



A project report on

# **PyPASS - A Desktop Password Manager**

Submitted in partial fulfilment of the requirements for the degree of

B. Tech in Information Technology

By

**Saket Pandey (1806413)**

**Sambeet Kumar Pani (1806512)**

**Tushar Abir (1806532)**

**Devyansh Singh (1806375)**

Under the guidance of

**Mrs. Bhaswati Sahoo**

School of Computer Engineering

Kalinga Institute of Industrial Technology

Deemed to be University

Bhubaneswar

April 2022



## **CERTIFICATE**

This is to certify that the project report entitled “PyPASS – A Desktop Password Manager” submitted by:

|                           |                 |
|---------------------------|-----------------|
| <b>SAKET PANDEY</b>       | <b>1806413</b>  |
| <b>SAMBEET KUMAR PANI</b> | <b>1806512</b>  |
| <b>TUSHAR ABIR</b>        | <b>1806532.</b> |
| <b>DEVYANSH SINGH</b>     | <b>1806375</b>  |

In partial fulfilment of the requirements for the award of the **Degree of Bachelor of Technology in Discipline of Engineering** is a bonafide record of the work carried out under my guidance and supervision at School of Computer Engineering, Kalinga Institute of Industrial Technology, Deemed to be University.

Signature of Supervisor 1  
NAME OF THE SUPERVISOR 1  
Academic affiliation  
Organization

.....  
**The Project was evaluated by us on 16/04/2022**

Mrs. Bhaswati Sahoo

## **ACKNOWLEDGEMENTS**

We express my deep sense of gratitude to Mrs. Bhaswati Sahoo whose guidance, encouragement, suggestion and very constructive criticism have contributed immensely to the evolution of our ideas on the project.

We are very much thankful to the Program Head and Dean of the School of CSE for their valuable guidance, keen interest and encouragement at various stages of our project.

**SAKET PANDEY**

**SAMBEET KUMAR PANI**

**TUSHAR ABIR**

**DEVYANSH SINGH**

## **ABSTRACT**

In today's technologically advancing world, cybersecurity is an all-time growing concern for every person, to protect their important and precious data from illegal miners and hackers. Our application helps to resolve such concerns in the form of a password manager, which primarily focuses on a single user, offline environment so far. We use a master password as the primary key to store all of the user's credentials for various sites, portals, etc. The entries are stored using encryption and hashing to ensure utmost security for the user's data.

## **TABLE OF CONTENTS**

| S.No       | Topic                                 | Pg. No |
|------------|---------------------------------------|--------|
| <b>I</b>   | Abstract                              | 4      |
| <b>II</b>  | Table of Contents                     | 5      |
| <b>III</b> | <b><u>CHAPTER A: INTRODUCTION</u></b> | 6      |
|            | <i>A.1</i> Aim of the Project         |        |
| <b>IV</b>  | <b><u>CHAPTER B: BACKGROUND</u></b>   | 7      |
|            | <i>B.1</i> About the Project          |        |
|            | <i>B.1.1</i> Working Concept          |        |
|            | <i>B.1.2</i> Technologies used        |        |
|            | <i>B.2</i> Study of Similar Projects  |        |
| <b>V</b>   | <b><u>CHAPTER C: FEATURES</u></b>     | 11     |
|            | <i>C.1</i> Features                   |        |
|            | <i>C.2</i> Architecture               |        |
| <b>VI</b>  | <b><u>CHAPTER D: CONCLUSION</u></b>   | 20     |
|            | <i>D.1</i> Future Work                |        |
|            | <i>D.2</i> Password Recommendations   |        |
| <b>VII</b> | References                            | 21     |

# **CHAPTER A**

## **INTRODUCTION**

**PyPASS** is an offline single-user desktop password application which helps people manage their passwords. The human-factor kicks in as people tend to forget their passwords, design bad passwords, or are not interested in security. As passwords have become increasingly ubiquitous, people either write them down on paper or re-use them for multiple platforms which puts this as a major weakness. We cannot remember every big unique password for every account. Hence a Password Manager can help us not worry about remembering all the different passwords.

We succeeded in implementing a barebones password manager application. It has all the necessary basic features required for user password management.

- All stored under a master password.
- Search, add, update, delete passwords
- Import-export external databases containing user passwords
- Storing passwords locally in place of any online centralized server
- Encrypting passwords BEFORE storing in the database.

### **A.1. AIM OF THE PROJECT**

We cannot trust to save all our passwords to an online Password Manager, since the data is transmitted via the internet and if in future their servers are compromised by some malware, with no chances of backup then, we will have to reset all those passwords before we can be exploited by hackers. So the best secure way to backup our password is an offline password manager with advanced security features which can be accessed easily and instantly.

Password databases can also be on an external physical piece of hardware making it safer from cyber-attacks. We make it easier to use a strong password that is sufficiently random, long and different for every one of your accounts.

## CHAPTER 2

# **BACKGROUND**

A password manager is an encrypted piece of software or a program that allows users to store, generate, and manage their passwords for local applications and online services, as well as other information, in one convenient location with one master password.

### **B.1.1 WORKING CONCEPT**

In this modern age, the majority of our sensitive data is stored online. With that, the concern for cyber-security is steadily increasing. Our project aims to provide safe storage for all your passwords without the use of the internet so that all your passwords stay secure only with you. Our program is focused on a single user and saves his details locally. Upon the first use, you are required to create a Master password for your profile, along with setting up multi-factor authentication. Our program saves all the passwords in an encrypted database file, with the key as the master password. When the user logs in using his master password, the program decrypts the CSV file to get the desired password the user seeks.

