

Paarsh Matrimonial Website System with Intelligent Matchmaking

Nikhil Rajput, Rahul Patel

A Information Technology Student, India

Abstract- This paper presents a web-based matrimonial website system designed for efficient partner matchmaking using structured datasets and user preference analysis. The system is implemented using Java Spring Boot, REST APIs, and MySQL database. It enables user registration, profile management, match searching, and secure communication. The study also includes system architecture, data flow diagrams, dataset usage, and research gaps. The proposed system improves matchmaking accuracy and ensures scalability and security.

Keywords: Matrimonial Website, Spring Boot, MySQL, Dataset, Matchmaking, Full Stack.

I. INTRODUCTION

Matrimonial websites are widely used platforms that help individuals find suitable life partners through digital systems. Traditional matchmaking methods are slow, limited to social connections, and lack scalability. These methods often depend on family networks and manual communication, which reduces efficiency.

With the advancement of internet technologies, matrimonial platforms provide a faster and more efficient solution. Users can create profiles, define preferences, and search for compatible partners easily. These systems reduce manual effort and improve accessibility.

However, existing systems still rely heavily on static filtering techniques and do not utilize intelligent data processing. This research focuses on developing a web-based matrimonial system using structured datasets and backend technologies to improve performance, scalability, and user experience.

II. LITERATURE REVIEW

Existing matrimonial platforms such as Shaadi.com and Jeevansathi use profile-based filtering methods. These platforms allow users to search based on attributes like age, location, education, and religion. However, research studies show that these systems

have several limitations. They do not analyze user behavior, interaction patterns, or emotional compatibility. As a result, matchmaking accuracy is limited.

Recent research in recommendation systems suggests that data-driven approaches can improve matching quality. However, implementation of such techniques in matrimonial systems is still limited.

III. DATASET USED

The system uses structured user profile datasets stored in MySQL database for matchmaking.

Dataset attributes include:

- Personal Details: Name, Age, Gender
- Social Details: Religion, Caste, Community
- Location Details: City, State, Country
- Professional Details: Education, Occupation
- Preferences: Partner age range, location, education

Dataset Sources:

- Kaggle Matrimonial Dataset
- User-generated profile data

The dataset is used for filtering, matching, and recommendation. Structured storage of data improves query performance and matching accuracy.

IV. METHODOLOGY

The system follows a modular full-stack development approach. Each module is developed independently and integrated using REST APIs.

- User Registration and Authentication
- Profile Creation and Management
- Matchmaking System
- Secure Messaging Module
- Admin Verification System

V. SYSTEM ARCHITECTURE

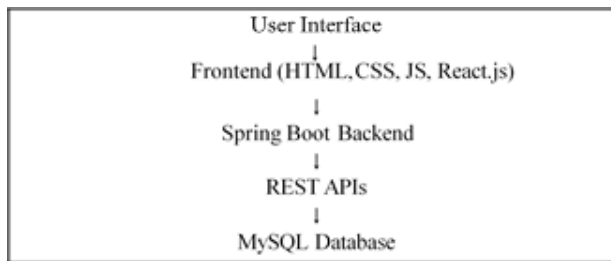


Figure 1. System Architecture

5.1. Frontend Development

The frontend of the system is implemented using HTML, CSS, Bootstrap, and JavaScript. It provides a user-friendly interface for registration, profile management, and matchmaking. The design ensures responsiveness and ease of use.

5.2. Backend Development

The backend is developed using Java Spring Boot. It handles business logic, API requests, and communication between frontend and database. REST APIs are used for efficient data exchange and system integration.

This paper presents a detailed matrimonial website system using Java Spring Boot, REST APIs, MySQL, and dataset-based matchmaking techniques.

VI. ENTITY RELATIONSHIP DIAGRAM

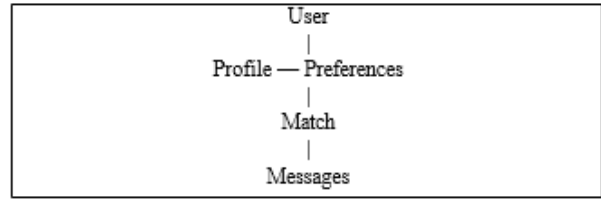


Figure 2. ER Diagram

The ER diagram shows relationships between users, profiles, matches, and messages.

