Summer 2021 CS 687 Capstone Project Progress Report
An Admin Management Dashboard with MERN for SME E-commerce

Pengfei (Chuck) Liu
Advisor: Sam Chung & Ahreum (Amy) Ju
MS in Computer Science
School of Technology and Computing (STC)
City University of Seattle (CityU)
liupengfei@cityuniversity.edu, juahreum@cityu.edu, chungsam@cityu.edu

Abstract

E-commerce has revolutionized the global retail industry, and E-commerce sales have tripled in the past seven years. Affected by the pandemic, the number of independent E-commerce companies ushered in exponential growth. As an invisible battlefield in the E-commerce competition, the admin management system aims to help E-commerce manage inventory and user data more conveniently and effectively. It is used to support business, optimize service processes, improve service efficiency, and provide data analysis functions to provide references for overall business adjustment and optimization. This project implements the admin management system mainly with MongoDB, Express, React, and NodeJS. By using MERN stack, it improves the overall performance of the system and reduces maintenance cost. The frontend is implemented with the single-page application which only renders necessary data to the page. The back is composed of MongoDB, Express, and NodeJS. The Express MVC helps to reduce the coupling between each part. The project increases the QPS by 60% by deploying with 2 AWS EC2 instances.

Keywords: Admin Management Dashboard, MERN Stack, Microservices Architecture, E-commerce

1. INTRODUCTION

Problem Statement

In order to meet the increasing demand for E-commerce platforms, higher requirements are put forward for the management capabilities of the back-end system. Implement a more efficient admin management system to fit E-commerce platforms' needs has become a crucial research topic. Due to the nature of the E-commerce business model, the admin management system must meet features of stability, scalability, and strong security (Yun, 2018). When designing product architecture, developers should fully consider the needs of business development and isolate each module. For example, build a commodity center for commodity modules, build a user management center for user modules. Only when there is a modular idea in product design, when business adjustments and new functions are added, development can be carried out quickly to avoid things that affect the whole business operation.

Motivation

According to the National Retail Federation (National Retail Federation) report, the consumption rate in the United States increased by 3.6% year on year to approximately 755 billion dollars during the Christmas holiday season in 2020. Online shopping sales were the main driving force for total sales growth, increasing by at least 20% to approximately US$202.0 billion (Sherman, 2021). Data proves that traditional physical stores and emerging E-commerce sales are polarizing, and the pandemic is accelerating the reshaping of the global retail industry. A recent report by McKinsey showed that affected by the pandemic, the number of newly opened E-commerce stores in the United States over the past year is equivalent to the sum of the past 10 years (Buck, et al., 2020). Due to the increase of consumers, E-commerce platforms should promote a full-category and full-scenario development strategy to meet all user needs. With the rapid development of the E-commerce industry, the demand for E-commerce admin management systems has also increased dramatically. The success of a website relies on a high-quality UI design, but the admin management system determines the success or failure of E-commerce platforms (Li, 2021).

A full-featured E-commerce admin management system does not only help E-commerce platforms to efficiently manage stores, but also improve employee work efficiency and meet user's personal needs. Requirements and functions are constantly iterating with the expansion of business, so product architecture and technical architecture are required to be highly scalable and change with business and users' requirements. The management system needs to cluster and integrate these functions for different purposes, split the E-commerce back-end into multiple components, clarify business boundaries, minimize the coupling between components, and efficiently support front-end business.

Approach

In this project, we investigated the MERN stack web development and created a single-page application E-commerce admin management system separated of back-end and front-end. In order to design a suitable management system for E-commerce, we followed the modular design, component design, and engineering model development.

Conclusions

This project implements a fully featured E-commerce management system through the MERN stack, including user and product management. Redux manages the status of application components to reduce the coupling between components, thereby supporting business expansion and reducing maintenance costs. Complete data visualization display through Echarts.

2. BACKGROUND

The admin management application is designed for supporting a certain business process. It is also known as the business-end (2B business) product. A good 2B business product can be measured from the following two aspects: first, the business need is more crucial than personal needs. Second, 2B business products generally have a strong purpose when used or need to reach a certain level of business operations or need to complete certain process approvals (Bein et al., 2020). In brief, when they are used, they are generally not used for personal emotional reasons. When starting to design a system, the first step is to figure out the business needs. Due to the particularity features of E-commerce, it requires to constantly adjust its products to meet the needs of different users. Admin management needs to make timely adjustments to products and user functions at different stages for different development strategies. For E-commerce companies, there are three core and difficult parts: commodities, orders, and inventory. The business logic and interaction between the systems are extremely complex and the rules are diverse.

