International Journal of Research Publications in Engineering, Technology and Management (IJRPETM)

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 9, Issue 3, May-June 2026||

DOI:10.15662/IJRPETM.2026.0903001

Low-Code Mobile Application Builder with AI-Assisted Features using Flutter & Firebase

Dr.V.Seedha Devi[1], Harshini R[2], Dhana Lakshmi E[3], Gayathri N[4], Nithesha P[5]

Associate Professor, Department of Information Technology, Jaya Engineering College, Anna University, Chennai, Tamil Nadu, India[1]

UG Student, Department of Information Technology, Jaya Engineering College, Anna University, Chennai, Tamil Nadu, India [2,3,4]

Publication History: Received: 03.04.2026; Revised: 28.04.2026; Accepted: 02.05.2026; Published: 06.05.2026.

ABSTRACT: The increasing demand for rapid application development has highlighted the limitations of traditional coding-intensive approaches, particularly in terms of development time, complexity, and accessibility. This project presents the design and implementation of a Low-Code Application Builder, a platform developed to streamline and simplify the process of creating fully functional applications with minimal manual coding. The system provides an intuitive visual interface that enables users to design applications using drag-and-drop components, predefined templates, and configurable elements. It supports key functionalities such as dynamic user interface generation, real-time preview, backend integration, and database connectivity within a unified development environment. This approach allows both technical and non-technical users to efficiently build, modify, and deploy applications.. By incorporating reusable components and automated workflows, the system significantly reduces development effort while maintaining consistency and reliability. In conclusion, the proposed Low-Code Application Builder enhances development efficiency, minimizes technical barriers, and accelerates the application lifecycle, making it a practical solution for modern software development requirements.

KEYWORDS Low-Code Development, Template-Based Design, Backend Customization, Database Management, Platform , Apk , Dart, Depolyment, App Builder, Firebase

I. INTRODUCTION

In the current digital era, the demand for rapid and reliable application development has grown significantly due to the increasing dependence on web and mobile platforms across industries. Businesses and individuals require applications that can be developed, tested, and deployed in shorter timeframes without compromising functionality or performance. However, conventional software development approaches are often constrained by their reliance on extensive coding, complex system architecture, and the need for skilled developers, making the process both time-intensive and resource-demanding .Low-code development has emerged as a transformative approach that addresses these challenges by abstracting complex programming tasks into simplified visual operations

II. LITERATURE SURVEY

Low-Code/No-Code (LCNC) platforms have emerged as an effective solution to simplify software development in the IT industry. These platforms enable users to build applications using visual tools, pre-built components, and minimal coding, reducing dependency on skilled developers and accelerating development processes. It also addresses challenges including limited customization, security concerns, scalability issues, and vendor lock-in. The findings suggest that with proper governance, security practices, and user training.[1]

No/Low-Code (LCNC) development platforms are emerging as effective solutions to reduce the complexity and time involved in traditional web development. This project presents the design and partial implementation of a web-based LCNC platform that enables users to create functional websites using drag-and-drop components, pre-built templates, and configurable design elements with minimal coding effort., improving extensibility, and enhancing system performance and security for real-world deployment.[2]

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 9, Issue 3, May-June 2026||

DOI:10.15662/IJRPETM.2026.0903001

The rapid pace of digital transformation across industries has created a growing demand for application development approaches that are faster, more flexible, and accessible beyond traditional software engineering teams. In response, Low-Code and No-Code (LCNC) platforms have emerged as a transformative paradigm in modern application development.. It examines how LCNC platforms bridge the gap between professional developers and non-technical users, promoting citizen development and accelerating innovation.[3]

Low-code and no-code (LCNC) development platforms are significantly transforming modern software architecture by simplifying the application development process. These platforms enable users to design and develop applications with minimal or no coding through the use of visual interfaces, drag-and-drop tools, and pre-built components. Therefore, careful implementation strategies, including proper governance, training, and system planning, are essential for effective adoption.[4]

Low-code and no-code (LCNC) platforms are transforming modern software development by reducing dependence on extensive programming skills and enabling faster application creation. These platforms utilize visual development environments, drag-and-drop components, pre-built templates, and automated workflows to simplify the development process often referred to as citizen developers, can efficiently participate in application design, testing, and deployment. their successful adoption depends on effective governance, strategic planning, and alignment with organizational objectives.[5]

III. PROBLEM STATEMENT

Traditional software development methodologies involve complex processes, including extensive coding, multi-layered system architecture, and integration of frontend, backend, and database components. These requirements demand skilled developers, increased development time, and higher costs, making application development less accessible to non-technical users and small-scale organizations. As the demand for rapid and scalable digital solutions continues to grow, these limitations create a significant gap between development capability and user requirements.

Low-Code and No-Code (LCNC) platforms have been introduced to address these challenges by providing visual development environments, drag-and-drop interfaces, and reusable components. While these platforms simplify application development and enable faster delivery, they are often constrained, restricted control over backend processes, and inadequate support for complex data handling. In addition, issues related to scalability, performance optimization, data security, and vendor dependency further reduce their effectiveness in real-world applications.

IV. RESEARCH METHODOLOGY

The proposed Low-Code Application Builder system follows a modular and integrated architecture designed to simplify application development while maintaining flexibility and scalability. The system is structured around a Flutter-based mobile interface that serves as the primary interaction layer for both users and administrators. This approach ensures cross-platform compatibility and a consistent user experience.The methodology begins with the Authentication Module, which handles user validation through secure login and signup mechanisms, including social authentication.

