Data Transfer Using Virtual Printer

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Engineering

Synopsis Report Sem VII *by*

Karishma Ghadge

Roll No.14

Sayali Jadhav

Roll No. 24

Chaitali Shevale

Roll No.47

Harshali Thale

Roll No. 53

Under the Supervision of

Prof .Mahesh Zemse

DEPARTMENT OF INFORMATION TECHNOLOGY KONKAN
GYANPEETH COLLEGE OF ENGINEERING KARJAT-410201

July 2021

Certificate

This is to certify that the project entitled **Recurrent Data Transfer using Virtual printer** is a bonafide work of **Karishma Ghadge (Roll No.14), Sayali jadhav(Roll No. 24),Chaitali Shevale (Roll No.47),Harshali Thale(Roll No.53)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **Undergraduate** in **DEPARTMENT OF INFORMATION TECHNOLOGY**.

Supervisor/Guide

Prof. Mahesh Zemse

Department of Information Technology

Head of Department	**Principal**
Prof.	
Department of Information	Dr. M.J. Lengare
Technology	Konkan Gyanpeeth College of Engineering

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and in- tegrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Signature

(Karishma Ghadge) Roll No.14

Signature

(Sayali Jadhav) Roll No.24

Signature

(Chaitali Shevale) Roll No.47

Signature

(Harshali Thale) Roll No.53

Date.

Abstract

The project aims is to share Securely encrypted file over cloud server using virtual printer driver And We can access that data/file anywhereLet's have a look at how virtual printer work First,we need to install virtual printer driver application in our windows/system and after installation, we will be able to see virtual printer driver will be shown in control panel (device and printer).

User will be able to prints a document through the virtual printer from any application. The virtual printer intercepts the print job and saves each printed page as EMF file.

The virtual printer converts **EMF files** to **Image** output formats. Data has been extracted from Image file into text or other human readable file format. after that data has been splitted into json /Xml file format Web api is being called to post that data over cloud server. Now, the data has been transferred securely from source to desired destination and available to access same data\file from anywhere.

Acknowledgements

We wish to express our profound and sincere gratitude to Prof.Mahesh Zemse, Department Information Technology, KGCE, Karjat, who guided us into the intricacies of this project with matchless magnanimity. We thank Prof.Anil kale, Head of the Dept. of Informa- tion Technology, KGCE Karjat and Dr. M. J. LENGARE, Principal, KGCE Karjat for extending their support during the Course of this investigation. We would be failing in our duty if we dont acknowledge the co-operation Rendered during various stages of im- age interpretation by. We are highly grateful to who evinced keen interest and invaluable support in the progress and successful completion of our project work. We are indebted to for their constant encouragement, co-operation and help. Words of Gratitude are not enough to describe the accommodation and fortitude which they have shown throughout my endeavour.

Contents

Chapter 1

INTRODUCTION

1.1 Introduction

In our project that is in Data Transfer Using Virtual Printer Driver we are going to implement application for to share Securely encrypted file over cloud server using virtual printer driver And We can access that data/file anywhere

Let’s have a look at how virtual printer work.

We are using Leadtools api’s for developing Virtual printer driver its help to for developing virtual printer driver in C# language to that its easy for developer for developing

First, we need to install virtual printer driver application in our windows/system and after installation, we will be able to see virtual printer driver will be shown in control panel (device and printer).

User will be able to prints a document through the virtual printer from any application. The virtual printer intercepts the print job and saves each printed page as EMF file.

The virtual printer converts EMF files to Image output formats. Data has been extracted from Image file into text or other human readable file format. after that data has been splitted into json /Xml file format Web api is being called to post that data over cloud server. Now, the data has been transferred securely from source to desired destination and available to access same data\file from anywhere.

Function:

Typical uses of virtual printers include:

- Saving a document to another format such as a PDF or multi-page TIFF file.
- Sending documents to a server and we can access from anywhere.
- Allowing user to control certain aspects of printing not supported natively, such as printing multiple pages per sheet without border, print letterhead, watermarks etc.
- This output can either be saved in a file for future printing or passed to another printer.

- Previewing a printed document before printing it, to save ink and paper.This functionality is also built into many GUI applications. Allowing remote printing of documents over the Internet.
- At least one example of this technology creates a virtual printer on one computer which actually converts the document and sends it to a remote server, from which the file can be printed to a printer attached to a PC in a remote location.
- Similar technology is also being used to allow printing from devices such as smart phones.
- In our project that is in Data Transfer Using Virtual Printer Driver we are going to implement application for to share Securely encrypted file over cloud server using virtual printer driver And We can access that data/file anywhere
- Let’s have a look at how virtual printer work.
- We are using Leadtools api’s for developing Virtual printer driver its help to for developing virtual printer driver in C# language to that its easy for developer for developing
- First, we need to install virtual printer driver application in our windows/system and after installation, we will be able to see virtual printer driver will be shown in control panel (device and printer).
- User will be able to prints a document through the virtual printer from any application. The virtual printer intercepts the print job and saves each printed page as EMF file.
- The virtual printer converts EMF files to Image output formats. Data has been extracted from Image file into text or other human readable file format. after that data has been splitted into json /Xml file format Web api is being called to post that data over cloud server. Now, the data has been transferred securely from source to desired destination and available to access same data\file from anywhere.

1.1 Objectives

The objective of virtual printer is to share securely encrypted file over cloud server using virtual printer driver and we can access that data Or file from anywhere.

1. Virtual printer work with print drivers that are coded to send their output to other applications rather than to a physical device.
2. The virtual printer can help save resources because it makes it possible to carry out tasks that would formerly have involved actual printing without wasting paper and ink And EMF file.
3. User will be able to print a document through the virtual printer from any application
4. Virtual printer converting documents of any type into **PDF** format or image files such as **JPEG**, **TIFF**, GIF, PNG, BMP, PCX or DCX.
5. The virtual printers converts EMF files to Image output format.
6. Data will be extracted from Image file into the text or other human readable file format.
7. With the help of virtual printers drivers data has been transferred securely from source to destination.

1.2 Purpose, Scope, and Applicability

1.1.1 Purpose

The purpose of the Project is to The objective of virtual printer is to share securely encrypted file over server using virtual printer driver and we can access that data Or file from anywhere.

The virtual printer can help save resources because it makes it possible to carry out tasks that would formerly have involved actual printing without wasting paper and ink And EMF file.

1.1.2 Scope

This application is for only desktop computer so in future we can develop for allowing remote printing of documents over the Internet

1.1.3 Applicability

The objective of virtual printer is to share securely encrypted file over cloud server using virtual printer driver and we can access that data Or file from anywhere.

The virtual printer can help save resources because it makes it possible to carry

out tasks that would formerly have involved actual printing without wasting paper and ink And EMF file. virtual printing is widely used. It is most commonly used for converting documents into images in various formats or for sending faxes from a computer.

1.3 Organisation of Report

We did Literature Survey on various standard papers like Control of distributed computer using virtual printer driver Execute Module, Virtual printer and Leadtools Virtual printer etc. After reading these papers we found that their are various technology and languages used like C,C++,C# and Xml. So that We are most comfortable with C# so that we decided to developing driver using Leadtools

Chapter 2

LITERATURE SURVEY

TABLE 2.1: Paper Comparison

Sr.No.	Title	Description
1	Control of distributed computerusing virtual printer driver Execute Module Control of distributed computerusing virtual printer driver Execute Module (2007)	Author :Fumihiko Iwata, Nagano-ken (JP); Nagano-ken (JP); Akihiro Sato, Nagano-ken (JP); Nagano-ken (JP); Togashi, Nagano-ken (JP) Working : The application program converts print data into data Suitable for the virtual printer based on the performance information and transmits the converted print data to the virtual printer driver
2	Application of Digital Virtual Prototype Technology in Simulation Design of Paper Delivery Mechanism of Printing Press (2020)	Author: Ming He Working: Digital virtual prototype technology has an important role in promoting many industries in China and the printing industry. The use of it not only meets the actual needs of the industry, but also effectively promotes the optimization of resources

Chapter 3

REQUIREMENTS AND ANALYSIS

Problem Definition

1) Time consumption

The time consuming is more in existing system . When user gives the inputs (print) In th form of any documents format (.pdf,.doc,.txt) to the virtual printer driver then very difficult to analyze the formt of particular input file and also take too much time to convert inyo desire output But in our proposed system we overcome tha problem we used leadtool 21

2) user interface

In existing system have less user friendly . So that user don't get see input file So far solution to this we can make our proposed system more user friendly so that can see their input file before sending to server

4.1 Requirements Specification

Software and Hardware Requirements

- *Intel processor of AMD dual core.*
- *2 GB RAM.*
- *500 GB Hard disk.*
- *Various software prerequisites are as follows:*
- *Windows XP,7,8,10.*
- *Visual Studio 17.*
- *SQL server 14.*
- *Leadtools.*
- *Programming languages called C# and ASP.Net.*

4.2 Scheduling and Planning

In Planning we did the Requirement Analysis of Project from last week of July to mid September After that we referred many of the Papers related to Sentence Completion task and Design our project from mid september to last Week of October and We will start the implementation of our project in the month of December and complete it by Second week of March and with implementation will also and also will start the documentaion from mid Feb and Complete it by the April month. The detailed information of our planning and scheduling is shown below.

