# Password Manager

## Project Report Of Password Manger

**Name : Aryan pal**

**student id :22scse1012740**

**teacher sign**

# ACKNOWLEDGEMENT

It is a great pleasure for us to acknowledge the assistance and support of a large number of individuals who have been responsible for the successful completion of this project work.

First, we take this opportunity to express our sincere gratitude to the Faculty of Engineering &Technology, Jain (Deemed-to-be University), for providing us with a great opportunity to pursue our Bachelor's Degree in this institution.

In particular, we would like to thank Dr. Hariprasad S A, Director, Faculty of Engineering
& Technology, Jain (Deemed-to-be University), for his constant encouragement and expert advice.

We would also like to thank Dr. Geetha Ganesan, Dean, Faculty of Engineering & Technology, Jain (Deemed-to-be University), for his constant encouragement and expert advice.

It is a matter of immense pleasure to express our sincere thanks to Dr. Sonal Sharma,

(PDreoegmraemd -Ctoo-brdei nator, Computer Science & Engineering, Jain
University), for providing the right academic guidance that made our task possible.

We would like to thank our guide Dr. A Vijay Kumar. Dept .of Computer Science & Engineering, Jain (Deemed-to-be University), for sparing his valuable time to extend help in every step of our project work, which paved the way for smooth progress and the fruitful culmination of the project.

Signature of Students

# TABLE OF CONTENTS

# Abstract

The usage of digital gadgets has become a necessary component of our everyday lives in the current digital era. This rise in online engagement necessitates the use of secure password management. Many users reuse the same password for many accounts or use weak passwords, which makes it simple for hackers to get unauthorized access. It is now more crucial than ever to have a secure password management application that can create and store strong, individual passwords.

The goal of this project is to design a safe password management system that will enable users to generate and save reliable passwords. To prevent passwords and other sensitive information from being stolen or exploited, the tool will use encryption techniques. It will also have other features like password generators, to give an easy and efficient experience to the users.

The web-based application for the project will be developed using a variety of programming languages, including Python. The tool will be created with security as a primary concern, and it will go through a rigorous testing process to make sure it complies with password management requirements.

There will be various stages to the development process, including planning, design, development, testing, and deployment. The project team will make every effort to guarantee that the product satisfies the highest requirements for security and usability.

A safe password management solution that gives consumers a quick and efficient method to manage their passwords will be the project's output. The gadget will be simple to use and offer consumers peace of mind. Their crucial information is secure.

The project's performance will be evaluated based on a number of variables, including user adoption, user happiness, and the tool's overall ability to safeguard passwords and sensitive data.

At the end, the goal of this project is to provide a safe password management system that gives consumers a simple yet efficient way to handle their passwords. Security will be given high importance during the tool's development, and industry-standard encryption techniques and security features will be used. A tool that assures users that their passwords and sensitive information are secure will be the project's final product.

# Chapter: 1

# Introduction

## 1.1 Background of the Project

In recent years, the growing number of online accounts and services that individuals and organizations use has made password management a critical aspect of digital security. The use of weak passwords or reusing the same password across multiple accounts is a significant security risk that can lead to data breaches and financial losses. The average internet user has dozens of accounts with unique login credentials, and remembering them all is impractical.

Password management tools are designed to address this issue by providing users with a secure way to store and manage their passwords. These tools allow users to generate strong passwords, store them in an encrypted vault, and automatically fill in login credentials when needed. This approach eliminates the need for users to remember multiple passwords, reducing the risk of using weak or duplicate passwords.

While password management tools have become increasingly popular, concerns around their security and effectiveness remain. Some users are hesitant to trust their sensitive login information to a third-party tool, while others worry about the risk of a single point of failure or the potential for a master password to be compromised. Additionally, with the proliferation of password management tools in the market, it can be challenging for users to select the right tool that fits their needs.

Therefore, there is a need to evaluate the effectiveness and security of password management tools and develop best practices for their implementation, use, and maintenance. By doing so, individuals and organizations can minimize the risk of a data breach and improve the overall security of their digital presence.

