National College of Ireland

A comparative study of AES encryption modes and hashing for blockchain applications - Configuration Manual

MSc Research Project
Cloud Computing

Debashish Sarangi
Student ID: 22162844

School of Computing
National College of Ireland

Supervisor: Dr Giovani Estrada

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Debashish Sarangi
Student ID:	22162844
Programme:	Cloud Computing
Year:	2023
Module:	MSc Research Project
Supervisor:	Dr Giovani Estrada
Submission Due Date:	25/04/2024
Project Title:	A comparative study of AES encryption modes and hashing for blockchain applications - Configuration Manual
Word Count:	402
Page Count:	9

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:	27th May 2024

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):	

A comparative study of AES encryption modes and hashing for blockchain applications - Configuration Manual

Debashish Sarangi
22162844

1 Introduction

The configuration guideline's goal is to provide a concise overview of the requirements required to build this application. It would provide direction for the exacting procedures needed to properly design, implement, test, and repeat the project. The following sections make up the remaining portions of this document. Module 2 displays the system's settings; Module 3 lists the required libraries; Module 4 contains database tables; and Module 5 implements AES encryption, hashing, and blockchain.

2 System Configuration

2.1 Configuring the ASP.Net Environment

The entire programme was written in the C# programming language. The code for the project was written using the Visual Studio platform. You can get Visual Studio off the internet and set it up for free. It was selected because it is a freeware package that allows you to build programmes in many languages and supports a wide range of platforms. The suggested text editor is Visual Studio 20221 version 19.0.

