UVOD:

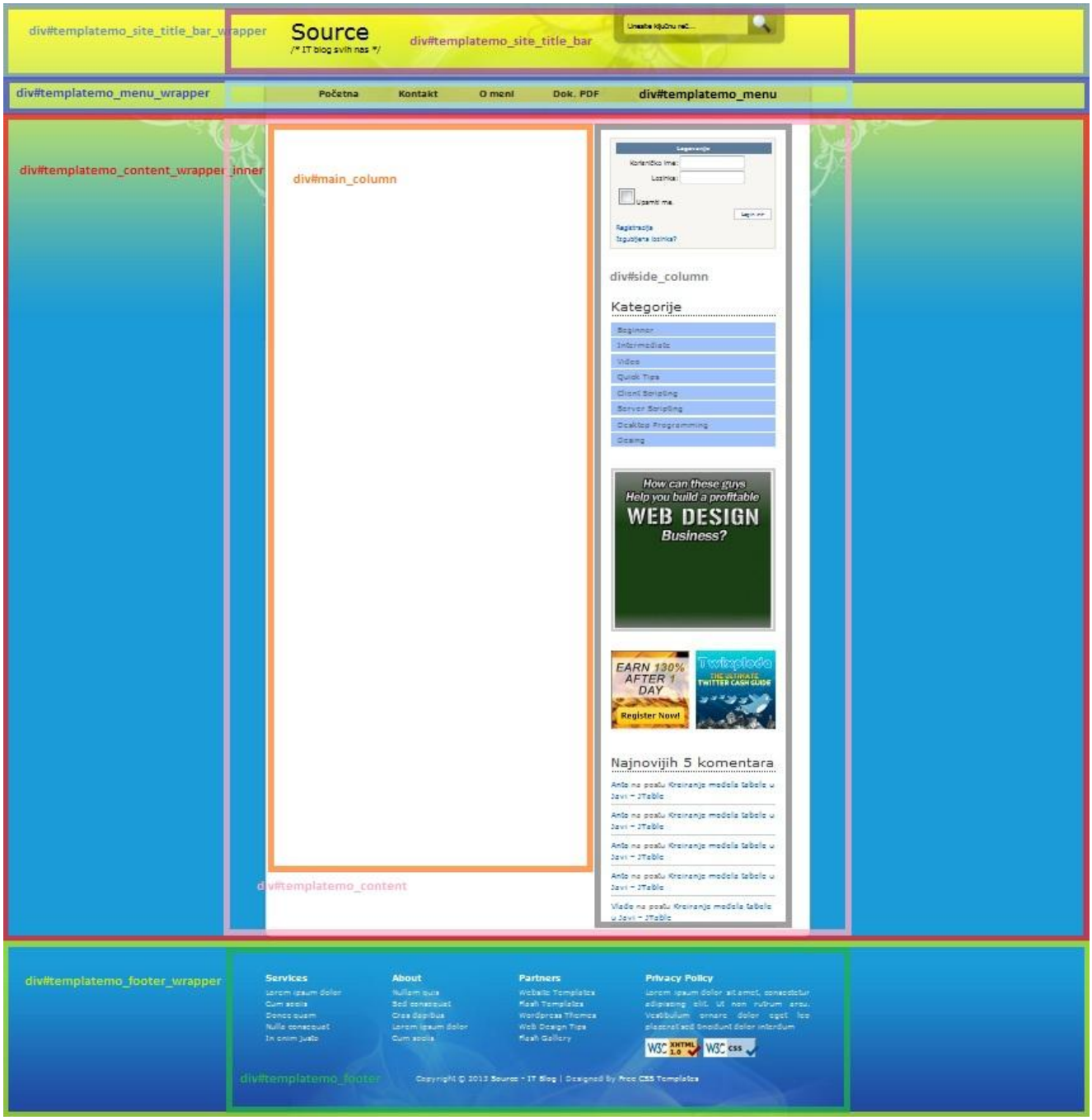
Korišćeni programski jezici: HTML, CSS, JavaScript, JQuery, AJAX, ASP.NET

Baza podataka: sql server 2008

JavaScript biblioteke: jquery.lightbox-0.5, jquery-latest, jquery – 1.9.1.min

Sajt sadrži sledeće funkcionalnosti: ***lightbox*** (na strani detaljnije.aspx, gde se prikazuje ceo sadržaj nekog posta (ako post sadrži slike klikom na istu otvara se lightbox)), ***kontakt formu*** (na stranici kontakt.aspx sa proverom unešenih podataka regularnim izrazima i na klijentskoj i na serverskoj strani sa slanjem e-maila na sourcekontakt@gmail.com), ***stranicu o autoru*** (omeni.aspx), ***straničenje*** (na stranama index.aspx i rezultat.aspx gde je prikaz po strani četiri posta), ***galeriju sa putanjama slika iz baze*** (realizovanu na strani detaljnije.aspx kod prikaza celokupnog posta, na detaljnije.aspx strani je realizovano i ostavljanje komentara na postu), ***pretragu sajta*** (pretraga se vrši po naslovu posta, takođe je omogućeno filtriranje postova po kategoriji kojoj pripadaju i autorima), ***logovanje uz registraciju*** (korišćene gotove "asp-ovske" kontrole, kada se korisnik registruje automatski mu se dodeljuje uloga korisnika. Admin je u mogućnosti da određene korisnike prebaci u ulogu autora), ***anketa sa ajax-om*** (nalazi se na index.aspx strani, anketa pravi kolačić kada se izvrši glasanje i upisuje se ip adresa u bazu podataka sa koje je korisnik glasao (ako se kolačić izbriše glasanje je ponovo omogućeno)), ***dinamički meni iz baze***.
Administracija sajta je urađena po datom kriterijumu (dinamičko dodavanje novog linka i nove stranice, dinamičko dodavanje, brisanje i editovanje sadržaja na sajtu (omogućeno za postove), dodavanje nove ankete, brisanje, modifikacija postojeće ankete).

Template:




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>IT Blog</title>
<meta name="keywords" content="yellow blog template, free html css layout" />
<meta name="description" content="yellow blog template, free html css layout from
TemplateMo.com" />
<link href="templatemo_style.css" rel="stylesheet" type="text/css" />

<script language="javascript" type="text/javascript">
    function clearText(field) {
        if (field.defaultValue == field.value) field.value = '';
        else if (field.value == '') field.value = field.defaultValue;
    }
</script>
    <asp:ContentPlaceHolder id="head" runat="server">
    </asp:ContentPlaceHolder>
</head>
<body>
    <form id="form1" runat="server">
    <div>
        <div id="templatemo_site_title_bar_wrapper">
       <div id="templatemo_site_title_bar">
           <div id="site_title">
            <h1><a href="index.aspx" target="_parent">Source
                <span>/* IT blog svih nas */</span>
            </a></h1>
        </div>

        <div id="search_box">
            <form action="#" method="get">

            <asp:TextBox ID="searchfield" runat="server" value="Unesite ključnu
reč..." name="q" size="10" onfocus="clearText(this)"
onblur="clearText(this)"></asp:TextBox>
            <asp:Button ID="searchbutton" runat="server" Text=""
onclick="searchbutton_Click" />

            </form>
        </div>

    </div>

</div>

<div id="templatemo_menu_wrapper">
       <div id="templatemo_menu">
           <ul>
            <asp:Label ID="LabelLinkoviSvi" runat="server"></asp:Label>
            <asp:Label ID="LabelLinkoviAdmin" runat="server"
Visible="False"></asp:Label>
            <asp:Label ID="LabelLinkoviKorisnici" runat="server"
Visible="False"></asp:Label>
            <asp:Label ID="LabelLinkoviAutori" runat="server"
Visible="False"></asp:Label>
        </ul>
    </div>
</div>

<div id="templatemo_content_wrapper_outer">
```

```
        <div id="templatemo_content_wrapper_inner">

        <div id="templatemo_content_wrapper">

             <div id="templatemo_content">
               <div class="content_bottom"></div>

                 <div id="main_column">

                     <asp:ContentPlaceHolder ID="ContentPlaceHolderSadrzaj"
runat="server">
                     </asp:ContentPlaceHolder>

                             </div> <!-- end of main column -->

                 <div id="side_column">

                     <asp:ContentPlaceHolder ID="ContentPlaceHolderOAutoru"
runat="server">
                     </asp:ContentPlaceHolder>


                 <div class="side_column_section">

                     <asp:LoginView ID="LoginView1" runat="server">
                         <AnonymousTemplate>
                             <asp:Login ID="Login1" runat="server"
BackColor="#F7F6F3" BorderColor="#E6E2D8"
                         BorderPadding="4" BorderStyle="Solid" BorderWidth="1px"
Font-Names="Verdana"
                         Font-Size="0.8em" ForeColor="#333333" Height="95px"
Width="250px" UserNameLabelText="Korisničko ime:" PasswordLabelText="Lozinka:"
RememberMeText="Upamti me." LoginButtonText="Login >>" TitleText="Logovanje"
CreateUserText="Registracija" CreateUserUrl="~/registracija.aspx"
PasswordRecoveryText="Izgubljena lozinka?"
PasswordRecoveryUrl="~/registracija.aspx">
                         <InstructionTextStyle Font-Italic="True" ForeColor="Black" />
                         <LoginButtonStyle BackColor="#FFFBFF" BorderColor="#CCCCCC"
BorderStyle="Solid"
                             BorderWidth="1px" Font-Names="Verdana" Font-Size="0.8em"
                             ForeColor="#284775" />
                         <TextBoxStyle Font-Size="0.8em" />
                         <TitleTextStyle BackColor="#5D7B9D" Font-Bold="True" Font-
Size="0.9em"
                             ForeColor="White" />
                         </asp:Login><br />
                             </AnonymousTemplate>
                             <LoggedInTemplate>
                                 <table style="width: 100%;">
                                     <tr>
                                         <td>
                                             <asp:Label ID="LabelKo" runat="server"
Text="Ulogovani ste kao:">
                                             </asp:Label><asp:LoginName ID="LoginName1"
runat="server" />
                                         </td>
                                     </tr>
```

```
                                <tr>
                                    <td>
                                        <asp:LoginStatus ID="LoginStatus1"
runat="server" LogoutText="Izlogujte se" />
                                    </td>
                                </tr>
                            </table>


                        </LoggedInTemplate>
                    </asp:LoginView>

                </div>

                <div class="cleaner_h30"> </div>

                <div class="side_column_section">
                    <asp:ContentPlaceHolder ID="ContentPlaceHolderAnketa"
runat="server">
                    </asp:ContentPlaceHolder>
                </div>

                <div class="cleaner_h30"> </div>

                <div class="side_column_section">
                    <h3>Kategorije</h3>
                    <ul>
                        <asp:Label ID="LabelKategorije"
runat="server"></asp:Label>
                     </ul>
                </div>

                <div class="cleaner_h30"> </div>

                <div class="side_column_section">

                <div class="ads_250_250">
                    <asp:AdRotator ID="AdRotatorBaneri250" runat="server"
                        DataSourceID="XmlDataSourceBaneri250" />
                    <asp:XmlDataSource ID="XmlDataSourceBaneri250" runat="server"
                        DataFile="~/baneri250.xml"></asp:XmlDataSource>
                </div>

                </div>

                <div class="cleaner_h30"> </div>

                <div class="side_column_section">

                     <div class="ads_125_125 right_padding_10">
                        <asp:AdRotator ID="AdRotatorBaner125l" runat="server"
                            DataSourceID="XmlDataSourceBaner125l" />
                        <asp:XmlDataSource ID="XmlDataSourceBaner125l"
runat="server"
                            DataFile="~/baneri125l.xml"></asp:XmlDataSource>
                    </div>

                    <div class="ads_125_125">
```

```
                        <asp:AdRotator ID="AdRotatorBaner125d" runat="server"
                            DataSourceID="XmlDataSourceBaner125d" />
                        <asp:XmlDataSource ID="XmlDataSourceBaner125d"
runat="server"
                            DataFile="~/baner125d.xml"></asp:XmlDataSource>
                    </div>

                </div>

                <div class="cleaner_h30"> </div>

                <div class="side_column_section">

                     <h3>Najnovijih 5 komentara</h3>
                    <asp:Label ID="LabelNKomentari" runat="server"></asp:Label>



                </div>

                </div> <!-- end of side column -->

              <div class="cleaner"></div>
            </div>

              <div class="cleaner"></div>
        </div>

        <div class="cleaner"></div>
    </div>

</div>

<div id="templatemo_footer_wrapper">

       <div id="templatemo_footer">


        <div class="section_w200">

              <h4>Services</h4>
              <ul class="footer_menu_list">
              <li><a href="#">Lorem ipsum dolor</a></li>
                <li><a href="#">Cum sociis</a></li>
                <li><a href="#">Donec quam</a></li>
                <li><a href="#">Nulla consequat</a></li>
                <li><a href="#">In enim justo</a></li>
            </ul>

        </div>

        <div class="section_w200">

               <h4>About</h4>
              <ul class="footer_menu_list">
                <li><a href="#">Nullam quis</a></li>
                <li><a href="#">Sed consequat</a></li>
                <li><a href="#">Cras dapibus</a></li>
```

```
              <li><a href="#">Lorem ipsum dolor</a></li>
                <li><a href="#">Cum sociis</a></li>
            </ul>

        </div>

        <div class="section_w200">

                    <h4>Partners</h4>
              <ul class="footer_menu_list">
              <li><a href="http://www.templatemo.com" target="_parent">Website
Templates</a></li>
                <li><a href="http://www.flashmo.com" target="_parent">Flash
Templates</a></li>
                <li><a href="http://www.layermo.com" target="_parent">Wordpress
Themes</a></li>
                    <li><a href="http://www.webdesignmo.com" target="_parent">Web
Design Tips</a></li>
                <li><a href="http://www.koflash.com" target="_blank">Flash
Gallery</a></li>
          </ul>

        </div>

        <div class="section_w260">

              <h4>Privacy Policy</h4>
              <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut non
rutrum arcu. Vestibulum ornare dolor eget leo placerat sed tincidunt dolor
interdum</p>

                    <div class="cleaner_h10"></div>

            <a href="http://validator.w3.org/check?uri=referer"><img
style="border:0;width:88px;height:31px" src="http://www.w3.org/Icons/valid-
xhtml10" alt="Valid XHTML 1.0 Transitional" width="88" height="31" vspace="8"
border="0" /></a>
                    <a href="http://jigsaw.w3.org/css-
validator/check/referer"><img style="border:0;width:88px;height:31px"
src="http://jigsaw.w3.org/css-validator/images/vcss-blue" alt="Valid CSS!"
vspace="8" border="0" /></a>

