Tushar Kathe, 2026, 14:2
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Paarsh Matrimony System

Author: Tushar Kathe
Guide : Sapna Bhimajiyani
Department -Information Technology
College- Parul University, Vadodara, Gujarat, India

Abstract- Finding a suitable life partner has always been an important but challenging task. Traditional methods of matchmaking are often time-consuming, less accurate, and lack privacy. This research paper presents the Paarsh Matrimony System, a modern web-based application designed to improve the matchmaking process using technology and intelligent features. The system is built using React.js, FastAPI, and MongoDB, ensuring a smooth and responsive user experience. It includes features such as secure user authentication, profile management, AI-based recommendations, real-time chat, and payment integration. The platform focuses on providing accurate match suggestions based on user preferences and behavior. It also ensures data security and user privacy. The results show that the system makes matchmaking faster, easier, and more reliable compared to traditional methods.

Keywords: Paarsh Matrimony System, Matchmaking, Web Application, React.js, FastAPI, MongoDB, User Authentication, Recommendation System, AI-Based Matching, Real-Time Chat, Data Security, Profile Management.

I. INTRODUCTION

With the rapid advancement of digital communication technologies, the way individuals establish connections and relationships has evolved significantly [1]. However, the process of selecting a compatible life partner continues to remain complex, as it involves multiple personal, social, and emotional factors [2]. Conventional matchmaking approaches, including manual searches and community-based references, often lack efficiency and may not consistently deliver accurate outcomes [3].

Several existing online matrimonial platforms provide basic profile browsing features but do not effectively utilize intelligent mechanisms for compatibility assessment [4]. As a result, users are required to manually explore numerous profiles, which increases time consumption and reduces the likelihood of finding suitable matches [5]. Furthermore, concerns related to data privacy, profile authenticity, and limited interaction capabilities negatively impact user trust and system effectiveness [6]. These challenges indicate the necessity for an advanced system capable of delivering precise and personalized matchmaking solutions [7].

The Paarsh Matrimony System is proposed as a technology-driven solution that integrates modern web frameworks with intelligent processing capabilities [8]. The system follows a full-stack architecture, where React.js is used for developing the user interface, FastAPI handles backend operations, and MongoDB is employed for flexible and secure data storage [9]. It offers essential functionalities such as secure authentication, dynamic profile management, and efficient communication features [10].

In addition, the system incorporates AI-based recommendation techniques to evaluate user preferences and behavioral patterns, enabling the generation of accurate and personalized partner suggestions [11]. Features such as real-time messaging, interest-based interactions, and sentiment analysis further enhance user engagement and platform

usability [12]. Strong security measures, including authentication protocols and data validation techniques, are implemented to ensure user privacy and system reliability [13].

An administrative module is also integrated to supervise user activities, verify profiles, and manage reported cases, thereby maintaining platform transparency and integrity [14]. The application is designed with a responsive interface to support accessibility across multiple devices, along with performance optimization strategies to ensure consistent system efficiency [15].

In summary, the Paarsh Matrimony System provides a structured, intelligent, and secure approach to digital matchmaking, offering improved accuracy, reduced effort, and enhanced user experience compared to traditional methods [16].

II. LITERATURE REVIEW

Several matrimony and matchmaking systems have been developed over time; however, many of them primarily rely on basic profile filtering techniques rather than intelligent compatibility analysis [1]. Traditional platforms generally use parameters such as age, location, religion, and profession to suggest matches, which limits the accuracy and relevance of recommendations [2]. These rule-based approaches often fail to capture deeper aspects of compatibility, resulting in lower user satisfaction [3].

Recent studies highlight the effectiveness of Artificial Intelligence (AI) and machine learning techniques in improving recommendation systems by analyzing user behavior, preferences, and interaction patterns [4]. AI-driven systems are capable of learning from user activity and generating personalized suggestions, thereby increasing the probability of successful matchmaking [5]. Despite these advancements, many existing solutions still lack real-time features such as instant messaging, dynamic suggestions, and continuous profile updates [6].

In addition, data management plays a critical role in modern matchmaking platforms. Traditional relational databases may face limitations in handling large volumes of unstructured and dynamic user data [7]. As a result, NoSQL databases such as MongoDB have gained attention due to their flexibility, scalability, and efficient handling of JSON-based data structures [8]. MongoDB enables faster data retrieval and seamless scalability, making it suitable for real-time applications like matrimonial systems [9].