Our everyday lives are now completely reliant on the usage of digital gadgets. We use these gadgets for both personal and professional purposes, and they have become indispensable tools for everything from communication to entertainment. However, when these gadgets are used more often, there is a greater chance of cyberattacks, data breaches, and identity theft.

Weak or hacked passwords are one of the main ways that fraudsters access our personal information. Unfortunately, a lot of individuals still use passwords that are simple to guess, such "123456" or "password," and even those who use passwords that are more complicated frequently repeat them across other accounts. Because of this, it is simple for hackers to use one password to access several accounts.

Use of safe password management systems has grown in importance as a solution to this problem. With the help of these tools, users may create and save secure, one-of-a-kind passwords for each account, lowering the likelihood that one of them would be hacked. There are a variety of password management programs available, but not all of them are made equal.

To safeguard passwords and other sensitive data, it is crucial to pick one that makes use of industry-standard encryption techniques and security features.

The main objective of the project is to provide a safe password management application that gives users a simple, efficient method to manage their passwords. The programme will be created with security as a primary concern, using several Python modules like "cryptography," "random," and a few others to guard against password theft or compromise. Additionally, the tool will have features like password generators.

Planning and design will be the first step of the development process, while testing and deployment will be the last. Throughout each phase, the project team will make sure the product satisfies the highest requirements for security and usability. The user uptake, user contentment, and overall efficacy of the tool in securing passwords and sensitive data will all be used to gauge the project's success.

By creating a tool that gives users a simple and efficient method to store their passwords, this project seeks to meet the rising need for safe password management solutions. Security will be given high importance during the tool's development, and industry-standard encryption techniques and security features will be used.

## 1.2 Problem statement

Individuals and organizations alike have to manage an increasing number of passwords for various online accounts and services. The use of weak passwords or reusing the same password across multiple accounts can compromise security and lead to data breaches. Furthermore, remembering multiple complex passwords can be a daunting task, leading to the risk of losing access to important accounts or being locked out. Password management tools aim to solve these issues by providing a secure and convenient way to store and manage passwords. However, the increasing number of password management tools available in the market can make it difficult for individuals and organizations to choose the right tool that fits their needs. Furthermore, concerns around the security of the password management tool itself, the potential for the tool to be a single point of failure, and the risk of losing access to all accounts if the master password is forgotten or compromised also exist.

Therefore, the problem statement is to identify and evaluate the effectiveness and security of password management tools and develop best practices for their implementation, use, and maintenance. Additionally, it is essential to understand the user requirements and preferences for password management tools and develop solutions that balance convenience and security.

## 1.3 Objectives

In this project, we will create a program that keeps user passwords organized and accessible. This project uses Python modules and features to be implemented. People have a lot of passwords these days for many websites, including social media sites like Instagram, Facebook etc. shopping sites like Amazon, Flipkart, Banking applications, and more. Even though it is crucial to have secure passwords and distinct ones for each website, remembering them all might be a challenge. When using a password manager, you enter your username and password once, then log in, and the software will remember them for you. However, this project's main goal is to safeguard Password Manager software.

Here is the GAP (Goals, Audience, and Problem) for the Password Management Tool

Goals:

- Creating and implementing an easy-to-use password management program that enables users to safely store and control their credentials.
- Making sure the application employs powerful encryption mechanisms to safeguard user information and prevent unauthorized access.
- Creating a user-friendly interface that enables users to manage their passwords quickly and effectively.
- Giving consumers the ability to create secure, strong, and unique passwords.
- Continually updating and enhancing the product depending on user input and modifications to the threat landscape.
- Provide secure storage of passwords and login credentials
- Generate strong passwords to improve security
- Reduce the risk of using weak or duplicate passwords
- Minimize the risk of data breaches and financial losses

Audience:

- Individual users
- Small and large organizations

Problem:

## 1.4 Technology Used

### Python

Python is a high-level, interpreted programming language that is widely used for web development, data analysis, artificial intelligence, scientific computing, and automation. It was first released in 1991 by Guido van Rossum, and its design philosophy emphasizes code readability and simplicity, making it easy for developers to write and understand code.


