Journal of Computer and Electrical Sciences (JCES)- Volume-1 Issue-4, April 2026
ISSN: 3049-2602

GADGET RECOMMENDATION

J. Sai Surya[1], B. Sasank[2], Y. Ramesh[3], P. Veerendra Kumar[4], B. Samba Naik[5], Mr. M. Samuel Sandeep[6]

[1,2,3,4,5] UG Student, Department of Computer Science and Engineering

[6]Assistant Professor, Department of Computer Science and Engineering

Andhra Loyola Institute of Engineering and Technology, Vijayawada, Andhra Pradesh, India

Email: saisuryajagilinki@gmail.com, boppanasasank2@gmail.com, yarraballiramesh43@gmail.com, veerendrapalchuri37@gmail.com, banavathsambanaik@gmail.com .

Abstract: The rapid expansion of the smartphone market has resulted in a wide range of devices with diverse features and price points, making smartphone selection challenging, particularly for users with limited technical knowledge. Many users find it difficult to compare specifications and identify devices that best match their requirements and budget. To address this issue, this project presents Gadget Recommendation, a web-based, usage-driven smartphone recommendation system designed to simplify the decision-making process. The system recommends smartphones based on user usage patterns such as gaming, photography, multitasking, and daily use, while considering key parameters including budget range, brand preference, processor performance, RAM, internal storage, camera quality, and battery capacity. A rule- based filtering approach combined with a scoring mechanism is employed to eliminate unsuitable devices and rank smartphones according to relevance. The application follows a modular frontend–backend architecture to ensure scalability, maintainability, and usability. Based on processed user inputs, the system generates optimized recommendations, including a best match and alternative options, thereby significantly reducing the time and effort required for smartphone selection. Overall, Gadget Recommendation provides an efficient and userfriendly decision support solution in the competitive smartphone market.

Keywords: E-commerce Recommendation System, Smartphone Recommendation, Fast API, MongoDB Atlas, NoSQL Database, Search and Filtering

1. INTRODUCTION

The rapid growth of the smartphone industry has created a large variety of devices with different specifications, features, and price ranges. While this diversity provides more choices to consumers, it also makes the process of selecting the right smartphone difficult for many users. People often struggle to compare technical specifications such as processor performance, RAM, storage, and battery capacity. Users who lack technical knowledge may find it challenging to identify a device that suits their needs and budget. As a result, many buyers rely on incomplete information or external opinions when making purchasing decisions. This situation highlights the need for a

https://doi.org/10.5281/zenodo.19614249

simplified system that can guide users in choosing suitable gadgets. Therefore, an intelligent recommendation system can help users make better and faster decisions.

The Gadget Recommendation System is designed to address this problem by providing personalized smartphone suggestions based on user preferences and usage patterns. The system analyzes factors such as budget range, brand preference, RAM, storage, processor capability, camera quality, and battery capacity. It also considers user activities such as gaming, photography, multitasking, and general daily usage. By evaluating these parameters, the system filters and ranks smartphones that best match the user's requirements. A rule-based filtering mechanism and scoring approach are used to remove unsuitable devices and highlight the most relevant options. This approach simplifies the complex comparison process involved in selecting smartphones. As a result, users receive recommendations that are practical and tailored to their needs.

The proposed system is implemented as a web-based application using modern technologies that ensure responsiveness and ease of use. The frontend interface allows users to enter their preferences and instantly receive recommendations. Smartphone data is retrieved from a backend service and processed to generate ranked suggestions. The system architecture is designed to be modular, scalable, and efficient for handling moderate datasets. By reducing the effort required to analyze multiple smartphone specifications, the system improves the overall decision-making experience. Additionally, the system provides alternative recommendations so users can compare different options easily. Overall, the Gadget Recommendation System serves as a helpful decision-support tool in the competitive smartphone market.

2. LITERATURE SURVEY

Early research in recommendation systems primarily focused on Collaborative Filtering (CF), which identifies patterns based on user-item interactions. While effective for large datasets, CF suffers from the "cold-start" problem when new gadgets are added without existing user ratings. To mitigate this, Content-Based Filtering was introduced, utilizing explicit item attributes (like RAM, Battery, and Display) to suggest products similar to those a user previously preferred.

