Front End Engineering-II

Project Report

Semester-IV (Batch-2022)

Title of the Project: -

PasswordHub

Supervised By:
Raveesh Samkaria

Submitted By:
Swapnil Dhamija
Roll Number: -2210990881
Group - 13

Department of Computer Science and Engineering
Chitkara University Institute of Engineering & Technology,
Chitkara University, Punjab

Abstract

This project report details the development of PasswordHub, a dynamic web application built using HTML, JavaScript, and Tailwind CSS, designed to revolutionize the creation of secure passwords. Users can explore a wide range of password options, from simple to complex, through an intuitive interface. PasswordHub enables users to easily generate, customize their preferred passwords in various formats. With its focus on user-friendly design and adaptability, PasswordHub meets the varied requirements of individuals seeking to enhance their digital security, marking a significant advancement in password management.

Table of Contents

Introduction

In the digital landscape of today, where visual appeal plays a critical role, the ability to utilize color effectively is key to creating compelling designs. Whether designing a website, crafting a logo, or developing marketing materials, the strategic use of color can significantly impact user engagement and brand perception. This report documents the development and deployment of PasswordHub, an innovative password generator web application, leveraging a dynamic combination of HTML, JavaScript, and Tailwind CSS.

PasswordHub emerges as an essential tool for users seeking to enhance their digital security, offering a convenient platform to generate strong and secure passwords tailored to their specific needs. The application's user-friendly interface and seamless functionality aim to simplify the password creation process, allowing users to generate, customize, and save their passwords effortlessly.

This introduction sets the context for the project, highlighting the importance of secure passwords in today's digital landscape and the role of PasswordHub in meeting this need. Subsequent sections will delve into the technical aspects of the application's development, including key features, design considerations, and implementation strategies specific to password generation. The report will also address any challenges encountered during the project and discuss the solutions devised to overcome them, offering valuable insights for future password generator development projects.

Problem Statement

In the digital landscape of today, where visual appeal plays a critical role, the ability to utilize color effectively is key to creating compelling designs. Whether designing a website, crafting a logo, or developing marketing materials, the strategic use of color can significantly impact user engagement and brand perception. This report documents the development and deployment of PasswordHub, an innovative password generator web application, leveraging a dynamic combination of HTML, JavaScript, and Tailwind CSS.

PasswordHub emerges as an essential tool for users seeking to enhance their digital security, offering a convenient platform to generate strong and secure passwords tailored to their specific needs. The application's user-friendly interface and seamless functionality aim to simplify the password creation process, allowing users to generate, customize, and save their passwords effortlessly.

This introduction sets the context for the project, highlighting the importance of secure passwords in today's digital landscape and the role of PasswordHub in meeting this need. Subsequent sections will delve into the technical aspects of the application's development, including key features, design considerations, and implementation strategies specific to password generation. The report will also address any challenges encountered during the project and discuss the solutions devised to overcome them, offering valuable insights for future password generator development projects.

Technical Details

PasswordHub is built using a blend of HTML, Tailwind CSS, and JavaScript to ensure a smooth user experience and secure password generation process.

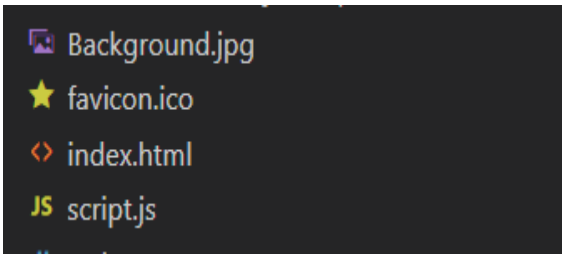
- **HTML:** Serving as the foundation of the application, HTML (Hypertext Markup Language) is utilized to structure the content and layout of PasswordHub. HTML is responsible for organizing elements such as input fields, buttons, and password display areas, ensuring the interface is clear and user-friendly. By structuring the content with HTML, PasswordHub guarantees accessibility and compatibility across various devices and browsers.
- **Tailwind CSS:** Tailwind CSS is employed to style and design the visual components of PasswordHub. Tailwind CSS offers a utility-first approach, allowing for rapid styling and customization through pre-defined utility classes. With Tailwind CSS, PasswordHub achieves a clean and modern look while maintaining flexibility and scalability. Tailwind CSS assists in styling elements such as buttons, forms, and alerts, ensuring a consistent and visually appealing user interface.
- **JavaScript:** JavaScript plays a crucial role in enabling the interactive features of PasswordHub. Through JavaScript, PasswordHub implements functionalities such as password generation algorithms, and user input validation. JavaScript enhances the user experience by enabling users to customize their password requirements. Additionally, JavaScript facilitates the copying of generated passwords, enhancing the usability and convenience of PasswordHub.

