International Journal of Engineering and Management Research e-ISSN: 2250-0758 | p-ISSN: 2394-6962
Volume-12, Issue-5 (October 2022)
www.ijemr.net https://doi.org/10.31033/ijemr.12.5.42

Efficiency of an E-Commerce Web Application with MERN Stack and Modern Tools

C.M. K De Silva[1], A. S De Silva[2], K.A. I Maduwantha[3], D.A.I.U Dewpura[4], D.I. De Silva[5] and R.R.P De Zoysa[6]
[1]Faculty of Computing, Sri Lanka Institute of Information Technology, New Kandy RD, Malabe, SRI LANKA
[2]Faculty of Computing, Sri Lanka Institute of Information Technology, New Kandy RD, Malabe, SRI LANKA
[3]Faculty of Computing, Sri Lanka Institute of Information Technology, New Kandy RD, Malabe, SRI LANKA
[4]Faculty of Computing, Sri Lanka Institute of Information Technology, New Kandy RD, Malabe, SRI LANKA
[5]Faculty of Computing, Sri Lanka Institute of Information Technology, New Kandy RD, Malabe, SRI LANKA
[6]Faculty of Computing, Sri Lanka Institute of Information Technology, New Kandy RD, Malabe, SRI LANKA

[1]Corresponding Author: imashakuruppu25@gmail.com

ABSTRACT

The majority of people in today's generation use technology to manage their life and take care of their basic necessities. Many of us in our generation shop for clothing, groceries, and electronics and even fancy items via e-commerce websites. We created a single chocolatier e-commerce web application utilizing the MERN stack, which consists of the MongoDB database, the Express.JS framework, the React.JS library, and the Node.JS platform. This program has several views for users and administrators, is fully functioning with 8 main functions. Depending on the interests of the consumer, we can purchase many chocolate types and brands through this website. In this project, we have the option to add and remove, edit, and save various products. For the website, we have created administrative features including login and logout options, admin dashboard, category and brand management for customers, payment method options for carts, review, and promotion management. Customers could search, sort, filter, and add goods to the cart. The customer can pay and purchase the things when the bills are created based on the cart contents. On the other hand, we have used the modern tools which are necessary to improve the functional and non-functional requirements of the application.

***Keywords*--** JavaScript, MERN Stack, Framework, Library, React.js, Node.js, Express.js, MongoDB, Azure boards, Selenium, GitHub, SonarQube

I. INTRODUCTION

As everyone is aware, technology is now a crucial instrument for online marketing. If we look around the world, most individuals are interested in making purchases online. However, we can see that a lot of grocers and small businesses sell their goods offline. Most of us will have unpleasant experiences with this style of selling. In some stores, for instance, the seller may have a product to sell in the offer, but the buyer may not be aware of it. Alternatively, the client may require the product urgently and visit the store, but the product may be out of stock in which case he will have a negative experience. Additionally, customers can choose from a variety of products based on their preferences and pricing, and through internet shopping, one can compare prices between different stores. Making an e-commerce web application is essential for browsing and purchasing in each store after recognizing all the flaws and issues with the offline shopping method. These days, there are a ton of e-commerce websites [5] like Flipkart, Amazon, and Myntra that one may use to effortlessly purchase their desired goods. One can purchase their goods while remaining at home by using these websites.

The MERN stack will be the finest alternative for assisting us in constructing the most effective and powerful web apps for these kinds of e-commerce [1].

II. PROBLEM IDENTIFICATION

The goal of this project is to create a web application that will make it simpler to find unique chocolates and facilitate their sale. The user-friendly E-commerce web application [7] has a visually appealing user interface and simple navigation. Customers can simply purchase things by adding them to their shopping carts, where they ca n then verify the total cost of the items they've added. a functional payment gateway has been installed, enabling debit or credit card payments. In order to increase the effectiveness of the application, we have also incorporated contemporary tools for project management, version control, and testing, such as Azure Boards, Git hub, Sonar Cube, and Selenium.

III. RELATED WORKS

If you want to find an E-commerce [2] app which is made with MERN stack there are many. We referred to many articles, but we will choose article [16] as a related

work. They have used the MERN stack for the E-Commerce application development. [7] We are using the same but with many new libraries to focus on efficiency of and E-Commerce web application with MERN stack and we are also using addition tools like Azure Boards, SonarQube, Selenium and GitHub for this.