Chapter 4

SYSTEM DESIGN

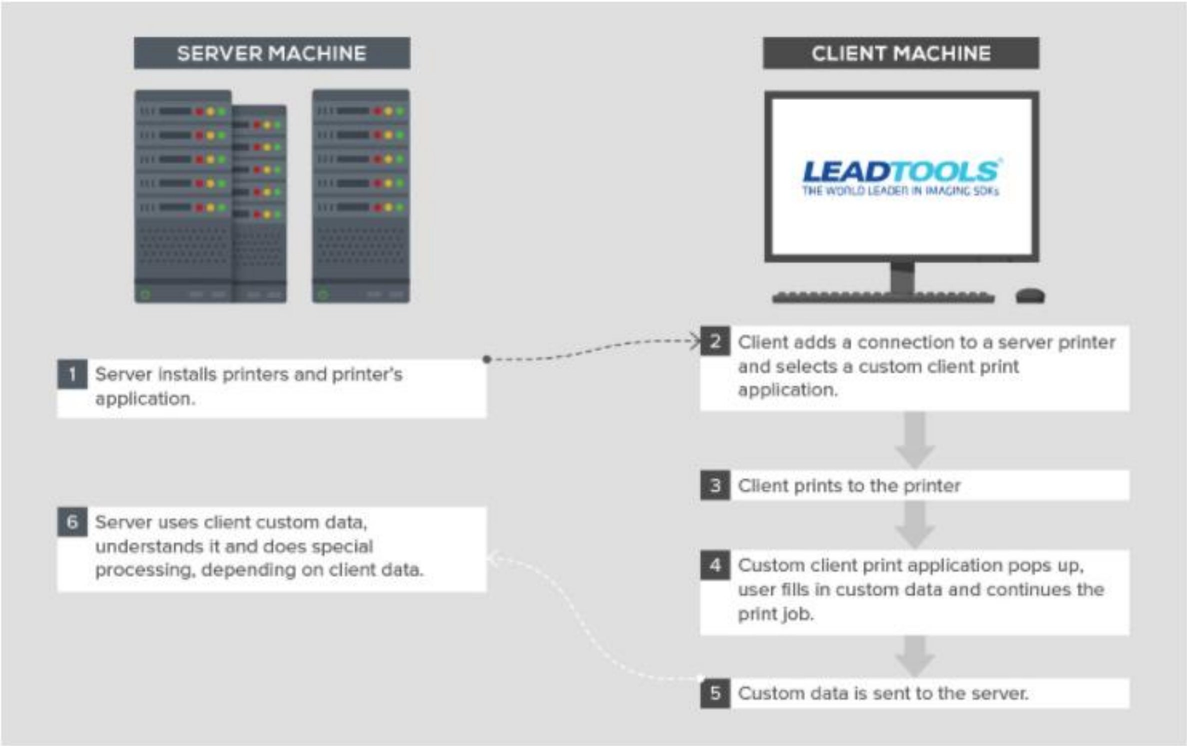
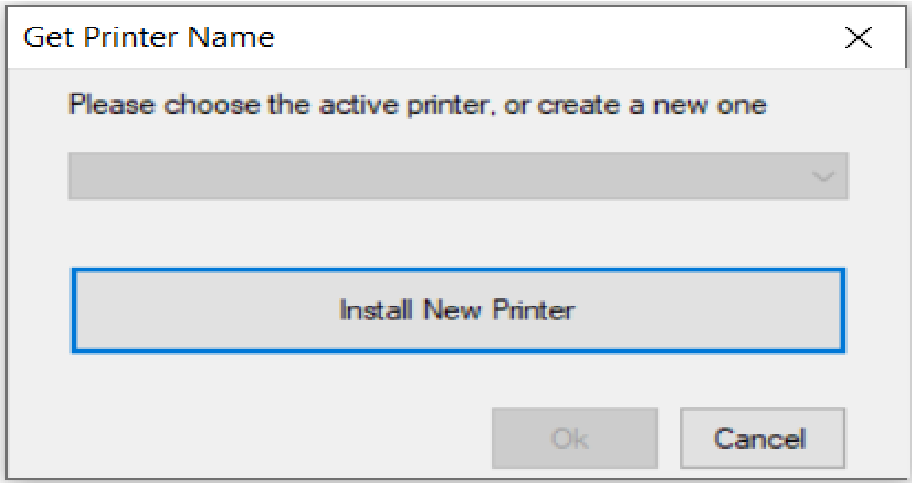

SERVER MACHINE
CLIENT MACHINE
LEADTOOLS
THE WORLD LEADER IN IMAGING SDKs
1 Server installs printers and printer's application.
2 Client adds a connection to a server printer and selects a custom client print application.
3 Client prints to the printer
4 Custom client print application pops up, user fills in custom data and continues the print job.
5 Custom data is sent to the server.
6 Server uses client custom data, understands it and does special processing, depending on client data.

