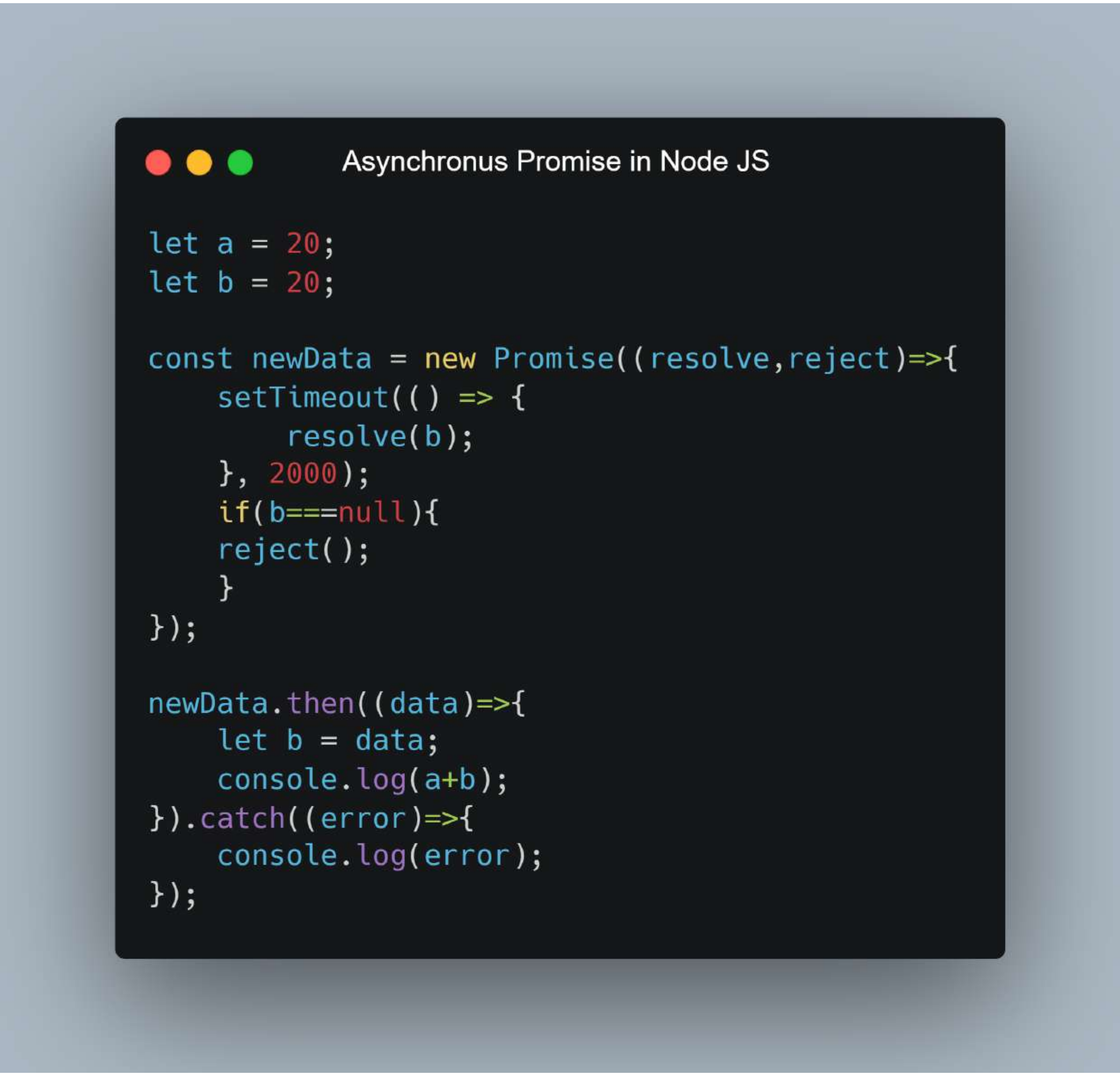
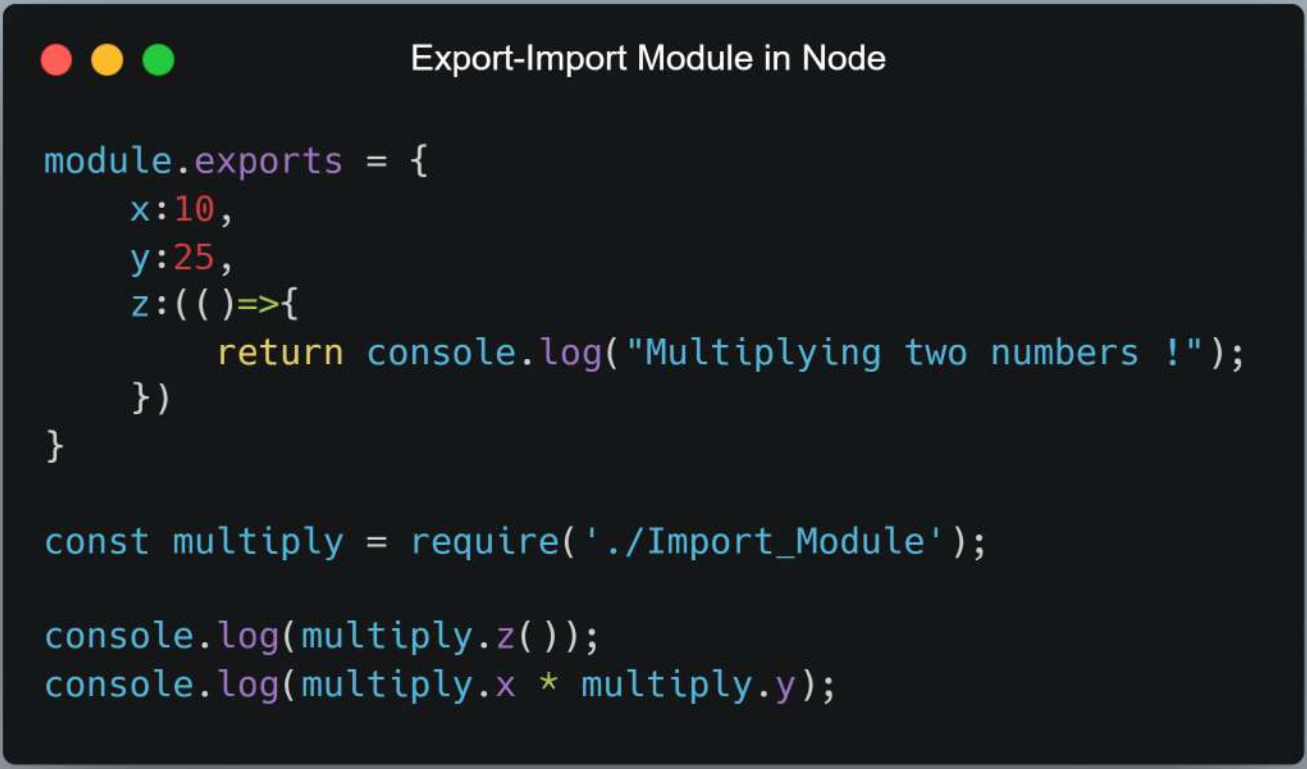


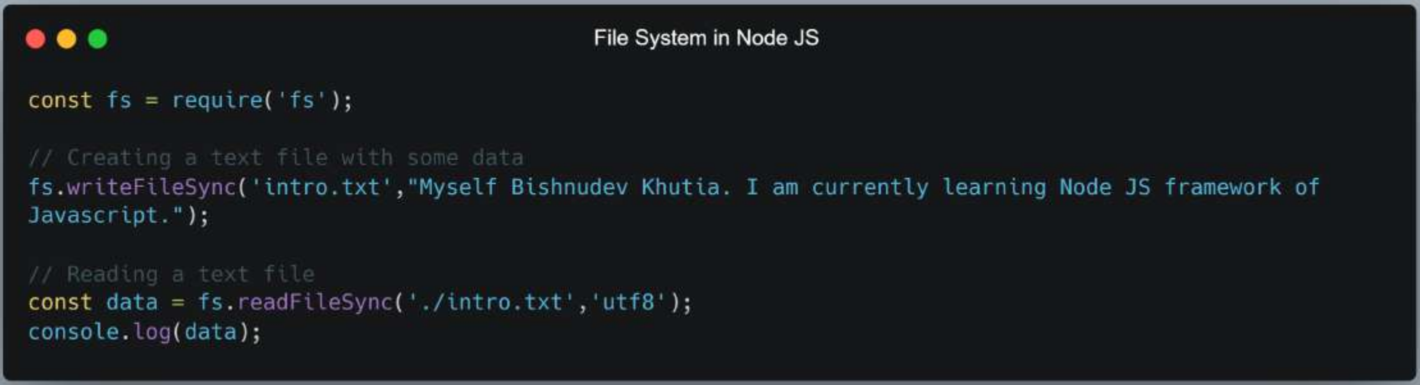
Asynchronus Promise in Node JS

```
let a = 20;
let b = 20;

const newData = new Promise((resolve,reject)=>{
    setTimeout(() => {
        resolve(b);
    }, 2000);
    if(b===null){
    reject();
    }
});

newData.then((data)=>{
    let b = data;
    console.log(a+b);
}).catch((error)=>{
    console.log(error);
});
```



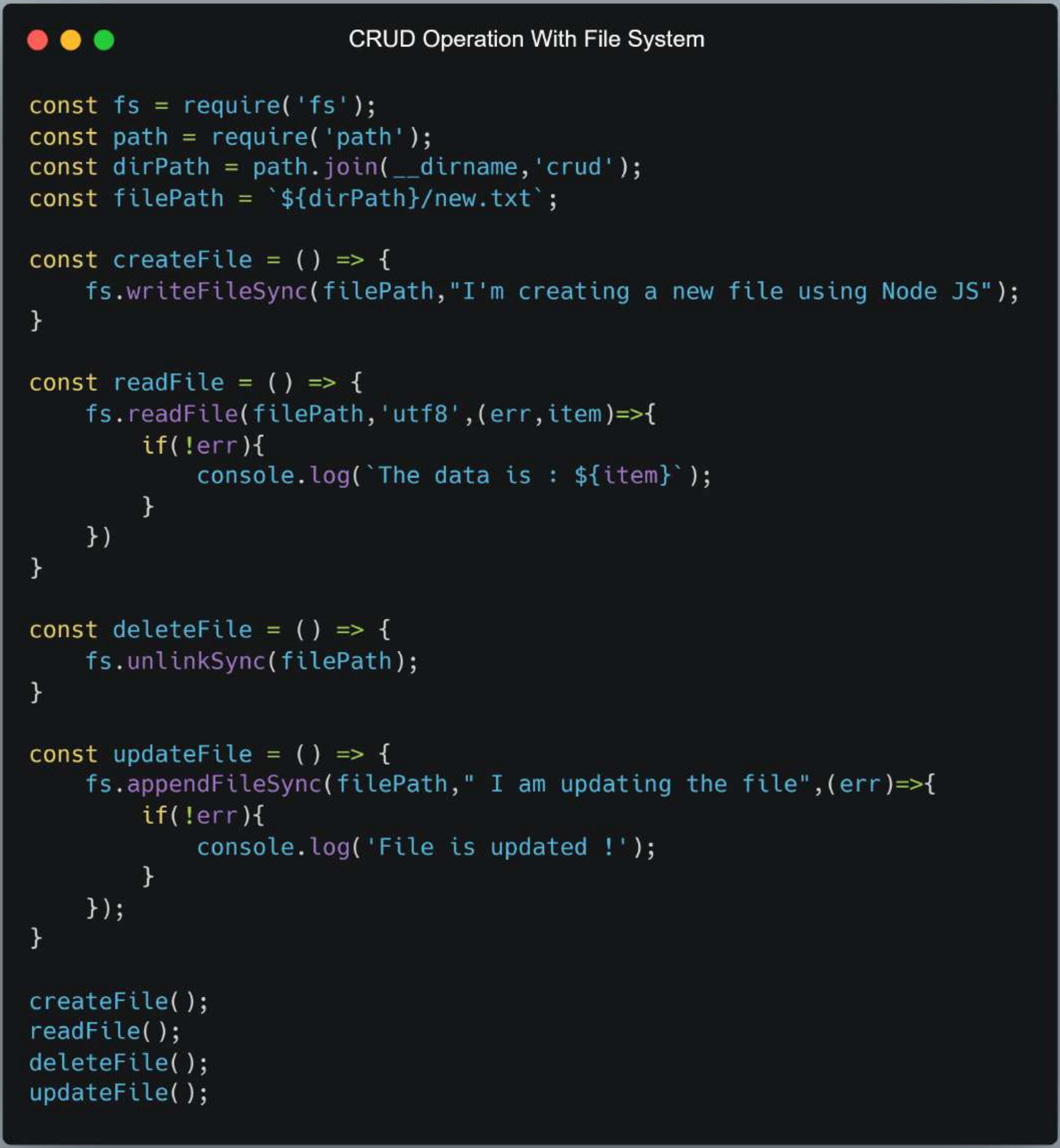
Export-Import Module in Node

```
module.exports = {
    x:10,
    y:25,
    z:(()=>{
        return console.log("Multiplying two numbers !");
    })
}

const multiply = require('./Import_Module');

console.log(multiply.z());
console.log(multiply.x * multiply.y);
```

