

|       |            |
|-------|------------|
| NAME  | Chirag     |
| UID   | 23BCS14318 |
| CLASS | 622-B      |

➤ NodeJs PRACTISE 5.1

- CODE

Server.js (Main server):

```
const express = require('express');
const mongoose = require('mongoose');
const app = express();
const productRoutes = require('./routes/products');

// Middleware to parse JSON
app.use(express.json());

// Connect to MongoDB
mongoose.connect('mongodb+srv://alpha:alpha2025@cluster0.gttfsb5.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0')
  .then(() => console.log('Connected to MongoDB'))
  .catch(err => console.error('Error connecting to MongoDB:', err.message));

// Mount the product routes
app.use('/products', productRoutes);

// Root route
app.get('/', (req, res) => {
  res.send('API is running. Use /products endpoint.');
```

```
// Start server
const PORT = process.env.PORT || 5000;
app.listen(PORT, () => console.log(`Server running on port ${PORT}`));
```

package.json :

```
{
  "name": "fundamentals",
  "version": "1.0.0",
  "main": "server.js",
  "scripts": {
    "server": "nodemon server.js",
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": "",

  "dependencies": {
    "body-parser": "^2.2.0",
    "express": "^5.1.0",
    "mongoose": "^8.19.1"
  }
}
```

/models/Product.js :

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

const ProductSchema = new mongoose.Schema({
  name: { type: String, required: true, trim: true },
  price: { type: Number, required: true, min: 0 },
  category: { type: String, required: true, trim: true },
  createdAt: { type: Date, default: Date.now }
});

module.exports = mongoose.model('Product', ProductSchema);
```

/routes/products.js :

```
// routes/products.js
const express = require('express');
const router = express.Router();
const Product = require('../models/Product');

// Get all products
router.get('/', async (req, res) => {
  try {
    const products = await Product.find();
    res.json(products);
  } catch (err) {
    res.status(500).json({ message: err.message });
  }
});

// Add a new product
router.post('/add', async (req, res) => {
  const product = new Product({
    name: req.body.name,
    price: req.body.price,
    category: req.body.category
  });

  try {
    const newProduct = await product.save();
    res.status(201).json(newProduct);
  } catch (err) {
    res.status(400).json({ message: err.message });
  }
});

// Update a product by ID
router.put('/:id', async (req, res) => {
  try {
    const updatedProduct = await Product.findByIdAndUpdate(
      req.params.id,
      {
        name: req.body.name,
        price: req.body.price,
        category: req.body.category
      },
      { new: true }
    );
  }
});
```

```

    res.json(updatedProduct);
  } catch (err) {
    res.status(400).json({ message: err.message });
  }
});

// Delete a product by ID
router.delete('/:id', async (req, res) => {
  try {
    await Product.findByIdAndDelete(req.params.id);
    res.json({ message: 'Product deleted' });
  } catch (err) {
    res.status(500).json({ message: err.message });
  }
});

module.exports = router;

```

## • OUTPUT

HTTP http://localhost:5000/products/

GET http://localhost:5000/products/

Params Authorization Headers (8) Body • Scripts Tests Settings

Body Cookies Headers (7) Test Results ↺

200 OK

{ } JSON Preview Visualize

```

1  [
2    {
3      "_id": "68f0bffe869105f9304bf760",
4      "name": "Keyboard",
5      "price": 1200,
6      "category": "Accessories",
7      "createdAt": "2025-10-16T09:50:54.418Z",
8      "__v": 0
9    },
10   {
11     "_id": "68f0fd7fc109caff1231351a",
12     "name": "Laptop",
13     "price": 9800,
14     "category": "Electronics",
15     "createdAt": "2025-10-16T14:13:19.909Z",
16     "__v": 0
17   },
18   {
19     "_id": "68f0fdb9c109caff1231351c",
20     "name": "Wireless Mouse",
21     "price": 560,
22     "category": "Accessories",
23     "createdAt": "2025-10-16T14:14:17.222Z",
24     "__v": 0
25   }

```

HTTP <http://localhost:5000/products/68f1021ecae15bccb029f337>

DELETE <http://localhost:5000/products/68f1021ecae15bccb029f337>

Params Authorization Headers (8) Body • Scripts Tests Settings

Body Cookies Headers (7) Test Results ↺

200 OK

{ } JSON ▾ ▶ Preview Visualize ▾

```
1 {
2   "message": "Product deleted successfully",
3   "deletedProduct": {
4     "id": "68f1021ecae15bccb029f337",
5     "name": "Tab",
6     "category": "Electronics"
7   }
8 }
```

HTTP <http://localhost:5000/products/add>

POST <http://localhost:5000/products/add>

Params Authorization Headers (8) Body • Scripts Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON ▾

```
1 {
2   "name": "Charger",
3   "price": 160,
4   "category": "Accessories"
5 }
```

Body Cookies Headers (7) Test Results ↺

201 Created

{ } JSON ▾ ▶ Preview Visualize ▾

```
1 {
2   "name": "Charger",
3   "price": 160,
4   "category": "Accessories",
5   "_id": "68f105aaca15bccb029f33f",
6   "createdAt": "2025-10-16T14:48:10.756Z",
7   "__v": 0
8 }
```

Output Link (Postman) : <http://localhost:5000/products/>

API's: GET <http://localhost:5000/products/>

DELETE [http://localhost:5000/products/\(id\)](http://localhost:5000/products/(id))

POST <http://localhost:5000/products/add>

