



IJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** IV **Month of publication:** April 2025

DOI: <https://doi.org/10.22214/ijraset.2025.69777>

www.ijraset.com

Call:  08813907089

E-mail ID: ijraset@gmail.com

Quick Web Craft – A Low Code SaaS Platform

TusharThite¹, Om Gaikwad², Yogesh Mane³, Samruddhi Patil⁴, Aniket Talwar⁵, S.V.Shinge⁶

^{1, 2, 3, 4, 5}UG Student, ⁶Assistant Professor, Computer Science and Engineering, D.Y.Patil College of Engineering and Technology, Kolhapur, Maharashtra, India

Abstract: *Quickwebcraft is a simplified Software as a Service (SaaS) platform created to make dynamic web development accessible to individuals without explicit technical backgrounds. It allows users to build professional-looking, mobile-friendly websites using drag- and- drop components, customizable templates and real-time previews. This platform streamlines both frontend and deployment processes, improving productivity and significantly reducing development time for users including entrepreneurs and small businesses.*

Keywords: *Low-codetool, SaaS Development, visual web editor, drag-and-drop interface, Quickwebcraft*

I. INTRODUCTION

In today's digital era, an online presence has become vital for businesses, freelancers, and creators. However, developing and deploying a website often requires expertise in front-end and back-end technologies, databases and deployment. For individuals lacking a technical background, these obstacles can be especially discouraging.

Quickwebcraft is emerged as a solution to these problems. It provides a low-code development environment that helps users build interactive and aesthetic websites with ease. This project intends to make web development accessible to a wider audience by encouraging participation in the website creation process.

The system integrates modern web technologies like React.js, Node.js, MongoDB, and Express.js into a unified platform, presented through a user-friendly interface. With drag-and-drop functionality, real-time previews, and deployment features, the tool is positioned as a comprehensive web solution.

II. LITERATURE REVIEW

The digital landscape is evolving faster, growing the demand for efficient, accessible, and user-friendly website development tools. Traditional coding methods often require specialized expertise, making it hard for non-technical users to build and maintain websites. Furthermore, manual development approaches can be time-consuming and costly, hindering quick digital transformation. Low-code SaaS solutions, particularly drag-and-drop website builders, have emerged as powerful alternatives, democratizing web development through visual interfaces, pre-built templates, and reusable components. These tools are especially valuable for small businesses and individuals with limited technical resources.

Recent studies have emphasized the importance of quality assurance (QA) in low-code and no-code projects, highlighting the need for robust QA processes to ensure security and scalability [1]. A systematic review conducted has shown that these platforms offer faster delivery and reduced costs, while also noting concerns like vendor lock-in and limited customization [2]. One of the research demonstrates how low-code solutions improve API development and backend integration, streamlining deployment for small and medium-sized enterprises (SMEs) [3].

Additionally, researchers have explored the combined power of low-code development and model-driven engineering (MDE), showing how abstraction and automation improve scalability [4]. Industry insights point to drag-and-drop tools that lower the learning curve for non-programmers, making website creation accessible for even non-programmers [5]. For larger organizations, successful adoption often depends on key features such as user-friendliness, scalability, and integration with existing systems [6]. Beyond website development, low-code tools are also helpful in automating business processes and improving operational efficiency through pre-built components for workflow management and data integration [7]. Low-code platforms are bridging the gap between businesses, technical and non-technical teams, incorporating collaboration and driving innovations [8].

In conclusion, low-code platforms are transforming software development by enabling rapid prototyping, reducing dependency on developers, and promoting inclusive innovation. While the benefits are substantial, addressing limitations such as customization constraints and quality control is vital for long-term success and scalability.

III. THEORETICAL BACKGROUND

The theoretical foundation of the Quickwebcraft project lies at the intersection of low-code software development, web technologies, and user-centric design. The project is guided by principles of rapid application development (RAD), which emphasizes minimal manual coding, visual interfaces, and reusability of components. By leveraging modern frontend frameworks like React.js, backend technologies such as Node.js and Express.js, and cloud-based services including Vercel and MongoDB, the platform delivers a seamless and scalable development experience. The system architecture supports efficient data handling, secure authentication, and streamlined deployment, enabling users—especially those with limited technical expertise—to build, publish, and manage websites effortlessly.

