

Quick-DB a Headless Content Management System

Anwarul Siddique¹, Mohammed Faizan², Sayyad Arasalan³, Hritik Nagpure⁴

Reyan Ahmed Ansari⁵, Mohd Inzemamul Haque⁶, Anshul Wagh⁷

¹Professor, Computer Science & Engineering, Anjuman College of Engineering & Technology, Nagpur, Maharashtra, India

²Final Year UG Student, Computer Science & Engineering, Anjuman College of Engineering & Technology, Nagpur, Maharashtra, India

³Final Year UG Student, Computer Science & Engineering, Anjuman College of Engineering & Technology, Nagpur, Maharashtra, India

⁴Final Year UG Student, Computer Science & Engineering, Anjuman College of Engineering & Technology, Nagpur, Maharashtra, India

⁵Final Year UG Student, Computer Science & Engineering, Anjuman College of Engineering & Technology, Nagpur, Maharashtra, India

⁶Final Year UG Student, Computer Science & Engineering, Anjuman College of Engineering & Technology, Nagpur, Maharashtra, India

⁷Final Year UG Student, Computer Science & Engineering, Anjuman College of Engineering & Technology, Nagpur, Maharashtra, India

-----***-----

Abstract - Website development has evolved to become a prominent and ever-growing field in today's world. Having an effective web presence is critical for achieving sustainable business growth. However, small scale businesses with limited budgets cannot afford to hire professionals to develop their website. In such a situation, a Content Management System is a cost-effective solution as it allows non-developers to build websites without the requirement of any coding knowledge. While in a traditional CMS, the frontend and backend are coupled together, in a headless CMS, the backend is completely separated from the frontend thus, making it more flexible and scalable. The purpose of this project is to design and develop a headless content management system and evaluate its performance.

Key Words: CMS, Headless CMS, WordPress, REST API, MySQL

1.INTRODUCTION

Headless CMS (content management system) is a relatively new approach to managing and delivering digital content. Traditional CMS platforms, such as WordPress or Drupal, have a monolithic architecture that tightly couples the content management and presentation layers. This architecture limits flexibility in delivering content on multiple channels, as changes to the presentation layer require modifications to the core code. In contrast, headless CMS platforms provide a more flexible architecture by separating the content management system from the presentation layer. This means that with a headless CMS, content creators can manage their digital content in an organized way without concern for how it will be presented on various channels.

Developers can then use APIs to access and deliver this content in various forms on different channels, such as websites, mobile apps or IoT devices. However, with the growing popularity of headless CMS platforms, it is important to conduct research into their effectiveness and usability. Such research can examine the advantages and limitations of headless CMS, as well as how it compares to traditional CMS in terms of ease of use, efficiency, and scalability.

Furthermore, research can help identify which industries and use cases are best suited for headless CMS adoption.

In recent years, there has been a surge of interest in headless CMS as an alternative to traditional monolithic CMS platforms. As such, conducting research on headless CMS has become increasingly important to help organizations make informed decisions about adopting this approach. The effectiveness and usability of headless CMS platforms have become an area of growing interest due to their flexible architecture that separates content management from the presentation layer. This approach enables content creators to efficiently manage their digital content without worrying about presentation, while developers can easily access and deliver the content across various channels using APIs.

However, despite its advantages, research is needed to explore the limitations and scalability of headless CMS platforms as compared to traditional CMS. Through such research, the best use cases and industries for headless CMS adoption can also be identified. Overall, research is crucial to gain a better understanding of the potential benefits and limitations of headless CMS platforms in comparison with traditional CMS.

Consequently, conducting research can assist organizations in making informed decisions and developing effective strategies for adopting headless CMS platforms to enhance their digital content management capabilities.

Overall, this research paper aims to provide detailed insights of development of QUICK-DB system using Headless CMS. Which aims to deliver our content through an API directly to the frontends, using the Headless CMS method.

