VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JNANA SANGAMA", BELAGAVI-590018

MINI PROJECT REPORT

On

"Custom Password Generator and Strength Analyzer"

Bachelor of Engineering

in

Information Science and Engineering

of

Visvesvaraya Technological University, Belagavi.

By

VAIBHAV SRIVASTAVA	**1CD22IS173**
TUSHAR MISHRA	**1CD22IS171**
PRATHAMESH SHAHAPURKAR	**1CD23IS408**
SHRIKISHAN RAIBAGI	**1CD22IS159**

Under the guidance of

Prof. Mamatha

Asst. Professor, Dept. of ISE Citech , Bengaluru

CAMBRIDGE INSTITUTE OF TECHNOLOGY

BANGALORE - 560 036

2024-2025

CAMBRIDGE INSTITUTE OF TECHNOLOGY

(Affliated to VTU- Belgaum)

K.R. Puram, Bangalore-560 036

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the Mini Project work entitled "Custom Password Generator and Strength Analyzer" is a bonified work carried out by **VAIBHAV SRIVASTAVA (1CD22IS173)**, **TUSHAR MISHRA (1CD22IS171), PRATHAMESH SHAHAPURKAR (1CD23IS408), SHRIKISHAN RAIBAGI (1CD22IS159)** in partial fulfilment in the requirement for V semester Bachelor of Engineering in Information Science and Engineering of Visvesvaraya Technological University, Belagavi during the year 2024-2025.

It is certified that all the corrections/suggestions indicated for internal assessment have been incorporated in the Report. The report has been approved as it satisfies the academic requirements in respect of Mini-Project work prescribed for said degree.

Signature of the Guide	**Signature of HOD**
Prof. Mamatha	**Dr. Preethi S**

ABSTRACT

In the modern digital era, securing sensitive information has become increasingly crucial, given the alarming frequency of cyberattacks, data breaches, and unauthorized access incidents. Among various security measures, passwords remain one of the most fundamental yet most vulnerable aspects of digital security. Weak or predictable passwords are a significant cause of compromised accounts, leaving personal and organizational data at risk. This project responds to this challenge by creating an interactive and user-friendly Password Generator and Strength Analyzer tool, which makes it easy to create and evaluate strong passwords.

This tool has been designed to have two functionalities. The first functionality is the password generator, which enables the users to generate highly secure passwords based on their specific needs. Users can input desired parameters such as password length, whether to include uppercase letters, lowercase letters, numbers, special characters, or even custom words or numbers. This approach ensures that the generated passwords meet user requirements while being in line with best security practices. The Password Strength Analyzer evaluates the strength of any provided password. It checks various criteria, including length, character diversity, and complexity, and calculates a strength percentage. The analyzer also offers actionable recommendations on how to strengthen the password by pointing out missing components.

The tool was built using web technologies, namely HTML, CSS, and JavaScript, which ensures the user experience is responsive, cross-platform, and aesthetically pleasing with a dark theme. This tool integrates customization, real-time feedback, and strength analysis to provide an intuitive and educational platform for improving hygiene in passwords. The project targets personal, educational, and corporate users alike, in the aim of enhancing general cybersecurity awareness and practices.

CONTENTS

List of Figures

CHAPTER 1

INTRODUCTION

1.1 Background

In the digital age, passwords are the most basic yet critical tools for securing access to personal, professional, and organizational systems. They are the gatekeepers of sensitive data, ensuring that only authorized individuals can access it. With the exponential growth of internet usage and the proliferation of online services, the importance of strong password security has become more prominent than ever. Unfortunately, many individuals and organizations fail to give this aspect the attention it deserves, leading to significant vulnerabilities in their security frameworks.

Weak or reused passwords are among the most common reasons for data breaches. Despite countless awareness campaigns, many users continue to choose passwords like "123456," "password," or their pet's name—passwords that are easy to guess. This is compounded by the fact that users often reuse passwords across multiple accounts, creating a domino effect: a breach in one account can lead to unauthorized access in several others.

The rise of cyberattacks, including brute force attacks, dictionary attacks, and phishing schemes, further emphasizes the need for robust password practices. Hackers employ increasingly sophisticated tools to exploit weak passwords, often compromising millions of accounts in one go. This alarming trend highlights the pressing need for tools that not only help users generate secure passwords but also educate them about password strength and best practices.

Another challenge is that complex, secure passwords are often difficult to remember, pushing users toward unsafe practices like writing them down or saving them in unencrypted files. Thus, a balance must be struck between security and usability. A tool that can generate strong, unique passwords and evaluate their strength in real-

time while being intuitive and user-friendly can significantly enhance cybersecurity practices.

This project addresses these challenges by offering a **Password Generator and Strength Analyzer Tool** that is both practical and educational. The tool enables users to create passwords based on specific criteria and provides a detailed analysis of their strength, including actionable recommendations to improve them. By integrating password generation and analysis functionalities into a single interface, this project bridges the gap between user convenience and security requirements, contributing to a safer digital environment.

1.2 Model Information

The **Password Generator and Strength Analyzer Tool** is designed with two core functionalities:

1. **Password Generator**: This module allows users to create secure passwords tailored to their specific needs. Users can define criteria such as password length, the inclusion of uppercase and lowercase letters, numbers, special characters, and even specific words or numbers. This level of customization ensures that the generated passwords are both secure and user-friendly.
2. **Password Strength Analyzer**: This module evaluates any given password against a set of predefined security criteria. It calculates a strength percentage based on factors like length, character diversity, and complexity. If the password does not meet certain criteria, the analyzer provides recommendations, such as including additional character types or increasing the password length.

The tool leverages modern web technologies, including **HTML**, **CSS**, and **JavaScript**, for its implementation. It features an aesthetically pleasing dark-themed interface that is both responsive and intuitive. The modular design ensures that users can easily switch between password generation and analysis functionalities, making it a one-stop solution for password management.

The model is designed to enhance both individual and organizational security practices by promoting better password hygiene. By providing real-time feedback and education on password strength, the tool empowers users to take proactive steps toward protecting their digital identities.

