Source Code

The following is the source code that use in the project to implement.

//Login admin

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace Coin\_Operated\_Wifi

{

public partial class frmAdLogin : Form

{

private OleDbConnection voucherDatabase;

String connectionString = @"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=dbVoucher.mdb";

public frmAdLogin()

{

InitializeComponent();

connectToVoucherDb();

}//end frmAdLogin

//function to connect to Database

private void connectToVoucherDb()

{

voucherDatabase = new OleDbConnection(connectionString);

voucherDatabase.Open();

MessageBox.Show("Connection Successful", "Authentication", MessageBoxButtons.OK, MessageBoxIcon.Information);

voucherDatabase.Close();

}//end connectToVoucherDb

//function to checkUser if it exist

private void checkUser()

{

voucherDatabase.Open();

String queryStr = "SELECT count(\*) FROM useraccount WHERE username='" + txtUser.Text + "' AND userpass ='" + txtPass.Text + "'";

OleDbCommand oleDbCmd = new OleDbCommand(queryStr, voucherDatabase);

int count = (int)oleDbCmd.ExecuteScalar();

if (count > 0)

{

voucherDatabase.Close();

frmRegister formReg = new frmRegister();

formReg.Show();

this.Hide();

}

else

{

MessageBox.Show("Invalid Account");

txtUser.Text = "";

txtPass.Text = "";

voucherDatabase.Close();

}

}//end checkUser

//function if button btnLogin is clicked

private void btnLogin\_Click(object sender, EventArgs e)

{

checkUser();

}//end btnLogin

//function if button btnExit is clicked

private void btnExit\_Click(object sender, EventArgs e)

{

this.Close();

}//end btnExit

}

}

//User Registration

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Runtime.InteropServices;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Threading;

using System.IO.Ports;

using System.Data.OleDb;

using System.Diagnostics;

namespace Coin\_Operated\_Wifi

{

public partial class frmRegister : Form

{

public Boolean check = false;

public delegate void UpdateTextCallback(string message);

public int InpAddress = 889;

public int valueOfPin = 0;

public static int c = 0;

public string availed = "";

private string tblname = "";

private String connectionString = @"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=dbVoucher.mdb";

private Thread thread;

private OleDbConnection voucherDatabase;

public frmRegister()

{

InitializeComponent();

this.clearFields();

StartThread();

}//end frmRegister

//function to Start the Thread

private void StartThread()

{

thread = new Thread(new ThreadStart(TestThread));

thread.Start();

}//end StartThread

//function TestThread

public void TestThread()

{

while (true)

{

Thread.Sleep(100);

valueOfPin = PortAccess.Input(InpAddress);

if (valueOfPin == 127)

{

c = c + 1;

txtCredits.Invoke(new UpdateTextCallback(this.UpdateText), new object[] { c.ToString() });

}

}

}//end TestThread

//function UpdateText with parameter message

private void UpdateText(string message)

{

txtCredits.Text = message;

}//end updateText

//function to clear all textFields

public void clearFields()

{

radDay.Checked = false;

radWeek.Checked = false;

radMonth.Checked = false;

txtName.Text = "";

txtCost.Text = "";

txtConsume.Text = "";

txtCredits.Text = c.ToString();

}//end clearFields

//function if button button2 or clear button is clicked

private void button2\_Click(object sender, EventArgs e)

{

clearFields();

}//end button2

//function if radio button radDay is clicked

private void radDay\_CheckedChanged(object sender, EventArgs e)

{

txtCost.Text = "30";

txtConsume.Text = "24";

this.availed = "1 Day";

this.tblname = "daily";

}//end radDay

//function if radio button radWeek is clicked

private void radWeek\_CheckedChanged(object sender, EventArgs e)

{

txtCost.Text = "100";

txtConsume.Text = "168";

this.availed = "1 Week";

this.tblname = "weekly";

