Laboratorijska Vježba 7.
ASP.NET Core MVC
Pogledi

Cilj vježbe:

- Upoznavanje sa načinom definisanja pogleda uz korištenje HTML, CSS i C# programskog jezika;
- Korištenje *bindinga* kontrola sa modelom podataka;
- Validacija podataka koristeći stereotipne anotacije u modelu podataka.

1. Definisanje pogleda koristeći kombinaciju HTML – CSS – C#

Za demonstraciju rada sa pogledima iskoristiti će se aplikacija koja simulira studentsku službu, za koju su automatski definisani kontroleri i pogledi u okviru prethodne laboratorijske vježbe. Ova web-aplikacija ima automatski definisanu *Home/Index* stranicu koja je prikazana u isječku koda ispod i sastoji se od jednostavnih HTML elemenata[1] kako slijedi:

```
@{
    ViewData["Title"] = "Home Page";
}

<div class="text-center">
    <h1 class="display-4">Welcome</h1>
    <p>Learn about <a href="https://docs.microsoft.com/aspnet/core">building Web
apps with ASP.NET Core</a>.</p>
</div>
```

Prvenstveno je potrebno primijetiti da kod koji je definisan u *Index* stranici nije potpun, jer postoji dio prikaza koji je zajednički za sve stranice i koji je definisan u **Shared** folderu. Ovaj folder sadrži zajedničku definiciju elemenata za prikaz poruka o validaciji i greškama, kao i **Layout**, odnosno osnovni oblik rasporeda elemenata na stranici. *Layout* zajednička stranica definisana je u isječku koda ispod kako slijedi:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>@ViewData["Title"] - StudentskaSluzba</title>
    <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />
    <link rel="stylesheet" href="~/css/site.css" />
</head>
<body>
    <header>
        <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-
white border-bottom box-shadow mb-3">
            <div class="container">
                <a class="navbar-brand" asp-area="" asp-controller="Home" asp-
action="Index">StudentskaSluzba</a>
```

[1] Predmet ove vježbe nije objašnjavanje značenja HTML kontrola. Početni link za istraživanje: https://www.w3schools.com/html/html_elements.asp.

```
                <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target=".navbar-collapse" aria-controls="navbarSupportedContent"
aria-expanded="false" aria-label="Toggle navigation">
                    <span class="navbar-toggler-icon"></span>
                </button>
                <div class="navbar-collapse collapse d-sm-inline-flex flex-sm-row-
reverse">
                    <ul class="navbar-nav flex-grow-1">
                        <li class="nav-item">
                            <a class="nav-link text-dark" asp-area="" asp-
controller="Home" asp-action="Index">Home</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link text-dark" asp-area="" asp-
controller="Home" asp-action="Privacy">Privacy</a>
                        </li>
                    </ul>
                </div>
            </div>
        </nav>
    </header>
    <div class="container">
        <main role="main" class="pb-3">
            @RenderBody()
        </main>
    </div>