A. Need for Project

- 1) **Rapid digital transformation:** These tools can significantly accelerate the process of creating and deploying websites, enabling businesses to quickly adapt to changing market conditions and seize new opportunities.
- 2) **Cost-effectiveness:** Low-code platforms often require less development time and resources compared to traditional coding methods, making them a more affordable option for businesses of all sizes.
- 3) **Time-to-market:** By streamlining the development process, these tools can help businesses bring their products or services to market faster, gaining a competitive advantage.
- 4) **Customization and scalability:** Low-code platforms often offer a high degree of customization, allowing businesses to tailor their websites to their specific needs. Additionally, they can scale to accommodate growth and increase user demand.
- 5) **Ease of use:** These tools are designed to be user-friendly, making them accessible to individuals with limited technical skills. This can empower a wider range of people to create and manage their own websites.

B. Technologies Used

1) React.js

Description: React.js is a popular JavaScript library used for building fast and interactive user interfaces using reusable components. It simplifies the development of complex UI systems with its component-based architecture.

Role in the Project: React.js is used to develop the frontend of Quickwebcraft, enabling the creation of a dynamic drag-and-drop interface and seamless user experience for building websites.

2) Node.js and Express.js

Description: Node.js is a JavaScript runtime environment that executes code outside a web browser, while Express.js is a lightweight web framework for building backend services.

Role in the Project: Node.js and Express.js formed the backend of the platform, managing APIs, user authentication, and data handling between the frontend and database.

3) MongoDB

Description: MongoDB is a NoSQL database that stores data in flexible, JSON-like documents, offering scalability and high performance for modern web applications.

Role in the Project: MongoDB was used to store user profiles, website data, and UI component configurations. MongoDB Atlas was used as a cloud-based solution for reliability and accessibility.

4) Vercel

Description: Vercel is a cloud platform for static site hosting and deployment, optimized for frontend frameworks like React.js. It allows fast, global deployments with minimal setup.

Role in the Project: The frontend of Quickwebcraft is deployed using Vercel, ensuring fast load times, continuous integration, and a smooth user experience.

5) Git and GitHub

Description: Git is a version control system, and GitHub is a platform for hosting and collaborating on Git repositories. **Role in the Project:** Git is used for tracking code changes and version control. GitHub hosted the repository, enabling collaboration, issue tracking, and documentation management among team members.

C. Problem Definition

The Quickwebcraft project responds to the increasing demand for quick and easy-to-use platforms for developing websites that are suitable for users with little technical knowledge. The creation of this project is driven by several significant issues as the need for a digital presence grows across multiple domains.

1) Lack of Technical Skills Among

Issue: Many individuals and small-scale businesses lack coding knowledge, which becomes a barrier to creating and managing professional websites.

Impact: Lack of technical expertise, businesses are often required to hire expensive developers or settle for an average web presence. Quickwebcraft offers a low-code solution that bridges this gap, enabling users to build functional websites through a visual interface.

2) Time-Consuming Development Processes

Issue: Traditional website development involves multiple stages—design, coding, testing, and deployment—which can be time-consuming, especially for small-scale projects.

Impact: The longer development time hinders the rapid deployment of web solutions. Quickwebcraft saves time and effort by promoting rapid and effective website construction with drag-and-drop components and reusable templates.

3) Inaccessibility of Scalable Hosting Solutions

Issue: Beginners and non-developers often struggle with setting up reliable hosting environments or continuous deployment pipelines.

Impact: This can lead to inconsistent uptime and poor user experience. By integrating with cloud platforms like Vercel, Quickwebcraft simplifies deployment and ensures seamless scalability with minimal user intervention.

4) Fragmentation of Frontend and Backend Technologies

Issue: Modern web development often requires managing multiple technologies for frontend, backend, database, and deployment.

Impact: This complexity is overwhelming for solo users or non-technical teams. Quickwebcraft unifies these technologies under a single platform, abstracting the complexity while still delivering professional-grade websites.

