

## Cerinte proiect ECBD 2024:

Va fi postata o singura arhiva .zip cu numele si grupa, care va contine:

### 1. (2p) Documentatie in format pdf (~4-5 pagini):

- titlu, nume student, grupa, cadru didactic coordinator – temele sunt individuale
- descrierea proiectului (scopul aplicatiei): tema este aleasa de studenti ~ 1/2 pagina
- structura colectiilor cu tipurile de date (minim 3 colectii), conform cerintei 2 a), cu explicatii ~ 1-2 pagini
- descrierea functionalitatilor si a operatiilor corespunzatoare din fisierele de script ~ 2-3 pagini
- documentatia nu va contine script-uri cu instructiuni

### 2. Cerinte proiect:

#### a) Structura colectiilor (care va fi detaliata in documentatie):

- colectiile sa aiba valori pentru `_id` autodefinite (*ObjectId*) si/sau definite de utilizator
- atributele sa aiba tipuri scalare de tip *String*, *Integer*, *Double*, *Boolean*, *Date/Timestamp* si cel putin unul din tipurile: *Binary data* sau *Regular expression*
- atribute de tip vector de valori scalare
- atribute de tip subdocument
- atribute de tip vector de subdocumente
- orice combinatie care e optima pentru problema de implementat

#### b) (2p) Operatii CRUD:

- inserarea datelor multiple in toate colectiile folosite
- modificarea si stergerea documentelor individuale
- actualizarea anumitor elemente din vectorii de subdocumente din colectiile folosite

#### c) (1.5p) Cautari simple:

- cautarea documentelor/subdocumentelor/elementelor din vector folosind *find()* cu diverse optiuni pentru cautari in vectori/subdocumente, proiectia atributelor, valori null, atribute inexistente, sortari
- cautare paginata pentru unele operatii de cautare (ca parametri se paseaza numarul de documente pe pagina si numarul paginii)

#### d) (3.5p) Cautari complexe:

- folosirea cursorilor pentru *find()* pentru procesari suplimentare
- folosirea frameworkului *aggregate* (operatori de tip *group*, *push/addFields*, *project*, *unwind*, *sort*, *limit*) si conceptul de *pipeline* pentru inlantuirea operatorilor
- folosirea operatorului *\$lookup* pentru interogarea datelor din mai multe colectii

3. **Creata scripturi** pentru fiecare dintre operatiile implementate (acestea sa fie grupate in fisiere pe directoare pe categoriile de cerinte de la punctul 2 – b, c, d)

### Barem de notare:

(1p) oficiu

(2p) documentatia corecta si completa conform cerintelor

(2p) scripturile cu operatiile de tip CRUD

(1.5p) scripturile cu operatiile de cautare simpla

(0.5p) cursori folositi in scripturile cu operatiile de cautare complexa

(1p) folosirea operatorului *\$lookup* in scripturile cu operatiile de cautare complexa

(2p) folosirea celorlalti operatori in scripturile cu operatiile de cautare complexa