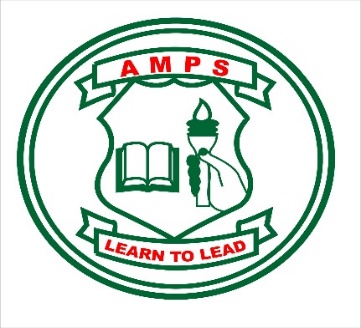
**ALWIN MEMORIAL PUBLIC SCHOOL**

**Indira Nagar, Selaiyur, Chennai – 600073**

****

**AISSCE 2020 – 2021**

**COMPUTER SCIENCE**

**E**

**12**

**MARIAM MATHEW PAZHAYAKALAM**

**NAME ……………………………………………..**

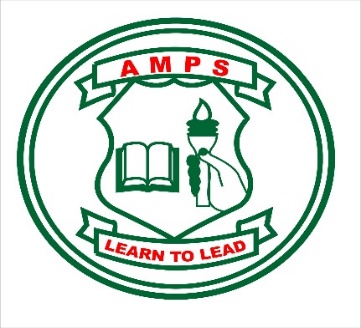
**REGISTER NO……………………………………**

**STD.….…………………SEC……………………..SUBJECT.………………………………………….**

**CryptOn**

**ALWIN MEMORIAL PUBLIC SCHOOL**

**Indira Nagar, Selaiyur, Chennai – 600073**

****

**BONAFIDE CERTIFICATE**

**This is to certify that the project titled \_\_\_\_\_\_\_\_\_\_\_ is a Bonafide work done by \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_in partial fulfillment of the requirements in computer science practical as prescribed by CBSE for AISSCE 2020 – 2021.**

**MARIAM MATHEW PAZHAYAKALAM**

**CryptOn**

**REGISTER NO. \_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**TEACHER-IN-CHARGE PRINCIPAL**

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**ACKNOWLEDGEMENT**

We would like to bring to light those who have helped us in the completion of our project work, without whom, this work would not have reached its destination.

At the very outset, we thank the God Almighty for showering his blessings and being a source of guidance and wisdom throughout the study, without whom, no human achievement is possible.

We would also like to extend our deepest gratitude to**Dr.N.Vijayan,M.A,M.Sc.,M.Ed.,M.L.,Ph.D**., Chairman, **Mr.V.Aldous R.Huxley,Vice-Chairman**, **Mrs.Rachel Georgiana**, Director, Mrs**.Jayanthi Sivakumar**, Dean, **Ms. Nazrina**, Principal,**Mr. Jesudas kirupakaran**, Vice-Principal, Alwin Memorial Public School, for providing all the facilities to carry out our work.

We would like to pay my special regardsto**Mrs.P.Sakuntalakarpagavalli**, Principal-Advisor, **Mr.SankaraSubramania Pillai**, Controller of Examinations, Alwin Memorial Public School , for profound belief in our work and our abilities

We are indebt to our Institutional guide**Mrs.ShanmugapriyaSabaridasan**, HOD of computer science department for her Rejuvenating inspiration, kind co-operation, valuable guidance and suggestions which helped us to complete every aspect of this project work.

We are immensely thankful to **Mrs.Ananthy**, for the valuable guidance, suggestions and constant support that she provided throught the project.

We express our sincere and deep sense of gratitude and heartfelt thanks to our **staffs**, and **mentors**, for their meticulous guidance, transcendent suggestions, constructive criticism and constant encouragement right from conceptualization of the project work to the preparation of the project Crypton.

We would like to thank all **our friends and classmates ,** faculty, Alwin Memorial Public School, for their kind help and good communication, which helped us for the successful completion of this work.

Words are simply not inadequate for penning down that deep sense of gratitude to **our Family**, without whose sacrifice, our professional career would never have seen the light of day. Last but not the least we owe a great deal to **our teaching and non-teaching staff** without whose co-operation this dissertation would not have been possible.

