PROJECT REPORT

ON

AI POWERED E COMMERCE WEBSITE

Submitted by

Jaival Chauhan (U23AI035)

Aditya Kumar (U23AI029)

Jaimin Vankar (U23AI058)

Vikram Singh Mehrolia (U23AI034)

DEPARTMENT OF ARTIFICIAL INTELLIGENCE

SARDAR VALLABHBHAI NATIONAL INSTITUTE OF TECHNOLOGY
SURAT-395007

MAY-2025

CONTENTS

ABSTRACT

In Today's world of Rapid Digitalization , E Commerce websites have emerged as the primary medium of online shopping. Users prefer online shopping over offline because of ease of use , much larger range of products , door to door service and no offline hassle. Nowadays AI is being used more and more to simplify and automate tasks and to provide personalized user experience .

This Project presents an AI-enhanced E-commerce website which incorporates AI powered product recommendation system , semantic search to find products and an AI powered Chatbot to help users find the best products according to their requirements.

Our Project uses a K-means clustering algorithm to display similar products and to display products by categories. For displaying the products by categories the products are clustered into different categories based on the product description , tags and Product name.

For searching products we have used semantic search technique , the product descriptions are converted into embeddings and stored into a database and the user query is also converted into embeddings and then compared semantically to get the most relevant products.

We have also developed an AI powered chatbot which uses cosine similarity to map the user input regarding their product requirements to the product description embeddings. This helps the users to enter their product requirements in natural language and get more intuitive and interactive experience.

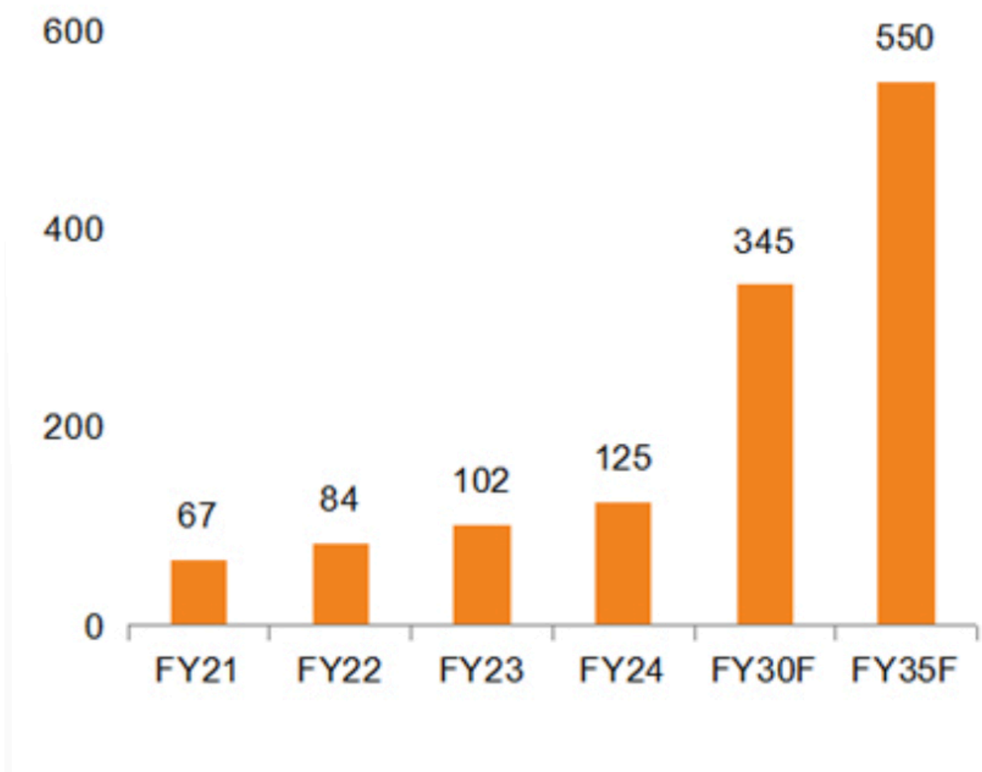
All these features have been employed on our Fully functional website where the users can Signup and Login securely , select the products they want to buy , add them to cart and finally complete the purchase. Our website's frontend interface has been made using Html , Css and Javascript and the Backend has been done using FastAPI. We have used various python libraries for the Machine learning tasks which we have employed like scikit-learn _____________ etc.

This Report gives a brief description of our system architecture , methodologies , implementation details , background of the problem which we are solving and the motivation for the same. Our work contributes to the growing field of AI based Ecommerce websites which provide much better and personalized experience to all the users.

1. INTRODUCTION

1.1. Background and Motivation of the Problem

The Growth of Ecommerce :-
The number of Smartphone and Internet users in India is expected to hit 1.1 billion by FY25. This has helped the growth of India’s digital space which will be valued at US$ 1 trillion by 2030. The GDP growth , increased internet penetration and rising median and mean incomes has helped the growth of the E commerce as well as quick commerce market in our country.
India’s Ecommerce industry was valued at US$125 billion in FY24 and is projected to grow to US$345 billion by FY30 reporting a compound annual growth rate (CAGR) of 15%. Further it’s expected to grow to US$550 billion by FY35. Report Titled ‘The AIdea of India: 2025’ 32% of Indian businesses are planning to invest in AI adoption.
As of Today E commerce has captured 45% of smartphone and 25% of laptop market, impact on home appliances sales has been relatively lower at 15-20%.



Source: News articles, F- Forecasted

Better User Experience :-
User experience has become a key differentiator among platforms. Consumers today expect more than just a Website they want seamless navigation , personalization product recommendations ,

and assistance regarding any queries related to the products. Poor user experience, irrelevant search results or irrelevant products being displayed, can lead to frustration and ultimately the user quitting the platform or moving on to an alternative.

Need of Personalized Recommendations :-
Research has shown that personalized product recommendations significantly impact consumer decision-making in online shopping. In an experimental study, Senecal and Nantel (2004) found that participants who consulted online recommendations selected the recommended product twice as often as those who did not. Importantly, recommender systems were more influential than traditional sources such as human experts or other consumers, demonstrating the power of personalization in guiding online purchase decisions.