A. Research Gap

In the research paper [16] they have used MERN stack for development of the E-Commerce web application [4]. And they focus on what E-Commerce is. We are using the same stack with many new libraries and tools to identify the efficiency we can achieve with using them for an E-Commerce web application. Our main focus is the efficiency we achieve from using MERN stack.

IV. METHODOLOGY

A. How we Started

As any project starts, our team members got together to discuss what we were going to do for this project. The first idea we all agreed on was to implement and E – commerce application. Since this application we were going to develop was mainly focused on selling something, the next problem that occurred was to select the product that we are going to sell through our application. First choice was to develop an application to sell tiles and bath ware. It was a good choice, but we thought why not make our application a delicious one and chose to focus our E-commerce application on selling chocolates. And this E – commerce application of ours is a Business to Consumer type application. Our basic idea was that the business will manage admin side and consumer can access the same website but only a user with an admin credentials can access the business side of the web application.

B. Main Functions of the Application

Then came the identifying of the main function that we need to implement for our application to become a successful application. After brainstorming we managed to identify seven main functions in our application. They are product management, user management, category management, brand management, order management, promotion management and review management. Each function has basic CRUD operations, search function and report generation. We were also determined to make the application more user responsive so that the user will have a good experience with our application. Product management function is focused on adding new chocolate products and managing them, it is essential for scaling up the business. Category management and brand management goes toe to toe with product management. It is easier for customer to find the products if the product can be found under a category or a brand. User management function is there so admins can manage the application better. They can assign new admins in case business is good. Order management is there to keep track of all the orders received by the business. Admins can track and change the status of orders easily. Review management is there so admins can get the honest reviews of the customers and increase their business by making their wrongs right. Lastly, promotion management is there to give promotions and increase the sales.

C. Technology

After getting the basics sorted out, the question we faced was what technology are we going to use to develop this application? First, we considered using Angular but since only one member of our team was well versed in Angular, we chose MERN stack which is related with JavaScript [8] to develop the E-commerce application. The main reason for is how efficient is MERN stack when developing a web application. We can develop a single component [15] and reuse the component again and again. Thus, increasing the code reusability. This component usage also makes the debugging easier, and it decreases the code complexity. It is also easy to identify any errors and Visual Studio Code (an open-source code editor we used for coding) also offers many extensions to ensure code quality and code readability. Best example for an extension like that is the prettier extension.

D. MERN Stack

If you are a newcomer to coding, you might be wondering what is MERN stack [16]? The simple answer is MERN stands for four different things. “M” is for MongoDB [13], which is the database we use in this stack. “E” stands for the Express JS [12]. The language we use to shape the backend of the application. “R” stands for React. What we use to design the frontend also known to many as the client-side of the application. And lastly, “N” stands for NodeJS [9], which together with Express makes up the middle tier of the application. NodeJS [10] is a very powerful and well-known JavaScript [11] server platform among many developers. MERN architecture is a three-tiered architecture. As mentioned before, frontend is React, backend or the server is Express and Node, and Database is the MongoDB. We used MERN stack for our application and many new libraries. Cloudinary was also used to store profile pictures.

E. Frontend

After selecting MERN stack as the technology we will be using to develop the web application, what we first did was starting to develop the interfaces for each main function. It was also our first milestone assessment. When designing the application, we discussed of following the KISS principle. Which stand for keep it simple and stupid. Which will make the interfaces more user friendly. We referred many chocolates related E-commerce web application [3] to get an idea about how they have designed their application. What we noticed was that almost all the applications have used two main colors with white to

e-ISSN: 2250-0758 | p-ISSN: 2394-6962
Volume-12, Issue-5 (October 2022)
https://doi.org/10.31033/ijemr.12.5.42

design their application. We also chose two main colors and started designing our interfaces using Figma. At the same time, we divided the workload between the group members and divided that workload into two sprints. Below images are the UIs of our application.

Figure 1: Home page

This is the home page of our application. When user enters the application, he will be directed to this page. User can login using the profile icon. If user scrolls down user can view the available products. User can also go to the about page to view description and products page to view all the available products. User can search for a specific product. User can view their wish list by clicking on the heart icon and view the cart by clicking on cart icon in the top bar.

