

# .NET Web Development 18 - Finalni test

Korišćenjem radnih okvira ASP.NET Core Web API (.NET 5.0), Entity Framework Core i Bootstrap i Fetch API-a realizovati Web aplikaciju za ribarnice i ribu koju prodaju. Aplikacija treba da obezbedi rad sa sledećim entitetima:

## Ribarnica:

- Id - Identifikator
- Naziv - obavezna tekstualna vrednost sa manje od 120 karaktera
- Godina otvaranja - obavezna celobrojna vrednost manja od 2023, a veća ili jednaka od 1950

## Riba:

- Id – Identifikator
- Sorta - obavezna tekstualna vrednost sa najmanje 2 karaktera, a najviše 30 karaktera
- Mesto ulova - obavezna tekstualna vrednost sa najmanje 3 karaktera, a najviše 120 karaktera
- Cena (dinari po kilogramu) – obavezna decimalna vrednost iz intervala [100, 10000]
- Dostupna količina (u kilogramima) – obavezna celobrojna vrednost iz intervala [1, 1000]
- Ribarnica - veza sa instancom klase Ribarnica (jedan proizvod (riba) može biti prodavana od strane samo jedne ribarnice, dok jedna ribarnica prodaje više proizvoda (riba))

Polazniku se ostavlja potpuna sloboda prilikom imenovanja entiteta i njihovih polja.

1) Pomoću radnih okvira ASP.NET Core Web API i Entity Framework Core implementirati sledeći REST API, vodeći računa da se kod rezultata odgovarajućih akcija za povezani objekat ne prikazuje nedostajuća vrednost (null):

- a) **GET api/ribarnice** - preuzimanje svih Ribarnica sortiranih po Nazivu rastuće;
- b) **GET api/ribarnice/{id}** - preuzimanje Ribarnice po zadatom id-u;
- c) **GET api/info?granica={vrednost}** - preuzimanje svih Ribarnica sa starošću Ribarnice (razlika sadašnje godine i godine otvaranja) i prosečnom Cenom svih Riba koje prodaju, pri čemu je prosečna Cena svih Riba manja od vrednosti granica, a sortiranih prema Nazivu Ribarnice opadajuće;
- d) **GET api/stanje** – preuzimanje svih Ribarnica gde za svaku Ribarnicu postoji podatak koji joj je Naziv, broj različitih Riba (proizvoda) koje prodaje i ukupna Dostupna količina za sve Riba koje prodaje, sortirano po broju različitih Riba opadajuće;
- e) **GET api/ribarnice/nadji?naziv={vrednost}** - preuzimanje svih Ribarnica čiji Naziv sadrži prosleđenu vrednost **naziv**, sortirano prema Godini osnivanja rastuće, a u slučaju da su dve Ribarnice osnovane iste godine, njih u tom slučaju sortirati po Nazivu opadajuće;
- f) **GET api/ribe** - preuzimanje svih Riba, sortirano po Sorti rastuće;
- g) **GET api/ribe/{id}** - preuzimanje Riba po zadatom id-u;
- h) **GET api/ribe/trazi?sorta={vrednost}** - preuzimanje svih Riba čija Sorta je jednaka prosleđenoj vrednosti **sorta**, sortirano prema Ceni rastuće;
- i) **POST api/ribe** - dodavanje nove Riba;
- j) **PUT api/ribe/{id}** - izmena postojeće Riba;
- k) **DELETE api/ribe/{id}** - brisanje postojeće Riba;
- l) **POST api/pretraga** - preuzimanje svih Riba sa Dostupnom količinom između dve unete vrednosti **[najmanje, najviše]**, sortirano tako da se prvo prikazuju skuplje (sortirano prema Ceni opadajuće);
- m) Registracija i prijava korisnika.

2) Jedinično testirati sledeće funkcionalnosti:

- a) Funkcionalnost **1) g)** sa jediničnim testom kada akcija vraća status kod 200 i objekat;
- b) Funkcionalnost **1) j)** sa jediničnim testom kada akcija vraća status kod 400;
- c) Funkcionalnost **1) k)** sa jediničnim testom kada akcija vraća status kod 404;
- d) Funkcionalnost **1) l)** sa jediničnim testom kada akcija vraća status kod 200 i više objekata.