Need of Chatbots :-
Users expect a human-like response , fast query resolution and personalized assistance like they would receive from an actual salesman. But most of the websites which are using traditional chatbots rely on Keyword based , hard coded answers to the users queries which degrades the user experience and is insufficient to provide proper assistance. These chatbots often redirect the user to an actual salesperson via call or text to solve the user's query if they are unable to solve it. This costs those ecommerce companies as they need to set up a call center to handle such queries and the user experience is also not great.
To deal with this our website employs a cosine similarity based chatbot which understands users requirements and recommends products based on that. The user can enter the requirements in natural language which is much more intuitive rather than selecting some filters or fixed values or attributes.

1.2. Literature Survey or Related Works

// Baaki hai

1.3. Contributions

This Project aims to show a detailed practical implementation of artificial intelligence in e-commerce to enhance user experience, streamline product discovery, and improve user engagement. The key contributions of this work are as follows:

A. AI-Based Recommendation System using K-Means Clustering
   We designed a recommendation system that categorises products based on their description , name and type and then whichever product the user is browsing similar products are shown from the same Cluster.

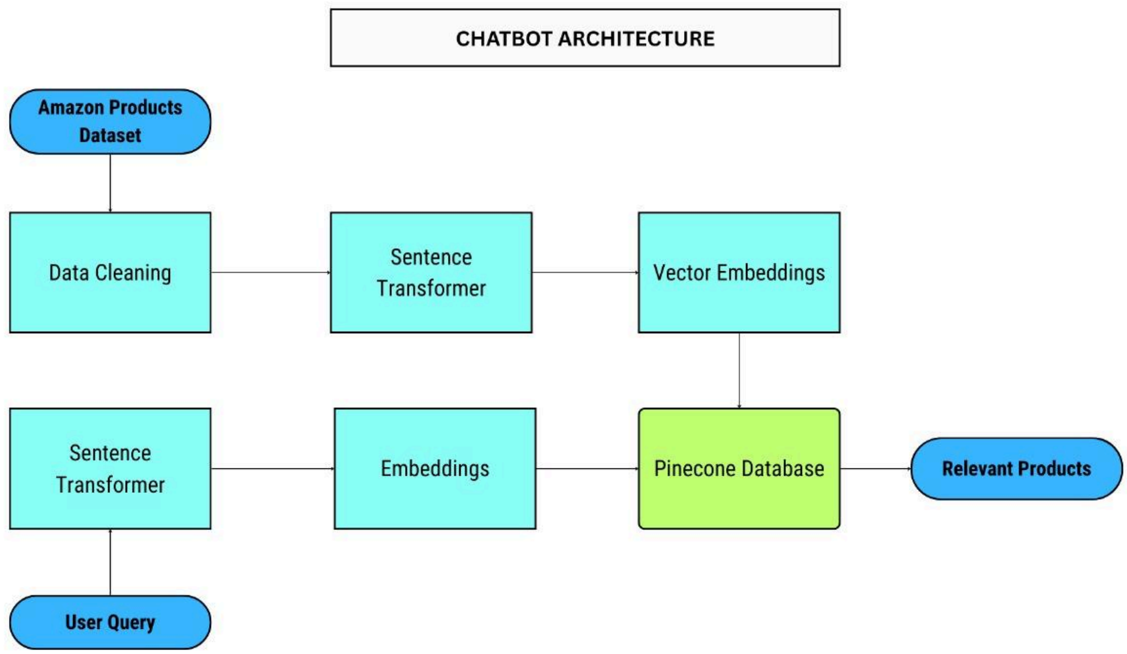
B. Semantic Search for Product Discovery
   Instead of using a traditional keyword based product search mechanism we have employed a custom semantic search mechanism which understands the users queries in

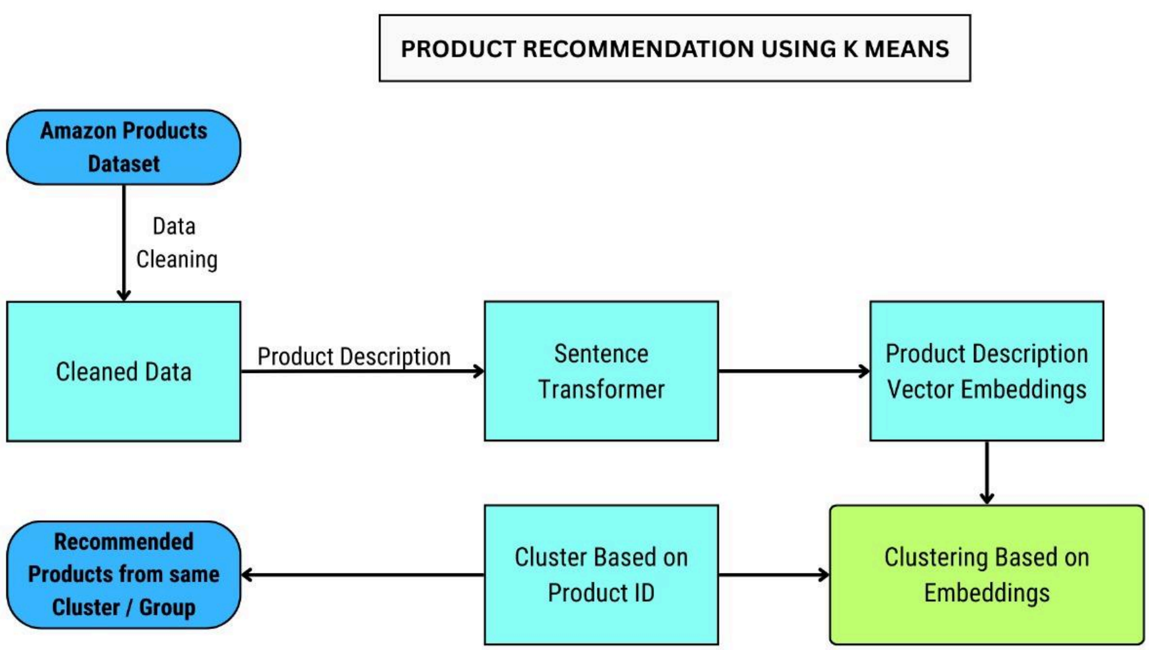
natural language. Through sentence embeddings and similarity matching, the system provides contextually appropriate product results even for vaguely described or non-exact user queries.

