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

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IV Apr 2025- Available at www.ijraset.com

Quick Web Craft – A Low Code SaaS Platform

TusharThite[1], Om Gaikwad[2], Yogesh Mane[3], Samruddhi Patil[4], Aniket Talwar[5], S.V.Shinge[6]
[1, 2, 3, 45]*UG Student,* [6]*AssistantProfessor, 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 toindividualswithoutexplicittechnicalbackgrounds.Itallowsuserstobuildprofessional-looking,mobile-friendly website susingdrag- and- drop components, customizable templates and real-time previews. This platform streamlines both frontend and deployment processes,improvingproductivityandsignificantly reducingdevelopmenttime forusers including entrepreneurs and smallbusinesses.***

Keywords: Low-codetool,SaaSDevelopment,visualwebeditor,drag-and-dropinterface,Quickwebcraft

I. INTRODUCTION

Intoday'sdigital era,anonlinepresence hasbecomevital for businesses, freelancers,and creators.However,developingand 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.

Quickwebcraftis emerged as a solution to these problems. It provides a low-code development environment that helps users build interactiveand aesthetic websites withease. Thisprojectintendsto make web developmentaccessibleto 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.Traditionalcodingmethodsoftenrequirespecializedexpertise,makingithardfornon-technicaluserstobuildandmaintain 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 ofqualityassurance (QA) in low-code and no-code projects, highlightingthe needforrobustQAprocessestoensuresecurityandscalability[1].Asystematicreviewconductedhasshownthattheseplatforms offer faster deliveryand reduced costs, while also noting concerns like vendor lock-in and limited customization [2]. One of the researchdemonstrates howlow-codesolutions improve API development and backend integration, streamliningdeployment for small and medium-sized enterprises (SMEs) [3].

Additionally,researchershaveexploredthecombinedpoweroflow-codedevelopmentandmodel-drivenengineering(MDE), showing how abstraction and automation improve scalability [4]. Industry insights point to drag-and-drop tools that lower the learningcurvefornon-programmers,makingwebsitecreationaccessibleforevennon-programmers[5].Forlargerorganizations, successfuladoptionoftendependsonkeyfeaturessuchasuser-friendliness,scalability,andintegrationwithexistingsystems[6]. Beyond website development, low-code tools are also helpful in automating business processes and improving operationalefficiencythroughpre-builtcomponentsforworkflowmanagementanddataintegration[7].Low-codeplatformsarebridgingthe gapbetweenbusinesses,technicalandnon-technicalteams,incorporatingcollaborationanddrivinginnovations[8]

Inconclusion,lowcodeplatformsaretransformingsoftwaredevelopmentbyenablingrapidprototyping,reducingdependencyondevelopers, 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.

©IJRASET: All Rights are Reserved

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 emphasizeminimalmanualcoding,visualinterfaces,andreusabilityofcomponents.Byleveragingmodernfrontendframeworks likeReact.js,backendtechnologiessuchasNode.jsandExpress.js,andcloud-basedservicesincludingVercelandMongoDBthe 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) Rapiddigitaltransformation:Thesetoolscansignificantlyacceleratetheprocessofcreatinganddeployingwebsites, enabling businesses to quickly adapt to changing market conditions and seize new opportunities.
2) Cost-effectiveness:Low-codeplatformsoftenrequirelessdevelopmenttimeandresourcescomparedtotraditionalcoding methods, making them a more affordable option for businesses of all sizes.
3) Time-to-market:Bystreamliningthedevelopmentprocess,thesetoolscanhelpbusinessesbringtheirproductsorservices to market faster, gaining a competitive advantage.
4) Customizationandscalability: Low-codeplatformsoftenofferahighdegreeofcustomization,allowingbusinessestotailor their websites to their specific needs. Additionally, they can scale to accommodate growth and increase user demand.
5) Easeofuse:Thesetoolsaredesignedtobeuser-friendly,makingthemaccessibletoindividualswithlimitedtechnicalskills. This can empower a wider range of people to create and manage their own websites

B. Technologies Used

1) React.js

*Description:*React.jsisapopularJavaScriptlibraryused forbuildingfastandinteractiveuserinterfacesusingreusable components. It simplifies the development of complex UI systems with its component-based architecture.
RoleintheProject: React.jsisusedtodevelopthe frontend ofQuickwebcraft,enablingthecreationofadynamicdrag-and-drop interface and seamless user experience for building websites.