1.3 Applications

The **Password Generator and Strength Analyzer Tool** has a wide range of applications, making it a versatile solution for various user groups.

1. **Personal Use**:
    - Individuals can use the tool to generate secure passwords for their online accounts, including social media platforms, email services, and banking portals.
    - The analyzer ensures that these passwords are robust enough to withstand hacking attempts, providing users with peace of mind.
2. **Corporate Use**:
    - Organizations can employ the tool to enforce strong password policies among employees, reducing the risk of internal breaches.
    - It can be integrated into training programs to educate employees about the importance of strong passwords and the risks associated with weak ones.
3. **Educational Institutions**:
    - Schools and universities can use the tool to teach students about cybersecurity.
    - By demonstrating password strength and best practices, educators can instill a culture of digital security in young learners.
4. **Cybersecurity Training Programs**:
    - The tool can be included in cybersecurity workshops and training sessions to provide hands-on experience in creating and evaluating passwords.
    - Trainees can experiment with different password combinations to understand the characteristics of strong passwords.
    - Provide an improving environment for the password strength analysis.

5. **Password Management Services**:
    - Developers of password management applications can integrate this tool as an additional feature, offering users a comprehensive solution for password creation and evaluation.
6. **Government and Financial Institutions**:
    - These sectors, which handle highly sensitive information, can use the tool to enforce stringent password policies among their employees and stakeholders.

By catering to such diverse applications, the tool serves as a valuable asset for promoting better cybersecurity practices across various domains.

1.4 Problem Statement

Passwords are a cornerstone of digital security, yet they are often the weakest link in protecting sensitive information. The increasing reliance on online services has made passwords a critical aspect of user authentication. However, the widespread adoption of weak, predictable, or reused passwords poses a significant risk to both individuals and organizations.

Several challenges contribute to this problem:

1. **Lack of Awareness**:
   Many users lack a clear understanding of what constitutes a strong password. They may not know the importance of using a mix of characters or the risks associated with reusing passwords across multiple accounts.
2. **Convenience Over Security**:
   Users often prioritize convenience, opting for passwords that are easy to remember but equally easy to guess. This tendency is a major factor behind the prevalence of weak passwords.
3. **Sophisticated Cyberattacks**:
   Hackers have developed advanced techniques, such as brute force attacks and dictionary attacks, to exploit weak passwords. The increasing frequency and

4. sophistication of these attacks underscore the urgent need for stronger password practices.
5. **Usability Challenges**:
   While strong passwords are essential for security, they are often difficult to remember. This creates a usability issue, as users may resort to unsafe practices like writing down their passwords or saving them in insecure formats.
6. **Lack of Tools for Evaluation**:
   Existing password generation tools often fail to provide real-time feedback on password strength. Users may not receive actionable recommendations for improving their passwords, leaving them unaware of potential vulnerabilities.

The **Password Generator and Strength Analyzer Tool** addresses these challenges by offering a comprehensive solution that simplifies the process of creating and evaluating passwords. It empowers users to create secure passwords tailored to their specific needs while providing detailed feedback on their strength. By bridging the gap between security and usability, this tool promotes better password hygiene and contributes to a safer digital environment.

CHAPTER 2

APPLICATION SURVEY

In the realm of cybersecurity, the importance of strong passwords cannot be overstated. Passwords are the foundation of digital authentication systems, securing personal, organizational, and financial information. However, as the prevalence of cyberattacks rises, the demand for effective password management solutions has grown significantly. This chapter explores various existing tools, technologies, and approaches to password generation and strength analysis, providing a detailed survey of their features, benefits, and limitations.

2.1 Overview of Existing Password Generators

Password generators are widely used tools that create random, strong passwords based on predefined criteria. These tools have become increasingly popular as users recognize the need for secure passwords to protect their accounts. Some of the most commonly used password generators include:

1. **Built-in Password Generators in Browsers**:
    - Modern web browsers like Google Chrome, Mozilla Firefox, and Microsoft Edge offer built-in password generators.
    - These tools automatically generate strong, random passwords when users create new accounts.
    - While convenient, they are often limited to the browser ecosystem, requiring users to rely on the browser's password management system.
2. **Standalone Password Generators**:
    - Software applications like LastPass, Dashlane, and 1Password include password generation features.
    - These tools allow users to customize passwords by setting length and character types.

- However, they often require subscriptions for full access, making them less accessible to casual users.

3. **Online Password Generators**:
    - Numerous websites offer free password generation services. Examples include Strong Password Generator and Norton's Password Generator.
    - While free and easy to use, online tools can pose privacy risks, as users may inadvertently share sensitive data.
4. **Mobile Applications**:
    - Mobile apps like Keeper and Bitwarden provide password generation and management capabilities.
    - They are particularly useful for users who manage multiple accounts on mobile devices.

While these tools provide robust password creation, many fail to offer detailed analysis of password strength or guidance on how to improve weak passwords.

2.2 Overview of Password Strength Analyzers

Password strength analyzers evaluate the robustness of passwords against predefined security criteria. These tools are crucial for educating users about weak passwords and helping them create stronger alternatives. Popular implementations include:

1. **Password Strength Meters**:
    - Found in many websites and applications, these meters provide a visual indicator (e.g., weak, medium, strong) of a password's strength.
    - They often assess length, complexity, and diversity of characters but rarely provide actionable feedback.
2. **Online Strength Analysis Tools**:
    - Websites like Kaspersky Password Checker evaluate password strength and suggest improvements.
    - However, using online tools can pose security risks, as users must input their passwords into third-party systems.
3. **Integrated Analyzers in Password Managers**:

- Tools like LastPass and 1Password analyze stored passwords and highlight weak or reused ones.
- These systems often require users to rely heavily on their ecosystems for password management.

4. **Custom Scripts and Open-Source Tools**:
    - Developers often create custom password analyzers using programming languages like Python or JavaScript.
    - Open-source tools provide flexibility and transparency but may require technical expertise to use effectively.

Although these analyzers are effective in evaluating password strength, they rarely integrate seamlessly with password generators to provide an all-in-one solution.