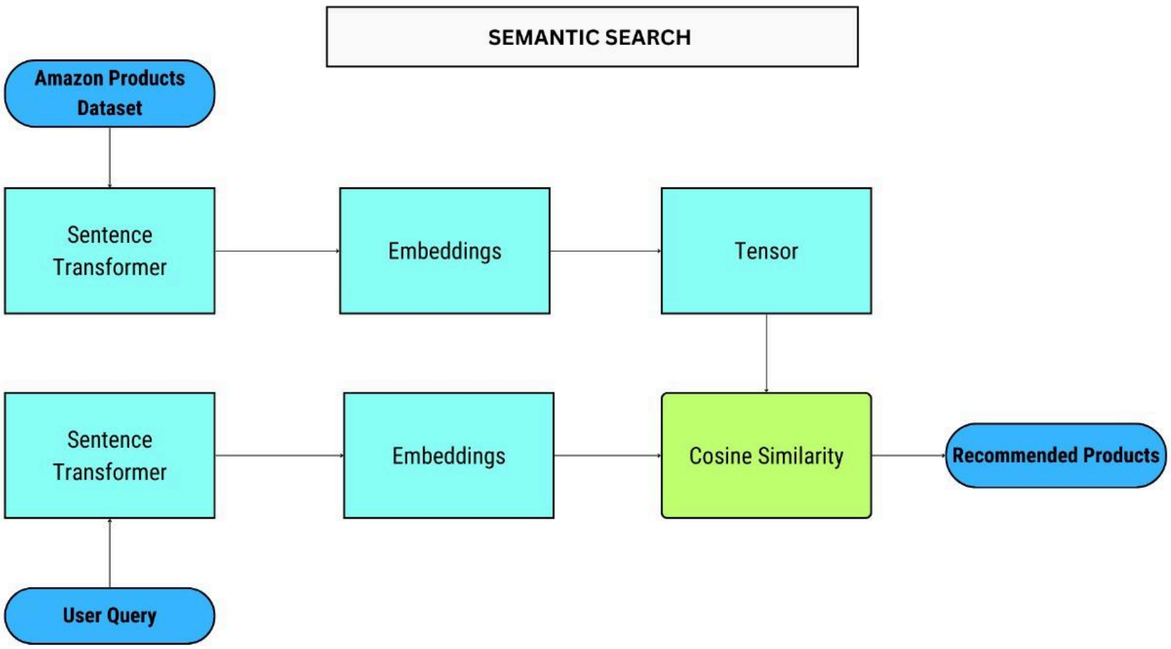
C. Product Recommendation Chatbot using cosine similarity
   We created an AI powered chatbot that engages with costumers and provides them accurate product recommendations by applying cosine similarity on the user queries and the product description database embeddings.

D. User Friendly Interface
   We Created a Simple , Intuitive , User friendly frontend for our website using Html , Css and Javascript for the frontend and FastAPI for the backend. This interface provides realistic E commerce usage scenario experience where we have employed various AI features which can be tested and their outputs can be visualized.

2. FLOWCHART/ SYSTEM DIAGRAM






SEMANTIC SEARCH
Amazon Products Dataset
Sentence Transformer
Embeddings
Tensor
Sentence Transformer
Embeddings
Cosine Similarity
Recommended Products
User Query

3. Methodology/Technique/Mechanism Name

Tech Stack:

Backend: FastAPI

Frontend: HTML, CSS, JavaScript

Database: MongoDB (For user data, product data, cart data, newsletter subscriber's data) and Pinecone (vector database for semantic search)

ML Models: Sentence Transformers [all-mpnet-base-v2] (for embeddings generation), KMeans Clustering. Cosine Similarity

Dataset: Amazon Product Dataset

A) The product recommendation system using K-Means clustering

1. Semantic Understanding and Embedding Generation :

   The system starts by embedding unstructured product descriptions into numerical representations using Sentence Transformers such as all-mpnet-base-v2 that encode semantic relations between products. The embeddings maintain contextual sense such that the model can learn that "wireless Bluetooth earphones" and "cord-free audio headsets" are essentially similar products although described differently.

2. Clustering Procedure :

   K-Means clustering clusters products by reducing intra-cluster variance in the embedding space. The algorithm divides the dataset into K clusters by iteratively updating centroids - abstract points that represent the center of each cluster. Products are mapped to clusters based on Euclidean distance to these centroids, building groups where items are similar in functional features, use cases, or complementary attributes.

3. Cluster Based Recommendation :

   When a user views a product, its precomputed cluster ID is retrieved from the enriched dataset. Recommendations are determined by choosing products within the same cluster, so that the recommendations are semantically consistent with the viewed item. This system is superior to keyword-based systems because it captures conceptual similarities, e.g., suggesting "ergonomic office chairs" with "posture-supporting workstations.".

4. Execution Procedure :

Data Preparation: Clean product descriptions and metadata from Amazon datasets.

Embedding Generation: Process text through Sentence Transformer to generate 384-768 dimensional vectors

Dimensionality Reduction: Optional PCA/t-SNE step for visualization and computational efficiency

Cluster Optimization: Find best K through elbow method or silhouette analysis

Cluster Assignment: Store cluster IDs along with product records for real-time retrieval.

Recommendation Engine: Get closest neighbors of the cluster with cosine similarity

Advantages and Expandability : This architecture supports real-time recommendations with O(1) cluster lookup time complexity. Precomputing embeddings and clusters, the system can handle large catalogs effectively with interpretability through cluster-level analysis. Hybrid implementations can take advantage of clustering combined with collaborative filtering, using clusters as features in more general recommendation models. Pragmatic Considerations The system needs to be retrained every now and then to support new products, the incremental clustering methods reducing computational overhead. Cluster stability can be enhanced using ensemble techniques or dimensionality regularization. Multilingual support can be provided using multilingual Sentence Transformers (e.g., paraphrase-multilingual-MiniLM-L12-v2) supporting cross-lingual product matching.

