

**Southwest University**

《C语言综合课程设计》

**题 目：**  基于CGI的云计算IaaS平台

**学 院：** 计算机与信息科学学院 软件学院

**级、专业：** 2017 **级** 计算机科学与技术 **专业** 2 **班**

**学生学号：** 222017321102054

**学生姓名：**  陈南西

**指导教师：**  胡小方

**提交日期：**  2018 **年** 7 **月** 21 **日**

目录

[**一、问题描述** 3](#_Toc519934896)

[**二、功能需求分析** 3](#_Toc519934897)

[**三、总体设计** 3](#_Toc519934898)

[(1). 本系统所采用的主要组件 3](#_Toc519934899)

[(2). 程序目录和文件与其对应功能 3](#_Toc519934900)

[(3). 系统工作简单流程 3](#_Toc519934901)

[**四、详细设计** 4](#_Toc519934902)

[（1）访问首页流程图 4](#_Toc519934903)

[（2）开通Web站点流程图 5](#_Toc519934904)

[（3）控制面板工作流程图 6](#_Toc519934905)

[**五、系统开发** 7](#_Toc519934906)

[（1）所使用的第三方C语言开源库 7](#_Toc519934907)

[（2）程序主要文件 7](#_Toc519934908)

[（3）程序源代码 7](#_Toc519934909)

[Main.c 7](#_Toc519934910)

[Functions.c 16](#_Toc519934911)

[Functions.h 35](#_Toc519934912)

[**六、测试** 39](#_Toc519934913)

[1. 打开首页 39](#_Toc519934914)

[2. 注册用户 39](#_Toc519934915)

[3. 登陆系统 40](#_Toc519934916)

[4. 开通Web站点 40](#_Toc519934917)

[5. 控制面板 41](#_Toc519934918)

[6. 连接FTP 41](#_Toc519934919)

[7. 打开Web站点首页进行访问 42](#_Toc519934920)

[**七、参考文献** 42](#_Toc519934921)

1. **问题描述**

使用C语言编写CGI程序程序与SQL Server数据库进行交互，配合HTML和CSS等前端技术，开发一个IaaS云计算Web服务的开通以及管理系统. 利用该系统可以方便地开通和管理自己的Web站点，还可以启动，停止和删除站点。开通站点的同时还会为每个站点分配一个FTP账户，方便用户对Web站点的内容进行修改。

1. **功能需求分析**

(1).具有注册登陆系统，可对外开放。

(2).可以在线开通Web站点并且分配对应的FTP管理用户

(3).拥有控制面板，可以轻松查看Web站点信息和FTP信息，以及对Web站点进行启动，停止和删除等操作。

(4).开通Web站点的时候自动为其分配一个FTP账户，方便对站点内容进行修改

1. **总体设计**

(1). 本系统所采用的主要组件

系统平台： Windows

数据库系统：SQL Server

HTTP服务器：IIS

脚本语言：Powershell

(2). 程序目录和文件与其对应功能

文件夹Scripts ：里面存放由Powershell所编写的功能脚本，供CGI程序调用

文件夹css：存放网页样式表文件

文件夹js：存放网页JavaScript脚本文件

文件夹font-awesome-4.7.0：网页图标框架

index.cgi：系统核心程序，起中央调度作用，连接该系统其他所有文件，能够

发起新的H TTP请求，或处理所接收的HTTP请求。并且与SQL Server数据库进行交互。

control.html：控制面板的网页模板，供index.cgi读取

index.html：首页的网页模板，供index.cgi读取

info.html：提示信息的网页模板，供index.cgi读取

login.html：登陆界面

new.html：开通Web站点界面

register.html：注册新用户界面

(3). 系统工作简单流程

1.用户访问网站加载默认的index.cgi，根据请求中所带的参数进行不同的处理。

如果没有参数，则会判断是否登陆，没登陆则输出标准的首页。

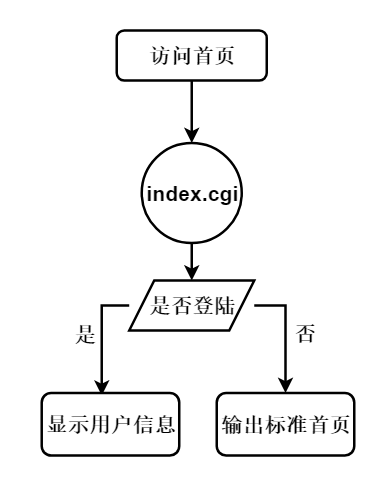
2.用户登陆后，会在服务器生成GUID标识符，该标识符会已Cookie的形式被记录到客户端中，同时还会记录到服务器数据库里面的T\_Session表里面，同时被记录到数据库的还有事件发生时的时间戳，用于以后判断登陆是否过期。

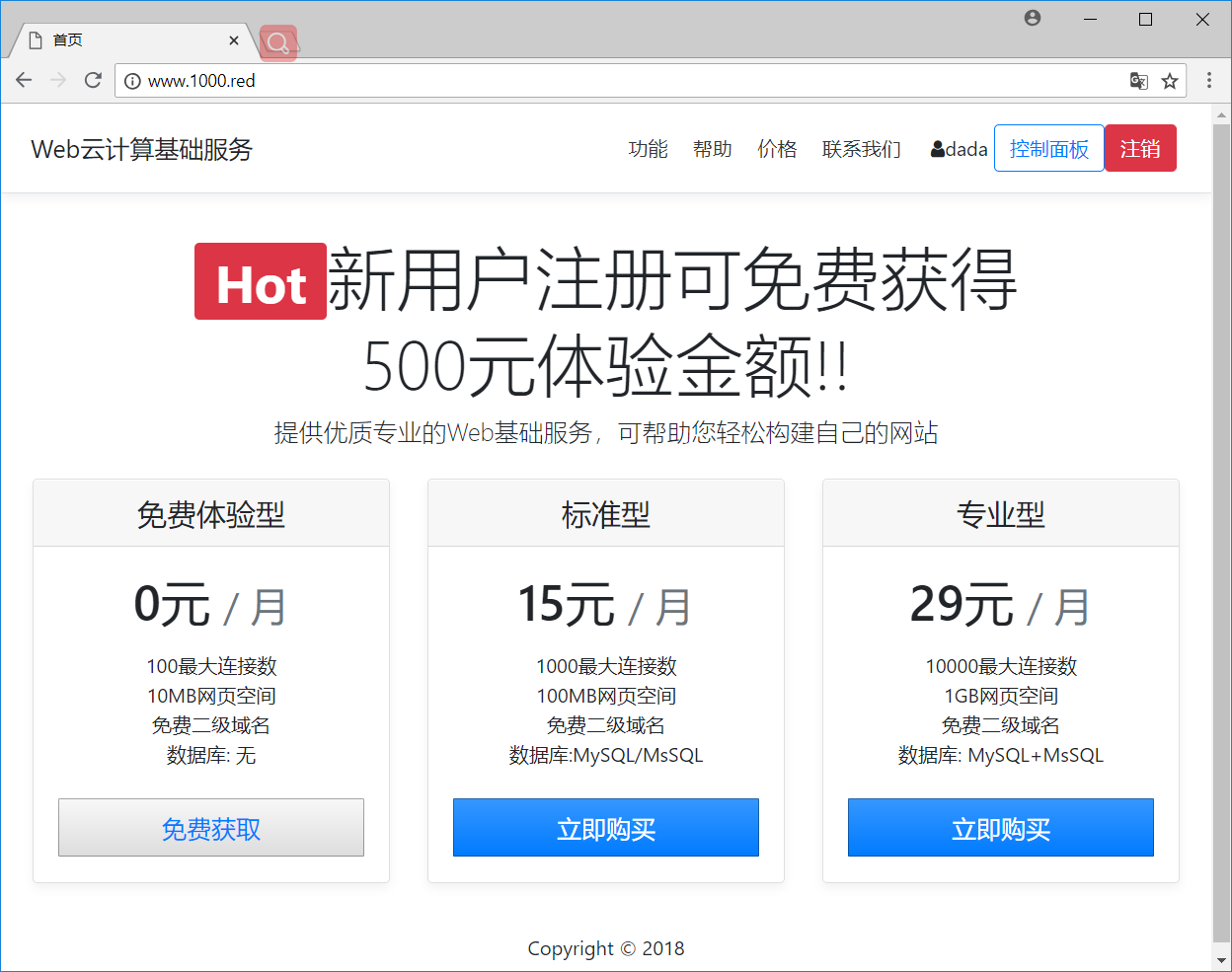
3.进入开通Web站点页面后，填写相关信息，以form表单的形式提交给index.cgi，index.cgi解析参数，然后把参数传递给Script文件夹下的脚本进行执行，根据相关参数在服务器下创建文件夹，新建站点，绑定域名，创建计算机普通用户，绑定FTP。

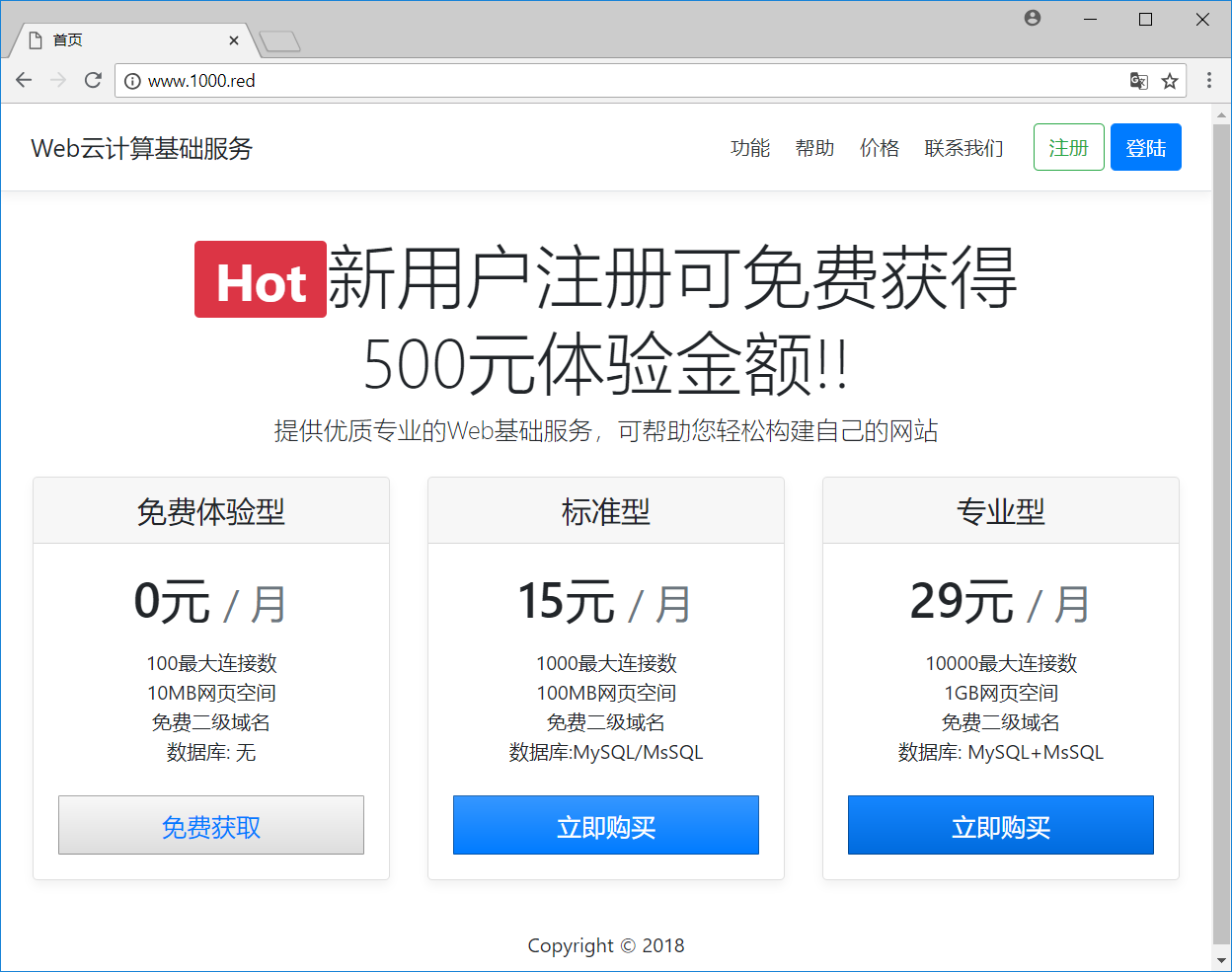
4.在管理面板中可以启动，停止，删除Web站点，其原理也是发送带参数的请求给index.cgi，然后index.cgi调用Script文件夹下的脚本进行相应操作。

