Developing a configurable Whitelabel solution for an Ecommerce platform

Pavankumar V Badiger[1], Utkarsh Singh[2], Prof. Jyoti Shetty[3]
[1]Student, RV College of Engineering, Bengaluru, India
[2]Student, RV College of Engineering, Bengaluru, India
[3]Assistant Professor, RV College of Engineering, Bengaluru, India

Abstract: *With the increase in people tending towards online shopping, any seller would want to set up their own E-commerce platform where they can sell their products. The easiest approach would be using a white-label E-commerce website. Many services on the market allow users to set up their own web applications by giving them a set of templates which they can configure. The White-label Ecommerce web application should have components that are configurable for different sellers using a Content Management System. The website is built on Next.js framework on top of Nodejs, Material UI is used for styling the components of UI which is developed using the mock-up designs built for reference. The seller configuration data are stored in Postgres DB and fetched using APIs hosted by Fastify. The product catalogue and cart details are fetched from the core Digital Retail Backbone APIs of existing e-commerce websites. A Content management system is also built using Typescript which will be used by sellers to configure the website at any time. A proper functioning e-commerce web application with home page, product listing page, product details page and a cart page are developed. Sellers/Brands can easily customize the look, feel and functionality of the various components in their website; Add pages, content and functionalities according to their needs and requirements. The sellers configure their platform using Jarvis – a content management system. A proof-of-concept application is built, and project will be continued in development to add more features and push this application to production. With this project, the barriers to entry in the E-commerce space are very low and minimal investment is required to get started on e-commerce journey. Seller registration, setting up one's store is an easy process. Such marketplaces expose a seller/brand to millions of potential customers. Anyone can start selling quickly and begin leveraging services to start, customize and change your ecommerce business forever.*

Keywords: **Whitelabel, E-commerce, Digital Retail, Content Management System**

1. Introduction

Several businesses who sell their products in stores don't have their own Ecommerce website. The idea of developing this platform as a service is generated targeting these business partners. The ability to build this platform using the existing architecture of an Ecommerce website, makes it stand out from the competition available in the market. The development of this service will be much easier, faster and based on already existing industry level architecture.

People now are more comfortable with having a lot of options displayed on their screen and getting the similar information available in their hand while sitting at their home. Many Ecommerce web applications provide rich user experience. Ecommerce giants like Amazon and Flipkart in India are growing in revenue rapidly. When the pandemic hit the world and people were forced to stay at home and all the shops were closed, the real need and use of Ecommerce websites was recognised. Companies have started to give more importance to their online websites. A lot of the companies who don't have their own website are trying to get into this online business. This is one of the reasons why this project is of importance, using this website the sellers can configure their own website. As we are reusing some of the existing resources from an Ecommerce website, the time and efforts to build this whitelabel platform is reduced compared to other websites which are built from scratch.

The Whitelabel Solution - The website is built on Next.js framework on top of NodeJS, Material UI is

used for the components of the user interface, the seller configuration data are stored in PostgresDB and fetched using APIs hosted by Fastify. A Content management system is built using Typescript which will be used by sellers to configure the website at any time. The product catalog and cart details are fetched from the core DRB (Digital Retail Backbone) APIs of an Ecommerce website. The seller configurations are stored in Google Cloud Platform, so whenever the seller updates the seller configurations in the CMS, the changes must be reflected in the E-commerce website. This needs for the CMS and Ecommerce web-app to be integrated together, any schema changes made in the seller configurations data in one application should also be reflected in the other application, although care is taken to ensure that if some part of the data gets deleted in the database, it remains unconfigured in the CMS.

The content management system should allow any seller with a particular seller id to set up their own website, using the CMS the seller enters the first set of seller configurations, which the seller can update and modify at any point of time later. The functions of the Ecommerce platform are to accept these updated seller configurations and change the look and feel of the site. The other functionality of Ecommerce platform is exactly like any other Ecommerce website, i.e., t display home page, product details and listing page and provide functionality for checkout journey. The customer journey remains the same when compared to other ecommerce sites but there is a huge difference in the journey from a seller/brand's point of view.