}//end radWeek

//function if radio button radMonth is clicked

private void radMonth\_CheckedChanged(object sender, EventArgs e)

{

txtCost.Text = "250";

txtConsume.Text = "720";

this.availed = "1 Month";

this.tblname = "monthly";

}//end radMonth

//function if toolStripMenuItem login as Administrator is clicked

private void loginAsAdministratorToolStripMenuItem\_Click(object sender, EventArgs e)

{

thread.Abort();

frmAdmin adlog = new frmAdmin();

adlog.Show();

this.Hide();

}//end loginAsAdministratorToolStripMenuItem

//function that initializes c to zero

public void ctozero()

{

c = 0;

clearFields();

}

//function if button btnAdd is clicked

private void btnAdd\_Click(object sender, EventArgs e)

{

check = OpenAccept();

string voucher = "";

int randomnum = 0;

Random random = new Random();

if (check == true && Convert.ToInt32(txtCredits.Text) >= Convert.ToInt32(txtCost.Text))

{

for (int i = 0; i <= 2000; i++)

{

randomnum = random.Next(1, 2000);

if (isAvailable(randomnum) == true)

{

thread.Abort();

voucher = Voucher(randomnum);

this.Hide();

frmConfirm confirm = new frmConfirm(txtName.Text, availed, txtCredits.Text, voucher,randomnum);

confirm.Show();

break;

}

else

{

Console.WriteLine("no id found");

}

}

}

else if (check == true && Convert.ToInt32(txtCredits.Text) < Convert.ToInt32(txtCost.Text))

{

MessageBox.Show("Please Insert Exact Coin Value According to Cost","Insert Coin",MessageBoxButtons.OK,MessageBoxIcon.Error);

}

else

{

MessageBox.Show("Some field/s missing!", "Register Form", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}//end btnAdd

//function if randomNumber generated is available in database

public Boolean isAvailable(int randomNUM) {

Boolean bol = false;

voucherDatabase = new OleDbConnection(connectionString);

voucherDatabase.Open();

String queryStr = "SELECT count(\*) FROM " + tblname + " WHERE ID =" + randomNUM + "";

OleDbCommand oleDbCmd = new OleDbCommand(queryStr, voucherDatabase);

int count = (int)oleDbCmd.ExecuteScalar();

if (count > 0)

{

bol = true;

}

voucherDatabase.Close();

return bol;

}//end isAvailable

//function to get voucher code from database

public string Voucher(int randomNUM)

{

voucherDatabase = new OleDbConnection(connectionString);

voucherDatabase.Open();

String queryStr = "SELECT " + tblname + " FROM " + tblname + " WHERE ID =" + randomNUM + "";

OleDbCommand oleDbCmd = new OleDbCommand(queryStr, voucherDatabase);

OleDbDataReader oldDbReader = oleDbCmd.ExecuteReader();

oldDbReader.Read();

String vouchercode = oldDbReader.GetValue(0).ToString();

voucherDatabase.Close();

return vouchercode;

}//end voucher

//function if All txtFields are filled and returns value check

public Boolean OpenAccept()

{

//if statement for every textboxes

if (txtConsume.Text == "")

check = false;

else if (txtCost.Text == "")

check = false;

else if (txtCredits.Text == "")

check = false;

else if (txtName.Text == "")

check = false;

else

check = true;

return check;

}//end openaccept

}

class PortAccess //new class called portAccess

{

[DllImport("inpout32.dll", EntryPoint = "Inp32")]

public static extern int Input(int adress);

}//end class

}

//Print Receipt

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

using System.Drawing.Printing;

namespace Coin\_Operated\_Wifi

{

public partial class frmConfirm : Form

{

private OleDbConnection voucherDatabase;

private String connectionString = @"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=dbVoucher.mdb";

private string timetouse;

private int rand;

private string tblname = "";

public frmConfirm(string name,string availed, string payment, string code,int random)