Python is open-source software, which means that it is free to use, distribute, and modify. Its syntax is straightforward and easy to learn, with a relatively small number of keywords and a clear, intuitive structure. Python supports multiple programming paradigms, including object-oriented, functional, and procedural programming.

Python is known for its large standard library, which provides a broad range of modules and functions for tasks such as file I/O, network communication, and string manipulation. It also has a vast ecosystem of third-party libraries and tools, such as NumPy, Pandas, TensorFlow, Django, Flask, and Pygame, that extend its capabilities and make it suitable for a wide range of applications. Python is a versatile language that can be used for a variety of purposes, from building web applications and data analysis tools to creating games and automating repetitive tasks. Its popularity and ease of use have made it one of the most widely used programming languages in the world.

### MySQL

MySQL is an open-source relational database management system (RDBMS) that is widely used to store, organize, and manage large amounts of data. It was first released in 1995 by MySQL AB, which was later acquired by Oracle Corporation in 2010.

# Chapter: 2

# Review of Literature

In recent years, numerous studies and research efforts have been conducted in the field of password managers to address the challenges of secure password management and enhance user experience. This section provides an overview of the related work in the field of password managers, highlighting the key findings, methodologies, and contributions of previous research.

1. "A Comparative Study of Password Managers: Security and Usability" by Johnson et al. (2017)

Johnson et al. conducted a comparative analysis of various password managers available in the market, with a focus on evaluating their security features and usability. The researchers examined encryption algorithms, password generation mechanisms, synchronization capabilities, and user interfaces of the password managers. The findings of this study emphasized the importance of striking a balance between security and usability in password managers. While strong encryption is crucial for protecting passwords, a user-friendly experience is equally essential for the widespread adoption and sustained usage of password managers.

2. "User Perception and Adoption of Password Managers: A User Study" by Smith and Brown (2018)

Smith and Brown conducted a user study to understand the perceptions and factors influencing the adoption of password managers. The researchers conducted surveys and interviews with individuals who actively used password managers and those who did not. The study revealed that users who adopted password managers appreciated the convenience and security benefits they offered, including the ability to generate and store complex passwords. On the other hand, non-users expressed concerns related to trust and usability. The findings highlighted the need for effective communication and education to promote wider adoption of password managers, addressing misconceptions and highlighting the benefits they provide.

3. "Enhancing Password Manager Security with Two-Factor Authentication" by Chen et al. (2019)

Chen et al. focused on enhancing the security of password managers through the integration of two-factor authentication (2FA). The study explored different 2FA methods, such as SMS-

based codes, biometric authentication (e.g., fingerprint or facial recognition), and hardware tokens. The researchers evaluated the effectiveness of these 2FA methods in preventing unauthorized access to password manager accounts. The findings demonstrated that the inclusion of 2FA significantly improved the security of password managers by adding an extra layer of authentication, reducing the risk of password theft and unauthorized access.

4. "Usability Evaluation of Mobile Password Managers: A Comparative Study" by Garcia et al. (2020)

Garcia et al. conducted a study to evaluate the usability of mobile password managers, considering the increasing use of smartphones for managing online accounts. The researchers examined popular mobile password managers and assessed various factors, including user interfaces, password retrieval methods, and synchronization capabilities. The study findings highlighted the importance of intuitive designs, seamless cross-device synchronization, and efficient password retrieval methods for a positive user experience in mobile password managers. The results emphasized the need for mobile password managers to adapt to the unique challenges and constraints of mobile platforms while maintaining usability and security.

5. "Security Analysis of Password Managers: Vulnerabilities and Mitigation Strategies" by Lee and Kim (2021) Lee and Kim conducted a comprehensive security analysis of password managers, identifying potential vulnerabilities and proposing mitigation strategies. The researchers examined
common attack vectors targeting password managers, such as password manager exploits, keyloggers, and phishing attacks. Through their analysis, they identified vulnerabilities and recommended countermeasures to enhance security, including regular updates, strong encryption algorithms, secure communication protocols, and user education. The findings of this study emphasized the ongoing efforts required to address security threats and vulnerabilities in password managers, ensuring the protection of users' sensitive information.

