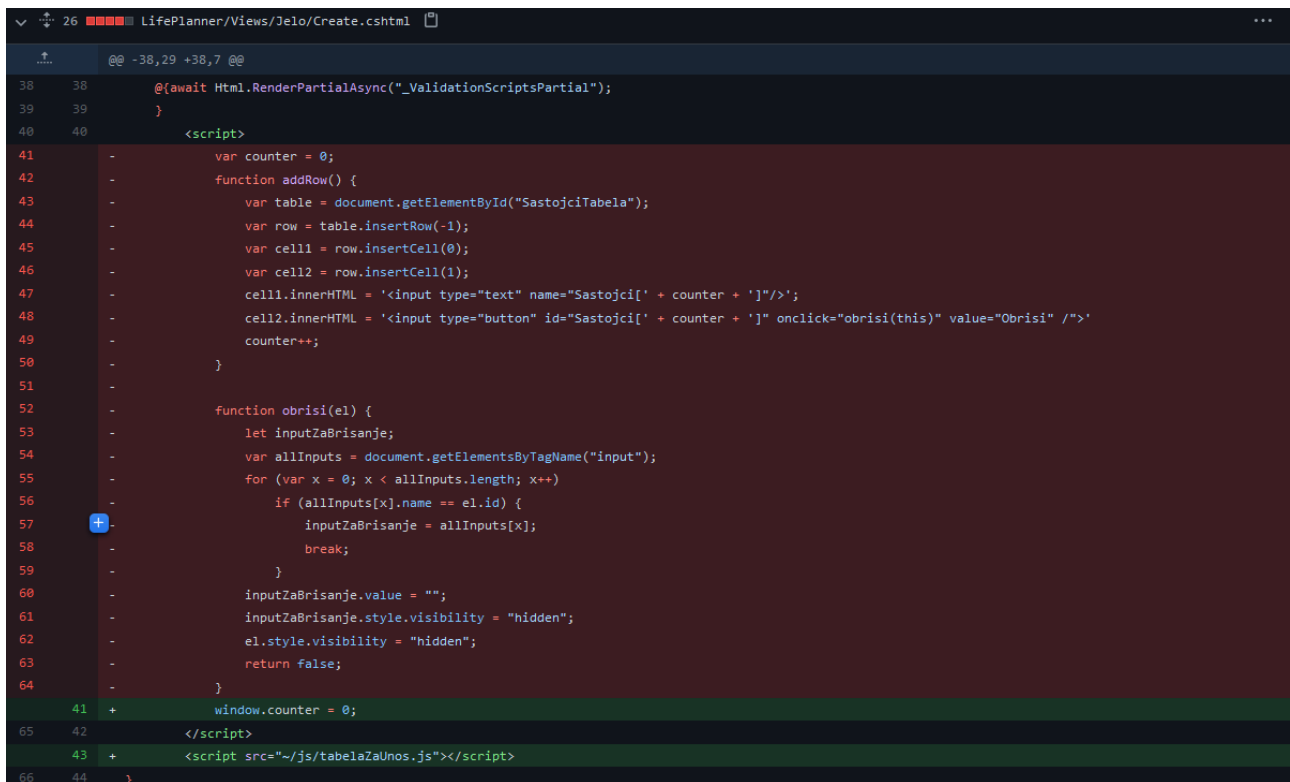


# Refaktorizacija

1. <https://github.com/ooad-2020-2021/Grupa2-r-queensmen/issues/32>

## Opis problema

Za dodavanje/uređivanje sastojaka i vježbi za jela i treninge respektivno koristili smo tabelu. U toj tabeli, svaki red predstavlja jedan sastojak/vježbu. Logika za tu tabelu je ista i ponavljala se u svakom .cshtml fajlu. Ovo je stanje prije:



```
38 38      @await Html.RenderPartialAsync("_ValidationScriptsPartial");
39 39  }
40 40  <script>
41  -      var counter = 0;
42  -      function addRow() {
43  -          var table = document.getElementById("SastojciTabela");
44  -          var row = table.insertRow(-1);
45  -          var cell1 = row.insertCell(0);
46  -          var cell2 = row.insertCell(1);
47  -          cell1.innerHTML = '<input type="text" name="Sastojci[' + counter + ']" />';
48  -          cell2.innerHTML = '<input type="button" id="Sastojci[' + counter + ']" onclick="obrisi(this)" value="Obrisi" />';
49  -          counter++;
50  -      }
51  -
52  -      function obrisi(el) {
53  -          let inputZaBrisanje;
54  -          var allInputs = document.getElementsByTagName("input");
55  -          for (var x = 0; x < allInputs.length; x++)
56  -              if (allInputs[x].name == el.id) {
57  +                  inputZaBrisanje = allInputs[x];
58  -                  break;
59  -              }
60  -          inputZaBrisanje.value = "";
61  -          inputZaBrisanje.style.visibility = "hidden";
62  -          el.style.visibility = "hidden";
63  -          return false;
64  -      }
65  +      window.counter = 0;
66  +      </script>
67  +      <script src="~/js/tabelaZaUnos.js"></script>
68  +  }
```

Crveno: Logika koja se nalazila u više .cshtml fajlova.

