

SVEUČILIŠTE U SPLITU
PRIRODOSLOVNO-MATEMATIČKI FAKULTET

SEMINARSKI RAD
PROJEKT IZ UPRAVLJANJA PODACIMA
MONGODB BAZA PODATAKA – VOJSKA

Profesor: Tonći Dadić

Studentica: Marina Mravak

Podatkovna znanost i inženjerstvo

Split, srpanj 2022.

Sadržaj

1	UVOD	3
1.1	Opis problema.....	3
2	IZRADA MODELA.....	4
2.1	Kolekcije.....	4
2.1.1	Djelatnik	4
2.1.2	Vrsta zaposlenja	4
2.1.3	Vojna jedinica	5
2.1.4	Vojarna	5
2.1.5	Vojna odora	5
2.1.6	Vrsta vojne odore	5
2.1.7	Oružje	5
2.1.8	Povijest plaće.....	5
3	IZRADA BAZE PODATAKA	6
3.1	Spajanje na MongoDB Atlas Cluster i PyMongo Driver	7
3.2	Kreiranje kolekcija u bazi podataka Vojska	11
4	MONGODB CRUD OPERACIJE	15
4.1	Insert (Create)	15
4.2	Select (Read).....	16
4.3	Update.....	18
4.4	Delete.....	20
5	ZADACI.....	21
6	POPIS SLIKA	28

1 UVOD

Mnogi se mladići i djevojke odlučuju na vojni poziv i službu u vojsci. Oružane snage posebno su ustrojena organizacija osposobljena i opremljena za vođenje svih oblika oružane borbe. Temeljna zadaća oružanih snaga Republike Hrvatske jest zaštita njezina suvereniteta i teritorijalne cjelovitosti. Oružane snage mogu obavljati i određene zadaće u izvanrednim okolnostima, npr. u slučajevima elementarnih nepogoda i drugih nesreća.

Oružane snage zadužene su za provođenje odluka nadležnih civilnih državnih organa o pitanju rata ili miru. Ako se odluče na rat, odredit će njegov konačni cilj i vojsci izdati nalog da ga provede. Vojska će svojom strukom i snagom odgovoriti na pitanje kako voditi rat da bi se ostvario konačni politički cilj.

Na službi u oružanim snagama, osim vojnih osoba, jesu i vojni službenici i vojni namještenici. Vojni su službenici osobe s visokom stručnom spremom koje su imenovane na dužnost u oružanim snagama i u svom zanimanju obavljaju poslove za koje su ovlaštene oružane snage. Vojni su namještenici osobe s višom ili srednjom stručnom spremom. Za uspješnost i učinkovitost u obavljanju ovih poslova, posebno je važan timski rad i učinkovito vođenje.

1.1 Opis problema

Za ovaj projekt potrebno je osmisliti bazu podataka na principu kako funkcionira vojska i koje su zadaće njenih djelatnika. Vodi se evidencija o djelatnicima koji su u službi vojske, obavljanju poslova i zadaća u vojarni prema dobivenim zapovijedima. Kakvi će to biti poslovi, ovisi o činu ili vrsti zaposlenja kojeg djelatnik ima i vojnoj jedinici u kojoj radi.

Dostupni podaci o djelatnicima su: ime, prezime, osobni identifikacijski broj. Bilježi se datum početka i kraja djelatnika koji služi u određenoj vojnoj jedinici. Obveza svakog djelatnika je da na radnom mjestu nosi vojnu odoru. Naoružani su samo onim oružjem koje mogu nositi sa sobom i borbenim vještinama stečenim dugotrajnom vojnom obukom i položenim raznim tečajevima. Vodi se evidencija o djelatnikovoj mjesečnoj plaći za vrijeme služenja određenog čina.

Zadaća baze podatka je omogućiti lakši pristup podacima svakog djelatnika vojske, njihovoj vrsti posla i službi u određenim vojnim jedinicama.

2 IZRADA MODELA

2.1 Kolekcije

Kolekcija u MongoDB je skupina dokumenata, te je slična tablici u SQL relacijskoj bazi podataka. Dakle, kolekcija u MongoDB je osnovni blok koji sadrži istu grupu dokumenata. Kolekcije najčešće imaju više dokumenata, a svaki dokument ima samo jednu vrijednost odnosno podatak.

MongoDB pohranjuje zapise podataka kao BSON dokumente. BSON je binarni prikaz JSON dokumenata, iako sadrži više tipova podataka od JSON-a. Struktura MongoDB dokumenata se sastoje od polja - vrijednosti parova.

Za model baze podataka koji opisuje princip rada vojske koriste se sljedeće kolekcije (Slika 1):

1. Djelatnik
2. Vrsta zaposlenja
3. Vrsta zaposlenja_Djelatnik
4. Vojna jedinica
5. Vojna jedinica_Djelatnik
6. Vojarna
7. Vojna odora
8. Vrsta vojne odore
9. Oružje
10. Vrsta oružja
11. Oružje_Djelatnik
12. Povijest plaće

2.1.1 Djelatnik

Središnja kolekcija baze je *Djelatnik* koja prikazuje podatke o svakom zaposleniku koji je u službi vojske. Opisana je poljima: *Imenom*, *Prezimenom*, jedinstvenim identifikacijskim brojem ili *OIB-om*, *Težinom* i *Visinom*.

2.1.2 Vrsta zaposlenja

Svaki djelatnik u oružanim snagama radi određenu vrstu posla npr. radi posao običnog vojnika, časnika, generala, itd. Kolekcija je opisana poljem *Naziv*. Tablica *Vrsta zaposlenja_Djelatnik* prikazuje podatke o datumu početka i kraja kada je djelatnik radio određenu vrstu zaposlenja.

2.1.3 Vojna jedinica

Kolekcija *Vojna jedinica* bilježi podatke o djelatnicima koji su u službi određene vojne jedinice. Opisuje se poljima *Naziv* i *Kapacitet* (broj djelatnika koji mogu služiti u određenoj jedinici). Nova tablica *Djelatnik_Vojna jedinica* prikazuje podatke o datumu početka i kraja kada je djelatnik radio za određenu vojnu jedinicu.

2.1.4 Vojarna

Poljima *Naziv* i *Kapacitet* određuju kolekciju *Vojarna*, mjesto gdje djelatnici borave i rade.

2.1.5 Vojna odora

Svaki djelatnik vojske na svojem radnom mjestu mora nositi *Vojnu odoru*. Kolekcija je opisana poljima: *Serijski broj* i *Veličina*.

