# Uvod

## Visual Studio Team Servics (VSTS)

Jedan korisnik (privatni Microsoft account npr [neko@hotmail.com](mailto:neko@hotmail.com) ili poslovni account-član domene, npr. [neko@edu.fit.ba](mailto:neko@edu.fit.ba)) može biti član na više Visual Studio Team Services (VSTS) accounts sa različitim ulogama (owner ili member).  
VSTS accounts ima poddomenu nesto.visualstudio.com i može imati besplatno 5 korisnika.  
Jedan VSTS accounts može imati više projekata, npr. nesto.visualstudio.com/projekt1  
Jedan projekat može imati više repozitorija.

1 - google: visual studio online -> prvi link  
<https://www.visualstudio.com/vso/>

2 - login sa edu-fit-ba emailom  
napraviti VSTS account, npr. poddomena.visualstudio.com tipa "AGILE" te "GIT" kao version control  
Za seminarski rad potrebno je da svaka grupa napravi VSTS account prema sljedećem spisku   
<https://edufit-my.sharepoint.com/:x:/g/personal/adil_fit_ba/ESWfRAybUn1HvUFgwUkQnGoBsoCknAwJFMCzD_JiqwhRAA?e=2a40aff36f1847cc866e841740356d11>

3. dodati testni projekt, npr. proba1  
<https://fitba1701.visualstudio.com/proba1>

4. dodati ostale članove grupe u sekciji users   
Odabrati Basic kao Access Level te "Project Administrators" za VSTS Group  
<https://fitba1701.visualstudio.com/_admin/_users>

5. kao članove VSTS accounta dodati  
[adil@fit.ba](mailto:adil@fit.ba)  
[denis@fit.ba](mailto:denis@fit.ba)

update sept 2018:

On September 10, 2018, Microsoft renamed **Visual Studio Team Services (VSTS)** to **Azure DevOps Services**. For more information about this change, see [this blog post](https://aka.ms/azurevsts).

umjesto: **neki-naziv.visualstudio.com** sada je **dev.azure.com/neki-naziv**

# Osnove C#

Osnove C# - Razlike C++ i C#  
<https://github.com/adil-fit-ba/rs1-2017-18/tree/master/02.1_Csharp_razlike>

# ORM, Entityframework 1

## Analizirati: Pristup DB-u na klasičan način (ADO.net)

<https://github.com/adil-fit-ba/rs1-2017-18/tree/master/03.1_ado.net>

## Pogledati Delegati in C#

<http://www.tutorialsteacher.com/csharp/csharp-anonymous-method>

Primjer c++  
<https://github.com/adil-fit-ba/rs1-2017-18/tree/master/03.2_delegati_cpp>

Pogledati Lamda Expression in C#

<http://www.tutorialsteacher.com/linq/linq-lambda-expression>

<http://www.tutorialsteacher.com/Content/images/linq/lambda-expression-1.png>  
<http://www.tutorialsteacher.com/Content/images/linq/lambda-expression-2.png>

## Analizirajte code (Priprema\_za\_ORM - Object-relational mapping)

<https://github.com/adil-fit-ba/rs1-2017-18/tree/master/03.3_dotnet_stari_Priprema_za_ORM>

## Analizirajte code (Implementacija\_vlastitog\_ORMa)

<https://github.com/adil-fit-ba/rs1-2017-18/tree/master/03.4_dotnet_stari_Implementacija_vlastitog_ORMa>

## Implementirajte .net core console aplikaciju

a. Napraviti novi dot.net core console projekat

b. Instalirati EF (nuget installer)  
Microsoft.EntityFrameworkCore  
Microsoft.EntityFrameworkCore.SqlServer  
Microsoft.EntityFrameworkCore.Tools

Pokrenuti aplikaciju

Stavke a i b se mogu izvršiti sa CLI unutar novog foldera:

**za .net core 2.0**  
dotnet new console   
dotnet add package Microsoft.EntityFrameworkCore   
dotnet add package Microsoft.EntityFrameworkCore.SqlServer   
dotnet add package Microsoft.EntityFrameworkCore.Tools   
dotnet restore   
dotnet run

c. Dodati klase u novi folder "model": Kupac i Opstina

d. Definisati novu klasu koja nasljeđuje DbContext, te dodati property DbSet<NazivModelKlase> za svaku tabelu

e. U klasu DbContext dodati  
protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder) {  
optionsBuilder.UseSqlServer("Server=10.10.10.18;Database=BrojIndeksa;Trusted\_Connection=False;MultipleActiveResultSets=true;User ID=sa;Password=test");  
}

f.  
Ukoliko želite koristiti CLI (cmd za EF):   
u ".csproj" unutar taga <ItemGroup> ...</ItemGroup> dodati

