

PRESENT-AI : AI-POWERED GIFT IDEA GENERATOR

¹ J. R. Arun Kumar ² Mayank ³ Mansi Gothwal ⁴ Akshita Sharma ⁵ Mohit Bambhani

¹ Professor, Department of CSE, Modern Institute of Technology and Research Centre, Rajasthan, India

^{2,3,4,5} UG Students, Department of CSE, Modern Institute of Technology and Research Centre, Rajasthan, India

Article Information

Received : Mar 27 2026
Revised : Mar 27 2026
Accepted : Mar 28 2026
Published : Mar 29 2026

Corresponding Author:

Mayank

Abstract— Choosing the right gift has become difficult today due to too many options and varying personal preferences. PresentAI simplifies this process by providing personalized gift suggestions based on factors like occasion, recipient, and budget. Instead of relying on complex AI models, it uses a structured, rule-based approach to filter and rank options, ensuring the results are relevant, diverse, and easy to understand. The platform also supports group gifting by allowing users to discuss, shortlist, and vote on ideas together. Built using React and Flask, PresentAI focuses on reliability, low cost, and clear explanations, making the overall gifting experience more practical and user-friendly..

Keywords: Gift Recommendation; Personalization; Rule-Based System; Group Gifting; Decision Support; Explainability

Copyright © 2026: J. R. Arun Kumar, Mayank, Mansi Gothwal, Akshita Sharma, Mohit Bambhani. This is an open access distribution, and reproduction in any medium, provided Access article distributed under the Creative Commons Attribution License the original work is properly cited License, which permits unrestricted use.

Citation: J. R. Arun Kumar, Mayank, Mansi Gothwal, Akshita Sharma, Mohit Bambhani. “PRESENT-AI: AI-POWERED GIFT IDEA GENERATOR”, Journal of Science, Computing and Engineering Research, 9(3), March 2026.

I. INTRODUCTION

In today's digital world, selecting a suitable gift has become more difficult than ever due to the large number of options available and the diversity in personal preferences. While online platforms provide easy access to products, they often fail to guide users in choosing gifts that are meaningful and appropriate. As a result, people spend a lot of time browsing without making confident decisions. This problem becomes even more complex when the recipient's preferences are not clearly known or when multiple people are involved in the decision-making process.

PresentAI is designed to simplify this process by providing a structured and intelligent gift recommendation system. It takes user inputs such as occasion, recipient details, interests, and budget, and converts them into personalized suggestions. Unlike many modern systems that rely heavily on complex AI models, PresentAI follows a deterministic, rule-based approach that ensures consistent and explainable results. The system not only helps individuals choose better gifts but also supports collaborative decision-making through group interaction features.

A. Overview of Gift Recommendation Systems

A gift recommendation system is designed to assist users in selecting suitable gifts by considering multiple factors such as occasion, recipient details, age group, preferences, and budget. Earlier methods were mostly manual, where users depended on their own judgment or advice from others, which often lacked consistency and personalization. Even many existing digital platforms provide only basic or trending suggestions that may not fit every situation. Modern systems aim to overcome these

limitations by using structured data and logical filtering techniques to generate more relevant recommendations. PresentAI follows a rule-based approach, ensuring that the suggestions are not only accurate but also easy to understand and explain, making the system more reliable and user-friendly.

B. Importance and Applications

Gift recommendation systems play an important role in simplifying decision-making for various occasions such as birthdays, anniversaries, festivals, and special events. They help users save time, reduce confusion, and make more thoughtful choices without feeling overwhelmed. In the context of e-commerce, such systems improve user experience by guiding customers toward products that are more likely to be appreciated. Additionally, features like group gifting make the process more collaborative, allowing friends or family members to discuss and finalize a gift together. As online shopping continues to grow, the importance of intelligent recommendation systems is increasing, making them a valuable tool for both users and businesses.

II. PROBLEM STATEMENT

In today's digital environment, selecting an appropriate gift has become a complex and time-consuming task. The availability of a large number of products often leads to confusion rather than convenience. Users are required to consider multiple factors such as personal preferences, occasion requirements, and budget limitations, which makes decision-making difficult. Existing platforms typically provide generic or trend-based suggestions that do not fully match user needs.

