# DATA GOVERNANCE AND SECURITY

**A Project Report**

*Submitted in the partial fulfillment for the award of the degree of*

**BACHELOR OF ENGINEERING**

**IN**

**Big Data Engineering**

**Submitted by:**

**Abhinav Sharma (20BCS3873)**
**Kanwar Arinjai Singh (20BCS3908)**
**Anupam Kumar (20BCS3945)**

**Under the supervision of:**
**MS. NAVJEET KAUR**

**CHANDIGARH UNVERSITY, GHARUAN,**

**MOHALI, PUNJAB**

CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

## BONAFIDE CERTIFICATE

Certified that this project report **“Data Governance and Security”** is the bonafide work of “**Abhinav Sharma, Kanwar Arinjai Singh and Anupam Kumar”** who carried out the project work under our supervision.

**SIGNATURE**

**SIGNATURE**

**HEAD OF THE DEPARTMENT**

**SUPERVISOR**

Submitted for the project viva- voce examination.

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

# ACKNOWLEDGEMENT

**I would like to express our deep and sincere gratitude to our Project respected supervisor Navjeet Kaur for giving us the opportunity to do the project and providing valuable guidance throughout this research. Their dynamism, vision and exquisite effortshave deeply inspired us. They taught us the methodology to carry outthe research and to present the research work as clearly as possible.**

**It was a great privilege for us to study and work under their guidance. We owe the completion of my project to our project Mentor for his continuous support and guidance**

# LIST OF FIGURES

# LIST OF TABLES

# TABLE OF CONTENTS

# ABSTRACT

This project is a comprehensive effort to enhance data governance and security within the organization. It involves assessing and categorizing data assets, developing robust policies and frameworks, implementing stringent access controls and encryption techniques, establishing real-time monitoring and auditing mechanisms, and conducting employee training programs. Additionally, the project includes the development of an incident response plan, ensuring compliance with data protection regulations, and a commitment to continuous improvement through regular evaluations. The ultimate goal is to safeguard sensitive data, mitigate security risks, ensure compliance, and bolster stakeholder trust in data handling practices.

The end result is  a password manager system which enforces best practices like password length and complexity, minimizing the use of weak passwords. Password managers store all passwords in one centralized location, making it easier to manage and update credentials. This reduces the reliance on memory or unsecured methods for storing passwords.

# INTRODUCTION

## 1.1 Identification of Contemporary issue

Solving data governance and security problems is imperative to protect sensitive information, maintain user trust, and comply with regulations. Effective solutions mitigate the risk of data breaches, safeguard against cyber threats, and ensure the integrity of data. This fosters a secure environment, enabling reliable decision-making and sustained project success while avoiding legal consequences.

1. Data Breaches and Cybersecurity Threats:

The frequency and sophistication of cyber attacks continue to rise, posing a significant threat to the security of sensitive data. Projects need to implement robust cybersecurity measures to safeguard against unauthorized access, data breaches, and other cyber threats.

2. Data Privacy Compliance:

With the introduction of regulations like the General Data Protection Regulation (GDPR) and others in various regions, ensuring compliance with data privacy laws has become a critical issue. Projects must navigate complex regulatory landscapes to avoid legal consequences and maintain user trust.

3. Third-Party Data Sharing and Outsourcing Risks:

Many projects involve collaboration with third-party vendors or outsourcing partners, introducing additional risks to data security. Managing and securing data shared with external entities is challenge that needs to be addressed to prevent data unauthorized.

4. Data Ownership and Accountability:

Defining and establishing clear data ownership and accountability within a project is crucial. Without a clear understanding of who owns the data and who is responsible for its security, there can be confusion and potential gaps in governance.

5. Data Governance Frameworks:

Establishing and maintaining a robust data governance framework is an ongoing challenge. This includes defining data policies, roles and responsibilities, and implementing mechanisms for monitoring and enforcement.

## 1.2 Identification of Problem

The project's primary focus is on addressing challenges related to data governance and security within the organization's data ecosystem. The proliferation of data sources, increasing data volumes, and the need to ensure data privacy and compliance have led to several critical issues that require resolution. The organization lacks a robust system for maintaining data quality and consistency across various data sources and databases. Inaccurate, incomplete, or inconsistent data hampers decision-making processes and undermines trust in the data. There is a lack of proper access controls and permissions management for data stored within the organization. Unauthorized access to sensitive data poses a significant security risk and could lead to breaches of confidentiality. The organization struggles to meet data privacy regulations and industry-specific compliance standards (such as GDPR, HIPAA, etc.). Non-compliance could result in legal penalties and damage to the organization's reputation.

## 1.3 Identification of Tasks

In a data governance and security project with a focus on implementing a password manager system, several critical tasks need to be identified and executed to ensure the robust protection of sensitive information. The project can be broken down into key components, each addressing specific aspects of data security and governance.

1. Design and Implementation of Password Manager:

The core task revolves around designing and implementing the password manager system. This includes selecting or developing a robust password management tool that adheres to industry best practices. The design phase should consider encryption standards, multi-factor authentication, and user-friendly interfaces to encourage widespread adoption.

2. Data Classification and Policy Development:

To govern data effectively, it's essential to classify information based on sensitivity. Data governance tasks involve defining and implementing policies specifying who has access to what data, and under what conditions. This includes determining password complexity requirements, access controls, and establishing protocols for periodic password updates.

3. Continuous Monitoring and Auditing:

Post-implementation, ongoing monitoring and auditing are critical tasks. Regularly reviewing access logs, conducting security audits, and analyzing potential threats ensure the system's integrity. Continuous improvement based on these assessments helps adapt security measures to emerging risks and vulnerabilities.

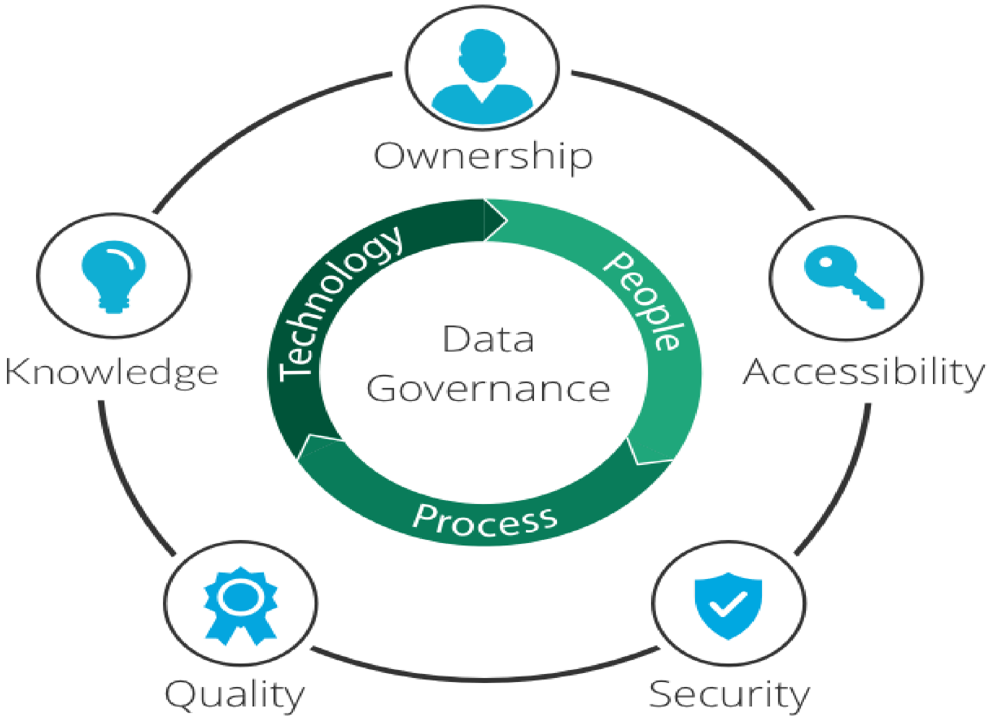


Figure1

# LITERATURE SURVEY

## 2.1 Bibliometric Analysis

Data governance and security are critical components of any organization that collects, stores, and utilizes data. In today's data-driven world, organizations are increasingly reliant on data to make informed decisions, drive innovation, and gain a competitive advantage. However, this reliance on data also introduces significant risks, as data breaches and privacy violations can have devastating consequences.

Data governance is the set of policies, procedures, and practices that ensure data is managed effectively and responsibly throughout its entire lifecycle. It encompasses data quality, data access controls, data lifecycle management, and data security. Data security, on the other hand, is the practice of protecting data from unauthorized access, use, disclosure, disruption, modification, or destruction.

Real-world scenarios of data governance and security projects

- A healthcare provider implements a data governance program to ensure patient data is protected and used appropriately. This program includes policies and procedures for data classification, access control, and data retention.
- A financial institution implements a data security project to protect customer financial information from cyberattacks. This project includes implementing firewalls, intrusion detection systems, and data encryption.
- A retail company implements a data governance program to improve data quality and ensure data is used for analytics and decision-making. This program includes data cleansing and validation processes.

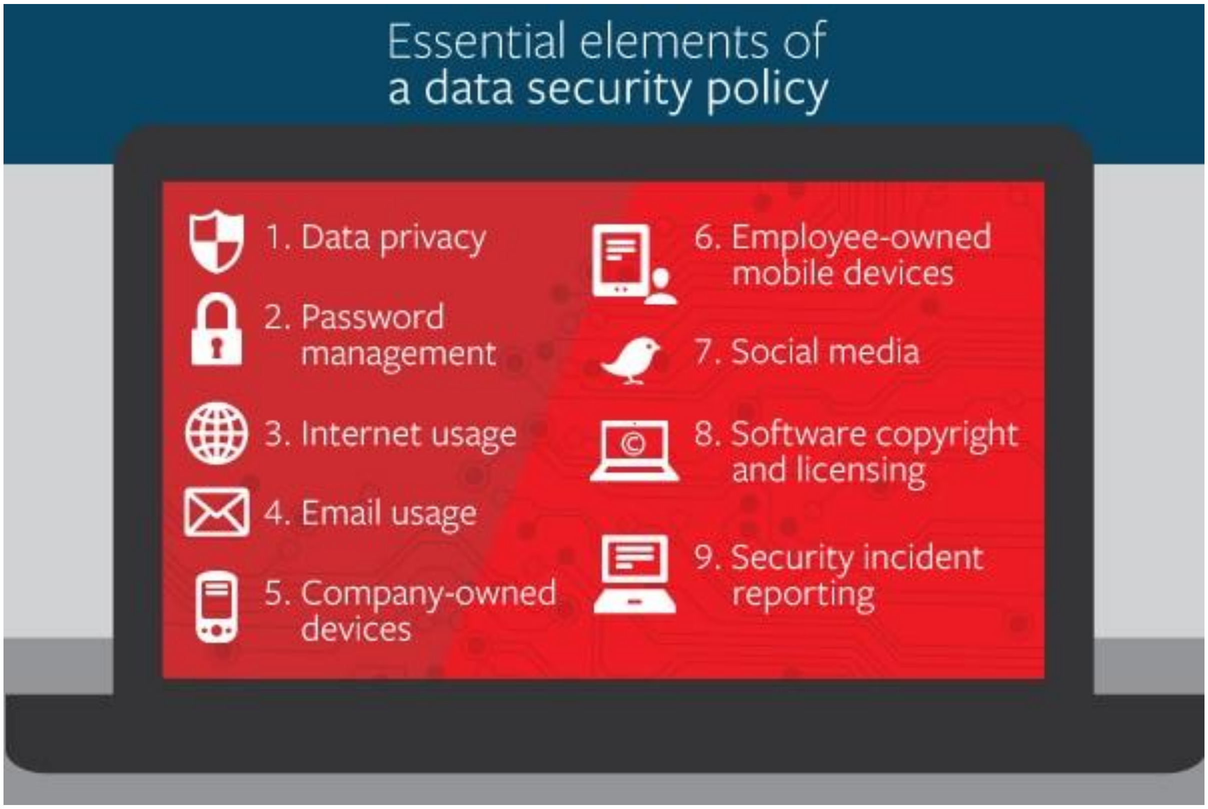


Figure 2

Recent publications in the field of data governance and security

- "Data Governance: The Key to Data Quality" by SAP
- "Data Governance and Security: Implementing Policies and Practices to Protect Data Integrity and Privacy" by Talent500
- "Data Governance: The Missing Piece of the Digital Puzzle" by Gartner
- A study by Alharbi et al. (2021) found that the most common topics in papers on data governance in cloud computing were data ownership, data security, and data quality. Another study by Zhang et al. (2022) found that the most common topics in papers on data governance in smart cities were data ownership, data sharing, and data privacy.
- The SME Quandary (2011) by Begg et al. This study examined the challenges of implementing data governance in small and medium-sized enterprises (SMEs). The study found that SMEs often lack the resources and expertise to implement data governance effectively.