Developers around the world are working to improve the quality of user interfaces and the development process of building applications to achieve projects and development requirements within a set deadline (Mehra et al., 2021). The MERN (MongoDB, Express, React, Node.js) technology stack is one of the early open-source technology stacks developed with the boom of the SPA and NoSQL. React in the MERN technology stack is a JavaScript library used to build user interfaces. MongoDB is a popular data storage NoSQL database. Node.js is a server-side JavaScript runtime environment, and Express is a web server built on Node.js. The MERN stack development ensures the development process is as smooth as possible. Moreover, React is a component-based development, which completes the encapsulation of functional modules, the decoupling of functions and business logic, and the reduction of code.

3. RELATED WORK

This project designed a single-page application of an admin management system (BMS) with a separated front-end and back-end server for small to medium E-commerce including front-end web applications and back-end applications. This BMS covered four major features include user management, classification management, commodity management, authority management for organizations to better manage their user and product information. This project uses the MERN stack to implement a responsive admin management system to help E-commerce businesses to run more efficiently and improve user experience. And it will follow the MVC framework to complete a low coupling E-commerce BMS.

Literature Review

According to Lv (2021), the biggest role of the admin management system is to support users' businesses. If an E-commerce company does not have a place to upload products and no place to process orders, then the user can do nothing. The second crucial aspect of the E-commerce management system is efficiency; this part is mainly about the management of content. Lv describes the meaning of the management system as to improve the efficiency of internal personnel, so the most important feature of considering the necessity of a management system is the cost performance. Torabizadeh, Yusof, Ma'aram, and Shaharoun (2020) illustrated a high-quality admin management system can reduce the time consumption of employees; however, most third-party management software on the market contains too many functions beyond the organization's needs, which caused employees to waste extra time on these operations. The training of each newcomer means double the investment of an old employee and a new employee, especially for a small company with limited funds and employees. A well-designed admin management system can reduce the cost of employee training so that employees can start regular work quickly by simple learning. Souri and Rezaei (2017), believed that to build an admin management system, developers need to focus on the following three dimensions: the business process is the soul, low coupling of functional modules, security, and integrity of data transmission.

An effective software development approach to reduce coupling is to separate the front-end and back-end development. Mardin (2018) explained front-end and back-end separation has become an industry-standard way for Internet project development, decoupling front-end and back-end development. And the separation of front and back ends will lay a solid foundation for large-scale distributed architecture, microservice architecture, and multi-terminal services in the future. The core idea of front-end separation is that the front-end HTML page calls the back-end API interface through AJAX and interacts through JSON data.

Malik and Kim (2017) compared the two most used API architecture styles in their articles about Restful and SOAP. And they summarized few advantages by following the RESTful API architecture. Arcuri (2019) also said RESTful API allows developers to separate the front-end and back-end server, also reduce the data throughput, and move security issues focus on the interface. It also provides some solutions to test the API design include both white-box and black-box tests.

Hasija and Kumar (2016) discussed MongoDB as a database suitable for agile development, its data model can be flexibly updated with the development of the application. Stepantsov mentions MongoDB is designed for high performance and high availability database usage. Using the advantages of in-memory computing, MongoDB can provide high-performance data CRUD operations.

Review Conclusions

By studying the MERN stack development, it can help build a single-page application responsive admin management. Through different E-commerce needs of the admin management system, some functions can still be solved manually or by other methods instead of developing the system, especially for teams with

a small number of people. So, during the designing process, we will consider the frequency of use of this function, how much time can be saved by the staff, and how much system risk this function can reduce. MongoDB increases the efficiency of using CRUD operation, which helps to improve the user experience of based management functions of this project.

4. APPROACH

User Requirement

The E-commerce admin management system is a back-end management system provided to relevant business personnel. They can use this admin management system to perform management functions includes user management/commodity classification management/commodity management/authority management. Separate front-end and back-end have become the industry standard for web project development which aims to decouple front-end and back-end development. The E-commerce admin management system is composed of several parts, and with the development of the company's business, functional modules will continue to expand. Since each subsystem is not isolated from other parts, the product architecture determines the requirements and design, and the technical architecture determines the technical framework and performance (Stallinger et al., 2011).

Design

The core idea of separate front-end and back-end is to allow the front-end HTML page to call the back-end API interface through AJAX, and interacts through JSON data, thereby reducing the concurrency/load pressure of the back-end server and improving performance and scalability.

