Add a button to your webform like this:

```
<asp:Button runat="server" ID="PopupButton" OnClientClick="window.open('PdfExample.aspx')"
Text="Printable Form" />
```

This will open a new window on your browser and display another WebForm in it (PdfExample.aspx in this case)

Download this ZipFile - > http://sourceforge.net/projects/itextsharp/files/latest/download

Unzip it and add a reference to the iTextSharp.dll in your Visual Studio Solution (Website -> Add reference)

Add a WebForm to your Project, name it better than PdfExample, perhaps use the original form name with PDF added to the end.

Add the following Imports:

```
Imports System
Imports System.Collections
Imports System.ComponentModel
Imports System.Data
Imports System.Drawing
Imports System.Web
Imports System.Web.SessionState
Imports System.Web.UI
Imports System.Web.UI.WebControls
Imports System.Web.UI.HtmlControls
Imports iTextSharp.text
Imports iTextSharp.text.pdf
Imports System.IO
```

Add the following Page_Load:

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
' step 1
' need to write to memory first due to IE wanting
' to know the length of the pdf beforehand
Dim m As MemoryStream = New MemoryStream()
Dim document As iTextSharp.text.Document = New Document()
    Try
        ' step 2: we set the ContentType and create an instance of the Writer
        Response.ContentType = "application/pdf"
        Dim writer As PdfWriter = PdfWriter.GetInstance(document, m)
        writer.CloseStream = False

        ' step 3
        document.Open()

        ' step 4
        PopulatePDF(document)

        Catch ex As DocumentException
                Console.Error.WriteLine(ex.StackTrace)
                Console.Error.WriteLine(ex.Message)
        End Try
        ' step 5: Close document
```

```
        document.Close()

        ' step 6: Write pdf bytes to outputstream
        Response.OutputStream.Write(m.GetBuffer(), 0, m.GetBuffer().Length)
        Response.OutputStream.Flush()
        Response.OutputStream.Close()
        m.Close()

End Sub
```

This code creates a PDF file in memory on the server and lets the client open it, all without creating a physical file on disk, it only lives in memory. This saves you all sorts of hassles with physical filenames and multiple clients overwriting each other. Plus this code will be the exact same for any printable form you wish to build. I call that boilerplate code.

The actual creation of the PDF file itself is done in PopulatePDF(), creating a PDF is much like building HTML tables and iTextSharp has complete syntax for making pretty much anything you want. Here's a little example I stole from their tutorials.

```
Protected Sub PopulatePDF(ByRef Doc As iTextSharp.text.Document)
    Doc.Add(New Paragraph(DateTime.Now.ToString()))
    Dim table As iTextSharp.text.Table = New iTextSharp.text.Table(3)
    table.BorderWidth = 1
    table.BorderColor = New iTextSharp.text.Color(0, 0, 255)
    table.Border = iTextSharp.text.Rectangle.TOP_BORDER Or
    iTextSharp.text.Rectangle.BOTTOM_BORDER
    table.Padding = 5
    table.Spacing = 5
    Dim cell As Cell = New Cell("header")
    cell.Header = True
    cell.BorderWidth = 3
    cell.Border = iTextSharp.text.Rectangle.TOP_BORDER Or
    iTextSharp.text.Rectangle.BOTTOM_BORDER
    cell.Colspan = 3
    table.addCell(cell)
    cell = New Cell("example cell with colspan 1 and rowspan 2")
    cell.Rowspan = 2
    cell.BorderColor = New iTextSharp.text.Color(255, 0, 0)
    cell.Border = iTextSharp.text.Rectangle.LEFT_BORDER Or
    iTextSharp.text.Rectangle.BOTTOM_BORDER
    table.addCell(cell)
    table.addCell("1.1")
    table.addCell("2.1")
    table.addCell("1.2")
    table.addCell("2.2")
    table.AddCell("cell test1")
    cell = New Cell("big cell")
    cell.Rowspan = 2
    cell.Colspan = 2
    cell.Border = iTextSharp.text.Rectangle.NO_BORDER
    cell.GrayFill = 0.9F
    table.addCell(cell)
    table.addCell("cell test2")
    Doc.Add(table)
```

```
End Sub
```

They have a support site here: http://itextpdf.com/

They are trying to earn some cash selling books/pdf's so the nice tutorials are buried behind paywalls now.

A little creative googling will help find gems.

http://www.mikesdotnetting.com/category/20/itextsharp