



**UNIVERSITATEA TEHNICĂ GHEORGHE ASACHI**

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**

**Gestionarea comenzilor dintr-un magazin  
online**

**- EXTRAGEREA CUNOȘTINȚELOR DIN  
BAZE DE DATE -**

STUDENT

**Paraschiv Florin-Vlăduț, 1409B**

# 1 Descriere proiect

Scopul proiectului este de a ușura procesul prin care se realizează o comandă la un magazin online. Prin intermediul celor 3 colecții, se rețin informațiile esențiale pentru o comandă, și anume utilizatorul care o plasează, produsele comandate și detalii despre comandă în sine, statusul și data la care s-a plasat. În cadrul aplicației, utilizatorii pot plasa mai multe comenzi, cât și o comandă cu mai multe produse, împreună cu cantitatea dorită.

## 2 Structura colecțiilor

În cadrul proiectului există 3 colecții:

- **Users:** este format din 6 proprietăți, din care 3 sunt obligatorii, și este folosită pentru a reține informații despre utilizatori. Pentru fiecare proprietate s-au definit și validatori, ca de exemplu email-ul să respecte formatul de email.
- **Products:** este format din 19 proprietăți, din care 4 sunt obligatorii, și este folosită pentru stocarea detaliilor despre comenzi. Comenzile pot avea categorii diferite, care vin și cu proprietăți diferite, și, la fel ca la colecția de utilizatori, s-au definit și validări pentru fiecare proprietate în parte.
- **Orders:** este format din 4 proprietăți, toate fiind obligatorii, fiind nevoie de utilizatorul care a plasat comanda, produsele pe care le-a comandat, statusul comenzii și data la care s-a plasat comanda.

Structura colecțiilor cu validarea acestora este următoarea:

```
1  bsonType: "object",
2  required: ["_id", "username", "email"],
3  properties: {
4    _id: {
5      bsonType: "int",
6      minimum: 1,
7      description: "Must be an integer greater than 0 and is required",
8    },
9
10   username: {
11     bsonType: "string",
12     minLength: 5,
13     maxLength: 30,
14     description: "must be a string and is required",
15   },
16
17   email: {
18     bsonType: "string",
19     pattern: "^.+@.+$",
20     description: "must be a string and match the regular expression
21       pattern",
22   },
23
24   firstName: {
25     bsonType: "string",
26     minLength: 4,
27     maxLength: 30,
28     pattern: "[a-zA-Z]+",
29     description: "must be a string and is not required",
```

```

29     },
30
31     lastName: {
32         bsonType: "string",
33         minLength: 4,
34         maxLength: 30,
35         description: "must be a string and is not required",
36     },
37
38     isAdmin: {
39         bsonType: "bool",
40         description: "must be a boolean and is not required",
41     }

```

Listing 1: Structura colecției utilizatori

```

1  bsonType: "object",
2  required: ["_id", "name", "category", "price"],
3  properties: {
4      _id: {
5          bsonType: "int",
6          minimum: 1,
7          description: "Must be an integer greater than 0 and is required",
8      },
9
10     name: {
11         bsonType: "string",
12         minLength: 3,
13         description: "Must be a string and is required",
14     },
15
16     price: {
17         bsonType: ["double", "int"],
18         minimum: 0,
19         description: "Must be a double and is required",
20     },
21
22     cateogry: {
23         enum: ["Laptop, Tablete & Telefoane", "Bacanie", "Fashion"],
24         description: "Can only be one of the enum values",
25     },
26
27     color: {
28         bsonType: "string",
29         minLength: 3,
30         description: "Must be a string",
31     },
32
33     diagonal_display: {
34         bsonType: ["int", "double"],
35         description: "Can only be one of the enum values",
36     },
37
38     hard_disk_type: {
39         enum: ["SSD", "HDD", "SSD+HDD"],
40         description: "Can only be one of the enum values",
41     },
42
43     hard_disk_memory_SSD: {
44         enum: [128, 256, 512, 1024, 2048, 4096],
45         description: "Can only be one of the enum values",