The admin management system should follow approaches of modularization, componentization, and engineering model development (Zhang et al., 2017). The updated iteration of a product is dependent on the collaboration of multiple teams such as product, design, development, and testing. With the improvement of product functions, the supporting team behind it is also growing. Product design and iteration are driven by componentization can increase team efficiency and reduce costs. The traditional design process is mostly parallel with attributes and multiple roles. The workflow from requirements to design, from design to display will involve different requirements. The lack of overlap between the same role and lack of communication will lead to a lot of repetitive work, which will cause redundancy works for each project cycle and management. These approaches can reduce the coupling, avoid redundancy and optimize the performance; therefore, management personnel can load resources on demand.

Implementation

This project followed the MERN stack to implement the admin management dashboard for SME E-commerce. In traditional web applications, DOM manipulation is generally a direct update operation that has been approved as a less efficient approach. To minimize the operation of the DOM, this project used React for the front-end. React provides a lightweight virtual DOM instead of direct DOM operations; thereby improving the efficiency of the entire project. To complete the single-page application, this application requires tools to manger routers. React Router is an important part of the React system, it was used for implementing URL management, switching components, and changing state. However, react is just an abstraction layer of DOM, not a complete solution for web applications. There are two aspects, it doesn't involve: code structure and data transfer between components. By using redux, it can achieve state sharing between components. Data interaction between the front and back ends was realized through the Axios and API tests implemented via Postman. The back-end construction of the project used the combination of Node, Express, Mongoose, and MongoDB. The modularization of this project was implemented by the ES6 and CommonJS. The data visualization was done by D3JS and Echarts-for-react.

Technologies Used

The front-end of this project used the React, bootstrap, axios, webpack and ES6, the backend used nodeJS, express, and mongoDB.

5. DATA COLLECTION

This project used the case analysis approach for the data collection. JingDong (JD), as one of the massive B2C online retailers in China, has ranked 59th in Fortune Global 500 in 2021 (Fortune, 2021). This project used the case analysis approach of JD for exploring the formation path of the admin management system from the different stages of B2C E-commerce enterprise development. JD was founded in 1998 as an SME that mainly sells electronic products. It has grown from an SME that sells less than 100 electronic products to today's commercial giant with annual revenue of over 110 billion dollars. While maintaining rapid development over the years, JD has accumulated hundreds of millions of loyal users and accumulated massive amounts of real

data. JD.com provides the most suitable products and services for more than 300 million active users. At present, JD Retail Group's third-party platform has signed more than 210,000 merchants, achieving full category coverage.

The usual role of the admin management system is to create, store, and process inventory and user information. It manages the normal operation of the entire E-commerce system. Through case studies, understanding the basic needs of the management system includes studying the needs of user management, inventory management, and product classification. And as the business grows, different management modules will be managed by separate departments. Authority management should also meet the actual needs of customers and have the most flexible authority system. As a case of transforming from a 3C online retailer to a comprehensive online retailer, JD's success helps to understand different product life cycles and make a reasonable combination according to the company's business and strategic goals, thereby improving the efficiency of the entire management system.

6. DATA ANALYSIS

JD provides various data information includes users, merchants, products, including content information, review information, and user interaction behaviors of merchants and products information. Figure 1 shows a small sample of JD's raw data. The raw data can divide into five major fields: behavior, user, account, product, and merchant. With combining of these five major fields, the admin dashboard management system analyzed user needs from four aspects: sales, members, products, and customers.

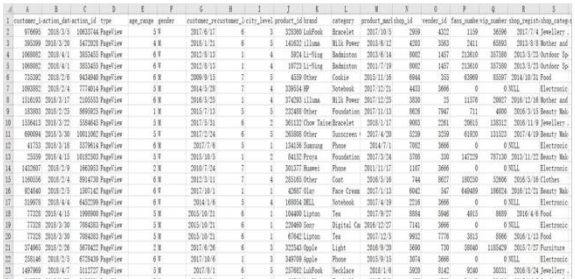

Figure 1. A Sample of JD's raw data.

The sales analysis mainly shows the changes in the number of orders for each behavior category of the user. It includes the changes in the total number of orders for the user's purchase of products on JD and the sales volume of the product categories purchased by the user.

Field	Type
customer_id	Bigint
product_id	Bigint
action_date	Date
action_id	Bigint
type	String
age_range	Int
gender	String
customer_register_date	Date
customer_level	Int
city_level	Int
brand	String
shop_id	Bigint
category	String
product_market_date	Date
vender_id	Bigint
fans_number	Int
vip_number	Int
shop_register_date	Date
shop_category	String
shop_score	Float

Figure 2. JD's data types.