**CONTENTS**

* + ABSTRACT
  + REQUIREMENT ANALYSIS
    - DESIGN AND APPLICATION
    - HARDWARE AND SOFTWARE REQUIREMENTS
  + FLOW CHART
    - PROJECT FLOW
    - PROGRAM FLOW
  + SOURCE CODE
  + OUTPUTS
  + FUTURE ENHANCEMENT
  + BIBLIOGRAPHY

# ABSTRACT

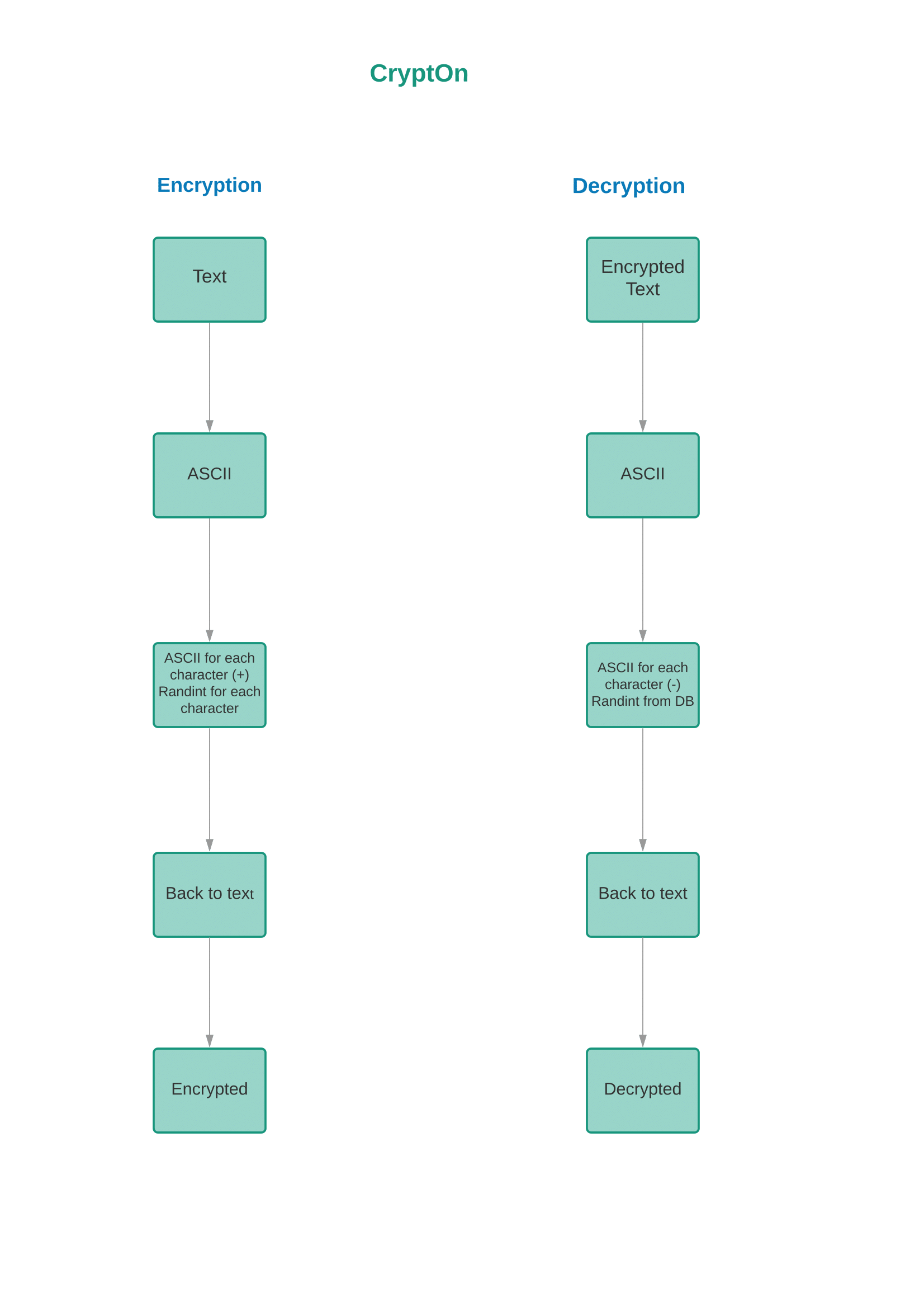
Encryption is the process of changing information in such a way as to make it unreadable by anyone except those possessing special knowledge [key] that allows them to change the information back to its original, readable form. Encryption is necessary to safeguard data. Encryption is important because it allows you to securely protect data that you don’t want anyone else to have access to.