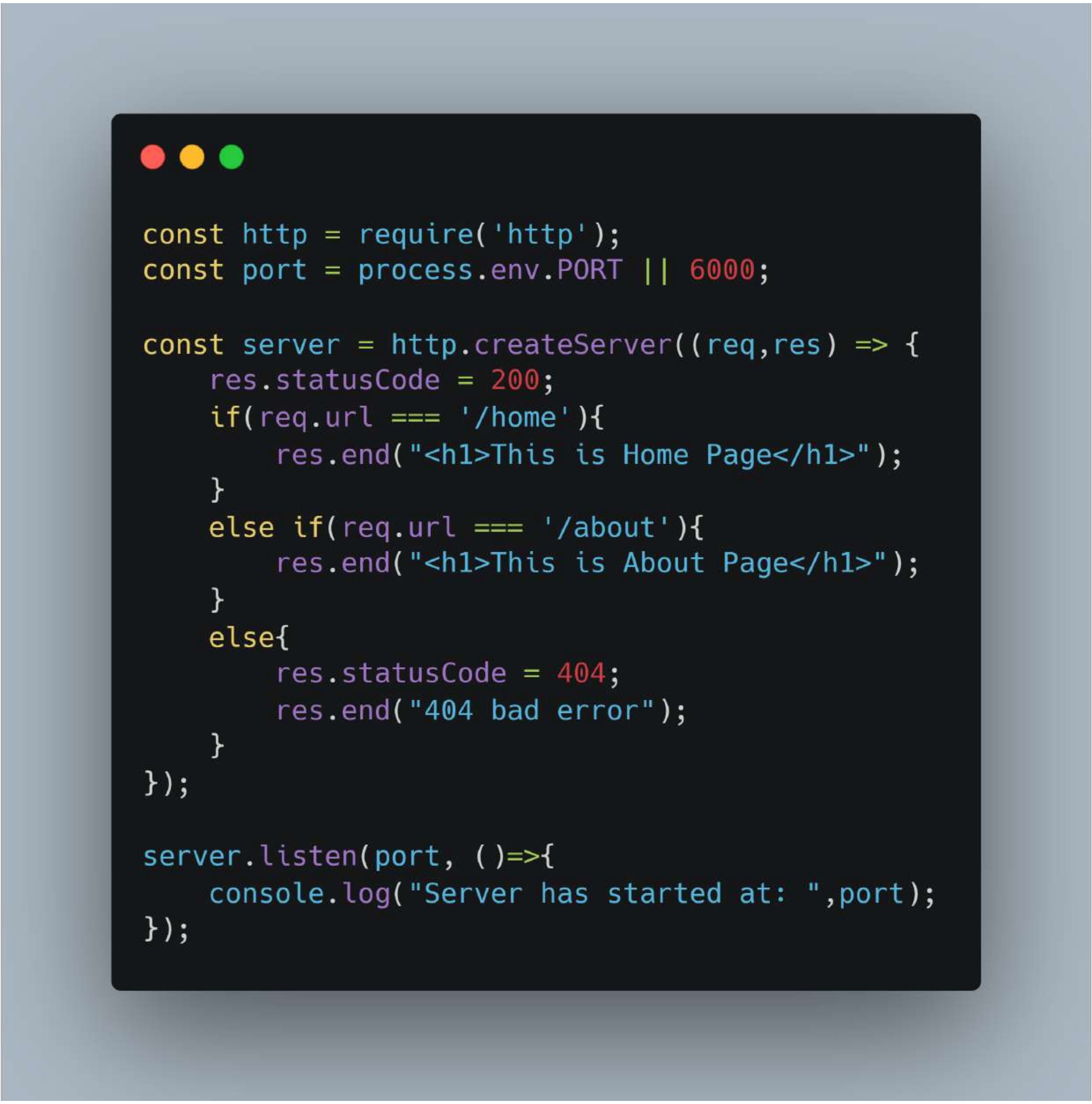


File System in Node JS

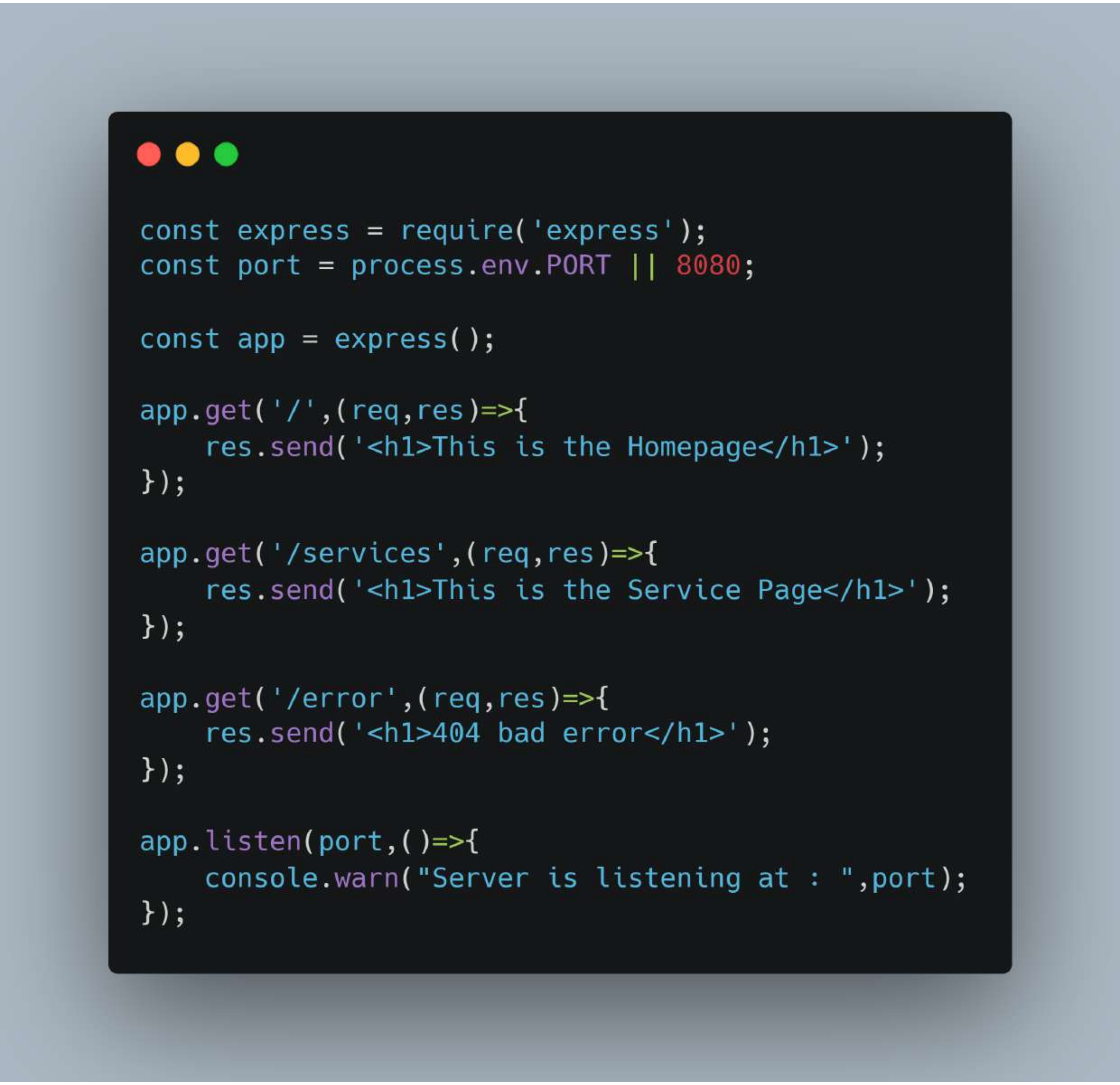
```javascript
const fs = require('fs');

// Creating a text file with some data
fs.writeFileSync('intro.txt',"Myself Bishnudev Khutia. I am currently learning Node JS framework of Javascript.");

// Reading a text file
const data = fs.readFileSync('./intro.txt','utf8');
console.log(data);
```