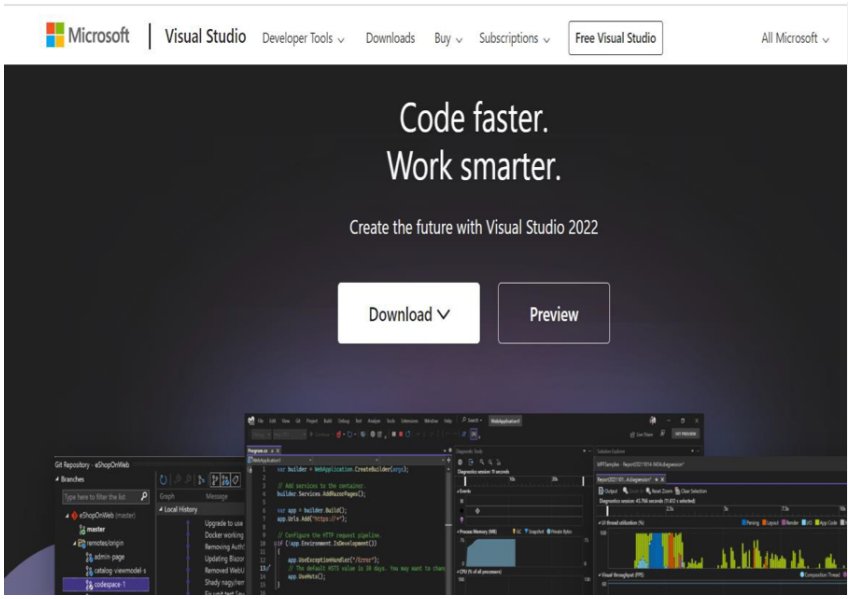


Figure 1: Download page of Visual Studio

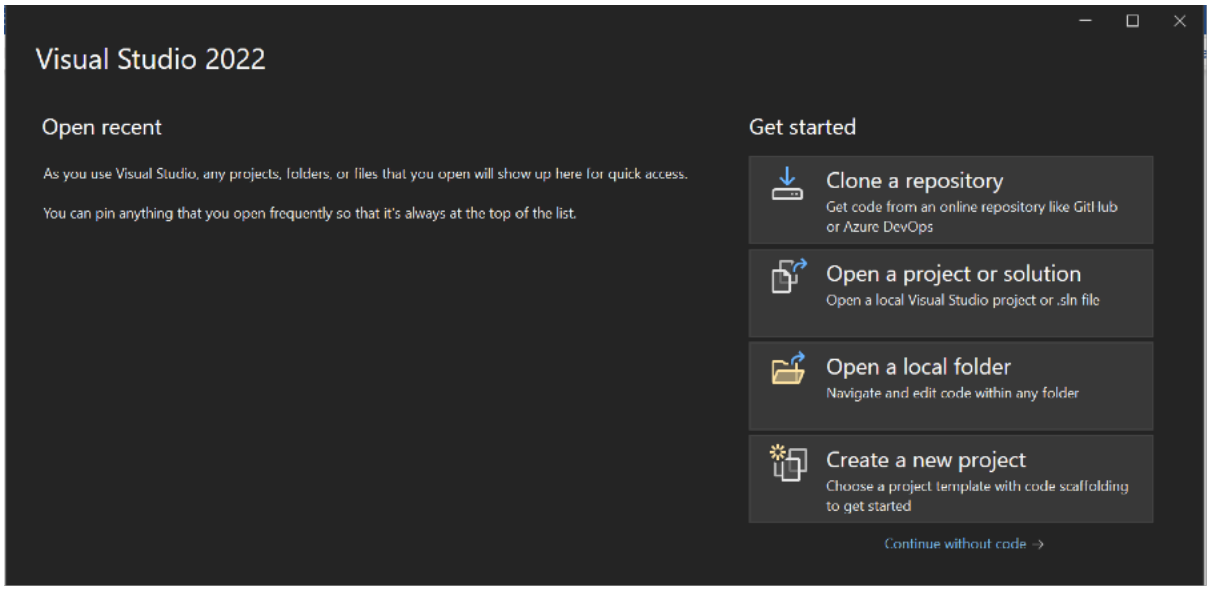


Figure 2: Welcome page of Visual Studio

2.2 Configuring a Database Server

A Microsoft SQL database is used by the project to store application data. We could connect our application to local server with this technology. This instance of SQL Server Management Studio is version 20.0.70.0. To connect to the database, the user must input the credentials. You can get free online versions of the SQL Management tool that are open source.