5) Lack of Reusability and Modularity in Existing Platforms

Issue: Many website builders restrict customization or lack reusable components, limiting design flexibility. *Impact:* Users are unable to scale or modify their websites according to their evolving needs. Quickwebcraft focuses on component-based architecture using React.js, which allows modular design and easy scalability.

6) Need for Collaborative Development and Community Support

Issue: Most low-code platforms do not emphasize community collaboration or code transparency.

Impact: This limits peer learning and adaptation for custom use cases. Quickwebcraft encourages collaboration by hosting its code on GitHub and allowing developers to contribute, fork, and improve the system.

IV. MOTIVATION

As computer science students, we've often seen how many people struggle to create a digital presence simply because they have limited access or lack coding skills. This inspired us to build something that makes web development easier and more accessible for everyone.

With Quickwebcraft, our goal is to simplify the process of building websites by using low-code tools and intuitive design. It's a way for us to apply what we've learned in development and design to solve a real-world problem—helping individuals and small businesses build quality websites without needing technical expertise. We're excited to contribute to making web creation more inclusive, efficient, and scalable.

V. SYSTEM ANALYSIS AND DESIGN

A. Development Approach

Agile Scrum: The Quickwebcraft project is developed using the Agile Scrum methodology. This iterative approach helped us break the work into manageable sprints, focus on regular feedback, and make continuous improvements throughout the development process. It also encouraged better teamwork and flexibility as we built and refined each module.

B. Project Inception

Objective: To define the purpose, end-user goals, and architecture for a low-code web development platform.

Activities:

- Identifying the need for a simplified website builder targeting non-technical users.
- Gathering user requirements such as drag-and-drop interface, theme customization, and instant preview.
- Outlining the core modules: User interface, Component library, Template library, and Deployment integration.

C. Module 1 – User Management

Objective: To handle user authentication, authorization, and profile management.

Activities:

- Enabling user registration and login with secure credential handling.
- Allowing password resets and profile updates.

D. Module 2 – Website Creation

Objective: Allow users to start building websites from templates or blank canvases.

Activities:

- Provide an option to begin from a blank canvas for full customization.
- Offer a range of pre-designed templates for quick starts.
- Let users name their websites and configure initial settings.

E. Module 3 – Website Editing

Objective: To provide a visual editor for customizing website content and layout.

Activities:

- Implementing drag-and-drop functionality using a React-based interface.
- Offering a library of reusable components like headers, footers, and buttons.
- Allowing inline content editing and theme customization (fonts, colors, layout).

F. Module 4 – Component Library

Objective: To manage reusable components for efficient design.

Activities:

- Allowing users to add, edit, and delete custom-built components.
- Providing tools to organize components by category or use case.

G. Module 5 – Template Management

Objective: To store and manage website templates.

Activities:

- Saving user-created and default templates in the system.
- Allowing full customization of templates before deployment.

H. Module 6 – Deployment

Objective: To make Effortless Deployment of websites.

Activities:

- Integrating with platforms like Vercel or custom hosting.

- Automating the build and deployment process with one-click options.
- Generating and managing public URLs for live websites.

I. Module 7 – Integration Module

Objective: To connect external services and enable API-based functionality.

Activities:

- Facilitating integration with email services and social media.
- Managing API calls and responses for third-party services.
- Ensuring real-time data synchronization between services and the builder.

The diagram illustrates the modular structure of the website builder platform. It begins with the User Management Module for handling user access. Users then create or edit websites through the Website Creation Module, which connects to the Website Editing Module for applying templates and components. Templates are managed by the Template Management Module, while components are accessed from the Component Library Module. Third-party services are integrated via the Integration Module, and the final website is published using the Deployment Module, resulting in a Live Website.