Recent studies, such as the work by Akhdan and Baizal (2024), have explored Ontology-based Conversational Recommender Systems. Their research on smartwatches demonstrates that mapping domain expertise into knowledge graphs can provide highly accurate, explainable results. However, these systems often require high computational overhead and complex backend logic, which can increase response times in mobile web environments.

https://doi.org/10.5281/zenodo.19614249

System / Paper	Recommendation Technique	Personalization Method	Real-Time Interaction	Complexity	Limitations
[1] Ontology-Based Conversational Recommender System for Smartwatches	Ontology-based conversational recommender	Maps user functional needs to device specifications using knowledge graphs	Medium	High	Requires complex ontology construction and domain knowledge
[2] Smartphone Recommender System using FCPC-RGC	Fuzzy Cognitive Pairwise Comparison with clustering	User preference ratings compared using fuzzy logic	Low	High	Computationally complex and difficult for real-time applications
[3] Collaborative Filtering Recommender Systems for E-Commerce	Collaborative filtering	Uses user similarity and historical behavior	Low	Medium–High	Cold start problem and requires large user datasets
[4] Hybrid Recommendation Systems for Online Shopping	Hybrid (content + collaborative filtering)	Combines product attributes and user interaction history	Medium	High	Complex architecture and heavy computational overhead
Gadget Recommendation System (Proposed)	Rule-based filtering with weighted scoring	Uses user preferences such as budget, usage type, and specifications	High	Low	Depends on dataset completeness

Fig.2.1 Comparison with other works

In contrast, the shift toward lightweight, frontend-driven architectures suggests that offloading scoring and ranking logic to the client-side (e.g., using React) can drastically improve responsiveness. Current literature reveals a significant gap: most commercial systems rely on massive backend ML models that lack real-time UI validation, often leading to "null results" where users select incompatible filter combinations. This project addresses this by implementing a rule-

https://doi.org/10.5281/zenodo.19614249

based weighted scoring engine directly in the frontend, ensuring instantaneous feedback and usage-oriented recommendations for non-technical users.

3.PROBLEM STATEMENT

The rapid growth of digital gadget products has made it difficult for users to easily identify the most suitable gadget according to their requirements. Users often need to browse through multiple products and sources, which is time-consuming and inefficient. There is a need for a system that can organize gadget information and allow users to search, filter, and retrieve products based on criteria such as brand and price. The **Gadget Recommendation** project aims to solve this problem by providing a structured and efficient platform for gadget exploration and recommendation.

4. PROPOSED SYSTEM

The proposed system is a web-based Gadget Recommendation application that helps users search and explore gadgets based on their requirements. It provides features such as searching by brand or model, filtering by price, viewing all gadgets, and retrieving product details.

The system is implemented using Fast API for backend development and MongoDB Atlas for storing gadget data. It uses RESTful APIs to connect the application logic with the database and provide fast and structured responses.

The proposed system simplifies gadget discovery and helps users make better decisions by organizing product information in a user-friendly and searchable format. It also provides a scalable base for future enhancements such as personalized recommendations and advanced comparison features.