2.3 Comparative Analysis of Existing Tools

While existing password generators and analyzers serve their purposes effectively, they often function as standalone systems. A comparative analysis reveals the following insights:

1. **Strengths**:
    - Password generators produce secure, random passwords, reducing the risk of brute force and dictionary attacks.
    - Strength analyzers educate users about password vulnerabilities and provide a benchmark for improvement.
    - Tools integrated into browsers and password managers offer convenience and accessibility.
2. **Limitations**:
    - Lack of Integration: Most tools do not combine generation and analysis in a single interface, requiring users to switch between multiple systems.
    - Limited Customization: Many generators provide only basic options, failing to accommodate specific user requirements.

- Privacy Concerns: Online tools often require users to input passwords, potentially exposing sensitive data.
- Usability Challenges: Tools with overly complex interfaces deter non-technical users, limiting their effectiveness.

3. **Emerging Trends**:
   - Increasing focus on user education and awareness about password security.
   - Integration of advanced algorithms, such as machine learning, to improve strength analysis.
   - Emphasis on user-centric design to make tools more accessible and intuitive.

2.4 Identified Gaps and Opportunities

The survey of existing tools highlights several gaps that the **Password Generator and Strength Analyzer Tool** aims to address:

1. **Seamless Integration**:
   - By combining password generation and strength analysis in a single platform, the tool eliminates the need for multiple systems.
2. **Customization Options**:
   - The tool allows users to define specific criteria, including the inclusion of custom numbers or words, enhancing usability.
3. **Privacy Protection**:
   - As a standalone application, the tool ensures that user data remains secure, avoiding the risks associated with online platforms.
4. **Actionable Feedback**:
   - Unlike basic strength meters, the tool provides detailed recommendations for improving weak passwords, empowering users to take proactive steps toward better security.
5. **User-Friendly Design**:
   - The dark-themed, aesthetically pleasing interface ensures that the tool is both functional and visually appealing, catering to a wide audience.

2.5 Relevance to the Project

The insights gained from this application survey serve as the foundation for the design and development of the **Password Generator and Strength Analyzer Tool**. By addressing the limitations of existing systems and leveraging their strengths, this project aims to deliver a comprehensive solution that promotes better password hygiene and enhances cybersecurity practices.

This chapter demonstrates the importance of understanding the current landscape to develop innovative solutions. By filling the identified gaps and incorporating user-centric features, the tool sets a new standard for password management systems.

CHAPTER 3

SYSTEM REQUIREMENTS SPECIFICATION

3.1 Hardware Requirements

The successful implementation of the **Password Generator and Strength Analyzer Tool** requires specific hardware components to ensure smooth operation and an optimal user experience. Given its lightweight nature as a web-based application, the hardware requirements are minimal, making it accessible to a wide range of users.

1. **Client-Side Requirements**:
    - **Device**: A desktop, laptop, tablet, or smartphone capable of running a modern web browser.
    - **Processor**: Any processor with a minimum clock speed of 1 GHz (e.g., Intel Pentium, AMD Ryzen, or equivalent).
    - **RAM**: At least 2 GB of RAM for standard operation, though 4 GB or more is recommended for multitasking.
    - **Storage**: Minimal storage space is required, primarily for temporary browser cache. No installation is needed as the tool is browser-based.
    - **Display**: A resolution of 1024x768 pixels or higher for optimal viewing of the tool's interface.
2. **Server-Side Requirements** (if hosted online):
    - **Server**: A hosting environment or virtual private server (VPS) for deploying the tool, with at least 2 CPU cores.
    - **RAM**: 4 GB or more to handle multiple user sessions simultaneously.
    - **Storage**: 20 GB of storage for storing application files, logs, and potential updates.
    - **Network**: A stable internet connection with a minimum bandwidth of 10 Mbps for efficient client-server communication.

With these modest hardware requirements, the tool is designed to run seamlessly across various platforms, ensuring accessibility and ease of use.

3.2 Software Requirements

The **Password Generator and Strength Analyzer Tool** relies on modern web development technologies and frameworks to deliver a smooth and secure user experience. The software requirements are categorized into two components: client-side and server-side.

1. **Client-Side Software**:
    - **Web Browser**: A modern browser such as Google Chrome, Mozilla Firefox, Microsoft Edge, or Safari, supporting the latest HTML5, CSS3, and JavaScript standards.
    - **Operating System**: Cross-platform compatibility ensures that the tool works on Windows, macOS, Linux, Android, and iOS.
2. **Development Tools and Frameworks**:
    - **HTML5**: For structuring the content and interface of the application.
    - **CSS3**: For styling the dark-themed, aesthetic user interface.
    - **JavaScript**: For implementing interactive features such as real-time password analysis and dynamic password generation.
3. **Server-Side Software** (if deployed online):
    - **Node.js or Python**: As the backend environment for hosting additional features or services, if required.
    - **Database Management** (Optional): A lightweight database, such as SQLite or MongoDB, for storing user preferences or session logs.
    - **Hosting Platform**: A web hosting service like AWS, Heroku, or Netlify to deploy the application for public or private use.
4. **Development Environment**:
    - **Code Editor**: Tools such as Visual Studio Code, Sublime Text, or Atom for developing and maintaining the application.
    - **Version Control**: Git for version control and GitHub for collaborative development and repository hosting.
5. **Libraries and APIs**:
    - Libraries like Bootstrap or Tailwind CSS for additional styling options.

- Optional cryptographic libraries (e.g., CryptoJS) for securing sensitive user data.

By meeting these software requirements, the tool ensures a secure, efficient, and user-friendly experience, adaptable to both local and online deployment

CHAPTER 4

DESIGN

4.1 Purpose

The **Password Generator and Strength Analyzer Tool** is designed to address the growing need for robust password security by combining a generator and strength analyzer into a single, user-friendly platform. Its primary purpose is to empower users to create strong, secure passwords while providing actionable insights into improving weak or moderate passwords.