3) Pomoću radnog okvira Bootstrap i Fetch API-a realizovati sledeću Single Page Application:

- a) Prilikom pokretanja, stranica sadrži poruku da korisnik nije prijavljen, dugmad za prijavu i registraciju, kao i tabelarni prikaz Riba po funkcionalnosti **1) f)**, kao što je prikazano na slici 1. Za Ribarnicu se prikazuje njen Naziv.

**Korisnik nije prijavljen na sistem**

Prijava

Registracija

**Riba na prodaju**

Sorta	Cena (din/kg)	Ribarnica	Kolicina (kg)
Saran	860	Dunav doo	30
Saran	780	Sveza riba	12
Som	1300	Tisa str	10
Smudj	1100	Sveza riba	20
Smudj	950	Dunav doo	15

Slika 1

- b) Ukoliko korisnik pritisne dugme **Registracija**, prikazaće mu se forma za registraciju korisnika, kao što je prikazano na slici 2. Ukoliko se korisnik uspešno registrovao, forma za registraciju će se očistiti, a korisniku će se prikazati forma za prijavu, čiji su izgled i ponašanje opisani u funkcionalnosti **3) c)**. Ukoliko je došlo do greške prilikom registracije, korisnik će biti obavešten pomoću **alert-a "Greska prilikom registracije!"**. Klikom na dugme **Odustajanje** korisniku se prikazuje sadržaj iz funkcionalnosti **3) a)**.

**Registracija korisnika**

Korisnicko ime

Email

Lozinka

Ponovi lozinku

Registruj se

Odustajanje

**Riba na prodaju**

Sorta	Cena (din/kg)	Ribarnica	Kolicina (kg)
Saran	860	Dunav doo	30
Saran	780	Sveza riba	12
Som	1300	Tisa str	10
Smudj	1100	Sveza riba	20
Smudj	950	Dunav doo	15

Slika 2

- c) Ukoliko korisnik pritisne dugme **Prijava** sa početne strane aplikacije ili kada se korisnik uspešno registruje na sistem, prikazaće mu se forma za prijavu korisnika kao što je prikazano na slici 3. Ukoliko je došlo do greške prilikom prijave, potrebno je obavestiti korisnika pomoću **alert-a “Greska prilikom prijave!”**. Ukoliko se korisnik uspešno prijavi na sistem, forma za prijavu se čisti i prikazuje mu se sadržaj sa slike 4. Klikom na dugme **Odustajanje** korisniku se prikazuje sadržaj iz funkcionalnosti **3) a)**.

**Prijava korisnika**

Korisničko ime  
Lozinka

Prijavi se

Odustajanje

  

**Riba na prodaju**

Sorta	Cena (din/kg)	Ribarnica	Kolicina (kg)
Saran	860	Dunav doo	30
Saran	780	Sveza riba	12
Som	1300	Tisa str	10
Smudj	1100	Sveza riba	20
Smudj	950	Dunav doo	15

