Configuration Manual

MSc Internship
Cybersecurity

Thamarai Kannan S.V
Student ID: x18105114

School of Computing
National College of Ireland

Supervisor: Imran khan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Thamarai Kannan Sabapathy Venkatachalapathy		
Student ID:	X18105114		
Programme:	MSc in Cyber Security	**Year:**	2019-2020
Module:	Academic Internship		
Lecturer:	Imran Khan		
Submission Due Date:	12 December 2019		
Project Title:	Improvising Jumbling Salting algorithm using even or odd technique		
Word Count: 611	**Page Count: 8**		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ………

Date: ………

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Thamarai Kannan S.v
X18105114

i

1 Introduction

The configuration manual explains how the system is set up to achieve result of the thesis “Securing the passwords using Jumbling-Salting Algorithm from cyber-attacks” in relation to hardware and software tools used with an explanation why they are chosen to create a website.

2 System Specification:

Hardware:

The hardware device specifications used to create and deploy the websites are recorded as follow.

Processor: Intel® Core™ i3-7100U CPU @ 2.40GHz, 2401 Mhz, 2 Core(s)

Memory: 12288MB RAM

GPU: 6230MB

Hard disk size: 1 TB storage

Software:

Operating System: Microsoft Windows 10

Software Tool

Microsoft Visual Studio 2019: It’s an Integrated development environment used to write code to develop website.

Programming Language:

.NET:

The .NET framework is used as front-end development for website.

.C Sharp:

.C sharp is used for back-end development for website.
Microsoft Word 2019 is used for writing final thesis draft.

3 Execution process:

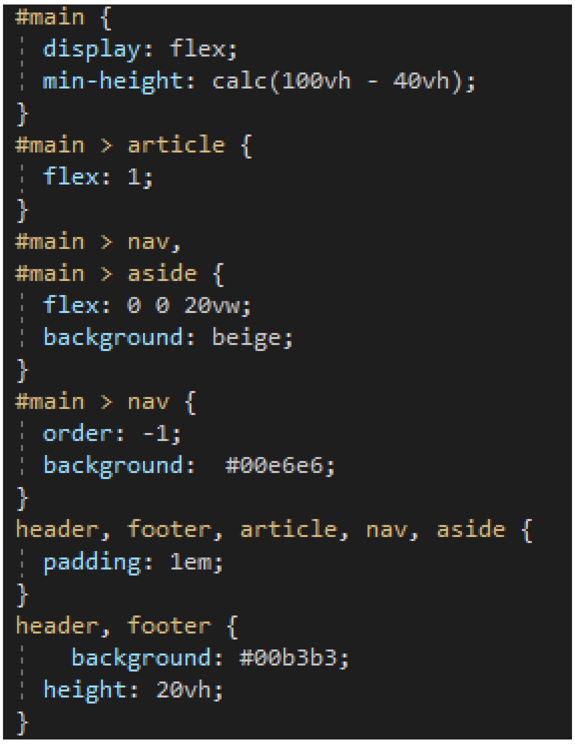
The undertaking website is executed utilizing different programming instruments and programming languages.

Step 1: Installation of Visual Studio

First step in execution process is installing visual studio in system.

Step 2: Creating home page for website

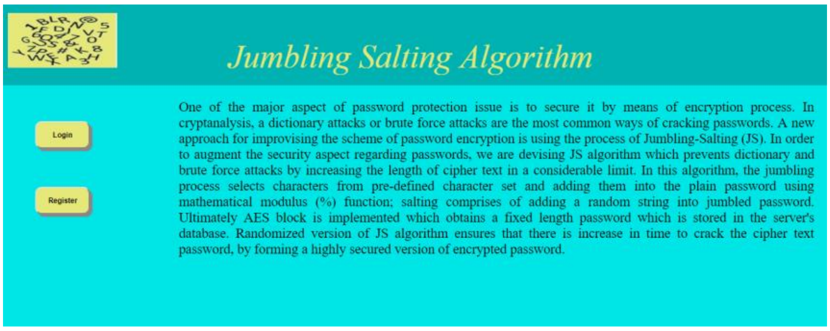
Code used:



```
#main {
  display: flex;
  min-height: calc(100vh - 40vh);
}
#main > article {
  flex: 1;
}
#main > nav,
#main > aside {
  flex: 0 0 20vw;
  background: beige;
}
#main > nav {
  order: -1;
  background:  #00e6e6;
}
header, footer, article, nav, aside {
  padding: 1em;
}
header, footer {
    background: #00b3b3;
  height: 20vh;
}
```

