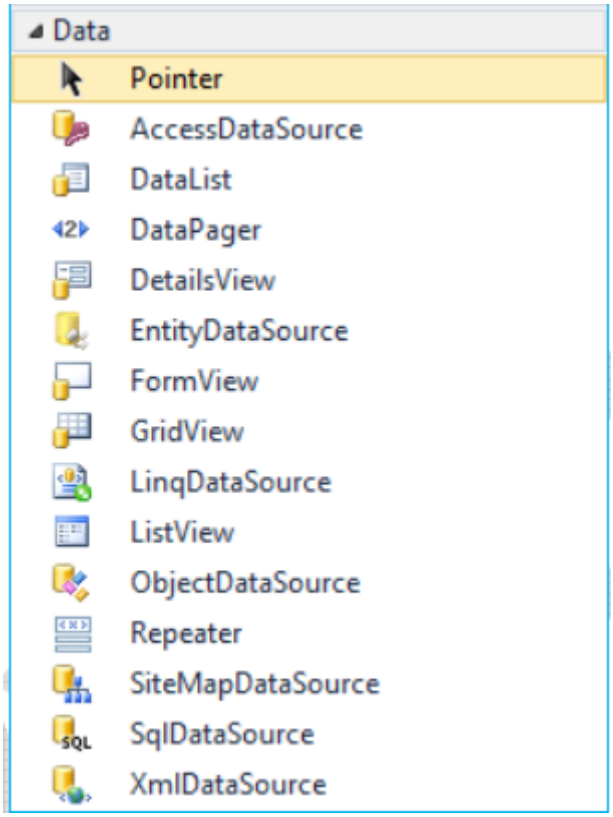
ASP web programiranje

Treća laboratorijska vježba

Velika Gorica, 2022.

1. ASP.NET kontrole iz skupine Data

Pri izradi web aplikacija vrlo često je nužno povezati se na bazu podataka, poslati SQL upit i prikazati korisniku rezultat SQL upita. U ASP.NET-u za povezivanje na bazu podataka, slanje SQL upita i prikaz rezultata, koriste se klase iz skupine Data koje pojednostavljuju cijeli postupak (Slika 1).



Slika 1

Klase koje omogućuju spajanje na bazu podataka i slanje SQL upita su:

- AccessDataSource
- EntityDataSource
- LinqDataSource
- ObjectDataSource
- SiteMapDataSource
- SqlDataSource
- XmlDataSource

Navedene klase nemaju grafičko sučelje, tj. nisu vidljive korisniku u internet pregledniku. Prilikom generiranja HTML dokumenta na poslužitelju navedene klase se ne postavljaju u HTML dokument, odnosno ostaju na poslužitelju.

Gore navedene klase služe za automatsko povezivanje na bazu podataka i slanje SQL upita. Podaci za spajanje na bazu podataka se mogu upisati direktno u sam objekt na za to predviđeno mjesto ili u konfiguracijsku datoteku. Preporučljivo je podatke koji se s vremenom mogu promijeniti zapisati u konfiguracijsku datoteku radi lakše izmjene.

Podaci koji se dobiju kao rezultat iz baze podataka se moraju prikazati na jednoj od preostalih klasa iz skupine Data (npr. GridView). Klasa GridView sadrži svojstvo DataSourceID preko kojeg se povezuje s jednom od klasa za spajanje na bazu podataka.

Klase iz skupine Data omogućuju jednostavniji prikaz većeg broja zapisa iz baze podataka bez potrebe za pisanjem izvornog koda.