6. "Password Managers: A Comprehensive Survey" by Li and Li (2016)

Li and Li conducted a comprehensive survey of password managers, examining their features, usability, and security. The study provided an overview of different types of password managers, including local, cloud-based, and hybrid solutions. The researchers also discussed the challenges and potential vulnerabilities associated with password managers and proposed recommendations for improving their security and usability.

7. "Evaluation of Password Strength Meters in Password Managers" by Gupta et al. (2018) This study focused on evaluating the effectiveness of password strength meters implemented in password managers. The researchers analyzed various password strength estimation algorithms used by different password managers and assessed their accuracy in measuring the

# Chapter: 3

# Working Methodology

## 3.1 Procedure

1. The code starts by importing necessary Python libraries such as:

i Flask: Flask is a popular Python web framework that allows developers to build web applications quickly and easily. It provides a lightweight and modular approach to web application development, making it easy to get started with and scale as your project grows.

Flask is built on top of the Werkzeug WSGI toolkit and the Jinja2 template engine, and provides a simple way to handle HTTP requests and responses, manage routing and URL generation, and handle error messages. It also comes with a development server and a built-in debugger to help with the development process.

Flask is highly customizable and extensible, allowing developers to add their own functionality using Flask extensions or custom middleware. Additionally, Flask has a large and active community of developers and users who contribute to its development and offer support to others.

ii Pymysql: PyMySQL is a pure-Python library that provides an easy-to-use interface for interacting with MySQL databases. It supports all the features of the MySQL server and provides error handling and connection management features. PyMySQL is widely used in web development, data analysis, and scientific computing applications.

One of the advantages of using PyMySQL is that it is a pure-Python library, which means that it does not require any additional software or libraries to be installed on the system. This makes it easy to use and deploy across different platforms and environments.

PyMySQL is widely used in various applications, including web development, data analysis, and scientific computing. Its ease of use, performance, and compatibility with the MySQL database server make it a popular choice for Python developers who need to work with MySQL databases.