So, we worked on a project that is almost completely based on Caesar's Cipher.In cryptography, a Caesar cipher, also known as Caesar's cipher, the shift cipher, Caesar's code or Caesar shift, is one of the simplest and most widely known encryption techniques. It is a type of substitution cipher in which each letter in the plaintext is replaced by a letter by some fixed number of positions down the alphabet. The encryption step performed by a Caesar cipher is often incorporated as part of more complex schemes. Thus, to cipher a given text we need an integer value, known as shift which indicates the number of position each letter of the text has been moved down. In this project Text is converted to Unicode, random generated number for length the same as the text is Added and then Converted back to text

The encryption can be represented using modular arithmetic by first transforming the letters into numbers, according to the scheme. We are implementing the algorithm on Python 3 and using PyMySQL module, we are able to store encrypted data on a database. Basically,PyMySQL is an interface for connecting to MySQL database server from Python. It implements the Python Database and contains a pure-Python MySQL client library. The goal of PyMySQL is to be a drop-in replacement for mysql.connector.

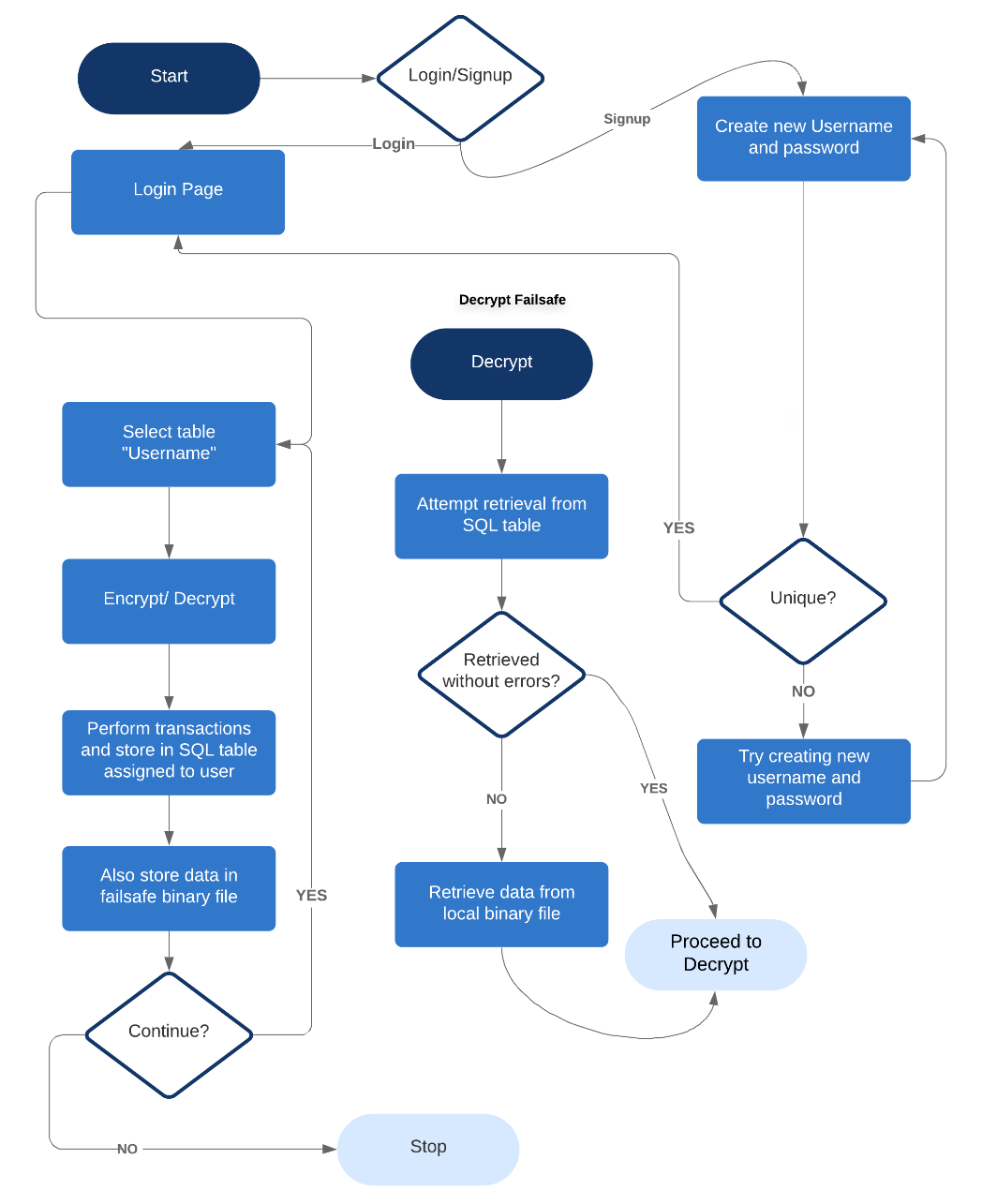
**REQUIREMENT ANALYSIS**

**Design and Application:**



The initial design of the program was quite just the proof of concept. It barely had any features or a contemplatable user interface.

This design had only two parts and all it did was receive text from the user for encryption, convert it to the constituent character’s Unicode values and add a random integer to each of the character’s Unicode value. When converted back to text, it will be a string of new characters or symbols, rendering the text unreadable in its present format. To make it readable again, we made a decryption unit which just did the same as encryption but in reverse: it subtracted the random numbers from the Unicode value of the encrypted text. We called this the engine of this project since it is the core module for its functionality. Of Course, this engine’s code was modified and altered so many times in order to scale up to an user friendly and error-free executable program.