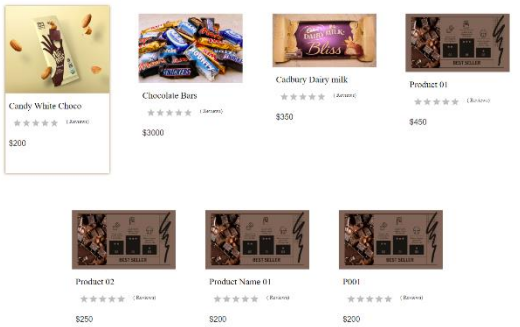

Figure 2: Home page featured products part

User can view the featured products by scrolling down in the home page. User can click on a product and user will be directed to the product page.

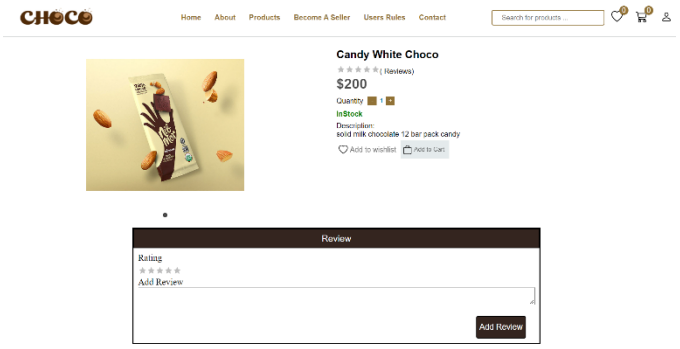

Figure 3: Single product page

User can view the products here. User can specify a number and click on add to cart button to add the product to the cart. User can click on the heart icon to add the product to the wish list. User can also use the review part to add a review on the product.

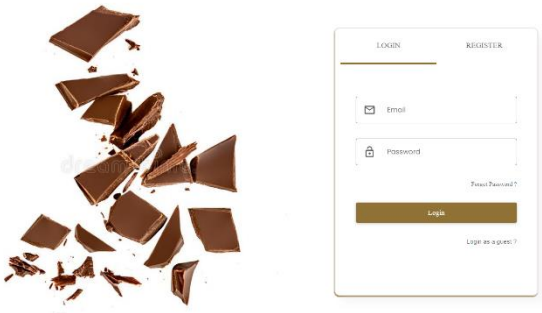

Figure 4: Login page

User can login to the application using this page. User can also register by clicking on the register icon.

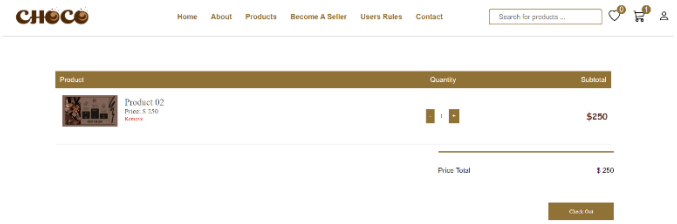

Figure 5: Cart page

User can view the cart by this page. User can increase or decrease the number of products in the cart from here.

e-ISSN: 2250-0758 | p-ISSN: 2394-6962
Volume-12, Issue-5 (October 2022)
https://doi.org/10.31033/ijemr.12.5.42

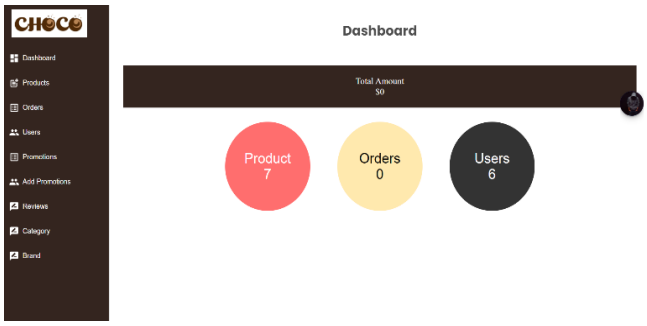

Figure 6: Dashboard page

Admin can view all the products, orders and users that are registered in the application from this page. Admin can navigate using the sidebar.