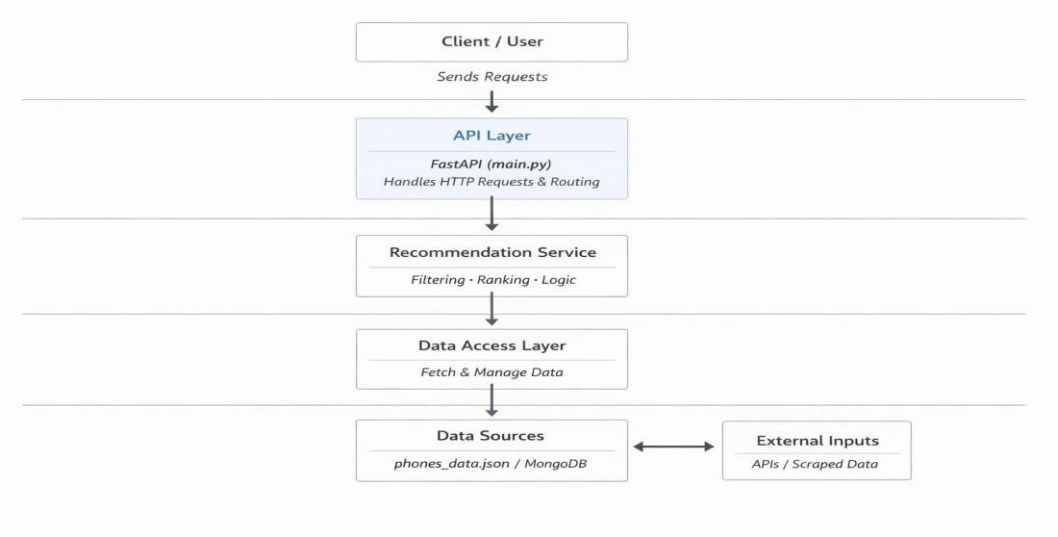
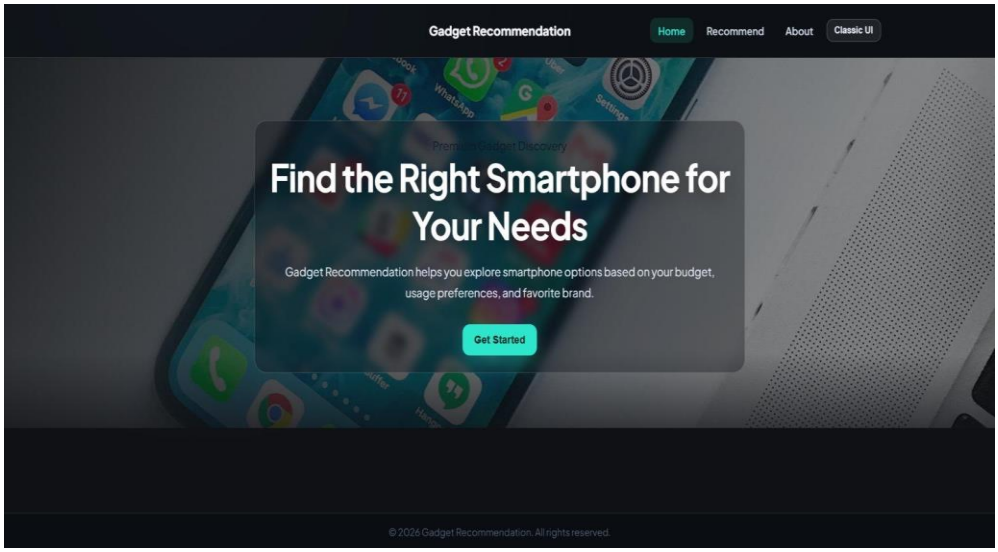


Fig: System Architecture

https://doi.org/10.5281/zenodo.19614249

3.1 System Components

1. User Interface Module: The user interface is developed using **React.js**. Users can interact with the system by selecting preferences such as budget range, usage requirements, and desired specifications. The interface also displays the recommended gadgets along with key specifications.