By leveraging HTML, Tailwind CSS, and JavaScript, PasswordHub delivers a secure and user-friendly password generator that empowers users to create strong and unique passwords effortlessly. The seamless integration of these technologies ensures optimal performance, accessibility, and interactivity, making PasswordHub a reliable tool for enhancing digital security practices.

File Structure

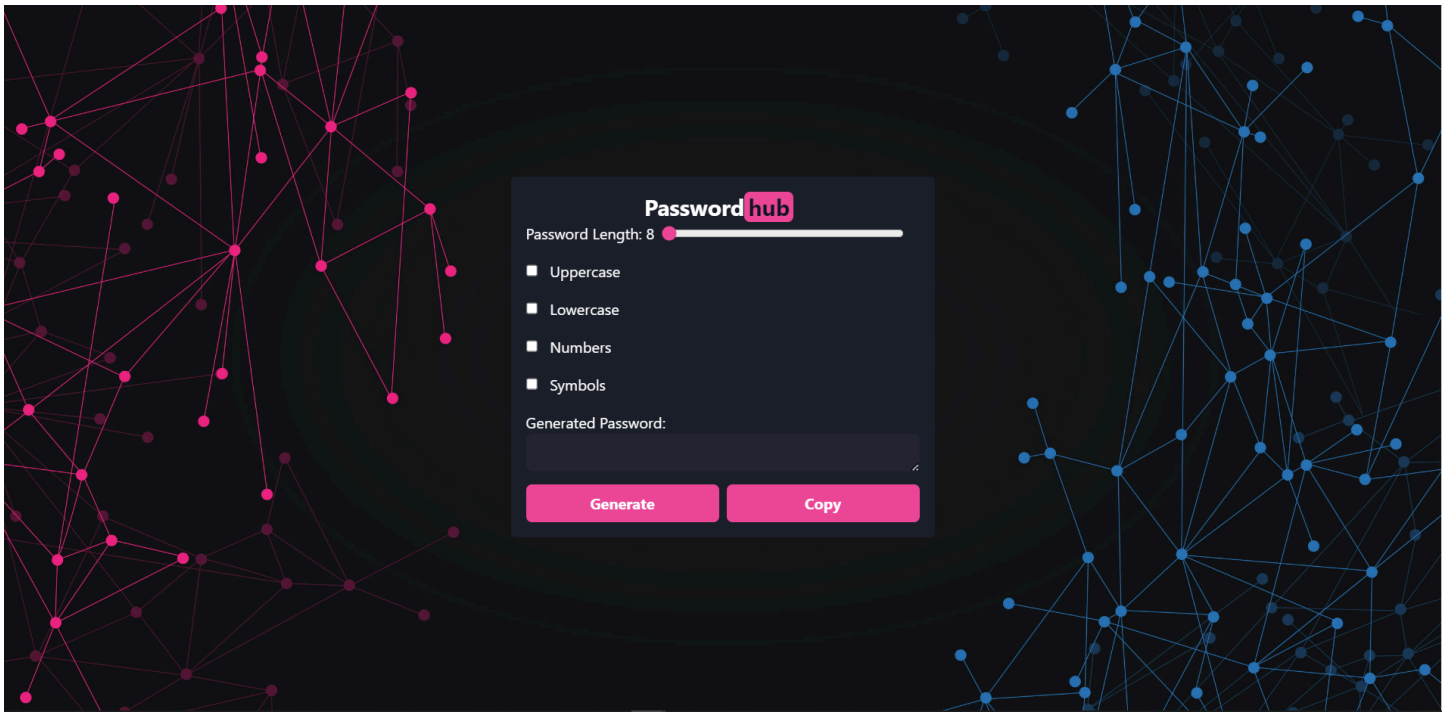
The file structure of ColorVerse adheres to a minimalist approach, comprising essential files to ensure efficient organization and streamlined development. Here's an overview of the file structure:

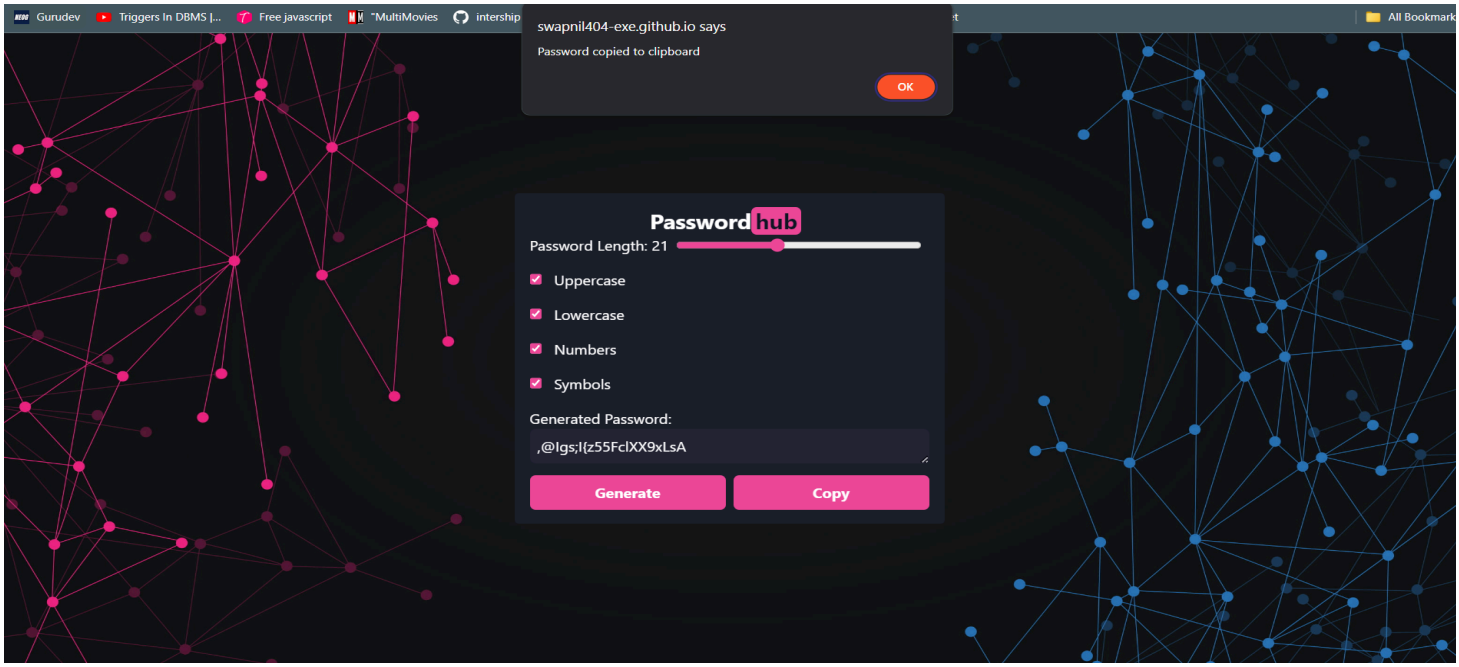
1. **index.html:** The heart of PasswordHub, index.html serves as the main entry point for the website. This file contains the HTML markup responsible for structuring the content and layout of PasswordHub. Elements such as input range, password preview, generate and copy buttons are defined within index.html, providing users with a seamless browsing experience.
2. **script.js:** The JavaScript functionality of PasswordHub is encapsulated within the script.js file. Here, essential functionalities such as real-time password generation, user interaction handling, and dynamic content updates are implemented. **script.js** ensures the interactivity and responsiveness of PasswordHub, enhancing the overall user experience.
3. **Background.jpg:** The background image used in PasswordHub, background.jpg, enhances the visual appeal of the website. This image provides a backdrop that complements the overall design aesthetic of PasswordHub, creating a visually engaging environment for users. The choice of background image is carefully selected to maintain a balance between aesthetics and usability, ensuring that the content remains easily readable and accessible. Through the use of background.jpg, PasswordHub achieves a cohesive and inviting visual style, enhancing the user experience and reinforcing the website's branding.
4. **Favicon.ico:** The favicon.ico file is a small but important element of PasswordHub's branding and user experience. This file contains the icon that appears in the browser tab when users visit the website. The favicon.ico icon is designed to be distinctive and easily recognizable, serving as a visual representation of PasswordHub's identity. By using a custom favicon.ico icon, PasswordHub enhances its brand visibility and provides users with a cohesive browsing experience across different tabs and bookmarks.