Figure 1: Code for Homepage

Output:



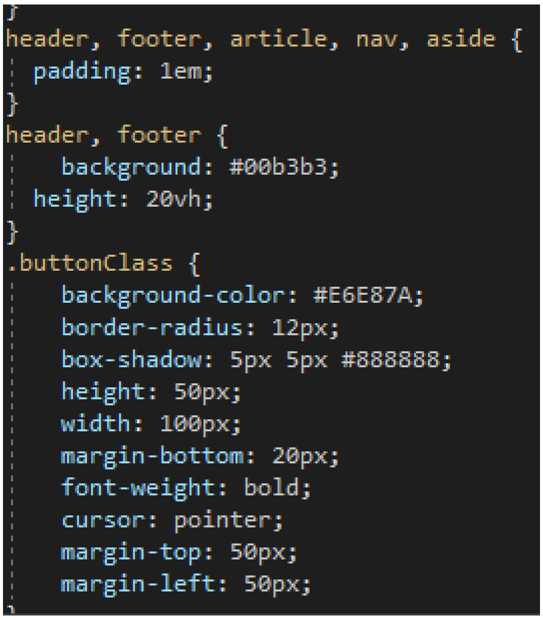
Jumbling Salting Algorithm

One of the major aspect of password protection issue is to secure it by means of encryption process. In cryptanalysis, a dictionary attacks or brute force attacks are the most common ways of cracking passwords. A new approach for improvising the scheme of password encryption is using the process of Jumbling-Salting (JS). In order to augment the security aspect regarding passwords, we are devising JS algorithm which prevents dictionary and brute force attacks by increasing the length of cipher text in a considerable limit. In this algorithm, the jumbling process selects characters from pre-defined character set and adding them into the plain password using mathematical modulus (%) function; salting comprises of adding a random string into jumbled password. Ultimately AES block is implemented which obtains a fixed length password which is stored in the server's database. Randomized version of JS algorithm ensures that there is increase in time to crack the cipher text password, by forming a highly secured version of encrypted password.

Figure 2: Homepage in localhost

Step 3: Creating Registration page in website for new users to register.

Code used:



```
}
header, footer, article, nav, aside {
  padding: 1em;
}
header, footer {
    background: #00b3b3;
  height: 20vh;
}
.buttonClass {
    background-color: #E6E87A;
    border-radius: 12px;
    box-shadow: 5px 5px #888888;
    height: 50px;
    width: 100px;
    margin-bottom: 20px;
    font-weight: bold;
    cursor: pointer;
    margin-top: 50px;
    margin-left: 50px;
}
```

Figure 3: Registration code

Output:

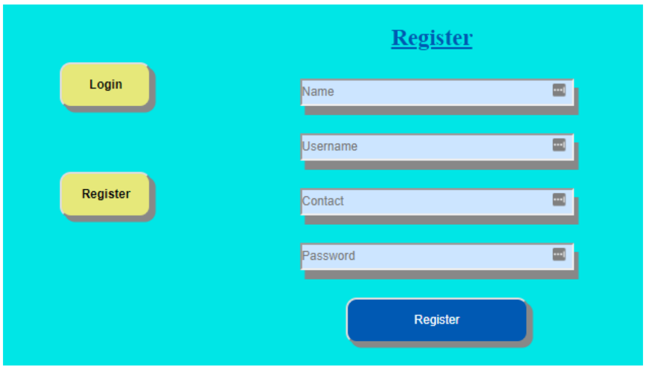
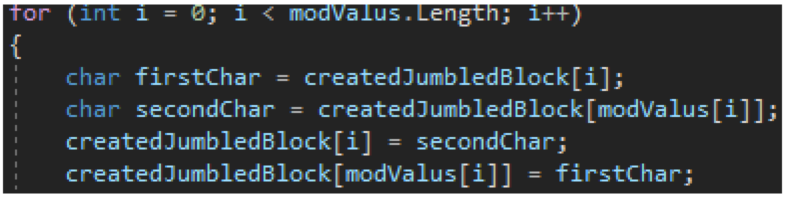


