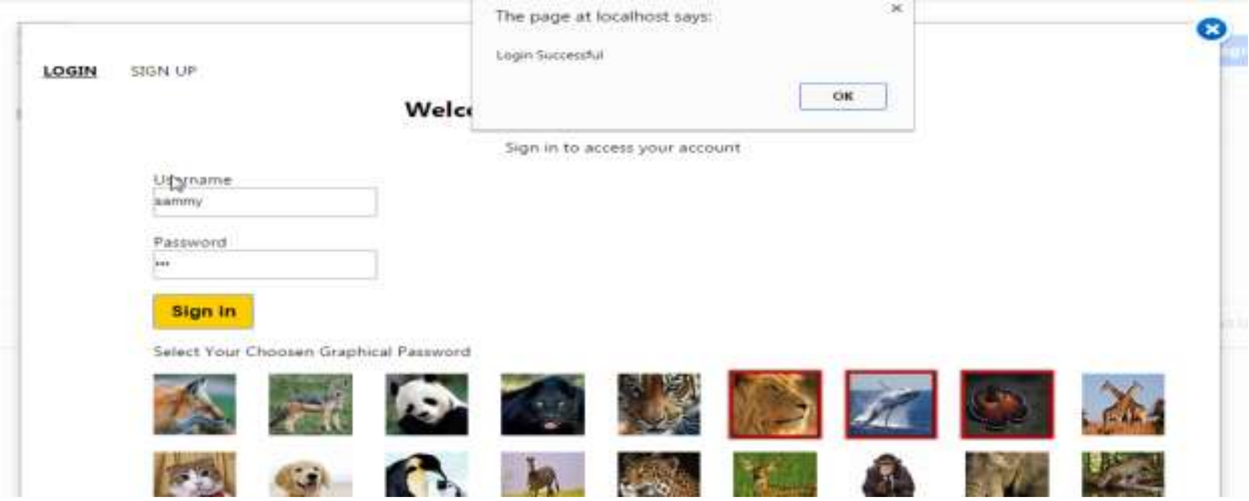
Sign in to access your account

Select Your Choosen Graphical Password

Fig 8.7 Login Page Module

Login successful Page

After entering correct login details prompt message will appear as login successful.



The page at localhost says:

Login Successful

Sign in to access your account

Username
sammy

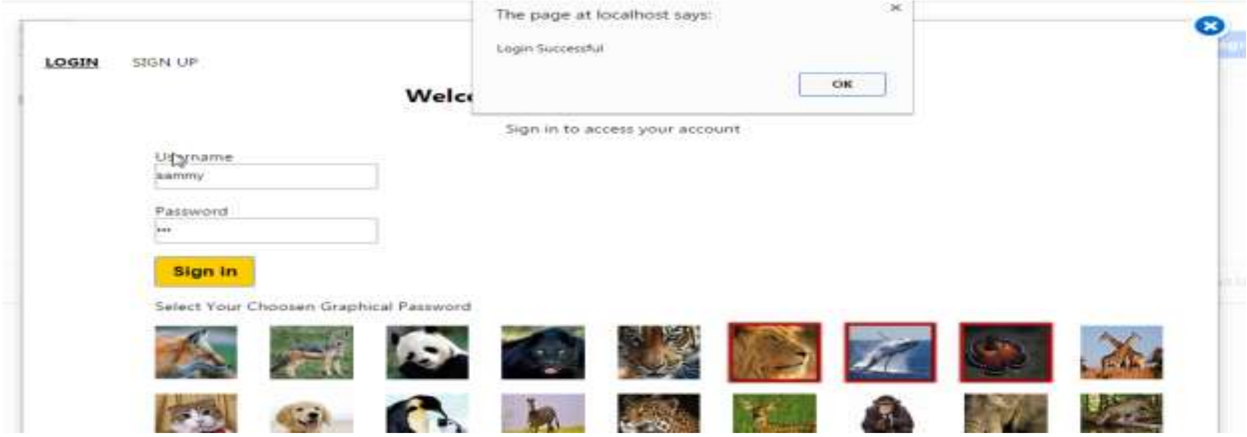
Password
•••

Select Your Choosen Graphical Password

Fig 8.8 Login Successful Module

File upload

Upload the file choosing correct pattern.



The page at localhost says:

Login Successful

Sign in to access your account

Username
sammy

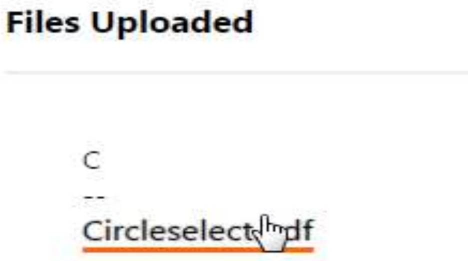
Password
•••

Select Your Choosen Graphical Password

Fig 8.9 File Upload Module

Choose the file

Select the file which one user has to upload.



