



Marwadi University Faculty of Technology

Department of Information and Communication Technology

Subject: Advanced Web Technology (01CT1625)

Aim: Database: Using MongoDB, build the CRUD operations with NodeJS

```
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);





Marwadi University Faculty of Technology

Department of Information and Communication Technology

Subject: Advanced Web Technology (01CT1625)

Aim: Database: Using MongoDB, build the CRUD operations with NodeJS

Experiment :- 12 Date:- 20-04-2025

Enrollment No:- 92200133003

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) => {
  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) => {
  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) => {
  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
```



Marwadi University Faculty of Technology

Department of Information and Communication Technology

Subject: Advanced Web Technology (01CT1625)

Aim: Database: Using MongoDB, build the CRUD operations with NodeJS

```
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 V http://localhost:5000/api/users

Query Headers 2 Auth Body 1 Tests Pre Run

Response Headers 6 Cookies Results Docs

JSON XML Text Form Form-encode GraphQL Binary

JSON Content

Format

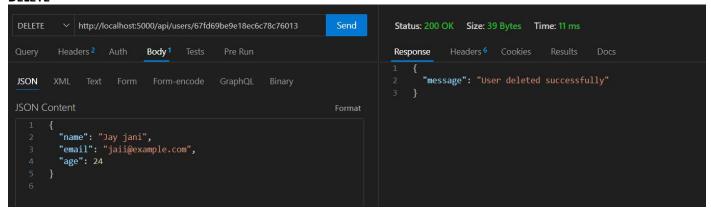
1 {
2    "name": "Prashant Sarvaiya",
3    "email": "prashant@example.com",
4    "age": 22,
5    "_id": "67fd69519e18ec6c78c76011",
6    "__v": 0
7 }

7 }
```

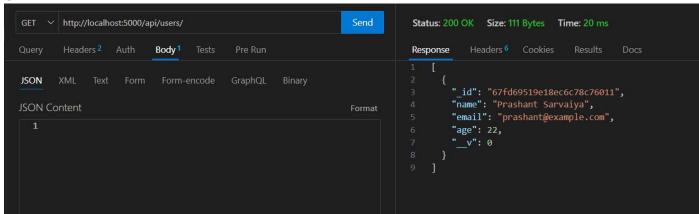
UPDATE



DELETE



GET



DB