### **B.1.2 TECHNOLOGY USED**

One of the primary focuses of the project was to keep it as simple and concise as possible, both for the user and the developer. Keeping this in mind, we have developed the project **entirely on Python**, which is a high-level, general-purpose interpreted language. Its key features are code readability made possible by the use of significant indentation along with its object oriented approach are ideal for our project.

Some of the most significant python packages and modules we have already used are listed below-

#### **1. NumPy**

It is a module that helps in working on multi-dimensional arrays as well as matrices. It contains multiple functions from the domains of linear algebra and Fourier transform as well.

## **2. python-dateutil**

It is a simple yet powerful extension that is added to the standard date time module of python.

## **3. cryptography**

With our project, data and application security is of utmost priority. This is where this module comes into play, which acts as a standard cryptographic library. It provides us with both, cryptographic recipes along with the primitives, among which both high-level recipes and low-level interfaces are available.

## **4. PyQt5**

It acts as the binder between the python environment and Qt, which is a cross-platform GUI toolkit. It implements in such a way as to allow python to be used as an alternative application development language to C++.

## **5. pyAesCrypt**

It is a file encryption module for python. It uses AES file-crypt format to encrypt/decrypt files along with binary streams.

The technologies for the newly implemented / in-progress features are listed below-

## **6. OpenCV – 4.5**

A real-time Computer vision and image-processing framework built on C/C++. We have integrated it into our application via the OpenCV-python package.

## **7. MediaPipe – 0.8.5**

A customizable, open-source and cross-platform machine learning solutions framework developed by Google. MediaPipe comes with some pre-trained ML models such as hand recognition, object detection, face detection etc.

## **8. Tensorflow – 2.5.0**

An open-source library for machine learning and deep learning developed by Google. It can be used across a range of tasks but has a particular focus on deep neural networks.



## **9. OneTimePass**

It is a library for generating one-time passwords in conjunction with Google Authenticator.

## **10. Secrets**

The secrets library is used for generating random numbers for managing important data, in this case, the one time password (OTP) used for multi factor authentication.

## **B.2 STUDY OF SIMILAR PROJECTS**

**i) Lastpass** - A password manager that saves the data encrypted in the cloud. It requires a login email id and a password associated with it. Recent policy changes have made it so the data saved is only shared among similar types of devices. Desktop and phones have separate cloud storage, within the same login info.

Lastpass provides a simple “one-click” process of entering the details in a website as long as you have saved login info with respect to that website beforehand.

Lastpass is based on Python, C, and Java. Objective-C is used for automatic suggestions when using an app.

**ii) 1Password** - A password manager made by Agilebytes. They make use of a subscription based model for more features and better security for the users. More features include storing licenses, card details, social security numbers, and more.

1Password files are synchronized using Dropbox, local WiFi, iCloud (A cloud service from Apple) and also their own website 1Password.com (paid subscription needed). They provide a chrome extension to help provide a simpler and better experience to sync your passwords across devices. They provide a simple and convenient way to save and enter your passwords that work pretty well on many phones.

Python, Go and Rust are used for the internal functioning of 1Password. Javascript and Objective-C work the basis of the external support with a GUI like browser extensions, apps, etc.

**iii) BitWarden** - Bitwarden is a password manager that's open source that can also store other sensitive information.

Being open source, Bitwarden has the highest flexibility on the platforms and how it runs. It has a command- line interface and also an application with a user interface, while also providing browser extension as a choice. It has 2-factor authentication using authenticator apps, emails, etc. Bitwarden also saves older passwords in their history which is completely encrypted and stored along with your current/ updated passwords.

The internal functioning has been programmed using C#. Depending on the platform, it also uses angular, javascript, HTML and typescript.

**iv) Apple Keychain** - Apple keychain designed to be used exclusively on Apple devices. It's a well-integrated service that saves your passwords and personal information and shares it only between your devices. It does have a browser extension for use on Windows which requires lots of prerequisites before being able to use it.

Apple Keychain has been made using Swift, a programming language that they have developed for their devices and apps.

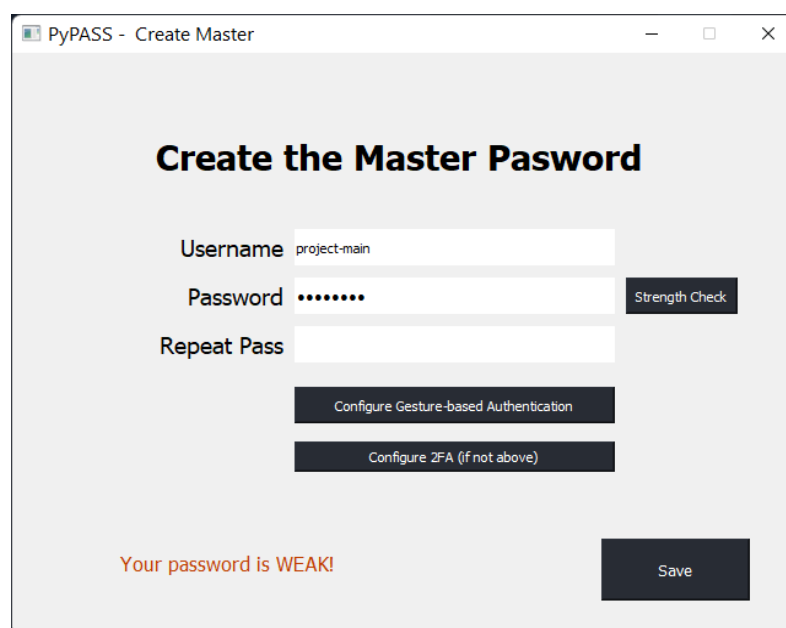
## CHAPTER C

# FEATURES

### C.1 FEATURES

#### Master Password

Our application uses the concept of a master password or master key. This way the user only needs to remember one key to get access to all their other passwords and credentials for safekeeping.



