RESEARCH ARTICLE OPEN ACCESS

A Review on MERN Stack

Gitesh Singh*, Hitesh Sumarwal*, Pawan Sen**

*B.Tech Student, Department of CSE, Arya Institute of Engineering & Technology, Jaipur
** Assistant Professor, Department of CSE, Arya College of Engineering, Jaipur

ABSTRACT

In today's fast-paced business world, it is critical to be able to understand client demand efficiently and ahead of time. If our customers could do business online and have access to our products or services at their fingertips, it would have a greater impact on their daily lives, as well as create an ecosystem of doing business online and serving customers on a large scale. Shopping and doing business online has become a way of life for the younger generation. E-commerce web applications, which retail a variety of products, have provided people with access to everything from basic necessities to luxury items. This project allows registered users to view various products on a web user interface and obtain desired products immediately use desired payment methods. This project provides a simple way for business owners to view orders placed. Several technologies have been studied and acknowledged in the process of developing an e-commerce web application. React.js, MongoDB, Node.js, and Express.js are among the technologies included. This is a project to make it easier to access various products and to create a web application in which a customer receives a comprehensive web application, as well as to understand the technologies used to demonstrate such an application. This paper will go over each of the fundamental technologies required to build and deploy an ecommerce web application.

Keywords — ReactJS, NodeJS, MongoDB, and Express JS.

I. INTRODUCTION

Doing business in products or services via the Internet is known as electronic commerce. Mobile commerce, electronic funds transfer, internet marketing, supply chain management, online transaction processing, inventory management systems, automated data collection systems, and electronic data interchange are all examples of technologies used in electronic commerce (EDI). Ecommerce has traditionally used the World Wide Web for at least one segment of the transaction life cycle, though it may also use other technologies such as email, etc. E-commerce businesses can use Online e-commerce web applications to continue retail sales to customers. Offering or participating in online marketplaces that process third-party B2C or C2C sales. Buying and selling between businesses. Obtaining and utilising demographic data via web application contacts and social media platforms. Marketing to prospective customers and entrenched customers by e-mail. Making plans in retail to introduce new services and products. According to a study conducted by the Associated Chambers of Commerce and Industry of India, the online retail market in India could grow to Rs.70 billion by 2015, up from Rs.20 billion in 2011. India has always been a country with enormous potential. The country's socioeconomic situation has improved many folds since independence, and India is now emerging as one of the world's leading countries. Furthermore, with a population of over 100 crores and a growth rate of more than 6%, it can be considered a marketing behemoth. As a result, it is clear why online business and shopping in India has grown at such a rapid pace over the years. As technology spreads to India's remotest villages and more job opportunities emerge for the youth, an increasing number of people are gaining awareness and the funds to purchase the expensive and luxurious product over any web application that provides retail products and services. Ecommerce web applications facilitate interaction between different parties involved in a commerce trade, as well as data management in the process via the internet on social media platforms. Electronic commerce is gaining traction as a start-up business and the momentum it gains as the business grows as a recognised and used business model.

II. DEVELOPMENT OF WEB PAGES

The foundations of today's commercial internet were laid in 1990. Tim Berners-Lee created the basic concepts of the World Wide Web and many tools for efficient web usage at the end of 1990. These tools include HyperText Transfer Protocol (HTTP), HyperText Markup Language (HTML), the first web browser and code editor, the first web server, and the first web page, which defined a new term, the world wide web, and a method for creating one's own web page. Since 1990, the internet has been rapidly evolving, and its evolution can be divided into four generations. Users could only read web content on the first generation of web pages, which were static and not frequently updated. The main motto was to read only the web. HTML was used to create all web pages, and HTTP was the primary communication protocol. The second generation begins in 2004, with terms such as various social networks, blogs, the ability for users to create web page content, and the enhancement of user experience browsing web interfaces. During that time, well-known social networks such as Facebook, Twitter, LinkedIn, and others emerged. These social networks allowed users all over the world to connect. At the same time, new technologies such as JavaScript, Document Object Model (DOM), Ajax, Cascading Style Sheets (CSS), eXtensible HTML (XHTML), eXtensible Markup Language (XML), eXtensible Stylesheet Language (XSL), and Flash were introduced, which enabled the presentation and delivery of web services without the problems associated with web distribution. The third web generation begins in 2010, and it is distinguished by the semantic web (the addition of semantics to the web), content personalization, intelligent search, and computers' ability to create a wide range of content. Ontologies are used to represent and reason about meaning. Aside from ontologies, other technologies used in the third web generation include Resource Description Framework (RDF), Web Ontology Language (OWL), and others. People may be able to refer to Internet 4.0 as the active web in the fourth generation. In the current web 3.0, the use of search engines is still critical, as it provides us with knowledge, in its largest content web applications, which we can use as needed. This will not be the case with Web 4.0. When fully established, it will not require several of the steps required when using web 3.0, making its use more direct and invisible.

