

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi – 590018, Karnataka



Python Application Programming **A mini project report on** **Password Manager**

Submitted By:

Shreyas M

(1GA16CS143)

Surya Pratap M G

(1GA17CS421)

Under the Guidance of
Sushmitha S
Assistant Professor



Department of Computer Science and Engineering

(Accredited by NBA 2019-2022)

Global Academy of Technology

Rajarajeshwarinagar, Bangalore – 560098

2019-2020

Global Academy of Technology

Department of Computer Science and Engineering



Certificate

This is to certify that the project entitled "**Password Manager**" is a bonafide work carried out by **Shreyas M (1GA16CS421)**, **Surya Pratap M G(1GA17CS421)** as a part of assignment in Python Applications Programming in Computer Science and Engineering during the year 2019-2020.

Max. marks	Marks obtained	Faculty Name and Signature
05		

ABSTRACT

We all use the platforms where we have to maintain the passwords to keep our account safe but also we often change our passwords just because we have forgotten it. Here is a solution for the problem we are facing by forgetting the passwords.

Password Manager helps you to store your password of your different accounts under a single platform. Here you can store your passwords with their services and access them whenever you want to.

So you now just have to remember a single password of the password manager instead of the multiple passwords for so many platforms.

TABLE OF CONTENTS

TOPIC	PAGE NO
1. INTRODUCTION	
1.1 Python Programming Language	
1.2 Applications of Python	
2. SYSTEM REQUIREMENTS	
2.1 Software Requirements	
2.2 Hardware Requirements	
3. IMPLEMENTATION AND RESULTS	
3.1 Project Code	
3.2. Results	
CONCLUSION	
REFERENCES	

ORGANIZATION OF THE REPORT

The report is divided into various chapters and is organized as follows:

Chapter 1: Introduction

This chapter includes brief introduction to Python Programming Language and its applications.

Chapter 2: System requirements

This chapter includes details of hardware and software requirements necessary for the execution of the project.

Chapter 3: Implementation and Results

This chapter includes the program code of the project and the results of successful runs of the code.

Conclusion

This section includes the conclusion about the project.

References

This section includes the bibliographical references used for the development of the project.

CHAPTER 1

INTRODUCTION

1.1. Python Programming language

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python was conceived in the late 1980s by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC language (itself inspired by SETL), capable of exception handling and interfacing with the Amoeba operating system. Its implementation began in December 1989.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Python was conceived in the late 1980s as a successor to the ABC language. Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system capable of collecting reference cycles. Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3.

The Python 2 language, i.e. Python 2.7.x, was officially discontinued on 1 January 2020 (first planned for 2015) after which security patches and other improvements will not be released for it.[30][31] With Python 2's end-of-life, only Python 3.5.x and later are supported.

Python interpreters are available for many operating systems. A global community of programmers develops and maintains CPython, an open source reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and CPython development.

1.2. Applications of Python programming language

Python supports cross-platform operating systems which makes building applications with it all the more convenient. Some of the globally known applications such as YouTube, Bit Torrent, Drop Box, etc. use Python to achieve their functionality.

- Web Development

Python can be used to make web-applications at a rapid rate. Why is that? It is because of the frameworks Python uses to create these applications. There is common-backend logic that goes into making these frameworks and a number of libraries that can help integrate protocols such as HTTPS, FTP, SSL etc. and even help in the processing of JSON, XML, E-Mail and so much more.

- Game Development

Python is also used in the development of interactive games. There are libraries such as PySoy which is a 3D game engine supporting Python 3, PyGame which provides functionality and a library for game development. Games such as Civilization-IV, Disney's Toontown Online, Vega Strike etc. have been built using Python.

- Machine Learning and Artificial Intelligence

Machine Learning and Artificial Intelligence are the talks of the town as they yield the most promising careers for the future. We make the computer learn based on past experiences through the data stored or better yet, create algorithms which makes the computer learn by itself.

- Data Science and Data Visualization

Data is money if you know how to extract relevant information which can help you take calculated risks and increase profits. You study the data you have, perform operations and extract the information required. Libraries such as Pandas, NumPy help you in extracting information.