1. **详细设计**

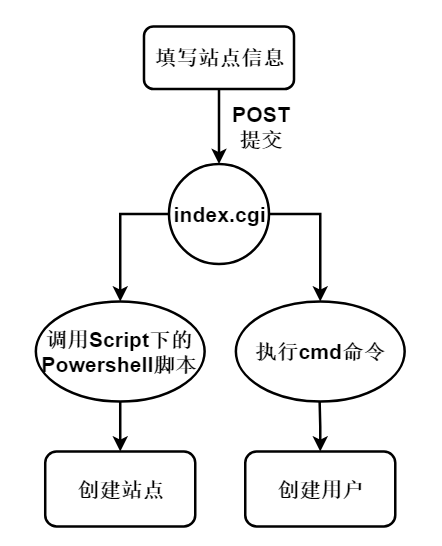
（1）访问首页流程图

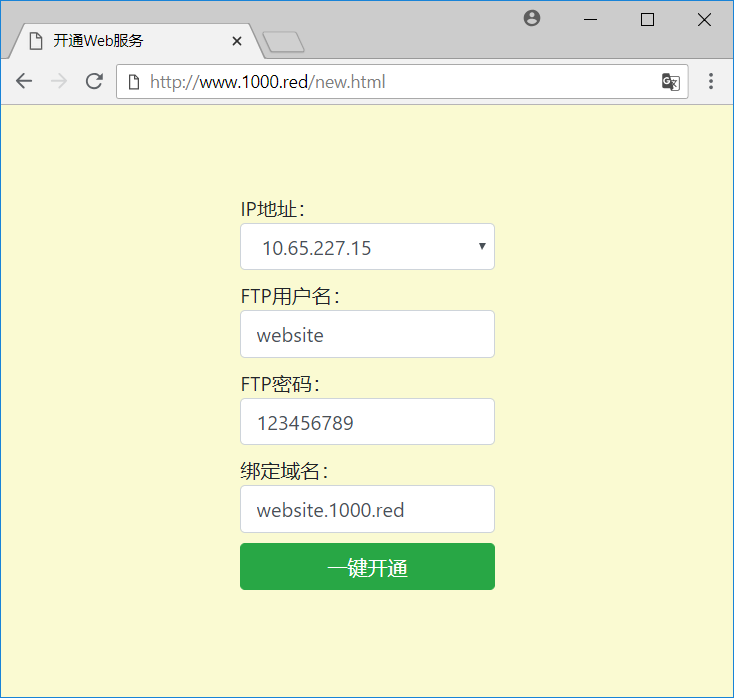




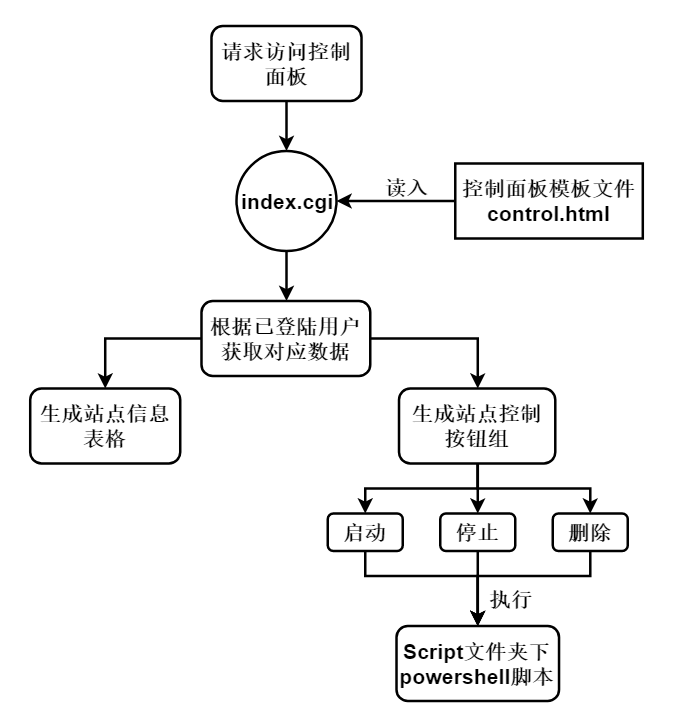


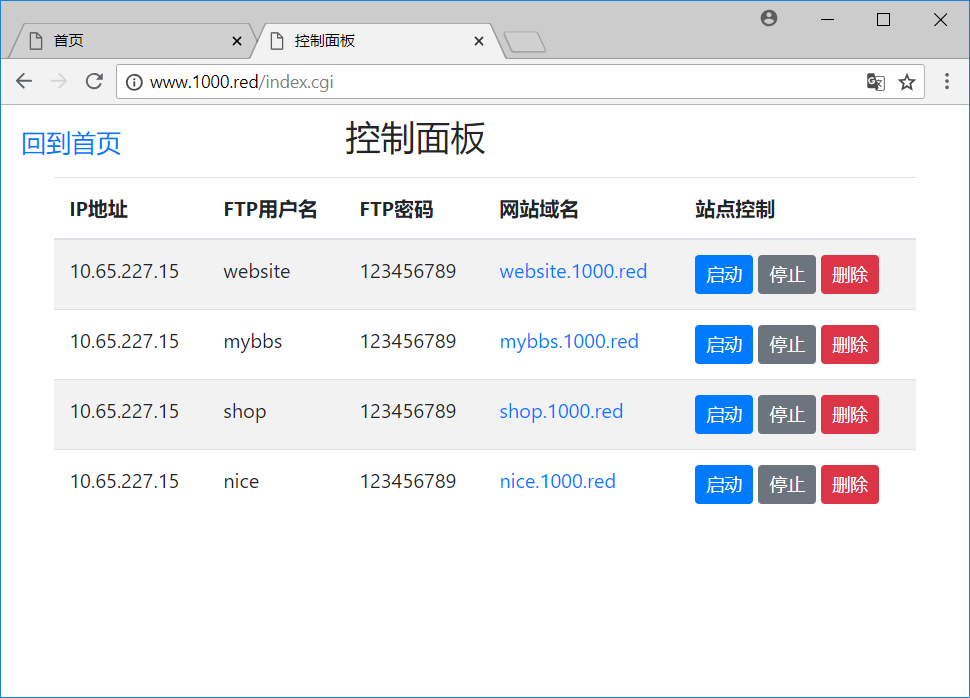
（2）开通Web站点流程图





（3）控制面板工作流程图





1. **系统开发**
2. 所使用的第三方C语言开源库

CGIC：官网https://boutell.com/cgic/

Libctemplate: 官网<http://libctemplate.sourceforge.net/>

1. 程序主要文件

**Main.c 主程序index.cgi的源代码**

**Functions.c 自定义的函数库**

**Functions.h 自定义函数的头文件**

1. 程序源代码

### Main.c

#pragma warning(disable : 4996)

#include <stdio.h>

#include <stdlib.h>

#include <winsock.h>

#include "Functions.h"

#include "cgic.h"

#include <Windows.h>

#include <sql.h>

#include <sqlext.h>

#include <sqltypes.h>

#include <string.h>

#include "ctemplate.h"

*//数据库信息*

#define DataSource "CloudComputeSystem"

#define SQL\_User "sa"

#define SQL\_Password "123456789"

**void** showInfoPage(**char**\* content);

**int** cgiMain() {

*//请求参数*

**char** action[20] = { 0 };    *//标识当前请求的类型，如 登陆，注册 等*

**char** username[20] = { 0 };

**char** password[20] = { 0 };

**char** email[20] = { 0 };

**char** phone[20] = { 0 };

**char** ip[20] = { 0 };

**char** ftpusername[20] = { 0 };

**char** ftppassword[20] = { 0 };

**char** domain[20] = { 0 };

*//获得请求参数*

    cgiFormString("action", &action, 20);

    cgiFormString("username", &username, 20);

    cgiFormString("password", &password, 20);

    cgiFormString("email", &email, 20);

    cgiFormString("phone", &phone, 20);

    cgiFormString("ip", &ip, 20);

    cgiFormString("ftpusername", &ftpusername, 20);

    cgiFormString("ftppassword", &ftppassword, 20);

    cgiFormString("domain", &domain, 20);

*//根据action的请求类型来进行相应操作*

    if (strcmpi(action, "register") == 0) {

        if (SQL\_QueryUser(username) != 0) {

*//printf("您所要注册的账号已经存在，请更换用户名！");*

            cgiHeaderContentType("text/html;charset=gb2312");

            showInfoPage("<h2>您所要注册的账号已经存在！<a href='register.html'>请点击返回重新注册</a></h2> ");

        }

        else

        {

            SQL\_RegisterAccount(username, password, email, phone);

            if (SQL\_QueryUser(username) != 0) {

*//printf("注册成功！");*

                cgiHeaderContentType("text/html;charset=gb2312");

                showInfoPage("<h2>注册成功！<a href='login.html'>点击进行登陆</a></h2> ");

            }

        }

    }

    else if (strcmpi(action, "login") == 0) {

        if (SQL\_QueryUser(username) == 0) {

*//printf("账号不存在");*

            cgiHeaderContentType("text/html;charset=gb2312");

            showInfoPage("<h2>账号不存在！<a href='login.html'>请点击返回重试</a></h2> ");