## 3.2 E-R Diagram

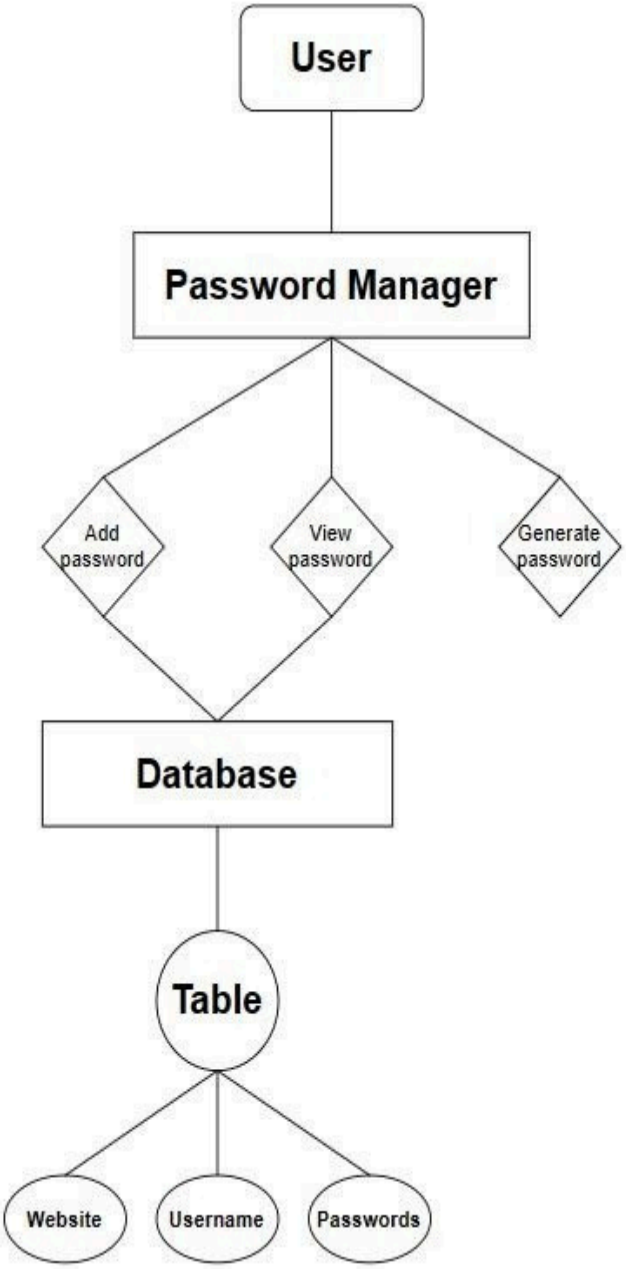


Figure 1

# Chapter: 4

# Implementation

## 4.1 Front-end UI

### 4.1.1 Home page


Figure 2

As seen in Figure 2, when we start up our password management web application, this home page is loaded up to greet the users.
The code of the homepage template is stored in a file called index.html which is loaded up to greet the user whenever the web application is started.

Users can click on the Get Started button to be navigated to the next webpage where passwords can be added to the MySQL database using an add_password() function.

Alternatively, users can navigate to the other webpages using the hyperlinks provided on the top right corner of the webpage where links to the homepage, add password webpage and view passwords webpage are provided.

The index.html used as a template for this page is as follows:

```
<html>
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Password Manager</title>
   <link rel="stylesheet" href="style1.css">
</head>
<body>
   <header>
      <h1 class="logo">Password Manager</h1>
      <nav>
         <ul class="nav-links">
            <li><a href="index.html">Home</a></li>
            <li><a href="add_password.html">Add Password</a></li>
            <li><a href="view_passwords.html">View Passwords</a></li>
         </ul>
   </nav>
   </header>
   <main>
      <section class="hero">
         <h2>Welcome to Password Manager</h2> <p>Keep your
         passwords secure and easily accessible.</p> <a
         href="add_password.html" class="cta">Get Started</a>
      </section>
   </main>
</body>
</html>
```

