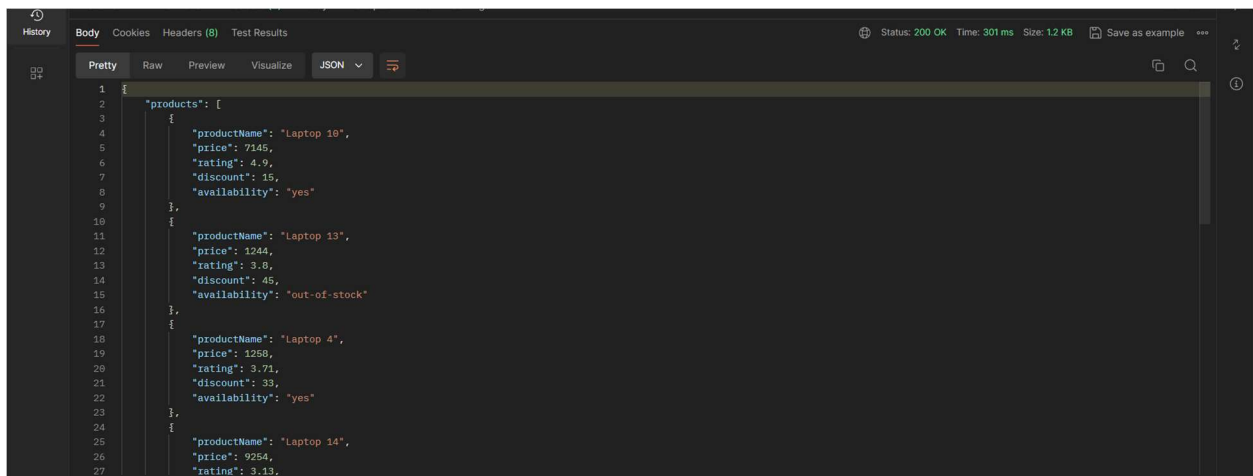


RollNo:HU21CSEN0100531

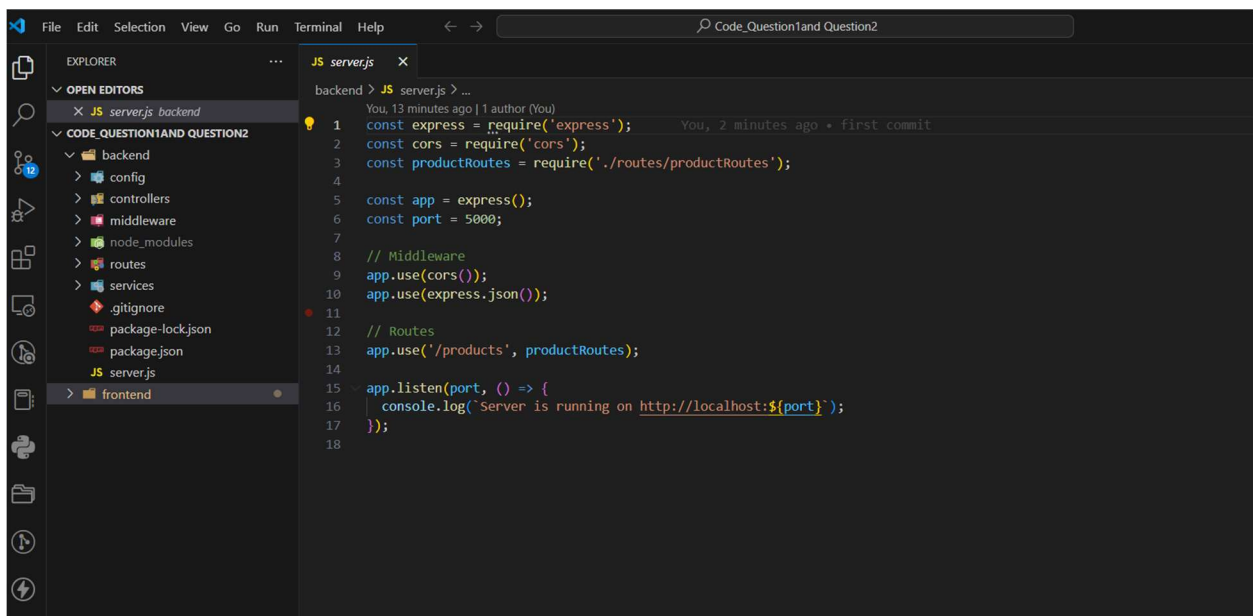
The screenshot displays the Afformate web application interface. At the top, there's a navigation bar with links for Home, Workspaces, API Network, and a Search Postman button. Below this, the main header shows the current workspace 'Afformate' and a 'New Request' button. The interface is divided into several sections: 'Params', 'Authorization', 'Headers (8)', 'Body', 'Scripts', 'Tests', and 'Settings'. The 'Body' tab is selected, showing a JSON request body for a POST request to 'http://20.244.56.144/test/auth'. The request body contains fields like 'companyName', 'clientId', 'clientSecret', 'ownerName', 'ownerEmail', and 'rollNo'. Below the request, the 'Body' tab shows the response, which is a JSON object containing 'token_type', 'access_token', and 'expires_in'. The status bar at the bottom indicates the request was successful (201 Created) with a time of 93 ms and a size of 108 KB.

