A Project/Dissertation Report on Graphical Password Authentication

Submitted in partial fulfillment of the

requirement for the award of the degree of

B.Tech Computer Science Specialization in Cyber Security

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervisionof

Dr. N Partheeban

Project guide

Submitted By:

Akash Uniyal 19SCSE1140031

Aman Singh Negi 19SCSE1140032

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA
NOVEMBER, 2021

Abstract

Considering the traditional username-password authentication, the alphanumeric passwords are either easy to guess or difficult to remember. Also, users generally keep the same passwords for all their accounts because it is difficult to remember a lot of them. Alternative authentication methods, such as biometrics, graphical passwords are used to overcome these problems associated with the traditional username-password authentication technique.

In a graphical password authentication system, the user has to select from images, in a specific order, presented to them in a graphical user interface (GUI). According to a study, the human brain has a greater capability of remembering what they see(pictures) rather than alphanumeric characters. Therefore, graphical passwords overcome the disadvantage of alphanumeric passwords.

Graphical passwords provide a promising alternative to traditional alpha numeric passwords. They are attractive since people usually remember pictures better than words. In this extended abstract, we propose a simple graphical password authentication system. We describe its operation with some examples, and highlight important aspects of the system

List of Figures

Table of Contents

Introduction

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. Alphanumerical passwords are required to satisfy two contradictory requirements. They have to be easily remembered by a user, while they have to be hard to guess by impostor. Users are known to choose easily guessable and/or short text passwords, which are an easy target of dictionary and brute-forced attacks. Alphanumerical passwords are required to satisfy two contradictory requirements. They have to be easily remembered by a user, while they have to be hard to guess by impostor. Users are known to choose easily guessable and/or short text passwords, which are an easy target of dictionary and brute-forced attacks.

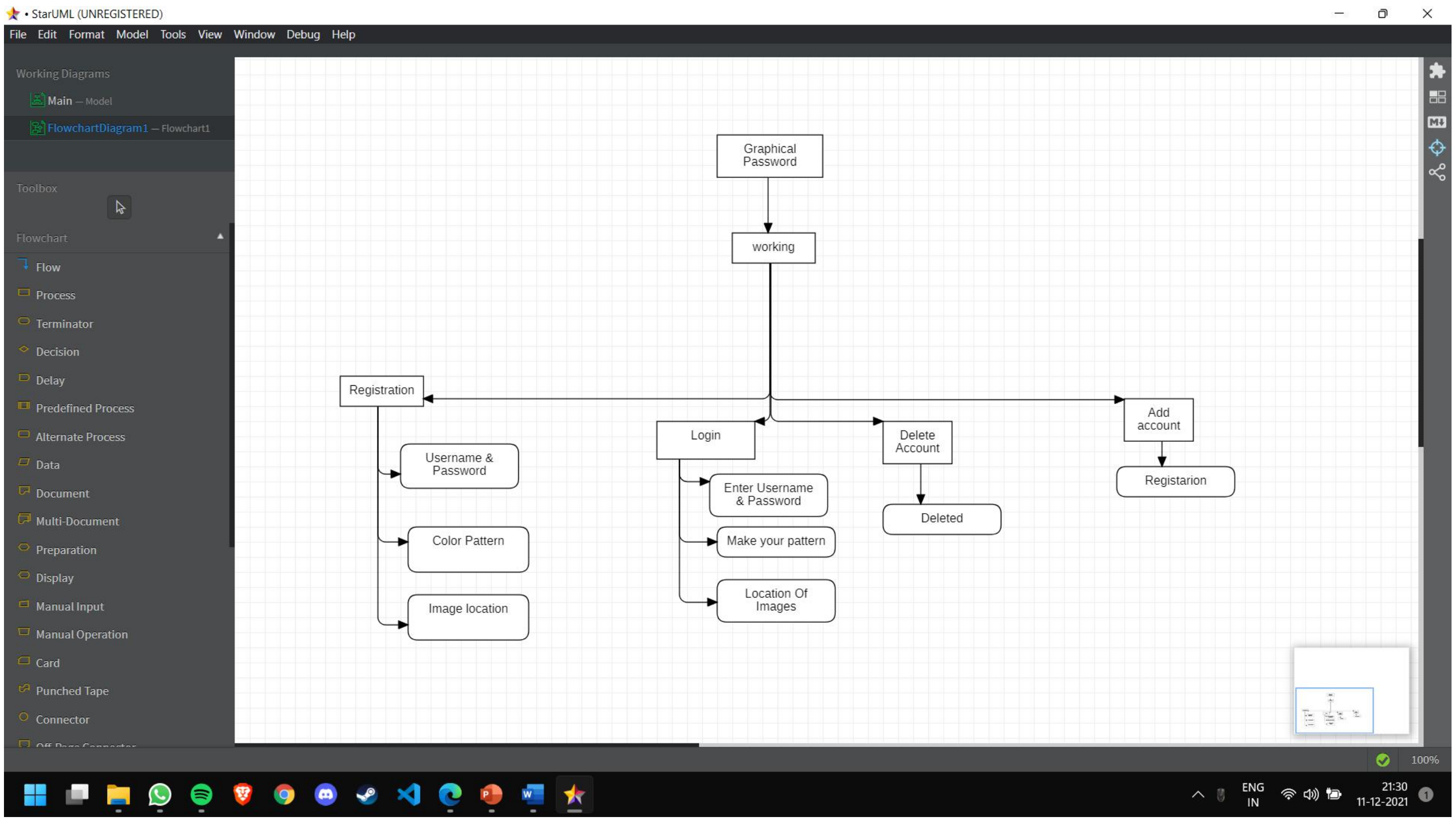
Enforcing a strong password policy sometimes leads to an opposite effect, as a user may resort to write his or her difficult-to-remember passwords on sticky notes exposing them to direct theft. In the literature, several techniques have been proposed to reduce the limitations of alphanumerical password. One proposed solution is to use an easy to remember long phrases (passphrase) rather than a single word. Another proposed solution is to use graphical passwords, in which graphics (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.