Slika 3

- d) Nakon labele Prijavljeni korisnik se prikazuje korisničko ime prijavljenog korisnika. Klikom na dugme **Odjava** korisnik se odjavljuje sa sistema i nakon toga mu se prikazuje sadržaj opisan u **3) a)**.
- e) Tabela Riba se proširuje sa još dve kolone sa desne strane, koje za svaku Ribu sadrže Mesto ulova i dugme **Obriši**. Klikom na dugme **Obriši** se briše odabrana Riba i osvežava prikaz u tabeli. Ovu akciju može da vrši **samo prijavljeni korisnik**.
- f) Iznad tabele se nalazi forma za pretragu Riba koja prati funkcionalnost **1) I)**. Ovu akciju može da vrši **samo prijavljeni korisnik**. Ribe koje ispunjavaju kriterijum pretrage se prikazuju u tabeli Riba i moguće je njihovo brisanje kao što je opisano u funkcionalnosti **3) e)**. Ukoliko je došlo do greške prilikom unosa kriterijuma za pretragu ili samog dobavljanja Ribe sa REST API-a, korisnika obavestiti putem **alert-a “Greska prilikom pretrage!”**.
- g) Ispod tabele Riba se prikazuje forma za dodavanje nove Ribe, koja prati funkcionalnost **1) i)**, kao što je prikazano na slici 4. Ovu akciju može da vrši **samo prijavljeni korisnik**. Ribarnica se bira iz padajućeg menija koji se popunjava pomoću funkcionalnosti **1) a)**. U padajućem meniju se prikazuje Naziv Ribarnice. Klikom na dugme **Dodavanje** vrši se dodavanje nove Ribe. Ukoliko je dodavanje bilo uspešno, forma za dodavanje će se očistiti, a u tabeli Riba će se osvežiti prikaz. Ukoliko je došlo do greške prilikom unosa podataka ili prilikom komunikacije sa REST API-em, korisnik će se obavestiti putem **alert-a “Greska prilikom dodavanja!”**. Klikom na dugme **Odustajanje** čiste se polja u formi za dodavanje.

Prijavljeni korisnik: **korisnicko\_ime**

**Odjava**

Pretraga ribe po ukupnoj količini

Najmanje (kg)

Najviše (kg)

**Traži**

**Riba na prodaju**

Sorta	Cena (din/kg)	Ribarnica	Kolicina (kg)	Mesto ulova	Akcija
Saran	860	Dunav doo	30	Dunav	<b>Obrisi</b>
Saran	780	Sveza riba	12	Ribnjak Ecka	<b>Obrisi</b>
Som	1300	Tisa str	10	Tisa	<b>Obrisi</b>
Smudj	1100	Sveza riba	20	Ribnjak Bager	<b>Obrisi</b>
Smudj	950	Dunav doo	15	Dunav	<b>Obrisi</b>

**Dodavanje nove ribe**

Sorta

Mesto ulova

Cena (din/kg)

Ukupna kolicina (kg)

Ribarnica

**Dodavanje**

**Odustajanje**

Slika 4

## Napomene:

- Test nosi 100 bodova, za polaganje testa je neophodno osvojiti 50 bodova uz poštovanje uslova za pregled testa.
- Uslov za pregled testa je implementiranje funkcionalnosti 1) a), 1) f) i 3) a).
- Zabranjena je upotreba eksternih alata za generisanje koda i Scaffolding-a. Sve funkcionalnosti koje su scaffold-ovane ili na neki način koriste scaffold-ovanu funkcionalnost bodovaće se sa 0 bodova.
- Ako ima potrebe, a u skladu sa dobrom programerskom praksom, koristiti AutoMapper, Dependency Injection koncept i Repository šablon.
- Svaka komunikacija sa kolegama je najstrožije zabranjena!
- Projekat se predaje u zip arhivi ImePrezime.zip (obrisati foldere bin i obj pre kreiranja arhive).
- Koristeći Code First migracije popuniti bazu podataka sledećim podacima:
  - Ribarnice

Id	Naziv	Godina osnivanja
1	Dunav doo	2015
2	Tisa str	2012
3	Sveza riba	2015

○ Ribe

Id	Sorta	Mesto ulova	Cena (din/kg)	Ukupna količina (kg)	Ribarnica
1	Smudj	Ribnjak Bager	1100	20	3
2	Saran	Dunav	860	30	1
3	Som	Tisa	1300	10	2
4	Saran	Ribnjak Ecka	780	12	3
5	Smudj	Dunav	950	15	1

**Bodovanje:**

Funkcionalnost	Bodovi
Model Ribarnica	2
Model Riba	4
Code first migracije i unos podataka	2
1) a)	2
1) b)	2
1) c)	7
1) d)	6
1) e)	4
1) f)	2
1) g)	3
1) h)	4
1) i)	3
1) j)	3
1) k)	3
1) l)	5
1) m)	3
2) a)	3
2) b)	2
2) c)	2
2) d)	3
3) a)	5
3) b)	4
3) c)	5
3) d)	3
3) e)	4
3) f)	7
3) g)	7
<b>Ukupno</b>	<b>100</b>