The screenshot shows a window titled "PyPASS - Create Master". The main heading is "Create the Master Password". Below this, there are three input fields: "Username" with the value "project-main", "Password" with masked characters "\*\*\*\*\*", and "Repeat Pass" which is empty. To the right of the Password field is a "Strength Check" button. Below the input fields are two buttons: "Configure Gesture-based Authentication" and "Configure 2FA (if not above)". At the bottom left, there is a red warning message: "Your password is WEAK!". At the bottom right is a "Save" button.

#### Elaborate Login Function

We provide our users with various login methods. It can be as simple as just using their username and master password to login or they can opt for more secure multi-factor options.

#### Multi Function Application

We provide our users with various quality of life functions within the application for ease of access and usage. These include-

- Search - easily search for stored credentials from the database.
- Add - add to the existing database with as many credentials as needed.
- Delete - delete any of the existing credentials when no longer required.

- Password Generator - give the user the choice to generate a random and secure password according to their needs.
- Strength Analyzer - provides the user with the option to check how secure any password they are using is.
- Clipboard Integration - copy any credentials straight from the application onto their clipboard for ease of use.

STATUS: Records Found(1)

master

Search

STATUS: No search string detected, type something!

The Passwords cannot be empty

PyPASS - Delete Password

Are you sure to delete the record ?

ID: 1  
Project: Fakebook  
Username: mzucky

After deleting, You cannot recover the record !

Cancel OK

PyPASS - Create new

Create a new Record Pasword

Project GitLab

Username project-main

Password ..... Strength Check

Generate Password

Notes 8th semester

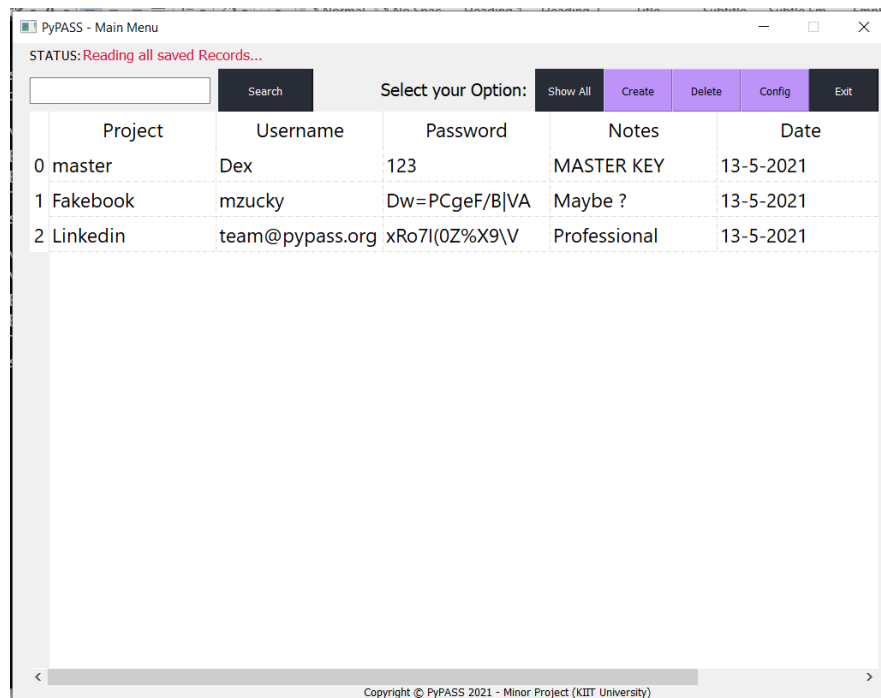
Save Clear All Close

Your password is STRONG

### Easy to use user interface

We provide our user with a very minimal but effective user interface. Any regular user will be able to fully utilise all domains of our application without any trouble understanding the

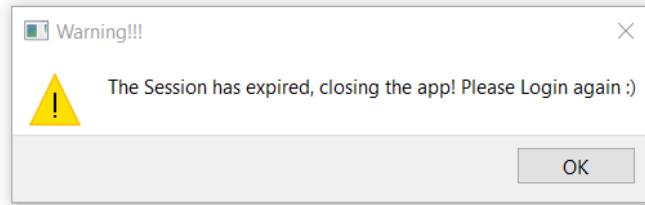
features or the user interface. The designing has been done with correspondence to modern industry standards and designs for smooth user experience.