Primjer

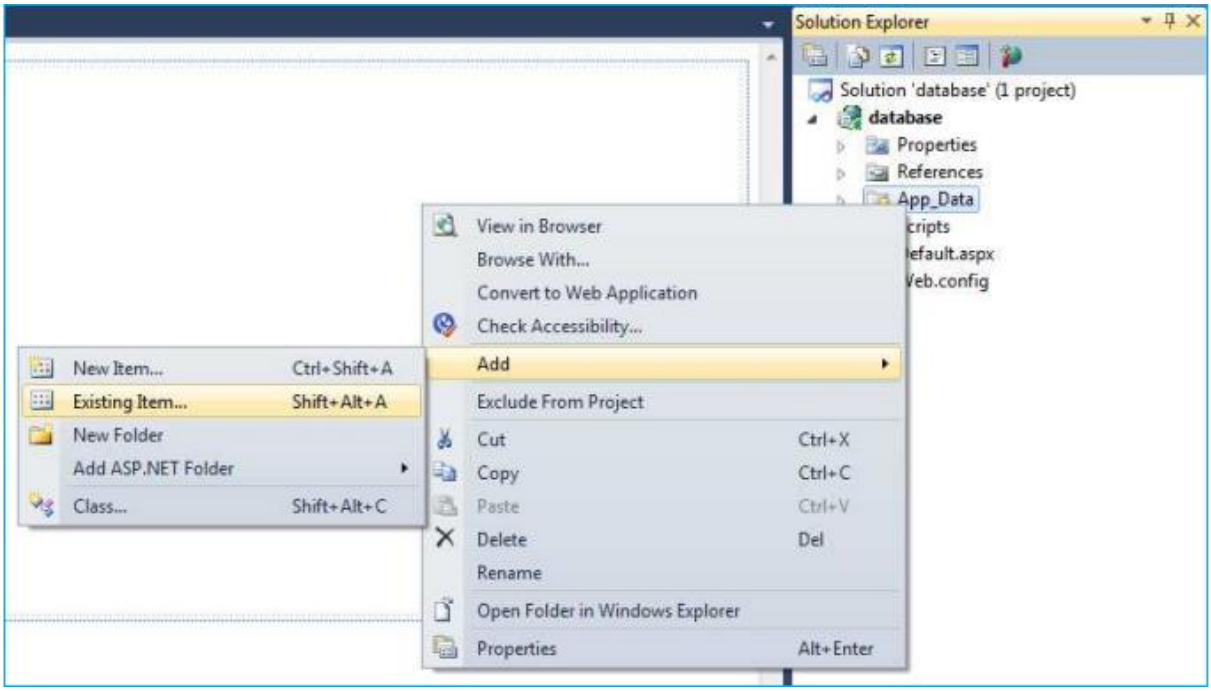
Povezivanje na bazu podataka, slanje SQL upita i prikaz rezultata na ASP.NET kontroli GridView.

Na web formu se dodaju dvije ASP.NET kontrole:

- AccessDataSource
- GridView

Napomena: Ukoliko unutar programskog alata Visual Studio nije dostupna kontrola *AccessDataSource*, može se upotrijebiti kontrola *SqlDataSource*. Ovisno o verziji platforme .NET Framework biti će prikazana ili sakrivena kontrola *AccessDataSource* u kartici Toolbox unutar programskog alata Visual Studio. Ukoliko se koristi starija verzija platforme .NET Framework (<4.0) biti će prikazana kontrola *AccessDataSource*, a ako se koristi novija verzija platforme .NET Framework (>=4.0) neće biti prikazana kontrola *AccessDataSource*.

Prije povezivanja na Access bazu podataka potrebno je postaviti bazu podataka u direktorij App_Data unutar projekta i dodati referencu na bazu podataka (Slika 2).



Slika 2. dodavanje reference na bazu podataka

Za kontrolu *AccessDataSource* je potrebno promijeniti dva svojstva (Slika 3):