Traditional gift selection methods rely heavily on personal judgment and manual searching, which are inefficient and inconsistent. Users often experience decision fatigue due to too many choices and lack of proper guidance. In group gifting scenarios, the challenge increases as multiple individuals need to coordinate and agree on a single option without having an effective system for communication.

Another important issue is the lack of transparency in recommendation systems. Most platforms do not explain why a particular gift is suggested, which reduces user trust. Therefore, there is a need for a system that can provide personalized, explainable, and reliable recommendations while also supporting collaborative decision-making in a structured manner.

III. RELATED WORK

Gift recommendation has been an important area of research in the field of recommender systems and e-commerce applications. Traditional recommendation approaches such as collaborative filtering and content-based filtering have been widely used to suggest products based on user behavior or item attributes. While these techniques have shown effectiveness in general product recommendation, they often face limitations in gifting scenarios where decisions are context-specific and may not rely on past user interactions.

With the advancement of artificial intelligence, many modern systems have adopted machine learning and deep learning techniques to generate recommendations. These AI-based systems can process large amounts of data and provide flexible suggestions; however, they often suffer from issues such as lack of explainability, inconsistency in results, and dependency on external APIs or cloud-based services.

Rule-based recommendation systems provide an alternative approach by using predefined logic and structured datasets. These systems offer better control, transparency, and reproducibility compared to purely AI-driven models. However, most existing systems do not integrate rule-based recommendations with collaborative features such as group discussions and voting mechanisms.

PresentAI addresses these limitations by combining a deterministic rule-based recommendation engine with collaborative decision-making tools. It provides a balanced approach that ensures accuracy, explainability, and user engagement within a single platform.

IV. SYSTEM ARCHITECTURE

A. User Interaction Layer:

The User Interaction Layer is implemented using a React-based web application that provides an intuitive and responsive interface. Users can enter gifting details, view recommendations, manage shortlisted items, and participate in group discussions. The interface is designed to be simple and user-friendly, ensuring that even non-technical users can easily navigate the system.

B. Recommendation Processing Layer

The Recommendation Processing Layer is responsible for generating gift suggestions. It uses a rule-based engine that processes structured user inputs and applies multiple filtering stages, including occasion matching, interest alignment, age suitability, and budget constraints. A scoring mechanism is used to rank the filtered items, and the final results are categorized into different groups such as safe, practical, creative, and fun.

C. Backend Service Layer

The Backend Service Layer is developed using Flask and handles all server-side operations, including API requests, authentication, and data processing. JWT-based authentication ensures secure access to system features. Real-time communication for group discussions and voting is supported using WebSocket-based technologies.

D. Data Storage Layer

The Data Storage Layer consists of a structured dataset of gift items organized based on categories such as occasion, interests, and budget. SQLite is used for storing user-related data and system records, while browser storage mechanisms are used to maintain temporary session data such as shortlisted items and feedback.

V. PROPOSED MODEL

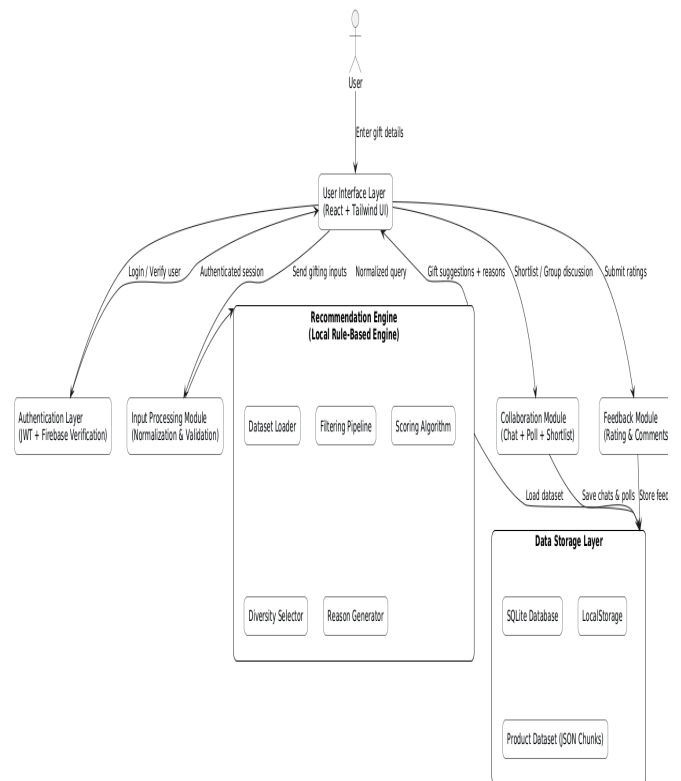


Fig.1: Proposed Model Diagram

The proposed model of PresentAI is a deterministic and structured gift recommendation system that transforms user inputs into personalized and explainable suggestions. The system follows a layered architecture where different modules handle user interaction, input processing, recommendation generation, collaboration, and data storage. This design ensures consistency, transparency, and efficient decision-making for both individual and group gifting scenarios.

