

Lab 6

1-) Write a function 'isPrime' that takes one parameter and returns true if the number is a prime number otherwise returns false.

2-) Change 'isPrime' that takes in a single number parameter and returns a new promise.

Using setTimeout and after 500 milliseconds, the promise will either resolve or reject.

if the input is prime number, the promise resolves with {prime: true}.

if the input is not a prime number, it rejects with {prime: false}.

3-) Create a NoSQL design model for an application to manage a library, taking into consideration the following requirements:

- Books have an ISBN number and are categorized by author and tagged by keywords to facilitate search
- Books can be borrowed by students, so the librarian will be able to check all borrowed books and their return date so he may contact students who are late returning their books.

4-) Create a new DB called lab6.

Develop an Express Application for the following operations:

Find/FindOne/Add/Delete

Try to use appropriate HTTP verbs for an entity called: lectures, document structure is:

```
lectures = [{_id, course, lecture}]
```

- Test your API using Postman
- Implement a route (GET /search/:q) to search if the lecture name contains the passed :q parameter.
- Send the results as JSON.

5-) Considering the following [restaurants](#) collection that has information about all restaurants in the USA. Import the data into a local/cloud DB server.

```
{
  "address": { "building": "1007",
               "coord": [ -73.856077, 40.848447 ],
               "street": "Morris Park Ave",
               "zipcode": "10462" },
  "district": "Bronx",
  "cuisine": "Bakery",
  "grades": [ {"date": {"$date": 1393804800000}, "grade": "A", "score": 2},
              {"date": {"$date": 1378857600000}, "grade": "A", "score": 6},
```

```
        {"date": {"$date": 1358985600000}, "grade": "A", "score": 10},
        {"date": {"$date": 1322006400000}, "grade": "A", "score": 9},
        {"date": {"$date": 1299715200000}, "grade": "B", "score": 14}],
    "name": "Morris Park Bake Shop",
    "restaurant_id": "30075445"
}
```

1. Write a MongoDB query to display all the documents in the collection restaurants.
2. Write a MongoDB query to display the fields restaurant_id, name, district and cuisine for all the documents in the collection restaurant.
3. Write a MongoDB query to display the fields restaurant_id, name, district and cuisine, but exclude the field _id for all the documents in the collection restaurant.
4. Write a MongoDB query to display all the restaurant which is in the district "Bronx".
5. Write a MongoDB query to display the first 5 restaurant which is in the district "Bronx".