FIGURE 4.1: System Design

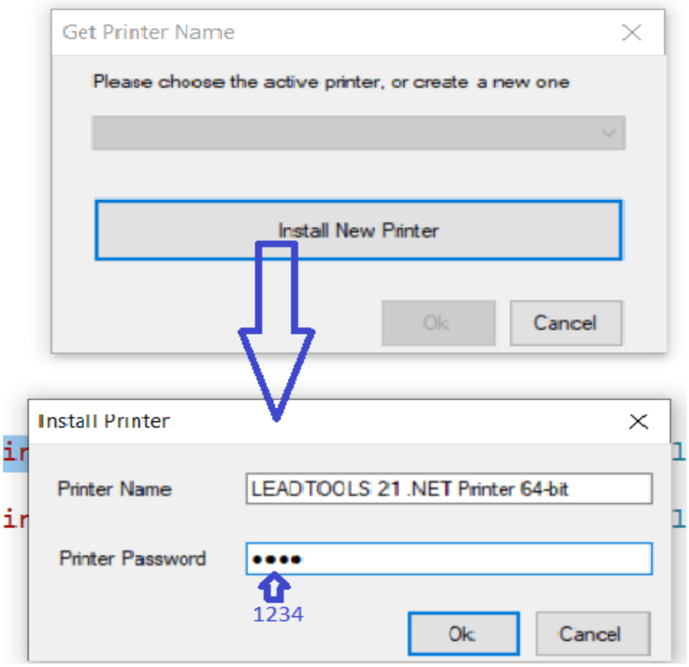
Chapter 5

Coding and Implementation



```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace PrinterDemo
{
  public partial class FrmGetPrinterName : Form
  {
    #region Constructor...
    public FrmGetPrinterName()
    {
      InitializeComponent();
    }

    public FrmGetPrinterName(string activePrinter)
    {
      InitializeComponent();
      _printerName = activePrinter;
    }
    #endregion

    #region Fields...
    PrintingUtilities _printingUtilities = new PrintingUtilities();
    string _printerName = string.Empty;
    #endregion
    #region Properties...
    public string PrinterName
    {
      get
      {
        return _printerName;
      }
    }
    #endregion

    #region Events...
    private void FrmGetPrinterName_Load(object sender, EventArgs e)
    {
      try
      {
        _cmbPrintersList.Items.Clear();
        _btnInstallNewPrinter.Visible = (_printerName == string.Empty);
        FillLeadtoolsPrintersList();
        EnableControls();
      }
      catch (Exception Ex)
      {
        MessageBox.Show(Ex.ToString(), "LEADTOOLS Printer Demo", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
      }
    }

    private void FrmGetPrinterName_FormClosed(object sender, FormClosedEventArgs e)
    {
      try
      {
        if (_cmbPrintersList.Items.Count > 0)
        {
          _printerName = _cmbPrintersList.Text;
        }
      }
      catch (Exception Ex)
      {
        MessageBox.Show(Ex.ToString(), "LEADTOOLS Printer Demo", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
      }
    }

    private void _btnInstallNewPrinter_Click(object sender, EventArgs e)
    {
      try
      {
        FrmInstallPrinter frmInstallPrinter = new FrmInstallPrinter();
        DialogResult dialogResult = frmInstallPrinter.ShowDialog();

        if (dialogResult == DialogResult.OK)
        {
          Cursor = Cursors.WaitCursor;
          string newPrinterName = _printerName = frmInstallPrinter.PrinterName;
          string newPrinterPassword = frmInstallPrinter.PrinterPassword;

          PrintingUtilities.InstallNewPrinter(newPrinterName, newPrinterPassword);
          _printerName = newPrinterName;
          FillLeadtoolsPrintersList();
        }
      }
      catch (Exception Ex)
      {
        MessageBox.Show(Ex.ToString(), "LEADTOOLS Printer Demo", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
      }
      finally
      {
        Cursor = Cursors.Default;
      }
    }
    #endregion

    #region Methods...

    private bool Is64Bit()
    {
      return IntPtr.Size == 8;
    }

    private void FillLeadtoolsPrintersList()
    {
      string setupPrinter = string.Empty;
      if (Is64Bit())
        setupPrinter = string.Format("LEADTOOLS {0} .NET Printer 64-bit",
Leadtools.Demos.DemosGlobal.LTVersion);
      else
        setupPrinter = string.Format("LEADTOOLS {0} .NET Printer 32-bit",
Leadtools.Demos.DemosGlobal.LTVersion);

      try
      {
        _cmbPrintersList.Items.Clear();
        _cmbPrintersList.Items.AddRange(PrintingUtilities.GetLeadtoolsPrintersList());

        if (_cmbPrintersList.Items.Count > 0)
        {
          if (_printerName != string.Empty)
          {
            _cmbPrintersList.Text = _printerName;
          }
          else
          {
            _cmbPrintersList.SelectedIndex = 0;
          }

          if (_printerName == string.Empty)
          {
            for (int i = 0; i < _cmbPrintersList.Items.Count; i++)
            {
              if ((_cmbPrintersList.Items[i] as string).ToLower() == setupPrinter.ToLower())
                _cmbPrintersList.SelectedIndex = i;
            }
          }
        }
        else
        {
          string errorMessage = "No printers are available.";
          MessageBox.Show(errorMessage, "LEADTOOLS Printer Demo", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
        }
        EnableControls();
      }
      catch (Exception Ex)
      {
        MessageBox.Show(Ex.ToString(), "LEADTOOLS Printer Demo", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
      }
    }

    private void EnableControls()
    {
      try
      {
        bool bprinterExist = (_cmbPrintersList.Items.Count > 0);
        _btnOk.Enabled = bprinterExist;
        _cmbPrintersList.Enabled = bprinterExist;
      }
      catch (Exception Ex)
      {
        MessageBox.Show(Ex.ToString(), "LEADTOOLS Printer Demo", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
      }
    }
    #endregion
  }
}
```

2)



Get Printer Name

Please choose the active printer, or create a new one

Install Printer

Printer Name: LEADTOOLS 21 .NET Printer 64-bit

Printer Password: ●●●●

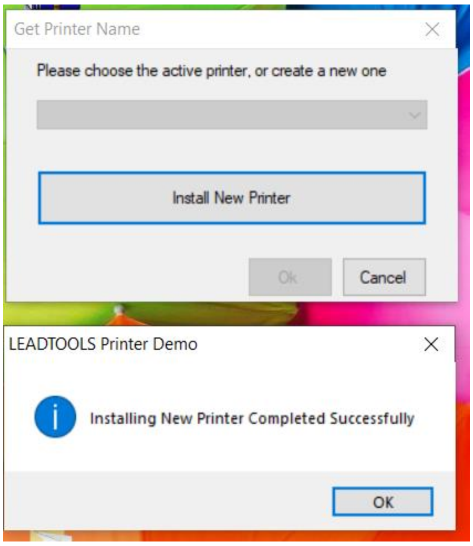
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace PrinterDemo
{
  public partial class FrmInstallPrinter : Form
  {
    #region Constructor...
    public FrmInstallPrinter()
    {
      InitializeComponent();
    }
    #endregion

    #region Fields...
    string _printerName = string.Empty;
    string _printerPassword = string.Empty;
    #endregion

    #region Properties...
    public string PrinterName
    {
```

```
    get
    {
      return _printerName;
    }
  }

  public string PrinterPassword
  {
    get
    {
      return _printerPassword;
    }
  }
  #endregion

  #region Events...
  private void FrmInstallPrinter_Load(object sender, EventArgs e)
  {
    try
    {
      EnableControls();
    }
    catch (Exception Ex)
    {
      MessageBox.Show(Ex.ToString(), "LEADTOOLS Printer Demo", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
    }
  }

  private void FrmInstallPrinter_FormClosing(object sender, FormClosingEventArgs e)
  {
    try
    {
      _printerName = _txtBoxPrinterName.Text;
      _printerPassword = _txtBoxPrinterPassword.Text;
    }
    catch (Exception Ex)
    {
      MessageBox.Show(Ex.ToString(), "LEADTOOLS Printer Demo", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
    }
  }

  private void _txtBoxPrinterName_TextChanged(object sender, EventArgs e)
  {
    try
    {
      EnableControls();
    }
    catch (Exception Ex)
    {
      MessageBox.Show(Ex.ToString(), "LEADTOOLS Printer Demo", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
    }
  }
  #endregion
```

```
    #region Methods...
    private void EnableControls()
    {
      try
      {
        bool bEnable = (_txtBoxPrinterName.Text != string.Empty);
        _btnOk.Enabled = bEnable;
      }
      catch (Exception Ex)
      {
        MessageBox.Show(Ex.ToString(), "LEADTOOLS Printer Demo", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
      }
    }
    #endregion
  }
}
```

b) Password window coding

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace PrinterDemo
{
  public partial class FrmPassword : Form
  {
    #region Constructor...
    public FrmPassword(bool bLock)
    {
      InitializeComponent();
      _bLock = bLock;
    }
    #endregion

    #region Fields...
    private string _password = string.Empty;
    bool _bLock = false;
    #endregion

    #region Properties...
    public string Password
    {
      get
      {
        return _password;
      }
    }
    #endregion

    #region Events...
    private void FrmPassword_FormClosing(object sender, FormClosingEventArgs e)
    {
      try
      {
        _password = _txtBoxPassword.Text;
      }
      catch (Exception Ex)
      {
        MessageBox.Show(Ex.ToString(), "LEADTOOLS Printer Demo", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
      }
    }
    #endregion

    private void FrmPassword_Load(object sender, EventArgs e)
    {
      try
      {
        string formTitle = string.Empty;

        if (_bLock)
        {
          formTitle = "Lock Printer";
        }
        else
        {
          formTitle="Unlock Printer";
        }
        this.Text = formTitle;
      }
      catch (Exception Ex)
      {
        MessageBox.Show(Ex.ToString(), "LEADTOOLS Printer Demo", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
      }
    }

  }
}
```