This module ensures that only authorized users can access the platform and perform operations within the system.The core component of the system is the Application Builder Module, which provides a visual drag-and-drop interface for designing user interfaces and configuring backend logic. This module enables users to create applications without extensive coding knowledge by utilizing predefined components and customizable configurations. It also supports backend customization, allowing users to define application behavior and data flow.

International Journal of Research Publications in Engineering, Technology and Management (IJRPETM)

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 9, Issue 3, May-June 2026||

DOI:10.15662/IJRPETM.2026.0903001

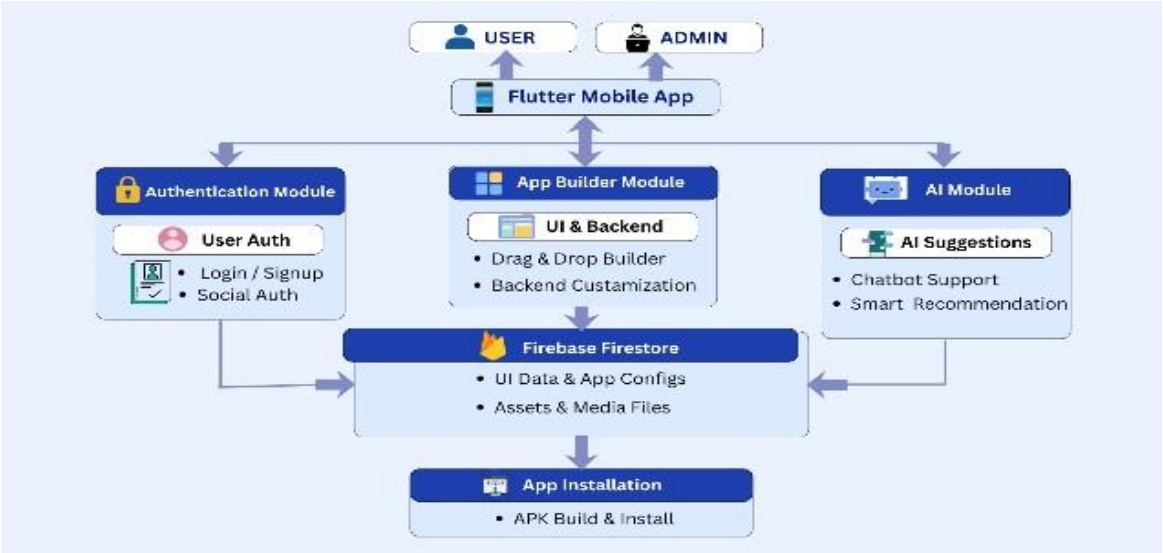

Fig.1 Architecture Diagram For AppForage

All application data, configurations, and media assets are managed through Firebase Firestore, which acts as the centralized cloud database for real-time data synchronization. The AI module enhances the system by providing intelligent suggestions, chatbot support, and smart recommendations to improve user experience and decision-making. After the app design and configuration are completed, the system proceeds to the final stage of APK generation and installation. This structured flow ensures a seamless transition from user interaction to application deployment, making the architecture efficient, user-friendly, and suitable for low-code development environments.

The system is designed to operate in a continuous feedback loop, where user interactions and application performance data are constantly monitored and refined. The AI module plays a key role here by analyzing usage patterns, detecting inefficiencies, and suggesting improvements in UI design, workflow, and feature usage. Admins can leverage these insights to optimize the application without rebuilding it from scratch. Additionally, the integration between modules ensures smooth data flow and minimal latency, which is critical for real-time applications The screen supports both **standard** authentication and third-party login options, such as social sign-in, to improve accessibility and user convenience. Input validation mechanisms are implemented to ensure correct data entry, while error handling provides appropriate feedback for invalid credentials or login failures.

1. Authentication Module

This module is responsible for managing secure user access and identity verification within the system. It supports multiple authentication methods, including traditional login/signup using email and password, as well as third-party social authentication such as Google or OAuth-based providers. The module ensures strong validation mechanisms, including input sanitization, password strength checks, and prevention of common vulnerabilities like SQL injection and brute-force attacks.

To enhance security, sensitive user data such as passwords are encrypted using hashing algorithms, and session management techniques like token-based authentication (JWT) are implemented. Role-based access control (RBAC) is applied to differentiate permissions between normal users and administrators, ensuring restricted access to sensitive features. Additionally, features like password recovery, multi-factor authentication (MFA), and session timeout handling can be integrated to further strengthen security. Overall, this module acts as a critical gateway, ensuring that only authorized users can access and interact with the system securely.

2. App Builder Module

This module is the core component of the platform, enabling users to create fully functional applications through a visual, low-code development environment. It provides an intuitive drag-and-drop interface where users can design UI layouts using pre-built components such as buttons, forms, images, and navigation elements. These components can be customized in terms of style, behavior, and responsiveness without requiring extensive programming knowledge.

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 9, Issue 3, May-June 2026||

DOI:10.15662/IJRPETM.2026.0903001

Beyond frontend design, the module also supports backend configuration, allowing users to define databases, APIs, and data models directly within the platform. Workflow automation features enable users to create logical connections between components, such as form submissions triggering database updates or API calls. Real-time preview functionality allows users to instantly visualize changes, improving development efficiency and reducing errors.The module also includes template selection, reusable components, and version control mechanisms to streamline the development process. Integration with external services such as cloud storage, payment gateways, and third-party APIs further extends its capabilities. By combining UI design, backend logic, and deployment features in a single environment, this module significantly reduces development time and simplifies full-stack application creation.