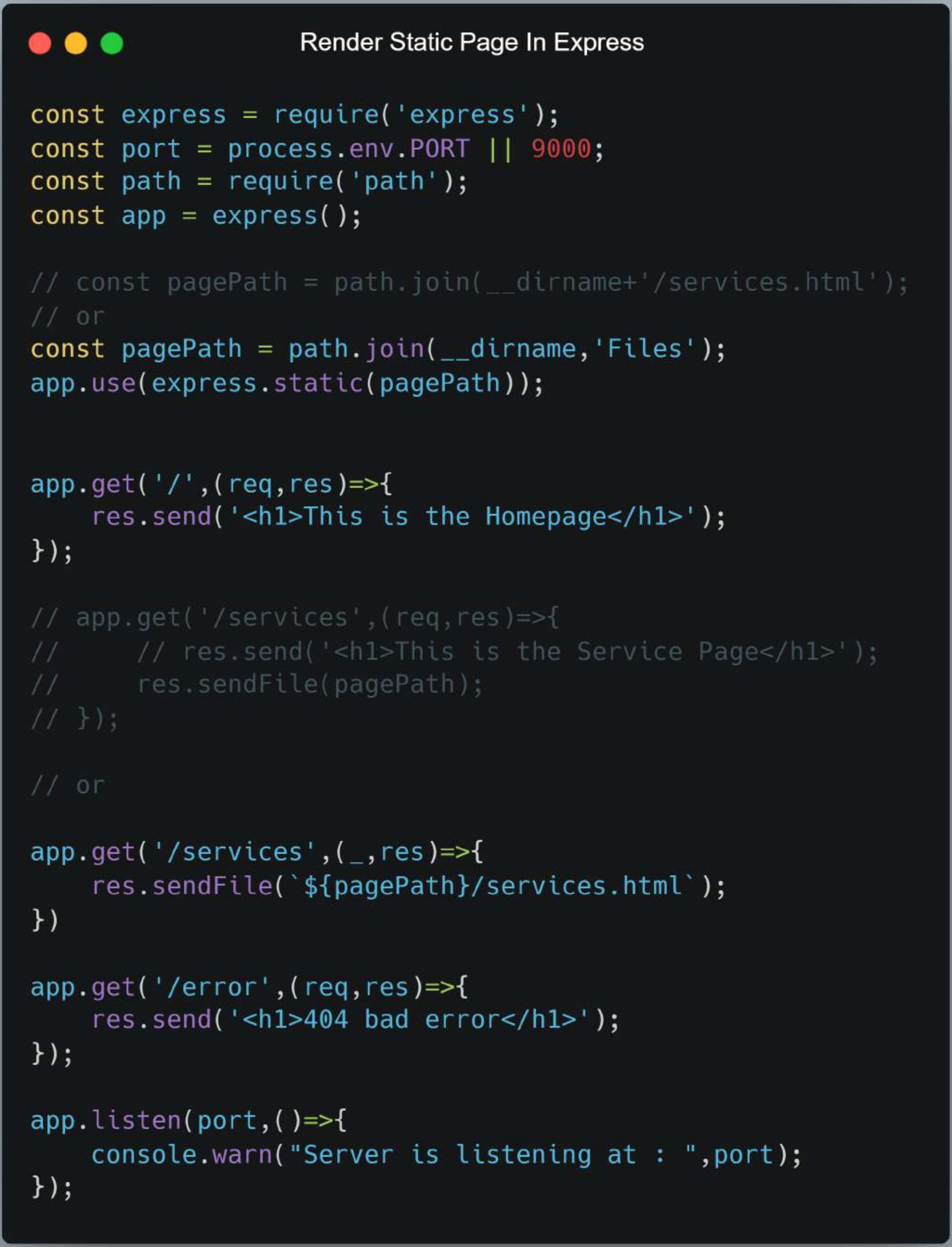
```
const fs = require('fs');
const path = require('path');
const dirPath = path.join(__dirname,'crud');
const filePath = `${dirPath}/new.txt`;

const createFile = () => {
    fs.writeFileSync(filePath,"I'm creating a new file using Node JS");
}

const readFile = () => {
    fs.readFile(filePath,'utf8',(err,item)=>{
        if(!err){
            console.log(`The data is : ${item}`);
        }
    })
}

const deleteFile = () => {
    fs.unlinkSync(filePath);
}

const updateFile = () => {
    fs.appendFileSync(filePath," I am updating the file",(err)=>{
        if(!err){
            console.log('File is updated !');
        }
    });
}

createFile();
readFile();
deleteFile();
updateFile();
```



```
const http = require('http');
const port = process.env.PORT || 6000;

const server = http.createServer((req,res) => {
    res.statusCode = 200;
    if(req.url === '/home'){
        res.end("<h1>This is Home Page</h1>");
    }
    else if(req.url === '/about'){
        res.end("<h1>This is About Page</h1>");
    }
    else{
        res.statusCode = 404;
        res.end("404 bad error");
    }
});

server.listen(port, ()=>{
    console.log("Server has started at: ",port);
});
```

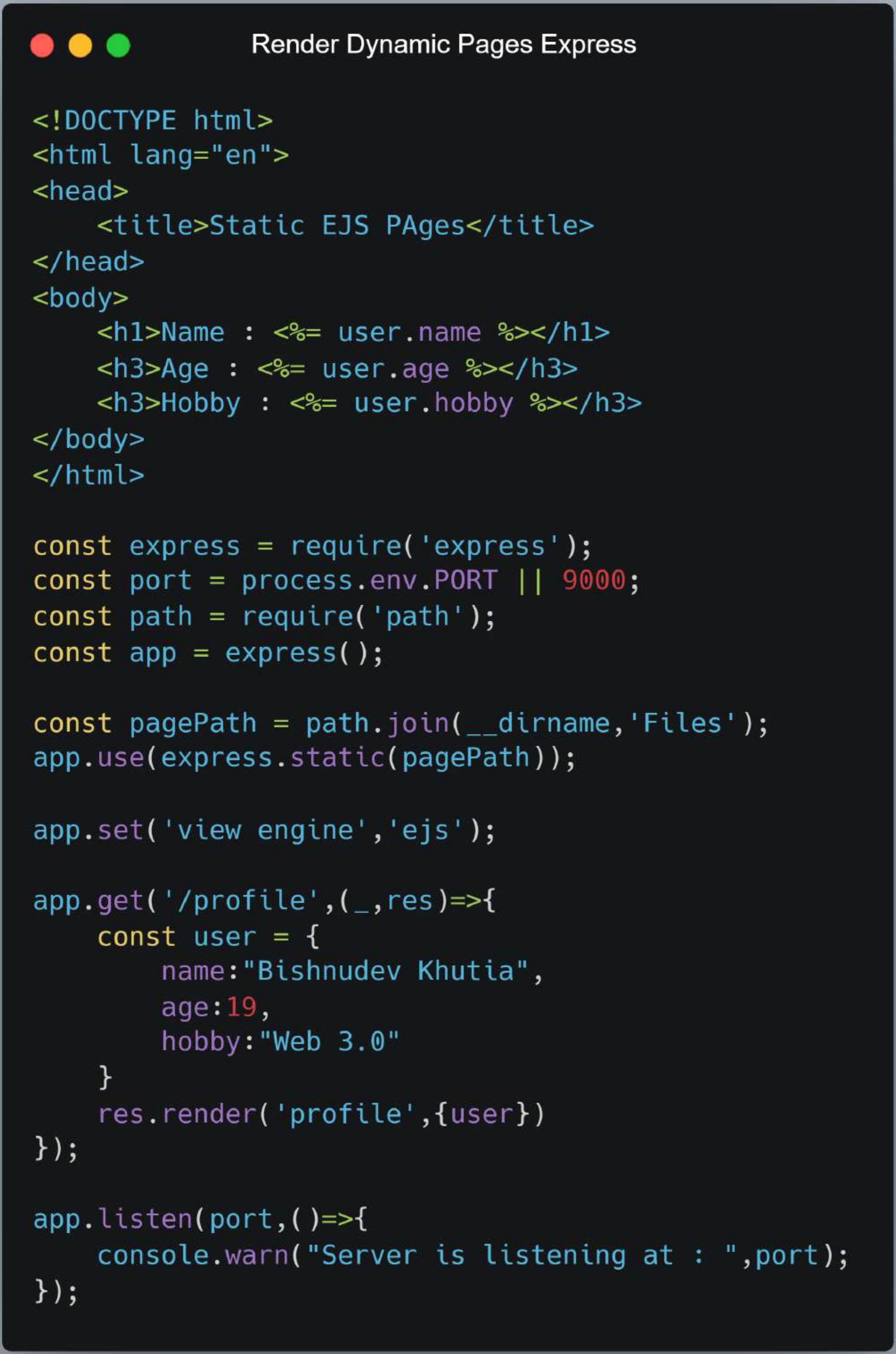


```
const express = require('express');
const port = process.env.PORT || 8080;

const app = express();

app.get('/',(req,res)=>{
    res.send('<h1>This is the Homepage</h1>');
});

app.get('/services',(req,res)=>{
    res.send('<h1>This is the Service Page</h1>');
});

app.get('/error',(req,res)=>{
    res.send('<h1>404 bad error</h1>');
});

app.listen(port,()=>{
    console.warn("Server is listening at : ",port);
});
```



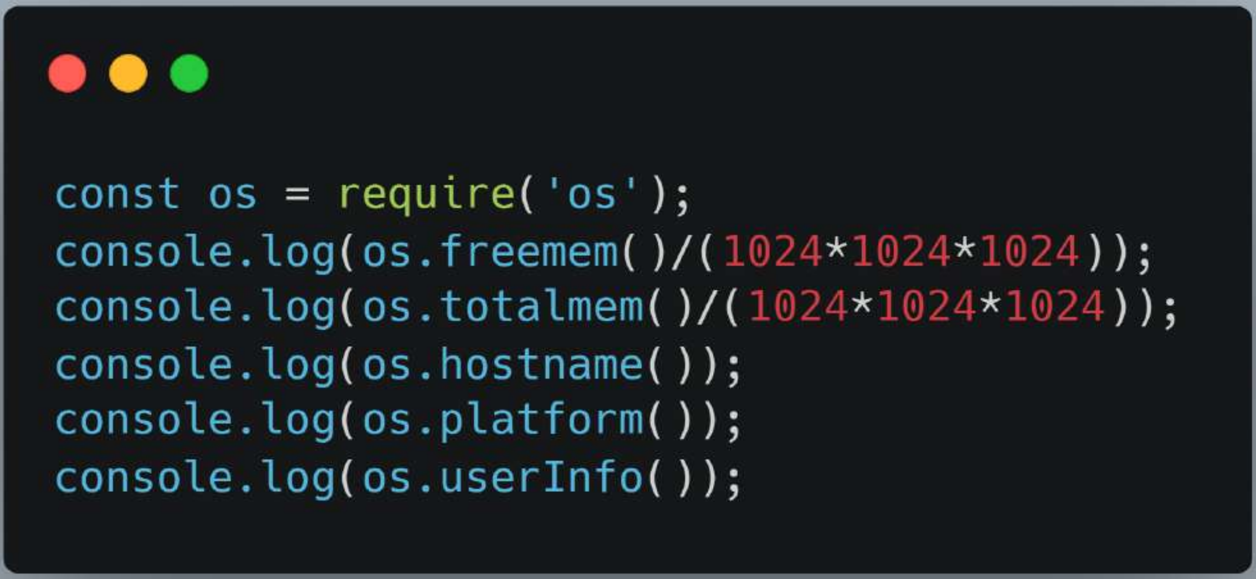
Render Static Page In Express

```
const express = require('express');
const port = process.env.PORT || 9000;
const path = require('path');
const app = express();

// const pagePath = path.join(__dirname+'/services.html');
// or
const pagePath = path.join(__dirname,'Files');
app.use(express.static(pagePath));


app.get('/',(req,res)=>{
    res.send('<h1>This is the Homepage</h1>');
});

// app.get('/services',(req,res)=>{
//     // res.send('<h1>This is the Service Page</h1>');
//     res.sendFile(pagePath);
// });

// or

app.get('/services',(_,res)=>{
    res.sendFile(`${pagePath}/services.html`);
})

app.get('/error',(req,res)=>{
    res.send('<h1>404 bad error</h1>');
});

app.listen(port,()=>{
    console.warn("Server is listening at : ",port);
});
```