        }

        else if (SQL\_CheckPassword(username, password) == 0)

        {

*//printf("密码错误");*

            cgiHeaderContentType("text/html;charset=gb2312");

            showInfoPage("<h2>密码错误！<a href='login.html'>请点击返回重试</a></h2> ");

        }

        else {

*////printf("登陆成功");*

**char** g\_username[20] = { 0 },g\_guid[50],g\_timestamp[20];

            CheckLogin(g\_username);

            if (SQL\_CheckGUID2(g\_username,g\_guid,g\_timestamp) != 0) {

                SQL\_DeleteGUID(g\_username);*//依据用户名删除GUID*

            }

*//*

**char** \*guid = createGuid();

            cgiHeaderCookieSetString("guid", guid, 60 \* 60 \* 24, "", "www.1000.red");

            cgiHeaderContentType("text/html;charset=gb2312");

            showInfoPage("<h2>登陆成功！<a href='index.cgi'>请点击返回</a></h2> ");

**char** time[20];

            sprintf(time, "%d", GetTimeStamp());

            SQL\_SetSession(username, guid, time); *//将GUID添加进服务器数据库*

        }

    }

    else if (strcmpi(action, "control") == 0) {

        cgiHeaderContentType("text/html;charset=gb2312");

**char** ip[20], ftpusername[20], ftppassword[20], domain[20];

        TMPL\_varlist \* ftplist = 0;

        TMPL\_varlist \* mainlist = 0;

        TMPL\_loop\* ftploop = 0;

**char** g\_username[20] = { 0 };

        CheckLogin(g\_username); *//得到已经登陆的用户名*

        SQLRETURN ret;

            SQLHENV henv;*//SQLHANDLE henv*

            SQLHDBC hdbc;*//SQLHANDLE hdbc*

            SQLHSTMT hstmt;*//SQLHANDLE hstmt*

            ret = SQLAllocHandle(SQL\_HANDLE\_ENV, NULL, &henv);*//申请环境句柄*

            ret = SQLSetEnvAttr(henv, SQL\_ATTR\_ODBC\_VERSION, (SQLPOINTER)SQL\_OV\_ODBC3, SQL\_IS\_INTEGER);*//设置环境属性*

            ret = SQLAllocHandle(SQL\_HANDLE\_DBC, henv, &hdbc);*//申请数据库连接句柄*

            ret = SQLConnect(hdbc, (SQLCHAR\*)DataSource, SQL\_NTS, (SQLCHAR\*)SQL\_User, SQL\_NTS, (SQLCHAR\*)SQL\_Password, SQL\_NTS);*//连接数据库*

            if (ret == SQL\_SUCCESS || ret == SQL\_SUCCESS\_WITH\_INFO) {

                ret = SQLAllocHandle(SQL\_HANDLE\_STMT, hdbc, &hstmt);*//申请SQL语句句柄*

                SQLCHAR sql[100];

                sprintf(sql, "SELECT \* FROM T\_SiteInfo where username='%s'", g\_username);

                ret = SQLExecDirect(hstmt, sql, SQL\_NTS);*//直接执行SQL语句*

                if (ret == SQL\_SUCCESS || ret == SQL\_SUCCESS\_WITH\_INFO) {

                    SQLCHAR user[20];*//, guid[50],timestamp[20];*

                    SQLINTEGER i\_len,f\_len,u\_len, p\_len, d\_len;

                    while (SQLFetch(hstmt) != SQL\_NO\_DATA) {

                        SQLGetData(hstmt, 2, SQL\_C\_CHAR, ip, 20, &i\_len);

                        SQLGetData(hstmt, 3, SQL\_C\_CHAR, ftpusername, 50, &f\_len);

                        SQLGetData(hstmt, 4, SQL\_C\_CHAR, ftppassword, 20, &p\_len);

                        SQLGetData(hstmt, 5, SQL\_C\_CHAR, domain, 20, &d\_len);

                        ftplist = TMPL\_add\_var(0, "IP", ip, "FTP\_Username", ftpusername, "FTP\_Password", ftppassword, "Domain", domain);

                        ftploop = TMPL\_add\_varlist(ftploop, ftplist);

                    }

                    mainlist = TMPL\_add\_loop(mainlist, "FTPLOOP", ftploop);

**int** number\_row;

                    ret = SQLRowCount(hstmt, &number\_row);

                    if (number\_row == 0) *//没有记录*

                    {

*//return 0;*

                    }

                    else {

*//return 1;*

                    }

                    SQLFreeHandle(SQL\_HANDLE\_STMT, hstmt);*//释放语句句柄*

                }

                else printf("查询数据库操作失败！\n");

                SQLDisconnect(hdbc);*//断开与数据库的连接*

            }

            else printf("连接数据库失败!\n");

            SQLFreeHandle(SQL\_HANDLE\_DBC, hdbc);*//释放连接句柄*

            SQLFreeHandle(SQL\_HANDLE\_ENV, henv);*//释放环境句柄*

        TMPL\_write("control.html", 0, 0, mainlist, cgiOut, cgiOut);

    }

    else if (strcmpi(action, "logout") == 0) {

*//删除服务器对应的GUID*

**char** g\_username[20] = { 0 };

        CheckLogin(g\_username); *//得到已经登陆的用户名*

        SQL\_DeleteGUID(g\_username);*//依据用户名删除GUID*

        cgiHeaderLocation("http://www.1000.red");

    }

    else if (strcmpi(action, "new") == 0) {

**char** g\_username[20] = { 0 };

        if (CheckLogin(g\_username)==0) {

            cgiHeaderContentType("text/html;charset=gb2312");

            showInfoPage("<h2>你好像还没登陆！<a href='login.html'>请点击返回重试</a></h2> ");

        }

        else *//已经是登陆状态*

        {

            if (SQL\_QueryFtpUser(ftpusername)==1) { *//查询对应ftp用户名存在*

                cgiHeaderContentType("text/html;charset=gb2312");

                showInfoPage("<h2>你填写的FTP用户名已经存在，请修改！<a href='new.html'>请点击返回重试</a></h2> ");

            }else{

                if (SQL\_QueryDomain(domain) == 1) {

                    cgiHeaderContentType("text/html;charset=gb2312");

                    showInfoPage("<h2>你填写的域名已经存在，请修改！<a href='new.html'>请点击返回重试</a></h2> ");

                }

                else {

**char** physicalPath[50] = { 0 };

                    sprintf(physicalPath, "D:\\localuser\\%s", ftpusername);

                    Create\_Directory(ftpusername);

                    Create\_Site(ftpusername, physicalPath, ip, "80", domain, "http");

                    SQL\_RecordSite(g\_username, ip, ftpusername, ftppassword, domain);

*//创建计算机账户*

**char** code[100] = { 0 };

                    strcat(code, "net user ");

                    strcat(code, ftpusername);

                    strcat(code, " ");

                    strcat(code, ftppassword);

                    strcat(code, " /add");

                    system(code);

                    cgiHeaderContentType("text/html;charset=gb2312");

                    showInfoPage("<form action=\"index.cgi\" method=\"POST\"><h2>开通完毕！<input type=\"hidden\" name=\"action\" value=\"control\"/><button type=\"submit\">请点击查看</button></h2></form> ");

                }

            }

        }

    }

    else if (strcmpi(action, "start") == 0) {

        Start\_Site(ftpusername);

        cgiHeaderContentType("text/html;charset=gb2312");

        showInfoPage("<form action=\"index.cgi\" method=\"POST\"><h2>启动完毕！<input type=\"hidden\" name=\"action\" value=\"control\"/><button type=\"submit\">点击返回</button></h2></form> ");

    }

    else if (strcmpi(action, "stop") == 0) {

        Stop\_Site(ftpusername);

        cgiHeaderContentType("text/html;charset=gb2312");

        showInfoPage("<form action=\"index.cgi\" method=\"POST\"><h2>停止完毕！<input type=\"hidden\" name=\"action\" value=\"control\"/><button type=\"submit\">点击返回</button></h2></form> ");

    }

    else if (strcmpi(action, "delete") == 0) {

**char** physicalPath[50] = { 0 };

        sprintf(physicalPath, "D:\\localuser\\%s", ftpusername);

        Delete\_Directory(ftpusername);

        Delete\_Site(ftpusername);

        SQL\_DeleteSite(ftpusername);

*//删除计算机账户*

**char** code[100] = { 0 };

        strcat(code, "net user ");

        strcat(code, ftpusername);

        strcat(code, " /delete");

        system(code);

        cgiHeaderContentType("text/html;charset=gb2312");

        showInfoPage("<form action=\"index.cgi\" method=\"POST\"><h2>删除完毕！<input type=\"hidden\" name=\"action\" value=\"control\"/><button type=\"submit\">点击返回</button></h2></form> ");

    }

    else {

**char** g\_username[20] = {0};

        if (CheckLogin(g\_username)==0) { *//表示没有登陆，加载默认首页*

            cgiHeaderContentType("text/html;charset=gb2312");

            TMPL\_varlist \*myList = 0;

            myList = TMPL\_add\_var(myList, "button\_register", "<a class=\"btn btn-outline-success\" href=\"register.html\">注册</a>", 0);

            myList = TMPL\_add\_var(myList, "button\_login", "<a class=\"btn btn-primary\" href=\"login.html\">登陆</a>", 0);

            TMPL\_write("index.html", 0, 0, myList, cgiOut, cgiOut);

        }

        else {

            cgiHeaderContentType("text/html;charset=gb2312");

            TMPL\_varlist \*myList = 0;

**char** str[1024] = { 0 };

            sprintf(str, "<p class=\"mb-2 mt-2\"><span class=\"fa fa-user\"></span>%s</p> &nbsp;<form action=\"index.cgi\" method=\"POST\"><input type=\"hidden\" name=\"action\" value=\"control\"/> <button formtarget=\"\_blank\" type=\"submit\" class=\"btn btn-outline-primary\">控制面板</button></form><form action=\"index.cgi\" method=\"POST\"><input type=\"hidden\" name=\"action\" value=\"logout\"/> <button type=\"submit\" class=\"btn btn-danger\">注销</button></form>",g\_username);

            myList = TMPL\_add\_var(myList, "userinfo", str, 0);

            TMPL\_write("index.html", 0, 0, myList, cgiOut, cgiOut);

        }

    }

    return 0;

}

**void** showInfoPage(**char**\* content) {

    TMPL\_varlist \*myList = 0;

**char** info[1024] = { 0 };

    sprintf(info, "%s", content);

    myList = TMPL\_add\_var(myList, "info", info, 0);

    TMPL\_write("info.html", 0, 0, myList, cgiOut, cgiOut);

}

### Functions.c

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <Windows.h>

#include <sqlext.h>

#include <sql.h>

#include <sqltypes.h>

#include <objbase.h> *//生成GUID所需*

#include "ctemplate.h"

*//数据库信息*

#define DataSource "CloudComputeSystem"

