

 <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 (01CT1625)</b>	<b>Aim:</b> Database: Using MongoDB, build the CRUD operations with NodeJS	
<b>Experiment :- 12</b>	<b>Date:- 20-04-2025</b>	<b>Enrollment No:- 92200133003</b>

## CODE :

### Index.js

```
const express = require('express');
const mongoose = require('mongoose');
const bodyParser = require('body-parser');
const userRoutes = require('./routes/userRoutes');

const app = express();
const PORT = 5000;

// Middleware
app.use(bodyParser.json());

// MongoDB Connection
mongoose.connect('mongodb://127.0.0.1:27017/crudDB', {
  useNewUrlParser: true,
  useUnifiedTopology: true,
})
.then(() => console.log('MongoDB Connected!'))
.catch((err) => console.log(err));

// Routes
app.use('/api/users', userRoutes);

// Start Server
app.listen(PORT, () => {
  console.log(`Server is running on port ${PORT}`);
});
```

### Model

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

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

module.exports = mongoose.model('User', userSchema);
```

 <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 (01CT1625)</b>	<b>Aim:</b> Database: Using MongoDB, build the CRUD operations with NodeJS	
<b>Experiment :- 12</b>	<b>Date:- 20-04-2025</b>	<b>Enrollment No:- 92200133003</b>

### Route

```
const express = require('express');
const router = express.Router();
const User = require('../model/user');
```

#### // CREATE

```
router.post('/', async (req, res) => {
  try {
    const user = new User(req.body);
    const savedUser = await user.save();
    res.status(201).json(savedUser);
  } catch (err) {
    res.status(400).json({ error: err.message });
  }
});
```

#### // READ All

```
router.get('/', async (req, res) => {
  try {
    const users = await User.find();
    res.json(users);
  } catch (err) {
    res.status(500).json({ error: err.message });
  }
});
```

#### // READ One

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

#### // UPDATE

```
router.put('/:id', async (req, res) => {
  try {
    const updatedUser = await User.findByIdAndUpdate(
      req.params.id,
      req.body,
      { new: true } // return updated doc
    );
    res.json(updatedUser);
  } catch (err) {
    res.status(400).json({ error: err.message });
  }
});
```

#### // DELETE

 <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 (01CT1625)</b>	<b>Aim:</b> Database: Using MongoDB, build the CRUD operations with NodeJS
<b>Experiment :- 12</b>	<b>Date:- 20-04-2025</b>
	<b>Enrollment No:- 92200133003</b>

```

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

```

```

module.exports = router;

```

## OUTPUT

### POST

POST

http://localhost:5000/api/users

Send

Query

Headers 2

Auth

Body 1

Tests

Pre Run

JSON

XML

Text

Form

Form-encode

GraphQL

Binary

JSON Content

Format

```

1 {
2   "name": "Prashant Sarvaiya",
3   "email": "prashant@example.com",
4   "age": 22
5 }
6

```

Status: 201 Created

Size: 109 Bytes

Time: 64 ms

Response

Headers 6

Cookies

Results

Docs

```

1 {
2   "name": "Prashant Sarvaiya",
3   "email": "prashant@example.com",
4   "age": 22,
5   "_id": "67fd69519e18ec6c78c76011",
6   "__v": 0
7 }

```

### UPDATE

PUT

http://localhost:5000/api/users/67fd69be9e18ec6c78c76013

Send

Query

Headers 2

Auth

Body 1

Tests

Pre Run

JSON

XML

Text

Form

Form-encode

GraphQL

Binary

JSON Content

Format

```

1 {
2   "name": "Jay jani",
3   "email": "jaii@example.com",
4   "age": 24
5 }
6

```

Status: 200 OK

Size: 96 Bytes

Time: 42 ms

Response

Headers 6

Cookies

Results

Docs

```

1 {
2   "_id": "67fd69be9e18ec6c78c76013",
3   "name": "Jay jani",
4   "email": "jaii@example.com",
5   "age": 24,
6   "__v": 0
7 }

```

 <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 (01CT1625)</b>	<b>Aim:</b> Database: Using MongoDB, build the CRUD operations with NodeJS	
<b>Experiment :- 12</b>	<b>Date:- 20-04-2025</b>	<b>Enrollment No:- 92200133003</b>

DELETE

DELETE

http://localhost:5000/api/users/67fd69be9e18ec6c78c76013

Send

Status: 200 OK

Size: 39 Bytes

Time: 11 ms

Query

Headers 2

Auth

Body 1

Tests

Pre Run

JSON

XML

Text

Form

Form-encode

GraphQL

Binary

JSON Content

Format

```

1 {
2   "name": "Jay jani",
3   "email": "jaii@example.com",
4   "age": 24
5 }
6

```

Response

Headers 6

Cookies

Results

Docs

1 {

2 "message": "User deleted successfully"

3 }

GET

GET

http://localhost:5000/api/users/

Send

Status: 200 OK

Size: 111 Bytes

Time: 20 ms

Query

Headers 2

Auth

Body 1

Tests

Pre Run

JSON

XML

Text

Form

Form-encode

GraphQL

Binary

JSON Content

Format

```

1

```

Response

Headers 6

Cookies

Results

Docs

1 [

2 {

3 "\_id": "67fd69519e18ec6c78c76011",

4 "name": "Prashant Sarvaiya",

5 "email": "prashant@example.com",

6 "age": 22,

7 "\_\_v": 0

8 }

9 ]

DB

users

+

127.0.0.1:27017 > crudDB > users

Open MongoDB shell

Documents 1

Aggregations

Schema

Indexes 2

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('67fd69519e18ec6c78c76011')
name: "Prashant Sarvaiya"
email: "prashant@example.com"
age: 22
__v: 0

```