        </div>

        <div class="cleaner_h20"></div>

        <div class="section_w860">
              Copyright © 2013 <a href="index.aspx">Source - IT Blog</a> |
Designed by <a href="http://www.templatemo.com" target="_parent">Free CSS
Templates</a>
        </div>

    </div> <!-- end of footer -->
</div>
    </div>
    </form>
</body>
</html>
```

c#:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Web.Configuration;
using System.Data;
using System.Web.Services;

public partial class MasterPage : System.Web.UI.MasterPage
{
    private string connectionString =
WebConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString;


    protected void Page_Load(object sender, EventArgs e)
    {


        if (!this.IsPostBack)
        {
            string upit1 = "SELECT gm.* FROM GlavniMeni gm INNER JOIN
UlogeGlavniMeni ugm ON gm.idStrane=ugm.idStrane INNER JOIN Uloge u ON
u.idUloga=ugm.idUloga WHERE u.idUloga=4";
            SqlConnection con1 = new SqlConnection(connectionString);
            SqlCommand com1 = new SqlCommand(upit1, con1);
            using (con1)
            {
                con1.Open();
                SqlDataReader rid = com1.ExecuteReader();
                while (rid.Read())
                {
                    LabelLinkoviSvi.Text += "<li><a href=" + rid["Putanja"] +
"><span></span>" + rid["ImeStrane"] + "</a></li>";
                }
                rid.Close();
            }

            string upit3 = "SELECT gm.* FROM GlavniMeni gm INNER JOIN
UlogeGlavniMeni ugm ON gm.idStrane=ugm.idStrane INNER JOIN Uloge u ON
u.idUloga=ugm.idUloga WHERE u.idUloga=2";
            SqlConnection con3 = new SqlConnection(connectionString);
            SqlCommand com3 = new SqlCommand(upit3, con3);
            using (con3)
            {
                con3.Open();
                SqlDataReader rid = com3.ExecuteReader();
                while (rid.Read())
                {
                    LabelLinkoviAdmin.Text += "<li><a href=" + rid["Putanja"] +
"><span></span>" + rid["ImeStrane"] + "</a></li>";
                }
                rid.Close();
```

```
            }

            string upit5 = "SELECT gm.* FROM GlavniMeni gm INNER JOIN
UlogeGlavniMeni ugm ON gm.idStrane=ugm.idStrane INNER JOIN Uloge u ON
u.idUloga=ugm.idUloga WHERE u.idUloga=1";
            SqlConnection con5 = new SqlConnection(connectionString);
            SqlCommand com5 = new SqlCommand(upit5, con5);
            using (con5)
            {
                con5.Open();
                SqlDataReader rid = com5.ExecuteReader();
                while (rid.Read())
                {
                    LabelLinkoviKorisnici.Text += "<li><a href=" + rid["Putanja"]
+ "><span></span>" + rid["ImeStrane"] + "</a></li>";
                }
                rid.Close();
            }

            string upit6 = "SELECT gm.* FROM GlavniMeni gm INNER JOIN
UlogeGlavniMeni ugm ON gm.idStrane=ugm.idStrane INNER JOIN Uloge u ON
u.idUloga=ugm.idUloga WHERE u.idUloga=3";
            SqlConnection con6 = new SqlConnection(connectionString);
            SqlCommand com6 = new SqlCommand(upit6, con6);
            using (con6)
            {
                con6.Open();
                SqlDataReader rid = com6.ExecuteReader();
                while (rid.Read())
                {
                    LabelLinkoviAutori.Text += "<li><a href=" + rid["Putanja"] +
"><span></span>" + rid["ImeStrane"] + "</a></li>";
                }
                rid.Close();
            }

            string upit2 = "SELECT * FROM Kategorije";
            SqlConnection con2 = new SqlConnection(connectionString);
            SqlCommand com2 = new SqlCommand(upit2, con2);
            using (con2)
            {
                con2.Open();
                SqlDataReader rid = com2.ExecuteReader();
                while (rid.Read())
                {
                    LabelKategorije.Text += "<li><a
href='rezultat.aspx?k="+rid["idKategorija"]+"'>"+rid["Naziv"]+"</a></li>";
                }
                rid.Close();
            }

            string upit4 = "SELECT TOP (5) k.AutorKomentara, k.WebSite, p.Naslov,
p.idPost FROM Postovi AS p INNER JOIN Komentari AS k ON k.idPost = p.idPost ORDER
BY k.DatumKom DESC";
            SqlConnection con4 = new SqlConnection(connectionString);
            SqlCommand com4 = new SqlCommand(upit4, con4);
            using (con4)
            {
```

```
                con4.Open();
                SqlDataReader rid = com4.ExecuteReader();
                while (rid.Read())
                {
                    LabelNKomentari.Text += "<div class='recent_post'> "
                                            + " <a href=" + rid["WebSite"] + ">" +
rid["AutorKomentara"] + " </a> na postu  "
                                            + "<a href='detaljnije.aspx?post=" +
rid["idPost"] + "'> " + rid["Naslov"] + "</a> "
                                           + " </div>";
                }
                rid.Close();
            }
        }
    }

    protected void searchbutton_Click(object sender, EventArgs e)
    {
        Response.Redirect("~/rezultat.aspx?kr="+searchfield.Text);
    }
}
```

CSS:

```
/*
CSS Credit: http://www.templatemo.com
*/

body {
       margin: 0;
       padding: 0;
       line-height: 1.5em;
       font-family: Verdana, Geneva, sans-serif;
       font-size: 12px;
       color: #000000;
       background: #1b9bd7;
}

a:link, a:visited { color: #0066CC; text-decoration: none; font-weight: normal; }
a:active, a:hover { color: #FF0066; text-decoration: underline; }

p { margin: 0px; padding: 0px; text-align: justify; }

img { margin: 0px; padding: 0px; border: none; }

.cleaner { clear: both; width: 100%; height: 0px; font-size: 0px;  }

.cleaner_h10 { clear: both; width:100%; height: 10px; }
.cleaner_h20 { clear: both; width:100%; height: 20px; }
.cleaner_h30 { clear: both; width:100%; height: 30px; }
.cleaner_h40 { clear: both; width:100%; height: 40px; }
.cleaner_h50 { clear: both; width:100%; height: 50px; }
.cleaner_h60 { clear: both; width:100%; height: 60px; }

.fl { float: left; }
.fr { float: right; }
```

```
.m_right { margin-right: 30px; }

.button_01 a {
       display: block;
       width: 90px;
       height: 24px;
       padding: 6px 0 0 30px;
    background: url(images/templatemo_button_01.jpg) no-repeat;

       color: #191717;
       font-size: 14px;
       font-weight: bold;
       text-align: center;
       text-decoration: none;
}

h1 {
       margin: 0px;
       padding: 2px 0;
       font-size: 30px;
       font-weight: bold;
}

h2 {
       position: relative;
       height: 25px;
       margin: 0px;
       padding: 10px 0 0 0;
       font-size: 20px;
       font-weight: bold;
       text-align: center;
}

h3 {
       margin: 0 0 10px 0;
       padding: 2px 0 5px 0;
       font-size: 20px;
       font-weight: normal;
       color: #333;
       border-bottom: 1px dotted #333;
}

h4 {
       margin: 0 0 0 0;
       padding: 0 0 5px 0;
       font-size: 14px;
       color: #ffffff;
       font-weight: bold;
}

.image_wrapper {
       border: 1px solid #ffffff;
       margin-top: 3px;
       margin-bottom: 5px;
}

.fl_image {
       float: left;
```

```
	margin-right: 15px
}

.fr_image {
	float: right;
	margin-left: 15px
}

.list_01 {
	margin: 0 0 20px 40px;
	padding: 0px;
}

.list_01 li {
	margin: 0px;
	padding: 0 0 10px 0;
	list-style-image: url(images/templatemo_list_icon_01.jpg);
}

.list_01 li a {
	color: #ffffff;
}

#templatemo_site_title_bar_wrapper {
	width: 100%;
	height: 110px;
	background: url(images/templatemo_title_bar_repeat.jpg) repeat-x;
}

#templatemo_site_title_bar {
	width: 780px;
	height: 110px;
	padding: 0 90px;
	margin: 0 auto;
	background: url(images/templatemo_title_bar_bg_with_search.jpg) center no-
repeat;
}

#templatemo_site_title_bar  #site_title {
	float: left;
	padding-top: 30px;
	padding-bottom: 10px;
	width: 500px;
}

#site_title h1 a {
	margin: 0px;
	padding: 0px;
	font-size: 36px;
	color: #000000;
	font-weight: normal;
	text-decoration: none;
}

#site_title h1 a:hover {
	font-weight: normal;
	text-decoration: none;
}
```

```
#site_title h1 a span {
	display: block;
	margin-top: 10px;
	font-size: 14px;
	color: #000000;
}

#templatemo_site_title_bar #search_box {
	float: right;
	width: 240px;
	height: 30px;
	margin: 15px 0 0 0;
	padding: 0 15px;
}

#search_box form {
	float: left;
	width: 240px;
	height: 30px;
	margin: 0;
	padding: 0;
}

#searchfield {
	height: 16px;
	width: 182px;
	padding: 5px;
	margin: 0px;
	color: #000000;
	font-size: 12px;
	font-variant: normal;
	line-height: normal;
	background: none;
	border: none;
}

#searchbutton {
	height: 30px;
	width: 42px;
	margin: 0px;
	padding: 6px 0;
	cursor: pointer;
	font-size: 12px;
	text-align: center;
	vertical-align: bottom;
	white-space: pre;
	color: #ffffff;
	background: none;
	border: none;
}

#templatemo_menu_wrapper {
	width: 100%;
	height: 60px;
	background: url(images/templatemo_menu_bar_repeat.jpg) repeat-x;
}
```

```
#templatemo_menu {
	width: 860px;
	height: 60px;
	margin: 0 auto;
	padding: 0 50px;
	background: url(images/templatemo_menu_bar_bg.jpg) center no-repeat;
}

#templatemo_menu ul {
	width: 750px;
	overflow: hidden;
	padding: 15px 0 0 0;
	margin: 0px auto;
	list-style: none;
}

#templatemo_menu ul li {
	padding: 0px;
	margin: 0px;
	display: inline;
}

#templatemo_menu ul li a {
	float: left;
	display: block;
	height: 25px;
	width: 100px;
	padding: 5px 5px 0 0;
	margin: 0 10px;
	font-size: 14px;
	text-align: center;
	font-weight: bold;
	text-decoration: none;
	color: #292922;
	background: url(images/templatemo_menu_right.jpg) right top no-repeat;
	outline: none;
}

#templatemo_menu li a:hover, #templatemo_menu li .current {
	position: relative;
	background: url(images/templatemo_menu_hover_right.png) right no-repeat;
}

#templatemo_menu li a:hover span, #templatemo_menu li .current span {
	position: absolute;
	top: 0px;
	left: -5px;
	width: 5px;
	height: 30px;
	background: url(images/templatemo_menu_hover_left.png) right no-repeat;
}

#templatemo_content_wrapper_outer {
	width: 100%;
	background: url(images/templatemo_content_wrapper_outrer_repeat.jpg) top
repeat-x;
}
```

```
#templatemo_content_wrapper_inner {
	width: 100%;
	margin: 0 auto;
	background: url(images/templatemo_content_wrapper_outrer.jpg) center top
no-repeat;
}

#templatemo_content_wrapper {
	width: 960px;
	margin: 0 auto;
	background: url(images/templatemo_content_bg.png) center repeat-y;
}

#templatemo_content {
	position: relative;
	width: 780px;
	margin: 0 auto;
	padding: 40px 90px 0 90px;
	margin-bottom: 15px;
	background: url(images/templatemo_content_wrapper_inner.jpg) center top no-
repeat;
}

#templatemo_content .content_bottom {
	position: absolute;
	width: 960px;
	height: 15px;
	bottom: -15px;
	left: 0px;
	background:url(images/templatemo_content_bottom.png) no-repeat;
}

#main_column {
	float: left;
	width: 450px;
}

#side_column {
	float: right;
	width: 260px;
	padding: 0 15px;
}

#main_column .post_box {
	clear: both;
	margin-bottom: 50px;
}

#main_column .post_boxDetaljnije {
	clear: both;
	margin-bottom: 5px;
}

.post_box h2 {
	font-size: 20px;
	color: #4f4f4f;
	padding: 2px 0;
	margin-bottom: 10px;
```

```
	text-align: left;
}

.post_box .post_info {
	width: 100%;
	color: #666;
	background: #fffd41;
}

.post_box .post_info a {
	color: #666;
	text-decoration: none;
}

.post_box .post_info a:hover {
	color: #333;
}

.post_boxDetaljnije h2 {
	font-size: 20px;
	color: #4f4f4f;
	padding: 2px 0;
	margin-bottom: 10px;
	text-align: left;
}

.post_boxDetaljnije .post_info {
	width: 100%;
	color: #666;
	background: #fffd41;
}

.post_boxDetaljnije .post_info a {
	color: #666;
	text-decoration: none;
}

.post_boxDetaljnije .post_info a:hover {
	color: #333;
}

.post_info .post_date {
	float: left;
	padding: 2px 0;
	width: 34%;
	text-align: center;
	background: #c3e368;
}

.post_info .post_author {
	float: left;
	padding: 2px 0;
	width: 32%;
	text-align: center;
	background: #71d2ff;
}

.post_info .post_comment {
```