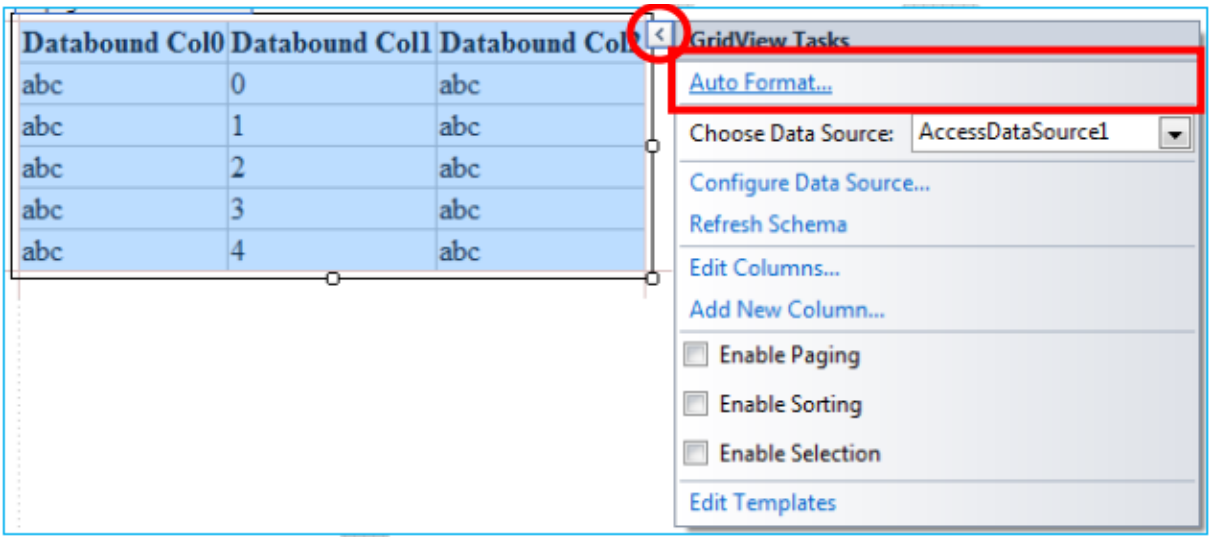
- DataFile (baza podataka, datoteka)
- SelectCommand (SQL SELECT upit, SELECT * FROM TableName)

```
<asp:AccessDataSource ID="AccessDataSource1" runat="server"
    DataFile="[redacted]" SelectCommand="[redacted]">
</asp:AccessDataSource>
```

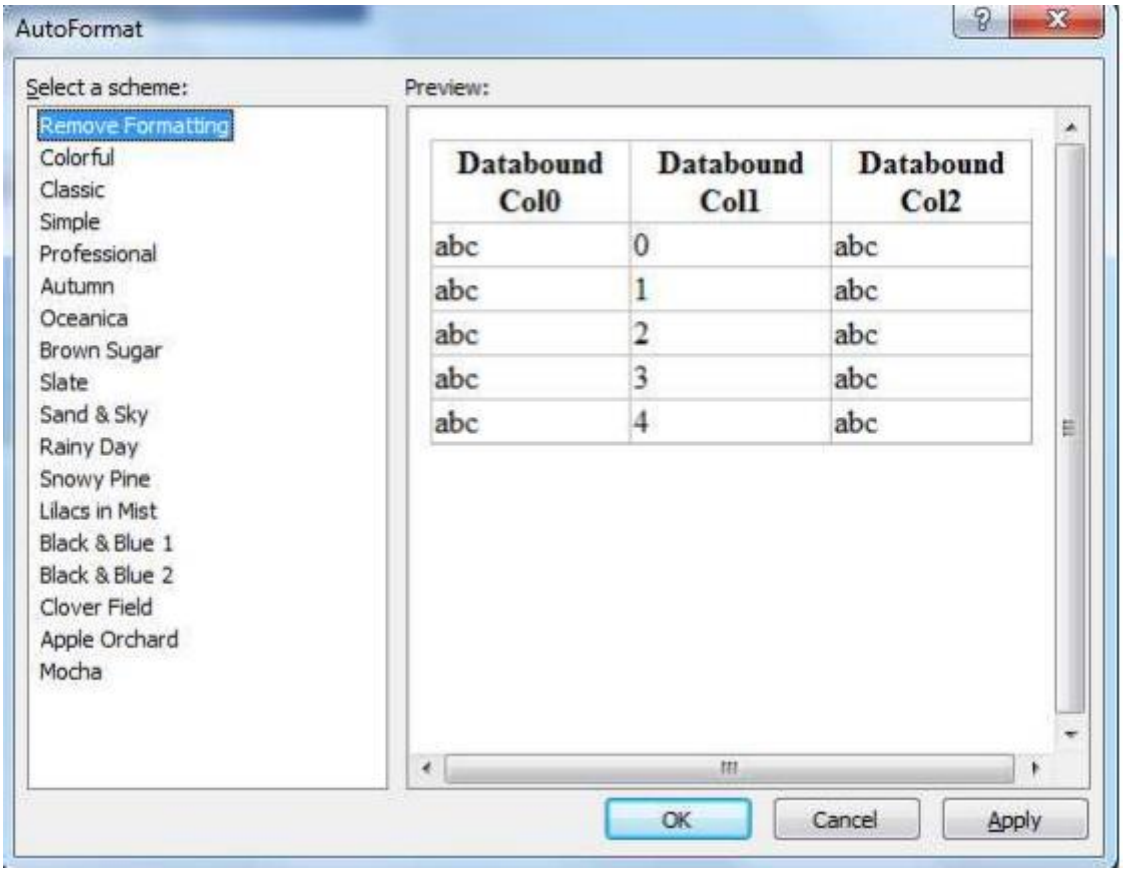
Slika 3. AccessDataSource

Na kontroli *GridView* potrebno je postaviti svojstvo *DataSourceID* na *AccessDataSource1* (ime *AccessDataSource* kontrole u ovom primjeru).