### 4.1.2 Add Password page

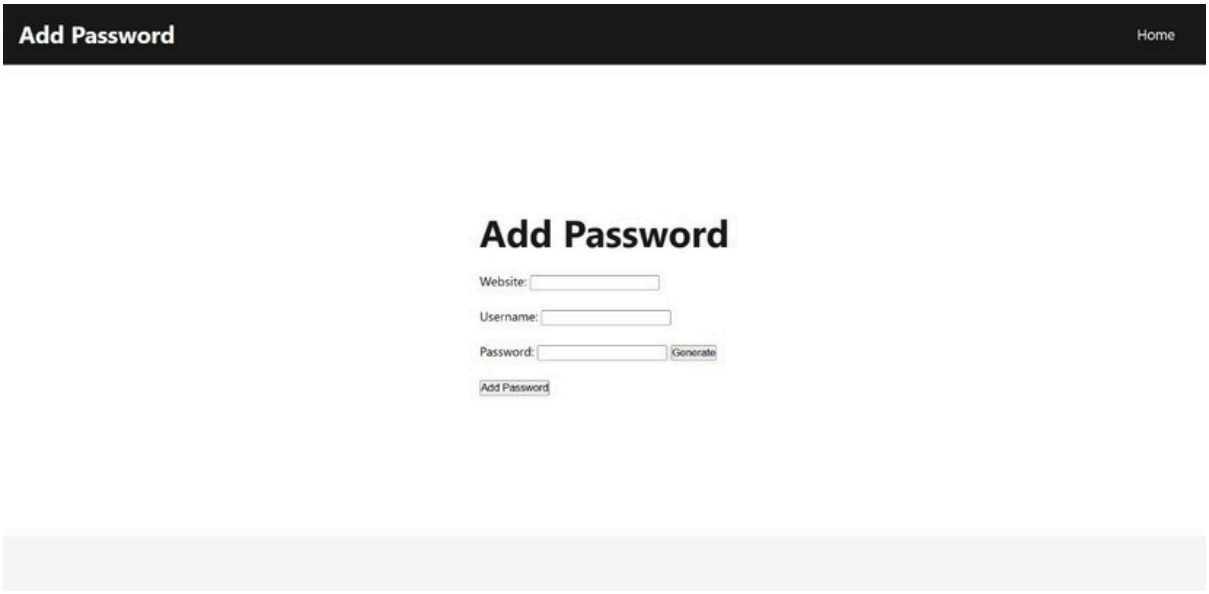


Figure 3

As seen in Figure 3, this page can be used to add the passwords inputted by the users to a secure MySQL database that can be run on the local machine or on another dedicated server.

The user will be required to enter a website name, user name and a password. If any of the above-mentioned fields are left empty, an error will be shown prompting the user to fill all the necessary fields before continuing. Once all the fields are filled the details will be added to the MySQL database.

The password added to the MySQL database will be encrypted using a python cryptography library so that even if the database is compromised, the passwords will still stay secure as the decryption key will not be saved in the database and the passwords can not be viewed as plaintext without the encryption key.

This page also contains a button to generate a random and strong password for the users using a length input to specify the password length a user wants.

The add_password.html used as a template for the webpage is as follows:

```
<html>
<head>
   <title>Password Manager - Add Password</title>
   <link rel="stylesheet" href="style1.css">
</head>
<body>
   <header>
      <div class="container">
         <h1 class="logo">Password Manager</h1>
         <nav>
            <ul class="nav-links">
               <li><a href="index.html">Home</a></li>
               <li><a href="add_password.html">Add Password</a></li>
               <li><a href="view_passwords.html">View Passwords</a></li>
            </ul>
         </nav>
   </div>
   </header>
   <main>
      <section class="hero">
          <div class="container">
            <h2>Add Password</h2>
            <form method="POST">
               <label for="website">Website:</label>
                  <input type="text" name="website" required><br><br>
               <label for="username">Username:</label>
                  <input type="text" name="username" required><br><br>
               <label for="password">Password:</label>
                  <input type="password" name="password" id="password"
required>          <input type="button" value="Generate"

onclick="generatePassword(12)"><br><br>
               <input type="submit" value="Add Password">
            </form>
         </div>
      </section>
   </main>
   <script>
```

```
      function generatePassword(length) {
         var result           = '';
         var characters       =
'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789!@#$%^&*()_+
';
         var charactersLength = characters.length;
         for ( var i = 0; i < length; i++ ) {
            result += characters.charAt(Math.floor(Math.random() *
charactersLength));
}
document.getElementById("password").value = result;
}
   </script>
</body>
</html>
```