2) Node.jsandExpress.js

*Description:*Node.jsisaJavaScriptruntimeenvironment thatexecutescodeoutsideawebbrowser,whileExpress.jsisa lightweight web framework for building backend services.
RoleintheProject: Node.jsandExpress.jsformedthebackendoftheplatform,managingAPIs,userauthentication,and data handling between the frontend and database.

3) MongoDB

Description: MongoDBisaNoSQLdatabasethatstoresdatainflexible,JSON-likedocuments,offeringscalabilityand high performance for modern web applications.
RoleintheProject: MongoDB wasusedtostoreuserprofiles,websitedata,andUIcomponentconfigurations.MongoDBAtlas was used as a cloud-based solution for reliability and accessibility.

4) Vercel

Description:Vercelisacloud platformforstaticsitehostinganddeployment,optimizedforfrontend frameworkslikeReact.js. It allows fast, global deployments with minimal setup.
*RoleintheProject:*ThefrontendofQuickwebcraftisdeployedusingVercel,ensuringfastloadtimes,continuousintegration, and a smooth user experience.

5) GitandGitHub

*Description:*Gitisaversioncontrolsystem,andGitHubisaplatformforhostingandcollaboratingonGitrepositories. *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.

©IJRASET: All Rights are Reserved

C. ProblemDefinition

TheQuickwebcraftprojectrespondstotheincreasingdemandforquickandeasytouseplatformsfordevelopingwebsitesthataresuitableforuserswithlittletechnicalknowledge.Thecreationofthisprojectisdrivenby several significantissuesas the need for a digital presence grows across multiple domains.

1) LackofTechnicalSkillsAmong

*Issue:*Manyindividualsandsmallscalebusinesseslackcodingknowledge, whichbecomesabarriertocreatingand managing professional websites.

Impact: Lack of technical expertise, businesses are often required to hire expensive developers or settle for an average web presence.Quickwebcraftoffersalow-codesolutionthatbridgesthisgap,enablingusersto buildfunctionalwebsitesthrougha visual interface.

2) Time-ConsumingDevelopmentProcesses

*Issue:*Traditionalwebsitedevelopmentinvolvesmultiplestages—design,coding,testing,anddeployment—whichcanbetime- consuming, especially for small-scale projects.

*Impact:*Thelongerdevelopmenttimehinderstherapiddeploymentofwebsolutions. Quickwebcraftsavestimeandeffortby promoting rapid and effective website construction with drag-and-drop components and reusable templates.

3) InaccessibilityofScalableHostingSolutions

Issue:Beginnersandnon-developersoftenstrugglewithsettingupreliablehostingenvironmentsorcontinuousdeployment pipelines.

Impact:Thiscanleadtoinconsistentuptimeandpooruserexperience.ByintegratingwithcloudplatformslikeVercel, Quickwebcraft simplifies deployment and ensures seamless scalability with minimal user intervention.

4) FragmentationofFrontendandBackendTechnologies

Issue:Modernwebdevelopmentoftenrequiresmanagingmultipletechnologiesforfrontend,backend,database,anddeployment.

Impact:Thiscomplexityisoverwhelmingfor solo usersor non-technicalteams. Quickwebcraft unifiesthesetechnologiesunder a single platform, abstracting the complexity while still delivering professional-grade websites.

5) LackofReusabilityandModularityinExistingPlatforms

Issue:Manywebsitebuildersrestrictcustomizationorlackreusablecomponents,limitingdesignflexibility.*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) NeedforCollaborativeDevelopmentandCommunitySupport

*Issue:*Mostlow-codeplatformsdonotemphasizecommunitycollaborationorcodetransparency.

Impact:Thislimitspeerlearningandadaptationforcustomusecases.Quickwebcraftencouragescollaborationbyhostingitscode 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.

WithQuickwebcraft,ourgoalistosimplifytheprocessofbuildingwebsitesbyusinglow-codetoolsandintuitivedesign.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 qualitywebsites without needingtechnical expertise. We're excited to contribute to making web creation more inclusive, efficient, and scalable.