**za .net core 2.0**  
<DotNetCliToolReference Include="Microsoft.EntityFrameworkCore.Tools.DotNet" Version="2.0.1" />  
zatim uradite build prjekta

g. za generisanje migracije (skripte za kreiranje i modifikaciju tabela)   
CLI  
**dotnet ef migrations add InicijalnaDB**

ili package manager komanda

**add-migration InicijalnaDB**

Pogledati sadržaj foldera Migrations

h. izvršavanje migracije (prethodno generisane migracije/skripte)

CLI  
**dotnet ef database update**

ili package manager komanda

**update-database**

i. pokrenuti aplikaciju preko cmd  
dotnet run

j. Testiranje EF: Dodati novi objekat Opstine u DB

k. Testiranje EF: Dodati novi objekat Opstine i studenta u DB

l. Testiranje EF: Ispisati sve studente i njihove opstine

Rješenje:  
<https://github.com/adil-fit-ba/rs1-2017-18/tree/master/03.5_ConsoleApp2a>

Online nastava 24.10.2017  
<https://youtu.be/2c1t43B-ILM>

# Entityframework 2

Rješenje zadataka 3. sedmice do zadatka 6i  
<https://github.com/adil-fit-ba/rs1-2017-18/tree/master/04.1_ConsoleApp3>

tutorijali  
<http://www.learnentityframeworkcore.com/walkthroughs/console-application>

Zadatak 1:

Svaki student treba da ponovi korake 6a do 6h koje su prethodno opisane. Student treba da implementira aplikaciju koja će imati minimalno tri entity klase koje su međusobno povezane (klase odabrati po svom izboru).

Prilikom implementacije konzolne aplikacije dodati izbornik, npr  
a) dodavanje i prikaz opština  
b) dodavanje kupca sa odabirom postojećih opština  
c) dodavanje kupca sa unosom nove opštine  
d) prikaz kupaca i njihovih opština

Možete pogledati rješenje od prošle sedmice. Link se nalazi iznad.

Evo još primjera v2  
<https://github.com/adil-fit-ba/rs1-2017-18/tree/master/04.2_ConsoleApp2b>

Rješenje sa online nastave  
<https://github.com/adil-fit-ba/rs1-2017-18/tree/master/04.3_ConsoleApp9>

Online nastava snimak  
<https://youtu.be/RMgKZDcvQlE>

# Asp.net core mvc – 1

Napraviti asp.net core mvc aplikaciju - koristiti sample MVC

- dodati EF core  
- implementirati akciju i view za prikaz klijenata

<https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/adding-view>

online nastava - video  
<https://youtu.be/M_qBIFV0dao>

<https://github.com/adil-fit-ba/rs1-2017-18/tree/master/05.2_Mvc_2017-11-11>

# Asp.net core mvc - 2

Online 04.12.2017 video i coder\*  
<https://youtu.be/csqhyJD5PBs>

<https://github.com/adil-fit-ba/rs1-2017-18/tree/master/06.1_Mvc_2017-12-04>

# Asp.net core mvc - 3

## Primjer html forme

* StudentiController/Dodaj1 i Snimi1 – forma bez ViewModela, “not strongly typed forms”
* StudentiController/Dodaj2 i Snimi2 – forma sa ViewModelom, “strongly typed forms”

<https://github.com/adil-fit-ba/rs1-2017-18/tree/master/07.1_primjer_linkova_i_formi>

## “classic” vs “mvc html helper” vs “mvc tag helper”

### Primjer "linkova"

**classic html - bez helpera**  
<a href="kontroler/akcija?parametar1=vrijednost1&parametar2=vrijednost2" class="btn btn-warning" style="border:dashed">Opis</a>

**mvc html helpers**  
@Html.ActionLink("Opis", "Akcija", "Kontroler", new {parametar1=vrijednost1, parametar2=vrijednost2}, new {@class="btn btn-warning", style="border:dashed"})  
<http://www.dotnettricks.com/learn/mvc/understanding-html-helpers-in-aspnet-mvc>

**mvc tag helpers**  
<a asp-controller="Account" asp-action="Register" asp-route-parametar1="vrijednost1" asp-route-parametar2="vrijednost2">Register</a>  
<https://docs.microsoft.com/en-us/aspnet/core/mvc/views/tag-helpers/intro>