`

Fig. 2: User Interface

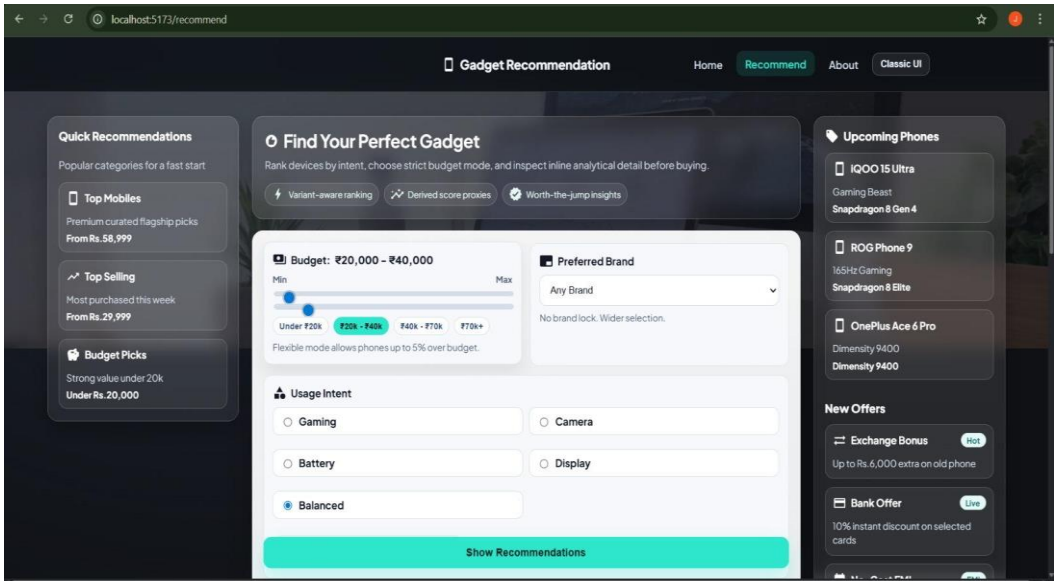


Fig. 3: User Interface

https://doi.org/10.5281/zenodo.19614249

2. Data Fetch Module: When the application loads, the frontend sends a request to the backend API endpoint (for example /phones). The backend retrieves the gadget dataset from the database and sends it to the frontend where it is stored in the **client cache**.

3. Data Preprocessing Module: The retrieved dataset is processed to ensure consistent formatting of attributes such as price, RAM, and storage. This step prepares the data for efficient filtering and scoring operations.

4. Validation Module: This module checks the combinations of user-selected filters and disables incompatible options dynamically. This ensures that users do not select impossible feature combinations and improves overall user experience.

5. Scoring and Ranking Module: The scoring engine evaluates each device using a rule-based weighted scoring algorithm. Different weights are applied based on user preferences such as gaming performance, camera quality, battery capacity, or overall performance. Devices are then ranked according to their calculated scores**.**

6. Recommendation Display Module: The top-ranked gadgets are displayed to the user along with important specifications and a brief explanation of why they were recommended.

5. METHODOLOGY

The **Gadget Recommendation** system is designed using a structured and modular methodology to provide efficient and user-oriented gadget suggestions. The working process of the system begins with user interaction and continues through data retrieval, preprocessing, validation, scoring, and final recommendation display. The methodology adopted for the system is explained below.

5.1 User Interaction and Preference Selection

The first stage of the methodology involves user interaction through a React.js-based frontend interface. The interface allows users to provide their preferences such as budget range, intended usage, and desired gadget specifications. These user inputs act as the basis for filtering and recommending suitable gadgets. The user interface is designed to be simple, interactive, and responsive, enabling easy navigation and selection of preferences.

5.2 Data Retrieval from Backend

After receiving user preferences, the frontend communicates with the backend system through RESTful API endpoints. The backend is developed using FastAPI, which processes incoming requests and retrieves gadget data from the MongoDB Atlas database. For example, when the application loads or when the user applies filters, the frontend sends a request to API endpoints such as /phones to obtain the required dataset.

https://doi.org/10.5281/zenodo.19614249

5.3 Data Preprocessing

Once the gadget data is fetched from the database, it undergoes a preprocessing stage. In this step, the system ensures that all important gadget attributes such as price, RAM, storage, and specifications are formatted consistently. This preprocessing phase improves the quality and usability of the data and prepares it for filtering, validation, and ranking operations.

5.4 Validation of User Inputs

The system then validates the user-selected combinations of preferences and specifications. This module ensures that incompatible or impossible combinations are automatically disabled or restricted. Such validation helps improve the overall user experience by preventing invalid search conditions and making the recommendation process more reliable.

