

Poročilo pri predmetu Orodja za razvoj aplikacij

Končni projekt

Avtor: Jaka Udovič

Cerknica, 6. 6. 2021

KAZALO

1.	UVOD	3
2.	NAČRTOVANJE PROJEKTA.....	4
2.1.	ZAHTEV	4
3.	VSTAVLJANJE PODATKOV V PODATKOVNO BAZO	5
4.	REST STORITVE.....	6
4.1.	IMPLEMENTACIJA RESTFUL STORITEV	6
5.	NAMIZNA APLIKACIJA	8
6.	SPLETNA APLIKACIJA.....	10

1. Uvod

Naročnik projekta gospod Bruno je strasten športnik, vendar mu urejanje statistike iz preteklih tekmovanj predstavlja velik časovni strošek. Želi da, bi ta čas lahko uporabil za rekreacijo. Njegova želja je informacijski sistem, ki bi mu olajšal pregled njegove statistike. Bruno si želi, da bi imel vpogled v podatke, na spletni strani in v aplikaciji, kjer bi jih lahko tudi urejal in si pogledal statistiko, tako na spletni strani kot tudi v aplikaciji.

2. Načrtovanje projekta

2.1. Zahteve

Iz podanega problema in pogovora z naročnikom smo določili naslednje zahteve za izdelavo spletne aplikacije:

- uvoz preteklih podatkov v podatkovno bazo (podatke bo priskrbel naročnik),
- prijava uporabnika,
- prikazovanje rezultatov na spletu,
- prikaz statistike rezultatov za posameznega tekmovalca,
- možnost pridobivanja rezultatov preko api-ja,
- sortiranje podatkov,
- iskanje po podatkih,
- dodajanje novega uporabnika in
- izvoz podatkov.

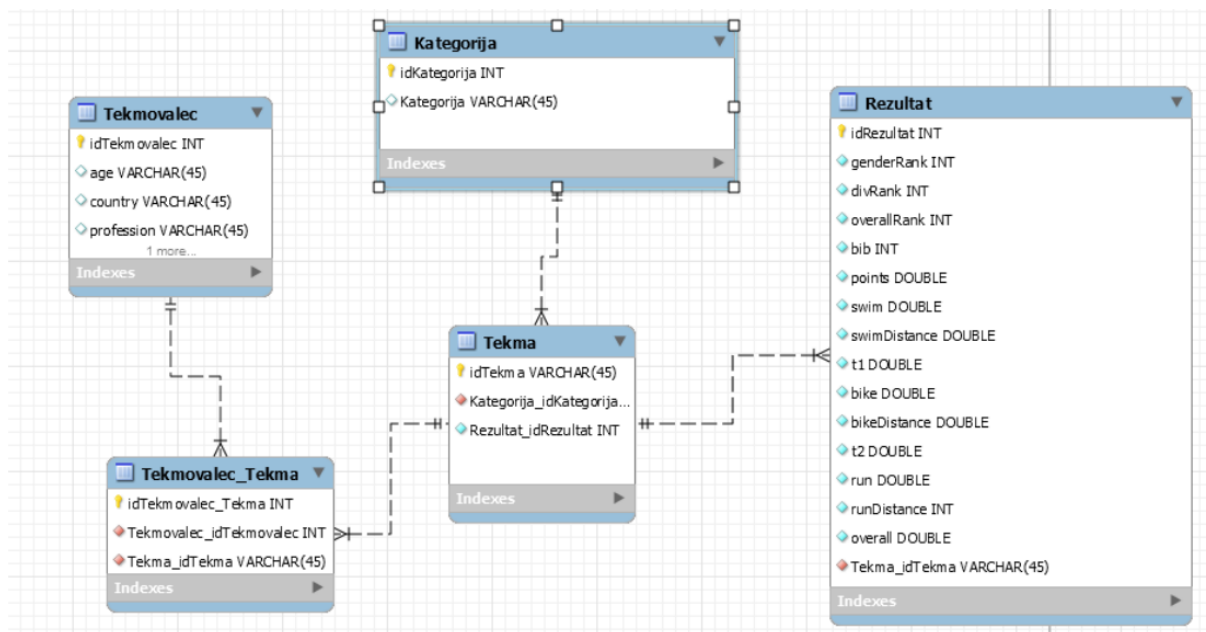
Zahteve pri izdelavi namizne aplikacije:

- uvoz preteklih podatkov v podatkovno bazo (podatke bo priskrbel naročnik),
- dvojezičnost,
- prikazovanje rezultatov na spletu,
- iskanje po podatkih,
- pridobivanja rezultatov in tekmovalcev preko api-ja,
- sortiranje podatkov,
- brisanje podatkov,
- izvoz podatkov,
- prijava v sistem in
- dodajanje novega uporabnika.

