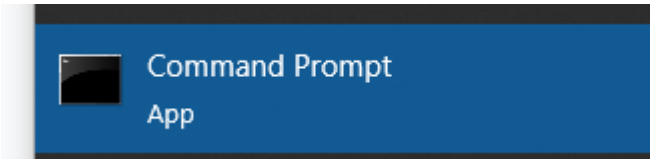
Vježba 2 - Testni Projekt

Uvod

U nastavku je ukratko opisana struktura C# MVC Web projekta, po direktorijima.

- Opći dio
  - **`Properties`** - manifest aplikacije - postavke za kompajliranje koda (assembly)
  - **`References`** - Povezani moduli
  - **`App_start`** - Funkcionalnosti za inicijalizaciju aplikacije kod pokretanja, veze prema js i css datotekama, preusmjeravanje zahtjeva klijenta (routing)
  - **`Content`** - Statičke datoteke za web klijent (css)
  - **`fonts`** - posebni znakovi
  - **`Scripts`** - Javascript datoteke za web klijent
  - **`Web.config`** datoteka - postavke web stranice i veze prema bazi podataka, dodatne postavke kompajliranja
  - **`Global.asax`** datoteka - pokreće se kod inicijalizacije aplikacije, npr. registrira rute za zahtjeve klijenta.
- MVC dio
  - **`Models`** - prikaz podataka kao C# klase
  - **`Views`** - Prikaz korisničkog sučelja u markup html-u
  - **`Controllers`** - Pristupna točka - temeljem poziva i poslanih parametara aplikacija vraća view
- ORM dio
  - **`DataAccess`** - funkcionalnosti za dohvat podataka iz baze
    - **`Repositories`** - standardne metode koje se pozivaju iz aplikacije, neovisno implementaciji baze podataka - cilj je apstrahirati implementaciju baze.
      - Add - dodavanje novog entiteta u bazu, prima instancu modela koji se dodaje
      - GetById - dohvat prema ID-ju - vraća instancu modela
      - GetAll - vraća listu (prebrojivu klasu) modela
      - Update - prima ID i instancu modela koji se ažurira

`Dohvat koda za vježbu 2 iz git repozitorija`

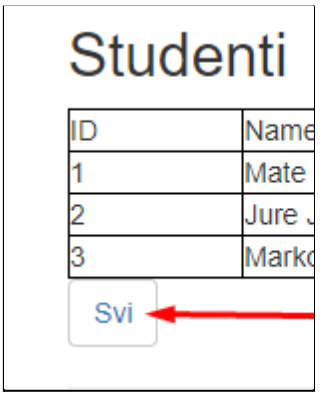



```
D:
cd TPP\Study
git fetch --all
git checkout app_study
git pull origin app_study
```

Ako git pull javi grešku:
git stash
git stash clear

ZADATAK 1

Pokrenuti aplikaciju i navesti na koji URL se upućuje poziv se upućuje kad korisnik klikne na:



ZADATAK 2

Korištenjem debuggera i postavljanjem breakpointa na odgovarajuće mjesto pogledati strukturu podataka i navesti ime klase za:

- podatak koji dohvati **`StudentController`** prije nego ga pošalje u View u sklopu metode **`List`**()
- podatak koji **`SqliteDataAccess`** dohvati iz baze u metodi **`GetEntities`**

ZADATAK 3

Na kartici studenta dodati polja: ID tečaja i Naziv tečaja. Za naziv tečaja iskoristiti metodu **`GetStudentWithCourse`**. Dodati vrijednosti ID tečaja i Naziv tečaja u **`ViewBag`** i prikazati ih u view. Na Moodle učitati izmijenjenu datoteku **`StudentController.cs`**

Osigurati da se aplikacija ne sruši ako CourseId nije unesen u bazi.

ZADATAK 4

Analogno funkcionalnosti "Svi" na listi studenata, izraditi funkcionalnost "Svi" na listi kolegija. Dodati potrebne controllere i viewove, te po potrebi modificirati postojeće. Na Moodleu navesti nazive novo kreiranih i izmijenjenih datoteka.

ZADATAK 5

U klasu **`StudentController`** dodati dvije metode sa sljedećim potpisima:

```
public string ReadJsonFile(string fileName);
```

- Čita json datoteku i vraća sadržaj u obliku stringa

```
public List<T> DeserializeJsonToType<T>(string json);
```

- Deserijalizira json sadržaj u listu zapisa klase modela (Student, Course)
- Koristiti klasu **`JsonConvert`**

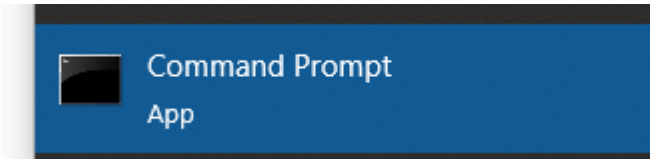
Pozvati metode unutar akcije **`List()`** kontrolera **`StudentController`**.

- Ime i putanja JSON datoteke se generira prema ključu:
  `D:\TPP\Study\Study\DataAccess\JsonData\IME_MODELA.json`
- Put do JSON datoteke generirati korištenjem .NET klase **`Path i`** interne klase **`Config`**.
  Primjer:
  ```
  var fileName = Path.Combine(Config.RootDirectory, "Study",
  "DataAccess", "JsonData", "Student.json");
  ```

Spremiti rezultat metode **`DeserializeJsonToType()`** u varijablu **`studentsFromJson`**.

Pokrenuti aplikaciju pomoću debuggera i postaviti breakpoint tako da je vidljiva struktura varijable **`studentsFromJson`**. Zamijeniti vrijednost svojstva **`ViewBag.Students`** s vrijednosti varijable **`studentsFromJson`**.

Spremanje rada u novi git branch




```
D:
cd TPP\Study
git checkout -b ime_prezime_2022_03_18
git add -A
git config --global user.email "vas.email@hhh"
git config --global user.name "ImePrezime"
git commit -m "initial commit"
```

Vježbe 3:

ZADATAK 6

Korištenjem koncepta Inversion of control i Dependency injection izraditi novu klasu **`JsonDataAccess`** koja implementira interface **`IDataAccess`** i čita podatke iz json datoteka **`Student.json`** i **`Course.json`** koje se nalaze u direktoriju **`DataAccess\JsonData`**.

Kopirati metode iz prethodnog zadatka (5) u klasu **`JsonDataAccess`**.

Ime datoteke za poziv metode **`ReadJsonFile`** unutar generičkih metoda interfacea **`IDataAccess`** generirati na način:

```
var fileName = Path.Combine(Config.RootDirectory, "Study", "DataAccess",
"JsonData", $"{typeof(T).Name}.json");
```

Na moodle učitati datoteku koja implementira vezu prema json datotekama.