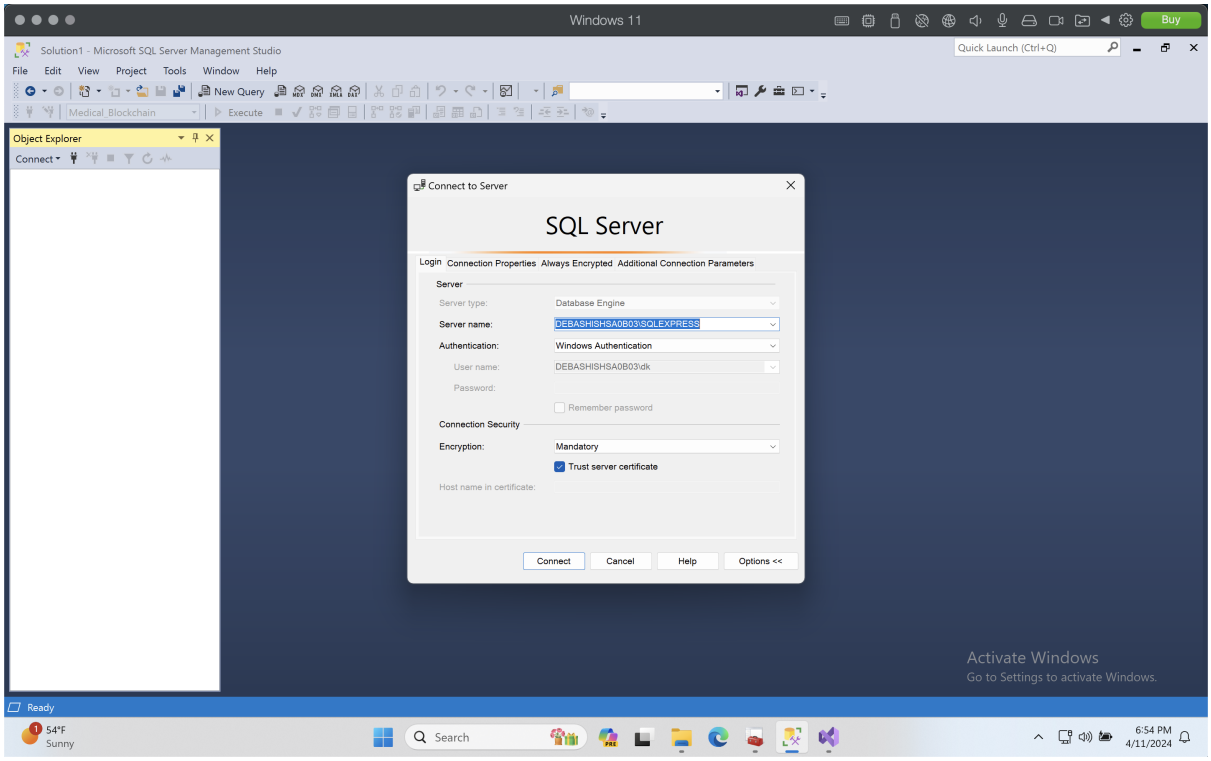


Figure 3: connecting to SQL server management system

```
version="1.0"?

more information on how to configure your ASP.NET application, please visit
://go.microsoft.com/fwlink/?LinkId=169433

guration>
Settings>
dd key="ChartImageHandler" value="storage=file;timeout=20;dir=c:\TempImageFiles\;"/>
pSettings>
nectionStrings>
--<add name="connect" connectionString="Server=contextbasedsearch.database.windows.net,1433;Initial Catalog=contextbasedsearch;Persist Security Info=False;User ID=
--<add name="dbConnection" connectionString="Server=tcp:medicalblockchain.database.windows.net,1433;Initial Catalog=medicalblockchain;Persist Security Info=False;U
dd name="dbConnection" connectionString="Data Source=DEBASHISHSA0B03\SQLEXPRESS;Initial Catalog=Medical_Blockchain;Integrated Security=True"/>
nnectionStrings>

r a description of web.config changes see http://go.microsoft.com/fwlink/?LinkId=235367.

e following attributes can be set on the <httpRuntime> tag.
<system.Web>
  <httpRuntime targetFramework="4.8.1" />
</system.Web>

tem.web>
essionState timeout="50000"/>
ustomErrors mode="Off"/>
ttpHandlers>
<add path="ChartImg.axd" verb="GET,HEAD,POST" type="System.Web.UI.DataVisualization.Charting.ChartHttpHandler, System.Web.DataVisualization, Version=4.0.0.0, Cultu
httpHandlers>
ages controlRenderingCompatibilityVersion="4.0">
<controls>
  <add tagPrefix="asp" namespace="System.Web.UI.DataVisualization.Charting" assembly="System.Web.DataVisualization, Version=4.0.0.0, Culture=neutral, PublicKeyToke
</controls>
pages>
ompilation debug="true" targetFramework="4.8.1">
<assemblies>
  <add assembly="system.web.datavisualization, version=4.0.0.0, culture=neutral, publickeytoken=31bf3856ad364e35"/>
  <!--<add assembly="microsoft.office.interop.word, version=15.0.0.0, culture=neutral, publickeytoken=71e9bce111e9429c"/>-->
</assemblies>
compilation>
```

Figure 4: Database connection in the web.Config file

2.3 Hardware specifications

- RAM: 8 GibaByte

- Processor: Apple M1
- Storage: 11 GigaByte

2.4 Software Used

- OS Used: Microsoft Windows 11
- Language: C# .Net
- IDE used: Visual Studio 2022

[1]

