



# SPARK-A Dating Application: A JAVA, Flutter and ANDRIOD-Based

Prof. Madhuri Parate<sup>1</sup>, Ishant Fulzele<sup>2</sup>, Yash Jugseniya<sup>3</sup>, Gaurav Katole<sup>4</sup>,  
Aditya Rewatkar<sup>5</sup>, Rutuja Bhende<sup>6</sup>

Professor Department Of Computer Science And Engineering,

Nagarjuna Institute Of Engineering Technology And Management, Nagpur, Maharashtra, India<sup>1</sup>

UG Student, Department Of Computer Science And Engineering, Nagarjuna Institute Of Engineering Technology &  
Management, Nagpur, Maharashtra, India<sup>2,3,4,5</sup>

## I.ABSTRACT

**Abstract:** The rapid growth of mobile applications has significantly transformed the way people connect and communicate. This project, titled “**Spark – A Modern Cross-Platform Dating Application,**” focuses on developing a user-friendly and efficient dating platform using Flutter and Firebase technologies. The application is designed to operate seamlessly across multiple platforms, including Android, iOS, and Web, using a single codebase.

The primary objective of the project is to provide a simple, secure, and real-time environment for users to discover potential matches, express interest through a swipe-based interface, and communicate instantly upon mutual matching. The system incorporates features such as user authentication using email and Google OAuth, profile creation and management, swipe-to-like functionality, automatic match detection, and real-time chat powered by Cloud Firestore.

**Keywords:**Flutter, Firebase, Cross-Platform Application, Real-Time Chat, Dating App, Cloud Firestore, User Authentication, Social Networking, Mobile Application Development.

## II.INTRODUCTION

The rapid growth of mobile technology has transformed the way people connect and communicate in today’s digital world. Dating applications have become a popular platform for individuals to discover and interact with potential partners conveniently. This project, “**Spark – A Modern Cross-Platform Dating Application,**” focuses on developing a user-friendly and efficient system that enables users to find matches and communicate in real time. Built using Flutter and Firebase, the application ensures a seamless experience across multiple platforms with secure and scalable functionality. With the rapid advancement of mobile technology and internet connectivity, digital platforms have become a primary medium for social interaction and relationship building. Among these, dating applications have gained significant popularity by enabling users to connect, communicate, and form relationships in a convenient and efficient manner.

The project “**Spark – A Modern Cross-Platform Dating Application**” is developed to provide a seamless and interactive platform for users to discover potential matches and engage in real-time communication. The application is built using Flutter, a powerful cross-platform development framework, allowing deployment on Android, iOS, and Web platforms through a single codebase.

## III.METHODOLOGY

The proposed methodology for the development of the **Spark – Cross-Platform Dating Application** follows a systematic and iterative approach to ensure the successful design, implementation, and deployment of a scalable and efficient system. The methodology is structured into multiple phases, including requirement analysis, system design, implementation, integration, testing, and deployment. This approach ensures that the system meets functional and non-functional requirements while maintaining high performance, usability, and security.

The initial phase involves **requirement analysis**, where the system requirements are identified based on user needs and current limitations of existing dating applications. The functional requirements include user registration and authentication, profile management, swipe-based interaction, mutual match detection, and real-time messaging. Non-functional requirements such as performance, scalability, security, and cross-platform compatibility are also considered to ensure a robust system design.



Following requirement analysis, the **system design phase** is carried out, which defines the overall architecture and structure of the application. The system adopts a **feature-first architectural approach**, where the application is divided into independent modules based on functionalities such as authentication, home/swipe, matches, chat, and profile management. This modular structure enhances code readability, maintainability, and scalability. The user interface is designed using Material Design principles to provide a consistent and intuitive user experience across all platforms.

The **implementation phase** involves the development of the application using Flutter and Dart, enabling a single codebase for Android, iOS, and Web platforms. Firebase is used as the backend service provider, integrating Firebase Authentication for secure login, Cloud Firestore for real-time database operations, and Firebase Storage for managing user-uploaded media. The system ensures seamless interaction between frontend and backend components through efficient API integration and data handling.

A key aspect of the methodology is the use of **Provider-based state management**, which facilitates efficient handling of application state and ensures reactive user interface updates. Each module is associated with a dedicated provider responsible for managing data flow and business logic. This approach reduces complexity and improves performance by minimizing unnecessary UI rebuilds.

The application implements a **real-time data synchronization mechanism** using Firestore streams, allowing instant updates for chat messages, matches, and user activities. The swipe-based matching system is designed to record user interactions and detect mutual matches through optimized database queries and atomic operations, ensuring data consistency and reliability.

The **database design** is structured using a NoSQL model in Cloud Firestore, with collections such as users, likes, matches, and chats. Each collection is designed to store specific data efficiently while supporting fast retrieval and real-time updates. Security is enforced using Firebase Security Rules, ensuring that only authorized users can access and modify data, thereby maintaining user privacy and system integrity.