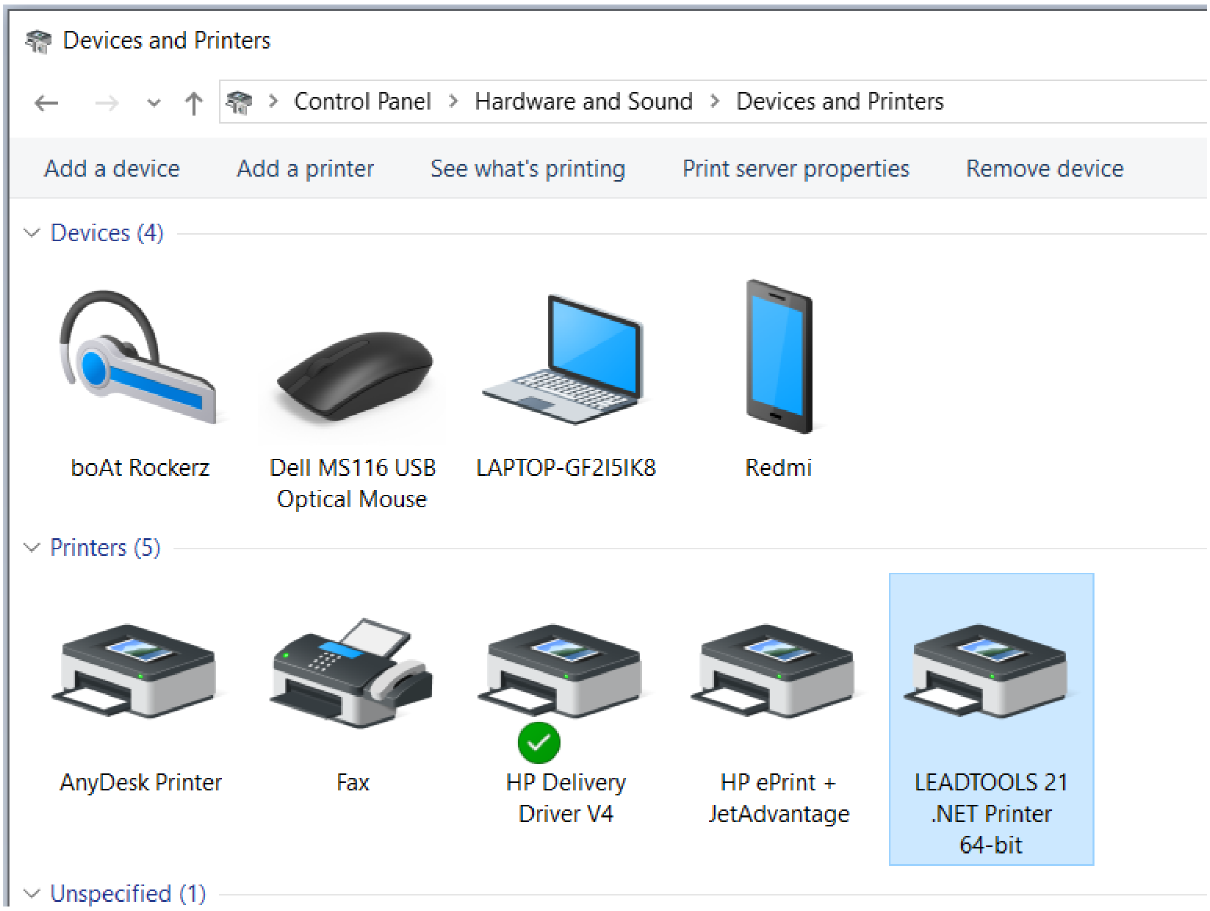
Get Printer Name

Please choose the active printer, or create a new one

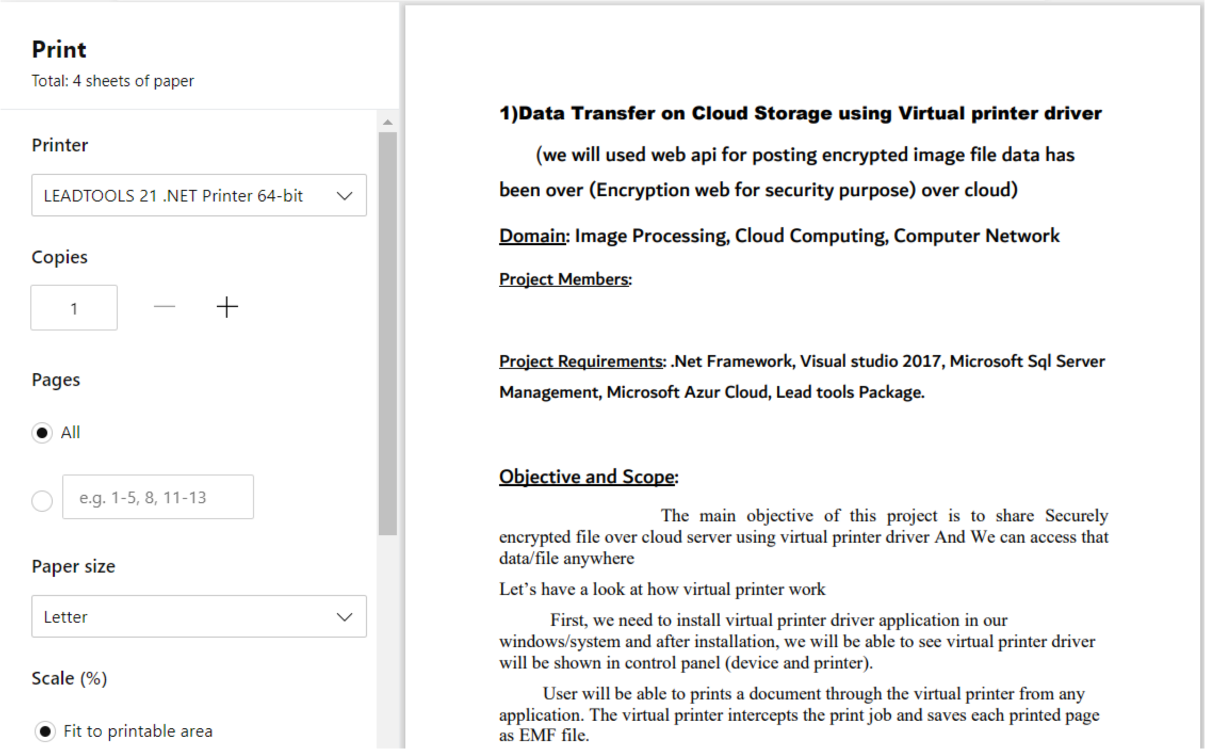
LEADTOOLS Printer Demo

Installing New Printer Completed Successfully

After Installation Successfully Then We can See Printer Driver Icon in Control panel Device and Printer setting



4)



Print

Total: 4 sheets of paper

Printer: LEADTOOLS 21 .NET Printer 64-bit

Copies: 1

Pages: ● All ○ e.g. 1-5, 8, 11-13

Paper size: Letter

Scale (%): ● Fit to printable area

1)Data Transfer on Cloud Storage using Virtual printer driver

(we will used web api for posting encrypted image file data has been over (Encryption web for security purpose) over cloud)

Domain: Image Processing, Cloud Computing, Computer Network

Project Members:

Project Requirements: .Net Framework, Visual studio 2017, Microsoft Sql Server Management, Microsoft Azur Cloud, Lead tools Package.

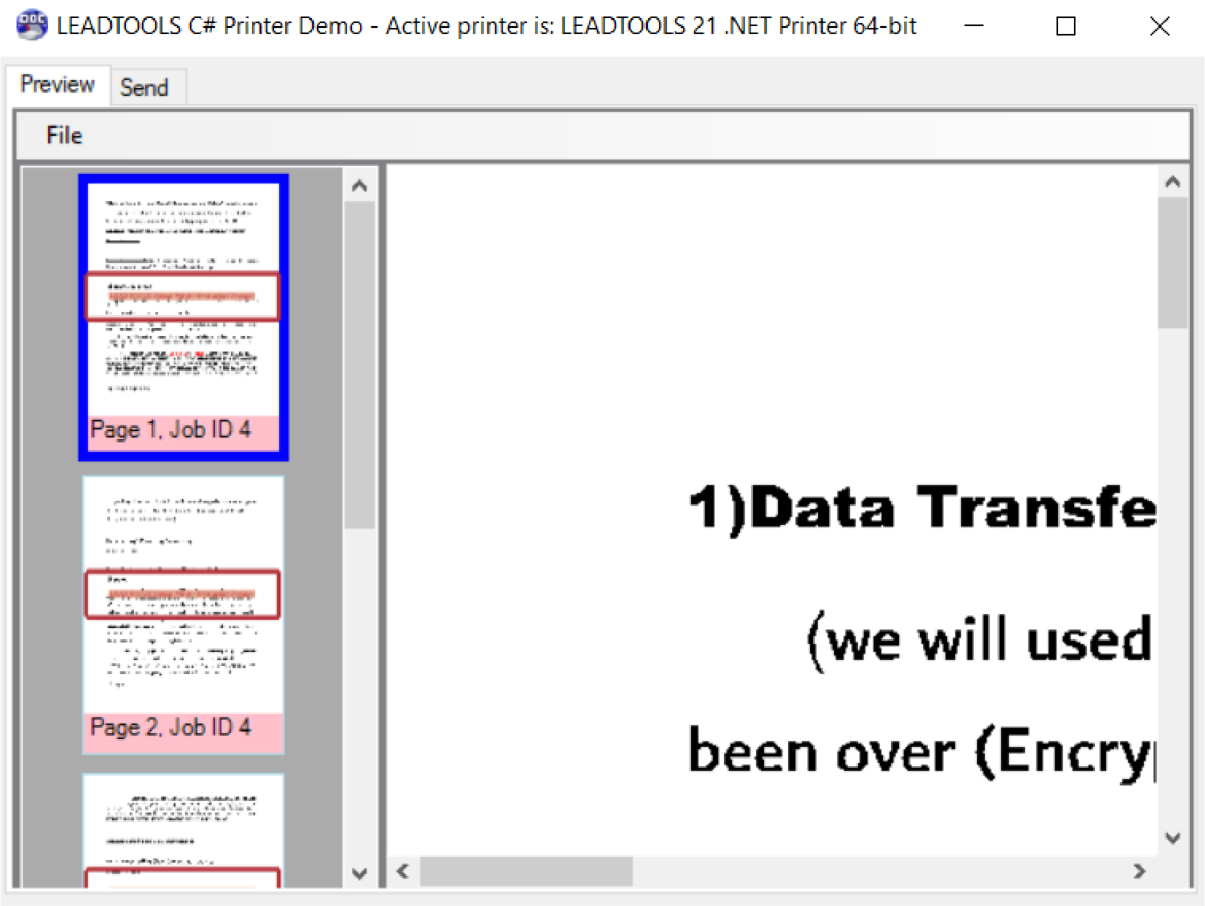
Objective and Scope:

The main objective of this project is to share Securely encrypted file over cloud server using virtual printer driver And We can access that data/file anywhere

Let's have a look at how virtual printer work

First, we need to install virtual printer driver application in our windows/system and after installation, we will be able to see virtual printer driver will be shown in control panel (device and printer).