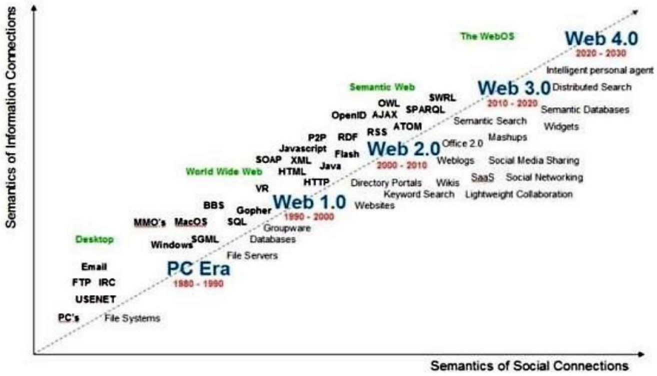

Fig 1. Web Generations

III. METHODOLOGY

MERN stack was used to create the web application. The web application was built with the MERN stack and the Mongoose and MongoDB databases. Chrome developer tools were used for simulation testing while using redux tools. The following section goes over the MERN stack components and how they are implemented.

A. NodeJS

Node.js is written in C++ and runs on the JS operating system. Node.js is a runtime environment for JS. For optimal performance, Node.js employs the Google Chrome V8 engine. Node.js is designed with a single-thread architecture and uses event-driven, asynchronous programming callback functions. The event-driven design of Node.js is the fundamental core concept for its environment, which has given us a variety of APIs that are event-based and asynchronous in nature, which has assisted us in building the website using Node.js for our back-end development. Because we used Node.js, it called the corresponding callback function based on the business logic of our web application. These callback functions are executed asynchronously, which means that while they appear to be registered sequentially in the logic, they do not rely on the code in which they appear and instead wait for the execution of the corresponding event to fire. The main benefit of event-driven and asynchronous programming is that it is single-threaded. The call back function code is executed without waiting for a specific code to complete, and the limited resources were used for other tasks that would be executed as part of our web application's business logic. This design was appropriate for our back-end development, which was also the system's goal. Handling synchronous requests was a major task in server development, and blocking was the cause of not fully utilising or wasting resources. We improved resource utilisation and website performance by using single thread architecture and asynchronous callback functions, which also provided us with the desired results during testing. We can see from the Node.js supported module that many functions, including file operations, are executed asynchronously, which is different from other languages. To make server development easier, Node.js employs large network modules such as HTTP, DNS, NET, UDP, HTTPS, TLS, and so on. Using these network modules, developers can create a Web server.

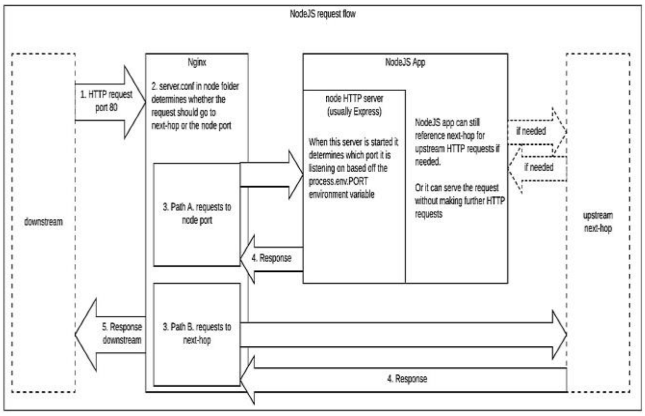
NodeJS request flow
downstream
1. HTTP request port 80
Nginx
2. server.conf in node folder determines whether the request should go to next-hop or the node port
3. Path A. requests to node port
4. Response
3. Path B. requests to next-hop
5. Response downstream
4. Response
NodeJS App
node HTTP server (usually Express)
When this server is started it determines which port it is listening on based off the process.env.PORT environment variable
NodeJS app can still reference next-hop for upstream HTTP requests if needed.
Or it can serve the request without making further HTTP requests
if needed
if needed
upstream next-hop