{

InitializeComponent();

txtName.Text = name;

txtAvailed.Text = availed;

txtPayment.Text = payment + " pesos";

txtVouch.Text = code;

timetouse = time2use(availed);

rand = random;

}//end frmConfirm

//function for setting time to use base from availed offer

private string time2use(string availed)

{

string a;

if (availed == "1 Day")

{ a = "24 hours"; this.tblname = "daily"; }

else if (availed == "1 Week")

{ a = "168 hours"; this.tblname = "weekly"; }

else

{ a = "720 hours"; this.tblname = "monthly"; }

return a;

}//end time2use

//function if button btnPrint is clicked

private void btnPrint\_Click(object sender, EventArgs e)

{

AddToDatabase();

DeleteVoucher();

printreceipt();

frmRegister frmReg = new frmRegister();

frmReg.ctozero();

this.Hide();

frmReg.Show();

}//end btnPrint

//funtion to Add record to database

public void AddToDatabase()

{

voucherDatabase = new OleDbConnection(connectionString);

using ( OleDbConnection mysql = new OleDbConnection(connectionString))

{

voucherDatabase.Open();

OleDbCommand cmd = new OleDbCommand();

cmd.CommandType = CommandType.Text;

cmd.CommandText = "insert into userregister ([username],[availed\_offer],[coin\_inserted],[time\_to\_use],[voucherCode]) values (?,?,?,?,?)";

cmd.Parameters.AddWithValue("@username",txtName.Text.ToString());

cmd.Parameters.AddWithValue("@availed\_offer", txtAvailed.Text.ToString());

cmd.Parameters.AddWithValue("@coin\_inserted", txtPayment.Text.ToString());

cmd.Parameters.AddWithValue("@time\_to\_use", timetouse.ToString());

cmd.Parameters.AddWithValue("@voucherCode", txtVouch.Text.ToString());

cmd.Connection = voucherDatabase;

cmd.ExecuteNonQuery();

voucherDatabase.Close();

}

}//end AddToDatabase

//function to delete record from voucher

public void DeleteVoucher()

{

voucherDatabase = new OleDbConnection(connectionString);

Console.WriteLine(rand);

OleDbDataAdapter oledbAdapter = new OleDbDataAdapter();

string queryStr = "DELETE FROM " + tblname + " WHERE id = " + rand;

voucherDatabase.Open();

oledbAdapter.DeleteCommand = voucherDatabase.CreateCommand();

oledbAdapter.DeleteCommand.CommandText = queryStr;

oledbAdapter.DeleteCommand.ExecuteNonQuery();

voucherDatabase.Close();

}//end deleteVoucher

//function for printing

private void printreceipt()

{

PrintDialog printdialog = new PrintDialog();

PrintDocument printdocument = new PrintDocument();

printdialog.Document = printdocument;

printdocument.PrintPage += new PrintPageEventHandler(printDocument\_PrintPage);

DialogResult result = printdialog.ShowDialog();

if (result == DialogResult.OK)

{

printdocument.Print();

}

}//end printreceipt

//function for the format of the print page

void printDocument\_PrintPage(Object sender, PrintPageEventArgs e)

{

Graphics graphic = e.Graphics;

Font font = new Font("Courire New", 12);

float fontHeight = font.GetHeight();

int startX = 10;

int startY = 10;

int offset = 10;

graphic.DrawString("Welcome to The Coin", new Font("Courier New", 10), new SolidBrush(Color.Black), startX, startY);

graphic.DrawString(" Operated Wifi", new Font("Courier New", 10), new SolidBrush(Color.Black), startX, startY + offset);

graphic.DrawString("Name : " + txtName.Text, new Font("Courier New", 8), new SolidBrush(Color.Black), startX, startY + offset + 32);

graphic.DrawString("Availed: " + txtAvailed.Text, new Font("Courier New", 8), new SolidBrush(Color.Black), startX, startY + offset + 44);

graphic.DrawString("Payment: " + txtPayment.Text, new Font("Courier New", 8), new SolidBrush(Color.Black), startX, startY + offset + 56);

graphic.DrawString("Time : " + timetouse, new Font("Courier New", 8), new SolidBrush(Color.Black), startX, startY + offset + 68);

graphic.DrawString("Code : " + txtVouch.Text, new Font("Courier New", 8), new SolidBrush(Color.Black), startX, startY + offset + 80);

graphic.DrawString("Thank You!Come Again.",new Font("Courier New", 10), new SolidBrush(Color.Black), startX, startY + offset +100);