3 Libraries Used

The following is a list of the primary libraries and import lines used to develop the application.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.IO;
using System.Security.Cryptography;
using System.Text;
using System.Net;
```

Figure 5: Libraries Included

[1] https://visualstudio.microsoft.com/

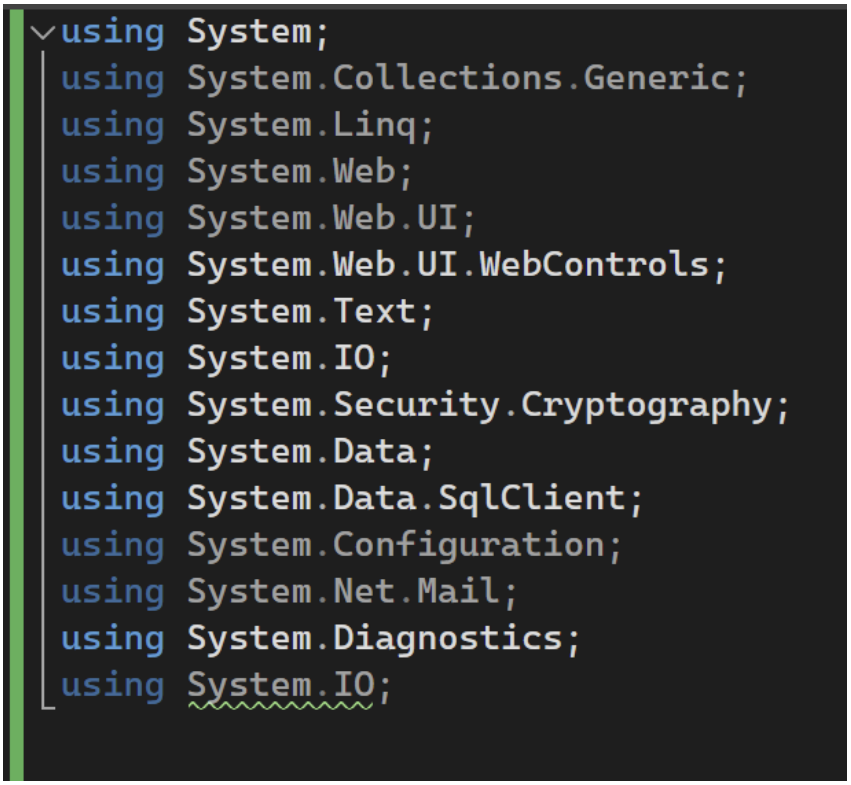


Figure 6: Libraries Included

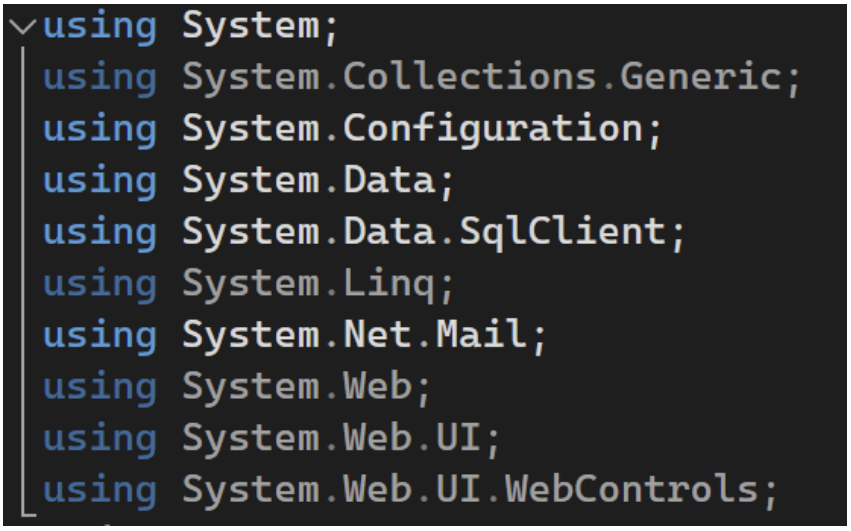


Figure 7: Libraries Included

4 Database Tables Used

These are the database table used in the project.

	Id	Date	File_Name	Patient_Id	Patient_Name	Description	Hash	doc_id	Key_Value	hash_match
1	1	2024-04-08	med_preis.jpg	A1	Rahul jain	fever	sF3TaqMRXSgDBRAtVe5gQx93EzZgMbg63Qq3tAVGH4M=	1	ATpCvYZG	ee9d8c0264a08e39382670f19cc220440e9ea9cc8067a4930...
2	2	2024-04-10	aap.jpg	A1	Rahul jain	aap	H9wPFRt0xMCppgqyemGQliU3qD7makYf3t/lS/xspz8=	1	2d2Wlqn6	50698041779feac71597cd3ef05e2c21eb42ef196379baba88...
3	3	2024-04-10	3.png	A1	Rahul jain	3	SFe0djuLx8W0L+T/kb2dFGNug3TRJ9laJVPhibpZVdA=	1	gv4AzXCL	37b215b4dd82db94ef16bf8550f9dc3311ae7ace9ba2fbfe2e6...
4	4	2024-04-11	code.png	A1	Rahul jain	test	lVSpsDtafJw5CCzvBE/x7Gfmxc+KmOgmSdmldJ3hdJw=	1	vQGYaxXe	909d6e5cfd077ca20400e73d58b9481f8b41ed620c2542cbb...
5	5	2024-04-11	n.png	A1	Rahul jain	test 1	E3ERwfhkgdVJFaDKNd+7fqcQEiv3ls8lM6sNkuVM6Go=	1	AYreWEwW	c29ac18e85e5e31433d172b1022ccb5ae77c66d0ae295a60...
