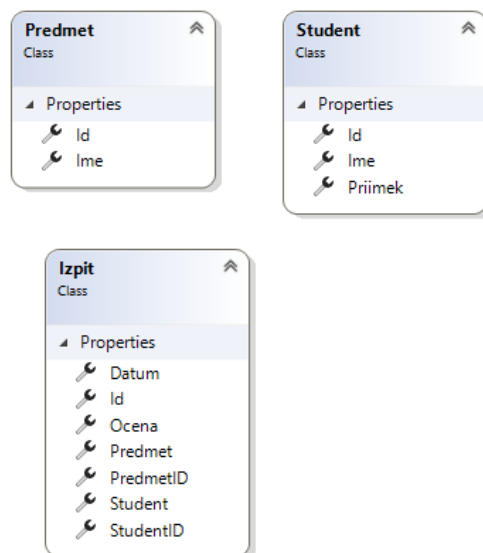


Vaja MVC s Code first

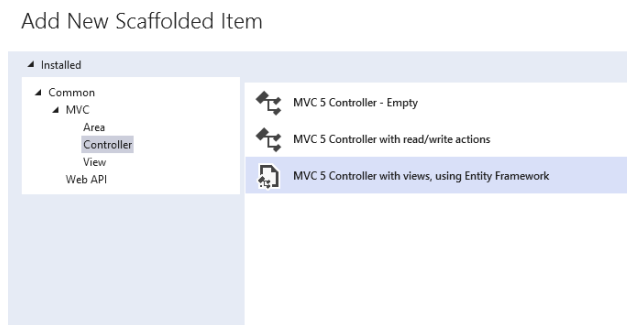
1. Izdelajte novo MVC aplikacijo. V aplikacijo, v mapo Models, dodajte 3 razrede, kot je na spodnjem diagramu



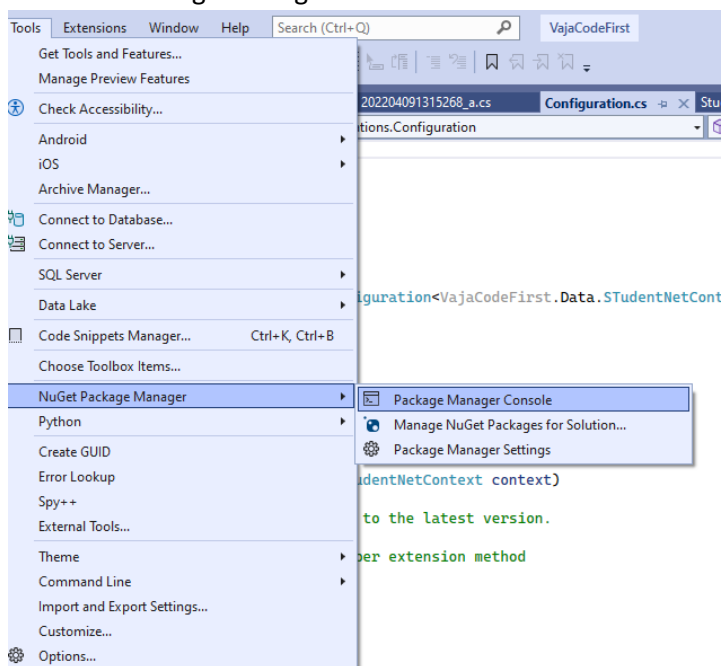
V aplikaciji želimo opisati študente, predmete in ocene študenta pri posameznem predmetu. Na spodnji sliki vidite še tipe lastnosti razreda Izpit. Spremenljivkama Predmet in Student dajte v glavo še besedico virtual.