### Primjer "html forme"

**classic html - bez helpera** | **mvc html helpers | mvc tag helpers **

<https://github.com/adil-fit-ba/rs1-2017-18/blob/master/07.1_primjer_linkova_i_formi/usporedba_forma.cshtml>

Primjer

<https://github.com/adil-fit-ba/rs1-2017-18/tree/master/07.2_primjer_mvc_strongly_typed_forme_A>

<https://github.com/adil-fit-ba/rs1-2017-18/tree/master/07.3_primjer_mvc_strongly_typed_forme_B>

# Seminarski rad – zadatak 5.2

a) Kreirajte Visual Studio Solution sa projekta:  
- ASP.NET MVC core projekat sa imenom "Nesto.Web"  
- Class Library projekat sa imenom "Nesto.Data"  
  
b) U folder **Models** dodajte entity klase. Entity klase pripremite tako što ćete pisati code ručno (prateći domain model ili dijagram klasa). Preporuka je da ne koristite opcije unutar UML alata za eksport koda iz dijagrama klasa.  
Napomena: Visual Paradigm Community (besplatna verzija) nema opciju za generisanje koda iz dijagrama.  
  
  
c) Dodajte **EntityFramework Core** reference putem **NuGet package manager**.  
  
d) Po potrebni konfigurisati Entitiy model preko Fluent API-a (funkcija **OnModelCreating**u izvedenoj klasi od **DBContext**-a)  
npr. ukoliko se prilikom izvršenja migracijske skripte (update-database) pojavi greška "Cycle Cascade Insert, Update, Delete" ili sl. onda uklonite Cascade na sljedeći način:  
  
  modelBuilder.Entity<Student>()

                .HasOne(pt => pt.Opstina)

                .WithMany()

                .HasForeignKey(pt => pt.OpstinaId)

                .OnDelete(DeleteBehavior.Restrict);

e) Na  Microsoft SQL Serveru  na app.fit.ba napraviti praznu bazu podataka na MS SQL sa nazivom p1700 (umjesto 00 stavite broj vašeg projekta, pročitajte upute za plesk <https://www.fit.ba/forum/yaf_postst1727_Upute-za-Plesk.aspx>)

Brojevi projekata su objavljeni u ovom dokumentu u poglavlu 1.1 Visual Studio Team Servics (VSTS).  
  
  
f) Dodajte connection string u fajl**appsettings.json, npr.**

 "ConnectionStrings": {

    "lokalni1": "Server=localhost;Database=RS1\_online1;Trusted\_Connection=true;MultipleActiveResultSets=true;User ID=;Password=",

    "adil-pc": "Server=10.10.10.18;Database=RS1\_online1;Trusted\_Connection=false;MultipleActiveResultSets=true;User ID=sa;Password=test",

    "fit-server1": "Server=app.fit.ba,1433;Database=RS1\_online1;Trusted\_Connection=false;MultipleActiveResultSets=true;User ID=nekiUser;Password=nekiPass"

  }

g) Dodajte klasu koja nasljeđuje **DBContext**. Parametar konstruktora iz bazne klase treba da sadrži naziv connection stringa iz konfiguracijskog objekta

**public MojContext(DbContextOptions<MojContext> options)**

**: base(options)**

**{**

**}**

u funkciju **StartUp::ConfigureServices** dodajte

*services.AddDbContext<MojContext>(options =>*

*options.UseSqlServer(Configuration.GetConnectionString("lokalni1")));*

h)  dodajte migraciju i izvršite skriptu (update-database)  
Ukoliko je sve ispravno konfigurisano, bit će kreirane tabele u bazi podataka. Provjerite tabele preko SQL Server Managment Studio-a

i) Dodajte HomeController. U funkciju Index dodajte  code za pristup bazi podataka. Npr. Preuzeti sve podatke iz neke tabele. Pokrenite web aplikaciju.

i) Jedan član grupe treba postaviti Visual Studio projekat na novi git repoziroj unutar vašeg VSTS team projekta. Ostali članovi trebaju klonirati repozitorij na svoj lokalni računar.

