

Headless Content Management System

Anushka Patil, Mitali Bamb, Sharvari Patil, Mrs. Supriya Patil

Department of Computer Engineering
Pimpri Chinchwad Polytechnic, Pune, Maharashtra, India

Abstract: *Website development has become a prominent and growing field in today's world. An effective web presence is essential to achieving sustained business growth. However, 's limited budget small business cannot afford to hire professionals to develop his website for. In this situation, Content Management System is a cost- effective solution. Because you don't need to be a developer to create a website without any coding knowledge. While traditional CMS combines frontend and backend, headless CMS completely separates backend from frontend, making it more flexible and scalable. The purpose of this research work is to design and develop a headless content management system and to evaluate its performance.*

Keywords: Web Development, Content Management System, Headless Content Management System, Shopify, GraphQL API, React, NextJS, Tailwind CSS, Netlify or Gatsby.

I. INTRODUCTION

The Internet has grown exponentially over the last 2 decades, with a significant increase in the number of businesses and users on the Internet. Today, most people prefer using the Internet for everything from research to business. Therefore, it is becoming increasingly important for companies to enter the digital world.

A well-developed website can have an incredible impact by complementing a company's digital marketing efforts, and can effectively increase sales. Therefore, many companies are increasing their investment in web development so that they can improve their business by meeting the needs of digital consumers. However, the process of developing a website is a time-consuming task and requires professionals with knowledge of various web development techniques. This can be a costly problem for start-ups. Add to this the rapid development of the field, making it difficult for even developers to keep up with the latest technology. This is where a content management system (CMS) comes in.

CMS is basically software that allows you to create and store content in digital form. This content is used to create a website based on a specific template. The user does not need her prior web development or coding knowledge. The traditional CMS consists of two parts. a. Content management application (CMA) that enables a user to create the content and that will be displayed on the website and a content delivery application (CDA) that delivers the content to the website once the user creates or updates the same. Additionally, there are three types of architecture that are commonly used. The first is the traditional or monolithic architecture, where the backend and frontend are coupled together into a single application. The most popular example of this is WordPress. A decoupled architecture comes next, in which the backend can either be integrated with the frontend or the content can be pushed to the frontend via API calls. But the data from the backend is continuously pushed somewhere even if there are no API calls being made. Lastly comes the headless architecture, where the backend is completely separated from the frontend or the presentation layer. The developed content is made accessible via REST API or GraphQL and can be displayed on any device possible.

Due to the difference in architecture, the performance of these CMS is also different. This research aims to develop a Headless Content Management System using Javascript, Next JS, React, Tailwind CSS, Netlify or Gatsby, providing insight into the pros and cons of headless CMS. Along with headless CMS, authentication service was also developed to provide level security for CMS applications.

II. RELATED WORKS

A comprehensive literature search was performed, with various referencing numerous peer-reviewed journals and various referencing papers published in standard journals. Most of the research in this area has involved evaluating and comparing the performance of existing content management systems such as WordPress, Drupal, and Joomla.

In 2018, a research study was concluded developing a WordPress website to provide the same content and analyze the

loading time of each. The hand-coded website was found to load times longer than WordPress for image, text, and video content, while WordPress made more API calls.

In 2018, another study was conducted to analyze the impact of WordPress on website development. At, we found that WordPress powers 29% of all websites. It is backward compatible with and provides numerous plugins and themes for developing websites.

The authors of “Comparative analysis of opensource content management systems” studied the use of open- source CMS for web content creation and management, influencing all major ministries of the Federal Government of India. Their results showed that WordPress was the most popular worldwide. Drupal is most commonly used at in government agencies

A unique 2015 study “Intuitive Content Management System” uses data-driven and mobile-driven technologies to replicate and multiply in the cloud to create standardized, user-friendly websites. The author has developed a CMS that allows users to create and manage websites without pre-installing his/her software. It provided better performance than the traditional CMS.

In a study conducted in 2014 “CMS: Comparative case study” Joomla, Drupal and WordPress ranked Alfresco, Typo3, Dotnetnuke, Drupal, Joomla, WordPress and Plone among System Requirements, Performance, Security, it turned out to be the top 3 built-in CMS. In function, in support. However, no CMS offers the best features and the choice of which one to use depends on the user's her needs.