2. Literature Survey

The current whitelabel Ecommerce platform includes woocommerce, shopify, mercado shops etc. These platforms have minimum features and basic modules, they don't charge transaction fees, support unlimited products, customized themes and product filtering. They have products, Shipping cart, payments, shipping and warehouse modules. These platforms don't provide the full control of operation to their sellers, their digital brand presence is also less. These platforms also charge commissions which the sellers are not satisfied with. Platforms like shopify provide several templates and customizable modules with drag and drop functionality. The seller can have their own categories, collections, and customize the navigation. They are also provided with tools to manage inventory, promotions and gift cards. If the seller wants the ability to develop and customize using code, that functionality is also available.

A Whitelabel Open banking platform is built in [1] using Model View Controller architecture, although more information is given about the background and reasons to build that platform instead of the methodology that is being used to build it. The research done in [2] compares the similarity and dissimilarity on how the user perceives the website among the top retailers based on electronic service quality(e-SQ) using data from the emerging market.

Automated deployment and performance analysis of a Whitelabel solution is done in [3] which is built using static site generators, AWS CDK (S3+Cloudfront CDN) and strapi. It discusses existing implementations and investigates potential solutions and provides a performance benchmark of several web technologies. The research in [4] discusses the use of micro front end to improve performance, reduce download time, leading to the possibility of concurrent development. A micro front end breaks down a large component into smaller parts. It also states one drawback of using micro front end, that it increases the overall development time in case of smaller projects with small number of developers, as developers spend more time on architecture support instead of feature development.

Research and analysis of front-end frameworks and libraries for E business development is done in [5], stating that Angular 2 is suitable for enormous commercial projects as it contains the most comprehensive functions and features, for small and medium scale applications which are developed for live stream, communication and blog, React and Vue are suitable. An interactive social networking component is implemented in an Ecommerce system in [6], which uses md5 to encrypt the password, also protecting the database from sql injection.

Modern Web-Development using ReactJS is discussed in [7], Lightweight DOM (Document Object Model) is used to improve the performance. A static checker is developed in [8] that takes Open API specifications as input, it checks whether the web API requests in JavaScript conform to these specifications. It performs a static analysis which can extract URLs, HTTP methods. The research done in [9] shares insights into the UX design that need to be followed while white labelling a software product. It explains about the synchronization that is needed between business objectives, user goals and technology parameters. It also states that white labeling a software product helps the clients gain competitive advantage over their competitors.

The features which affect user perceived web quality are discussed in [10], which include both technical and non-technical features. It was found that not only technical features like ease of use, design, smartphone and tablet responsivity, information, but also non-technical features like trust, empathy, free shipping and discount affect the user perceived web quality. This research collected the data to process this information only from Italy.

3. Methodology

The methodology followed to develop the Whitelabel platform was an agile scrum methodology , which is a project management system that relies on incremental development. The goal was to make a platform for the sellers so they can develop an E-commerce solution for creating, managing, and optimizing their customer's digital experience; For the E-commerce application, create landing pages, homepage, product details page etc. and provide the option to customize and configure the available details on this page and others. The user interface developed for this application is in accordance with the design mock-ups provided by the Design Team. Material UI is used for the making of this application which is made in ReactJS. JavaScript and CSS is used to make the application more attractive and serve users with better UX. This was followed by development of Backend Services in NodeJS - Back-end development focuses on the side of the website users can't see. Back-end development includes tasks like Building APIs and code to add functionality to Front-end, Troubleshooting and debugging web applications, Database management and Framework utilization. After the services are made, testing is done. The purpose of testing is to expose faults in the interaction between integrated units. Once all the modules have been unit tested, integration testing is performed. We perform Jest Testing on the Front-end components and have included Cypress Tests in our application. We also use Postman to perform API Testing to check whether all parts of the application are working as expected. Development of the Content Management System in TypeScript - The configuration by the seller is done through Jarvis which is a single page application. It enables developers & business users to operate and configure the different Tenant capabilities grouped by business domains. This if followed by deployment of the application - The deployment is done with Kubernetes on Google Cloud Platform (GCP). Gitlab is used for maintaining the code versions and handling the CI/CD pipeline. Gitlab also allows teammates to contribute and work together on the application.