Fig 2. NodeJs Request Flow

B. ExpressJS

Because Express is a Node.js framework, we used it. While developing the application, we discovered that instead of creating a slew of Node modules and writing code in NodeJS. Express made it simpler and easier to write and implement back-end code in a structured format. Express aided us in the design of our web applications and APIs for our project because it supports many middlewares, making the code shorter and easier to write. The most significant advantages of using Express in our application are asynchronous programming and single-threaded architecture. A robust API is required for our application. To begin our express project, we created a new folder and added a command to the command prompt to initialise the package.json file. Following that, we had to accept the default settings and proceed. The command to begin is npm init.

Express is a Node.js web application framework that offers a variety of features for developing web and mobile applications. It can be used to create a single-page, multi-page, or hybrid web application.

It is a layer built on top of Node js that aids in the management of servers and routes.

- Express was designed to make it simple to create APIs and web applications.
- It cuts coding time in half while still producing web and mobile applications.
- Mobile applications are useful.
- Another reason to use express is that it is written in javascript, which is a simple language to learn even if you have no prior experience.
- Fluency in any language. Express enables a large number of new developers to enter the field of web development.

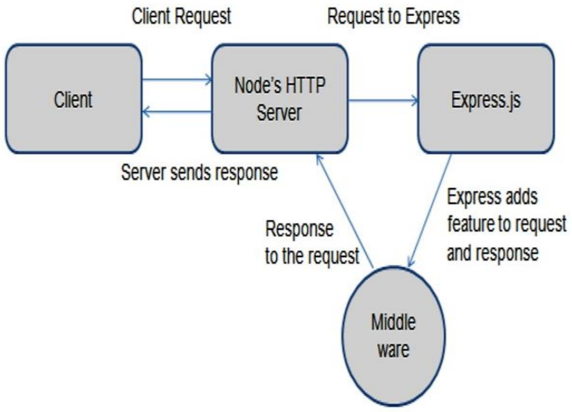

Fig 3. ExpressJs Request Flow

C. ReactJS

React.JS is the JavaScript programming language's front-end library. We built our web application's user interface with React.JS, which is used for single-page application development because it can render dynamically changing data at a high speed. React allows developers to write code in JS and create UI components. We investigated virtual DOM objects in React.JS, which we used in our project. Any changes we made to our e-commerce web application caused the entire UI to re-render the virtual DOM. This enables us to contrast the potential differences between the DOM Object and the Virtual DOM. We used JSX because it made writing code in a React application easier and faster. Components are used in React.JS. Components are the User-Interface building blocks, with each component having a logic related to our e-commerce application and contributing to the overall User-Interface of our web application. Components can be reused, which made our web application code easier to understand by other developers and improved the overall performance of the web application. To begin our react application, we first installed createreact-app with npm or yarn. npm instal yarn global adds OR npm instal create-react-app global The two commands for using npm or yarn, respectively, are create-react-app. Following that, we used to create a new react app. ourappname create-react-app Then, in our app name folder, type yarn start or npm start to launch your application.