- Data Governance for Security in IoT & Cloud Converged Environments (2016) by Al-Ruithe et al. This study investigated the challenges of data governance in IoT and cloud-based environments. The study found that these environments pose new challenges for data governance, such as the need to manage data across multiple platforms and the need to protect data from cyber threats.
- Towards a data governance framework for third generation platforms (2019) by Juan Yebenes. The purpose of this study is to meet the performance and security needs of various processes by examining concerns pertaining to cloud computing resource utilization.
- Designing data governance (2010) by Vijay Khatri. The aim of this paper is to present a general framework for data governance that may be utilized by practitioners to create an efficient data governance strategy, approach, and design, and by researchers to concentrate on significant data governance challenges.

These publications provide insights into the challenges and opportunities of data governance and security, as well as best practices for implementation.

Benefits of data governance and security projects

- Improved data quality
- Enhanced data security
- Increased compliance with data regulations
- Improved decision-making
- Increased customer trust

Challenges of data governance and security projects

- Gaining executive buy-in
- Changing organizational culture
- Integrating data governance and security into existing processes

- Keeping up with evolving data security threats

Recommendations for successful data governance and security projects

- Start with a clear understanding of the business goals.
- Involve key stakeholders from across the organization.
- Develop a comprehensive data governance framework.
- Implement appropriate security controls.
- Continuously monitor and update the data governance and security program.

Data governance and security are critical for any organization that collects, stores, and utilizes data. By implementing effective data governance and security programs, organizations can protect their data assets, improve decision-making, and build trust with their customers.

## 2.2 Problem Definition

In an era marked by the proliferation of digital data, organizations find themselves in a precarious position. While data has become an invaluable asset that drives decision-making, innovation, and competitiveness, it has also ushered in a new era of unprecedented vulnerabilities and risks. Data breaches, privacy violations, regulatory non-compliance, and unauthorized access pose imminent threats, compromising an organization's integrity, customer trust, and financial stability. In this context, the problem at hand is the pressing need to enhance data governance and security within our organization comprehensively.

Firstly, organizations grapple with the sheer volume and diversity of data they generate and collect. The exponential growth of data, both structured and unstructured, has made it a daunting task to classify, organize, and manage this data effectively. The problem lies in not only understanding what data an organization possesses but also determining its relevance, quality, and potential risks.

Second, the complexity of regulations makes data governance initiatives much more urgent. Organizations are subject to strict guidelines for data management, storage, and sharing under data protection and privacy laws like GDPR, HIPAA, and CCPA. Serious fines and reputational harm may arise from noncompliance. Data encryption is a fundamental aspect of data security, but its complexities are a persistent issue. Effective encryption requires not only selecting appropriate encryption algorithms but also managing encryption keys securely. The challenge is implementing encryption without introducing latency or operational bottlenecks, while still guaranteeing data confidentiality.

Real-time monitoring and auditing of data access and usage are critical for identifying and mitigating security incidents. However, many organizations struggle with the implementation of comprehensive monitoring tools and effective auditing processes, leaving them vulnerable to undetected threats.

The human element introduces yet another dimension to the problem. Employees, while essential to an organization's success, can unintentionally expose sensitive information due to a lack of awareness and understanding of data governance and security best practices. The challenge is to provide continuous training and create a culture of data security awareness.

Finally, companies frequently don't have clear incident response procedures, which makes them unprepared to deal with security breaches. The effect may worsen in the absence of a well-defined plan for locating, stopping, and immediately addressing data breaches.

The problems outlined above collectively contribute to a high level of risk and inefficiency in the organization's data management practices. These issues not only pose a threat to data security but also hinder the organization's ability to leverage data as a strategic asset. Addressing these problems through the Data Governance and Security Project is essential to mitigate risks, enhance data integrity, maintain compliance, and empower the organization to make informed decisions based on reliable data.

In this context, the problem statement encompasses the multifaceted challenge of governing and securing data amidst exponential growth, complex regulations, evolving access control and encryption requirements, monitoring deficiencies, employee vulnerabilities, and the need for continuous improvement. Solving this problem requires a comprehensive approach that addresses organizational culture, policies, processes, and technology.

## 2.3 Goals/Objectives

In today's increasingly interconnected and digital world, the importance of password security cannot be overstated. Password managers have emerged as essential tools for individuals and organizations alike, offering a comprehensive solution to managing passwords effectively and securely.

At the core of password management systems lies the primary objective of enhancing password security. Password managers help users create, store, and manage strong, unique passwords for all their online accounts, significantly reducing the risk of password-related compromises. By eliminating the need to remember multiple complex passwords, password managers also improve password usability, making it easier and more convenient for users to access their online accounts.

Another crucial goal of password management systems is to mitigate the risks associated with weak or reused passwords. Weak passwords, such as those that are easily guessable or consist of common words, can be easily cracked by cybercriminals, granting them unauthorized access to sensitive data. Password managers address this issue by encouraging users to create strong, unique passwords for each account, preventing the domino effect of a single compromised password exposing multiple accounts.

Protecting sensitive data is another overarching objective of password management systems. By securely storing passwords and other sensitive information in an encrypted vault, password managers ensure that this critical data remains protected from unauthorized access. This protection extends to mitigating the risk of phishing attacks, where cybercriminals attempt to trick users into revealing their login credentials by posing as legitimate websites or services.

Password managers also aim to streamline the process of creating, managing, and accessing passwords, saving users time and effort. Instead of juggling multiple passwords or relying on insecure methods like writing them down, password managers provide a centralized and secure repository for all passwords, accessible with a single master password or through biometric authentication.

Promoting password hygiene is another critical objective of password management systems. Password hygiene refers to the practices and habits that individuals and organizations adopt to maintain strong password security. Password managers encourage better password practices by prompting users to create strong passwords, avoiding password reuse, and enabling two-factor authentication, which adds an extra layer of security beyond just a password.

Enhancing overall cybersecurity is the goal of password management systems. By reducing the attack surface associated with weak or reused passwords, password managers play a vital role in strengthening an organization's cybersecurity posture. Additionally, password managers can integrate with other cybersecurity measures, such as firewalls and intrusion detection systems, to provide a comprehensive approach to protecting sensitive data and preventing unauthorized access.

Data governance and security also contribute to fostering a culture of data awareness and responsibility within the organization. This involves educating employees about the importance of data protection, privacy, and compliance. Training programs and awareness campaigns can help reduce the risk of human errors and insider threats.

Ultimately, the goals of data governance and security are intertwined, working together to create a resilient and well-managed data environment. By establishing clear policies, implementing robust security measures, and fostering a data-aware culture, organizations can harness the full value of their data assets while mitigating risks and ensuring regulatory compliance.

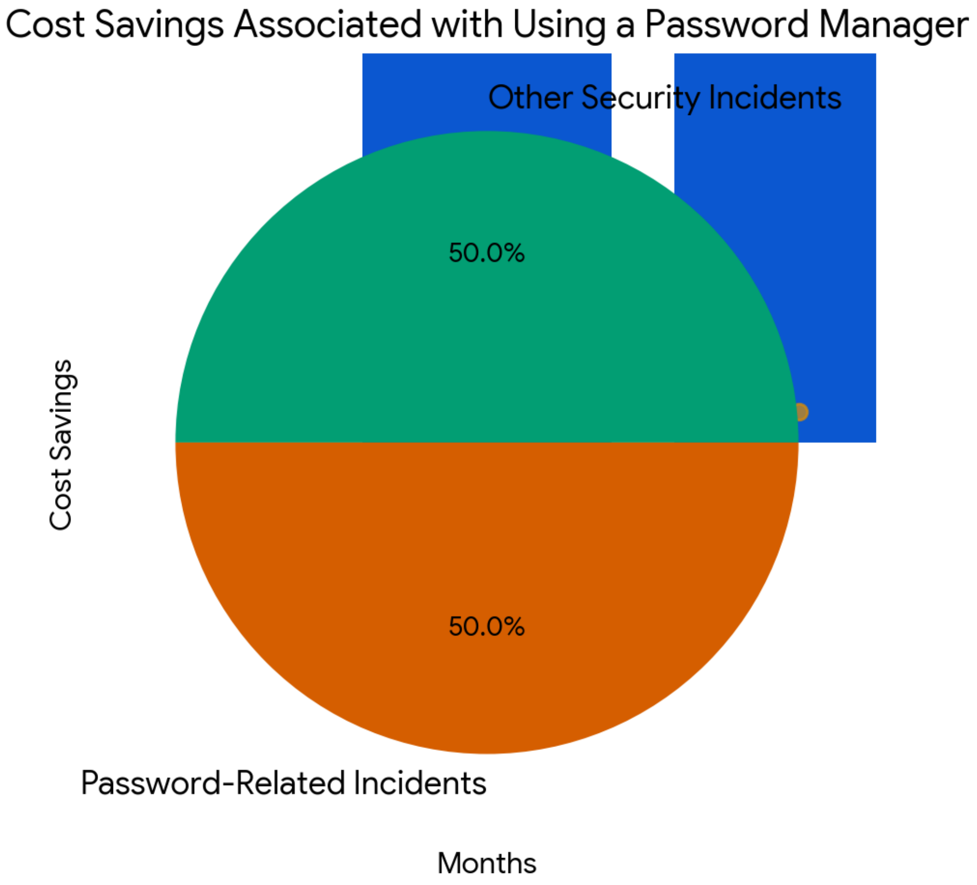

Cost Savings Associated with Using a Password Manager
Other Security Incidents
50.0%
Cost Savings
50.0%
Password-Related Incidents
Months

Figure 3

# DESIGN AND FLOW PROCESS

## 3.1 Evaluation & Selection of Specifications/Features

Evaluating and selecting a password management system (PMS) is a crucial decision for individuals and organizations alike. With a vast array of options available, it's essential to carefully consider the specific needs and requirements to ensure the chosen PMS effectively addresses password security and usability concerns.

