ABSTRACT

Introduction

In the digital age, individuals and organizations alike are faced with the challenge of managing an ever-growing number of login credentials for various online services and applications. The proliferation of accounts and the need for unique, complex passwords for each one has made it increasingly difficult for users to maintain secure practices without resorting to risky behaviours such as password reuse or relying on easily guessable combinations. This project aims to address this critical issue by developing a robust, user-friendly password safe application that allows users to securely store and manage their credentials.

Problem Statement and Overview

The primary problem of this project seeks to solve is the secure storage and management of user credentials. In today's interconnected world, individuals often have dozens, if not hundreds, of online accounts, each requiring unique login information. Remembering all these credentials securely is virtually impossible for most people, leading to poor password practices that compromise security.

Tools and Applications Used

The project utilizes Python, renowned for its versatility and extensive library support, Django for rapid web application development, Django REST framework for creating powerful APIs, PostgreSQL for Database management, and frontend technologies like HTML, CSS, JavaScript, ReactJS for a responsive user interface.

Existing System and Proposed Plan

To address the concern of single points of failure, the system incorporates a key derivation function (KDF) that generates the encryption key from the user's master password. This approach ensures that the encryption key is never stored directly, reducing the risk of key compromise.

Architecture and Design

The system architecture involves several key components:

1. **Backend**: Built on Python and Django, handling video processing, storage, and user authentication.
2. **Database**: Storing encrypted credentials and user information in tables using PostgreSQL utilizing advantages of SQL properties.
3. **Frontend**: Implemented using HTML, CSS, JavaScript and ReactJS to create an intuitive interface for users to search for videos, select formats, and initiate downloads.

Conclusion and Expected Output

This Python-based password-safe project leveraging Django provides a robust and secure solution for credential management. By combining strong encryption, multi-factor authentication, and user-friendly features, the application offers a comprehensive platform for individuals and organizations to safeguard their sensitive information in an increasingly digital world. The project's emphasis on security, usability, and scalability makes it a valuable tool for addressing the growing challenges of credential management and data protection.