3. AI Module

This module introduces intelligence into the system by leveraging artificial intelligence and machine learning techniques to enhance user experience and optimize application development. It provides real-time suggestions during app creation, such as recommending UI layouts, color schemes, or component placements based on best practices and user behavior patterns.The AI-powered chatbot assists users by answering queries, guiding them through the app-buildingprocess, and resolving common issues, reducing dependency on manual support

Additionally, the module analyzes user interactions and application data to provide predictive insights, such as identifying potential errors, suggesting performance improvements, and recommending feature enhancements.Advanced capabilities may include natural language processing (NLP) for converting user instructions into functional components, automated code generation, and intelligent debugging support Additionally, the integration between modules ensures smooth data flow and minimal latency, which is critical for real-time applications The screen supports both **standard** authentication and third-party login options. The AI continuously learns from user activity and system data, enabling adaptive improvements over time. This results in smarter, more efficient applications and a more interactive and personalized user experience.

V. RESULT

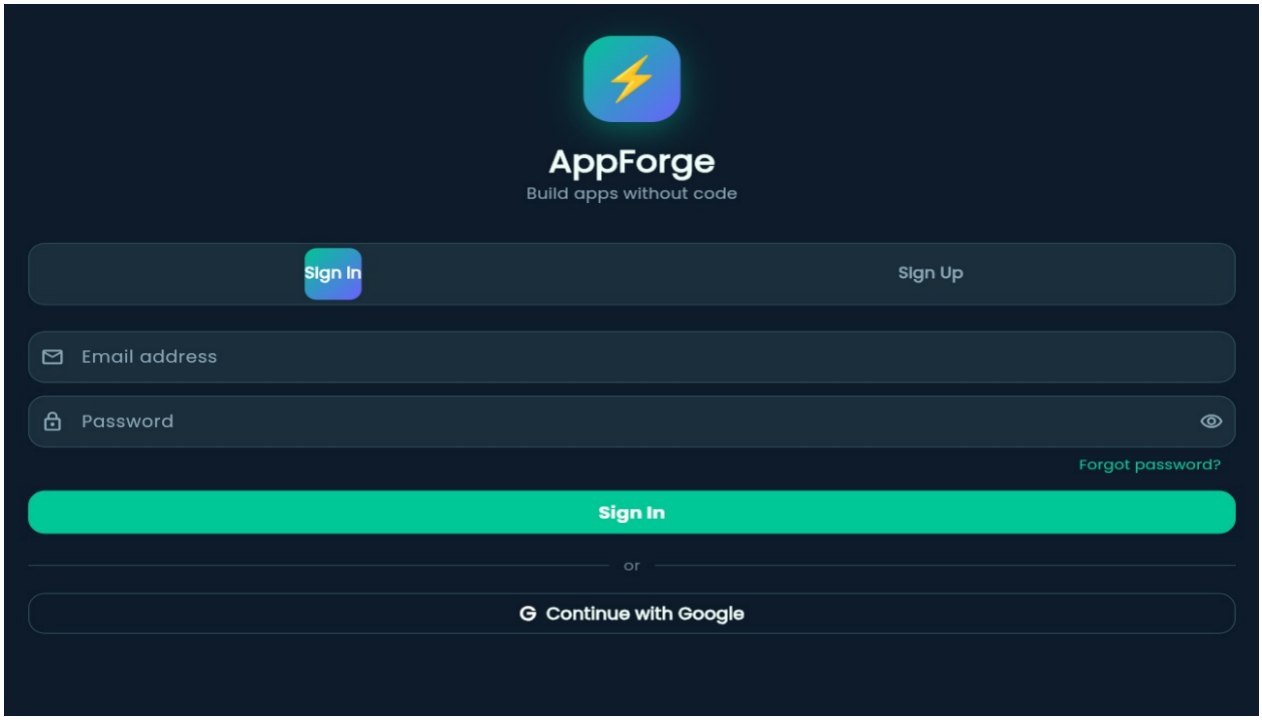

Fig:2 Login Screen

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 9, Issue 3, May-June 2026||

DOI:10.15662/IJRPETM.2026.0903001

Fig. 2 shows the Login Screen serves as the primary entry point to the Low-Code Application Builder platform, ensuring secure and authenticated access for users and administrators. It is designed with a clean and user-friendly interface that allows users to enter their credentials, including email and password

