

 <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 Express		
<b>Experiment :- 4</b>	<b>Date:- 05-03-2025</b>	<b>Enrollment No:- 92200133003</b>	

**CODE : -**

#### Student / Model.js

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

const studentSchema = new mongoose.Schema({
  name: {
    type: String,
    required: true
  },
  age: Number,
  department: String
});

module.exports = mongoose.model('Student', studentSchema);
```

#### Student / routes.js

```
const express = require('express');
const router = express.Router();
const Student = require('../models/students');

// CREATE
router.post('/', async (req, res) => {
  try {
    const student = new Student(req.body);
    await student.save();
    res.status(201).json(student);
  } catch (error) {
    res.status(400).json({ error: error.message });
  }
});

// READ ALL
router.get('/', async (req, res) => {
  try {
    const students = await Student.find();
    res.json(students);
  } catch (error) {
    res.status(500).json({ error: error.message });
  }
});

// READ ONE
router.get('/:id', async (req, res) => {
  try {
    const student = await Student.findById(req.params.id);
    if (!student) return res.status(404).json({ message: 'Student not found' });
    res.json(student);
  } catch (error) {
    res.status(500).json({ error: error.message });
  }
});

// UPDATE
router.put('/:id', async (req, res) => {
  try {
    const student = await Student.findByIdAndUpdate(req.params.id, req.body, { new: true });
    res.json(student);
  } catch (error) {
    res.status(400).json({ error: error.message });
  }
});
```

 <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 Express	
<b>Experiment :- 4</b>	<b>Date:- 05-03-2025</b>	<b>Enrollment No:- 92200133003</b>

```
// DELETE
router.delete('/:id', async (req, res) => {
  try {
    await Student.findByIdAndDelete(req.params.id);
    res.json({ message: 'Student deleted' });
  } catch (error) {
    res.status(500).json({ error: error.message });
  }
});

module.exports = router;

index.js
const express = require('express');
const mongoose = require('mongoose');

const app = express();
app.use(express.json());

mongoose.connect('mongodb://127.0.0.1:27017/studentdb', {
  useNewUrlParser: true,
  useUnifiedTopology: true,
});

const studentSchema = new mongoose.Schema({
  name: String,
  age: Number,
  department: String,
});

const Student = mongoose.model('Student', studentSchema);

app.post('/students', async (req, res) => {
  const student = new Student(req.body);
  await student.save();
  res.send(student);
});

app.get('/students', async (req, res) => {
  const students = await Student.find();
  res.send(students);
});

app.listen(3001, () => {
  console.log('Server is running on http://localhost:3001');
});