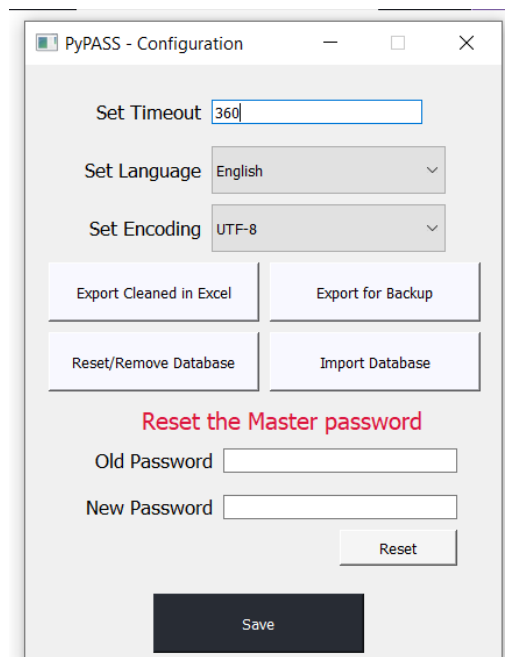
## **Secure Data**

With our application, security is of utmost concern. Keeping this in mind, we have incorporated multiple layers of security to attain the most secure data.

- Offline Storage - one of the key features our application boasts is the ability to store the user data locally. This way no one, not even us, apart from the user has any access to their data. All the data is stored only on a single device with no need for the internet or third party access.
- Secure Storage, even locally - We strive for utmost security of the user's data. To achieve this, we use multiple encryption, decryption and hashing techniques and store the data in a CSV format post encryption so that anyone with access to the device apart from the user cannot access their data.
- Session Timeout - An added quality of life feature, this ensures that if the user leaves the application without logging out, the application will shut itself down after a certain amount of inactivity detected by the application itself. This ensured any unintentional third party access to data due to human error.



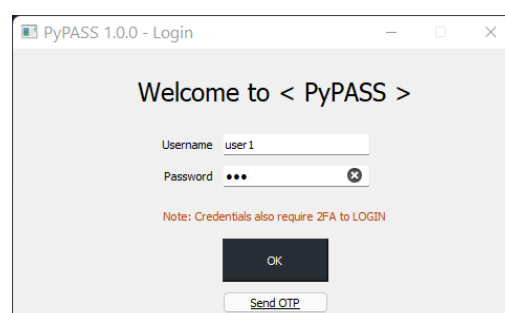
- Export/Import Data - we store the data in an easy to handle CSV format, which in turn allows us to provide the user with the option to export or import their data to/from another device if needed.

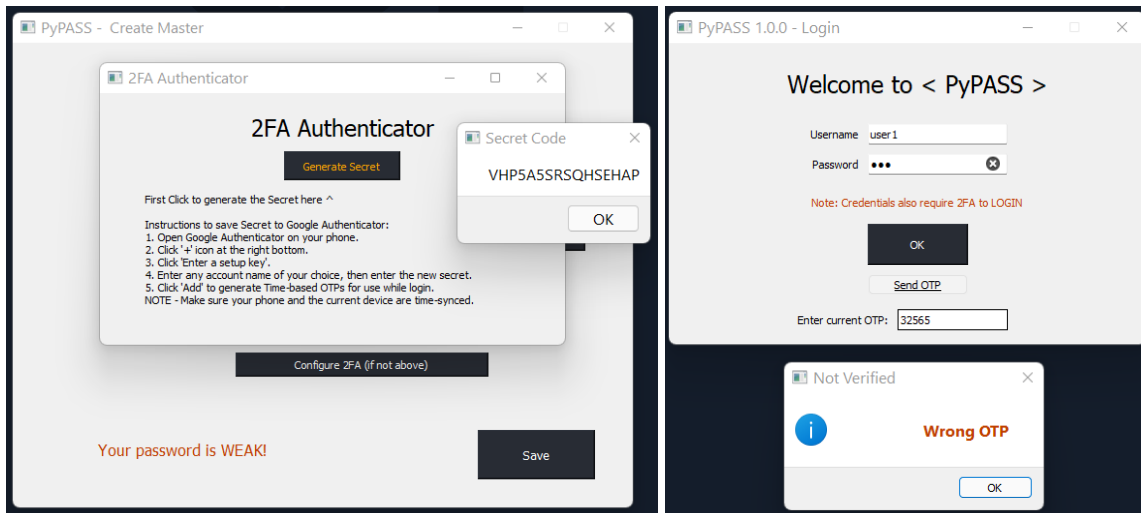


## **Multi Factor Authentication**

In today's world of increasing cyber threats, the need to secure data on digital devices is as high as ever. A simple password simply does not cut it anymore. Hence, we provide our users with various multi factor authentication methods for utmost security.