V. SYSTEM ANALYSIS AND DESIGN

A. DevelopmentApproach

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. ProjectInception

*Objective:*Todefinethepurpose,end-usergoals,andarchitectureforalow-codewebdevelopmentplatform.
Activities:

- Identifyingtheneedforasimplifiedwebsitebuildertargetingnon-technicalusers.
- Gatheringuserrequirementssuchasdrag-and-dropinterface,themecustomization,andinstantpreview.
- Outliningthecoremodules:Userinterface,Componentlibrary,Templatelibrary,andDeploymentintegration.

C. Module1 –UserManagement

*Objective:*Tohandleuserauthentication,authorization,andprofilemanagement.
Activities:

- Enablinguserregistrationandloginwithsecurecredentialhandling.
- Allowingpasswordresetsandprofileupdates

D. Module2–Website Creation

Objective:Allowuserstostartbuildingwebsitesfromtemplatesorblankcanvases.
Activities:

- Provideanoptiontobeginfromablankcanvasforfullcustomization.
- Offerarangeofpre-designedtemplatesforquickstarts.
- Letusersnametheirwebsitesandconfigureinitial settings.

E. Module3–WebsiteEditing

Objective:Toprovideavisualeditorforcustomizingwebsitecontentandlayout.
Activities:

- Implementingdrag-and-dropfunctionalityusingaReact-basedinterface.
- Offeringalibraryofreusablecomponentslikeheaders,footers,andbuttons.
- Allowinginlinecontenteditingandthemecustomization(fonts,colors,layout).

F. Module4 –ComponentLibrary

*Objective:*Tomanagereusablecomponentsforefficientdesign.
Activities:

- Allowinguserstoadd,edit,anddeletecustom-builtcomponents.
- Providingtoolstoorganizecomponentsbycategoryorusecase.

G. Module5 –Template Management

Objective:Tostoreandmanagewebsitetemplates.
Activities:

- Savinguser-createdanddefaulttemplatesinthesystem.
- Allowingfullcustomizationoftemplatesbeforedeployment.

H. Module6 –Deployment

*Objective:*TomakeEffortlessDeploymentofwebsites.
Activities:

- IntegratingwithplatformslikeVercelorcustomhosting.

©IJRASET: All Rights are Reserved

- Automatingthebuildanddeploymentprocesswithone-clickoptions.
- GeneratingandmanagingpublicURLsforlivewebsites.

I. Module7–IntegrationModule

Objective:ToConnectexternalservicesandenableAPI-basedfunctionality.

Activities:

- Facilitatingintegrationwithemailservicesandsocialmedia.
- ManagingAPIcallsandresponsesforthird-partyservices.
- Ensuringreal-timedatasynchronizationbetweenservicesandthebuilder.

Thediagramillustratesthemodularstructureofthewebsitebuilderplatform.ItbeginswiththeUserManagementModule 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 LibraryModule. Third-party services are integrated via the Integration Module, and the final website is published using the Deployment Module, resulting in a Live Website.