VII. DATA FLOW DIAGRAM

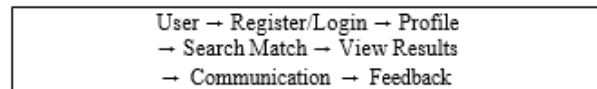


Figure 3. DFD Level 0

This diagram represents the flow of data within the system from user input to matchmaking and communication.

VIII. TECHNOLOGY USED

- Front end: HTML, CSS, Bootstrap, JavaScript, React.js
- Backend: Java (Spring Boot)
- Database: MySQL
- API: REST API
- Concepts: OOP (Encapsulation, Inheritance, Polymorphism, Abstraction)

IX. DETAILED WORKING

The system begins with user registration and authentication. Users create profiles by entering personal details and preferences. These details are stored securely in the MySQL database.

When a user searches for matches, the system applies filtering logic based on dataset attributes such as age, location, and preferences. The backend processes the request and retrieves matching profiles.

The results are displayed on the frontend interface. Users can communicate through the messaging system. The admin module monitors activities and removes fake profiles to ensure system reliability.

X. ADVANTAGES OF PROPOSED SYSTEM

- Fast and efficient matchmaking
- Secure data handling
- User-friendly interface
- Scalable architecture
- Easy profile management

XI. RESEARCH GAPS

- Lack of intelligent AI-based matchmaking
- Weak fake profile detection systems
- Limited use of behavioral data
- Privacy and security concerns
- No real-time compatibility scoring

XII. RESULTS

The developed system provides improved performance compared to traditional methods. It allows users to search for matches quickly and communicate securely. The system ensures efficient data handling and scalability.

XIII. CONCLUSION

The matrimonial website system developed using Spring Boot and MySQL provides an efficient and scalable solution for partner match-making. The system successfully integrates frontend, backend, and database components to deliver a smooth and secure user experience. By using structured datasets, the system improves matchmaking accuracy compared to traditional methods. It allows users to create profiles, search for suitable matches, and communicate effectively.

The use of REST APIs ensures efficient data exchange and system performance.

Furthermore, the system highlights the importance of digital transformation in matchmaking services.

It reduces manual effort, improves accessibility, and enhances user satisfaction. Overall, the proposed system demonstrates how modern full-stack technologies can be applied to solve real-world problems efficiently.

XIV. FUTURE WORK

The proposed system can be further enhanced by incorporating advanced technologies and intelligent algorithms. Future improvements include:

- Implementation of machine learning algorithms for better compatibility prediction
- AI-based recommendation system using user behavior and interaction data
- Advanced fraud detection and profile verification techniques
- Real-time matchmaking system for faster results
- Integration of mobile applications for improved accessibility
- Use of cloud computing for scalability and performance optimization

XV. ACKNOWLEDGEMENT

The author would like to express sincere gratitude to the project guide and faculty members for their valuable guidance, support, and encouragement throughout the development of this project.

Special thanks to the institution for providing the necessary resources and environment to successfully complete this work. The author also appreciates the support of peers and friends who contributed ideas and assistance during various stages of the project development. This project would not have been possible without the continuous motivation and guidance received during the research and implementation phases.

REFERENCES

1. I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.

2. J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Tech-niques. Elsevier, 2011.
3. Spring Boot Documentation, Available: <https://spring.io/projects/spring-boot>
4. MySQL Documentation, Available: <https://dev.mysql.com/doc/>
5. REST API Design Guide, Available: <https://restfulapi.net/>
6. A. Ricci, L. Rokach, and B. Shapira, Recommender Systems Hand-book. Springer, 2015.
7. Kaggle Dataset: Matrimonial Profiles Dataset

Screen Shots



1.1 Login Page



1.2 Home Dashboard