2.1.6 Vrsta vojne odore

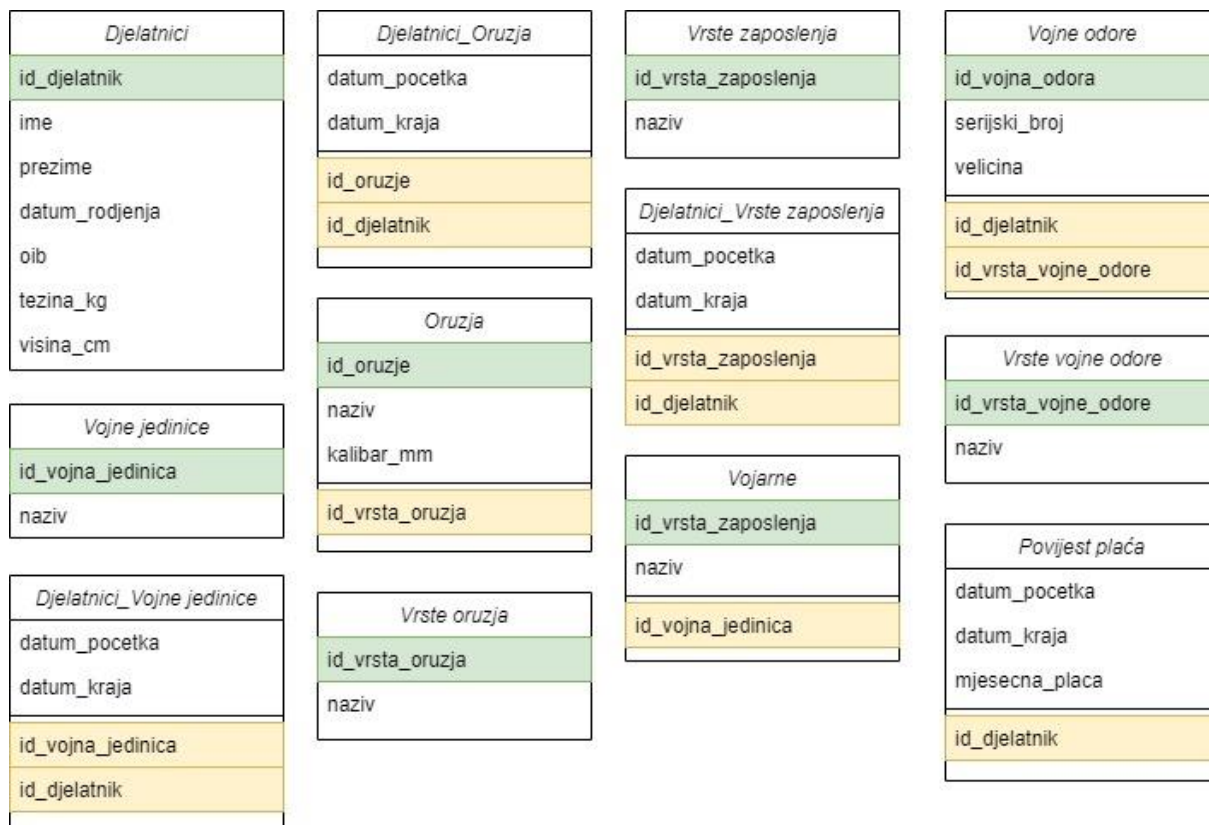
Svaki djelatnik ima zaduženu vojnu odore određene vrste npr. prikladna, svečana itd. Kolekcija je opisana poljem *Naziv*.

2.1.7 Oružje

Svaki djelatnik raspolaže sa određenim *Oružjem* koji mu služi za vojne obuke i obranu. Kolekcija je opisana poljima: *Naziv*, *Masa_kg*, *Kalibar_mm* i *Dodatan_opis*. Oružje se dijeli na više vrsta pa se povezuje s entitetom *Vrsta_oružja* npr. pištolj, puška, mitraljez. Kolekcija je opisana poljem *Naziv*. Svaki djelatnik zadužuje određeno oružje pa se bilježi datum početka i kraja zaduženja kolekcijom *Oružje_Djelatnik*.

2.1.8 Povijest plaće

Kolekcijom *Povijest plaće* se bilježe povijesni podaci mjesečne plaće određenog djelatnika vojske, kada i koliko se promijenila u određenom vremenu.



Slika 1 Struktura baze podataka Vojska

3 IZRADA BAZE PODATAKA

MongoDB je baza podataka dokumenata dizajnirana za jednostavnost razvoja i skaliranja. Nudi lokalnu implementaciju i implementaciju u oblaku:

- Za lokalnu implementaciju, MongoDB nudi i verziju baze podataka za Enterprise i za Community:
 1. MongoDB Community je besplatno izdanje MongoDB.
 2. MongoDB Enterprise dostupan je kao dio MongoDB Enterprise Advanced pretplate i uključuje sveobuhvatnu podršku za implementaciju MongoDB. Također, dodaje značajke usmjerene na poduzeća kao što su podrška za LDAP i Kerberos, enkripcija na disku i revizija.
- MongoDB Atlas je opcija usluge MongoDB Enterprise u oblaku koja ne zahtijeva instalaciju i nudi besplatnu probu korištenja.

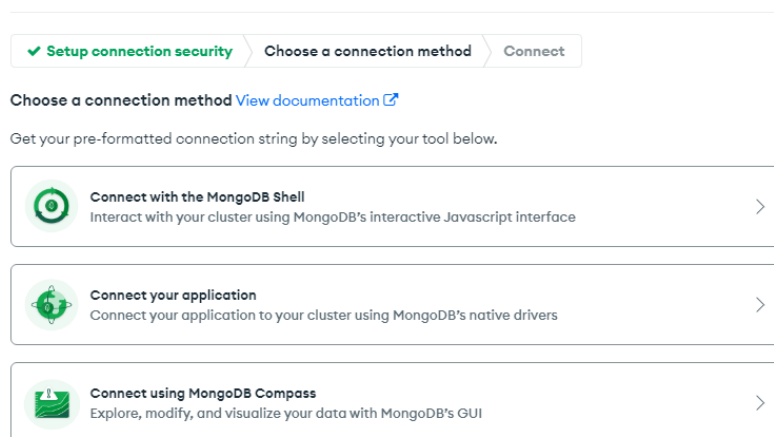
U ovom projektu koristit će se opcija implementacije MongoDB u oblaku, te programski jezik Python kojim će se upravljati navedenom dokument bazom podataka Vojska u besplatnom Jupyter okruženju Google Colab koje je namijenjeno za rad u oblaku.

3.1 Spajanje na MongoDB Atlas Cluster i PyMongo Driver

MongoDB Atlas Cluster je ponuda NoSQL baze podataka kao usluge u javnom oblaku (dostupna u Microsoft Azure, Google Cloud Platform, Amazon Web Services). Ovo je upravljana MongoDB usluga i sa samo nekoliko klikova se može postaviti radni MongoDB klaster kojem se može pristupiti iz web preglednika. MongoDB Atlas besplatni klasteri mogu se koristiti kao malo razvojno okruženje za smještaj podataka. Besplatni klasteri nikada ne istječu i pružaju pristup podskupu Atlasovih značajki. Prvi korak je spajanje na klaster koji predstavlja zbirku baza podataka kojom upravlja jedna instanca pokrenutog poslužitelja baze podataka. (Slika 2)

U ovom slučaju, bira se metoda povezivanja s aplikacijom, odnosno koristeći izvorni MongoDB driver. Zatim se odabira driver i verzija, odnosno MongoDB Python driver verzije 3.12, gdje se nakon odabira prikaže nit za povezivanje (*connection string*) koja se kopira te kasnije služi za spajanje na bazu podataka. (Slika 3)

Connect to Cluster0



✓ Setup connection security > Choose a connection method > Connect

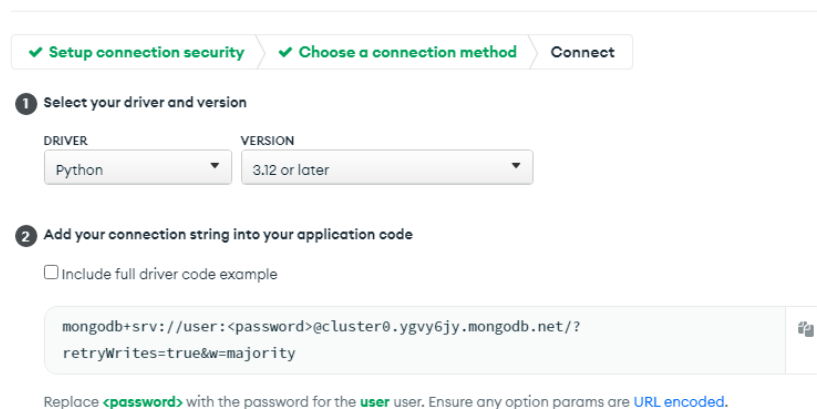
Choose a connection method [View documentation](#)

Get your pre-formatted connection string by selecting your tool below.

