COLLEGE CODE : 9111

COLLEE NAME : SRM MADURAI COLLEGE FOR ENGINEERING AND TECHNOLOGY

DEPARTMENT : COMPUTER SCIENCE AND ENGINEERING

STUDENT NM ID ROLL NO : 1E6803F2FCA0F4CAA1343345D8F2708B

DATE : 06/10/2025

PROJECT NAME : IBM-FE-PRODUCT CATALOG

PROJECT PHASE : 05

SUBMITTED BY: BRINDHA B(3RD YEAR,B.E CSE)

MOBLE NO : 9080174974

# Project Title: Product Catalog

## Final Demo Walkthrough

The Product Catalog project demonstration includes a walkthrough of the web application, highlighting key functionalities such as browsing products, searching by category, filtering by price range, and viewing detailed product descriptions. The demo also covers the admin interface used for adding, editing, or deleting products. The walkthrough emphasizes the responsive design and user-friendly navigation.

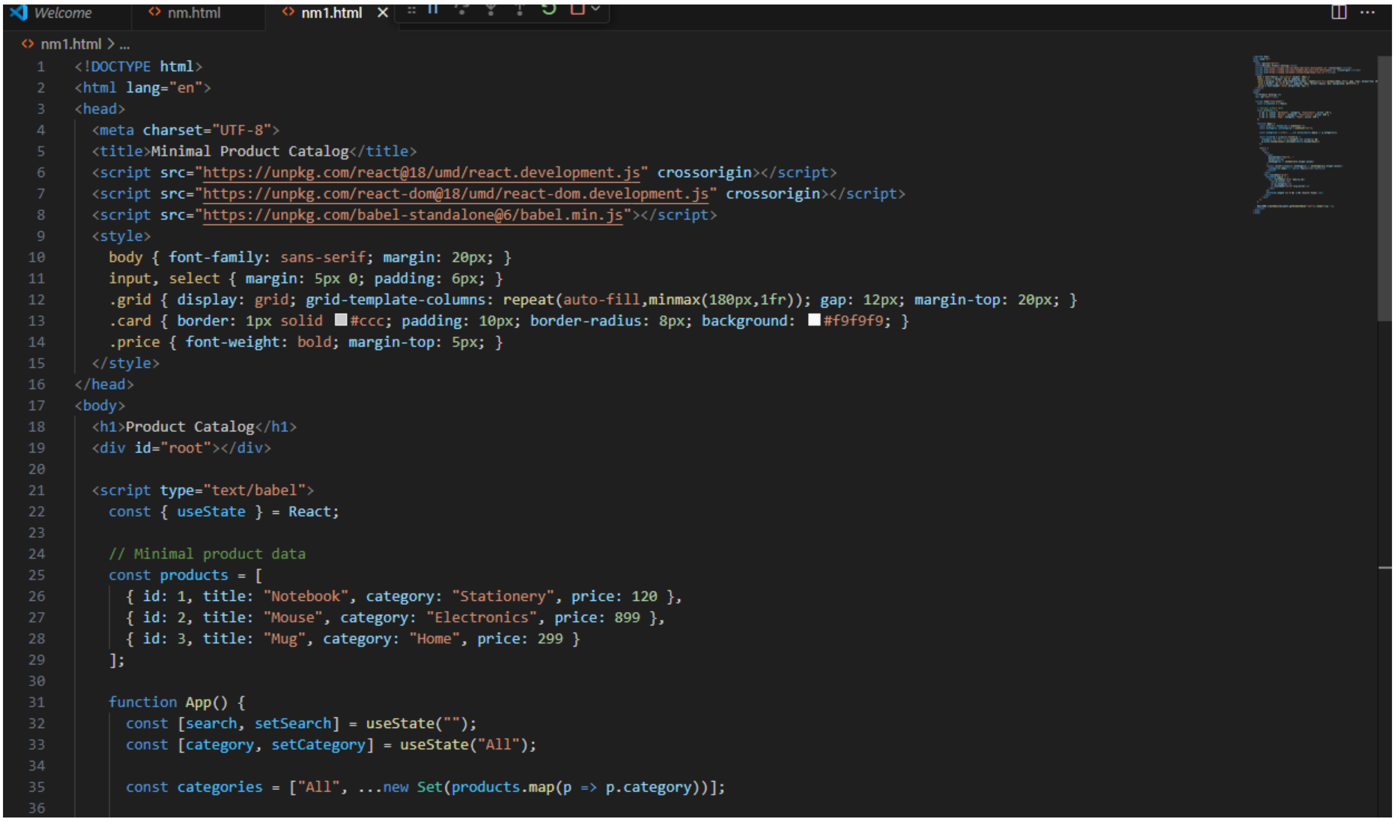
## Project Report

The Product Catalog project aims to create an organized digital platform that allows users to browse and explore various products efficiently. The system is built using modern web technologies such as HTML, CSS, JavaScript, and a backend framework (like Node.js or Django) connected to a database for storing product details. It provides features such as product categorization, search functionality, and dynamic product listings. The report covers system design, database schema, implementation steps, and performance evaluation.