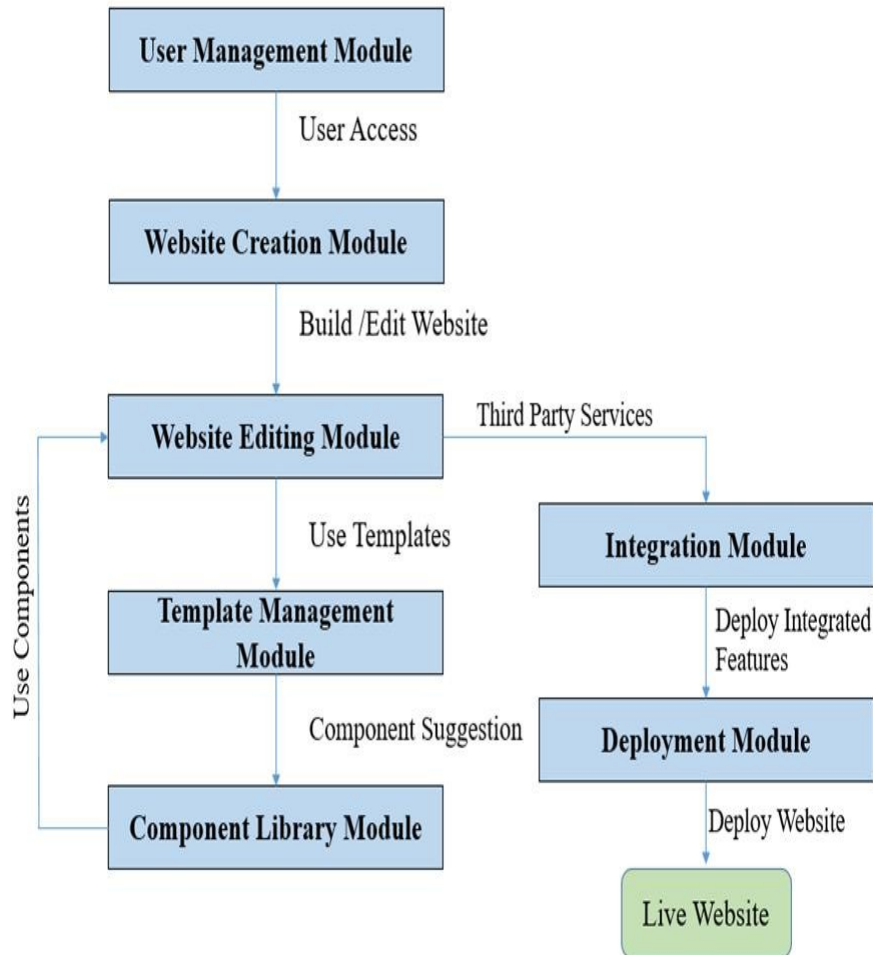


Fig. 1 Modular Structure of Quickwebcraft

This system architecture illustrates the modular structure of the Quickwebcraft low-code SaaS platform, showcasing the integration of frontend, backend services, databases, third-party APIs, and deployment layers. It emphasizes scalability, maintainability, and flexibility through distinct modules such as drag-and-drop UI, real-time services, API gateway, and auto-scaling, supporting seamless development and deployment workflows.