#define SQL\_User "sa"

#define SQL\_Password "123456789"

*//将参数拼接成完整命令*

**void** BuildCode(**char**\* code, **char**\* base, **char**\* script, **char**\* param);

**void** Create\_Directory(**char** \*name) {

**char** base[] = "powershell";

**char** script[] = "./Scripts/New-Item#Directory.ps1";

**char** param[100] = { 0 };

**char** code[1024] = { 0 };

    sprintf(param, "-path D:\\localuser\\%s",name);

    BuildCode(code, base, script, param);

    system(code);*//执行命令*

}

**void** Delete\_Directory(**char** \*name) {

**char** base[] = "powershell";

**char** script[] = "./Scripts/Remove-Item#Directory.ps1";

**char** param[100] = { 0 };

**char** code[1024] = { 0 };

    sprintf(param, "-path D:\\localuser\\%s", name);

    BuildCode(code, base, script, param);

    system(code);*//执行命令*

}

**void** Create\_Site(**char**\* name,**char**\* PhysicalPath, **char**\* ip, **char**\* port, **char**\* hostname, **char**\* protocol) {

**char** base[] = "powershell";

**char** script[] = "./Scripts/New-IISSite.ps1";

**char** param[100] = { 0 };

**char** code[1024] = { 0 };

    sprintf(param, "-Name %s -PhysicalPath %s -BindingInformation '%s:%s:%s' -Protocol %s ", name, PhysicalPath,ip,port,hostname,protocol);

    BuildCode(code, base, script, param);

    system(code);*//执行命令*

}

**void** Delete\_Site(**char**\* name) {

**char** base[] = "powershell";

**char** script[] = "./Scripts/Remove-IISSite.ps1";

**char** param[100] = { 0 };

**char** code[1024] = { 0 };

    sprintf(param, "-Name %s", name);

    BuildCode(code, base, script, param);

    system(code);*//执行命令*

}

**void** Start\_Site(**char**\* name) {

**char** base[] = "powershell";

**char** script[] = "./Scripts/Start-IISSite.ps1";

**char** param[100] = { 0 };

**char** code[1024] = { 0 };

    sprintf(param, "-Name %s",name);

    BuildCode(code, base, script, param);

    system(code);*//执行命令*

}

**void** Stop\_Site(**char**\* name) {

**char** base[] = "powershell";

**char** script[] = "./Scripts/Stop-IISSite.ps1";

**char** param[100] = { 0 };

**char** code[1024] = { 0 };

    sprintf(param, "-Name %s", name);

    BuildCode(code, base, script, param);

    system(code);*//执行命令*

}

*//将参数拼接成完整命令*

**void** BuildCode(**char**\* code, **char**\* base, **char**\* script, **char**\* param){

    strcat(code,base);

    strcat(code, " ");

    strcat(code, script);

    strcat(code, " ");

    strcat(code, param);

}

*//依靠GUID进行检查是否登陆，并且可以给g\_username赋值,如果已经成功登陆且在期限内返回1，没有返回0*

**int** CheckLogin(**char** \*g\_username) {

*//取得本地Cookie*

**char** c\_guid[50] = { 0 };

    cgiCookieString("guid", c\_guid, 50);

*//取得服务器GUID 和 TimeStamp*

**char** g\_guid[50] = { 0 }, g\_timestamp[20] = { 0 };

    if (SQL\_CheckGUID(g\_username, c\_guid, g\_timestamp) == 0) { *//等于0代表服务器中不存在客户端的GUID*

        return 0;

    }

    else

    {

        if (GetTimeStamp() - atoi(g\_timestamp) >= 60 \* 60 \* 24) {

            return 0; *//超过24小时*

        }

        else

        {

            return 1;

        }

    }

}

*//查询用户是否存在，若存在返回1，不存在返回0*

**int** SQL\_QueryUser(**char**\* username)

{

    SQLRETURN ret;

    SQLHENV henv;*//SQLHANDLE henv*

    SQLHDBC hdbc;*//SQLHANDLE hdbc*

    SQLHSTMT hstmt;*//SQLHANDLE hstmt*

    ret = SQLAllocHandle(SQL\_HANDLE\_ENV, NULL, &henv);*//申请环境句柄*

    ret = SQLSetEnvAttr(henv, SQL\_ATTR\_ODBC\_VERSION, (SQLPOINTER)SQL\_OV\_ODBC3, SQL\_IS\_INTEGER);*//设置环境属性*

    ret = SQLAllocHandle(SQL\_HANDLE\_DBC, henv, &hdbc);*//申请数据库连接句柄*

    ret = SQLConnect(hdbc, (SQLCHAR\*)"CloudComputeSystem", SQL\_NTS, (SQLCHAR\*)"sa", SQL\_NTS, (SQLCHAR\*)"123456789", SQL\_NTS);*//连接数据库*

    if (ret == SQL\_SUCCESS || ret == SQL\_SUCCESS\_WITH\_INFO) {

        ret = SQLAllocHandle(SQL\_HANDLE\_STMT, hdbc, &hstmt);*//申请SQL语句句柄*

        SQLCHAR sql[100];

        sprintf(sql, "SELECT \* FROM T\_UserInfo where username='%s'", username);

*//printf("sql is %s\n",sql);*

        ret = SQLExecDirect(hstmt, sql, SQL\_NTS);*//直接执行SQL语句*

        if (ret == SQL\_SUCCESS || ret == SQL\_SUCCESS\_WITH\_INFO) {

            SQLCHAR user[20], pass[20];

            SQLINTEGER u\_len, p\_len;

            while (SQLFetch(hstmt) != SQL\_NO\_DATA) {

                SQLGetData(hstmt, 1, SQL\_C\_CHAR, user, 20, &u\_len);

                SQLGetData(hstmt, 2, SQL\_C\_CHAR, pass, 20, &p\_len);

            }

*//printf("user is %s , pass is %s \n", user,pass);*

**int** number\_row;

            ret = SQLRowCount(hstmt, &number\_row);

            if (number\_row == 0) *//没有记录*

            {

                return 0;

            }

            else {

                return 1;

            }

            SQLFreeHandle(SQL\_HANDLE\_STMT, hstmt);*//释放语句句柄*

        }

        else printf("查询数据库操作失败！\n");

        SQLDisconnect(hdbc);*//断开与数据库的连接*

    }

    else printf("连接数据库失败!\n");

    SQLFreeHandle(SQL\_HANDLE\_DBC, hdbc);*//释放连接句柄*

    SQLFreeHandle(SQL\_HANDLE\_ENV, henv);*//释放环境句柄*

}

*//查询FTP用户是否存在，若存在返回1，不存在返回0*

**int** SQL\_QueryFtpUser(**char**\* ftpusername)

{

    SQLRETURN ret;

    SQLHENV henv;*//SQLHANDLE henv*

    SQLHDBC hdbc;*//SQLHANDLE hdbc*

    SQLHSTMT hstmt;*//SQLHANDLE hstmt*

    ret = SQLAllocHandle(SQL\_HANDLE\_ENV, NULL, &henv);*//申请环境句柄*

    ret = SQLSetEnvAttr(henv, SQL\_ATTR\_ODBC\_VERSION, (SQLPOINTER)SQL\_OV\_ODBC3, SQL\_IS\_INTEGER);*//设置环境属性*

    ret = SQLAllocHandle(SQL\_HANDLE\_DBC, henv, &hdbc);*//申请数据库连接句柄*

    ret = SQLConnect(hdbc, (SQLCHAR\*)"CloudComputeSystem", SQL\_NTS, (SQLCHAR\*)"sa", SQL\_NTS, (SQLCHAR\*)"123456789", SQL\_NTS);*//连接数据库*

    if (ret == SQL\_SUCCESS || ret == SQL\_SUCCESS\_WITH\_INFO) {

        ret = SQLAllocHandle(SQL\_HANDLE\_STMT, hdbc, &hstmt);*//申请SQL语句句柄*

        SQLCHAR sql[100];

        sprintf(sql, "SELECT \* FROM T\_SiteInfo where ftpusername='%s'", ftpusername);

        ret = SQLExecDirect(hstmt, sql, SQL\_NTS);*//直接执行SQL语句*

        if (ret == SQL\_SUCCESS || ret == SQL\_SUCCESS\_WITH\_INFO) {

            SQLCHAR user[20], pass[20];

            SQLINTEGER u\_len, p\_len;

            while (SQLFetch(hstmt) != SQL\_NO\_DATA) {

                SQLGetData(hstmt, 1, SQL\_C\_CHAR, user, 20, &u\_len);

                SQLGetData(hstmt, 2, SQL\_C\_CHAR, pass, 20, &p\_len);

            }

**int** number\_row;

            ret = SQLRowCount(hstmt, &number\_row);

            if (number\_row == 0) *//没有记录*

            {

                return 0;

            }

            else {

                return 1;

            }

            SQLFreeHandle(SQL\_HANDLE\_STMT, hstmt);*//释放语句句柄*

        }

        else printf("查询数据库操作失败！\n");

        SQLDisconnect(hdbc);*//断开与数据库的连接*

    }

    else printf("连接数据库失败!\n");

    SQLFreeHandle(SQL\_HANDLE\_DBC, hdbc);*//释放连接句柄*

    SQLFreeHandle(SQL\_HANDLE\_ENV, henv);*//释放环境句柄*

}

*//查询Domain是否存在，若存在返回1，不存在返回0*

**int** SQL\_QueryDomain(**char**\* domain)

{

    SQLRETURN ret;

    SQLHENV henv;*//SQLHANDLE henv*

    SQLHDBC hdbc;*//SQLHANDLE hdbc*

    SQLHSTMT hstmt;*//SQLHANDLE hstmt*

    ret = SQLAllocHandle(SQL\_HANDLE\_ENV, NULL, &henv);*//申请环境句柄*

    ret = SQLSetEnvAttr(henv, SQL\_ATTR\_ODBC\_VERSION, (SQLPOINTER)SQL\_OV\_ODBC3, SQL\_IS\_INTEGER);*//设置环境属性*

    ret = SQLAllocHandle(SQL\_HANDLE\_DBC, henv, &hdbc);*//申请数据库连接句柄*

    ret = SQLConnect(hdbc, (SQLCHAR\*)"CloudComputeSystem", SQL\_NTS, (SQLCHAR\*)"sa", SQL\_NTS, (SQLCHAR\*)"123456789", SQL\_NTS);*//连接数据库*

*//ret = SQLConnect(hdbc, (SQLCHAR\*)DataSource, SQL\_NTS, (SQLCHAR\*)SQL\_User, SQL\_NTS, (SQLCHAR\*)SQL\_Password, SQL\_NTS);//连接数据库*

    if (ret == SQL\_SUCCESS || ret == SQL\_SUCCESS\_WITH\_INFO) {

        ret = SQLAllocHandle(SQL\_HANDLE\_STMT, hdbc, &hstmt);*//申请SQL语句句柄*

        SQLCHAR sql[100];

        sprintf(sql, "SELECT \* FROM T\_SiteInfo where domain='%s'", domain);