2.2. Entitetno-Relacijski Model

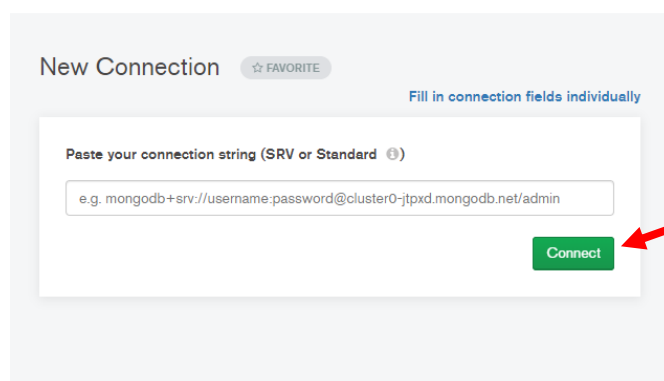
Za izdelavo ER modela smo uporabili orodje MySQL Workbench. Ustvarili smo 5 entitet: tekmovalec, tekma, kategorija, rezultat in pa tekmovalec_tekma, s katero smo normalizirali naš

model. Nato smo vsem entitetam določili pripadajoče attribute, katerim smo nato dodelili lastnosti kot so: primarni ključ, samodejno povečevanje, atribut ni nič.



3. Vstavljanje podatkov v podatkovno bazo

Za podatkovno bazo smo izbrali MongoDB, ki deluje po načinu NoSQL. Za začetek uporabe je potrebno prenesti podatkovni strežnik, ki ga lahko najdemo na uradni strani MongoDB na naslednji povezavi: <https://www.mongodb.com/try/download/community>. Ko namestimo strežnik s čarovnikom, ga zaženemo z odpiranjem datoteke mongod.exe, ki se nahaja v mongodb/bin direktoriju. Za lažji pregled nad podatki si nato namestimo še MongoDB compass z naslednje povezave: <https://www.mongodb.com/try/download/compass>. MongoDB compass prav tako namestimo s pomočjo čarovnika in ga nato zaženemo. Za povezavo na naš strežnik samo pritisnemo gumb »Connect«, nato pa Compass samodejno izpolni parametre za povezavo na Localhost.



Slika 1: Povezava na podatkovni strežnik, iz aplikacije Compass

Nato smo preko HTTP povezave pridobili podatke, ki nam jih je pripravil Bruno v obliki kompresiranih csv in Json datotek. Te datoteke so tudi priložene projektu. Za začetek obdelovanja teh podatkov smo zagnali Visual Studio 2019, v katerem smo ustvarili novo konzolno aplikacijo. Najprej smo izdelali razredne modele, ki smo jih predstavili v ER diagramu.

```
public class Tekmovalec
{
    [BsonId]
    [BsonRepresentation(BsonType.ObjectId)]
    public string Id { get; set; }
    public string name { get; set; }
    public string age { get; set; }
    public string country { get; set; }
    public string profession { get; set; }
}
```

Slika 2: Razred tekmovalec

Nato smo preko NuGeta prenesli vtičnik MongoDB.Driver, ki smo ga v programu uporabili z ukazi using in dodali povezavo do podatkovnega strežnika.

```
using MongoDB.Bson.Serialization.Attributes;
using MongoDB.Driver;
using MongoDB.Bson;
```

Slika 3: Uporaba vtičnika MongoDB.driver

Uporabnik, ki odpre naš paket, ki že vključuje vtičnike, jih ne rabi ponovno namestiti, saj to NuGet izvede samodejno. Nato smo izdelali funkcije, ki preberejo podatke iz csv datoteke jih pregledajo in vnesejo v podatkovno bazo. Za uporabo programa odpremo datoteko OZRA1.sln. Počakamo, da se prenesejo NuGet paketi ter stisnemo tipki ctrl in F5 ter počakamo, da se vnos podatkov konča. Vnesene podatke si nato lahko ogledamo v programu Compass.