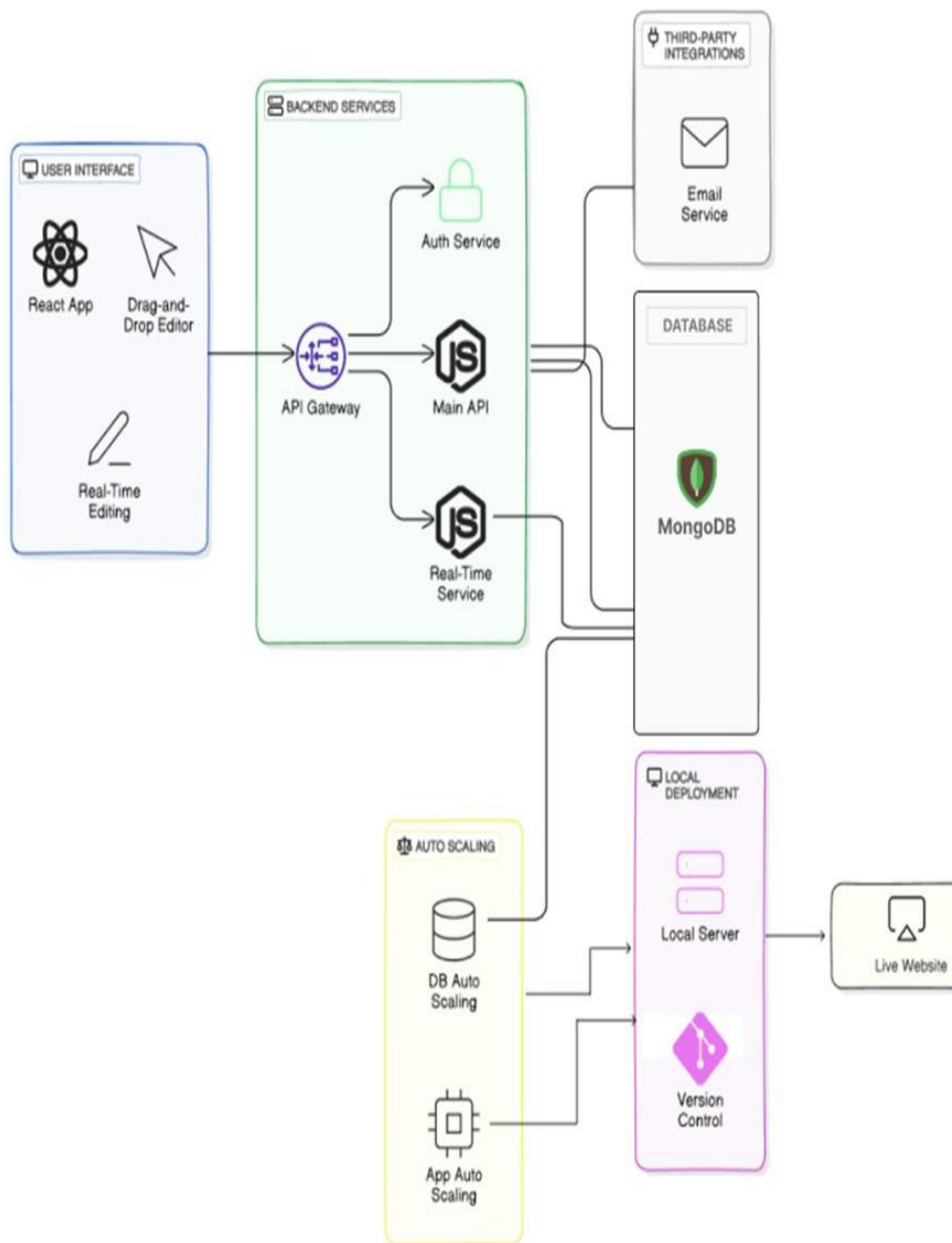


Fig.2 System Architecture of Quickwebcraft

This Data Flow Diagram (DFD) represents the Level 1 flow of the website builder system, illustrating interactions between the user, main modules, and the database. The user initiates actions like registering or logging in, which are handled by the User Management Module and connected to the Database for storing or retrieving user information. The Website Creation Module enables users to access features like template selection and blank canvas creation, while the Website Editing Module facilitates website customization using resources from the Component Library Module and Template Management Module. The Deployment Module manages website hosting and domain setup, integrating with third-party services through the Integration Module. The Database ensures seamless data storage and retrieval across the system.

