

 <b>Marwadi University</b> Marwadi Chandarana Group		<b>Marwadi University</b> <b>Faculty of Technology</b> <b>Department of Information and Communication Technology</b>	
<b>Subject: Advanced Web Technology</b>	<b>Aim:</b> Develop a web application which involves database operations using NodeJS		
<b>Experiment :- 2</b>	<b>Date:- 29-03-2025</b>	<b>Enrollment No:- 92200133003</b>	

#### CODE :-

##### index.js

```
const express = require("express");
const app = express();
const port = 3001;
const cors = require("cors");
app.use(
  cors({
    origin: "http://localhost:5173",
  })
);
//middleware
app.use(express.json());
app.use(express.urlencoded({ extended: true }));

const dbConfig = require("./src/config/db.config");

const mongoose = require("mongoose");

mongoose.Promise = global.Promise;

mongoose
  .connect(dbConfig.url, {
    useNewUrlParser: true,
    useUnifiedTopology: true,
  })
  .then(() => {
    console.log("connection done with database");
  })
  .catch((err) => {
    console.log("error in db connection ", err);
    process.exit();
  });

const blogRouter = require("./src/blog/blog.routes");

app.use("/api/blog", blogRouter);

app.get("/", (req, res) => res.send("Hello World!"));
app.listen(port, () => console.log(`Example app listening on port ${port}!`));
```

##### blog.controller.js

```
const blogService = require("./blog.service");

exports.create = async (req, res) => {
  const { title, description, author } = req.body;
  if (!title || !description || !author) {
    return res.status(400).json({ message: "Fields are empty" });
  }

  try {
    const blog = await blogService.createBlog(title, description, author);
    res.status(201).json({ message: "Blog created successfully", blog });
  } catch (err) {
    console.error("Error creating blog:", err);
    res.status(500).json({ error: "Internal server error" });
  }
};
```

 <b>Marwadi University</b> Marwadi Chandarana Group		<b>Marwadi University</b> <b>Faculty of Technology</b> <b>Department of Information and Communication Technology</b>
<b>Subject: Advanced Web Technology</b>	<b>Aim:</b> Develop a web application which involves database operations using NodeJS	
<b>Experiment :- 2</b>	<b>Date:- 29-03-2025</b>	<b>Enrollment No:- 92200133003</b>

```

exports.findAll = async (req, res) => {
  try {
    const blogs = await blogService.findAllBlogs();
    res.send({ message: "Blogs data", blogs });
  } catch (err) {
    console.error("Error fetching blogs:", err);
    res.status(500).json({ error: "Internal server error" });
  }
};

exports.findAllByTitle = async (req, res) => {
  const title = req.query.title;
  try {
    const blogs = await blogService.findBlogsByTitle(title);
    res.send({ message: "Blogs data", blogs });
  } catch (err) {
    console.error("Error fetching blogs by title:", err);
    res.status(500).json({ error: "Internal server error" });
  }
};

exports.findOne = async (req, res) => {
  const id = req.params.id;
  try {
    const blog = await blogService.findBlogById(id);
    if (!blog) {
      return res.status(400).json({ message: "No blog found with id: " + id });
    }
    res.json({ message: "Data found", blog });
  } catch (err) {
    console.error("Error fetching blog by id:", err);
    res.status(500).json({ error: "Internal server error" });
  }
};

exports.update = async (req, res) => {
  const id = req.params.id;
  const updateData = req.body;
  try {
    const result = await blogService.updateBlog(id, updateData);
    if (!result) {
      return res.status(400).json({ message: "No blog found with id: " + id });
    }
    res.json({ message: "Blog updated successfully" });
  } catch (err) {
    console.error("Error updating blog:", err);
    res.status(500).json({ error: "Internal server error" });
  }
};

exports.delete = async (req, res) => {
  const id = req.params.id;
  try {
    const result = await blogService.deleteBlog(id);
    if (!result) {
      return res.status(400).json({ message: "No blog found with id: " + id });
    }
    res.json({ message: "Blog deleted successfully" });
  } catch (err) {
    console.error("Error deleting blog:", err);
    res.status(500).json({ error: "Internal server error" });
  }
};