Files Uploaded

C

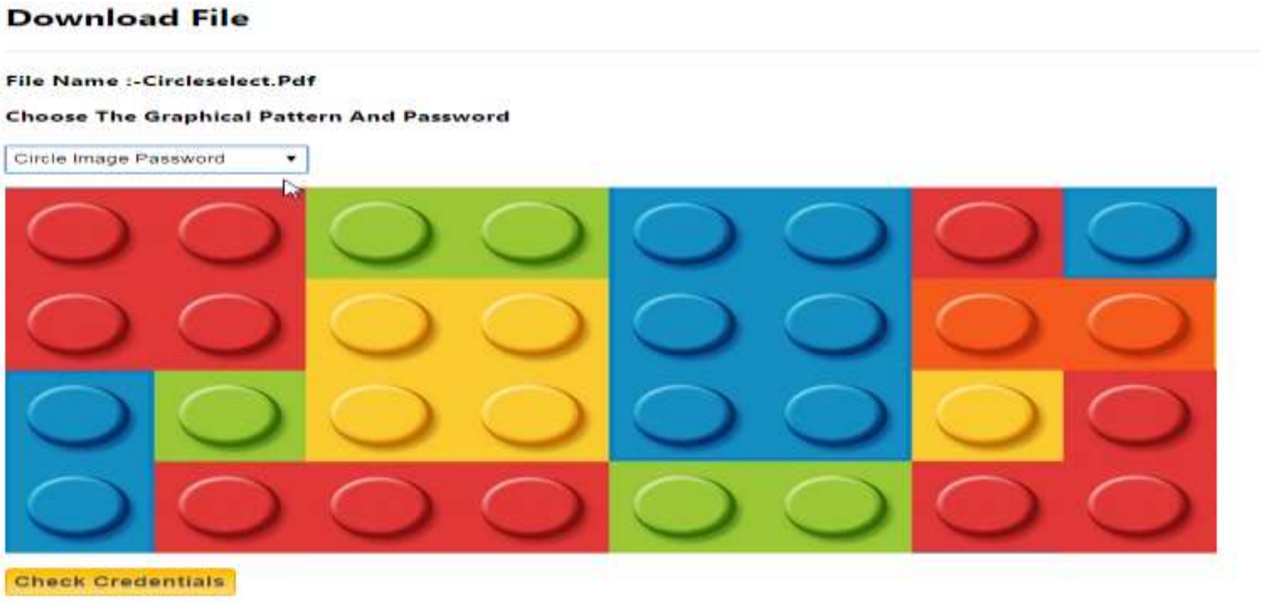
--

Circleselect.pdf

Fig 8.10 Choose the file

Select circle type pattern

Choose the circle type pattern to download the file



Download File

File Name :-Circleselect.Pdf

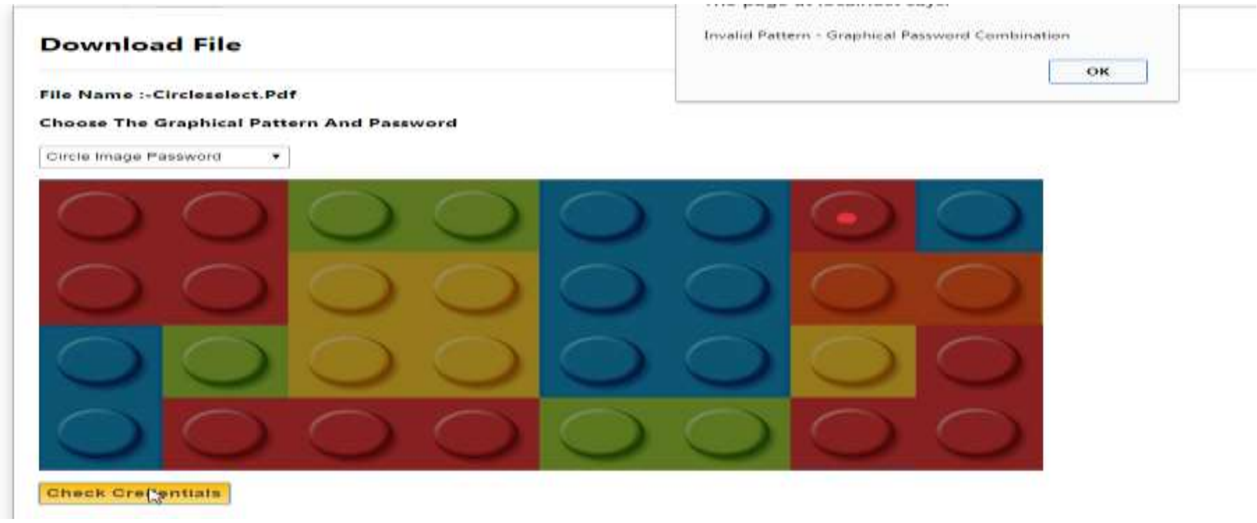
Choose The Graphical Pattern And Password

Circle Image Password

Fig 8.11 Circle type pattern

Choose the wrong pattern

Selecting the wrong pattern which is to download the file.