Za uređenje izgleda GridView kontrole mogu se iskoristiti postojeći predlošci koji su dostupni kroz dizajner web formi kao na donjoj slici (Slika 4, Slika 5), modificirati postojeći predlošci ili kreirati novi predložak.

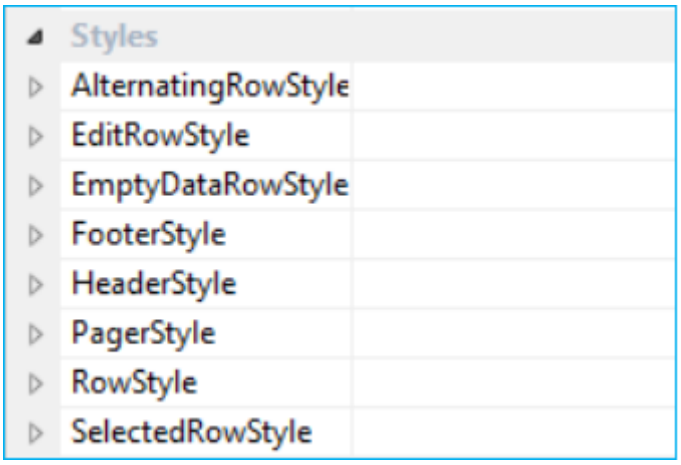


Slika 4. GridView, AutoFormat



Slika 5. GridView, odabir predloška za dizajn

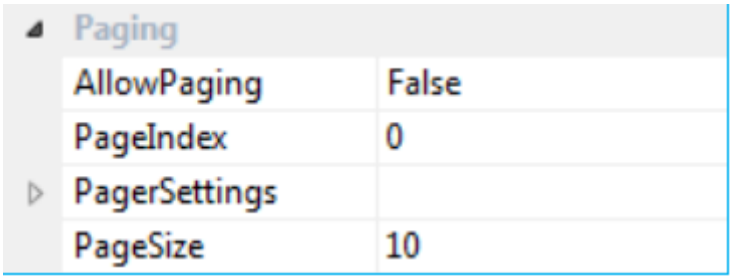
Za modificiranje postojećeg predloška (nakon odabira predloška) ili kreiranje novog predloška koristi se skupina Styles u kartici Properties (Slika 6).



Slika 6. GridView, Styles

Ukoliko se za rezultat SQL SELECT upita dobije velik broj zapisa, koriste se svojstva iz skupine *Paging* kontrole *GridView* kako bi se prilagodio prikaz zapisa (Slika 7).

Prije bilo kakve izmjene svojstava iz skupine *Paging*, potrebno je postaviti svojstvo *AllowPaging* na true inače se preostala svojstva neće primijeniti na kontroli *GridView*.



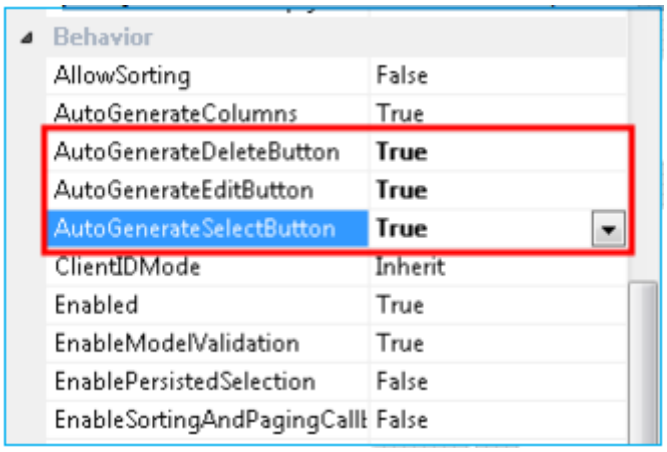
Slika 7. GridView, Paging

Klasa *GridView* omogućuje označavanje jednog retka u cijelog tablici. U tu svrhu potrebno je postaviti svojstvo *AutoGenerateSelectButton* na vrijednost True (Slika 8).