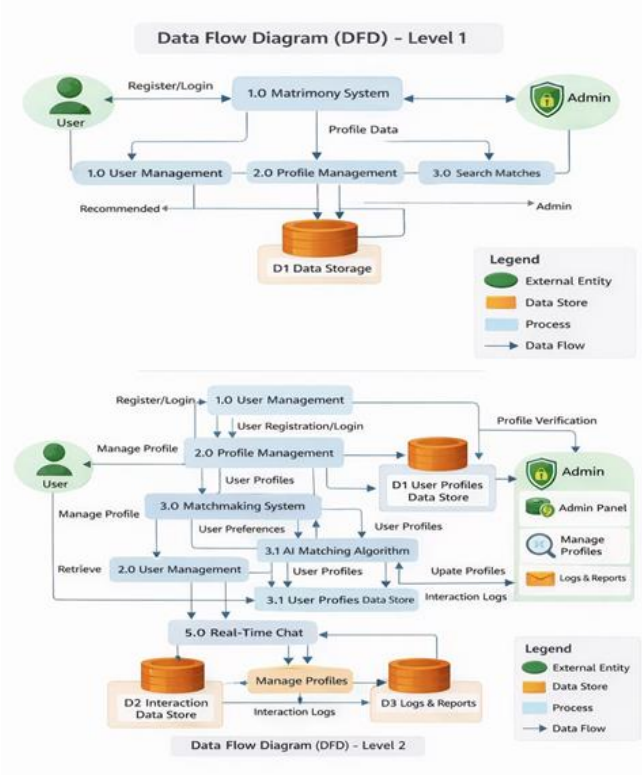
Security and privacy are also major concerns in existing platforms. Several studies emphasize the importance of secure authentication mechanisms, data encryption, and user verification processes to prevent misuse and protect sensitive information [10]. However, not all systems implement strong security measures, which can reduce user trust and platform reliability [11].

Furthermore, user engagement is often limited in traditional systems due to the absence of interactive features and feedback mechanisms [12]. Incorporating functionalities such as real-time chat, sentiment analysis, and AI-based profile evaluation can significantly enhance user experience and system effectiveness [13].

This study aims to address the identified gaps by integrating intelligent matchmaking algorithms, real-time user interaction, and secure data management into a single unified platform [14]. By combining AI-based recommendations, MongoDB for efficient data handling, and modern web technologies, the proposed Paarsh Matrimony System seeks to provide a more accurate, scalable, and user-centric solution compared to existing approaches [15].

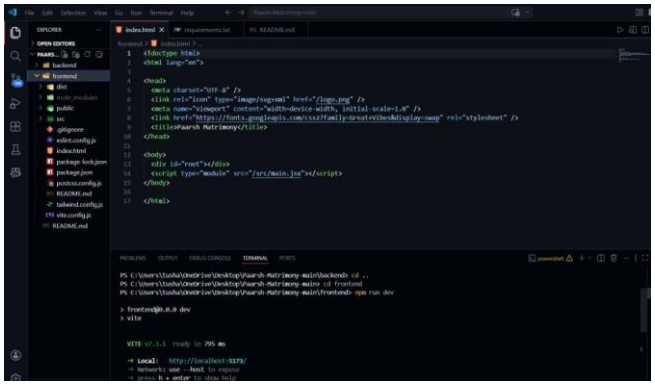
III. METHODOLOGY

The Paarsh Matrimony System is designed using a structured and modular approach that integrates modern web technologies, intelligent algorithms, and secure data handling techniques. The methodology focuses on improving matchmaking accuracy, enhancing user interaction, and ensuring system reliability. The
overall system follows a client–server architecture and is divided into multiple functional modules as described below.



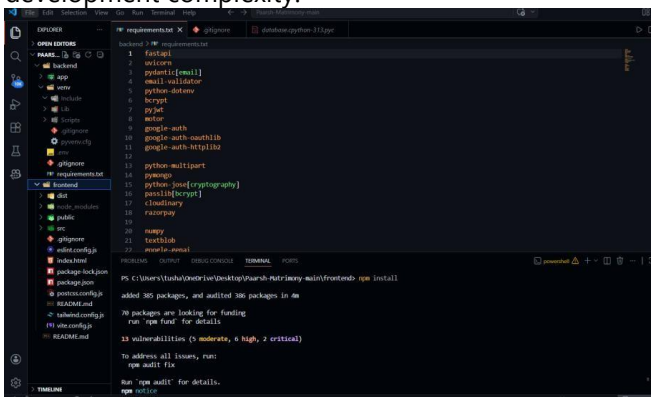
Frontend Development Framework

The frontend of the system is implemented using React.js, which enables the creation of a dynamic and interactive user interface. It follows a component-based architecture, allowing reusable UI components such as login forms, profile cards, dashboards, and chat interfaces.



