ENCRYPTION USING SUBSTITUTION TECHNIQUES

A Project Report

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

# Bachelor of Technology in

Department of Computer Science and Engineering

By

K SIDHARTH RAO (2010030443)

ABHIRAM VIDIYIYALA (2010030180)

under the supervision of

# < Name and

# Designations of the Supervisor>



Department of Computer Science and Engineering

K L University Hyderabad,

Aziz Nagar, Moinabad Road, Hyderabad – 500 075, Telangana, India.

March, 2022

**Declaration**

The Project Report entitled “ENCRYPTION USING SUBSTITUTION TECHNIQUES” is a record of bonafide work of ABHIRAM VIDIYIYALA (2010030180), K SIDHARTH RAO(2010030443), submitted in partial fulfillment for the award of B.Tech in the Department of Computer Science and Engineering to the K L University, Hyderabad. The results embodied in this report have not been copied from any other Departments/University/Institute.

K SIDHARTH RAO (2010030443)

ABHIRAM VIDIYIYALA (2010030180)

**Certificate**

This is to certify that the Project Report entitled “ENCRYPTION USING SUBSTITUTION TECHNIQUES” is being submitted by ABHIRAM VIDIYIYALA (2010030180), K SIDHARTH RAO (2010030443), submitted in partial fulfillment for the award of B.Tech in Computer Science and Engineering to the K L University, Hyderabad is a record of bonafide work carried out under our guidance and supervision.

The results embodied in this report have not been copied from any other departments/ University/Institute.

## Signature of theSupervisor

Name andDesignation

## Signature oftheHOD Signature of the ExternalExamine

**ACKNOWLEDGEMENT**

First and foremost, we thank the lord almighty for all his grace & mercy showered upon us, for completing this project successfully.

We take grateful opportunity to thank our beloved who has given constant encouragement during our course and motivated us to do this project. We are grateful to our Principal **Dr. L. Koteswara Rao** who has been constantly bearing the torch for all the curricular activities undertaken by us.

We pay our grateful acknowledgement & sincere thanks to our Head of the Department **Dr. Chiranjeevi Manike**for her exemplary guidance, monitoring and constant encouragement throughout the course of the project. We thankof our department who has supported throughout this project holding a position of supervisor.

We whole heartedly thank all the teaching and non-teaching staff of our department without whom we won’t have made this project a reality. We would like to extend our sincere thanks especially to our parent, our family members and friends who have supported us to make this project a grand success.

**ABSTRCAT**

As multimedia applications are used increasingly, security becomes an important issue of communication and storage of data. In the current scenario, where there are many incidents of data hacking data leakages, it is extremely important to store the data in a very secured manner. In this project we have discussed about one of the encryption techniques which will help us in ensuring the security of data while storing and transferring it from one source to the other.

# **Table of Contents**

# Introduction

# 

# Literature Survey

# 

# Selected model and Techniques

# Hardware and software components

# Proposed System

# Implementation

# Result Discussion

# Conclusion and Future Work

# 

**INTRODUCTION**

Security is one of the issues often face by web developer and software engineer. In today's life, computer and networking is essential to today's communication and transferring of data is essential when doing online banking, data transferring, sharing and more.

The art and science of keeping messages secure is called cryptography. It is practiced by cryptographers, and cryptanalyst are practitioners of cryptanalysis, the art and science of breaking the cipher text. Earlier the requirement of information security within an organization was primarily provided by physical and administrative means. But the concept of network security became quite evident with the introduction of computers and later with introduction of distributed systems.

The need of cryptographic algorithm is to avoid threat to integrity, confidentiality and availability.

**LITERATURE SURVEY**

Substitution technique is a classical encryption technique where the characters present in the original message are replaced by the other characters or numbers or by symbols.