```

 <b>Marwadi University</b> Marwadi Chandarana Group	NAAC <b>A+</b>	<b>Marwadi University</b> <b>Faculty of Technology</b> <b>Department of Information and Communication Technology</b>	
<b>Subject: Advanced Web Technology</b>	<b>Aim:</b> Develop a web application which involves database operations using NodeJS		
<b>Experiment :- 2</b>	<b>Date:- 29-03-2025</b>	<b>Enrollment No:- 92200133003</b>	

### blog.model.js

```
const mongoose = require("mongoose");
```

```
var schema = new mongoose.Schema(
{
  title: {
    type: String,
    required: true,
  },
  description: {
    type: String,
    required: true,
  },
  author: {
    type: String,
    required: true,
  },
},
{
  timestamps: true,
}
);
```

```
var blog = new mongoose.model("mublog", schema);
module.exports = blog;
```

### blog.routes.js

```
const blog = require("./blog.controller");
const express = require("express");
const router = express.Router();
```

```
router.post("/", blog.create);
```

```
router.get("/", blog.findAll);
```

```
router.get("/title/", blog.findAllByTitle);
```

```
router.get("/:id", blog.findOne);
```

```
router.put("/:id", blog.update);
```

```
router.delete("/:id", blog.delete);
```

```
module.exports = router;
```

### blog.service.js

```
const BlogModel = require("./blog.model");
```

```
exports.createBlog = async (title, description, author) => {
  const blog = new BlogModel({
    title,
    description,
    author,
  });
  return await blog.save();
};
```

```
exports.findAllBlogs = async () => {
  return await BlogModel.find();
};
```

 <b>Marwadi University</b> Marwadi Chandarana Group 	<b>Marwadi University</b> <b>Faculty of Technology</b> <b>Department of Information and Communication Technology</b>	
<b>Subject: Advanced Web Technology</b>	<b>Aim:</b> Develop a web application which involves database operations using NodeJS	
<b>Experiment :- 2</b>	<b>Date:- 29-03-2025</b>	<b>Enrollment No:- 92200133003</b>

```
exports.findBlogsByTitle = async (title) => {
  const condition = title
    ? { title: { $regex: new RegExp(title), $options: "i" } }
    : {};
  return await BlogModel.find(condition);
};

exports.findBlogById = async (id) => {
  return await BlogModel.findById(id);
};

exports.updateBlog = async (id, updateData) => {
  return await BlogModel.findByIdAndUpdate(id, updateData, {
    useFindAndModify: false,
  });
};

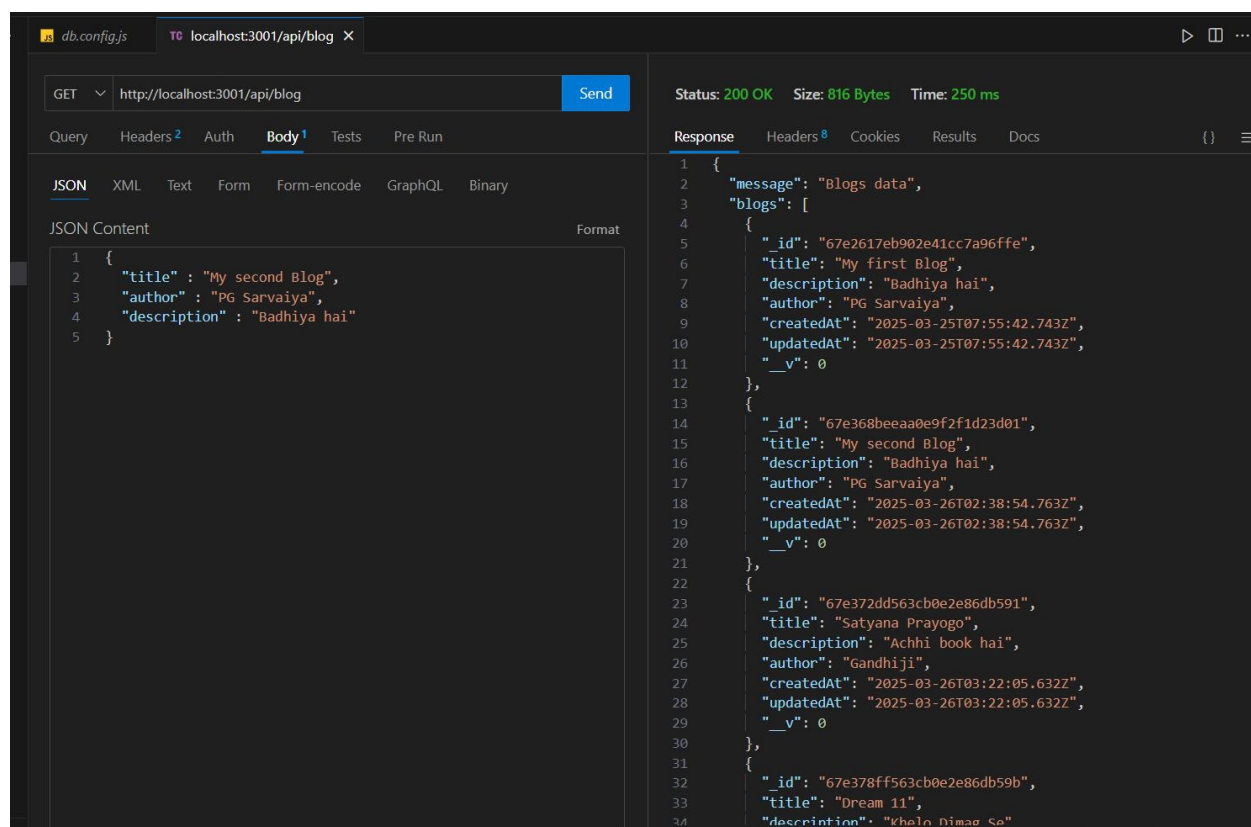
exports.deleteBlog = async (id) => {
  return await BlogModel.findByIdAndDelete(id);
};
```

