**HOMEWORK 3**

**PREREQUISITES:**

1. Install nodemon and express dependencies (if not installed already)

2. Install mongoDB and mongoose package

3. Makesure mongo is running (on port 27017) by setting DB path. Run Mongo and Mongod commands

3. Create a repo for your homework tasks on **Github** (https://github.com/) or **git.epam.com**.

4. Provide your mentor with the link to the repo and add read access permissions.

5. Create package.json by running the following commands npm init or npm init -y.

**Task 1:**

Implement an express server that runs on 3000 port and configure the below:

1. Enable the server to parse incoming requests with JSON payloads
2. Enable the server to serve static files (use built-in middleware)
3. Enable the server to parse incoming requests with URL encoded
4. Change the server settings to accept maximum request body size to 10 MB (this applies for JSON and URL encoded)
5. Enable router config on express
6. Connect to mongo server with database “**booksdb**” which is running on port 27017
   1. Enable useNewUrlParser
   2. Disable AutoIndex
   3. Set poolsize to 10
   4. Eable KeepAlive & add keepAliveInitialDelay
   5. Listen or below events
      1. Connecting
      2. Connected
      3. Disconnecting
      4. close

**Task 2:**

Modify the above server implementation to meet below needs:

We are trying to build APIs for a bookstore. Each book should have valid properties as mentioned below

* **create a schema called book with below properties and turn it to a model with name “books”**

Book name – string, required

ISBN – International Standard Book Number – string, required

Publisher Name – string, required, **index required**

Author Name – string, required, **index required**

Author ID – number, required

Price – number, required, min-0, max – 10000, **index required**

Published year – number, required

Reviewers – Array, required

Number of pages: number, required

Country: string, required

Rating: Number, required

Create a sample books.json file and import this data manually (using DB command) into mongoDB

Implement the below endpoints – all must have proper HTTP error codes, exception handling

**/books** – this is a GET call to retrieve all books information

**/books/:id** – this should retrieve a single book info from the book database (**id** should refer **ISBN in schema)**

**/books?year=2019 –** this should retrieve all books published in 2019

**/books?pricestart=<number>&priceend=<number> -** this should retrieve all books between the specified price

***Task 3:***

Implement Mongodump and Mongoexport against books collectionfrom node (hint: try using node.js child\_process way)

**Task 4:**

Modify your server implementation to POST new books to MongoDB

1. Use custom middleware with express JSON parser to add a new book from POSTMAN
2. Perform schema validation to check if the provided data is of correct format – use 3rd party schema validators like JOI or AJV etc) – two books can’t share same ISBN & title
3. Add new book data in mongoDB
4. Update all books author to “Adam Freeman” whose ISBN number is 9781484249789 (use appropriate HTTP verb)
5. Update author name “John Papa” to “John” for his 3rd and 4th books based on published year (hint: try using skip, sort and limit)
6. /books/:ISBN – this should delete the book from mongoDB (if ISBN is not valid, print appropriate custom error message)
7. Update the schema to add a nested doc about author address – Array of 2 docs
   1. Permanent address -
      1. House Number – string, required
      2. Street – string, required
      3. Area – string, required
      4. Pin code – number required, min – 6, max-6
   2. Temporary address
      1. House number – string, required
      2. Street – string, required
      3. Area – string, required
      4. Pin code – number required, min – 6, max-6
8. /books/pincode/<number> - this should print all authors from a specific pincode
9. Implement a custom schema validator for Pincode which should have DDDDDD format
10. Implement an aggregation pipeline to accomplish below
    1. Print all books of all authors where published year between 2015-2020 and price range between 500-100 in price descending order

{

“Book1” :

}

* 1. For each author – print all books authored by him and each book should have reviewers as an array