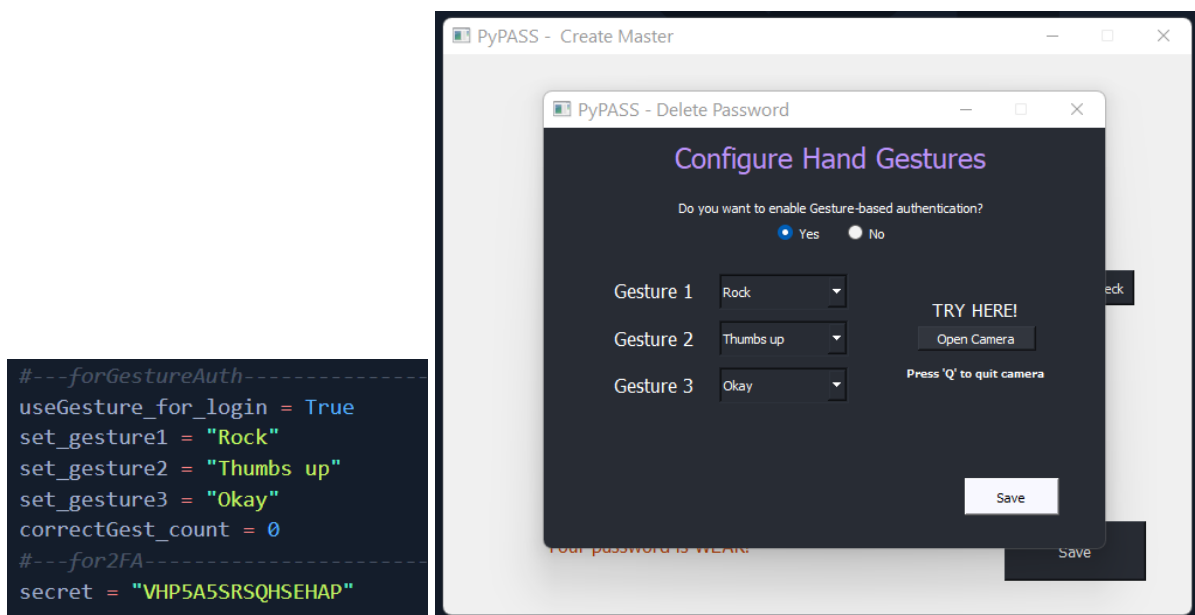
- *Third Party Authentication apps* - if the user opts for this method, he will be prompted with a randomly generated one time password (OTP) via a third party generator app such as Google Authenticator every time he wishes to login into the application.



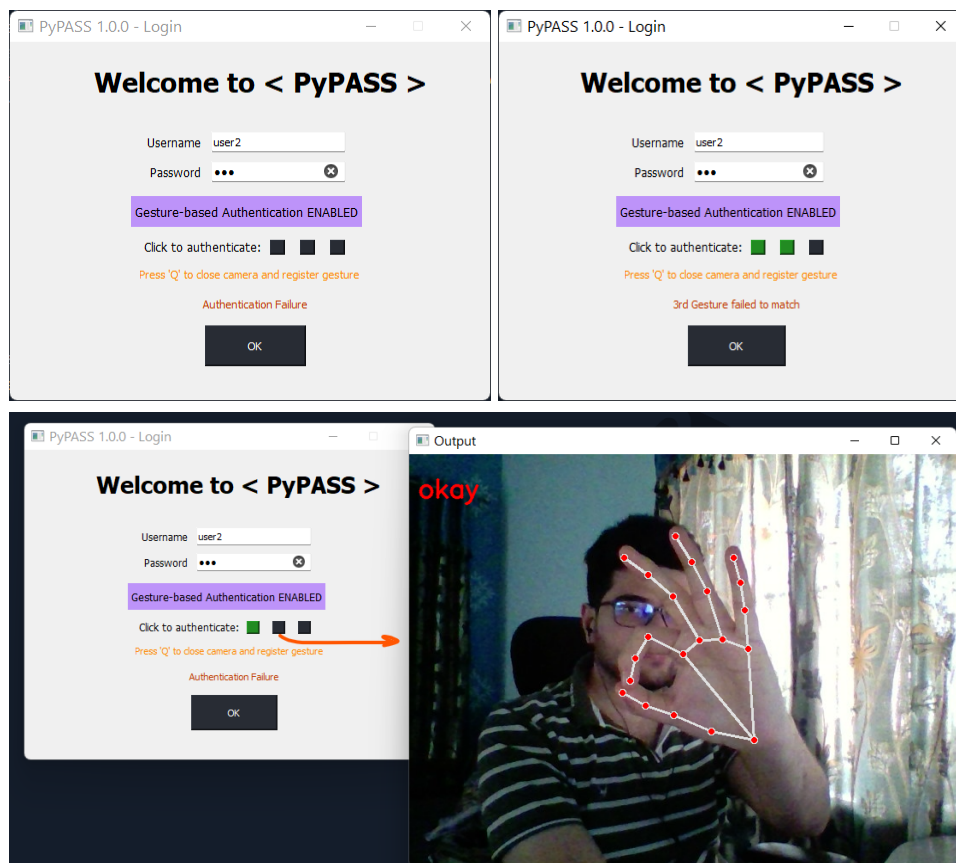
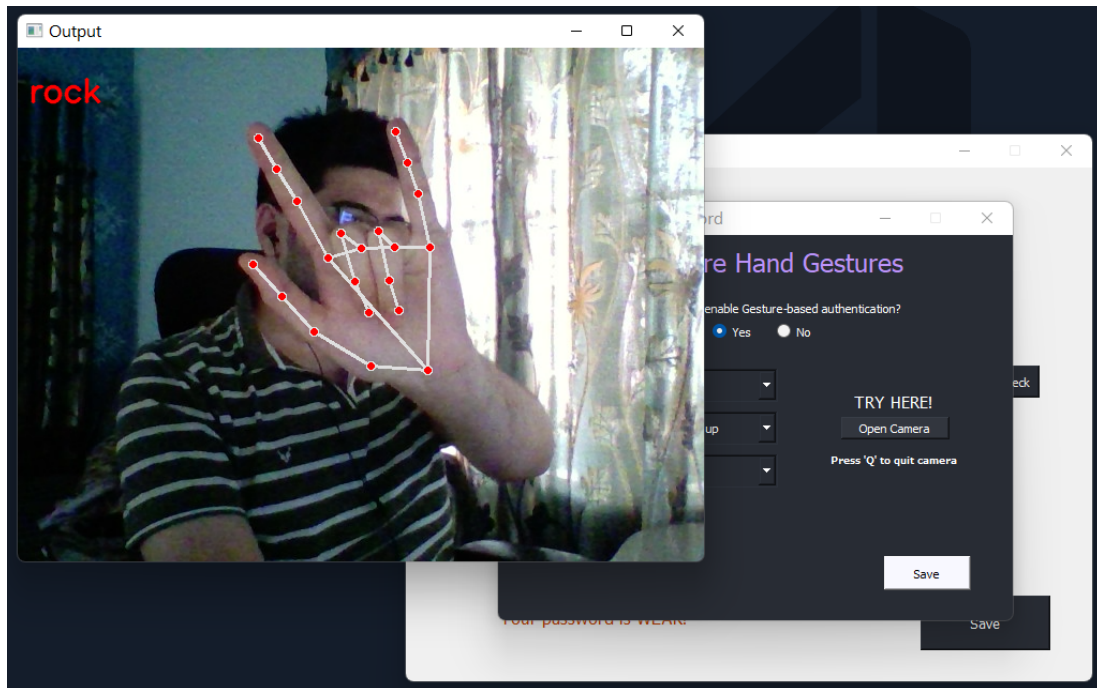