B) Search Pipeline Architecture

Technique used: Sentence Transformer + Pinecone

This framework substitutes keyword matching with concept-based retrieval with Pinecone's vector database8. When the user searches for terms such as "wireless earbuds with long battery," the Sentence Transformer model all-MiniLM-L6-v2 translates the query into a 384-dimensional vector that preserves semantic intent78. Pinecone then conducts approximate nearest neighbor (ANN) search over pre-indexed product embeddings and returns items that conceptually match the query instead of being exact keyword matches.

Execution Process

1. Embedding Generation: Amazon product description datasets are run through Sentence Transformers to create dense vector representations
2. Vector Indexing: Pinecone indexes these embeddings in distance metric-configurable optimized vector indexes (cosine similarity by default)

3. Real-time Querying: User queries initiate concurrent encoding and vector retrieval via Pinecone's low-latency API
4. Result Ranking: Similarity scores rank returned products, with metadata such as product names and prices added.

Technical Details :

The system deals with complex queries via semantic comprehension - it returns affordable ANC headsets when looking for "affordable noise-canceling headphones" even if product descriptions do not use exact words. Pinecone's distributed framework guarantees millisecond latency for massive catalogs, while its hybrid indexing accommodates sparse-dense vector pairs for improved accuracy.

Implementation Steps :

Developers launch Pinecone using pinecone.init(api_key="YOUR_KEY"), declare indexes with a specific dimensionality, and add vectors using index.upsert(). Query management involves translating user input with SentenceTransformer.encode() and retrieving results with index.query(vector=query_embedding, top_k=5).

Scalability Features :

Pinecone scales infrastructure automatically, hosting billions of vectors with sub-100ms latency. Hybrid search (sparse/dense vector mix) can be added to the system through pinecone-text utilities or multilingual support through models such as paraphrase-multilingual-MiniLM-L12-v2.

Pragmatic Factors :

The deployment comprises periodic refreshing of indexes for new products and model selection judiciously trading between accuracy/speed. Cost optimization includes right-sizing indexes and adding caching layers for high-frequency queries

C) AI Powered Chatbot

Technique used: Sentence Transformer + Cosine Similarity

This natural language interface employs robust semantic matching to provide accurate product recommendations. The site translates raw product information in Amazon's inventory into numerical information using Sentence Transformers such as all-mpnet-base-v2, which can identify subtle query and product similarities. When the customer asks for "Show me rugged hiking boots that cost less than $100," the chatbot follows a three-step procedure:

1. Semantic Encoding: Converts the query into a 384-dimensional vector that captures the intrinsic meaning, including price conditions and durability requirements.

2. Similarity Analysis: Computes cosine similarity scores between query vector and previously computed product embeddings.

3. Intelligent Ranking: Selects most appropriate matches where angular similarity of vectors indicates conceptual relevance

4. Technical Implementation
The architecture employs PyTorch-based embeddings in tensor storage for efficient computation. Cosine similarity's focus on direction rather than magnitude makes it a good fit for product description comparison regardless of length. For example, a search for "wireless earbuds with long battery life" accurately matches products described as "30-hour playtime Bluetooth earphones" despite the different words used.

5. Primary Benefits

   i. Conceptual Matching: Aware of the fact that "laptop for students" and "affordable college notebook" must match conceptually
   ii. Real-Time Performance: Executes queries in milliseconds using optimized vector operations.
   iii. Natural Interaction: Controls conversational improvements, like "Display lighter alternatives," without requiring a restart.
   iv. Execution Procedure

```
python
# Generate product embeddings
model = SentenceTransformer('all-mpnet-base-v2')
product_embeddings = model.encode(descriptions)

# Process user query
query_embedding = model.encode("ergonomic office chair under $200")
    similarities = cosine_similarity(query_embedding, product_embeddings)
    top_5_indices = np.argsort(similarities)[-5:][::-1]
```

Scalability The system efficiently processes large catalogs via batch embedding computation and parallel similarity calculation. Leveraging pre-trained models, it attains state-of-the-art semantic comprehension without domain-specific training. The design is a suitable basis for customized e-commerce conversations and retains interpretability via clear similarity scoring.

D) Other Functionalities:

1. **User Authentication:** Implemented using JWT for secure login and signup
2. **Cart System:** Users can add, view, and modify products in their shopping cart
3. **Newsletter Subscription**: Users can subscribe to updates and offers

4. IMPLEMENTATION ENVIRONMENT

1) Backend and API

- fastapi==0.115.11
- fastapi-cli==0.0.7
- uvicorn==0.34.0
- starlette==0.46.1
- python-dotenv==1.0.1
- python-multipart==0.0.20
- email_validator==2.2.0
- fastapi-mail==1.4.1

2) Security and Authentication

- bcrypt==4.3.0
- cryptography==44.0.2
- python-jose==3.4.0
- passlib==1.7.4
- rsa==4.9

3) Data Processing and Machine Learning

- pandas==2.2.3
- numpy==2.2.4
- scikit-learn==1.6.1
- scipy==1.15.2
- joblib==1.4.2
- threadpoolctl==3.6.0

4) Date and Time

- python-dateutil==2.9.0.post0
- pytz==2025.2
- tzdata==2025.2

5) Typing and Parsing

- pydantic==2.10.6
- pydantic_core==2.27.2
- typing_extensions==4.12.2
- annotated-types==0.7.0

6) Networking and Communication

- httpx==0.28.1
- httpcore==1.0.7
- h11==0.14.0
- httptools==0.6.4
- websockets==15.0.1
- sniffio==1.3.1
- anyio==4.9.0
- dnspython==2.7.0