**Video sa online nastave na kojoj je riješen zadatak 5.2.2.** (22.12.2017.)  
<https://youtu.be/35IcfQxPQt4>

<https://github.com/adil-fit-ba/rs1-2017-18/tree/master/08.1_Mvc_2017-12-22>

# Video – Zadatak sa praktičnog dijela ispita (1 od 3)

code + video

<https://github.com/adil-fit-ba/rs1-2017-18/tree/master/09.1_mvc_video1_ispit1>

<https://youtu.be/A4GztNh5ZfE>

# Video – Zadatak sa praktičnog dijela ispita (2 od 3)

code + video

<https://github.com/adil-fit-ba/rs1-2017-18/tree/master/10.1_mvc_video2_ispit2>

<https://youtu.be/61rmAms13Oo>

# Video – Zadatak sa praktičnog dijela ispita (3 od 3)

code + video

<https://github.com/adil-fit-ba/rs1-2017-18/tree/master/11.1_mvc_video3_Ispit3>

<https://youtu.be/dUG7OHDejjs>

# Validacija unosa

### Validacija

<https://docs.microsoft.com/en-us/aspnet/core/mvc/models/validation>

video

<https://youtu.be/g4juaF7GYaM>

### Regex sintaksa

<https://ryanstutorials.net/regular-expressions-tutorial/regular-expressions-basics.php>

**Regular expressions** are used in [search engines](https://en.wikipedia.org/wiki/Search_engine), search and replace dialogs of [word processors](https://en.wikipedia.org/wiki/Word_processor) and [text editors](https://en.wikipedia.org/wiki/Text_editor), in text processing utilities such as [sed](https://en.wikipedia.org/wiki/Sed) and [AWK](https://en.wikipedia.org/wiki/AWK) and in [lexical analysis](https://en.wikipedia.org/wiki/Lexical_analysis). Many [programming languages](https://en.wikipedia.org/wiki/Programming_language) provide regex capabilities, built-in or via [libraries](https://en.wikipedia.org/wiki/Library_(computing)).

# PartialView i View Components

video

<https://youtu.be/yEI2NY7K8kA>

## PartialView

<https://docs.microsoft.com/en-us/aspnet/core/mvc/views/partial>  
<https://github.com/adil-fit-ba/rs1-2017-18/tree/master/13.1_mvc-video5_PartialViews>

ViewComponent

<https://docs.microsoft.com/en-us/aspnet/core/mvc/views/view-components>

<https://github.com/adil-fit-ba/rs1-2017-18/tree/master/13.2_mvc-video5_ViewComponents>

# Autentifikacija i Autorizacija

video

<https://youtu.be/cjsyzLD1Vyo>

## Session

<http://benjii.me/2016/07/using-sessions-and-httpcontext-in-aspnetcore-and-mvc-core/>

## ActionFilters

<http://www.c-sharpcorner.com/article/working-with-filters-in-asp-net-core-mvc/>

# AREA za asp.net core

video

<https://youtu.be/14EdQYgE1wg>

<https://tahirnaushad.com/2017/08/25/asp-net-core-2-0-mvc-areas/>

# Dodatno

## Način za unaprijeđene asp.net core aplikacije

<http://asp.net-hacker.rocks/2016/02/18/extending-razor-views.html>

## Jquery

<https://api.jquery.com/>  
<https://www.w3schools.com/jquery/default.asp>

## EF core i connection string

<https://jessedotnet.com/2016/12/29/asp-net-core-injecting-your-db-context-into-your-controllers/>

## Error logs

Kako aktivirati ispis error loga u fajlove na serveru.

<https://scottsauber.com/2017/04/10/how-to-troubleshoot-an-error-occurred-while-starting-the-application-in-asp-net-core-on-iis/>

## Korisni alati

### WebApi

<https://docs.microsoft.com/en-us/aspnet/core/tutorials/web-api-help-pages-using-swagger?tabs=visual-studio>

### Upload fajlova

<https://haacked.com/archive/2010/07/16/uploading-files-with-aspnetmvc.aspx/>

# Git

### Osnove

Naučite git [https://try.github.io](https://try.github.io/)

PPT

<https://courses.cs.washington.edu/courses/cse403/13au/lectures/git.ppt.pdf>

### git proba

Link git repozitorija može se pronaći na VSTS-u, npr. <https://nekoime.visualstudio.com/_git/proba1>

1. otvoriti cmd i navigirati na desktop

2. klonirati git repozitorij  
**git clone nekoime.visualstudio.com/\_git/proba1  
cd proba1**

3. dodati sadržaj na repozitorij  
a) dodati podfolder **zadaci** i   
**md zadaci**

b) ubaciti testne docx fajlove (npr. RS1 zadaci 1 ... 4)

4) dozvoli da git prati sve fajlove, otvoriti root folder

**cd..**  
**git add .**

5) izvršiti commit i push **git commit -m"Poruka odnosno komentar"  
  
git push**

pogledati fajlove na git-repozitoriju u web browseru

### git i VS projekat

VS projekat sadrži neke fajlove (temporary files, build results, itd.) koje ne treba postaviti na git repozitorij fajlovi . U “.**gitignore”** fajlu se može definisat koje šta sve ne treba dodati na git.