2.LITERATURE SURVEY

Sr No.	Paper Title	Findings
1.	J. Cabot, "WordPress: A Content Management System to Democratize Publishing," in IEEE Software, vol. 35, no. 3, pp. 89-92, May/June 2018	This paper is about open-source product known as WordPress which is the best CMS for frontend. It also describes how it becomes the multibillion-dollar company and how it still growing
2.	A. Kumar, A. Kumar, H. Hashmi and S. A. Khan, "WordPress: A Multi-Functional Content Management System," <i>2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART)</i> , Moradabad, India, 2021, pp. 158-161	This paper was about WordPress which is a well-known open-source CMS platform. It covered the main features of WordPress and drawbacks of using the same as CMS tool.
3.	Huynh, Le Dinh, Pham Quang Khang, and Phan Duy Hung. "A Collaborative Engine Design for Quick-CMS by React and Sails." In <i>Cooperative Design, Visualization, and Engineering: 19th International Conference, CDVE 2022, Virtual Event, September 25–28, 2022, Proceedings</i> , pp. 204-213. Cham: Springer International Publishing, 2022.	This paper discusses the benefits of using a Content Management System (CMS) for fast development of websites and digital content. The Quick CMS engine is presented as an example of a collaborative design that significantly reduces development time and costs, and has been successfully deployed across multiple projects.
4.	Alqadhi, Mohamed, Abdulrahman Alabduljabbar, Kyle Thomas, Saeed Salem, DaeHun Nyang, and David Mohaisen. "Do Content Management Systems Impact the Security of Free Content Websites?" In <i>Computational Data and Social Networks: 11th International Conference, CSoNet 2022, Virtual Event, December 5–7, 2022, Proceedings</i> , pp. 141-154. Cham: Springer Nature Switzerland, 2023.	This study analyzes over 1,500 free content websites to identify potential causes of vulnerabilities and maliciousness. It examines the use of content management systems (CMS) and their contribution to these

		<p>issues, finding that even a small number of unpatched vulnerabilities in popular CMS's can lead to significant maliciousness. The study also highlights varying trends in CMS usage across different types and categories of content.</p>
<p>5.</p>	<p>Giner-Miguel, Joan, Abel Gómez, and Jordi Cabot. "Enabling Content Management Systems as an Information Source in Model-Driven Projects." In <i>Research Challenges in Information Science: 16th International Conference, RCIS 2022, Barcelona, Spain, May 17–20, 2022, Proceedings</i>, pp. 513-528. Cham: Springer International Publishing, 2022.</p>	<p>This paper proposes a model-based framework for integrating headless Content Management Systems (CMSs) into software development processes. The framework facilitates discovery and representation of information schema behind the CMS, generating interactions for platform-agnostic access by client applications. The framework is open-source and available online.</p>

3. RESEARCH METHODOLOGY

Quick-DB is a headless CMS (Content Management System) designed to simplify the process of managing and delivering content across multiple platforms like Website, App, etc. Significant thing about Quick-DB is that, it is a platform that stores all of your content, including text, images, videos, and other media files, in a structured format. This makes it easy to manage and organize your content, and to deliver it to different platforms and devices.

One of the key benefits of using Quick-DB is that it separates content from presentation which means that the Front-end and Back-end is loosely coupled. This means that you can manage your content independently of how it will be displayed on your website or app. This is particularly useful for organizations that need to deliver content across multiple channels, such as websites, mobile apps, social media, and email.

In summary, Quick-DB is a powerful and flexible headless CMS that can help you streamline your content management and delivery process, and ensure that your content is delivered consistently across all channels and devices.

Module 1: Authentication

Authentication is a fundamental aspect of application development that plays a critical role in securing user data and authorization. It is the process of verifying the identity of a user or a system, usually by requiring the user to provide some form of credentials, such as a username and password, before granting access to a system or application.

Quick-DB's registration process is designed to collect personal information from users to create a unique account that can be used for future logins. This information includes the user's first and last name, email address, and a password that is created by the user.

The registration form may also include additional fields, such as a phone number or address, depending on the requirements of the application.

Once the user has completed the registration process, their personal information is stored in a secure database. The password is typically encrypted, meaning that it is transformed into a string of characters that cannot be easily read by anyone who gains access to the database. This helps to protect the user's password and prevent unauthorized access to their account. When the user logs in to their account, they are required to provide their registered email address and password. The login process involves retrieving the user's registered data from the database and comparing it to the entered information. If the information matches, the user is granted access to their account. However, if the entered information does not match the registered data, the user is denied access, and they may be prompted to try again or reset their password.