Today, React's popularity has surpassed that of all other front-end development frameworks. This is why:

- Ease of creating dynamic applications: React makes it easier to create dynamic web applications because it requires less coding and provides more functionality than JavaScript, where coding can quickly become complex.
- Improved performance: React uses Virtual DOM to create web applications more quickly. Virtual DOM compares the previous states of the components and updates only the items in the Real DOM that have changed, rather than updating all of the components again, as traditional web applications do.
- Reusable components: Components are the foundation of any React application, and a single app is typically made up of multiple components. These components have their own logic and controls, and they can be reused throughout the application, reducing development time significantly.
- Unidirectional data flow: React uses a one-way data flow. As a result, when creating a React app, developers frequently nest child components within parent components. Because data flows in a single direction, it is easier to debug errors and pinpoint the location of a problem in an application at any given time.

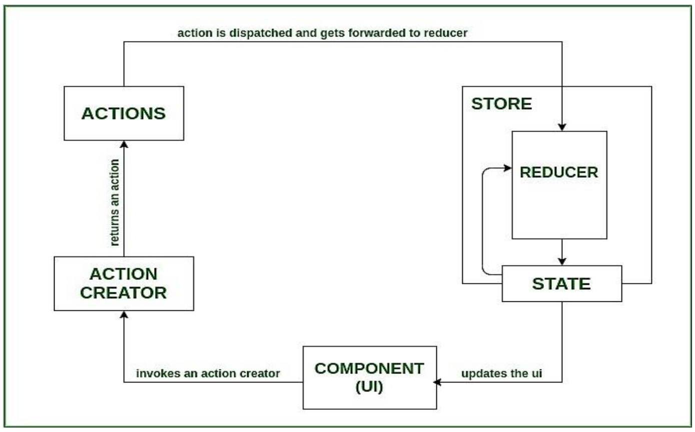

Fig 4. React-Redux Application Flow

D. MongoDB

For our project, we used a Document-Oriented Database (MongoDB). MongoDB is a database that stores data in the form of documents. MongoDB converts our JSON data behind the scenes on the server into a binary version that can be stored and queried more efficiently. MongoDB queries the database using BSON. MongoDB stores BSON format both internally and over the network, but that doesn't mean it's a JSON database. We can represent any data in JSON format, which can be natively stored in MongoDB and retrieved in JSON format just as easily. We can say that MongoDB is flexible because it allows users to create schema, databases, tables, and so on. After installing MongoDB, we had the option of using Mongo shell, which provides a JavaScript interface through which users can interact and perform querying operations. Because MongoDB is a document-oriented database, indexing documents is simple. As a result, it handles responses at a faster rate. Scalability is a feature of MongoDB. We handled large amounts of data in the MongoDB database by organising it into a nested documented structure. MongoDB is a database server that can host multiple databases.

It is a fantastic feature provided by MongoDB. A schema-less database allows multiple types of documents to be stored in a single collection. In other words, in the MongoDB database, a single collection can hold multiple documents, each of which can have a different number of fields, content, and size. It is not necessary for one document to be similar to another, as in relational databases. MongoDB's cool feature gives databases a lot of flexibility.

- In MongoDB, all data is stored in documents rather than tables as in RDBMS. In these documents, data is stored in fields (key-value pairs) rather than rows and columns, making the data much more flexible than in RDBMS. And each document has a distinct object id.
- Every field in the documents in the MongoDB database is indexed with primary and secondary indices, making it easier and taking less time to get or search data from the pool of data. If the data is not indexed, the database must search through each document with the specified query, which takes a long time and is inefficient.
- Sharding in MongoDB allows for horizontal scalability. Sharding is the process of distributing data across multiple servers. A large amount of data is partitioned into data chunks using the shard key, and these data chunks are distributed evenly across shards that reside across many physical servers. It will also add new machines to an already active database.