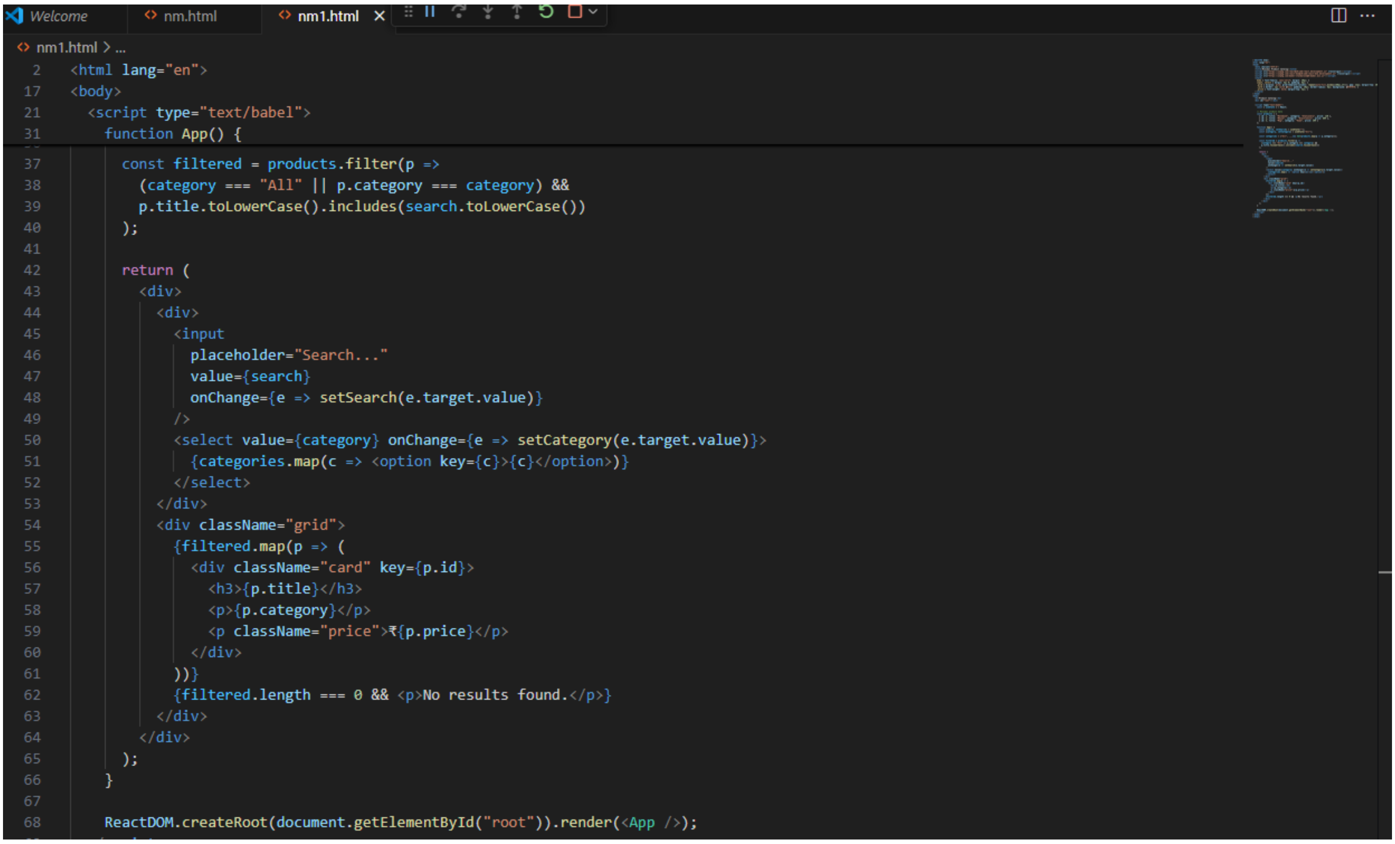
## Screenshots / API Documentation

Screenshots demonstrate the user interface for product browsing, search functionality, and admin management. API documentation outlines endpoints for retrieving, adding, updating, and deleting product information. Example endpoints include '/api/products', '/api/products/:id', and
'/api/categories'. Each endpoint includes details about request methods, parameters, and sample responses.

