**Experiment Number : 10 Date:06/04/2025**

# **Experiment 10: Create a Web Application with MongoDB and Node.js**

## **PRE LAB EXERCISE**

### **Objective:**

* Develop a web application using **Node.js** and **MongoDB**.
* Implement **CRUD (Create, Read, Update, Delete)** operations.
* Use **Express.js** for handling server requests.
* Create a **RESTful API** for managing data.

### **QUESTIONS:**

1. What is MongoDB, and how is it different from relational databases?

MongoDB is a NoSQL database that stores data in flexible, JSON-like documents instead of tables with rows and columns. It allows dynamic schemas, making it more scalable and suitable for unstructured data.

1. How does **Express.js** help in building a web server?

Express.js is a lightweight framework for Node.js that simplifies server-side development by providing easy routing, middleware support, and handling HTTP requests and responses.

1. What is **Mongoose**, and why is it used?

Mongoose is an Object Data Modeling (ODM) library for MongoDB in Node.js. It provides schema validation, query building, and middleware support, making database interactions easier.

1. What are the HTTP methods used in a RESTful API?

 GET – Retrieve data

 POST – Create new data

 PUT – Update existing data

 DELETE – Remove data

 PATCH – Partially update data

1. How does **MongoDB store data**, and what is a **document** in MongoDB?

MongoDB stores data in collections (similar to tables in SQL), and each entry is a **document**. A document is a JSON-like object containing key-value pairs, which can have nested structures and arrays.

## **IN LAB EXERCISE**

**Create a Web Application with MongoDB and Node.js**

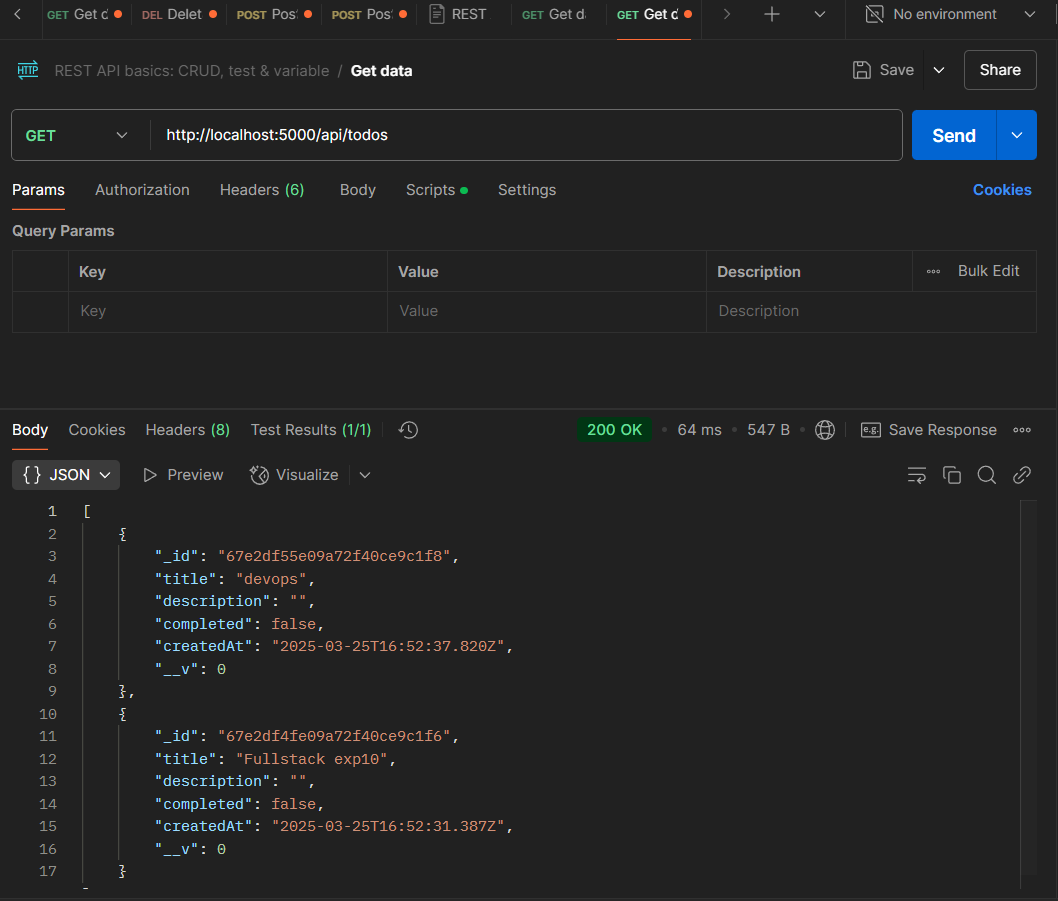
**A screenshot of a computer program

AI-generated content may be incorrect.**

**GET(Display the todo)**

A screenshot of a computer

AI-generated content may be incorrect.