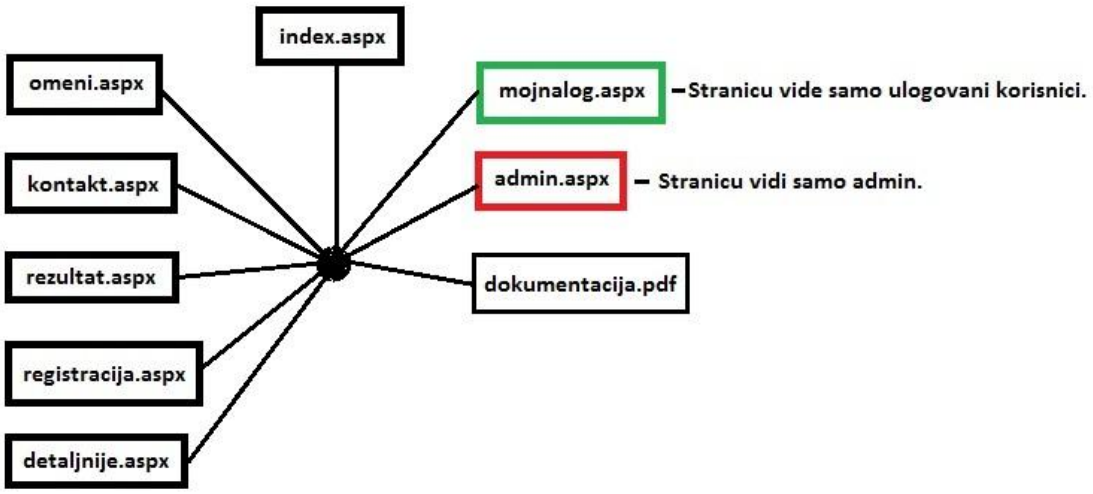
```
        float: left;
        padding: 2px 0;
        width: 34%;
        text-align: center;
        background: #88b4fe;
}

.post_box .post_body {
        clear: both;
        padding: 30px 0 20px 0;
        background: url(images/templatemo_post_body_bg.png) top center no-repeat;
}

.post_body img {

        margin-bottom: 15px;
}

.post_body p {
        margin-bottom: 10px;
}

.post_box .continue a {
        display: block;
        width: 120px;
        padding: 2px 0;
        text-align: center;
        color: #4c7fd4;
        font-weight: bold;
        background: #dfe0e1;
        border-left: 5px solid #88b4fe;
}

#side_column .ads_250_250 {
        border: 5px solid #ccc;
}

#side_column .side_column_section {
        clear: both;
        width: 260px;
}

.ads_125_125 {
        float: left;
        width: 125px;
        height: 125px;
        margin: 0 0 10px 0;
}

.right_padding_10 {
        margin-right: 10px;
}

.recent_post {
        color: #747474;
        margin-bottom: 5px;
        padding-bottom: 5px;
        border-bottom: 1px solid #ccc;
```

```
}

.recent_post h4 a {
	font-size: 12px;
	color: #3fa7d8;
	font-weight: bold;
}
.recent_post h4 a:hover {
	text-decoration: none;
	color: #FF0066;
}

.side_column_section ul {
	margin: 0px;
	padding: 0px;
	list-style: none;
}

.side_column_section li {
	margin: 0 0 3px 0;
	padding: 0px;
}

.side_column_section li a {
	display: block;
	width: 240px;
	padding: 2px 10px;
	color: #666;
	outline: none;
	background: #a0c3ff;
}

.side_column_section li a:hover {
	background: #d9ee9f;
	text-decoration: none;
}

#templatemo_footer_wrapper {
	width: 100%;
	background: url(images/templatemo_footer_wrapper.jpg) bottom repeat-x;
}

#templatemo_footer {
	width: 860px;
	min-height: 180px;
	margin: 0 auto;
	padding: 50px;
	color: #d9dcc6;
	background: url(images/templatemo_footer_bg.jpg) center bottom no-repeat;
}

.section_w860 {
	text-align: center;
	width: 920px;
}

.section_w860 a {
	color: #ffffff;
```

```
}

.section_w200 {
	float: left;
	width: 180px;
	padding: 0 20px 0 0;
}

.section_w260 {
	float: left;
	width: 260px;
	padding: 0 0 0 0;
}

.footer_menu_list {
	margin: 0px;
	padding: 0px;
	list-style: none;
}

.footer_menu_list li {
	margin: 0px;
	padding: 0px;
}

.footer_menu_list li a {
	color: #d9dcc6;
	font-weight: normal;
	text-decoration: none;
}

.footer_menu_list li a:hover {
	color: #ffffff;
	text-decoration: none;
}

.OZaAutora{ border: 1px solid #F7F6F3;}

.OZaAutora a{
    color: #666;
	text-decoration: none;
}

.OZaAutora a:hover{
    color: #333;
}

.HZaAutora{ background-color:#F7F6F3; text-align:center; margin: 5px;}

.SiTOA{ margin:5px;}

.SA {margin-right:5px;}

.PZaA {display:block; margin:0px; text-align: left;}

#strLinkovi_1 { text-align:center; }
#strLinkovi_1 a { margin-right:10px; }
.aktivnaStrana_1 {background-color:Silver;}
```

ORGANIZACIJA:

Organizaciona šema:



Mapa sajta:

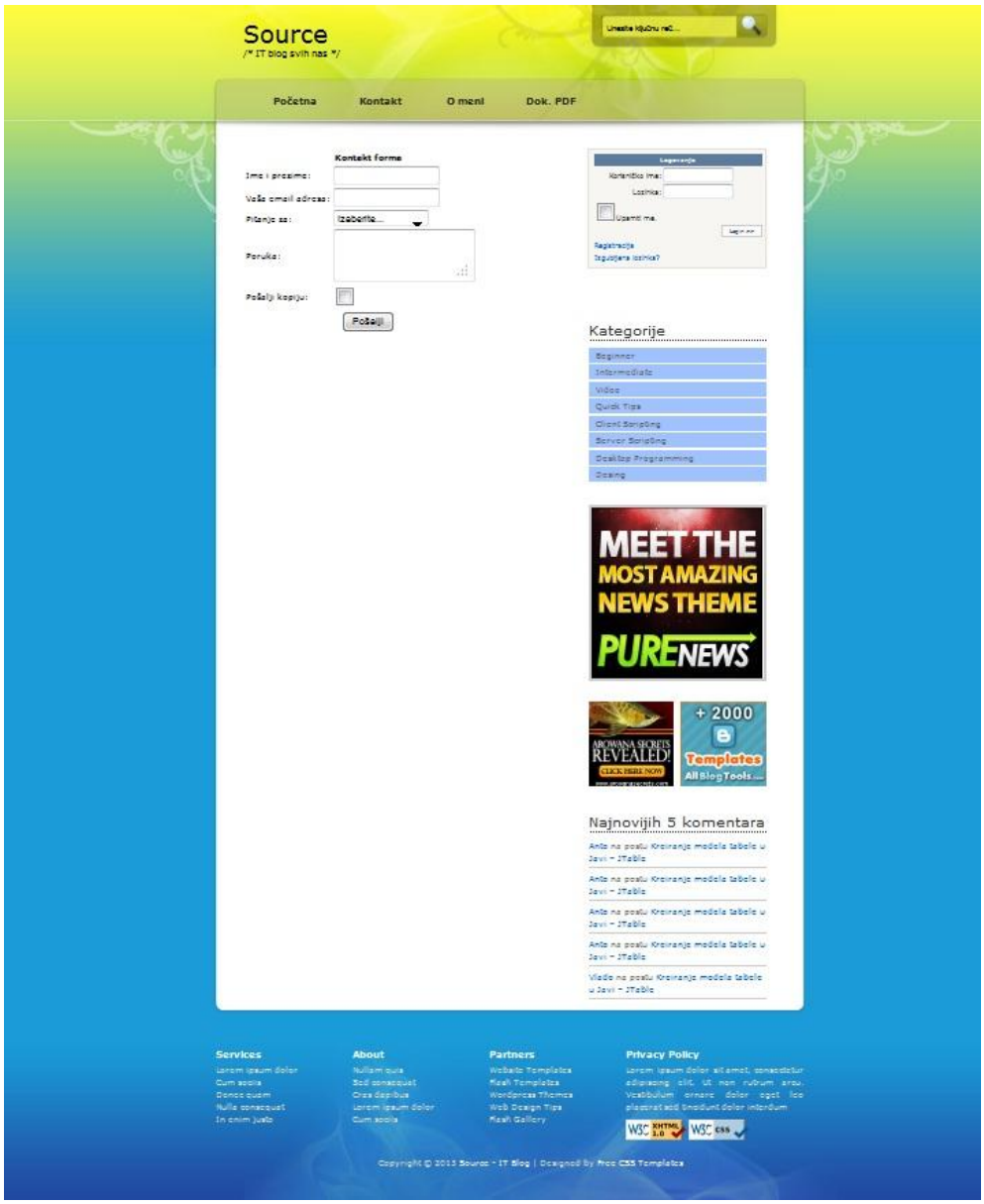
```
<?xml version="1.0" encoding="utf-8" ?>
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
 <siteMapNode url="" title=""  description="">
  <siteMapNode url="index.aspx" title="Početna"  description="" />
  <siteMapNode url="kontakt.aspx" title="Kontakt"  description="" />
  <siteMapNode url="omeni.aspx" title="O meni"  description="" />
  <siteMapNode url="dokumentacija.pdf" title="Dokumentacija" description="" />
  <siteMapNode url="mojnalog.aspx" title="Moj nalog"  roles="*" description="" />
 <siteMapNode url="admin.aspx" title="Admin panel" roles="admin" description="" />
    </siteMapNode>
</siteMap>
```

Slike stranica i opis funkcionalnosti:

index.aspx :

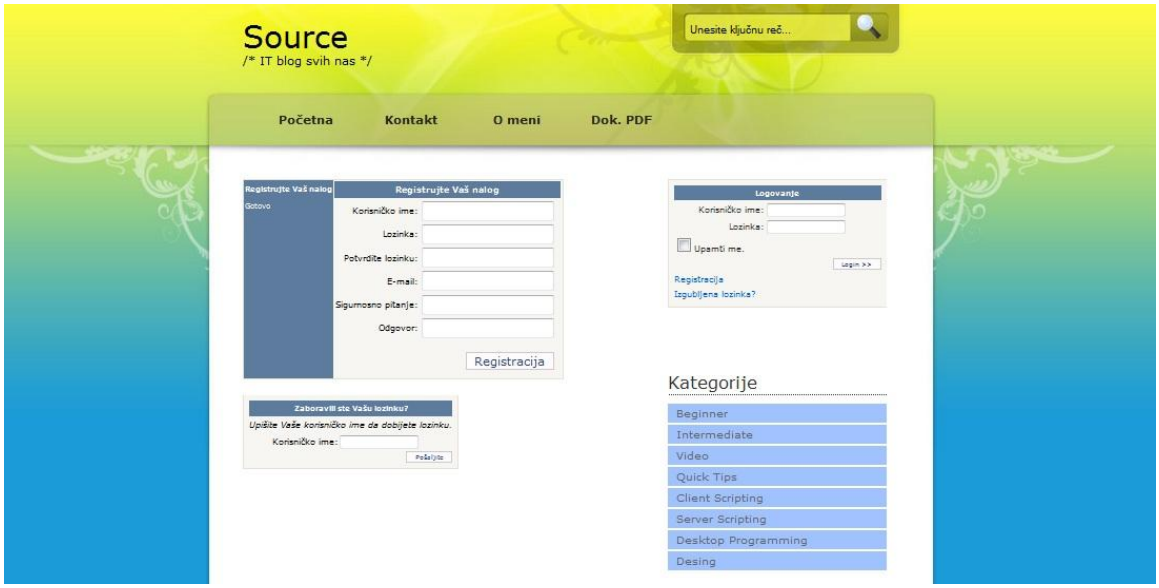
Na strani index.aspx se prikazuju svi postovi koji postoje u bazi podataka. Postovi su sortirani po datumu postavljanja. Ubačeno je **straničenje** pa se na svakoj strani prikazuju po četiri posta. Klikom na naslov, sliku ili link "Pročitajte više" otvara se strana detaljnije.aspx na kojoj se vidi ceo post. Na ovoj strani je realizovana i **ajax anketa**. **Logovanje je omogućeno na svim stranama koje mogu videti neregistrovani korisnici. Pretraga i filtriranje je takođe omogućeno na svim stranama.**

kontakt.aspx:



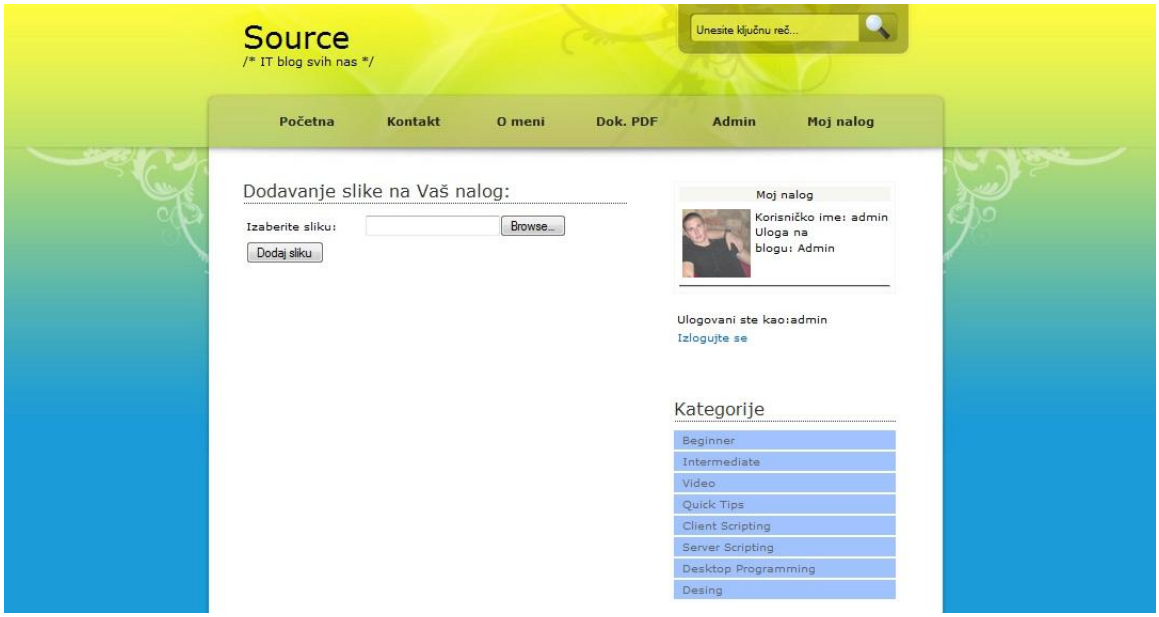
Kontakt forma sa proverom na klijentskoj i serverskoj strani i slanje e-mail-a na sourcekontakt@gmail.com.

registracija.aspx:



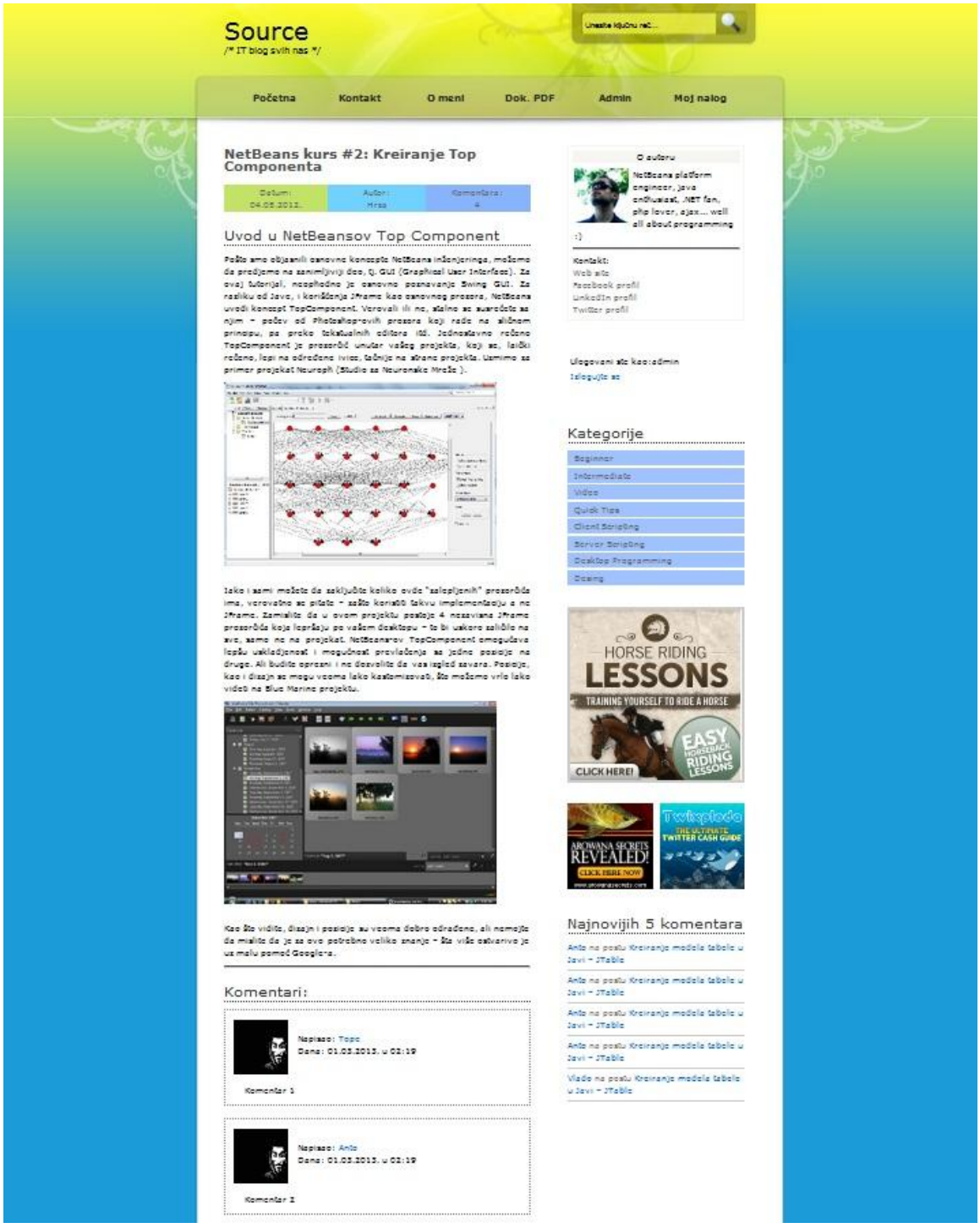
Registracija realizovana pomoću asp-ovskih kontrola. Takođe je omogućena i promena zaboravljene lozinke.

mojnalog.aspx:



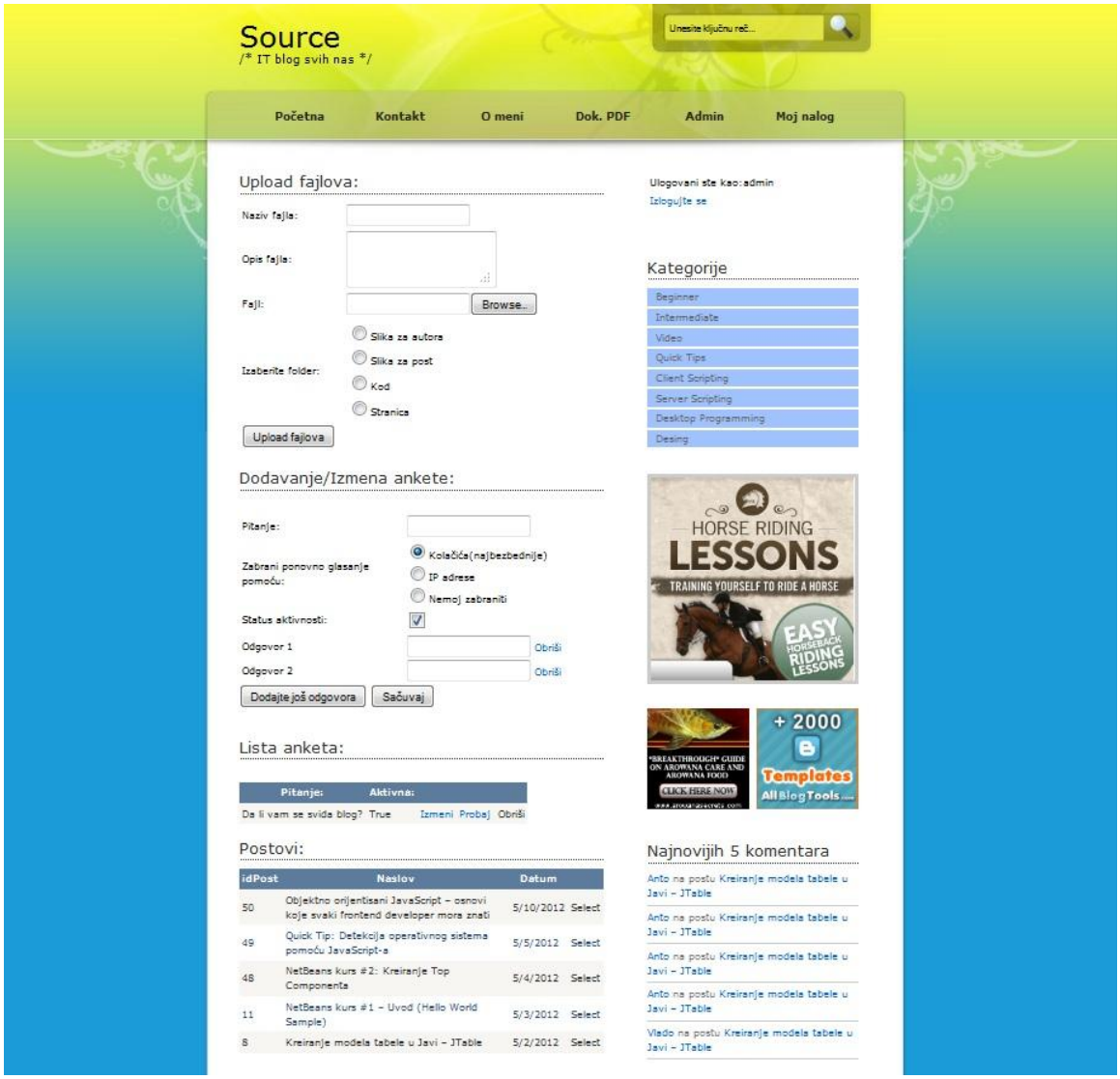
Na ovoj stranici korisnik može postaviti na svoj nalog sliku koja će se videti kada korisnik bude ostavljao komentare na postovima.

detaljnije.aspx:



Korisnik na ovoj strani može pogledati ceo tekst posta sa pratećim kodom i slikama kao i da ostavlja komentare. Kada korisnik ostavi komentar prikaže se njegova slika i podaci kada je komentar napisan sa pratećim tekstom.

admin.aspx:



Strana namenjena adminu. Ceo sadržaj se može izmeniti, obrisati i dodati na ovoj strani.

KODOVI:

Ubacio sam samo razliku u odnosu na kod template-a koji je ubečen ranije u dokumentaciji.

```
<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
<link href="poll.css" rel="stylesheet" type="text/css" />
<script src="scripts/jquery-latest.js" type="text/javascript"></script>
<script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>
    <script type="text/javascript">
       $(document).ready(function() {
              var prikazPoStrani_1 = 4;
              var brProizvoda_1 = $('#stranicenje').children().size();
              var brStrana_1 = Math.ceil(brProizvoda_1/prikazPoStrani_1);
              $('#trenutnaStrana_1').val(0);
              $('#prikazPoStrani_1').val(prikazPoStrani_1);
              var strLinkoviHtml_1 = '<a class="prethodniLink_1"
href="javascript:prethodna_1();"><<<</a>';
              var trenutniLink_1 = 0;
              while(brStrana_1 > trenutniLink_1){
                     strLinkoviHtml_1 += '<a class="linkStrane_1"
href="javascript:idiNaStranu_1(' + trenutniLink_1 +')" longdesc="' +
trenutniLink_1 +'">' + (trenutniLink_1 + 1) +'</a>';
                     trenutniLink_1++;
              }
              strLinkoviHtml_1 += '<a class="sledeciLink_1"
href="javascript:sledeca_1();">>>></a>';
              $('#strLinkovi_1').html(strLinkoviHtml_1);
              $('#strLinkovi_1 .linkStrane_1:first').addClass('aktivnaStrana_1');
              $('#stranicenje').children().css('display', 'none') ;
              $('#stranicenje').children().slice(0,
prikazPoStrani_1).css('display', 'block');
       });

       function prethodna_1(){
              novaStrana_1 = parseInt($('#trenutnaStrana_1').val()) - 1;
              if($('.aktivnaStrana_1').prev('.linkStrane_1').length==true){
                     idiNaStranu_1(novaStrana_1);
              }
       };

       function sledeca_1(){
              novaStrana_1 = parseInt($('#trenutnaStrana_1').val()) + 1;
              if($('.aktivnaStrana_1').next('.linkStrane_1').length==true){
                     idiNaStranu_1(novaStrana_1);
              }
       };

       function idiNaStranu_1(brStrane_1){
              var prikazPoStrani_1 = parseInt($('#prikazPoStrani_1').val());
              kreni_1 = brStrane_1 * prikazPoStrani_1;
              zavrsi_1 = kreni_1 + prikazPoStrani_1;
              $('#stranicenje').children().css('display', 'none').slice(kreni_1,
zavrsi_1).css('display', 'block');
```

```
            $('.linkStrane_1[longdesc=' + brStrane_1
+']').addClass('aktivnaStrana_1').siblings('.aktivnaStrana_1').removeClass('aktivn
aStrana_1');
            $('#trenutnaStrana_1').val(brStrane_1);
};

$(document).ready(function () {
    var imgPoll = new Image();
    imgPoll.src = 'images/red-bar.png';

    if ($("#divVoted").length > 0)
    {
        animateResults();
    }
    else {
        $("#rdoPoll0").attr("checked", "checked");
        $("#btnSubmit").click(function () {
            $("#divPoll").css("cursor", "wait");
$("#btnSubmit").attr("disabled", "true")
            var pID = $("input[id$=hidPollID]").val();
            var cID = $("input[name='rdoPoll']:checked").val();
            var data = "{'pID':'" + pID + "', 'cID':'" + cID + "'}";
            $.ajax(
                {
                    type: "POST",
                    url: "index.aspx/UpdatePollCount",
                    data: data,
                    contentType: "application/json; charset=utf-8",
                    dataType: "json",
                    success: function (msg)
                    {

                        $("#divPoll").css("cursor", "default");
$("#btnSubmit").attr("disabled", "false")

$("div[id$=divAnswers]").fadeOut("fast").html(msg.d).fadeIn("fast", function () {
animateResults(); });
                    }
                });
        });
    }

    function animateResults() {
        $("div[id$=divAnswers] img").each(function () {
            var percentage = $(this).attr("val");
            $(this).css({ width: "0%" }).animate({ width: percentage }, 'slow');
        });
    }
});

</script>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolderSadrzaj"
Runat="Server">
    <asp:Label ID="LabelPostovi" runat="server"></asp:Label>
</asp:Content>
<asp:Content ID="Content3" ContentPlaceHolderID="ContentPlaceHolderOAutoru"
Runat="Server">
```

```
</asp:Content>
<asp:Content ID="Content4" ContentPlaceHolderID="ContentPlaceHolderAnketa"
runat="server">
    <h3>Asp.Net AJAX Anketa</h3>
    <br />
    <div runat="server" id="divMsg" class="mInfo" visible="false" />
    <div id="divPoll" class="poll-box" runat="server">
        <div class="poll-question">
            <asp:Literal ID="litQuestion" runat="server" />
        </div>
        <div id="divAnswers" runat="server">
            <asp:Repeater runat="server" ID="rptChoices">
                <ItemTemplate>
                    <p>
                        <label>
                            <input type="radio" value='<%# Eval("PollChoiceID")
%>' name="rdoPoll" id='rdoPoll<%# Container.ItemIndex %>' />
                            <%# Eval("Choice") %>
                        </label>
                    </p>
                </ItemTemplate>
                <FooterTemplate>
                    <p>
                        <input type="button" value="Glasajte" class="submit"
id="btnSubmit" />
                    </p>
                </FooterTemplate>
            </asp:Repeater>
        </div>
    </div>
    <asp:HiddenField runat="server" ID="hidPollID" />
</asp:Content>
```

c#:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.Configuration;
using System.Data.SqlClient;
using System.Data;
using System.Web.Services;

public partial class index : System.Web.UI.Page
{
    private string connectionString =
WebConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString;

    protected void Page_Load(object sender, EventArgs e)
    {
        this.Title = "IT Blog | Početna";

        divMsg.Visible = false;
```

```
        if (!this.IsPostBack)
        {


            string[] brk = new string[50];
            string upit2 = "SELECT COUNT(k.idPost) AS BrK FROM Komentari AS k
RIGHT OUTER JOIN Postovi AS p ON p.idPost = k.idPost GROUP BY k.idPost, p.Datum
ORDER BY p.Datum DESC";
            SqlConnection con2 = new SqlConnection(connectionString);
            SqlCommand com2 = new SqlCommand(upit2, con2);
            using (con2)
            {
                int i = 0;
                con2.Open();
                SqlDataReader rid = com2.ExecuteReader();
                while (rid.Read())
                {
                    brk[i] = rid["BrK"].ToString();
                    i++;
                }
                rid.Close();
            }



            string upit = "SELECT p.*, a.* FROM Postovi p INNER JOIN Autori a ON
p.idAutor=a.idAutor ORDER BY p.Datum DESC";
            SqlConnection con = new SqlConnection(connectionString);
            SqlCommand com = new SqlCommand(upit, con);
            using (con)
            {
                con.Open();
                SqlDataReader rid = com.ExecuteReader();
                LabelPostovi.Text = "<input type='hidden' id='trenutnaStrana_1' />
"
                                          + " <input type='hidden'
id='prikazPoStrani_1' /> "
                                    + " <div id='stranicenje'> ";
                int i = 0;
                while (rid.Read())
                {
                    DateTime datum = (DateTime)rid["Datum"];
                    if (rid["OpisPosta"].ToString().Length > 230)
                    {
                        LabelPostovi.Text += "<div class='post_box'><h2><a
href='detaljnije.aspx?post=" + rid["idPost"] + "'>" + rid["Naslov"] +
"</a></h2><br/><div class='post_info'><div class='post_date'>Datum:<br/> " +
datum.ToString("dd.MM.yyyy.") + "</div> "
                                                 + " <div
class='post_author'>Autor:<br/> <a href='rezultat.aspx?a=" + rid["idAutor"] + "'>"
+ rid["KorisničkoIme"] + "</a></div><div class='post_comment'>Komentara:<br/>" +
brk[i] + " "
                                                 + " </div><div
class='cleaner'></div></div><div class='post_body'><table><tr><td><a
href='detaljnije.aspx?post=" + rid["idPost"] + "'><img style='margin-right:15px;
margin-top:10px;' src=" + rid["Slika"] + " alt=" + rid["OpisSlike"] + "
/></a></td><td> "
```