Download File

File Name :-Circleselect.Pdf

Choose The Graphical Pattern And Password

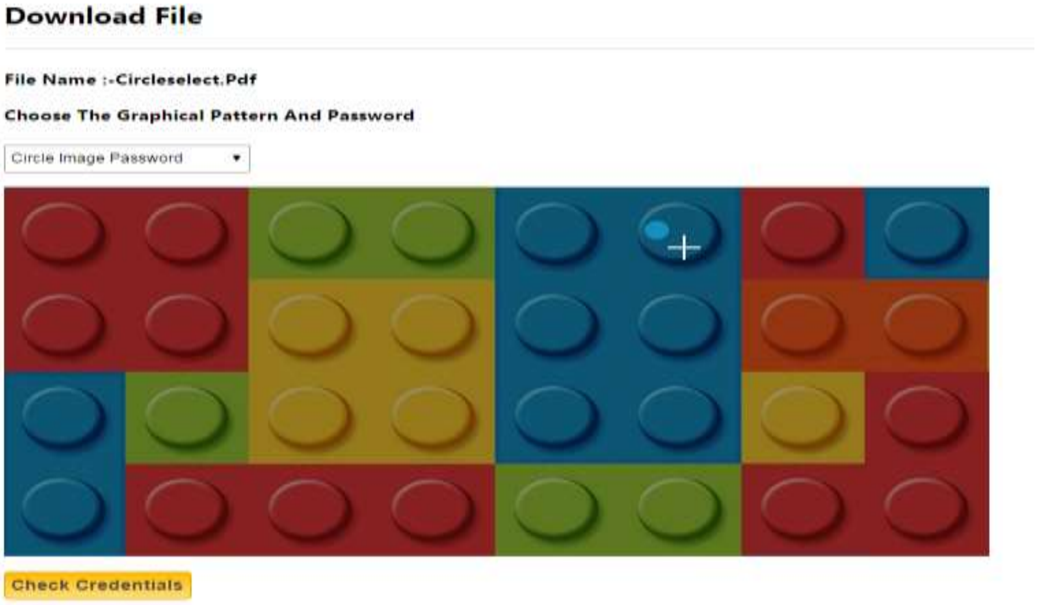
Circle Image Password

Invalid Pattern - Graphical Password Combination

Fig 8.12 Select the wrong Circle pattern

Circle pattern

Select the correct circle pattern to download the file.



Download File

File Name :-Circleselect.Pdf

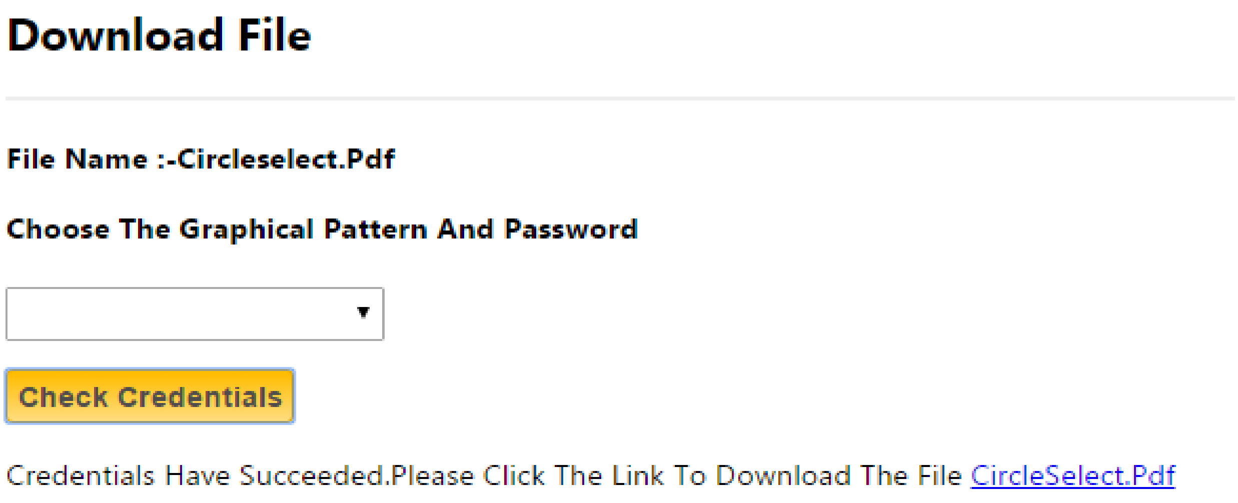
Choose The Graphical Pattern And Password

Circle Image Password

Fig 8.13 Select the correct circle pattern

Download the file

After selecting the correct pattern file is ready to download.



