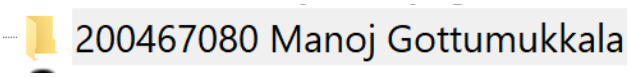
Information Encoding Standards

Assignment: 2

1.Create a folder in the BDAT1001-<CRN> FTP folder using the following information and format:



Created the folder with the given naming conventions.

3.Convert you myimage.jpg to Base64 encoding using this [website] application

Converted myimage.jpg to Base64 encoding. Here the highlighted region is the Base64 encoding of my image.

4.Use the HTML code from the [Base64 C# Converter] page to build a basic web page to describe yourself

```
<!DOCTYPE html>

<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta charset="utf-8" />
    <title></title>
</head>
<body>
    <div>
        <dl>
            <dt>First Name</dt>
            <dd>Manoj</dd>
            <dt>Last Name</dt>
            <dd>Gottumukkala</dd>
            <dt>Date of Birth</dt>
            <dd>18/10/1997</dd>
            <dt>My picture</dt>
            <dd>
                <img src="
            </dd>
        </dl>

    </div>
</body>
</html>
```

The HTML script for building a website with my first name, last name, date of birth, and the picture is shown above. I have provided the base64 encoding of my image in the HTML script for displaying my picture.

First Name
Manoj
Last Name
Gottumukkala
Date of Birth
18/10/1997
My picture

This is how it looks when the webpage is loading/opened.

5.Add the code from [Base64 C# Converter] to the Converter application from [Assignment 1 - Data Security, Encoding, Encryption, and Privacy].

```
namespace Assignment2
{
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {

            //Get the location of the user's Desktop
            string desktopPath = Environment.GetFolderPath(Environment.SpecialFolder.Desktop);

            //Build a image file path from the original image path
            FileInfo fileinfo = new FileInfo($"{desktopPath}\\myimage.jpg .jpg");

            //Provide an Image from a file on your Desktop
            Image image = Image.FromFile(fileinfo.FullName);

            //Convert Image to Base64 encoded text
            string base64image = Converter.ImageToBase64(image, ImageFormat.Jpeg);

            //Output the Base64 encoded text
            Console.WriteLine(base64image);

            //Build a new image file path from the original image path
            FileInfo newfileinfo = new FileInfo($"{fileinfo.Directory}\\{fileinfo.Name} - New from Base64{fileinfo.Extension}");

            //Save Base64 encoded text to Image object
            Image imageFromBase64 = Converter.Base64ToImage(base64image);

            //Save the Image to a file
            imageFromBase64.Save(newfileinfo.FullName, ImageFormat.Jpeg);

            string[] lines = { "StudentId,FirstName,LastName,DateOfBirth,ImageData", "200467080,Manoj,Gottumukkala,18/10/97, /9j/4AAQSk
```

This is the code I have used in the program.cs

Flow of execution:

1. This is what happens when we execute the first three statements of the program.

desktopPath gets the location of my desktop. Fileinfo will have the absolute path of the image specified in the statement, and the image variable has the myimage.jpg

2.

```
//Convert Image to Base64 encoded text
string base64image = Converter.ImageToBase64(image, ImageFormat.Jpeg);

//Output the Base64 encoded text
Console.WriteLine(base64image);
```

When these statements are given, the sequence of execution moves to the ImageToBase64 class in the convertor.cs file where the image is converted to base64 format. The output of this conversion is displayed. The ImageToBase64 class is explained in detail with the comments in convertor.cs file.

3.