VI. TECHNOLOGY STACK

The PresentAI system is developed using a modern full-stack technology stack that ensures scalability, performance, and ease of development. The selected technologies support efficient user interaction, secure backend processing, real-time collaboration, and reliable data management.

A. Frontend Technologies

The frontend of the system is built using React.js, which enables the development of a dynamic and responsive user interface. It allows smooth navigation between different components such as input forms, recommendation pages, shortlist views, and chat interfaces. Tailwind CSS is used for styling, providing a clean and consistent design with minimal effort. JavaScript is used for handling client-side logic, API calls, and dynamic updates.

B. Backend Technologies

The backend is implemented using Flask, a lightweight Python-based web framework that supports RESTful API development. It handles user requests, processes recommendation logic, and manages communication between different system components. Flask is chosen for its simplicity, flexibility, and efficient performance in handling structured workflows.

C. Authentication and Security

The system uses JWT (JSON Web Token) for secure authentication and session management. It ensures that only authorized users can access protected features. Additionally, Firebase Authentication is integrated to support social login and simplify user identity verification.

D. Real-Time Communication

To support collaborative features such as group chat and poll voting, Flask-SocketIO is used for real-time communication. It enables instant data exchange between users, ensuring smooth interaction in group gifting scenarios.

E. Database and Storage

SQLite is used as the primary database for storing user data, group information, chat records, and system-related data. For client-side persistence, LocalStorage and SessionStorage are used to maintain temporary data such as shortlisted items and session states. The product dataset is stored in structured JSON format, which is used by the recommendation engine.

F. Recommendation Engine Technologies

The core recommendation engine is implemented in Python using a rule-based approach. It processes structured data, applies filtering and scoring logic, and generates personalized suggestions. This approach ensures deterministic outputs, low operational cost, and explainable results.

G. Deployment and Integration

The frontend can be deployed using platforms like Vercel or Netlify, while the backend can be hosted on services such as Render or Railway. The system is designed to integrate easily with external APIs or e-commerce platforms for future enhancements.

