**ЕТШ „Никола Тесла“ Ниш**

**Семинарски рад**

**Предмет:**

**Веб Програмирање**

**Ученик:**

**Никола Симоновић 3/4**

Назив задатка: Тенкови ВС

Дата је текстуална датотека под називом „vazduhoplovi.txt“ која представља податке о авионима. Сваки ред у текстуалној датотеци је поредставља податке о једном авиону у облику:

**Шифра|Назив|Количина|Рад|Калибар|Произвођач|Година производње|Слика**

Креирати веб апликацију под називом „Ратно ваздухопловство“ која се састоји од две странице:

* Почетна;
* Измени.

На страници „Почетна“ поставити поља за унос параметара за претраживање података о ваздухоплову, тастер „Тражи“ и линк на прелаз на страницу „Измени“. Потребно је омогућити претраживање странице „Почетна“ по следећим пољима:

* Шифра – поље за унос текста са одговарајућом ознаком (TextBox),
* Назив – поље за унос текста са одговарајућом ознаком (TextBox),
* Количина - поље за унос текста са одговарајућом ознаком (TextBox),
* Рад - поље за унос текста са одговарајућом ознаком (TextBox),
* Калибар - поље за унос текста са одговарајућом ознаком (TextBox),
* Произвођач – падајућа листа (Drop Down List) са одговарајућом ознаком са предефинисаним вредностима,
* Година производње – падајућа листа (Drop Down List) са одговарајућом ознаком са предефинисаним вредностима.

Код текстуалних поља омогућити претраживање тако да се унети узорак садржи у запису који се тражи. Поље за избор године реализовано падајућом листом напунити подацима о свим различитим годинама издања који се јављају у текстуалној датотеци са подацима.

Имплементирати механизам за претрагу који ће према задатом критеријуму приказати албуме који одговарају параметрима претраге.

Резултат претраге представити у облику табеле која садржи сва поља (колоне) која су дефинисана у датотеци са подацима укључујући и слику омота коју треба приказати у првој колони.

На страници „О ваздухоплову“ написати податке о авиону, кратко упуство за коришћење веб апликације и линк за прелаз на страницу „Сви ваздухоплови“

CSS код издвојити у посебну екстерну датотеку и учитати је на основну страницу при стартовању апликације.

Први корак у прављењу апликација овог типа је прављење класе у којој ћемо направити пропортије особина. У овом случају је то класа “Tenkovi”.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace WebTenkovi

{

public class Tenkovi

{

public string Sifra { get; set; }

public string Naziv { get; set; }

public string Kolicina { get; set; }

public string Rad { get; set; }

public string Kalibar { get; set; }

public string Proizvodjac { get; set; }

public string GodinaProizvodnje { get; set; }

public string Slika { get; set; }

}

}

Свака особина има свој тип и мора бити public, да би је могли користити у другим класама. Следећи корак је прављење класе „DataProvider“. У класи је потребно унети using System.IO за коришћење потребних функција. У тој класи правимо функције за манипулисање података из текстуалног фајла. То су функције:

* UcitajPodatke;
* DodajPodatakUFajl;
* ObrisiPodatakIzListe;
* ObrisiPodatakIzFajla
* Pronadji(ili filtriraj podakte)
* IzmeniPodatak;
* GetTenkovi;
* UcitajGodine;
* UcitajProizvodjaca;

Такође, у претходно наведеној класи морамо декларисати приватну особину која ће нам касније требати за путању до текстуалног фајла. Прави се истанца која ће се користити касније за долазак до текстуалног фајла.

public DataProvider(string putanja)

{

\_putanja = putanja;

}

Класа UcitajPodatke:

public List<Tenkovi> UcitajPodatke()

{

List<Tenkovi> listaTenkova = new List<Tenkovi>();

try

{

using(StreamReader sr = new StreamReader(\_putanja))

{

string red;

while((red = sr.ReadLine())!=null)

{

var niz = red.Split('|');

Tenkovi tenk = new Tenkovi

{

Sifra = niz[0],

Naziv = niz[1],

Kolicina = niz[2],

Rad = niz[3],

Kalibar = niz[4],

Proizvodjac = niz[5],

GodinaProizvodnje = niz[6],

Slika = niz[7]

};

listaTenkova.Add(tenk);

}

}

return listaTenkova;

}

catch(Exception ex)

{

Poruka = ex.Message;

return null;

}

}

У класи се декларише ред у фајлу и низ у коме ћемо стављати те редове. Користе се наредбе:

* Try;
* Catch;
* Using;

Прави се листа у којој се после ти редови стављају и спремни су за штампу.

Класа DodajUFajl

public bool DodajPodatakUFajl(Tenkovi tenkovi)

{

try

{

using(StreamWriter sw = new StreamWriter(\_putanja, true))

{

string red = string.Format($"{tenkovi.Sifra}|{tenkovi.Naziv}|{tenkovi.Kolicina}|{tenkovi.Rad}");

sw.WriteLine(red);

}

Poruka = "Upesno dodat novi ucenik u fajl";

return true;

}

catch (Exception ex)

{

Poruka = ex.Message;

return false;

}

}

Додају се нови подаци у фајл.