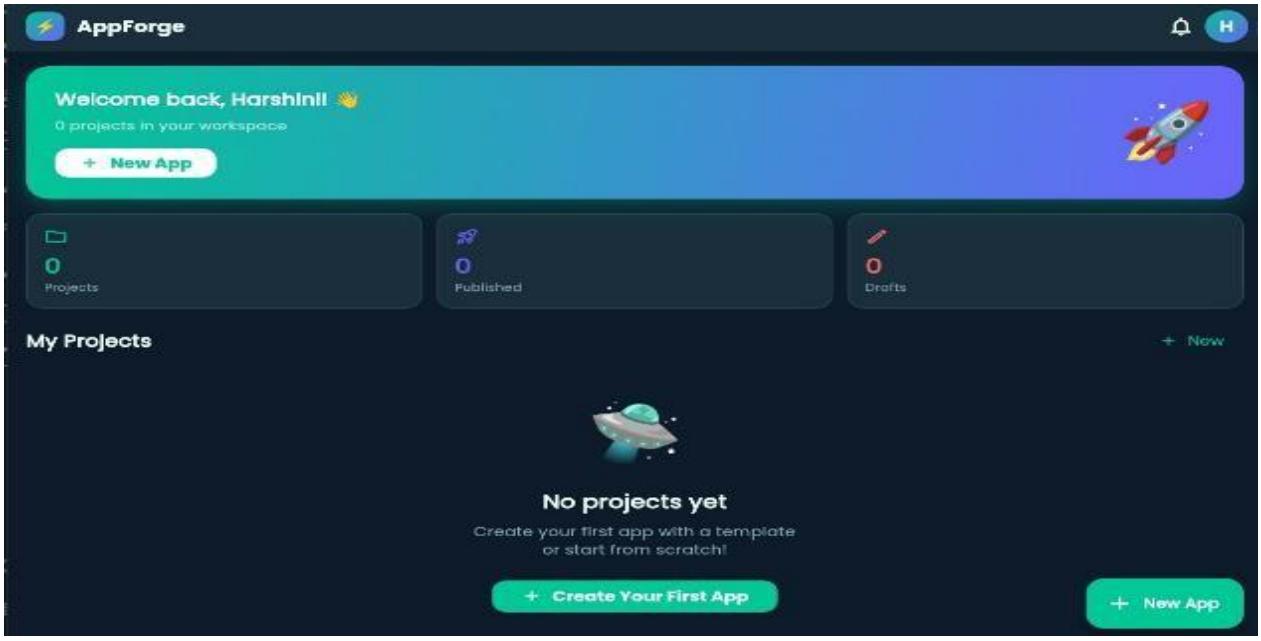

Fig.3 Dashboard Interface of Low-Code App

Fig. 3 shows the dashboard screen in a low-code app builder should act as the central control panel where users get a clear overview of their projects and system activity

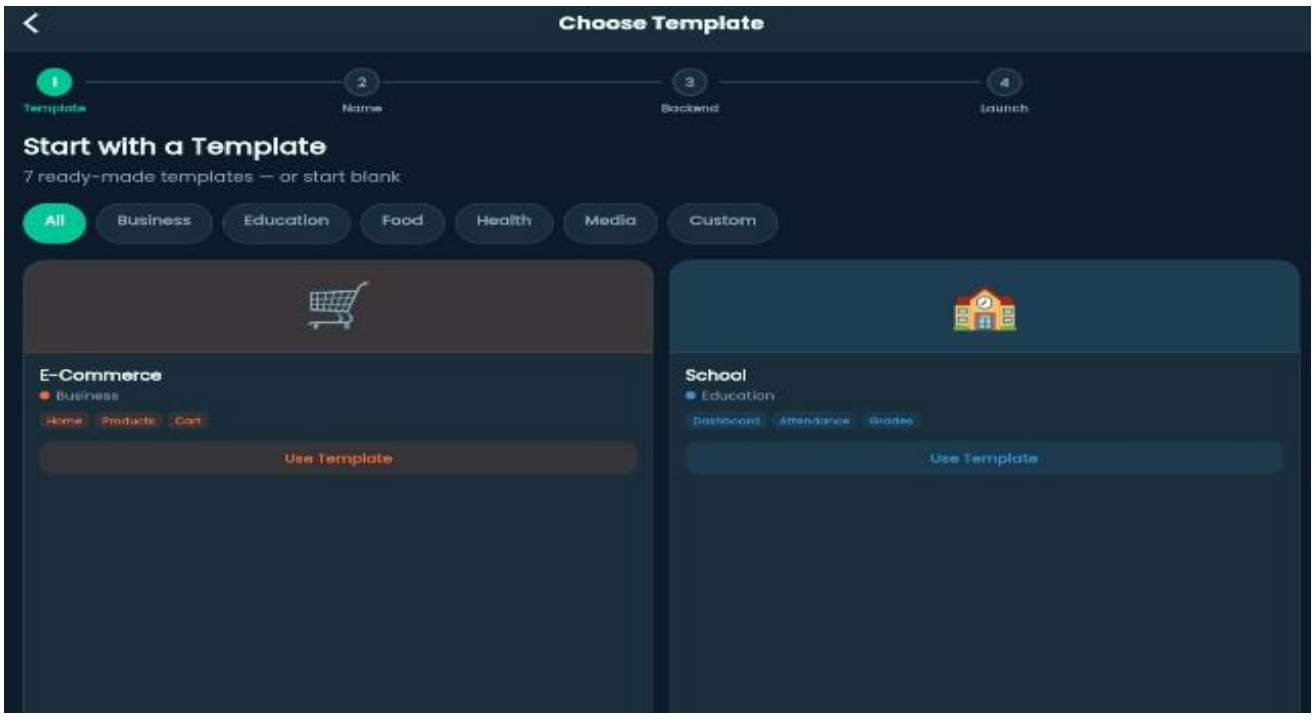

Fig. 4 Template Selection Interface

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 9, Issue 3, May-June 2026||

DOI:10.15662/IJRPETM.2026.0903001

Fig. 4 shows the template selection screen is where users choose the starting point for their application, making it a critical step in the low-code development process.