4. REST storitve

Rest je arhitektura, kje je vsak vir predstavljen kot spletna storitev z enoznačnim naslovom URL, ki opredeljuje kako uporabljati obstoječe standarde. Za pridobivanje in spreminjanje virov uporabljamo standardne metode HTTP: GET, PUT, POST in DELETE. Arhitektura REST je zelo primerna za prenos podatkov, ki nimajo vizualnih komponent. (Rajnar, 2014)

4.1. Implementacija RESTful storitev

RESTful so spletne storitve, ki ustrezajo principom REST arhitekturnega stila. V okviru našega projekta smo v Visual Studio 2019 ustvarili nov asp.net web API projekt. Najprej smo ustvarili modele za naše razrede, ki smo jih lahko prevzeli iz prejšnjega projekta, v katere smo vnašali podatke v podatkovno bazo. Nato smo preko NuGet upravljalnika dodali vtičnike za uporabo MongoDB podatkovnega strežnika, Swashbuckla in WEB API-ja. Z dodajanje BrunoService in RezultatService smo definirali povezavo na podatkovni strežnik in funkcije, ki jih lahko izvajamo nad podatkovno bazo. Dodali smo funkcije za dodajanje, pridobivanje, spreminjanje in odstranjevanje podatkov. Nato smo dodali še API krmilnik, ki je naš vmesnik med

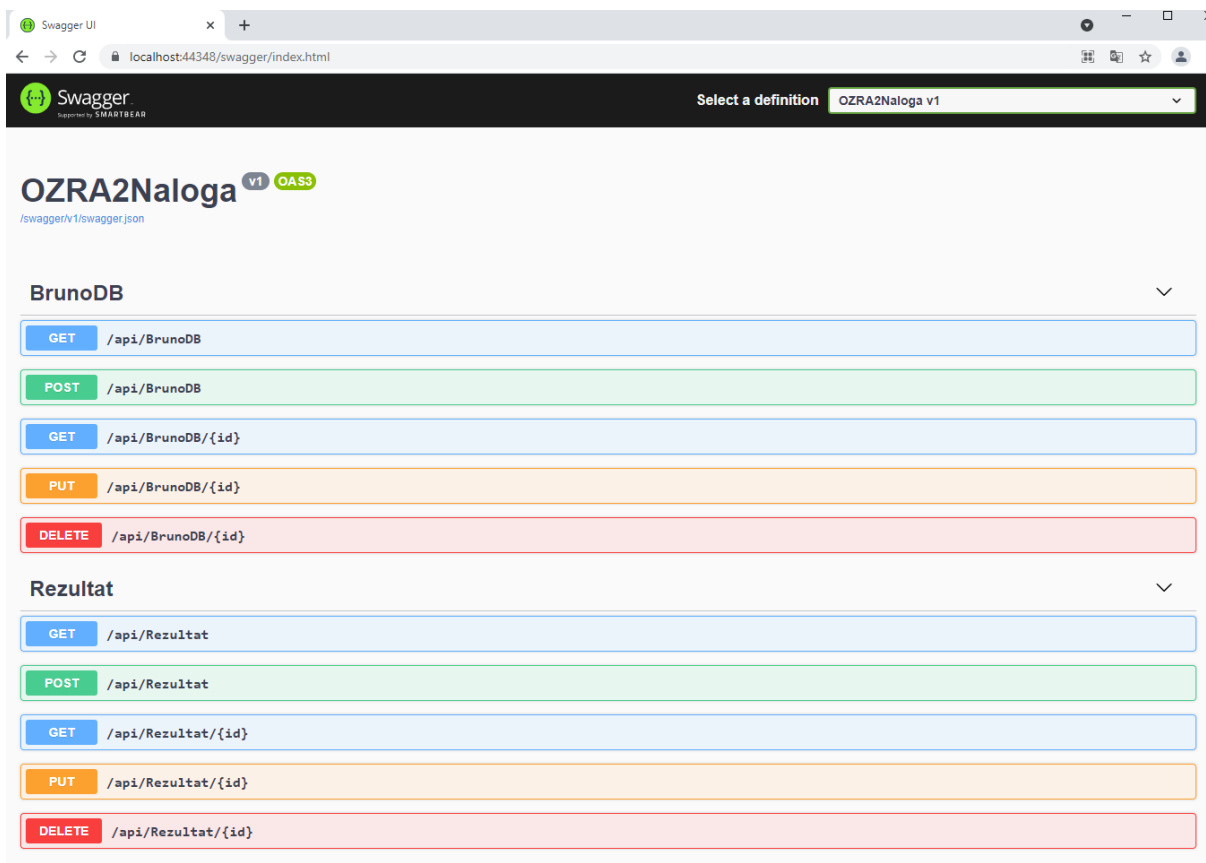
uporabnikom in podatkovnim strežnikom. Za izgled in preprostejšno uporabo nam pomaga storitev Swagger.