Класа ObrisiPodatkeIzListe:

public List<Tenkovi> ObrisiPodatakIzListe(string sifra)

{

var listaTenkova = UcitajPodatke();

Tenkovi tenkovi = listaTenkova.FirstOrDefault(t => t.Sifra == sifra);

listaTenkova.Remove(tenkovi);

return listaTenkova;

}

Бришу се сви подаци из листе.

Класа IzmeniPodatak:

public bool IzmeniPodatak(Tenkovi t\_tenk)

{

var listaTenkova = UcitajPodatke();

var tenk = listaTenkova.FirstOrDefault(t => t.Sifra == t\_tenk.Sifra);

if (tenk != null)

{

tenk.Sifra = t\_tenk.Sifra;

tenk.Naziv = t\_tenk.Naziv;

tenk.Kolicina = t\_tenk.Kolicina;

tenk.Rad = t\_tenk.Rad;

}

bool podatakIzmenjen = UpisiPodatke(listaTenkova);

return podatakIzmenjen;

}

Мења податке које смо унели у ту класу.

Класа UpisiPodatke:

public bool UpisiPodatke(List<Tenkovi> listaTenkova)

{

try

{

using(StreamWriter sw = new StreamWriter(\_putanja))

{

foreach(var tenk in listaTenkova)

{

string red = string.Format($"{tenk.Sifra}|{tenk.Naziv}|{tenk.Kolicina}|{tenk.Rad}|{tenk.Kalibar}|{tenk.Proizvodjac}|{tenk.Proizvodjac}|{tenk.GodinaProizvodnje}|{tenk.Slika}");

sw.WriteLine(red);

}

}

Poruka = "Uspesno!";

return true;

}

catch(Exception ex)

{

Poruka = ex.Message;

return false;

}

}

Са овом класом уписујемо нове податке.

Класа ObrisiPodatkeIzFajla:

public bool ObrisiPodatakIzFajla(string sifra)

{

return UpisiPodatke(ObrisiPodatakIzListe(sifra));

}

Комбиновањем две класе добијамо нову класу којом бришемо одређене податке из фајла.

Класа Pronadji:

public List<Tenkovi> Pronadji(string sifra = "", string naziv ="", string kolicina = "", string rad ="", string kalibar = "", string proizvodjac = "", string godinaProizvodnje = "")

{

var listaTenkova = UcitajPodatke();

var listaRezultata = new List<Tenkovi>();

foreach(var Tenk in listaTenkova)

{

if(

(string.IsNullOrWhiteSpace(sifra) || Tenk.Sifra.Contains(sifra))&&

(string.IsNullOrWhiteSpace(naziv) || Tenk.Naziv.Contains(naziv))&&

(string.IsNullOrWhiteSpace(kolicina) || Tenk.Kolicina.Contains(kolicina))&&

(string.IsNullOrWhiteSpace(rad) || Tenk.Rad.Contains(rad))&&

(string.IsNullOrWhiteSpace(kalibar) || Tenk.Kalibar.Contains(kalibar))&&

(string.IsNullOrWhiteSpace(proizvodjac) || Tenk.Proizvodjac.Contains(proizvodjac))&&

(string.IsNullOrWhiteSpace(godinaProizvodnje) || Tenk.GodinaProizvodnje.Contains(godinaProizvodnje))

)

{

listaRezultata.Add(Tenk);

}

}

return listaRezultata;

}

Ова класа филтрира податке и са њом проналазимо жељене податке.

Класа UcitajGodinu I UcitajProizvoddaca

public List<string> UcitajProizvodjaca()

{

var listaTenkova = UcitajPodatke();

var listaPro = new List<string>();

if (listaTenkova != null)

{

foreach (var tenk in listaTenkova)

{

string proizvodjac = tenk.Proizvodjac;

if (!listaPro.Contains(proizvodjac))

{

listaPro.Add(proizvodjac);

}

listaPro.Sort();

}

}

return listaPro;

}

public List<string> UcitajGodine()

{

var listaTenkova = UcitajPodatke();

var listaGod = new List<string>();

if (listaTenkova != null)

{

foreach (var tenk in listaTenkova)

{

string godinaProizvodnje = tenk.GodinaProizvodnje;

if (!listaGod.Contains(godinaProizvodnje))

{

listaGod.Add(godinaProizvodnje);

}

}

listaGod.Sort();

}

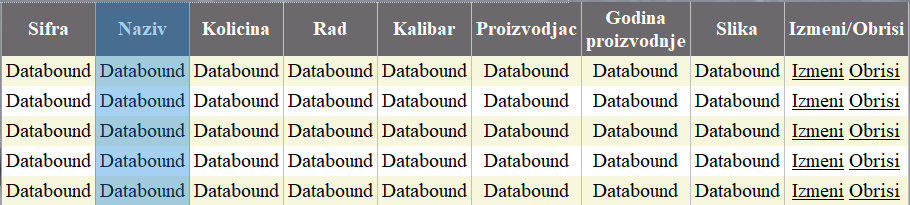
return listaGod;

}

}

Функције које нам омогућавају читање одређених података из фајла. У овом случају су то Године произодње и произвођач. Користе се у проналажењу података.