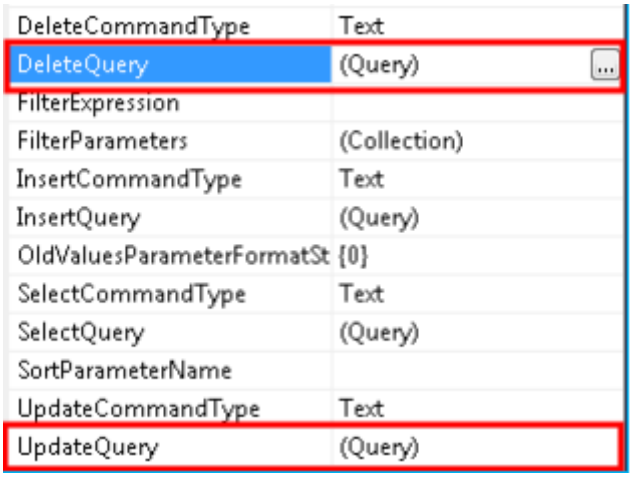
Klasa *GridView* omogućuje uz prikaz podataka i jednostavnu manipulaciju (kreiranje jednostavnih SQL upita) podacima u bazi podataka. Za tu svrhu potrebno je postaviti svojstva *AutoGenerateDeleteButton* i *AutoGenerateUpdateButton* na vrijednost *True* (Slika 8)

Za dodavanje akcija za *DeleteButton* i *UpdateButton* potrebno je dodati odgovarajući SQL upit kontroli AccessDataSource na svojstva *DeleteQuery* i *UpdateQuery* (Slika 9) i definirati na kontroli *GridView* koji stupac sadrži primarni ključa tablice koristeći svojstvo *DataKeyNames* (Slika 10).

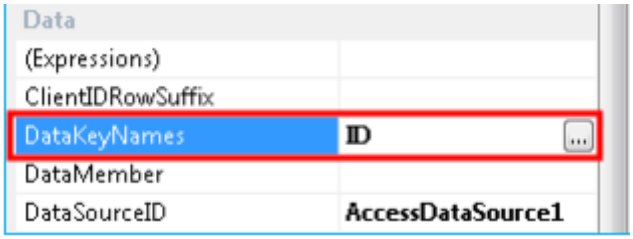
Za svojstva *DeleteQuery* i *UpdateQuery* potrebno je napisati SQL upite prema primjerima na slikama u nastavku (Slika 11, Slika 12).