System Overview

A three-level authentication system that utilizes multiple forms of authentication. The first level is the conventional username and password authentication. The second level is a random pattern of red, green, and blue that the user specifies. The third level is a 4x4 grid where the user is presented with 4 images that can be dragged and dropped anywhere on the grid. When the user is logging in, they are once again presented with the same 4 images, placed in random locations within the grid. The user has to drag and drop the images to the same locations they chose during the registration stage. User information is hashed using the "crypto" Node.js module. We have tested the project on a computer that does not have Node.js installed and there were no errors. However, should the project not run on your computer, kindly install Node.js version 10 or later and try rerunning the project since the missing module might then be the problem. The bundle.js file located in /dist/js is bundled using Webpack module bundler. The unbundled source code is located in /src/js. The CSS code is compiled using node-sass npm package. The source code is located in /src/sass. The project uses the localStorage property of the window object to store user information. A 'reset all users' button was added to the header of the UI which removes the information saved by the project in the local storage of the browser.

System Design

URL DIAGRAM



Registration

The users of the system have to firstly register with the application before going ahead and logging into it. The registration consists of firstly choosing the set of images that the user desires for setting the password, out of each those images the user selects the area in the image which least likely guessable. For more security the user also gives a text password which will in turn be hidden in the set of images that the user had selected before this will be done by the concept of image steganography.

Image Steganography

Image steganography is performed during the registration and login process of the application. It is the process of hiding a data within another. If any type of data such as image, text etc. are hidden within an image, it is known as image steganography. During registration process, a textual password is asked from the user, which is then stored within the images.

While the login process, the user is asked to re-enter the password, which will then be compared with the one retrieved from the image. If the retrieved password matches with the one stored during the login process, the user is considered authenticated.

Login Process

During logging in the user is asked to type the text password which is matched to the text which will be retrieved from the images which was stored during the registration. Along with the text, the colors that the user had selected during the registration will be displayed out of which the user will have to click on the same colors that were clicked before.

Graphical Password Authentication

Personality traits are often regarded as a weak link in a computer security system. If we point out that they exist three large areas where there is human communication with computers important: authentication, security performance, and to develop secure systems. Here we focus on the verification problem. User proving authenticity is an important factor for many computer security conditions. Research has shown that from the user they can only remember a limited number of passwords, themselves they usually write them down or they will use the same passwords different accounts. Dealing with problems with username verification-common password, alternative verification methods, such as biometrics, have been used. However, in this paper we will focus on another alternative: using image as passwords.

Image passwords refer to the use of images (again drawings) as passwords. In theory, such photo passwords easy to remember, because people remember pictures better than words . Also, they have to be very resistant to them a vicious attack, as the search site is real endless.

In general, image password strategies are categorized into two main categories: image processing based on recognition and memory-based .

Proposed System

These systems are usually in danger of slipping on the shoulder to the extent that in most cases, the whole drawing is the same visible on the screen as it is installed, and thus the attacker needs to view accurately or record only one login to unlock the entire password. Shoulder shaving is a bad aspect of the image password confirmation. To overcome what we have built an SSR shield (Shoulder Surfing Resistant). A shield containing many false mouse points
sorted in such a way that it goes random at the location of the image and the actual identifier will look exactly the same pointless mouse guides. This shield provides a superior layer grid clicks and confusing someone else.

Formulation of problem

The problems of knowledge-based authentication are extremely text-based passwords are well known. Users often needs to create memorable passwords that are easy for attackers to guess, but Strong system-assigned passwords are difficult for users to remember, a graphical password authentication system should encourage users with strong passwords as well as memorable. So they came through with new ideas like

->recognition based (pass faces)
->recall based (Das)
->persuasion using cued click points (pass points)

In the area of graphical passwords, Recognition based (pass faces) having high level of online guessing attacks. Recall based (draw a secret) shows high password strength but it needs very low level of attempts to crack the password. Cued click points is the latest technique which gives hot spot images(i.e.)highlighting the points to the attackers. This paper overcomes the above issue with following authentication ideas incorporate with graphical password techniques.

LANGUAGES USED IN THIS SOFTWARE:

- JavaScript
- CSS
- SCSS
- HTML
- Node.js V10

Languages

- **HTML**

The Hyper Text Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets and scripting languages such as JavaScript.