### db.config.js

```
module.exports = {
  url: "mongodb://127.0.0.1:27017/mu_blog_db",
};
```

## OUTPUT :-

### Get



The screenshot shows a REST client interface with the following details:

- URL:** http://localhost:3001/api/blog
- Method:** GET
- Status:** 200 OK
- Size:** 816 Bytes
- Time:** 250 ms
- Response (JSON):**

```
{
  "message": "Blogs data",
  "blogs": [
    {
      "_id": "67e2617eb902e41cc7a96ffe",
      "title": "My first Blog",
      "description": "Badhiya hai",
      "author": "PG Sarvaiya",
      "createdAt": "2025-03-25T07:55:42.743Z",
      "updatedAt": "2025-03-25T07:55:42.743Z",
      "__v": 0
    },
    {
      "_id": "67e368beaaa0e9f2f1d23d01",
      "title": "My second Blog",
      "description": "Badhiya hai",
      "author": "PG Sarvaiya",
      "createdAt": "2025-03-26T02:38:54.763Z",
      "updatedAt": "2025-03-26T02:38:54.763Z",
      "__v": 0
    },
    {
      "_id": "67e372dd563cb0e2e86db591",
      "title": "Satyana Prayogo",
      "description": "Achhi book hai",
      "author": "Gandhiji",
      "createdAt": "2025-03-26T03:22:05.632Z",
      "updatedAt": "2025-03-26T03:22:05.632Z",
      "__v": 0
    },
    {
      "_id": "67e378ff563cb0e2e86db59b",
      "title": "Dream 11",
      "description": "Khalo Dimag So"
    }
  ]
}
```

 <b>Marwadi University</b> Marwadi Chandarana Group 	<b>Marwadi University</b> <b>Faculty of Technology</b> <b>Department of Information and Communication Technology</b>	
<b>Subject: Advanced Web Technology</b>	<b>Aim:</b> Develop a web application which involves database operations using NodeJS	
<b>Experiment :- 2</b>	<b>Date:- 29-03-2025</b>	<b>Enrollment No:- 92200133003</b>

## POST

POST

http://localhost:3001/api/blog

Send

Query

Headers<sup>2</sup>

Auth

Body<sup>1</sup>

Tests

Pre Run

JSON

XML

Text

Form

Form-encode

GraphQL

Binary

JSON Content

Format

```

1 {
2   "title": "ICT Department",
3   "author": "Chandrasinh Parmar",
4   "description": "Guranteed Placement"
5 }