Finally, a 2013 study compared 13 available Java- based open-source CMSs based on 29 features such as admin control, automatic code generation, and multi-user support, we concluded that Alfresco is the best available open-source CMS.

III. METHODOLOGY

This section describes the development process for creating a headless CMS. The development of the entire application was divided into four tasks. The first is to develop an authentication service that provides user registration and login functionality and enables access control using authentication tokens. The second is the design and development of a headless CMS service that allows to create content and store content. A third task entails integrating authentication services with CMS services. Finally the whole application is docked.

3.1 Development of the Authentication Service

This includes developing functionality to allow users to register and log in to their CMS accounts. Each time a user logs into her, a unique her authentication token is generated for each registered user. This is stored as key-value pairs in Redis, a cached database. This token is used to maintain access to your CMS account and after the expiration date is set, the token becomes invalid and the user is logged out of their account. The authentication service is accessed through API calls.

As part of this service, we used shopify database to store user data and API endpoints for registration and login are created. Login is then integrated with Mongo to generate for each user his unique authentication tokens. For the login API, the user supplied credentials are sent as request body parameters of the API call. The login service then queries to see if the credentials match in the database. If the credentials match it will return a success message, but if the username is correct but the password is incorrect, he will get a proper message. Finally, if neither username nor password match, message is returned prompting the user to register. Similarly, for the Registration API, where a new user is created, the credentials and details entered by the user are sent as request body parameters in the API call, which sends a query to add the data to the MongoDB database.

3.2 Development of the Authentication Service

This is the core of our research and includes front-end and back-end development of content management systems. The service should allow users to create new orders, add products to them, and dynamically update them. Also, users should be able to create own cart for a particular selected product. In addition, user should be able to edit and delete products from cart page and its respective fields and entries. This is the core of our research and includes front-end and back-end development of content management systems. The service should allow users to create new orders, add fields

to them, and dynamically update them. In addition, user should be able to edit and delete products from cart and its respective fields and entries.

Collection page provides various products collection for the customer. The user can view different products and can add to cart the products or proceed for checkout which is handled by the shopify.

3.3 Integration of the Authentication Service

The CMS service has an integrated registration and login page with a developed authentication service. This is so that all API calls from the CMS that access the database service are authorized via each user's unique herauth token once the user is logged into their account. When the authentication token expires, the user is automatically logged out. This provides an extra layer of security.

3.4 Containerization of the Application

It is important to ensure that changes to the system on which the application is deployed do not cause deployment or run-time problems. For this reason, Container is used to containerize the authentication and CMS services. In our case we have used the netlify or Gatsby platform to deploy our website which provides better developer and customer experience .

IV. DESIGN OF THE PROPOSED SYSTEM

This section describes the structure of the overall system. The system can be divided into three main parts. These include CMS Service Frontend, CMS Service Backend, and Authentication Service. To get a better high-level overview of the overall system, the overall architecture of the proposed system is shown in Figure 1.

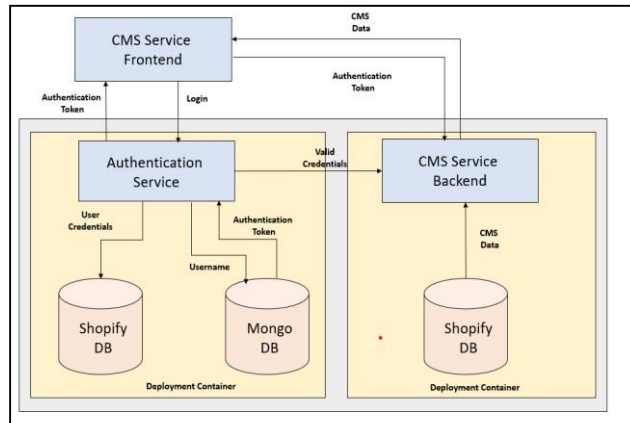


Fig-1: Architecture for the Design and Development of a Headless CMS

In order to understand the system from the user's perspective, an activity diagram was designed which can be seen in Fig-2. The user is first asked for a login. Credentials are accepted if the user has an account. Otherwise, the user goes to the registration page and creates a new account to log in to the CMS service. When users log in, they see the Home page where they view, select, order products. All products added to the cart are store in the shopify database. When a user logs in, a unique auto token is generated for each user.