In summary, authentication is an essential aspect of application development that helps to ensure the security and privacy of user data. Quick-DB's registration and login processes are designed to collect user information securely and prevent unauthorized access to user accounts. By following these best practices, Quick-DB can provide a secure and reliable platform for users to store and access their data.

Module 2: Database Creation

You have the choice with Quick-DB to select the best database for your project. Here user will be able to select the Database type and be able to create the database as per their requirements. It may be configured, set up, and customized according to your preferences. It increases their level of free choice and flexibility in the design process. They can build a smooth user experience and make sure the application or website is operating correctly from beginning to end by understanding both sides of development.

They might also structure construct databases, collections, handle authentication, set up permissions, and more. Developers may use their true creativity when they have complete both the front end and the back end are under your control.

Module 3: API Generation

API Generation- The user will get the URL and the API will be created in this module. The content-types can be accessed via API endpoints using the REST API. Quick-DB offers a collection of dynamic APIs that enable programmers to design unique API endpoints. When a content-type is generated, Quick-DB automatically builds API endpoints. When requesting information from an API endpoint. Like with all projects, creating a new API begins with planning, which helps identify the best tools to use for the task. Beginning with a central platform from which one may choose their preferred frameworks and development tools.

An open-source headless content management system like Quick-DB provides developers with an easy Graphical User Interface to generate and manage content using MySQL and supports both RESTful API. They might also structure construct databases, collections, handle authentication, set up permissions, and more. Developers may use their true creativity when they have complete both the front end and the back end are under your control.

Module 4: Database Management

In our Quick DB project user can create their database on the bases of their needs. And our project can help to fulfil their need. User can easily insert, delete, modify, etc. some operation performs in this database section. After user make a database, the user can modify or give the new look of their database depend on their requirements his/her can easily changes perform on database when any updating required on their project.

User can insert an integer format data, float format data, character format data, string format data depend on their requirement. When user does not use previous database, he/she can make a new database. Here in the section of database management the history of database is available user can check every updating and changes in the database management section.

In our Quick-DB project we create one section of database management. Where the information of the database is store. When user make a database the history of database is available in the database management section. User can modify or edit anything in database the changes of database are available in the database management. Every time user make change in his/her database the history of database update and show every changes history in database management.

User can modify the any information in project the information is history of project is available in database section. When user can make any change in her/his database her/his can easily modify their database information at any time when his/her can changes

some or more data in database. Every time the updating of database is shown in database management section and user can check which updating done by the need of their requirement.