User will be able to prints a document through the virtual printer from any application. The virtual printer intercepts the print job and saves each printed page as EMF file.



LEADTOOLS C# Printer Demo - Active printer is: LEADTOOLS 21 .NET Printer 64-bit

Page 1, Job ID 4

Page 2, Job ID 4

1)Data Transfe

(we will used

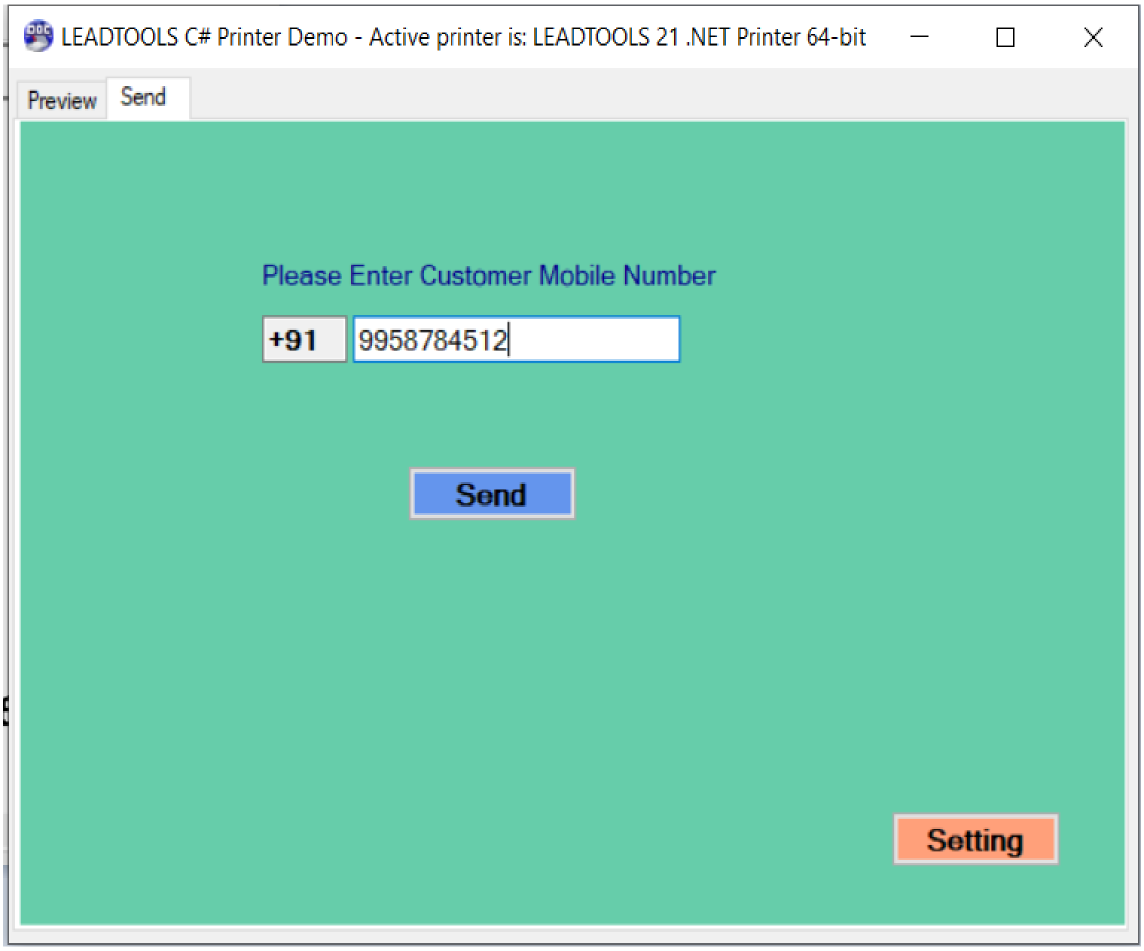
been over (Encry

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Drawing.Imaging;
using System.Drawing.Printing;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.Runtime.InteropServices;
using Leadtools;
using Leadtools.Demos;
using Leadtools.Demos.Dialogs;
using Leadtools.Printer;
using Leadtools.Document.Writer;
using Leadtools.Codecs;
using PrinterDemo.UI;
using System.Data.SqlClient;


namespace PrinterDemo
{

  struct JobInfoStruct
  {
    private int _jobId;
    private int _pageNo;

    public JobInfoStruct(int jobId, int pageNo)
    {
      _jobId = jobId;
      _pageNo = pageNo;
    }

    public int JobId
    {
      get { return _jobId; }
    }

    public int PageNo
    {
      get { return _pageNo; }
    }
  }

#region Properties...
    private bool ReceiveJobEvent
    {
      get
      {
        return _miEventsJob.Checked;
      }
    }

    private bool ReceiveEmfEvent
    {
      get
      {
        return _miEventsEmf.Checked;
      }
    }

    private bool ViewOutputFile
    {
      get
      {
        if (_miViewOutputFile == null)
          return true;

        return _miViewOutputFile.Checked;
      }
    }
    #endregion
  public partial class FrmMain : Form
  {
    //public Image img = null;
     SqlCommand Com;
     //------------------------------
    [DllImport("kernel32.dll")]
    static extern IntPtr GlobalLock(IntPtr hMem);

    [DllImport("kernel32.dll")]
    static extern bool GlobalUnlock(IntPtr hMem);

    [DllImport("kernel32.dll")]
    static extern bool GlobalFree(IntPtr hMem);

#if LEADTOOLS_V175_OR_LATER
    //Dotnet does not have functions to add or remove font files, we will have to use the API ones
    [DllImport("gdi32")]
    public static extern int AddFontResource(string lpFileName);

    [DllImport("gdi32")]
    public static extern int RemoveFontResource(string lpFileName);
#endif

    private const Int32 DM_OUT_BUFFER = 14;

    #region Main...
    /// <summary>
    /// The main entry point for the application.
    /// </summary>
    [STAThread]
    static void Main(string[] args)
    {
      try
      {
#if LEADTOOLS_V175_OR_LATER
        if (!Support.SetLicense())
          return;
#else
        Support.Unlock(false);
#endif // #if LEADTOOLS_V175_OR_LATER
```

```
        if (args.Length > 0)
        {
          FrmMain.StartedPrinter = args[0];

        }

        Application.EnableVisualStyles();
        Application.SetCompatibleTextRenderingDefault(false);
        Application.Run(new FrmMain());
      }
      catch
      {
      }
    }
    #endregion

    #region Constructor...
    public FrmMain()
    {
      try
      {
        if (InitClass())
          InitializeComponent();
        else
          Close();
      }
      catch (Exception Ex)
      {
        MessageBox.Show(Ex.Message, "LEADTOOLS Printer Demo", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        Close();
      }
    }
    #endregion

    #region Fields...
    PrintDocument _printDocument = new PrintDocument();
    Printer _printer;
    FrmProgress _frmProgress = new FrmProgress();
    List<IntPtr> _lstMetaFiles = new List<IntPtr>();
    List<JobInfoStruct> _lstJobInfo = new List<JobInfoStruct>();
    int _pageNo = 0;
    int _jobId = 0;
    string _currentPrinterName = string.Empty;
    static string StartedPrinter = string.Empty;
    bool bSelectedPrinter = true;
    RasterCodecs _codec;
    List<string> _tempFiles = new List<string>();
#if LEADTOOLS_V175_OR_LATER
    //Array to hold the font names
    List<string> _fonts = new List<string>();
    //Path to save the font files to
    string _fontsPath;
#endif
    #endregion
```

```
private void FrmMain_Load(object sender, EventArgs e)
  {
   try
   {
    _codec = new RasterCodecs();
    _miPrinterDefaultSpecs.Visible = true;
    if (bSelectedPrinter == false)
     this.Close();
    else
    {
     _printDocument.PrintPage += new PrintPageEventHandler(_printDocument_PrintPage);
    }

