NEXT GEN EMPLOYABILITY PROGRAM

CREATING A FUTURE-READY WORKFORCE

Student Name :
Kadam Nitisha Narayan

Student ID : **INTERNSHIP_173951615267aee8f889303**

College Name :
Karnatak Science College

CAPSTONE PROJECT SHOWCASE

Project Title

UrbanAura :ECommerce platform using MERN Stack

Abstract | Problem Statement | Project Overview | Proposed Solution | Technology Used | Modelling & Results | Conclusion | Q&A

Abstract

1. This project is an e-commerce website designed to showcase and sell a variety of products categorized under sections like Men, Women, Kids, Gadgets, Accessories, and more.
2. The website allows users to browse products, filter them by category, view detailed information about each item, add products to the cart, and proceed to checkout.
3. It uses React for the frontend and Node.js with Express and MongoDB for the backend.
4. Image uploads are supported, and each product can have an associated image, description, and pricing details. The site aims to provide a user-friendly interface with responsive design and smooth navigation.

Problem Statement

In today's digital era, customers increasingly prefer shopping online for convenience, variety, and ease of access. However, many small to medium businesses lack a user-friendly and responsive e-commerce platform that offers personalized product recommendations, smooth navigation, and essential functionalities like cart management and product filtering.

The goal of this project is to design and develop a **fully functional and visually appealing e-commerce website** that allows users to browse products by categories, view product details, add products to a cart, and proceed to checkout with a simple and intuitive interface. The platform should include a responsive design, dynamic product rendering from a backend API, and image handling for product listings.

Project Overview

- A modern and responsive web application that allows users to browse a wide range of products, filter by categories, view detailed descriptions, and manage their shopping cart. It includes features such as add to cart, buy now, and category-based browsing. Each product can be individually explored through a clean and intuitive UI.

Proposed Solution

- This ecommerce project is a user-friendly web application designed to showcase and sell various products categorized into sections like Women, Men, Gadgets, etc. It features a dynamic front-end built using **React.js**, with state and route management handled by **React Router** and **Context API**. The backend, powered by **Node.js** and **Express.js**, connects to a **MongoDB database** to manage products, images, and orders. Product data is fetched via RESTful APIs, with images stored locally or using an upload folder. Each product includes fields like name, price, image URL, description, and category, and users can view detailed pages, add items to a cart, or proceed to checkout. Styling is enhanced using **custom CSS** and can be improved further with **CSS modules or frameworks like Tailwind or Bootstrap** to provide a modern, responsive, and visually appealing UI. The home page features a unique design, while other pages follow a consistent theme, with flexibility to apply page-specific styling as needed.

Technology used

Frontend
•**React.js** – For building dynamic and responsive user interfaces.
•**React Router** – For client-side routing between different components/pages.
•**Axios** – For making HTTP requests to the backend API.
•**Context API** – For managing global state like the shopping cart.
•**CSS** – For styling the website, with custom component-level and global styles.

Backend
•**Node.js** – JavaScript runtime for server-side logic.
•**Express.js** – Web framework for building RESTful APIs.
•**MongoDB** – NoSQL database to store products, users, and orders.
•**Mongoose** – ODM for interacting with MongoDB using JavaScript.

Others
•**Postman** – For testing and verifying API endpoints.
• **Local Uploads** – To handle and store product images.
•**Vite** – For fast frontend development and optimized builds.

Modelling & Result

Modelling (System Design & Architecture)

•**Frontend**:

Developed using **React.js** with routing via react-router-dom.

• Components include:

- •Home Page
- •Product Listing Page
- •Product Details Page
- •Add Product Page
- •Cart & Checkout Pages

Modelling & Result

Backend:

Built with **Express.js** and **Node.js**. Handles:

- Product CRUD operations
- Image upload using multer
- API endpoints for frontend communication

Database:

MongoDB stores product information with fields like:

- name, price, offerprice, image, description, category, quantity

Image Upload:

- Stored in /uploads/ directory
- Image path saved in MongoDB and accessed via URL like:

http://localhost:5000/uploads/image.jpg

Modelling & Result

Results (Output Screens & Features)

•Add Product Form

Admin can enter product details and upload images, which are saved and displayed.

•Product Page

Users can view all products, filter by category, and view detailed product pages.

• **Dynamic Image Rendering**

Uploaded images correctly load via backend URL and render on product cards.

•Category Filter

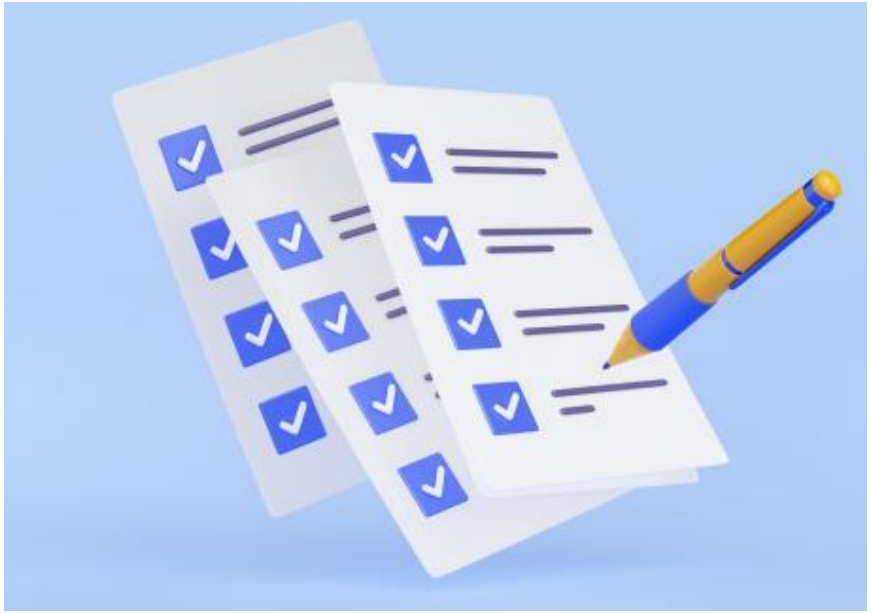
Products can be filtered by category like *Men*, *Women*, *Gadgets*, etc.

• **Cart Functionality**

Users can add items to the cart and proceed to checkout.

Conclusion

- This e-commerce website is a basic attempt to build an online shopping platform with features like product listings, filtering, cart management, and a simple checkout process. It uses technologies such as React, Node.js, MongoDB, and Express to create a full-stack application. While there is still room for improvement, this project helped me understand the core concepts of web development and how to connect the frontend with the backend.



edunet
foundation