Key objectives of the tool include:

1. **Enhancing Password Security**:
    - By generating passwords based on customizable user inputs and established security standards, the tool ensures that users create strong and unique passwords, reducing vulnerability to cyberattacks.
2. **Educating Users**:
    - The integrated strength analyzer educates users about password weaknesses, providing detailed feedback on how to strengthen their credentials by highlighting missing elements such as uppercase letters, numbers, or symbols.
3. **Convenience and Accessibility**:
    - Designed with a modern, dark-themed interface, the tool offers a seamless experience across devices and platforms, catering to a diverse audience, including non-technical users.
4. **Privacy and Security**:
    - As a standalone or locally deployable application, the tool ensures that sensitive user data remains private and is not transmitted to third-party servers.
5. **Customization**:
    - Unlike generic tools, this tool allows users to specify numbers, words, or special characters they wish to include in generated passwords, ensuring personalization without compromising security.

By achieving these purposes, the tool contributes to better password hygiene, a critical component of modern cybersecurity practices.

4.2 System Architecture

The **Password Generator and Strength Analyzer Tool** is built using a modular, client-centric architecture, ensuring flexibility, scalability, and efficiency. The architecture includes the following core components:

1. **User Interface (UI)**:
    - The UI serves as the primary interaction layer, designed using HTML5, CSS3, and JavaScript.
    - It features a sleek, dark-themed design with intuitive controls, including input fields, sliders, and buttons for password generation and analysis.
    - Key modules in the UI include:
        - Password Generator Interface: Collects user inputs (e.g., length, character types, specific words/numbers).
        - Password Strength Meter: Provides real-time feedback on the strength of generated or inputted passwords.
2. **Application Logic**:
    - The core logic is implemented using JavaScript, enabling dynamic functionality and real-time analysis.
    - Password generation logic:
        - Uses randomization algorithms to create secure passwords based on user-defined criteria.
        - Ensures compliance with security standards such as length, character diversity, and randomness.
    - Password strength analysis logic:
        - Evaluates passwords against predefined criteria, including length, uppercase/lowercase letters, numbers, symbols, and uniqueness.
        - Generates a percentage score representing overall strength and provides actionable recommendations for improvement.

3. **Backend (Optional)**:
    - For online deployments, a backend service using Node.js or Python can handle additional features like session management or secure data storage.
    - If no backend is required, the tool operates entirely on the client-side, ensuring privacy and simplicity.
4. **Data Flow**:
    - User inputs (criteria for password generation or password for analysis) are processed by the application logic.
    - The output (generated password or strength score) is displayed in real time on the UI.
    - Feedback and recommendations are dynamically updated based on user interactions.
5. **Security Measures**:
    - Encryption libraries (e.g., CryptoJS) are integrated to secure any sensitive operations.
    - No passwords are stored or transmitted, ensuring user privacy.

Flow of Operations:

1. The user accesses the tool via a web browser.
2. For password generation:
    - The user specifies criteria such as length, character types, and custom inputs.
    - The tool generates a secure password, displays it, and analyzes its strength.
3. For password analysis:
    - The user inputs a password for evaluation.
    - The tool calculates a strength score and provides recommendations for improvement.

By leveraging this architecture, the tool ensures a robust, efficient, and user-centric approach to password security.

CHAPTER 5

SYSTEM DEVELOPMENT

5.1 Objectives

The **Password Generator and Strength Analyzer Tool** aims to achieve several objectives that address the challenges of modern password security. These objectives serve as a guiding framework for the tool's development and functionality:

1. **Facilitating Strong Password Creation**:
    - Generate passwords that meet robust security standards, including a combination of uppercase letters, lowercase letters, numbers, and symbols.
    - Allow users to customize passwords by including specific words or numbers as per their requirements.
2. **Enhancing User Awareness**:
    - Educate users on the importance of strong passwords through real-time feedback on password strength.
    - Provide actionable recommendations to improve weak passwords, fostering better password hygiene.
3. **Streamlined User Experience**:
    - Develop an intuitive, aesthetically pleasing interface with a modern dark theme to enhance usability.
    - Ensure that the tool is accessible across multiple devices and platforms, catering to a diverse audience.
4. **Privacy and Security**:
    - Operate entirely on the client-side to ensure user data privacy.
    - Implement secure algorithms for password generation and strength analysis without storing or transmitting sensitive data.
5. **Customization and Flexibility**:
    - Offer extensive customization options for password generation, such as length, character types, and specific inputs.

- Provide a modular structure that can be easily extended or integrated into other applications.

By meeting these objectives, the tool positions itself as a comprehensive solution for addressing the challenges of password security in a user-friendly manner.

5.2 Methodology

The development of the **Password Generator and Strength Analyzer Tool** followed an agile methodology to ensure iterative improvement, user feedback integration, and efficient resource utilization. The methodology included the following phases:

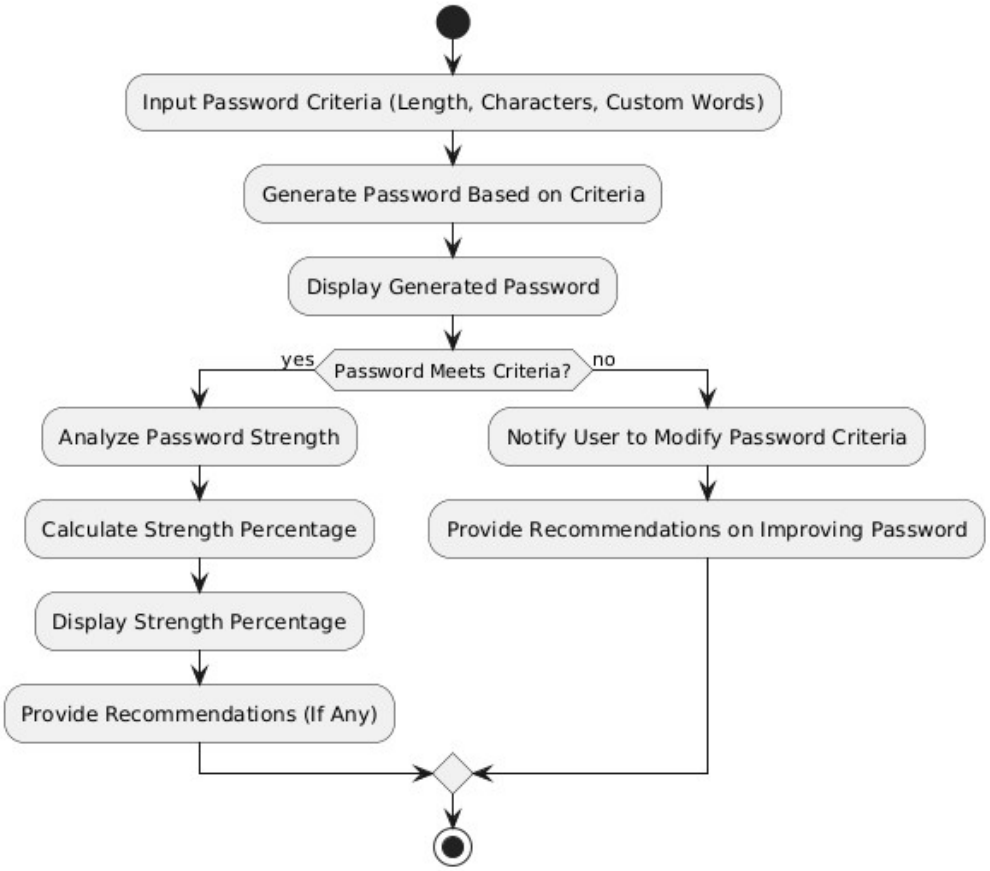
1. **Requirement Analysis**:
   - Identified the need for an integrated tool to generate secure passwords and analyze their strength.
   - Conducted an application survey to understand the limitations of existing tools.
2. **Design and Planning**:
   - Defined the tool's architecture and functionality based on the identified objectives.
   - Created wireframes and prototypes to visualize the user interface and user experience.
3. **Development**:
   - Implemented the frontend using HTML5, CSS3, and JavaScript for real-time interactivity.
   - Developed core algorithms for password generation and strength analysis, focusing on efficiency and security.
   - Incorporated dynamic feedback mechanisms to educate users on improving password strength.
4. **Testing and Debugging**:
   - Conducted unit testing for individual modules, including the generator, analyzer, and feedback components.
   - Performed integration testing to ensure seamless operation across all components.

- Carried out user acceptance testing to validate the tool's usability and effectiveness.

5. **Deployment**:
   - Deployed the tool as a standalone web application, accessible via modern browsers.
   - Provided documentation and user guides to facilitate adoption and usage.
6. **Maintenance and Updates**:
   - Regularly updated the tool to incorporate user feedback and enhance functionality.
   - Ensured compatibility with evolving web standards and technologies.

5.3 Flow Chart

The flow chart below represents the logical flow of operations within the **Password Generator and Strength Analyzer Tool**:



5.4 System Design

The system design of the **Password Generator and Strength Analyzer Tool** is divided into multiple components to ensure modularity, scalability, and ease of development.

Frontend Design

1. **User Interface**:
    - A visually appealing dark-themed interface using CSS3 and modern design principles.
    - Input fields and sliders for specifying password generation criteria, such as length and character types.
    - Real-time password strength meter with dynamic feedback and recommendations.
2. **Interactive Elements**:
    - Buttons for generating and analyzing passwords.
    - Visual indicators (e.g., progress bars, color-coded meters) to represent password strength.
3. **Error Handling**:
    - Input validation to ensure user criteria are appropriate for password generation.
    - Friendly error messages to guide users in resolving issues.

Core Logic

1. **Password Generation Algorithm**:
    - A randomization algorithm that combines uppercase and lowercase letters, numbers, symbols, and user-specified inputs.
    - Ensures compliance with industry-standard password security guidelines.
2. **Strength Analysis Algorithm**:
    - Evaluates passwords against criteria such as length, complexity, and character diversity.
    - Calculates a strength score (percentage) and identifies missing elements.
    - Provides actionable feedback to users, such as “Add a special character for a stronger password.”

Backend Design

For deployment as an online tool, a lightweight backend can be implemented:

1. **API for Password Generation**:
    - Exposes an API endpoint for generating passwords based on user inputs.
2. **Session Management**:
    - Tracks user interactions to enhance the experience.

CHAPTER 6

IMPLEMENTATION

6.1 Language

The implementation of the **Password Generator and Strength Analyzer Tool** leverages modern web development languages and frameworks to ensure a seamless and interactive user experience. The chosen languages and their roles are outlined below:

1. **HTML5**:
    - Used for structuring the content of the tool.
    - Provides the framework for input fields, buttons, and other user interface elements.
2. **CSS3**:
    - Responsible for the styling and aesthetic design of the application.
    - Implements a modern dark theme, enhancing visual appeal and reducing eye strain.
    - Ensures responsiveness, allowing the tool to work on different screen sizes and devices.
3. **JavaScript**:
    - Powers the core functionality of the tool.
    - Implements algorithms for password generation, strength analysis, and real-time feedback.
    - Handles user interactions, such as updating the strength meter dynamically.