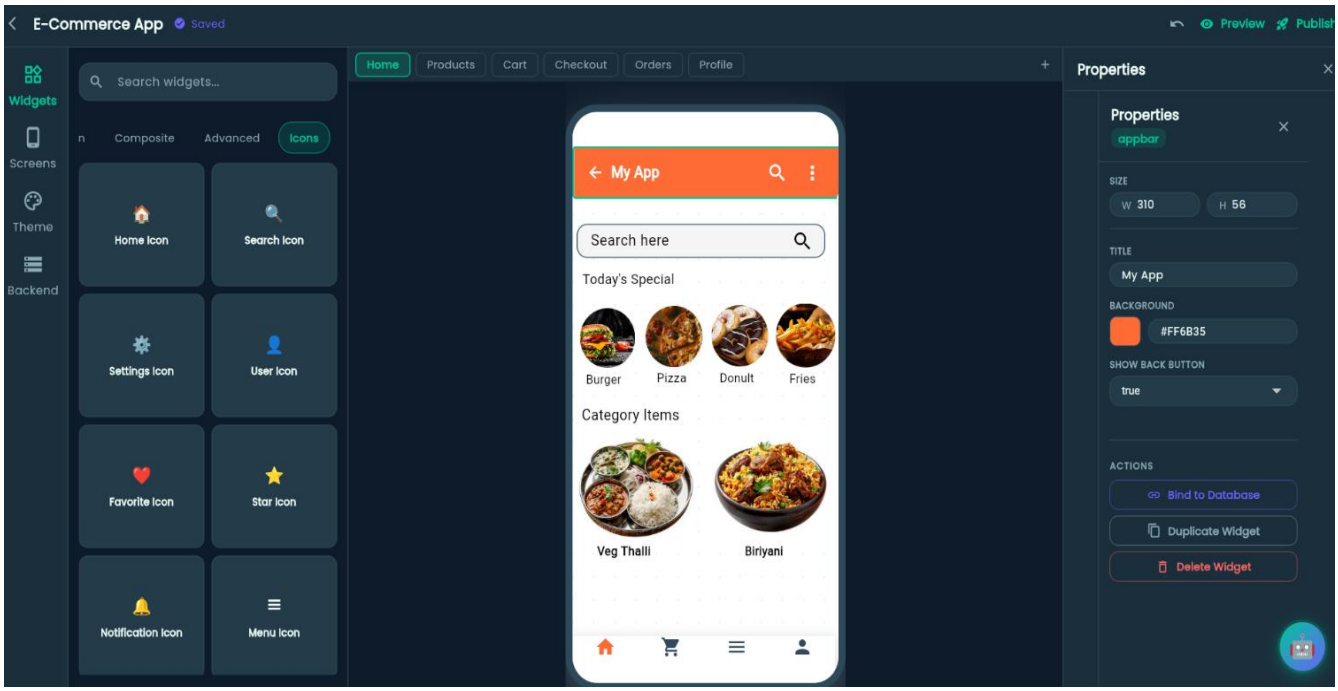

Fig. 5 App Builder Interface with Drag-and-Drop Functionality

Fig. 5 shows the app builder screen is the core workspace where users design and develop their application using a visual, drag-and-drop interface. It allows users to add UI components like buttons, text fields, images, and layouts, and position them easily without writing code.

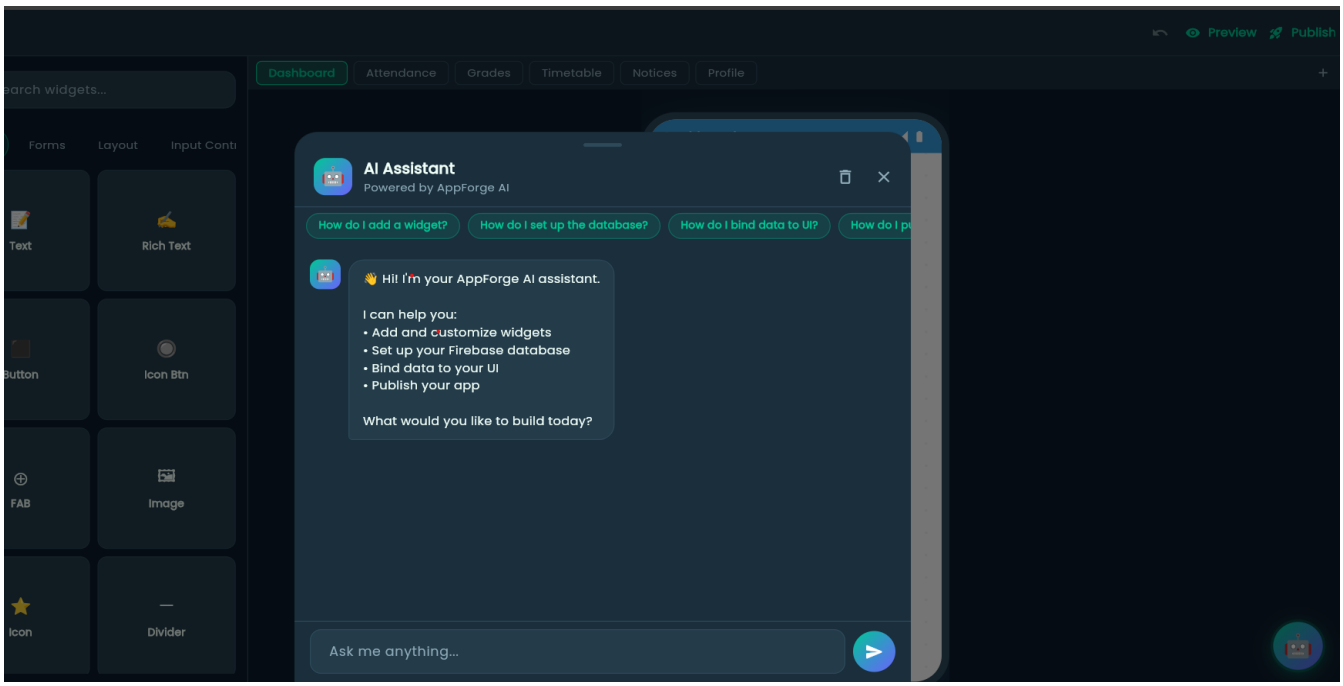

Fig :6 AI Chat Bot

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 9, Issue 3, May-June 2026||

DOI:10.15662/IJRPETM.2026.0903001

Fig. 6 shows the AI Chat Screen enables users to interact with an intelligent assistant that helps in building and modifying applications efficiently. It provides real-time guidance, suggestions, and automation support based on user inputs

.

