

FLUTTERCHAT: A Real-time Firebase Chat Application with AI-based Chatbot

S. Umamaheswari¹, S. Saranya^{2,*}

Abstract

Recently, the development and deployment of chatbots have gathered significant attention from both developers and researchers. Chatbots represent AI-driven conversational systems capable of understanding and responding to human language using advanced techniques like Natural Language Processing (NLP) and Neural Networks (NN). A cutting-edge real-time chat application has been crafted using Flutter and OpenAI, seamlessly integrating an AI-powered chatbot with an innovative image generator to enrich user interaction and engagement. The Firebase Chatbot Application is its ability to transcend conventional image sharing methods. Users can effortlessly transmit multiple images within a conversation, which are intelligently compiled into a single PDF file. This innovation streamlines the sharing process, ensuring that recipients receive a cohesive and organized collection of images. Moreover, the application empowers users to engage with PDF content directly within the chat interface. By integrating PDF viewing capabilities, individuals can access and review documents shared within conversations without leaving the app. This seamless integration enhances user convenience and fosters efficient collaboration, whether it involves sharing project proposals, presentations, or important documents. The recommended chatbot can be developed using technologies like Flutter, Dart, Android Studio, and Firebase. Furthermore, the implementation will utilize Machine Learning (ML) and Deep Learning (DL) techniques, including Neural Machine Translation (NMT) and Deep Reinforcement Learning (RL) models, to enhance functionality and performance.

Keywords: Real-time chat application, AI-based chatbot, image generator, Flutter, OpenAI, personalized responses, natural language processing

INTRODUCTION

Flutter Chat Application with PDF Scanner and Reader and Event Reminder is a powerful mobile application that allows users to communicate with each other in real-time, scan and read PDF files, and set reminders for important events. The Flutter Chat feature enables users to send and receive messages, share image and event remainder [1].

*Author for Correspondence

S. Saranya
E-mail: seesaran08@gmail.com

¹Professor, Department of Information Technology, C. Abdul Hakeem College of Engineering and Technology, Melvisharam, Tamil Nadu, India

²Assistant Professor, Department of Computer Science and Engineering, C. Abdul Hakeem College of Engineering and Technology, Melvisharam, Tamil Nadu, India

Received Date: December 19, 2024

Accepted Date: January 09, 2025

Published Date: January 24, 2025

Citation: S. Umamaheswari, S. Saranya. FLUTTERCHAT: A Real-time Firebase Chat Application with AI-based Chatbot. Journal of Mobile Computing, Communications & Mobile Networks. 2025; 12(1): 42–51p.

Event Remainder

The Event Reminder feature in our application is specifically designed to help users schedule and send messages at the desired time, ensuring timely communication with their contacts. This feature is particularly useful for individuals who need to send messages at specific times, such as appointment reminders, birthday greetings, or work-related notifications. With the Event Reminder feature, users can simply compose their message and select the date and time they want it to be sent. The feature

will then automatically send the message at the scheduled time, even if the user is not actively using the application. This ensures that important messages are delivered promptly, even if the user forgets or is unable to send them manually [2].

Image-to-PDF

Our app includes a feature that enables users to convert images into PDF files, simplifying the process of sharing and storing visual content in a consistent and easily accessible format. This feature is particularly useful for individuals who need to share images with others, such as business professionals, educators, or students. To use this feature, users can simply select one or more images and choose the “Convert to PDF” option. This feature is designed to be user-friendly and efficient, allowing users to convert images quickly and easily to PDF format without the need for additional software or tools. By integrating this capability into our application, we aim to enhance user productivity, streamline document handling, and provide a valuable tool for anyone looking to digitize and organize visual content effectively [3].

PDF Viewer

The Firebase Chat Application is a robust messaging platform that enables real-time communication between users. A standout feature of this app is its ability to display PDF files containing only images. This is especially beneficial for users who need to share visual content like diagrams, charts, or graphics in a uniform and easily accessible format. The app includes an integrated PDF viewer, allowing users to view and interact with image-only PDF files. This feature is integrated directly into the app, providing a seamless and convenient way for users to view and share visual content without the need to switch to a separate app or tool [4].