Key Considerations for PMS Evaluation:

1. Security:

- Encryption: Ensure the PMS uses strong encryption algorithms to protect stored passwords and sensitive data from unauthorized access.
- Multi-Factor Authentication (MFA): Prioritize PMSs that support MFA, adding an extra layer of security beyond just a password.
- Password Generation: Choose a PMS that can automatically generate strong, unique passwords for new accounts.
- Password Sharing: Assess the secure password sharing capabilities if the need arises.
- Password Auditing: Consider PMSs that offer password auditing tools to identify and remediate weak or reused passwords.

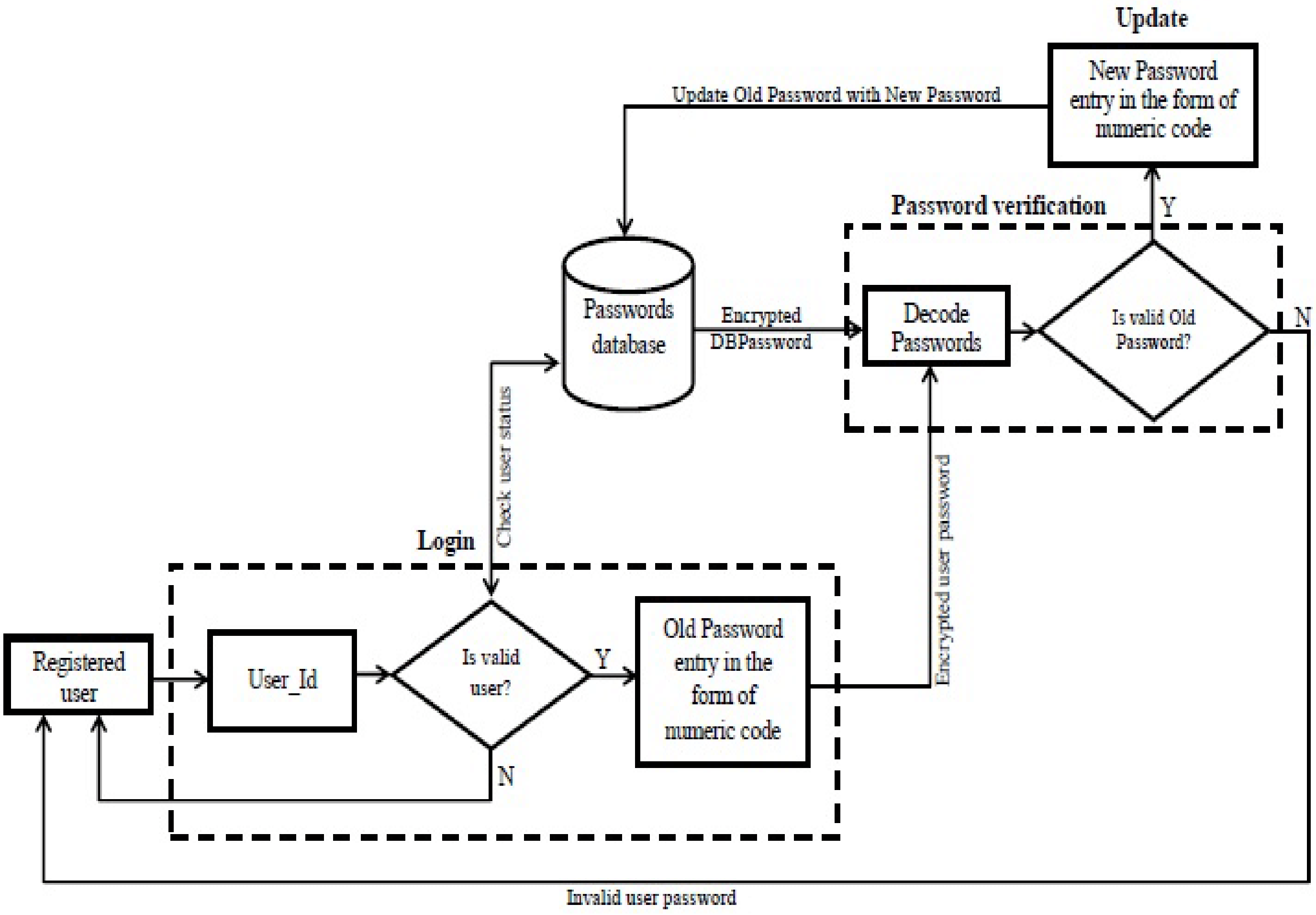
2. Usability:

- Cross-Platform Compatibility: Select a PMS that works seamlessly across various devices, including desktops, laptops, smartphones, and tablets.
- User Interface (UI): Choose a PMS with an intuitive and user-friendly interface that simplifies password management.
- Synchronization: Opt for a PMS that can synchronize passwords across multiple devices, ensuring consistent access.
- Password Autofill: Evaluate PMSs that offer autofill functionality for convenient logins.
- Ease of Use: Prioritize PMSs that are easy to set up and use, minimizing the learning curve.

3. Additional Features:

- Password Import/Export: Consider PMSs that allow importing and exporting passwords for seamless transitions.

- Password History: Evaluate PMSs that maintain a history of past passwords for recovery purposes.
- Secure Notes: Assess the PMS's ability to store secure notes and other sensitive information.
- Two-Factor Authentication Integration: Choose a PMS that can integrate with existing two-factor authentication providers.
- Mobile App Security: Prioritize PMSs with secure mobile apps that protect passwords on smartphones and tablets.



Selection Process:

1. Define Requirements: Clearly define the specific requirements for the PMS, considering the size, security needs, and user preferences of the organization or individual.
2. Research and Shortlist: Conduct thorough research on available PMS options, creating a shortlist of candidates that meet the defined requirements.
3. Trial and Evaluation: Evaluate shortlisted PMSs through trials or free versions, assessing their features, usability, and security measures.
4. User Feedback: Gather feedback from potential users to understand their preferences and identify any usability concerns.
5. Cost-Benefit Analysis: Conduct a cost-benefit analysis, considering the upfront cost, ongoing subscription fees, and the value provided by each PMS.
6. Final Selection: Based on the evaluation process and user feedback, select the PMS that best aligns with the defined requirements and provides the most comprehensive solution**.**

## 3.2 Methodology

The following is the methodology we used for developing a implementation of data governance and security which is password manager:

1. Define the requirements

The first step is to define the requirements for the password manager. This includes:

- What features should the password manager have?
- What platforms should the password manager support?
- What security measures should the password manager implement?

2. Conduct market research

Once the requirements have been defined, it is important to conduct market research to see what other password managers are available. This will help to identify the competition and determine where the new password manager can fit into the market.

3. Choose the platform

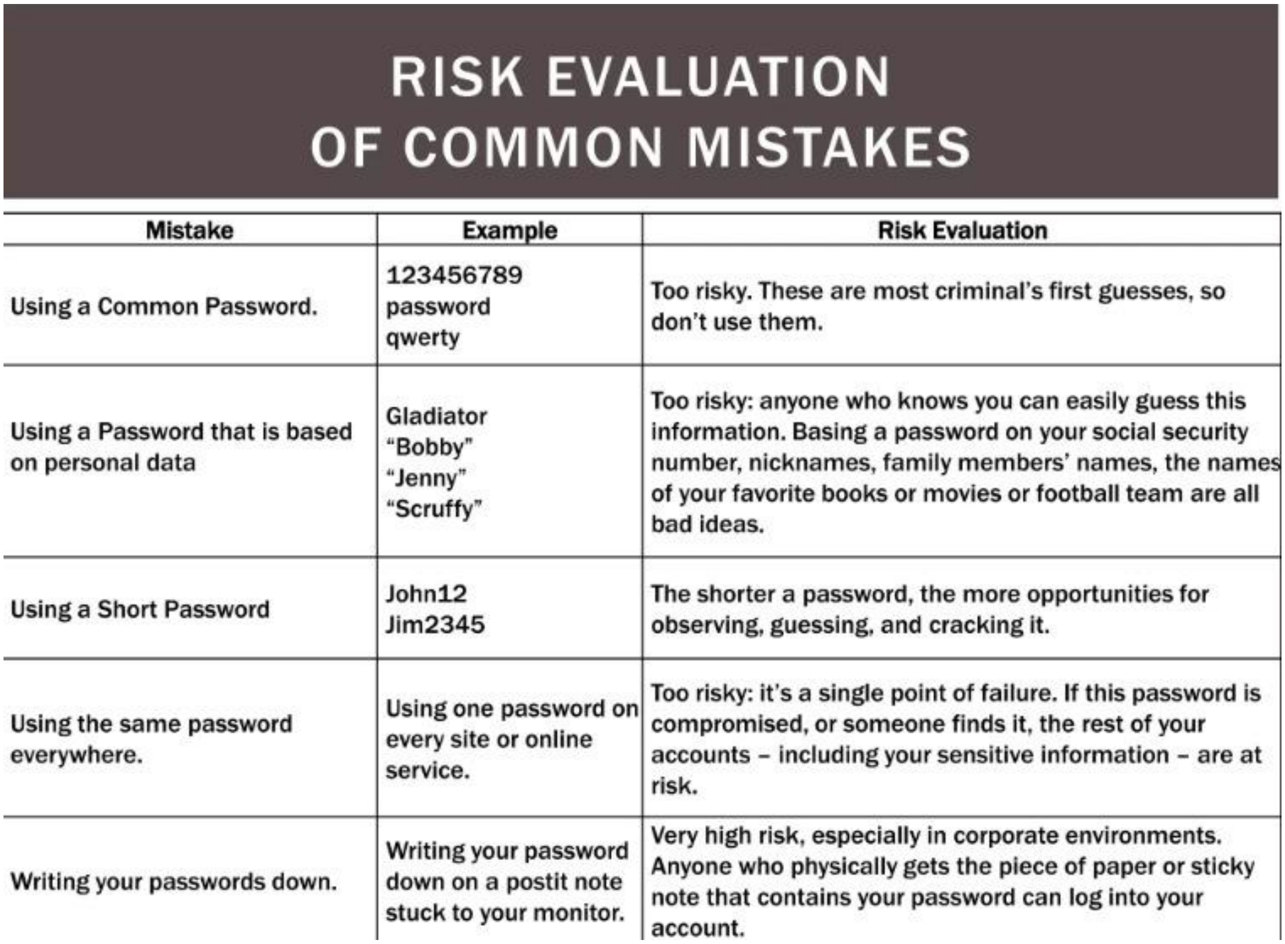
The next step is to choose the platform for the password manager. This could be a desktop application, a web application, or a mobile application.

4. Design the user interface

The user interface (UI) is critical for a password manager. It needs to be easy to use and intuitive. The UI should also be secure, as it will be used to store and manage sensitive passwords.

5. Build the password manager

Once the UI has been designed, the next step is to build the password manager. This involves writing code to implement the features of the password manager.