- *Hand Gesture Recognition* - another key proposition in our application is the method to use a sequence of hand gestures to login to the application as a multi factor option. We use computer vision integrated with machine learning which then allows any device with a camera to use this technique. The user sets a sequence of gestures as an authentication medium. Once set, every time the user wishes to log in, he will be prompted to gesture the preset sequence facing the device's camera. Once the application verifies the gestures and their order are both correct, the user will be logged in.

This solution will improve the user experience and strengthen security at the same time.

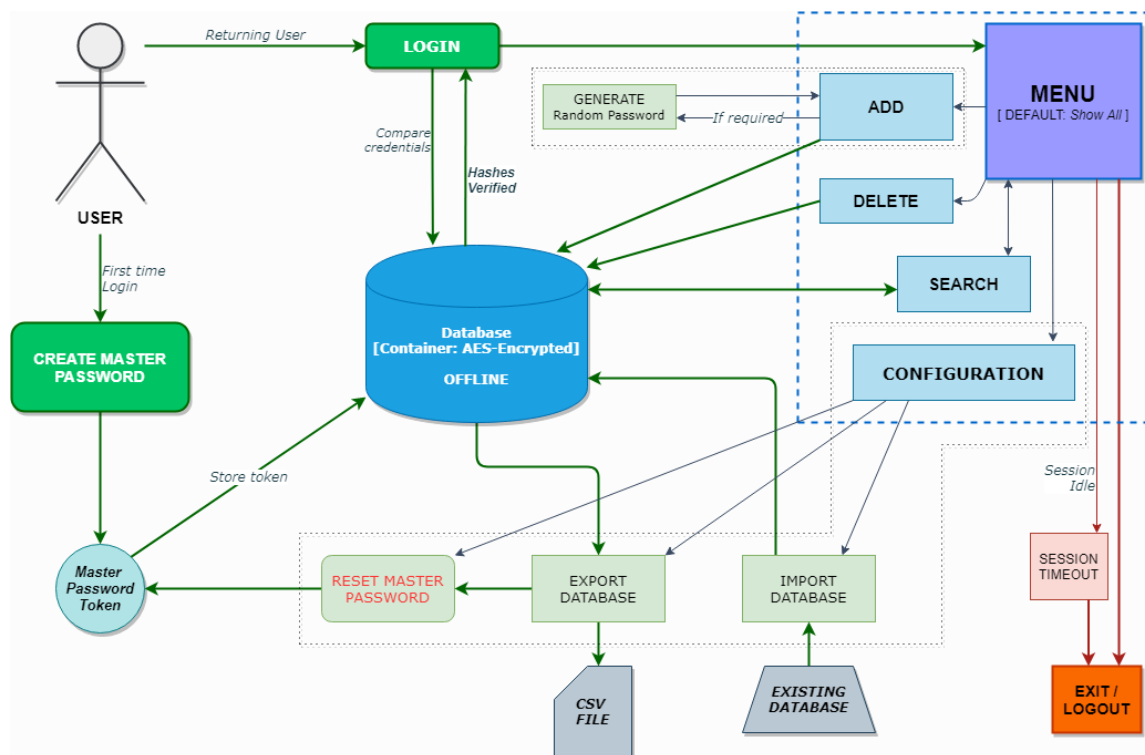


- We'll first use MediaPipe to recognize the hand and the hand's key points. MediaPipe returns a total of 21 key points for each detected hand. These key points will be fed into Tensorflow's pretrained gesture recognizer network by keras, to recognize the hand pose. MediaPipe can detect multiple hands in a single frame, but we'll detect only one hand at a time in this project.