The screenshot displays the Afformate web application interface. At the top, there's a navigation bar with 'Home', 'Workspaces', and 'API Network' tabs. Below this, a 'New Request' form is visible, showing a GET request to 'http://localhost:5000/products?company=AMZ&category=Laptop&minPrice=1&maxPrice=10000&topN=10'. The 'Body' tab is selected, showing a JSON payload for a user. The bottom section shows the 'Body' tab with a JSON response for the products, including details like 'productName', 'price', 'rating', 'discount', and 'availability'.



```
1 {
2   "products": [
3     {
4       "productName": "Laptop 10",
5       "price": 7145,
6       "rating": 4.9,
7       "discount": 15,
8       "availability": "yes"
9     },
10    {
11      "productName": "Laptop 13",
12      "price": 1244,
13      "rating": 3.8,
14      "discount": 45,
15      "availability": "out-of-stock"
16    },
17    {
18      "productName": "Laptop 4",
19      "price": 1258,
20      "rating": 3.71,
21      "discount": 33,
22      "availability": "yes"
23    },
24    {
25      "productName": "Laptop 14",
26      "price": 9254,
27      "rating": 3.13,
```

Code

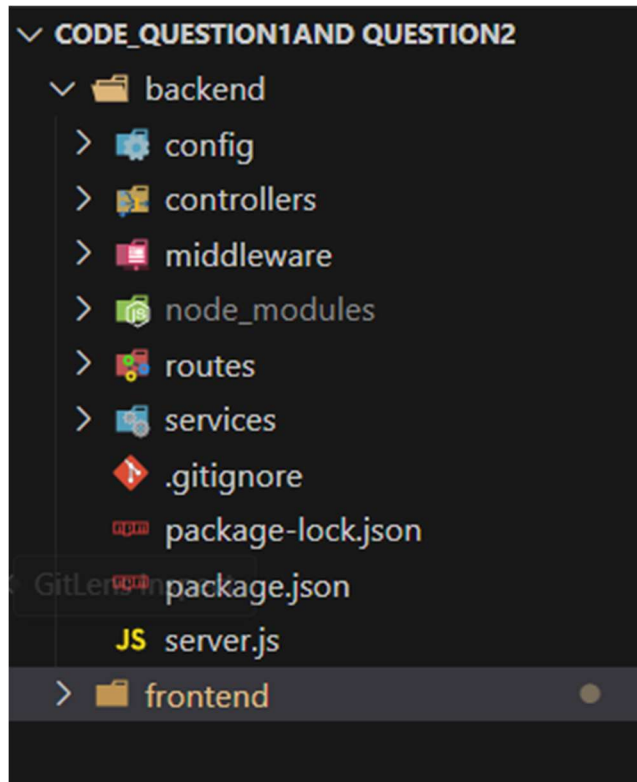


```
File Edit Selection View Go Run Terminal Help
Code_Question1and Question2

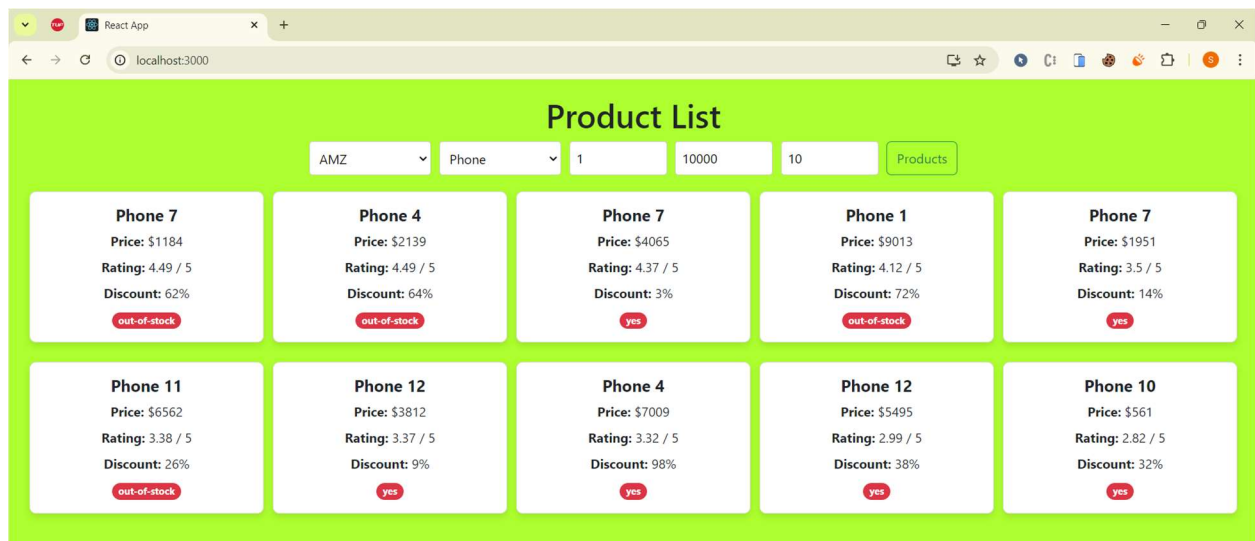
EXPLORER
  OPEN EDITORS
    JS server.js backend
  CODE QUESTION1AND QUESTION2
    backend
      config
      controllers
      middleware
      node_modules
      routes
      services
      .gitignore
      package-lock.json
      package.json
      JS server.js
    frontend

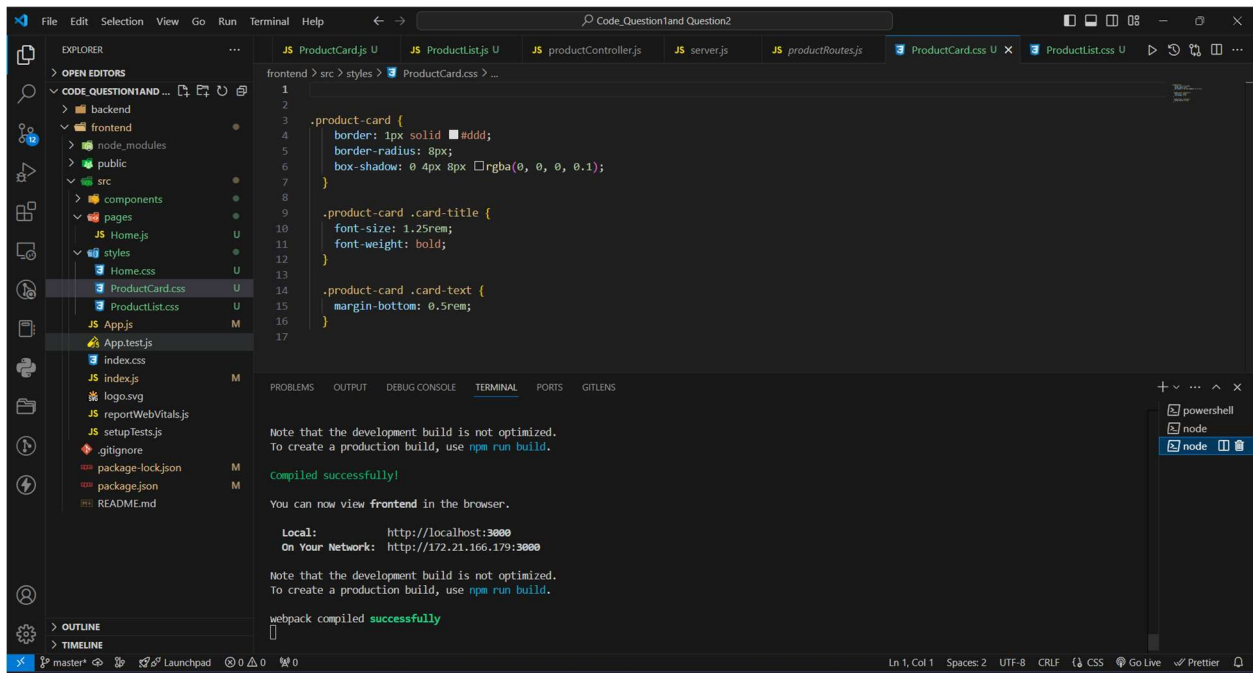
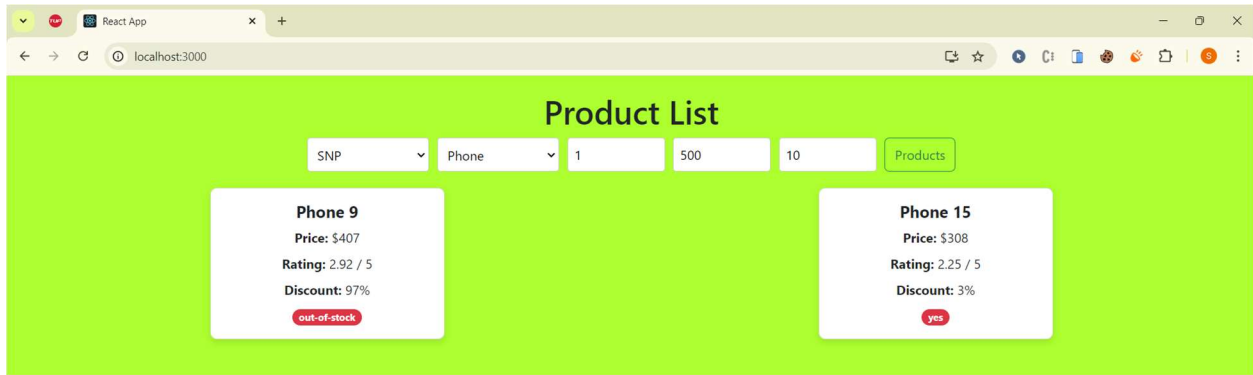
JS server.js
backend > JS server.js > ...
You, 13 minutes ago | 1 author (You)
1 const express = require('express');
2 const cors = require('cors');
3 const productRoutes = require('./routes/productRoutes');
4
5 const app = express();
6 const port = 5000;
7
8 // Middleware
9 app.use(cors());
10 app.use(express.json());
11
12 // Routes
13 app.use('/products', productRoutes);
14
15 app.listen(port, () => {
16   console.log(`Server is running on http://localhost:${port}`);
17 });
18
```

Folder structure



Question-2





Note that the development build is not optimized.
To create a production build, use `npm run build`.

Compiled successfully!

You can now view **frontend** in the browser.

Local: http://localhost:3000
On Your Network: http://172.21.166.179:3000

Note that the development build is not optimized.
To create a production build, use `npm run build`.

webpack compiled **successfully**

█