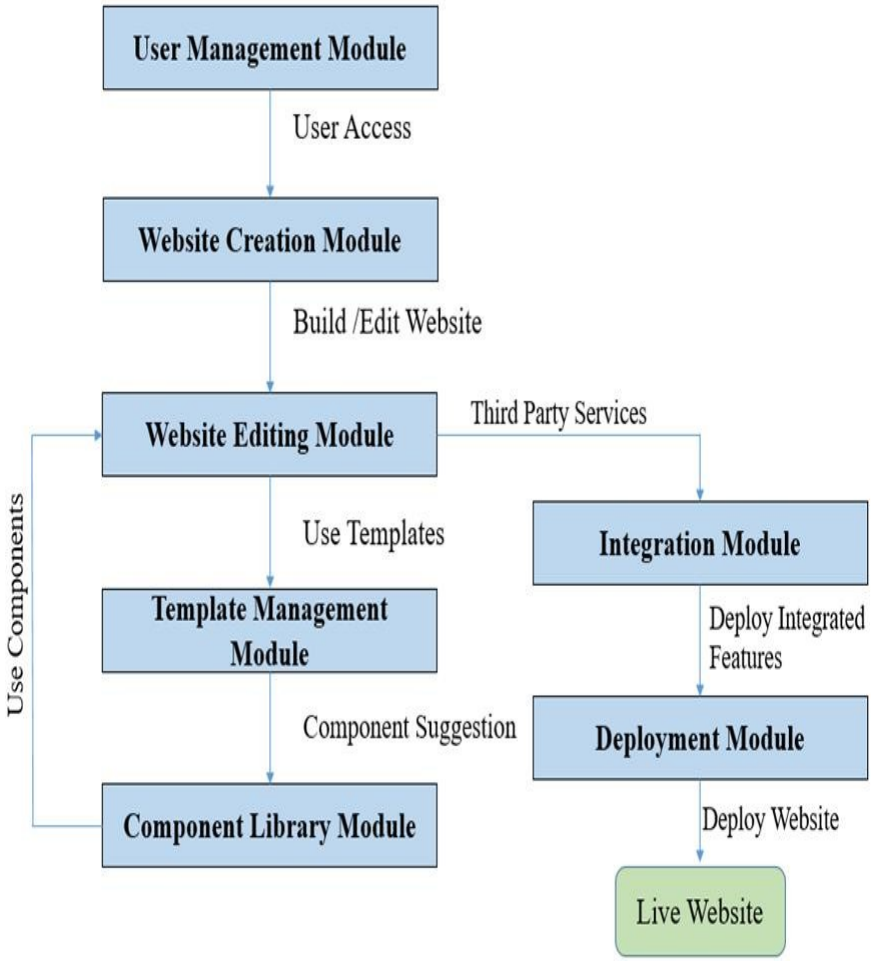


Fig.1ModularStructureofQuickwebcraft

Thissystemarchitectureillustratesthemodularstructureofthe Quickwebcraftlow-codeSaaSplatform,showcasingthe integrationof frontend,backend services, databases,third-partyAPIs,and deploymentlayers.Itemphasizesscalability, maintainability,andflexibility throughdistinctmodulessuchasdrag-and-dropUI,real-timeservices,APIgateway,and auto-scaling, supporting seamless development and deployment workflows.