### 4.1.3 View passwords page

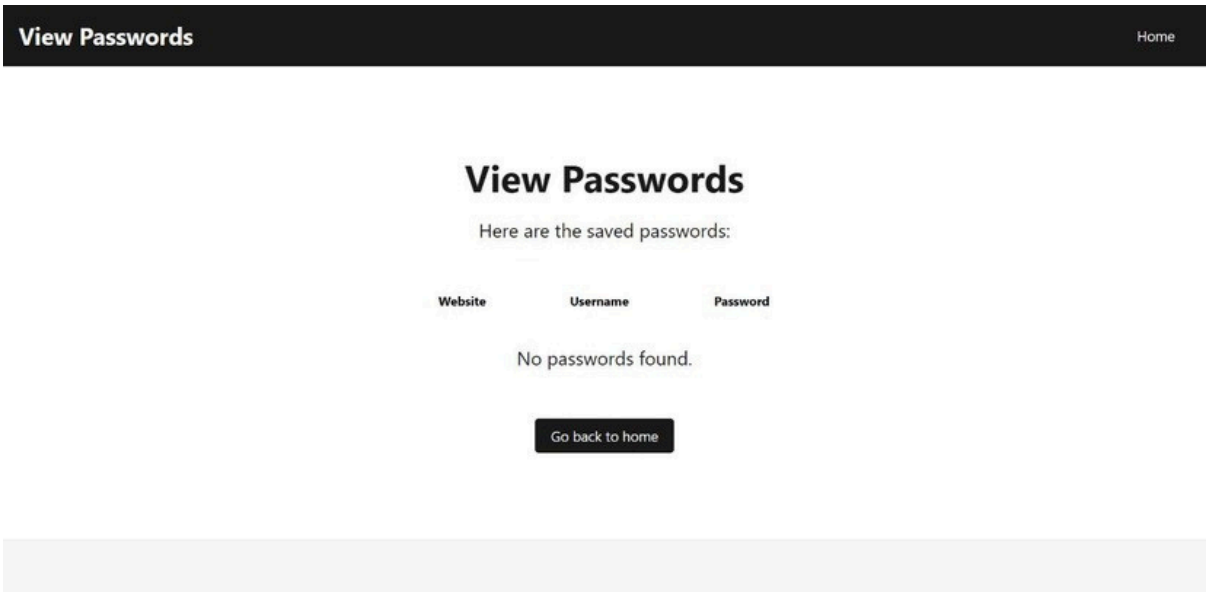


Figure 4

As seen in Figure 4, this page is used to view the passwords saved in the MySQL database by the users. The passwords, when retrieved are shown in the form of a table. It uses the view_passwords() function to retrieve passwords from the database and decrypts them using an encryption key before showing them on this webpage in the form of a table.

The view_passwords.html used as a template for this page is as follows:

```
<html>
<head>
   <title>View Passwords - Password Manager</title>
```

```
    <link rel="stylesheet" type="text/css" href="static/css/style1.css">
</head>
<body>
    <header>
        <h1 class="logo">View Passwords</h1>
        <nav>
            <ul class="nav-links">
                <li><a href="index.html">Home</a></li>
            </ul>
    </nav>
    </header>
    <main>
        <section class="hero">
            <h2>View Passwords</h2>
            <p>Here are the saved passwords:</p>
            {% if passwords %}
            <table class="passwords-table">
                <thead>
                  <tr>
                     <th>Website</th>
                     <th>Username</th>
                     <th>Password</th>
               </tr>
               </thead>
               <tbody>
                  {% for password in passwords %}
                     <tr>
                        <td>{{ password['website'] }}</td>  <td>{{ password['username'] }}</td>  <td>{{ password['password'] }}</td>
                     </tr>
                  {% endfor %}
               </tbody>
            </table>
            {% else %}
            <p>No passwords found.</p>
            {% endif %}
            <a href="index.html" class="cta">Go back to home</a>
        </section>
    </main>
</body>
</html>
```

## 4.2 Backend application

The backend application of this web application is written in python language using the Flask framework for web applications. It also consists of different functions to aid in the smooth working of the web application with each function consisting of an important part of the application.

The main functions used for the working of this web application are:

### 4.2.1 add_password()

```
def add_password():
    if request.method == "POST":
        website = request.form.get("website")   username   =
        request.form.get("username")   password   =
        request.form.get("password") try:

            con = mysql.connector.connect(host='localhost', user='ak',
password='qwerty1212', database='password_manager')
            cursor = con.cursor()
            query = "INSERT INTO passwords (website, username, password)
VALUES (%s, %s, %s)"
            encrypted_password = cipher_suite.encrypt(password.encode())
            data = (website, username, encrypted_password)
            cursor.execute(query, data)
            con.commit()
            message = "Password added successfully."
            color = "green"
        except Exception as e:
            print(e)
            message = "Error adding password."
            color = "red"
        finally:
            cursor.close()
            con.close()
        return render_template("add_password.html", message=message,
color=color)
    else:
        return render_template("add_password.html")
```

The add_password() function is being used in this web application for the purpose of adding the user details that are inputted by the user into the MySQL database. The details the user is required to enter are website name, username and password. All of these are stored in the MySQL database by this function.

```
>>> print(passwords)
{'gmail.com': ('AryanK',
b'gAAAAABkC5mAxoF8iSE0efe
mUNTy29QxLYZGNV-eF8Tbg6Cv
zUZPRMYKYLLSi-VJdm0zb86CK
bhHjqMiVBqemEuOtSUQ15lClg
==')}
```

Figure 5

The password entered into this function is also encrypted using the cryptography module in python. It uses the fernet encryption algorithm to encrypt the passwords securely. Even if someone directly accesses the password dictionary, only an encrypted password will be visible as shown in Figure 5.