    <footer class="border-top footer text-muted">
        <div class="container">
            &copy; 2021 - StudentskaSluzba - <a asp-area="" asp-controller="Home"
asp-action="Privacy">Privacy</a>
        </div>
    </footer>
    <script src="~/lib/jquery/dist/jquery.min.js"></script>
    <script src="~/lib/bootstrap/dist/js/bootstrap.bundle.min.js"></script>
    <script src="~/js/site.js" asp-append-version="true"></script>
    @RenderSection("Scripts", required: false)
</body>
</html>
```

Ova stranica je mnogo kompleksnija od *Index* stranice, jer sadrži sve elemente koje je potrebno navesti kako bi se svi pogledi mogli ispravno prikazivati. Ključni elementi na *Shared* stranici su:

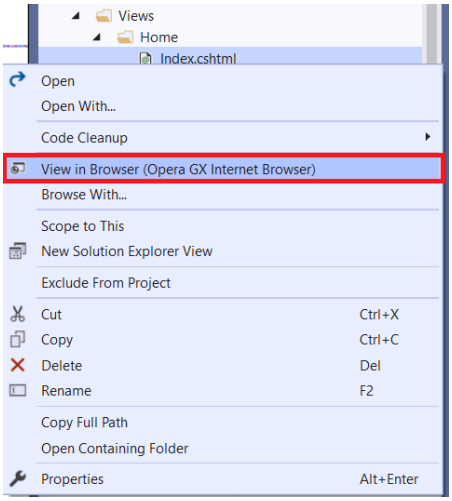
- Korištenje **bootstrap**[2] biblioteke koja omogućava responzivni web-dizajn, pri čemu se izgled kontrola automatski mijenja kada se mijenja i veličina stranice.

- Korištenje **nav** elementa, koji omogućava navigaciju kroz meni kontrola (kontrole su linkovi definisani kao **a** kontrole). Ovaj meni prikazuje se na vrhu svake stranice u aplikaciji i na njemu se prikazuju kontrole koje omogućavaju navigaciju na različite stranice (inicijalno je omogućena navigacija na *Index* i *Privacy*, a kasnije će se dodati mogućnost navigacije na *Predmets* stranice).

[2] Način na koji ova biblioteka poboljšava prikaz i omogućava responzivnost web-dizajna, kao i savjeti za lakše raspoređivanje kontrola mogu se pronaći na sljedećem linku: https://aspnetcore.readthedocs.io/en/stable/client-side/bootstrap.html.

- Korištenje *RenderBody()* C# funkcije koja vrši umetanje koda trenutne stranice na specificiranom mjestu, čime se osnovni kod za prikaz pogleda nadopunjuje sa elementima koji su definisani zasebno u odvojenom CSHTML *file*-u. Ovo zapravo znači da se pri učitavanju svake web-stranice prvo učitava *Layout* stranica, a zatim se na ovo mjesto umeće kod za željenu stranicu (npr. *Index*). Simbol **@** služi za specifikaciju da je naredba u C# jeziku i da je potrebno interpretirati, što će biti objašnjeno u sljedećem poglavlju ove vježbe.

Kako bi se izvršio pregled stranice, nema potrebe za pokretanjem cijele aplikacije (što nekad može trajati duže vremena), već je dovoljno u *Solution Exploreru* izvršiti desni klik na stranicu koja se želi pregledati i odabrati opciju *View in Browser*, kao što je prikazano na Slici 1. Nakon što se izvrše željene promjene na stranici, dovoljno je spasiti promjene i osvježiti stranicu u web-pregledniku, nakon čega će biti prikazan novi izgled stranice.



Slika 1. Opcija za prikaz izgleda web-stranice

Na ovaj način omogućava se prikaz *Home/Index* stranice bez pokretanja aplikacije. Ova stranica ima predefinisani izgled koji će se sada promijeniti u svrhu demonstriranja korištenja HTML i CSS *markup* sintakse za definisanje pogleda. Neka se želi omogućiti da stranica *Index* prikazuje sliku fakulteta, tekst sa uputstvima za korištenje, link koji korisnika preusmjerava na c2 i video koji se automatski pokreće kada se stranica učita.

Prvo se vrši definisanje prikaza teksta i slike fakulteta. Slika koja se želi koristiti nalazi se na internet lokaciji (dozvoljeno je i korištenje slika iz projekta sa relativnim putanjama[3]). **style** atribut koristi se za deklaraciju CSS koda (iako je CSS kod moguće izdvojiti u zaseban *file*) kojim se definiše stil željene kontrole. **img** kontrola postavljena je na cijelu širinu stranice, a **div** kontrola ima definiciju visine (koja iznosi 20 relativnih jedinica *em*), kao i definiciju da sve što prelazi zadanu visinu bude sakriveno. Na ovaj način postiže se izrezivanje slike, odnosno zbog veoma velike rezolucije slike potrebno je izrezati jedan njen dio kako ne bi zauzimala preveliki prostor. U isječku koda ispod definisana je slika, kao i tekstualne kontrole stranice, a sam prikaz pogleda prikazan je na Slici 2.

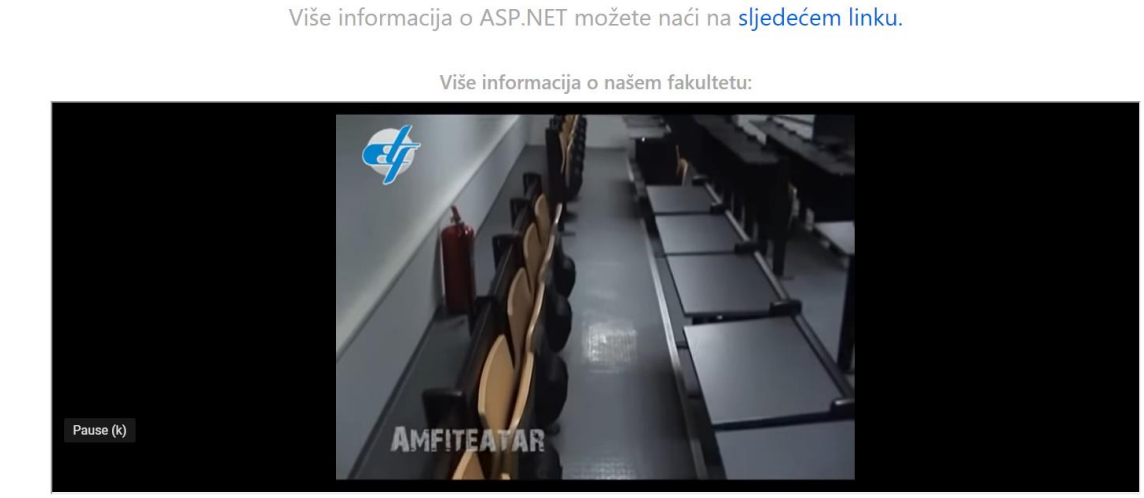
[3] Više informacija o korištenju relativnih putanja dostupno je na sljedećem linku: https://xspdf.com/resolution/317315.html.