Zeleno: Dodana globalna *counter* varijabla u svakom .cshtml fajlu da joj možemo pristupiti iz *tabelaZaUnos.js* fajla. Također, istoimeni fajl je uključen u svaki .cshtml fajl.

Ovo je stanje nakon refaktorizacije (uključujući i zeleno sa gornje slike). Jednostavno smo svu tu logiku prebacili u jedan .js fajl i uključili ga gdje je potrebno:

```

1 + function addRow() {
2 +     let table;
3 +     let inHtml1, inHtml2;
4 +     if (location.href.includes('Vjezba')) {
5 +         table = document.getElementById("VjezbeTabela");
6 +         inHtml1 = '<input type="text" name="Vjezbe[' + counter + ']" />';
7 +         inHtml2 = '<input type="button" id="Vjezbe[' + counter + ']" onclick="obrisi(this)" value="Obrisi" />';
8 +     }
9 +     else if (location.href.includes('Jelo')) {
10 +         table = document.getElementById("SastojciTabela");
11 +         inHtml1 = '<input type="text" name="Sastojci[' + counter + ']" />';
12 +         inHtml2 = '<input type="button" id="Sastojci[' + counter + ']" onclick="obrisi(this)" value="Obrisi" />';
13 +     }
14 +     var row = table.insertRow(-1);
15 +     var cell1 = row.insertCell(0);
16 +     var cell2 = row.insertCell(1);
17 +     cell1.innerHTML = inHtml1;
18 +     cell2.innerHTML = inHtml2;
19 +     counter++;
20 + }
21 +
22 + function obrisi(el) {
23 +     let inputZaBrisanje;
24 +     var allInputs = document.getElementsByTagName("input");
25 +     for (var x = 0; x < allInputs.length; x++)
26 +         if (allInputs[x].name == el.id) {
27 +             inputZaBrisanje = allInputs[x];
28 +             break;
29 +         }
30 +     inputZaBrisanje.value = "";
31 +     inputZaBrisanje.style.visibility = "hidden";
32 +     el.style.visibility = "hidden";
33 +     return false;
34 + }