User analysis bases on the user's gender, age, occupation, location; through the user's basic attributes and behavioral attributes, they help the merchants to determine the user portrait of the stable user population, so they can launch more new products to attract these users. The data shows that the number of customers of JD increased rapidly from 2008 to 2010 because JD launched a series of new promotion activities after it went online in 2007. After 2012, the rate of new user growth has dropped significantly. The number of users increasing each year has remained stable. JD maintains a stable growth rate in sales by targeting specific groups of people. By dividing the user registration time range, users registered after 2016 defines as new customers, users registered during 2010-2015 are mid-term users, and users registered before 2010 are old customers. Figure 3 shows that mid-term customers are the main force of platform sales. The platform can enhance user stickiness with old users by having some promotion activities.

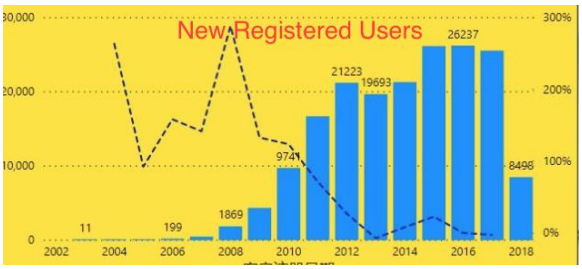

Figure 3. New Registered Users.

Overall, the electronics, clothing, and beauty products on the JD platform are the best-selling categories; among them, users prefer to buy mobile phones, notebooks, digital cameras, and other electronic products on the JD platform. The users of the JD are mainly male users. The membership level of male users is higher than that of female users, and the repurchase rate of female users is only 20%, which is only half of that of male users. Improving female users' experience of page and product experience on JD, optimizing female products and related store brand adjustments, and promoting more promotional activities that are conducive to women's purchases will bring more users to the JD. Based on the analysis of JD, it provides aspects that need to cover for the admin management dashboard. Our project uses three different patterns including bar chart, pie chart, and line chart, to provide better data visualization for merchants so they can make a better decision based on these data.

7. FINDING

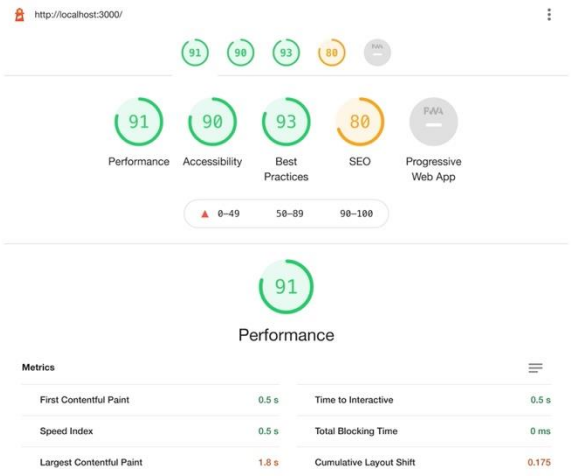

Figure 4. The overall performance of Admin Dashboard

Figure 4 shows the measure score of the project by Google's lighthouse. There are two aspects that need to pay attention to: performance and SEO. Overall, the project reached a higher performance with a score of 91 out of 100. The use of React plays a key role in this project. It allows developers to build complex and powerful products faster and implement iteration more efficiently. The JavaScript revolution has rewritten the rules of web development. It allows the development of single-page applications using JS technology to obtain some data through JavaScript, and then build a dynamic website. However, no matter how popular this method seems, DOM manipulation is still relatively slow. React.js excels at building dynamic and engaging web interfaces, surpassing other JavaScript frameworks (such as Angular and Ember) because of React's virtual DOM. React through the introduction of virtual DOM, instead of updating the entire components every time during the update process to achieve SSR, The React compares the virtual DOM and previous DOM then only updates the differences to the DOM, thereby improving the performance of the entire website. In addition, React's reusable components make the code of the project modular and more organized, and they can be reused anywhere without any additional settings. And combined with the virtual DOM can ensure that the UI is updated quickly and efficiently. Moreover, isomorphic JavaScript can quickly render web pages. The dashboard is usually built as a client-side SPA. In a highly complex or high-load SPA, it only takes 3-5 seconds to fully load everything from the server. Figure 5 shows the load test after scaling with the AWS. For 5000 clients, the average response time was only 16ms.