6.2 Code Imports:

INDEX.html

```
    <!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Custom Password Generator and Analyzer</title>
  <link rel="stylesheet" href="styles.css">
```

```
</head>
<body>
  <div class="container">
    <h1>🔒 Custom Password Generator & Strength Analyzer</h1>

    <!-- Password Generator Section -->
    <div id="password-generator" class="card">
      <h2>Password Generator</h2>
      <form id="generator-form">
        <div class="form-group">
          <label for="length">Password Length:</label>
          <input type="number" id="length" min="4" max="20" value="12">
        </div>
        <div class="form-group">
          <label>Include:</label>
          <div>
            <input type="checkbox" id="uppercase" checked> Uppercase Letters
            <input type="checkbox" id="lowercase" checked> Lowercase Letters
          </div>
          <div>
            <input type="checkbox" id="numbers" checked> Numbers
            <input type="checkbox" id="special" checked> Special Characters
          </div>
        </div>
        <div class="form-group">
          <label for="custom-input">Custom Words/Numbers to Include:</label>
          <input type="text" id="custom-input" placeholder="Enter words or
numbers (e.g., 123, myWord)">
        </div>
        <button type="button" id="generate-btn" class="btn">Generate
Password</button>
      </form>
      <div>
        <h3>Generated Password:</h3>
        <p id="password-output" class="output-box">Click "Generate
Password"!</p>
      </div>
    </div>

    <!-- Password Strength Analyzer Section -->
    <div id="password-analyzer" class="card">
      <h2>Password Strength Analyzer</h2>
      <form id="analyzer-form">
        <div class="form-group">
          <label for="password-input">Enter Password:</label>
          <input type="text" id="password-input" placeholder="Enter or paste
your password">
        </div>
```

```
        <button type="button" id="analyze-btn" class="btn">Analyze
Strength</button>
      </form>
      <div id="strength-results">
        <h3>Password Strength:</h3>
        <p id="strength-percentage"></p>
        <h4>Recommendations:</h4>
        <ul id="recommendations"></ul>
      </div>
    </div>
  </div>

  <script src="script.js"></script>
</body>
</html>
```

STYLES.CSS

```
/* General Styles */
body {
    font-family: 'Poppins', sans-serif;
    margin: 0;
    padding: 0;
    background-color: #121212;
    color: #e0e0e0;
  }

  .container {
    width: 90%;
    max-width: 600px;
    margin: 40px auto;
    text-align: center;
  }

  h1 {
    color: #00adb5;
    margin-bottom: 20px;
  }

  .card {
    background-color: #1e1e1e;
    padding: 20px;
    border-radius: 8px;
    margin-bottom: 20px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
  }
```

```css
h2 {
  color: #00adb5;
  margin-bottom: 15px;
}

.form-group {
  margin-bottom: 15px;
  text-align: left;
}

label {
  display: block;
  margin: 5px 0;
}

input[type="text"],
input[type="number"] {
  width: 100%;
  padding: 10px;
  margin-top: 5px;
  background-color: #2a2a2a;
  color: #e0e0e0;
  border: 1px solid #555;
  border-radius: 4px;
}

input[type="checkbox"] {
  margin-right: 10px;
}

.btn {
  display: inline-block;
  padding: 10px 20px;
  background-color: #00adb5;
  color: #121212;
  font-weight: bold;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}

.btn:hover {
  background-color: #02848c;
}

.output-box {
  background-color: #2a2a2a;
```

```
    padding: 10px;
    margin-top: 10px;
    border-radius: 4px;
    font-family: 'Courier New', Courier, monospace;
    word-break: break-all;
  }

  /* Password Strength Output */
  #strength-percentage {
    font-size: 1.2em;
    font-weight: bold;
    margin-top: 10px;
  }

  #strength-percentage.weak {
    color: #ff5252;
  }

  #strength-percentage.moderate {
    color: #ff9800;
  }

  #strength-percentage.strong {
    color: #4caf50;
  }

  #recommendations {
    list-style-type: disc;
    text-align: left;
    margin: 10px auto;
    padding: 0;
    color: #ff5252;
  }



  #recommendations li {
    margin: 5px 0;
  }
```

SCRIPT.JS

```
// Password Generator with Custom Input
document.getElementById('generate-btn').addEventListener('click', () => {
    const length = parseInt(document.getElementById('length').value);
    const includeUppercase = document.getElementById('uppercase').checked;
    const includeLowercase = document.getElementById('lowercase').checked;
```

```
    const includeNumbers = document.getElementById('numbers').checked;
    const includeSpecial = document.getElementById('special').checked;
    const customInput = document.getElementById('custom-input').value.trim();

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

    let allChars = '';
    if (includeUppercase) allChars += upper;
    if (includeLowercase) allChars += lower;
    if (includeNumbers) allChars += numbers;
    if (includeSpecial) allChars += special;

    if (!allChars && !customInput) {
      document.getElementById('password-output').textContent = 'Please select
at least one character set or provide custom input.';
      return;
    }

    let password = customInput; // Start the password with custom input
    const remainingLength = length - customInput.length;

    if (remainingLength < 0) {
      document.getElementById('password-output').textContent = 'Custom input
exceeds the specified password length.';
      return;
    }

    // Generate random characters to fill the remaining length
    for (let i = 0; i < remainingLength; i++) {
      const randomIndex = Math.floor(Math.random() * allChars.length);
      password += allChars[randomIndex];
    }

    // Shuffle the password to mix custom input and random characters
    password = password.split('').sort(() => Math.random() - 0.5).join('');

    document.getElementById('password-output').textContent = password ||
'Failed to generate password.';
  });

  // Password Strength Analyzer
  document.getElementById('analyze-btn').addEventListener('click', () => {
    const password = document.getElementById('password-input').value;

    // Criteria for password strength
```

```
    const hasUppercase = /[A-Z]/.test(password);
    const hasLowercase = /[a-z]/.test(password);
    const hasNumbers = /[0-9]/.test(password);
    const hasSpecialChars = /[\W_]/.test(password);
    const isLongEnough = password.length >= 8;

    // Calculate password strength percentage
    let strengthScore = 0;
    if (hasUppercase) strengthScore += 20;
    if (hasLowercase) strengthScore += 20;
    if (hasNumbers) strengthScore += 20;
    if (hasSpecialChars) strengthScore += 20;
    if (isLongEnough) strengthScore += 20;

    // Display strength percentage
    const strengthPercentageEl = document.getElementById('strength-
percentage');
    strengthPercentageEl.textContent = `Strength: ${strengthScore}%`;

    // Change color based on strength score
    strengthPercentageEl.className = '';
    if (strengthScore <= 40) {
      strengthPercentageEl.classList.add('weak');
    } else if (strengthScore <= 80) {
      strengthPercentageEl.classList.add('moderate');
    } else {
      strengthPercentageEl.classList.add('strong');
    }

    // Generate recommendations
    const recommendations = [];
    if (!hasUppercase) recommendations.push('Add at least one uppercase
letter.');
    if (!hasLowercase) recommendations.push('Add at least one lowercase
letter.');
    if (!hasNumbers) recommendations.push('Include at least one number.');
    if (!hasSpecialChars) recommendations.push('Add special characters (e.g.,
@, #, $).');
    if (!isLongEnough) recommendations.push('Increase the length to at least 8
characters.');

    const recommendationsEl = document.getElementById('recommendations');
    recommendationsEl.innerHTML = recommendations.length
      ? recommendations.map(rec => `<li>${rec}</li>`).join('')
      : '<li>Your password is strong and meets all criteria!</li>';
  });
```