Substitution Technique:

1. Caesar Cipher
2. Monoalphabetic Cipher
3. Playfair Cipher
4. Hill Cipher
5. Polyalphabetic Cipher
6. One-Time Pad

Cryptology, which is the study of cryptosystems, encompasses two disciplines: cryptography and cryptanalysis. It is one of the major areas of study in information security. Cryptography is defined as the science and art of encrypting and decrypting data using some special measures. Encryption is the method of disguising plaintext in such a way as to hide its substance while decryption (which is the opposite of encryption) is unhiding the substance by changing the ciphertext to its original plaintext. Cryptanalysis, on the other hand, is the branch of science which deals with breaking the codes and extracting the hidden meaning, while the whole system which comprises both cryptography and cryptanalysis is called cryptosystem.

**Cryptographic Algorithms (Ciphers)**

A cipher is an algorithmic function which converts plaintext messages into unreadable forms by applying a set of transformation techniques to each letter in the plaintext. The particular transformations employed at any time are controlled by a secret key, called cryptographic key, used at that time. The security of the ciphertext is said to rest majorly on the secrecy of this key.

In a study in 2009, Poschmann wrote that ciphers can be classified using several criteria. According to one of such criteria, the ciphers are classified as symmetric key and asymmetric key. In symmetric key ciphers, the same key is used for both encryption and decryption. The key used for encryption is called the public key and the key used for decryption is called the private key. The public key can be revealed without compromising the security of the system while the corresponding private key, however, must not be revealed to any party.

Poschmann (2009) further wrote that, in symmetric encryption, ciphers can be classified into stream ciphers and block ciphers; stream ciphers obtain ciphertext by using the XOR of the plaintext and keystream (bi-wise).

They are grouped into two: synchronous stream cipher, whose key sequence does not depend on the plaintext and ciphertext but only on the previous elements of the key sequence and the initial key, e.g. One-time password (OTP); and asynchronous stream cipher, whose keystream depends on the plaintext or ciphertext, e.g., Cipher Feedback mode (CFB). Other examples of stream ciphers include RC4 and SEAL.

Block ciphers, on the other hand, operate on a fixed length block size. It can be considered simply as a large lookup-table (substitution cipher). In particular, identical plaintext blocks encrypt to identical cipher text blocks. Examples include Data Encryption Standard (DES), 3Data Encryption Standard (3DES), Advanced Encryption Standard (AES), Blowfish, etc.

The earliest known use of a substitution cipher, and the simplest, was by Julius Caesar. Since Julius Caesar used an additive cipher to communicate with his officers; for this reason, additive ciphers are sometimes referred to as the Caesar Cipher. In Cryptography, Caesar cipher is one of the most widely known encryption-decryption algorithms.

**HARDWARE AND SOFTWARE REQUIREMENTS**

HARDWARE:

* Modern Operating System:
  + Windows 7 or 10
  + Mac OS X 10.11 or higher, 64-bit
  + Linux: RHEL 6/7, 64-bit (almost all libraries also work in Ubuntu)
* x86 64-bit CPU (Intel / AMD architecture)
* 4 GB RAM
* 5 GB free disk space

SOFTWARE:

* Python3
* Pycharm community (IDE)

**PROPOSED SYSTEM**

To carry out the process of encryption, we will be working on the Substitution Technique called Caesar Cipher. In this encryption, it uses the substitution cipher in which each letter in the plaintext is replaced by some fixed number of positions down the alphabet.

This method is names after Julius Caesar who using this method to communicate with his generals.

The following terms will be used in this substitution technique:

Plain-text: This in an intelligible piece of information i.e., original text that needs to transferred safely to the receiver. It is the main input to the encryption algorithm.

Secret Key: This is another input to the encryption and decryption algorithm, which is the main component used for converting the plain-text to cipher-text i.e., an unintelligible form which has the useful content hidden in a way.

Encryption Algorithm: This is the actual process by which we are converting the plain-text into ciphertext.

Cipher-text: This is the output of encryption process in which we are taking plain-text and secret key as input and processed by encryption algorithm. The cipher-text can be understood as a scrambled piece of text which has useful information in secret form.

Decryption Algorithm: This algorithm is the reverse of the encryption algorithm which takes in cipher-text and secret key as inputs and produces plain-text as the output.

A Caesar cipher is a simple method of encoding messages. Caesar ciphers use a substitution method where letters in the alphabet are shifted by some fixed number of spaces to yield an encoding alphabet. A Caesar cipher with a shift of 11 would encode an A as a B, an M as an N, and a Z as an A, and so on.