The newest design was sculpted to maximize scalability and to provide room for future enhancement.

In this latest design, the program can work heedlessly with or without the presence of an SQL server since this doesn’t completely rely on it. During encryption, the runtime input variables are stored in SQL as well as a binary file at the same time. The key feature of this new formulation is to have a failsafe in case the retrieval from SQL doesn’t go as expected; thus, solving the biggest problem in the initial build. The initial build couldn’t handle more than a word for encryption and retrieval from the SQL table was quite painful since SQL has a hard time bringing back a really long Unicode string without encountering an error of sorts. This build fixed that with the binary file. By all means, it attempts to retrieve data from the SQL and only in case of an error, the binary file protocol is invoked. Our program can now handle strings of any length.

The application of this project is very large but the market for this is quite narrow since this is a very vulnerable security tool. But still, this can be implemented to satisfy the necessities of small-scale businesses and schools. Our target clients are

schools since most schools design their question papers in the computer labs and students have access to the same system as the teachers and it doesn’t take long to browse, search for the file and secretly email it to themselves. Annually, 12% of schools’ in India have their question leaked before the internal exam due to this malpractice and often are not able to take action since they cannot pinpoint on the one who did this. We are willing to provide them this encryption solution which would solve this crisis. Even if the question paper was stolen, they would not be able to retrieve what’s stored in it without the encryption key. Similar applications exist for small-scale businesses’ and their employees. In India, 32.64% of small business with fresh IPOs lose crores due to private stock manipulation and shut down because of information being sold to other organizations without a trace or simply without the means to trace it. What was that?“It would be much easier of the company had its data encrypted with an affordable algorithm?” That’s our cue.