- Connect with the MongoDB Shell**
Interact with your cluster using MongoDB's interactive Javascript interface
- Connect your application**
Connect your application to your cluster using MongoDB's native drivers
- Connect using MongoDB Compass**
Explore, modify, and visualize your data with MongoDB's GUI

Slika 2 Spajanje na Cluster0

Connect to Cluster0



✓ Setup connection security > ✓ Choose a connection method > Connect

1 Select your driver and version

DRIVER: Python | VERSION: 3.12 or later

2 Add your connection string into your application code

☐ Include full driver code example

mongodbsrv://user:<password>@cluster0.ygvy6jy.mongodb.net/?retryWrites=true&w=majority

Replace <password> with the password for the user user. Ensure any option params are [URL encoded](#).

Slika 3 Odabir driver-a i verzije

Kod idućeg koraka dodaj se vlastita IP adresa na popis pouzdanih IP adresa. Dakle, IP jedinstveno identificira uređaj koji se povezuje na mrežu. (Slika 4) U Atlasu se može povezati s klasterom samo s pouzdane IP adrese. Unutar Atlasa može se stvoriti popis pouzdanih IP adresa, koji se naziva popis IP pristupa. Popis IP pristupa definira IP adrese koje se mogu povezati s klasterom i pristupiti podacima, te bilo koje povezivanje IP uređaja čija IP adresa nije na popisu je neuspješno. (Slika 5)

Add IP Access List Entry

Atlas only allows client connections to a cluster from entries in the project's IP Access List. Each entry should either be a single IP address or a CIDR-notated range of addresses. [Learn more.](#)

ADD CURRENT IP ADDRESS

Access List Entry:

31.217.110.193

Comment:

Optional comment describing this entry

Slika 4 Postavljanje vlastite IP adrese uređaja

IP Access List

Peering

Private Endpoint

You will only be able to connect to your cluster from the following list of IP Addresses:

IP Address	Comment	Status	Actions
31.217.118.146/32		<div></div> Active	<div>EDIT</div> <div>DELETE</div>
0.0.0.0/0 (includes your current IP address)		<div></div> Active	<div>EDIT</div> <div>DELETE</div>

Slika 5 Popis IP pristupa

Iz sigurnosnih razloga, Atlas zahtijeva od klijenata da se autentificiraju kao korisnici MongoDB baze podataka za pristup klasterima.


Korisnici baze podataka odvojeni su od korisnika Atlasa:

- Korisnici baze podataka mogu pristupiti bazama podataka koje se nalaze u Atlasu.
- Korisnici Atlasa mogu se prijaviti na Atlas, ali nemaju pristup MongoDB bazama podataka.

Dakle, da bi se pristupilo vlastitom klasteru, potrebno je kreirati vlastitog korisnika. Potrebno je postaviti korisničko ime (*username*) i lozinku (*password*) novog korisnika.

Ova kombinacija korisničkog imena i lozinke koristi se da se korisniku dodijeli pristup bazama podataka i dokumentima u vlastitom klasteru u Atlas projektu. (Slika 6) Nakon kreiranja novog korisnika, navedeni se stvori na popisu korisnika koji imaju pristup bazi podataka. (Slika 7)

Add New Database User

Create a database user to grant an application or user, access to databases and collections in your clusters in this Atlas project. Granular access control can be configured with default privileges or custom roles. You can grant access to an Atlas project or organization using the corresponding [Access Manager](#) 

Authentication Method

Password

Certificate

AWS IAM
(MongoDB 4.4 and up)

MongoDB uses [SCRAM](#) as its default authentication method.

Password Authentication

mmravak

..... SHOW

Slika 6 Kreiranje korisnika baze podataka

Database Users		Custom Roles	
User Name ↕	Authentication Method ▲	MongoDB Roles	Resources
 user	SCRAM	readWriteAnyDatabase@admin	All Resources

Slika 7 Popis korisnika koji imaju pristup bazi podataka

Kako bi se povezali s MongoDB bazom podataka, postoji nekoliko Python upravljačkih programa. Može se dodati jedan od sljedećih upravljačkih programa aplikaciji za rad s MongoDB u Pythonu:

- PyMongo za sinkrone Python aplikacije i
- Driver za asinkrone Python aplikacije.

U ovom slučaju koristit će se službeni MongoDB driver za sinkrone Python aplikacije pod nazivom PyMongo, a instalira se pomoću *pip* naredbe u Pythonu.

Također, potrebno je instalirati *dnspython* za povezivanje *pymongo[srv]* protokola. (Slika 8) MongoDB podržava DNS-konstruiran početni popis. Korištenje DNS-a za izradu popisa dostupnih poslužitelja omogućuje veću fleksibilnost implementacije i mogućnost promjene poslužitelja bez ponovnog konfiguriranja klijenata.

Kako bi se iskoristio DNS popis, koristi se prefiks niza veze *mongodb+srv* umjesto standardnog *mongodb*. *+srv* ukazuje klijentu da ime hosta koje slijedi odgovara DNS SRV zapisu. Upravljački program tada će zatražiti od DNSa zapis kako bi odredilo koji host pokreće mongod instancu.

```
!pip install pymongo[srv]
!pip install dnspython # restart runtime
```

Slika 8 Instalacija PyMongo drivera i DNSa

Prvi korak pri radu s PyMongom je stvaranje MongoClienta, gdje klasa MongoClient omogućuje da se uspostavi uspješna MongoDB poslužiteljska veza s aplikacijom, te se pri pozivu prosljeđuje *connection string* koji se dobije pri spajanju na *Cluster0*. Kreiranom objektu *myclient* prosljeđuje se naziv baze podataka koja se želi stvoriti. Zatim, MongoDB kreira bazu podataka ako ne postoji i uspostavlja vezu s njom. (Slika 9)

```
myclient = pymongo.MongoClient("mongodb+srv://user:ITygu2ACRhMGV6cq@cluster0.ygvv6jy.mongodb.net/?retryWrites=true&w=majority")
vojska_db = myclient["Vojska"]
```

Slika 9 Spajanje na mongod instancu i kreiranje baze Vojska

Nakon kreiranja baze podataka, provjerava se da li je stvorena pomoću naredbe *list_database_names()*, a kao rezultat vraća se lista sistemskih baza podataka. Navedene su dvije baze pod nazivima: *admin* i *local*. (Slika 10)