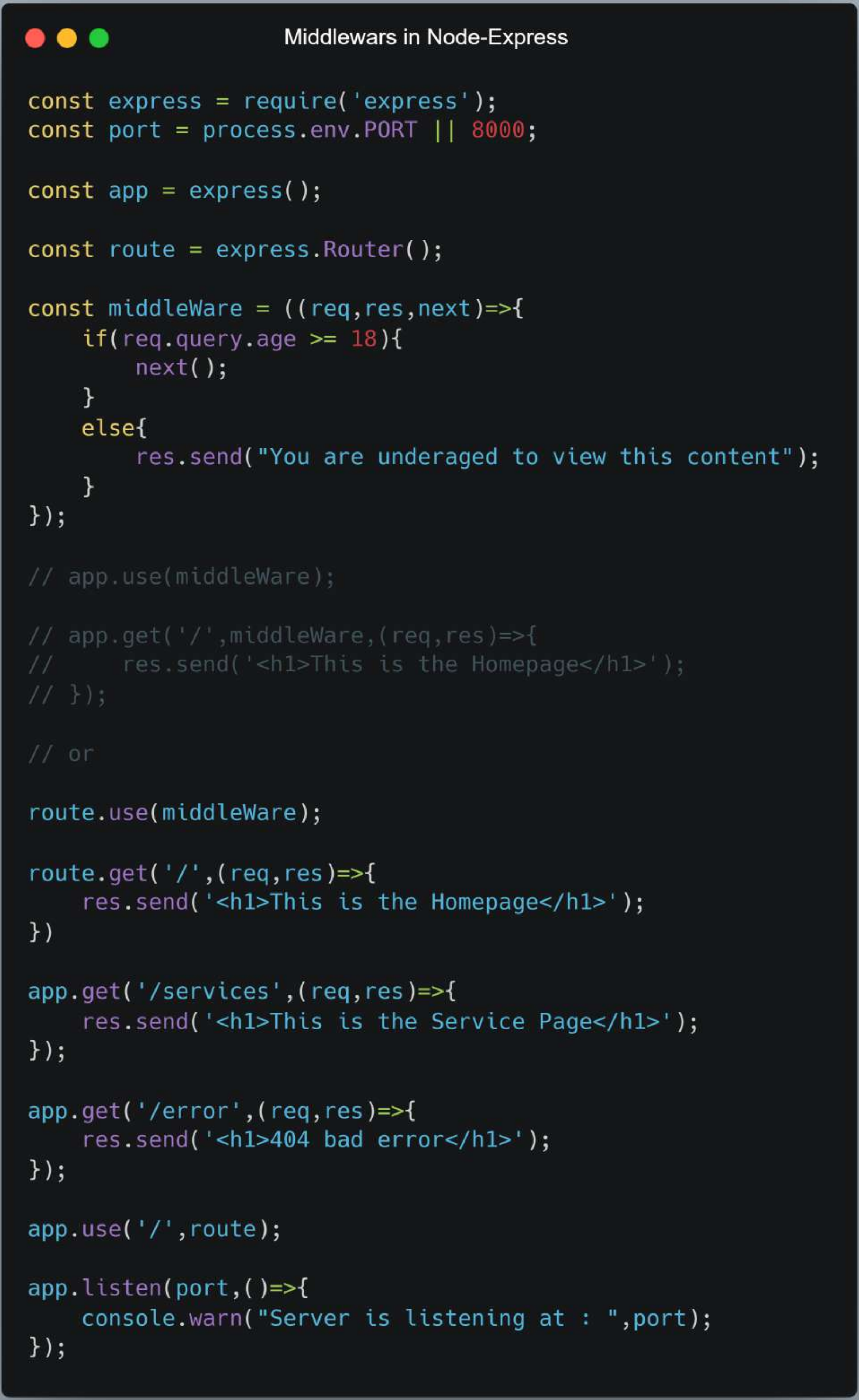


Render Dynamic Pages Express

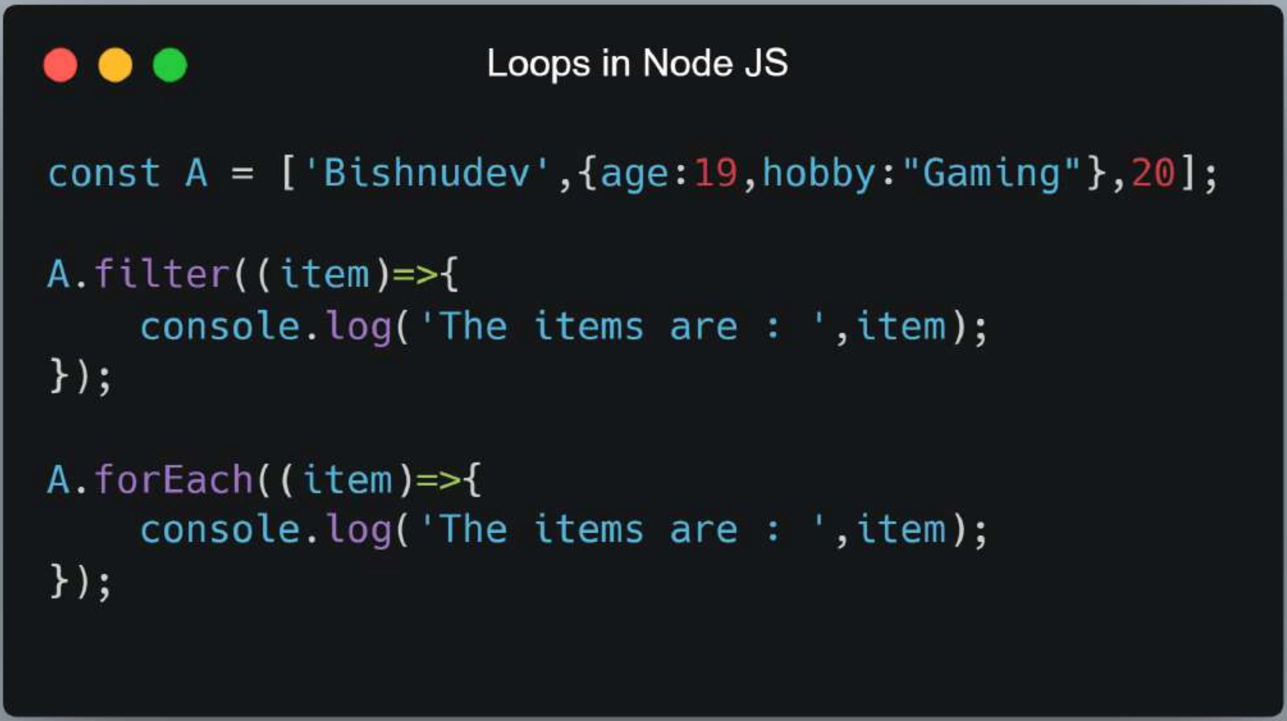
```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Static EJS PAges</title>
</head>
<body>
    <h1>Name : <%= user.name %></h1>
    <h3>Age : <%= user.age %></h3>
    <h3>Hobby : <%= user.hobby %></h3>
</body>
</html>

const express = require('express');
const port = process.env.PORT || 9000;
const path = require('path');
const app = express();

const pagePath = path.join(__dirname,'Files');
app.use(express.static(pagePath));

app.set('view engine','ejs');

app.get('/profile',(_,res)=>{
    const user = {
        name:"Bishnudev Khutia",
        age:19,
        hobby:"Web 3.0"
    }
    res.render('profile',{user})
});

app.listen(port,()=>{
    console.warn("Server is listening at : ",port);
});
```



```
const os = require('os');
console.log(os.freemem()/(1024*1024*1024));
console.log(os.totalmem()/(1024*1024*1024));
console.log(os.hostname());
console.log(os.platform());
console.log(os.userInfo());
```

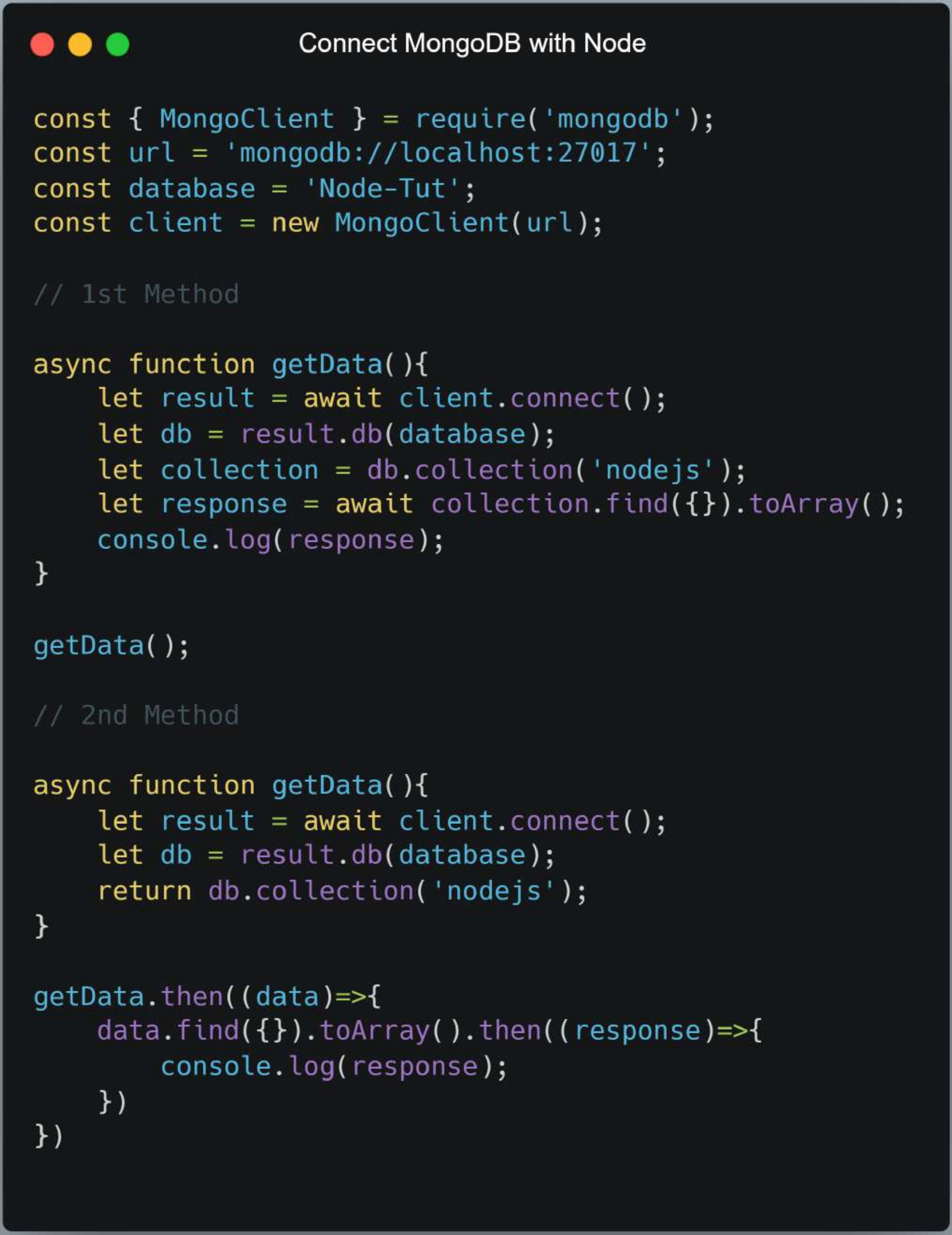


```
const express = require('express');
const port = process.env.PORT || 8000;

const app = express();

const route = express.Router();

const middleWare = ((req,res,next)=>{
    if(req.query.age >= 18){
        next();
    }
    else{
        res.send("You are underaged to view this content");
    }
});

// app.use(middleWare);

// app.get('/',middleWare,(req,res)=>{
//     res.send('<h1>This is the Homepage</h1>');
// });

// or

route.use(middleWare);

route.get('/',(req,res)=>{
    res.send('<h1>This is the Homepage</h1>');
})

app.get('/services',(req,res)=>{
    res.send('<h1>This is the Service Page</h1>');
});

app.get('/error',(req,res)=>{
    res.send('<h1>404 bad error</h1>');
});

app.use('/',route);

app.listen(port,()=>{
    console.warn("Server is listening at : ",port);
});
```