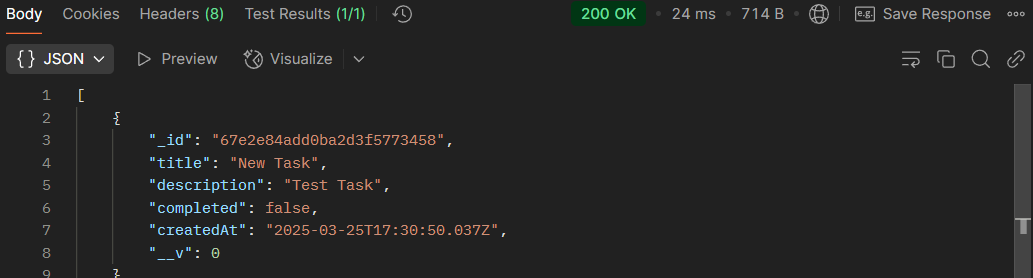
**A screenshot of a computer

AI-generated content may be incorrect.**

**Create a Todo(POST):**

A screenshot of a computer

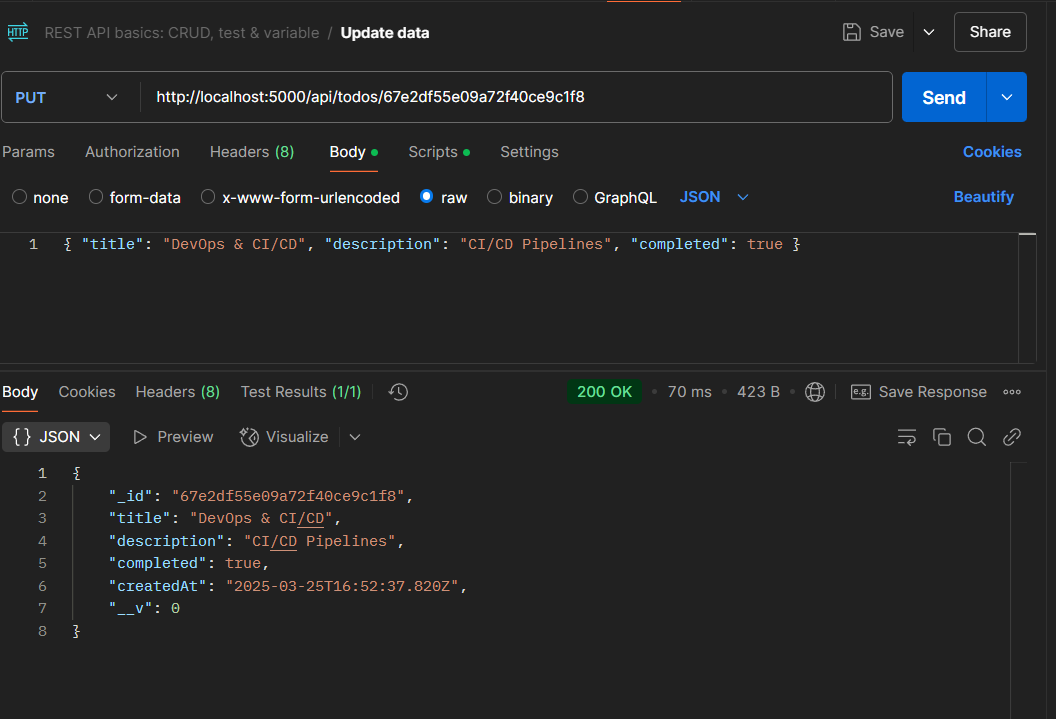
AI-generated content may be incorrect.



A screenshot of a computer

AI-generated content may be incorrect.

**Update a Todo(PUT):**



A screenshot of a computer

AI-generated content may be incorrect.

**Delete a Todo(Delete):**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

## **POST LAB EXERCISE**

### **QUESTIONS:**

1. What is the purpose of **Mongoose** in a Node.js application?

Mongoose helps manage MongoDB in a Node.js application by providing a structured way to define schemas, validate data, and simplify database operations.

1. How do you create a new document in MongoDB using Mongoose?

1.const User = new UserModel({ username: "John", password: "hashedpassword" });

User.save();

2. using async/await:

await UserModel.create({ username: "John", password: "hashedpassword" });

1. Explain the significance of **method-override** in this application.

method-override allows browsers (which support only GET and POST) to use other HTTP methods like PUT and DELETE, enabling full RESTful API functionality.

1. What are the different HTTP methods used in this CRUD application?

 **GET** – Retrieve data

 **POST** – Add new data

 **PUT** – Update existing data

 **DELETE** – Remove data

1. How can you improve security in a MongoDB-based web application?

 Use **bcrypt** for password hashing

 Enable **authentication and access control** in MongoDB

 Use **environment variables** for sensitive data

 Validate and sanitize user input

 Implement **rate limiting** to prevent attacks

**ASSESSMENT PATTERN.**

|  |  |  |
| --- | --- | --- |
| **Description** | **Max Marks** | **Marks Awarded** |
| Pre Lab Exercise | **5** |  |
| In Lab Exercise | **10** |  |
| Post Lab Exercise | **5** |  |
| Viva | **10** |  |
| **Total** | **30** |  |
| **Faculty Signature** | |  |