```

Status: 201 Created Size: 258 Bytes Time: 21 ms

Response

Headers<sup>8</sup>

Cookies

Results

Docs

```

1 {
2   "message": "Blog created successfully",
3   "blog": {
4     "title": "ICT Department",
5     "description": "Guranteed Placement",
6     "author": "Chandrasinh Parmar",
7     "_id": "67e82ae88b816e0f149f2bb7",
8     "createdAt": "2025-03-29T17:16:24.946Z",
9     "updatedAt": "2025-03-29T17:16:24.946Z",
10    "__v": 0
11  }
12 }

```

## DELETE :

blog.routes.js

localhost:3001/api/blog X

DELETE

http://localhost:3001/api/blog/67e82ae88b816e0f149f2bb7

Send

Query

Headers<sup>2</sup>

Auth

Body<sup>1</sup>

Tests

Pre Run

JSON

XML

Text

Form

Form-encode

GraphQL

Binary

JSON Content

Format

```

1 {
2   "title": "ICT Department",
3   "author": "Chandrasinh Parmar",
4   "description": "Guranteed Placement"
5 }

```

Status: 200 OK Size: 39 Bytes Time: 13 ms

Response

Headers<sup>8</sup>

Cookies

Results

Docs

```

1 {
2   "message": "Blog deleted successfully"
3 }

```

## PUT:

blog.routes.js

localhost:3001/api/blog \* X

PUT

http://localhost:3001/api/blog/67e2617eb902e41cc7a96ffe

Send

Query

Headers<sup>2</sup>

Auth

Body<sup>1</sup>

Tests

Pre Run

JSON

XML

Text

Form

Form-encode

GraphQL

Binary

JSON Content

Format

```

1 {
2   "title": "ICT Department",
3   "author": "Chandrasinh Parmar",
4   "description": "Guranteed Placement"
5 }

