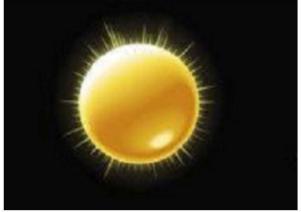
ЗАДАТАК 10: Временска прогноза (А18)

Дата је попуњена XML датотека са подацима о временској прогнози и слике које репрезентују податак о времену (ведро, облачно, киша и снег).

Креирати MVC апликацију користећи Razor View Engine за приказ података о временској прогнози прочитаних из XML датотеке. Апликација треба да садржи две странице:

Страницу за приказ временске прогнозе изабраног места

ВРЕМЕНСКА	ПРОГНОЗА
Назив места: Б	еоград ▼
Прика	ажи
Мимимлана температура:	0
Максимална температура:	10
Врем	ме:
1	41



Страница за максималне температуре

Странице за приказ максималних температура у свим местима

МАКСИМАЛНА ТЕМПЕРАТУРА

Назив места	Максимална температура
Београд	10
Младеновац	7
Москва	-10
Токио	20

Страница са прогнозом

РЕШЕЊЕ:

- 1. Креирати MVC апликацију, обрисати непотребне фолдере, фајлове и садржај одговарајућег фајла (Site.css).
- 2. Преузети XML датотеку и поставити у **App_Data** фолдер.
- 3. Креирати фолдер Slike, преузети слике и поставити их у фолдер Slike.
- 4. Global.asax.cs
 - Обрисати непотребне методе и креирати руте као на слици испод:

5. Креирати модел Prognoza.cs

```
namespace VremenskaPrognozaV1.Models
{
    public class Prognoza
    {
        public int Mesto { get; set; }
        public string NazivMesta { get; set; }
        public double MinTemperatura { get; set; }
        public double MaxTemperatura { get; set; }
        public string Vreme { get; set; }
}
```

НАПОМЕНА: Подаци који се дефинишу у оквиру класе зависе од података у XML датотеци, због десеријализације.

6. Креирати Home контролер, HomeController.cs

```
// dodati
using System.Xml;
                                    // dodati
using System.Xml.Serialization;
using VremenskaPrognozaV1.Models;
                                    // dodati
namespace VremenskaPrognozaV1.Controllers
{
    public class HomeController : Controller
        public ActionResult Index()
            List<Prognoza> prognoze = new List<Prognoza>();
            string putanja = Server.MapPath("~/App_Data/VremenskaPrognoza.xml");
            XmlReader reader = XmlReader.Create(putanja);
            XmlSerializer ser = new XmlSerializer(typeof(List<Prognoza>), new
                                    XmlRootAttribute("Prognoze"));
            prognoze = (List<Prognoza>)ser.Deserialize(reader);
            var gradovi = new List<SelectListItem>();
```

```
for (var i = 0; i < prognoze.Count(); i++)</pre>
                gradovi.Add(
                    new SelectListItem {
                        Value = i.ToString(),
                        Text = prognoze[i].NazivMesta
                    });
            ViewBag.gradovi = new SelectList(gradovi, "Value", "Text");
            ViewBag.Min = TempData["Min"] ??
            ViewBag.Max = TempData["Max"] ?? "";
            ViewBag.Vreme = TempData["Vreme"] ?? "";
            ViewBag.Vidljivo = TempData["Vreme"] == null ? "none" : "initial";
             return View();
        }
        public ActionResult Prikazi()
            List<Prognoza> prognoze = new List<Prognoza>();
            string putanja = Server.MapPath("~/App_Data/VremenskaPrognoza.xml");
            XmlReader reader = XmlReader.Create(putanja);
            XmlSerializer ser = new XmlSerializer(typeof(List<Prognoza>), new
                                XmlRootAttribute("Prognoze"));
            prognoze = (List<Prognoza>)ser.Deserialize(reader);
            var index = int.Parse(Request["nazivMesta"]);
            TempData["Min"] = prognoze[index].MinTemperatura.ToString();
            TempData["Max"] = prognoze[index].MaxTemperatura.ToString();
            TempData["Vreme"] = prognoze[index].Vreme.ToString();
            return RedirectToAction("Index");
        }
        public ActionResult maksimalneTemp()
            List<Prognoza> prognoze = new List<Prognoza>();
            string putanja = Server.MapPath("~/App_Data/VremenskaPrognoza.xml");
            XmlReader reader = XmlReader.Create(putanja);
            XmlSerializer ser = new XmlSerializer(typeof(List<Prognoza>), new
                                XmlRootAttribute("Prognoze"));
            prognoze = (List<Prognoza>)ser.Deserialize(reader);
            return View(prognoze);
        }
    }
}
```