MOTIVATION

The motivation behind the Firebase chat application with an inbuilt image-to-PDF converter, PDF viewer, event reminder, and chat functionality is to provide a comprehensive and user-friendly platform for communication and collaboration. The addition of an image-to-PDF converter enables users to transform images into PDF format, simplifying document sharing and management within the app. The built-in PDF viewer allows users to open and read PDF files directly in the app, removing the need for external PDF viewers. The event reminder feature helps users stay organized and on top of their schedules by sending reminders for important events. The app allows users to chat with each other and share images, making it a versatile platform for communication and collaboration. The app's integration with Firebase ensures that data is stored securely and synced in real-time, providing a seamless user experience, making it an ideal tool for personal and professional use. Overall, the motivation behind this app is to provide a powerful and user-friendly platform for communication and collaboration, leveraging the latest technologies to deliver an exceptional user experience [5].

PROBLEM STATEMENT

The problem statement for this application is the need for a comprehensive and user-friendly platform that integrates image-to-PDF conversion, PDF viewing, event reminders, and chat functionality. Currently, users have to switch between multiple apps to perform these tasks, leading to inefficiencies and time wastage. The proposed solution is a Firebase chat application that integrates all of these features in to one platform, providing a seamless user experience and enhancing productivity [6].

LITERATURE SURVEY

The existing system does not have any built-in features for creating or viewing PDF files or setting event reminders. The search results do not offer detailed information about the current system but provide some useful insights into PDF viewers and PDF creators. The first search result is a manual for a software application for regulatory compliance. The manual mentions a PDF viewer, but it does not provide any information about creating PDF files or setting event reminders. The second search result is a user guide for Nitro Pro, a software application for creating and editing PDF files. The guide

mentions a PDF viewer and a PDF creator, but it does not provide any information about setting event reminders. The third search result is a documentation page for the Stream Chat API, which includes a section on reminders [7].

The section provides information about how to use reminders in a chat application, but it does not provide any information about creating PDF files or setting event reminders. The fourth search result is a Stack Overflow question about canceling a background process when leaving a component in an Angular application. The question is not related to PDF files or event reminders, but it does mention a PDF viewer library for Angular. The fifth search result is a user guide for Nitro Pro 11, which includes information about creating and editing PDF files. The guide mentions a PDF viewer and a PDF creator, but it does not provide any information about setting event reminders [8].

SYSTEM CONFIGURATION

Software Requirements

The software required for this application include:

- *Flutter*: A cross-platform mobile app development framework used to build the user interface and functionality of the chat application. Flutter is an open-source UI development kit created by Google that enables the creation of natively compiled applications for mobile, web, and desktop from a single codebase, offering flexible and expressive design options. It is a cross-platform UI framework developed by Google for building applications that are natively compiled across mobile, web, and desktop platforms using a single codebase [9].
- *Firebase*: A backend-as-a-service platform used for real-time messaging, data storage, and user authentication in the application. Firebase is an all-in-one platform for mobile and web app development, providing features such as real-time databases, authentication, cloud storage, hosting, and analytics, making backend infrastructure management easier for developers. Firebase is a comprehensive platform by Google for building mobile and web applications, offering services like real-time database, authentication, hosting, analytics, and more, simplifying backend development and scaling [10].
- *Android Studio*: Android Studio is the official Integrated Development Environment (IDE) for Android app development, offering tools for coding, debugging, and testing. It supports Kotlin and Java, providing powerful features to build high-quality mobile applications efficiently, and is also used for Flutter development and Firebase integration.
- *Dart*: Dart is the programming language used with Flutter to develop the application's logic and functionality. Created by Google, Dart is recognized for its flexibility in building web, mobile, and server applications. It is often used with Flutter for cross-platform app development, featuring a strong type system and Just-In-Time (JIT) compilation. Dart is a programming language created by Google, primarily used for developing mobile, web, and desktop applications, particularly in conjunction with Flutter.