```

Status: 200 OK Size: 816 Bytes Time: 10 ms

Response

Headers<sup>8</sup>

Cookies

Results

Docs

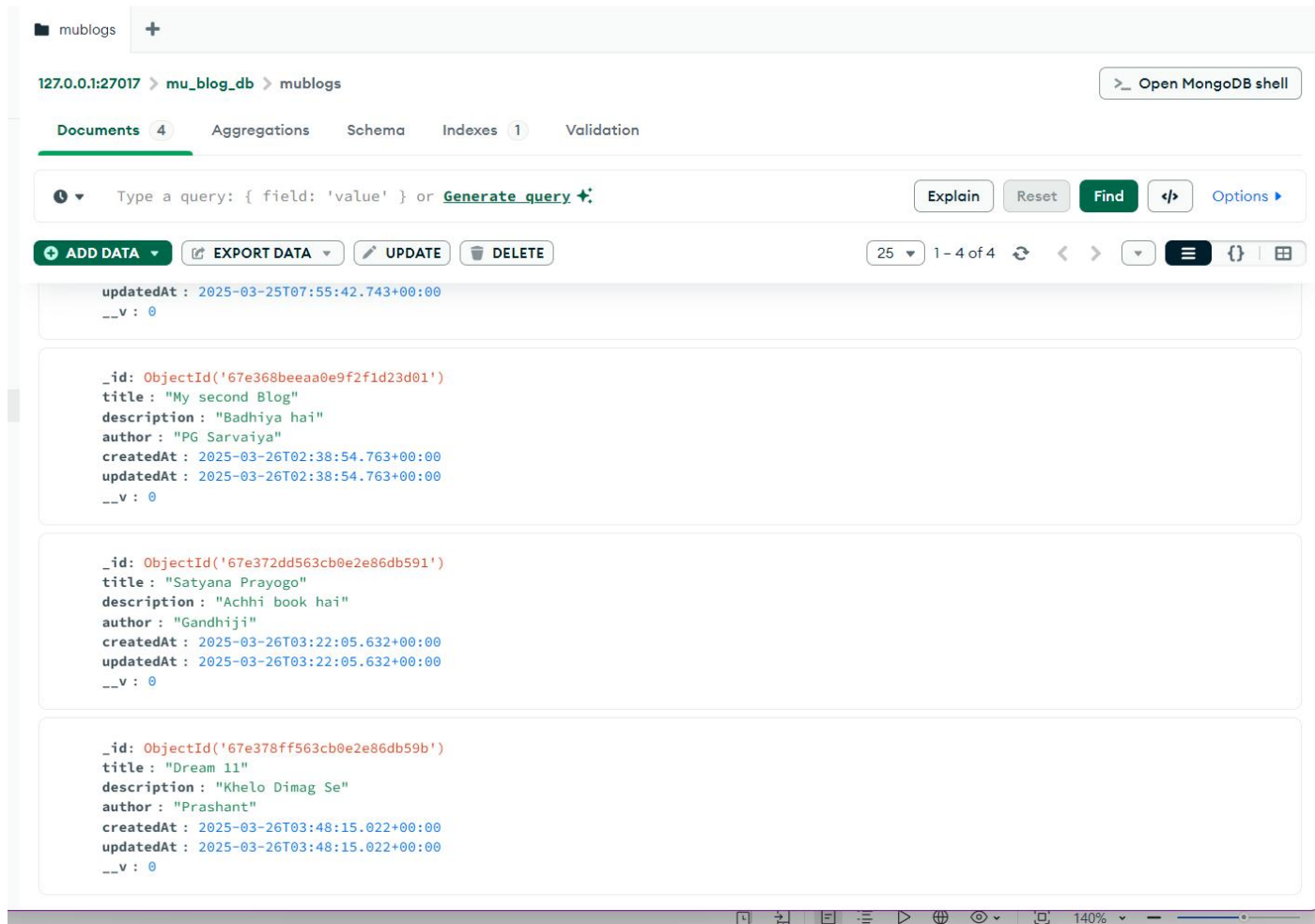
```

1 {
2   "message": "Blogs data",
3   "blogs": [
4     {
5       "_id": "67e2617eb902e41cc7a96ffe",
6       "title": "My first Blog",
7       "description": "Badhiya hai",
8       "author": "PG Sarvaiya",
9       "createdAt": "2025-03-25T07:55:42.743Z",
10      "updatedAt": "2025-03-25T07:55:42.743Z",
11      "__v": 0
12    },
13    {
14      "_id": "67e2617eb902e41cc7a96ffe",
15      "title": "My first Blog",
16      "description": "Badhiya hai",
17      "author": "PG Sarvaiya",
18      "createdAt": "2025-03-25T07:55:42.743Z",
19      "updatedAt": "2025-03-25T07:55:42.743Z",
20      "__v": 0
21    }
22  ]
23 }

```

 <b>Marwadi University</b> Marwadi Chandarana Group 	<b>Marwadi University</b> <b>Faculty of Technology</b> <b>Department of Information and Communication Technology</b>	
<b>Subject: Advanced Web Technology</b>	<b>Aim:</b> Develop a web application which involves database operations using NodeJS	
<b>Experiment :- 2</b>	<b>Date:- 29-03-2025</b>	<b>Enrollment No:- 92200133003</b>

## MONGODB :



The screenshot shows the MongoDB Compass interface. The database is 'mu\_blog\_db' and the collection is 'mublogs'. There are 4 documents in the collection. The documents are as follows:

```

{
  "updatedAt": "2025-03-25T07:55:42.743+00:00",
  "__v": 0
}

{
  "_id": "67e368beaaa0e9f2f1d23d01",
  "title": "My second Blog",
  "description": "Badhiya hai",
  "author": "PG Sarvaiya",
  "createdAt": "2025-03-26T02:38:54.763+00:00",
  "updatedAt": "2025-03-26T02:38:54.763+00:00",
  "__v": 0
}

{
  "_id": "67e372dd563cb0e2e86db591",
  "title": "Satyana Prayogo",
  "description": "Achhi book hai",
  "author": "Gandhiji",
  "createdAt": "2025-03-26T03:22:05.632+00:00",
  "updatedAt": "2025-03-26T03:22:05.632+00:00",
  "__v": 0
}

{
  "_id": "67e378ff563cb0e2e86db59b",
  "title": "Dream 11",
  "description": "Khelo Dimag Se",
  "author": "Prashant",
  "createdAt": "2025-03-26T03:48:15.022+00:00",
  "updatedAt": "2025-03-26T03:48:15.022+00:00",
  "__v": 0
}

```