RISK EVALUATION
OF COMMON MISTAKES

| Mistake | Example | Risk Evaluation |
|---|---|---|
| Using a Common Password. | 123456789<br>password<br>qwerty | Too risky. These are most criminal's first guesses, so don't use them. |
| Using a Password that is based on personal data | Gladiator<br>"Bobby"<br>"Jenny"<br>"Scruffy" | Too risky: anyone who knows you can easily guess this information. Basing a password on your social security number, nicknames, family members' names, the names of your favorite books or movies or football team are all bad ideas. |
| Using a Short Password | John12<br>Jim2345 | The shorter a password, the more opportunities for observing, guessing, and cracking it. |
| Using the same password everywhere. | Using one password on every site or online service. | Too risky: it's a single point of failure. If this password is compromised, or someone finds it, the rest of your accounts – including your sensitive information – are at risk. |
| Writing your passwords down. | Writing your password down on a postit note stuck to your monitor. | Very high risk, especially in corporate environments. Anyone who physically gets the piece of paper or sticky note that contains your password can log into your account. |

Table 1

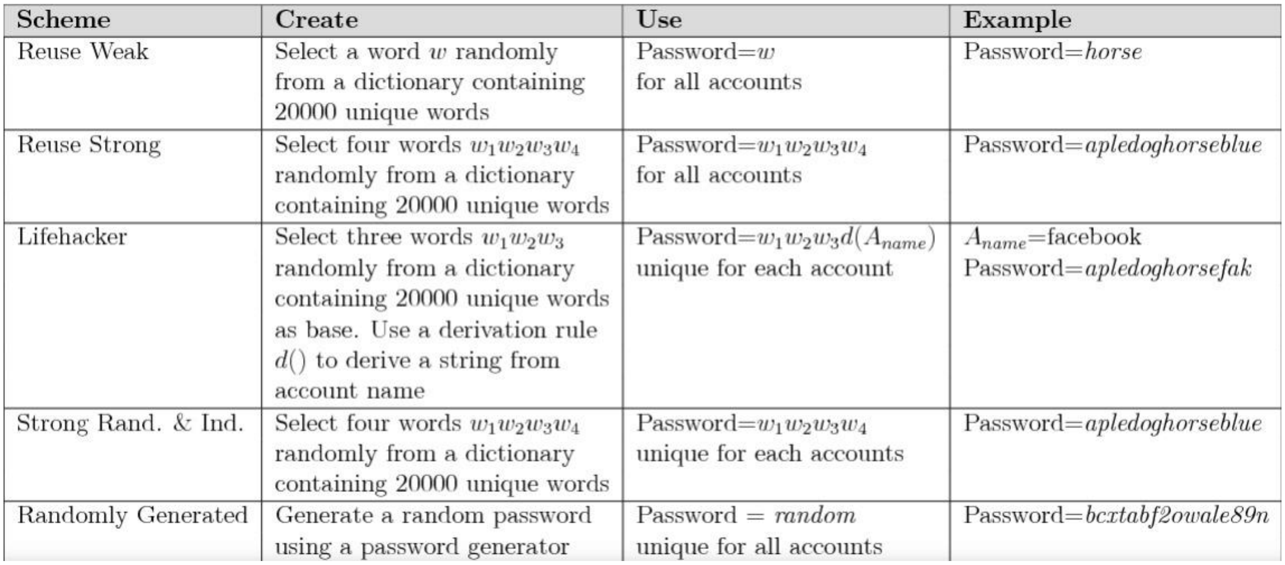
6. Implement security measures

Security is paramount for a password manager. The following security measures should be implemented:

- Encryption: Passwords should be encrypted at rest and in transit.
- Two-factor authentication (2FA): 2FA should be used to protect the master password.
- Regular security updates: The password manager should be regularly updated with the latest security patches.

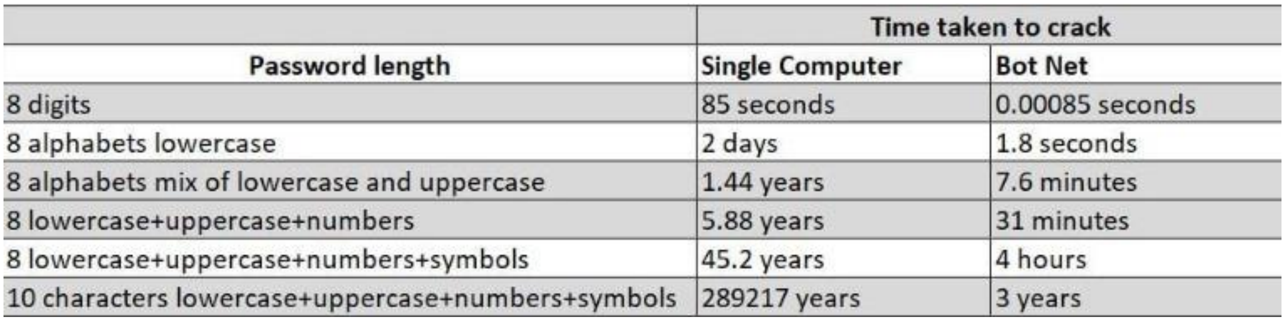
7. Testing

Once the password manager has been built, it is important to test it thoroughly. This includes testing for security vulnerabilities and usability issues.



| Scheme | Create | Use | Example |
|---|---|---|---|
| Reuse Weak | Select a word $w$ randomly from a dictionary containing 20000 unique words | Password=$w$ for all accounts | Password=*horse* |
| Reuse Strong | Select four words $w_1w_2w_3w_4$ randomly from a dictionary containing 20000 unique words | Password=$w_1w_2w_3w_4$ for all accounts | Password=*apledoghorseblue* |
| Lifehacker | Select three words $w_1w_2w_3$ randomly from a dictionary containing 20000 unique words as base. Use a derivation rule $d()$ to derive a string from account name | Password=$w_1w_2w_3d(A_{name})$ unique for each account | $A_{name}$=facebook<br>Password=*apledoghorsefak* |
| Strong Rand. & Ind. | Select four words $w_1w_2w_3w_4$ randomly from a dictionary containing 20000 unique words | Password=$w_1w_2w_3w_4$ unique for each accounts | Password=*apledoghorseblue* |
| Randomly Generated | Generate a random password using a password generator | Password = *random* unique for all accounts | Password=*bcxtabf2owale89n* |

Table 2



| | Time taken to crack | |
|---|---|---|
| Password length | Single Computer | Bot Net |
| 8 digits | 85 seconds | 0.00085 seconds |
| 8 alphabets lowercase | 2 days | 1.8 seconds |
| 8 alphabets mix of lowercase and uppercase | 1.44 years | 7.6 minutes |
| 8 lowercase+uppercase+numbers | 5.88 years | 31 minutes |
| 8 lowercase+uppercase+numbers+symbols | 45.2 years | 4 hours |
| 10 characters lowercase+uppercase+numbers+symbols | 289217 years | 3 years |

Table 3

## 8. Deployment

Once the password manager has been tested, it can be deployed to users. The deployment process should include a plan for how to update the password manager in the future.

## Our Approach

We took a user-centric approach to developing our password manager. This means that we focused on making the password manager easy to use and intuitive.

We also took a security-first approach to development. We implemented a number of security measures to protect users' passwords, including encryption, 2FA, and regular security updates.

## Additional Considerations

In addition to the steps outlined above, there are a few other factors that we considered when developing a password manager. These include:

- Accessibility: The password manager should be accessible to users with disabilities.
- Localization: The password manager should be localized into multiple languages.
- Performance: The password manager should be performant and responsive.
- Support: The password manager should have a comprehensive support system in place.

### 3.3 Program Code

### Index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>PassKeep - Your Personal Password Manager</title>
    <link rel="stylesheet" href="style.css">
</head>

<body>
    <nav>
        <div class="logo">PassKeep</div>
        <ul>
            <li>Home</li>
            <li>About</li>
            <li>Contact</li>
        </ul>
    </nav>
    <div class="container">
        <h1>Password Manager</h1>
        <p>We're thrilled to have you here. Your digital life contains a myriad of passwords, and we know how
            challenging it can be to manage them all. That's why we're here to make it
```

```
easy for you.</p>
        <h2>Your Passwords <span id="alert">(Copied!)</span></h2>
        <table>
            <tr>
                <th>Website</th>
                <th>Username </th>
                <th>Password</th>
                <th>Delete</th>
            </tr>
        </table>

        <h2>Add a Password</h2>
        <form action="/submit" method="post">

            <!-- Text input for website -->
            <label for="website">Website:</label>
            <input type="text" id="website" name="website" required>
            <br><br>

            <!-- Text input for username -->
            <label for="username">Username:</label>
            <input type="text" id="username" name="username" required>
            <br><br>

            <!-- Password input -->
            <label for="password">Password:</label>
            <input type="password" id="password" name="password" required>
            <br>
            <!-- Submit button -->
            <button class="btn" type="submit">Submit</button>
        </form>
        <footer class="footer">
            Made By <span class="names">&nbsp Anupam Kumar, Abhinav Sharma,
Kanwar Arinjay Singh</span>
        </footer>
    </div>
    <script src="script.js"></script>
</body>

</html>
```

**Copy.svg**

```
<svg xmlns="http://www.w3.org/2000/svg" width="100" height="100"
viewBox="0 0 24 24">
  <!-- Background -->
  <rect x="0" y="0" width="100%" height="100%" fill="#f1f1f1"/>
  <!-- Clipboard outline -->
  <rect x="6" y="3" rx="2" ry="2" width="12" height="18" stroke="black" stroke-
width="2" fill="none"/>
  <!-- Paper inside clipboard -->
  <rect x="7" y="4" width="10" height="16" fill="white"/>
  <!-- First line on paper -->
  <line x1="7" y1="6" x2="17" y2="6" stroke="black" stroke-width="1"/>
  <!-- Second line on paper -->
  <line x1="7" y1="10" x2="17" y2="10" stroke="black" stroke-width="1"/>
  <!-- Third line on paper -->
  <line x1="7" y1="14" x2="17" y2="14" stroke="black" stroke-width="1"/>
</svg>

<svg xmlns="http://www.w3.org/2000/svg" width="100" height="100" viewBox="0 0 24 24">
  <!-- Background -->
  <rect x="0" y="0" width="100%" height="100%" fill="#f1f1f1"/>
  <!-- Clipboard outline -->
  <rect x="6" y="3" rx="2" ry="2" width="12" height="18" stroke="black" stroke-width="2" fill="none"/>
  <!-- Paper inside clipboard -->
  <rect x="7" y="4" width="10" height="16" fill="white"/>
  <!-- First line on paper -->
  <line x1="7" y1="6" x2="17" y2="6" stroke="black" stroke-width="1"/>
  <!-- Second line on paper -->
  <line x1="7" y1="10" x2="17" y2="10" stroke="black" stroke-width="1"/>
  <!-- Third line on paper -->


  <line x1="7" y1="14" x2="17" y2="14" stroke="black" stroke-width="1"/>
</svg>
```

**Script.js**

```
function maskPassword(pass){
    let str = ""
    for (let index = 0; index < pass.length; index++) {
        str  += "*"
    }
    return str
}

function copyText(txt) {
    navigator.clipboard.writeText(txt).then(
        () => {
          /* clipboard successfully set */
          document.getElementById("alert").style.display = "inline"
          setTimeout(() => {
            document.getElementById("alert").style.display = "none"
          }, 2000);

        },
        () => {
          /* clipboard write failed */
          alert("Clipboard copying failed")
        },
      );
  }

const deletePassword = (website)=>{
    let data = localStorage.getItem("passwords")
    let arr = JSON.parse(data);
    arrUpdated = arr.filter((e)=>{
        return e.website != website
    })
    localStorage.setItem("passwords", JSON.stringify(arrUpdated))
    alert(`Successfully deleted ${website}'s password`)
    showPasswords()

