源代码参考：

```
    /*   PE initialization code after reset */
    /* Initialization of the registers INITRG, INITRM, INITEE is done to protect them to be written accidentally later by the application */
    /*lint -save   -e950 -e923 Disable MISRA rule (1.1,11.3) checking. */
    *(byte*)INITRG_ADR = 0x00U;                /* Set the register map position */
    asm("nop");                                /* nop instruction */
    /*lint -restore Enable MISRA rule (1.1,11.3) checking. */
    /*  INITRM:  RAM15=0,RAM14=0,RAM13=0,RAM12=0,RAM11=1,??=0,??=0,RAMHAL=1 */
    setReg8(INITRM, 0x09U);                    /* Set the RAM map position */
    /* INITEE: EE15=0,EE14=0,EE13=0,EE12=0,??=0,??=0,??=0,EEON=1 */
    setReg8(INITEE, 0x01U);                    /* Set the EEPROM map position */
    /* MISC: ??=0,??=0,??=0,??=0,EXSTR1=1,EXSTR0=1,ROMHM=0,ROMON=1 */
    setReg8(MISC, 0x0DU);
    /* PEAR: NOACCE=0,??=0,PIPOE=0,NECLK=0,LSTRE=0,RDWE=0,??=0,??=0 */
    setReg8(PEAR, 0x00U);
    /*   System clock initialization */
    /* CLKSEL: PLLSEL=0,PSTP=0,SYSWAI=0,ROAWAI=0,PLLWAI=0,CWAI=0,RTIWAI=0,COPWAI=0 */
    setReg8(CLKSEL, 0x00U);                    /* Select clock source from XTAL and set bits in CLKSEL reg. */
    /* PLLCTL: CME=1,PLLON=0,AUTO=1,ACQ=1,??=0,PRE=0,PCE=0,SCME=1 */
    setReg8(PLLCTL, 0xB1U);                    /* Disable the PLL */
    /*** End of PE initialization code after reset ***/
    /*lint -save   -e950 Disable MISRA rule (1.1) checking. */
    __asm("jmp _Startup");                     /* Jump to C startup code */
    /*lint -restore Enable MISRA rule (1.1) checking. */
}
#pragma CODE_SEG DEFAULT
void PE_low_level_init(void)
{
    /* Int. priority initialization */
    /* HPRIO: PSEL7=1,PSEL6=0,PSEL5=0,PSEL4=1,PSEL3=0,PSEL2=0,PSEL1=1,??=0 */
    setReg8(HPRIO, 0x92U);                     /* Set the highest interrupt priority to the ivVcan4rx interrupt */
```

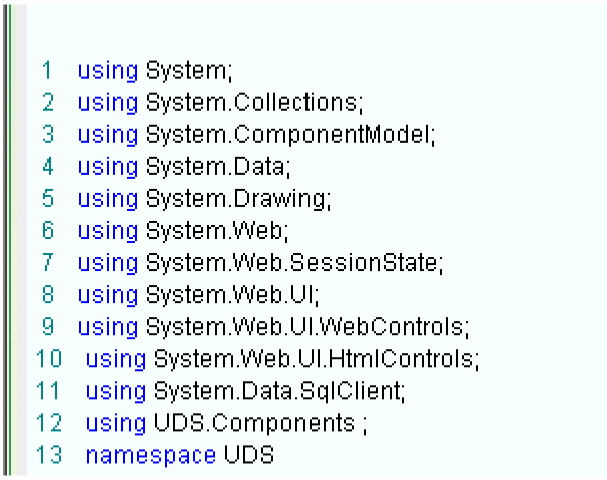
源代码提供要求：

源代码排版在 A4 纸上，每页不低于 50 行（页中不得有空行）。若全部源代码不超过 60 页，需要全部顺序排版在 A4 WORD 文档里；若超过 60 页，只需提供前、后各连续的 30 页，并且保证提供的源代码最后一页满页、是模块结束页；

代码的要求

1.

代码要求是提供原始的代码 不是关键代码 语法上要求完整
例如 C++代码应该是 include 之类开头的 而不是直接一开始就是函数
C#代码应该是 using 之类开头的 而不是直接一开始就是函数
例如 以下为完整的 C#代码开头



```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;
using UDS.Components ;
namespace UDS
```

要求提供的是对应的代码文件的原始文本

2.