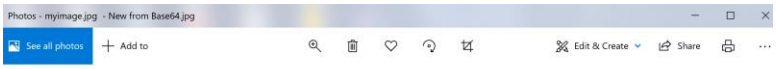
```
//Build a new image file path from the original image path
FileInfo newfileinfo = new FileInfo($"{fileinfo.Directory}\\{fileinfo.Name} - New from Base64{fileinfo.Extension}");

//Save Base64 encoded text to Image object
Image imageFromBase64 = Converter.Base64ToImage(base64image);

//Save the Image to a file
imageFromBase64.Save(newfileinfo.FullName, ImageFormat.Jpeg);
```

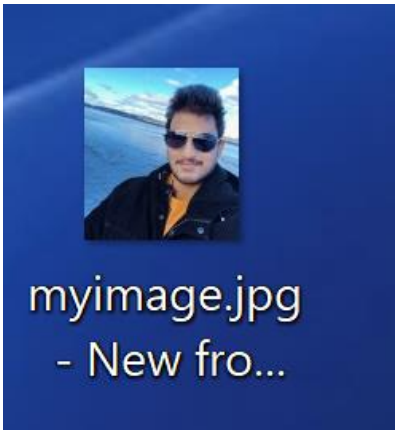
The first statement creates an empty image file. During statement 2, the base64image (This has base64 equivalent of the image) is taken as a parameter by Converter. Base64ToImage. The Base64ToImage module/class in the converter.cs file converts the base64 to image format. During the 3rd statement, the output image is saved.

6.a Output the Base64 encoded image to the console.

It is displaying the base 64 encoded version of the image.

Converted the above base64 encoded version back to image:





We can see an image file is created with a name mentioned in the code.

7. Create a text file named info.csv that contains the following text

```
string[] lines = { "StudentId,FirstName,LastName,DateOfBirth,ImageData", "200467080,Manoj,Gottumukkala,18/10/97, /9j/4AAQSkZJRgABAQEAYABgAAD/4Tr+RXhp

// Set a variable to the Documents path.
string docPath =
  Environment.GetFolderPath(Environment.SpecialFolder.Desktop);


using (StreamWriter outputFile = new StreamWriter(Path.Combine(docPath, "info.csv")))
{
    foreach (string line in lines)
        outputFile.WriteLine(line);
}
```

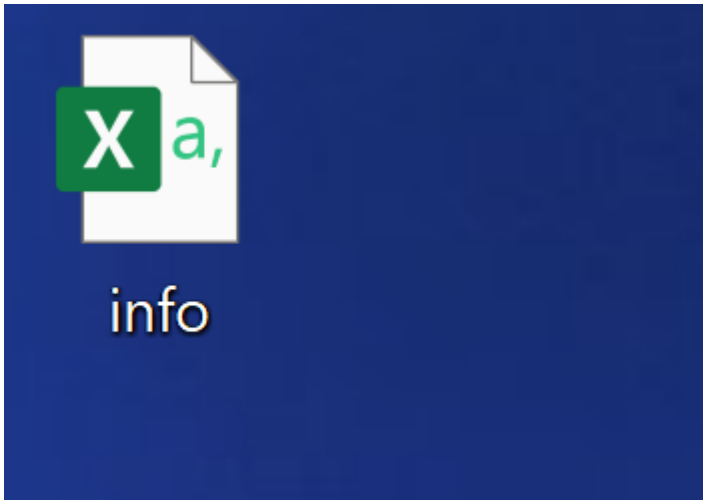
This is the code I have used for executing the task.

Flow of execution:

1.I have declared a string named lines containing the information that I need to store in the text file. Commas (,) in the string separate the text between rows and columns.

2. In the next statement, the console gets the location of the user's desktop.

3. I have created an empty text file named info.csv on the desktop and added the text in the lines string.



We can observe a info.csv text file on the Desktop.

StudentId	FirstName	LastName	DateOfBirt	ImageData
2E+08	Manoj	Gottumukk	18/10/97	/9j/4AAQSkZJRgABAQEAYABgAAD/4Tr+RXhpZgAATU0AKgAAAAgABgALAAIAAAAmAAAIYgESAAMAAAABAAEAAAExAAIAAAAmAAAliAEyAAIAAAAUAAAIrodpAAQAAAABAAAIwuocAAcAAAgMAAAAVgAAEUYc

This is the content in the text file.