* The formula of encryption of the text is: En (x) = (x + n) mod 26

En (x) = Cipher-text

x = Numeric value of each alphabet in the text

n = Search-key

**IMPLEMENTATION**

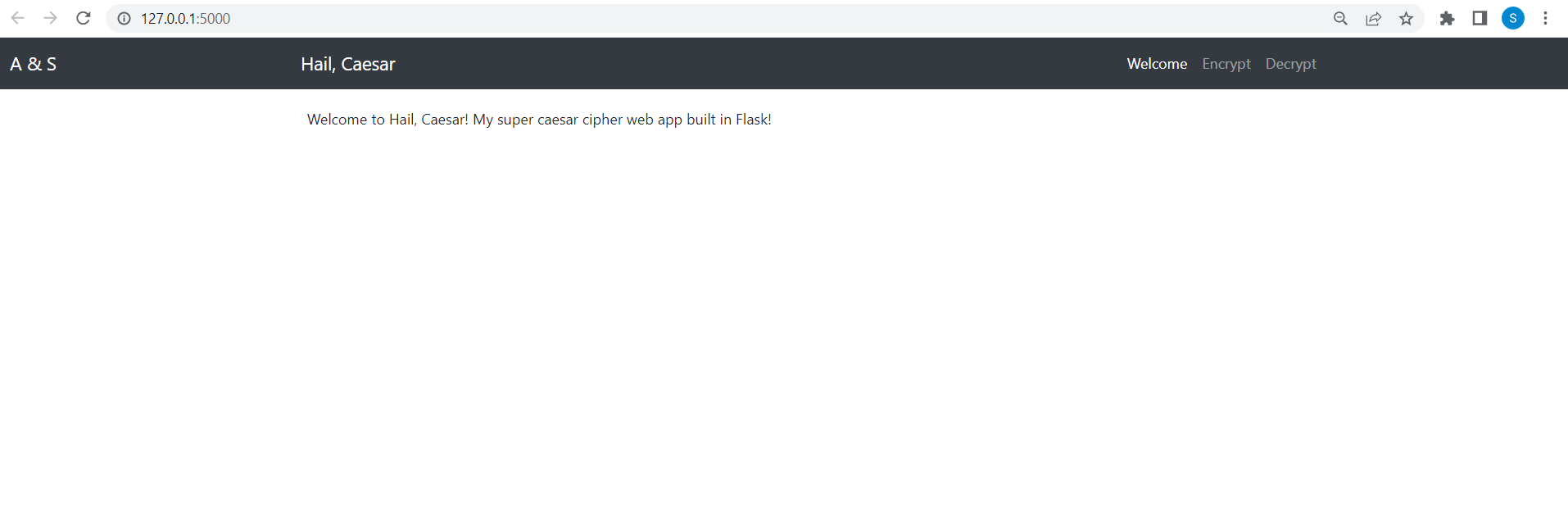
**CODE**

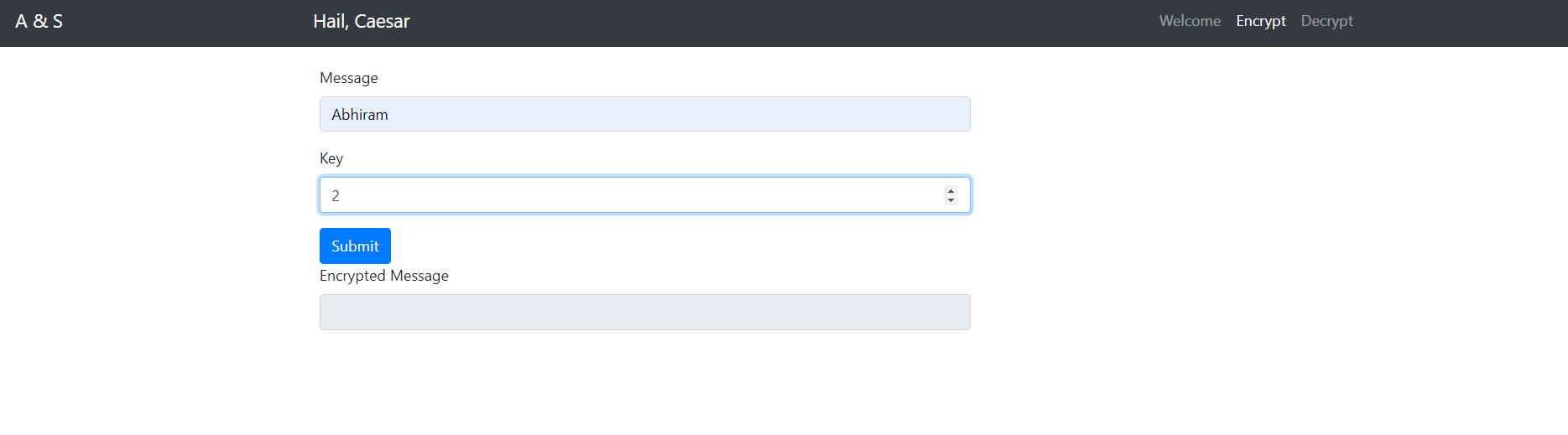
App.py

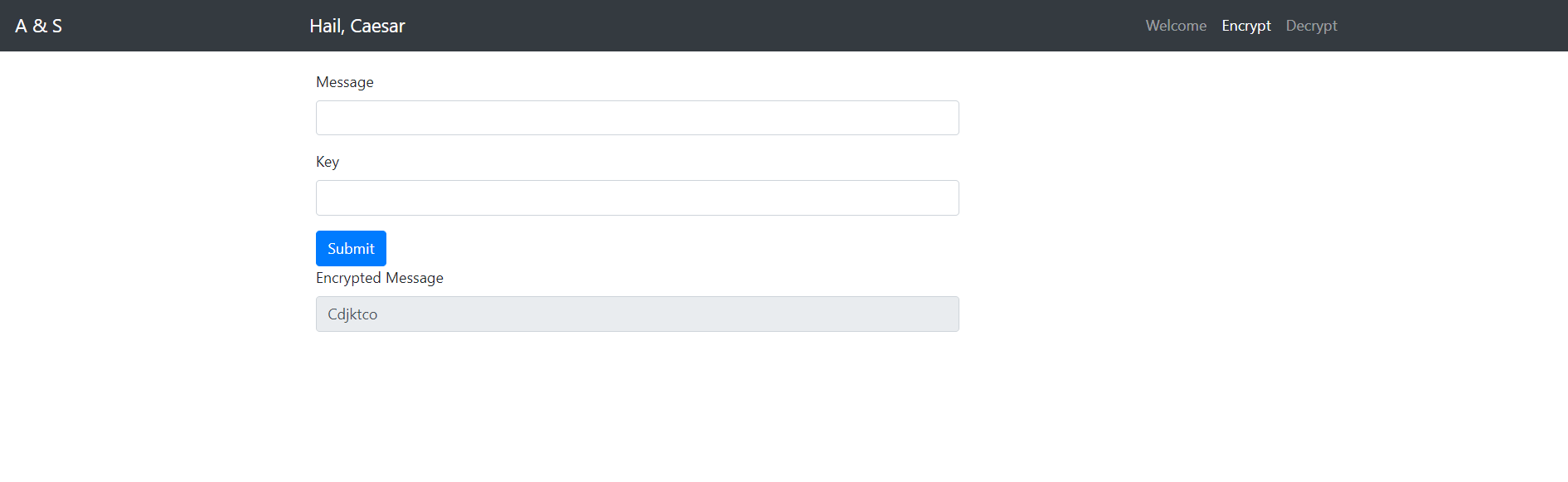
*"""  
This is our web application! It's built using a 'framework' (a bunch of tools) called Flask http://flask.pocoo.org/  
"""*# import some modules from flask  
from flask import Flask, render\_template, request, url\_for  
# import some 2 functions from helpers.py  
from helpers import sanitise\_text\_input, sanitise\_number\_input  
  
# Instantiate (create) our flask app object  
app = Flask(\_\_name\_\_)  
# debug mode so our browser doesn't cache (save) our page  
app.debug = True  
  
