COLLEGE CODE:9620

COLLEGE NAME : SATYAM COLLEGE OF ENGGINEERING AND TECHNOLOGY

DEPARTMENT : CSE

STUDENT NM-:F9CAD8FBF48A76E9CFA4302C26A88CEA

ROLL NO :962023104027

DATE :18-10-2025

Completed The Project named as

Phase 5 TECHNOLOGY PROJECT

NAME : IBM-FE-Product catalog with filters

SUBMITTED BY,

NAME :MANI BHARATHI

MOBILE NO : 6374788746

Phase 5 - Project Demonstration & Documentation

1.Final Demo Walkthrough

- Homepage loads with a list of products displayed in a responsive grid layout.

- Sidebar filters allow users to filter products by category, price range, brand, and color.

- As filters are selected, the product list updates instantly to reflect matching items — no page reload.

- Users can reset filters to view the full catalog again.

- The catalog is fully responsive, working seamlessly on mobile, tablet, and desktop.

- Products can be clicked (optional) to view more details or add to cart (if implemented).

2. PROJECT REPORT

➢ **Objectives**

- To create a user-friendly product listing interface.
- To enable dynamic filtering of products.
- To enhance the shopping experience with quick and easy navigation.

➢ **Tech Stack**

- Frontend: HTML, CSS, JavaScript
- Backend (Optional): JSON / Node.js
- Tools: VS Code, GitHub

➢ **Features**

- Product listing with images, name, and price.
- Real-time filters (Category, Price Range, Rating).
- Search bar for quick product lookup.
- Responsive design for mobile and desktop.

3.Screenshots /API Documentation

Screenshots:

- Home page with products and filters.
- Filtered products view.
- Product details view.

API Endpoints:

- GET /api/products – Get all products.
- GET /api/products?category=&price=&rating= – Get filtered products.
- GET /api/products/<id> – Get product details.

4.CHALLENGES & DOCUMENTATION

- **Challenge:** Slow data loading when multiple filters applied.
  Solution: Implemented server-side filtering and caching for faster response times.
- **Challenge:** Managing complex filter combinations.
  Solution: Used dynamic query building and state management to handle multiple filters efficiently.
- **Challenge:** UI responsiveness on different devices.
  Solution: Integrated real-time API updates and validation checks

GITHUP REDME SETUP GUIDE

Project: Product Catalog with Filters
A web app that displays products with category, price, and rating filters.

Tech Stack: React, Node.js, Express, MongoDB

Setup Steps:

1. Clone the repo: `git clone https://github.com/username/product-catalog.git`
2. Install dependencies: `npm install`
3. Add `.env` file with DB connection details.
4. Run backend: `npm run server`
5. Run frontend: `npm start`

Features:

- Filter, search, and sort products
- Responsive UI

FINAL SUBMISSION (Repo + Deployed Link)

GitHub Repository:

🖇️ **https://github.com/kanimalathi50-jpg/marketing**

Deployed Link:
🌐 https://kanimalathi50-jpg.github.io/marketing/

Description:
A responsive product catalog web app featuring search, filter, and sort functionalities built using React, Node.js, and MongoDB.

Includes:

- Complete source code (frontend + backend)
- API documentation
- Setup guide and deployment instructions