Dakle, važno je naglasiti da se u MongoDB-u baza podataka ne stvara dok ne dobije sadržaj. MongoDB čeka dok se ne stvori kolekcija s barem jednim dokumentom (zapisom) prije nego što zapravo stvori bazu podataka.

```
# Checking if database Vojska exists
print(myclient.list_database_names())

['admin', 'local']
```

Slika 10 Popis sistemskih baza podataka

3.2 Kreiranje kolekcija u bazi podataka Vojska

Za kreiranje kolekcije ili zbirke u MongoDB, koristi se objekt baze podataka i pomoću metode `create_collection()` i uz naveden naziv se stvora željena zbirka. MongoDB kreira kolekciju ako ne postoji u bazi podataka. (Slika 11)

```
# Creating the collection Djelatnici
# djelatnici = vojska_db["Djelatnici"]
djelatnici = vojska_db.create_collection('Djelatnici')

{'$clusterTime': {'clusterTime': Timestamp(1656606340, 33),
  'signature': {'hash': b'\xcb\x9a\xbe\x89\xa8\xa7\x87\xa5&\xf6\x825\xee\xe3dP\xcd\xb8\xd6\xe9',
    'keyId': 7065736675304931333}},
  'nIndexesWas': 1,
  'ns': 'Vojska.Vozila',
  'ok': 1.0,
  'operationTime': Timestamp(1656606340, 32)}
```

Slika 11 Stvaranje kolekcije Djelatnici

Zbog jednostavnog načina unosa podataka u bazu, pomoću Excel tablice kreiran je dokument koji sadrži zapise nekoliko djelatnika vojske. Navedeni zapisi čitaju se iz *xlsx* dokumenta i spremaju u podatkovni okvir, odnosno *DataFrame* koristeći *pandas* biblioteku. (Slika 12) Kako bi se zapisi spremili u bazu potrebno ih je pretvoriti u JSON format što se vrši naredbom `to_dict()`, te se pozivanjem metode `insert_many()` nad kolekcijom *Djelatnici*, konvertirani zapisi spremaju u bazu podataka. (Slika 13) Također, postoji mogućnost unosa pojedinačnog djelatnika u bazu podataka Vojska te se pri tome koristi metoda `insert_one()` gdje se prosljeđuje zapis s potrebnim parovima polje-vrijednost. (Slika 14)

```
# Inserting data from Excel
df_djelatnici = pd.read_excel('djelatnici.xlsx')

# Printing data using Dataframe
df_djelatnici.head()
```

	_id	ime	prezime	datum_rodjenja	oib	tezina_kg	visina_cm
0	1	Janko	Stržić	1.5.1995.	12345678911	69	174
1	2	Vice	Maras	23.8.1985.	28655123523	73	188
2	3	Nino	Abramović	23.8.1996.	41562492214	82	190
3	4	Dino	Mandžukić	22.7.1965.	21388299110	95	195
4	5	Tomislav	Pavletić	1.1.1982.	86182295624	75	192

Slika 12 Podaci o djelatnicima iz Excel tablice

```
# Converting data to JSON format
data = df_djelatnici.to_dict(orient = 'records')

# Inserting data to the collection Djelatnici
djelatnici.insert_many(data)
```

Slika 13 Konverzija zapisa i spremanje u bazu

```
# Inserting djelatnik using method insert_one()

djelatnik_doc = { "_id": 100,
                  "ime": "Joško",
                  "prezime": "Jozić",
                  "datum_rodjenja": '1981-10-2',
                  "oib": 77345578912,
                  "tezina_kg": 75,
                  "visina_cm": 189,
                  "djelatnici_id_djelatnik": 3}

x = djelatnici.insert_one(djelatnik_doc)
```

Slika 14 Unos zapisa za jednog djelatnika

Popis dokumenata odnosno zapisa koji su se spremili mogu se ispisati pomoći *pretty* ispisa. (Slika 15) Također, u bazi podataka kreiranoj u MongoDB Atlas klasteru može se vidjeti stvorena baza *Vojska*, kolekcija *Djelatnici* i spremljeni dokumenti u navedenoj kolekciji. Svaki dokument sadrži parove polje-vrijednost, te za svako polje prikazan je tip podatka, ranije definiran kod zapisa u Excel tablici. (Slika 16)

```
# Pretty printing data from the collection Djelatnici
from pprint import pprint

print('Broj dokumenata u kolekciji Djelatnici je:',
      djelatnici.estimated_document_count())

for x in djelatnici.find():
    pprint(x)

Broj dokumenata u kolekciji Djelatnici je: 11
{'_id': ObjectId('62c4994d4c04d6833910ea4c'),
 'datum_rodjenja': datetime.datetime(1995, 5, 1, 0, 0),
 'id_djelatnik': 1,
 'ime': 'Janko',
 'oib': 12345678911,
 'prezime': 'Strižić',
 'tezina_kg': 69,
 'visina_cm': 174}
{'_id': ObjectId('62c4994d4c04d6833910ea4d'),
 'datum_rodjenja': datetime.datetime(1985, 8, 23, 0, 0),
 'id_djelatnik': 2,
 'ime': 'Vice',
 'oib': 28655123523,
 'prezime': 'Maras',
 'tezina_kg': 73,
 'visina_cm': 188}
```

Slika 15 Ispis zapisa u kolekciji Djelatnici

Vojska.Djelatnici

STORAGE SIZE: 20KB TOTAL DOCUMENTS: 11 INDEXES TOTAL SIZE: 20KB

Find Indexes Schema Anti-Patterns ① Aggregation Search Indexes ●

INSERT DOCUMENT

FILTER { field: 'value' } **OPTIONS** **Apply**

QUERY RESULTS: 1-11 OF 11

1	_id: ObjectId("62c4994d4c04d6833910ea4c")	ObjectId
2	id_djelatnik: 1	Int32
3	ime: "Janko"	String
4	prezime: "Strižić"	String
5	datum_rodjenja: 1995-05-01T00:00:00.000+00:00	Date
6	oib: 12345678911	Int64
7	tezina_kg: 69	Int32
8	visina_cm: 174	Int32

Slika 16 Dokumenti kreirani u kolekciji Djelatnici

Nadalje, ako se žele ispisati samo neki od parova polje-vrijednost, to se vrši pomoću naredbe *find()* u kojoj se šalje željeni upit, a koji se izvršava nad cijelim skupom zapisa. U ovom slučaju, iz cijele kolekcije žele se dobiti ispis zapisa na način da se ispišu vrijednosti polja *prezime*, *oib* i *datum_rodjenja*. (Slika 17)

```
# Printing djelatnici using find method with specific query
for x in djelatnici.find({}, {"_id": 0,
                              'prezime': 1,
                              "datum_rodjenja": 1,
                              "oib": 1 }):
    pprint(x)

{'datum_rodjenja': datetime.datetime(1995, 5, 1, 0, 0),
 'oib': 12345678911,
 'prezime': 'Strižić'}
{'datum_rodjenja': datetime.datetime(1985, 8, 23, 0, 0),
 'oib': 28655123523,
 'prezime': 'Maras'}
{'datum_rodjenja': datetime.datetime(1996, 8, 23, 0, 0),
 'oib': 41562492214,
 'prezime': 'Abramović'}
{'datum_rodjenja': datetime.datetime(1965, 7, 22, 0, 0),
 'oib': 21388299110,
 'prezime': 'Mandžukić'}
{'datum_rodjenja': datetime.datetime(1982, 1, 1, 0, 0),
 'oib': 86182295624,
 'prezime': 'Pavletić'}
{'datum_rodjenja': datetime.datetime(1988, 9, 24, 0, 0),
 'oib': 77139039464,
 'prezime': 'Maras'}
```

