

فرض کنید که می‌خواهید برای کاربری پس از ثبت اطلاعاتش در سایت، کارتی به فرمت PDF صادر کنید تا آن را دریافت و سپس چاپ کند. حتماً از این دست موارد زیاد مشاهده کرده اید؛ مانند دریافت کارت ورود به جلسات امتحانی، کارت ورود به همایش‌ها و کنسرت‌های موسیقی و ...

برای تهیه فایل PDF، به غیر از کتابخانه‌های گزارش‌گیری تجاری، می‌توان از کتابخانه‌ی [iTextSharp](#) که گزینه‌ای سوری باز، با کیفیت و محبوب است، استفاده کرد. متأسفانه این کتابخانه دارای محیط گرافیکی طراحی گزارش نیست و کار با آن فقط از طریق کدنویسی میسر است که صد البته انعطاف‌پذیری و پویایی قابل توجهی را برای تهیه‌ی گزارش نسبت به ابزارهای طراحی گرافیکی، در اختیار برنامه‌نویس قرار می‌دهد. البته می‌توان از برنامه‌ی Open Office برای [طراحی قالب گزارش](#) نیز استفاده کرد، اما من پس از استفاده، به کیفیت و انعطاف‌پذیری و امکانات مورد نظر نتوانستم دست یابم و تصمیم گرفتم برای تهیه‌ی کارت، مستقیماً با iTextSharp کد نویسی انجام دهم.

در این مقاله به نحوه‌ی تهیه یک کارت با فرمت PDF با استفاده از کتابخانه iTextSharp خواهیم پرداخت که این کتابخانه به فناوری خاصی گره نخورده است و در تمامی برنامه‌های ASP.NET، WPF، Windows Form و در کل هر کجا که دات‌نت فریم‌ورک در دسترس است، قابل استفاده می‌باشد.

فرض کنید کارتی به شکل زیر می‌خواهیم بسازیم (تمامی تصاویر از سطح اینترنت جمع‌آوری شده‌اند):



طبق این عکس، طرح بندی کارت مورد نظر به این شکل است که در بالای کارت باید آرم سازمان، عنوان در وسط و عکس فرد در سمت چپ باشد و در ادامه نام و مشخصات فرد درج شده باشند. همچنین عکسی به صورت شفاف، تمام پس زمینه کارت را پوشانده است.

فرقی نمی‌کند که تکنولوژی مورد استفاده شما چیست، برای سادگی کار این مثال را با یک Console Application آغاز کنید. برای نصب iTextSharp نیز فرمان زیر را در کنسول NuGet وارد کنید:

```
Install-Package iTextSharp
```

شروع کار با iTextSharp

معمولا برای کار با iText یک سری روال تکراری از قبیل انتخاب نام فایل نهایی، [تعریف فونت](#)، سایز کاغذ، حاشیه بندی و ... را باید طی کنید که کدهای آن را در ذیل مشاهده می‌کنید:

```
var fileStream = new FileStream("card.pdf", FileMode.Create, FileAccess.Write, FileShare.None);
var docFont = GetFont();
var pageSize = PageSize.A6.Rotate(); // سایز کارت را اینجا باید مشخص کرد
var doc = new Document(pageSize);
doc.SetMargins(18f, 18f, 15f, 2f);
var pdfWriter = PdfWriter.GetInstance(doc, fileStream);
doc.Open();
```

- در اینجا سایز کارت، بر روی کاغذ A6 در حالت افقی قرار داده شده است. بدیهی است که مطابق نیاز خودتان می‌توانید این سایز را تغییر دهید.
- تابع GetFont یک تابع کمکی است که در سورس نهایی ارائه شده است و [نکته تعریف فونت در iTextSharp](#) در آن رعایت شده است.
- بقیه موارد نیز جزء الزامات کار با این کتابخانه است.

برای درج عکس به صورت شفاف در پس زمینه کارت باید از کد زیر استفاده کرد:

```
// درج لوگوی مسابقات به صورت شفاف در پس زمینه
var canvas = pdfWriter.DirectContentUnder;
var logoImg = Image.GetInstance(competitionImagePath);
logoImg.SetAbsolutePosition(0, 0);
logoImg.ScaleAbsolute(pageSize);
var graphicsState = new PdfGState { FillOpacity = 0.2F };
canvas.SetGState(graphicsState);
canvas.AddImage(logoImg);
```

چیدمان و طرح بندی بندی عناصر در iTextSharp

برای طراحی کارت یا کلا کار طراحی، باید با نحوه‌ی قرار دادن و طرح بندی عناصر مثل تصاویر و نوشته‌ها و ابزارهای مورد نیاز برای این کار، آشنا شوید. خوشبختانه در iText برای این کار ابزارهای خوبی وجود دارد.