©IJRASET: All Rights are Reserved

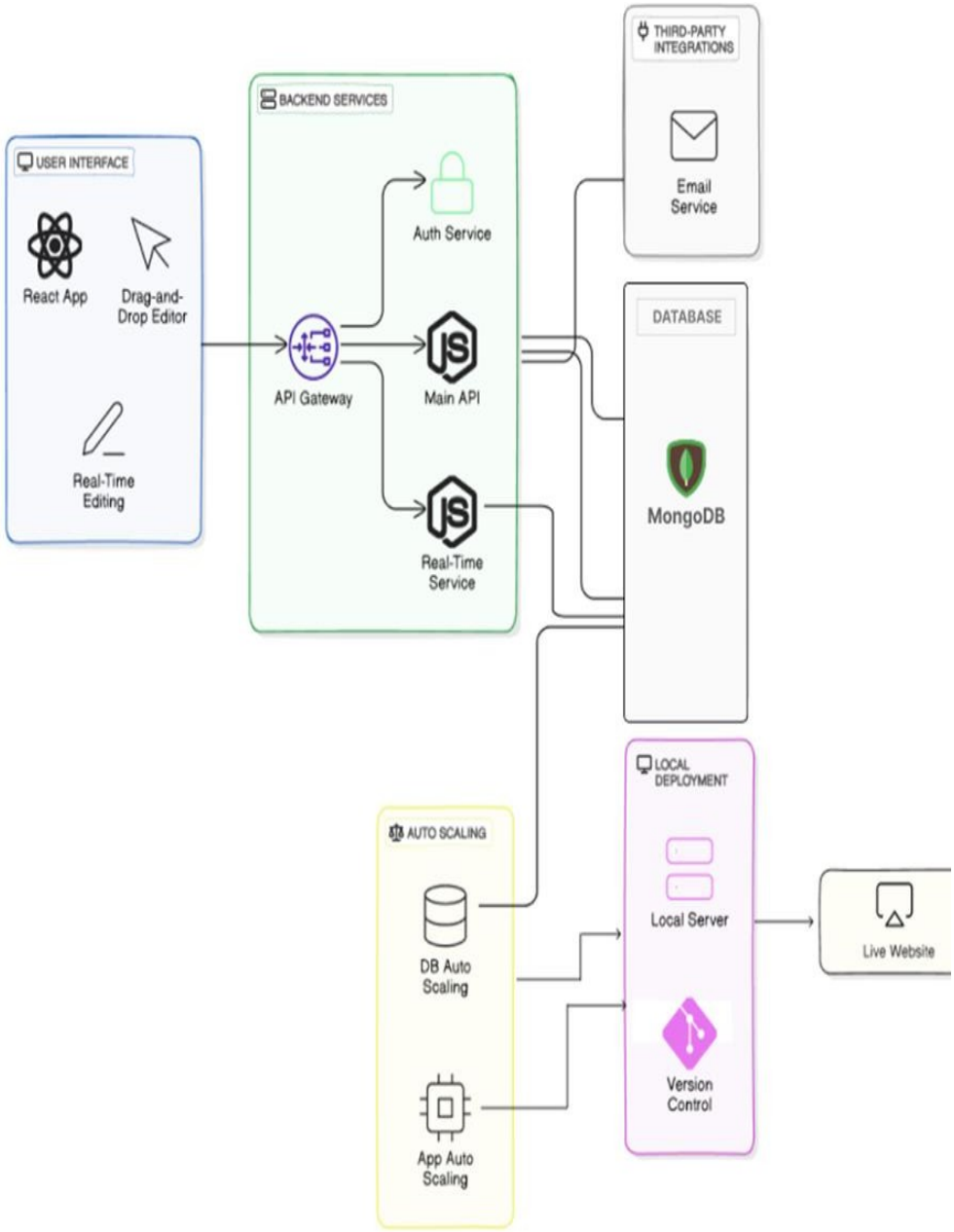


Fig.2SystemArchitectureofQuickwebcraft

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.

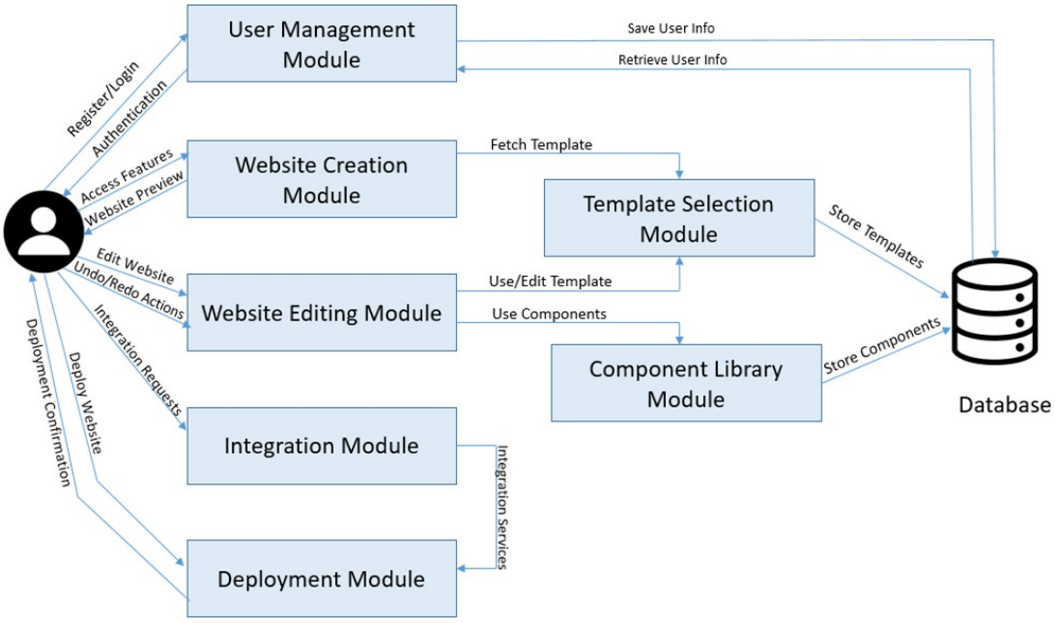


Fig.3DFDLevelDiagram

VI. CONCLUSION AND FUTURE WORK

Thedevelopmentofthelow-codeSaaSplatformhighlightsa modularanduser-friendlyapproachto websitecreation,editing, and deployment. By leveraging modern technologies and a well-structured architecture, the system aims to simplify web development for users with minimalcodingknowledge. Maintainingaclear workflowand regular iterationhasbeenessentialin building a functional and scalable platform.