}//end printDocument\_PrintPage()

}

}

//Admin Log-in to view the user registered

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace Coin\_Operated\_Wifi

{

public partial class frmAdmin : Form

{

private OleDbConnection voucherDatabase;

private String connectionString = @"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=dbVoucher.mdb";

public frmAdmin()

{

InitializeComponent();

}//end frmAdmin

//function to checkUser if it exist

private void checkUser()

{

voucherDatabase = new OleDbConnection(connectionString);

voucherDatabase.Open();

String queryStr = "SELECT count(\*) FROM useraccount WHERE username='" + txtUser.Text + "' AND userpass ='" + txtPass.Text + "'";

OleDbCommand oleDbCmd = new OleDbCommand(queryStr, voucherDatabase);

int count = (int)oleDbCmd.ExecuteScalar();

if (count > 0)

{

this.Hide();

voucherDatabase.Close();

frmAdminPanel adPa = new frmAdminPanel();

adPa.Show();

}

else

{

MessageBox.Show("Invalid Account");

txtUser.Text = "";

txtPass.Text = "";

voucherDatabase.Close();

}

}//end checkUser

//function if button btnLog is clicked

private void btnLog\_Click(object sender, EventArgs e)

{

checkUser();

}//end btnLog

//function if button btnExit is clicked

private void btnExit\_Click\_1(object sender, EventArgs e)

{

this.Hide();

frmRegister frmReg = new frmRegister();

frmReg.ctozero();

frmReg.Show();

}//end btnExit

//function if button button1 or close Program is clicked

private void button1\_Click(object sender, EventArgs e)

{

Application.Exit();

}//end close program button

}

}

//admin panel

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.OleDb;

using System.Drawing;

using System.Drawing.Printing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Coin\_Operated\_Wifi

{

public partial class frmAdminPanel : Form

{

private String connectionString = @"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=dbVoucher.mdb";

private OleDbConnection voucherDatabase;

private string idnumber;

public frmAdminPanel()

{

InitializeComponent();

getDataTable();

btnReprint.Enabled = false;

btnDelete.Enabled = false;

}//end frmAdminPanel()

//function to display contents of userregister table from database to Datagrid

private void getDataTable()

{

Users\_Datagrid.DataSource = null;

Users\_Datagrid.Rows.Clear();

Users\_Datagrid.Refresh();

string query = "SELECT \* FROM userregister";

using (voucherDatabase = new OleDbConnection(connectionString))

{

using (OleDbDataAdapter adapter = new OleDbDataAdapter(query, voucherDatabase))

{

DataTable datatable = new DataTable();

adapter.Fill(datatable);

for (int i = 0; i < datatable.Rows.Count; i++)

{

Users\_Datagrid.Rows.Add(datatable.Rows[i][0],datatable.Rows[i][1],datatable.Rows[i][2],datatable.Rows[i][3],datatable.Rows[i][4],datatable.Rows[i][5]);

}

}

}

}//end getDataTable

//logout from admin page

private void btnLogout\_Click(object sender, EventArgs e)

{

this.Hide();

frmRegister reg = new frmRegister();

reg.ctozero();

reg.Show();

}//end btnLogout

//function for printing

private void printreceipt()

{

PrintDialog printdialog = new PrintDialog();

PrintDocument printdocument = new PrintDocument();

printdialog.Document = printdocument;

printdocument.PrintPage += new PrintPageEventHandler(printDocument\_PrintPage);