4.2. Uporaba REST storitev

Za uporabo aplikacije zaženemo program OZRA2Naloga.sln in počakamo da se prenesejo vsi NuGet paketi. Nato vnesemo ime naše podatkovne baze v RezultatService.

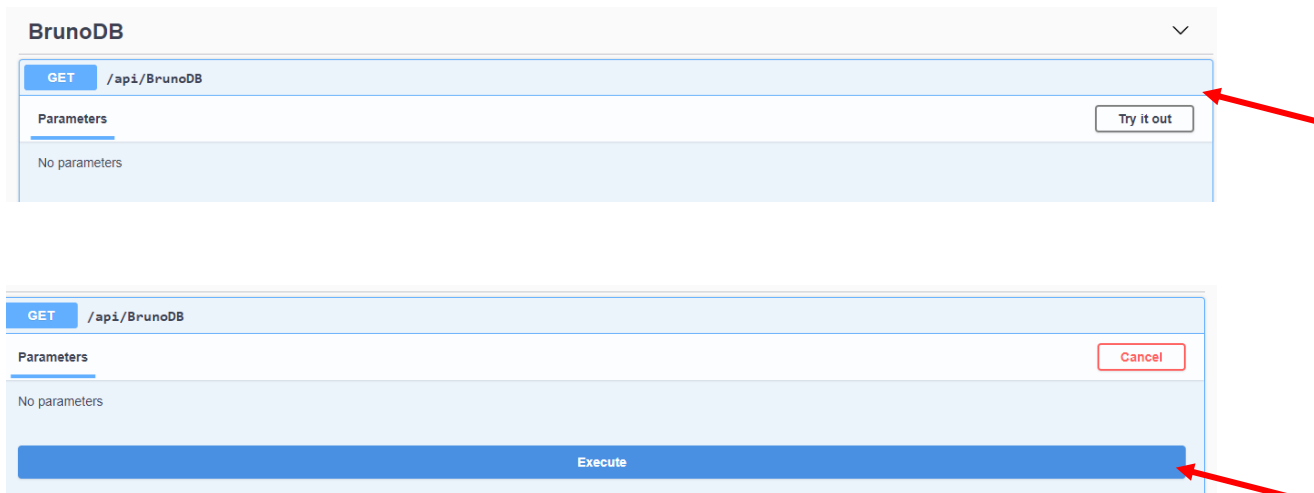
```
0 references
public RezultatService(IRezultatDatabaseSettings settings)
{
    var client = new MongoClient("mongodb://localhost:27017");
    var database = client.GetDatabase("BrunoDB");
    try
    {
        _rezultat = database.GetCollection<Rezultat>("Rezultat");
    }
    catch
    { }
}
```

Ko imamo vse potrebne vtičnike stisnemo gumba ctrl in F5. Odpre se nam spletna aplikacija na kateri lahko vidmo definirane Rest storitve.