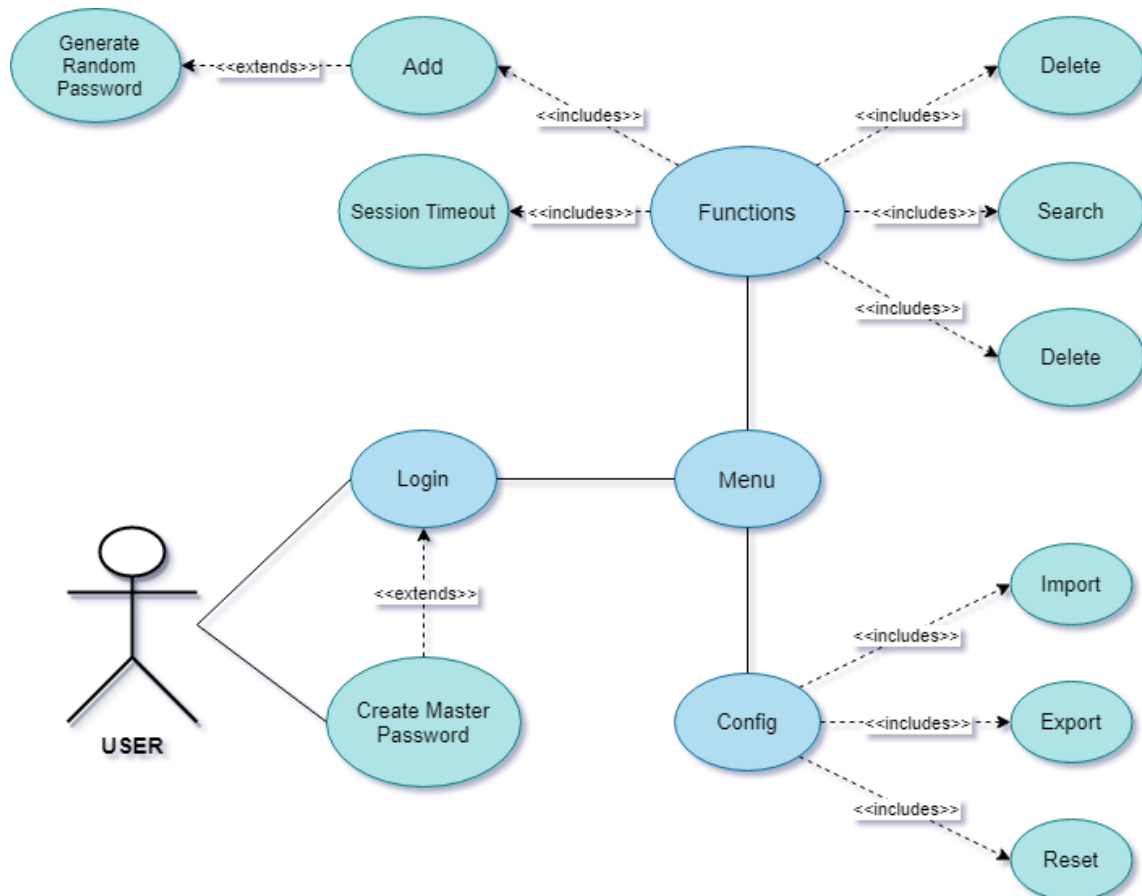




## C.2 ARCHITECTURE



Basic Application Flow Diagram



UML Diagram

## **CHAPTER D**

# **CONCLUSION**

To sum up the whole experience, the project is a great learning experience for all the members. We are learning a lot about the threats associated with the digital world and how we can contribute our share towards enhancing the security of the same. Going through all the documentations and templates online, they are helping us to get on par with the industry standards and requirements necessary, and match our project to that level as close as possible.

### **4.1 FUTURE WORK**

We also hope to add more features to the application in future updates, some of which are mentioned below –

- Adding full portability so that it can be transferred as easily as just copying.
- Integrating the application to store the data using a SQL database.
- In the current state, our application can handle only one user per device. We hope to expand this to multi users per device, so public devices can also use our service.
- We also hope to make GUI based improvements for improved user experience.

### **4.2 RECOMMENDATIONS**

Consider using a password manager, which is an application that can run on a computer, smartphone, or in the cloud, that securely tracks and stores passwords. Most password managers can also generate strong, random passwords for each account. As long as the password to access the password manager is strong and unique, and two-factor authentication is being utilized, this technique can be effective.

Another technique to assist in building strong, unique passwords, is to choose a repeatable pattern for your password, such as choosing a sentence that incorporates something unique about the website or account, and then using the first letter of each word as your password. For example, the sentence: "This is my January password for the Center for Internet Security website." would become "TimJp4tCfISw." This password capitalizes 5 letters within the sentence, swaps the word "for" to the number

"4," and adds the period to include a symbol. The vulnerability in this technique is that if multiple passwords from the same user are exposed it may reveal the pattern. Variations on this technique include using the first letters from a line in a favorite song or a poem.