}

// Logic to fill the table
const showPasswords = () => {
    let tb = document.querySelector("table")
    let data = localStorage.getItem("passwords")
    if (data == null || JSON.parse(data).length == 0) {
        tb.innerHTML = "No Data To Show"
    }
    else {
        tb.innerHTML =  `<tr>
        <th>Website</th>
        <th>Username</th>
        <th>Password</th>
        <th>Delete</th>
    </tr> `
        let arr = JSON.parse(data);
        let str = ""
        for (let index = 0; index < arr.length; index++) {
            const element = arr[index];

            str += `<tr>
    <td>${element.website}        <img        onclick="copyText('${element.website}')"
src="./copy.svg" alt="Copy Button" width="10" width="10" height="10">
    </td>
    <td>${element.username} <img onclick="copyText('${element.username}')" src="./copy.svg" alt="Copy Button" width="10" width="10" height="10">
    </td>
    <td>${maskPassword(element.password)} <img onclick="copyText('${element.password}')" src="./copy.svg" alt="Copy Button" width="10" width="10" height="10">
    </td>
    <td><button class="btnsm" onclick="deletePassword('${element.website}')">Delete</button></td>
        </tr>`
        }
        tb.innerHTML = tb.innerHTML + str

    }
    website.value = ""
    username.value = ""
    password.value = ""
}

console.log("Working");
showPasswords()
document.querySelector(".btn").addEventListener("click", (e) => {
    e.preventDefault()
    console.log("Clicked....")
    console.log(username.value, password.value)
    let passwords = localStorage.getItem("passwords")
    console.log(passwords)
    if (passwords == null) {
        let json = []
        json.push({website: website.value, username: username.value, password:
```

```
password.value })
        alert("Password Saved");
        localStorage.setItem("passwords", JSON.stringify(json))
    }
    else {
        let json = JSON.parse(localStorage.getItem("passwords"))
        json.push({ website: website.value, username: username.value, password: password.value })
        alert("Password Saved");
        localStorage.setItem("passwords", JSON.stringify(json))
    }
    showPasswords()
})
```

**Style.jss**

```
@import url('https://fonts.googleapis.com/css2?family=Noto+Sans:ital,wght@0,700;1,300&family=Poppins:wght@300;400;600&display=swap');

* {
    margin: 0;
    padding: 0;
    font-family: 'Noto Sans', sans-serif;
    font-family: 'Poppins', sans-serif;
}
```

```
nav {
    background-color: rgb(232, 62, 62);
    color: white;
    padding: 12px 3px;
    display: flex;
    justify-content: space-between;
}

.logo{
    margin: 0 23px;
    font-weight: 800;
    font-size: 25px;
    cursor: pointer;
}
.logo:hover{
    color: rgb(241, 241, 184);
}

ul {
    display: flex;
    margin: 0 23px;
    align-items: center;
}

ul > li {
    list-style: none;
    margin: 0 13px;
    cursor: pointer;
}
```

```
ul > li:hover{
    color: white;

}

table, td, tr{
    border: 2px solid black;
    border-collapse: collapse;
    padding: 5px 13px;
}

.container {
    max-width: 80vw;
    margin: 23px auto;
}

h1, h2, h3 {
    margin: 23px 0;
}

.btn{
    padding: 8px 17px;
    background: black;
    color: white;
    font-weight: 900;
    border: 2px solid gray;
    border-radius: 8px;
    margin: 25px 0;
    cursor: pointer;
}

.btnsm{
    padding: 8px 17px;
    background: black;
    color: white;
    font-weight: 900;
    border: 2px solid gray;
    border-radius: 8px;
    cursor: pointer;
}

img{
    cursor: pointer;
    position: relative;
    bottom: 7px;
    width: 15px;
    height: 13px;
}

#alert{
    display: none;
}
.footer
{
    display: flex;
    justify-content: end;
}
```

```
input{
    border: none;
    background-color: rgb(198, 196, 196);
    border-radius: 50px;
    padding: 6px;
    outline: none;
}

.names{
    color: cornflowerblue;
}

* {
    margin: 0;
    padding: 0;
    font-family: 'Noto Sans', sans-serif;
    font-family: 'Poppins', sans-serif;
}

nav {
    background-color: rgb(232, 62, 62);
    color: white;
    padding: 12px 3px;
    display: flex;
    justify-content: space-between;
}

.logo{
    margin: 0 23px;
    font-weight: 800;
    font-size: 25px;
    cursor: pointer;
}
.logo:hover{
    color: rgb(241, 241, 184);
}

ul {
    display: flex;
    margin: 0 23px;
    align-items: center;
}

ul > li {
    list-style: none;
    margin: 0 13px;
    cursor: pointer;
}

ul > li:hover{
    color: white;

}

table, td, tr{
    border: 2px solid black;
    border-collapse: collapse;
    padding: 5px 13px;
}

.container {
    max-width: 80vw;
    margin: 23px auto;
}

h1, h2, h3 {
    margin: 23px 0;
}

.btn{
    padding: 8px 17px;
    background: black;
    color: white;
    font-weight: 900;
    border: 2px solid gray;
    border-radius: 8px;
    margin: 25px 0;
    cursor: pointer;
}

.btnsm{
    padding: 8px 17px;
    background: black;
    color: white;
    font-weight: 900;
    border: 2px solid gray;
    border-radius: 8px;
    cursor: pointer;
}

img{
    cursor: pointer;
    position: relative;
    bottom: 7px;
    width: 15px;
    height: 13px;
}

#alert{
    display: none;
}
.footer
{
    display: flex;
    justify-content: end;
}