CHAPTER 7

SCREENSHOTS

User page

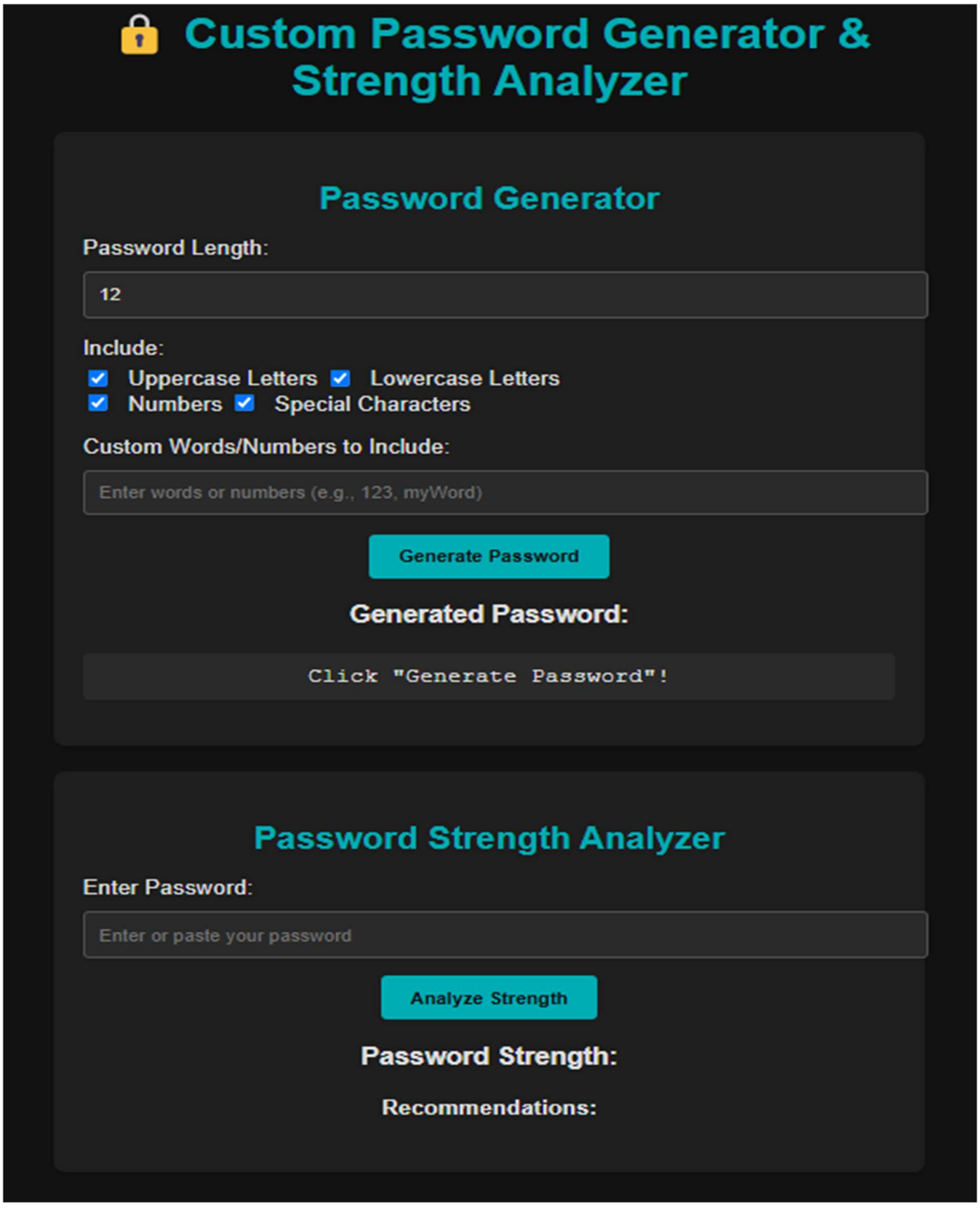


Fig 7.1

Password Generated

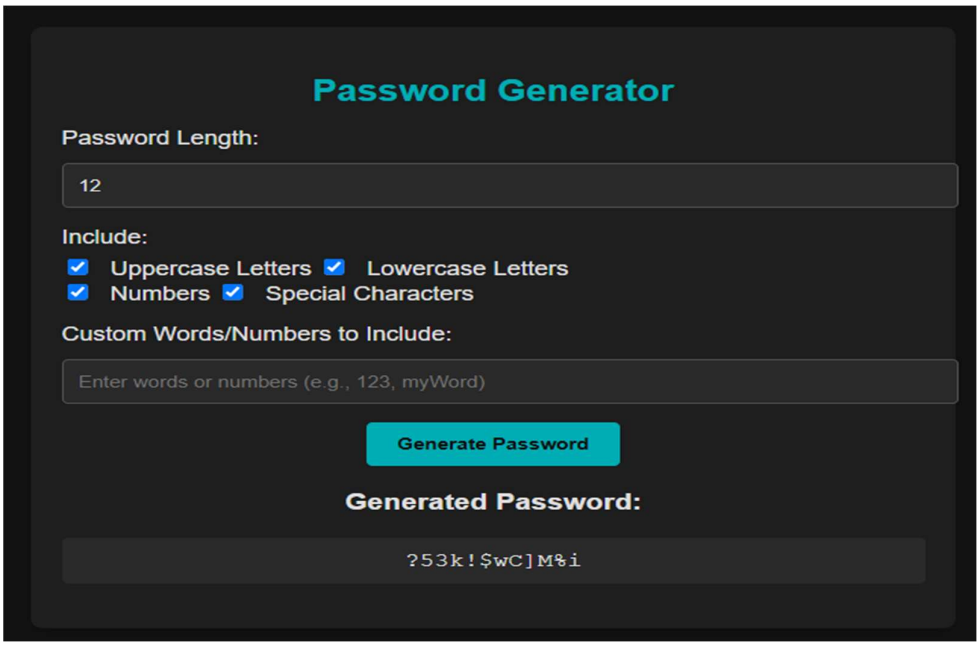


Fig 7.2

Analyzed the strength of a password

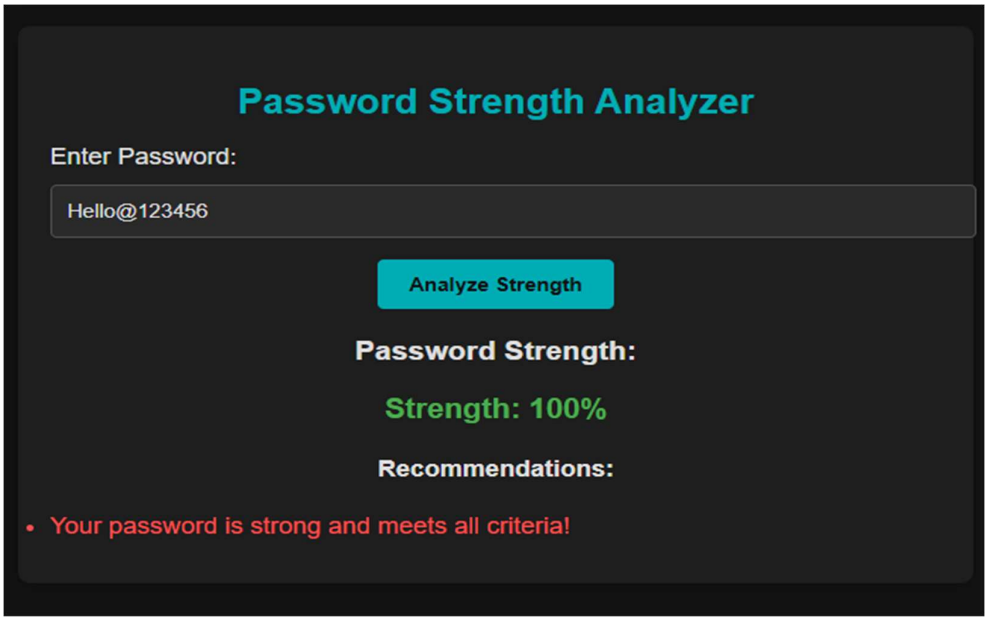


Fig 7.3

CHAPTER 8

CONCLUSION

In an era dominated by digital interactions and online services, the importance of strong password security cannot be overstated. The **Password Generator and Strength Analyzer Tool** addresses a pressing need for enhancing personal and organizational cybersecurity by offering a comprehensive, user-friendly solution for creating and evaluating passwords.

This project highlights the integration of robust security principles with accessible technology, empowering users to take control of their digital security. By combining a password generator and a strength analyzer into a single tool, the project ensures that users not only create secure passwords but also understand the characteristics of a strong password. This dual approach fosters better password hygiene and reduces the risk of cyberattacks such as hacking, phishing, and brute-force attempts.

The tool's functionality is built on well-established principles of password security. The generator module produces highly secure passwords based on user-defined criteria, including the length, the inclusion of specific character types, and custom inputs such as specific numbers or words. The strength analyzer evaluates both generated and user-input passwords, providing real-time feedback on their strength. This dynamic feedback educates users by highlighting weaknesses and recommending improvements, such as adding special characters or increasing the length of the password.

Aesthetics and user experience are central to the tool's design. The dark-themed interface ensures a modern and visually appealing experience while enhancing usability across devices. The tool's modularity and flexibility allow for future enhancements, such as the inclusion of password history tracking or integration with third-party authentication systems.