```
    <h1 style="text-align: center">Studentska Služba</h1>
    <div style="height: 20em; overflow: hidden">
        <img src="https://www.tk.etf.unsa.ba/wp-
content/uploads/2019/09/dotfeesa_small_-68.jpg" style="width: 100%">
    </div>
        <h3 style="margin-top: 3%">Aplikacija za registrovanje studenata i
predmeta</h3>
        <div style="margin-top: 2%; color: darkgray; font-size: 1.5em">
            <b>
                Ova web-aplikacija namijenjena je za rad sa studentima i predmetima
fakulteta.
                <br>
                Sastoji se od korisničkih interfejsa koji omogućavaju vršenje
operacija dodavanja, editovanja i brisanja.
                <br>
                Trenutno je izvršen razvoj samo za predmete, kojima je moguće
pristupiti pomoću stavke menija na vrhu stranice.
            </b>
```

Slika 2. Prikaz slike i teksta na web-stranici

Potrebno je još definisati link i video kontrole. Link se definiše koristeći kontrolu **a**, pri čemu **href** atribut sadrži internet lokaciju na koju je potrebno preusmjeriti korisnika. Linkovi se mogu koristiti i za rutiranje, o čemu će biti riječi u nastavku. Za dodavanje prikaza video sadržaja koristi se **iframe** kontrola. Veoma je važno ispravno definisati **src** atribut, koji mora biti oblika **https://www.youtube.com/embed/<id videa>**, pri čemu je za automatsko pokretanje potrebno dodati parametar **autoplay**, a za gašenje zvuka parametar **mute**. ID videa može se pronaći na web-adresi videa nakon što se isti pokrene u web-pretraživaču. U isječku koda ispod prikazan je način definisanja ovih kontrola, a Slika 3 prikazuje izgled stranice nakon pokretanja.

