```
const express = require('express');
const mongoose = require('mongoose');
const bodyParser = require('body-parser');

const app = express();
const port = 3000;

mongoose.connect('mongodb://localhost/food-nutrition-db', { useNewUrlParser: true,
useUnifiedTopology: true });
const db = mongoose.connection;

db.on('error', console.error.bind(console, 'MongoDB connection error:'));
db.once('open', () => {
  console.log('Connected to MongoDB');
});

const foodSchema = new mongoose.Schema({
  name: { type: String, required: true },
  calories: { type: Number, required: true },
  protein: { type: Number, required: true },
  carbs: { type: Number, required: true },
  fat: { type: Number, required: true },
});

const Food = mongoose.model('Food', foodSchema);

app.use(bodyParser.json());

app.get('/api/foods', async (req, res) => {
  try {
    const foods = await Food.find();
    res.json(foods);
  } catch (error) {
    console.error(error);
    res.status(500).json({ error: 'Internal Server Error' });
  }
});

app.post('/api/foods', async (req, res) => {
  try {
    const newFood = new Food(req.body);
    const savedFood = await newFood.save();
    res.status(201).json(savedFood);
  } catch (error) {
    console.error(error);
    if (error.name === 'ValidationError') {
      res.status(400).json({ error: error.message });
    } else {
      res.status(500).json({ error: 'Internal Server Error' });
    }
  }
});

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