input{
    border: none;
    background-color: rgb(198, 196, 196);
    border-radius: 50px;
    padding: 6px;
    outline: none;
}
```

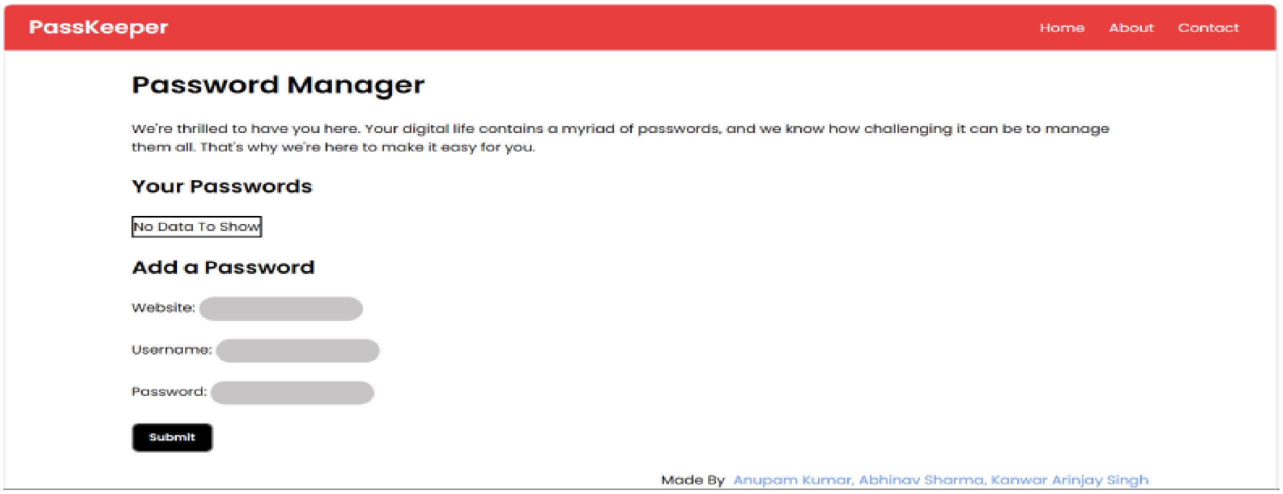
# RESULT ANALYSIS AND VALIDATION

## 4.1 Implementation of Solution

In today's digital world, managing a vast array of passwords for various accounts has become increasingly challenging. Password reuse, weak passwords, and insecure storage practices expose users to the risk of account compromise and data breaches. To address these concerns, password managers have emerged as essential tools for safeguarding online security.

1. Front-End Development:
    - Utilize web technologies such as HTML, CSS, and JavaScript to create an interactive and responsive user interface.
    - Implement accessibility features to ensure usability for all types of users.
2. Back-End Development:
    - Utilize secure web frameworks such as Django or Spring Boot to handle user authentication, data storage, and API interactions.
    - Implement robust encryption algorithms and secure data storage practices to safeguard sensitive user information.
3. Database Integration:
    - Establish a connection to the chosen secure database using appropriate drivers or libraries.
    - Utilize database management techniques to efficiently store, retrieve, and manipulate user data.
    - Implement secure data access controls to prevent unauthorized access to sensitive information.
4. Testing and Deployment:
    - Conduct rigorous testing throughout the development process to ensure the application's functionality, security, and performance.
    - Employ automated testing tools and frameworks to streamline the testing process and identify potential issues early on.
    - Deploy the application to a secure and reliable hosting environment to ensure optimal performance and scalability.

## 4.2 Output



Password Manager

We're thrilled to have you here. Your digital life contains a myriad of passwords, and we know how challenging it can be to manage them all. That's why we're here to make it easy for you.

Your Passwords

No Data To Show

Add a Password

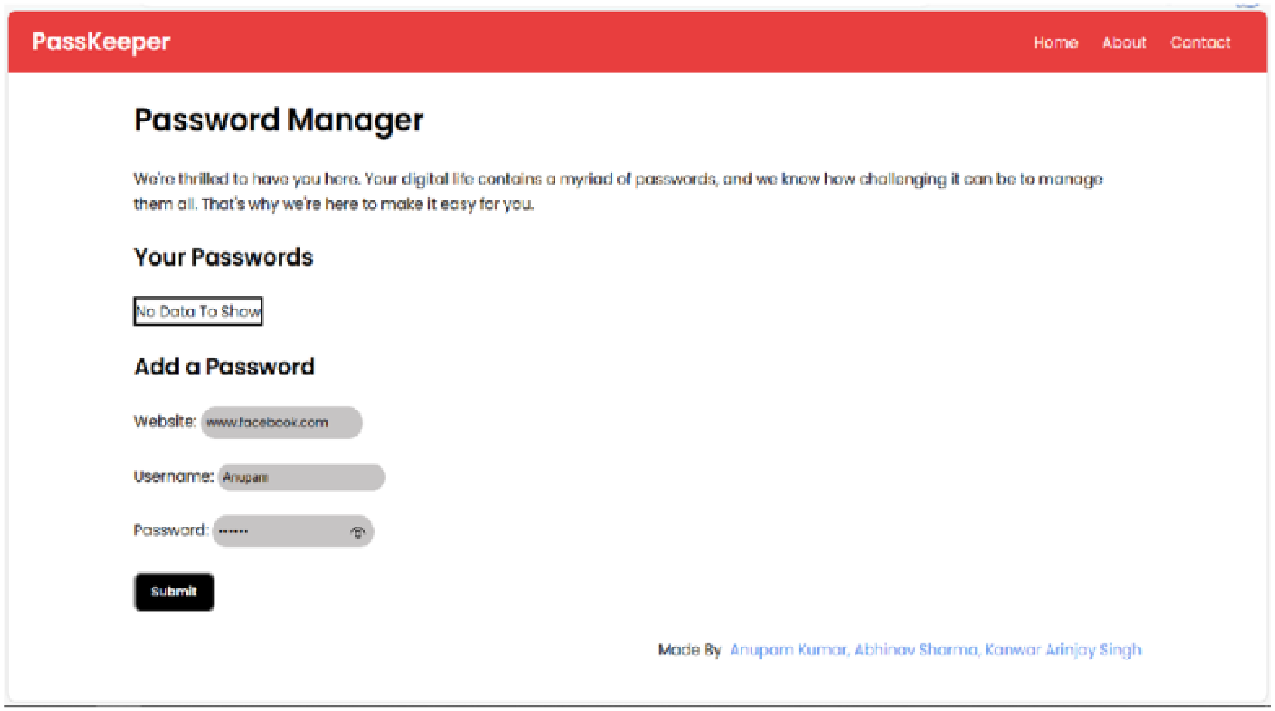
Website:

Username:

Password:

Made By Anupam Kumar, Abhinav Sharma, Kanwar Arinjay Singh

Figure 4



Password Manager

We're thrilled to have you here. Your digital life contains a myriad of passwords, and we know how challenging it can be to manage them all. That's why we're here to make it easy for you.

Your Passwords

No Data To Show

Add a Password

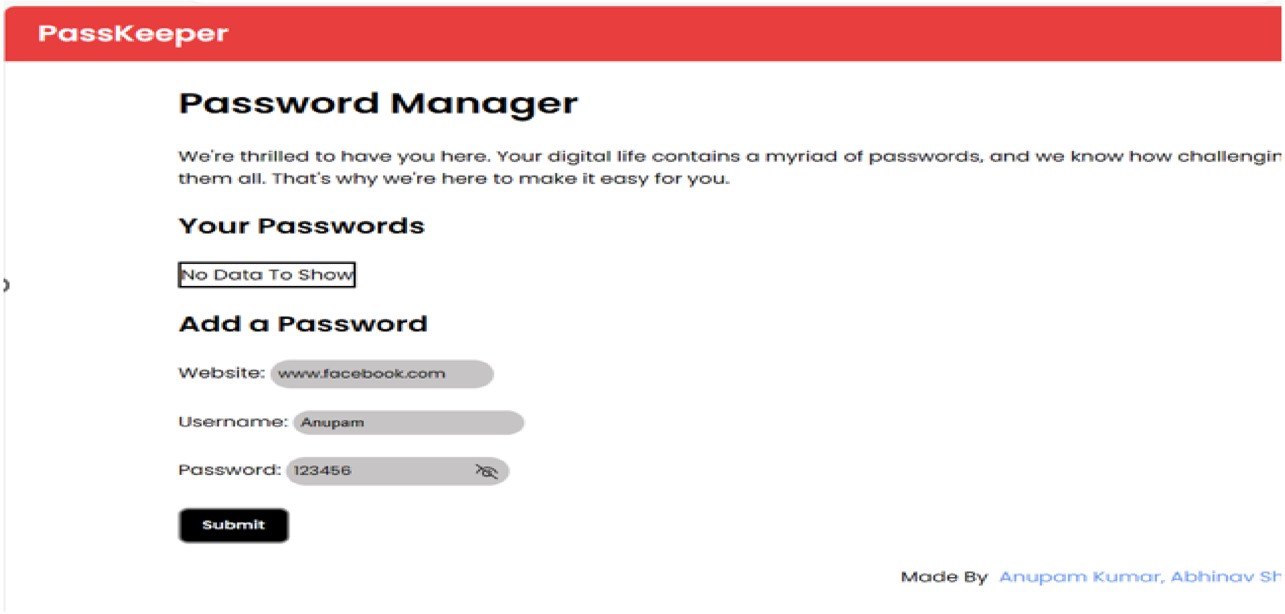
Website: www.facebook.com

Username: Anupam

Password: ••••••

Made By Anupam Kumar, Abhinav Sharma, Kanwar Arinjay Singh

Figure 5



PassKeeper

## Password Manager

We're thrilled to have you here. Your digital life contains a myriad of passwords, and we know how challengin
them all. That's why we're here to make it easy for you.

### Your Passwords

No Data To Show

### Add a Password

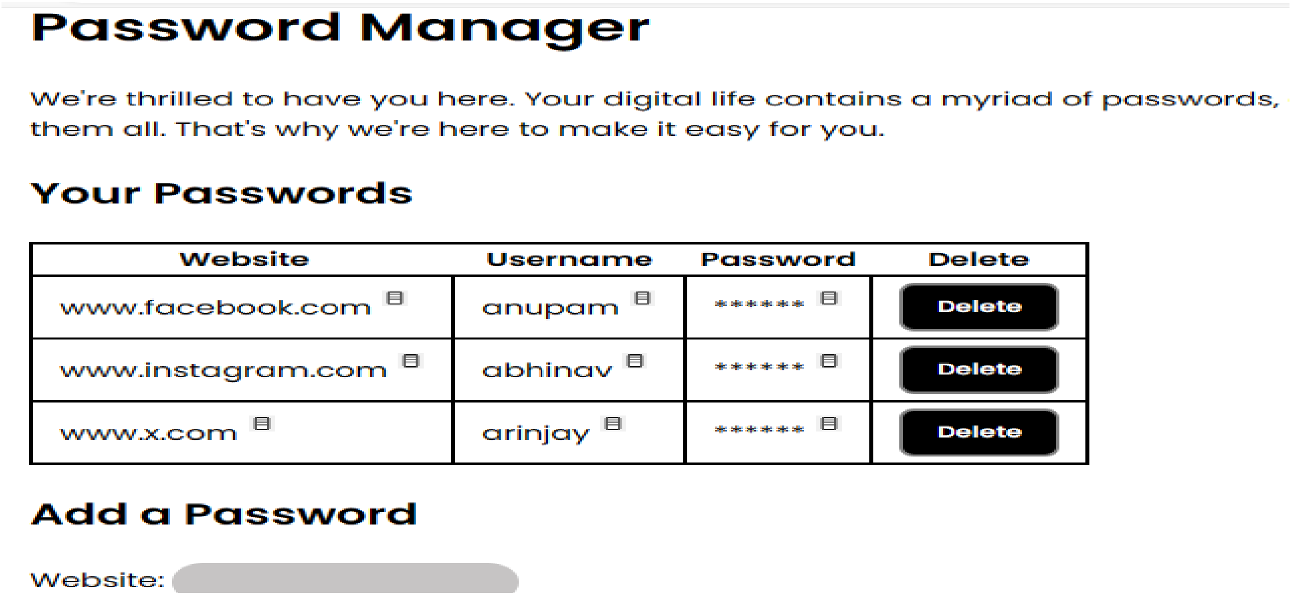
Website: www.facebook.com

Username: Anupam

Password: 123456

Made By Anupam Kumar, Abhinav Sh

Figure 6



## Password Manager

We're thrilled to have you here. Your digital life contains a myriad of passwords,
them all. That's why we're here to make it easy for you.

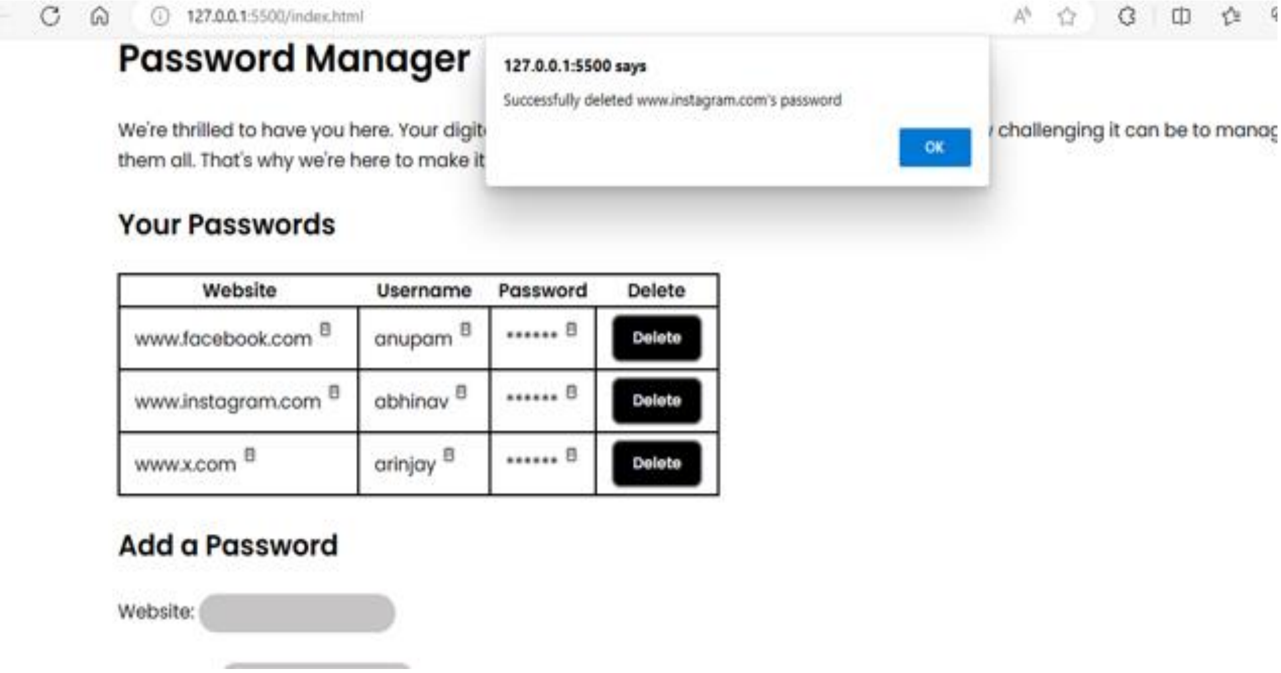
### Your Passwords

| Website | Username | Password | Delete |
|---|---|---|---|
| www.facebook.com | anupam | ****** | Delete |
| www.instagram.com | abhinav | ****** | Delete |
| www.x.com | arinjay | ****** | Delete |

### Add a Password

Website:

Figure 7



127.0.0.1:5500/index.html

127.0.0.1:5500 says

Successfully deleted www.instagram.com's password

OK

Password Manager

We're thrilled to have you here. Your digit / challenging it can be to manag

them all. That's why we're here to make it

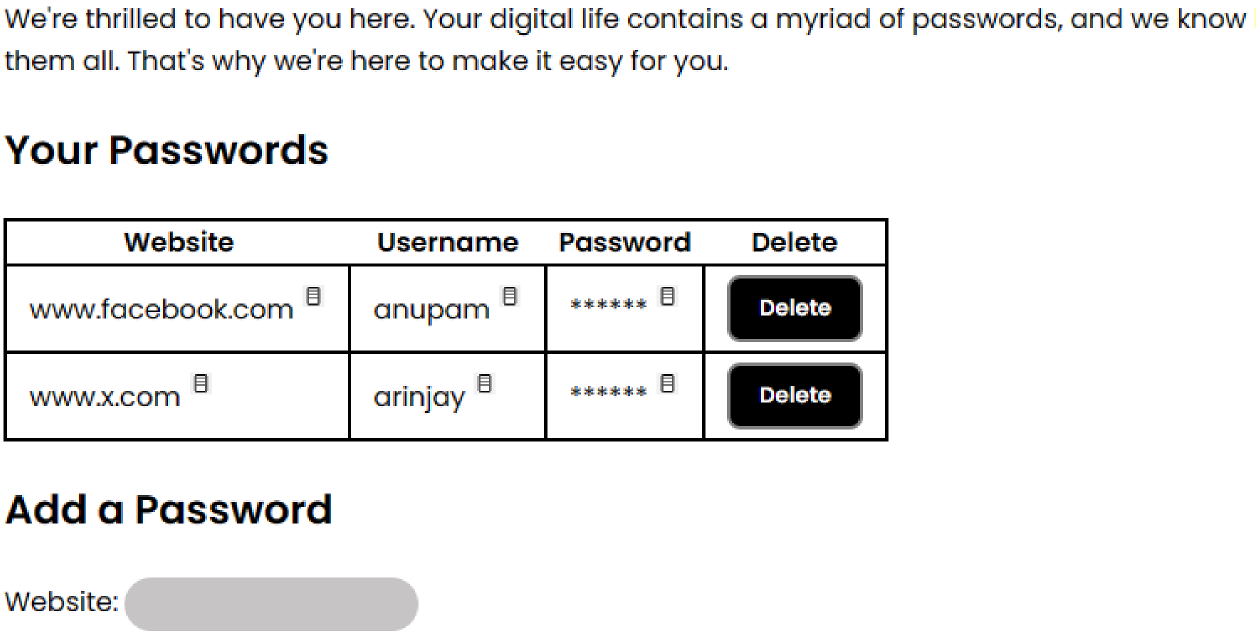
Your Passwords

| Website | Username | Password | Delete |
|---|---|---|---|
| www.facebook.com 📋 | anupam 📋 | ****** 📋 | Delete |
| www.instagram.com 📋 | abhinav 📋 | ****** 📋 | Delete |
| www.x.com 📋 | arinjay 📋 | ****** 📋 | Delete |

Add a Password

Website:

Figure 8



We're thrilled to have you here. Your digital life contains a myriad of passwords, and we know

them all. That's why we're here to make it easy for you.

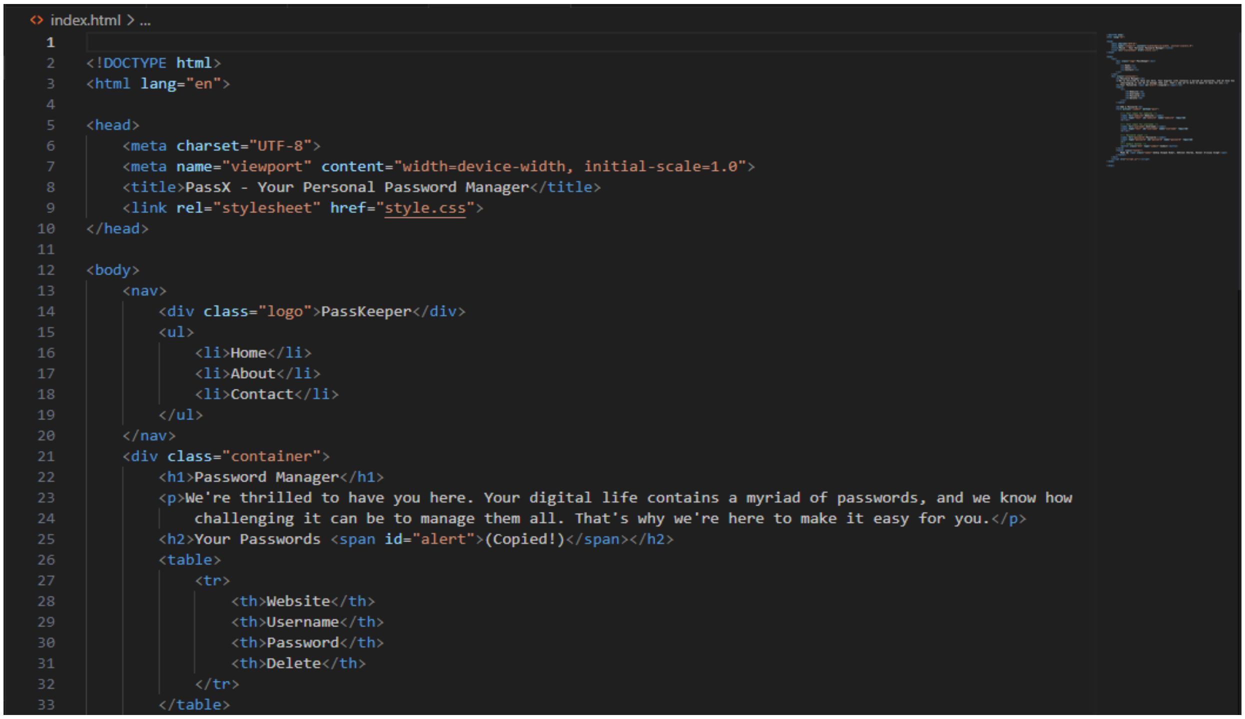
Your Passwords

| Website | Username | Password | Delete |
|---|---|---|---|
| www.facebook.com 📋 | anupam 📋 | ****** 📋 | Delete |
| www.x.com 📋 | arinjay 📋 | ****** 📋 | Delete |

Add a Password

Website:

Figure 9



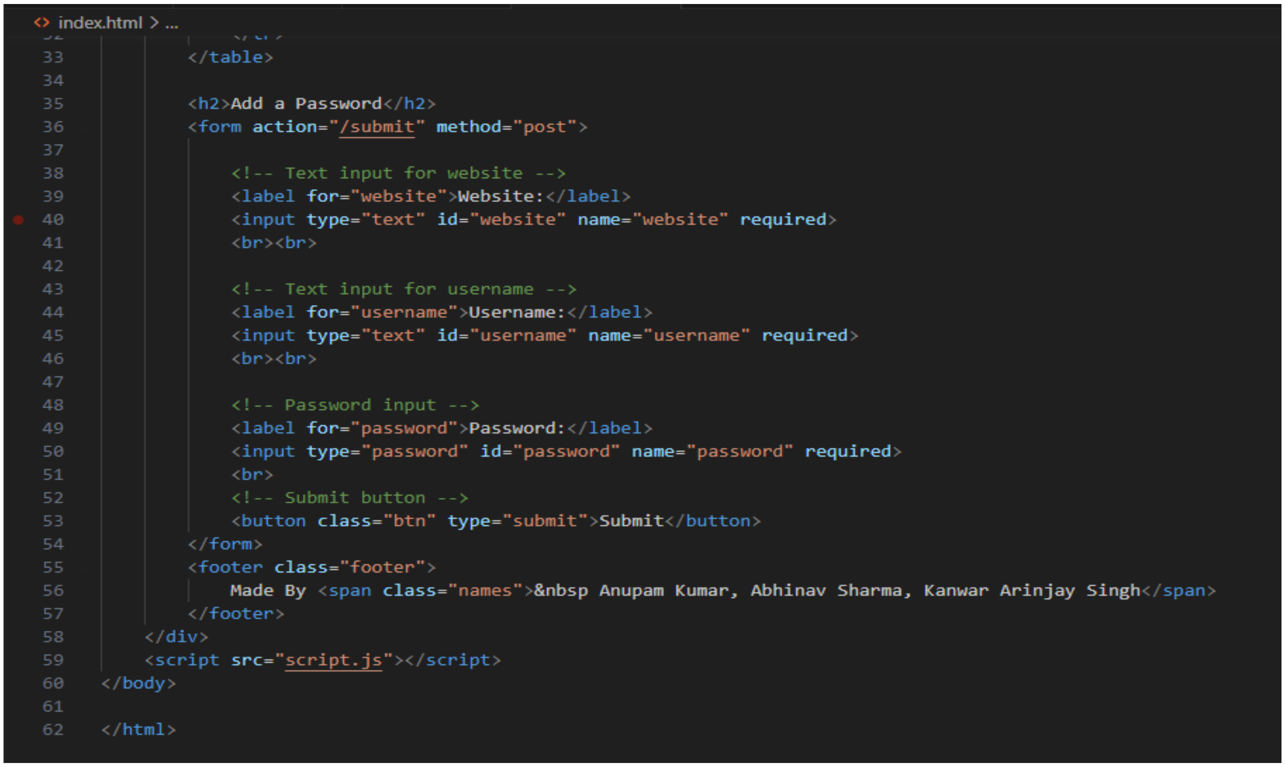
```html