*//printf("sql is %s\n",sql);*

        ret = SQLExecDirect(hstmt, sql, SQL\_NTS);*//直接执行SQL语句*

        if (ret == SQL\_SUCCESS || ret == SQL\_SUCCESS\_WITH\_INFO) {

            SQLCHAR user[20], pass[20];

            SQLINTEGER u\_len, p\_len;

            while (SQLFetch(hstmt) != SQL\_NO\_DATA) {

                SQLGetData(hstmt, 1, SQL\_C\_CHAR, user, 20, &u\_len);

                SQLGetData(hstmt, 2, SQL\_C\_CHAR, pass, 20, &p\_len);

            }

**int** number\_row;

            ret = SQLRowCount(hstmt, &number\_row);

            if (number\_row == 0) *//没有记录*

            {

                return 0;

            }

            else {

                return 1;

            }

            SQLFreeHandle(SQL\_HANDLE\_STMT, hstmt);*//释放语句句柄*

        }

        else printf("查询数据库操作失败！\n");

        SQLDisconnect(hdbc);*//断开与数据库的连接*

    }

    else printf("连接数据库失败!\n");

    SQLFreeHandle(SQL\_HANDLE\_DBC, hdbc);*//释放连接句柄*

    SQLFreeHandle(SQL\_HANDLE\_ENV, henv);*//释放环境句柄*

}

*//查询密码是否正确，正确返回1，不正确返回0*

**int** SQL\_CheckPassword(**char**\* username, **char**\* password)

{

    SQLRETURN ret;

    SQLHENV henv;*//SQLHANDLE henv*

    SQLHDBC hdbc;*//SQLHANDLE hdbc*

    SQLHSTMT hstmt;*//SQLHANDLE hstmt*

    ret = SQLAllocHandle(SQL\_HANDLE\_ENV, NULL, &henv);*//申请环境句柄*

    ret = SQLSetEnvAttr(henv, SQL\_ATTR\_ODBC\_VERSION, (SQLPOINTER)SQL\_OV\_ODBC3, SQL\_IS\_INTEGER);*//设置环境属性*

    ret = SQLAllocHandle(SQL\_HANDLE\_DBC, henv, &hdbc);*//申请数据库连接句柄*

    ret = SQLConnect(hdbc, (SQLCHAR\*)DataSource, SQL\_NTS, (SQLCHAR\*)SQL\_User, SQL\_NTS, (SQLCHAR\*)SQL\_Password, SQL\_NTS);*//连接数据库*

    if (ret == SQL\_SUCCESS || ret == SQL\_SUCCESS\_WITH\_INFO) {

        ret = SQLAllocHandle(SQL\_HANDLE\_STMT, hdbc, &hstmt);*//申请SQL语句句柄*

        SQLCHAR sql[100];

        sprintf(sql, "SELECT \* FROM T\_UserInfo where username='%s' AND password='%s'", username,password);

        ret = SQLExecDirect(hstmt, sql, SQL\_NTS);*//直接执行SQL语句*

        if (ret == SQL\_SUCCESS || ret == SQL\_SUCCESS\_WITH\_INFO) {

            SQLCHAR user[20], pass[20];

            SQLINTEGER u\_len, p\_len;

            while (SQLFetch(hstmt) != SQL\_NO\_DATA) {

                SQLGetData(hstmt, 1, SQL\_C\_CHAR, user, 20, &u\_len);

                SQLGetData(hstmt, 2, SQL\_C\_CHAR, pass, 20, &p\_len);

            }

**int** number\_row;

            ret = SQLRowCount(hstmt, &number\_row);

            if (number\_row == 0) *//没有记录*

            {

                return 0;

            }

            else {

                return 1;

            }

            SQLFreeHandle(SQL\_HANDLE\_STMT, hstmt);*//释放语句句柄*

        }

        else printf("查询数据库操作失败！\n");

        SQLDisconnect(hdbc);*//断开与数据库的连接*

    }

    else printf("连接数据库失败!\n");

    SQLFreeHandle(SQL\_HANDLE\_DBC, hdbc);*//释放连接句柄*

    SQLFreeHandle(SQL\_HANDLE\_ENV, henv);*//释放环境句柄*

}

*//注册账户，成功返回1，失败返回0*

**int** SQL\_RegisterAccount(**char**\* username, **char**\* password, **char**\* email, **char**\* phone)

{

    SQLRETURN ret;

    SQLHENV henv;*//SQLHANDLE henv*

    SQLHDBC hdbc;*//SQLHANDLE hdbc*

    SQLHSTMT hstmt;*//SQLHANDLE hstmt*

    ret = SQLAllocHandle(SQL\_HANDLE\_ENV, NULL, &henv);*//申请环境句柄*

    ret = SQLSetEnvAttr(henv, SQL\_ATTR\_ODBC\_VERSION, (SQLPOINTER)SQL\_OV\_ODBC3, SQL\_IS\_INTEGER);*//设置环境属性*

    ret = SQLAllocHandle(SQL\_HANDLE\_DBC, henv, &hdbc);*//申请数据库连接句柄*

    ret = SQLConnect(hdbc, (SQLCHAR\*)DataSource, SQL\_NTS, (SQLCHAR\*)SQL\_User, SQL\_NTS, (SQLCHAR\*)SQL\_Password, SQL\_NTS);*//连接数据库*

    if (ret == SQL\_SUCCESS || ret == SQL\_SUCCESS\_WITH\_INFO) {

        ret = SQLAllocHandle(SQL\_HANDLE\_STMT, hdbc, &hstmt);*//申请SQL语句句柄*

        SQLCHAR sql[100];

        sprintf(sql, "insert into T\_UserInfo values('%s','%s','%s','%s')", username,password,email,phone);

        ret = SQLExecDirect(hstmt, sql, SQL\_NTS);*//直接执行SQL语句*

        if (ret == SQL\_SUCCESS || ret == SQL\_SUCCESS\_WITH\_INFO) {

            return 1;

            SQLFreeHandle(SQL\_HANDLE\_STMT, hstmt);*//释放语句句柄*

        }

        else return 0;

        SQLDisconnect(hdbc);*//断开与数据库的连接*

    }

    else return 0;;

    SQLFreeHandle(SQL\_HANDLE\_DBC, hdbc);*//释放连接句柄*

    SQLFreeHandle(SQL\_HANDLE\_ENV, henv);*//释放环境句柄*

}

*//设置Session，成功返回1，失败返回0*

**void** SQL\_SetSession(**char**\* username, **char**\* guid, **char**\* timestamp)

{

    SQLRETURN ret;

    SQLHENV henv;*//SQLHANDLE henv*

    SQLHDBC hdbc;*//SQLHANDLE hdbc*

    SQLHSTMT hstmt;*//SQLHANDLE hstmt*

    ret = SQLAllocHandle(SQL\_HANDLE\_ENV, NULL, &henv);*//申请环境句柄*

    ret = SQLSetEnvAttr(henv, SQL\_ATTR\_ODBC\_VERSION, (SQLPOINTER)SQL\_OV\_ODBC3, SQL\_IS\_INTEGER);*//设置环境属性*

    ret = SQLAllocHandle(SQL\_HANDLE\_DBC, henv, &hdbc);*//申请数据库连接句柄*

    ret = SQLConnect(hdbc, (SQLCHAR\*)DataSource, SQL\_NTS, (SQLCHAR\*)SQL\_User, SQL\_NTS, (SQLCHAR\*)SQL\_Password, SQL\_NTS);*//连接数据库*

    if (ret == SQL\_SUCCESS || ret == SQL\_SUCCESS\_WITH\_INFO) {

        ret = SQLAllocHandle(SQL\_HANDLE\_STMT, hdbc, &hstmt);*//申请SQL语句句柄*

        SQLCHAR sql[100];

        sprintf(sql, "insert into T\_Session values('%s','%s','%s')", username, guid, timestamp);

        ret = SQLExecDirect(hstmt, sql, SQL\_NTS);*//直接执行SQL语句*

        if (ret == SQL\_SUCCESS || ret == SQL\_SUCCESS\_WITH\_INFO) {

*//return 1;*

            SQLFreeHandle(SQL\_HANDLE\_STMT, hstmt);*//释放语句句柄*

        }

        SQLDisconnect(hdbc);*//断开与数据库的连接*

    }

    SQLFreeHandle(SQL\_HANDLE\_DBC, hdbc);*//释放连接句柄*

    SQLFreeHandle(SQL\_HANDLE\_ENV, henv);*//释放环境句柄*

}

*//获得对应用户session的GUID和Timestamp*

**int** SQL\_GetSession(**char**\* username, **char**\* guid, **char**\* timestamp)

{

    SQLRETURN ret;

    SQLHENV henv;*//SQLHANDLE henv*

    SQLHDBC hdbc;*//SQLHANDLE hdbc*

    SQLHSTMT hstmt;*//SQLHANDLE hstmt*

    ret = SQLAllocHandle(SQL\_HANDLE\_ENV, NULL, &henv);*//申请环境句柄*

    ret = SQLSetEnvAttr(henv, SQL\_ATTR\_ODBC\_VERSION, (SQLPOINTER)SQL\_OV\_ODBC3, SQL\_IS\_INTEGER);*//设置环境属性*

    ret = SQLAllocHandle(SQL\_HANDLE\_DBC, henv, &hdbc);*//申请数据库连接句柄*

    ret = SQLConnect(hdbc, (SQLCHAR\*)DataSource, SQL\_NTS, (SQLCHAR\*)SQL\_User, SQL\_NTS, (SQLCHAR\*)SQL\_Password, SQL\_NTS);*//连接数据库*

    if (ret == SQL\_SUCCESS || ret == SQL\_SUCCESS\_WITH\_INFO) {

        ret = SQLAllocHandle(SQL\_HANDLE\_STMT, hdbc, &hstmt);*//申请SQL语句句柄*

        SQLCHAR sql[100];

        sprintf(sql, "SELECT \* FROM T\_Session where username='%s'", username);

        ret = SQLExecDirect(hstmt, sql, SQL\_NTS);*//直接执行SQL语句*

        if (ret == SQL\_SUCCESS || ret == SQL\_SUCCESS\_WITH\_INFO) {

            SQLCHAR user[20];*//, guid[50],timestamp[20];*

            SQLINTEGER u\_len, g\_len, t\_len;

            while (SQLFetch(hstmt) != SQL\_NO\_DATA) {

                SQLGetData(hstmt, 1, SQL\_C\_CHAR, user, 20, &u\_len);

                SQLGetData(hstmt, 2, SQL\_C\_CHAR, guid, 50, &g\_len);

                SQLGetData(hstmt, 3, SQL\_C\_CHAR, timestamp, 20, &t\_len);

            }

**int** number\_row;

            ret = SQLRowCount(hstmt, &number\_row);

            if (number\_row == 0) *//没有记录*

            {

                return 0;