حتما با تگ Table در HTML آشنایی دارید. در سال‌های دور، حتی کل صفحه‌ی وب را به وسیله‌ی Table ساختار دهی می‌کردند. در iTextSharp نیز کلاسی به نام PdfPTable در دسترس است که می‌توان از آن به عنوان قالبی برای قرار دادن عناصر، در صفحه استفاده کرد. این Table همانند هر جدولی دارای یک سری سطر و ستون است که می‌توانیم عناصر مورد نظرمان مثل تصویر و نوشته و... را در آن قرار دهیم.

همانطور که از تگ Table در HTML می‌توان برای رسم جدول و قرار دادن عناصر در سطر و ستون‌های آن استفاده کرد، در iText نیز می‌توان از کلاس PdfPTable برای ترسیم جدول و از متد AddCell آن برای افزودن سلول به آن استفاده کرد. کار با کلاس

PdfPTable نیز ساده است. کافی هست به هنگام ساخت نمونه‌ای از آن، در سازنده‌اش تعداد ستون‌های جدول را ذکر کنید. سپس با استفاده از متد AddCell آن، پارامتری از جنس PdfPCell برای آن ارسال کنید تا به جدول، سلول جدیدی اضافه شود و در صورتیکه تعداد سلول‌های جدید، از تعداد ستون‌های تعریف شده بیشتر شود، iText به صورت خودکار سلول‌های اضافی را به ردیف جدیدی منتقل می‌کند. البته برای افزودن سطر و ستون، روش‌های دیگری نیز هست؛ ولی گویا [روش مرجع](#) همین روش است. کلاس PdfPCell که نقش سلول‌های جدول را بازی می‌کند، نیز می‌تواند شامل متن، تصویر و یا حتی یک جدول تودرتو باشد. در ادامه کدهای جدول بالایی کارت شامل لوگوی دانشگاه، عنوان مسابقات و عکس فرد را مشاهده می‌کنید:

```
// جدولی که برای چیدمان عناصر ارم دانشگاه و عنوان و عکس شخص استفاده می‌شود
var topTable = new PdfPTable(3)
{
    WidthPercentage = 100,
    RunDirection = PdfWriter.RUN_DIRECTION_RTL,
    ExtendLastRow = false,
};

var universityLogoImage = Image.GetInstance(universityLogoPath);
universityLogoImage.ScaleAbsolute(70, 100);

topTable.AddCell(new PdfPCell(universityLogoImage)
{
    HorizontalAlignment = Element.ALIGN_LEFT,
    Border = 0
});

topTable.AddCell(new PdfPCell(new Phrase("کارت مسابقات دانشگاه آزاد اسلامی", docFont))
{
    RunDirection = PdfWriter.RUN_DIRECTION_RTL,
    HorizontalAlignment = Element.ALIGN_CENTER,
    Border = 0,
});

var userImage = Image.GetInstance(userModel.ImagePath);
userImage.Border = Rectangle.TOP_BORDER | Rectangle.RIGHT_BORDER | Rectangle.BOTTOM_BORDER
| Rectangle.LEFT_BORDER;
userImage.BorderWidth = 1f;
userImage.BorderColor = new BaseColor(204, 204, 204); // gray color
userImage.ScaleAbsolute(70, 100);

topTable.AddCell(new PdfPCell(userImage)
{
    HorizontalAlignment = 2,
    Border = 0
});

int[] topTableColumnWidth = { 10, 25, 10 };
topTable.SetWidths(topTableColumnWidth);
doc.Add(topTable);
```

- در ابتدا یک جدول سه ستونه تعریف شده است. تعداد ستون‌ها در هنگام نمونه سازی از کلاس PdfPTable، در سازنده‌ی آن ذکر شده است.
- در iText برای کار با تصاویر، باید از کلاس Image و متد GetInstance فراهم شده توسط خود کتابخانه استفاده کرد. سپس این تصویر را باید به عنوان پارامتر به سلول جدول ارسال کرد.
- به کمک متد AddCell، می‌توان به جدول، سلول اضافه کرد و به صورت خودکار سلول‌های جدیدی که از تعداد ستون‌ها بیشتر می‌شوند، به سطر جدید منتقل می‌شوند.
- اگر می‌خواهید در سلولی متنی نمایش دهید، از کلاس Phrase و تعیین صریح فونت آن استفاده کنید؛ چرا که در غیر این صورت ممکن است متون فارسی نمایش داده نشود.
- در انتها هم جدول مورد نظر را باید به شی doc از جنس کلاس Document تعریف شده اضافه کرد.
- بدیهی هست که اطلاعات شخص مثل نام، نام خانوادگی و ... را نیز باید در یک جدول چهار ستونه قرار داد و نکته‌ی خاص اضافه‌تری ندارد.