```

2. <https://github.com/ooad-2020-2021/Grupa2-r-queensmen/issues/41>

## Opis problema

Index ruta Zadatak kontrolera ima veliki broj zaduženja: slanje savjeta, ljudski formatiranog i C# nativnog datuma, zadataka i raspoloženja na View. Sve ovo je bilo rađeno preko ViewBag-a. Stanje se počelo komplikovati. Stoga smo odlučili sve to fino upakovati u jedan ViewModel koji će biti ugnježdjena klasa u kontroleru i isti poslati na View.

```
63 64
64 - string randomSavjet = savjeti[new Random().Next(savjeti.Count)];
65 - ViewBag.savjet = randomSavjet;
66 - ViewBag.datum = datum;
67 -
68 65 //https://stackoverflow.com/questions/30887367/how-to-compare-sql-datetime-and-c-sharp-datetime
69 66 DateTime RuzniNepotrebnDatum = DateTime.ParseExact(DateTime.Today.ToString("d_M_yyyy"), "d_M_yyyy", null);
70 67
71 68 var raspolozenje = await _context.Raspolozenja.FirstOrDefaultAsync(r => r.Datum == RuzniNepotrebnDatum && r.Korisnik == korisnik);
72 - ViewBag.danasnjiDatum = DateTime.Today;
73 - if (raspolozenje != null)
74 - {
75 -     ViewBag.raspolozenje = true;
76 -     ViewBag.raspolozenjeId = raspolozenje.Id;
77 - }
78 - else
79 - {
80 -     ViewBag.raspolozenje = false;
81 - }
69 +
82 70 viewModel.Korisnik = korisnik;
83 - viewModel.Savjet = randomSavjet;
71 + viewModel.Savjet = savjeti[new Random().Next(savjeti.Count)];
84 72 viewModel.OdabraniDatum = datum;
85 73 viewModel.taskoviZaDan = taskoviZaDan;
86 74 viewModel.Raspolozenje = raspolozenje;
```

Sada ta ruta izgleda ovako:

```
// GET: Zadatak
public async Task<IActionResult> Index(string datumString)
{
    ViewModel viewModel = new ViewModel();

    DateTime datum = DateTime.ParseExact(datumString, "d_M_yyyy", null);
    var korisnik = await _userManager.GetUserAsync(User);
    List<Zadatak> taskoviZaDan = new List<Zadatak> { };

    if (korisnik == null)
    {
        taskoviZaDan = HttpContext.Session.GetObjectFromJson<List<Zadatak>>("NeRegZadaci");
        if (taskoviZaDan == null)
        {
            taskoviZaDan = new List<Zadatak> { };
        }
    }
    else
    {
        taskoviZaDan = await _context.Zadaci.Where(z => z.Datum == datum && z.Korisnik == korisnik).ToListAsync();
    }

    //https://stackoverflow.com/questions/30887367/how-to-compare-sql-datetime-and-c-sharp-datetime
    DateTime RuzniNepotrebnDatum = DateTime.ParseExact(DateTime.Today.ToString("d_M_yyyy"), "d_M_yyyy", null);

    var raspolozenje = await _context.Raspolozenja.FirstOrDefaultAsync(r => r.Datum == RuzniNepotrebnDatum && r.Korisnik == korisnik);

    viewModel.Korisnik = korisnik;
    viewModel.Savjet = savjeti[new Random().Next(savjeti.Count)];
    viewModel.OdabraniDatum = datum;
    viewModel.taskoviZaDan = taskoviZaDan;
    viewModel.Raspolozenje = raspolozenje;
    return View(viewModel);
}
```

3. <https://github.com/ooad-2020-2021/Grupa2-r-queensmen/issues/40>

## Opis problema

Problem se odnosi na MoodController i sličan je kao prijašnji. ViewBag smo punili nepotrebno atributima samog modela Raspoloženje pa smo jednostavno poslali sam Model. Ovo su izmjene koje smo napravili u Index rutu:

```
LifePlanner/Controllers/VodaController.cs
@@ -28,23 +28,11 @@ public VodaController(ApplicationDbContext context, UserManager<RegistrovaniKori
28 28      public async Task<IActionResult> Index(string datumString)
29 29      {
30 30          DateTime datum = DateTime.ParseExact(datumString, "d_M_yyyy", null);
31 -         var vodaVecPostoji = await _context.KolicineVode.FirstOrDefaultAsync(v => v.Datum == datum);
32 -         if (vodaVecPostoji == null)
33 -         {
34 -             ViewBag.voda = 0;
35 -             ViewBag.postoji = false;
36 -             ViewBag.id = null;
37 -         }
38 -         else
39 -         {
40 -             ViewBag.voda = vodaVecPostoji.Kolicina;
41 -             ViewBag.postoji = true;
42 -             ViewBag.id = vodaVecPostoji.Id;
43 -         }
44 +         var korisnik = await _userManager.GetUserAsync(User);
45 +         var vodaVecPostoji = await _context.KolicineVode.FirstOrDefaultAsync(v => v.Datum == datum && v.Korisnik == korisnik);
46 +         ViewBag.datum = DajDatumZaPrikaz(datum);
47 +         ViewBag.datumFull = datum;
48 +         var korisnik = await _userManager.GetUserAsync(User);
49 +         return View(await _context.KolicineVode.Where(v => v.Korisnik.Id == korisnik.Id).ToListAsync());
50 +         return View(vodaVecPostoji);
51     }
52
53     // POST: Voda/Create
```