            }

            else {

                return 1;

            }

            SQLFreeHandle(SQL\_HANDLE\_STMT, hstmt);*//释放语句句柄*

        }

        else printf("查询数据库操作失败！\n");

        SQLDisconnect(hdbc);*//断开与数据库的连接*

    }

    else printf("连接数据库失败!\n");

    SQLFreeHandle(SQL\_HANDLE\_DBC, hdbc);*//释放连接句柄*

    SQLFreeHandle(SQL\_HANDLE\_ENV, henv);*//释放环境句柄*

}

*//依靠GUID来查询GUID,没找到返回0，有则返回1*

**int** SQL\_CheckGUID(**char**\* g\_username, **char**\* c\_guid, **char**\* g\_timestamp)

{

    SQLRETURN ret;

    SQLHENV henv;*//SQLHANDLE henv*

    SQLHDBC hdbc;*//SQLHANDLE hdbc*

    SQLHSTMT hstmt;*//SQLHANDLE hstmt*

    ret = SQLAllocHandle(SQL\_HANDLE\_ENV, NULL, &henv);*//申请环境句柄*

    ret = SQLSetEnvAttr(henv, SQL\_ATTR\_ODBC\_VERSION, (SQLPOINTER)SQL\_OV\_ODBC3, SQL\_IS\_INTEGER);*//设置环境属性*

    ret = SQLAllocHandle(SQL\_HANDLE\_DBC, henv, &hdbc);*//申请数据库连接句柄*

    ret = SQLConnect(hdbc, (SQLCHAR\*)DataSource, SQL\_NTS, (SQLCHAR\*)SQL\_User, SQL\_NTS, (SQLCHAR\*)SQL\_Password, SQL\_NTS);*//连接数据库*

    if (ret == SQL\_SUCCESS || ret == SQL\_SUCCESS\_WITH\_INFO) {

        ret = SQLAllocHandle(SQL\_HANDLE\_STMT, hdbc, &hstmt);*//申请SQL语句句柄*

        SQLCHAR sql[100];

        sprintf(sql, "SELECT \* FROM T\_Session where guid='%s'", c\_guid);

        ret = SQLExecDirect(hstmt, sql, SQL\_NTS);*//直接执行SQL语句*

        if (ret == SQL\_SUCCESS || ret == SQL\_SUCCESS\_WITH\_INFO) {

*//SQLCHAR user[20];//, guid[50],timestamp[20];*

            SQLINTEGER u\_len, g\_len, t\_len;

            SQLCHAR guid[50];

            while (SQLFetch(hstmt) != SQL\_NO\_DATA) {

                SQLGetData(hstmt, 1, SQL\_C\_CHAR, g\_username, 20, &u\_len);

                SQLGetData(hstmt, 2, SQL\_C\_CHAR, guid, 50, &g\_len);

                SQLGetData(hstmt, 3, SQL\_C\_CHAR, g\_timestamp, 20, &t\_len);

            }

**int** number\_row;

            ret = SQLRowCount(hstmt, &number\_row);

            if (number\_row == 0) *//没有记录*

            {

                return 0;

            }

            else {

                return 1;

            }

            SQLFreeHandle(SQL\_HANDLE\_STMT, hstmt);*//释放语句句柄*

        }

        else printf("查询数据库操作失败！\n");

        SQLDisconnect(hdbc);*//断开与数据库的连接*

    }

    else printf("连接数据库失败!\n");

    SQLFreeHandle(SQL\_HANDLE\_DBC, hdbc);*//释放连接句柄*

    SQLFreeHandle(SQL\_HANDLE\_ENV, henv);*//释放环境句柄*

}

*//依靠username来查询GUID,没找到返回0，有则返回1*

**int** SQL\_CheckGUID2(**char**\* c\_username, **char**\* g\_guid, **char**\* g\_timestamp)

{

    SQLRETURN ret;

    SQLHENV henv;*//SQLHANDLE henv*

    SQLHDBC hdbc;*//SQLHANDLE hdbc*

    SQLHSTMT hstmt;*//SQLHANDLE hstmt*

    ret = SQLAllocHandle(SQL\_HANDLE\_ENV, NULL, &henv);*//申请环境句柄*

    ret = SQLSetEnvAttr(henv, SQL\_ATTR\_ODBC\_VERSION, (SQLPOINTER)SQL\_OV\_ODBC3, SQL\_IS\_INTEGER);*//设置环境属性*

    ret = SQLAllocHandle(SQL\_HANDLE\_DBC, henv, &hdbc);*//申请数据库连接句柄*

    ret = SQLConnect(hdbc, (SQLCHAR\*)DataSource, SQL\_NTS, (SQLCHAR\*)SQL\_User, SQL\_NTS, (SQLCHAR\*)SQL\_Password, SQL\_NTS);*//连接数据库*

    if (ret == SQL\_SUCCESS || ret == SQL\_SUCCESS\_WITH\_INFO) {

        ret = SQLAllocHandle(SQL\_HANDLE\_STMT, hdbc, &hstmt);*//申请SQL语句句柄*

        SQLCHAR sql[100];

        sprintf(sql, "SELECT \* FROM T\_Session where username='%s'", c\_username);

        ret = SQLExecDirect(hstmt, sql, SQL\_NTS);*//直接执行SQL语句*

        if (ret == SQL\_SUCCESS || ret == SQL\_SUCCESS\_WITH\_INFO) {

*//SQLCHAR user[20];//, guid[50],timestamp[20];*

            SQLINTEGER u\_len, g\_len, t\_len;

            SQLCHAR username[20];

            while (SQLFetch(hstmt) != SQL\_NO\_DATA) {

                SQLGetData(hstmt, 1, SQL\_C\_CHAR, username, 20, &u\_len);

                SQLGetData(hstmt, 2, SQL\_C\_CHAR, g\_guid, 50, &g\_len);

                SQLGetData(hstmt, 3, SQL\_C\_CHAR, g\_timestamp, 20, &t\_len);

            }

**int** number\_row;

            ret = SQLRowCount(hstmt, &number\_row);

            if (number\_row == 0) *//没有记录*

            {

                return 0;

            }

            else {

                return 1;

            }

            SQLFreeHandle(SQL\_HANDLE\_STMT, hstmt);*//释放语句句柄*

        }

        else printf("查询数据库操作失败！\n");

        SQLDisconnect(hdbc);*//断开与数据库的连接*

    }

    else printf("连接数据库失败!\n");

    SQLFreeHandle(SQL\_HANDLE\_DBC, hdbc);*//释放连接句柄*

    SQLFreeHandle(SQL\_HANDLE\_ENV, henv);*//释放环境句柄*

}

*//依靠username来删除GUID*

**void** SQL\_DeleteGUID(**char**\* username)

{

    SQLRETURN ret;

    SQLHENV henv;*//SQLHANDLE henv*

    SQLHDBC hdbc;*//SQLHANDLE hdbc*

    SQLHSTMT hstmt;*//SQLHANDLE hstmt*

    ret = SQLAllocHandle(SQL\_HANDLE\_ENV, NULL, &henv);*//申请环境句柄*

    ret = SQLSetEnvAttr(henv, SQL\_ATTR\_ODBC\_VERSION, (SQLPOINTER)SQL\_OV\_ODBC3, SQL\_IS\_INTEGER);*//设置环境属性*

    ret = SQLAllocHandle(SQL\_HANDLE\_DBC, henv, &hdbc);*//申请数据库连接句柄*

    ret = SQLConnect(hdbc, (SQLCHAR\*)DataSource, SQL\_NTS, (SQLCHAR\*)SQL\_User, SQL\_NTS, (SQLCHAR\*)SQL\_Password, SQL\_NTS);*//连接数据库*

    if (ret == SQL\_SUCCESS || ret == SQL\_SUCCESS\_WITH\_INFO) {

        ret = SQLAllocHandle(SQL\_HANDLE\_STMT, hdbc, &hstmt);*//申请SQL语句句柄*

        SQLCHAR sql[100];

        sprintf(sql, "DELETE FROM T\_Session WHERE username='%s'", username);

        ret = SQLExecDirect(hstmt, sql, SQL\_NTS);*//直接执行SQL语句*

        if (ret == SQL\_SUCCESS || ret == SQL\_SUCCESS\_WITH\_INFO) {

            SQLFreeHandle(SQL\_HANDLE\_STMT, hstmt);*//释放语句句柄*

        }

        else printf("查询数据库操作失败！\n");

        SQLDisconnect(hdbc);*//断开与数据库的连接*

    }

    else printf("连接数据库失败!\n");

    SQLFreeHandle(SQL\_HANDLE\_DBC, hdbc);*//释放连接句柄*

    SQLFreeHandle(SQL\_HANDLE\_ENV, henv);*//释放环境句柄*

}

*//记录Web服务开通信息，成功返回1，失败返回0*

**int** SQL\_RecordSite(**char**\* username, **char**\* ip, **char**\* ftpusername, **char**\* ftppassword,**char**\* domain)

{

    SQLRETURN ret;

    SQLHENV henv;*//SQLHANDLE henv*

    SQLHDBC hdbc;*//SQLHANDLE hdbc*

    SQLHSTMT hstmt;*//SQLHANDLE hstmt*

    ret = SQLAllocHandle(SQL\_HANDLE\_ENV, NULL, &henv);*//申请环境句柄*

    ret = SQLSetEnvAttr(henv, SQL\_ATTR\_ODBC\_VERSION, (SQLPOINTER)SQL\_OV\_ODBC3, SQL\_IS\_INTEGER);*//设置环境属性*

    ret = SQLAllocHandle(SQL\_HANDLE\_DBC, henv, &hdbc);*//申请数据库连接句柄*

    ret = SQLConnect(hdbc, (SQLCHAR\*)DataSource, SQL\_NTS, (SQLCHAR\*)SQL\_User, SQL\_NTS, (SQLCHAR\*)SQL\_Password, SQL\_NTS);*//连接数据库*

    if (ret == SQL\_SUCCESS || ret == SQL\_SUCCESS\_WITH\_INFO) {

        ret = SQLAllocHandle(SQL\_HANDLE\_STMT, hdbc, &hstmt);*//申请SQL语句句柄*

        SQLCHAR sql[100];

        sprintf(sql, "insert into T\_SiteInfo values('%s','%s','%s','%s','%s')", username,ip , ftpusername, ftppassword, domain);

        ret = SQLExecDirect(hstmt, sql, SQL\_NTS);*//直接执行SQL语句*

        if (ret == SQL\_SUCCESS || ret == SQL\_SUCCESS\_WITH\_INFO) {

            return 1;

            SQLFreeHandle(SQL\_HANDLE\_STMT, hstmt);*//释放语句句柄*

        }

        else return 0;

        SQLDisconnect(hdbc);*//断开与数据库的连接*

    }

    else return 0;;