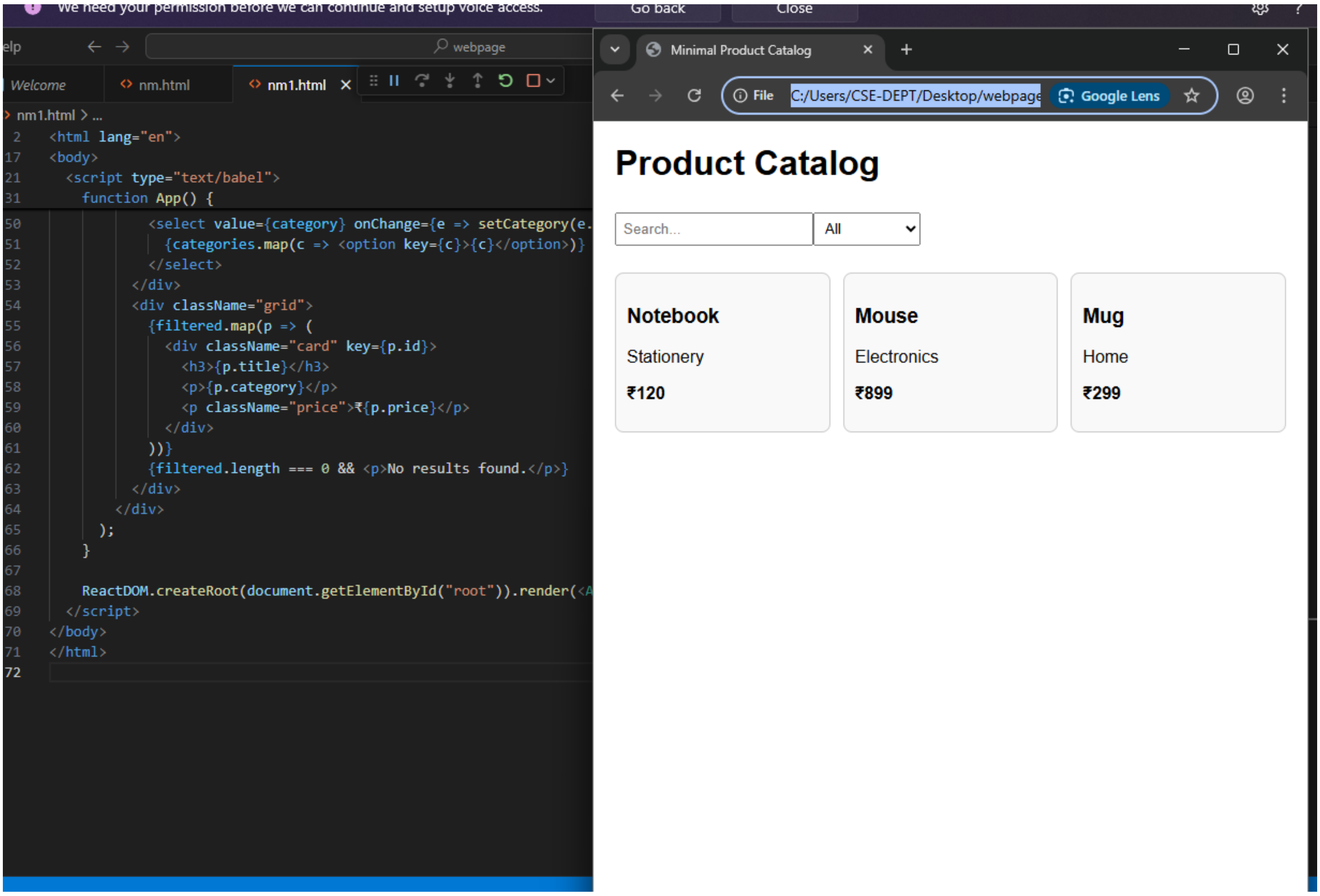
**Coding with output:**



```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Minimal Product Catalog</title>
  <script src="https://unpkg.com/react@18/umd/react.development.js" crossorigin></script>
  <script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js" crossorigin></script>
  <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>
  <style>
    body { font-family: sans-serif; margin: 20px; }
    input, select { margin: 5px 0; padding: 6px; }
    .grid { display: grid; grid-template-columns: repeat(auto-fill,minmax(180px,1fr)); gap: 12px; margin-top: 20px; }
    .card { border: 1px solid #ccc; padding: 10px; border-radius: 8px; background: #f9f9f9; }
    .price { font-weight: bold; margin-top: 5px; }
  </style>
</head>
<body>
  <h1>Product Catalog</h1>
  <div id="root"></div>

  <script type="text/babel">
    const { useState } = React;

    // Minimal product data
    const products = [
      { id: 1, title: "Notebook", category: "Stationery", price: 120 },
      { id: 2, title: "Mouse", category: "Electronics", price: 899 },
      { id: 3, title: "Mug", category: "Home", price: 299 }
    ];

    function App() {
      const [search, setSearch] = useState("");
      const [category, setCategory] = useState("All");

      const categories = ["All", ...new Set(products.map(p => p.category))];

```



```html
<html lang="en">
<body>
  <script type="text/babel">
    function App() {
      const filtered = products.filter(p =>
        (category === "All" || p.category === category) &&
        p.title.toLowerCase().includes(search.toLowerCase())
      );

      return (
        <div>
          <div>
            <input
              placeholder="Search..."
              value={search}
              onChange={e => setSearch(e.target.value)}
            />
            <select value={category} onChange={e => setCategory(e.target.value)}>
              {categories.map(c => <option key={c}>{c}</option>)}
            </select>
          </div>
          <div className="grid">
            {filtered.map(p => (
              <div className="card" key={p.id}>
                <h3>{p.title}</h3>
                <p>{p.category}</p>
                <p className="price">₹{p.price}</p>
              </div>
            ))}
            {filtered.length === 0 && <p>No results found.</p>}
          </div>
        </div>
      );
    }

    ReactDOM.createRoot(document.getElementById("root")).render(<App />);
```