6	6	2024-04-11	Picture 3.jpg	A1	Rahul jain	new	/c26I+Q5HVjihBJV7MAz9gKCkosFbwo+Q2CzF+Fo7HA=	1	KyChGQod	be948ee58619a81dd0d06b802a56d699509ab75d171547b4...
7	7	2024-04-11	8.png	A1	Rahul jain	test	2UO4+VR1EsMHaqng8UMnsMPK8BqU7uyeXnLw+thSa+E=	1	xQLJkQlx	01633945d6578f438ef146d10a175bf36bce1e2fc0dbdc84580...
8	8	2024-04-11	2.png	A1	Rahul jain	testing	xBmXF5AysB9vRlBuZcxOvW15Wf7tw5teA5dyBuMMJs0=	1	ozr1xbag	afe07195b0f0f15f1f30341457a1777292ed3ad42d561a63b4b4...
9	9	2024-04-11	Three-types-of-cryptography-Symmetric-key-public...	A1	Rahul jain	try hard	VLef3ZFHUL/DBzUoFgxLLM32JKIZUDj9v+jBba/WtHo=	1	uR2m8Qs3	5831aacd81e3202841dba99e1bae412e9827295a6a4ffa56d9...
10	10	2024-04-11	Symmetric-Key-Encryption.png	A1	Rahul jain	new	RvedMkKvMmFRZaGbz3Meb6q+25yKffkZ6SA+naJfYdI=	1	qXRVjKLj	36ad2ed157a9cd85d6610cf2df17bc82fc290cb749107c5a8fa...
11	11	2024-04-11	sys.png	A1	Rahul jain	sys	Jm4gR/I4O/Xo9WeqzYcOMEXpLaqw2m9oeCbGs3lee8M=	1	FMCcsb18	36ad2ed157a9cd85d6610cf2df17bc82fc290cb749107c5a8fa...
12	12	2024-04-11	re volt.png	A1	Rahul jain	normal	F4gh2vdLOaFGlVmkxkshBEs9oid7215v6g1FE684y14=	1	Sq2fkEL7	ba132604515d43bc240fb1b5ca0c6170c594f38a729c272ca8...
13	13	2024-04-11	Signature.png	A1	Rahul jain	sign	km/zmPkys2y8NXdexYUpMVvbBBLtphx8snSDkZB8b74=	1	1RE6ARzT	622459404a040eafd3f9051d6e27502ccf23316234e31a481b2...
14	14	2024-04-11	google.png	A1	Rahul jain	sig	dahUFH3CXfznROD7oPwl1MLiRfALGFE4E6Wl0nEpE=	1	lHsDpq13	f56308874d6ae836a95a1301e16681cae57b1dd7a84d0ec41...
