Assignment II

1. Create an index having fields of the following types. Disable dynamic mapping for this index.

```
PUT college
        {
         "mappings":{
          "_doc":{
           "dynamic":false,
           "properties":{
            "name":{
             "type":"text"
            },
            "Contact":{
             "properties":{
              "phone_num":{
               "type":"keyword"
              },
              "website":{
               "type":"keyword"
              }
             }
            },

             "established_date":{
             "type":"date",
             "format":"MM-dd-yyyy || yyyy/mm/dd"

            },
           "total_land_area":{
             "type":"double"
            },

            "total_students":{
             "type":"integer"
            },
```

```
        "teachers":{
          "type":"nested",
          "properties":{
            "name":{
              "type":"text"
            },
            "subjects_taught":{
              "type":"text"
            }
          }

        },
        "class_requirements_fulfilled":{
          "type":"boolean"
        },
        "fees":{
          "type":"long"
        },
        "classroom_area":{
          "type":"integer_range"
        }
       }
      }
     }
    }
```

2. Insert two valid documents, that is, with fields which match the types mentioned in (1)

```
PUT college/_doc/1
{
 "name":"Dwit",
 "Location":{
  "lat":51.2,
  "lon":47.5
 },
 "Contact":{
  "phone_num":"4455665",
  "website":"www.deerwalk.edu.np"
 },
```

```
 "established_date":"02-03-2014",
 "total_land_area":512.56 ,
 "total_students":250,
 "teachers":{
  "name":"Ritu Raj Lamsal",
  "subjects_taught":["Digital Logic","Computer Networks"]
 },
 "class_requirements_fulfilled":true,
 "fees":1200000,
 "classroom_area":{
  "gt":80,
  "lt":90
 }
 }

PUT college/_doc/2
{
 "name":"Texas",
 "Location":{
  "lat":50.1,
  "lon":47
 },
 "Contact":{
  "phone_num":"4409876",
  "website":"www.texas.edu.np"
 },
 "established_date":"2019/03/02",
 "total_land_area":417.5,
 "total_students":400,
 "teachers":{
  "name":"Subash Sharma",
  "subjects_taught":["Artificial Intelligence"]
 },
 "class_requirements_fulfilled":"true",
 "fees":800000,
 "classroom_area":{
  "gt":70,
  "lt":80
 }
  }
```

3. Try inserting an invalid document to see the exception thrown.

```
PUT college/_doc/3
 {
  "name":"Brilliant College",
  "Location":{
   "lat":30.2,
   "lon":28.9
  },
  "Contact":{
   "phone_num":4414573,
   "website":"www.brilliant.edu.np"

  },
  "established_date":"2019-02-01",
  "total_land_area":546.3,
  "total_students":"890",
  "teachers":{
   "name":"Sweekriti Gautam",
   "subjects_taught":["Information Retrieval"]
  },
  "class_requirements_fulfilled":true,
  "fees":"asgcb",
  "classroom_area":{
   "gt":"70",
   "lt":"77"
  }

 }
```

4. Use curl command along with _bulk API to insert the documents available in the file provided in mail (name: accounts.json) into accounts index.

```
curl -H "Content-Type: application/json" -XPOST "http://localhost:9200/accounts/_doc/_bulk?pretty" --data-binary @accounts.json
```

5. Perform queries using Request URI to find the following:

all documents

http://localhost:9200/accounts/_doc/_search?q=*
GET /accounts/_doc/_search?q=*

age greater than equal to 30 and less than equal to 70
http://localhost:9200/accounts/_doc/_search?q=age:[30%20TO%2070]
GET /accounts/_doc/_search?q=age:[30 TO 70]

females with age less than equals 25
http://localhost:9200/accounts/_doc/_search?q=gender:F%20AND%20age:[*%20TO%2025]
GET /accounts/_doc/_search?q=gender:F AND age:[* TO 25]

males belonging to ME state
http://localhost:9200/accounts/_doc/_search?q=gender:M%20and%20state=ME
GET /accounts/_doc/_search?q=gender:M AND state:ME

6. Perform following _update_by_query operations on accounts:

1. **Add a new field expense_list whose value is empty array [ ] for all documents.**

```
POST /accounts/_update_by_query?conflicts=proceed
{
 "query":{
  "match_all":{}
 },
 "script": {
  "source": "ctx._source.expense_list = []",
  "lang": "painless"
 }

    }
```