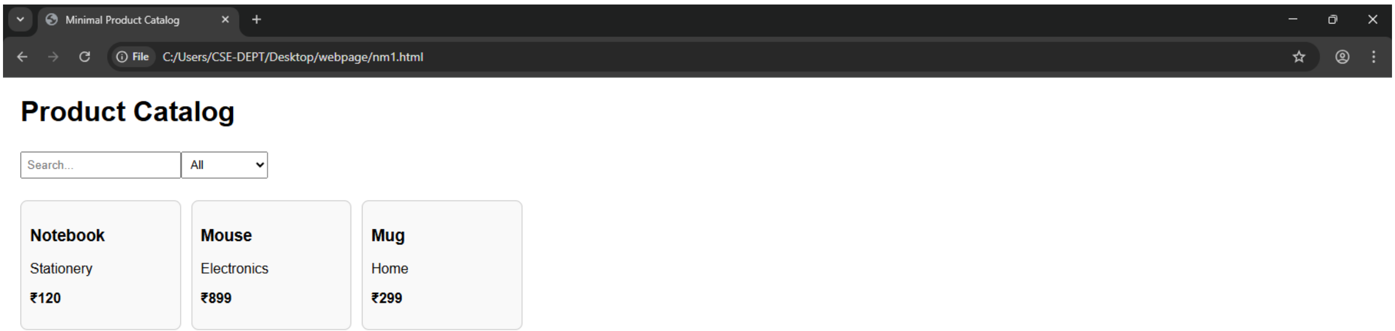

We need your permission before we can continue and setup voice access.
Go back
Close
webpage
Welcome
nm.html
nm1.html
nm1.html > ...
Minimal Product Catalog
File
C:/Users/CSE-DEPT/Desktop/webpage
Google Lens

```
<html lang="en">
<body>
  <script type="text/babel">
    function App() {
            <select value={category} onChange={e => setCategory(e.
              {categories.map(c => <option key={c}>{c}</option>)}
            </select>
          </div>
          <div className="grid">
            {filtered.map(p => (
              <div className="card" key={p.id}>
                <h3>{p.title}</h3>
                <p>{p.category}</p>
                <p className="price">₹{p.price}</p>
              </div>
            ))}
            {filtered.length === 0 && <p>No results found.</p>}
          </div>
        </div>
      );
    }

    ReactDOM.createRoot(document.getElementById("root")).render(<A
  </script>
</body>
</html>
```

Product Catalog

Search... | All

| | | |
|---|---|---|
| **Notebook**<br>Stationery<br>**₹120** | **Mouse**<br>Electronics<br>**₹899** | **Mug**<br>Home<br>**₹299** |



Product Catalog

Search... | All

| | | |
|---|---|---|
| **Notebook**<br>Stationery<br>**₹120** | **Mouse**<br>Electronics<br>**₹899** | **Mug**<br>Home<br>**₹299** |

## Challenges & Solutions

During development, challenges included handling large datasets efficiently and ensuring a seamless user experience. Implementing search optimization and pagination helped overcome performance issues. Ensuring responsive design across devices was another challenge, solved by using CSS frameworks such as Bootstrap and media queries. Integrating the backend API with the frontend required debugging CORS and routing errors, which were resolved by proper API configuration and testing.

## GitHub README & Setup Guide

The GitHub repository includes a detailed README file containing setup instructions. Steps include cloning the repository, installing dependencies using npm or pip, configuring environment variables, and running the server using the appropriate command. The guide also provides information about deployment options such as hosting on Vercel or Heroku.

Github links: 1. RESHMA SRI R - https://github.com/ReshmaSriR/PRODUCT-CATLOG.git

2. VIJI N - https://github.com/vijinagaraj/product-catalog.git
3. KAVIYA M - https://github.com/kaviyakanishk12-design/product-catelog.git
4. BRINDHA B - https://github.com/Brindha87/product-cateelog.git

## Final Submission (Repo + Deployed Link)

The final submission includes the GitHub repository link containing source code, documentation, and setup guide, along with the deployed project link demonstrating the live product catalog system. Both links ensure accessibility for evaluation and demonstration purposes.

VIDEO LINK:
https://drive.google.com/file/d/18qePaIXlcR6UC1VLrzIzx2XXpUufmMFz/view?usp=drive_link