- Embedded Applications

Python is based on C which means that it can be used to create Embedded C software for embedded applications. This helps us to perform higher-level applications on smaller devices which can compute Python.

- CAD Applications

Computer-Aided Designing is a very complicated application to make as many things have to be taken care of. Objects and their representation, functions are just the tip of the iceberg when it comes to something like this. Python makes this simple too and the most well-known application for CAD is Fandango.

CHAPTER 2

SYSTEM REQUIREMENTS

2.1 Software Requirements

- Operating system : Windows 7/8/10
- IDE : Python IDLE 3.6
- DB Browser (SQLite)

2.2 Hardware Requirements

Processor – Intel Core 2 Duo or later versions

- Memory - 356MB RAM
- 60MB Hard Disk Drive
- Mouse or other Pointing Device
- Keyboard
- Display Device

CHAPTER 3

IMPLEMENTATION AND RESULTS

3.1 About Project

This project demonstrates the working of an Password Manager in the simplest possible way i.e. You can store a password of different services and retrieve the passwords in this program.

3.2 Project code

```
import sqlite3

ADMIN_PASSWORD = "12"

connect = input("What is your password?\n")

while connect != ADMIN_PASSWORD:
    connect = input("Wrong Password...!! Try Again..\n")

conn = sqlite3.connect('pass_manager.db')

def get_password(service1):
    cursor = conn.execute("SELECT (PASS_KEY) from KEYS WHERE SERVICE=" +
    ''' + service1 + ''')
    cursor1 = ""
    for row in cursor:
        cursor1 = row[0]
    return cursor1
```

```
def add_password(service1, np1):
    command = 'INSERT INTO KEYS (PASS_KEY, SERVICE) VALUES    (%s, %s);' %
('"' + np1 + '"', '"' + service1 + '"',)
    conn.execute(command)
    conn.commit()
    print("\n Password stored Successfully")

if connect == ADMIN_PASSWORD:

    print("\n" + "*" * 15)
    print("Commands:")
    print("gp = get password")
    print("sp = store password")
    print("*" * 15)
    input_ = input("")

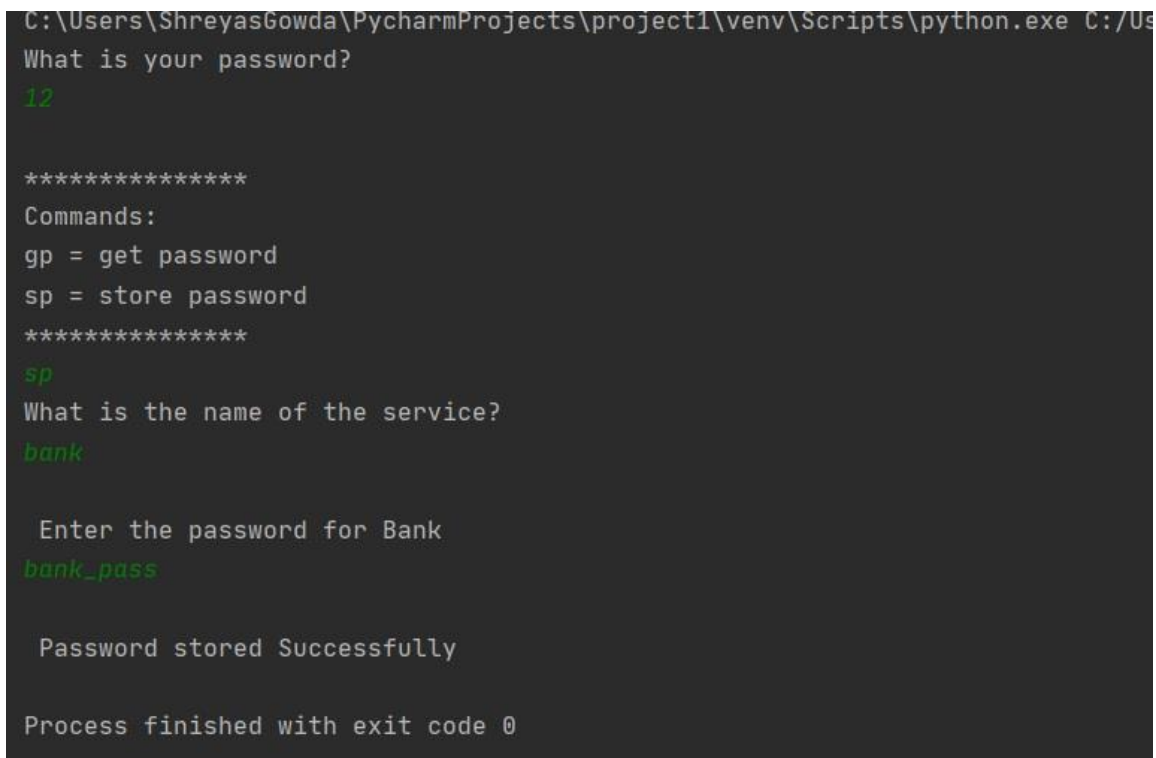
    if input_ == "sp":
        service = input("What is the name of the service?\n")
        np = input("\n Enter the password for " + service.capitalize() +
"\n")
        add_password(service, np)
        exit()

    if input_ == "gp":
        service = input("What is the name of the service?\n")
        print("\n" + service.capitalize() + " password:\n" +
get_password(service))
        exit()

    else:
        print('Wrong Selection')
```