5.5 Filtering, Scoring, and Ranking

After validation, the system evaluates the available gadgets using a rule-based weighted scoring mechanism. Different weights are assigned to gadget attributes depending on the user's requirements, such as gaming performance, camera quality, battery backup, or overall performance. Based on these weighted values, each gadget is scored and ranked. This allows the system to prioritize the most relevant devices according to the user's needs.

5.6 Recommendation Generation

Following the ranking process, the system selects the top-performing gadgets and generates personalized recommendations. These recommendations are based on the matching score obtained during the ranking stage. This step ensures that users receive the most suitable gadget suggestions instead of manually browsing through all available products.

5.7 Recommendation Display

Finally, the recommended gadgets are displayed to the user through the frontend interface. The results include important gadget specifications, ranking position, and a brief explanation of recommendation relevance. This presentation helps users easily compare and understand the suggested gadgets before making a decision

https://doi.org/10.5281/zenodo.19614249

6. RESULTS

The system successfully performed operations such as searching gadgets, filtering by price and brand, retrieving product details, and generating ranked recommendations based on user preferences. The output confirmed that the recommendation process worked correctly and provided relevant gadget suggestions to the user.

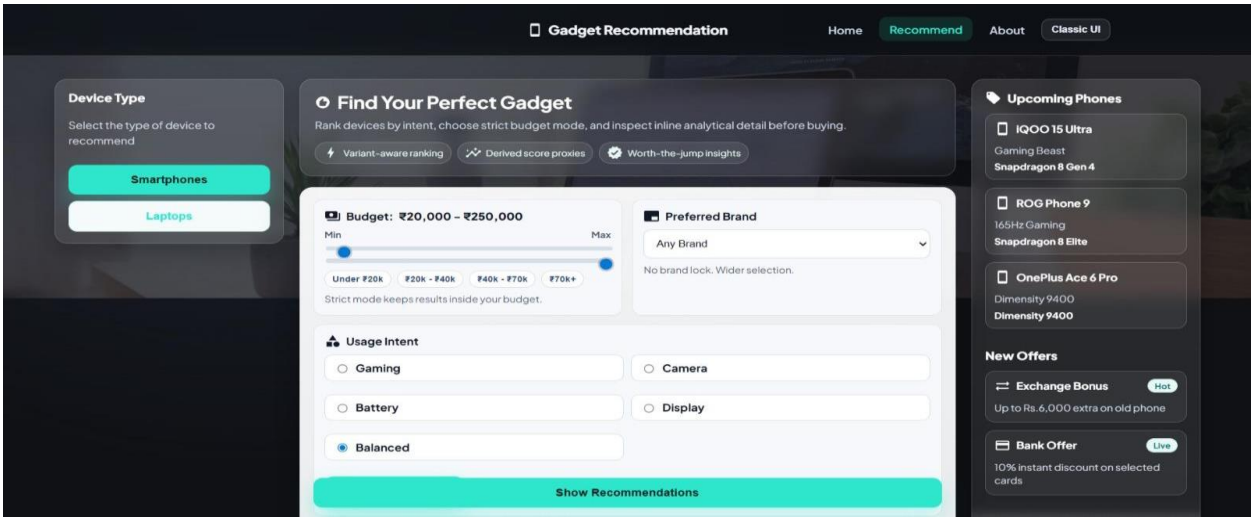


Fig1: Specifying Requirements

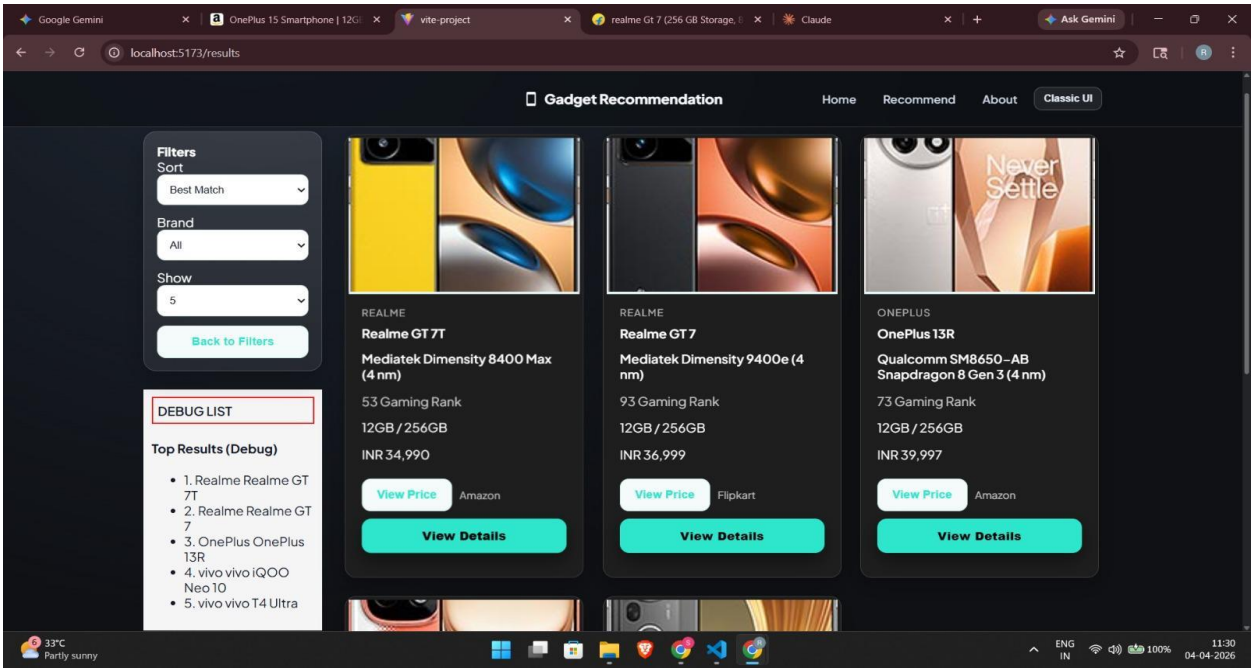


Fig2: recommended Gadgets

https://doi.org/10.5281/zenodo.19614249

7. CONCLUSION

The **Gadget Recommendation** project successfully presents the design and development of a **web-based recommendation and comparison system** that helps users search, filter, and evaluate gadgets more effectively. The system was developed using **FastAPI** for backend services and **MongoDB Atlas** for cloud-based data storage, providing a scalable and efficient platform for managing product information.