DialogResult result = printdialog.ShowDialog();

if (result == DialogResult.OK)

{

printdocument.Print();

}

}//end printreceipt()

//function for the format of the print page

void printDocument\_PrintPage(Object sender, PrintPageEventArgs e)

{

Graphics graphic = e.Graphics;

Font font = new Font("Courire New", 12);

float fontHeight = font.GetHeight();

int startX = 10;

int startY = 10;

int offset = 10;

graphic.DrawString("Welcome to The Coin", new Font("Courier New", 10), new SolidBrush(Color.Black), startX, startY); //

graphic.DrawString(" Operated Wifi", new Font("Courier New", 10), new SolidBrush(Color.Black), startX, startY + offset); //

graphic.DrawString("Name : " + txtName.Text, new Font("Courier New", 8), new SolidBrush(Color.Black), startX, startY + offset + 32); //

graphic.DrawString("Availed: " + txtAvailed.Text, new Font("Courier New", 8), new SolidBrush(Color.Black), startX, startY + offset + 44); // this part are

graphic.DrawString("Payment: " + txtPayment.Text, new Font("Courier New", 8), new SolidBrush(Color.Black), startX, startY + offset + 56); // the ones to be printed

graphic.DrawString("Time : " + txtConsume.Text, new Font("Courier New", 8), new SolidBrush(Color.Black), startX, startY + offset + 68); // in the receipt

graphic.DrawString("Code : " + txtVoucher.Text, new Font("Courier New", 8), new SolidBrush(Color.Black), startX, startY + offset + 80); //

graphic.DrawString("Thank You!Come Again.",new Font("Courier New", 10), new SolidBrush(Color.Black), startX, startY + offset +100); //

}//end printDocument\_PrintPage()

//function to clear all textboxes

private void clrTxtBox()

{

txtName.Text = "";

txtAvailed.Text = "";

txtConsume.Text = "";

txtPayment.Text = "";

txtVoucher.Text = "";

btnReprint.Enabled = false;

btnDelete.Enabled = false;

}//end clrTextBox

//print the chosen user

private void btnReprint\_Click(object sender, EventArgs e)

{

printreceipt();

clrTxtBox();

}//end btnReprint

//if cell is double clicked shows contents to the textboxes

private void Users\_Datagrid\_CellDoubleClick(object sender, DataGridViewCellEventArgs e)

{

btnReprint.Enabled = true;

btnDelete.Enabled = true;

foreach (DataGridViewRow row in Users\_Datagrid.SelectedRows)

{

idnumber = row.Cells[0].Value.ToString();

txtName.Text = row.Cells[1].Value.ToString();

txtPayment.Text = row.Cells[3].Value.ToString();

txtConsume.Text = row.Cells[4].Value.ToString();

txtAvailed.Text = row.Cells[2].Value.ToString();

txtVoucher.Text = row.Cells[5].Value.ToString();

}

}//end Users\_Datagrid\_CellDoubleClick

//deletes selected record

private void btnDelete\_Click(object sender, EventArgs e)

{

voucherDatabase = new OleDbConnection(connectionString);

OleDbDataAdapter oledbAdapter = new OleDbDataAdapter();

string queryStr = "DELETE FROM userregister WHERE id = " + idnumber;

voucherDatabase.Open();

oledbAdapter.DeleteCommand = voucherDatabase.CreateCommand();

oledbAdapter.DeleteCommand.CommandText = queryStr;

int rows = oledbAdapter.DeleteCommand.ExecuteNonQuery();

if (rows > 0)

{

MessageBox.Show("User Record Successfully Deleted from Database!", "User Delete Record", MessageBoxButtons.OK, MessageBoxIcon.Information);

voucherDatabase.Close();

getDataTable();

clrTxtBox();

}

else

voucherDatabase.Close();

}//end btnDelete

}//end of class

}