One of the most significant achievements of this project is its focus on privacy and security. By operating entirely on the client side, the tool ensures that sensitive user data is neither transmitted nor stored externally, addressing common concerns about the security of online

password management tools. This local-first approach makes the tool particularly suitable for individuals and organizations that prioritize data privacy.

The project also emphasizes education and awareness. By providing actionable recommendations for improving password strength, the tool helps users understand the importance of incorporating diversity, randomness, and adequate length into their passwords. This educational aspect contributes to long-term behavioral change, encouraging users to adopt stronger security practices in other areas of their digital lives.

While the tool achieves its primary objectives, it also lays the groundwork for future development. Potential enhancements include:

- Integration with cloud storage for securely saving passwords.
- Advanced analysis features that detect reused or weak passwords across multiple accounts.
- Support for multi-language interfaces to cater to a global audience.
- Incorporation of multi-factor authentication suggestions for enhanced security.

CHAPTER 9

REFERENCES

These references provide foundational knowledge for password generation, security standards, and web development practices used in this project.

1. **NIST Special Publication 800-63B**:
    - National Institute of Standards and Technology. (2017). *Digital Identity Guidelines: Authentication and Lifecycle Management*. Retrieved from https://nvlpubs.nist.gov/
2. **OWASP Password Policy Recommendations**:
    - Open Web Application Security Project. (2021). *Password Policy Recommendations*. Retrieved from https://owasp.org/
3. **Cryptographic Random Password Generators**:
    - Ferguson, N., Schneier, B., & Kohno, T. (2010). *Cryptography Engineering: Design Principles and Practical Applications*. Wiley Publishing.
4. **Web Development Technologies**:
    - Duckett, J. (2014). *HTML and CSS: Design and Build Websites*. Wiley Publishing.
5. **Cybersecurity Best Practices**:
    - Anderson, R. (2020). *Security Engineering: A Guide to Building Dependable Distributed Systems*. John Wiley & Sons.