- **CSS**

Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

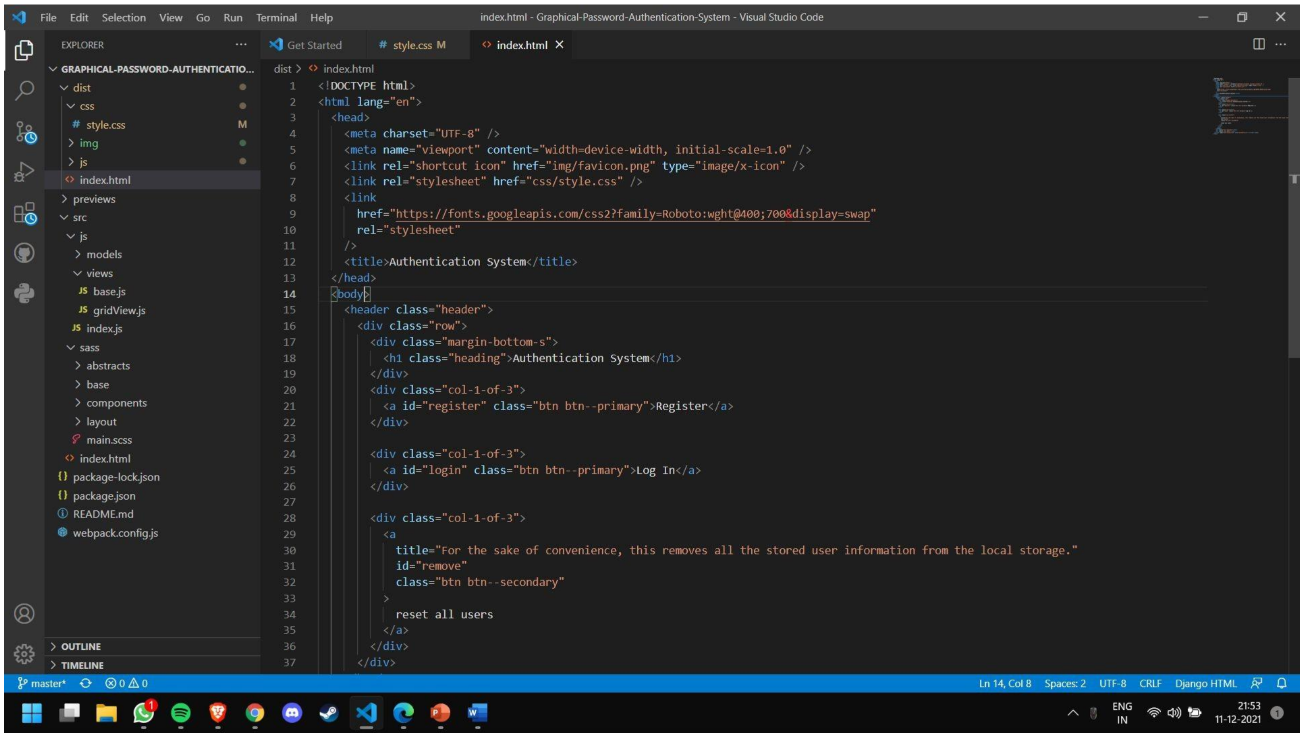
- **Java Script**

JavaScript, often abbreviated JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. Over 97% of websites use JavaScript on the client side for web page behavior, often incorporating third-party libraries.

- **Java Script in HTML**

JavaScript is a text-based programming language used both on the client-side and server-side that allows you to make web pages interactive. Where HTML and CSS are languages that give structure and style to web pages, JavaScript gives web pages interactive elements that engage a user.

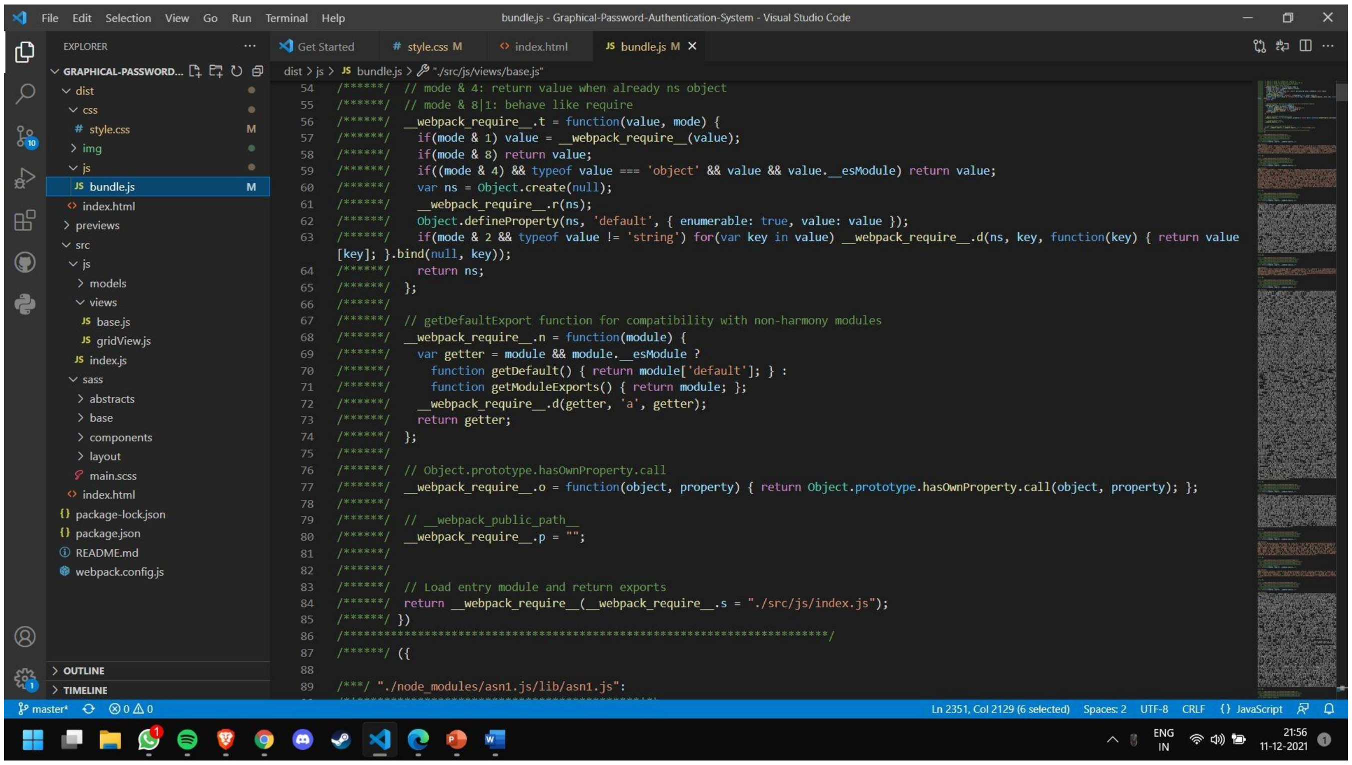
Implementation



```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link rel="shortcut icon" href="img/favicon.png" type="image/x-icon" />
    <link rel="stylesheet" href="css/style.css" />
    <link
      href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;700&display=swap"
      rel="stylesheet"
    />
    <title>Authentication System</title>
  </head>
  <body>
    <header class="header">
      <div class="row">
        <div class="margin-bottom-s">
          <h1 class="heading">Authentication System</h1>
        </div>
        <div class="col-1-of-3">
          <a id="register" class="btn btn--primary">Register</a>
        </div>

        <div class="col-1-of-3">
          <a id="login" class="btn btn--primary">Log In</a>
        </div>

        <div class="col-1-of-3">
          <a
            title="For the sake of convenience, this removes all the stored user information from the local storage."
            id="remove"
            class="btn btn--secondary"
          >
            reset all users
          </a>
        </div>
      </div>
```