Ukoliko preko VS-a kreirate repozitoj, onda će VS kreirati sljedeći gitignore fajl.

6) Preuzmite defaultni gitignore fajl i dodajte ga u korijen git repozitorija.

<https://raw.githubusercontent.com/adil-fit-ba/rs1-2017-18/master/.gitignore> (desni klik -> Save As ili Save Link As)

7) dodati podfolder web\_aplikacija1, napraviti novi VS asp.net core MVC projekat,

**md web\_aplikacija1**

**cd web\_aplikacija1  
dotnet new mvc**

Testirati projekat.

8) dozvoli da git prati sve fajlove (osim što je navedeno u gitignore), otvoriti root folder  
**cd..**

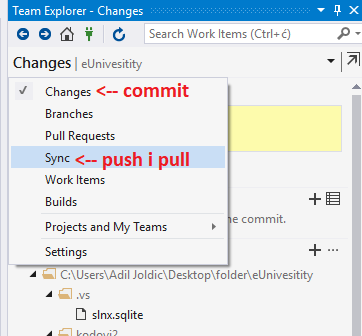
**git add .**

9) izvršiti commit i push **git commit**upisati komentar, ctrl+c, upisati ":exit", tipka enter **git push**

6) pogledati fajlove u web browseru

8) modificirajte projekt u VS-u

9) izvršiti **commit** i **push** kroz VS – Team Explorer prozor.



10) testirati git na drugim računaru (ili na istom računaru ali uraditi clone u drugi folder)

### Git Hosting

Popis git javnih hosting provajdera.

<https://git.wiki.kernel.org/index.php/GitHosting>

| **Provider** | **Framework is open-source?** | **Support for other SCM** | **Open-source repositories** | **Space (GB)** | **Free private repositories** |
| --- | --- | --- | --- | --- | --- |
| [**Assembla**](http://www.assembla.com/catalog/51-free-private-git-repository-package?type=private&ad=git-wiki) | No | SVN/Hg/P4 | Yes | 0.15 | 1 project, 3 users |
| [**Beanstalk**](http://beanstalkapp.com/?ad=git-wiki) | No | SVN | No | 0.1 | 1 projects, 1 user |
| [**bitbucket.org**](http://bitbucket.org/) | No | Mercurial | Yes | Unlimited | Unlimited projects, 5 collaborators |
| [**Codetidy**](https://codetidy.net/) | No | No | No | 0.1 | 5 repositories, 5 collaborators |
| [**Codebase**](http://codebasehq.com/) | No | Mercurial/SVN | Public access available | 0.05 | 1 project (unlimited repos), 2 collaborators |
| [**CloudForge**](http://www.cloudforge.com/) | No | CVS/SVN | Yes | 0.2 | 1 user only |
| [**Deveo**](https://deveo.com/) | No | Mercurial/SVN | No | Unlimited | Unlimited projects, 6 collaborators |
| [**GitEnterprise**](http://www.gitenterprise.com/) | No | No | No | 1 | Unlimited projects, 10 collaborators |
| [**GitHub**](http://github.com/) | No | SVN | Yes | Unlimited | No |
| [**GitLab.com**](https://about.gitlab.com/gitlab-com/) | Yes | No | Yes | Unlimited | Unlimited projects, unlimited collaborators |
| [**Pikacode**](http://pikacode.com/) | No | Mercurial | Yes | 1 | No |
| [**ProjectLocker**](http://www.projectlocker.com/) | No | SVN | Read-only http | 0.2 | 1 project, 2 collaborators |
| [**repo.or.cz**](http://repo.or.cz/) | Yes | No | Yes | 0.4 | No |
| [**RocketGit**](https://rocketgit.com/) | Yes | No | Yes | Unlimited | Unlimited |
| [**SourceForge.net**](http://sourceforge.net/) | Yes | Hg, SVN | Yes | Unlimited | No |
| [**Unfuddle**](http://unfuddle.com/) | No | SVN | Yes | 0.2 | 1 project, 2 collaborators |
| [**Visual Studio Team Services**](https://visualstudio.com/team-services/) | No | TFVC | No | Unlimited | Unlimited, 5 users |

### Repozitorij za predmet RS1-2017-18

Preuzimite repozitorij RS1-2017-18 sa <https://github.com/adil-fit-ba/rs1-2017-18>