```
            <br>
            Više informacija o ASP.NET možete naći na <a
href="https://c2.etf.unsa.ba/course/view.php?id=111"> sljedećem linku.</a>
            <br />
            <br />
            <h5>Više informacija o našem fakultetu:</h5>
            <iframe style="width: 100%" height="400"
               src="https://www.youtube.com/embed/8VXW4Uq_PC0?autoplay=1&mute=1">
            </iframe>
```



Slika 3. Prikaz linka i video sadržaja na web-stranici

Neka se sada želi omogućiti prelazak na kontroler predmeta putem meni trake na vrhu stranice. Kao što je prethodno objašnjeno, sama definicija meni trake nalazi se u **Shared** folderu i **Layout** stranici, gdje su definisana postojeća dva linka *Index* i *Privacy* (u **nav** kontroli u listi deklarisanoj putem kontrole **ul**). Potrebno je dodati novi element liste kojim će se definisati rutiranje na kontroler *Predmets*, tako da cijela lista dobiva izgled prikazan u isječku koda ispod. Novi element liste ima definisanu **asp-controller** vrijednost koja preusmjerava stranicu na *Predmets* kontroler, kao i **asp-action** vrijednost koja preusmjerava stranicu na *Index* stranicu ovog kontrolera.

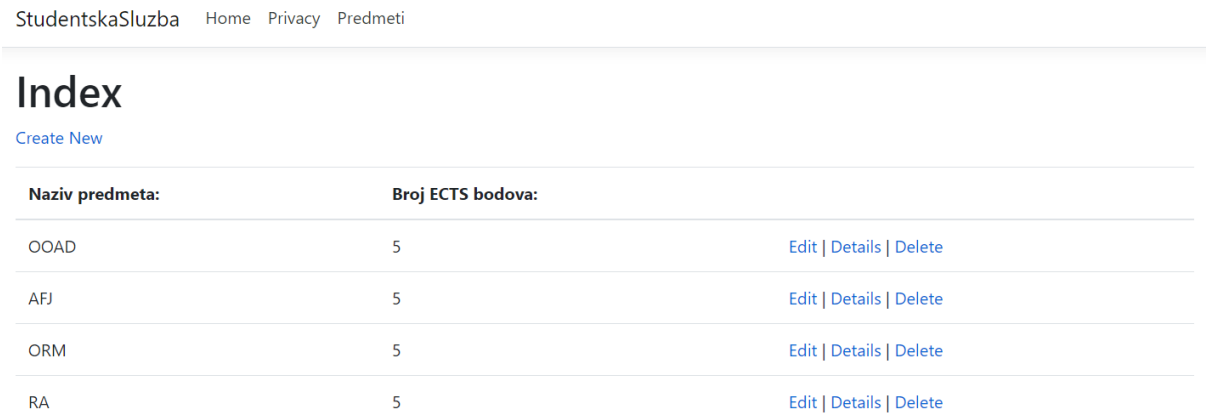
```
                    <ul class="navbar-nav flex-grow-1">
                        <li class="nav-item">
                            <a class="nav-link text-dark" asp-area="" asp-
controller="Home" asp-action="Index">Home</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link text-dark" asp-area="" asp-
controller="Home" asp-action="Privacy">Privacy</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link text-dark" asp-area="" asp-
controller="Predmets" asp-action="Index">Predmeti</a>
                        </li>
                    </ul>
```

Novi izgled liste sa dodatnim elementom vidljiv je u meniju prikazanom na Slici 4, a sada se klikom na taj element korisnik preusmjerava na *Index* stranicu kontrolera za predmete, koja je kreirana u prethodnoj laboratorijskoj vježbi i koja će se koristiti u nastavku.

Slika 4. Prikaz elementa koji je dodan u listu menija

2. *Binding* kontrola sa modelom podataka

Na Slici 5 prikazan je izgled *Predmets/Index* stranice. Ova stranica vrši prikaz svih predmeta koji se nalaze u predefinisanoj listi predmeta (budući da se u aplikaciji ne koristi perzistencija podataka). U CSHTML *file*-ovima moguće je specificirati vrijednosti elemenata koristeći **Razor sintaksu**, koja omogućava da se korištenjem specijalnog znaka **@** (uz eventualnu enkapsulaciju više linija koda koristeći vitičaste zagrade **{ }**) deklariše kod napisan u C# programskom jeziku koji se pri pokretanju aplikacije interpretira i daje statičke ili dinamičke vrijednosti koje se zatim dodjeljuju željenim kontrolama.



Slika 5. Predefinisani izgled Predmets/Index stranice

U programskom kodu ispod prikazana je deklaracija *Predmets/Index* stranice. Ova stranica sadrži sljedeće elemente definisane koristeći Razor sintaksu:

1. **@model** element, kojim se definiše varijabla *model*. Ovoj varijabli dodjeljuje se vrijednost **IEnumerable<StudentskaSluzba.Models.Predmet>**, čime se označava da je u pitanju kolekcija podataka tipa *Predmet*.

2. **@{ ViewData["Title"] = "Index"; }**, čime se definiše naziv stranice. Ovaj naziv prikazuje se u web-pretraživaču kao ime taba u kojem je stranica otvorena.

3. **@Html.DisplayNameFor(model => model.Naziv)**, čime se vrši prikaz naziva atributa za željenu varijablu. U konkretnom slučaju, vrši se prikaz definisanog naziva za atribut *Naziv* klase *Predmet*. Naziv za atribut definiše se putem stereotipa, što će biti prikazano u okviru validacije elemenata u sljedećem poglavlju vježbe.

4. **@foreach (var item in Model) { }**, čime se vrši iteracija kroz kolekciju podataka *model*.

5. **@Html.DisplayFor(modelItem => item.Naziv)**, čime se vrši prikaz vrijednosti atributa za željenu varijablu. U konkretnom slučaju, vrši se prikaz vrijednosti atributa *Naziv* za trenutni element klase *Predmet* u *foreach* petlji.

6. **@item.ID**, čime se dobavlja vrijednost ID atributa za trenutni element klase *Predmet* u *foreach* petlji. Ova vrijednost koristi se za rutiranje, što omogućava da se klikom na link *Edit* pošalje zahtjev na adresu *Predmets/Edit/<ID predmeta>*.