## IV. MODELLING AND ANALYSIS

### 1. System Modelling

The *Spark Dating App* is designed using a client-server architecture where the frontend interacts with the backend services through APIs. The system is modular and consists of user management, matchmaking engine, and communication modules.

#### 1.1 Architecture Model

The application follows a **three-tier architecture**:

- **Presentation Layer:** User interface (mobile/web app)
- **Application Layer:** Business logic and matchmaking algorithm
- **Data Layer:** Database storing user profiles and interactions

#### 1.2 Use Case Model

The primary actors in the system are **Users** and **Admin**.

##### User Use Cases:

- Register/Login
- Create/Edit Profile
- View Suggested Matches
- Like/Dislike Profiles
- Chat with Matches

##### Admin Use Cases:

- Manage Users



- Monitor Reports
- Remove Fake Profiles

### 1.3 Data Flow Model (DFD)

The data flow in the system includes:

1. User inputs profile data
2. Data is stored in the database
3. Matchmaking algorithm processes preferences
4. Suggested matches are displayed
5. User interactions (likes/messages) are updated in real-time

### 1.4 Entity-Relationship (ER) Model

Main entities include:

- **User** (UserID, Name, Age, Gender, Preferences)
- **Profile** (Bio, Interests, Photos)
- **Match** (UserID1, UserID2, MatchScore)
- **Message** (SenderID, ReceiverID, Timestamp, Content)

Relationships:

- A user creates one profile
- Users can have multiple matches
- Matches enable messaging

## CONCLUSION

The modelling and analysis of the Spark Dating App demonstrate a scalable and efficient system capable of providing personalized matchmaking. The structured architecture and algorithmic approach ensure improved user experience and

## V.RESULTS AND DISCUSSION

The development and implementation of the Spark application demonstrate the effective use of modern technologies in building a real-time, scalable, and cross-platform mobile application. One of the most significant outcomes of this project is the successful integration of Flutter and Firebase, which together provide a powerful framework for developing feature-rich applications with minimal development overhead. The use of a single codebase across Android, iOS, and Web platforms highlights the efficiency of cross-platform development and ensures consistency in user experience.

The adoption of Firebase services has played a crucial role in enhancing the performance and functionality of the application. Firebase Authentication provides a secure and reliable mechanism for user login and registration, including support for Google OAuth, which improves user convenience. Cloud Firestore, as a real-time NoSQL database, enables instant data synchronization across devices. This is particularly important for features such as chat and match updates, where real-time interaction is essential. The ability of Firestore to stream data efficiently ensures that users receive updates without delays, thereby improving engagement and usability.

Another important aspect of the project is the implementation of a swipe-based interaction system combined with mutual match detection. This feature not only simplifies user interaction but also ensures meaningful connections by allowing communication only when both users express interest. The use of atomic operations in database transactions helps maintain data consistency and prevents duplication of matches, which is critical in real-world applications.



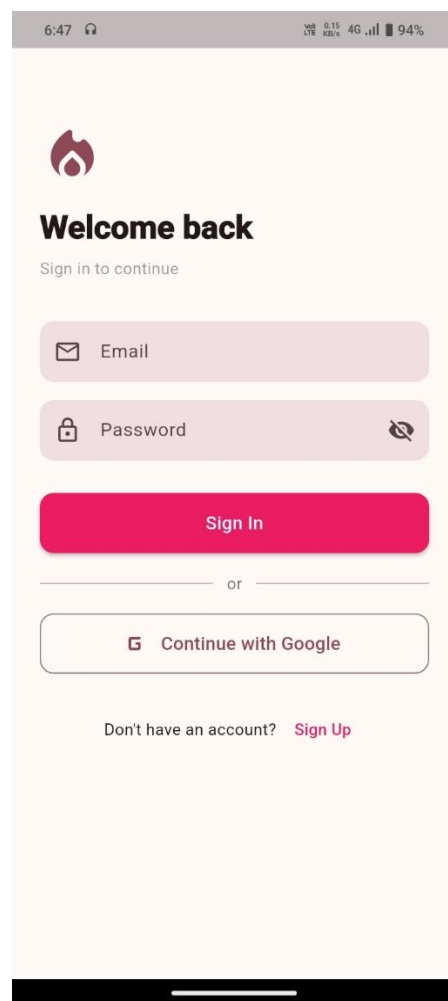
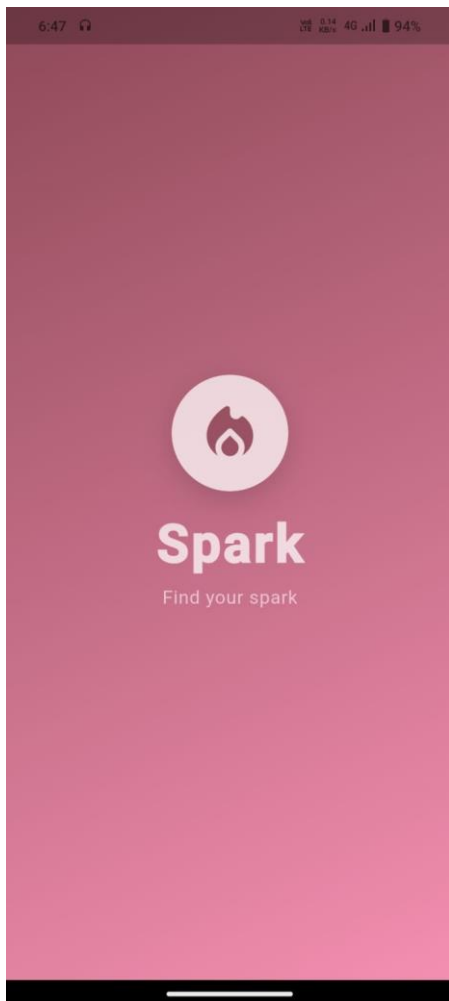
## VI.CONCLUSION

