Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 7, July : 2025

SOULYN SHAADISAGA: A REAL–TIME MATCHMAKING PLATFORM

Soumya Ranjan Patel, 4th Year, Department of CSE, Gandhi Institute for Technology, BPUT, India soumya.patel2022@gift.edu.in
Soumya Ranjan Behera 4th Year, Department of CSE, Gandhi Institute for Technology, BPUT, India soumyarb2022@gift.edu.in
Subhashree Sukla, Assistant Professor, Department of CSE, Gandhi Institute for Technology, BPUT, India

Abstract–
SOULYN ShaadiSaga is a full-stack matchmaking platform developed to connect individuals seeking life partners through a secure, scalable, and user-centric system. Leveraging modern technologies such as Spring Boot (Java) for backend services and React JS for responsive frontend interfaces, the application enables users to register, authenticate securely, browse and interact with potential matches, and manage their profile data. This system also includes administrative control for user verification and moderation. With layered architecture and real-time data flow, the application ensures robust performance, data security, and user satisfaction.

Keywords:
Spring Boot, React JS, Full–Stack Java, MySQL

I. INTRODUCTION

Introducing our **SOULYN ShaadiSaga**, which aims to address the cultural and technical challenges of online matchmaking by providing a responsive, efficient, and reliable platform. The system allows users to create and manage detailed profiles, browse compatible matches, and interact through a protected environment. Built with scalable and maintainable architecture, the platform emphasizes personalized user experience, efficient matchmaking algorithms, and admin-level moderation tools.

II. LITERATURE REVIEW

The literature review focuses on the analysis of existing matchmaking systems, social networking apps, and secure authentication methods. Research explores the evolution of online matrimony platforms, the importance of privacy in user data, and interface usability. Studies on **microservice architecture**, **token-based authentication (JWT)**, and **user behavior analytics** provide valuable insights. Comparisons with legacy matrimonial systems highlight the need for real-time features, better recommendation engines, and mobile-first interfaces.

III. SYSTEM DESIGN

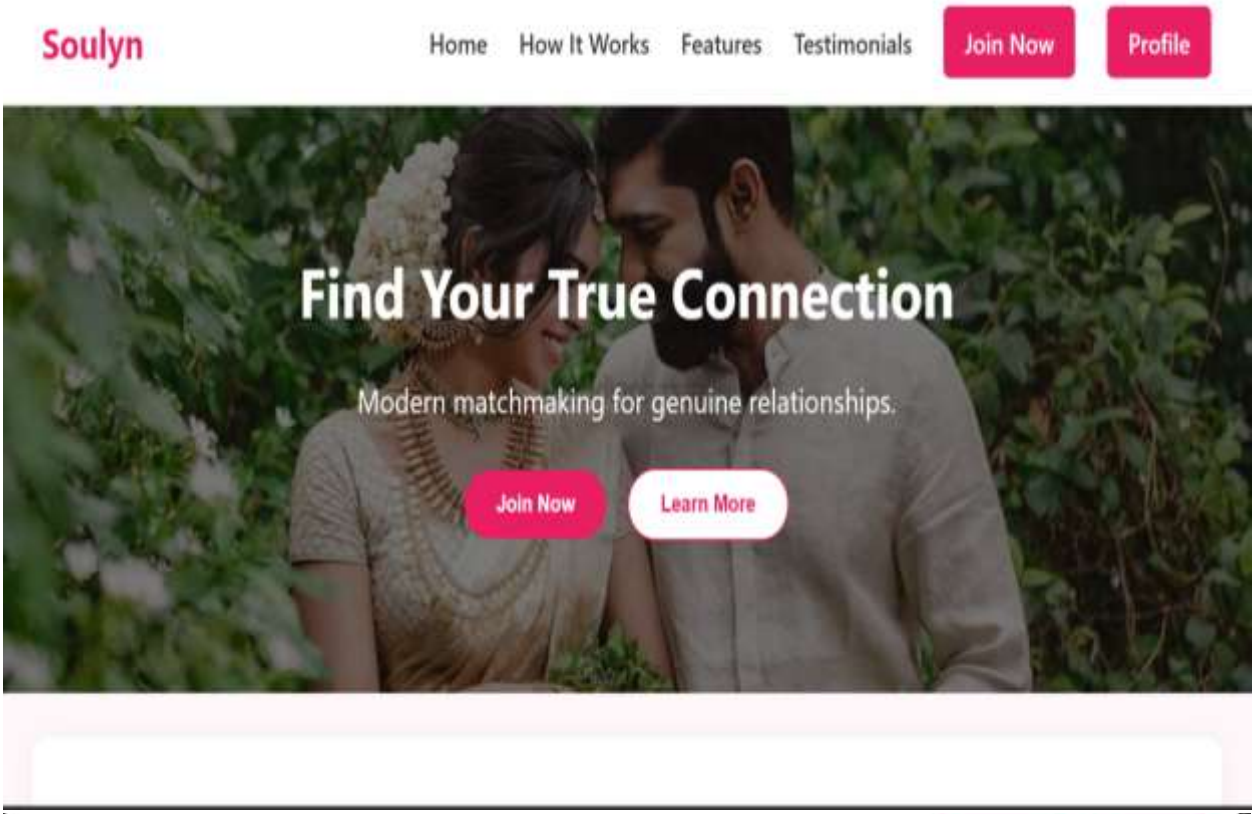
The system is designed using a layered architecture that clearly separates responsibilities across different components. The frontend, built with React JS, manages user registration, login, match browsing, and profile management through dynamic routing and reusable components. The backend, implemented using Spring Boot, handles API requests, data validation, security protocols, and core business logic. Custom JWT-based authentication ensures secure and reliable login and registration processes. The database layer, utilizing either MySQL or PostgreSQL, stores essential data such as user profiles, match records, and administrative metadata. An integrated admin panel supports content moderation, user management, and analytics tracking. This design promotes modularity, scalability, and ease of deployment across various cloud platforms.