VII. RESULT SCREENSHOTS

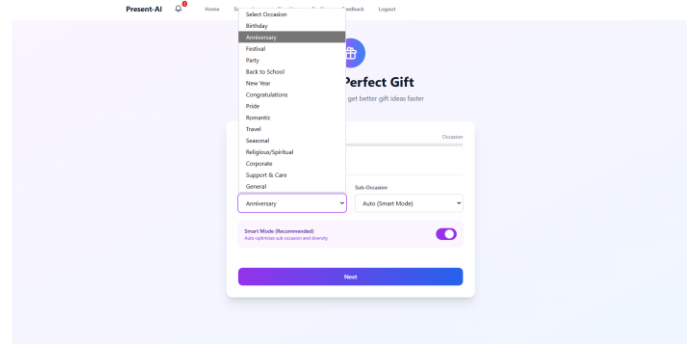


Fig. 2: Home Page

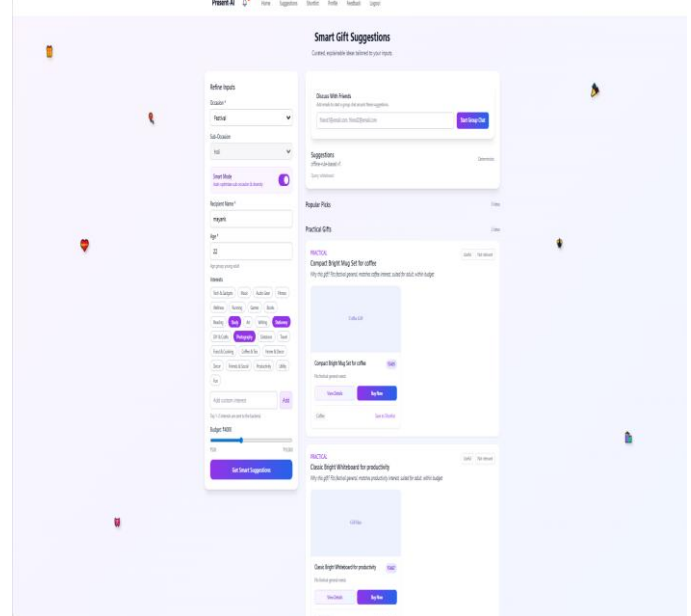


Fig. 3: Suggestions Page

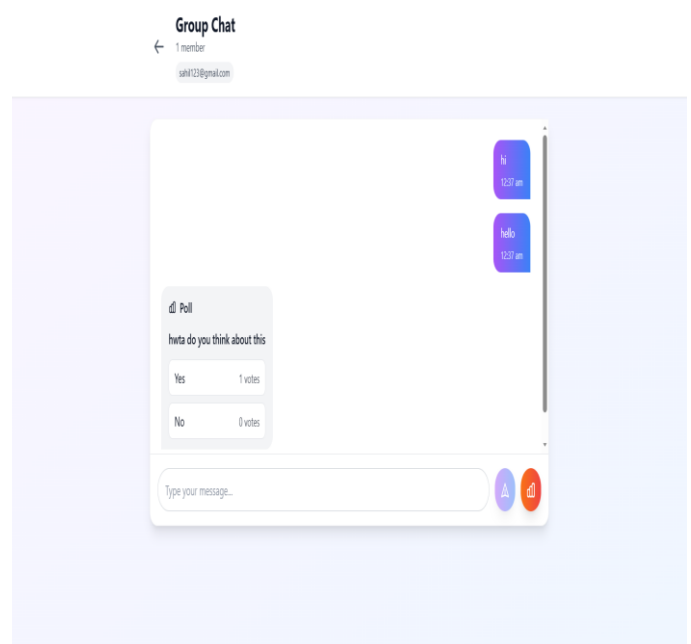


Fig. 4: Group Chatroom

VIII. CONCLUSIONS

PresentAI successfully demonstrates how a structured and rule-based approach can simplify the complex task of gift selection. The system effectively transforms user inputs such as occasion, recipient details, interests, and budget into personalized and meaningful gift suggestions. By using a deterministic recommendation engine, the system ensures consistent and reliable outputs, avoiding the unpredictability often associated with AI-based models.

One of the key strengths of the system is its explainability, where each recommendation is supported by a clear reason. This improves user trust and helps users make confident decisions. Additionally, the integration of collaborative features such as group chat, shortlisting, and poll-based voting makes the system more practical for real-world use, especially in group gifting scenarios.

The implementation of a full-stack architecture using React, Flask, and real-time communication technologies ensures smooth performance and scalability. The system also maintains a balance between simplicity and functionality by avoiding heavy dependency on external AI services while still delivering accurate and relevant results.

In conclusion, PresentAI provides a reliable, user-friendly, and cost-effective solution for modern gift recommendation challenges. It highlights that a well-designed rule-based system can deliver efficient and explainable results while enhancing the overall gifting experience.

REFERENCES

- [1] F. Ricci, L. Rokach, and B. Shapira, *Recommender Systems Handbook*, 2nd ed. New York, NY, USA: Springer, 2015.
- [2] C. C. Aggarwal, *Recommender Systems: The Textbook*. Cham, Switzerland: Springer, 2016.
- [3] M. Grinberg, *Flask Web Development: Developing Web Applications with Python*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2018.
- [4] A. Banks and E. Porcello, *Learning React: Modern Patterns for Developing React Apps*. Sebastopol, CA, USA: O'Reilly Media, 2020.
- [5] L. Ramalho, *Fluent Python*. Sebastopol, CA, USA: O'Reilly Media, 2015.
- [6] D. Hardt, "The OAuth 2.0 Authorization Framework," *Internet Engineering Task Force (IETF)*, 2012.
- [7] R. T. Fielding, "Architectural Styles and the Design of Network-Based Software Architectures," Doctoral dissertation, University of California, Irvine, 2000.
- [8] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Boston, MA, USA: Addison-Wesley, 1994.