```
@model IEnumerable<StudentskaSluzba.Models.Predmet>

@{
    ViewData["Title"] = "Index";
}

<h1>Index</h1>

<p>
    <a asp-action="Create">Create New</a>
</p>
<table class="table">
    <thead>
        <tr>
            <th>
                @Html.DisplayNameFor(model => model.Naziv)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.ECTS)
            </th>
            <th></th>
        </tr>
    </thead>
    <tbody>
@foreach (var item in Model) {
        <tr>
            <td>
                @Html.DisplayFor(modelItem => item.Naziv)
            </td>
            <td>
```

```
                @Html.DisplayFor(modelItem => item.ECTS)
            </td>
            <td>
                <a asp-action="Edit" asp-route-id="@item.ID">Edit</a> |
                <a asp-action="Details" asp-route-id="@item.ID">Details</a> |
                <a asp-action="Delete" asp-route-id="@item.ID">Delete</a>
            </td>
        </tr>
}
    </tbody>
</table>
```

Na ovaj način moguće je poglede direktno povezati sa modelima podataka po želji, kao i vršiti ekstrakciju i prikaz atributa i njihovih naziva, a zatim vršiti izmjene na modelu koristeći rutiranje uz slanje parametara zahtjeva.

3. Validacija modela

Još jedna važna osobina ASP.NET Core MVC koju je potrebno iskoristiti pri definisanju aplikacije je mogućnost dodatne definicije modela podataka. Pod tim se misli na dodatno definisanje osobina atributa kao što su: označavanje polja obaveznim, definisanje imena atributa za prikaz, dozvoljena dužina, format i sl. Neke od ovih osobina već su demonstrirane u prethodnoj vježbi, pri čemu su korištene **Key**, **Required** i **NotMapped** anotacije. U nastavku će biti objašnjeno korištenje drugih vrsta anotacija[4] za što bolju validaciju modela podataka.

Kako je ID predmeta primarni ključ koji se automatski definiše u konstruktoru modela, za ovaj atribut ne definišu se nikakve anotacije za validaciju. Naziv predmeta predstavlja string od minimalno 3, a maksimalno 50 karaktera, pri čemu je dozvoljeno samo korištenje velikih, malih slova engleskog alfabeta, brojeva i razmaka. Broj ECTS bodova mora biti između 1 i 10. Potrebno je i odabrati proizvoljna imena za prikaz za sve atribute. Sva ova validacija dodaje se koristeći anotacije prikazane u isječku koda ispod. Za definisanje raspona za brojčane vrijednosti koristi se **Range** stereotip. Za definisanje raspona za stringove koristi se **StringLength**, a za definisanje dozvoljenog formata podataka **RegularExpression** stereotip. Za definisanje imena atributa za prikaz na formama koristi se **DisplayName** stereotip.