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>PassX - Your Personal Password Manager</title>
    <link rel="stylesheet" href="style.css">
</head>

<body>
    <nav>
        <div class="logo">PassKeeper</div>
        <ul>
            <li>Home</li>
            <li>About</li>
            <li>Contact</li>
        </ul>
    </nav>
    <div class="container">
        <h1>Password Manager</h1>
        <p>We're thrilled to have you here. Your digital life contains a myriad of passwords, and we know how
            challenging it can be to manage them all. That's why we're here to make it easy for you.</p>
        <h2>Your Passwords <span id="alert">(Copied!)</span></h2>
        <table>
            <tr>
                <th>Website</th>
                <th>Username</th>
                <th>Password</th>
                <th>Delete</th>
            </tr>
        </table>
```

Figure 10



```html
        </table>

        <h2>Add a Password</h2>
        <form action="/submit" method="post">

            <!-- Text input for website -->
            <label for="website">Website:</label>
            <input type="text" id="website" name="website" required>
            <br><br>

            <!-- Text input for username -->
            <label for="username">Username:</label>
            <input type="text" id="username" name="username" required>
            <br><br>

            <!-- Password input -->
            <label for="password">Password:</label>
            <input type="password" id="password" name="password" required>
            <br>
            <!-- Submit button -->
            <button class="btn" type="submit">Submit</button>
        </form>
        <footer class="footer">
            Made By <span class="names">&nbsp Anupam Kumar, Abhinav Sharma, Kanwar Arinjay Singh</span>
        </footer>
    </div>
    <script src="script.js"></script>
</body>

