Research Report: Give me the link of E-Commerce project using MERN stack

Research Query: Give me the link of E-Commerce project using MERN stack

Generated: June 12, 2025 at 13:37

Generated by: NEXUS AI Agentic Backend v3.0.0

Sections: 5

Search Results Summary	
Total Sources	8
Media Items	10
Credibility Score	68%
Processing Time	45.23s

Table of Contents

1. Executive Summary

best result got all link

2. Key Findings

ABC

3. Source Analysis

Sources and References

#	Title	Domain	Credibility	Type
1	E-commerce Website using MERN Stack -...	www.geeksforgeeks.org	85%	Webpage
2	E-commerce Website using MERN Stack -...	www.geeksforgeeks.org	85%	Webpage
3	mern-ecommerce · GitHub Topics	github.com	75%	Repository
4	eCommerce Platform Project - MERN Sta...	github.com	75%	Repository
5	MERN-E-Commerce-Website - laxmansingh...	laxmansinghnegi10.github.io	55%	Webpage
6	Fusion Electronics: A MERN-Stack E-co...	ankit7rma.github.io	55%	Webpage
7	E-commerce website using the MERN Sta...	reactjsexample.com	55%	Webpage
8	Build an E-Commerce website with MERN...	dev.to	55%	Webpage

Detailed Source Information:

1. E-commerce Website using MERN Stack - GeeksforGeeks

URL: https://www.geeksforgeeks.org/e-commerce-website-using-mern-stack/

Summary: This article details building an e-commerce website using the MERN stack (MongoDB, Express.js, React, Node.js). The backend, built with Node.js and Express, utilizes MongoDB to store product data and provides an API for data retrieval. The frontend, developed with React, displays product information fetched from this API. Crucial steps include setting up the backend (including installing necessary packages like `express`, `mongoose`, and `cors`), defining a product schema in MongoDB, and creating an API endpoint to handle product data requests. The frontend then consumes this API to render the products on the user interface. The guide emphasizes using CORS middleware to handle cross-origin requests between the client and server. The example code snippets illustrate setting up the Express server, connecting to MongoDB, and defining a basic product schema. The complete project involves creating React components to display product details effectively to the user.

2. E-commerce Website using MERN Stack - GeeksforGeeks

URL: https://www.geeksforgeeks.org/mern/e-commerce-website-using-mern-stack/

Summary: This article details building an e-commerce website using the MERN stack (MongoDB, Express.js, React, Node.js). The backend, built with Node.js and Express, utilizes MongoDB to store product data and provides an API for data access. The frontend, developed with React, displays products fetched from this API. Crucial steps include setting up the backend (including installing necessary packages like `express`, `mongoose`, and `cors`), defining a product schema in MongoDB, and creating an API endpoint to retrieve product data. The frontend then consumes this API to render product information. The tutorial emphasizes using CORS middleware to handle cross-origin requests. The provided code snippets illustrate setting up the Express server, connecting to MongoDB, and defining a basic product schema. The complete project involves creating React components to display products effectively, offering a functional e-commerce platform.

3. mern-ecommerce · GitHub Topics

URL: https://github.com/topics/mern-ecommerce

Summary: GitHub showcases numerous public MERN stack e-commerce repositories, with over 90 projects using JavaScript and TypeScript. Top repositories like `mohamedsamara/mern-ecommerce` (1.9k stars) and `jigar-sable/flipkart-mern` (410 stars) demonstrate robust, full-stack applications featuring functionalities such as user profiles, order tracking, admin dashboards, and payment gateways (e.g., Paytm). Many projects integrate technologies like Redux, MongoDB, Express.js, Node.js, and cloud services (Cloudinary, SendGrid). These examples highlight the versatility of the MERN stack for building feature-rich e-commerce platforms, ranging from basic e-commerce sites to complex applications with advanced features like Stripe API integration and JWT authentication. The projects vary in complexity and features, offering developers numerous examples and starting points for their own e-commerce projects.

4. eCommerce Platform Project - MERN Stack - GitHub

URL: https://github.com/ajaybor0/MERN-eCommerce

Summary: This GitHub repository (https://github.com/ajaybor0/MERN-eCommerce) presents a fully functional e-commerce platform built using the MERN stack (MongoDB, Express.js, React, Node.js). The platform offers a robust user experience with features including a full-featured shopping cart, product reviews and ratings, a product search, user profiles with order history, and a comprehensive admin dashboard. Admin functionalities encompass user and product management, order details viewing, and marking orders as delivered. Razorpay integration ensures secure payment processing. The project includes a database seeder for easy setup. A live demo is available (though note potential delays due to Render's free tier inactivity timeout), showcasing the platform's capabilities. Getting started requires forking the repository, cloning it locally, setting up a MongoDB database and a Razorpay account, and configuring environment variables.

5. MERN-E-Commerce-Website - laxmansinghnegi10.github.io

URL: https://laxmansinghnegi10.github.io/-MERN-E-Commerce-Website-/

Summary: This project details a fully functional e-commerce website built using the MERN stack (MongoDB, Express.js, React.js, Node.js) and Redux. Key features include a responsive design adaptable to various devices, a scalable Node.js/Express.js backend for efficient handling of increasing user traffic and transactions, and robust data management via MongoDB. Redux ensures smooth state management in the React frontend, contributing to a consistent user experience. The architecture is designed for future scalability and adaptability. The provided instructions detail installation prerequisites (Node.js, npm, MongoDB), cloning the repository, installing dependencies, setting environment variables (MONGO_URI, JWT_SECRET, PORT), and starting both the server and React development server. The application is accessible locally at http://localhost:3000 after successful setup. The project's structure is clearly defined, separating the client-side (React) and server-side code.

4. Media Gallery

Media Gallery

#	Title	Type	Source
1	geeksforgeeks	Image	https://www.geeksforgeeks.o...
2	Screenshot-2024-03-19-013706	Image	https://www.geeksforgeeks.o...
3	Screenshot-2024-03-10-104441	Image	https://www.geeksforgeeks.o...
4	Screenshot-2024-03-10-104433	Image	https://www.geeksforgeeks.o...
5	ww	Image	https://www.geeksforgeeks.o...
6	course-img	Image	https://www.geeksforgeeks.o...
7	course-img	Image	https://www.geeksforgeeks.o...
8	course-img	Image	https://www.geeksforgeeks.o...

5. Conclusion

thanks

Document Information

Generated by	NEXUS AI Ultra Premium Agentic Search Engine

Query	Give me the link of E-Commerce project using MERN stack
Generation Time	2025-06-12 13:37:40 UTC
Total Sections	5
Sources Analyzed	8
Media Items	10