```
        #region Properties

        [Key]
        [Required]
        [DisplayName("ID predmeta:")]
        public int ID { get; set; }

        [Required]
        [StringLength(maximumLength: 50, MinimumLength = 3, ErrorMessage = "Naziv
predmeta smije imati između 3 i 50 karaktera!")]
        [RegularExpression(@"[0-9| |a-z|A-Z]*",
         ErrorMessage = "Dozvoljeno je samo korištenje velikih i malih slova,
brojeva i razmaka!")]
        [DisplayName("Naziv predmeta:")]
```

[4] Lista svih anotacija dostupna je na sljedećem linku: https://docs.microsoft.com/en-us/aspnet/core/mvc/models/validation?view=aspnetcore-5.0.

```
        public string Naziv { get; set; }

        [Required]
        [Range(1.0, 10.0, ErrorMessage = "Broj ECTS bodova mora biti između 1 i
10!")]
        [DisplayName("Broj ECTS bodova:")]
        public double ECTS { get; set; }

        [NotMapped]
        public List<Student> UpisaniStudenti { get; set; }

        #endregion
```

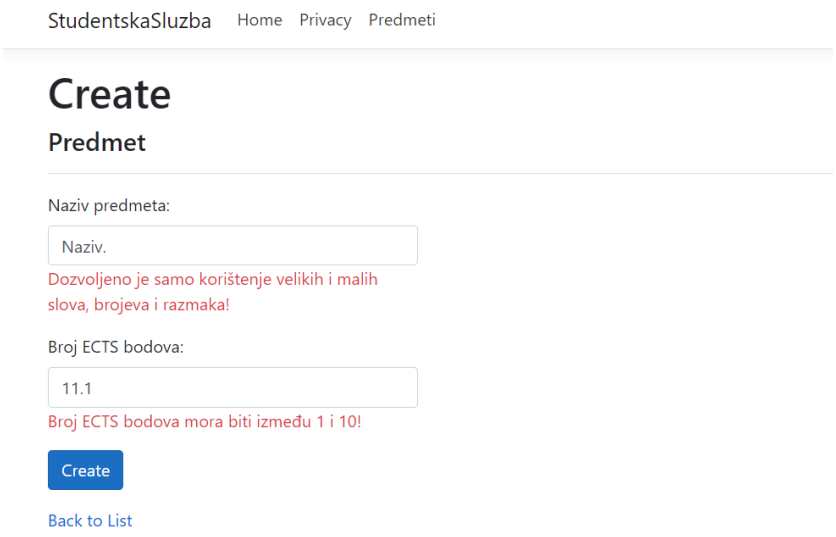
Promjene koje su izvršene u modelu podataka automatski se propagiraju na prethodno generisane kontrolere, tako da ih nije potrebno nanovo automatski generisati. Važno je napomenuti da se preporučuje korištenje pristupa gdje se prvo definiše validacija, a zatim *scaffold* kontrolera i pogledi kako bi se izbjegla pojava bilo kakvih problema.

Validacija polja na formi definisana je u okviru kontrola koje omogućavaju korisnički unos. Te kontrole nalaze se na pogledu *Predmets/Create*. Definicija ovog pogleda prikazana je u isječku koda ispod i vidljivo je da se kao model definiše element klase *Predmet*. Ključni elementi kojima se definiše validacija su **asp-validation-summary** kojim se specificira prikaz validacijskih poruka samo na osnovu anotacija definisanih u modelu, kao i **asp-validation-for** kojima se definišu pojedinačne validacije za atribute modela. Validacijske poruke prikazuju se ispod elemenata za unos, što je i definisano na formi. Na dnu pogleda nalazi se definicija kojom se učitavaju skripte za prikaz validacije. Na ovaj način omogućava se prikaz poruka validacije za polja za koja se vrši unos pri kreiranju novog predmeta.

