

**Name: Rohan Lalchandani**

**Class: D15A      Roll no: 25**

## **Lab Exercise 7**

**Title:** NodeJs and Express

### **Objectives:**

Develop a product management API using Express.js that allows users to view and search for products in a catalog. The application should provide different endpoints for retrieving product information, including the ability to view all products, view details for a specific product by ID, and search for products based on query parameters.

### **Theory:**

#### **1. What is Express.js, and why is it used in Node.js applications?**

**Express.js** is a web application framework for **Node.js** that simplifies the process of building robust web applications and APIs. It provides a minimalistic set of features while allowing developers to add various middleware as needed. Here's why Express.js is widely used in Node.js applications:

1. **Simplified Routing:** Express makes it easier to define routes for different HTTP methods (GET, POST, etc.), allowing better organization of application logic.
2. **Middleware Support:** Express has built-in middleware handling for tasks like parsing request bodies, handling cookies, or serving static files, and it can integrate third-party middleware to extend functionality.
3. **Asynchronous Nature:** Since Express is built on Node.js, it supports asynchronous code execution, which is crucial for handling non-blocking operations in web applications.
4. **Fast and Lightweight:** Express is lightweight with minimal overhead, making it suitable for high-performance applications.
5. **Scalability:** Express allows developers to build scalable web applications by providing flexibility in structuring applications and organizing modules.

## 2. How are dynamic routes handled in Express.js?

In **Express.js**, dynamic routes allow the server to respond to requests with varying parameters. This is handled by defining route patterns where parts of the URL are treated as variables. These variables are accessible through the `req.params` object.

Example:

```
const express = require('express');
const app = express();

// Dynamic route
app.get('/user/:id', (req, res) => {
  const userId = req.params.id; // Access dynamic route parameter
  res.send(`User ID: ${userId}`);
});

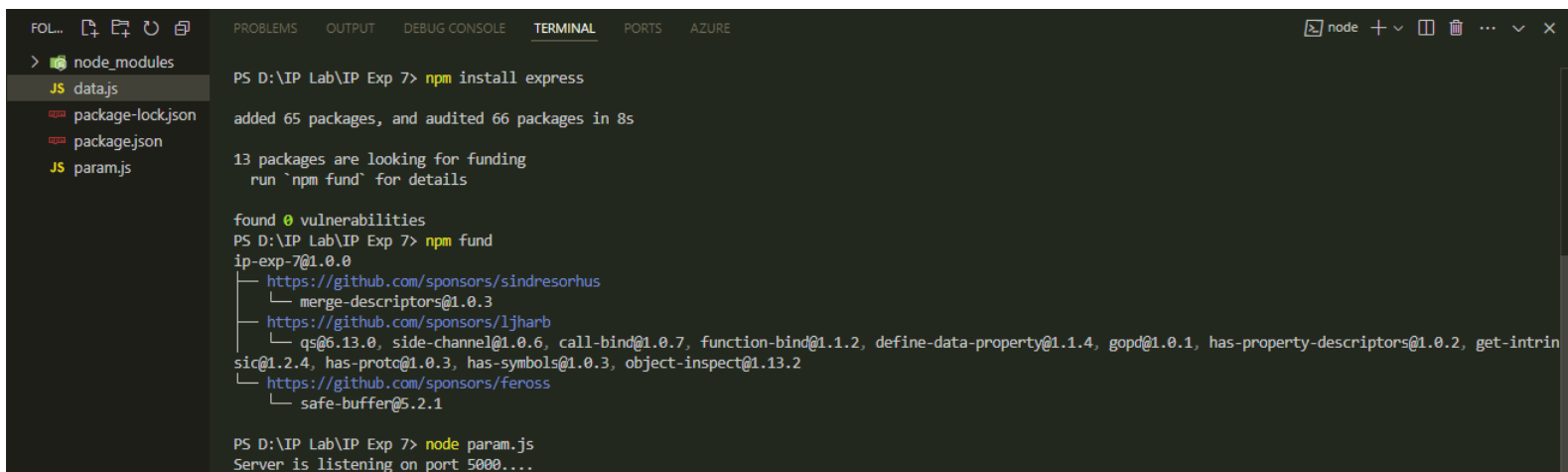
app.listen(3000, () => {
  console.log('Server is running on port 3000');
});
```

### Code and Screenshot:

GitHub Repo for link:

<https://github.com/Rohan-Lalchandani08/IP-Exp7>

Screenshot of the Client and Server Execution:



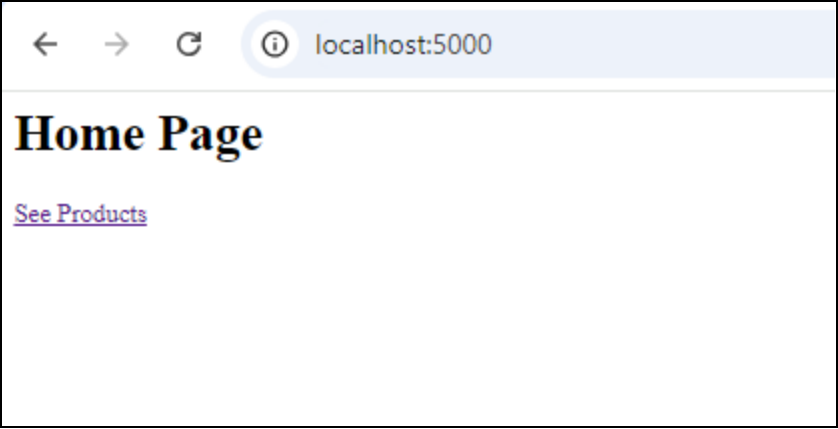
```
FOL... node_modules
JS data.js
package-lock.json
package.json
JS param.js

PS D:\IP Lab\IP Exp 7> npm install express
added 65 packages, and audited 66 packages in 8s

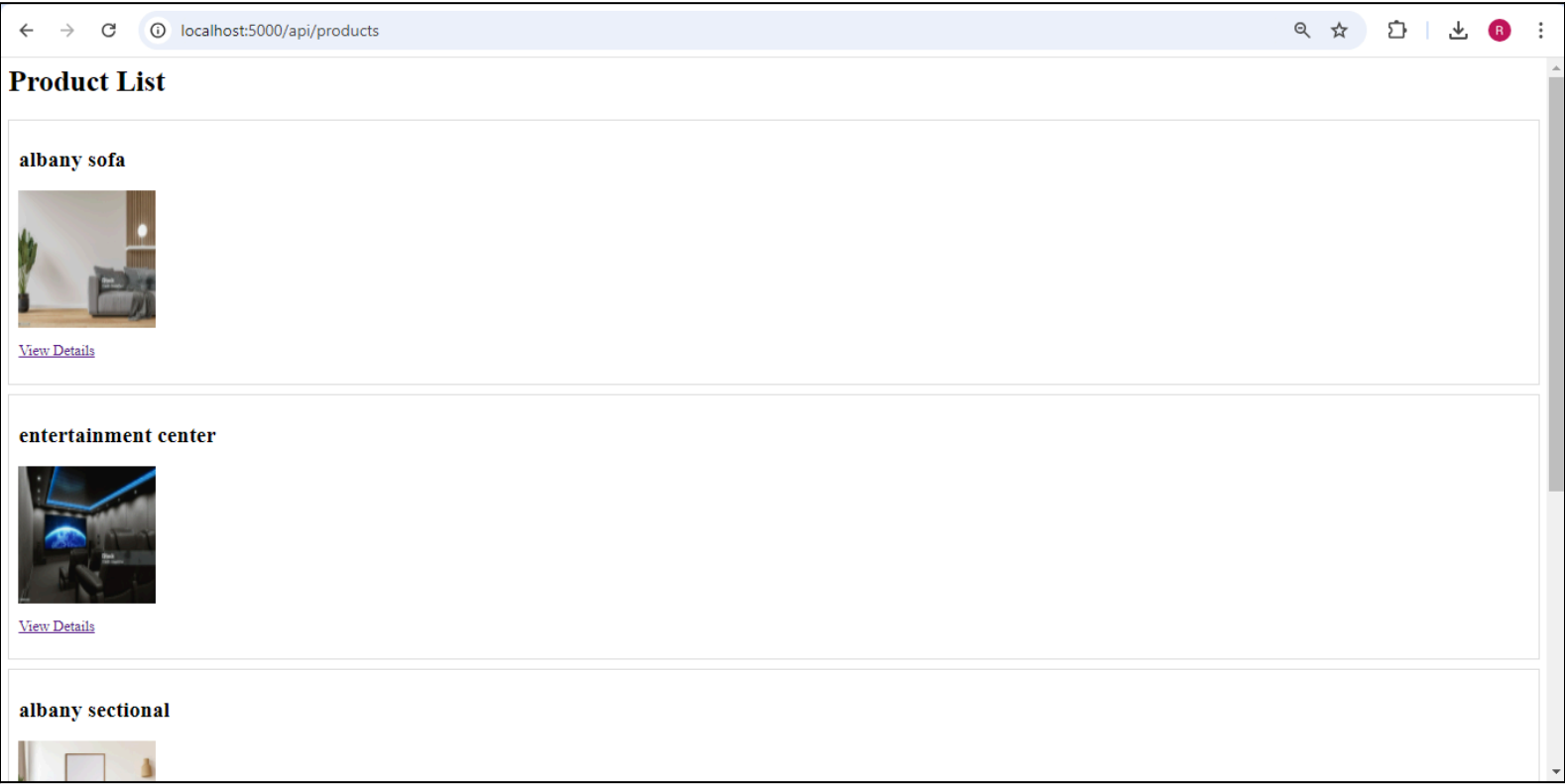
13 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS D:\IP Lab\IP Exp 7> npm fund
ip-exp-7@1.0.0
├─ https://github.com/sponsors/sindresorhus
├─ merge-descriptors@1.0.3
├─ https://github.com/sponsors/ljharb
├─ qs@6.13.0, side-channel@1.0.6, call-bind@1.0.7, function-bind@1.1.2, define-data-property@1.1.4, gopd@1.0.1, has-property-descriptors@1.0.2, get-intrins
sic@1.2.4, has-proto@1.0.3, has-symbols@1.0.3, object-inspect@1.13.2
├─ https://github.com/sponsors/feross
└─ safe-buffer@5.2.1

PS D:\IP Lab\IP Exp 7> node param.js
Server is listening on port 5000....
```