Name	Type	Modifier
Methods		
<add method>		
Properties		
Datum	DateTime	public
Id	int	public
Ocena	int	public
Predmet	Predmet	public
PredmetID	int	public
Student	Student	public
StudentID	int	public
<add property>		
Fields		

2. Aplikacijo prevedite. Nato dodajte kontroler Studenti, tako da uporabite MVC5 Controller with Views using Entity framework



3. Poženite aplikacijo, da vam generira podatkovno bazo, v brskalnik vpišite naslov <http://localhost:xxxxx/Studenti> (kjer xxx nadomestite s številko vrat).
4. Vnesite vsaj dva študenta.
5. Nato dodajte na enak način še kontroler za Predmete. Ko boste poizkusili pognati aplikacijo, boste dobili napako. Vaša podatkovna baza se ne ujema s trenutnim stanjem v kontekstu, zato morate narediti t.i. migracijo.
6. Zaženete Package manager Console:



Omogočite migracije

```
PM> Enable-Migrations
```

Dodate migracijo

```
PM> Add-Migration a
```

In posodobite podatkovno bazo

```
PM> Update-Database
```

7. Preverite, če program sedaj pravilno deluje (dodajte vsaj dva predmeta).

8. Na koncu bi želeli dodati še kontroler za vnos izpitov. Postopamo enako kot pri dodajanju predmetov. Dodamo kontroler IzpitiController, nato pa dodamo novo migracijo (drugi in tretji ukaz v PM> konzoli, pri čemer ime migracije – prej a - spremenimo, na primer v a1)
9. Če poženete aplikacijo, bo sicer delovala, a ni prav taka, kot si jo želimo. V vnosu izpitov nimamo možnosti vnosa študenta, pa tudi predmeta ne. Najlažje bo, če izbrišete kontroler Izpiti in mapo Izpiti v Views mapi, nato pa kontroler še enkrat poženete (prepoznal bo tuje ključne in dodal ustrezno kodo).
10. Vnesite nekaj izpitov.
11. Na koncu urediti aplikacijo tako, da bo menijska vrstica delovala za primer ŠtudentNeta. (Zamenjajte Home, About, Contact s Študenti, Izpiti, Predmeti) in ustrezno popravite kodo v _Layout.cshtml.
12. Uredimo še vnose datumov in števil. Želeli bi seveda, da lahko vnašamo datume v slovenskem zapisu (13.1.2022) in ne v ameriškem (1/13/2022). Tudi realna števila bi radi vnašali z decimalno vejico. V ta namen si najprej prek NugetPackage managerja dodamo knjižnico jQuery.Validation.Globalize. Nato pa si v razred BundleConfig.cs dodamo kodo (rumeno pobarvane vrstice so že dodane, dodamo samo vmesno kodo):

```

bundles.Add(new ScriptBundle("~/bundles/jqueryval").Include(
    "~/Scripts/jquery.validate*"));
bundles.Add(new ScriptBundle("~/bundles/jqueryval").Include(
    "~/Scripts/cldr.js"));
bundles.Add(new ScriptBundle("~/bundles/jqueryval").Include(
    "~/Scripts/globalize.js"));
bundles.Add(new ScriptBundle("~/bundles/jqueryval").Include(
    "~/Scripts/globalize/date.js"));
bundles.Add(new ScriptBundle("~/bundles/jqueryval").Include(
    "~/Scripts/globalize/number.js"));
bundles.Add(new ScriptBundle("~/bundles/jqueryval").Include(
    "~/Scripts/jquery.validate.globalize.js"));
// Use the development version of Modernizr to develop with and
learn from. Then, when you're
// ready for production, use the build tool at
https://modernizr.com to pick only the tests you need.
bundles.Add(new ScriptBundle("~/bundles/modernizr").Include(
    "~/Scripts/modernizr-*"));

```

13. Preverite sedaj vnos datumov in realnih števil (na primer pri vnosu novega izpita ali vnosu ocene)
14. Za tiste, ki jim ostaja čas: Izdelajte si nov kontroler, kjer boste izpisovali ocene za posameznega študenta. Na strani izberete študenta, nato pa na isti strani izpišete njegove ocene.