```
                                            + " <p>" +
rid["OpisPosta"].ToString().Substring(0, 230) + "...</p></div><div
class='continue'><a href='detaljnije.aspx?post=" + rid["idPost"] + "'>Pročitajte
više</a></td></tr></table></div></div>";
                        i++;
                    }
                    else
                    {
                        LabelPostovi.Text += "<div class='post_box'><h2><a
href='detaljnije.aspx?post=" + rid["idPost"] + "'>" + rid["Naslov"] +
"</a></h2><br/><div class='post_info'><div class='post_date'>Datum:<br/> " +
datum.ToString("dd.MM.yyyy.") + "</div> "
                                            + " <div
class='post_author'>Autor:<br/> <a href='rezultat.aspx?a=" + rid["idAutor"] + "'>"
+ rid["KorisničkoIme"] + "</a></div><div class='post_comment'>Komentara:<br/>" +
brk[i] + " "
                                            + " </div><div
class='cleaner'></div></div><div class='post_body'><table><tr><td><a
href='detaljnije.aspx?post=" + rid["idPost"] + "'><img style='margin-right:15px;
margin-top:10px;' src=" + rid["Slika"] + " alt=" + rid["OpisSlike"] + "
/></a></td><td> "
                                            + " <p>" + rid["OpisPosta"] +
"</p></div><div class='continue'><a href='detaljnije.aspx?post=" + rid["idPost"] +
"'>Pročitajte više</a></td></tr></table></div></div>";
                        i++;
                    }

                }
                LabelPostovi.Text += " </div><div id='strLinkovi_1'></div> ";
                rid.Close();
            }

            if (User.IsInRole("admin"))
            {
                Master.FindControl("LabelLinkoviAdmin").Visible = true;
            }
            else
            {
                Master.FindControl("LabelLinkoviAdmin").Visible = false;
            }
            if (User.IsInRole("korisnik"))
            {
                Master.FindControl("LabelLinkoviKorisnici").Visible = true;
            }
            else
            {
                Master.FindControl("LabelLinkoviKorisnici").Visible = false;
            }
            if (User.IsInRole("autor"))
            {
                Master.FindControl("LabelLinkoviAutori").Visible = true;
            }
            else
            {
                Master.FindControl("LabelLinkoviAutori").Visible = false;
            }
```

```
            if (!String.IsNullOrEmpty(Request.QueryString["pid"]))
               {
                   hidPollID.Value = Request.QueryString["pid"];
               }
               else
               {
                   Poll objPoll = new Poll();
                   int pollID = objPoll.getRandomActivePollID();

                   if (pollID > 0)
                       hidPollID.Value = pollID.ToString();
                   else
                   {
                       divMsg.InnerHtml = "<b>Obaveštenje: </b>Nema
aktivnih anketa.";
                       divPoll.Visible = false;
                       divMsg.Visible = true;
                       return;
                   }
               }
               showPoll();
           }
       }

        private void showPoll()
        {
            Poll objPoll = new Poll();
            DataSet dsPoll =
objPoll.SelectPoll(int.Parse(hidPollID.Value));

            if (dsPoll.Tables[0].Rows.Count > 0)
            {
                litQuestion.Text =
dsPoll.Tables[0].Rows[0]["Question"].ToString();

                string blockMode =
dsPoll.Tables[0].Rows[0]["BlockMode"].ToString();

                bool isPolled = isPolled =
CheckAlreadyPolled(blockMode);


                if (isPolled)
                {
                    divAnswers.InnerHtml = getResultHTML(dsPoll);
```

```
                divAnswers.InnerHtml += "<div class='poll-total'
id='divVoted'>Vaš glas je upisan.</div>";
            }
            else
            {
                rptChoices.DataSource = dsPoll.Tables[1];
                rptChoices.DataBind();
                rptChoices.Visible = rptChoices.Items.Count > 0;
            }
        }
        else
        {
            divMsg.InnerHtml = "<b>Obaveštenje: </b>Anketu koju
ste tražili nije nađena.";
            divMsg.Visible = true;
        }
    }

    private bool CheckAlreadyPolled(string blockMode)
    {
        bool isPolled = false;

        if (blockMode == Poll.BlockMode.IP_ADDRESS.ToString())
        {
            string ip =
HttpContext.Current.Request.ServerVariables["REMOTE_ADDR"];
            Poll objPoll = new Poll();
            int id =
objPoll.SelectPollIP(int.Parse(hidPollID.Value), ip);
            if (id > 0) isPolled = true;
        }
        else if (blockMode == Poll.BlockMode.COOKIE.ToString())
        {
            if (Request.Cookies["Poll"] != null &&
Request.Cookies["Poll"]["ID"] != null)
            {
                string commaSeperatedPollIDs =
Request.Cookies["Poll"]["ID"];

                string[] pollIDs =
commaSeperatedPollIDs.Split(",".ToCharArray());

                foreach (string pID in pollIDs)
                {
                    if (pID == hidPollID.Value)
                    {
```

```
                    isPolled = true;
                    break;
                }
            }
        }
    }

    return isPolled;
}

[WebMethod]
public static string UpdatePollCount(int pID, int cID)
{
    Poll objPoll = new Poll();
    objPoll.UpdateChoiceVote(cID);

    string ip =
HttpContext.Current.Request.ServerVariables["REMOTE_ADDR"];
    objPoll.InsertPollIP(pID, ip);

    HttpCookie pollCookie;
    string valueToStore = "";

    if (HttpContext.Current.Request.Cookies["Poll"] != null
&& HttpContext.Current.Request.Cookies["Poll"]["ID"] != null)
    {
        pollCookie =
HttpContext.Current.Request.Cookies["Poll"];
        valueToStore =
HttpContext.Current.Request.Cookies["Poll"]["ID"] + "," +
pID.ToString();
    }
    else
    {
        pollCookie = new HttpCookie("Poll");
        valueToStore = pID.ToString();
    }
    pollCookie.Values["ID"] = valueToStore;
    pollCookie.Expires = DateTime.MaxValue;
    HttpContext.Current.Response.Cookies.Add(pollCookie);

    DataSet dsPoll = objPoll.SelectPoll(pID);
    return getResultHTML(dsPoll);
}

private static string getResultHTML(DataSet dsPoll)
```

```
    {
        int totalVotes =
int.Parse(dsPoll.Tables[1].Compute("Sum(VoteCount)",
String.Empty).ToString());
        System.Text.StringBuilder sbResult = new
System.Text.StringBuilder();

        foreach (DataRow dr in dsPoll.Tables[1].Rows)
        {
            decimal percentage = 0;
            if (totalVotes > 0)
                percentage =
decimal.Round((decimal.Parse(dr["VoteCount"].ToString()) /
decimal.Parse(totalVotes.ToString())) * 100,
MidpointRounding.AwayFromZero);

            string alt = dr["VoteCount"].ToString() + " votes out
of " + totalVotes.ToString();

            sbResult.Append("<label class='poll-
result'>").Append(dr["Choice"]).Append("
(").Append(dr["VoteCount"]).Append(" glas/a -
").Append(percentage).Append("%)</label>");
            sbResult.Append("<div class='poll-chart'><img
src='images/red-bar.png' width='0%'
val='").Append(percentage).Append("%' height='15px'
alt='").Append(alt).Append("' title='").Append(alt).Append("' />
").Append("</div>");
        }
        sbResult.Append("<div class='poll-total'>Ukupno glasova:
").Append(totalVotes).Append("</div>");
        return sbResult.ToString();
    }
}
```

kontakt.aspx:

```
<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
<script type="text/javascript">
    function ValidacijaOsoba(objSource, objArgs) {
        if (objArgs.Value != "0") {
            objArgs.IsValid = true;
        } else {
            objArgs.IsValid = false;
        }

    }
```

```
    function ValidacijaPoruka(objSource, objArgs) {

        if (objArgs.Value.length > 10) {
            objArgs.IsValid = true;
        } else {
            objArgs.IsValid = false;
        }
    }
</script>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolderSadrzaj"
Runat="Server">
 <table class="tabela">
        <tr>
            <th colspan="3">
                Kontakt forma
            </th>

        </tr>
        <tr>
            <td>
                <asp:Label ID="LabelImePrezime" runat="server" Text="Ime i
prezime:" AssociatedControlID="TextBoxImePrezime" CssClass="label"></asp:Label>
            </td>
            <td>
                <asp:TextBox ID="TextBoxImePrezime" runat="server" CssClass="text-
box"></asp:TextBox>
            </td>
            <td class="error">
                <asp:RequiredFieldValidator ID="RequiredFieldValidatorImePrezime"
runat="server" ErrorMessage="Ime i prezime je obavezno !!!" Text="*"
ControlToValidate="TextBoxImePrezime"></asp:RequiredFieldValidator>
                <asp:RegularExpressionValidator
ID="RegularExpressionValidatorImePrezime"
                    runat="server" ErrorMessage="Ime i prezime nije u dobrom
formatu !!!" Text="*"
                    ControlToValidate="TextBoxImePrezime"
ValidationExpression="^\w{3,}\s\w{3,}$"></asp:RegularExpressionValidator>

            </td>
        </tr>
        <tr>
            <td>
                <asp:Label ID="LabelEmail" runat="server" Text="Vaša email
adresa:" AssociatedControlID="TextBoxEmail" CssClass="label"></asp:Label>
            </td>
            <td>
                <asp:TextBox ID="TextBoxEmail" runat="server" CssClass="text-
box"></asp:TextBox>
            </td>
            <td class="error">
                <asp:RequiredFieldValidator ID="RequiredFieldValidatorEmail"
runat="server" ErrorMessage="Email je obavezan !!!" Text="*"
ControlToValidate="TextBoxEmail"></asp:RequiredFieldValidator>
                <asp:RegularExpressionValidator
ID="RegularExpressionValidatorEmail"
                    runat="server" ErrorMessage="Email nije u dobrom formatu !!!"
Text="*"
```

```
                    ControlToValidate="TextBoxEmail"
                    ValidationExpression="\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-
.]\w+)*"></asp:RegularExpressionValidator>
                </td>
            </tr>
            <tr>
                <td>
                    <asp:Label ID="LabelOsoba" runat="server" Text="Pitanje za:"
AssociatedControlID="DropDownListOsoba" CssClass="label"></asp:Label>
                </td>
                <td>
                    <asp:DropDownList ID="DropDownListOsoba" runat="server"
CssClass="drop-down-list">
                    </asp:DropDownList>
                </td>
                <td class="error">

                    <asp:CustomValidator ID="CustomValidatorOsoba" runat="server"
                        ErrorMessage="Morate izabrati osobu !!!"
ControlToValidate="DropDownListOsoba"
                        Text="*"
onservervalidate="CustomValidatorOsoba_ServerValidate"
ClientValidationFunction="ValidacijaOsoba"></asp:CustomValidator>
                </td>
            </tr>
            <tr>
                <td>
                    <asp:Label ID="LabelPoruka" runat="server" Text="Poruka:"
AssociatedControlID="TextBoxPoruka" CssClass="label"></asp:Label>
                </td>
                <td>
                    <asp:TextBox ID="TextBoxPoruka" runat="server"
TextMode="MultiLine" CssClass="text-box"></asp:TextBox>
                </td>
                <td class="error">
                    <asp:RequiredFieldValidator ID="RequiredFieldValidatorPoruka"
runat="server" ErrorMessage="Morate uneti nešto u polje Poruka !!!" Text="*"
ControlToValidate="TextBoxPoruka"></asp:RequiredFieldValidator>
                    <asp:CustomValidator ID="CustomValidatorPoruka" runat="server"
                        ErrorMessage="Morate uneti poruku duzu od 10 karaktera"
Text="*"
                        ControlToValidate="TextBoxPoruka"
                        onservervalidate="CustomValidatorPoruka_ServerValidate"
ClientValidationFunction="ValidacijaPoruka"></asp:CustomValidator>
                </td>
            </tr>
            <tr>
                <td>
                    <asp:Label ID="LabelSlanjeKopije" runat="server" Text="Pošalji
kopiju:" AssociatedControlID="CheckBoxKopija" CssClass="label"></asp:Label>
                </td>
                <td>
                    <asp:CheckBox ID="CheckBoxKopija" runat="server" />
                </td>
                <td>
                     
                </td>
            </tr>
```

```
        <tr>
            <td colspan="3" align="center">
                <asp:Button ID="ButtonPosalji" runat="server" Text="Pošalji"
CssClass="button"
                    onclick="ButtonPosalji_Click" />
            </td>
        </tr>