Hardware Requirements

RAM:	Minimum 8 GB or Higher
GPU:	8 GB
Processor:	Intel Pentium 5 or Higher
SSD:	16 GB or Higher
Device:	Android Device

METHODOLOGY

Crafting robust algorithms is pivotal for implementing advanced functionalities like AI-based chatbots and image generation within applications. These algorithms, ranging from natural language processing to deep learning models, undergo extensive research and experimentation to refine conversation quality and accuracy. Ensuring the security of the application is paramount, involving the implementation of secure authentication mechanisms, robust data encryption techniques, and stringent

access control measures. These measures ensure the protection of data's confidentiality, integrity, and availability. Integration of security protocols such as SSL/TLS establishes secure communication channels, safeguarding sensitive information against unauthorized access or manipulation. Testing and validation procedures are fundamental components of the methodology, guaranteeing the functionality, performance, and reliability of the system. Using different testing methods such as unit testing, integration testing, and user acceptance testing helps identify and resolve any issues or flaws. Validation ensures the system meets user expectations and requirements, ultimately providing a seamless and engaging chat experience for users.

Authentication Module

This module manages user authentication, enabling users to log in with their Gmail accounts. Once authenticated, users can view and update their profiles. Utilize Firebase Authentication to enable Gmail sign-in and store user profiles in Fire store.

Google Login Process

To log in to a Google Chat application, users need to follow a specific process outlined in the provided. When a Chat app is invoked, it requires authentication to access user data beyond basic identity information. By default, Chat apps can read basic identity data like display name, user ID, email address, and avatar image. If a chat app requires additional data access, it requests permission from the user, who must then approve it. This authentication process ensures secure access to user data. Integrating Google login into a Firebase Chat Application involves configuring Firebase Authentication to accept Google accounts as a sign-in method.

After setting up the Firebase project and enabling Google Sign-In in the Firebase console, developers need to add the necessary configuration files to their Android (`google-services.js on`) or iOS (`GoogleService-Info.plist`) projects. Then, they install the Firebase SDK and implement the Google Sign-In button in the app's UI. The sign-in logic is handled by initializing the `Google SignInClient`, launching the sign-in intent, and handling the result. Upon successful sign-in, Firebase Authentication verifies the Google credentials and provides the user with a Firebase user token, allowing access to the Firebase services. Developers must also manage the user authentication status and address any errors that may occur during the sign-in process.

Chat Interface Module

This module provides the user interface for real-time messaging between users. Support for text formatting, emojis, and multimedia messages. Chat interface module in a Firebase Chat Application provides users with a simple yet intuitive platform for real-time communication. It includes features like a message input field for sending texts, a display area for viewing message history with timestamps, and user avatars for visual identification. Powered by Firebase's real-time database, the interface ensures instant message delivery and synchronization across devices. Additional features such as message status indicators, typing indicators, and customization options enhance user engagement. Overall, the chat interface module prioritizes efficiency, reliability, and ease of use, enabling seamless communication within the Firebase Chat Application.

Event Remainder Interface Module

This module helps users set reminders for events. Users have the ability to create new events or modify existing ones. Use Firebase Fire store to store event data and schedule reminder notifications using Firebase Cloud Messaging local. Users can set reminders for important events directly within the chat, utilizing intuitive controls and interactive elements. The module displays upcoming events in a visually appealing format, providing relevant details such as event title, date, time, and any associated notes. Users can easily create, edit, and delete reminders, with changes instantly reflected in the chat interface. Firebase's real-time database ensures that reminders are synchronized across all devices, enabling users to stay updated regardless of their platform. Additionally, notification alerts for

upcoming events further enhance usability, ensuring users never miss important appointments or deadlines. Overall, the event reminder interface module streamlines task management within the Firebase Chat Application, offering a convenient and efficient solution for organizing events and staying productive.