HTML CODE

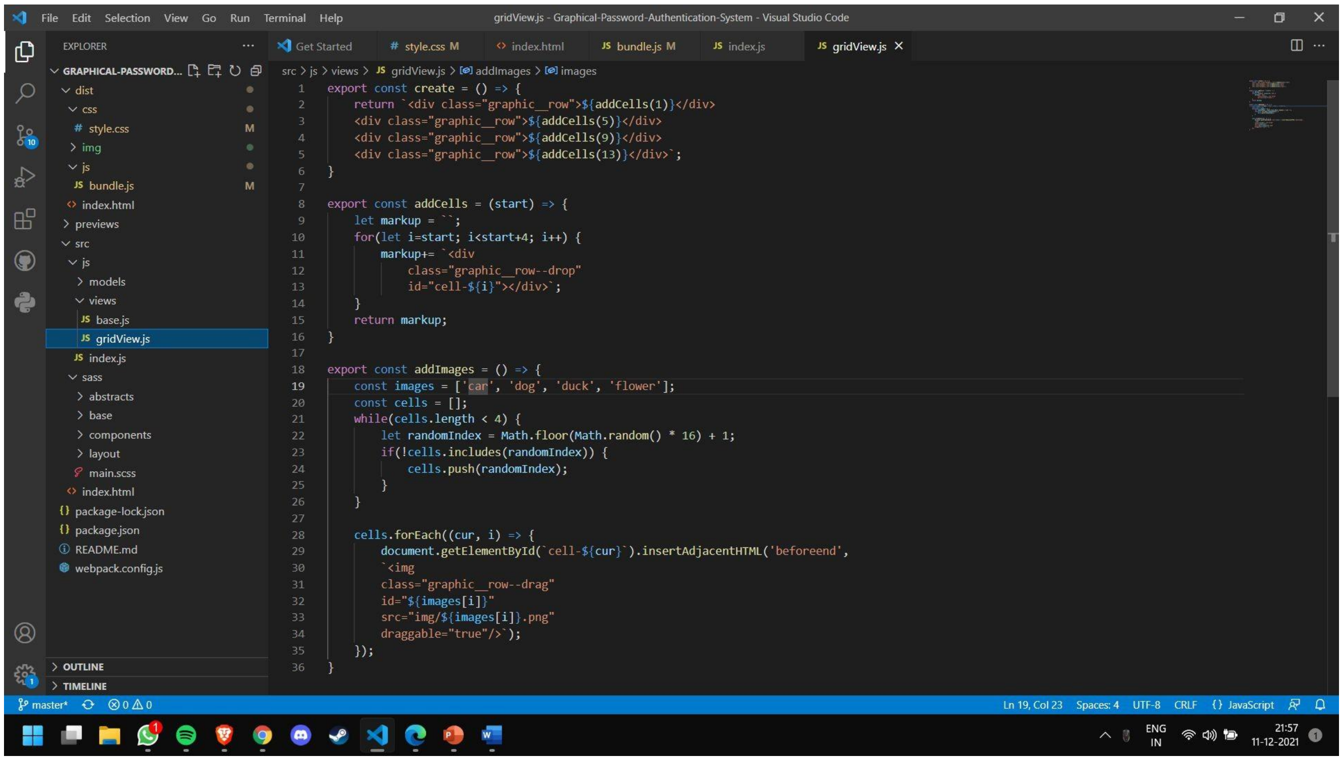
Implementation



```
/******/  // mode & 4: return value when already ns object
/******/  // mode & 8|1: behave like require
/******/  __webpack_require__.t = function(value, mode) {
/******/    if(mode & 1) value = __webpack_require__(value);
/******/    if(mode & 8) return value;
/******/    if((mode & 4) && typeof value === 'object' && value && value.__esModule) return value;
/******/    var ns = Object.create(null);
/******/    __webpack_require__.r(ns);
/******/    Object.defineProperty(ns, 'default', { enumerable: true, value: value });
/******/    if(mode & 2 && typeof value != 'string') for(var key in value) __webpack_require__.d(ns, key, function(key) { return value
[key]; }.bind(null, key));
/******/    return ns;
/******/  };
/******/
/******/  // getDefaultExport function for compatibility with non-harmony modules
/******/  __webpack_require__.n = function(module) {
/******/    var getter = module && module.__esModule ?
/******/      function getDefault() { return module['default']; } :
/******/      function getModuleExports() { return module; };
/******/    __webpack_require__.d(getter, 'a', getter);
/******/    return getter;
/******/  };
/******/
/******/  // Object.prototype.hasOwnProperty.call
/******/  __webpack_require__.o = function(object, property) { return Object.prototype.hasOwnProperty.call(object, property); };
/******/
/******/  // __webpack_public_path__
/******/  __webpack_require__.p = "";
/******/
/******/
/******/  // Load entry module and return exports
/******/  return __webpack_require__(__webpack_require__.s = "./src/js/index.js");
/******/ })
/************************************************************************/
/******/ ({

/***/ "./node_modules/asn1.js/lib/asn1.js":
```