Result

Webpage Screenshots: -





Code

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Password Generator</title>
  <script src="https://cdn.tailwindcss.com"></script>
  <link rel="stylesheet" href="style.css">
  <link rel="icon" href="./favicon.ico" type="image/x-icon">
</head>
```

Figure 1 HTML- Head

```html
<body class="bg-gray-100 h-screen flex justify-center items-center">
  <div id="main-app" class="max-w-md w-full p-4 bg-white rounded  ">
    <div id="heading" class="flex justify-center">
    <h1 class="0 text-white text-2xl flex justify-center font-bold rounded-md
    ">Password</h1><h1 class="text-gray-900 mr-2 font-bold text-2xl bg-pink-500 rounded-md px-1">hub</h1>
    </div>
    <div class="mb-4">
      <div class="flex items-center">
        <label for="length" class="mr-2 text-white text-l">Password Length: <span id="lengthValue">8</span></label>
        <input type="range" id="length" name="length" min="8" max="40" class="w-64 h-6 bg-gray-300 rounded-full  outline-none
        accent-pink-500 cursor-pointer" value="8" >
      </div>
    </div>
    <div class="mb-4" id="checkboxes " class="">
      <input type="checkbox" id="uppercase" class="mr-2 text-white  accent-pink-500 bg-white ">
      <label for="uppercase" class="text-l text-white">Uppercase</label>
    </div>
    <div class="mb-4">
      <input type="checkbox" id="lowercase" class="mr-2 text-white accent-pink-500 ">
      <label for="lowercase" class="text-l text-white">Lowercase</label>
    </div>
```

Figure 2 HTML - Body

```
    <div class="mb-4">
      <input type="checkbox" id="numbers" class="mr-2 text-white accent-pink-500">
      <label for="numbers" class="text-l text-white">Numbers </label>
    </div>
    <div class="mb-4">
      <input type="checkbox" id="symbols" class="mr-2 text-white accent-pink-500">
      <label for="symbols" class="text-l text-white">Symbols </label>
  </div>
    <label for="password" class="block text-white">Generated Password:</label>
    <textarea id="password" class="w-full p-2  rounded-md text-white" rows="1" readonly></textarea>
    <div class="flex flex-row gap-2">
    <button id="generate" class="w-full bg-pink-500 text-white py-2 rounded-md font-bold mt-2">Generate </button>
    <button id="copy" class="w-full bg-pink-500 text-white py-2 rounded-md font-bold mt-2">Copy </button>
  </div>
  </div>
  <script src="script.js"></script>
</body>

</html>
```

Figure 3 HTML Body

```
body {
    background-image: url('./Background.jpg');
    background-size: cover;
    background-position: center;
}
#main-app{
  background-color: rgba(73, 79, 154, 0.156);
  backdrop-filter: blur(300px);
}

#password, #length{
  background-color: rgba(56, 50, 79, 0.263);
  color: white;
}
```

Figure 4 External CSS

```
const generateBtn = document.getElementById('generate');
const lengthInput = document.getElementById('length');
const numberInput = document.getElementById('numbers');
const symbolInput = document.getElementById('symbols');
const uppercaseInput = document.getElementById('uppercase');
const lowercaseInput = document.getElementById('lowercase');
const passwordTextarea = document.getElementById('password');
const lengthValue = document.getElementById('lengthValue');
const copyBtn = document.getElementById('copy');

lengthInput.addEventListener('input', () => {
  lengthValue.innerText = lengthInput.value;
});
```

Figure 5 Javascript

```
generateBtn.addEventListener('click', () => {
  const length = +lengthInput.value;
  const hasUppercase = uppercaseInput.checked;
  const hasLowercase = lowercaseInput.checked;
  const hasSymbols = symbolInput.checked;
  const hasNumbers = numberInput.checked;

  const uppercaseChars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ';
  const lowercaseChars = 'abcdefghijklmnopqrstuvwxyz';
  const numbers = '0123456789';
  const symbols = '!@#$%^&*()_+-=[]{}|;:,.<>?';

  let chars = '';
  if (hasUppercase) chars += uppercaseChars;
  if (hasLowercase) chars += lowercaseChars;
  if (hasNumbers) chars += numbers;
  if (hasSymbols) chars += symbols;

  if(!hasLowercase && !hasNumbers && !hasSymbols && !hasUppercase) {
    alert('Choose atleast one option.');
    return;
  }
  let password = '';
  for (let i = 0; i < length; i++) {
    const randomIndex = Math.floor(Math.random() * chars.length);
    password += chars[randomIndex];
  }

  passwordTextarea.value = password;
});
```

Figure 6 Javascript

```
copyBtn.addEventListener('click', () => {
    if (passwordTextarea.value.trim() === '') {
        alert('Generate a password first');
    } else {
        passwordTextarea.select();
        document.execCommand('copy');
        alert('Password copied to clipboard');
    }
});
```

Figure 7 Javascript

References

1. https://www.w3schools.com/
2. https://fonts.google.com/
3. https://tailwindcss.com/
4. https://cdn.tailwindcss.com
5. https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.3.0/css/all.min.css