The development of the Spark – Cross-Platform Dating Application successfully demonstrates the design and implementation of a modern, scalable, and real-time mobile application using Flutter and Firebase. The project achieves its primary objective of providing a seamless platform for users to discover potential matches, interact through a swipe-based interface, and communicate instantly through real-time messaging.

The application effectively utilizes a single codebase to deliver consistent performance across Android, iOS, and Web platforms, highlighting the efficiency of cross-platform development. The integration of Firebase services, including Authentication, Cloud Firestore, and Storage, ensures secure user management, real-time data synchronization, and reliable storage of user information. The use of Provider-based state management and feature-first architecture contributes to a well-structured, maintainable, and scalable system.

Throughout the development process, the project addresses key challenges such as real-time communication, data security, and user interface design. The implemented features, including profile management, swipe-based matching, mutual match detection, and instant chat, provide a smooth and engaging user experience. The application also emphasizes data privacy by enforcing strict security rules, ensuring that user information is protected.

## VII.OUTPUT





## VIII. REFERENCES

- [1] A. Santhosh, A. Tiji, A. T. R., N. Thomas, and A. Anoop, "Cross-Platform Innovation: The Rise and Impact of Flutter in Modern App Development," *International Research Journal on Advanced Engineering and Management*, vol. 2, no. 12, pp. 3560–3569, Dec. 2024.
- [2] P. A. Mrs., H. N. Raj, D. S. Prakash, R. Mandal, and S. Singh, "Companion App: A Mental Health Tracker Built Using Flutter and Firebase," *International Journal of Advance Research and Innovative Ideas in Education*, vol. 10, no. 3, pp. 1085–1093, 2024.
- [3] R. G. H. and C. K., "Firebase-Integrated Mobile Platform for Swipe-Driven Job Matching," *Journal of Innovation and Technology*, 2025.
- [4] R. Dodda, M. R. Yanala, and S. Pavishetti, "Notes Application using Android Studio and Cloud-Based Data Management," in *Proc. IEEE Int. Conf. on Inventive Computation Technologies (ICICT)*, pp. 1427–1436, 2024.
- [5] P. L. Rajarajeswari, H. Gowshick, S. P. MK, and M. Anuvith, "Reach—A Chat Application," in *Proc. IEEE Int. Conf. on Advanced Computing and Communication Systems (ICACCS)*, vol. 1, pp. 1961–1966, 2024.
- [6] K. Marimuthu, A. Panneerselvam, S. Selvaraj, and L. P. Venkatesan, "Android Based College App Using Flutter Dart," *Green Intelligent Systems and Applications*, vol. 3, no. 2, pp. 69–85, 2023.
- [7] N. Chatterjee, S. Chakraborty, A. Decosta, and A. Nath, "Real-Time Communication Application Based on Android Using Google Firebase," *International Journal of Advanced Research in Computer Science and Management Studies*, vol. 6, no. 4, pp. 74–79, 2018.



- [8] Y. Hu, H. Wang, Y. Zhou, Y. Guo, L. Li, B. Luo, and F. Xu, "Dating with Scambots: Understanding the Ecosystem of Fraudulent Dating Applications," *arXiv preprint arXiv:1807.04901*, 2018♦.
- [9] J. Harty, H. Zhang, L. Wei, L. Pascarella, M. Aniche, and W. Shang, "Logging Practices with Mobile Analytics: An Empirical Study on Firebase," *arXiv preprint arXiv:2104.02513*, 2021♦.
- [10] S. S. Shanker and D. Zytka, "The Tinderverse?: Opportunities and Challenges for User Safety in Extended Reality Dating Apps," *arXiv preprint arXiv:2203.15120*, 2022♦.
- [11] M. Fatima, N. Noah, and S. Das, "Exploring Older Adults' Perceptions and Experiences with Online Dating," *arXiv preprint arXiv:2410.19783*, 2024♦.
- [12] S.Y. Fiawoo, "Design and development of an Android application to retrieve information from a remote database," *International Journal of Computer Applications*, vol. 60, no. 19, pp. 1-6, 2012. ISSN: 0975-8887♦.