15	15	2024-04-13	Bard_Chart_Image.png	A1	Rahul jain	gg	kNc8MMf2jAt/oH8c2h6uS5HJwr5prR6x87ODniZv3Tw=	1	xkq9ciTb	a96e5f68a46867e6d2bec4eeba42ac12afed231ffd7469b148...
16	16	2024-04-13	Bard_Chart_Image-2.png	A1	Rahul jain	gg	6ogmx142WYzTKBe03dC5/Ckt9o/VAGWBZFbPHofmDzE=	1	TZ5p6FMv	f991437939339050 1bcaf672ca2260cf556ec5349797e0d47e1...
17	17	2024-04-13	129589429_1702237761613_page-0001.jpg	A1	Rahul jain	h	ckcQtqJvEjpV29aywYaAWITbXxByO99VpxoTjHww7E8=	1	eWEAzJSp	eb78cf0693f14ef1c383ea887d99e0b929e4f984447b19d9fd5...

Figure 8: Database for File Uploaded

	Id	Full_Name	Degree	Specialist	Email	Phone	Password	Address	Hash
1	1	Debashish Sarangi	PHD	medicine	debashishsarangi0@gmail.com	8822307638	EB5lmDpa	dublin	NULL

Figure 9: Database for Doctor

	User_id	Patient_ID	Doc_Id	Full_Name	Password	Contact_no	Email_id	Address	Description	Remark	Date	Gen_Hash
1	1	A1	1	Rahul jain	Jxlcl0uq	7008498462	dkinireland@gmail.com	dublin	Fever	###	2024-04-08	9NIEXWECJmq6w0Odu/mGshmetvKNJD0KRc2G3inOigY=

Figure 10: Database for Patients

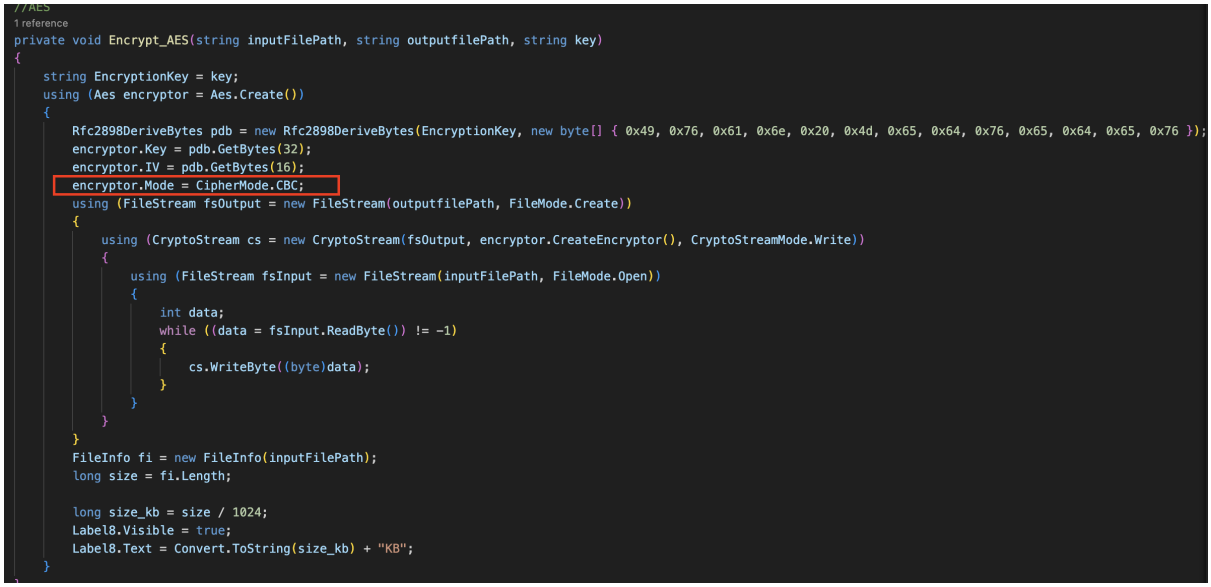
	id	name	email	contact	address	password
1	1	bupa corp	debashishsarangi2@gmail.com	7008498468	dublin	6110786

Figure 11: Database for Clients

5 Implementation of AES encryption and Hashing

The key is obtained using PBKDF2 and the SHA-256 hashing technique, and the data is secured using AES-256 encryption. Additionally, the user receives the key by mail via SMTP.

To switch between the available AES modes in practice, you would change this line to set the preferred one.