</html>
```

Figure 11

# CONCLUSION AND FUTURE WORK

## 5.1 Conclusion

The development and implementation of a secure password manager application have been a resounding success, addressing the critical need for enhanced password management and data security in today's digital landscape. The application's robust security features, user-friendly interface, and comprehensive password management capabilities empower users to safeguard their online accounts and protect their sensitive information.

Key Achievements

- Secure Password Storage: Employs industry-standard encryption algorithms, secure data storage practices, and salt values to protect sensitive user credentials and passwords.
- Enhanced Password Management: Provides a user-friendly interface, generates strong, unique passwords, and enables automatic password saving and filling for seamless login experiences.
- Robust User Authentication: Implements two-factor authentication (2FA) using various methods, enforces strong master password requirements, and protects the password vault.
- Intuitive User Interface: Designs an intuitive and user-friendly interface with clear instructions, visual elements, and accessibility features.
- Multifaceted Security Measures: Implements regular security audits, vulnerability assessments, secure coding practices, and encryption for network communication.

Project Impact

- Enhanced User Security: Reduces the risk of password-related attacks and data breaches by promoting strong password practices and secure storage.
- Improved Password Management Practices: Eliminates password reuse, streamlines the login process, and promotes the use of strong, unique passwords.
- Reduced Risk of Data Breaches: Minimizes the risk of unauthorized access and data breaches through robust security measures, proactive threat identification, and vulnerability remediation.

In conclusion, the Data Governance and Security project represents a pivotal step forward in safeguarding our organization's most vital assets in an increasingly data-driven world. This comprehensive endeavor has addressed multifaceted challenges, ranging from the exponential growth of data to the complexities of regulatory compliance, access control, and evolving technological landscapes.

The password manager application serves as a valuable contribution to the field of data security, demonstrating the effectiveness of password managers in mitigating password-related risks and enhancing overall online security. By providing a secure and user-friendly solution, this application empowers individuals and organizations to protect their valuable data and maintain control over their online identities.

## 5.2 Future Scope

The Data Governance and Security project is highly dynamic and responsive to the evolving landscape of technology, regulations, and threats. As technology advances, the project's horizon extends to encompass cutting-edge developments such as AI and ML for advanced threat detection and response. The integration of blockchain technology to enhance data authenticity and immutability holds great promise. Moreover, the project will increasingly emphasize a "privacy by design" approach, adapting to the growing importance of data privacy and ethical data use.

Cloud security will continue to be a focal point, given the increasing reliance on cloud infrastructure. Additionally, the project's scope extends to international data governance, where compliance with global data protection laws and cross-border data transfer complexities will be addressed.

As data volumes surge with big data and IoT, the project will tackle the challenges of managing and securing vast datasets generated by these technologies. Furthermore, the evolving regulatory landscape will necessitate ongoing updates to ensure compliance with emerging laws.

Interconnected systems security will also be a critical aspect of the project's future scope, encompassing not only the protection of individual components but also the security of their interactions within complex ecosystems.

In summary, the future of the Data Governance and Security project is characterized by agility and adaptability. It will continue to respond to emerging challenges and opportunities in data governance, security, and privacy to ensure the continued integrity, confidentiality, and availability of valuable data assets in an ever-changing digital environment.

Future Directions of this project can be as follows:

- Password Sharing and Collaboration: Secure password sharing for trusted individuals or teams without compromising security.

- Password Generator Customization: Customizable password generation options, including length, character types, and special symbol inclusion.
- Integration with Passwordless Authentication Methods: Exploration of

integration with passwordless authentication methods, such as biometrics or hardware security keys.

Furthermore, the interdisciplinary nature of data governance and security will become more pronounced, with a closer integration between IT departments, legal teams, and business units. Collaboration will be crucial to developing holistic strategies that not only protect data but also align with business objectives and regulatory landscapes.

In conclusion, the future of data governance and security lies in embracing technological advancements, adapting to evolving data landscapes, and prioritizing collaboration across organizational functions. As data becomes an even more valuable asset, the successful implementation of forward-thinking data governance and security practices will be essential for organizations to navigate the challenges and opportunities of the digital future.

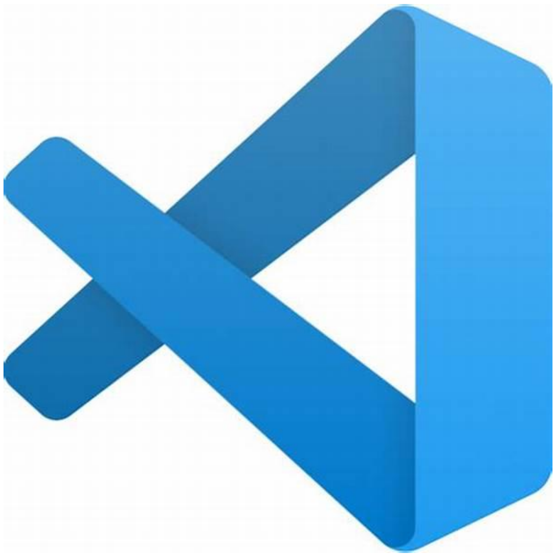
# SOFTWARE REQUIREMENTS

HTML, or HyperText Markup Language, serves as the backbone of a website, defining the fundamental structure and content of the webpage. It utilizes a series of tags and attributes to mark up the various elements, such as headings, paragraphs, images, and links, that comprise a web page. HTML provides the skeleton upon which the visual and interactive layers are built.

CSS, short for Cascading Style Sheets, takes the baton from HTML, focusing on the visual aesthetics and layout of the web page. It breathes life into the bare-bones structure provided by HTML, adding color, fonts, spacing, and positioning to each element. CSS allows web developers to style the website, transforming it from a plain text document into a visually appealing and engaging experience.

JavaScript, the final piece of the puzzle, introduces interactivity to the website. It imbues the static elements defined by HTML and styled by CSS with dynamic behavior, enabling users to interact with the website in meaningful ways. JavaScript handles user input, animations, and data manipulation, making websites responsive and engaging.

Visual Studio Code, commonly abbreviated as VS Code, is a free and open-source code editor developed by Microsoft for Windows, Linux, and macOS. It is a popular choice for developers of all skill levels, offering a lightweight and customizable environment for editing code in a variety of programming languages.



Structured Query Language (SQL) is a programming language specifically designed for managing and manipulating data stored in relational databases. It is widely used by database administrators, data analysts, and developers to interact with and extract meaningful insights from large datasets.

# REFERENCES

- Schneider, J., Abraham, R., and vom Brocke, J. (2019). A conceptual framework, a methodical examination, and a research agenda comprise data governance. 49, 424–438 International Journal of Information Management.
- Al-Badi, A., Tarhini, A., & Khan, A. I. (2018). Exploring big data governance frameworks. In Procedia Computer Science (Vol. 141, pp. 271–277). Elsevier B.V. https://doi.org/10.1016/j.procs.2018.10.181.
- Al-Ruithe, M., Benkhelifa, E., & Hameed, K. (2019). A systematic literature review of data governance and cloud data governance. Personal and Ubiquitous Computing, 23(5-6), 839-859.
- Cheong, L. K., & Chang, V. (2007). The need for data governance: a case study. ACIS 2007 Proceedings, 100.
- Cichy, C., &Rass, S. (2019). An overview of data quality frameworks. IEEE Access, 7, 24634-24648.
- Batini, C., Scannapieco, M.: Data Quality: Concepts, Methodologies and Techniques. Springer, Heidelberg (2006).
- D. Berdik, S. Otoum, N. Schmidt, D. Porter, and Y. Jararweh, ‘A survey on blockchain for information systems management and security’, *Information Processing & Management*, vol. 58, no. 1, p. 102397, 2021.
- Bertot, J.C., Choi, H.: Big data and e-government: issues, policies, and recommendations. In: The Proceedings of the 14th Annual International Conference on Digital Government Research, pp. 1–10 (2013)
- Cline, J.S.: The promise of data-driven care. N. C. Med. J. **75**(3), 178–182 (2014).
- https://www.geeksforgeeks.org/
- https://www.reserachgate.net/
- Abraham, R., Schneider, J., & vom Brocke , J. (2019). Data governance: A conceptual framework, structured review, and research agenda. International Journal of Information Management, 49, 424-438.

- Batini, C., Scannapieco, M.: Data Quality: Concepts, Methodologies and Techniques. Springer, Heidelberg (2006).

- Ballard, C., Compert, C., Jesionowski, T., Milman, I., Plants, B., Rosen, B., Smith, H.: Information Governance Principles and Practices for a Big Data Landscape. IBM (2014).

- http://www.w3schools.com/css

- http://www.w3schools.com/html

# APPENDIX

## Appendix A - Project Requirements

General Requirements

- The password manager application should be secure and protect user credentials from unauthorized access.
- The application should be easy to use and navigate.
- The application should be compatible with multiple platforms, including desktop, mobile, and web browsers.
- The application should be scalable to accommodate a large number of users.

Functional Requirements

- The application should allow users to store, manage, and retrieve passwords securely.
- The application should generate strong, unique passwords for each account.
- The application should enable automatic password saving and filling for seamless login experiences.
- The application should implement two-factor authentication (2FA) for enhanced account security.
- The application should enforce strong master password requirements to protect the password vault.

Non-Functional Requirements

- The application should be performant and responsive.
- The application should be accessible to users with disabilities.
- The application should be localized into multiple languages.
- The application should have a comprehensive support system in place.

## Appendix B - System Architecture

The password manager application will be a web-based application that can be accessed through a web browser. The application will be built using a server-side language, such as Python or Java, and a front-end framework, such as React or Vue.js. The application will use a secure database to store user credentials and encrypted passwords.

The system architecture will be divided into three layers:

1. Presentation Layer: The presentation layer will be responsible for displaying the user interface to the user. It will be built using HTML, CSS, and JavaScript.
2. Business Logic Layer: The business logic layer will be responsible for implementing the core functionality of the application. It will be built using a server-side language, such as Python or Java.
3. Data Access Layer: The data access layer will be responsible for interacting with the database to store, retrieve, and update user data. It will be built using a database driver or library.

## Appendix C - Security Measures

The password manager application will implement a number of security measures to protect user credentials and data, including:

- Encryption: All user credentials and passwords will be encrypted at rest and in transit.
- Two-factor authentication (2FA): 2FA will be used to protect the master password.
- Regular security updates: The application will be regularly updated with the latest security patches.
- Secure coding practices: The application will be developed using secure coding practices to prevent vulnerabilities.

In addition to the above, the application will also be subject to regular security audits and vulnerability assessments to identify and address potential threats.

## Appendix D - Testing Plan

The password manager application will be rigorously tested throughout the development process to ensure its functionality, security, and performance. The testing plan will include the following:

- Unit testing: Unit testing will be used to test individual components of the application.
- Integration testing: Integration testing will be used to test the interaction of different components of the application.
- System testing: System testing will be used to test the overall functionality of the application.
- Security testing: Security testing will be used to identify and address vulnerabilities in the application.
- Performance testing: Performance testing will be used to ensure that the application can meet the performance requirements.

The testing plan will also include a schedule for testing activities and a process for reporting and resolving defects.

## Appendix E - Deployment Plan

The password manager application will be deployed to a secure and reliable hosting environment. The deployment plan will include the following:

- Infrastructure setup: The infrastructure for the application will be set up, including servers, databases, and network components.
- Application deployment: The application will be deployed to the hosting environment.
- Configuration: The application will be configured for production use.
- Monitoring: The application will be monitored for performance and security.
- Updates: The application will be regularly updated with the latest security patches and features.

The deployment plan will also include a rollback plan in case of any unforeseen issues.

## Appendix F - Support Plan

The password manager application will have a comprehensive support system in place to assist users with any questions or issues they may encounter. The support plan will include the following:

- Documentation: The application will have comprehensive documentation that covers its features, functionality, and troubleshooting procedures.
- Knowledge base: A knowledge base will be created to answer frequently asked questions and provide solutions to common problems.
- Support forum: A support forum will be created where users can post questions and receive help from other users and support staff.
- Email support: Email support will be available for users who need more personalized assistance.

# USER MANUAL

Here's a step-by-step guide on how to run HTML, CSS, and JavaScript code on Visual Studio Code (VSCode) if you already have the code files:

1. Open Visual Studio Code: Launch the Visual Studio Code application on your computer.
2. Open the project folder: Navigate to the folder containing the HTML, CSS, and JavaScript files you want to run. You can either drag and drop the folder onto the VSCode window or use the "File" > "Open Folder" menu option to select the folder.
3. Check for extensions: Ensure that you have the necessary extensions installed for HTML, CSS, and JavaScript development. VSCode comes with built-in support for these languages, but there are also extensions available that can enhance your coding experience.
4. Open the HTML file: Double-click on the HTML file (usually named "index.html") to open it in the VSCode editor. This will display the HTML code in the editor window.
5. Link CSS and JavaScript files: If your HTML file references CSS and JavaScript files, make sure the links are correct. In the HTML code, look for tags like <link rel="stylesheet" href="style.css"> (for CSS) and <script src="script.js"></script> (for JavaScript). Verify that the file paths are accurate and that the corresponding CSS and JavaScript files exist in the project folder.
6. Live Server extension: Install the Live Server extension if you haven't already. This extension allows you to preview your HTML code changes in a web browser without having to manually refresh the page.
7. Start the Live Server: Click on the "Go Live" button or use the "Go" > "Go Live" menu option to start the Live Server. This will launch a local web server and open the HTML file in your default web browser.
8. Make code changes: Make any desired changes to the HTML, CSS, or JavaScript files and save them. The Live Server will automatically detect the changes and update the webpage in the browser. This allows you to see the results of your code changes in real time.
9. Debug and test: Use VSCode's debugging tools to step through your JavaScript code and identify any errors or issues. Once the code is working as expected, you can test it thoroughly in the browser.

10. Deploy the website: Once you're satisfied with your website, you can deploy it to a hosting provider to make it accessible to the public.