    </table>
    <asp:ValidationSummary ID="ValidationSummary1" runat="server"
CssClass="validacija-error" />
    <div id="podaciForma" class="post sakri" runat="server">
    </div>
</asp:Content>
<asp:Content ID="Content3" ContentPlaceHolderID="ContentPlaceHolderOAutoru"
Runat="Server">
</asp:Content>
```

c#:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Web.Configuration;
using System.Net.Mail;

public partial class kontakt : System.Web.UI.Page
{
    private string connectionString =
WebConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString;

    protected void Page_Load(object sender, EventArgs e)
    {
        this.Title = "IT Blog | Kontakt";

        if (!this.IsPostBack)
        {
            if (User.IsInRole("admin"))
            {
                Master.FindControl("LabelLinkoviAdmin").Visible = true;
            }
            else
            {
                Master.FindControl("LabelLinkoviAdmin").Visible = false;
            }
            if (User.IsInRole("korisnik"))
            {
                Master.FindControl("LabelLinkoviKorisnici").Visible = true;
            }
            else
            {
                Master.FindControl("LabelLinkoviKorisnici").Visible = false;
            }
```

```
            if (User.IsInRole("autor"))
            {
                Master.FindControl("LabelLinkoviAutori").Visible = true;
            }
            else
            {
                Master.FindControl("LabelLinkoviAutori").Visible = false;
            }

            Dictionary<int, string> osobe = new Dictionary<int, string>();
            osobe.Add(0, "Izaberite...");
            osobe.Add(1, "Autora");
            osobe.Add(2, "Administratora");

            DropDownListOsoba.DataSource = osobe;
            DropDownListOsoba.DataTextField = "Value";
            DropDownListOsoba.DataValueField = "Key";
            DropDownListOsoba.DataBind();

        }
    }

    protected void ButtonPosalji_Click(object sender, EventArgs e)
    {
        if (this.IsValid)
        {
            string imePrezime = TextBoxImePrezime.Text;
            string email = TextBoxEmail.Text;
            string osoba = DropDownListOsoba.SelectedValue;
            string poruka = TextBoxPoruka.Text;
            bool kopija = CheckBoxKopija.Checked;

            BulletedList podaci = new BulletedList();
            podaci.Items.Add(new ListItem(imePrezime));
            podaci.Items.Add(new ListItem(email));
            podaci.Items.Add(new ListItem(osoba));
            podaci.Items.Add(new ListItem(poruka));
            podaci.Items.Add(new ListItem(kopija.ToString()));



            try
            {
                MailMessage mail = new MailMessage();
                mail.From = new MailAddress(imePrezime, "Poslato od: " +
imePrezime);
                mail.To.Add(new MailAddress(email, "Poslato na mail " + email));
                mail.Subject = "Kontakt";
                mail.Body = "<html><body><p>Ovo je mail poslat sa sajta:
</p><p><b>" + poruka + "</b></p></body></html>";
                mail.IsBodyHtml = true;
                mail.Priority = MailPriority.Normal;
                SmtpClient klijent = new SmtpClient("smtp.gmail.com");
                klijent.Credentials = new
System.Net.NetworkCredential("sourcekontakt@gmail.com", "pass");
                klijent.Send(mail);
                podaci.Items.Add(new ListItem("E-mail je poslat!"));
            }
```

```
            catch
            {
                podaci.Items.Add(new ListItem("E-mail nije poslat!"));
            }

            podaciForma.Controls.Add(podaci);
            podaciForma.Attributes["class"] = "post";
        }
    }
    protected void CustomValidatorOsoba_ServerValidate(object source,
ServerValidateEventArgs args)
    {
        if (args.Value != "0")
        {
            args.IsValid = true;
        }
        else
        {
            args.IsValid = false;
        }
    }
    protected void CustomValidatorPoruka_ServerValidate(object source,
ServerValidateEventArgs args)
    {
        if (args.Value.Length > 10)
        {
            args.IsValid = true;
        }
        else
        {
            args.IsValid = false;
        }
    }
}
```

detaljnije.aspx:

```
<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">

        <script type="text/javascript" src="lightBox/js/jquery.js"></script>
              <script type="text/javascript" src="lightBox/js/jquery.lightbox-
0.5.js"></script>
              <link rel="stylesheet" type="text/css"
href="lightBox/css/jquery.lightbox-0.5.css" media="screen" />


              <script language="JavaScript" type="text/javascript">
        $(function () {
            $('.image a').lightBox();
        });


    </script>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolderSadrzaj"
Runat="Server">
    <asp:Label ID="LabelDetaljnijeOpis" runat="server"
        Text="Post sa traženim ID-jem ne postoji!"></asp:Label>
    <asp:Label ID="LabelDetaljnijeTekst" runat="server"></asp:Label><hr /><br />
    <h3>Komentari:</h3>
    <asp:ScriptManager ID="ScriptManager1" runat="server">
    </asp:ScriptManager>
    <asp:UpdatePanel ID="UpdatePanel1" runat="server">
        <ContentTemplate>


    <asp:Label ID="LabelDetaljnijeKomentari" runat="server"
Text=""></asp:Label><br />
    <h3>Da li imate nešto da dodate?</h3>
    <table style="width: 100%;">
        <tr>
            <td>
                <asp:Label ID="LabelIme" runat="server" Text="Ime:"></asp:Label>
            </td>
            <td>
                <asp:TextBox ID="TextBoxIme" runat="server"></asp:TextBox>
            </td>
            <td>
                <asp:RequiredFieldValidator ID="RequiredFieldValidatorIme"
runat="server" ErrorMessage="<img align='absmiddle' src='images/warn.gif' />"
ControlToValidate="TextBoxIme"
SetFocusOnError="true"></asp:RequiredFieldValidator>
            </td>
        </tr>
        <tr>
            <td>
                <asp:Label ID="LabelEmail" runat="server" Text="E-
mail:"></asp:Label>
            </td>
            <td>
                <asp:TextBox ID="TextBoxEmail" runat="server"></asp:TextBox>
            </td>
```

```
            <td>
                <asp:RequiredFieldValidator ID="RequiredFieldValidatorEmail"
runat="server" ErrorMessage="<img align='absmiddle' src='images/warn.gif' />"
ControlToValidate="TextBoxEmail"></asp:RequiredFieldValidator>
                <asp:RegularExpressionValidator
ID="RegularExpressionValidatorEmail"
                    runat="server" ErrorMessage="<img align='absmiddle'
src='images/warn.gif' /> Format"
                    ControlToValidate="TextBoxEmail"
                    ValidationExpression="\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-
.]\w+)*" ></asp:RegularExpressionValidator>
            </td>
        </tr>
        <tr>
            <td>
                <asp:Label ID="LabelSite" runat="server"
Text="WebSite:"></asp:Label>
            </td>
            <td>
                <asp:TextBox ID="TextBoxSite" runat="server"></asp:TextBox>
            </td>
            <td>
                <asp:RegularExpressionValidator
ID="RegularExpressionValidatorSite"
                    runat="server" ErrorMessage="<img align='absmiddle'
src='images/warn.gif' /> Format"
                    ControlToValidate="TextBoxSite"
                    ValidationExpression="http(s)?://([\w-]+\.)+[\w-]+(/[\w-
./?%&amp;=]*)?" ></asp:RegularExpressionValidator>
            </td>
        </tr>
        <tr>
            <td>
                <asp:Label ID="LabelKomentar" runat="server"
Text="Komentar:"></asp:Label>
            </td>
            <td>
                <asp:TextBox ID="TextBoxKomentar" runat="server"
TextMode="MultiLine"
                    Height="88px" Width="291px"></asp:TextBox>
            </td>
            <td>
                <asp:RequiredFieldValidator ID="RequiredFieldValidatorKomentar"
runat="server" ErrorMessage="<img align='absmiddle' src='images/warn.gif' />"
ControlToValidate="TextBoxKomentar"></asp:RequiredFieldValidator>
            </td>
        </tr>
        <tr>
            <td colspan="3" align="center">
                <asp:Button ID="ButtonDodajKomentar" runat="server"
Text="Komentar"
                    onclick="ButtonDodajKomentar_Click" />
            </td>
        </tr>
    </table>
    </ContentTemplate>
    </asp:UpdatePanel><br />
```

```
</asp:Content>
<asp:Content ID="Content3" ContentPlaceHolderID="ContentPlaceHolderOAutoru"
Runat="Server">
    <asp:Label ID="LabelOAutoru" runat="server" Text="Label"></asp:Label><br />
</asp:Content>
```

c#:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.Configuration;
using System.Data.SqlClient;
using System.Data;
using System.Web.Services;

public partial class detaljnije : System.Web.UI.Page
{
    private string connectionString =
WebConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString;

    protected void Page_Load(object sender, EventArgs e)
    {
        this.Title = "IT Blog | Detaljnije";



        if (Request.QueryString.ToString() == "")
        {
            Response.Redirect("index.aspx");
        }
        if (!this.IsPostBack)
        {
            string id = Request.QueryString["post"].ToString();
            string brk = "";
            string upit2 = "SELECT COUNT(idKomentar) AS BrK FROM Komentari WHERE
idPost=" + id;
            SqlConnection con2 = new SqlConnection(connectionString);
            SqlCommand com2 = new SqlCommand(upit2, con2);
            using (con2)
            {
                con2.Open();
                SqlDataReader rid = com2.ExecuteReader();
                while (rid.Read())
                {
                    brk = rid["BrK"].ToString();
                }
                rid.Close();
            }
```

```
            string upit = "SELECT p.*, t.*, a.*, k.* FROM Postovi p INNER JOIN
Tekstovi t ON p.idPost=t.idPost INNER JOIN Autori a ON p.idAutor=a.idAutor INNER
JOIN Kontakti k ON a.idAutor=k.idAutor WHERE p.idPost="+id;
            SqlConnection con = new SqlConnection(connectionString);
            SqlCommand com = new SqlCommand(upit, con);
            using (con)
            {
                con.Open();
                SqlDataReader rid = com.ExecuteReader();
                while (rid.Read())
                {
                    DateTime datum = (DateTime)rid["Datum"];
                    LabelDetaljnijeOpis.Text = "<div
class='post_boxDetaljnije'><h2>" + rid["Naslov"] + "</h2><br/><div
class='post_info'><div class='post_date'>Datum:<br/> " +
datum.ToString("dd.MM.yyyy.") + "</div> "
                                            + " <div
class='post_author'>Autor:<br/> <a href='rezultat.aspx?a=" + rid["idAutor"] + "'>"
+ rid["KorisničkoIme"] + "</a></div><div class='post_comment'>Komentara:<br/>" +
brk + "</div><div class='cleaner'></div></div></div>";

                    LabelOAutoru.Text = "<div class='side_column_section'><div
class='OZaAutora'><div class='HZaAutora'>O autoru</div><div
class='SiTOA'><table><tr><td><p class='PZaA'><img class='SA' align='left' src=" +
rid["SlikaAutora"] + " alt='slikaAutora' />" + rid["TekstOAutoru"] +
"</p></td></tr></table><hr/><p>Kontakt:</p><a href=" + rid["WebSite"] + ">Web
site</a><br/><a href=" + rid["Facebook"] + ">Facebook profil</a><br/><a href=" +
rid["LinkedIn"] + ">LinkedIn profil</a><br/><a href=" + rid["Twitter"] + ">Twitter
profil</a></div></div></div><div class='cleaner_h30'> </div>";

                    LabelDetaljnijeTekst.Text += " <br/><div
class='post_boxDetaljnije'><div class='post_body'><h3>" + rid["Podnaslov"] +
"</h3> "
                                            + " <p>" + rid["Tekst"] +
"</p></div></div>";

                }
                rid.Close();
            }

            string upit1 = "SELECT DISTINCT k.*, f.Putanja FROM Komentari AS k
INNER JOIN Fajlovi AS f ON k.imeZaSliku = f.Korisnik WHERE k.idPost = " + id + "
AND f.OpisFajla='Slika korisnika' ORDER BY k.DatumKom";
            SqlConnection con1 = new SqlConnection(connectionString);
            SqlCommand com1 = new SqlCommand(upit1, con1);
            using (con1)
            {
                con1.Open();
                SqlDataReader rid = com1.ExecuteReader();
                while (rid.Read())
                {

                    DateTime datum = (DateTime)rid["DatumKom"];
                    LabelDetaljnijeKomentari.Text += "<table class='komentari'
style='width: 100%; border-width:1px; border-style:dotted; border-
color:grey;'><tr><td rowspan='2' style='width:100px;'><img width='80px'
height='80px' src=" + rid["Putanja"] + " alt='SlikaAutoraKomentara'
style='margin:10px;' /></td> "
```

```
                                                    + " <td><p>Napisao: <a
href=" + rid["WebSite"] + ">" + rid["AutorKomentara"] + "</a></p><p>Dana: " +
datum.ToString("dd.MM.yyyy. u HH:mm") +
"</p></td></tr><tr><td></td><td></td></tr><tr><td colspan='3'><p style='margin:0px
10px 10px 10px;'>    " + rid["Komentar"] +
"</p></td></tr></table><br />";

                }
                rid.Close();
            }

            if (User.IsInRole("admin"))
            {
                Master.FindControl("LabelLinkoviAdmin").Visible = true;
            }
            else
            {
                Master.FindControl("LabelLinkoviAdmin").Visible = false;
            }
            if (User.IsInRole("korisnik"))
            {
                Master.FindControl("LabelLinkoviKorisnici").Visible = true;
            }
            else
            {
                Master.FindControl("LabelLinkoviKorisnici").Visible = false;
            }
            if (User.IsInRole("autor"))
            {
                Master.FindControl("LabelLinkoviAutori").Visible = true;
            }
            else
            {
                Master.FindControl("LabelLinkoviAutori").Visible = false;
            }


        }
    }

    protected void ButtonDodajKomentar_Click(object sender, EventArgs e)
    {
        string[] korisniciSaSlikom = new string[50];
        string upit1 = "SELECT DISTINCT Korisnik FROM Fajlovi WHERE OpisFajla =
'Slika korisnika'";
        SqlConnection con1 = new SqlConnection(connectionString);
        SqlCommand com1 = new SqlCommand(upit1, con1);
        using (con1)
        {
            int i = 0;
            con1.Open();
            SqlDataReader rid = com1.ExecuteReader();
            while (rid.Read())
            {

                korisniciSaSlikom[i] = rid["Korisnik"].ToString();
                i++;
```

```
            }
            rid.Close();
        }

        if (this.IsValid)
        {
            string ime = TextBoxIme.Text;
            string email = TextBoxEmail.Text;
            string site = TextBoxSite.Text;
            string komentar = TextBoxKomentar.Text;
            string id = Request.QueryString["post"].ToString();

            string upit = "INSERT INTO [Komentari] ([Komentar], [idPost],
[AutorKomentara], [DatumKom], [Email], [WebSite], [imeZaSliku]) VALUES (@Komentar,
@idPost, @AutorKomentara, @Datum, @Email, @WebSite, @imeZaSliku)";
            SqlConnection con = new SqlConnection(connectionString);
            SqlCommand com = new SqlCommand(upit, con);
            using (con)
            {
                string imeZaSliku = "";
                com.Parameters.AddWithValue("Komentar", komentar);
                com.Parameters.AddWithValue("idPost", id);
                com.Parameters.AddWithValue("AutorKomentara", ime);
                com.Parameters.AddWithValue("Datum", DateTime.Now);
                com.Parameters.AddWithValue("Email", email);
                com.Parameters.AddWithValue("WebSite", site);
                if (User.Identity.Name != "")
                {
                    for (int i = 0; i < korisniciSaSlikom.Length; i++)
                    {
                        if (User.Identity.Name == korisniciSaSlikom[i])
                        {
                            imeZaSliku = User.Identity.Name;
                            break;
                        }
                    }
                    if (User.Identity.Name != imeZaSliku)
                    {
                        imeZaSliku = "nemaSlike";
                    }
                }
                else
                {
                    imeZaSliku = "nemaSlike";
                }
                com.Parameters.AddWithValue("imeZaSliku", imeZaSliku);


                con.Open();
                com.ExecuteNonQuery();