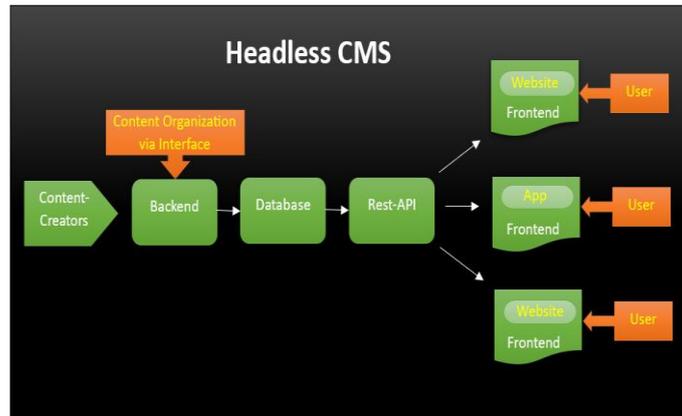


Fig-1: State Diagram

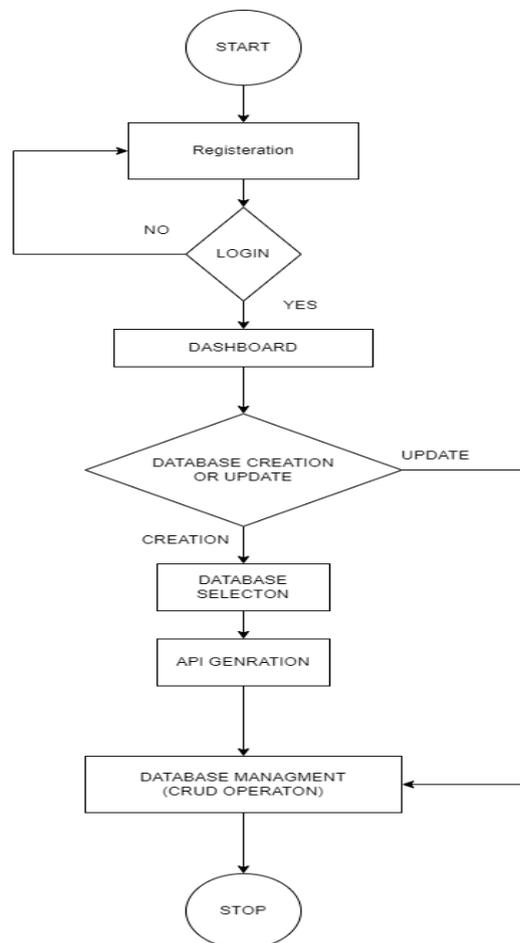


Fig-2: Flow Diagram

Fig-1 and Fig-2 shows the state and the flow of Quick-Db. Overall, this research methodology aims to develop a working prototype of a Quick-DB using Headless CMS technology. It involves designing of system architecture and Website, and testing and validating the system to ensure its accuracy and reliability

4.RESULTS AND DISCUSSION

Overall, in research it can be seen that a headless content management system is more flexible and performs better in terms of faster page load time. Additionally, it is a better option in the long run as the decoupling of the frontend and the backend ensures that the entire system doesn't have to be redesigned when any changes need to be made. It is also more secure as there is no direct access to the content that is used to create the website. Hence, the use of a headless CMS can have a great impact in developing a good website and the research demonstrates the feasibility of implementing this technology in real-world scenarios and provides a foundation for future research on the topic.

5.CONCLUSION & FUTURE SCOPE

In conclusion, as headless CMS gain popularity as an alternative to traditional CMS, conducting research into their efficacy and feasibility has become increasingly important. Research can help organizations understand the advantages and limitations of headless CMS platforms, as well as determine the best use cases and industries for adoption.

As for future scope, there are several areas where further research can be conducted to improve and enhance the system. These includes adding more databases type (like Mongo DB, No SQL, etc.) and more API implementation (GraphQL, etc.).

REFERENCES

1. 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
2. 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
3. J. Cabot, "WordPress: A Content Management System to Democratize Publishing," in *IEEE Software*, vol. 35, no. 3, pp. 89-92, May/June 2018
4. A. K. Phulre, S. Pagare and A. Chakrawati, "Automated Framework for Web Content Security Through Content Management System," 2022 10th International Conference on Emerging Trends in Engineering and Technology - Signal and Information Processing (ICETET-SIP-22), 2022, pp. 1-4.
5. A. Kumar, A. Kumar, H. Hashmi and S. A. Khan, "WordPress: A Multi-Functional Content Management System," 2021 10th International Conference on System Modelling & Advancement in Research Trends (SMART), 2021, pp. 158-161
6. Alqadhi, M., Alabduljabbar, A., Thomas, K., Salem, S., Nyang, D., Mohaisen, D. (2023). Do Content Management Systems Impact the Security of Free Content Websites? In: Dinh, T.N., Li, M. (eds) *Computational Data and Social Networks. CSoNet 2022. Lecture Notes in Computer Science*, vol 13831. Springer, Cham.
7. Giner-Miguel, J., Gómez, A., Cabot, J. (2022). Enabling Content Management Systems as an Information Source in Model-Driven Projects. In: Guizzardi, R., Ralyté, J., Franch, X. (eds) *Research Challenges in Information Science. RCIS 2022. Lecture Notes in Business Information Processing*, vol 446. Springer, Cham.
8. Huynh, L.D., Khang, P.Q., Hung, P.D. (2022). A Collaborative Engine Design for Quick-CMS by React and Sails. In: Luo, Y. (eds) *Cooperative Design, Visualization, and Engineering. CDVE 2022. Lecture Notes in Computer Science*, vol 13492. Springer, Cham.
9. J. Cabot, "WordPress: A Content Management System to Democratize Publishing," in *IEEE Software*, vol. 35, no. 3, pp. 89-92, May/June 2018