    _miViewOutputFile.Checked = true;
    this.Text = "LEADTOOLS C# Printer Demo - Active printer is: " + _currentPrinterName;

    string newGuid = Guid.NewGuid().ToString("N");
    //Get the path to the shared documents
    _fontsPath = Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments),
newGuid);
    Directory.CreateDirectory(_fontsPath);
   }
   catch (Exception Ex)
   {
    MessageBox.Show(Ex.Message, "LEADTOOLS Printer Demo", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
   }
   finally
   {
   }
  }

public void _printer_EmfEvent(object sender, EmfEventArgs e)
  {
   this.Enabled = false;
   string tempFile = Path.GetTempFileName();
   _tempFiles.Add(tempFile);
   File.WriteAllBytes(tempFile, e.Stream.ToArray());

   Metafile metaFile = new Metafile(e.Stream);
   _pageNo++;

   if (_frmProgress.Visible)
   {
    _frmProgress.SetProgressState(_pageNo, _jobId);
   }

   Image emfImage = metaFile.GetThumbnailImage(_lstBoxPages.Width, _lstBoxPages.ItemHeight, null,
IntPtr.Zero);
    //----------------------
   // img = emfImage;
    //---------------------------
   int nLastIndex = _lstBoxPages.Items.Add(emfImage);
   _lstMetaFiles.Add(metaFile.GetHenhmetafile());
   _lstJobInfo.Add(new JobInfoStruct(_jobId, _pageNo));
  }
```

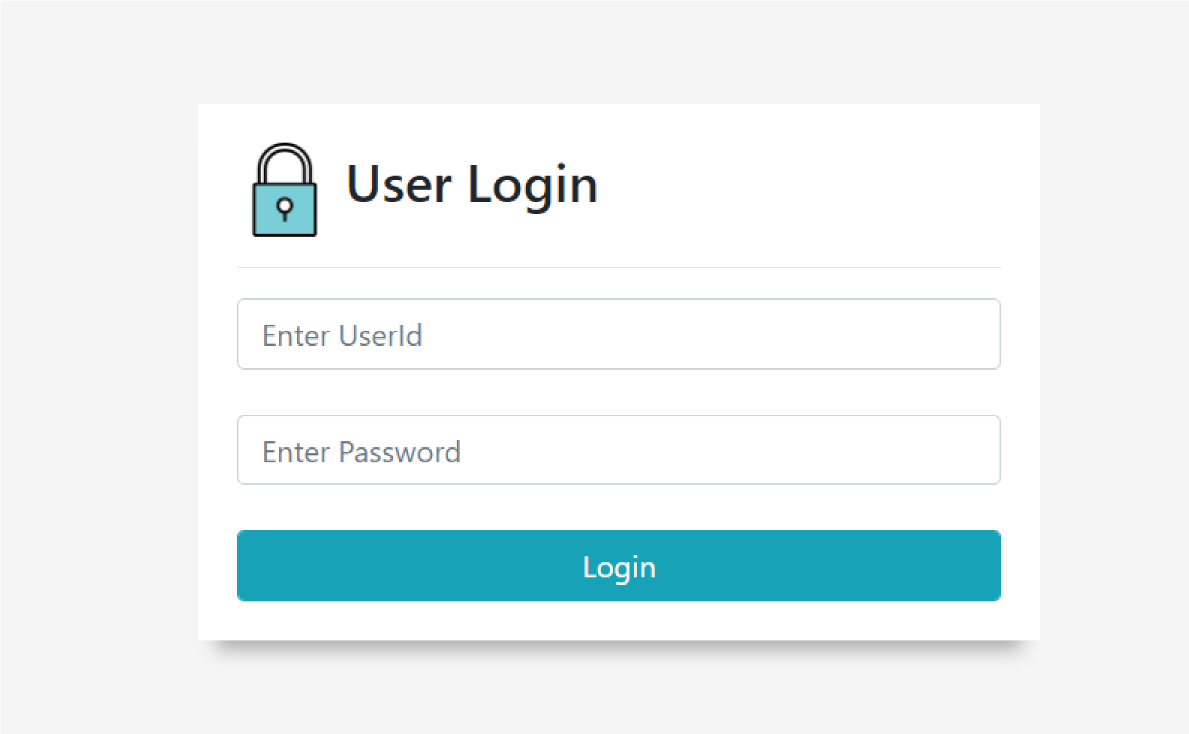
```
public void _printer_JobEvent(object sender, JobEventArgs e)
{
  if (e.JobEventState == EventState.JobStart)
  {
    this.Enabled = true;
    this.BringToFront();
    this.Focus();
    _pageNo = 0;
    _jobId = e.JobID;
    if (!_frmProgress.Visible)
    {
      _frmProgress = new FrmProgress(e.PrinterName, _printer);
      _frmProgress.Show();
    }
  }
  else if (e.JobEventState == EventState.JobEnd)
  {
    _frmProgress.Close();
    //job-end event we may have embedded font attached to the job
    //we will save the font files so we can use them when saving
    AddAndInstallFonts(e.JobID);
    this.Enabled = true;
    _lstBoxPages.SelectedIndex = 0;
    this.BringToFront();
    this.Focus();
  }
}
```

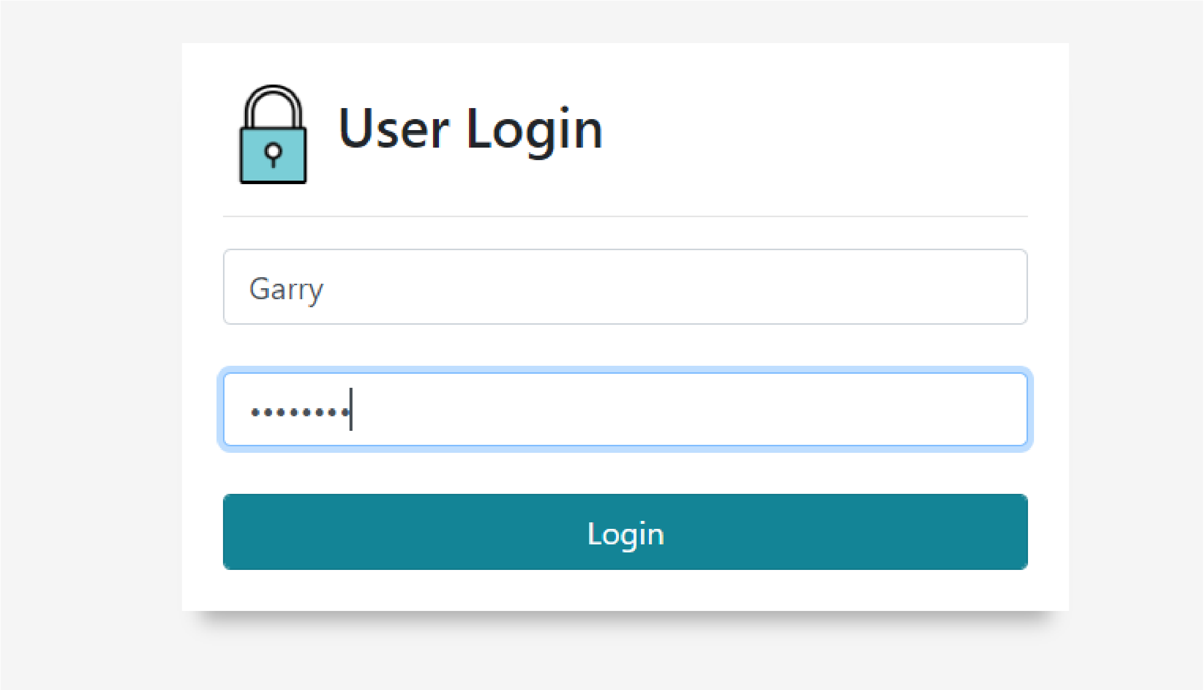

LEADTOOLS C# Printer Demo - Active printer is: LEADTOOLS 21 .NET Printer 64-bit
Preview
Send
Please Enter Customer Mobile Number
+91
9958784512
Send
Setting

```
private void Btn_Settings_Click(object sender, EventArgs e)
    {
      ConfigSetting Obj = new ConfigSetting();
      Obj.ShowDialog();
    }

    #region Btn_Send_Click
    private void Btn_Send_Click(object sender, EventArgs e)
    {
      try
      {
        if(_lstBoxPages.Items!=null && _lstBoxPages.Items.Count > 0)
        {
          foreach (Image img in _lstBoxPages.Items)
          {
            //--------------------------------------------
            SqlCommand com;
            string MachineName = Environment.MachineName.ToString();
            DateTime Date = System.DateTime.Today;
            long Number = Convert.ToInt64(Txt_MobileNo.Text.ToString());
            if (Number == null)
            {
              MessageBox.Show("Please Enter Mobile Number");
              Txt_MobileNo.Focus();
              return;
            }
            else
            {
              //---------------------------------------------------
              string SQlConnectionstring = "Data Source=LAPTOP-GF2I5IK8;Initial Catalog=VP;Integrated
Security=True";
              SqlConnection Con = new SqlConnection(SQlConnectionstring);
              Con.Open();
              //----------------------------------------------------