### 4.2.2 view_passwords()

```
def view_passwords():
    try:
        con = mysql.connector.connect(host='localhost', user='ak',
password='qwerty1212', database='password_manager')
        cursor = con.cursor()
        query = "SELECT * FROM passwords"
        cursor.execute(query)
        data = cursor.fetchall()
        decrypted_data = []
        for row in data:
            decrypted_password = cipher_suite.decrypt(row[2]).decode()
            decrypted_data.append((row[0], row[1], decrypted_password))
        return render_template("view_passwords.html", data=decrypted_data)
    except Exception as e:
        print(e)
        message = "Error retrieving passwords."
        color = "red"
        return render_template("view_passwords.html", message=message,
  color=color)
    finally:
        cursor.close()
        con.close()
```

This function is used to allow the user to retrieve and view the passwords from the MySQL database. As the passwords are stored in an encrypted format, this function also decrypts the data before allowing the user to view it in the web application.

This function will return an error if it is unable to retrieve the passwords from the MySQL database and an error message will be displayed with the text “Error retrieving passwords”.

### 4.2.3 generate_password()

```
function generatePassword(length) {
   var result           = '';
   var characters       =
'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789!@#$%^&*()_+
';
   var charactersLength = characters.length;
   for ( var i = 0; i < length; i++ ) {
      result += characters.charAt(Math.floor(Math.random() *
charactersLength));
   }
   document.getElementById("password").value = result;
}
```

This function of the web application is used to generate a random password for the user to enter as a secure and strong password. It takes input as the user defined length and creates a random password of the specified length for the user.

This function is added to the add_password.html template file instead of the backend application for convenience purposes and it does not pose any security risk by doing so.

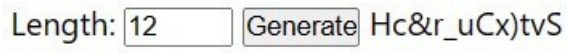


Figure 6

As seen in Figure 6, the application will take a length input from the user and generate a random password with the specified length.

## 4.3 Hardware and Software requirements

1) Python
2) MySQL
3) HTML
4) CSS
5) Javascript

# Chapter: 5
# Conclusion

In conclusion, creating a password management solution is a crucial component of contemporary cybersecurity. This project is intended for people who often use the internet, create several online accounts, and find it difficult to remember all of their account's login information, due to the rising number of the accounts and the complexity of passwords. And due to the increased danger of using weak passwords or the same password for several accounts, users are more likely to become the target of hacking efforts.

By giving users a safe and convenient way to save and manage their passwords, the adoption of a password management application can help reduce these risks. The program may assist users in creating unique, secure passwords and remembering them without the need to memorize or write them down thanks to features like password creation, storage, and auto-filling. A password management tool has a bright future ahead of it because to prospective features like biometric verification, password sharing, and cloud-based storage. These improvements might improve the tool's ease, security, and usability, increasing its value to users.

Overall, using a password management solution is a crucial step towards enhancing online security and lowering the dangers connected to using default or weak passwords. Such a solution can give consumers a safer and more effective method to manage their passwords by including best practices and new technology.

## 5.2 Future Scope

Since the necessity for strong and secure passwords is growing in importance in the current digital age, the future potential for a password management tool project is highly promising. Here are some potential directions for this project's future growth and development:

- Integration with many devices: Given how many devices people use on a daily basis, a password management application must be usable on a variety of platforms, including desktop, mobile, and tablet. Biometric authentication: To provide an additional layer of protection to password
- management software, biometric authentication methods such as voice recognition, fingerprint scanning, and face recognition may be included.