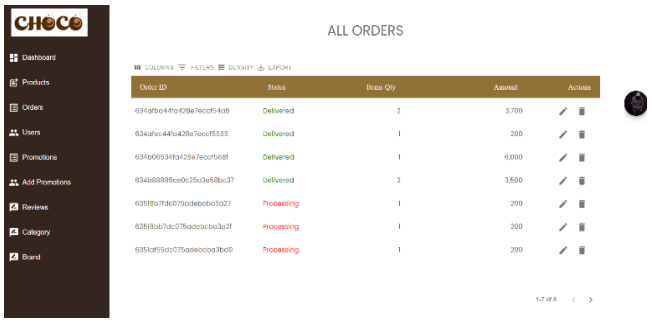

Figure 7: All orders page

Admin can view all the products from this page.

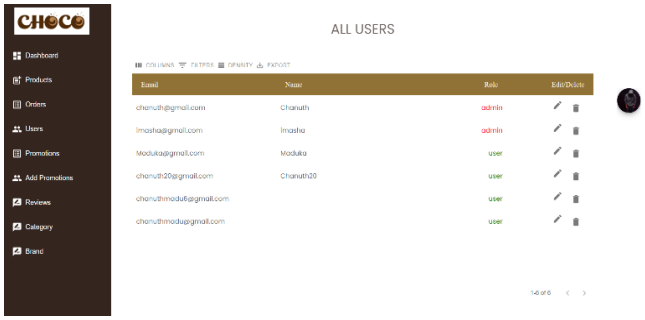

Figure 8: All user's page

Admin can view all the users from this page.

Figure 9: Promotion Page

Admin can view all the promotions from here. Admin can click on the add new promotion button to add a new promotion. Admin can also change the active status by clicking on the active button. Admin can go to the generate report page using generate report button.

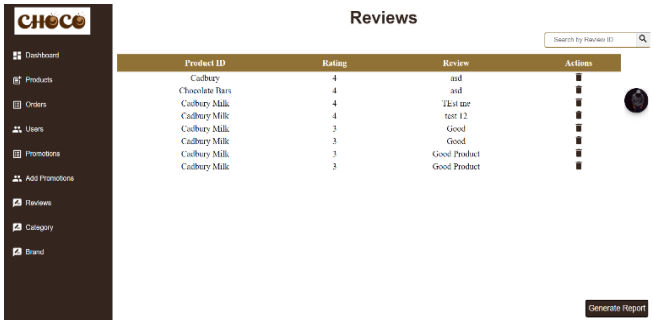

Figure 10: Review page

Admin can use this page to view all the review. Admin can search for a specific product to get all the reviews on that product.

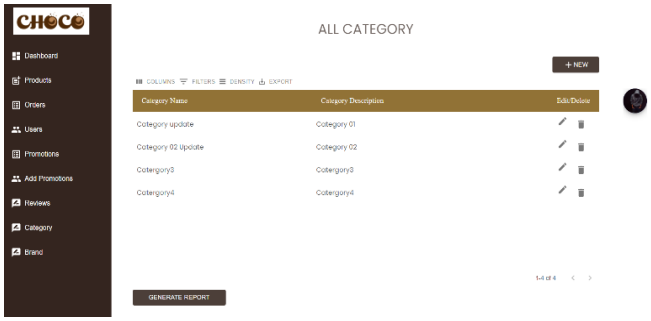

Figure 11: Category page

Admin can use this page to view al the categories. Admin can add a new category using the add button. Admin can use the generate report button and export button to generate report.
Admin can edit the brand using this page.

e-ISSN: 2250-0758 | p-ISSN: 2394-6962
Volume-12, Issue-5 (October 2022)
https://doi.org/10.31033/ijemr.12.5.42

1) GitHub

Firstly, we created a boilerplate and a good folder structure for the frontend. Then it was added to GitHub and each member created a branch for each function. Separate branch named development was created to merge the branches before pushing into the main branch. Using GitHub for the frontend made the collaboration with team members very efficient.