Java Script Code

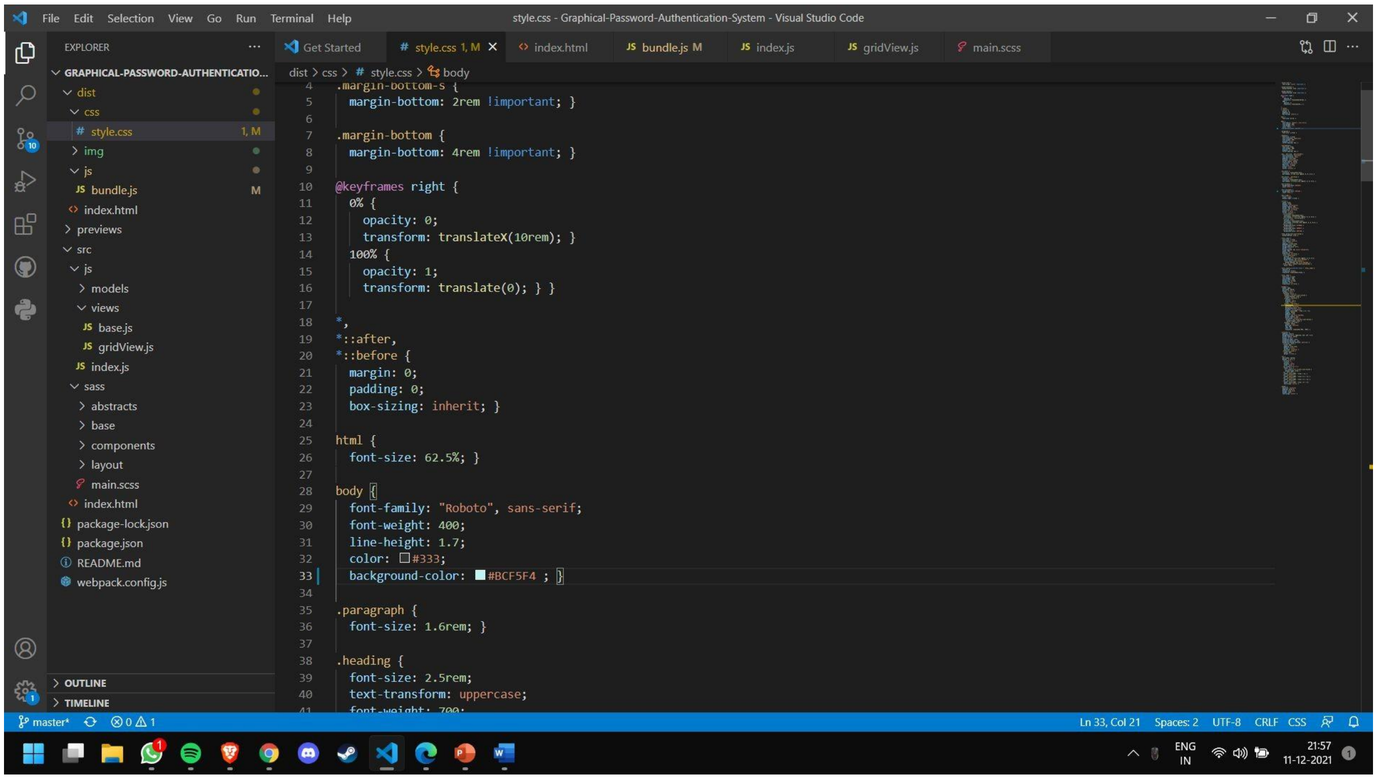
Implementation



```javascript
export const create = () => {
    return `<div class="graphic__row">${addCells(1)}</div>
    <div class="graphic__row">${addCells(5)}</div>
    <div class="graphic__row">${addCells(9)}</div>
    <div class="graphic__row">${addCells(13)}</div>`;
}

export const addCells = (start) => {
    let markup = ``;
    for(let i=start; i<start+4; i++) {
        markup+= `<div
            class="graphic__row--drop"
            id="cell-${i}"></div>`;
    }
    return markup;
}

export const addImages = () => {
    const images = ['car', 'dog', 'duck', 'flower'];
    const cells = [];
    while(cells.length < 4) {
        let randomIndex = Math.floor(Math.random() * 16) + 1;
        if(!cells.includes(randomIndex)) {
            cells.push(randomIndex);
        }
    }

    cells.forEach((cur, i) => {
        document.getElementById(`cell-${cur}`).insertAdjacentHTML('beforeend',
        `<img
        class="graphic__row--drag"
        id="${images[i]}"
        src="img/${images[i]}.png"
        draggable="true"/>`);
    });
}
```