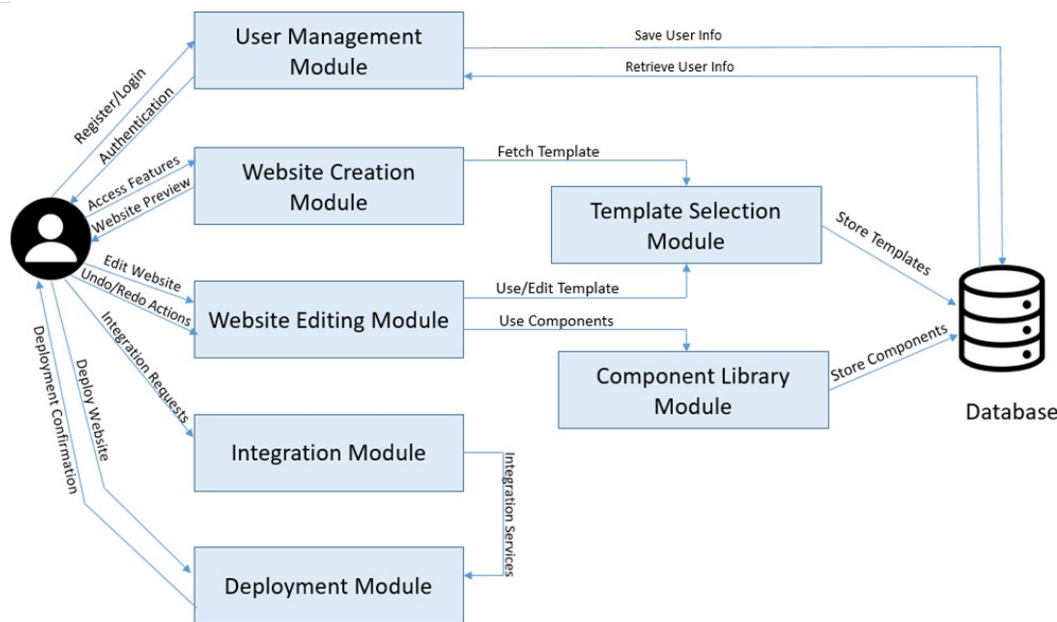


Fig.3DFDLevelDiagram

VI. CONCLUSION AND FUTURE WORK

The development of the low-code SaaS platform highlights a modular and user-friendly approach to website creation, editing, and deployment. By leveraging modern technologies and a well-structured architecture, the system aims to simplify web development for users with minimal coding knowledge. Maintaining a clear workflow and regular iteration has been essential in building a functional and scalable platform.

In the future, the platform can be enhanced with AI-driven design suggestions, real-time collaboration features, and support for more third-party integrations. Continued improvements will focus on boosting performance, expanding template and component libraries, and enriching the overall user experience.

VII. ACKNOWLEDGMENT

We would like to express our sincere gratitude to our project guide and faculty members for their consistent support and guidance throughout the development of this project. We also extend our thanks to our peers for their valuable feedback and collaboration, which played a significant role in shaping the final outcome of our low-code SaaS platform.

REFERENCES

- [1] De Silva, D. I., Shangavie, R., & R. A. A. L Ranathunga, "Role of Quality Assurance in Low-Code/No-Code Projects.", International Conference on Information (2024)
- [2] pp.789-794. doi:10.1109/ICOIN59985.2024.10572203
- [3] H. El Kamouchi, M. Kissi and O. El Beggar, "Low-code/No-code Development : A systematic literature review," 14th International Conference on Intelligent Systems: Theories and Applications (SITA), Casablanca, Morocco, (2023)
- [4] pp.1-8. doi:10.1109/SITA60746.2023.10373712.
- [5] P. Dhoke and P. Lokulwar, "Evaluating the Impact of No-Code/Low-Code Backend Services on API Development and Implementation: A Case Study Approach," 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), Delhi, India, (2023), pp. 1-5, doi:10.1109/ICCCNT56998.2023.10306945.
- [6] Di Ruscio, D., Kolovos, D., de Lara, J. et al. Low-code development and model-driven engineering: Two sides of the same coin?. *Softw Syst Model* 21,437–446 (2022). <https://doi.org/10.1007/s10270-021-00970-2>
- [7] UXPin Studio. "Low-Code and No-Code Tools – What Are They?" UXPin Blog, 22 Feb. 2022, <https://www.uxpin.com/studio/blog/low-code-no-code-tools/>.
- [8] A. Oberer, and B. Döbel, "Low-Code Platforms: An Analysis of Key Success Factors," *Bus. Inf. Syst. Eng.*, vol. 63, no. 3, pp. 277-291, (2021), doi:10.1007/s12599-021-00726-8.
- [9] S. R. Hilt, and M. Wagner, "Leveraging Low-Code Development for Business Process Automation," *Adv. Inf. Syst. Eng.*, vol. 423, pp. 533-546, (2021), doi:10.1007/978-3-030-89912-7_41.
- [10] R. Benac, and T.K. Mohd, "Recent Trends in Software Development: Low-Code Solutions," in: Arai, K. (eds) *Proceedings of the Future Technologies Conference (FTC) 2021, Volume 3. FTC 2021. Lecture Notes in Networks and Systems*, vol. 360. Springer, Cham. (2021), https://doi.org/10.1007/978-3-030-89912-7_41



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)