حال اگر بیاوریم این تکه کدها را کنار هم قرار بدهیم به کدی قابل اجرا خواهیم رسید.

- ابتدا کلاسی را که در برگیرنده‌ی اطلاعات فرد است، تعریف می‌کنیم:

```
public class UserModel
{
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public string StudentNumber { get; set; }
    public string NationalCode { get; set; }
    public string UniversityName { get; set; }
    public string ImagePath { get; set; }
}
```

- سپس کلاس CardReport را که اصل و اساس بحث ما بود، تعریف می‌کنیم.

```
using System;
using System.IO;
using iTextSharp.text;
using iTextSharp.text.pdf;
using Font = iTextSharp.text.Font;
using Image = iTextSharp.text.Image;
using Rectangle = iTextSharp.text.Rectangle;

namespace ITextSharpCardSample
{
    public class CardReport
    {
        public static void Generate(UserModel userModel, string competitionImagePath, string
universityLogoPath)
        {
            var fileStream = new FileStream("card.pdf", FileMode.Create, FileAccess.Write,
FileShare.None);

            var docFont = GetFont();

            var pageSize = PageSize.A6.Rotate(); // سایز کارت را اینجا باید مشخص کرد

            var doc = new Document(pageSize);

            doc.SetMargins(18f, 18f, 15f, 2f);

            var pdfWriter = PdfWriter.GetInstance(doc, fileStream);

            doc.Open();

            // درج لوگوی مسابقات به صورت شفاف در پس زمینه
            var canvas = pdfWriter.DirectContentUnder;
            var logoImg = Image.GetInstance(competitionImagePath);
            logoImg.SetAbsolutePosition(0, 0);
            logoImg.ScaleAbsolute(pageSize);
            var graphicsState = new PdfGState { FillOpacity = 0.2f };
            canvas.SetGState(graphicsState);
            canvas.AddImage(logoImg);

            // جدولی که برای چیدمان عناصر ارم دانشگاه و عنوان و عکس شخص استفاده می‌شود
            var topTable = new PdfPTable(3)
            {
                WidthPercentage = 100,
                RunDirection = PdfWriter.RUN_DIRECTION_RTL,
                ExtendLastRow = false,
            };

            var universityLogoImage = Image.GetInstance(universityLogoPath);

            universityLogoImage.ScaleAbsolute(70, 100);

            topTable.AddCell(new PdfPCell(universityLogoImage)
            {
                HorizontalAlignment = Element.ALIGN_LEFT,
                Border = 0
            });

            topTable.AddCell(new PdfPCell(new Phrase("کارت مسابقات دانشگاه آزاد اسلامی", docFont))
            {
                RunDirection = PdfWriter.RUN_DIRECTION_RTL,
                HorizontalAlignment = Element.ALIGN_CENTER,
```

```

        Border = 0,
    });

    var userImage = Image.GetInstance(userModel.ImagePath);
    userImage.Border = Rectangle.TOP_BORDER | Rectangle.RIGHT_BORDER | Rectangle.BOTTOM_BORDER
| Rectangle.LEFT_BORDER;
    userImage.BorderWidth = 1f;
    userImage.BorderColor = new BaseColor(204, 204, 204); // gray color
    userImage.ScaleAbsolute(70, 100);

    topTable.AddCell(new PdfPCell(userImage)
    {
        HorizontalAlignment = 2,
        Border = 0
    });

    int[] topTableColumnsWidth = { 10, 25, 10 };
    topTable.SetWidths(topTableColumnsWidth);
    doc.Add(topTable);

    // جدول مشخصات شرکت کننده مثل نام و نام خانوادگی
    var infoTable = new PdfPTable(4)
    {
        WidthPercentage = 100,
        RunDirection = PdfWriter.RUN_DIRECTION_RTL,
        ExtendLastRow = false,
        SpacingBefore = 15,
    };

    infoTable.AddCell(new PdfPCell(new Phrase("نام:", docFont))
    {
        RunDirection = PdfWriter.RUN_DIRECTION_RTL,
        HorizontalAlignment = Element.ALIGN_LEFT,
        Border = 0,
        PaddingBottom = 15
    });

    infoTable.AddCell(new PdfPCell(new Phrase(userModel.FirstName, docFont))
    {
        RunDirection = PdfWriter.RUN_DIRECTION_RTL,
        HorizontalAlignment = Element.ALIGN_LEFT,
        Border = 0
    });

    infoTable.AddCell(new PdfPCell(new Phrase("نام خانوادگی:", docFont))
    {
        RunDirection = PdfWriter.RUN_DIRECTION_RTL,
        HorizontalAlignment = Element.ALIGN_LEFT,
        Border = 0
    });

    infoTable.AddCell(new PdfPCell(new Phrase(userModel.LastName, docFont))
    {
        RunDirection = PdfWriter.RUN_DIRECTION_RTL,
        HorizontalAlignment = Element.ALIGN_LEFT,
        Border = 0
    });

    infoTable.AddCell(new PdfPCell(new Phrase("دانشجویی\ن شماره:", docFont))
    {
        RunDirection = PdfWriter.RUN_DIRECTION_RTL,
        HorizontalAlignment = Element.ALIGN_LEFT,
        Border = 0,
        PaddingBottom = 15
    });

    infoTable.AddCell(new PdfPCell(new Phrase(userModel.StudentNumber, docFont))
    {
        RunDirection = PdfWriter.RUN_DIRECTION_RTL,
        HorizontalAlignment = Element.ALIGN_LEFT,
        Border = 0
    });

    infoTable.AddCell(new PdfPCell(new Phrase("کد ملی:", docFont))
    {
        RunDirection = PdfWriter.RUN_DIRECTION_RTL,
        HorizontalAlignment = Element.ALIGN_LEFT,
        Border = 0
    });