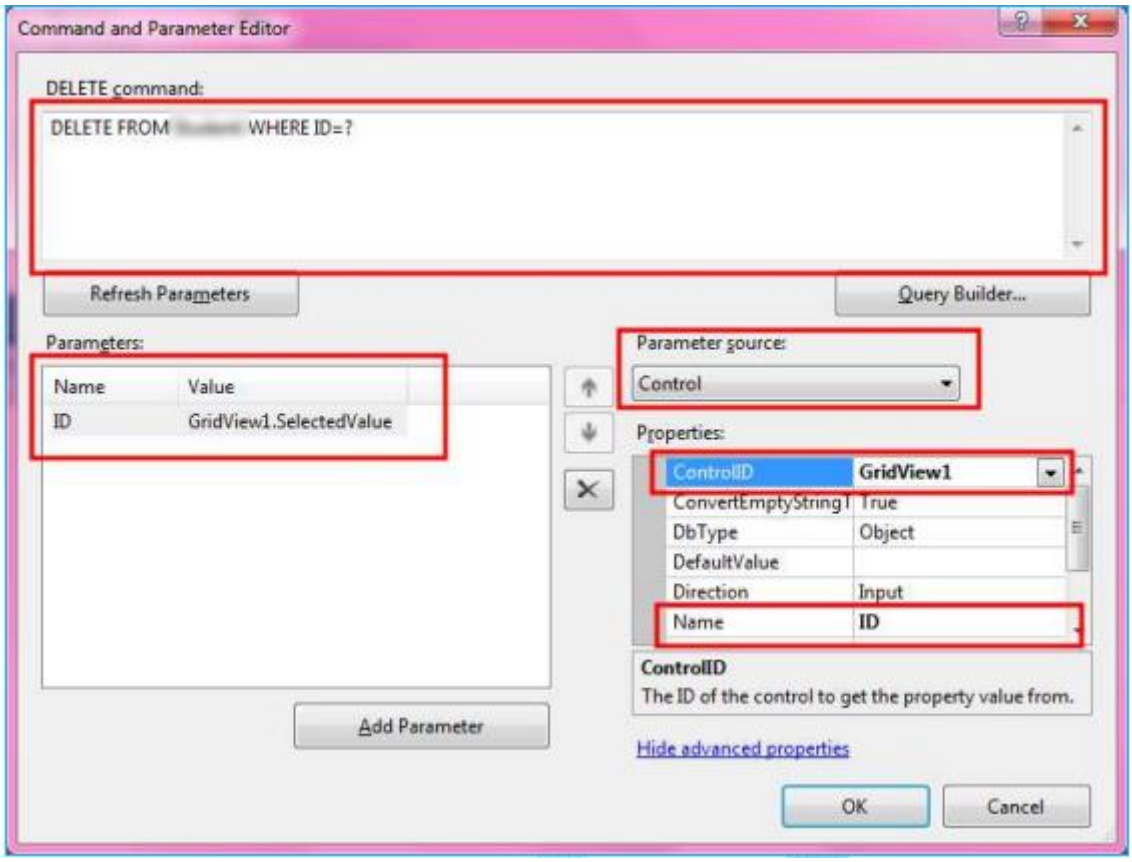
Slika 8. GridView, AutoGenerateButton



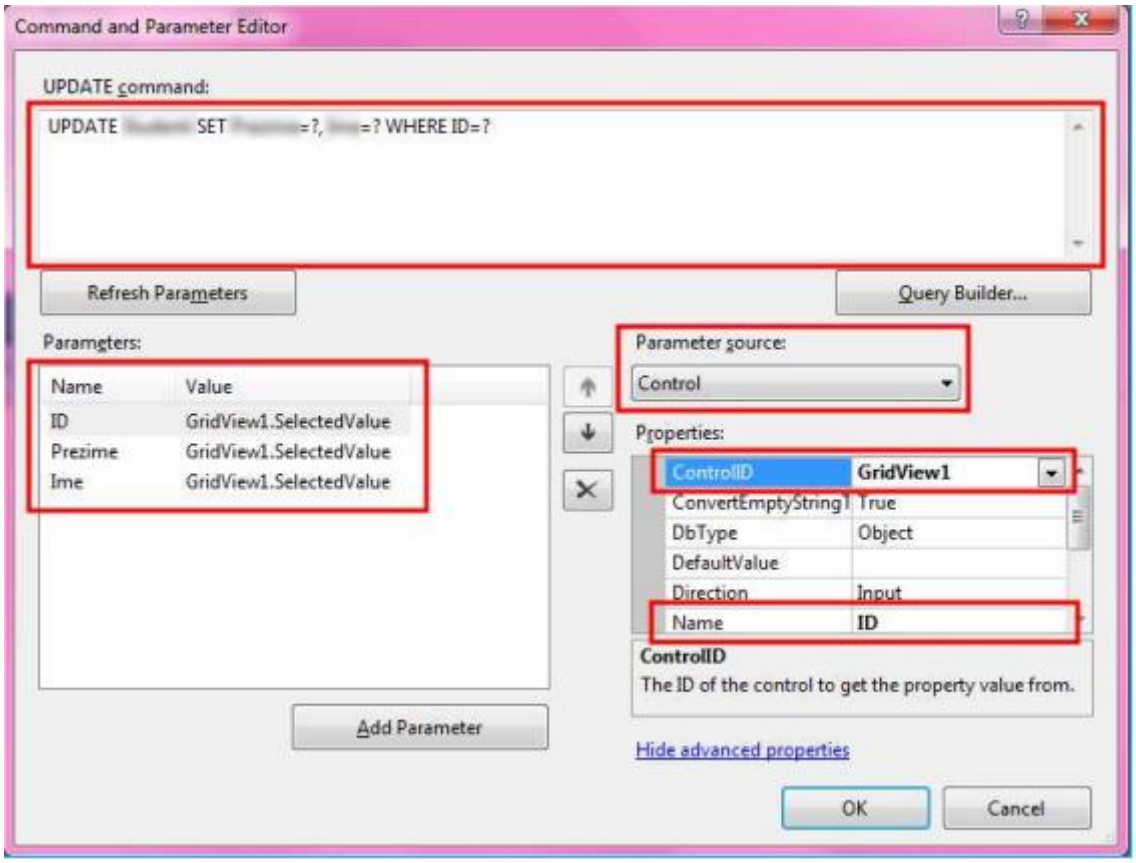
Slika 9. AccessDataSource, DeleteQuery i UpdateQuery