Backend Development (FastAPI)

The backend is developed using FastAPI, a lightweight and high-performance web framework used for handling API requests, business logic, and communication between the frontend and database. FastAPI is based on asynchronous programming, which allows the system to handle multiple requests simultaneously without blocking execution. This improves system efficiency and scalability. It also provides automatic API documentation and validation, ensuring structured data exchange and reducing development complexity.



Database Layer (MongoDB)

The system uses MongoDB as its primary database for storing user profiles, interaction data, and system-related information. MongoDB follows a NoSQL, document-oriented approach where data is stored in JSON-like format. MongoDB offers high flexibility by allowing dynamic schema design, making it

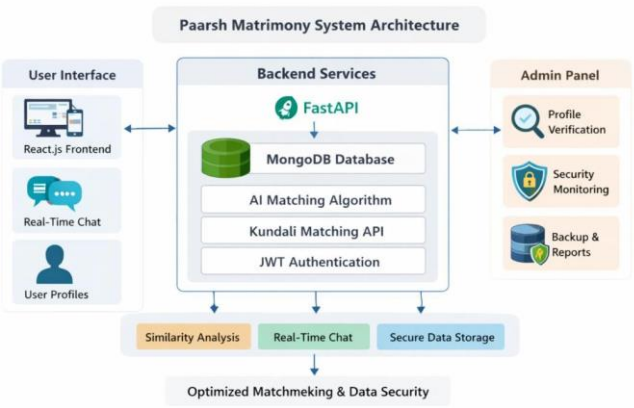
suitable for applications with varying data structures such as matrimonial profiles. It supports horizontal scaling and efficient querying, which enhances system performance when handling large volumes of user data. Its ability to store nested and complex data structures makes it ideal for managing user preferences and interaction history.

IV. ARCHITECTURE

System Architecture

The system follows a three-tier architecture consisting of the frontend, backend, and database layers. The frontend is developed using React.js, which provides a responsive and interactive user interface. The backend is implemented using FastAPI, which handles API requests, business logic, and communication between components. MongoDB is used as the database due to its flexibility in handling unstructured and semi-structured data.

A three-tier architecture improves scalability and maintainability by separating presentation, logic, and data layers. MongoDB, being a NoSQL database, stores data in JSON-like documents, which makes it suitable for dynamic user profiles and real-time updates. This architecture allows efficient data flow and supports high-performance applications.



User Authentication and Security

The system implements secure authentication using JSON Web Tokens (JWT). Users must register and log in to access system features. Passwords are securely stored using hashing techniques, and session management is handled through token-based authentication.

JWT authentication ensures stateless and secure communication between client and server. It reduces server load and prevents unauthorized access. Data validation and encryption techniques further enhance user privacy and protect sensitive information from security threats.

Profile Management System

Users can create, update, and manage their profiles by providing personal, educational, and preference-related details. The system validates user input and stores the data in MongoDB for efficient retrieval and updates.

Profile management plays a crucial role in matchmaking systems as it forms the basis for recommendation algorithms. Using MongoDB allows flexible schema design, enabling the system to handle diverse user attributes without strict structure limitations.

Matchmaking Algorithm (Similarity-Based Matching)

The system uses a similarity-based matching algorithm to recommend potential partners. It compares user profiles based on attributes such as age, interests, preferences, and background. Similarity-based algorithms calculate the degree of match between users using weighted parameters. This approach improves recommendation accuracy by considering multiple factors instead of simple filters. It helps in generating personalized and relevant match suggestions.

AI-Based Recommendation System

The system incorporates AI techniques to analyze user behavior, preferences, and interaction patterns. Based on this analysis, it generates intelligent recommendations that improve over time.

AI-based recommendation systems use data-driven approaches to learn from user activity. Techniques such as pattern recognition and behavioral analysis enable the system to provide more accurate and adaptive suggestions, enhancing user satisfaction.

Real-Time Chat System

The platform provides real-time communication between matched users through an integrated chat system. This allows users to interact directly and build connections.