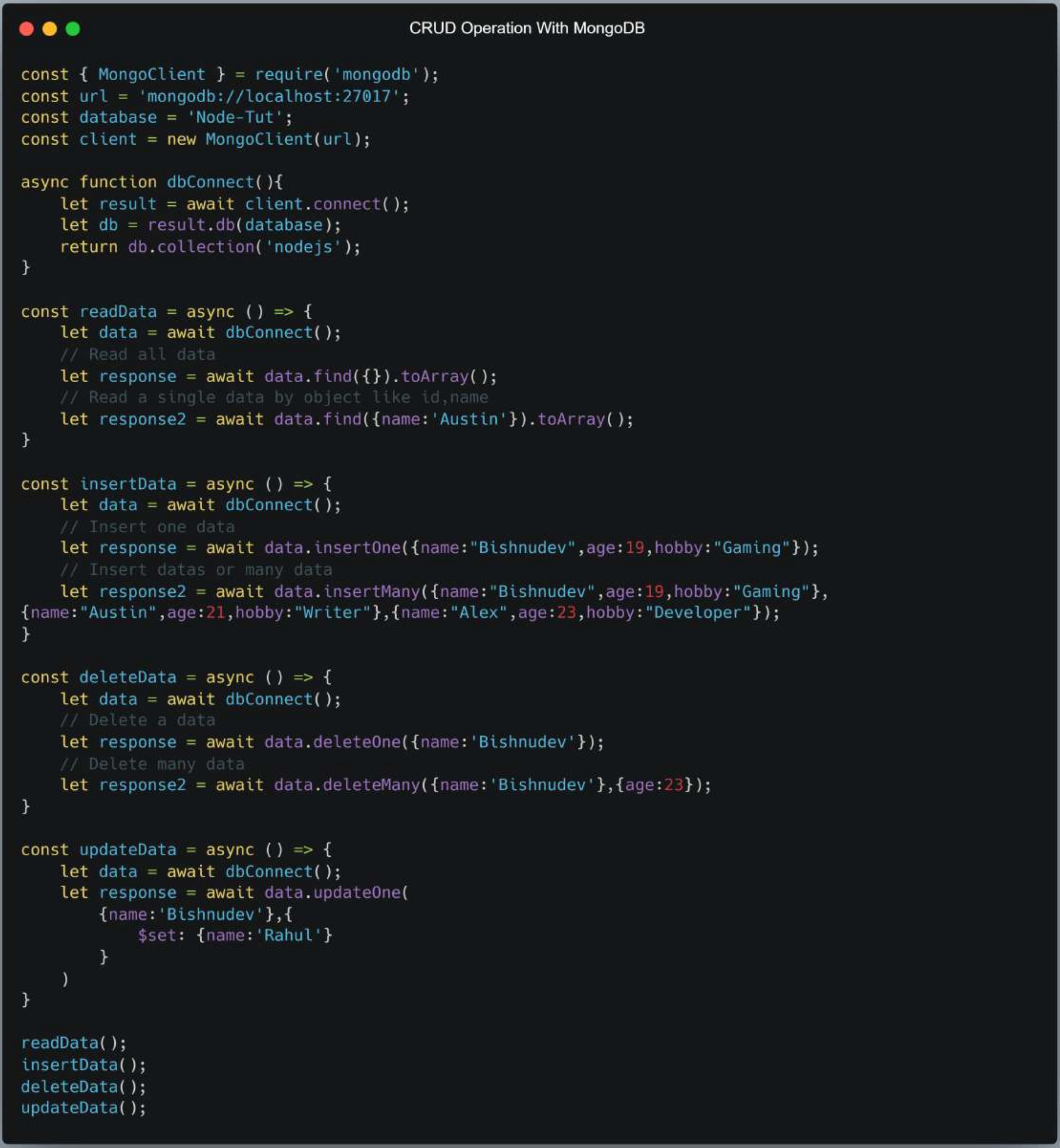
Loops in Node JS

```
const A = ['Bishnudev',{age:19,hobby:"Gaming"},20];

A.filter((item)=>{
    console.log('The items are : ',item);
});

A.forEach((item)=>{
    console.log('The items are : ',item);
});
```

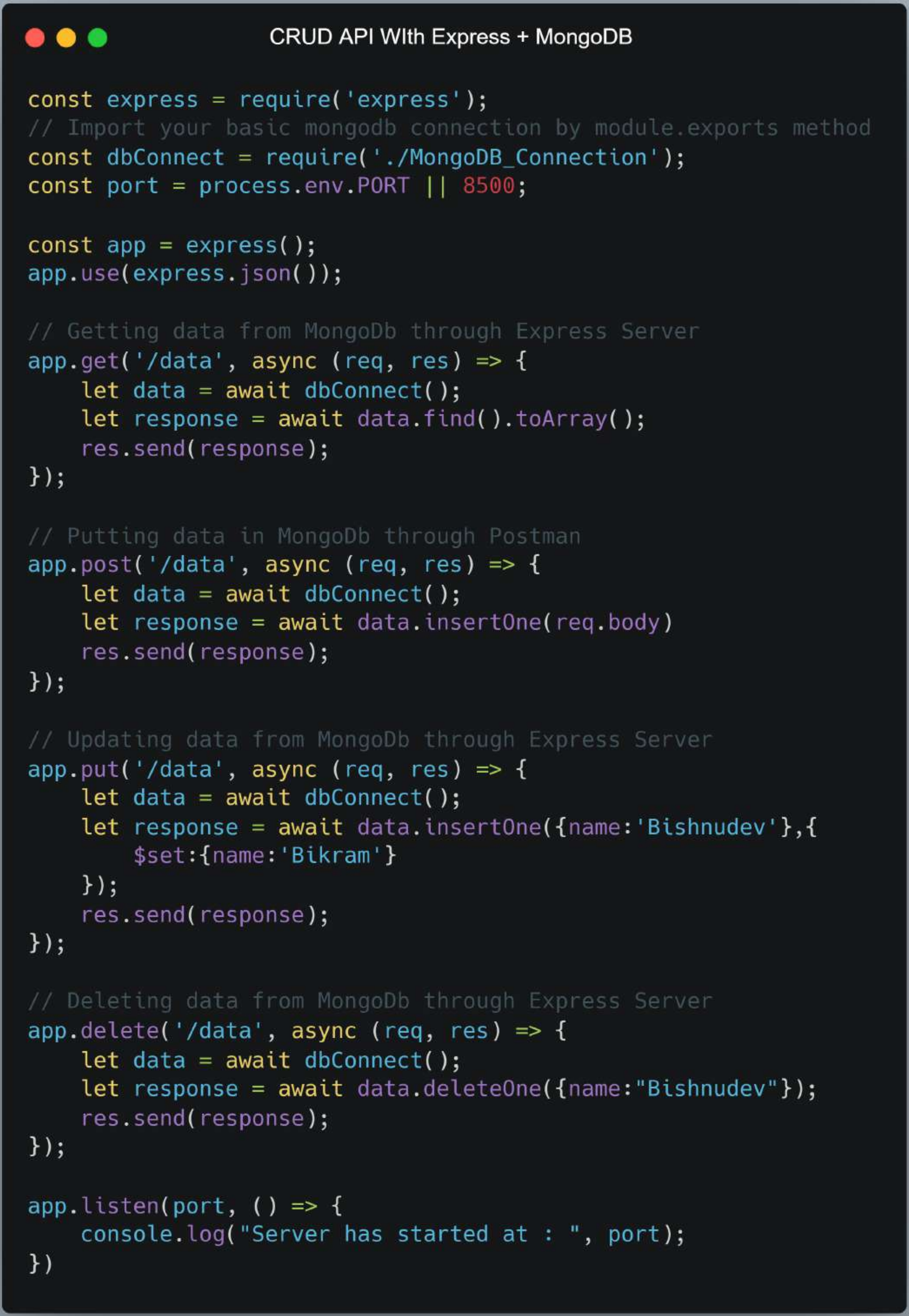


Connect MongoDB with Node

```
const { MongoClient } = require('mongodb');
const url = 'mongodb://localhost:27017';
const database = 'Node-Tut';
const client = new MongoClient(url);

// 1st Method

async function getData(){
    let result = await client.connect();
    let db = result.db(database);
    let collection = db.collection('nodejs');
    let response = await collection.find({}).toArray();
    console.log(response);
}

getData();

// 2nd Method

async function getData(){
    let result = await client.connect();
    let db = result.db(database);
    return db.collection('nodejs');
}

getData.then((data)=>{
    data.find({}).toArray().then((response)=>{
        console.log(response);
    })
})
```

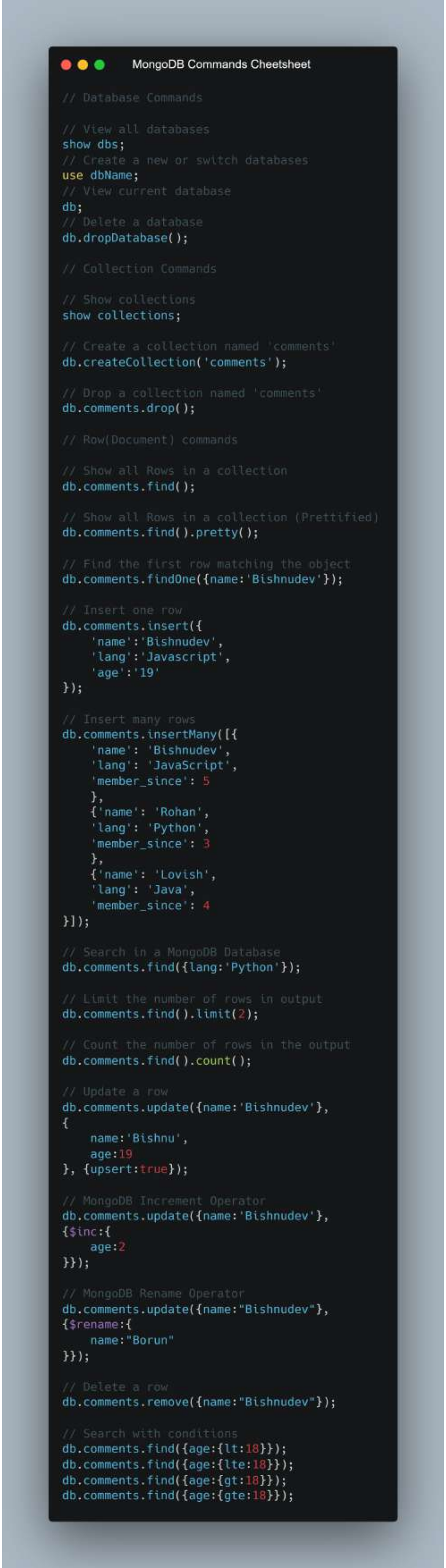


```
const { MongoClient } = require('mongodb');
const url = 'mongodb://localhost:27017';
const database = 'Node-Tut';
const client = new MongoClient(url);

async function dbConnect(){
    let result = await client.connect();
    let db = result.db(database);
    return db.collection('nodejs');
}

const readData = async () => {
    let data = await dbConnect();
    // Read all data
    let response = await data.find({}).toArray();
    // Read a single data by object like id,name
    let response2 = await data.find({name:'Austin'}).toArray();
}

const insertData = async () => {
    let data = await dbConnect();
    // Insert one data
    let response = await data.insertOne({name:"Bishnudev",age:19,hobby:"Gaming"});
    // Insert datas or many data
    let response2 = await data.insertMany({name:"Bishnudev",age:19,hobby:"Gaming"},
{name:"Austin",age:21,hobby:"Writer"},{name:"Alex",age:23,hobby:"Developer"});
}

const deleteData = async () => {
    let data = await dbConnect();
    // Delete a data
    let response = await data.deleteOne({name:'Bishnudev'});
    // Delete many data
    let response2 = await data.deleteMany({name:'Bishnudev'},{age:23});
}

const updateData = async () => {
    let data = await dbConnect();
    let response = await data.updateOne(
        {name:'Bishnudev'},{
            $set: {name:'Rahul'}
        }
    )
}

readData();
insertData();
deleteData();
updateData();
```