Java Script

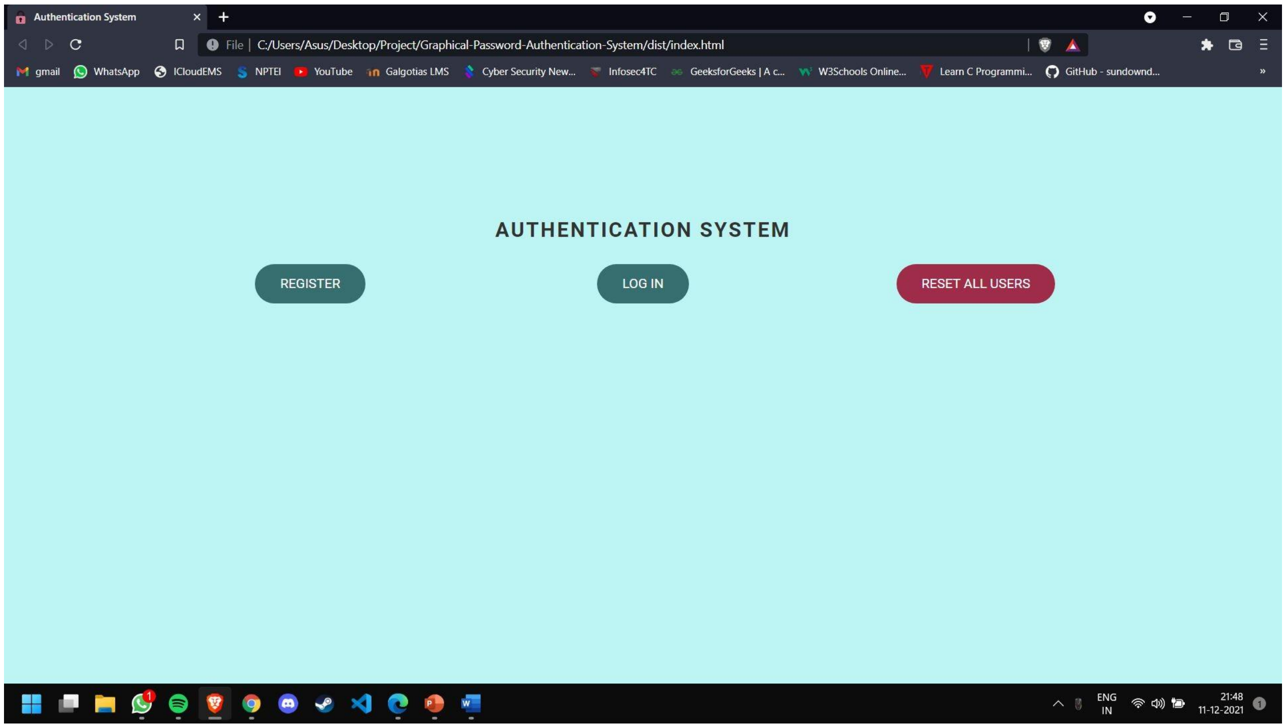
Implementation


style.css - Graphical-Password-Authentication-System - Visual Studio Code
EXPLORER
GRAPHICAL-PASSWORD-AUTHENTICATIO...
Get Started
style.css 1, M
index.html
bundle.js M
index.js
gridView.js
main.scss
dist > css > # style.css > body
master*
Ln 33, Col 21 Spaces: 2 UTF-8 CRLF CSS

```css
.margin-bottom-s {
  margin-bottom: 2rem !important; }

.margin-bottom {
  margin-bottom: 4rem !important; }

@keyframes right {
  0% {
    opacity: 0;
    transform: translateX(10rem); }
  100% {
    opacity: 1;
    transform: translate(0); } }

*,
*::after,
*::before {
  margin: 0;
  padding: 0;
  box-sizing: inherit; }

html {
  font-size: 62.5%; }

body {
  font-family: "Roboto", sans-serif;
  font-weight: 400;
  line-height: 1.7;
  color: #333;
  background-color: #BCF5F4 ; }

.paragraph {
  font-size: 1.6rem; }

.heading {
  font-size: 2.5rem;
  text-transform: uppercase;
  font-weight: 700;
```