            }
            TextBoxIme.Text = "";
            TextBoxEmail.Text = "";
            TextBoxSite.Text = "";
            TextBoxKomentar.Text = "";

            LabelDetaljnijeKomentari.Text = "";
```

```
            string upit2 = "SELECT DISTINCT k.*, f.Putanja FROM Komentari AS k
INNER JOIN Fajlovi AS f ON k.imeZaSliku = f.Korisnik WHERE k.idPost = " + id + "
AND f.OpisFajla='Slika korisnika' ORDER BY k.DatumKom";
            SqlConnection con2 = new SqlConnection(connectionString);
            SqlCommand com2 = new SqlCommand(upit2, con2);
            using (con2)
            {
                con2.Open();
                SqlDataReader rid = com2.ExecuteReader();
                while (rid.Read())
                {
                    DateTime datum = (DateTime)rid["DatumKom"];
                    LabelDetaljnijeKomentari.Text += "<table class='komentari'
style='width: 100%; border-width:1px; border-style:dotted; border-
color:grey;'><tr><td rowspan='2' style='width:100px;'><img width='80px'
height='80px' src=" + rid["Putanja"] + " alt='SlikaAutoraKomentara'
style='margin:10px;' /></td> "
                                                        + " <td><p>Napisao: <a
href=" + rid["WebSite"] + ">" + rid["AutorKomentara"] + "</a></p><p>Dana: " +
datum.ToString("dd.MM.yyyy. u HH:mm") +
"</p></td></tr><tr><td></td><td></td></tr><tr><td colspan='3'><p style='margin:0px
10px 10px 10px;'>    " + rid["Komentar"] +
"</p></td></tr></table><br />";

                }
                rid.Close();
            }
        }
    }
}
```

mojnalog.aspx:

```
<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolderSadrzaj"
Runat="Server">



    <h3>Dodavanje slike na Vaš nalog:</h3>
    <table style="width: 100%;">
                        <tr>
                            <td class="style1">
                                <asp:Label ID="LabelSlikaZaNalog" runat="server"
Text="Izaberite sliku:"></asp:Label>
                            </td>
                            <td>
                                <asp:FileUpload ID="FileUploadSlika"
runat="server" />
                            </td>
                            <td>

                            </td>
                        </tr>
                        <tr>
```

```
                            <td colspan="3"><asp:Button ID="ButtonUplooad"
runat="server" Text="Dodaj sliku"
                                    onclick="ButtonUplooad_Click" /></td>
                        </tr>
                        <tr>
                            <td colspan="3"><asp:Label ID="Label1" runat="server"
Text=""></asp:Label></td>
                        </tr>
                    </table>

</asp:Content>
<asp:Content ID="Content3" ContentPlaceHolderID="ContentPlaceHolderOAutoru"
Runat="Server">
    <asp:Label ID="LabelMojNalog" runat="server"></asp:Label><br />
</asp:Content>
```

c#:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.IO;
using System.Data.SqlClient;
using System.Web.Configuration;
using System.Data;

public partial class mojnalog : System.Web.UI.Page
{
    private string folderZaFajlove;
    private string uloga;
    protected void Page_Load(object sender, EventArgs e)
    {
        Page.Title = "IT Blog | Moj nalog";

        if (!User.Identity.IsAuthenticated)
        {
            Response.Redirect("index.aspx");
        }

        if (User.IsInRole("admin"))
        {
            Master.FindControl("LabelLinkoviAdmin").Visible = true;
            uloga = "Admin";
        }
        else
        {
            Master.FindControl("LabelLinkoviAdmin").Visible = false;
        }
        if (User.IsInRole("korisnik"))
        {
            Master.FindControl("LabelLinkoviKorisnici").Visible = true;
            uloga = "Korisnik";
        }
        else
```

```
        {
            Master.FindControl("LabelLinkoviKorisnici").Visible = false;
        }
        if (User.IsInRole("autor"))
        {
            Master.FindControl("LabelLinkoviAutori").Visible = true;
            uloga = "Autor";
        }
        else
        {
            Master.FindControl("LabelLinkoviAutori").Visible = false;
        }

        if (!this.IsPostBack)
        {
            string upit = "SELECT TOP 1 Putanja FROM Fajlovi WHERE OpisFajla =
'Slika korisnika' AND Korisnik = '" + User.Identity.Name + "' ORDER BY Datum DESC
";
            SqlConnection veza = new SqlConnection();
            veza.ConnectionString =
WebConfigurationManager.ConnectionStrings["ConnectionString"].ToString();
            SqlCommand com = new SqlCommand(upit, veza);
            using (veza)
            {
                veza.Open();
                SqlDataReader rid = com.ExecuteReader();

                    if (rid.Read() == false)
                    {
                        rid.Close();
                        LabelMojNalog.Text = "<div
class='side_column_section'><div class='OZaAutora'><div class='HZaAutora'>Moj
nalog</div><div class='SiTOA'><table><tr><td><p class='PZaA'><img width='80px'
height='80px'class='SA' align='left' src='Fajlovi/Slike/Korisnici/bezSlike.jpeg'
alt='slikaKorisnika' />Korisničko ime: " + User.Identity.Name + "<br/>Uloga
na blogu: " + uloga + "</p></td></tr></table><hr/></div></div></div>";
                    }
                    else
                    {
                        rid.Close();
                        rid = com.ExecuteReader();
                        rid.Read();

                            LabelMojNalog.Text = "<div
class='side_column_section'><div class='OZaAutora'><div class='HZaAutora'>Moj
nalog</div><div class='SiTOA'><table><tr><td><p class='PZaA'><img width='80px'
height='80px'class='SA' align='left' src=" + rid["Putanja"] + "
alt='slikaKorisnika' />Korisničko ime: " + User.Identity.Name + "<br/>Uloga
na blogu: " + uloga + "</p></td></tr></table><hr/></div></div></div>";

                    }

                    rid.Close();
            }
        }

        folderZaFajlove = Path.Combine(Request.PhysicalApplicationPath,
"Fajlovi");
```

```
    }
    protected void ButtonUplooad_Click(object sender, EventArgs e)
    {
        string relativnaPutanja;

        if (FileUploadSlika.PostedFile.FileName != "")
        {

            string ekstenzija =
Path.GetExtension(FileUploadSlika.PostedFile.FileName);
            switch (ekstenzija.ToLower())
            {
                case ".png":
                case ".bmp":
                case ".gif":
                case ".jpg":
                case ".jpeg": break;
                default: Label1.Text = "Niste odabrali sliku!"; return;
            }

            folderZaFajlove += "/Slike/Korisnici/";
            relativnaPutanja = "Fajlovi/Slike/Korisnici/";

            string ime = FileUploadSlika.PostedFile.FileName;
            string celokupnaPutanja = Path.Combine(folderZaFajlove,
User.Identity.Name + ekstenzija);
            try
            {
                FileUploadSlika.PostedFile.SaveAs(celokupnaPutanja);
                Label1.Text = "Fajl sa imenom: " + ime;
                Label1.Text += " je uploadovana uspesno u: " + celokupnaPutanja;
            }
            catch (Exception ex)
            {
                Label1.Text = ex.Message;
            }
            try
            {
                SqlConnection veza = new SqlConnection();
                veza.ConnectionString =
WebConfigurationManager.ConnectionStrings["ConnectionString"].ToString();
                SqlCommand komanda = new SqlCommand();
                komanda.CommandType = CommandType.Text;
                komanda.CommandText = "INSERT INTO dbo.Fajlovi(Putanja, OpisFajla,
Datum, Korisnik) VALUES(@putanja, @opisfajla, @datum, @korisnik)";
                komanda.Connection = veza;
                komanda.Parameters.AddWithValue("@putanja", relativnaPutanja +
User.Identity.Name + ekstenzija);
                komanda.Parameters.AddWithValue("@opisfajla", "Slika korisnika");
                komanda.Parameters.AddWithValue("@datum", DateTime.Now);
                komanda.Parameters.AddWithValue("@korisnik", User.Identity.Name);
                veza.Open();
                int brojRedova = komanda.ExecuteNonQuery();
                veza.Close();
                Label1.Text += "<br/>Broj upisanih redova: " +
brojRedova.ToString();
            }
            catch (Exception ex)
```

```
            {
                Label1.Text = ex.Message;
            }
        }
    }
}
```

Poll.cs:

```
using System;
using System.Data;
using System.Configuration;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

using Microsoft.Practices.EnterpriseLibrary.Common;
using Microsoft.Practices.EnterpriseLibrary.Data;

/// <summary>
/// Summary description for Poll
/// </summary>
public class Poll
{

    Database db ;

    /// <summary>
    /// ConString is the DB connection string specified in
web.config
    /// </summary>
      public Poll()
      {
         db = DatabaseFactory.CreateDatabase("ConnectionString");
      }


    /// <summary>
    /// Select poll by using ID
    /// </summary>
    /// <param name="pollID">ID of the poll to select.</param>
```

```
    public DataSet SelectPoll(int pollID)
    {
        return db.ExecuteDataSet("usp_poll_select", pollID);
    }

 /// <summary>
    /// Get a random active poll ID
    /// </summary>
    public int getRandomActivePollID()
    {
        object pollID =
db.ExecuteScalar("usp_Poll_getRandomActive");
        return pollID == null ? 0 : int.Parse(pollID.ToString());
    }


    /// <summary>
    /// list all the polls
    /// </summary>
    public DataTable ListPoll()
    {
        return db.ExecuteDataSet("usp_poll_list").Tables[0];
    }

    /// <summary>
    /// Insert a poll and return the ID
    /// </summary>
    /// <param name="question"></param>
    /// <returns>PollID</returns>
    public int InsertPoll(string question, string blockMode, bool
active)
    {
        object pollID = db.ExecuteScalar("usp_poll_insert",
question, blockMode, active);
        return pollID == null ? 0 : int.Parse(pollID.ToString());

    }

    /// <summary>
    /// Update the poll and return the number of rows affected
    /// </summary>
    /// <param name="pollID"></param>
    /// <param name="question"></param>
    /// <returns>Number of rows affected</returns>
    public int UpdatePoll(int pollID, string question, string
blockMode, bool active)
```

```
    {
        return db.ExecuteNonQuery("usp_poll_update", pollID,
question, blockMode, active);
    }

    /// <summary>
    /// Delete a poll by using ID
    /// </summary>
    /// <param name="pollID"></param>
    /// <returns>Number of rows affected</returns>
    public int DeletePoll(int pollID)
    {
        return db.ExecuteNonQuery("usp_poll_delete", pollID);
    }

    /// <summary>
    /// Insert choice of a poll
    /// </summary>
    /// <param name="pollID"></param>
    /// <param name="choice"></param>
    /// <returns>ID of the inserted choice</returns>
    public int InsertChoice(int pollID, string choice)
    {
        return db.ExecuteNonQuery("usp_PollChoices_insert",
pollID, choice);
    }

    /// <summary>
    /// Update a choice by its ID
    /// </summary>
    /// <param name="pollChoiceID"></param>
    /// <param name="choice"></param>
    /// <returns>Number of rows affected</returns>
    public int UpdateChoice(int pollChoiceID, string choice)
    {
        return db.ExecuteNonQuery("usp_PollChoices_update",
pollChoiceID, choice);
    }

    /// <summary>
    /// Delete multiple choices by IDs seperated by comma
    /// </summary>
    /// <param name="pollChoiceIDs">Comma seperated string of
IDs</param>
    /// <returns>Number of rows affected</returns>
    public int DeleteChoice(int choiceID)
```

```
    {
        return db.ExecuteNonQuery("usp_PollChoices_delete",
choiceID);
    }

    /// <summary>
    /// Update the vote count for a choice by its ID
    /// </summary>
    /// <returns>Number of rows affected</returns>
    public int UpdateChoiceVote(int pollChoiceID)
    {
        return db.ExecuteNonQuery("usp_PollChoices_vote",
pollChoiceID);
    }

    /// <summary>
    /// Log the IP address of the user and the poll id
    /// </summary>
    /// <returns></returns>
    public int InsertPollIP(int pollID, string ip)
    {
        return db.ExecuteNonQuery("usp_PollIPs_insert", pollID,
ip);
    }

/// <summary>
    /// Select an entry by the IP address and the poll id
    /// </summary>
    /// <returns>Positive number if exists</returns>
    public int SelectPollIP(int pollID, string ip)
    {
        object id = db.ExecuteScalar("usp_PollIPs_select",
pollID, ip);
        try { return id == null ? 0 : int.Parse(id.ToString()); }
        catch { return 0; }
    }

    public enum BlockMode
    {
        COOKIE = 1, IP_ADDRESS = 2, NONE = 3
    };

}
```

Anketa urađena po tutorijalu sa sajta:
http://www.clientsideasp.net/2009/03/01/aspnet-ajax-poll-using-jquery-a-complete-implementation-with-admin-part/

admin.aspx - c# kod:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.IO;
using System.Data.SqlClient;
using System.Data;
using System.Web.Configuration;
using System.Web.Services;

public partial class admin : System.Web.UI.Page
{
    private string folderZaFajlove;

    protected void Page_Load(object sender, EventArgs e)
    {
        this.Title = "IT Blog | Admin panel";

        divMsg.Visible = false;

        if (!this.IsPostBack)
        {
            if (User.IsInRole("admin"))
            {
                Master.FindControl("LabelLinkoviAdmin").Visible = true;
            }
            else
            {
                Response.Redirect("index.aspx");
                Master.FindControl("LabelLinkoviAdmin").Visible = false;
            }

            if (Request.QueryString["pid"] != null)
            {
                hidPollID.Value = Request.QueryString["pid"];
                ShowExistingPoll();
            }
        }

        folderZaFajlove = Path.Combine(Request.PhysicalApplicationPath,
"Fajlovi");
    }
    protected void DetailsViewPostovi_ItemUpdated(object sender,
DetailsViewUpdatedEventArgs e)
    {
        GridViewPostovi.DataBind();
    }
```

```
    protected void DetailsViewPostovi_ItemDeleted(object sender,
DetailsViewDeletedEventArgs e)
    {
        GridViewPostovi.DataBind();