    SQLFreeHandle(SQL\_HANDLE\_DBC, hdbc);*//释放连接句柄*

    SQLFreeHandle(SQL\_HANDLE\_ENV, henv);*//释放环境句柄*

}

*//依靠ftpusername来删除Site*

**void** SQL\_DeleteSite(**char**\* ftpusername)

{

    SQLRETURN ret;

    SQLHENV henv;*//SQLHANDLE henv*

    SQLHDBC hdbc;*//SQLHANDLE hdbc*

    SQLHSTMT hstmt;*//SQLHANDLE hstmt*

    ret = SQLAllocHandle(SQL\_HANDLE\_ENV, NULL, &henv);*//申请环境句柄*

    ret = SQLSetEnvAttr(henv, SQL\_ATTR\_ODBC\_VERSION, (SQLPOINTER)SQL\_OV\_ODBC3, SQL\_IS\_INTEGER);*//设置环境属性*

    ret = SQLAllocHandle(SQL\_HANDLE\_DBC, henv, &hdbc);*//申请数据库连接句柄*

    ret = SQLConnect(hdbc, (SQLCHAR\*)DataSource, SQL\_NTS, (SQLCHAR\*)SQL\_User, SQL\_NTS, (SQLCHAR\*)SQL\_Password, SQL\_NTS);*//连接数据库*

    if (ret == SQL\_SUCCESS || ret == SQL\_SUCCESS\_WITH\_INFO) {

        ret = SQLAllocHandle(SQL\_HANDLE\_STMT, hdbc, &hstmt);*//申请SQL语句句柄*

        SQLCHAR sql[100];

        sprintf(sql, "DELETE FROM T\_SiteInfo WHERE ftpusername='%s'", ftpusername);

        ret = SQLExecDirect(hstmt, sql, SQL\_NTS);*//直接执行SQL语句*

        if (ret == SQL\_SUCCESS || ret == SQL\_SUCCESS\_WITH\_INFO) {

            SQLFreeHandle(SQL\_HANDLE\_STMT, hstmt);*//释放语句句柄*

        }

        else {

            cgiHeaderContentType("text/html;charset=gb2312");

            showInfoPage("<h2>发生错误，请联系管理员QQ：1367248805！<a href='index.cgi'>请点击返回重试</a></h2> ");

        }

        SQLDisconnect(hdbc);*//断开与数据库的连接*

    }

    else printf("连接数据库失败!\n");

    SQLFreeHandle(SQL\_HANDLE\_DBC, hdbc);*//释放连接句柄*

    SQLFreeHandle(SQL\_HANDLE\_ENV, henv);*//释放环境句柄*

}

*//返回GUID标识符指针*

**char**\* createGuid()

{

**static** **char** \_guid[100];

    GUID guid = { 0 };

    CoCreateGuid(&guid);

    sprintf(\_guid, "{%08X-%04X-%04x-%02X%02X-%02X%02X%02X%02X%02X%02X}",

        guid.Data1,

        guid.Data2,

        guid.Data3,

        guid.Data4[0],

        guid.Data4[1],

        guid.Data4[2],

        guid.Data4[3],

        guid.Data4[4],

        guid.Data4[5],

        guid.Data4[6],

        guid.Data4[7]);

    return \_guid;

}

*//返回时间戳*

**int** GetTimeStamp() {

    time\_t t;

    t = time(NULL);

**int** ii = time(&t);

    return ii;

}

Functions.h

#pragma once

*//网站相关函数*

**void** Create\_Directory(**char** \*name); *//在D:\localuser\下创建文件夹*

**void** Delete\_Directory(**char** \*name);

**void** Create\_Site(**char**\* name, **char**\* PhysicalPath, **char**\* ip, **char**\* port, **char**\* hostname, **char**\* protocol); *//新建站点*

**void** Delete\_Site(**char**\* name);

**void** Start\_Site(**char**\* name); *//启动站点*

**void** Stop\_Site(**char**\* name); *//停止站点*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*int SQL\_QueryUser(char\* username)*

*参数：username：用户名*

*功能：查询对应用户名在数据库中是否存在*

*返回：用户存在返回1，不存在返回0*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

**int** SQL\_QueryUser(**char**\* username);

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*int SQL\_QueryUser(char\* username)*

*参数：username：用户名*

*功能：查询对应用户名的密码是否正确*

*返回：密码正确返回1，不存在返回0*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

**int** SQL\_CheckPassword(**char**\* username, **char**\* password);

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*int SQL\_RegisterAccount(char\* username, char\* password, char\* email, char\* phone)*

*参数：用户名，密码，邮箱，手机号*

*功能：注册新用户*

*返回：注册成功返回1，否则返回0*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

**int** SQL\_RegisterAccount(**char**\* username, **char**\* password, **char**\* email, **char**\* phone);

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*void SQL\_SetSession(char\* username, char\* guid, char\* timestamp);*

*参数：用户名，GUID唯一标识符，时间戳*

*功能：将用户登陆产生的GUID标识符以及当前时间的时间戳记录到服务器中，*

*返回：无*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

**void** SQL\_SetSession(**char**\* username, **char**\* guid, **char**\* timestamp);

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*int SQL\_GetSession(char\* username, char\* guid, char\* timestamp)*

*参数：用户名，用于存储GUID唯一标识符，用于存储时间戳*

*功能：向服务器取得对应用户的GUID标识符以及时间戳*

*返回：取得成功返回1，否则返回0*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

**int** SQL\_GetSession(**char**\* username, **char**\* guid, **char**\* timestamp);

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*int SQL\_CheckGUID(char\* username, char\* guid, char\* timestamp)*

*参数：用于存储用户名，GUID唯一标识符，用于存储时间戳*

*功能：向服务器查询指定GUID标识符，若存在则再取回对应的用户名和时间戳*

*返回：对应GUID存在返回1，否则返回0*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

**int** SQL\_CheckGUID(**char**\* username, **char**\* guid, **char**\* timestamp);

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*void SQL\_DeleteGUID(char\* username)*

*参数：用户名*

*功能：删除服务器中对应用户的GUID*

*返回：无*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

**void** SQL\_DeleteGUID(**char**\* username);

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*int SQL\_CheckGUID2(char\* c\_username, char\* g\_guid, char\* g\_timestamp)*

*参数：用户名，用于存储GUID，用于存储时间戳*

*功能：向服务器中查询对应用户，并获得其GUID和时间戳*

*返回：查询成功返回1，否则返回0*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

**int** SQL\_CheckGUID2(**char**\* c\_username, **char**\* g\_guid, **char**\* g\_timestamp);

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*int CheckLogin(char \*g\_username)*

*参数：用于存储用户名*

*功能：检查当前状态是否已经登陆，如果登陆成功可返回对应用户名*

*返回：已经成功登陆返回1，否则返回0*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*///*

**int** CheckLogin(**char** \*g\_username);

*//记录Web服务开通信息，成功返回1，失败返回0*

**int** SQL\_RecordSite(**char**\* username, **char**\* ip, **char**\* ftpusername, **char**\* ftppassword, **char**\* domain);