Slika 17 Metoda *find()* sa željenim queryjem

Nakon kreiranja baze *Vojska* i prve kolekcije *Djelatnici* sa zapisima, može se provjeriti da li sada baza postoji na listi sistemskih baza podataka. Kao rezultat dobije se lista baza, te se sada ispisuje i nova kreirana baza (Slika 18)

```
# Checking if database Vojska exists
print(myclient.list_database_names())

['Vojska', 'admin', 'local']
```

Slika 18 Lista sistemskih baza nakon kreiranja kolekcije *Djelatnici*

Na istom principu kreirano je ukupno 12 kolekcija. (Slika 19)

```
# List of collection names in database Vojska
print('Broj kolekcija u bazi vojska je:',
      len(vojska_db.list_collection_names()))
vojska_db.list_collection_names()

Broj kolekcija u bazi vojska je: 12
['Vojne odore',
 'Djelatnici',
 'Djelatnici_Vojne_jedinice',
 'Vrste zaposlenja',
 'Djelatnici_Oruzja',
 'Povijest placa',
 'Oruzja',
 'Vojarne',
 'Vojne_jedinice',
 'Djelatnici_Vrste_zaposlenja',
 'Vrste_vojne_odore',
 'Vrste_oruzja']
```

Slika 19 Lista svih kolekcija baze Vojska

4 MONGODB CRUD OPERACIJE

Prije izvršavanja CRUD operacija nad dokumentima, potrebno je dohvatiti kolekciju nad kojom se žele izvršavati promjene, a to se vrši pomoću metode `get_collection()` kojoj se proslijeđuje naziv kolekcije, u ovom slučaju to je kolekcija *djelatnici*. MongoDB CRUD operacije omogućuju kreiranje (*create*), čitanje (*read*), ažuriranje (*update*) i brisanje (*delete*) dokumenata kolekcija u bazi podataka. Za početak, dohvaća se kolekcija nad kojoj će se vršiti promjene, te se ispisuje broj dokumenata koje sadrži. (Slika 20)

```
djelatnici = vojska_db.get_collection('Djelatnici')

num = djelatnici.estimated_document_count()
print('Broj zapisa u kolekciji Djelatnici je:', num)

Broj zapisa u kolekciji Djelatnici je: 11
```

Slika 20 Dohvaćanje željene kolekcije

4.1 Insert (Create)

Za umetanje dokumenta pod nazivom *djelatnik_doc1* u kolekciju *djelatnici* baze podataka koristiti se metoda `insert_one()`. Isto tako, za umetanje više od jednog zapisa upotrebljava se metoda `insert_many()` koja je već korištena prilikom ubacivanja dokumenata u 12 navedenih kolekcija u bazu podataka *Vojska*.

Prikazani su primjeri gdje se koristi metoda *insert_one()* za unos jednog dokumenta u kolekciju *djelatnici*, a potom *insert_many()* za unos dva dokumenta. (Slika 21)

```
# Inserting djelatnik using method insert_one()

djelatnik_doc1 = { "id_djelatnik": 14,
                  "ime": "Karlo",
                  "prezime": "Maras",
                  "datum_rodjenja": dt.datetime(1997, 11, 2),
                  "oib": 45362713476,
                  "tezina_kg": 85,
                  "visina_cm": 185}

djelatnici.insert_one(djelatnik_doc1)

# Inserting multiple djelatnici using method insert_many()
djelatnik_doc2 = { "id_djelatnik": 16,
                  "ime": "Mate",
                  "prezime": "Matić",
                  "datum_rodjenja": dt.datetime(1995, 5, 5),
                  "oib": 32648911730,
                  "tezina_kg": 74,
                  "visina_cm": 176}

djelatnik_doc3 = { "id_djelatnik": 17,
                  "ime": "Josip",
                  "prezime": "Janjac",
                  "datum_rodjenja": dt.datetime(1989, 8, 17),
                  "oib": 13489547724,
                  "tezina_kg": 88,
                  "visina_cm": 191}

djelatnici.insert_many([djelatnik_doc2, djelatnik_doc3])
```

Slika 21 CREATE operacija dokumenata u kolekciju

4.2 Select (Read)

Čitanje dokumenta u MongoDB-u vrlo je jednostavno, a za to se upotrebljava metoda koja se zove *find_one()* koja prima *query* ili *filter* odnosno upit u kojem se definira koji dokument se želi ispisati. Kao što njeno ime pokazuje, jednostavno će pronaći samo jedan dokument koji se želi dobiti iz kolekcije. Naveden je primjer koji pokazuje kako se već ranije unesen dokument pod nazivom *djelatnik_doc1* čita pomoću metode *find_one()* u koju se proslijeđuje upit s definiranim *id_djelatnik* koji odgovara navedenom djelatniku.

Nadalje, za čitanje više od jednog dokumenta neke kolekcije, koristi se metoda *find()*. Kako bi se ispisao pronađeni dokument, koristi se *pprint()* metoda gdje se podaci ispišu na strukturirani način JSON formata. Prikazan je primjer kako se pomoću *find()* metode čitaju dokumenti pod nazivima *djelatnik_doc2* i *djelatnik_doc3* koji su dodani u prethodnom odjeljku. (Slika 22)

```
# Checking if djelatnik with id = 14 exists by method find_one()
myquery = { "id_djelatnik": 14 }
djelatnici.find_one(myquery)

{'_id': ObjectId('62cc1e36e3d381ce71ebf9c7'),
 'datum_rodjenja': datetime.datetime(1997, 11, 2, 0, 0),
 'id_djelatnik': 14,
 'ime': 'Karlo',
 'oib': 45362713476,
 'prezime': 'Maras',
 'tezina_kg': 85,
 'visina_cm': 185}

# Checking if djelatnici with ids = [16, 17] exist by method find()
myquery = {'id_djelatnik' : {"$in" : [16, 17]}}
for d in djelatnici.find(myquery):
    pprint(d)

{'_id': ObjectId('62cc174be3d381ce71ebf9bf'),
 'datum_rodjenja': datetime.datetime(1995, 5, 5, 0, 0),
 'id_djelatnik': 16,
 'ime': 'Mate',
 'oib': 32648911730,
 'prezime': 'Matić',
 'tezina_kg': 74,
 'visina_cm': 176}
{'_id': ObjectId('62cc174be3d381ce71ebf9c0'),
 'datum_rodjenja': datetime.datetime(1989, 8, 17, 0, 0),
 'id_djelatnik': 17,
 'ime': 'Josip',
 'oib': 13489547724,
 'prezime': 'Janjac',
 'tezina_kg': 88,
 'visina_cm': 191}
```