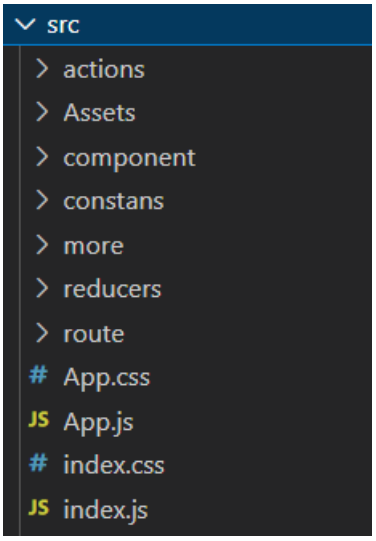

Figure 12:- Frontend Folder Structure

Above image shows our frontend folder structure. Actions folder contains all the API calls of our application as functions. We exported those functions from there and imported the function when necessary, so, we don't have to write the same API call again and again. Assets folder contains all the images and media files we used for our application. If our application is an element, component [15] folder contains all the molecules and atoms of the element. Constants folder contains all the constants in the application. More and Reducers contains the files we used to integrate Redux into our application to store global variables. Routes folder has all the routes of our application

2) Axios and Redux

For all the API calls we made for our backend we used Axios. All the API calls were coded in the actions file as a function in actions folder and were used again and again. Main Axios calls we used were GET, POST, DELETE, PUT and PATCH request. GET is used to fetch data. POST call is used to upload data. PUT and PATCH was used for update purposes and DELETE request was used to delete data. We also used redux to store global variables, so it is easier to manage state. Using Axios and Redux has made the web application development much easier and more flexible. And passing data is done in JSON format. In the Axios post request we are passing the JSON object. And using Axios for API calls is much more organized than simply using fetch function.

3) Material UI and React

Material UI, also known as the MUI, is a react component [15] library, we used to design our application. This library offers variety of components with default styles which make the frontend interface development super easy and efficient. It also allows us to write our own style for each component. This library eased our work and made the development process much efficient, and the work done was equally beautiful.

When developing the frontend, we mostly used functional components, because using hooks is much easier than managing states in a class component. Still, we used class components too. Using react for frontend development was much efficient because of reusability of components. And we also used conditional rendering in few places. When specifying the routes for the frontend. We had protected routes which will only allow admins to access them. This is one of the security aspects of our application. Admins also have access to a simple but complete dashboard so they can manage the business process easily. They can view all the ongoing activities from there. Also, customer user interface is simple, and they have the option of adding products to cart and wish-list which makes their shopping experience better.

4) Dynamic Messages

To make the application more user friendly we have implemented a notification system using Material UI Toast component. All the user interactions with the application will give user a notification and informing that task was a success or a failure. It will also ask for user confirmation for certain tasks. For an example, to delete a data admin will get a confirm deletion message.

5) Functionalities of the application

The main users of our application are customers and admins.

As a customer, a user can register into our application. Customers have to verify their email using the link sent to their email after registration. Then they can login to the app using the login credentials. Customers can navigate the app to select the chocolate product they want to buy; they can add it to their wish list or cart. They can select the amount they want to add to the cart, they can also clear the cart. Customers also have the option to give review to a product. Each of these functions of customers can do will give a response to them as a notification, whether the action they want to perform was successful or not. Users can also upload images for their profile. And they can manage their profile details from there.

As an admin, user have to login to the application. Admin will have a function to view dashboard. This is not available to customers. Admins can view all the activities of the application via dashboard. In the side bar they can

e-ISSN: 2250-0758 | p-ISSN: 2394-6962
Volume-12, Issue-5 (October 2022)
https://doi.org/10.31033/ijemr.12.5.42

navigate to product management, user management, order management, category management, promotion management and review management. They can do the basic crud applications in each of these pages. Admin can also add an image of a product when creating or editing a product. And in the promotion management page, they can change the active status of a promotion. In the user management stage, admin can specify the user role. In the order management stage, admin can change the order status. In each of the above-mentioned pages, delete function will ask user for confirmation and each function will give user feedback as a notification. Error messages will be shown in red and success messages in green. Other than that, there is also report generation function. Admin can use the export function just above the table to export the table data as a CSV file or as a PDF file. Admin can also directly print it. In each of the above pages there is also filter functions and search functions to make user experience better.

F. Backend

1) GitHub

Same as the frontend, our backend also started with a boilerplate and a GitHub repository. Branches were made for each main function and separate branch named development was also created in the backend repository for merging purpose. As we are developing from four different machines, GitHub made the development process and merging code much easier and efficient.

As you can see in the below image, this is the folder structure of our backend.