Figure 4: Registration Page

Step 4: Shuffling of plaintext password

Code used:

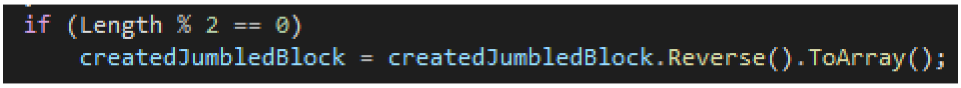


```
for (int i = 0; i < modValus.Length; i++)
{
    char firstChar = createdJumbledBlock[i];
    char secondChar = createdJumbledBlock[modValus[i]];
    createdJumbledBlock[i] = secondChar;
    createdJumbledBlock[modValus[i]] = firstChar;
```

Figure 5: Shuffling

Step 5: Check the length of given plain text is even or odd

Code used:

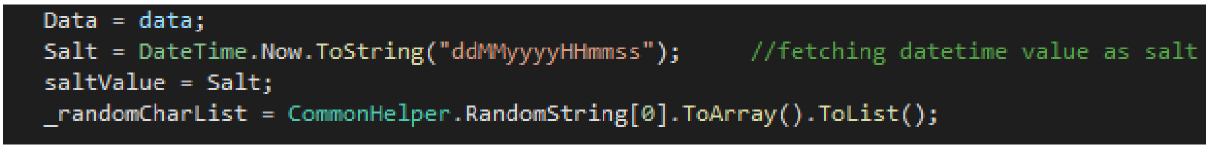


```
if (Length % 2 == 0)
    createdJumbledBlock = createdJumbledBlock.Reverse().ToArray();
```

Figure 6: Checking length whether even or odd

Step 6: Addition of salt value

Code used:

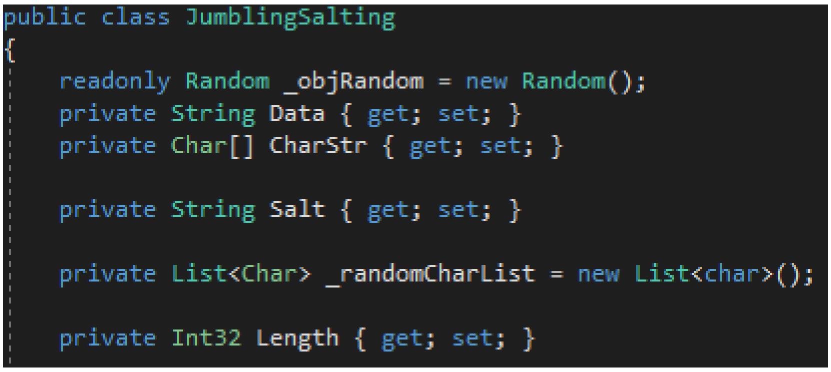


```
Data = data;
Salt = DateTime.Now.ToString("ddMMyyyyHHmmss");     //fetching datetime value as salt
saltValue = Salt;
_randomCharList = CommonHelper.RandomString[0].ToArray().ToList();
```

Figure 7: Salt process

Step 7: Initializing jumbling block from random values generated from pre-defined set.

Code used:

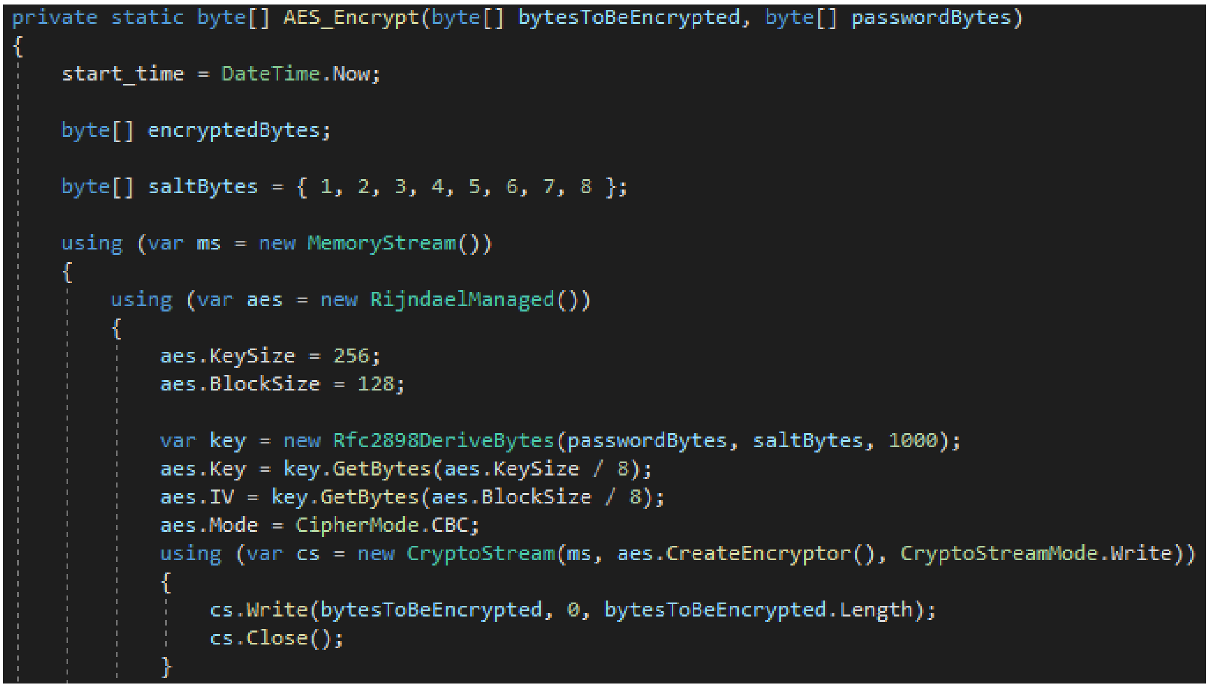


```
public class JumblingSalting
{
    readonly Random _objRandom = new Random();
    private String Data { get; set; }
    private Char[] CharStr { get; set; }

    private String Salt { get; set; }

    private List<Char> _randomCharList = new List<char>();

    private Int32 Length { get; set; }
```

Figure 8: Random value for Jumbling process

Step 8: Encrypting final block with AES algorithm.

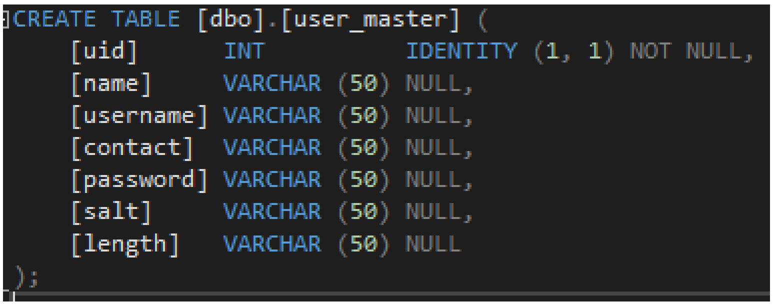
Code used:



```
private static byte[] AES_Encrypt(byte[] bytesToBeEncrypted, byte[] passwordBytes)
{
    start_time = DateTime.Now;

    byte[] encryptedBytes;

    byte[] saltBytes = { 1, 2, 3, 4, 5, 6, 7, 8 };

    using (var ms = new MemoryStream())
    {
        using (var aes = new RijndaelManaged())
        {
            aes.KeySize = 256;
            aes.BlockSize = 128;

            var key = new Rfc2898DeriveBytes(passwordBytes, saltBytes, 1000);
            aes.Key = key.GetBytes(aes.KeySize / 8);
            aes.IV = key.GetBytes(aes.BlockSize / 8);
            aes.Mode = CipherMode.CBC;
            using (var cs = new CryptoStream(ms, aes.CreateEncryptor(), CryptoStreamMode.Write))
            {
                cs.Write(bytesToBeEncrypted, 0, bytesToBeEncrypted.Length);
                cs.Close();
            }
```