4. Proposed Approach

There are a total of 6 modules in our proposed architecture. In this we discuss the results at every module which are:

Module 1: Development of a basic prototype e-commerce web application - This included the making of basic pages i.e., the home page, Product Listing and Product Details page, Cart page etc. Upon development, we analyze the important features, components and configurable options in these sections and define the project scope and requirements of the service.

Module 2: Development of the Front-end User Interface - The user interface developed for this application is in accordance with the design mock-ups provided by the Design Team. A UI mock-up is a visual representation of a final digital product or website, including layout/hierarchy, color, typography, icons, and other UI elements. Material UI is used for the making of this application which is made in ReactJS. JavaScript and CSS are used to make the application more attractive and serve users with better UX.
Module 3: Development of Backend Services - Back-end development focuses on the side of the website users can't see. It's what makes a site interactive. We can also refer to the back end as the "server side" of a website. Databases are run from a server, which is essentially a remote computer. Back-end development will help manage this database, as well as the site contents stored on it. Back-end development includes tasks like Building APIs and code to add functionality to Front-end, Troubleshooting and debugging web applications, Database management and Framework utilization.
Module 4: Unit testing and Integration Testing - Unit Testing is defined as a type of software testing where individual components of a software are tested. Unit Testing of software products was carried out during the development of an application. An individual component may be either an individual function or a procedure. Integration testing is the process of testing the interface between two software units or modules. Its focus is on determining the correctness of the interface. The purpose of integration testing is to expose faults in the interaction between integrated units. Once all the modules have been unit tested, integration testing is performed. We perform Jest Testing on the Front-end components and have included Cypress Tests in our application. We also use Postman to perform API Testing to check whether all parts of the application are working as expected.
Module 5: Development of the Content Management System - The configuration by the seller is done through Jarvis which is a single page application hosted in Firebase that exposes Osmos (aka Catalyst Core) capabilities on a user-friendly interface. It enables developers & business users to operate and configure the different Tenant capabilities grouped by business domains. e.g., Product Catalog, Inventory, etc. Jarvis is connected to postman collections through patch seller APIs, so whenever a seller wants to update the UI, the seller can make changes in the config server (Jarvis) which updates the postman collection and UI. The Jarvis application is built in TypeScript and unit testing for the different components are also taken care of.
Module 6: Deployment of the application - The deployment is done with Kubernetes on Google Cloud Platform (GCP). Gitlab is used for maintaining the code versions and handling the CI/CD pipeline. Gitlab also allows teammates to contribute and work together on the application.

5. Results

The final product of this project is a full-fledged E-commerce web application with customisable and configurable components. Whenever the seller wants to change any configuration, like change the label on the Home page's "Offers" component to "Discounted products", he just needs to update that in the Jarvis application which acts like a content management system. This is related to the Content Management System, and that can be analyzed by giving different values as input. But for the Ecommerce website of the sellers, we use the Google Lighthouse for analyzing the performance.
Experimental dataset for analyzing the performance of our website is fixed, the website uses the seller configurations which are stored in Google cloud platform, and other related data like product information, cart information search results come from the DRB APIs of an Ecommerce website.
Performance analysis is done using google lighthouse, in the following image, we can see that the performance score of our website is 70. The total blocking time is 0ms. First contentful Paint is 1.7s as well as Time to interactive is 1.7s. We also get the best practices followed report along with the report, and the score for our website is 83.

