

# Projektni zadatak

Projektni zadatak je izraditi .NET Web API aplikaciju. Specifični zadaci za svakog studenta definirani su u zasebnom dokumentu.

## Potrebni alati:

- Visual Studio 2022
- .NET 8
- ASP.NET Web API predložak projekta
- Entity Framework Core
- MS SQL Server

---

## Minimalni uvjeti

Projekti koji ne ispunjavaju ove zahtjeve neće biti uzeti u obzir.

- Projekt se uspješno pokreće (nema pogrešaka prilikom pokretanja projekta)
- Migracije baze podataka mogu se pokrenuti
- Swagger se uspješno pokreće
- Najmanje 5 tablica u bazi podataka
- CRUD operacije za barem jedan odnos "jedan-prema-više"
- CRUD operacije za barem jedan odnos "više-prema-više"
- UML dijagram baze podataka aplikacije (priložiti digram uz projekt)

---

## Bodovanje

Bodovi su podijeljeni po funkcionalnostima. Da biste ostvarili bodove za određenu funkcionalnost, potrebno je implementirati tu značajku **i objasniti kako radi**. Da bi se ostvarili svi bodovi, implementacija mora biti prisutna za sve dostupne endpointove. CRUD zahtijeva 5 endpointova (*ReadAll*, *ReadOne*, *Create*, *Update*, *Delete*). Bodovi se mogu ostvariti za sve funkcionalnosti (osnovne i dodatne)

**Funkcionalnosti:**

<b>Funkcionalnost</b>	<b>Opis</b>	<b>Max Bodovi</b>
<b>UML dijagram baze podataka</b>	Dijagram koji prikazuje sve tablice baze podataka i njihove odnose.	5
<b>Modeli i Migracije</b>	Svi modeli su definirani, a baza podataka je kreirana kroz migracije. Migracije se uspješno pokreću.	15
<b>CRUD za odnos "Jedan-prema-više"</b>	Endpointovi omogućuju upravljanje odnosom između roditeljskog i podređenog objekta. Relacije su vraćene u detaljnom prikazu roditelja.	15
<b>CRUD za odnos "Više-prema-više"</b>	Omogućiti CRUD operacije za "više-prema-više" odnos (barem s jedne strane). Druga strana vraća povezane objekte u detaljnom prikazu (ReadOne).	15
<b>Autentifikacija i autorizacija</b>	Implementirati shemu autentifikacije/autorizacije (basic auth, session ili token). Lozinke moraju biti hashirane u bazi.	20 10 – basic 15 – session ili JWT 20 JWT sa refresh tokenom
<b>DTO-ovi</b>	Svi endpointovi koriste DTO-ove umjesto Model klase.	10
<b>Validacija podataka</b>	Podaci se validiraju koristeći data annotations.	5
<b>Asinkrone operacije</b>	Sve operacije implementirane su kao asinkrone.	5
<b>.http datoteke</b>	Kreirane su .http datoteke za testiranje svih endpointova.	5
<b>Paginacija</b>	Dodati paginaciju na jedan endpoint (koristiti query parametre <i>page</i> i <i>pageSize</i> ).	5
<b>Ukupno</b>		<b>100</b>

**DODATNI BODOVI:**

<b>Funkcionalnost</b>	<b>Opis</b>	<b>Max Bodovi</b>
<b>Controller-Service-Repository</b>	Razdijeliti implementaciju u Controller – Service – Repository pattern koristeći DI.	10
<b>Generički repository</b>	Kreirati generički repository umjesto zasebnog za svaki entitet.	5
<b>Logiranje</b>	Implementirati mehanizam za logiranje.	5
<b>UNIT testovi</b>	Implementirati barem 3 UNIT testa koji testiraju specifične dijelove koda.	5
<b>Uloge korisnika</b>	Kreirati različite uloge (npr. Admin i User) i ograničiti pristup određenim rutama.	5
<b>Identity Core</b>	Implementirati autentifikaciju koristeći Identity Core.	5
<b>Data seederi</b>	Implementirati početni unos podataka u bazu.	5
<b>Ukupno</b>		<b>40</b>

**Ocjenjivanje:**

<b>Bodovi</b>	<b>Ocjena</b>
< 60	1
60-69	2
70-79	3
80-89	4
90+	5

## **Dostava projekata**

- Projekti se dostavljaju putem Merilna. Cijeli projekt stavite u zip folder. Ne zaboravite staviti dijagram baze podataka.
- U roku od 7 dana povratno javim informaciju zadovoljava li projekt minimalne uvijete za obranu. Ako je ok, možete se prijaviti za obranu projekta na slijedećem ispitnom roku. Ako nije ok, poslati ću vam što treba ispraviti.
- Rokovi će biti naknadno definirani
- Obzirom da je obrana u usmenom formatu, radi transparentnosti ispiti će se održavati u grupama od 3 studenta.

## **Prezentacija projekta**

- Projekt treba biti prezentiran, a bodovi će se dodijeliti tijekom prezentacije.
- Za bodovanje, student mora pokazati rad funkcionalnosti, prikazati kod i objasniti kako radi. Implementacija koju student ne zna objasniti donosi 0 bodova.
- Prezentacija započinje prikazom UML dijagrama baze podataka i objašnjenjem odnosa između modela te funkcionalnosti implementiranih u projektu.
- Nakon toga će student prezentirati korištenje endpointa putem Swaggera, Postmana ili .http datoteka.
- Pregledat će se kod i student će trebati objasniti određene dijelove implementacije.
- Na kraju će se pregledati lista funkcionalnosti i odrediti bodovi za svaku od njih.
- Studenti mogu prezentirati projekt na svom računalu ili mogu pokrenuti projekt na računalu u učionici koje je spojeno na projektor

**Savjeti za tijek rada:**

1. Kreirajte prazan ASP.NET Web API projekt.
2. Obrišite model i kontroler *WeatherForecast*.
3. Instalirajte NuGet paket *EntityFrameworkCore.SqlServer*.
4. Kreirajte *DataContext* i postavite konekciju na SQL Server.
5. Kreirajte modele aplikacije unutar direktorija *Models*.
6. Postavite odnose između modela.
7. Kreirajte migraciju za kreiranje baze podataka.
8. Pokrenite migraciju.
9. Provjerite bazu podataka i kreirajte dijagram baze u SQL Server Management Studio. Sliku dijagrama spremite unutar projekta.
10. Kreirajte kontrolere koristeći opciju "Controller using Entity Framework" za generiranje CRUD endpointova.
11. Kreirajte DTO-ove i mappere za specifične endpointove.
12. Dodajte data annotations za validaciju podataka u DTO-ove.
13. Dodajte paginaciju na jedan GET endpoint koristeći query parametre.
14. Koristite Endpoint Explorer za kreiranje .http datoteka za testiranje endpointova.
15. Implementirajte autentifikaciju i provjeru korisničkih podataka u bazi.
16. DODATNI BODOVI: Pročitajte na internetu o pojedinoj temi. Saznajte čemu služi i kako se koristi. Implementirajte funkcionalnost u svom projektu.