Download File

File Name :-Circleselect.Pdf

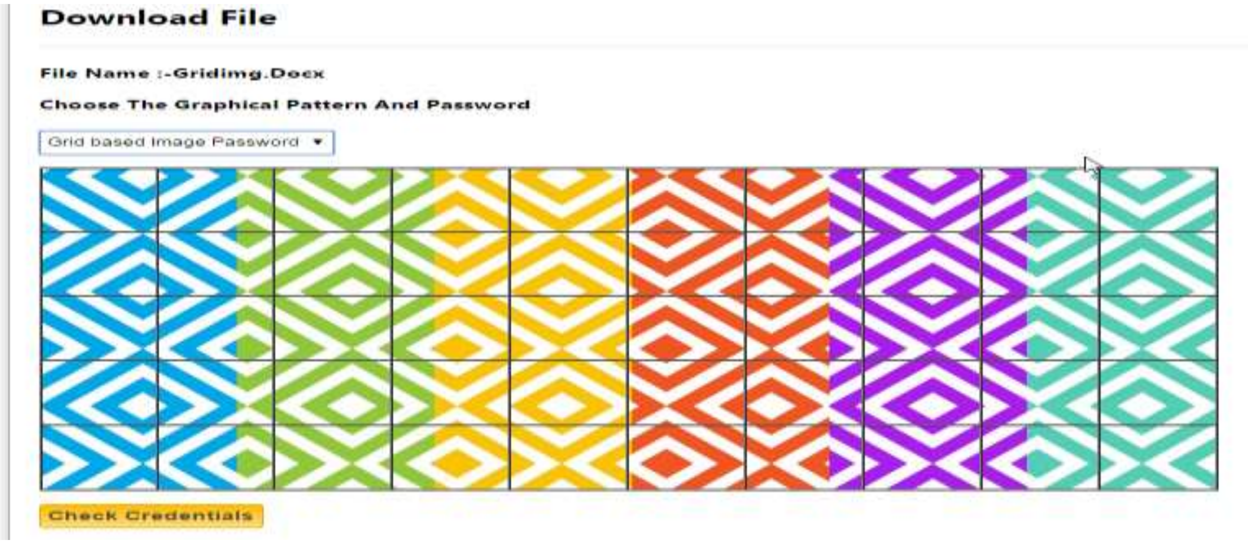
Choose The Graphical Pattern And Password

Credentials Have Succeeded.Please Click The Link To Download The File CircleSelect.Pdf

Fig 8.14 File link is ready for download

Grid Based Image password

Grid based password page will appear to select the grids.



Download File

File Name :-Gridimg.Docx

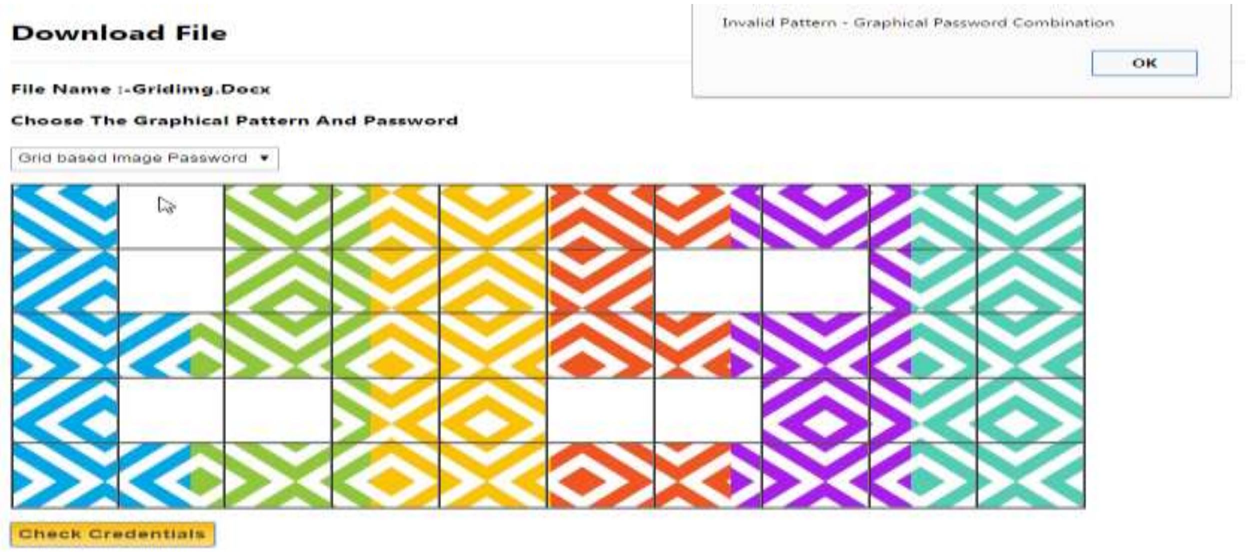
Choose The Graphical Pattern And Password