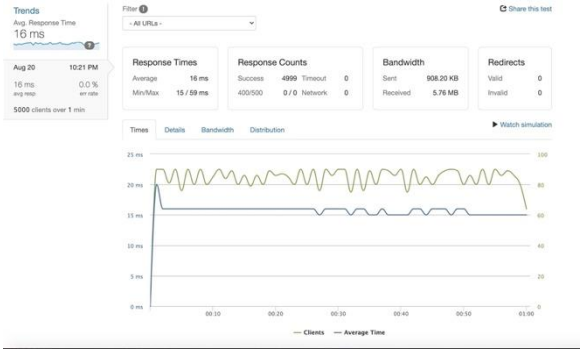

Figure 5. Load test after scaling with AWS.

However, building a client-side rendered SPA may have some defects that can cause serious business problems, as a tradeoff of SPA, SEO performance is always bad for SPA. SEO stands for search engine optimization. As a SPA built with React, a web application running only on the client-side cannot provide HTML for crawlers; this generally leads to poor SEO performance. If company's business mainly relies on users' traffic to expand its revenue, then this will be a serious

problem. But as a dashboard application, SEO is not an important factor.

8. CONCLUSION

The fundamental role of the management system is to create, store and process information. It manages the normal operation of the entire business. The E-commerce management system provides the management of merchandise and users; and with the increase of business, it provides authority management to enable managers to perform their duties and improve efficiency. This project uses the separation of front and back ends, the front end is built by React scaffolding, thereby reducing project configuration time and deployment time. The front end also implements a single-page application (SPA) in order to improve the overall performance of the application. The backend completes a data-driven service using NodeJS, Express, and MongoDB. A Restful API is created to obtain data, which reduces the coupling between servers.

9. FUTURE WORK

There are two parts that need to implement for this project. First, in order to handle more products in a more efficient way, this project will use Redis to cache data. As a cost-effective method, the Redis will also deploy on the AWS ElastCache. Secondly, the next iteration of the project will include more features such as logistic management, order management, and financial management.

Demo Link

10. REFERENCE

Aggarwal, S., & Verma, J. (2018). Comparative analysis of MEAN stack and MERN stack. *International Journal of Recent Research Aspects*, 5(1), 127-32.

Bein, A. S., Graha, Y. I., & Pangestu, A. P. (2020). Pandawan Website Design Based Content Management System As Media E-commerce Transaction. *Aptisi Transactions On Technopreneurship (ATT)*, *2*(1), 87-97. DOI: https://doi.org/10.34306/att.v2i1.73

Bhatti, A., Akram, H., Basit, H. M., Khan, A. U., Raza, S. M., & Naqvi, M. B. (2020). E-commerce trends during COVID-19 Pandemic. *International Journal of Future Generation Communication and Networking*, 13(2), 1449-1452.

Latha, A. G., Raju, R. N., Satyanarayana, C., & Srinivas, Y. (2011). Hotel management system an E-commerce application using resource description framework (RDF). *International Journal of Computer Science and Information Technologies*, 2(5), 2267-2272.

Linton, J. D., & Solomon, G. T. (2017). Technology, innovation, entrepreneurship and the small business—technology and innovation in small business. *Journal of small business management*, *55*(2), 196-199.

Lv, X. (2021). Applications of the Small and Medium Enterprise Management System Using the Clustering Algorithm. Mobile Information Systems, 2021

Mehra, M., Kumar, M., Maurya, A., & Sharma, C. (2021). MERN Stack Web Development. *Annals of the Romanian Society for Cell Biology*, *25*(6), 11756-11761.

Paivarinta, T., & Munkvold, B. E. (2005, January). Enterprise content management: an integrated perspective on information management. *In Proceedings of the 38th Annual Hawaii International Conference on System Sciences* (pp. 96-96). IEEE. DOI: 10.1109/HICSS.2005.244

Stallinger, F., Neumann, R., Schossleitner, R., & Kriener, S. (2011, May). Migrating towards evolving software product lines: Challenges of an SME in a core customer-driven industrial systems engineering context. *International Workshop on Product Line Approaches in Software Engineering* (pp. 20-24). DOI: 10.1145/1985484.1985490

Stepantsov, A. (2018). Development of a centralized database backup management system with nodeJS and React.

Yun, Q. (2019, July). Design and Implementation of E-commerce Platform based on Vue. js and MySQL. *Computer Engineering, Information Science & Application Technology* (ICCIA 2019).

Zhang, N., Cao, Y., & Zhang, S. (2017, May). Research of web front-end engineering solution in public cultural service project. *Computer and Information Science (ICIS)* (pp. 623-626). IEEE DOI: 10.1109/ICIS.2017.7960067