Slika 10. GridView, DataKeyNames



Slika 11. AccessDataSource, DeleteQuery



Slika 12. AccessDataSource, UpdateQuery

2. Primjeri komunikacije web aplikacije s bazom podataka

Sljedeći primjeri se mogu upotrijebiti ako se za povezivanje i komunikaciju s bazom podataka ne koriste ASP.NET kontrole iz skupine Data. Primjeri prikazuju naredbe kojima se otvara i zatvara konekcija na bazu podataka, šalje SQL upit prema bazi podataka i pohranjuje odgovor dobiven iz baze podataka.

Oznaka *ConnectionStrings* unutar konfiguracijske datoteke *web.config* sadrži podatke potrebne za spajanje na baze podataka.

Primjer 1 (zapis podataka za spajanje na bazu podataka u *web.config* datoteci):

```
<add name="konekcijaNaBazu"
     connectionString="Data Source=(LocalDB)\MSSQLLocalDB;
     AttachDbFilename=|DataDirectory|\TestDatabase.mdf;Integrated Security=True"
  providerName="System.Data.SqlClient" />
```

Primjer 2 (slanje SQL upita prema bazi podataka):

```
SqlConnection konekcija = new SqlConnection

(ConfigurationManager.ConnectionStrings["konekcijaNaBazu"].ConnectionString);

            // čitanje podataka iz konfiguracijske datoteke i kreiranje konekcije na
bazu podataka
            try
            {
                konekcija.Open(); // otvaranje konekcije na bazu podataka
                SqlCommand upit = new SqlCommand("SELECT * FROM Polaznik", konekcija);

                // kreiranje SQL upita za slanje prema bazi podataka
                SqlDataReader podaci = upit.ExecuteReader();

                // slanje SQL upita prema bazi podataka i spremanje rezultata
                while (podaci.Read())
                {
                    // naredba Read čita jedan zapis, te se zatim obavlja obrada
podataka
                    // naredba Read čita jedan zapis u svakoj iteraciji while petlje
sve dok se ne
                    // obrade svi dohvaćeni zapisi iz baze podataka
                }
                podaci.Close(); // prekid čitanja podataka dohvaćenih iz baze podataka
            }
            catch { }
            finally
            {
                konekcija.Close(); // zatvaranje konekcije prema bazi podataka
            }
```

Primjer 3 (parametrizirani SQL upit):

```
              SqlCommand upit = new SqlCommand
                    ("SELECT * FROM Polaznik WHERE Ime=@ime", konekcija);
              upit.Parameters.AddWithValue("@ime", this.TextBox1.Text);
              SqlDataReader podaci = upit.ExecuteReader();
```

Zadaci:

1) Kreirajte novu ASP.NET aplikaciju u programskom alatu Visual Studio.

2) Preuzmite s Gaudeamusa (gaudeamus.vvg.hr) bazu podataka za treću laboratorijsku vježbu.

3) Dodajte potrebne ASP.NET kontrole i metode za ispis podataka iz baze podataka iz tablice studenti.
4) Proširite tablicu studenti u bazi podataka s dodatnim stupcima za kontakt podatke.

5) Dodajte u ASP.NET aplikaciju potrebne kontrole i izvorni kod za promjenu podataka studenata.

6) Dodajte u ASP.NET aplikaciju mogućnost dodavanja novih studenata u bazu podataka.

7) Dodajte u ASP.NET aplikaciju mogućnost brisanja studenata iz baze podataka.

8) Dodajte u ASP.NET aplikaciju mogućnost sortiranja prikazanih podatak