```
@model StudentskaSluzba.Models.Predmet

@{
    ViewData["Title"] = "Create";
}

<h1>Create</h1>

<h4>Predmet</h4>
<hr />
<div class="row">
    <div class="col-md-4">
        <form asp-action="Create">
            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
            <div class="form-group">
                <label asp-for="Naziv" class="control-label"></label>
                <input asp-for="Naziv" class="form-control" />
                <span asp-validation-for="Naziv" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="ECTS" class="control-label"></label>
                <input asp-for="ECTS" class="form-control" />
                <span asp-validation-for="ECTS" class="text-danger"></span>
            </div>
            <div class="form-group">
                <input type="submit" value="Create" class="btn btn-primary" />
            </div>
```

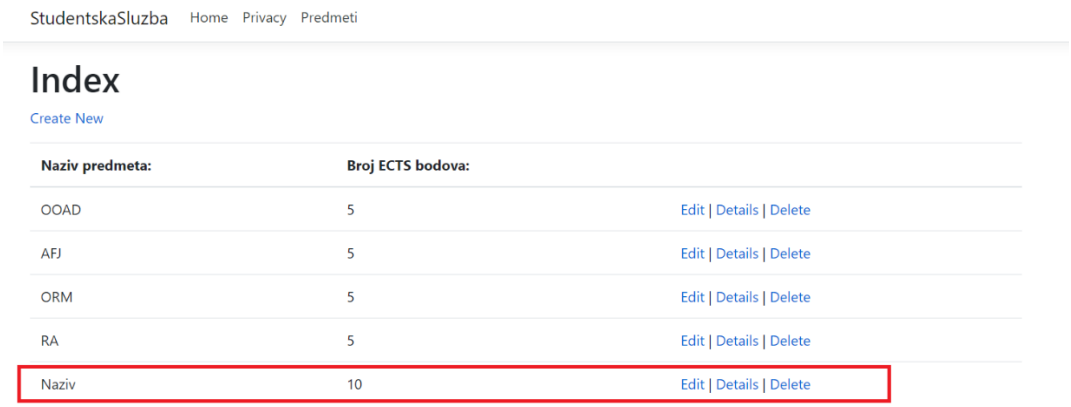
```
        </form>
    </div>
</div>

<div>
    <a asp-action="Index">Back to List</a>
</div>

@section Scripts {
    @{await Html.RenderPartialAsync("_ValidationScriptsPartial");}
}
```

Kao što je prethodno objašnjeno, polje ID predmeta, koje predstavlja njegov primarni ključ i koje se automatski generiše u konstruktoru, nije prikazano na formi i nije dozvoljen manuelni unos vrijednosti za ovo polje. Ukoliko se isto želi dozvoliti, potrebno je ručno dodati polja na formu za dodavanje predmeta. Izgled forme sa validacijskim porukama za neispravan unos vrijednosti u preostala polja (naziv i ECTS) prikazan je na Slici 6.



Slika 6. Prikaz validacijskih poruka na formi

Nakon što se izvrši ispravna specifikacija podataka i izvrši klik na dugme **Create**, poziva se POST metoda *Create* kontrolera *Predmets* u okviru koje se vrši dodavanje novog predmeta. Nakon toga dobiva se prikaz kao na Slici 7, odakle je vidljivo da je novi predmet dodan u listu.



Slika 7. Prikaz stranice sa svim predmetima gdje je dodan novi predmet

4. Zadaci za samostalni rad

Programski kod aplikacije koja je kreirana u vježbi dostupan je u repozitoriju na sljedećem linku: https://github.com/ehlymana/OOADVjezbe, na *branchu* **lv7**.

1. Definisati stranicu *Privacy* u aplikaciji koja je kreirana u vježbi. Ova stranica treba posjedovati dvije slike: logo UNSA i logo ETF koje se nalaze na sredini stranice jedna pored druge u ravni. Iznad prve treba pisati tekst „Univerzitet u Sarajevu“, a iznad druge „Elektrotehnički fakultet“. Ispod slika potrebno je dodati proizvoljan opis i linkove koji vode do zvaničnih stranica UNSA i ETF.

2. Napraviti preraspodjelu stavki menija za navigaciju tako da taj meni ima dva elementa: *Početna* (koji vodi do stranice *Home/Index*) i *Predmeti* (koja vodi do stranice *Predmets/Index*). Na početnu stranicu dodati meni ispod menija za navigaciju koji omogućava pristup *Index* i *Privacy* stranicama.

3. Dodati validaciju za klase *Student* i *UpisNaPredmet* po želji. Provjeriti da li se sve validacije ispravno prikazuju na formama.

4. Omogućiti prikaz ID vrijednosti za predmete na svim formama. Da li se ID automatski mijenja pri dodavanju predmeta? Šta je potrebno uraditi kako bi se to izvršilo?