Image-to-PDF Converter Module

The image-to-PDF module seamlessly converts images into PDF files within the chat interface. Users can select images, apply settings, and generate PDFs instantly. This feature enhances document sharing and archiving, optimizing communication efficiency directly within the chat application. The image-to-PDF converter module enhances the functionality of the Firebase Chat Application by allowing users to convert images shared within conversations into PDF files seamlessly. Integrated directly into the chat interface, this module enables users to select multiple images and initiate the conversion process with a simple click. Once converted, the PDF file is intelligently compiled, ensuring a cohesive and organized collection of images. Users can then share the PDF within the chat, facilitating easy dissemination of visual content while preserving the original quality and layout. The module leverages Firebase's real-time database to ensure swift conversion and synchronization across all devices. Additionally, users can view and interact with PDF content directly within the chat interface, enhancing convenience and usability. Overall, the image-to-PDF converter module enhances the versatility of the Firebase Chat Application, providing a seamless solution for managing and sharing visual content efficiently.

Integration Module

It ensures seamless integration of authentication, chat, file sharing, event reminder, and task highlights modules. Utilize Flutter's testing framework for writing and executing tests, and Firebase Test Lab for testing on various devices and configurations. The integration module of the Firebase Chat Application enhances its capabilities by allowing smooth integration with external services and platforms. This module allows users to incorporate various features and services directly into their chat conversations, enhancing collaboration, productivity, and user experience. Through APIs and SDKs, developers can integrate functionalities such as file sharing, task management, calendar scheduling, and more, empowering users to perform diverse tasks without leaving the chat interface.

Chatbot-AI

The creation of the chatbot system necessitates a detailed research approach to guarantee its efficiency, ease of use, and security. This approach includes multiple stages, beginning with the collection of requirements through interviews, surveys, and conversations with stakeholders. These requirements form the foundation of the system, addressing both functional and non-functional elements essential for its successful development. Stakeholders, such as potential users, domain experts, and developers, offer crucial insights into user needs, system features, and specific functionalities required for the chatbot. Once requirements are gathered, the next phase involves system modeling.

Various techniques, including use case diagrams, entity-relationship diagrams, and activity diagrams, are used to represent the structure and behavior of the chatbot system. These models provide a comprehensive view of the system's components, relationships, and interactions, helping to enhance the understanding of its architecture and information flow. This phase sets the foundation for the following design and development stages. This phase also involves identifying entities, attributes, and relationships essential for storing and retrieving data. Database management systems and data modeling tools are utilized to design a robust data architecture that can accommodate the needs of chatbot while ensuring optimal performance and reliability. User interface design is an essential part of the methodology, concentrating on developing intuitive and visually attractive interfaces for the chatbot application.

DESIGN AND IMPLEMENTATION

Creating a chat app with scheduled messaging, image-to-PDF conversion, and PDF reading using Flutter and Firebase is technically feasible. Operational aspects focus on user-friendly design and integration.

Economic considerations involve development costs and potential productivity gains. Deployment timelines hinge on resource availability, with stakeholder engagement crucial for goal alignment and project success. System architecture diagrams assist in maintaining consistency and ensuring alignment with both the system requirements and business goals. They provide clarity on how different components interact and contribute to the system's functionality, enabling stakeholders to make informed decisions about design choices, technology selection, and resource allocation. Overall, system architecture diagrams are essential tools for effectively designing, implementing, and managing complex systems in various domains, from software applications to large-scale infrastructure deployments. Figure 1 depicts components such as databases, servers, modules, interfaces, and their relationships, utilizing symbols and diagrams to represent the flow of data or processes.

Encoding and Decoding

Several studies conducted in the past 5 years have been reviewed to enhance the development of chatbots. These studies are analyzed based on their objectives, methodologies, strengths, and outcomes. One prominent area of investigation involves neural machine translation (NMT), which employs neural network models to learn statistical models for machine translation. The NMT model adopts a Sequence-to-Sequence (Seq2Seq) architecture with encoder-decoder framework.

Encoder

The encoder component handles the input data (e.g., source language sentences in machine translation) and converts it into a fixed-length vector representation. This is commonly done using recurrent neural networks (RNNs) or long short-term memory (LSTM) networks. The encoder's final hidden state holds the summarized information of the input sequence.