*//依靠ftpusername来删除Site*

**void** SQL\_DeleteSite(**char**\* ftpusername);

*//查询FTP用户是否存在，若存在返回1，不存在返回0*

**int** SQL\_QueryFtpUser(**char**\* ftpusername);

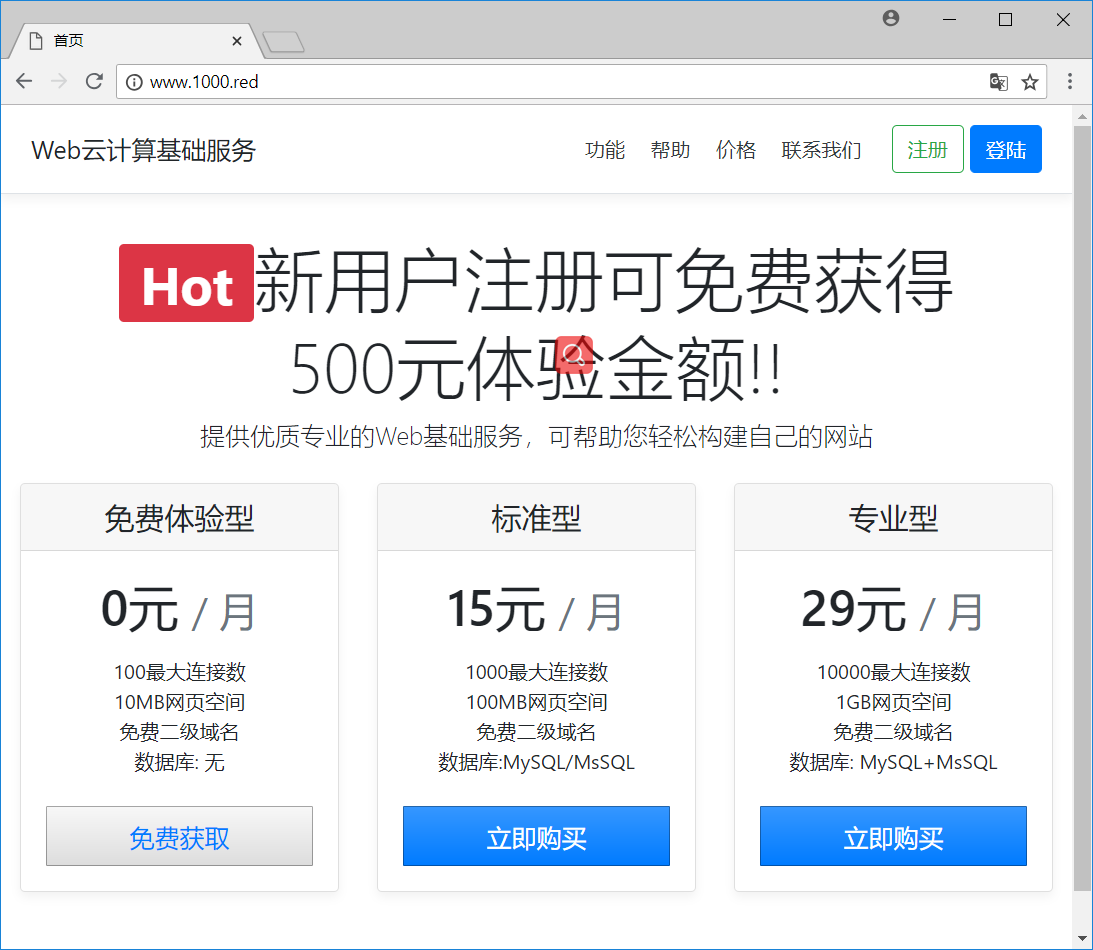
*//查询Domain是否存在，若存在返回1，不存在返回0*

**int** SQL\_QueryDomain(**char**\* domain);

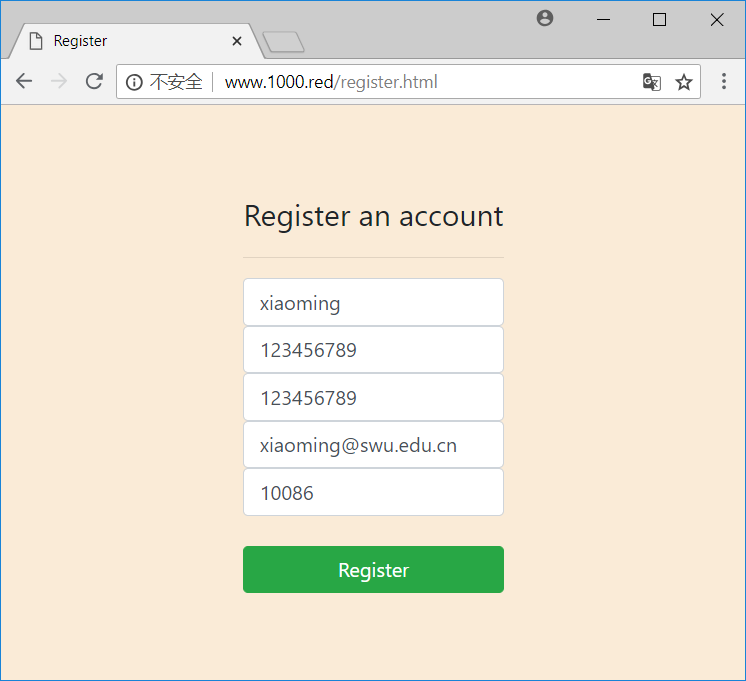
**char**\* createGuid(); *//生成GUID*

**int** GetTimeStamp(); *//得到时间戳*

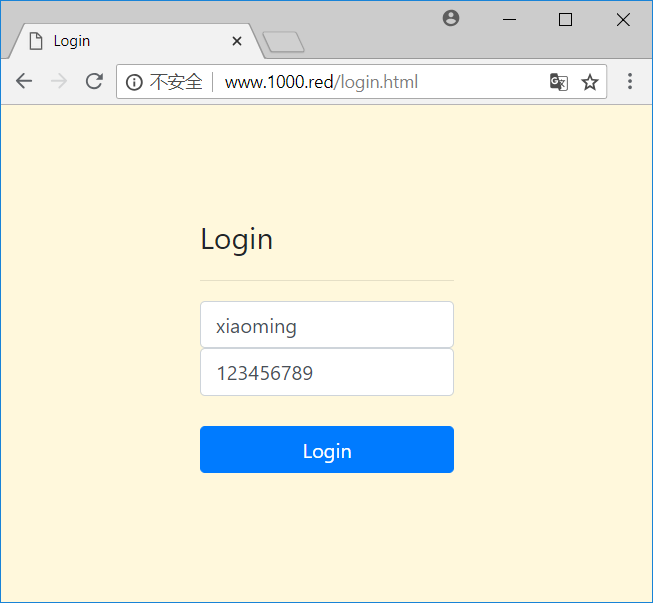
1. **测试**
2. 打开首页



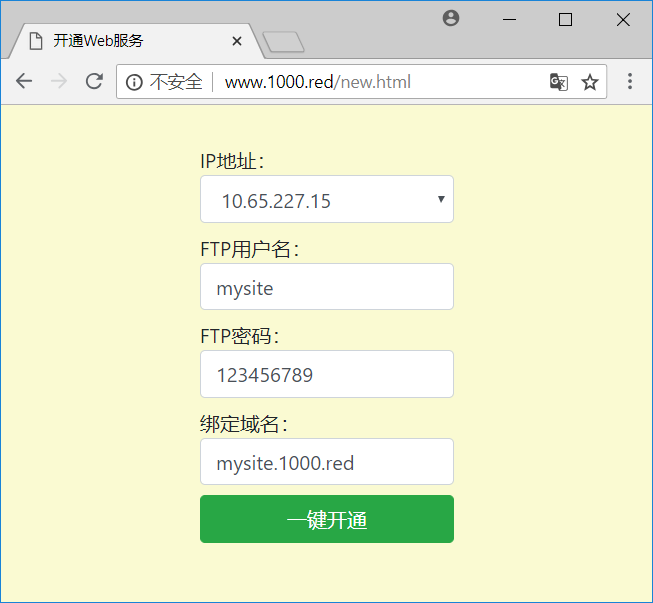
1. 注册用户



1. 登陆系统



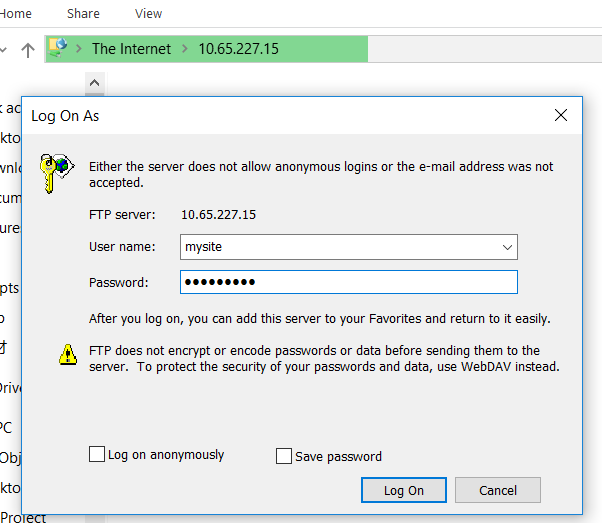
1. 开通Web站点



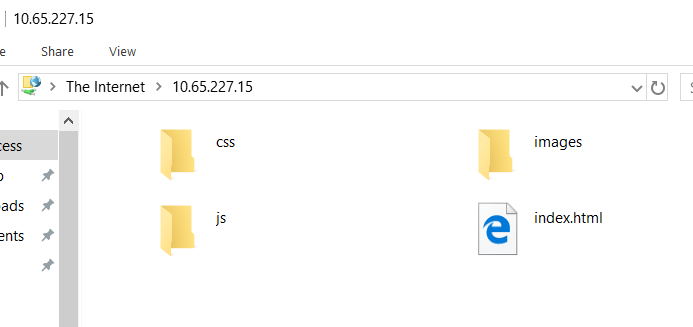
1. 控制面板



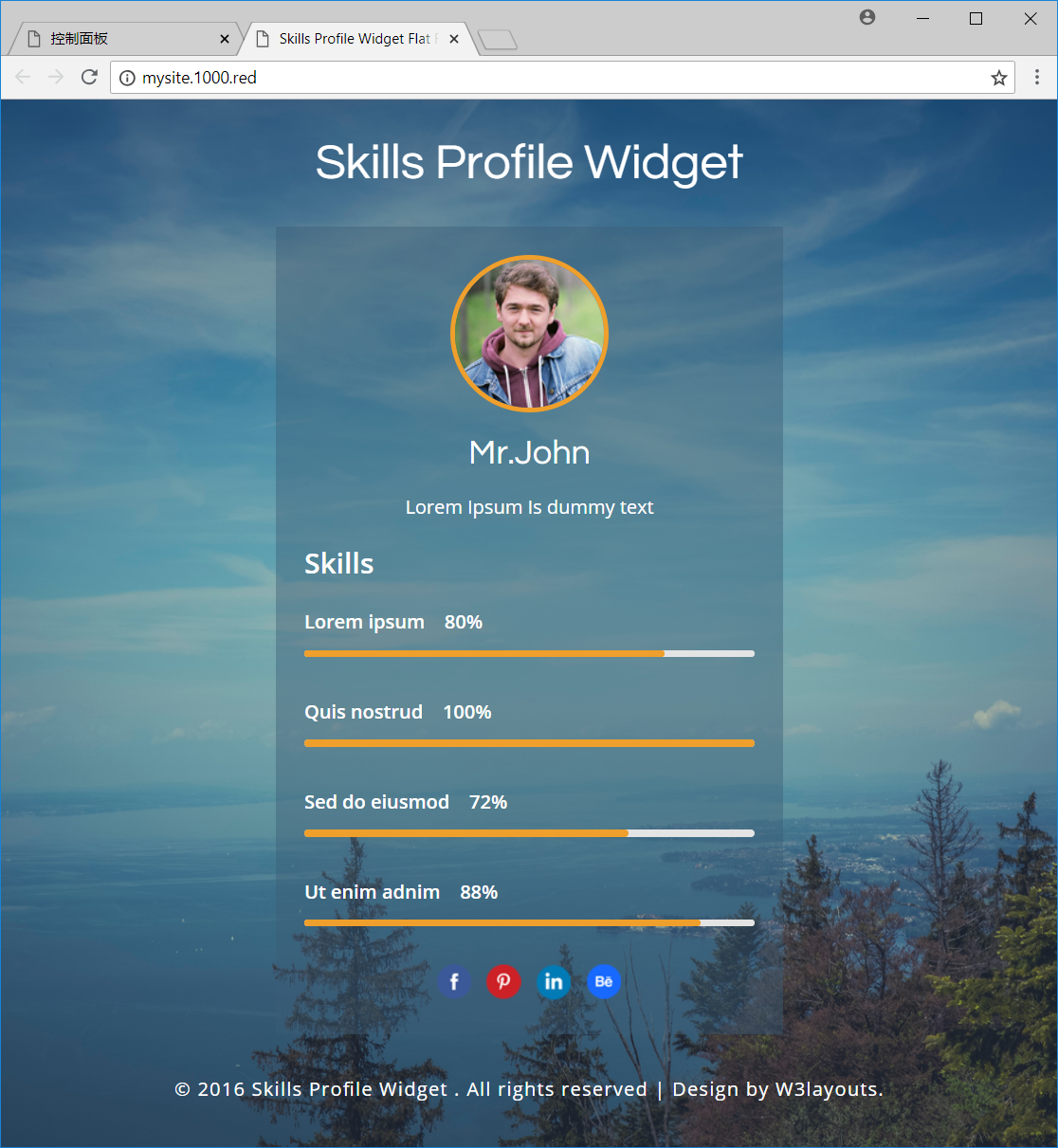
## 连接FTP



上传自己的网页文件



## 打开Web站点首页进行访问



1. **参考文献**
2. Windows Powershell实战指南（第三版）
3. Learn Windows PowerShell in a Month of Lunches
4. SQL Server 2016 从入门到实战
5. Microsoft IIS 10.0 Cookbook
6. Windows-Server-2016-Cookbook
7. manning-learn-windows-iis-in-a-month-of-lunches-2014
8. http://libctemplate.sourceforge.net/doc.html
9. https://boutell.com/cgic/
10. http://getbootstrap.com/docs/4.1/getting-started/introduction/

课程设计报告评级指标

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 评价指标 | A好 | B较好 | C一般 | D较差 | E差 |
| 报告符合规定的格式 |  |  |  |  |  |
| 模块的划分合理 |  |  |  |  |  |
| 数据结构和算法的时空性能高 |  |  |  |  |  |
| 图表规范 |  |  |  |  |  |
| 代码符合规定的风格要求 |  |  |  |  |  |

请在上表中打勾,然后在下面给出总成绩:

**课程设计报告总成绩：□A □B □C □D □E**

**指导教师（签字）：**

**年 月 日**