MONGODB CONNECTION

Experiment_No_04 > project-1-basic-api > controllers > taskController.js > ...

```javascript
1    const Task = require('../models/Task');
2
3
4    // @route    GET
5    exports.getTasks = async (req, res) => {
6      try {
7        const tasks = await Task.find();
8        res.status(200).json({ success: true, count: tasks.length, data: tasks });
9      } catch (err) {
10       res.status(500).json({ success: false, error: 'Server Error' });
11     }
12   };
13
14   // @desc     Create a task
15   // @route    POST
16   exports.createTask = async (req, res) => {
17     try {
18       const task = await Task.create(req.body);
19       res.status(201).json({ success: true, data: task });
20     } catch (err) {
21       res.status(400).json({ success: false, error: err.message });
22     }
23   };
24
25   // @desc     Update a task
26   // @route    PUT
27   exports.updateTask = async (req, res) => {
28     try {
29       const task = await Task.findByIdAndUpdate(req.params.id, req.body, {
30         new: true,
31         runValidators: true,
32       });
```

Fig 1.0

```javascript
25   // @desc     Update a task
26   // @route    PUT
27   exports.updateTask = async (req, res) => {
28     try {
29       const task = await Task.findByIdAndUpdate(req.params.id, req.body, {
30         new: true,
31         runValidators: true,
32       });
33       if (!task) {
34         return res.status(404).json({ success: false, error: 'Task not found' });
35       }
36       res.status(200).json({ success: true, data: task });
37     } catch (err) {
38       res.status(500).json({ success: false, error: 'Server Error' });
39     }
40   };
41
42   // @desc     Delete a task
43   // @route    DELETE
44   exports.deleteTask = async (req, res) => {
45     try {
46       const task = await Task.findByIdAndDelete(req.params.id);
47       if (!task) {
48         return res.status(404).json({ success: false, error: 'Task not found' });
49       }
50       res.status(200).json({ success: true, data: {} });
51     } catch (err) {
52       res.status(500).json({ success: false, error: 'Server Error' });
53     }
54   };
```

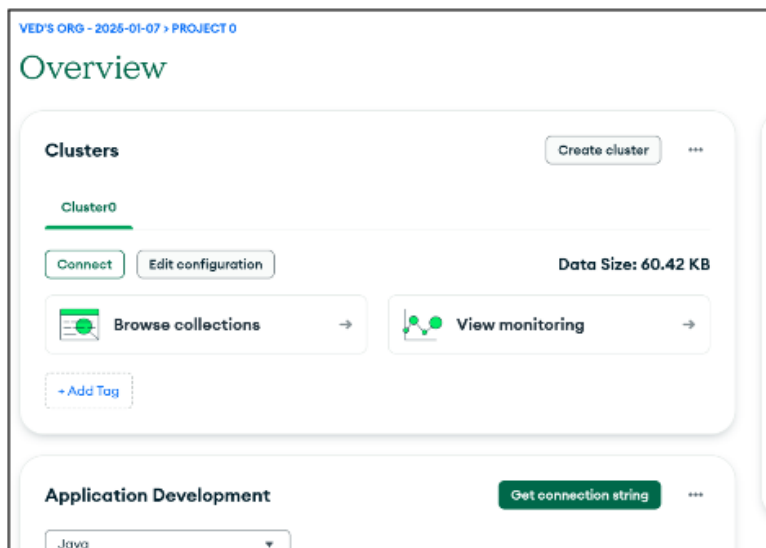Fig. 1.1

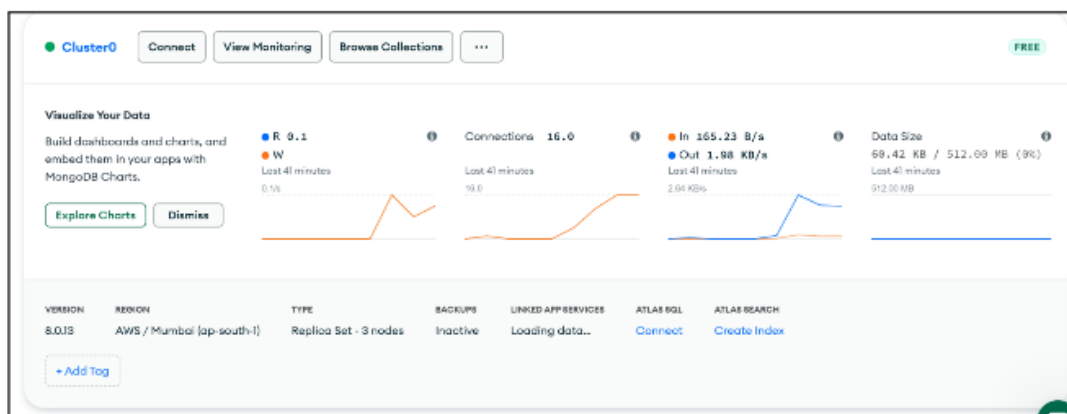Fig2.0

Fig 2.1

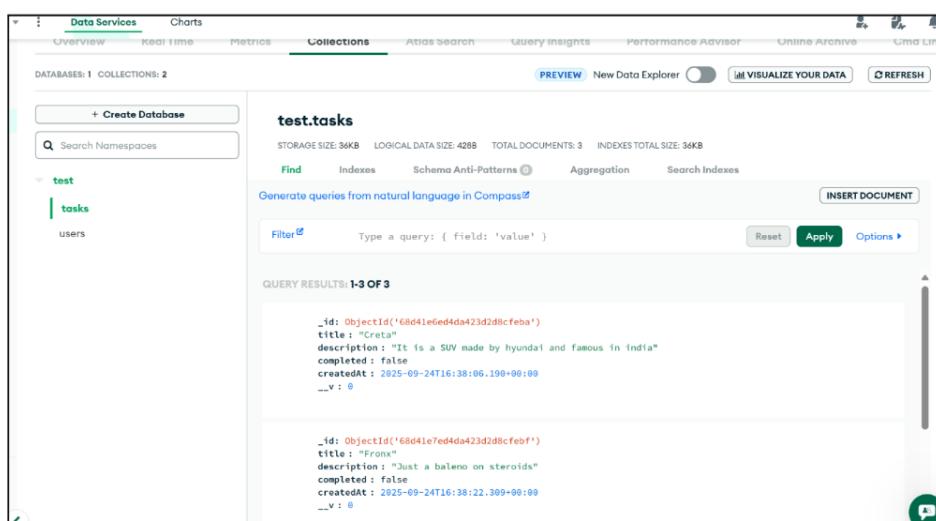Fig. 2.2

Fig 2.3

Login

Email

Password

Login

New user? Register here

Fig3.0

Car Information

Logout

| Car Name | Description | Add Car |

User role user is not authorized

Name	Description	Completed	Actions
Creta	It is a SUV made by hyunda	☐	Delete
Fronx	Just a baleno on steroids	☐	Delete
Wagonr	A car that everyone uses fo	☐	Delete

Fig 3.1