Experiment 7:

Install Express (npm install express). Set up a basic server that responds with "Hello, Express!" at the root endpoint (GET /). Create a REST API. Implement endpoints for a Product resource: GET : Returns a list of products. POST : Adds a new product. GET /:id: Returns details of a specific product. PUT /:id: Updates an existing product. DELETE /:id: Deletes a product. Add middleware to log requests to the console. Use express.json() to parse incoming JSON payloads.

Step-by-Step Process:

1. **Install Node.js and npm**

   o If you don't have Node.js installed, go to the official website: Node.js and install the LTS version. This will automatically install npm (Node Package Manager) as well.

1. create a folder expressjs
2. cd expressjs
3. npm init -y
4. npm install express or npm i express
5. npm i -D nodemon
6. now create a server.js file
7. under server.js file copy and paste the below code

server.js

```
import express, { json } from 'express'; // Use require instead of import
const app = express();
app.use(json()); // This is required to parse JSON data

app.get('/', (req, res) => {
    res.send('Hello, World!');
});
const products = [
    {
        id: 1,
        name: "mi"
    },
    {
        id: 2,
        name: "iphone"
    },
    {
```

```javascript
        id: 3,
        name: "oppo"
    }
]
app.get('/products', (req, res) => {
    res.send(products);
});

app.get('/products/:id', (req, res) => {
    const newData = products.filter(item => item.id.toString() ===
req.params.id)
    res.send(newData);
});

app.post('/addproducts', (req, res) => {
    const { id, name } = req.body;
    const newProduct = { id, name };
    products.push(newProduct);
    res.send(newProduct);
});

app.put('/updateproducts/:id', (req, res) => {
    const product = products.find(item => item.id.toString() ===
req.params.id);
    Object.assign(product, req.body);
    res.send(product);
});

app.delete('/deleteproducts/:id', (req, res) => {
    const index = products.findIndex(item => item.id.toString() ===
req.params.id);
    const deletedProduct = products.splice(index, 1);
    res.send(deletedProduct);
});

app.listen(3000, () => {
    console.log('Server running on http://localhost:3000');
});
```

Under package.json ensure all the following things are there or not
```json
{
  "name": "expressjs",
  "version": "1.0.0",
  "description": "",
  "license": "ISC",
  "author": "",
  "type": "module",
```

```
      "main": "index.js",
      "scripts": {
       "test": "echo \"Error: no test specified\" && exit 1",
       "start": "node server.js",
       "server": "nodemon server.js"
      },
      "dependencies": {
       "express": "^4.21.2"
      },
      "devDependencies": {
       "nodemon": "^3.1.9"
      }
     }
```

o/p:

npm run server

next install the postman

Install Postman

1. Visit the [Postman website](#) and download the application for your operating system (Windows, Mac, or Linux).

2. Install Postman by following the prompts.

Next click on new there select http after selecting http here you can see the all apis like get, get/id, put/id, post, delete/id…etc
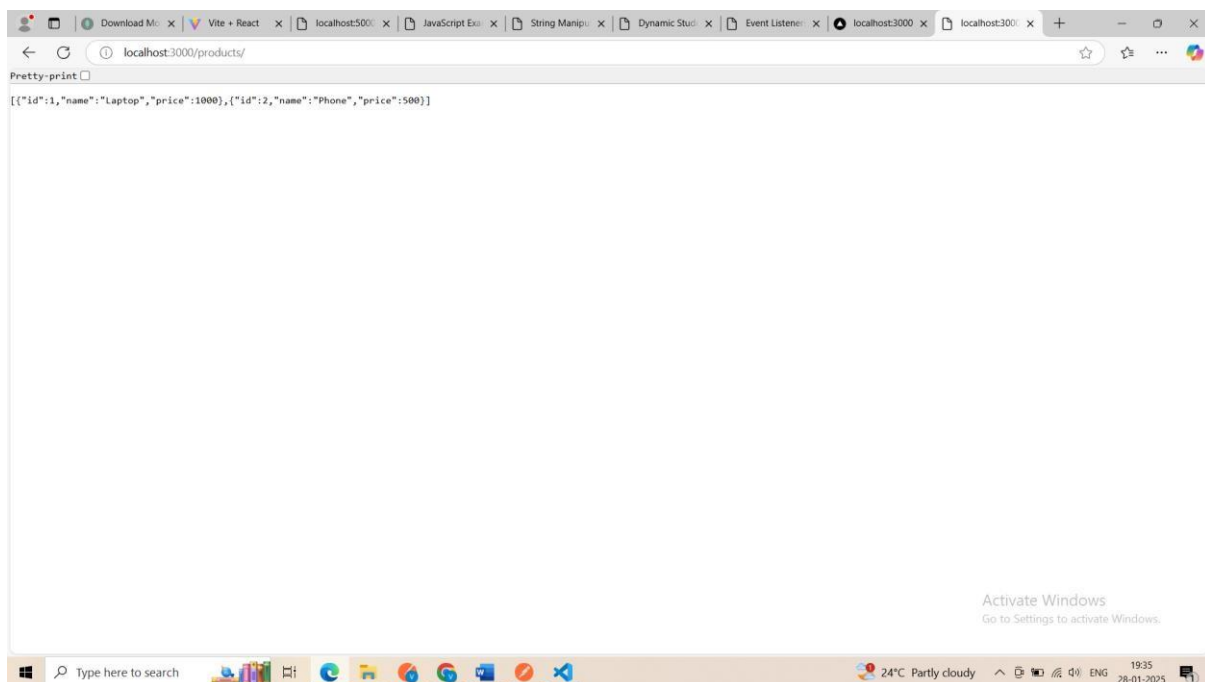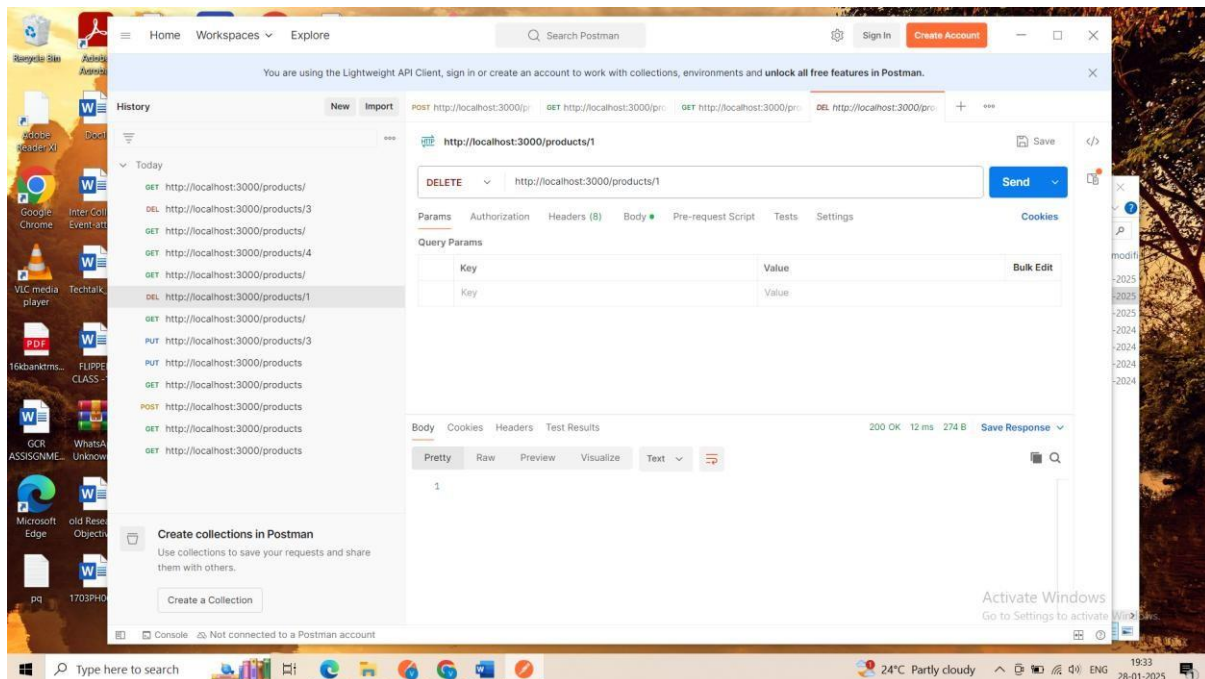
Get: http://localhost:3000/products

Post: http://localhost:3000/addproducts

Put/id: http://localhost:3000/updateproducts/2

Get/id: http://localhost:3000/products/2

Delete: http://localhost:3000/deleteproducts/1

Json data Experiment

```json
{
    "id": 1,
    "name": "Laptop"
}
```

```
Mode              LastWriteTime        Length Name
----              -------------        ------ ----
d-----       28-01-2025    14:11              express-product-api


PS D:\New folder\New Volume\FSD JS\fsd lab\7> cd .\express-product-api\
PS D:\New folder\New Volume\FSD JS\fsd lab\7\express-product-api> curl http://localhost:3000/products/1


StatusCode        : 200
StatusDescription : OK
Content           : {"id":1,"name":"Laptop","price":1000}
RawContent        : HTTP/1.1 200 OK
                    Connection: keep-alive
                    Keep-Alive: timeout=5
                    Content-Length: 37
                    Content-Type: application/json; charset=utf-8
                    Date: Tue, 28 Jan 2025 14:06:55 GMT
                    ETag: W/"25-bYHbDBiVbRFgXIJuIKeZ...
Forms             : {}
Headers           : {[Connection, keep-alive], [Keep-Alive, timeout=5], [Content-Length, 37], [Content-Type,
                    application/json; charset=utf-8]...}
Images            : {}
InputFields       : {}
Links             : {}
ParsedHtml        : mshtml.HTMLDocumentClass
RawContentLength  : 37


PS D:\New folder\New Volume\FSD JS\fsd lab\7\express-product-api>
```

1. Visit the Postman website and download the application for your operating system (Windows, Mac, or Linux).

2. Install Postman by following the prompts.

Step 2: Set Up Your Express Server