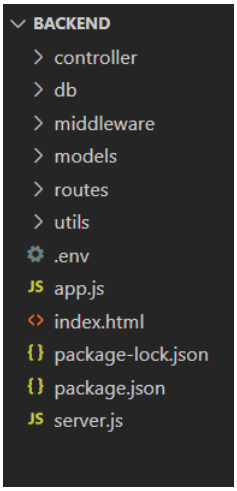

Figure 13: Backend Folder Structure

Controller folder has all the files with backend logic and functions. They are exported as constants. DB folder hold the file which initiate the connection with the database. This connection is also exported. Middleware folder holds the files which has the code to catch errors and authenticate users. Models folder holds all the files which describes the database schemas. Routes folder contains all the routes described by us. Utils folder has files which contains code for the backend feature, token generation for authentication, sending mail for verification and handling errors. .env file holds all the environment variable we have mentioned like Mongo DB URL for the cluster.

2) Mongoose and CORS

Mongoose is the frontend to Mongo DB; We used mongoose for model creation. This mongoose library offers wide variety of techniques for both creating schema and storing data in databases. Models assist us in storing data in the MongoDB database. Also must mention the importing and using CORS here because we faced a CORS error and struggled a lot there. We imported CORS in the backend and used it there to overcome that issue.

3) Express JS

Express is a framework purely based on Node JS. Express offers strong structures to develop an efficient backend. Express support HTTP and middleware method which makes the APIs made using Express JS relatively more robust and simpler to use. Express offers many tools for us to create a better backend environment. We must also take note that famous NodeJS frameworks like JS Sails and MEAN includes Express JS as a core function. We break down the simple route into two parts as controller and routes. This makes managing the routes easier because complex functions are in a different file. Also, we can reuse the same function again and again which is increasing the code reusability.

4) Database

Database we used is MongoDB, which is a very popular NOSQL database. It is also open source and millions of users utilizes it for work. Few reasons for choosing MongoDB as the database is it is open source, offers high performance and high availability cost free. It has the concept of Collection and Documents. It is also a cross- platform data storage system which opens up many paths in development. The data we send to the Mongo DB was passed as JSON objects. The main Mongo DB functions we used are find,

V. PROPOSED SYSTEM

Our proposed system consists of a simple but elegant looking responsive user interface developed with react and robust backend developed with Express JS. The database used for this is MongoDB. The efficiency of the system is increased by using the same code again and again with react components and efficiency of data transmission is ensured by passing JSON type data into the database. Authentication is there to ensure the system safety and vibrant notifications will make the user experience better. User input validations and search function will increase the user experience. Report functions are also available for the

business purposes. Usage of Redux and Axios makes the code less complex and more dynamic.

VI. MODERN TOOLS

A. Description

The tools we used for our application is Azure Boards [20] , GitHub [19] , SonarQube [17] and Selenium [18]. These tools were helpful because it made the developing much easier. The best tool among them is GitHub which made the merging of code and collaborating with the team members much easier. Azure Board were also helpful when it came to collaboration and keeping track of the work we have done. And assigning task to a member after finding a bug or an issue with the code is made easier with azure boards. SonarQube and Selenium was also helpful in debugging and ensuring the code quality. Let's see how each tool was helpful for us in this project.

B. Azure Boards

We used Azure Boards [20] to manage the work we did. Our team leader first divided our tasks and put it into Azure Board and assigned each task to us. We then put our own work and put it on Azure Board as a task and signed it completed when we did that task. After we used SonarQube we found few bugs and Azure Boards lets us to state those bugs clearly. Our leader put those as a bug fix task and assigned each member to fix those bugs. Basically, what we achieve from using Azure Boards is managing our work efficiently and collaborating with team members easily. It also helps team members to track other members work and help them.

C. GitHub

We used GitHub [19] mainly to collaborate with each team member when building the project. Through GitHub we can automatically merge our code rather than doing it manually which makes using GitHub more efficient. Firstly, we created two separate GitHub repositories for our Frontend implementation and Backend implementation. For the frontend repository we created 8 branches for each function and a branch name development to merge our code before pushing to the main branch. Keeping track of commits also allowed us to track the work done by other members of the team. Opening a pull request and reviewing the code before merging is also very helpful when it comes to resolving conflicts. Thus, GitHub plays a huge role in making the development of E-commerce more efficient and easier.