Figure 9: AES encryption

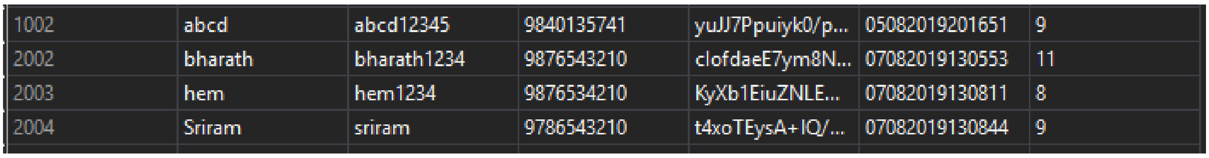
Step 9: Creating table in database to store information in SQL server.



```
CREATE TABLE [dbo].[user_master] (
    [uid]      INT          IDENTITY (1, 1) NOT NULL,
    [name]     VARCHAR (50) NULL,
    [username] VARCHAR (50) NULL,
    [contact]  VARCHAR (50) NULL,
    [password] VARCHAR (50) NULL,
    [salt]     VARCHAR (50) NULL,
    [length]   VARCHAR (50) NULL
);
```

Figure 10: Creation of Table

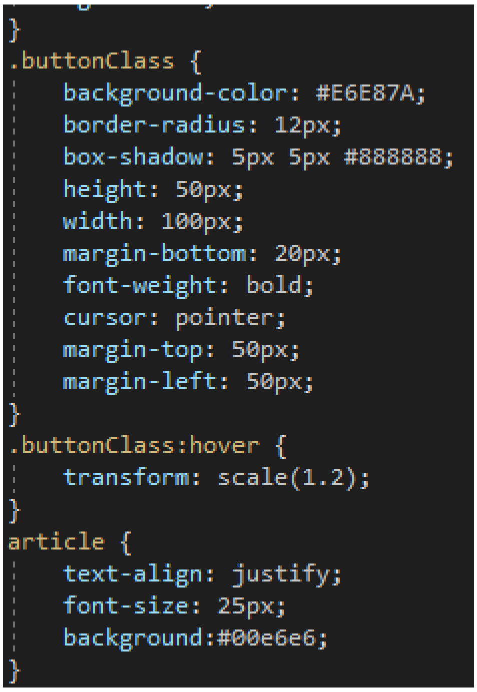
Output:



1002	abcd	abcd12345	9840135741	yuJJ7Ppuiyk0/p...	05082019201651	9
2002	bharath	bharath1234	9876543210	clofdaeE7ym8N...	07082019130553	11
2003	hem	hem1234	9876534210	KyXb1EiuZNLE...	07082019130811	8
2004	Sriram	sriram	9786543210	t4xoTEysA+lQ/...	07082019130844	9

Step 10: Creation of Login page.

Code used:



```
}
.buttonClass {
    background-color: #E6E87A;
    border-radius: 12px;
    box-shadow: 5px 5px #888888;
    height: 50px;
    width: 100px;
    margin-bottom: 20px;
    font-weight: bold;
    cursor: pointer;
    margin-top: 50px;
    margin-left: 50px;
}
.buttonClass:hover {
    transform: scale(1.2);
}
article {
    text-align: justify;
    font-size: 25px;
    background:#00e6e6;
}
```

Figure 11: Login page

Output:

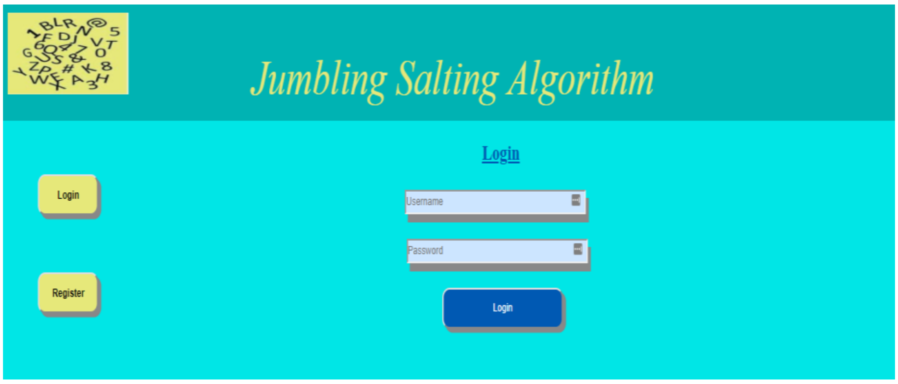


Figure 12: Login Page