# REFERENCES

## Research Thesis

1. Introducing Password Managers into Multiple-Password Environments (2016)

Sonny Johansson - Luleå University of Technology

## Reference websites

1. <https://link.springer.com/article/10.1007/s10207-019-00429-y>
2. <https://digitalguardian.com/blog/uncovering-password-habits-are-users-password-security-habits-improving-infographic>
3. <https://pypi.org/>
4. <https://wiki.python.org/moin/>
5. <https://www.w3schools.com/>
6. <https://www.welivesecurity.com/2020/06/26/what-is-password-manager-why-is-it-useful/>
7. <https://it.uottawa.ca/security/identity-authentication-theft#:~:text=Passwords%20provide%20the%20first%20line,all%20accounts%20on%20your%20computer.>
8. <https://www.cyberark.com/what-is/passwordless-authentication/>

# INDIVIDUAL CONTRIBUTION REPORT

## PyPASS - A Desktop Password Manager

SAKET PANDEY

1806413

**Abstract:** In today's technologically advancing world, cybersecurity is an all-time growing concern for every person, to protect their important and precious data from illegal miners and hackers. Our application helps to resolve such concerns in the form of a password manager, which primarily focuses on a single user, offline environment so far. We use a master password as the primary key to store all of the user's credentials for various sites, portals, etc. The entries are stored using encryption and hashing to ensure utmost security for the user's data.

**Individual contribution and findings:** My primary contribution to the project was the complete implementation of multi-factor authentication using hand gesture sequence recognition, alongwith the back end development and logical functioning of some security aspects of our application. This includes implementing database encryption and password hashing for storage, encoding the data-in-transit between the database, app functions and end user. I explored documentations and libraries for inspiration for the app and choosing various available libraries and technologies to be used in the final current draft of our project. I also am the *team leader* who planned and distributed work resulting in our team to successfully complete this project on time.

**Individual contribution to project report preparation:** I contributed to the following chapters/segments of the report:

- Key Features & App functions
- Architecture / Flow Diagram
- Project Showcase – Gesture-based Authentication as 2FA
- Project Planning
- Conclusion - Password Recommendations

Full Signature of Supervisor:

.....

Full signature of the student:



# INDIVIDUAL CONTRIBUTION REPORT

## PyPASS - A Desktop Password Manager

TUSHAR ABIR

1806532

**Abstract:** In today's technologically advancing world, cybersecurity is an all-time growing concern for every person, to protect their important and precious data from illegal miners and hackers. Our application helps to resolve such concerns in the form of a password manager, which primarily focuses on a single user, offline environment so far. We use a master password as the primary key to store all of the user's credentials for various sites, portals, etc. The entries are stored using encryption and hashing to ensure utmost security for the user's data.

**Individual contribution and findings:** My primary contribution to the program was in the back end development along with the logical and structural working of the application. I was responsible for creating the working logic behind the application, which includes the interactions of various functions with each other and how the application communicates with the database and the end user. I explored documentations and libraries for inspiration for the program and choosing various available libraries and technologies to be used in the final current draft of our project.

In the updated application, I continued to work on the back-end coding and implemented features, namely the password strength analyser and the password generator function.

**Individual contribution to project report preparation:** I contributed to the following chapters/segments of the report:

- Abstract
- About the project
- Working concept
- Technologies used
- Key features
- Project Showcase – Back end (Coding, Logic)
- Architecture / Use-Case Diagram

Full Signature of Supervisor:

.....

Full signature of the student:

**Tushar Abir**

# INDIVIDUAL CONTRIBUTION REPORT

## PyPASS - A Desktop Password Manager

SAMBEET KUMAR PANI

1806512

**Abstract:** In today's technologically advancing world, cybersecurity is an all-time growing concern for every person, to protect their important and precious data from illegal miners and hackers. Our application helps to resolve such concerns in the form of a password manager, which primarily focuses on a single user, offline environment so far. We use a master password as the primary key to store all of the user's credentials for various sites, portals, etc. The entries are stored using encryption and hashing to ensure utmost security for the user's data.

**Individual contribution and findings:** My contribution to the project was front-end development, and the GUI of the application. I contributed to make the visual elements that users see and interact within the application, from the login page to the settings menu. PyQt5, a python library has been used to easily develop the GUI. Online tutorials and documentation helped figure out the programming of this. My work included stuff like making sure a new window opens accordingly, settings the user configures, are saved, and also elements that interact and do things according to what the user wants them to.

**Individual contribution to project report preparation:** I contributed to the following chapters/segments of the report:

- Introduction
- Project Objectives
- Project Showcase (Frontend)
- Future Updates

Full Signature of Supervisor:

.....

Full signature of the student:

**Sambeet Kumar Pani**

# INDIVIDUAL CONTRIBUTION REPORT

## PyPASS - A Desktop Password Manager

DEVYANSH SINGH

1806375

**Abstract:** In today's technologically advancing world, cybersecurity is an all-time growing concern for every person, to protect their important and precious data from illegal miners and hackers. Our application helps to resolve such concerns in the form of a password manager, which primarily focuses on a single user, offline environment so far. We use a master password as the primary key to store all of the user's credentials for various sites, portals, etc. The entries are stored using encryption and hashing to ensure utmost security for the user's data.

**Individual contribution and findings:** I contributed to the project, working on the authentication process of the application. I have worked on the implementation of a TOTP (Time based OTP) solution. The key to all of the methods discussed below is that once a user has verified his or her username and password, he or she must enter a dynamic password that is constantly generated and changed before they can access the system. The library used here is OneTimePass.

**Individual contribution to project report preparation:** I contributed to the following chapters/segments of the report:

- Project Planning
- Project Showcase – Time-based OTP system as 2FA

Full Signature of Supervisor:

.....

Full signature of the student:

**Devyansh Singh**



# PLAGIARISM REPORT

(Main Report Content ONLY)