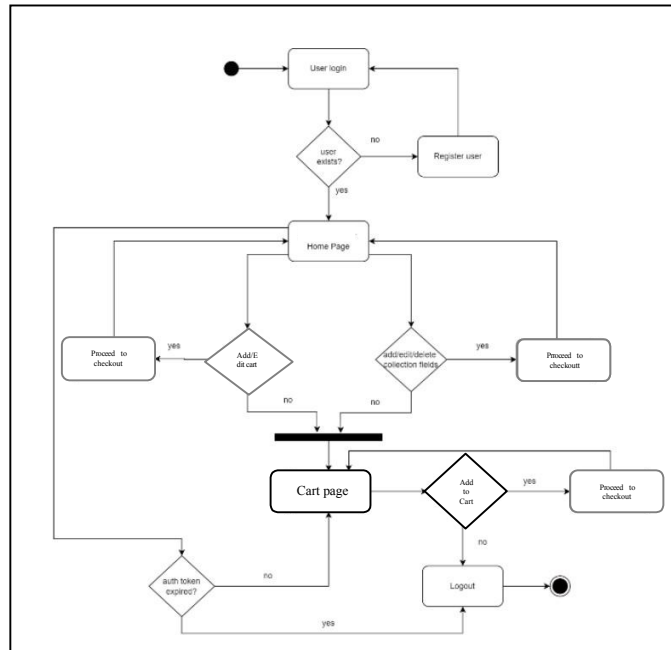


Fig-2: Activity Diagram for the Design and Development of a Headless CMS

Otherwise, the user can add to cart a product and place order. If a user doesn't want to add, edit the products they can move it to cart page that allows you to store the products and order later whenever needed.

V. CONCLUSION

This study presents the design and development of headless CMS. The system designed is effective at providing content creation and storage capabilities, but does not include workflow management for controlling different user-level permissions and managing different roles and permissions. Since workflow management is an important part of any business use case, its implementation can make the developed system commercially viable. Additionally, the introduction of drag and drop functionality allows marketers to model their own content based on specific templates. This may be useful in designing some custom components. Finally, the feature that allows integration of third-party data feeds such as S3 buckets reduces manual effort for users by eliminating the need to manually add collection entries. Overall, we find that the headless content management system is more flexible and performs better in terms of faster page load times. Additionally, decoupling the front-end and back-end is a better option in the long run, as it eliminates the need to redesign the entire system if changes are required. It is also more secure as there is no direct access to the content used to build the website. Therefore, using a headless CMS can have a significant impact on developing a business-friendly website, as it will greatly improve the ranking of your website due to its excellent performance in terms of load time.

VI. RESULT

This section provides information about the results obtained from this survey run. It also describes the results of tests performed on the system.

We have successfully developed a headless CMS that allows customers to buy products. This system has been extensively tested using the Jest test library. Run unit, integration, and system tests to ensure that each component performs the required functionality, verify data flow between different components, and determine if a particular component is affected by its presence. I have verified that there is no or missing of another component. More than 90% Code coverage was achieved on this system.

The system designed is easy to use and allows you to create the orders add product to cart and place order successfully. This data can be effectively used to build the website via API calls. A headless CMS completely decouples the backend from the frontend, so there is no CDA associated with the application and the data created can be retrieved via API calls. The developed application supports the same.

As part of the research, a headless content management system was developed. This differs from his traditional CMS as it completely separates the backend or content creation and storage part from the frontend or presentation layer. Such an architecture accesses data through API calls, so content can be delivered across multiple channels and modified as needed before being displayed on different devices. This also allows content reuse as the front end is designed independently. This is not possible with traditional architectures because the front-end and back-end are coupled, and there is a structured presentation layer that pulls data from the back-end content database. Unlike a traditional CMS, a headless CMS has no the back-end.