Slika 22 READ operacija dokumenata u kolekciju

4.3 Update

Za ažuriranje dokumenta iz kolekcije baze podataka, u navedenom primjeru postavljaju se kriteriji kako bi se ažurirao željeni dokument na određeni način. Dakle, ažurira se *ime* djelatnika jer je pri unosu podataka došlo do duplih zapisa. Za to treba koristiti metodu *find_one_and_update()* kojoj se proslijeđuju dva argumenta, jedan od njih zadužen je za čitanje dokumenta iz kolekcije, a drugi za ažuriranje gdje se koristi operator *\$set* s kojim se postavljaju novi podaci za navedeni dokument pohranjen u kolekciji baze podataka. Da bi se provjerilo navedeno ažuriranje, koriste se *find_one()* metoda i upit kojim je definirano *ime* koje se ažuriralo s odgovarajućim *id_djelatnik*. (Slika 23)

```
# Updating specific value in document of djelatnik with id = 14
find_one = {'id_djelatnik': 14}
update = {'$set': {'ime': 'Jure'}}
djelatnici.find_one_and_update(find_one, update)

{'_id': ObjectId('62cc1e36e3d381ce71ebf9c7'),
 'datum_rodjenja': datetime.datetime(1997, 11, 2, 0, 0),
 'id_djelatnik': 14,
 'ime': 'Karlo',
 'oib': 45362713476,
 'prezime': 'Maras',
 'tezina_kg': 85,
 'visina_cm': 185}

# Checking if djelatnik with name = 'Jure' exist by method find_one()
myquery = {'ime' : 'Jure', 'id_djelatnik': 14}
djelatnici.find_one(myquery)

{'_id': ObjectId('62cc1e36e3d381ce71ebf9c7'),
 'datum_rodjenja': datetime.datetime(1997, 11, 2, 0, 0),
 'id_djelatnik': 14,
 'ime': 'Jure',
 'oib': 45362713476,
 'prezime': 'Maras',
 'tezina_kg': 85,
 'visina_cm': 185}
```

Slika 23 UPDATE operacija dokumenta u kolekciju

Može se vidjeti razlika jer se ažurirao dokument koji pokazuje promjenu polja *ime* iz *Ivan* u *Jure*. Dakle, proces je bio uspješan i može se reći da se radi o istom dokumentu jer je *id_djelatnik* isti kao i prije.

Također, MongoDB pruža značajku koja je poznata kao ugrađeni ili ugniježđeni dokument. Ugrađeni dokumenti ili ugniježđeni dokumenti su vrste dokumenata koji sadrže dokument unutar drugog dokumenta. Ili drugim riječima, kada kolekcija ima dokument, ovaj dokument sadrži drugi dokument, drugi dokument sadrži drugi pod-dokument i tako dalje.

U MongoDB-u može se jednostavno ugraditi dokument u drugi dokument. Kao što znamo da su u mongo ljusci dokumenti predstavljeni pomoću vitičastih zagrada ({ }) i unutar tih vitičastih zagrada imamo parove vrijednosti-polja. No, unutar ovih polja može se ugraditi drugi dokument pomoću vitičastih zagrada { } i ovaj dokument može sadržavati parove polja-vrijednosti ili neki drugi pod-dokument. Prikazan je primjer gdje se u kolekciji *djelatnici* pomoću metode *update_one()* i koristeći operator *\$push* ažurira djelatnik s *id_djelatnik* = 14 u kojem se dodaje ugrađeni dokument *adresa* s poljima *ulica*, *kucni_broj*, *mjesto* i *postanski_broj*. (Slika 24)

```
# Update djelatnik with id = 14 using embedded document adresa
myquery = {'id_djelatnik': 14}
push = {'$push':
        { 'adresa': {
            'ulica': 'Solinska ulica',
            'kucni_broj': 5,
            'mjesto': 'Split',
            'postanski_broj': 21000
        }
    }}
djelatnici.update_one(myquery, push)
```

```
# Checking if djelatnik with id = 14 contains
# embedded document adresa
myquery = {'id_djelatnik': 14}
djelatnici.find_one(myquery)

{'_id': ObjectId('62cc1e36e3d381ce71ebf9c7'),
 'adresa': [{'kucni_broj': 5,
             'mjesto': 'Split',
             'postanski_broj': 21000,
             'ulica': 'Solinska ulica'}],
 'datum_rodjenja': datetime.datetime(1997, 11, 2, 0, 0),
 'id_djelatnik': 14,
 'ime': 'Jure',
 'oib': 45362713476,
 'prezime': 'Maras',
 'tezina_kg': 85,
 'visina_cm': 185}
```

Slika 24 Ugrađeni dokument 'adresa'

4.4 Delete

Proces brisanja dokumenta vrlo je jednostavan kao i čitanje dokumenta. Dakle, koristi se metoda *delete_one()* koja isto tako kao i metoda *find_one()* prima argument koji predstavlja upit kojim se filtrira odnosno kod brisanja, briše željeni dokument. U primjeru, upotrebljava se polje *id_djelatnik* za pronalazak i brisanje dokumenta za djelatnika navedenog id-ija. Nakon brisanja, vrši se provjera popisa dokumenata u kolekciji *djelatnik* koristeći metodu *find_one()* kojoj se proslijeđuje upit kojim je definirano da se traži djelatnik s istim *id_djelatnik*, te se vidi da je brisanje bilo uspješno te da takav zapis više ne postoji u kolekciji *djelatnici*. Za brisanje više od jednog dokumenta koristi se metoda *delete_many()*. (Slika 25)

```
# Deleting djelatnik with id = 15

myquery = { "id_djelatnik": 15}

djelatnici.delete_one(myquery)

<pymongo.results.DeleteResult at 0x7f1014798d50>

# Checking if djelatnik with id = 15 exist by method find()
myquery = {'id_djelatnik': 15}
for d in djelatnici.find(myquery):
    pprint(d)

print('Djelatnik je izbrisan.')
print('Postoji', djelatnici.count_documents(myquery), 'zapisa.')

Djelatnik je izbrisan.
Postoji 0 zapisa.

# Deleting multiple djelatnici with prezime = 'Maras'
find_many = {'prezime': 'Maras'}
print('Postoje', djelatnici.count_documents(find_many), 'zapisa.')
djelatnici.delete_many(find_many)

print('Djelatnici su izbrisani.')
print('Postoji', djelatnici.count_documents(myquery), 'zapisa.')

Postoje 2 zapisa.
Djelatnici su izbrisani.
Postoji 0 zapisa.
```

Slika 25 DELETE operacija dokumenta u kolekciju

5 ZADACI

Zadatak 1. Prikaži ime i prezime djelatnika i naziv oružja kojeg je djelatnik zadužio u određenom datumu, te sortiraj djelatnike po prezimenu uzlazno. (Slika 26 i 27)

```
task_1 = vojska_db.get_collection('Djelatnici_Oruzja').aggregate( [
    {
        '$lookup':
        {
            'from': "Djelatnici",
            'localField': "id_djelatnik",
            'foreignField': "id_djelatnik",
            'as': "djelatnik"
        }
    },
    {
        '$lookup':
        {
            'from': "Oruzja",
            'localField': "id_oruzje",
            'foreignField': "id_oruzje",
            'as': "oruzje"
        }
    },
    {
        '$project':
        {
            '_id': 0,
            'datum_pocetka': 1,
            'datum_kraja': 1,
            'djelatnik.ime': 1,
            'djelatnik.prezime': 1,
            'oruzje.naziv': 1
        }
    },
    {
        '$sort':
        {
            'djelatnik.prezime': 1
        }
    }
] )

for r in task_1:
    pprint(r)
```