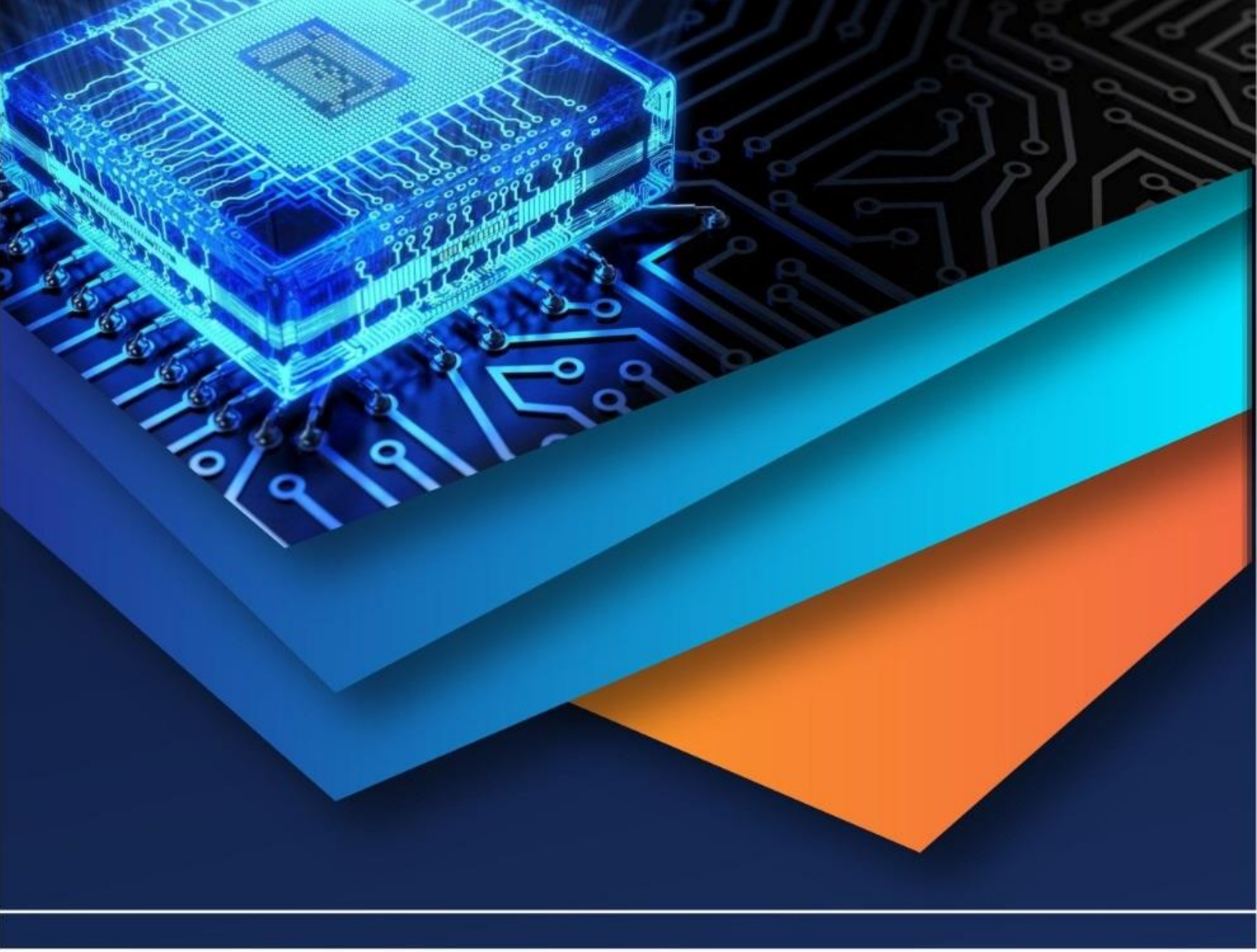
Inthefuture,theplatformcanbeenhancedwithAI-drivendesignsuggestions,real-timecollaborationfeatures,andsupportfor morethird-partyintegrations.Continuedimprovements willfocusonboostingperformance,expandingtemplateandcomponent libraries, and enriching the overall user experience.

VII.ACKNOWLEDGMENT

Wewouldliketoexpressoursinceregratitudetoourprojectguideandfacultymembersfortheirconsistentsupportandguidance 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 onInformation (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 onIntelligent 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 StudyApproach,"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,andM.Wagner,"LeveragingLow-CodeDevelopmentforBusinessProcessAutomation,"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 TechnologiesConference(FTC) 2021, Volume3. FTC 2021.LectureNotes inNetworks and Systems, vol360. 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)