  
# our index (main) page  
@app.route('/')  
def index():  
 return render\_template('index.html')  
  
  
# encyption page  
@app.route('/encrypt', methods=['GET', 'POST'])  
def encrypt():  
 # if the user is requesting the page via GET request, just return our webpage in html  
 if request.method == 'GET':  
 return render\_template('encrypt.html')  
 # if the user POSTs us some data, let's encrypt it and return the encrypted message  
 elif request.method == 'POST':  
 message = sanitise\_text\_input(request.form["message"])  
 key = sanitise\_number\_input(request.form['key'])  
 encryptedmessage = encrypt\_caesar(message, key)  
 return render\_template('encrypt.html', encrypted\_message=encryptedmessage)  
  
  
# decryption page  
@app.route('/decrypt', methods=['GET', 'POST'])  
def decrypt():  
 if request.method == 'GET':  
 return render\_template('decrypt.html')  
 elif request.method == 'POST':  
 message = sanitise\_text\_input(request.form["message"])  
 key = sanitise\_number\_input(request.form['key'])  
 decryptedmessage = decrypt\_caesar(message, key)  
 return render\_template('decrypt.html', decrypted\_message=decryptedmessage)  
  
  
# function to encrypt users message using key provided  
def encrypt\_caesar(message, key):  
 encryption\_str = ''  
 for i in message:  
 if i.isupper():  
 temp = 65 + ((ord(i) - 65 + key) % 26)  
 encryption\_str = encryption\_str + chr(temp)  
 elif i.islower():  
 temp = 97 + ((ord(i) - 97 + key) % 26)  
 encryption\_str = encryption\_str + chr(temp)  
 else:  
 encryption\_str = encryption\_str + i  
 # *TODO* return encryption\_str  
  
  
def decrypt\_caesar(message, key):  
 string2 = ''  
 for k in range(len(message)):  
 poz = ord(message[k])  
 if message[k].islower():  
 if poz + key < 97:  
 dif = (poz - key) - 97  
 if dif < 0:  
 string2 += chr(122 + dif + 1)  
 else:  
 string2 += chr(poz - key)  
 elif message[k].isupper():  
 if poz - key < 65:  
 dif = (poz - key) - 65  
 if dif < 0:  
 string2 += chr(90 + dif + 1)  
 else:  
 string2 += chr(poz - key)  
 else:  
 string2 += message[k]  
 return string2