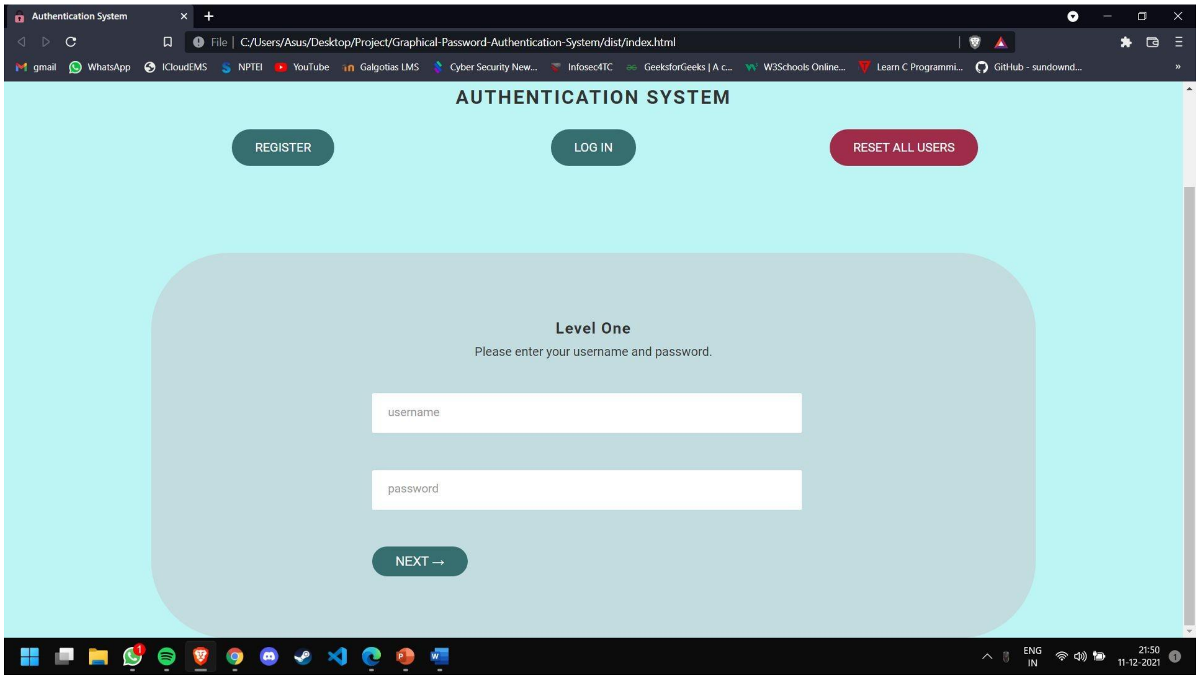
Style.CSS Code

Output Screen


Authentication System
C:/Users/Asus/Desktop/Project/Graphical-Password-Authentication-System/dist/index.html
AUTHENTICATION SYSTEM
REGISTER
LOG IN
RESET ALL USERS

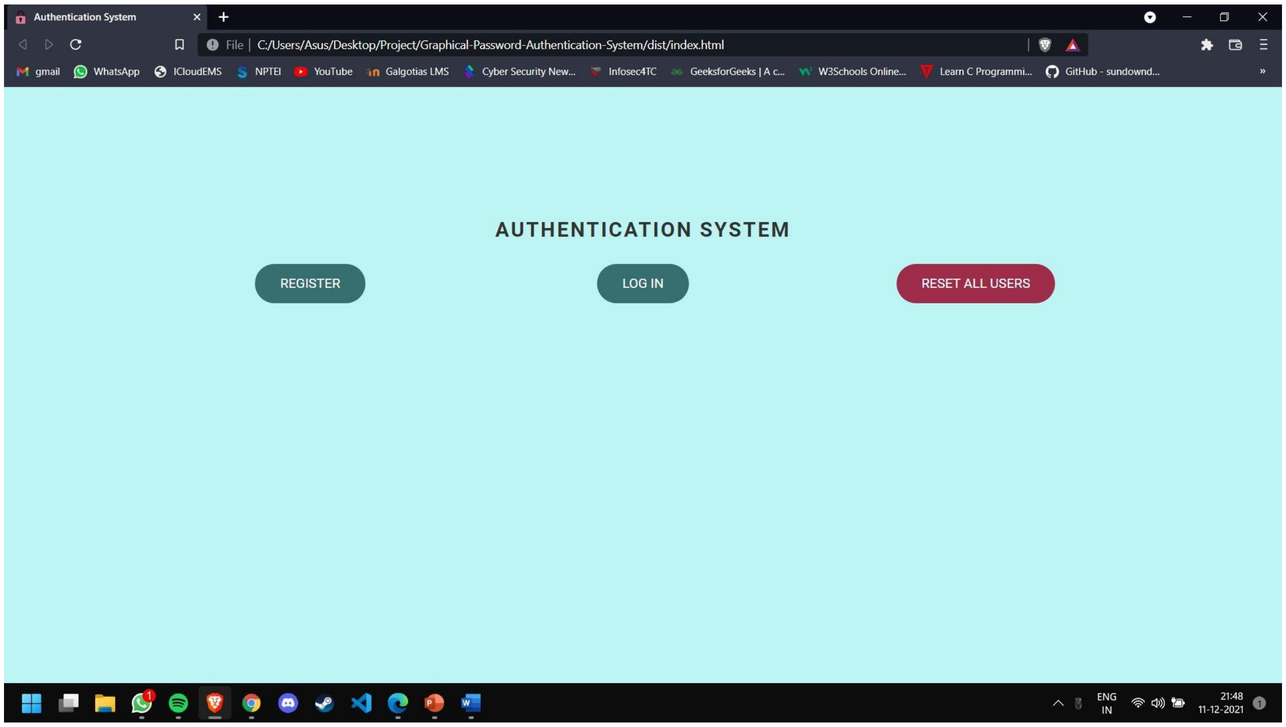
Login Page Screen

Output Screen


Authentication System
C:/Users/Asus/Desktop/Project/Graphical-Password-Authentication-System/dist/index.html
AUTHENTICATION SYSTEM
REGISTER
LOG IN
RESET ALL USERS
Level One
Please enter your username and password.
username
password
NEXT →

Registration Screen

Output Screen


Authentication System
File | C:/Users/Asus/Desktop/Project/Graphical-Password-Authentication-System/dist/index.html
AUTHENTICATION SYSTEM
REGISTER
LOG IN
RESET ALL USERS

eset Screen

Literature Survey

Wantong Zheng and Chunfu Jia proposed a method Combined PWD. This scheme proposes an online secret phrase verification component, combined PWD, through embedding separators (e.g., spaces) into the passwords to reinforce the current secret word validation framework. This plan uses the custom of the clients input. In this examination, site clients can embed spaces in their secret word where they need to stop when they register a record and the site back-end records the number of spaces in each hole.