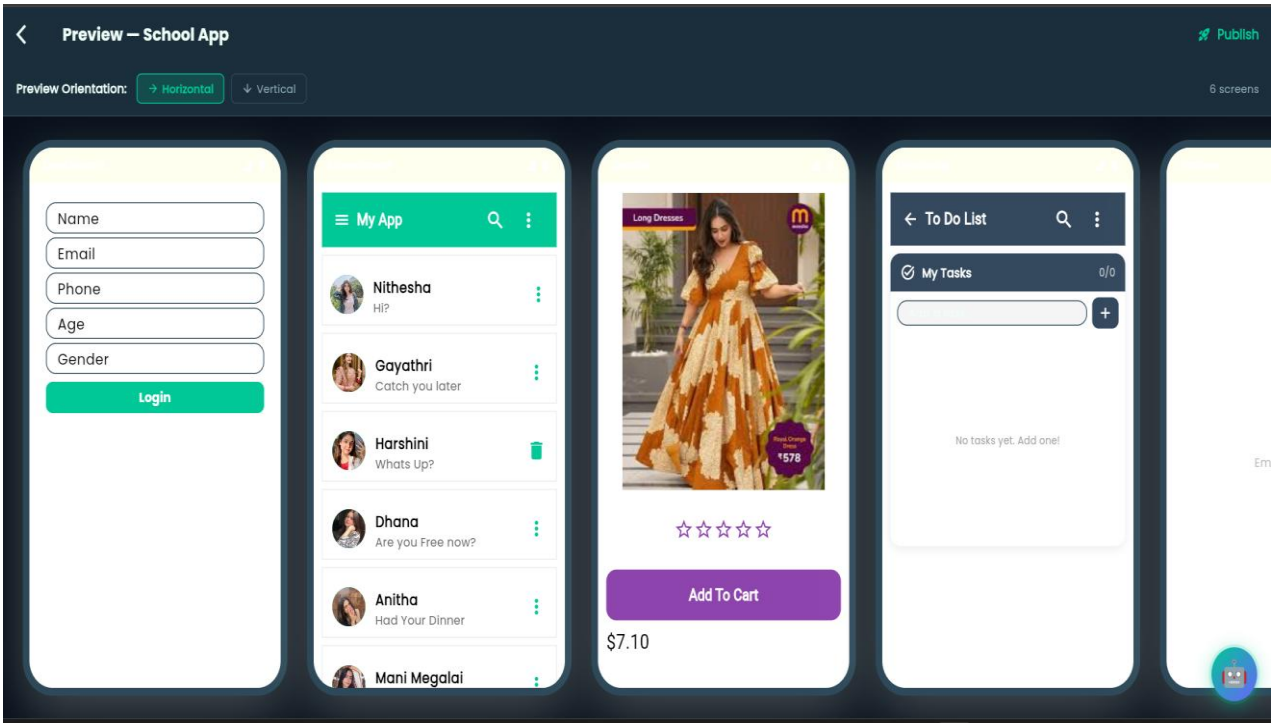

Fig:7 Preview Screen For App Forage

Fig. 7 shows the Preview Screen allows users to view their application in real-time before publishing. It provides an interactive simulation of how the app will appear and behave on end-user devices.

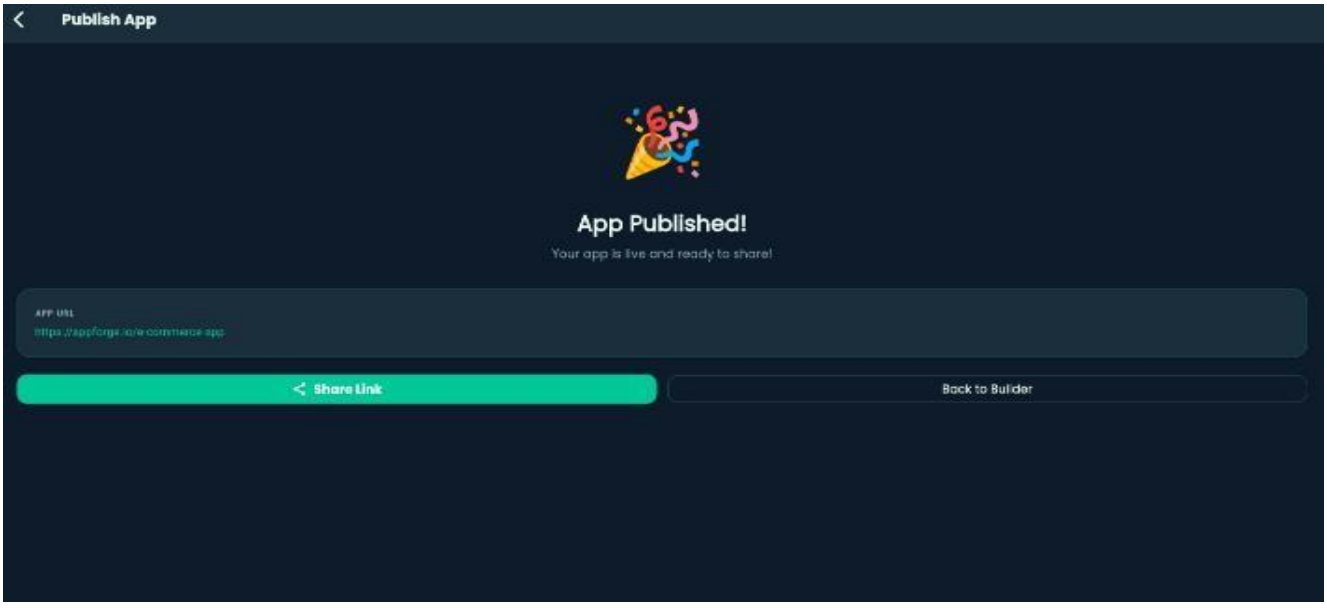

Fig. 8 Application Published Output with Shareable Link

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 9, Issue 3, May-June 2026||

DOI:10.15662/IJRPETM.2026.0903001

Fig. 8 shows the application publish screen is the final stage where users prepare and deploy their app for real-world use. It allows users to configure important settings such as app name, version, icon, permissions, and target platform before publishing.