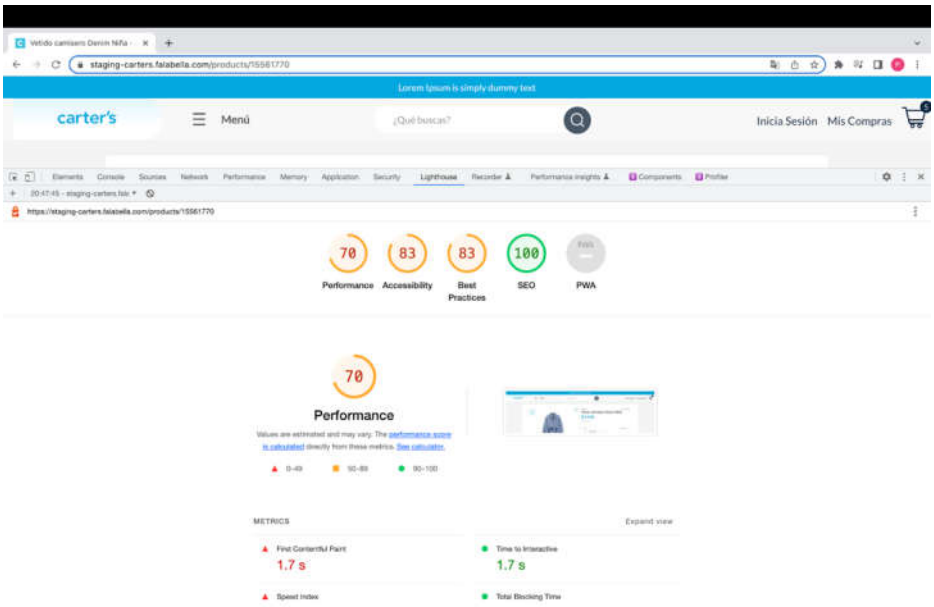

Figure 5.1: Google Lighthouse Performance Analysis Report

6. Conclusion

The current Whitelabel solution has all the pages developed which include home, product details, product listing and cart page. None of the pages have any hard coded values, all the product and cart related API's come from core DRB API's and seller configuration data come from the GCP database. The checkout and payment gateway needs to be integrated, and the CMS webpage needs to be enhanced. This platform will provide a lot of business opportunities to numerous sellers/brands. The website is built using the latest technologies, like the MUI for the components and react web framework and uses Fastify which provides quick routing.

The development time of this platform is reduced because the existing browse and commerce APIs are used to fetch product and cart details. If it was to be built from scratch, then it would require time to build these API's and use them in our Whitelabel Ecommerce platform. The design of User Interface is developed using the mock-up design created by UI developers. Along with the desktop view, the mobile view design is also developed parallelly. Before developing the CMS platform, Postman application was used to update and modify the seller configurations, which change the look and feel of the seller's website.

Whenever a component is introduced in any of the pages, the corresponding unit tests are written, covering the function, branch and statements in that component. After integrating all the components, the cypress test for that page is written. Currently the UI designs for CMS website are being created, once they are ready, corresponding pages for CMS will be developed. The CMS uses PATCH request API for the database containing the seller configuration. There are different tabs provided in the CMS like home page, product listing page, cart page. Entering a new value for configuration will be directly in the Ecommerce platform.

REFERENCES

[1] *Di Fatta, Davide & Musotto, Roberto & Vesperi, Walter. "Analyzing E-Commerce Websites: A Quali-Quantitive Approach for the User Perceived Web Quality (UPWQ)" International Journal of Marketing Studies. 2016, 8. 33-44. 10.5539/ijms.v8n6p33.*

[2] *Sanchit Aggarwal et al. "Modern Web-Development using ReactJS" International Journal of Recent Research Aspects ISSN: 2349-7688, Vol. 5, Issue 1, March 2018, pp. 133-137*