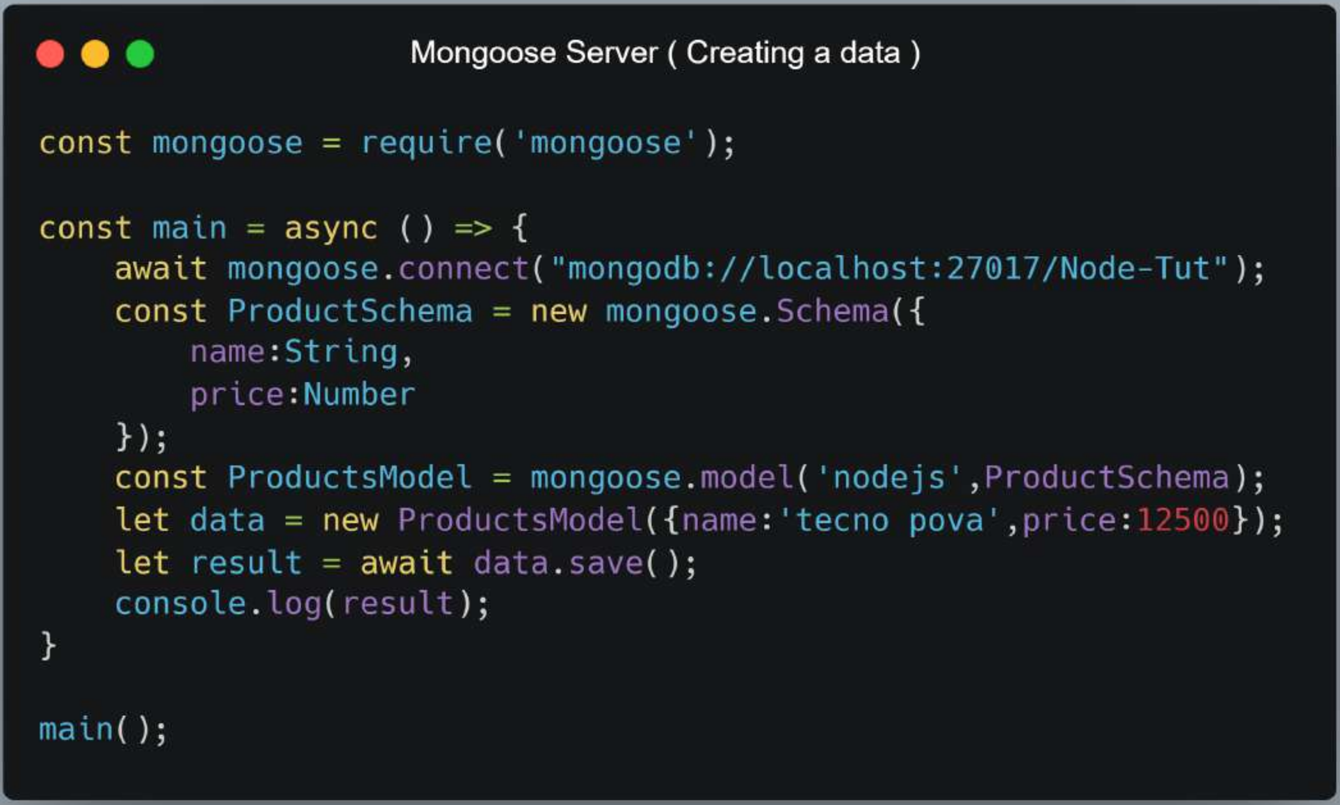
CRUD API WIth Express + MongoDB

```
const express = require('express');
// Import your basic mongodb connection by module.exports method
const dbConnect = require('./MongoDB_Connection');
const port = process.env.PORT || 8500;

const app = express();
app.use(express.json());

// Getting data from MongoDb through Express Server
app.get('/data', async (req, res) => {
    let data = await dbConnect();
    let response = await data.find().toArray();
    res.send(response);
});

// Putting data in MongoDb through Postman
app.post('/data', async (req, res) => {
    let data = await dbConnect();
    let response = await data.insertOne(req.body)
    res.send(response);
});

// Updating data from MongoDb through Express Server
app.put('/data', async (req, res) => {
    let data = await dbConnect();
    let response = await data.insertOne({name:'Bishnudev'},{
        $set:{name:'Bikram'}
    });
    res.send(response);
});

// Deleting data from MongoDb through Express Server
app.delete('/data', async (req, res) => {
    let data = await dbConnect();
    let response = await data.deleteOne({name:"Bishnudev"});
    res.send(response);
});

app.listen(port, () => {
    console.log("Server has started at : ", port);
})
```



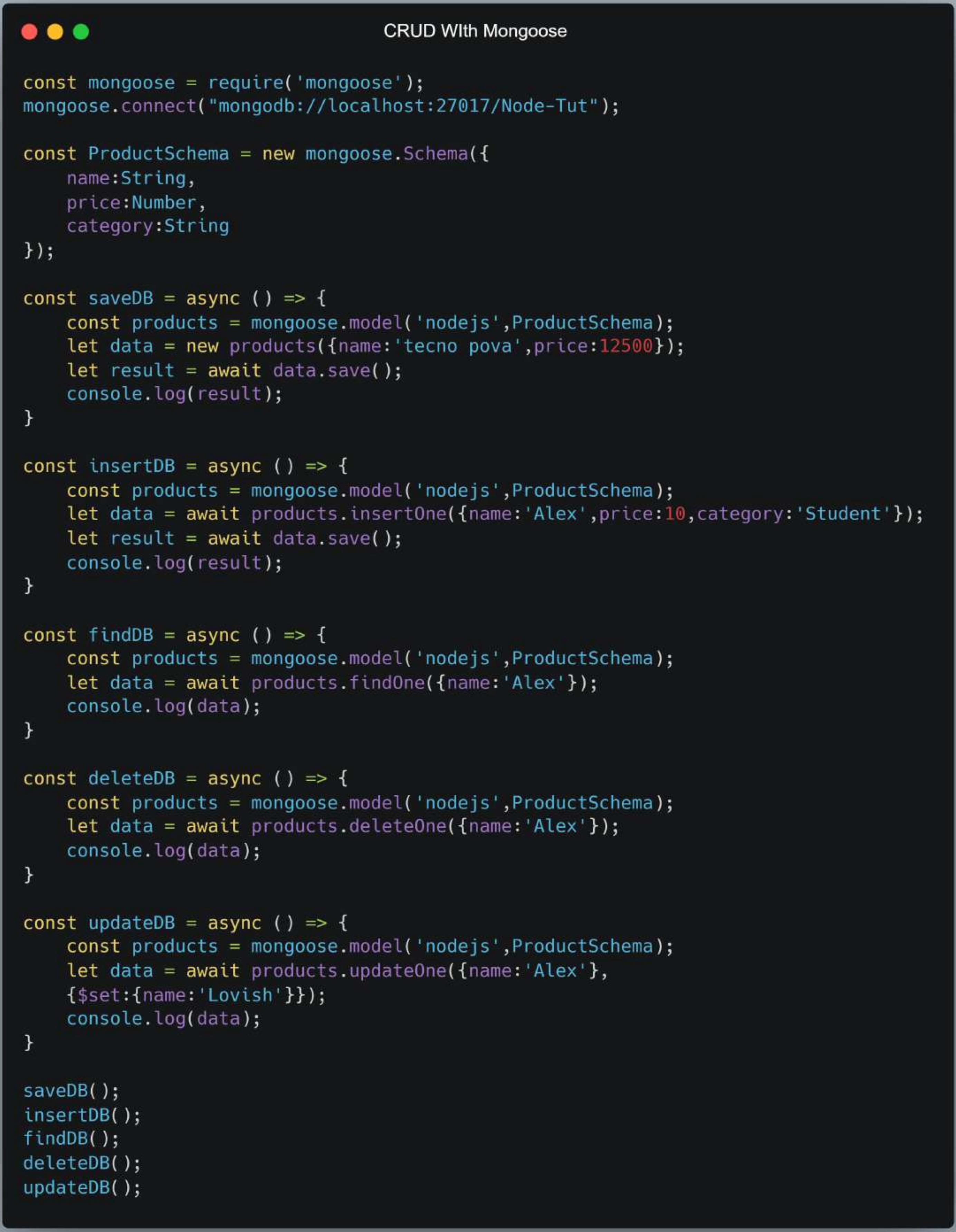
MongoDB Commands Cheetsheet

```
// Database Commands

// View all databases
show dbs;
// Create a new or switch databases
use dbName;
// View current database
db;
// Delete a database
db.dropDatabase();

// Collection Commands

// Show collections
show collections;

// Create a collection named 'comments'
db.createCollection('comments');

// Drop a collection named 'comments'
db.comments.drop();

// Row(Document) commands

// Show all Rows in a collection
db.comments.find();

// Show all Rows in a collection (Prettified)
db.comments.find().pretty();

// Find the first row matching the object
db.comments.findOne({name:'Bishnudev'});

// Insert one row
db.comments.insert({
    'name':'Bishnudev',
    'lang':'Javascript',
    'age':'19'
});

// Insert many rows
db.comments.insertMany([{
    'name': 'Bishnudev',
    'lang': 'JavaScript',
    'member_since': 5
    },
    {'name': 'Rohan',
    'lang': 'Python',
    'member_since': 3
    },
    {'name': 'Lovish',
    'lang': 'Java',
    'member_since': 4
}]);

// Search in a MongoDB Database
db.comments.find({lang:'Python'});

// Limit the number of rows in output
db.comments.find().limit(2);

// Count the number of rows in the output
db.comments.find().count();

// Update a row
db.comments.update({name:'Bishnudev'},
{
    name:'Bishnu',
    age:19
}, {upsert:true});

// MongoDB Increment Operator
db.comments.update({name:'Bishnudev'},
{$inc:{
    age:2
}});

// MongoDB Rename Operator
db.comments.update({name:"Bishnudev"},
{$rename:{
    name:"Borun"
}});

// Delete a row
db.comments.remove({name:"Bishnudev"});

// Search with conditions
db.comments.find({age:{lt:18}});
db.comments.find({age:{lte:18}});
db.comments.find({age:{gt:18}});
db.comments.find({age:{gte:18}});
```