Backend:

```
BEGIN TRANSACTION;
CREATE TABLE IF NOT EXISTS "KEYS" (
    "SERVICE" TEXT,
    "PASS_KEY" TEXT NOT NULL,
    PRIMARY KEY("SERVICE")
);
INSERT INTO "KEYS" VALUES ('facebook','password_facebook');
INSERT INTO "KEYS" VALUES ('insta','insta_pass');
INSERT INTO "KEYS" VALUES ('watsapp','watsapp_pass');
INSERT INTO "KEYS" VALUES ('bank','bank_pass');
COMMIT;
```

3.3 Results

```
C:\Users\ShreyasGowda\PycharmProjects\project1\venv\Scripts\python.exe C:/Us
What is your password?
12

*****
Commands:
gp = get password
sp = store password
*****
sp
What is the name of the service?
bank

Enter the password for Bank
bank_pass

Password stored Successfully

Process finished with exit code 0
```

Snapshot 3.1: Storing Password

```
What is your password?  
12  
  
*****  
Commands:  
gp = get password  
sp = store password  
*****  
gp  
What is the name of the service?  
insta  
  
Insta password:  
insta_pass  
  
Process finished with exit code 0  
|
```

Snapshot 3.2: Getting Password

```
What is your password?  
12  
  
*****  
Commands:  
gp = get password  
sp = store password  
*****  
ap  
Wrong Selection  
  
Process finished with exit code 0  
|
```

Snapshot 3.3: Wrong Selection

DB Browser for SQLite - C:\Users\ShreyasGowda\PycharmProjects\project1\pass_manager.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project

Database Structure Browse Data Edit Pragmas Execute SQL

Table: KEYS

	SERVICE	PASS_KEY
	Filter	Filter
1	facebook	password_facebook
2	insta	insta_pass
3	watsapp	watsapp_pass
4	bank	bank_pass

Snapshot 3.4: Backend Data Storage

CONCLUSION

The project has achieved its core functionality by allowing users to make simple functionalities like storing the password and retrieving it but it should also be improvised by implementing the encryption and decryption of the passwords.

The application has been tested with various test cases to ensure users are not able to store and retrieve the passwords.

The application has a backend part to view the data stored.

REFERENCES

Bibliography:

- [1] Charles Dierbach, "Introduction to Computer Science Using Python", 1st Edition, Wiley India Pvt Ltd. ISBN-13: 978-8126556014
- [2] Mark Lutz, "Programming Python", 4th Edition, O'Reilly Media, 2011. ISBN-13: 978-9350232873
- [3] Wesley J Chun, "Core Python Applications Programming", 3rd Edition, Pearson Education India, 2015. ISBN-13: 978-9332555365

Websites:

- [1] <https://docs.python.org/3/library>
- [2] <https://stackoverflow.com/>
- [3] <https://geeksforgeeks.com/>