[3] *Kamble, Mr & Kamble, Ms & Jadhav, Ms. "Online Shopping E-commerce Android Application and E-commerce Website Development" International Journal for Research in Applied Science and Engineering Technology. 2022, 10. 3216-3220. 10.22214/ijraset.2022.41950.*

[4] *Jibril, Muhammad. "Development and Implementation of E-Commerce System" International Journal of Advanced Research in Computer Science, 2018*

[5] *Silva, Bruno & Mendes, Marilia. "Building a White Label Open Banking Platform to Improve Financial Inclusion and Human Capital in Russas, Northeast Brazil", 2021*

[6] *Prateek Kalia, Justin Paul, "E-service quality and e-retailers: Attribute-based multi-dimensional scaling" Computers in Human Behaviour", 2021, Volume 115*

[7] *E. Wittern et al., "Opportunities in Software Engineering Research for Web API Consumption," 2017 IEEE/ACM 1st International Workshop on API Usage and Evolution (WAPI), 2017, pp. 7-10, doi: 10.1109/WAPI.2017.1.*

[8] *Niek Candaele, "Automated deployment and performance analysis of a white-label web application" Thesis, Howest University of Apllied sciences, 2021*

[9] *Ashwin Umathay and Aunindra Kumar Sinha. "User Experience Strategy for white labeling a software product" In Proceedings of the 8th Indian Conference on Human Computer Interaction (IHCI '16). Association for Computing Machinery, New York, NY, USA, 2016, 102–110.*

[10] *Pavlenko, Andrei & Askarbekuly, Nursultan & Megha, Swati & Mazzara, Manuel. "Micro-frontends: application of microservices to web front-ends." 2020*

[11] *Silva, Franklin & de Souza, Renata Maria & Machado, Ivan."Taming and Unveiling Software Reuse opportunities through White Label Software in Startups." Conference on Software Engineering and Advanced Applications (SEAA), 28 August 2020*

[12] *L. Li, W. Chou, W. Zhou and M. Luo, "Design Patterns and Extensibility of REST API for Networking Applications," in IEEE Transactions on Network and Service Management, vol. 13, no. 1, pp. 154-167, March 2016, doi: 10.1109/TNSM.2016.2516946.*

[13] *S. Delcev and D. Draskovic, "Modern JavaScript frameworks: A Survey Study," 2018 Zooming Innovation in Consumer Technologies Conference (ZINC), 2018, pp. 106-109, doi: 10.1109/ZINC.2018.8448444.*

[14] *Valášková, Katarína & Majerova, Jana & Krizanova, Anna. "Consumer Perception of Private Label Products: An Empirical Research." Journal of Competitiveness. 2018, 10. 149-163. 10.7441/joc.2018.03.10.*

[15] *X. Huang, "Research and Application of Node.js Core Technology," 2020 International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI), 2020, pp. 1-4, doi: 10.1109/ICHCI51889.2020.00008.*

[16] *L. P. Chitra and R. Satapathy, "Performance comparison and evaluation of Node.js and traditional web server (IIS)," 2017 International Conference on Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET), 2017, pp. 1-4, doi: 10.1109/ICAMMAET.2017.8186633.*

[17] *Farid Aulia Tanjung, Wawan Dhewanto, "Formulation of E-Commerce Website Development Plan Using Multidimensional Approach for Web Evaluation", Procedia - Social and Behavioral Sciences, 2014, Volume 115, Pages 361-372, ISSN 1877-0428*

[18] *Anwar, Nahid & Kar, Susmita. "Review Paper on Various Software Testing Techniques & Strategies." Global Journal of Computer Science and Technology. 2019, 43-49. 10.34257/GJCSTCVOL19IS2PG43.*

[19] *K. Sneha and G. M. Malle, "Research on software testing techniques and software automation testing tools," 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), 2017, pp. 77-81, doi: 10.1109/ICECDS.2017.8389562.*

[20] *Srivastav, Manoj & Nath, Asoke, "Web Content Management System," International Journal of Innovative Research in Advanced Engineering (IJIRAE), 2016*