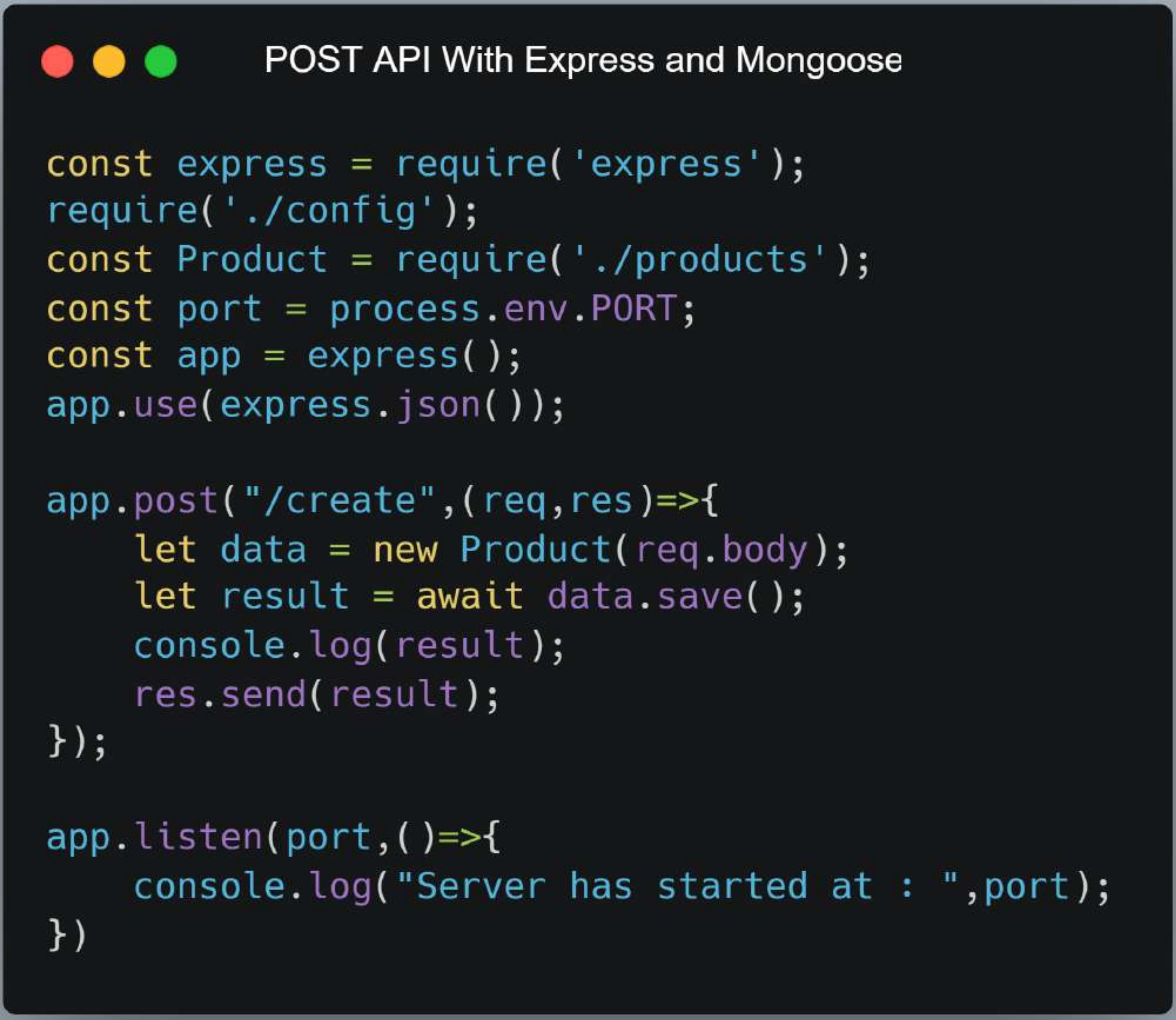
Mongoose Server (Creating a data)

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

const main = async () => {
    await mongoose.connect("mongodb://localhost:27017/Node-Tut");
    const ProductSchema = new mongoose.Schema({
        name:String,
        price:Number
    });
    const ProductsModel = mongoose.model('nodejs',ProductSchema);
    let data = new ProductsModel({name:'tecno pova',price:12500});
    let result = await data.save();
    console.log(result);
}

main();
```



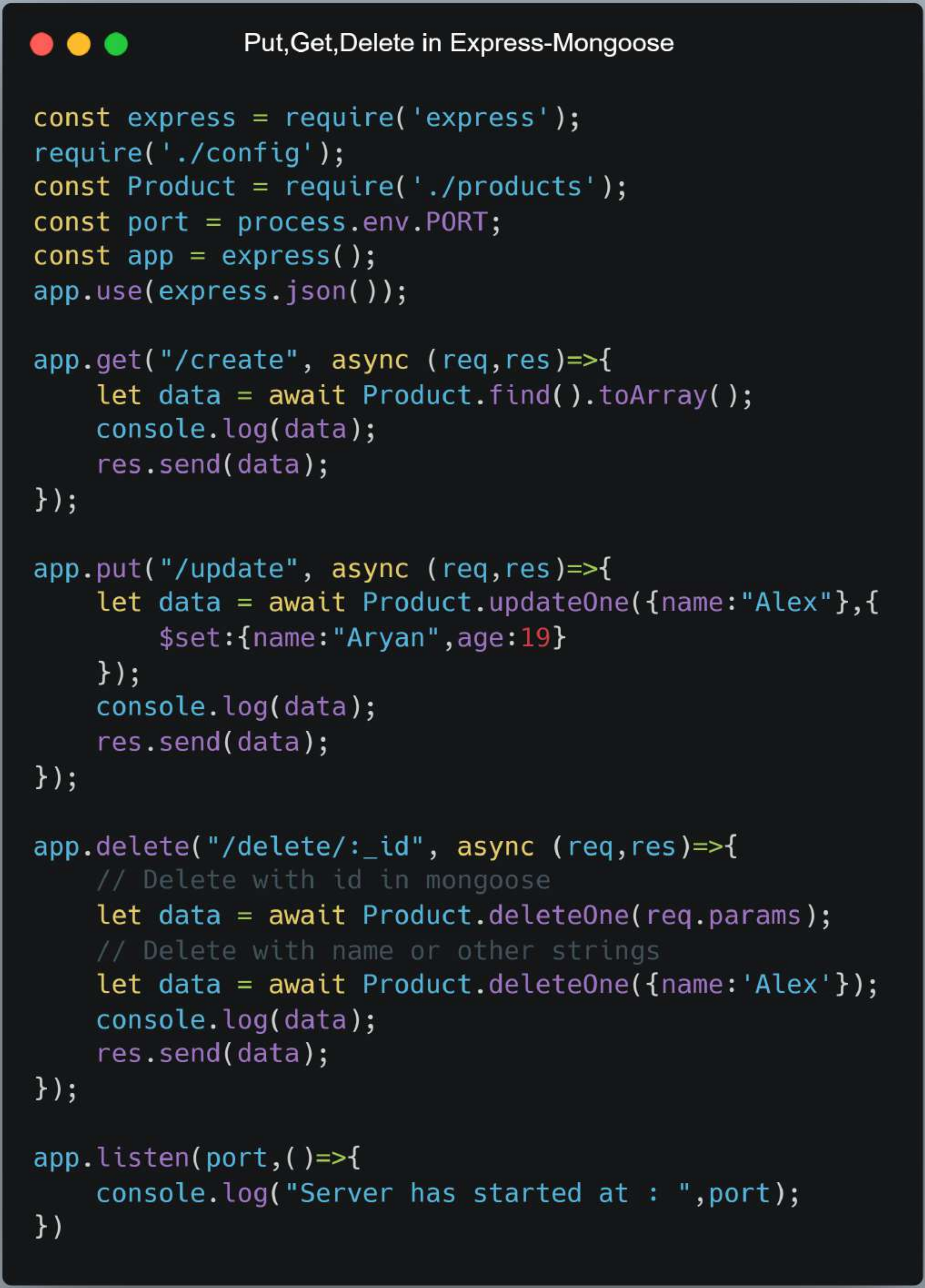
CRUD WIth Mongoose

```javascript
const mongoose = require('mongoose');
mongoose.connect("mongodb://localhost:27017/Node-Tut");

const ProductSchema = new mongoose.Schema({
    name:String,
    price:Number,
    category:String
});

const saveDB = async () => {
    const products = mongoose.model('nodejs',ProductSchema);
    let data = new products({name:'tecno pova',price:12500});
    let result = await data.save();
    console.log(result);
}

const insertDB = async () => {
    const products = mongoose.model('nodejs',ProductSchema);
    let data = await products.insertOne({name:'Alex',price:10,category:'Student'});
    let result = await data.save();
    console.log(result);
}

const findDB = async () => {
    const products = mongoose.model('nodejs',ProductSchema);
    let data = await products.findOne({name:'Alex'});
    console.log(data);
}

const deleteDB = async () => {
    const products = mongoose.model('nodejs',ProductSchema);
    let data = await products.deleteOne({name:'Alex'});
    console.log(data);
}

const updateDB = async () => {
    const products = mongoose.model('nodejs',ProductSchema);
    let data = await products.updateOne({name:'Alex'},
    {$set:{name:'Lovish'}});
    console.log(data);
}

saveDB();
insertDB();
findDB();
deleteDB();
updateDB();
```



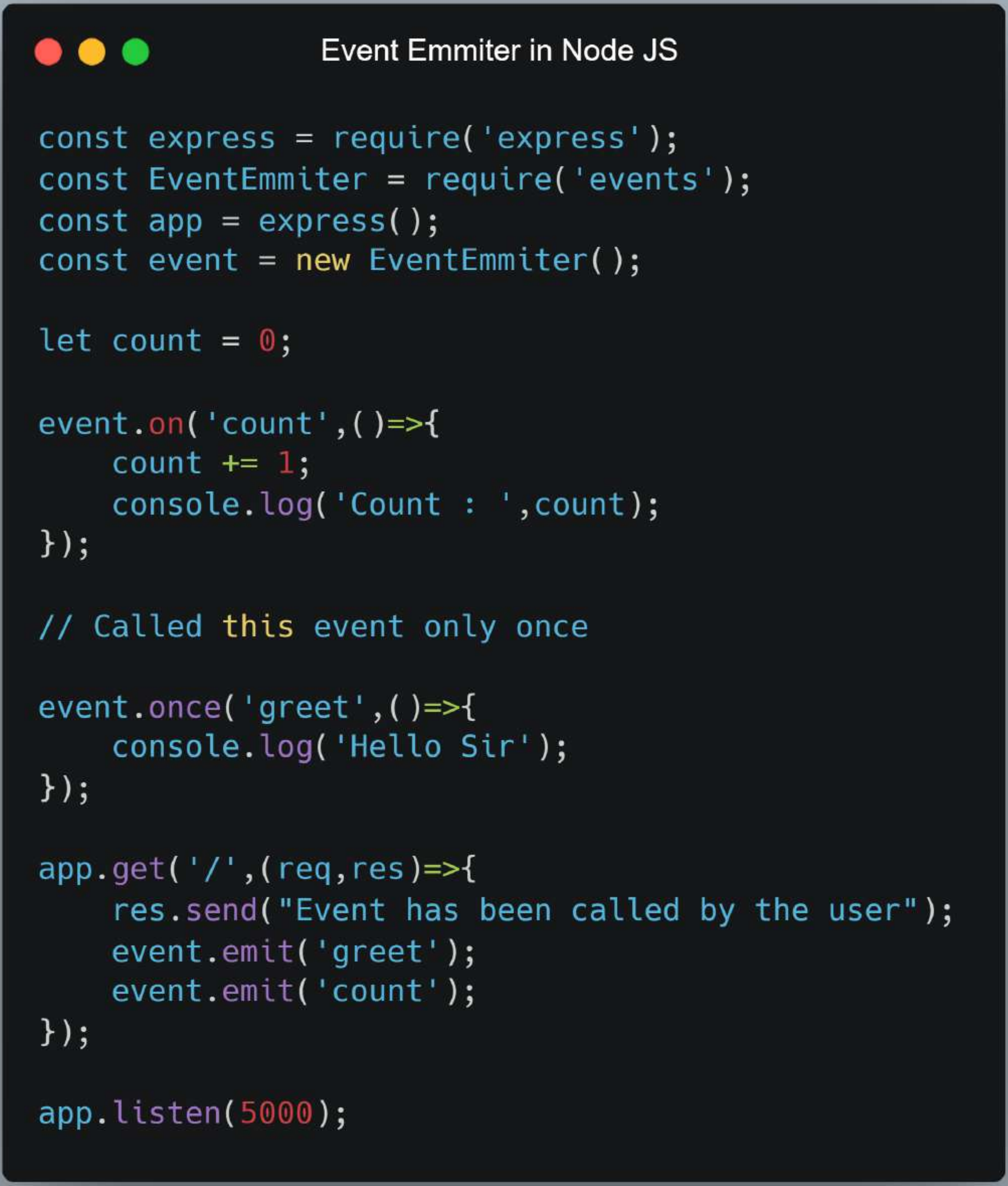
POST API With Express and Mongoose

