index.js

```
const express = require("express"); // Import the Express framework
const mongoose = require("mongoose"); // Import Mongoose for MongoDB interaction
const Product = require("./product"); // Import the Product model

const app = express(); // Create an Express application
const PORT = process.env.PORT || 2000; // Use process.env.PORT for Heroku deployment or default to 2000

app.use(express.json()); // Middleware to parse JSON bodies
app.use(express.urlencoded({ extended: true })); // Middleware to parse URL-encoded bodies

// MongoDB connection setup
mongoose.connect("mongodb+srv://e20280:mongodb@cluster0.bnio9tt.mongodb.net/appwithbackend")
    .then(() => {
        console.log("Connected to MongoDB"); // Log success message on successful connection
    })
    .catch((err) => {
        console.error("Error connecting to MongoDB", err); // Log error message on connection failure
    });

// POST API to add a product
app.post("/api/add_product", async (req, res) => {
    try {
        // Create a new product instance using the request body
        let newProduct = new Product({
            pname: req.body.pname,
            pprice: req.body.pprice,
            pdesc: req.body.pdesc
        });

        // Save the product to the database
        let savedProduct = await newProduct.save();
        // Respond with the created product and a 200 status code
        res.status(200).json(savedProduct);
        console.log("Added");
    } catch (error) {
        // Handle any errors and respond with a 400 status code
```

```
        res.status(400).json({ error: error.message });
    }
});

// GET API to retrieve all products
app.get("/api/get_product", async (req, res) => {
    try {
        // Retrieve all products from the database
        let products = await Product.find({});
        // Respond with the retrieved products and a 200 status code
        res.status(200).json(products);
        console.log("Getting");
    } catch (error) {
        // Handle any errors and respond with a 400 status code
        res.status(400).json({ error: error.message });
    }
});

// PUT API to update a product
app.put("/api/update/:id", async (req, res) => {
    try {
        // Update the product by its ID with the request body
        let updatedProduct = await Product.findByIdAndUpdate(req.params.id, req.body, { new: true });
        // Respond with the updated product and a 200 status code
        res.status(200).json(updatedProduct);
        console.log("Updating");
    } catch (error) {
        // Handle any errors and respond with a 400 status code
        res.status(400).json({ error: error.message });
    }
});

// DELETE API to delete a product
app.delete("/api/delete/:id", async (req, res) => {
    try {
        // Delete the product by its ID from the database
        await Product.findByIdAndDelete(req.params.id);
        // Respond with a 204 status code indicating no content
```

```
        res.status(204).send();
        console.log("Deleted");
    } catch (error) {
        // Handle any errors and respond with a 400 status code
        res.status(400).json({ error: error.message });
    }
});

// Start the server
app.listen(PORT, () => {
    console.log(`Server is running on http://localhost:${PORT}`); // Log the server start message
});

```