D. SonarQube

We used SonarQube [17] to test the code quality. Even though our code was working fine there were many issues when we first ran the code with SonarQube. There were many imports which were not used, and many best practices were breached. It also showed us what our project was missing in security, reliability, and maintainability aspects. We fixed those issues with help of this tool. Maintaining the code quality of the e-commerce web application was made efficient and quicker with SonarQube.

E. Selenium

Using selenium [18] was easy because the testing is automated. All you have to do is install the selenium extension, start the test and selenium will do the job. The hard part is fixing the issues we found. We tested the application few times and we also found many issues that was not found with SonarQube. We fixed those issues easily because the report we get from selenium is descriptive. Debugging the code was made efficient and quicker with selenium [19]. It also saved a lot of time and fixed many issues in our code thus increasing not only efficiency but security also.

VII. DISCUSSION

Our purpose was to find out whether the efficiency of an e-commerce app will increase with MERN [16] stack and usage of modern tools. What we found out was it actually increases the efficiency. React itself is making use of the same code again and again, adding Redux, Material UI and Axios libraries to it not only increases the efficiency, but it also makes the development process much easier. Paired with using JSON for data transmission and robust APIs from Express JS backend make the application more efficient. Using GitHub made the merging of code much easier and efficient because it will automatically merge and points out if there are any conflicts there. SonarQube was also really helpful for us to ensure our codes standard. It was also helpful to find many issues we had with our code. Changes made after that also increased the code efficiency. Using selenium to automate the testing of our web application saved us much time because otherwise it will take a long time to write test cases. And it helped us find many bugs and errors in our application which helped us to increase the application efficiency. Looking at above mentioned details we can see that using modern tools with MERN stack [16] we can make an E-commerce web application more efficient.

VIII. CONCLUSION

The main goal is to create an E-commerce web application that sells chocolatier items online using the frontend, backend, and database. This website is a completely functional website, starting with login authentication, admin authorization, adding things to the shopping cart, and using the payment gateway. Any branding or category sector can be used, whether on a small or large scale. They can easily use the web application, and without much effort, they can add products and establish

new categories. The buyer will find it quite appealing to view the products while seated at home or at work. Small-scale businesses will benefit greatly from being able to sell products directly to consumers rather than through wholesalers or huge retail intermediaries which make both parties happy. And as we have used more modern tools like Azure boards for the project management and GitHub for version controlling that make much easier in the development stage and also for the planning stage. Also Testing tools such as Selenium and SonarQube are useful for having error free web application and it benefits [6] both the client side and server side.

ACKNOWLEDGMENT

The authors gratefully acknowledge the contributions and the guidance of our Lecture in charge, Senior Lecturer Dr. Dilshan De Silva and our assessor Miss R.R.P De Zoysa for their work on the document.

REFERENCES

[1] Mai, N. (2020). *E-commerce application using MERN stack*.
[2] Ullah, S. E., Alauddin, T. & Zaman, H. U. (2016, Jan). Developing an e-commerce website. In: *International Conference on Microelectronics, Computing and Communications (MicroCom)*, pp. 1-4.
[3] King, D. N. & King, D. N. (2004). *Introduction to e-commerce*. Prentice Hall.
[4] Nemat, R. (2011). Taking a look at different types of e-commerce. *World Applied Programming, 1*(2), 100-104.
[5] Niranjanamurthy, M., Kavyashree, N., Jagannath, S. & Chahar, D. (2013). Analysis of e-commerce and m-commerce: advantages, limitations, and security issues. *International Journal of Advanced Research in Computer and Communication Engineering, 2*(6), 2360-2370.
[6] Niranjanamurthy, M., Kavyashree, N., Jagannath, S. & Chahar, D. (2013). Analysis of e-commerce and m-commerce: advantages, limitations and security issues. *International Journal of Advanced Research in Computer and Communication Engineering, 2*(6), 2360-2370.
[7] *Advantages and disadvantages of e- commerce businesses*.
[8] JavaScript .
[9] NodeJS Introduction.
[10] NodeJS Pros and Cons.
[11] NodeJS use cases.
[12] Express/Node introduction.
[13] MongoDB.
[14] Virtual-DOM.
[15]Component.
[16] E-Commerce web Application by using MERN Technology
[17] SonarQube.
[18] Selenium.
[19] What is GitHub?.
[20] Azure Boards- How to get started with agile planning on azure?.
[21] Selenium Tutorial – Learn how to perform automation testing using selenium web driver.