VI. CONCLUSION AND FUTURE ENHANCEMENT

This paper presented the design and implementation of a Low-Code Application Builder platform aimed at simplifying the process of application development for both technical and non-technical users. The system integrates a visual development environment with support for template-based and customized application creation, enabling users to design interfaces, configure backend functionality, and manage data within a unified platform .The inclusion of real-time preview, database integration, and backend connectivity ensures that the developed applications are not limited to static structures but support dynamic and functional operations. Furthermore, the platform reduces the dependency on extensive coding knowledge, thereby accelerating the development lifecycle and improving productivity. The integration of intelligent features such as guided workflows and AI-assisted support enhances usability and enables users to build applications with better design consistency and fewer errors. This approach not only minimizes development time but also bridges the gap between idea and implementation.

The current Low-Code Application Builder provides a functional foundation for simplified application development; however, several enhancements can significantly improve its capability, scalability, and real-world usability. One major area of improvement is advanced backend integration, where support for custom APIs, third-party services, and microservices architecture can be introduced. This will allow developers to extend application functionality beyond predefined configurations and build more complex, enterprise-level solutions .The platform can be enhanced by incorporating real-time collaboration features, enabling multiple users to work on the same project simultaneously. This would improve team productivity and align the system with modern collaborative development practices.

REFERENCES

1. F. Sufi, “Algorithms in Low-Code-No-Code for Research Applications: A Practical Review,” Algorithms, vol. 16, no. 2, 2023.
2. Saravanan, M. (2026). Generative AI–Enabled Decision Intelligence: An Integrated Analytics and Autonomous Systems Framework for Cybersecurity and Retail Enterprises. International Journal of Engineering & Extended Technologies Research (IJEETR), 8(1), 14-20.
3. Aarthi, K., Thirumoorthy, P., Tamizharasu, K., Manoja, R., Kalyanasundaram, P., & Rajasekar, M. (2025, September). Improved Network lifetime using Cluster based Power-Aware Balanced Routing Protocol for Device to Device Communication. In 2025 6th International Conference on Electronics and Sustainable Communication Systems (ICESC) (pp. 1005-1010). IEEE.
4. Rengarajan, A. (2025). Cloud-Based AI-Driven Threat Detection Framework for Smart Grid Cybersecurity. International Journal of Future Innovative Science and Technology (IJFIST), 8(6), 16065.
5. Mathew, A. (2024). Cloud data sovereignty governance and risk implications of cross-border cloud storage. Information Systems Audit and Control Association.
6. Seedha Devi, V., Namitha, B., Divya Dharshini, J., & Livetha, K. (2026). A hybrid biometric and geo-fencing based smart attendance system. International Journal of Advanced Research in Computer Science and Technology (IJARCST), 9(3), 794–802. https://doi.org/10.15662/IJARCST.2026.0903002
7. V. Viswanadhapalli, “The Future of Intelligent Automation: How Low-Code/No-Code Platforms are Transforming AI Decisioning,” International Journal of Engineering and Computer Science, 2025.
8. Jaiswal, M., Manchattahalli, U. R., Devi, V. S., & Jacob, M. S. (2026, January). Predictive Analysis of Internet Use and Risk Among Children and Adolescents via Machine Learning. In 2026 Sixth International Conference on Advances in Electrical, Computing, Communications and Sustainable Technologies (ICAECT) (pp. 1-6). IEEE.
9. B. John, “Low-Code and No-Code Platforms: Accelerating Enterprise Software Development,” World Journal of Engineering and Technology, 2025.
10. Murugeshwari, B., Selvaraj, D., Sudharson, K., & Radhika, S. (2023). Data Mining with Privacy Protection Using Precise Elliptical Curve Cryptography. Intelligent Automation & Soft Computing, 35(1).
11. Seedha Devi, V., Nivedha, S., Harisha, V., Mol, D. R., & Janaranjini, J. R. (2026). Enhanced prediction of PCOS and PCOD using deep learning for early diagnosis and clinical risk stratification. International Journal of Advanced Research in Computer Science & Technology (IJARCST), 9(3), 783–793.

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 9, Issue 3, May-June 2026||