This project addresses the challenge of selecting suitable gadgets from a large and diverse set of available products. By incorporating features such as **search functionality, brand-based filtering, price-based filtering, and structured product retrieval**, the system improves the user's ability to make informed purchasing decisions. The use of **RESTful APIs** ensures smooth communication between the backend and the database, making the application efficient and modular.

From a technical perspective, the project demonstrates the practical implementation of **modern backend architecture**, **NoSQL database integration**, and **API-driven development** for a real-world consumer application. It also establishes a strong foundation for future enhancements such as **personalized recommendations, machine learning integration, user authentication, review analysis, and advanced product comparison features**.

In conclusion, the **Gadget Recommendation** system serves as a useful, scalable, and user-oriented solution for digital product exploration and selection. It combines modern web technologies with practical consumer needs, making it a valuable application in the field of **e-commerce and intelligent recommendation systems**.

8.REFERENCES

[1]. Smart Phone Recommendation System Using Machine Learning – *Atlantis Press*, 2025.

[2]. Recommendation Systems in E-Commerce Applications with Machine Learning Methods – *arXiv*, 2025.

[3]. Conflict Aware Retail Recommendation Decision Support System – *Springer*, 2026.

[4]. Web Recommendation System for E-Commerce Applications – *IJERT*, 2018.

[5]. Evolution of Modern Web Services – REST API with its Architecture and Design – *IJRESM*, 2021.

[6]. A Comparative Study of MongoDB and Document-Based MySQL for Big Data Application Data Management – *MDPI*, 2022.

[7]. E-Commerce Web Design at XYZ Store with Recommendation System Features – *JSAI*, 2025.

[8]. Recommendation Systems for Decision Support – *Decision Support Systems*, 2008.

https://doi.org/10.5281/zenodo.19614249