              //---------------------------------------------
              string query = "Insert into PrinterData values(@payload,'" + MachineName.ToString() + "','" +
Date.ToString("dd/MM/yyyy") + "','" + Number + "')";
              //-------------------------
              MemoryStream tmpStream = new MemoryStream();
              img.Save(tmpStream, ImageFormat.Png); // change to other format
              tmpStream.Seek(0, SeekOrigin.Begin);
              byte[] imgBytes = new byte[tmpStream.Length];
              tmpStream.Read(imgBytes, 0, imgBytes.Length);

              //---------------
              Com = new SqlCommand(query, Con);
              IDataParameter par = Com.CreateParameter();
              par.ParameterName = "payload";
              par.DbType = DbType.Binary;
              par.Value = imgBytes;
              Com.Parameters.Add(par);
              Com.ExecuteNonQuery();
              //--------------------------
              // Com = new SqlCommand(query, Con);
              //// conv_photo();
              // Com.ExecuteNonQuery();
          Con.Close();
          MessageBox.Show("Data Send Successwfully");
        }
    }
      this.Close();

    }
  }
  catch (Exception ex)
  {
    MessageBox.Show(ex.Message);
  }



}

#endregion


private void Txt_MobileNo_KeyPress(object sender, KeyPressEventArgs e)
{
  if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar) && (e.KeyChar != '.'))
  {
    e.Handled = true;
  }
}
```

2) Second Phase Coding

Login Page





Garry

••••••••

Front end code:

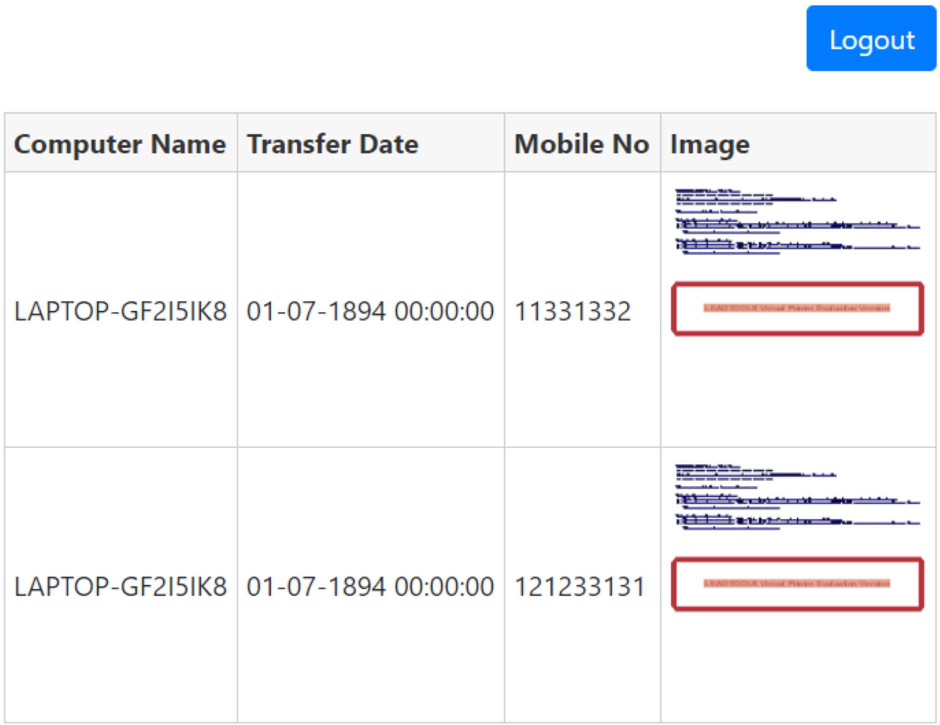
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Login.aspx.cs"
Inherits="LoginApp.Login" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title>User Login</title>
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
    integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
crossorigin="anonymous"/>

  <style>

    .wrapper1
    {
      height:100vh !important;
      display:flex;
      align-items:center;
      flex-direction:column;
      justify-content:center;
      width:100% !important;
      padding:20px;
      background-color:#f5f5f5 !important;
    }

    .logincontainer
    {
      border-radius:0px;
      background-color:#fff;
```

```
            width:90%;
            max-width:450px;
            position:relative;
            padding:20px;
            border:1px white solid ;
            box-shadow:0 15px 10px -10px #acacac;
        }

    </style>

</head>
<body>
    <form id="form1" runat="server">
        <div class="wrapper1">

            <div class="logincontainer">

                <h3>
                    <img src="Content/login.png" width="50px" height="50px" /> User Login
                </h3>

                <hr />

                <asp:TextBox runat="server" ID="txtUserId" placeholder="Enter UserId" CssClass="form-control"/>
                <br />

                <asp:TextBox runat="server" ID="txtPassword" placeholder="Enter Password"
TextMode="Password"
                    CssClass="form-control"/>
                <br />

                <asp:Button runat="server" ID="btnLogin" CssClass="btn btn-info form-control" Text="Login"
                    OnClick="btnLogin_Click"/>
                <br />
                <asp:Label runat="server" ID="txtInfo" />
            </div>

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

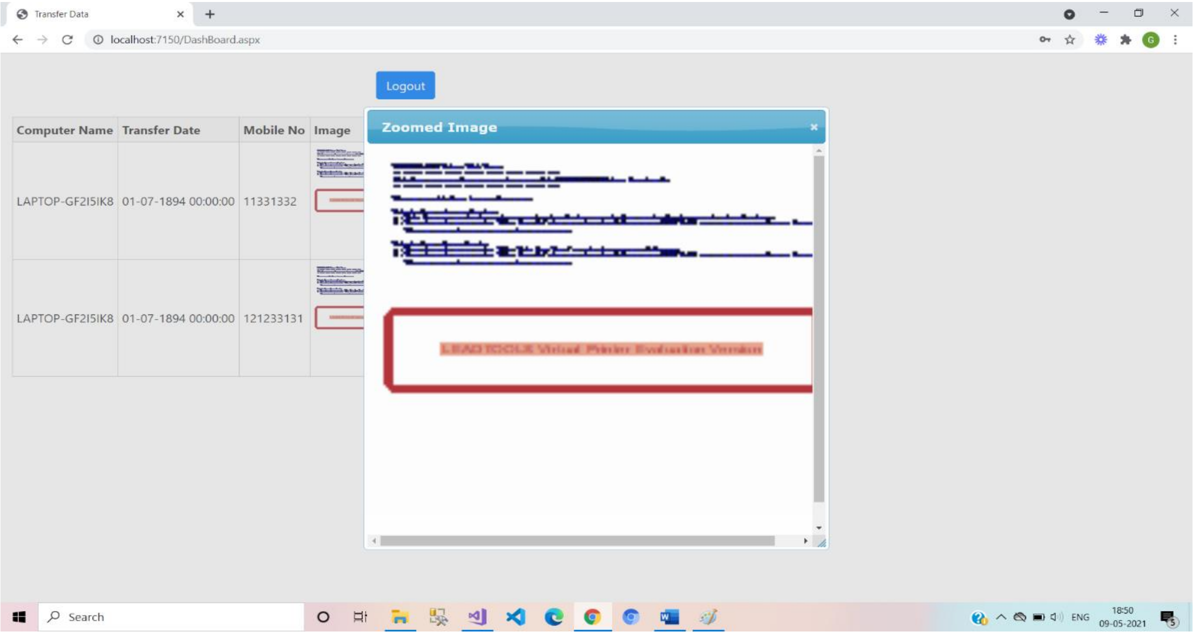
Backend code :

```
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;

namespace LoginApp
{
    public partial class Login : System.Web.UI.Page
    {
        static string connectionString = @"Data Source=LAPTOP-GF2I5IK8;Initial Catalog=VP;Integrated
Security=True";
        protected void Page_Load(object sender, EventArgs e)
        {

        }
        protected void btnLogin_Click(object sender, EventArgs e)
        {
            try
            {
                SqlConnection con = new SqlConnection(connectionString);
                con.Open();
                SqlCommand cmd = new SqlCommand("Select * from login where username='" + txtUserId.Text + "'
and pwd ='" + txtPassword.Text + "'", con);
                SqlDataAdapter da = new SqlDataAdapter(cmd);
                DataTable dt = new DataTable();
                da.Fill(dt);
               if (dt.Rows.Count > 0)
                {

                    txtInfo.Text = "Login Successful!";


                    con.Close();