Slika 26 Kód zadatka 1

```
{
  'datum_kraja': datetime.datetime(2020, 2, 22, 0, 0),
  'datum_pocetka': datetime.datetime(2010, 7, 27, 0, 0),
  'djelatnik': [{'ime': 'Nino', 'prezime': 'Abramović'}],
  'oruzje': [{'naziv': 'HS 2000'}]}
{
  'datum_kraja': datetime.datetime(2022, 6, 30, 0, 0),
  'datum_pocetka': datetime.datetime(2021, 1, 20, 0, 0),
  'djelatnik': [{'ime': 'Patrik', 'prezime': 'Antić'}],
  'oruzje': [{'naziv': 'VHS'}]}
{
  'datum_kraja': datetime.datetime(1999, 9, 1, 0, 0),
  'datum_pocetka': datetime.datetime(1994, 5, 20, 0, 0),
  'djelatnik': [{'ime': 'Matko ', 'prezime': 'Babić'}],
  'oruzje': [{'naziv': 'M4'}]}
{
  'datum_kraja': datetime.datetime(2021, 10, 1, 0, 0),
  'datum_pocetka': datetime.datetime(2005, 7, 1, 0, 0),
  'djelatnik': [{'ime': 'Dino', 'prezime': 'Mandžukić'}],
  'oruzje': [{'naziv': 'FN F2000'}]}
{
  'datum_kraja': datetime.datetime(2021, 4, 1, 0, 0),
  'datum_pocetka': datetime.datetime(2020, 3, 1, 0, 0),
  'djelatnik': [{'ime': 'Vice', 'prezime': 'Maras'}],
  'oruzje': [{'naziv': 'ERO'}]}
{
  'datum_kraja': datetime.datetime(2021, 4, 1, 0, 0),
  'datum_pocetka': datetime.datetime(2020, 5, 1, 0, 0),
  'djelatnik': [{'ime': 'Janko', 'prezime': 'Strižić'}],
  'oruzje': [{'naziv': 'Springfield Armory XDM'}]}
{
  'datum_kraja': datetime.datetime(2021, 8, 8, 0, 0),
  'datum_pocetka': datetime.datetime(2019, 3, 15, 0, 0),
  'djelatnik': [{'ime': 'Karlo', 'prezime': 'Tomić'}],
  'oruzje': [{'naziv': 'HS 2000'}]}
```

Slika 27 Rješenje zadatka 1

Zadatak 2. Prikaži oib i prezime djelatnika grupiranih po nazivu vojne jedinice u kojoj djelatnici rade i borave. (Slika 28 i 29)

```
task_2 = vojska_db.get_collection('Djelatnici_Vojne_jedinice').aggregate( [
{
  '$lookup':
  {
    'from': "Djelatnici",
    'localField': "id_djelatnik",
    'foreignField': "id_djelatnik",
    'as': "djelatnik"
  }
},
{
  '$lookup':
  {
    'from': "Vojne_jedinice",
    'localField': "id_vojna_jedinica",
    'foreignField': "id_vojna_jedinica",
    'as': "vojna_jedinica"
  }
},
],
```

```

{
    '$match':
    {
        'vojna_jedinica.naziv': { "$in" : ['vod', 'desetina']}
    }
},
{
    '$group':
    {
        '_id' : "$vojna_jedinica.naziv",
        'djelatnici': {
            '$push':
            {
                'prezime': '$djelatnik.prezime',
                'oib': '$djelatnik.oib'
            }
        }
    }
}
] )

for r in task_2:
    pprint(r)

```

Slika 28 Kôd zadatka 2

```

{'_id': ['vod'],
 'djelatnici': [{ 'oib': [12345678911], 'prezime': ['Strižić']},
                 { 'oib': [23925600058], 'prezime': ['Antić']}]},
{'_id': ['desetina'],
 'djelatnici': [{ 'oib': [13272077197], 'prezime': ['Tomić']},
                 { 'oib': [41562492214], 'prezime': ['Abramović']}]}}

```

Slika 29 Rješenje zadatka 2

Zadatak 3. Prikaži ime, prezime djelatnika grupirane po nazivu vrste zaposlenja iz liste ['običan vojnik', 'general-pukovnik'] koju je djelatnik obavljao u određenom datumu. (Slika 30 i 31)

```

task_3 = vojska_db.get_collection('Djelatnici_Vrste_zaposlenja').aggregate( [
    {
        '$lookup':
        {
            'from': "Djelatnici",
            'localField': "id_djelatnik",
            'foreignField': "id_djelatnik",
            'as': "djelatnik"
        }
    },
    {
        '$lookup':
        {
            'from': "Vrste_zaposlenja",
            'localField': "id_vrsta_zaposlenja",
            'foreignField': "id_vrsta_zaposlenja",
            'as': "vrsta_zaposlenja"
        }
    },
    {

```

```

    '$match':
    {
        'vrsta_zaposlenja.vrsta':
        {
            "$in" : ['obični vojnik', 'general-pukovnik']
        }
    },
    {
        '$group':
        {
            '_id' : '$vrsta_zaposlenja.vrsta',
            'djelatnici': {
                '$push':
                {
                    'ime': '$djelatnik.ime',
                    'prezime': '$djelatnik.prezime',
                    'oib': '$djelatnik.oib',
                    'datum_pocetka': '$datum_pocetka',
                    'datum_kraja': '$datum_kraja'
                }
            }
        }
    }
] )

```

Slika 30 Kôd zadatka 3

```

for r in task_3:
    pprint(r)

{'_id': ['obični vojnik'],
 'djelatnici': [{'datum_kraja': datetime.datetime(2022, 6, 30, 0, 0),
                  'datum_pocetka': datetime.datetime(2021, 1, 20, 0, 0),
                  'ime': ['Patrik'],
                  'oib': [23925600058],
                  'prezime': ['Antić']},
                {'datum_kraja': datetime.datetime(2021, 8, 8, 0, 0),
                  'datum_pocetka': datetime.datetime(2019, 3, 15, 0, 0),
                  'ime': ['Karlo'],
                  'oib': [13272077197],
                  'prezime': ['Tomić']},
                {'datum_kraja': datetime.datetime(2020, 2, 22, 0, 0),
                  'datum_pocetka': datetime.datetime(2010, 7, 27, 0, 0),
                  'ime': ['Nino'],
                  'oib': [41562492214],
                  'prezime': ['Abramović']}]},
{'_id': ['general-pukovnik'],
 'djelatnici': [{'datum_kraja': datetime.datetime(1999, 9, 1, 0, 0),
                  'datum_pocetka': datetime.datetime(1994, 5, 20, 0, 0),
                  'ime': ['Matko'],
                  'oib': [94184066307],
                  'prezime': ['Babić']}]},