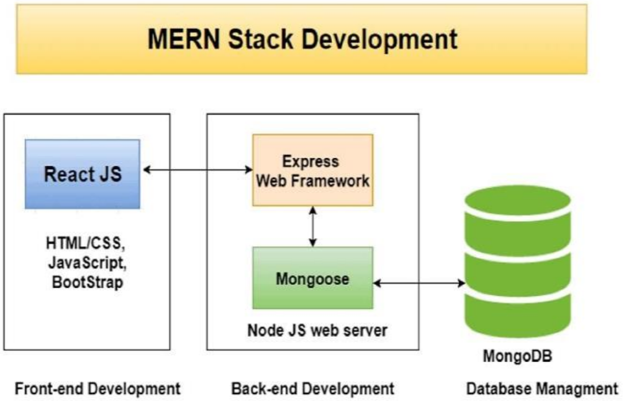

Fig 5. MERN Stack Architecture

IV. CONCLUSIONS

As of now, the current web application on e-commerce business aims to provide a critical review of relevant literature in the eBusiness field, as well as to describe key aspects of the methodology used throughout the project. More research and understanding of new technologies, frameworks, and testing tools will aid the web application's growth and performance. Web development using various technologies will be useful to many businesses in India and around the world in the future, as it creates a digitization ecosystem and makes business easier for everyone on the planet.

REFERENCES

[1] Office for National Statistics, Internet users in the UK: 2016. Retrieved September 26, 2017, from https:// www.ons.gov.uk/ businessindustryandtrade/itandinternetindustry/bulletins/ internetus ers/ 2016.

[2] Liang, L., Zhu, L., Shang, W., Feng, D., Xiao, Z. (2017). Express supervision system based on NodeJS and MongoDB.

[3] M. R. Solanki, A. Dongaonkar, A Journey of human comfort: web1.0 to web 4.0, International Journal of Research and Scientific Innovation (IJRSI), Volume III, Issue IX, pp. 75-78, 2016

[4] Javeed, A. (2019). Performance Optimization Techniques for ReactJS. 2019

[5] J. M. Spool, Content and design are inseparable work partners, 2014. Retrieved September 29, 2017, fromhttps://articles.uie.com/ content and design [6] Bozikovic, H., Stula, M. (2018). Web design Past, present and future. 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO).

[6] Kiran Ahuja, Harsh Sekhawat, Shilpi Mishra, Pradeep Jha (2021). Machine Learning in Artificial Intelligence: Towards a Common Understanding. Turkish Online Journal of Qualitative Inquiry, 12(8):1143-1152.

[7] H. Arora, M. Kumar, T. Rasool and P. Panchal (2022) Facial and Emotional Identification using Artificial Intelligence. 2022 6th International Conference on Trends in Electronics and Informatics (ICOEI), 1025-1030.

[8] Dr. Himanshu Aora, Kiran Ahuja, Himanshu Sharma, Kartik Goyal, Gyanendra Kumar (2021). Artificial Intelligence and Machine Learning in Game Development. Turkish Online Journal of Qualitative Inquiry (TOJQI), 12(8):1153-1158.

[9] Abhinav Agarwal, Himanshu Arora, Shilpi Mishra, Gayatri Rawat, Rishika Gupta, Nomisha Rajawat, Khushbu Agarwal (2023). Security and Privacy in Social Network. Sentiment Analysis and Deep Learning.

Advances in Intelligent Systems and Computing 1432:569577.

[10] S. Mishra, M. Kumar, N. Singh and S. Dwivedi (2022). A Survey on AWS Cloud Computing Security Challenges & Solutions. IEEE 6th International Conference on Intelligent Computing and Control Systems (ICICCS), 614-617.

[11] Shweta Pachauri, Deeksha Sharma, Dr. Rahul Misra (2022). Role of Computer Education in Indian Schools. International Journal of Recent Research and Review, XV(3), 15-18.

[12] Rahul Misra, Dr. Ramkrishan Sahay (2018). Evaluation of Five-Class Student Model based on Hybrid Feature Subsets. International Journal of Recent Research and Review, XI(1):80-86.

[13] S. Sharma, D. Arora, G. Shankar, P. Sharma and V. Motwani (2023). House Price Prediction using Machine Learning Algorithm. IEEE 7th International Conference on Computing Methodologies and Communication (ICCMC), 982-986.

[14] Dr. Himanshu Arora, Gaurav Kumar soni, Deepti Arora (2018). Analysis and Performance Overview of RSA Algorithm. International Journal of Emerging Technology and Advanced Engineering. 8(4), 10-12.

[15] Shilpi Mishra, Divyapratap Singh, Divyansh Pant, Akash Rawat (2022). Secure Data Communication Using Information Hiding and Encryption Algorithms. IEEE 2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS), 1448-1452.