7. Креирати View (за **Index** и **maksimalneTemp**)

8. index.cshtml

```
<html>
   <head>
       <title>Vremenska Prognoza</title>
       <link href="../../Content/Site.css" rel="stylesheet" />
   </head>
   <body>
       <div id="centar">
       @using (Html.BeginForm("Prikazi", "Home", FormMethod.Post)){
           @Html.Label("nm", "Назив места: ")
           @Html.DropDownList("nazivMesta", (SelectList)ViewBag.gradovi)
           <br /><br />
           <input type="submit" value="Прикажи"/>
       }
       <br />
       @Html.Label("minTemp", "Мимимлана температура: ")
       @Html.TextBox("min", null ,new { @readonly= "readonly" , @value = "@ViewBag.Min"})
```

9. maksimalnaTemp.cshtml

```
@using VremenskaPrognozaV1.Models
@model IEnumerable<Prognoza>
<html>
   <head>
       <title>Maksimalne temperature</title>
       <link href="../../Content/Site.css" rel="stylesheet" />
   </head>
   <body>
       <div id="naslov">МАКСИМАЛНА ТЕМПЕРАТУРА</div>
       <br /><br />
           Hазив места
                  Maксимална температура
              @foreach (var prognoza in Model) {
                  @prognoza.NazivMesta
                  @prognoza.MaxTemperatura
              <br />
       <div id="centar">
       @Html.ActionLink("Страница са прогнозом", "Index")
       </div>
   </body>
</html>
```

10. Site.css

МОДИФИКОВАНА ВЕРЗИЈА:

Global.asax.cs

```
using VremenskaPrognozaV1.Models; // dodati
namespace VremenskaPrognozaV1
     public class MvcApplication : System.Web.HttpApplication
          public static void RegisterRoutes(RouteCollection routes)
            routes.MapRoute("Z10", "", new { Controller = "Home", Action = "Index" });
routes.MapRoute("Z11", "prikazi", new { Controller = "Home", Action = "Prikazi" });
routes.MapRoute("Z12", "maxtemp", new { Controller = "Home", Action = "maksimalneTemp" });
         protected void Application_Start()
               RegisterRoutes(RouteTable.Routes);
               Application["sveprognoze"] = UcitajXML();
         private List<Prognoza> UcitajXML()
               List<Prognoza> prognoze = new List<Prognoza>();
               string putanja = Server.MapPath("~/App_Data/VremenskaPrognoza.xml");
               XmlReader reader = XmlReader.Create(putanja);
              XmlSerializer ser = new XmlSerializer(typeof(List<Prognoza>), new
                                            XmlRootAttribute("Prognoze"));
               prognoze = (List<Prognoza>)ser.Deserialize(reader);
               return prognoze;
         }
    }
}
```

HomeController.cs

```
public ActionResult maksimalneTemp()
{
    var prognoze = (List<Prognoza>)HttpContext.Application["sveprognoze"];
    return View(prognoze);
}
```

НАПОМЕНА: Измена се односи на методу **maksimalneTemp**, изменити и за остале две методе.

Модификована верзија, решава проблем учитавање XML података. Дакле, зашто у свакој методи да учитавамо XML податке (укупно 3 пута), кад то можемо да урадимо само једном у оквиру методе Application_Start.