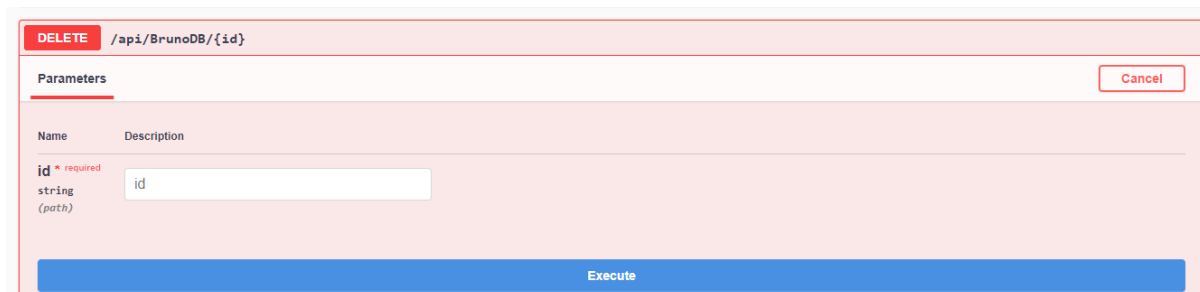


Slika 4: Spletna aplikacija za naše RESTful storitve

Get storitve lahko preizkusimo s pritiskoma na gumb »Try it out« in nato »Execute«, kot lahko vidimo na spodnji sliki. Za izvedbo GET storitev je potrebno počakati nekaj časa, saj se procesira precejšnja količina podatkov.



Podatki se nam prikažejo v JSON obliki. Za brisanje podatkov, si izberemo ID primerka, ki ga želimo izbrisati in ga vpišemo v polje ID in zaženemo brisanje s pritiskom na gumb »Execute«. Če je brisanje uspešno nam aplikacija izpiše kodo 200.



Slika 5: Obrazec za brisanje podatkov

Iskanje po Idju izvedemo podobno kot brisanje, le da izberemo storitev GET, vpišemo ID in izvedemo klic. Nov primerek tekmovalca lahko dodamo s storitvijo POST, v katero vpišemo željene podatke v spodnji obliki.

```
{
  "id": "string",
  "name": "string",
  "age": "string",
  "country": "string",
  "profession": "string"
}
```



Če je dodajanje uspešno, nam aplikacija ponovno vrne kodo 200. Urejanje izvedemo s klicem storitve PUT, In to tako, da vpišemo Id primerka, ki ga želimo spremeniti in željene nove podatke v formatu JSON.

5. Namizna aplikacija

Pri izdelavi spletne aplikacije smo se osredotočali na funkcionalnosti, ki si jih je zaželel Bruno. Namizno aplikacijo zaženemo z zagonom zagonske datoteke Ozra3.exe. Ppred tem pa moramo poskrbeti, da aplikacija z RESTful storitvami še vedno teče.

Competitor

Results

Id	Name	Age	Country
60bf52a362425dbb9	Neil Anderson	116	GBR
60608d4bb3129030f	Andrew Caune	32	AUS
60608d4ab3129030f	Urs Mueller	32	CHE
60bf52a362425dbb9	Emma Oneill	32	AUS
60bf52a362425dbb9	Andrew Caune	32	AUS
60608d4ab3129030f	Emma Oneill	32	AUS
60bf52a362425dbb9	Nathan Campbell	32	AUS
60608d4bb3129030f	Nathan Campbell	32	AUS

Slika 6: Polje z tekmovalci

Takoj, ko se naloži namizna aplikacija se nam prikažejo podatki tekmovalcev, ki jih lahko sortiramo s klikom na željeni atribut v glavi polja. Za dodajanje novega tekmovalca vpišemo željene podatke, kjer mora Id biti BSON Id, ki je 24-mesto heksadecimalno število. Paziti moramo tudi, da se Id ne ponovi. Dodajanje izvedemo z pritiskom na gumb »Dodaj« ali »Add«, če uporabljamo angleški vmesnik.

Id:

Name:

Age:

Profession:

Delete

Add New

Search

Export

Slika 7: Forma za dodajanje uporabnika

Za brisanje tekmovalca, le tega izberemo v polju s tekmovalci in pritisnemo gumb »Delete« oziroma »Izbriši«. Iskanje lahko izvedemo z vpisom željenega Id-ja v polje Id in pritiskom na »Search« oziroma »Iskanje«. Če želimo izvoziti tekmovalce izberemo gumb »Export« oziroma »Izvozi« in izberemo mesto na našem računalniku, kamor želimo shraniti datoteko. Za upravljanje z rezultati uporabljamo iste postopke kot pri tekmovalcih, le da se nam pri rezultatih za dodajanje novega rezultata odpre novo okno, kamor vpišemo podatke.