In the paper, a novel time-based unique password was contributed to avoiding challenges of using a third party such as one- time password email, test and token device, the client will set an underlying secret word to characterize how the secret key will be changing throughout a characterized time, we tracked down that the framework. Then found that the system retains the strength of the dynamic password and improves the usability of the system in terms of availability.

A strong password authentication scheme was proposed by Yang Jingo. The one-time password authentication schemes can be divided into two types namely weak-password authentication schemes and strong-password
authentication schemes. In this paper, we survey the as of

1. Kus scheme and it also shows an attack against his protocol. And also found that strong passwords have higher strength and easily guessing is not possible. Later, we present a strong password authentication scheme. This paper expands W. C. Ku's plan so that the alteration convention can oppose Stolen-verifier assault. The changed convention is built without loss of effectiveness.

Here, we use a picture password for the second authentication. So, no need for complex textual passwords. Users can use any basic textual password. The system is classified into three modules.

Hua Wang, Yao Guo proposes another reuse- situated secret phase authentication system, called Desktop Password Authentication Centre (DPAC), to reuse counter-measures among applications, along these lines lessening the expense of protecting passwords against dangers. This arrangement can take out a ton of tedious work and reduces the expense essentially, we demonstrate the feasibility overpack by implementing a prototype, in which we migrate the widely used OpenSSH to DPAC and implement two example countermeasures. Password authentication code (PAC) is a very important issue in many applications such as web- sites and database systems etc. Salah Refish proposes a PAC-RMPN scheme. In this paper, PAC between two clients to affirm verification between them has been introduced. This research presents a novel solution to the era-long problem of password authentication at the incoming level. They should discover a strategy to secure this a secret word from anticipated attackers. A legitimate user types his password only and presses enter to propagate it to another user which he wants to be authenticated. A secure password authentication scheme is proposed which gives more security. This method uses a combination of pattern, key, and dummy digits. For this, the client needs to perceive and enlist design as area numbers from the network, register key qualities that guide esteem to secret password, and attach faker qualities to misguide the attacker. From that point forward to log in, the client needs to review the example and guides the secret key from design with enrolled key qualities, making a secret word by including sham digits. It minimizes shoulder surfing, brute- force attacks, cross site scripting etc. due to the high complexity of guessing passwords in multi-levels: first from the pattern, then from key, and then from dummy values.

The secret key is he fundamental key to get approval however programmers are a lot of fruitful in secret phrase breaking because of the frail secret key chose by the client. To reinforce the secret key stockpiling, the proposed framework utilizes the Honeyword procedure alongside Honey encryption. Honeywords are falsepasswords

which are put away with unique secret word to draw the aggressor. The basic idea behind Honeyword is the insertion of false passwords. These are to lure the attack.
To generate the Honeyword of original password different techniques like Chaffing-with-tweaking, Chaffing-with- password model, etc. are available, but in theexisting approach.

References

[1] Antonella De Angeli, Lynne Coventry, Graham John-son, and Karen Renaud. Does the picture really cost a thousand words? to assess the feasibility of explicit verification systems. International Journal of Computer Studies, 63: 128–152, July 2005.

[2] Xiaoyuan Suo, Ying Zhu, and G. Scott Owen. With pictures passwords: Survey. In the Computer System of the Year Defense Claims Conference, pages 463–472, 2005.

[3] K. Renaud, "Graphical design guidelines the authenticity of the path, "International Journal of Information and Computer Security, vol. 3, no. 1, pages 60-85, June 2009.

[4] A.De Angeli, L. Coventry, G. Johnson, and K. Renaud, "Is a picture really worth a thousand words? Possibility testing of photographic verification systems, "International Journal of Human-Computer Studies, vol. 63, nxa. 1-2, pages 128-152, 2005.

[5] F. Craik and J. McDowd, "The age difference in memory once recognition, "Journal of Experimental Psychology: Learning, Memory, and Understanding, vol. 13, no. 3, pages 474-479, July 1987. [6] K.-P. L. Vu, R. Proctor, A. Bhargav-Spantzel, B.-L. Tai, J.Cook, and E. Schultz, "Improving password security once remembering personal and organizational protection information, "International Journal of Human-Computer Studies, volume. 65, pages 744-757, 2007.

[7] S. Chiasson, A. Forget, E. Stobert, P. C. van Oorschot, and R. Biddle, "Disruption of multiple passwords in text and clicking graphic passwords. "on ACM Computer as well Communication Security (CCS), November 2009.

[8] L. Standing, J. Conezio, and R. Haber, "Perception and photographic memory: One 2500 visual test reading incentives, "Psychonomic Science, vol. 19, no. 2, p. 7374, 1970.

[9] D. Nelson, V. Reed, and J. Walling, "Pictorial Superiority
Impact, "Journal of Experimental Psychology: Human Learning and Memory, vol. 2, no. 5, pp. 523-528, 1976.