2. **Add a value 'student_loan' into the expense_list array for members having age greater than equals 10 and less than equals 25**

```
POST /accounts/_update_by_query?conflicts=proceed
{
 "query":{

  "match_all":{}
   },
 "script":{
```

```
    "lang":"painless",

   "source":"""
  String val= "student_loan";
   if(ctx._source.age>=10 && ctx._source.age<=25){
    ctx._source.expense_list.add(val)
   }

   """
  }
}
```

3. **Add two values ‘car_loan’ and ‘house_loan’ into expense_list array for members having age greater than 25 and less than equals 50**

```
POST /accounts/_update_by_query?conflicts=proceed
{
 "query":{
  "match_all":{}
 },

 "script":{
  "lang":"painless",

  "source":"""
  String val1="car_loan";
  String val2="house_loan";
  if(ctx._source.age>=25 && ctx._source.age<=50){
   ctx._source.expense_list.add(val1);
   ctx._source.expense_list.add(val2)
  }
  """
 }
}
```

4. **Add a value ‘recreation’ for members having balance greater than equals 40000.**

```
POST /accounts/_update_by_query?conflicts=proceed
{
 "query":{
  "match_all":{}
 },
```

```
  "script":{
    "lang":"painless",

    "source":"""
    String val="recreation";

    if(ctx._source.balance>=40000){
      ctx._source.expense_list.add(val)

    }
    """
  }
}
```

5. Decrease the balance by 2000 of members of state PA.

```
POST /accounts/_update_by_query?conflicts=proceed
{
  "query":{
    "match_all":{}
  },
  "script":{
    "lang":"painless",
    "source":"""
    if(ctx._source.state=="PA"){
      ctx._source.balance-=params.val
    }
    """,
    "params":{
      "val":2000
    }
  }
}
```

7. Perform _delete_by_query to delete all records belonging to state: KY.

8. Perform following queries using Request Body with any values you want to:

- **Term query**

```
GET accounts/_search
{
  "query":{
```

```
   "term":{
     "gender":{
       "value":"F"
     }
  }
}
}
```

- **Range query**

```
GET accounts/_search
{
  "query":{
  "range": {
    "age": {
      "gte": 10,
      "lte": 50
    }
  }
}

}
```

- **Prefix query**

```
GET accounts/_search
{"query":{
  "prefix":{
    "firstname":"Ju"

  }
}

}
```

- **Wildcard Query**

```
GET product/_search
{"query":{
  "wildcard":{
    "firstname":"R?b"

  }
}

}
```

9.

A. Create an index college having following fields:

- **batch (integer type): example values, 2017, 2018**
- **students (nested type, i.e. array of inner objects): each inner object can have two properties id and name.**

```
PUT college1
{
 "mappings": {
  "_doc":{
   "properties":{
   "batch":{
    "type":"integer"
   },
   "students":{
   "type":"nested",
    "properties":{
     "id":{
      "type":"integer"

     },
     "name":{
      "type":"text"
     }
    }
   }

   }
  }
 }
}
```

B. Insert a document with certain id (example, 1), your batch (example, 2017), and an array of 3 students in index college.

```
PUT college1/_doc/1
{
 "batch":2019,
 "students":{
```

```
  "student_id":[547, 544, 521],
  "name":["Shreeya Pandey","Sweekriti Gautam","Rachana Banjade"]
 }
}
```

C. Create an index workshop having following fields

- **students_id**
- **workshop_about**
- **enrolled_year**

```
PUT workshop
{
 "mappings":{
  "_doc":{
   "properties":{
    "student_id":{
     "type":"integer"
    },
    "workshop_about":{
     "type":"text"
    },
    "enrolled_year":{
     "type":"date",
     "format":"yyyy"
    }
   }
  }
 }
}
```

D. Bulk insert 5 documents in index workshop.

```
POST workshop/_doc/_bulk
{"index":{"_id":1}}
{"student_id":547,"workshop_about":"Elastic Search","enrolled_year":"2019" }
{"index":{"_id":2}}
{"student_id":544,"workshop_about":"Machine Learning","enrolled_year":"2019"}
{"index":{"_id":3}}
{"student_id":321,"workshop_about":"Android","enrolled_year":"2018"}
{"index":{"_id":4}}
```

{"student_id":420,"workshop_about":"Android","enrolled_year":"2018" }
{"index":{"_id":5}}
{"student_id":530,"workshop_about":"Elastic Search","enrolled_year":"2019" }

E. Using terms query, find the students of your batch enrolled in any workshop.

```
GET workshop/_search
{
  "query": {
    "terms": {
      "student_id": {
        "index": "college1",
        "type": "_doc",
        "id": 1,
        "path": "student.id"
      }
    }
  }
}
```