Grid based Image Password

Fig 8.15 Grid Based Image Password Module

Invalid grid chosen

User select the invalid grid or he select it not sequentially.



Invalid Pattern - Graphical Password Combination

Download File

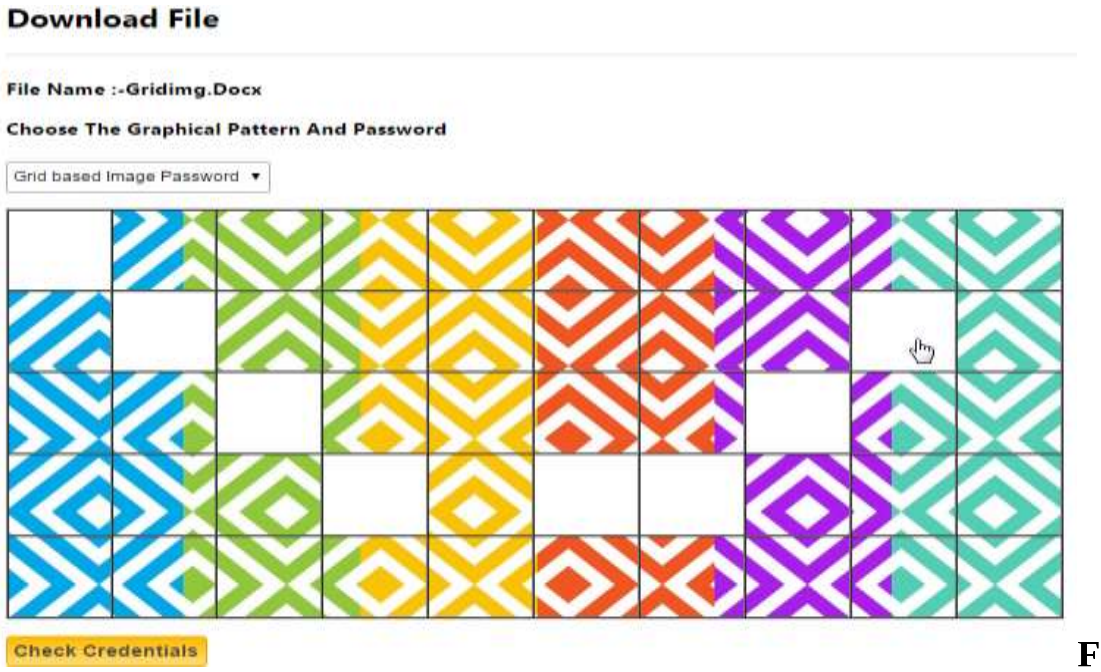
File Name :-Gridimg.Docx

Choose The Graphical Pattern And Password

Fig 8.16 Invalid grid chosen

Select the grid

Choose the 8 grid and set it as a password



Download File