```

```

    });

    infoTable.AddCell(new PdfPCell(new Phrase(userModel.NationalCode, docFont))
    {
        RunDirection = PdfWriter.RUN_DIRECTION_RTL,
        HorizontalAlignment = Element.ALIGN_LEFT,
        Border = 0
    });

    infoTable.AddCell(new PdfPCell(new Phrase("واحد دانشگاهی", docFont))
    {
        RunDirection = PdfWriter.RUN_DIRECTION_RTL,
        HorizontalAlignment = Element.ALIGN_LEFT,
        Border = 0
    });

    infoTable.AddCell(new PdfPCell(new Phrase(userModel.UniversityName, docFont))
    {
        RunDirection = PdfWriter.RUN_DIRECTION_RTL,
        HorizontalAlignment = Element.ALIGN_LEFT,
        Border = 0
    });

    // دو سلول بعدی صرفاً جهت تکمیل شدن یک ردیف است تا عملکرد صحیح خود را داشته باشد
    infoTable.AddCell(new PdfPCell(new Phrase("", docFont))
    {
        RunDirection = PdfWriter.RUN_DIRECTION_RTL,
        HorizontalAlignment = Element.ALIGN_LEFT,
        Border = 0
    });

    infoTable.AddCell(new PdfPCell(new Phrase("", docFont))
    {
        RunDirection = PdfWriter.RUN_DIRECTION_RTL,
        HorizontalAlignment = Element.ALIGN_LEFT,
        Border = 0
    });

    int[] infoTableColumnsWidth = { 20, 15, 20, 15 };
    infoTable.SetWidths(infoTableColumnsWidth);
    doc.Add(infoTable);
    doc.Close();
}

private static Font GetFont()
{
    const string fontName = "Iranian Sans";

    if (FontFactory.IsRegistered(fontName))
        return FontFactory.GetFont(fontName, BaseFont.IDENTITY_H, BaseFont.EMBEDDED);

    var fontPath = "Fonts/irsans.ttf"; // مسیر فونت
    FontFactory.Register(fontPath);

    return FontFactory.GetFont(fontName, BaseFont.IDENTITY_H, BaseFont.EMBEDDED);
}
}
}

```

نکته : حتماً به تعریف فونت در پوشه‌ی Fonts و عکس‌ها در پوشه Images توجه فرمایید.

و در انتها نحوه‌ی استفاده از کلاس CardReport در یک برنامه‌ی Console:

```

class Program
{
    static void Main(string[] args)
    {
        var userModel = new UserModel
        {
            FirstName = "علی",

```

```
        LastName = "احمدی",  
        NationalCode = "1234567890",  
        StudentNumber = "23242342",  
        UniversityName = "آزاد",  
        ImagePath = "Images/avatar.jpg"  
    };  
  
    CardReport.Generate(userModel, "Images/competition_logo.jpg",  
        "Images/university_logo.png");  
  
    System.Diagnostics.Process.Start("card.pdf");  
    }  
}
```

[دریافت نمونه کد در ویژوال استادیو 2015](#)