7)Utilities and Supporting Tools

- click==8.1.8
- typer==0.15.2
- colorama==0.4.6
- shellingham==1.5.4
- watchfiles==1.0.4

8) Templating and Formatting

- Jinja2==3.1.6
- MarkupSafe==3.0.2
- PyYAML==6.0.2
- markdown-it-py==3.0.0
- mdurl==0.1.2
- Pygments==2.19.1
- rich==13.9.4
- rich-toolkit==0.13.2

9) Cryptographic and Low-Level Libraries

- cffi==1.17.1
- pycparser==2.22

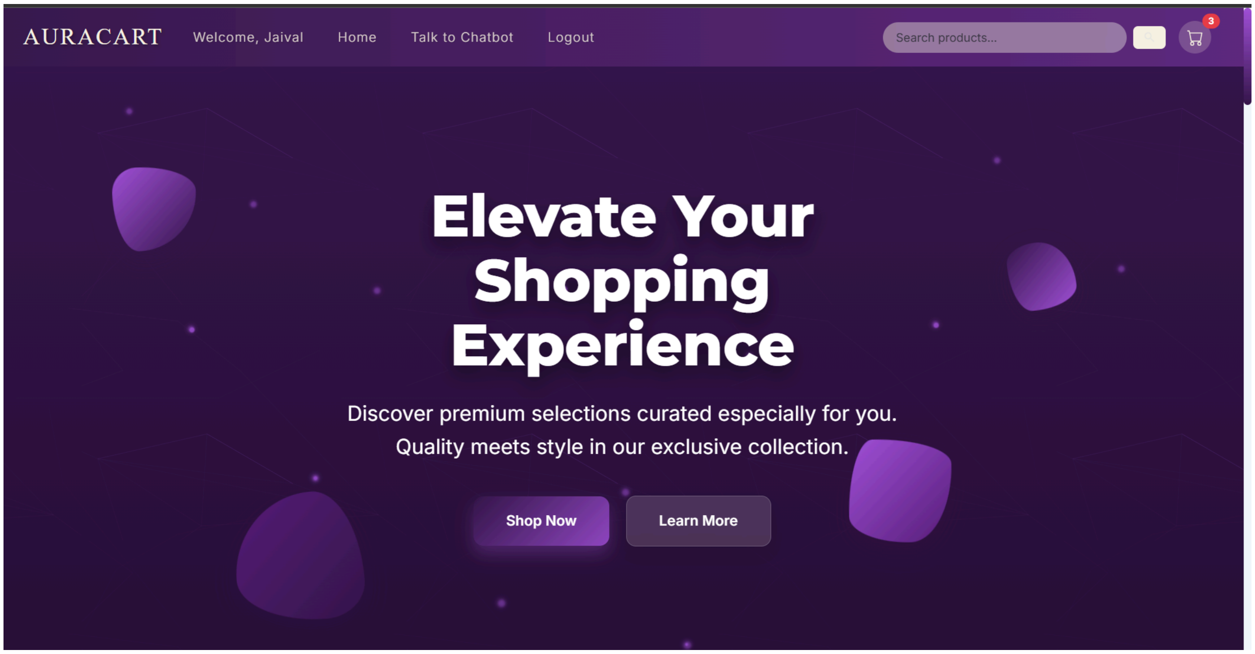
- ecdsa==0.19.1
- pyasn1==0.4.8

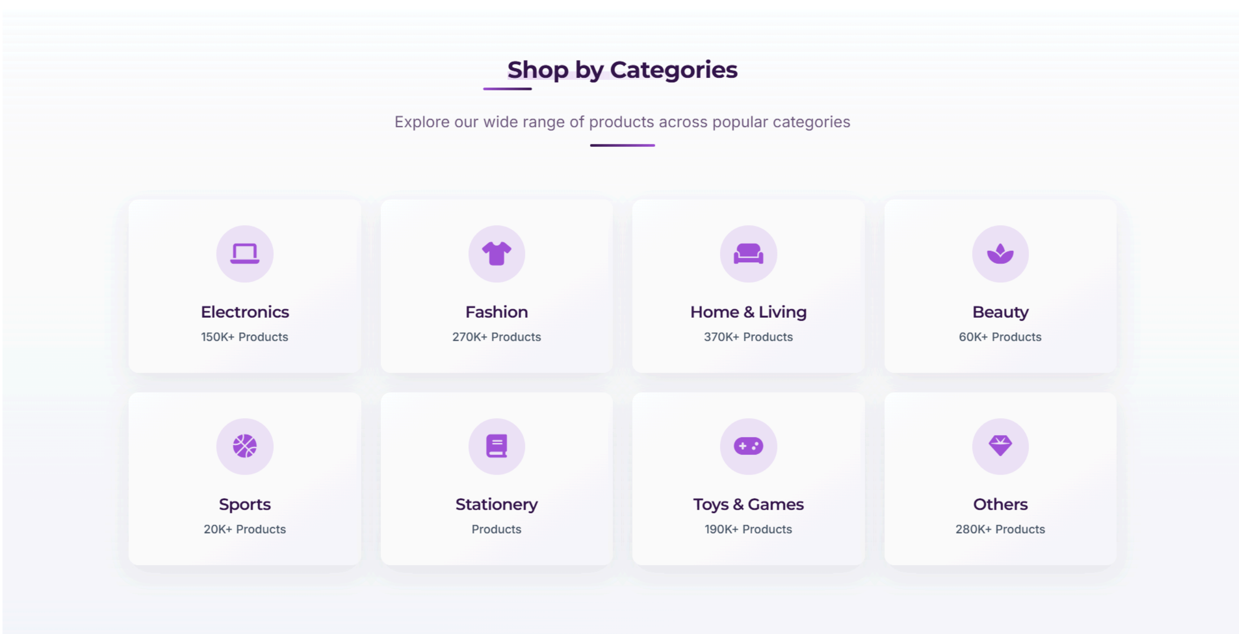
8) Others

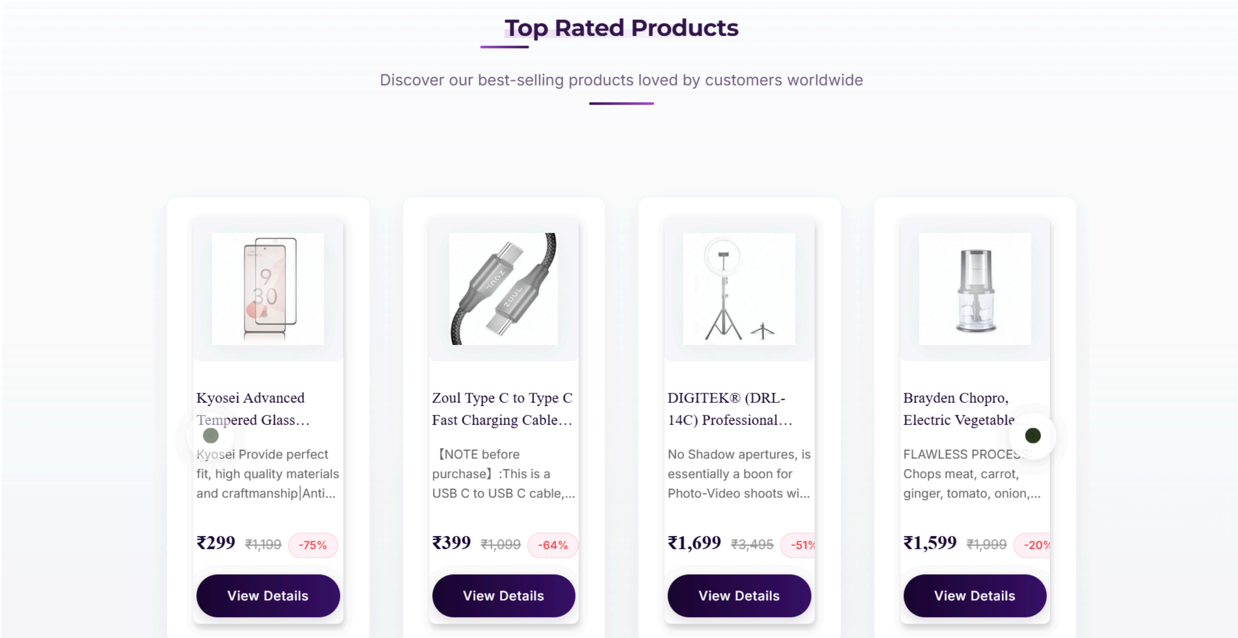
- certifi==2025.1.31
- idna==3.10
- six==1.17.0
- pymongo==4.11.3
- pinecone *(used for vector database functionality)*
- polars *(high-performance data frame library)*

5. Results

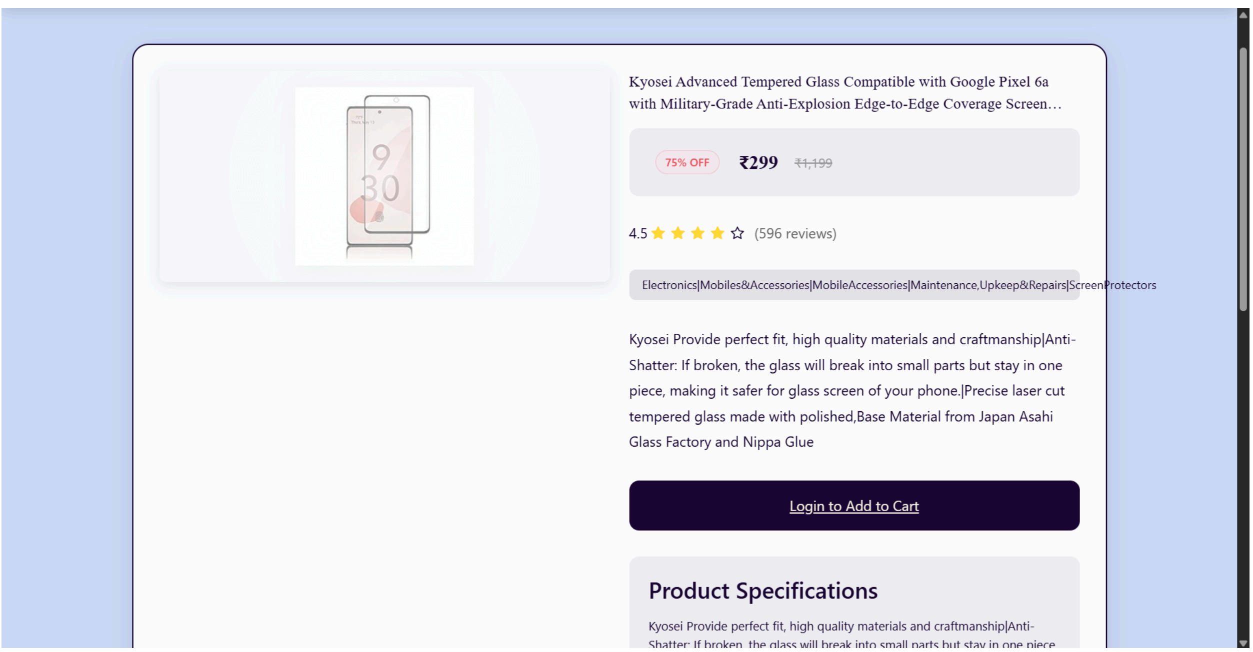
Home Page:

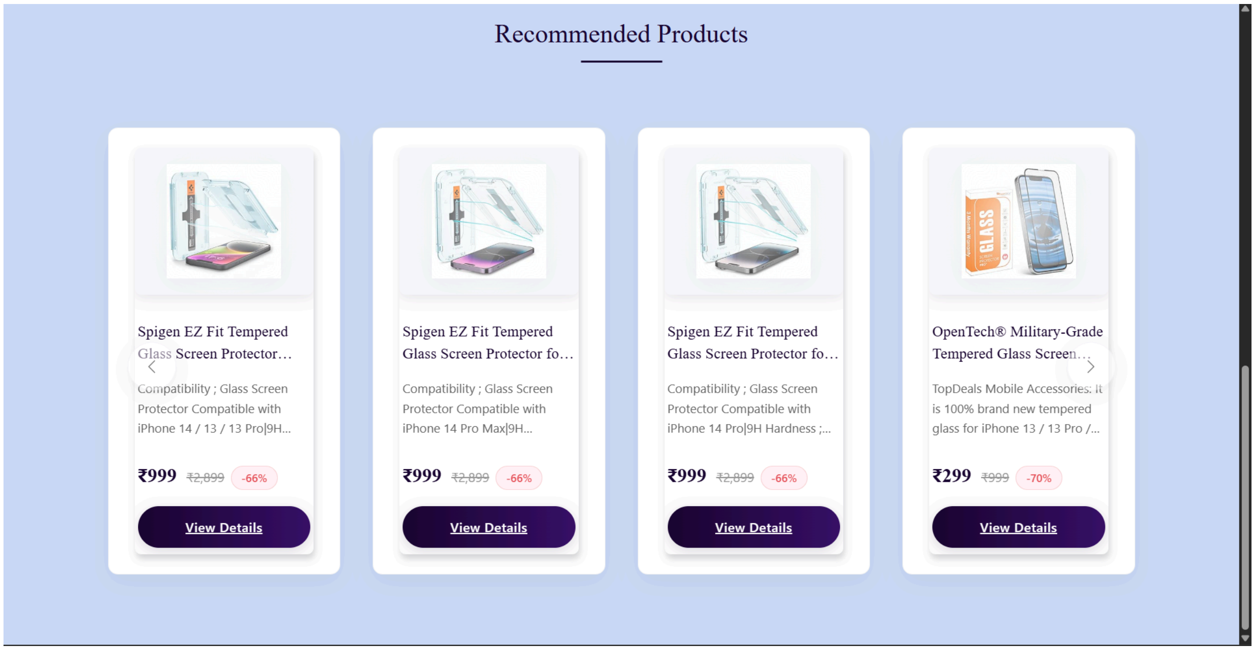




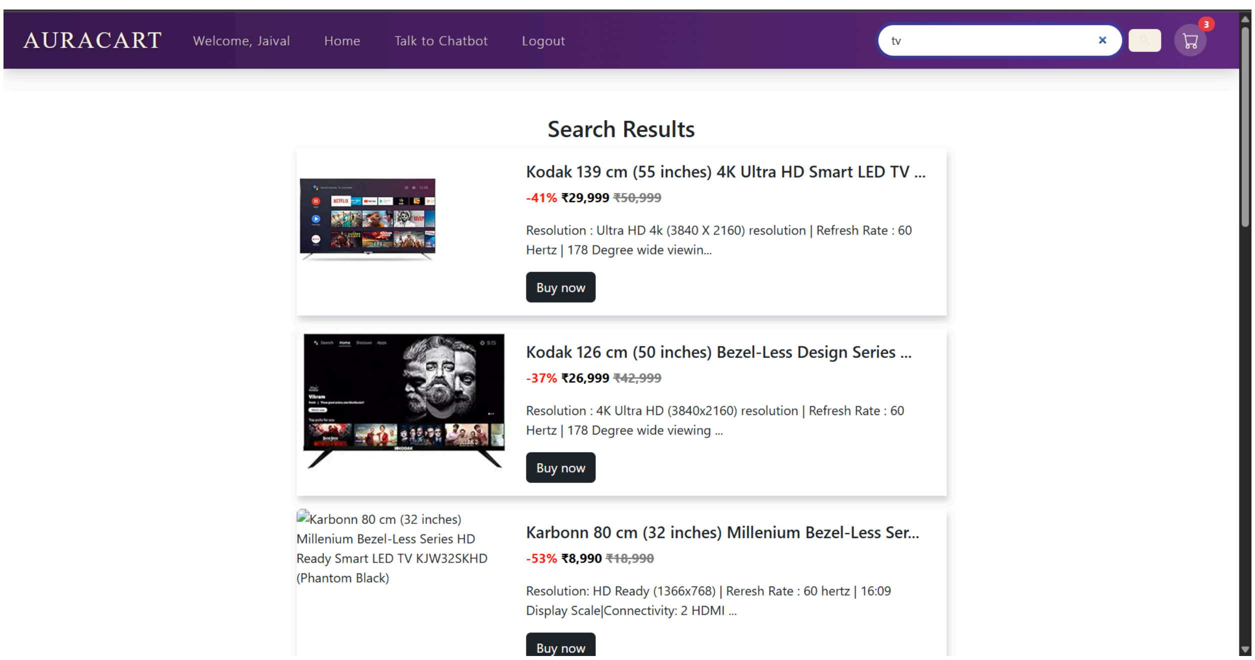


Recommending Products using Clustering:

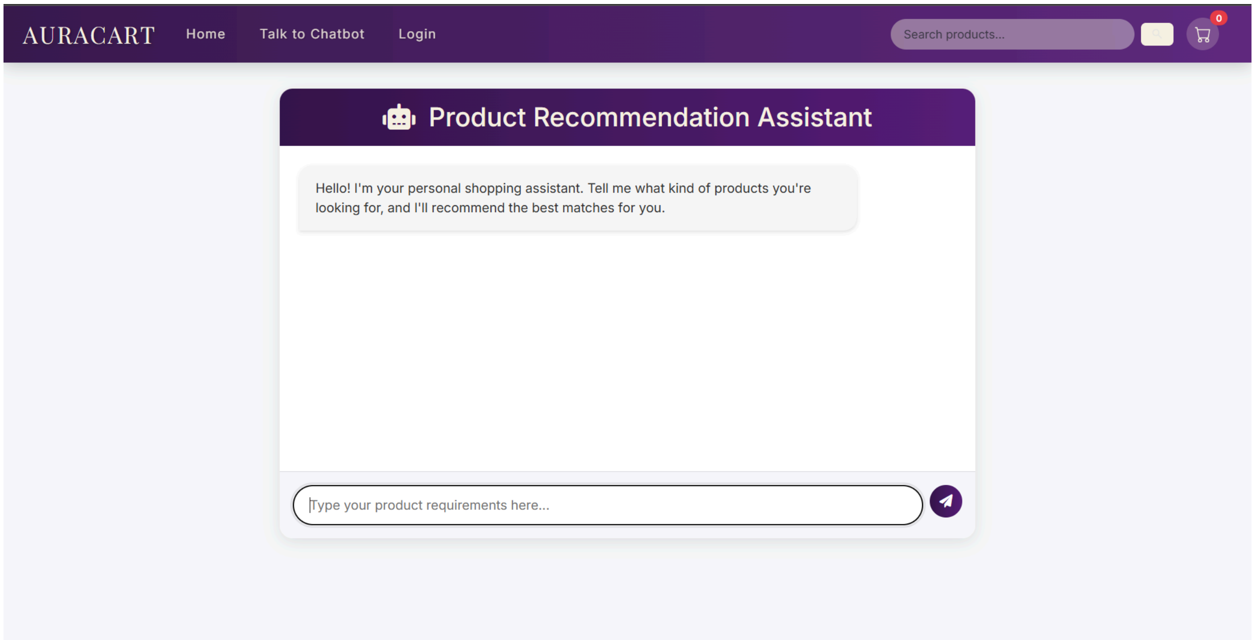


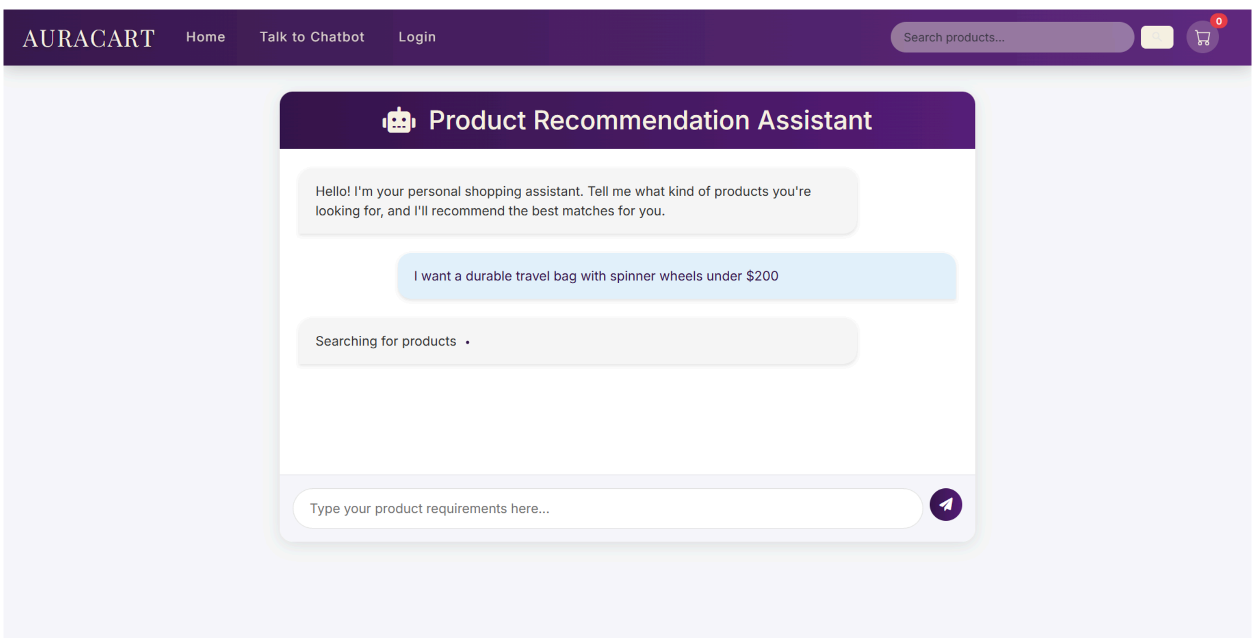


Product Search using Semantic Search:



Product Recommendation Chatbot:





Recommended Products

![Carry On Luggage with Spinner Wheels 20in Lightweight Suitcase Built in TSA Aluminum Frame PC Hardside Rolling Suitcases Travel Case (Titanium Gold)]

Carry On Luggage with Spinner Wheels 20in Lightweight Suitcase Built in TSA Aluminum Frame PC Hardside Rolling Suitcases Travel Case (Titanium Gold)

$129.99

★★★★★

View Product →

![Travel Softside 28 Inch Luggage with Spinner Wheels Lightweight Expandable Large Suitcase 8-wheel Spinners]

Travel Softside 28 Inch Luggage with Spinner Wheels Lightweight Expandable Large Suitcase 8-wheel Spinners

$109.99

★★★★☆

View Product →

![Travel Softside 28 Inch Luggage with Spinner Wheels Lightweight Expandable Large Suitcase 8-wheel Spinners]

Travel Softside 28 Inch Luggage with Spinner Wheels Lightweight Expandable Large Suitcase 8-wheel Spinners

$109.99

★★★★☆

View Product →

![Travel Softside 28 Inch Luggage with Spinner Wheels Lightweight

![Reaction Rugged Roamer Luggage Collection Lightweight

![Suitcase Spinner Wheels 26'' Checked Luggage Hardshell

6. CONCLUSION

7. References

1. https://www.ibef.org/industry/ecommerce
2. https://www.researchgate.net/publication/222519112_The_Influence_of_Online_Product_Recommendations_on_Consumers'Online_Choices/link/5a01b4c5aca272e53ebb2804/download?_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6InB1YmxpY2F0aW9uIiwicGFnZSI6InB1YmxpY2F0aW9uIn19
3. https://retailtechinnovationhub.com/home/2024/8/26/understanding-semantic-search-with-pinecone
4. https://www.ai.codersarts.com/post/pinecone-for-semantic-search
5. https://www.pinecone.io/learn/series/nlp/sentence-embeddings/
6. https://www.pinecone.io/learn/series/nlp/sentence-embeddings/