File Name :-Gridimg.Docx

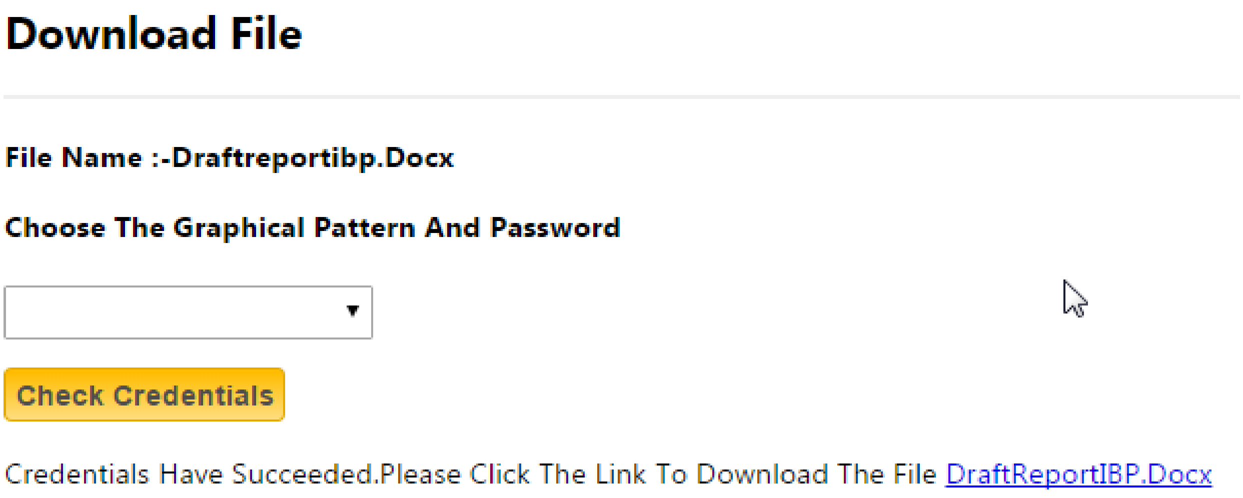
Choose The Graphical Pattern And Password

Fig 8.16

Select 8 grid as a password

Download File

After selecting the valid grid file is ready to download.



Download File

File Name :-Draftreportibp.Docx

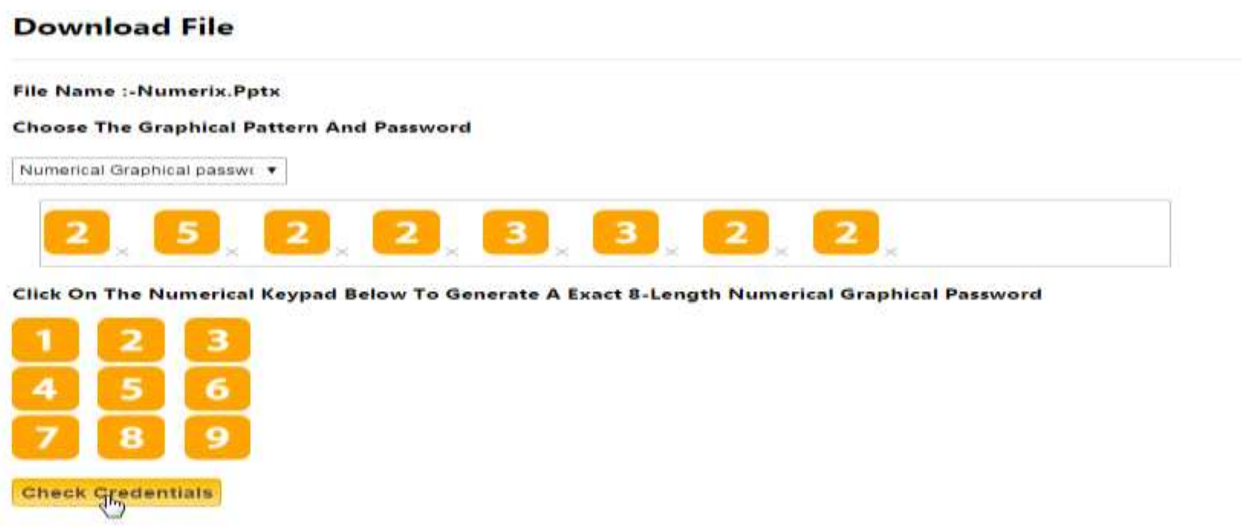
Choose The Graphical Pattern And Password

Credentials Have Succeeded.Please Click The Link To Download The File DraftReportIBP.Docx