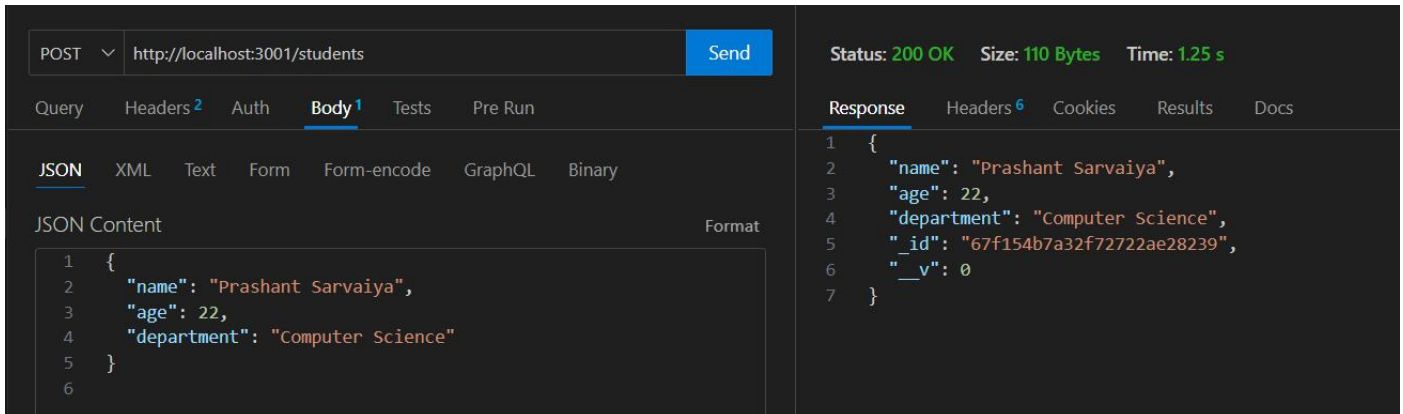
.env

MONGO_URI=mongodb://localhost:27017/studentdb
PORT=3001
```

 <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 Express		
<b>Experiment :- 4</b>	<b>Date:- 05-03-2025</b>	<b>Enrollment No:- 92200133003</b>	

## OUTPUT:

### POST Method



POST <http://localhost:3001/students> **Send**

Status: **200 OK** Size: **110 Bytes** Time: **1.25 s**

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   "name": "Prashant Sarvaiya",
3   "age": 22,
4   "department": "Computer Science"
5 }
6

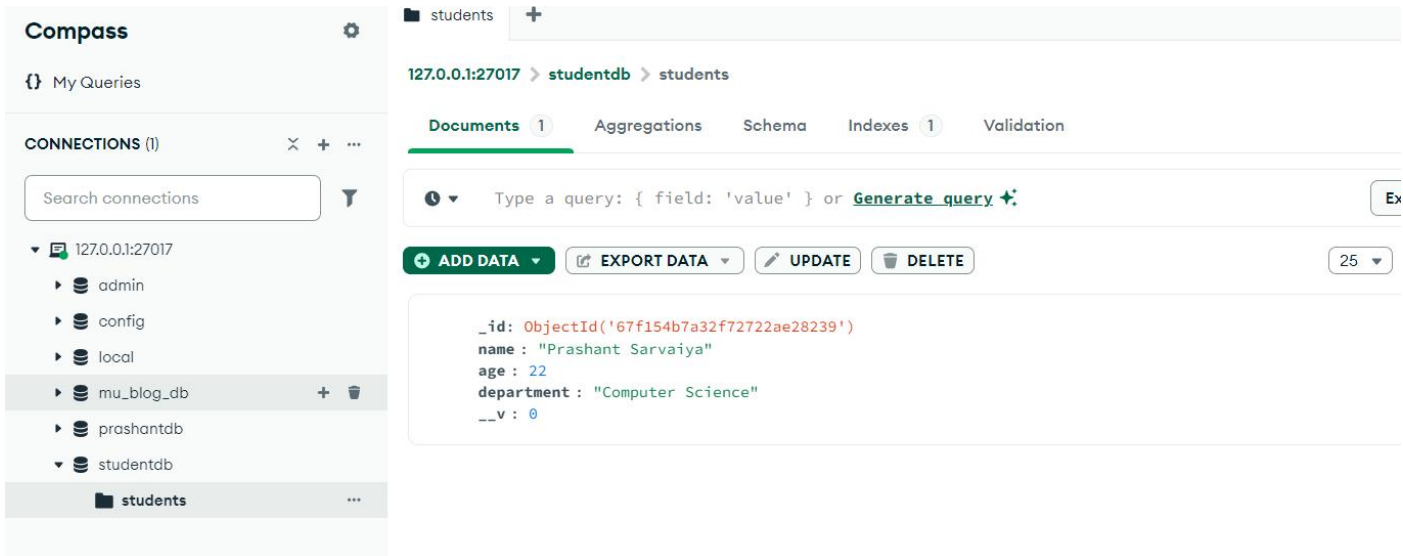
```

Response Headers <sup>6</sup> Cookies Results Docs

```

1 {
2   "name": "Prashant Sarvaiya",
3   "age": 22,
4   "department": "Computer Science",
5   "_id": "67f154b7a32f72722ae28239",
6   "__v": 0
7 }

```



**Compass**

{ } My Queries

CONNECTIONS (1)

127.0.0.1:27017

- admin
- config
- local
- mu\_blog\_db
- prashantdb
- studentdb
- students**

students +

127.0.0.1:27017 > studentdb > students

Documents <sup>1</sup> Aggregations Schema Indexes <sup>1</sup> Validation

Type a query: { field: 'value' } or [Generate query](#)