                    Response.Redirect("~/DashBoard.aspx");
                }
                else
                {
                    txtInfo.Text = "Invalid credentials";
                }

                con.Close();

            }
            catch(Exception ex)
            {

            }
        }
    }
}
```

Transfer Data Page :

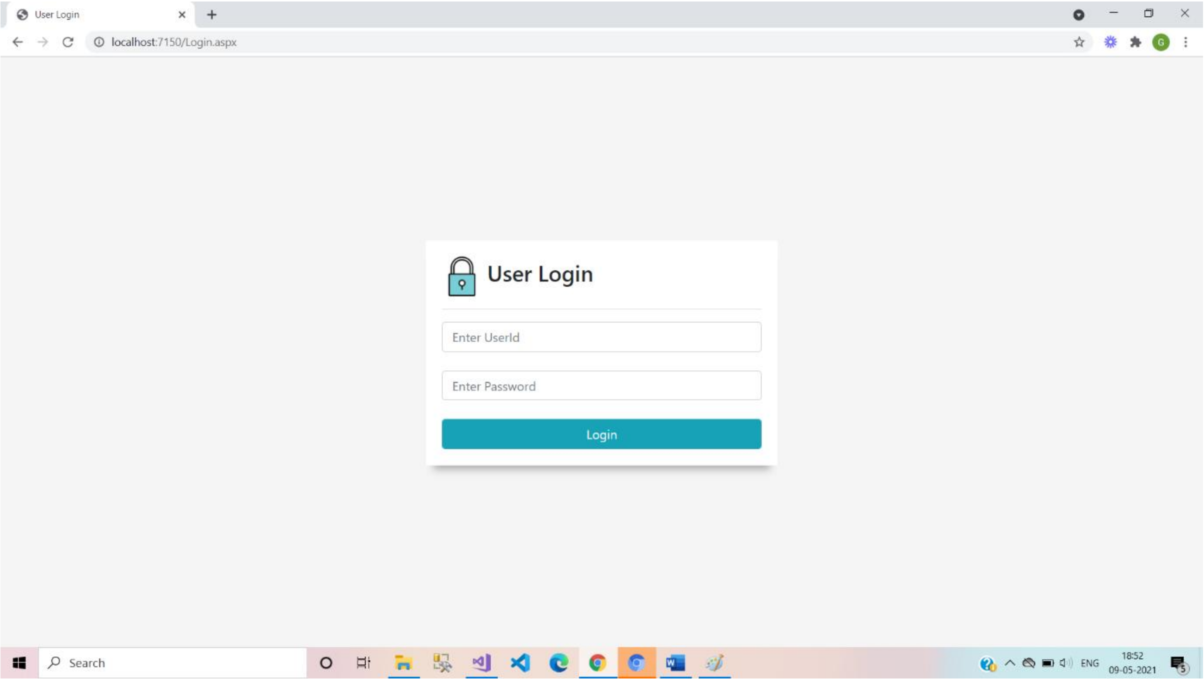


Computer Name	Transfer Date	Mobile No	Image
LAPTOP-GF2I5IK8	01-07-1894 00:00:00	11331332	
LAPTOP-GF2I5IK8	01-07-1894 00:00:00	121233131	



Transfer Data
localhost:7150/DashBoard.aspx
Logout
Zoomed Image
LEADTOOLS Virtual Printer Evaluation Version
Search
ENG
18:50
09-05-2021

Computer Name	Transfer Date	Mobile No	Image
LAPTOP-GF2I5IK8	01-07-1894 00:00:00	11331332	
LAPTOP-GF2I5IK8	01-07-1894 00:00:00	121233131	



Front end Code :-

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="DashBoard.aspx.cs"
Inherits="LoginApp.DashBoard" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title> Transfer Data</title>
  <style type="text/css">
    body
    {
      font-family: Arial;
      font-size: 10pt;
    }
    table
    {
      border: 1px solid #ccc;
      border-collapse: collapse;
    }
    table th
    {
      background-color: #F7F7F7;
      color: #333;
      font-weight: bold;
    }
    table th, table td
    {
      padding: 5px;
      border: 1px solid #ccc;
    }
    table img
    {
      height: 150px;
```

```
            width: 150px;
            cursor: pointer;
        }
        #dialog img
        {
            height: 550px;
            width: 575px;
            cursor: pointer;
        }
      /*div#main1
        {
            width:330px;
            margin-right: auto;
            margin-left: auto;
            text-align: center;
        }*/
    </style>

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" >

</head>
<body>

     <form id="form1" runat="server">
    <div class="col-lg-8">

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
         &nbsp;&nbsp;&nbsp;


<asp:Button ID="BtnLogout" runat="server" Text="Logout" CssClass="btn btn-primary"
OnClick="btnLogout_Click" />
        </div>
  <div>
      <br />
      <asp:GridView ID="gvImages" runat="server" AutoGenerateColumns="false"
OnRowDataBound="OnRowDataBound" style="border-collapse:collapse;margin-left: 15px;" >
        <Columns>
            <asp:BoundField DataField="ComputerName" HeaderText="Computer Name" />
            <asp:BoundField DataField="DateTime" HeaderText="Transfer Date" />
             <asp:BoundField DataField="MobNo" HeaderText="Mobile No" />
            <asp:TemplateField HeaderText="Image">
                <ItemTemplate>
                    <asp:Image ID="Image1" runat="server" />
                </ItemTemplate>
            </asp:TemplateField>
        </Columns>
    </asp:GridView>
           &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
           </div>
         <br />
        <asp:Label runat="server" ID="txtInfo" />
    <div id="dialog" style="display: none">

    </div>
        <script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js"></script>
```

```
    <link rel="stylesheet" href="https://ajax.googleapis.com/ajax/libs/jqueryui/1.8.24/themes/start/jquery-ui.css" />
    <script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/jqueryui/1.8.24/jquery-ui.min.js"></script>
    <script type="text/javascript">
        $(function () {
            $("#dialog").dialog({
                autoOpen: false,
                modal: true,
                height: 600,
                width: 600,
                title: "Zoomed Image"
            });
            $("[id*=gvImages] img").click(function () {
                $('#dialog').html('');
                $('#dialog').append($(this).clone());
                $('#dialog').dialog('open');
            });
        });
    </script>
    </form>
</body>
</html>
```

Backend Code:

```
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;

namespace LoginApp
{
    public partial class DashBoard : System.Web.UI.Page
    {
        static string connectionString = @"Data Source=LAPTOP-GF2I5IK8;Initial Catalog=VP;Integrated Security=True";
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                SqlConnection con = new SqlConnection(connectionString);
                con.Open();
                SqlCommand cmd = new SqlCommand("Select * from PrinterData ", con);
                SqlDataAdapter da = new SqlDataAdapter(cmd);
                DataTable dt = new DataTable();
                da.Fill(dt);
                gvImages.DataSource = dt;
                gvImages.DataBind();
                //if (dt.Rows.Count > 0)
                //{
```

```
                //    txtInfo.Text = "Login Successful!";


                //    con.Close();


                //}
                //else
                //{
                //    txtInfo.Text = "Invalid credentials";
                //}

                con.Close();

            }
            catch (Exception ex)
            {

            }
        }

        protected void OnRowDataBound(object sender, GridViewRowEventArgs e)
        {
            if (e.Row.RowType == DataControlRowType.DataRow)
            {
                DataRowView dr = (DataRowView)e.Row.DataItem;
                string imageUrl = "data:image/jpg;base64," + Convert.ToBase64String((byte[])dr["Image"]);
                (e.Row.FindControl("Image1") as Image).ImageUrl = imageUrl;
            }
        }

        protected void btnLogout_Click(object sender, EventArgs e)
        {
            Response.Redirect("~/Login.aspx");
        }
    }
}
```

Chapter 6:

Future Scope and Conclusion

6.1. Future scope

- Saving a document to another format such as a pdf or multi-page tiff file.
- Sending documents to a fax server.
- Allowing user to control certain aspects of printing not supported natively, such as printing multiple pages per sheet without border, print letterhead, watermarks etc. This output can either be saved in a file for future printing or passed to another printer.
- Previewing a printed document before printing it, to save ink and paper. This functionality is also built into many GUI applications.
- Allowing remote printing of documents over the Internet. At least one example of this technology creates a virtual printer on one computer which actually converts the document and sends it to a remote server, from which the file can be printed to a printer attached to a PC in a remote location. Similar technology is also being used to allow printing from devices such as smart phone.

6.2. Conclusion

The project is to objective of virtual printer is to share securely encrypted file over server using virtual printer driver and we can access that data or file from anywhere.