```
const express = require('express');
require('./config');
const Product = require('./products');
const port = process.env.PORT;
const app = express();
app.use(express.json());

app.post("/create",(req,res)=>{
    let data = new Product(req.body);
    let result = await data.save();
    console.log(result);
    res.send(result);
});

app.listen(port,()=>{
    console.log("Server has started at : ",port);
})
```

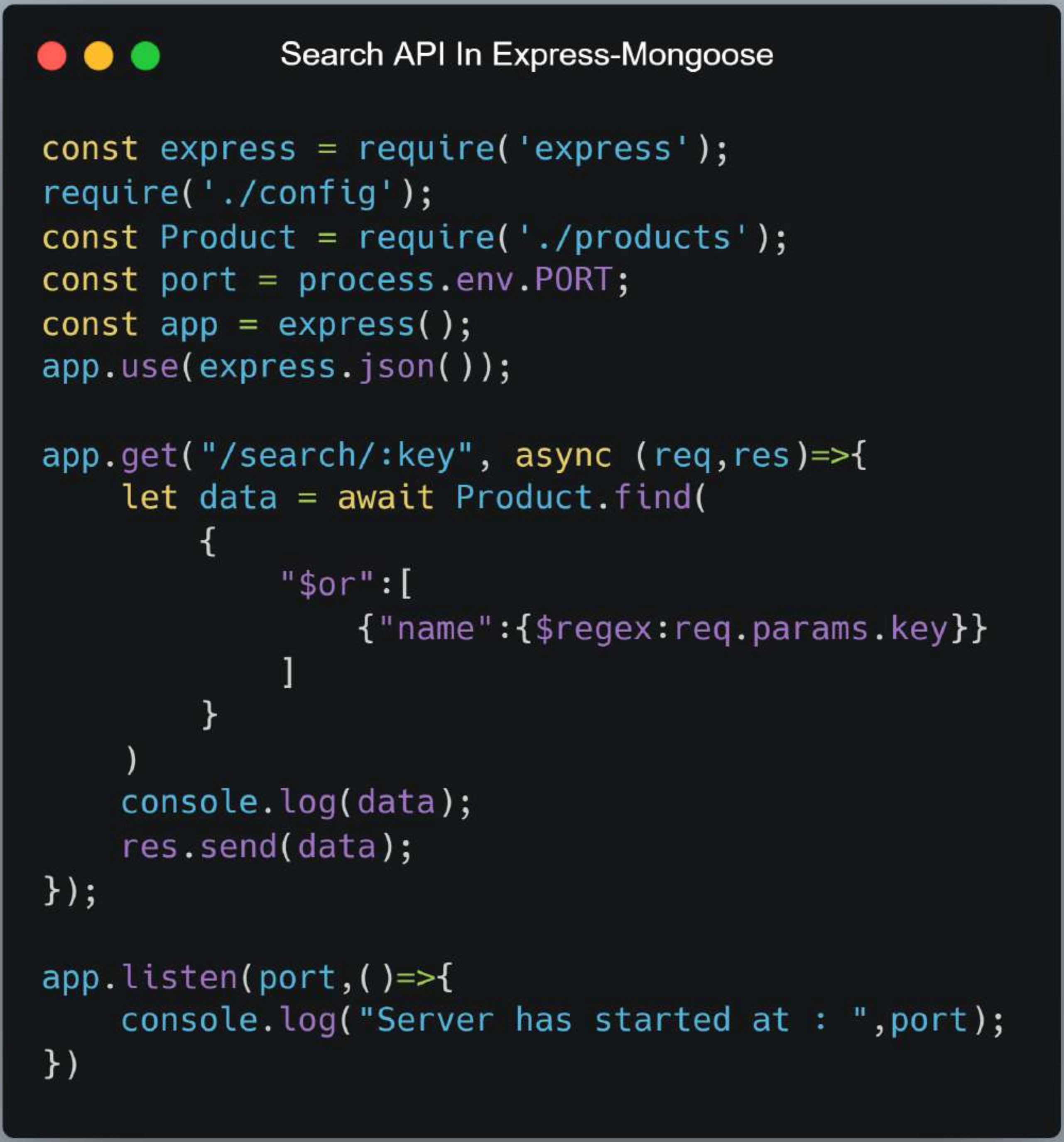


Put,Get,Delete in Express-Mongoose

```javascript
const express = require('express');
require('./config');
const Product = require('./products');
const port = process.env.PORT;
const app = express();
app.use(express.json());

app.get("/create", async (req,res)=>{
    let data = await Product.find().toArray();
    console.log(data);
    res.send(data);
});

app.put("/update", async (req,res)=>{
    let data = await Product.updateOne({name:"Alex"},{
        $set:{name:"Aryan",age:19}
    });
    console.log(data);
    res.send(data);
});

app.delete("/delete/:_id", async (req,res)=>{
    // Delete with id in mongoose
    let data = await Product.deleteOne(req.params);
    // Delete with name or other strings
    let data = await Product.deleteOne({name:'Alex'});
    console.log(data);
    res.send(data);
});

app.listen(port,()=>{
    console.log("Server has started at : ",port);
})
```

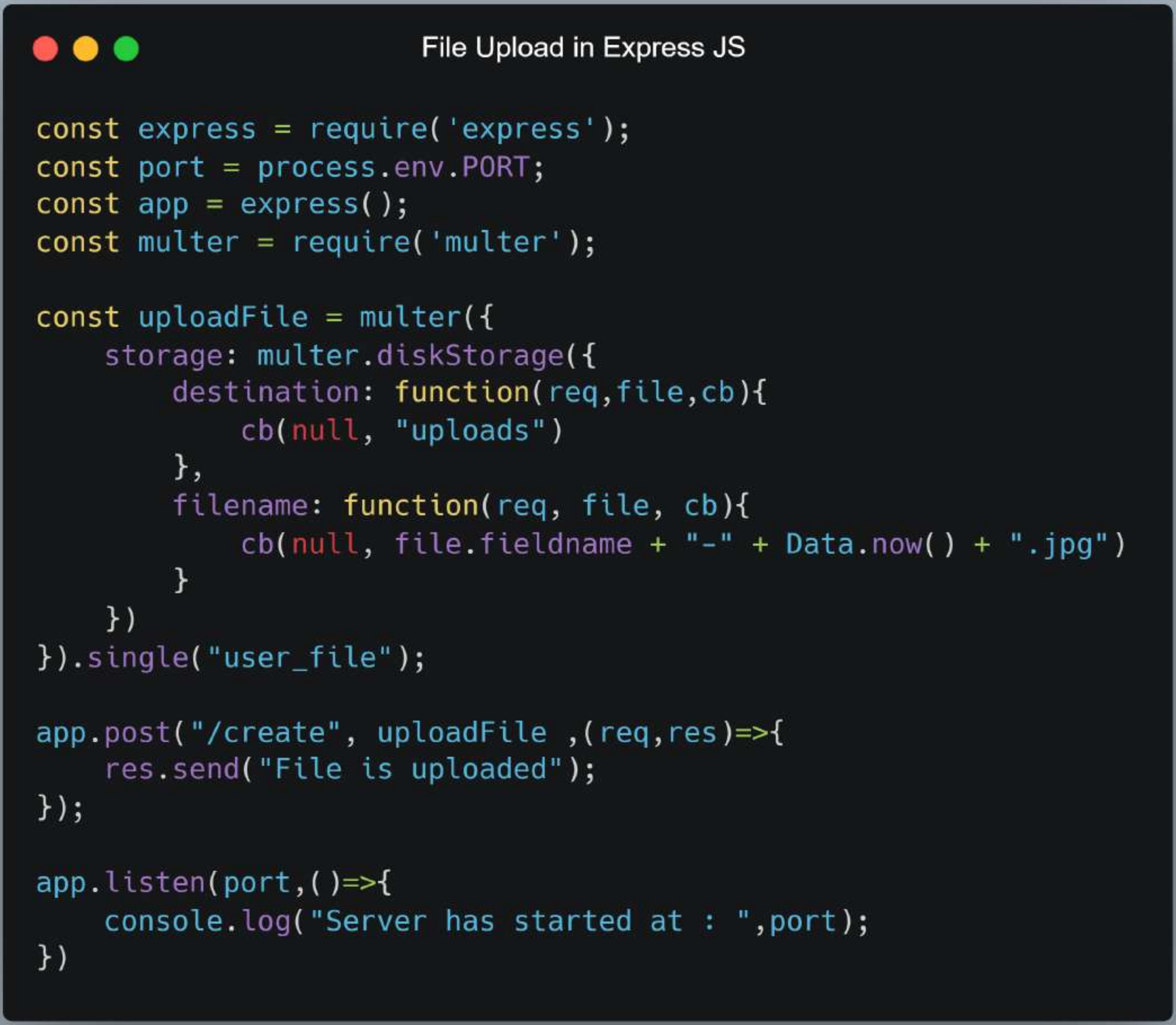


```javascript
const express = require('express');
const EventEmmiter = require('events');
const app = express();
const event = new EventEmmiter();

let count = 0;

event.on('count',()=>{
    count += 1;
    console.log('Count : ',count);
});

// Called this event only once

event.once('greet',()=>{
    console.log('Hello Sir');
});

app.get('/',(req,res)=>{
    res.send("Event has been called by the user");
    event.emit('greet');
    event.emit('count');
});

app.listen(5000);
```



Search API In Express-Mongoose

```
const express = require('express');
require('./config');
const Product = require('./products');
const port = process.env.PORT;
const app = express();
app.use(express.json());

app.get("/search/:key", async (req,res)=>{
    let data = await Product.find(
        {
            "$or":[
                {"name":{$regex:req.params.key}}
            ]
        }
    )
    console.log(data);
    res.send(data);
});

app.listen(port,()=>{
    console.log("Server has started at : ",port);
})
```



File Upload in Express JS

```
const express = require('express');
const port = process.env.PORT;
const app = express();
const multer = require('multer');

const uploadFile = multer({
    storage: multer.diskStorage({
        destination: function(req,file,cb){
            cb(null, "uploads")
        },
        filename: function(req, file, cb){
            cb(null, file.fieldname + "-" + Data.now() + ".jpg")
        }
    })
}).single("user_file");

app.post("/create", uploadFile ,(req,res)=>{
    res.send("File is uploaded");
});

app.listen(port,()=>{
    console.log("Server has started at : ",port);
})
```