    }
    protected void DetailsViewPostovi_ItemInserted(object sender,
DetailsViewInsertedEventArgs e)
    {
        GridViewPostovi.DataBind();
    }
    protected void DetailsViewPostovi_ItemDeleting(object sender,
DetailsViewDeleteEventArgs e)
    {
        string idPost = GridViewPostovi.SelectedDataKey.Value.ToString();
        SqlDataSourceBrisanje.DeleteCommand = "DELETE FROM Tekstovi WHERE idPost =
"+idPost;
        SqlDataSourceBrisanje.Delete();
        SqlDataSourceBrisanje.DeleteCommand = "DELETE FROM PostoviKategorije WHERE
idPost = " + idPost;
        SqlDataSourceBrisanje.Delete();
    }
    protected void DetailsViewDetaljiOAutoru_ItemDeleted(object sender,
DetailsViewDeletedEventArgs e)
    {
        GridViewAutori.DataBind();
    }
    protected void DetailsViewDetaljiOAutoru_ItemInserted(object sender,
DetailsViewInsertedEventArgs e)
    {
        GridViewAutori.DataBind();
    }
    protected void DetailsViewDetaljiOAutoru_ItemUpdated(object sender,
DetailsViewUpdatedEventArgs e)
    {
        GridViewAutori.DataBind();
    }
    protected void DetailsViewDetaljiOAutoru_ItemDeleting(object sender,
DetailsViewDeleteEventArgs e)
    {
        string idAutor = GridViewAutori.SelectedDataKey.Value.ToString();
        SqlDataSourceBrisanje.DeleteCommand = "DELETE FROM Postovi WHERE idAutor =
" + idAutor;
        SqlDataSourceBrisanje.Delete();
        GridViewPostovi.DataBind();
    }
    protected void DetailsViewDOTekstovima_ItemDeleted(object sender,
DetailsViewDeletedEventArgs e)
    {
        GridViewTekstovi.DataBind();
    }
    protected void DetailsViewDOTekstovima_ItemInserted(object sender,
DetailsViewInsertedEventArgs e)
    {
        GridViewTekstovi.DataBind();
    }
    protected void DetailsViewDOTekstovima_ItemUpdated(object sender,
DetailsViewUpdatedEventArgs e)
```

```
    {
        GridViewTekstovi.DataBind();
    }
    protected void DetailsViewKategorije_ItemDeleted(object sender,
DetailsViewDeletedEventArgs e)
    {
        GridViewKategorije.DataBind();
    }
    protected void DetailsViewKategorije_ItemInserted(object sender,
DetailsViewInsertedEventArgs e)
    {
        GridViewKategorije.DataBind();
    }
    protected void DetailsViewKategorije_ItemUpdated(object sender,
DetailsViewUpdatedEventArgs e)
    {
        GridViewKategorije.DataBind();
    }
    protected void DetailsViewPK_ItemDeleted(object sender,
DetailsViewDeletedEventArgs e)
    {
        GridViewPK.DataBind();
    }
    protected void DetailsViewPK_ItemInserted(object sender,
DetailsViewInsertedEventArgs e)
    {
        GridViewPK.DataBind();
    }
    protected void DetailsViewPK_ItemUpdated(object sender,
DetailsViewUpdatedEventArgs e)
    {
        GridViewPK.DataBind();
    }
    protected void DetailsViewDKA_ItemDeleted(object sender,
DetailsViewDeletedEventArgs e)
    {
        GridViewKA.DataBind();
    }
    protected void DetailsViewDKA_ItemInserted(object sender,
DetailsViewInsertedEventArgs e)
    {
        GridViewKA.DataBind();
    }
    protected void DetailsViewDKA_ItemUpdated(object sender,
DetailsViewUpdatedEventArgs e)
    {
        GridViewKA.DataBind();
    }
    protected void ButtonUploadFajla_Click(object sender, EventArgs e)
    {
        string relativnaPutanja;

        if (FileUploadFajl.PostedFile.FileName == "" || TextBoxNazivFajla.Text ==
"" || TextBoxOpisFajla.Text == "")
        {
            LabelInfo.Text = "Niste dobro popunili stavke za upload!";
        }
        else
```

```
        {
            string ekstenzija =
Path.GetExtension(FileUploadFajl.PostedFile.FileName);
            switch (ekstenzija.ToLower())
            {
                case ".png":
                case ".bmp":
                case ".gif":
                case ".jpg":
                case ".jpeg":
                case ".zip":
                case ".rar":
                case ".aspx":
                case ".cs": break;
                default: LabelInfo.Text = "Niste odabrali odgovarajući fajl!";
return;
            }

            if (RadioButtonListFilter.SelectedValue.ToString() == "autor")
            {
                folderZaFajlove += "/Slike/Autori/";
                relativnaPutanja = "Fajlovi/Slike/Autori/";
            }
            else if (RadioButtonListFilter.SelectedValue.ToString() == "post")
            {
                folderZaFajlove += "/Slike/Postovi/V/";
                relativnaPutanja = "Fajlovi/Slike/Postovi/V/";
            }
            else if (RadioButtonListFilter.SelectedValue.ToString() == "kod")
            {
                folderZaFajlove += "/Kodovi/";
                relativnaPutanja = "Fajlovi/Kodovi/";
            }
            else if (RadioButtonListFilter.SelectedValue.ToString() == "stranica")
            {
                folderZaFajlove = Request.PhysicalApplicationPath;
                relativnaPutanja = "";
            }
            else
            {
                LabelInfo.Text = "Niste odabrali stavku radioButtonListe!!!";
                return;
            }

            string ime = TextBoxNazivFajla.Text;
            string celokupnaPutanja = Path.Combine(folderZaFajlove, ime +
ekstenzija);
            try
            {
                FileUploadFajl.PostedFile.SaveAs(celokupnaPutanja);
                LabelInfo.Text = "Fajl sa imenom: " + ime + ekstenzija;
                LabelInfo.Text += " je uploadovana uspesno u: " +
celokupnaPutanja;
            }
            catch (Exception ex)
            {
                LabelInfo.Text = ex.Message;
            }
```

```
            try
            {
                SqlConnection veza = new SqlConnection();
                veza.ConnectionString =
WebConfigurationManager.ConnectionStrings["ConnectionString"].ToString();
                SqlCommand komanda = new SqlCommand();
                komanda.CommandType = CommandType.Text;
                komanda.CommandText = "INSERT INTO dbo.Fajlovi(Putanja, OpisFajla,
Datum, Korisnik) VALUES(@putanja, @opisfajla, @datum, @korisnik)";
                komanda.Connection = veza;
                komanda.Parameters.AddWithValue("@putanja",
relativnaPutanja+ime+ekstenzija);
                komanda.Parameters.AddWithValue("@opisfajla",
TextBoxOpisFajla.Text);
                komanda.Parameters.AddWithValue("@datum", DateTime.Now);
                komanda.Parameters.AddWithValue("@korisnik", User.Identity.Name);
                veza.Open();
                int brojRedova = komanda.ExecuteNonQuery();
                veza.Close();
                LabelInfo.Text += "<br/>Broj upisanih redova: "
+brojRedova.ToString();
            }
            catch (Exception ex)
            {
                LabelInfo.Text = ex.Message;
            }
        }
    }

    private void ShowExistingPoll()
    {

        Poll objPoll = new Poll();
        DataSet dsPoll = objPoll.SelectPoll(int.Parse(hidPollID.Value));

        if (dsPoll.Tables[0].Rows.Count > 0)
        {
            txtQuestion.Text = dsPoll.Tables[0].Rows[0]["Question"].ToString();
            string blockMode = dsPoll.Tables[0].Rows[0]["BlockMode"].ToString();
            if (blockMode == Poll.BlockMode.COOKIE.ToString())
                rdoCookie.Checked = true;
            else if (blockMode == Poll.BlockMode.IP_ADDRESS.ToString())
                rdoIP.Checked = true;
            else
                rdoNone.Checked = true;
            chkActive.Checked =
bool.Parse(dsPoll.Tables[0].Rows[0]["Active"].ToString());

            ShowChoices(dsPoll.Tables[1]);
            hidRowIndex.Value = dsPoll.Tables[1].Rows.Count.ToString();
            btnSave.Text = "Ažuriraj anketu";
        }
        else
        {
            hidPollID.Value = "";
            btnSave.Text = "Sačuvaj anketu";
        }
    }
```

```
    private void ShowChoices(DataTable dtChoice)
    {
        System.Text.StringBuilder sb = new System.Text.StringBuilder();
        sb.Append("<table id='tableChoices' width='100%'>\n");
        for (int i = 0; i < dtChoice.Rows.Count; i++)
        {
            sb.Append("<tr id='pRow").Append(i).Append("'>\n");
            sb.Append("<td width='200' class='tdLabel'>Odgovor
").Append(i+1).Append("</td>\n");
            sb.Append("<td><input type='hidden'
id='hidPollChoiceID").Append(i).Append("'
name='hidPollChoiceID").Append(i).Append("'
Value='").Append(dtChoice.Rows[i]["PollChoiceID"]).Append("' />\n");
            sb.Append("<input type='text' class='text'
id='txtChoice").Append(i).Append("' name='txtChoice").Append(i).Append("'
value='").Append(dtChoice.Rows[i]["Choice"]).Append("'/> \n");
            sb.Append("<a href='#'
onclick='removeFormField(\"#pRow").Append(i).Append("\");return
false;'>Obriši</a></td>\n");
            sb.Append("</tr>");
        }
        sb.Append("</table>");
        hidRowIndex.Value = dtChoice.Rows.Count.ToString();
        divChoices.InnerHtml = sb.ToString();
    }

    protected void rptChoices_ItemDataBound(object sender, RepeaterItemEventArgs
e)
    {
        if (e.Item.ItemType == ListItemType.Item || e.Item.ItemType ==
ListItemType.AlternatingItem)
        {
            if (e.Item.ItemIndex <= 1)
                e.Item.FindControl("linkRemove").Visible = false;
            else if (e.Item.ItemIndex > 1)
            {
                ((RequiredFieldValidator)e.Item.FindControl("rqvChoice")).Visible
= false;
            }
        }
    }

    protected void btnSave_Click(object sender, EventArgs e)
    {
        if (txtQuestion.Text != "")
        {
            Poll objPoll = new Poll();
            int pollID = 0;

            string blockMode;
            if (rdoCookie.Checked)
                blockMode = Poll.BlockMode.COOKIE.ToString();
            else if (rdoIP.Checked)
                blockMode = Poll.BlockMode.IP_ADDRESS.ToString();
            else
                blockMode = Poll.BlockMode.NONE.ToString();
```

```
            if (hidPollID.Value == string.Empty)
            {
                pollID = objPoll.InsertPoll(txtQuestion.Text.Trim(), blockMode,
chkActive.Checked);


                if (pollID > 0)
                {
                    hidPollID.Value = pollID.ToString();
                    divMsg.InnerHtml = "<b>Obaveštenje:</b> Nova anketa je uspešno
dodata.";
                    divMsg.Visible = true;

                    GridViewListaAnketa.DataBind();
                }
            }
            else
            {
                pollID = int.Parse(hidPollID.Value);
                objPoll.UpdatePoll(pollID, txtQuestion.Text.Trim(), blockMode,
chkActive.Checked);

                divMsg.InnerHtml = "<b>Obaveštenje:</b> Vaša anketa je izmenjena
uspešno.";
                divMsg.Visible = true;
            }

            if (pollID > 0)
            {
                InsertUpdateChoices(pollID);
                ShowExistingPoll();
            }
            else
            {
                divMsg.InnerHtml = "<b>Greška:</b> Došlo je do greške, anketa ne
može biti snimljena.";
                divMsg.Attributes.Add("class", "mWarn");
                divMsg.Visible = true;
            }
        }
        else
        {
            divMsg.InnerHtml = "<b>Pitanje je obavezno.</b>";
            divMsg.Visible = true;
        }
    }

    private void InsertUpdateChoices(int pollID)
    {
        Poll objPoll = new Poll();

        foreach (string key in Request.Form)
        {
            if (key.IndexOf("txtChoice") >= 0)
            {
                string hidPollChoiceID = "hidPollChoiceID" +
key.Substring(key.Length - 1, 1);
```

```
                int choiceID = 0;
                if (Request.Form[hidPollChoiceID] != null &&
int.TryParse(Request.Form[hidPollChoiceID], out choiceID))
                {
                    if (Request.Form[key].Trim().Length > 0)
                        objPoll.UpdateChoice(choiceID, Request.Form[key]);
                    else
                        objPoll.DeleteChoice(choiceID);
                }
                else if (Request.Form[key].Trim().Length > 0)
                {
                    objPoll.InsertChoice(pollID, Request.Form[key].Trim());
                }
            }
        }
    }

    [WebMethod]
    public static int DeletePollChoice(int cID)
    {
        Poll objPoll = new Poll();
        return objPoll.DeleteChoice(cID);
    }
}
```

web.config:

```
<?xml version="1.0"?>

<!--
  For more information on how to configure your ASP.NET application, please visit
  http://go.microsoft.com/fwlink/?LinkId=169433
  -->

<configuration>

    <connectionStrings>
        <add name="ConnectionString" connectionString="Data
Source=.\SQLEXPRESS;AttachDbFilename='C:\Users\Tope\Documents\Visual Studio
2010\WebSites\ProbaZaTem157\App_Data\Sajt.mdf';Integrated Security=True;User
Instance=True"
            providerName="System.Data.SqlClient" />
    </connectionStrings>
    <system.web>
        <httpRuntime requestValidationMode="2.0" />
        <authentication mode="Forms" />
        <compilation debug="false" targetFramework="4.0" />
        <customErrors mode="On" defaultRedirect="greska.aspx">
          <error statusCode="403" redirect="greska403.aspx"/>
          <error statusCode="404" redirect="greska404.aspx"/>
        </customErrors>
          <membership defaultProvider="SqlProvider" userIsOnlineTimeWindow="15">
        <providers>
          <clear />
          <add name="SqlProvider" type="System.Web.Security.SqlMembershipProvider"
applicationName="ProbaZaTem157" connectionStringName="ConnectionString"
```

```
enablePasswordRetrieval="false" requiresQuestionAndAnswer="true"
requiresUniqueEmail="true" enablePasswordReset="true" passwordFormat="Hashed" />
        </providers>
      </membership>
      <roleManager defaultProvider="SQLRoleProvider" enabled="true"
cacheRolesInCookie="true" cookieName=".ASPROLES" cookieTimeout="30" cookiePath="/"
cookieRequireSSL="false" cookieSlidingExpiration="true" cookieProtection="All" >
        <providers>
          <clear />
          <add name="SQLRoleProvider" type=" System.Web.Security.SqlRoleProvider"
connectionStringName="ConnectionString" applicationName="ProbaZaTem157" />
        </providers>
      </roleManager>
  </system.web>

</configuration>
```