```

```

46     },
47     hard_disk_memory_HDD: {
48         enum: [128, 256, 512, 1024, 2048, 4096],
49         description: "Can only be one of the enum values",
50     },
51
52     storage_capacity: {
53         enum: [32, 64, 128, 256, 512],
54         description: "Can only be one of the enum values",
55     },
56
57     sim_slot: {
58         enum: ["Dual SIM", "Single SIM", "Hybrid SIM", "Triple SIM"],
59         description: "Can only be one of the enum values",
60     },
61
62     sim_type: {
63         enum: ["Nano SIM", "Micro SIM", "SIM"],
64         description: "Can only be one of the enum values",
65     },
66
67     number_of_cameras: {
68         bsonType: "int",
69         description: "Must be an int and is not required",
70     },
71
72     principal_camera_resolution: {
73         bsonType: "int",
74         description: "Must be an int and is not required",
75     },
76
77     technology: {
78         enum: ["3G", "4G", "5G"],
79         description: "Can only be one of the enum values",
80     },
81
82     pack_type: {
83         enum: ["Doza", "Sticla"],
84         description: "Can only be one of the enum values",
85     },
86
87     sex: {
88         enum: ["M", "F", "Unisex"],
89         description: "Can only be one of the enum values",
90     },
91
92     cloth_size: {
93         enum: ["XS", "S", "M", "L", "XL"],
94         description: "Can only be one of the enum values",
95     },
96
97     shoe_size: {
98         bsonType: "array",
99         items: {
100             enum: [ 35, 35.5, 36, 36.5, 37, 37.5, 38, 38.5, 39, 39.5, 40, 40.5,
101                    41, 41.5, 42, 42.5, 43, 43.5, 44, 45, 46 ],
102             description: "Can only be one of the enum values",
103         },
104     },
105 }

```

Listing 2: Structura colecției produse

```

1  bsonType: "object",
2  required: ["user", "products", "order_date", "status"],
3  properties: {
4    user: {
5      bsonType: "object",
6      description: "Must be an object and is required",
7    },
8
9    products: {
10     bsonType: "array",
11     required: ["product", "quantity"],
12     properties: {
13       product: {
14         bsonType: "object",
15         description: "Must be an object and is required",
16       },
17
18       quantity: {
19         bsonType: "int",
20         minimum: 0,
21         description: "Must be a integer and is required",
22       },
23     },
24   },
25
26   order_date: {
27     bsonType: "date",
28     description: "Must be a date and is required",
29   },
30
31   status: {
32     enum: ["Pending", "In progress", "Done"],
33     description: "Can only be one of the enum values and is required",
34   }

```

Listing 3: Structura colecției comenzi

## 3 Descrierea funcționalităților și a operațiilor

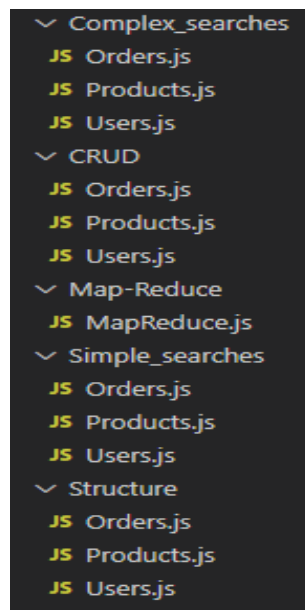


Figure 1: Scripturile proiectului

### 3.1 Structure

În directorul **Structure** sunt scripturile de creare a colecțiilor, împreună cu validările pe proprietăți, care au fost prezentate și mai sus.

### 3.2 CRUD

În directorul **CRUD** sunt scripturile pentru inserarea datelor de testare pentru cele 3 colecții. Ne-am folosit de multitudinea de metode, pe care le oferă MongoDB, pentru a insera în colecții, inserând unul câte unul, mai mulți deodată și folosindu-ne și de bulkWrite. Pe lângă inserare, s-au folosit și comenzi pentru actualizare și stergere a datelor.

### 3.3 Căutări simple

În directorul **Simple\_searches** s-au folosit metode simple de căutare a documentelor, sub-documentelor și elementelor din vector prin diverse opțiuni. Înainte de fiecare căutare s-a pus un comentariu legat de scopul acelei căutări.

### 3.4 Căutări complexe

În directorul **Complex\_searches** s-au realizat căutări mai complexe, folosind metoda *aggregate* și conceptul de pipeline pentru înlanțuirea operatorilor. La fel ca și la căutările simple, înainte de fiecare căutare s-a pus un comentariu legat de scopul căutării.

### 3.5 Map-Reduce

În directorul **Map-Reduce** sunt operațiile de acest tip. Au fost create 3 operații, care sunt descrise la început de operație, printr-un comentariu.