Decoder

The decoder component takes the encoded vector representation generated by the encoder as input and produces the output sequence (e.g., target language sentences in machine translation). It starts with a special token (e.g., <start> token) and iteratively generates the output tokens. An attention mechanism can be used to selectively focus on specific parts of the input sequence during the decoding process.

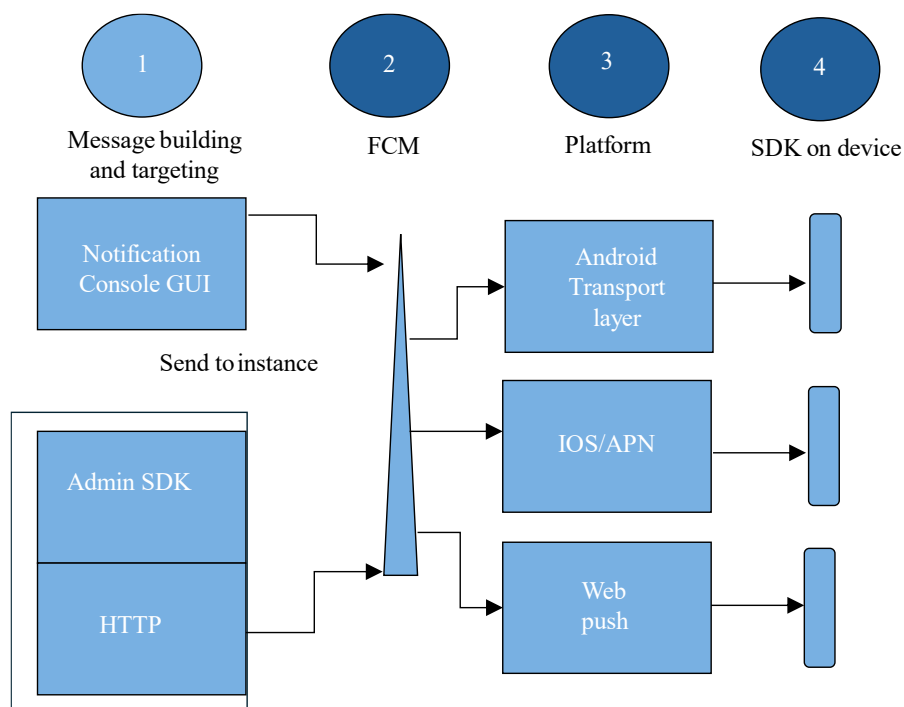


Figure 1. Architecture Diagram.

Embedding Layer

Transforms input tokens into compact vector representations that are compatible with processing by the neural network. Figure 2 shows the components such as databases, servers, modules, interfaces, and their relationships, utilizing symbols and diagrams. Overall, the Seq2Seq architecture with encoder-decoder framework forms the basis for many NMT models and has been successfully applied in various sequence-to-sequence tasks, including machine translation, text summarization, and chatbot response generation.

LSTM/RNN Layer

Long short-term memory (LSTM) or recurrent neural network (RNN) layers are used in sequential data processing. These layers capture the temporal relationships in the input sequence and produce context vectors.

RESULT AND DISCUSSION

This groundbreaking real-time chat application combines AI-powered chatbot technology and an image generator using Flutter and OpenAI. The chatbot utilizes advanced natural language processing algorithms to deliver context-aware responses, enhancing user engagement. The image generator produces custom visuals to enhance conversations. By utilizing Flutter's cross-platform capabilities and user-friendly interface design, this chatbot sets a new benchmark for chat applications, offering tailored responses and engaging visuals. This highlights AI's ability to enhance user engagement while setting the stage for future advancements in chat technology. Looking forward, potential improvements might incorporate help for bunch talks, start to finish encryption, and combination with other Firebase administrations.