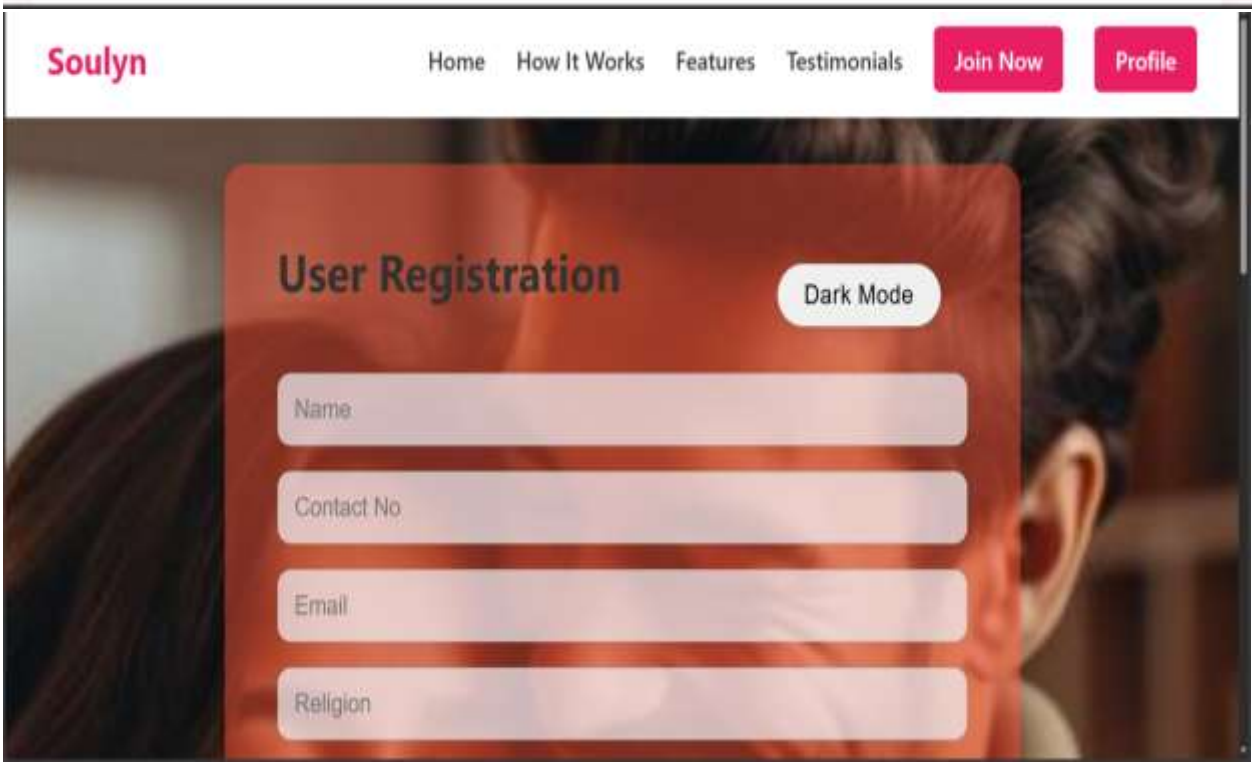
IV. IMPLEMENTATION

The system is designed using a layered architecture that clearly separates responsibilities across different components. The frontend, built with React JS, manages user registration, login, match

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 7, July : 2025

browsing, and profile management through dynamic routing and reusable components. The backend, implemented using Spring Boot, handles API requests, data validation, security protocols, and core business logic. Custom JWT-based authentication ensures secure and reliable login and registration processes. The database layer, utilizing either MySQL or PostgreSQL, stores essential data such as user profiles, match records, and administrative metadata. An integrated admin panel supports content moderation, user management, and analytics tracking. This design promotes modularity, scalability, and ease of deployment across various cloud platforms.





V. RESULTS

The SOULYN platform successfully demonstrated end-to-end functionality from user registration to match interaction. Admins could moderate users in real-time, and authenticated users could securely explore matches. The responsive UI ensured usability on mobile and desktop devices. Performance testing showed stable throughput under concurrent user loads, while security testing validated the robustness of JWT authentication and data access restrictions.

VI. CONCLUSION

In conclusion, the development of the **SOULYN ShaadiSaga** matchmaking platform demonstrates a modern approach to solving real-world social challenges using full-stack technologies. The system combines a secure backend with a responsive frontend to offer personalized matchmaking features and smooth user interaction. With support for secure login, profile management, and admin-level control, it ensures both functionality and safety.

The project's layered architecture promotes scalability and maintainability, making it suitable for long-term use and future upgrades. Moving forward, enhancements such as machine learning-based match suggestions and integrated chat features could further improve the platform’s effectiveness and user engagement.

ACKNOWLEDGEMENT

We sincerely thank our project guide, Prof Subhashree Sukla, for their unwavering support, expert guidance, and constructive feedback throughout the development of SOULYN ShaadiSaga. We also extend our appreciation to the faculty of the Department of Computer Science and Engineering for providing the necessary academic and technical support. Our heartfelt thanks go to our peers and testers whose feedback helped refine the platform's features and usability. We gratefully acknowledge the open-source communities and documentation platforms that supported our technical implementation, as well as our families and friends for their constant encouragement and moral support during this project.

REFERENCES

- https://spring.io/projects/spring-boot
- https://react.dev/
- https://jwt.io/introduction
- https://axios-http.com/
- https://developer.mozilla.org/en-US/
- https://dev.mysql.com/doc/workbench/en/