Fig 8.18 File is ready to download

Numerical based password

Select the numbers as password



Download File

File Name :-Numerix.Pptx

Choose The Graphical Pattern And Password

Numerical Graphical passw ▾

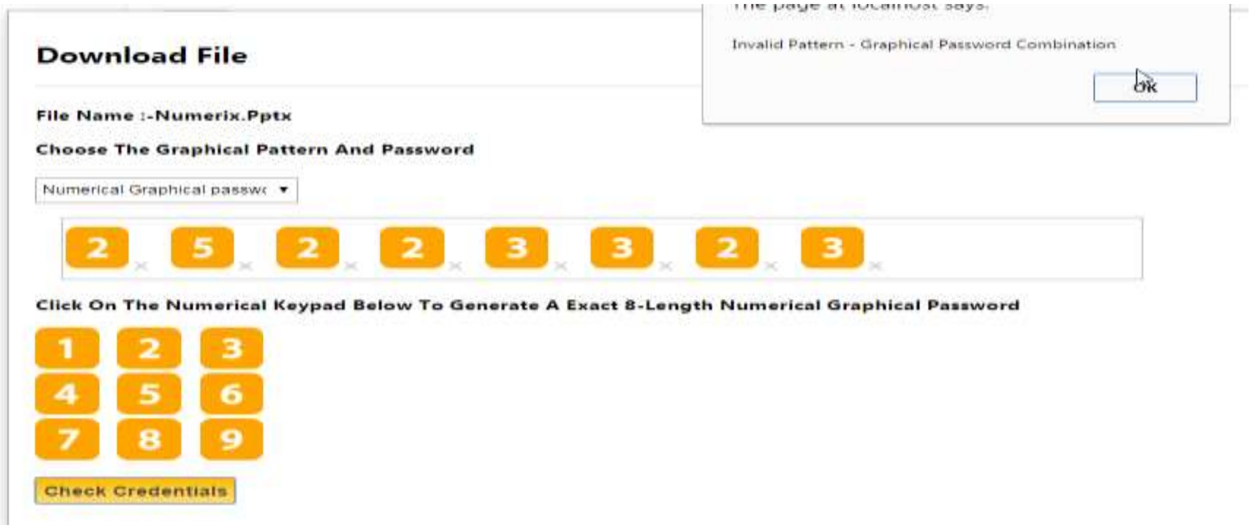
2 5 2 2 3 3 2 2

Click On The Numerical Keypad Below To Generate A Exact 8-Length Numerical Graphical Password

Fig 8.19 Numerical based password

Invalid numbers

User select the wrong numbers prompt message will appear as a wrong password.



The page at localhost says:

Invalid Pattern - Graphical Password Combination

Download File

File Name :-Numerix.Pptx

Choose The Graphical Pattern And Password

Numerical Graphical passw ▾

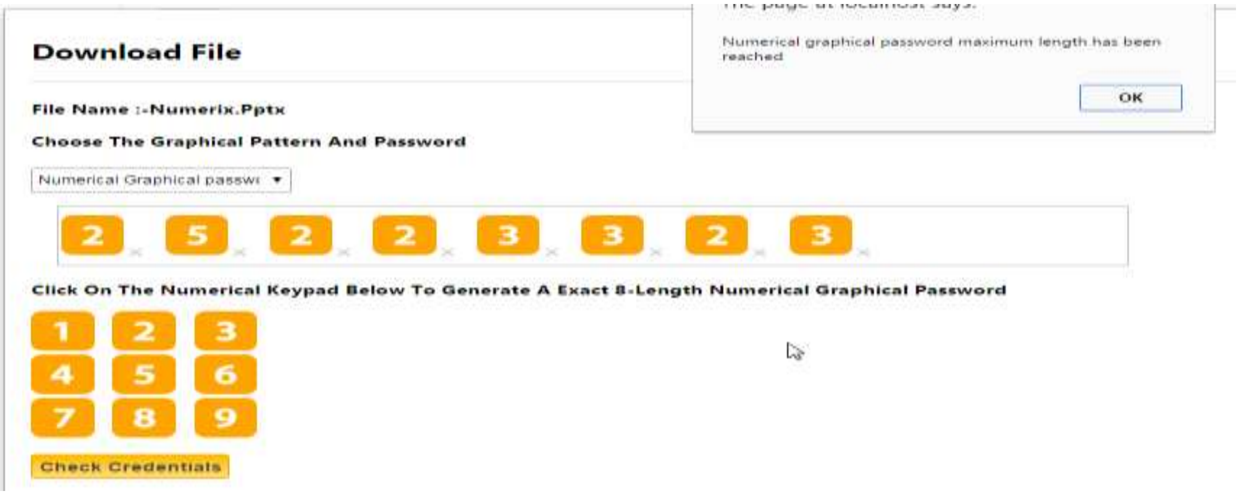
2 5 2 2 3 3 2 3

Click On The Numerical Keypad Below To Generate A Exact 8-Length Numerical Graphical Password

Fig 8.20 Invalid numerical password

Valid numerical password

Chosen the valid numerical password



Download File

File Name :-Numerix.Pptx

Choose The Graphical Pattern And Password

Numerical Graphical passw

2 5 2 2 3 3 2 3

Click On The Numerical Keypad Below To Generate A Exact 8-Length Numerical Graphical Password