DOI:10.15662/IJRPETM.2026.0903001

12. Kesavan, R., Anuradha, T., Kalaiarasi, P., Jacob, M. S., Anand, A. J., & Devi, V. S. (2025, November). Deep Learning-Based Diagnostic Support System for Breast Tissue Classification Using Mammographic Images. In 2025 IEEE International Conference on Intelligent Signal Processing and Effective Communication Technologies (INSPECT) (pp. 1-6). IEEE.
13. Gopalakrishnan, S., Dhinakaran, D., Raja, S. E., Raghavan, P., & Girija, M. S. (2026). Fusion-Driven Medical Image Encryption Framework with Entropy-Calibrated Control and Integrity Assurance. KSII Transactions on Internet & Information Systems, 20(2).
14. Mathew, A. (2025). Deep seek vs. ChatGPT: A deep dive into AI Language mastery. Int J Multidisciplinary Res, 7(1), 1-5.
15. Vimal, V. R., & Banerjee, J. S. (2025). Integrating PSO, GA, and ACO for Optimized ECG Feature Selection and Classification of Cardiac Disorders. SGS-Engineering & Sciences, 1(5).
16. G. L. Liwanag, R. Ebardo, and D. Cheng, "Low-Code and No-Code Development in the Era of Artificial Intelligence: A Systematic Review," Data and Metadata, 2025.
17. H. El Kamouchi, M. Kissi, and O. El Beggar, "Low-Code/No-Code Development: A Systematic Literature Review," IEEE Conference on Intelligent Systems, 2023.
18. N. Ahmed et al., "The Synergy of Low-Code/No-Code and AI/ML: Enhancing Intelligent Automation," World Journal of Engineering and Technology, 2025.
19. Dhinakaran, D., Prathap, P. J., Selvaraj, D., Kumar, D. A., & Murugeshwari, B. (2022). Mining privacy-preserving association rules based on parallel processing in cloud computing. International Journal of Engineering Trends and Technology, 70(3), 284-294.
20. Seedha Devi, V., Kumar, M. D., & Kumar, C. A. (2026). Flutter-based SOS alert and location tracking application with volunteer assist and rescue. International Journal of Research and Applied Innovations (IJRAI), 9(3), 521–530. https://doi.org/10.15662/IJRAI.2026.0903003
21. A. C. Bock and U. Frank, "Low-Code Platform," Business & Information Systems Engineering, 2021.
22. Y. Luo, P. Liang, C. Wang, M. Shahin, and J. Zhan, "Characteristics and Challenges of Low-Code Development," ACM/IEEE ESEM, 2021.
23. Vimal, V. R. (2025). Next Generation Enterprise Architecture for SAP Cloud Systems Leveraging AI Driven Analytics and Hybrid Infrastructure. International Journal of Engineering & Extended Technologies Research (IJEETR), 7(6), 11174-11182.
24. Jacob, M. S., Rekha, V. B., Manchattahalli, U. R., & Devi, V. S. (2025, October). Enhancing the Performance of Energy-Harvesting Wireless Sensor Networks through Machine Learning-based Energy Prediction. In 2025 10th International Conference on Communication and Electronics Systems (ICCES) (pp. 1558-1563). IEEE.
25. IEEE, "Low-Code/No-Code Platforms and Modern ERP Systems," IEEE Conference Publication, 2023.
26. Anbazhagan, K. (2025). Next-Generation Enterprise Cloud AI for Healthcare: Secure CNN Pipelines and Privacy Controls. International Journal of Future Innovative Science and Technology (IJFIST), 8(6), 15980.
27. IEEE, "Low-Code/No-Code Development: A Systematic Literature Review," IEEE Conference, 2023.
28. Devi, V. S., Kumar, G. S., Jacob, M. S., Jeevitha, S., & Jegatheesan, A. (2025, October). Comparative Study of Traditional and Deep Learning Approaches for Human Ear-Based Biometric Identification. In 2025 2nd International Conference on Software, Systems and Information Technology (SSITCON) (pp. 1-6). IEEE.
29. M. Desmond et al., "A No-Code Low-Code Paradigm for Authoring Business Automations Using Natural Language," arXiv, 2022.
30. Rajasekar, M. (2025). Risk-Aware Generative AI and Machine Learning Frameworks for Privacy-Preserving Banking and Trade Analytics over Cloud and 5G Networks. International Journal of Computer Technology and Electronics Communication, 8(4), 11078-11086.
31. Sugumar, R. (2025). Federated AI in Offline-First Mobile Health Architectures for Privacy-Preserving Clinical Intelligence. International Journal of Science, Research and Technology, 8(4), 14589-14600.
32. O. Ogundare et al., "No Code AI: Automatic Generation of Function Block Diagrams," arXiv, 2023.
33. S. Curty, F. Härer, and H. G. Fill, "Blockchain Application Development Using Low-Code Platforms: A Survey," arXiv, 2022.
34. Anujaa, T., Thajudeen Ali Ahamed, A. F., Baranwal, V., Thanikaiselvan, V., Subashanthini, S., Sivaranjani Devi, C., & Rengarajan, A. (2025). A lightweight multi round confusion-diffusion cryptosystem for securing images using a modified 5D chaotic system. Scientific Reports, 15(1), 31986.
35. Anbazhagan, K. (2025). AI Driven Zero Trust Security Model for Enterprise Data Protection and Intelligent Infrastructure Management. International Journal of Technology, Management and Humanities, 11(03), 101-107

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 9, Issue 3, May-June 2026||

DOI:10.15662/IJRPETM.2026.0903001

36. Prabha, S. P., & Rengarajan, A. (2025). ENHANCING CLOUD RESOURCE ALLOCATION WITH VISION TRANSFORMER, DEEP REINFORCEMENT LEARNING, AND IMPROVED SHRIKE OPTIMIZATION ALGORITHM. Corrosion Management ISSN: 1355-5243, 35(2), 233-245.
37. Seedha Devi, V., Selvi, D., Uma Maheshwari, K., & Yuvashree, G. (2026). Food linker: A smart system for global waste reduction. International Journal of Engineering & Extended Technologies Research (IJEETR), 8(3), 5012–5021. https://doi.org/10.15662/IJEETR.2026.0803002
38. Mathew, A., Jackson, E., & Tobesman, A. (2025). Evaluating the Efficacy of WPA3 against Advanced Attacks: A Comparative Analysis with WPA2 in Real-World. J Inform Techn Int, 3(1), 105.
39. J. Cabot, “Positioning of the Low-Code Movement within Model-Driven Engineering,” ACM/IEEE MODELS Conference, 2020.