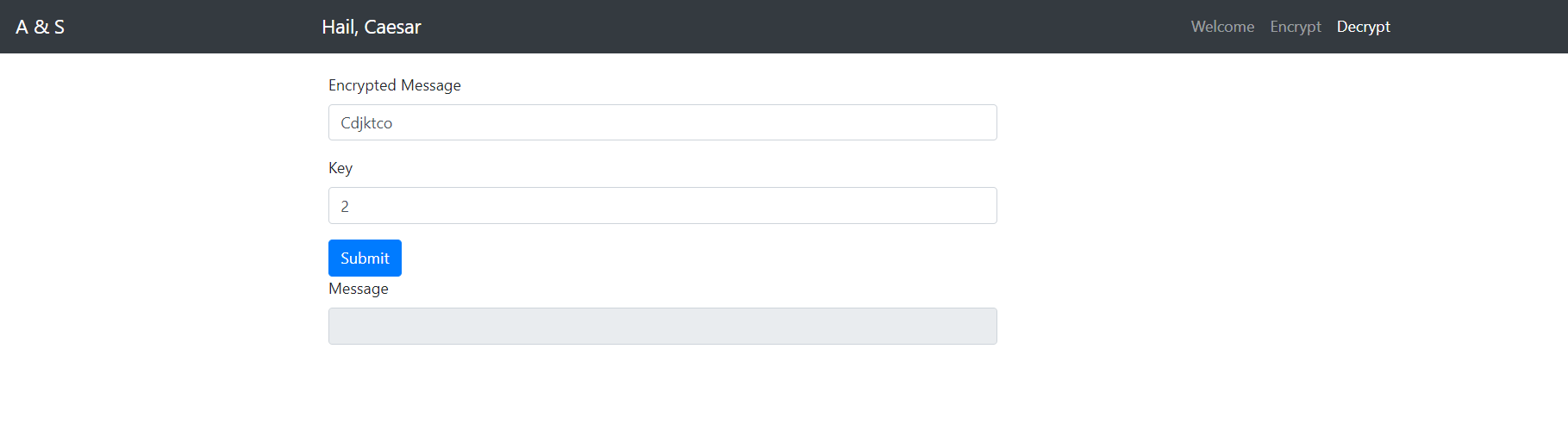
**RESULT AND DISCUSSION**

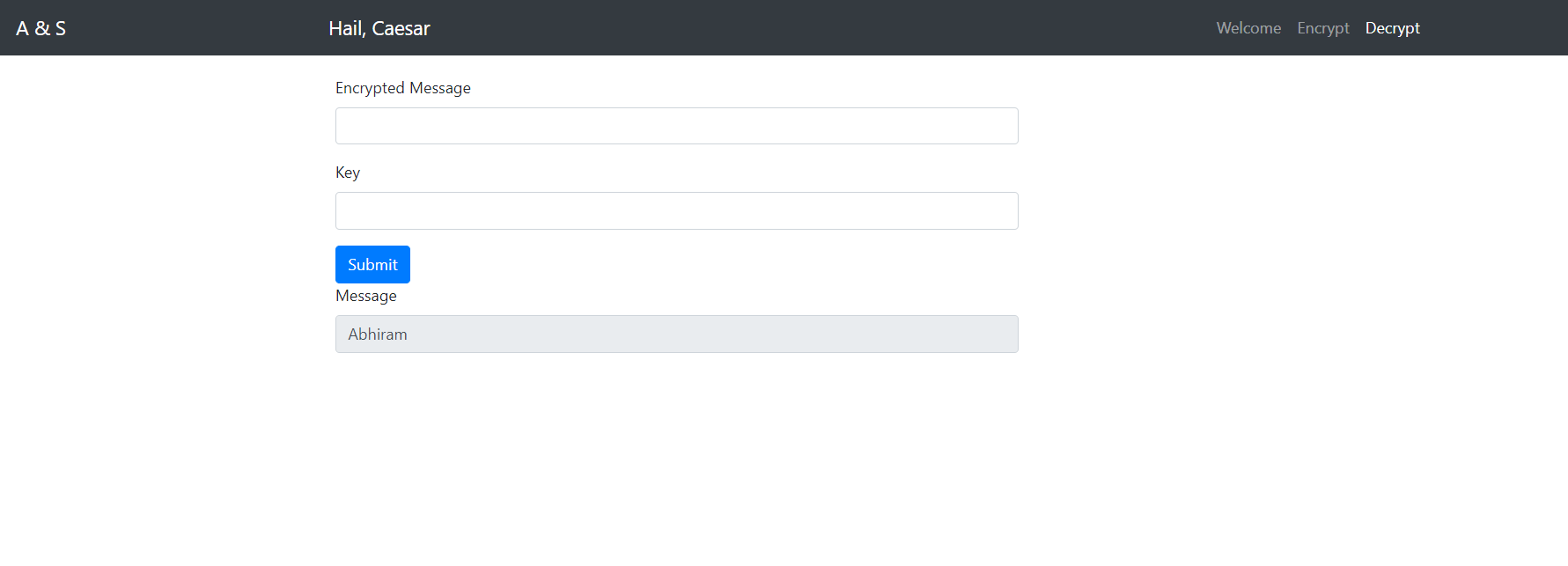
**OUTPUT**

****

****

****

****

****

**CONCLUSION AND FUTURE WORK**

With the usage of Caesar Cipher algorithm, we have seen how a plain text can be converted into Cipher text/encrypted text, which helps in ensuring data security while transferring it from one source to the other.

This method is useful in many ways:

* It is very easy to implement.
* This method is the simplest method of cryptography.
* It requires only a few computing resources.
* With this, we can get the ASCII values and the binary representation of the given text.

We can further develop this project by giving an option to the users in such a way that, even they can get the encrypted and decrypted text of special characters too.