**HARDWARE AND SOFTWARE REQUIREMENTS**

1. **Minimum Hardware Requirements:**

RAM: 128MB,

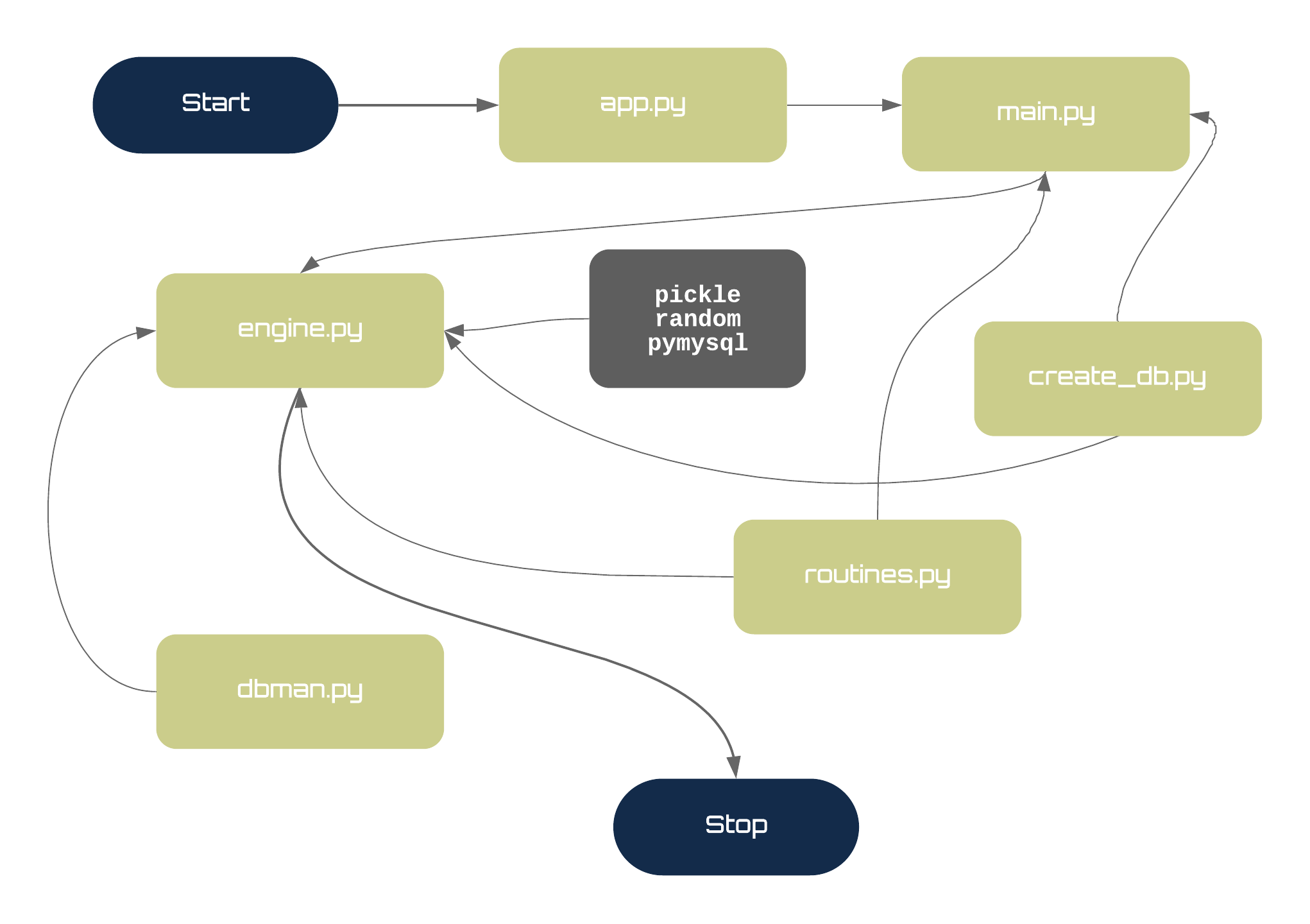
CPU: 1GHz

STORAGE: 100MB

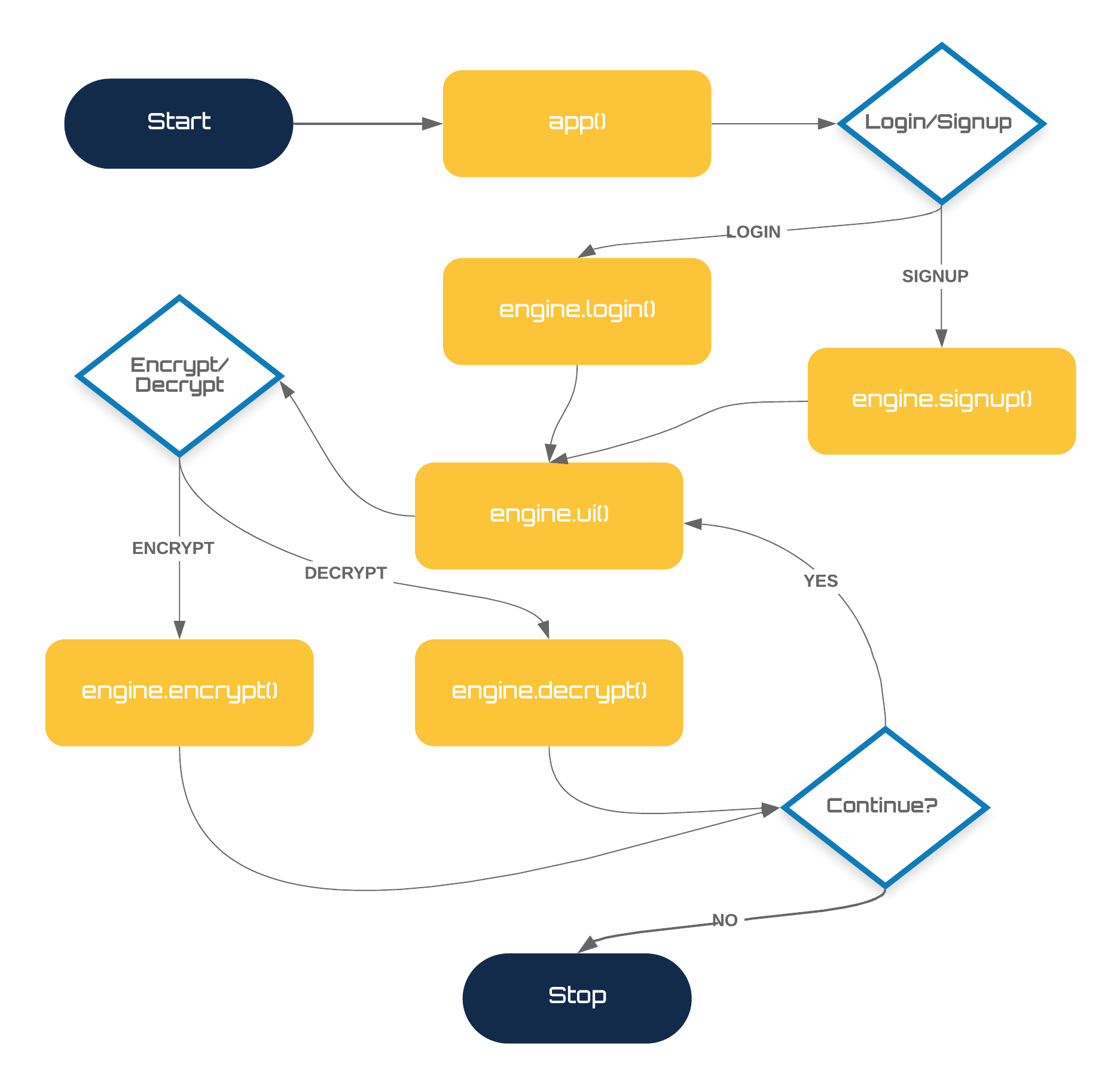
1. **Software Requirements:**

* Android/Windows/macOS/Linux/Unix/iOS
* Any platform with Python and SQL
* Python 3.8
* MySQL Server
* Pip package manager
* PyMySQL module for Python
* Pickle module for Python
* Random module for Python

