**Objective**: To explore more on developing db-driven Node/Express app

**Description:** we are going to develop a new Node/Express app program extracting data from databases. We will design a simple app and use MongoDB Atlas based on the given instructions in Q1 and Q2.

**Question 1)** You are asked to develop a sample Node/Express app which interact with MongoDB database in Atlas

**Step1:** Create new Node/Express app (in new folder named “Asn4-mongo-yourname”) as follow:

* Add the following dependencies to your project

|  |
| --- |
| * "dependencies": { * "express-validator": "^6.14.2", |
| o "express": "^4.18.1", o "mongoose": "^6.7.2" o } |

* Make sure to have the following project structure:

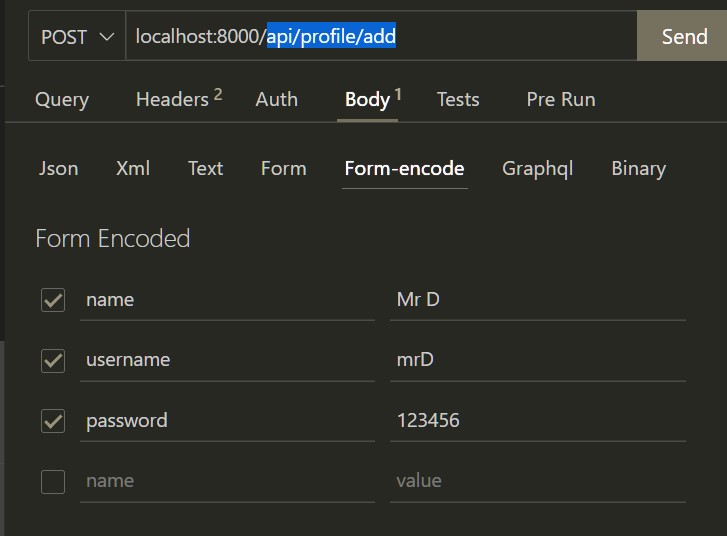
|  |  |
| --- | --- |
| Create a “config” folder which contains  “settings.js” (will be used for database connection  parameters for MongoDB.)    Create “models” folder which contains “Person.js” (will be used to create employee schema and  model.)    Create “routes/api” folder which contains all the routes related to solution.    Create “index.js” in the project root.    Make sure to have proper project settings and dependencies in “package.json” |  |

**Step2:** Using the attach file (codenippet-mongo.txt), copy/paste the related code to “config/settings.js”, “model/Person.js”, “routes/api/profile.js” and “index.js”.

* *Note: Create a new MongoDB database in your local machine and update the “url” in database.js accordingly*

**Step 3**: Run the application and test it using the following routes:

* A) Open Postman or thunderclient, choose POST method and use the url [http://localhost:8000/api/profile/ad](http://localhost:8000/api/profile/add)[d](http://localhost:8000/api/employees)



o In the Body, choose “x-www-form-urlencoded” and add three pair of key values as follow o Click on send. What is the output? o Check the console in VScode, what is the output? o Check the Compass, is the new record added to “myPeople” collection in Database?

* B) Open browser and enter http://localhost:8000/api/profile/get [.](http://localhost:8000/api/employees) What is the output?
* C) Using any username of any employee records in database, run the following query in the browser:

o localhost:8000/api/profile/get/mrA [.](http://localhost:8000/api/employees/618cf962f36b27c5379212b7) What is the output?

**Step 4**: Base on your observation, answer the following questions:

1. How does the Step3:A, B,C work? Explain the work flow, route, and the way the query executed.
2. What is the role of:
   1. module.exports = Person = mongoose.model('myPerson', PersonSchema)

1. Using the idea of Step3:C, try to update one of the record in the myPeople collection .Find related route ☺ in the code and explain how it works.
2. Using the idea of Step3:C, try to delete one of the record in the employee table. Find related route ☺ in the code and explain how it works.

*Note:*

* 1. *It is important to explain how this app works in your video demonstration*

**Question 2)** You are asked to redesign Question 1 by using the given dataset of Assignment2.

* Step 1: Create a new MongoDB database in Atlas based on the given dataset of Assignment2.
* Step 2: Redesign the route/code in Question1 and set it up to work with book-data instead of employee data.
  + - You may need to change the “model” and routes.
    - Your app should have the following features and Demonstrate how app works using Postman/ThunderClient. (similar to Q1):
      * Show all book-info
      * Show a specific book (based on the \_id or isbn)
      * Insert a new book
      * Delete an existing book (based on the \_id or isbn)
      * Update book\_title & pageCount of an existing book (based on the \_id or isbn)

o Using Handlebar and Form complete the followings (*hint:use ideas from Assingment2*):

* + - * Show all book-info
      * Insert a new book
* *Note:*

o *It is important to explain how this app works in your video demonstration*