Ta ruta sa izgleda puno jednostavije (ViewBag je samo korišten za datume):

```
// GET: Mood
public async Task<IActionResult> Index(string datumString)
{
    DateTime datum = DateTime.ParseExact(datumString, "d_M_yyyy", null);
    var korisnik = await _userManager.GetUserAsync(User);
    var moodVecPostoji = await _context.Raspolozenja.FirstOrDefaultAsync(r => r.Datum == datum && r.Korisnik == korisnik);
    ViewBag.datum = datum.ToString("d.M.yyyy");
    ViewBag.datumFull = datum;
    return View(moodVecPostoji);
}
```

4. <https://github.com/ooad-2020-2021/Grupa2-r-queensmen/issues/33>

## Opis problema

Dosta je bilo miješanja built-in ruta i naših ruta, miješanja engleskog i bosanskog jezika. Veliki dio frontend-a je prepisan na naš jezik pa je ostalo da i rute i view-ove prepravimo. Na sljedećoj slici su prepravke za imena ruta i pripadajućih metoda, a ista imena imaju i View-ovi.

```
43 - [ActionName("Details")]
44 - public async Task<IActionResult> Index(Guid? id)
43 + [ActionName("Opis")]
44 + public async Task<IActionResult> Opis(Guid? id)
45 45 {
46 46     if (id == null)
47 47     {
@@ -59,18 +59,18 @@ public async Task<IActionResult> Index(Guid? id)
59 59     }
60 60
61 61     // GET: Trening/Create
62 - [ActionName("Create")]
63 - public IActionResult DodajTrening()
62 + [ActionName("Kreiraj")]
63 + public IActionResult KreirajTrening()
64 64 {
65 65     return View();
66 66 }
67 67
68 68     // POST: Trening/Create
69 69     // To protect from overposting attacks, enable the specific properties you want to bind to, for
70 70     // more details, see http://go.microsoft.com/fwlink/?LinkId=317598.
71 - [HttpPost, ActionName("Create")]
71 + [HttpPost, ActionName("Kreiraj")]
72 72     [ValidateAntiForgeryToken]
73 - public async Task<IActionResult> DodajTrening([Bind("Id,Naziv,Vjezbe")] Trening trening)
73 + public async Task<IActionResult> KreirajTrening([Bind("Id,Naziv,Vjezbe")] Trening trening)
74 74 {
75 75     if (ModelState.IsValid)
76 76     {
@@ -93,7 +93,7 @@ public async Task<IActionResult> DodajTrening([Bind("Id,Naziv,Vjezbe")] Trening
93 93     }
94 94
95 95     // GET: Trening/Edit/5
96 - [ActionName("Edit")]
96 + [ActionName("Uredi")]
97 97     public async Task<IActionResult> UrediTrening(Guid? id)
98 98 {
```

Naravno, svi linkovi (akcije) su također prepravljani i u svim potrebnim .cshtml fajlovima.

Na veliku žalost, aplikacija je prejednostavna za bilo kakav značajniji refaktor. Ovo su refaktori koji su se prirodno nametnuli i koje smo uspjeli prepoznati.

S druge strane, sa zadovoljstvom možemo saopštiti da smo implementovali 95% funkcionalnosti koje smo obećali. Uz to, “potrošili” smo preko 12 sati na uređivanje samog izgleda aplikacije koristeći najbolje prakse za responzivni dizajn: grid i flexbox (). Naravno, ovo ništa ne bi bilo moguće bez naše divne profesorice i asistentice. Hvala vam za sav uloženi trud, strpljenje, želju da prenesete znanje i na svim savjetima kako biste nam maksimalno olakšali.

Sve najbolje vam želi grupa r/queensmen <3