**FLOW CHART**

**Project Flow**

**Program Flow**

****

**ALGORITHM**

Here we convert a user provided text into the Unicode equivalent for each of the character Unicode. Unicode is an international encoding standard for use with different languages and scripts, by which each letter, digit, or symbol is assigned a unique numeric value that applies across different platforms and programs.And Unicode is a widely used standard for encoding text documents on computers. This encoding system not only lets a computer store a document as a series of numbers, but also lets it share such documents with other computers that use the Unicode system. We generate a random integer of the same length and add it to the Unicode value of the text. We convert this new number back into text form. We will store this random integer i.e., the encryption and encrypted text in a database and a binary file to decrypt later.

When the user attempts to decrypt the text later, the engine will break down the encrypted text into its constituent characters and convert into its Unicode counterpart. Each character’s Unicode will be subtracted by the encryption key. This will return the original, readable, text.

**Example:**

**Input: ‘**hello**’**

**[‘**h’,’e’,’l’,’l’,’o’**] [104,101,108,108,111] ± [5,9,8,6,3]**

**[‘m’,’n’,’t’,’r’,’r’] [109,110,116,114,114]**

**Output: ‘**mntrr**’**

**SOURCE CODE**

**1.app.py**

from main import app

app()

**2.main.py**

import engine

import routines

import create\_db

create\_db.create()

create\_db.db\_select()

create\_db.profiles()

routines.cls()

def app():

try:

print(routines.logo)

print("Hi!")

response1=(input("Do you have an account? (Y|N): ")).lower()

if response1 == 'y':

engine.login()

elif response1 == 'n':

engine.signup()

else:

routines.invalid\_res()

except KeyboardInterrupt:

print("\n\n\n\nKeyBoard Interrupt Detected.\nThank you for using CryptOn. We hope to

See you again.")

print(routines.logo)

exit(0)

**3.engine.py**

import dbman

import random

import create\_db

import routines

import pickle

def encrypt(username):

string = input("Enter the word(s) you want to encrypt: ")

string\_ord = []; n = len(string)

task = [string\_ord.append(ord(i)) for i in list(string)]

enc\_key\_list = list(str(random.randint(10\*(n-1), (10\*n)-1)))

enc\_string = ''

for i in range(len(string)):

x = string\_ord[i]+int(enc\_key\_list[i])

enc\_string += chr(x)

enc\_key = ''

for i in enc\_key\_list:

enc\_key+=i

enc\_key = int(enc\_key)

dbman.cursor.execute(f"INSERT INTO{username}VALUES(%s,%s);",(enc\_string,enc\_key))

with open('failsafe.dat', 'rb+') as failsafe:#Failsafe

r = pickle.load(failsafe)

r['enc\_string'] = enc\_key

with open('failsafe.dat', 'wb') as failsafe:

pickle.dump(r,failsafe)

print("Your text is encrypted. You may need this to decrypt the text later:",enc\_string)

def decrypt(username):

string = input("Enter the encrypted text: ")

string\_ord = []; n = len(string)

task = [string\_ord.append(ord(i)) for i in list(string)]

dbman.cursor.execute(f"Select \* from {username};")

task2 = dict(dbman.cursor.fetchall())

with open('failsafe.dat','rb') as failsafe:

task3 = pickle.load(failsafe)

if string in task2.keys():

enc\_key = task2[string]

enc\_key\_list = list(str(enc\_key))

dec\_string = ''

for i in range(n):

x = string\_ord[i]-int(enc\_key\_list[i])

dec\_string += chr(x)

print("Your text has been decrypted: ",dec\_string,'\n ')

elif string in task3.keys():

print("Your text has been decrypted: ",task3[string],'\n')

elif string not in task2.keys():

print("No record found. Please try again.")

decrypt(username)

def signup():

dbman.cursor.execute("Select \* from Profiles;")

unique\_check = dict(dbman.cursor.fetchall())

username = (input("Enter username: ")).strip()

if username in unique\_check.keys():

print("Username Taken. Please try again.")