Real-time communication improves user engagement and decision-making. Technologies such as asynchronous APIs and event-driven communication ensure instant message delivery and smooth interaction between users.

Watchlist and Interaction Module

Users can save profiles, send interests, and manage their connections through the watchlist feature.

This module enhances user experience by allowing users to track potential matches and revisit profiles. It supports decision-making by providing easy access to shortlisted candidates.

Admin Panel and Monitoring

The system includes an admin panel for managing users, verifying profiles, and handling reported issues. It ensures the platform remains safe and reliable. Administrative control is essential for maintaining system integrity. Monitoring tools help in detecting suspicious activities, ensuring data authenticity, and improving overall system trustworthiness.

Data Storage and Management

All user data, chat records, and system information are stored in MongoDB. MongoDB uses a document-oriented model, which allows flexible and scalable data storage. It supports high-speed read and write operations, making it suitable for real-time applications like matchmaking systems. Its scalability ensures that the system can handle increasing user data efficiently.

Performance Optimization

The system implements efficient data handling, API optimization, and memory management techniques to ensure smooth performance.Performance optimization reduces latency and improves system responsiveness. Techniques such as caching, efficient query handling, and asynchronous processing help maintain system stability under high user load.

V. RESULTS AND DISCUSSION

The Paarsh Matrimony System improves the matchmaking process by making it faster and more efficient. Users receive better match suggestions based on their preferences, reducing the time spent searching for profiles.

The real-time chat feature enhances communication between users, while the secure authentication system ensures data safety. Overall, the system provides a better experience compared to traditional matrimony methods.

VI. CONCLUSION

The Paarsh Matrimony System presents a modern and efficient approach to digital matchmaking by integrating advanced web technologies with intelligent recommendation techniques. The system successfully addresses the limitations of traditional matchmaking methods by providing a secure, scalable, and user-friendly platform. Features such as AI-based matching,

real-time communication, and dynamic profile management enhance the overall user experience and improve the accuracy of partner suggestions.

The use of React.js, FastAPI, and MongoDB ensures high performance, flexibility, and efficient data handling. Security mechanisms such as JWT-based authentication and data validation contribute to maintaining user privacy and system reliability. Based on the implementation and testing, the system demonstrates improved efficiency, reduced user effort, and better matchmaking outcomes. Overall, the proposed system offers a practical and technology-driven solution for modern matrimonial services.

VII. FUTURE SCOPE

The Paarsh Matrimony System can be further enhanced by incorporating more advanced technologies and features. Future improvements may include:

- Integration of advanced machine learning models for more accurate and predictive matchmaking
- Implementation of voice-based interaction and chatbot assistants for better accessibility
- Use of blockchain technology to enhance data security and profile authenticity
- Development of a mobile application for improved accessibility and user convenience
- Integration of video calling features for real-time communication between users
- Enhanced analytics dashboard for admins to monitor user behavior and system performance
- Multi-language support to expand usability across different regions

These enhancements can make the system more intelligent, secure, and widely accessible.

Acknowledgement

The authors would like to express their sincere gratitude to all those who contributed to the successful development of this project. Special thanks to the project guide and faculty members for their valuable guidance, support, and encouragement throughout the research and development process.
We also acknowledge the support of our institution for providing the necessary resources and environment to complete this work. Additionally, we are grateful to our peers and team members for their cooperation and contributions in various stages of the project.

REFERENCES

1. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.
2. F. Chollet, Deep Learning with Python. Manning Publications, 2017.
3. MongoDB Inc., "MongoDB Documentation," [Online]. Available: https://www.mongodb.com/docs/
4. S. Tilkov and S. Vinoski, "Node.js: Using JavaScript to Build High-Performance Network Programs,"
5. IEEE Internet Computing, vol. 14, no. 6, pp. 80–83, 2010.
6. FastAPI, "FastAPI Documentation," [Online]. Available: https://fastapi.tiangolo.com/
7. React, "React Documentation," [Online]. Available: https://react.dev/
8. J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques. Elsevier, 2011.
9. Ricci, L. Rokach, and B. Shapira, Recommender Systems Handbook. Springer, 2015.
10. N. Z. Gong and D. Song, "Privacy-Preserving Recommender Systems," IEEE Security & Privacy, vol. 13,no.5,pp.20–27,2015.
11. K. Jain, "Online Matrimonial Systems and Their Impact," International Journal of Computer Applications, vol. 182, no. 45, 2019.