Податке на ВебФорми штампамо уз помоћ “GridView”-a.



GridView са подацима спајамо тако што унесемо путање до фајлова.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace WebTenkovi

{

public partial class WebForm1 : System.Web.UI.Page

{

DataProvider dp;

protected void Page\_Load(object sender, EventArgs e)

{

string putanja = Server.MapPath("~/Slike/Tenkovi.txt");

dp = new DataProvider(putanja);

if(!IsPostBack)

{

GridView1.DataSource = dp.UcitajPodatke();

DropDownList1.DataSource = dp.UcitajProizvodjaca();

DropDownList2.DataSource = dp.UcitajGodine();

DataBind();

}

}

protected void dodaj\_Click(object sender, EventArgs e)

{

Session["Tenkovi"] = null;

Response.Redirect("Izmena.aspx");

}

protected void GridView1\_RowDeleting(object sender, GridViewDeleteEventArgs e)

{

//string sifra = GridView1.Rows[e.RowIndex].Cells[0].Text;

dp.ObrisiPodatakIzFajla(GridView1.Rows[e.RowIndex].Cells[0].Text);

Response.Redirect("Pocetna.aspx");

}

protected void GridView1\_RowEditing(object sender, GridViewEditEventArgs e)

{

Tenkovi tenkovi = dp.GetTenkovi(GridView1.Rows[e.NewEditIndex].Cells[0].Text);

Session["Tenkovi"] = tenkovi;

Response.Redirect("Izmena.aspx");

}

protected void btnPronadji\_Click1(object sender, EventArgs e)

{

string sifra = txtSifra.Text;

string naziv = txtNaziv.Text;

string kolicina = txtKolicina.Text;

string rad = txtRad.Text;

string kalibar = txtKalibar.Text;

string proizvodjac = DropDownList1.SelectedValue;

string godinaProizvodnje = DropDownList2.SelectedValue;

GridView1.DataSource = dp.Pronadji(sifra, naziv, kolicina, rad, kalibar, proizvodjac, godinaProizvodnje);

GridView1.DataBind();

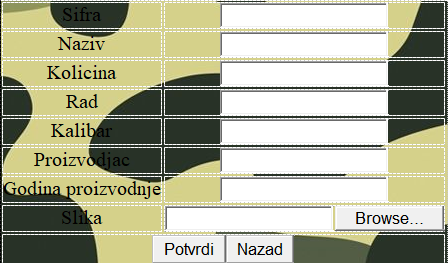
}

}

}

У овој радној класи Почетне стране се налазе и функције за button-е.

На страници Измени имамо: 6 textbox-a и један file uploader.



Као и за почетну, тако и за страну Измени имамо помоћну класу у којој се налазе функције потребне за жељени рад странице.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace WebTenkovi

{

public partial class Izmena : System.Web.UI.Page

{

DataProvider dp;

Tenkovi tenkovi;

protected void Page\_Load(object sender, EventArgs e)

{

string putanja = Server.MapPath("~/Slike/Tenkovi.txt");

dp = new DataProvider(putanja);

tenkovi = Session["Tenkovi"] as Tenkovi;

if(!IsPostBack)

{

if (tenkovi != null)

{

txtSifraIzmena.Text = tenkovi.Sifra;

txtNazivIzmena.Text = tenkovi.Naziv;

txtKolicinaIzmena.Text = tenkovi.Kolicina;

txtRadIzmena.Text = tenkovi.Rad;

txtKalibarIzmena.Text = tenkovi.Kalibar;

txtProizvodjacIzmena.Text = tenkovi.Proizvodjac;

txtGodinaIzmena.Text = tenkovi.GodinaProizvodnje;

}

}

}

protected void Potvrdi\_Click(object sender, EventArgs e)

{

Tenkovi t = new Tenkovi()

{

Sifra = txtSifraIzmena.Text,

Naziv = txtNazivIzmena.Text,

Kolicina = txtKolicinaIzmena.Text,

Rad = txtRadIzmena.Text,

Kalibar = txtKalibarIzmena.Text,

Proizvodjac = txtProizvodjacIzmena.Text,

GodinaProizvodnje = txtGodinaIzmena.Text,

Slika = FileUpload1.FileName

};

if(t!=null)

{

dp.DodajPodatakUFajl(t);

}

Response.Redirect("Pocetna.aspx");

}

protected void Nazad\_Click(object sender, EventArgs e)

{

Response.Redirect("Pocetna.aspx");

}

}

}