signup()

else:

password = input("Enter a unique password: ")

q = "INSERT INTO PROFILES VALUES(%s,%s);"

dbman.cursor.execute(q,(username,password))

create\_db.users(username)

routines.cls()

print('Successfully signed up!')

ui(username)

def login():

username = (input("Enter username: ")).strip()

dbman.cursor.execute("Select \* from Profiles;")

cred\_check = dict(dbman.cursor.fetchall())

if username in cred\_check.keys():

password = input("Enter password: ")

if cred\_check[username] == password:

ui(username)

elif cred\_check[username] != [password]:

routines.login\_err()

login()

elif username not in cred\_check.keys():

print(f"Username {username} not found!! Try again.")

login()

def ui(username):

print(f"Welcome {username}. What would you like to do?")

response2=input("Encryption or Decryption? (Enter 1 or 2): ")

if response2 == '1':

encrypt(username)

response3 = (input("Do you wish to continue? (Y|n): ")).lower()

if response3 == 'y':

ui(username)

elif response3 == 'n':

exit("Thank you for using CryptOn.")

else:

routines.invalid\_res()

elif response2 == '2':

decrypt(username)

response3 = (input("Do you wish to continue? (Y|n): ")).lower()

if response3 == 'y':

ui(username)

elif response3 == 'n':

print(routines.logo,'\n')

exit("Thank you for using CryptOn.")

else:

routines.invalid\_res()

else:

routines.invalid\_res()

**4.create\_db.py**

import dbman

def create():

dbman.cursor.execute("CREATE DATABASE IF NOT EXISTS CRYPTON;")

def profiles():

dbman.cursor.execute("CREATE TABLE IF NOT EXISTS PROFILES(USERNAME

TEXT, PASSWORD TEXT);")

def db\_select():

dbman.cursor.execute("USE CRYPTON;")

def users(username):

dbman.cursor.execute(f"CREATE TABLE IF NOT EXISTS {username}(enc\_text

longtext, enc\_key longtext);")

**5.routines.py**

import os

def cls():

os.system('cls')

def invalid\_res():

print("Sorry, that's an invalid response.")

def login\_err():

print("Invalid username or password.")



**6.dbman.py**

import pymysql

import os

connection = pymysql.connect(host='localhost',

user='root',

password='1234',

autocommit=True)

cursor = connection.cursor()

**FUTURE ENHANCEMENTS**

CryptOn is capable of massive future enhancements which can exponentially improve the quality and functioning of the program. Currently, this project is a CLI (Command-Line Interface) tool so, the first task would be to construct a scalable, cross-platform graphical interface that is designed for the ease of regular usage. The current build, though perfectly usable for compact necessities, has major vulnerabilities that can be exploited using modern day technology. The encryption algorithm has to be upgraded to a more secure and efficient 256-bit encryption like the SHA256/MD5 or better.

Post-Haste, Two-Factor Authorization will be built into the engine with prompt for an OTP sent to the registered mobile number. This would lay the groundwork for the Cloud Suite integration with which CryptOn enters the IT Product and Service Industry with its suite of Security products like On-Device encrypted data indexing (Access encrypted files without decrypting) and Server Hashing utilities.

The Go-To-Market strategy would be to team up with ISPs and service companies for small scale network and device security solutions including remote encryption key requests and secured data decryption using the presence of pre-defined devices on the network. Furthermore, paid cloud storage service that would be more secure compared to traditional remote storage providers.

**BIBLIOGRAPHY**

**websites:**

* https://[www.google.co.in](http://www.google.co.in)
* https://en.wikipedia.org
* https://www.geeksforgeeks.org
* <https://www.lucidchart.com>
* <https://www.canva.com>
* <https://www.python.org>
* <https://pypi.org>
* <https://www.youtube.com>

**books:**

* computer science with python by Sumitha Aurora
* core python programming by Dr.Nageshwara Rao
* computer science with python by Preethi Aurora
* Python programming for beginners by Guido Van Rossum