View all products:




Search by name query:

http://localhost:5000/api/v1/query?search=albany&limit=2

localhost:5000/api/v1/query?search=albany&limit=2


# Search Results

albany sofa



[View Details](#)

albany sectional

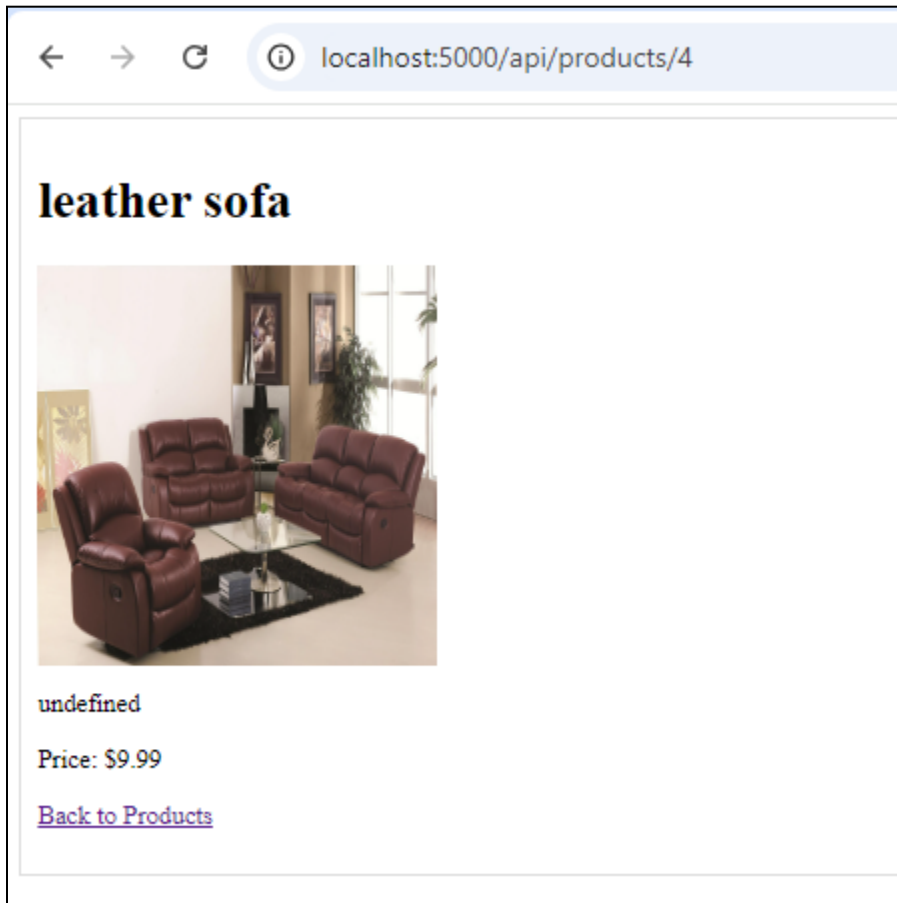


[View Details](#)

[Back to Products](#)

**Search by id query:**

http://localhost:5000/api/products/4



**Product not found message:**