ADD DATA EXPORT DATA UPDATE DELETE

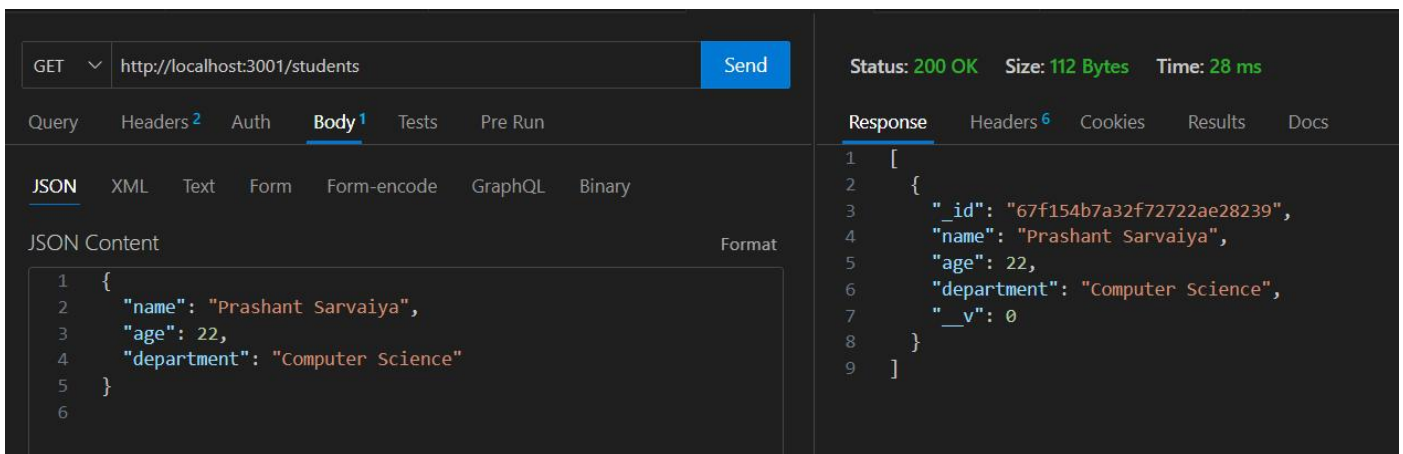
25

```

_id: ObjectId('67f154b7a32f72722ae28239')
name: "Prashant Sarvaiya"
age: 22
department: "Computer Science"
__v: 0

```

### GET Method



GET <http://localhost:3001/students> **Send**

Status: **200 OK** Size: **112 Bytes** Time: **28 ms**

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   "name": "Prashant Sarvaiya",
3   "age": 22,
4   "department": "Computer Science"
5 }
6

```

Response Headers <sup>6</sup> Cookies Results Docs

```

1 [
2   {
3     "_id": "67f154b7a32f72722ae28239",
4     "name": "Prashant Sarvaiya",
5     "age": 22,
6     "department": "Computer Science",
7     "__v": 0
8   }
9 ]

```

 <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 Express	
<b>Experiment :- 4</b>	<b>Date:- 05-03-2025</b>	<b>Enrollment No:- 92200133003</b>

#### PUT Method:

PUT

http://localhost:3001/students/67f154b7a32f72722ae28239

Send

Status: 200 OK

Size: 91 Bytes

Time: 34 ms

Query

Headers 2

Auth

Body 1

Tests

Pre Run

JSON

XML

Text

Form

Form-encode

GraphQL

Binary

JSON Content

Format

```

1 {
2   "name": "Harsh Doshi",
3   "age": 21,
4   "department": "ICT"
5 }
6

```

Response

Headers 6

Cookies

Results

Docs

```

1 {
2   "_id": "67f154b7a32f72722ae28239",
3   "name": "Harsh Doshi",
4   "age": 21,
5   "department": "ICT",
6   "_v": 0
7 }

```

Documents 1

Aggregations

Schema

Indexes 1

Validation

Type a query: { field: 'value' } or [Generate query](#)

Explain

Reset

Find

</>

Options

ADD DATA

EXPORT DATA

UPDATE

DELETE

25

1 - 1 of 1

↺

↻

↷

☰

{ }

⌂

```

_id: ObjectId('67f154b7a32f72722ae28239')
name : "Harsh Doshi"
age : 21
department : "ICT"
_v : 0

```

#### DELETE Method

DELETE

http://localhost:3001/students/67f154b7a32f72722ae28239

Send

Status: 200 OK

Size: 144 Bytes

Time: 36 ms

Query

Headers 2

Auth

Body 1

Tests

Pre Run

JSON

XML

Text

Form

Form-encode

GraphQL

Binary

JSON Content

Format

```

1 {
2   "name": "Harsh Doshi",
3   "age": 21,
4   "department": "ICT"
5 }
6

```

Response

Headers 6

Cookies

Results

Docs

```

1 {
2   "message": "Student deleted successfully",
3   "student": {
4     "_id": "67f154b7a32f72722ae28239",
5     "name": "Harsh Doshi",
6     "age": 21,
7     "department": "ICT",
8     "_v": 0
9   }
10 }

```

students

+

127.0.0.1:27017 > studentdb > students

Open MongoDB shell

Documents 1

Aggregations

Schema

Indexes 1

Validation

Type a query: { field: 'value' } or [Generate query](#)

Explain

Reset

Find

</>

Options

ADD DATA

EXPORT DATA

UPDATE

DELETE

25

0 - 0 of 0

↺

↻

↷

☰

{ }

⌂

This collection has no data

It only takes a few seconds to import data from a JSON or CSV file.

Import Data