Generally speaking, this task shows Firebase's adaptability in building responsive correspondence applications. Its dependability, versatility, and simplicity of combination pursue it an appealing decision for designers expecting to make highlight rich, constant correspondence answers for satisfying the needs of the cutting-edge computerized scene. The Firebase Chat Application offers a dynamic and feature-rich platform for seamless communication enriched with multimedia capabilities. With its intuitive chat interface, real-time messaging powered by Firebase, and innovative features like image-to-PDF conversion, it redefines digital interaction. By prioritizing user convenience, security, and scalability, this application fosters efficient collaboration and connectivity across various platforms. Whether for personal or professional use, the Firebase Chat Application stands as a testament to the convergence of communication and technology, providing users with a versatile solution that meets the evolving needs of modern communication in a reliable and efficient manner.

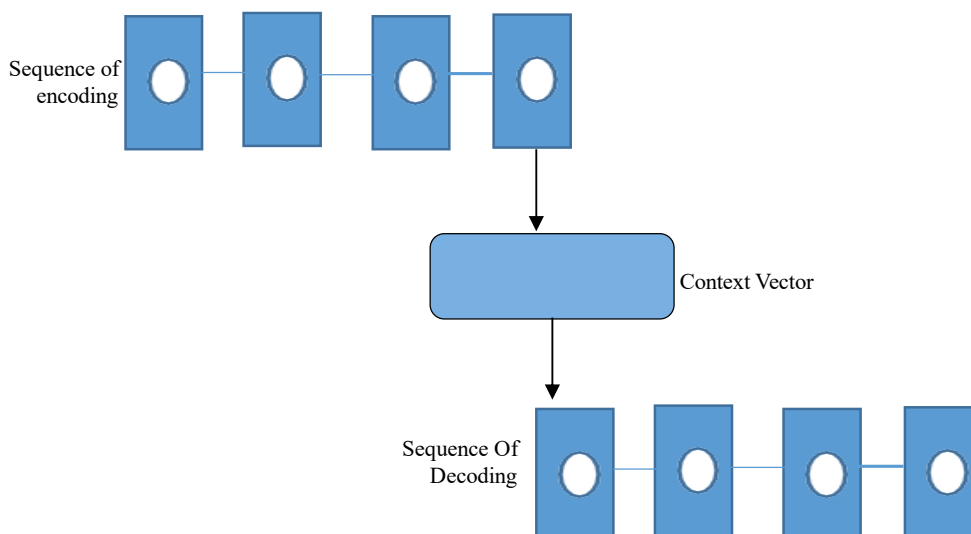


Figure 2. Encoding and Decoding diagram: Seq2Seq.



Figure 3. Login Page.



Figure 4. User Chat Interface.

SCREENSHOTS

Figure 3 indicates login page is a web page or interface that allows users to authenticate themselves before accessing a certain service, platform, or application. Its primary purpose is to verify the identity of users by requiring them to input credentials such as a username and password. In Figure 4, user chat interface is a graphical application that facilitates real-time interactions, enabling users to communicate with either other individuals or automated systems like chatbots.

It is commonly found in messaging apps, customer support platforms, and various online communication tools. Figure 5 depicts that sending and receiving images in a chat interface enhances communication by allowing users to share visual content. Figure 6 shows that event reminder interface module for a Firebase Chat Application enhances user productivity and organization by integrating event scheduling features seamlessly into the chat interface.

CONCLUSION

In conclusion, FLUTTERCHAT sets a new standard for chat applications, combining AI-driven interactions, personalized visuals, and extensive integrations with essential business tools.

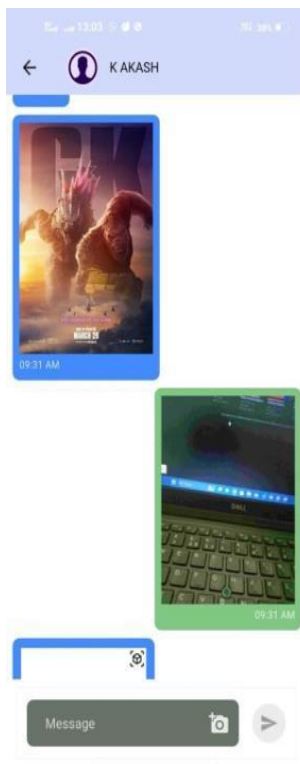


Figure 5. Sending images and PDF.