REFERENCES

- [1]. Kumar, A. Kumar, H. Hashmi and S. A. Khan, "WordPress: A Multi-Functional Content Management System," 2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART), 2021, pp. 158- 161, doi:10.1109/SMART52563.2021.9675311.
- [2]. G. Maragatham, S. N. A. Balaji, K. Sai Karthikeyan, Gokula krishnan and M. Siddharth, "A Study on Performance Analysis for Different WordPress and Hand Code Webpages," 2018 International Conference on Smart Systems and Inventive Technology (ICSSIT), 2018, pp. 191-204, doi: 10.1109/ICSSIT.2018.8748564.
- [3]. A. Mirdha, A. Jain and K. Shah, "Comparative analysis of open source content management systems," 2014 IEEE International Conference on Computational Intelligence and Computing Research, 2014, pp. 1-4, doi: 10.1109/ICCIC.2014.7238337.
- [4]. C. Hoong and M. A. Ameen, "Intuitive Content Management System," 2015 International Conference on Computer, Communications, and Control Technology (I4CT), 2015, pp. 541-543, doi: 10.1109/I4CT.2015.7219637.
- [5]. M. Nath and A. Arora, "Content management system: Comparative case study," 2010 IEEE International Conference on Software Engineering and Service Sciences, 2010, pp. 624-627, doi: 10.1109/ICSESS.2010.5552271.
- [6]. Cabot, Jordi. (2018). WordPress: A Content Management System to Democratize Publishing. IEEE Software. 35. 89-92. 10.1109/MS.2018.2141016.
- [7]. A. Kumar, A. Kumar, H. Hashmi and S. A. Khan, "WordPress: A Multi-Functional Content Management System," 2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART), 2021, pp. 158-161, doi: 10.1109/SMART52563.2021.9675311.
- [8]. N. A. Khan and H. Ahangar, "Use of Open Content Management Systems in Government Sector," 2018 5th
- [9]. Yermolenko, Andrei & Golchevskiy, Yuriy. (2021). Developing Web Content Management Systems – from the Past to the Future. SHS Web of Conferences. 110. 05007.10.1051/shsconf/202111005007.
- [10]. H. Liduo and C. Yan, "Design and implementation of Web Content Management System by J2EE-based three-tier architecture: Applying in maritime and shipping business," 2010 2nd IEEE International Conference on Information Management and Engineering, 2010, pp. 513-517, doi: 10.1109/ICIME.2010.5477670.
- [11]. Esperança and A. Pereira, "Content management system for e-Government portals," 2016 11th Iberian Conference on Information Systems and Technologies (CISTI), 2016, pp. 1-6, doi: 10.1109/CISTI.2016.7521369.
- [12]. Michelinakis, "Open Source Content Management Systems: An Argumentative Approach", Award MSc Electron. Bus. Manag, pp. 113, 2004.
- [13]. X. Cao and W. Yu, "Using Content Management System Joomla! to Build a Website for Research Institute Needs," 2010 International Conference on Management and Service Science, 2010, pp. 1-3, doi: 10.1109/ICMSS.2010.5577465.
- [14]. P. Kiatruangkrai, P. Phusayangkul, S. Viniyakul, N. Prompoon and P. Kanongchaiyos, "Design and Development of Real-Time Communication Content Management System for E-Commerce," 2010 Second International Symposium on Data, Privacy, and ECommerce, 2010, pp. 111-116, doi: 10.1109/ISDPE.2010.24.

- [15]. Chang Liu, Kirk P. Arnett, "Exploring the factors associated with Web site success in the context of electronic commerce", Information & Management, Volume 38, Issue 1, 2000, Pages 23-33, ISSN 0378-7206, doi: 10.1016/S0378-7206(00)00049-5.
- [16]. Martinez-Caro, Jose-Manuel, Antonio-Jose AledoHernandez, Antonio Guillen-Perez, Ramon SanchezIborra, and Maria-Dolores Cano. 2018. "A Comparative Study of Web Content Management Systems" Information 9, no. 2: 27. <https://doi.org/10.3390/info902002>