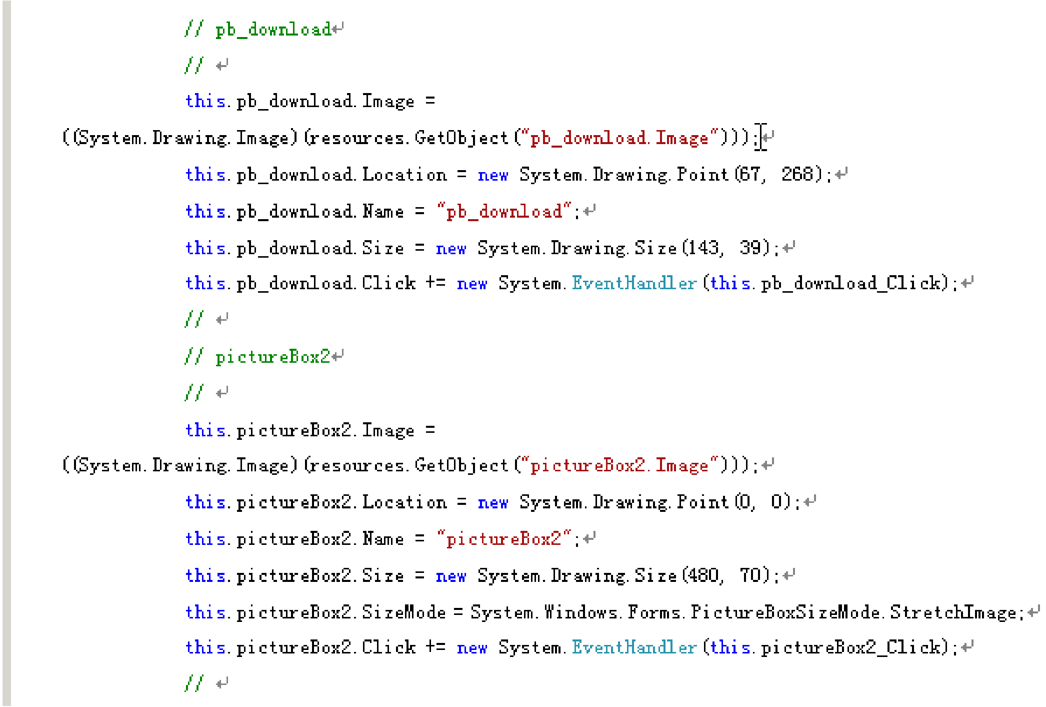
第一页应该是以下一种情况所在的页面的原始代码

1. 主函数
2. 程序的入口 比如登录函数
3. 主页 比如 index default 页面

3.

尽量少提供或者不提供设计器生成的代码
以 C#语言为例 设计器生成的代码语言文件 一般为 *XXXt.designer.cs*：

内容比如



```
            // pb_download
            //
            this.pb_download.Image =
((System.Drawing.Image)(resources.GetObject("pb_download.Image")));
            this.pb_download.Location = new System.Drawing.Point(67, 268);
            this.pb_download.Name = "pb_download";
            this.pb_download.Size = new System.Drawing.Size(143, 39);
            this.pb_download.Click += new System.EventHandler(this.pb_download_Click);
            //
            // pictureBox2
            //
            this.pictureBox2.Image =
((System.Drawing.Image)(resources.GetObject("pictureBox2.Image")));
            this.pictureBox2.Location = new System.Drawing.Point(0, 0);
            this.pictureBox2.Name = "pictureBox2";
            this.pictureBox2.Size = new System.Drawing.Size(480, 70);
            this.pictureBox2.SizeMode = System.Windows.Forms.PictureBoxSizeMode.StretchImage;
            this.pictureBox2.Click += new System.EventHandler(this.pictureBox2_Click);
            //
```

4.

代码量按前、后各连续 30 页，共 60 页，（不足 60 页全部提交）第 60 页为模块结束页，每页不少于 50 行（结束页除外）