DodajanjeRezultata

Id:	Gender rank:	Bib:	Division:
Points:	Swim:	Swim distance:	T1:
Bike:	Bike distance:	T2:	Run:
Overall rank:	Run distance:	Div rank:	Overall:
Tekma:			

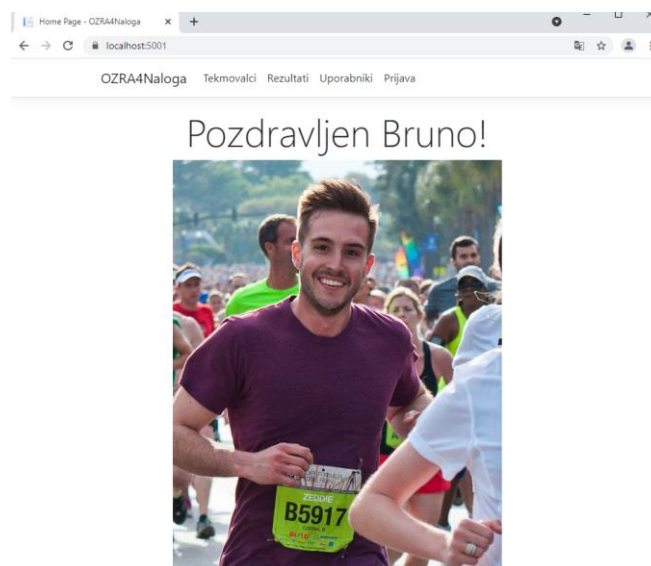
Add New

Slika 8: Okno za dodajanje rezultatov

Za spreminjanje jezika naše aplikacije v zgornjem desnem kotu izberemo zastavo države z željenim jezikom.

6. Spletna Aplikacija

Pred uporabo spletne aplikacije zopet poskrbimo da naš spletni API še vedno teče nemoteno. Nato odpremo zagonsko datoteko OZRA4Naloga.exe in v brskalnik vpišemo naslov, ki se izpiše v terminalu. Če spletna stran še vedno ne deluje, zaženemo OZRA4Naloga.sln in počakamo, da se naložijo NuGet paketi ter nato zopet zaženemo aplikacijo.



Slika 9: Prva stran spletne aplikacije

V navigacijski vrstici izberemo zavihek »Uporabniki« in s pritiskom na tekst »Create new« pričnemo z ustvarjanjem novega upravnika, s katerim se bomo vpisali v aplikacijo. Pri vpisovanju podatkov nas aplikacija opozarja ob nepravilnih vnosih. Zato pazimo, da le te popravimo. Ko so vsi podpisani podatki zadovoljivi uporabnika dodamo s pritiskom na gumb »Create«. Nato v Navigacijski vrstici izberemo tekst Prijava in se prijavimo z uporabniškim računom, ki smo ga ravnokar ustvarili.

OZRA4Naloga Tekmovalci Rezultati Uporabniki Prijava

Create

Uporabnik

Id

Title

ime

priimek

rojDatum

emso

naslov

postna

posta

drzava

mail

geslo

ConfirmPassword

[Create](#)

[Back to List](#)

Slika 10: Forma za vnos novega uporabnika

Nato si lahko ogledamo tekmovalce in njihove rezultate z izbiro v navigacijski vrstici. V spletni aplikaciji se nam izpišejo vsi podatki, ki jih lahko sortiramo z izbiro atributov, ki so modre barve na vrhu polja s podatki. Lahko pa tudi iščemo po imenu z vpisom imena v tekstovno polje »Find by name« in pritiskom na gumb »Search«, ki nam izpiše najdene tekmovalce. S pritiskom

na gumb »delete« ob določenem tekmovalcu ali rezultatu lahko le tega tudi izbrišemo. Vse rezultate in tekmovalce lahko tudi ivozimo v XML datoteko s pritiskom na gumb »Save File Now« in izbiro prostora za shranjevanje. Prav tako kot uporabnika lahko dodamo tudi nov rezultat in tekmovalca z izbiro teksta »Create new« in izpolnjevanjem obrazca, ki vsebuje validacijo podatkov, tako da je potrebno vpisati primerne podatke.