```

Slika 31 Rješenje zadatka 3

Zadatak 4. Prikaži naziv, mjesto i adresu vojarne koja sadrži vojne jedinice iz liste s nazivima ['vod', 'desetina'] grupirano po nazivu vojne jedinice. (Slika 32 i 33)

```
task_4 = vojska_db.get_collection('Vojarne').aggregate( [
{
    '$lookup':
    {
        'from': "Vojne jedinice",
        'localField': "id_vojna_jedinica",
        'foreignField': "id_vojna_jedinica",
        'as': "vojna_jedinica"
    }
},
{
    '$match':
    {
        'vojna_jedinica.naziv': { "$in" : ['vod', 'desetina']}
    }
},
{
    '$group':
    {
        '_id' : '$vojna_jedinica.naziv',
        'vojarna': {
            '$push':
            {
                'naziv': '$naziv',
                'mjesto': '$mjesto',
                'adresa': '$adresa'
            }
        }
    }
}
] )

for r in task_4:
    pprint(r)
```

Slika 32 Kôd zadatka 4

```
{ '_id': ['desetina'],
  'vojarna': [{ 'adresa': 'Ulica Nikole Tordinca bb',
                'mjesto': 'Vinkovci',
                'naziv': '5. gardijska brigada "Slavonski sokolovi"',
                'adresa': nan, 'mjesto': 'Pula', 'naziv': 'Vargarola'},
              { 'adresa': nan,
                'mjesto': 'Ploče',
                'naziv': '116. brigada HV "Neretvanski gusari"'}] },
{ '_id': ['vod'],
  'vojarna': [{ 'adresa': 'Ulica Nikole Tesle bb',
                'mjesto': 'Đakovo',
                'naziv': '3. gardijska brigada "Kune"',
                'adresa': 'Ulica kralja Tomislava 118',
                'mjesto': 'Našice',
                'naziv': '132. brigada'},
              { 'adresa': 'Ulica Martina Frankopana bb',
                'mjesto': 'Petrinja',
                'naziv': 'Pukovnik Predrag Matanović'},
              { 'adresa': 'Ulica Kaniška 102',
                'mjesto': 'Gospić',
                'naziv': '9. gardijska brigada "Vukovi"'}] }
```

Slika 33 Rješenje zadatka 4

Zadatak 5. Prikaži prezime i oib djelatnika, naziv vrste zaposlenja te serijski broj vojne odore grupirano po veličini vojne odore gdje veličina odgovara prema listi ['S', 'M']. (Slika 34 i 35)

```
# Task_5: Select prezime, oib of djelatnici, vrsta_vojne_odore.naziv and
# serijski_broj of vojne odore group by vojna.odora.velicina.

task_5 = vojska_db.get_collection('Vojne odore').aggregate( [
    {
        '$lookup':
        {
            'from': "Vrste vojne odore",
            'localField': "id_vrsta_vojne_odore",
            'foreignField': "id_vrsta_vojne_odore",
            'as': "vrsta_vojne_odore"
        }
    },
    {
        '$lookup':
        {
            'from': "Djelatnici",
            'localField': "id_djelatnik",
            'foreignField': "id_djelatnik",
            'as': "djelatnik"
        }
    },
    {
        '$match':
        {
            'velicina': { "$in" : ['S', 'M']}
        }
    },
    {
        '$group':
        {
            '_id' : '$velicina',
            'djelatnici_vrsta_odore': {
                '$push':
                {
                    'prezime': '$djelatnik.prezime',
                    'oib': '$djelatnik.oib',
                    'naziv': '$vrsta_vojne_odore.naziv',
                    'serijski_broj': '$serijski_broj'
                }
            }
        }
    }
] )

for r in task_5:
    pprint(r)
```

Slika 34 Kód zadatka 5

```

{'_id': 'M',
 'djelatnici_vrsta_odore': [{ 'naziv': ['službena'],
                              'oib': [12345678911],
                              'prezime': ['Strižić'],
                              'serijski_broj': '8yWqdRcj'},
                             { 'naziv': ['svečana'],
                              'oib': [21388299110],
                              'prezime': ['Mandžukić'],
                              'serijski_broj': 'QzxaVmdi'},
                             { 'naziv': ['službena'],
                              'oib': [12872703676],
                              'prezime': ['Jožić'],
                              'serijski_broj': '5DbCTTbQ'},
                             { 'naziv': ['posebna odjeća i obuća'],
                              'oib': [13362713476],
                              'prezime': ['Pavelić'],
                              'serijski_broj': 'qwPnz8YA'},
                             { 'naziv': ['službena'],
                              'oib': [13362713476],
                              'prezime': ['Pavelić'],
                              'serijski_broj': 'hQinqiSv'},
                             { 'naziv': ['svečana'],
                              'oib': [23925600058],
                              'prezime': ['Antić'],
                              'serijski_broj': 'X25QsZb2'},
                             { 'naziv': ['službena'],
                              'oib': [23925600058],
                              'prezime': ['Antić'],
                              'serijski_broj': 'Y2kgAZim'}]
},
{'_id': 'S',
 'djelatnici_vrsta_odore': [{ 'naziv': ['prikrivna'],
                              'oib': [41562492214],
                              'prezime': ['Abramović'],
                              'serijski_broj': '9XJZgjuJ'},
                             { 'naziv': ['prikrivna'],
                              'oib': [12872703676],
                              'prezime': ['Jožić'],
                              'serijski_broj': 'PZdZRT94'}]
}
```

Slika 35 Rješenje zadatka 5

6 POPIS SLIKA

Slika 1 Struktura baze podataka Vojska.....	6
Slika 2 Spajanje na Cluster0.....	7
Slika 3 Odabir driver-a i verzije	7
Slika 4 Postavljanje vlastite IP adrese uređaja	8
Slika 5 Popis IP pristupa	8
Slika 6 Kreiranje korisnika baze podataka	9
Slika 7 Popis korisnika koji imaju pristup bazi podataka	9
Slika 8 Instalacija PyMongo drivera i DNSa	10
Slika 9 Spajanje na mongod instancu i kreiranje baze Vojska.....	10
Slika 10 Popis sistemskih baza podataka	10
Slika 11 Stvaranje kolekcije Djelatnici	11
Slika 12 Podaci o djelatnicima iz Excel tablice	11
Slika 13 Konverzija zapisa i spremanje u bazu.....	12
Slika 14 Unos zapisa za jednog djelatnika	12
Slika 15 Ispis zapisa u kolekciji Djelatnici	13
Slika 16 Dokumenti kreirani u kolekciji Djelatnici.....	13
Slika 17 Metoda find() sa željenim queryjem	14
Slika 18 Lista sistemskih baza nakon kreiranja kolekcije Djelatnici	14
Slika 19 Lista svih kolekcija baze Vojska.....	15
Slika 20 Dohvaćanje željene kolekcije.....	15
Slika 21 CREATE operacija dokumenata u kolekciju	16
Slika 22 READ operacija dokumenata u kolekciju.....	17
Slika 23 UPDATE operacija dokumenta u kolekciju.....	18
Slika 24 DELETE operacija dokumenta u kolekciju	20
Slika 25 Kôd zadatka 1.....	21

Slika 26 Rješenje zadatka 1	22
Slika 27 Kôd zadatka 2.....	23
Slika 28 Rješenje zadatka 2.....	23
Slika 29 Kôd zadatka 3.....	24
Slika 30 Rješenje zadatka 3.....	24
Slika 31 Kôd zadatka 4.....	25
Slika 32 Rješenje zadatka 4.....	25
Slika 33 Kôd zadatka 5.....	26
Slika 34 Rješenje zadatka 5.....	27