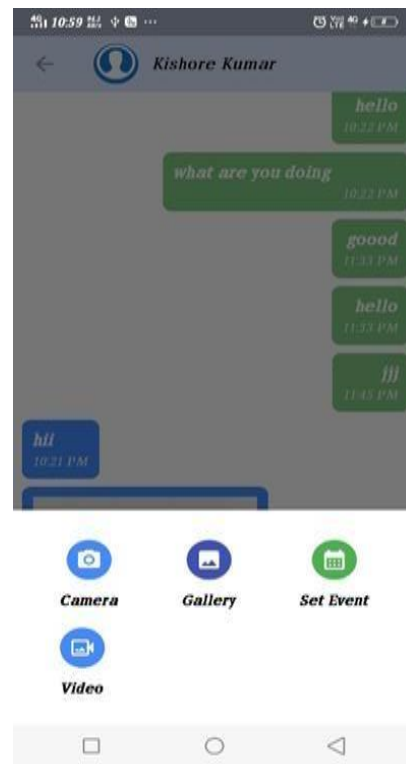


Figure 6. Integration of chat, file sharing, event reminder, and task highlights modules.

Its robust platform compatibility, real-time collaboration features, and insightful analytics underscore its potential to transform communication and workflow management. This innovative application not only enhances user engagement but also paves the way for future advancements in chat technology, demonstrating the limitless possibilities of AI in improving productivity and connectivity. By exploring these future developments, FLUTTERCHAT aims to stay at the forefront of innovation in chat technology, continually enhancing its features and capabilities. From advanced AI and AR integration to expanded business tool integrations and enhanced security measures, FLUTTERCHAT is committed to meeting the evolving needs of users and businesses worldwide. This forward-thinking approach ensures that FLUTTERCHAT remains a leader in providing seamless, personalized, and secure communication solutions across multiple platforms and diverse user demographics.

REFERENCES

1. Smyth N. *Firestore essentials-Android edition*. Payload Media, Inc.; North Carolina. 2017 Jul 18.
2. Tanna M, Singh H. *Serverless Web Applications with React and Firebase: Develop real-time applications for web and mobile platforms*. Packt Publishing Ltd, United Kingdom; 2018 Apr 10. 1–259.
3. Khawas C, Shah P. Application of firebase in android app development-a study. *Int J Comput Appl*. 2018 Jun; 179(46): 49–53.
4. Ревенчук ІА, Осташко ЕВ. Minimizing the costs of building complex cross-platform Applications with flutter and firebase. *Біоніка інтелекту*. 2022 Dec 30; 1(98): 49–57.
5. Oleksiuk V, Verbovetskyi DV, Hrytsai ІА. Design and development of a game application for learning Python. In *Proceedings of the 6th Workshop for Young Scientists in Computer Science & Software Engineering (CS&SE@ SW 2023)*. Virtual Event, Kryvyi Rih, Ukraine, CEUR Workshop Proceedings. 2024 Feb 2; 3662: 111–124.
6. Kumar A. *Mastering Firebase for Android Development: Build real-time, scalable, and cloud-enabled Android apps with Firebase*. Packt Publishing Ltd, United Kingdom; 2018 Jun 29.

7. Kukade A, Ramteke A. Android Mobile Monitoring Application with Google Firebase Real-Time Database. *International Journal of Scientific Research & Engineering Trends (IJSRET)*. 2021; 7(3): 1462–5.
8. Dodda R, Yanala MR, Pavishetti S. Notes Application using Android Studio and Cloud-based Data Management. In *2024 IEEE International Conference on Inventive Computation Technologies (ICICT)*. 2024 Apr 24; 1427–1436.
9. Chatterjee N, Chakraborty S, Decosta A, Nath A. Real-time communication application based on android using Google firebase. *Int J Adv Res Comput Sci Manag Stud*. 2018 Apr; 6(4): 74–79.
10. Rajarajeswari PL, Gowshick H, MK SP, Anuvidh M. Reach—A Chat Application. In *2024 IEEE 10th International Conference on Advanced Computing and Communication Systems (ICACCS)*. 2024 Mar 14; 1: 1961–1966.