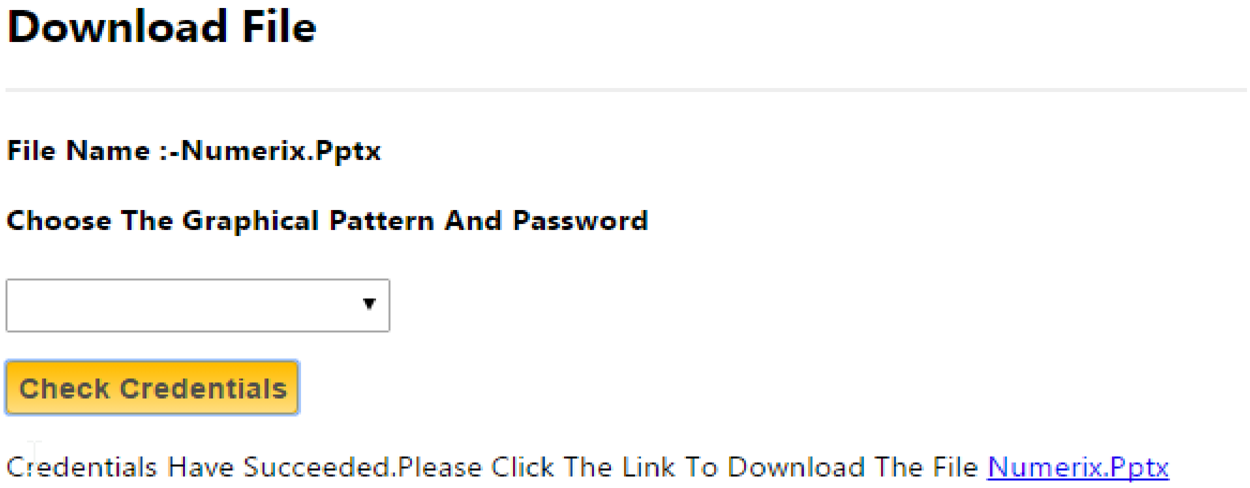
The page at localhost says:

Numerical graphical password maximum length has been reached

Fig 8.21 Valid numerical password

<u>Ready to download</u>

After selecting pattern file is ready to download



Download File

File Name :-Numerix.Pptx

Choose The Graphical Pattern And Password

Credentials Have Succeeded.Please Click The Link To Download The File Numerix.Pptx

Fig 8.22 File is ready to download

CONCLUSION AND FUTURE WORK

From the above project details, we can conclude that security is a difficult task. Today many security primitives are based on hard mathematical problems. Using hard AI problems for security is emerging as an exciting new paradigm, but has been underexplored. IBP addresses a number of security problems altogether, such as online guessing attacks, relay attacks, and, if combined with dual-view technologies, shoulder-surfing attacks. IBP is not a panacea, but it offers reasonable security and usability and appears to fit well with some practical applications for improving online security. Existing System Security primitives are based on hard mathematical problems. A fundamental task in security is to create cryptographic primitives based on hard mathematical problems that are computationally intractable. Using hard AI problems for security is new approach and will soon be widely accepted and used. Like Captcha, IBP utilizes unsolved AI problems. However, a password is much more valuable to attackers than a free email account that Captcha is typically used to protect. Therefore there are more incentives for attackers to hack IBP than Captcha. That is, more efforts will be attracted to the following win-win game by IBP than ordinary Captcha: If attackers succeed, they contribute to improving AI by providing solutions to open problems such as segmenting 2D texts. Otherwise, our system stays secure, contributing to practical security. As a framework, IBP does not rely on any specific Captcha scheme. When one Captcha scheme is broken, a new and more secure one may appear and be converted to an IBP scheme.

BIBLIOGRAPHY

- R. Biddle, S. Chaisson, and P. C. van Borscht, “Graphical passwords: Learning from the first twelve years,” ACM Computer Surveys, vol. 44, no. 4, 2012.
- The Science Behind Passfaces [Online]. Available: http://www.realuser.com/published/ScienceBehindPassfaces.pdf
- I. Jermyn, A. Mayer, F. Monrose, M. Reiter, and A. Rubin, “The design and analysis of graphical passwords,” in Proc. 8th USENIX Security Symp., 1999, pp. 1–15.
- H. Tao and C. Adams, “Pass-Go: A proposal to improve the usability of graphical passwords,” Int. J. Netw. Security, vol. 7, no. 2, pp. 273–292, 2008. S. Wiedenbeck, J. Waters, J. C. Birget, A. Brodskiy, and N. Memon.
- “PassPoints: Design and longitudinal evaluation of a graphical password system,” Int. J. HCI, vol. 63, pp. 102–127, Jul. 2005. ZHU et al.: NEW SECURITY PRIMITIVE BASED ON HARD AI PROBLEMS 903
- P. C. van Oorschot and J. Thorpe, “On predictive models and userdrawn graphical passwords,” ACM Trans. Inf. Syst. Security, vol. 10, no. 4, pp. 1–33, 2008.
- K. Golofit, “Click passwords under investigation,” in Proc. ESORICS, 2007, pp. 343–358.