```
//AES
1 reference
private void Encrypt_AES(string inputFilePath, string outputfilePath, string key)
{
    string EncryptionKey = key;
    using (Aes encryptor = Aes.Create())
    {
        Rfc2898DeriveBytes pdb = new Rfc2898DeriveBytes(EncryptionKey, new byte[] { 0x49, 0x76, 0x61, 0x6e, 0x20, 0x4d, 0x65, 0x64, 0x76, 0x65, 0x64, 0x65, 0x76 });
        encryptor.Key = pdb.GetBytes(32);
        encryptor.IV = pdb.GetBytes(16);
        encryptor.Mode = CipherMode.CBC;
        using (FileStream fsOutput = new FileStream(outputfilePath, FileMode.Create))
        {
            using (CryptoStream cs = new CryptoStream(fsOutput, encryptor.CreateEncryptor(), CryptoStreamMode.Write))
            {
                using (FileStream fsInput = new FileStream(inputFilePath, FileMode.Open))
                {
                    int data;
                    while ((data = fsInput.ReadByte()) != -1)
                    {
                        cs.WriteByte((byte)data);
                    }
                }
            }
        }
        FileInfo fi = new FileInfo(inputFilePath);
        long size = fi.Length;

        long size_kb = size / 1024;
        Label8.Visible = true;
        Label8.Text = Convert.ToString(size_kb) + "KB";
    }
}
```

Figure 12: Changing diffrent AES modes

For example, if you want to move to another mode such as Electronic Codebook (ECB) mode, you would alter the mode as follows: **encryptor.Mode = CipherMode.ECB;** Similarly, if you want to move to another mode such as Counter (CTR) mode, you would alter the mode as follows: **encryptor.Mode = CipherMode.CTR;**

```
//AES
1 reference
private void Encrypt_AES(string inputFilePath, string outputfilePath, string key)
{
    string EncryptionKey = key;
    using (Aes encryptor = Aes.Create())
    {
        Rfc2898DeriveBytes pdb = new Rfc2898DeriveBytes(EncryptionKey, new byte[] { 0x49, 0x76, 0x61, 0x6e, 0x20, 0x4d, 0x65, 0x64, 0x76, 0x65, 0x64, 0x65, 0x76 });
        encryptor.Key = pdb.GetBytes(32);
        encryptor.IV = pdb.GetBytes(16);
        using (FileStream fsOutput = new FileStream(outputfilePath, FileMode.Create))
        {
            using (CryptoStream cs = new CryptoStream(fsOutput, encryptor.CreateEncryptor(), CryptoStreamMode.Write))
            {
                using (FileStream fsInput = new FileStream(inputFilePath, FileMode.Open))
                {
                    int data;
                    while ((data = fsInput.ReadByte()) != -1)
                    {
                        cs.WriteByte((byte)data);
                    }
                }
            }
        }
    }
}
```

Figure 13: AES encryption and PBKDF2 key derivation

```
private string performHash(string data)
{
    byte[] bytes = Encoding.Unicode.GetBytes(data);
    SHA256Managed hashstring = new SHA256Managed();
    byte[] hash = hashstring.ComputeHash(bytes);
    string hashString = string.Empty;
    foreach (byte x in hash)
    {
        hashString += String.Format("{0:x2}", x);
    }
    return hashString;
}
```

Figure 14: SHA-256 Hashing method

```
SmtpClient SmtpServer = new SmtpClient();
MailMessage mail = new MailMessage();
SmtpServer.Credentials = new System.Net.NetworkCredential("debashishsarangi0@gmail.com", "ycuoiqqagkmvfpgz");
SmtpServer.Port = 587;
SmtpServer.EnableSsl = true;
SmtpServer.Host = "smtp.gmail.com";
mail = new MailMessage();
mail.From = new MailAddress("debashishsarangi0@gmail.com");

mail.To.Add(email);
mail.Subject = "Authentication  Details";
mail.Body = "File has been shared to your account to acces that file your Key is "+ key;
SmtpServer.Send(mail);
```

Figure 15: SMTP Method

```
protected void btnSave_Click(object sender, EventArgs e)
{

    Blockchain phillyCoin = new Blockchain();
    //phillyCoin.AddGenesisBlock();

    //string hash=phillyCoin.AddBlock(new Block(DateTime.Now,null ,"5"));  //use this for upload
    string prev_hash = "";
    string file_id = "";
    string count = "";
    if (FileUpload1.HasFile)
    {
        using (SqlCommand cmd3 = new SqlCommand("Select top 1 hash from log_master where patient_id=@id order by id desc", cn))
        {
            cmd3.Parameters.AddWithValue("@id", DropDownList1.SelectedItem.Text);
            using (SqlDataAdapter adp3 = new SqlDataAdapter(cmd3))
            {
                DataTable dt3 = new DataTable();
                adp3.Fill(dt3);
                {
                    prev_hash = dt3.Rows[0]["hash"].ToString();
                    B_Count.prev_hash = prev_hash;
                }
            }

        }
```

Figure 16: Block chain implementation

References