BSIT-S25-004

03-135221-033 Muhammad Faizan Tahir

03-135221-021 Muhammad Faeez Salman

ALF: The Ultimate Customer Support Bot

In partial fulfilment of the requirements for the degree of

Bachelor of Science in Information Technology

Dr Junaid Nasir Qureshi

Department of Computer Sciences

Bahria University, Lahore Campus

December 2025

C e r t i f i c a t e

We accept the work contained in the report titled

"ALF: The Ultimate Customer Support Bot"

written by

Muhammad Faizan Tahir

Muhammad Faeez Salman

as a confirmation to the required standard for the partial fulfilment of the degree of Bachelor of Science in Information Technology.

Approved by:

Dr Junaid Nasir Qureshi

____(Signature)

December 05, 2025

DECLARATION

We hereby declare that this project report is based on our original work except for citations and quotations which have been duly acknowledged. We also declare that it has not been previously and concurrently submitted for any other degree or award at Bahria University or other institutions.

Enrolment	Name	Signature
03-135221-033	Muhammad Faizan Tahir	
03-135221-021	Muhammad Faeez Salman	

Date : December 05, 2025

Specially dedicated to

My Beloved Grandmother, Mother, Father and Teachers

(Muhammad Faizan Tahir)

My Beloved Grandmother, Mother and Father

(Muhammad Faeez Salman)

ACKNOWLEDGEMENTS

We would like to express our sincere gratitude to our research supervisor, Junaid Nasir, for his incredibly helpful guidance, constant support, and amazing patience right up until this project was finished. His knowledge and thoughtful feedback really made our work better and more successful.

We're also so grateful to our parents, friends and Teachers for always being there to encourage us, keep us motivated, and support us throughout every single step of this project.

Muhammad Faizan Tahir
Muhammad Faeez Salman

ALF: The Ultimate Customer Support Bot

ABSTRACT

ALF: Modern customer support systems often struggle with delayed responses, limited scalability, and inefficiency in resolving complex queries, particularly those involving visual or contextual product-related issues. Traditional chatbots lack real-time access to dynamic databases and fail to interpret multimodal inputs like images or nuanced textual descriptions, leading to generic or inaccurate solutions[1]. This gap negatively impacts user satisfaction, increases operational costs, and strains human support teams. ALF: The Ultimate Customer Support Bot addresses these challenges by offering an intelligent, automated solution capable of processing both visual and textual inputs while integrating se amlessly with a company's backend systems. By enabling real-time access to product catalogues, order statuses, and delivery timelines, ALF ensures precise, context-aware responses, reducing resolution time and enhancing customer experience[2]. Its significanc e lies in bridging the automation-human support divide, optimizing resource allocation, and fostering brand loyalty through instant, reliable assistance. ALF leverages advanced Large language models (LLM) to interpret user queries, whether textual (e.g., "hair oil for strong hair") or image -based (e.g., product photos). Integrated with RESTful APIs[3], the bot fetches real-time data from the company's product and order databases to provide accurate recommendations. A deep neural network[4] model trained on product attributes and historical customer interactions enables ALF to suggest alternatives when exact matches are unavailable. The system's frontend, designed for web[5], features an intuitive UI with chat functionality and image-upload support. Backend integration employs cloud - based microservices for scalability, while security protocols ensure data privacy. By combining LLM, image recognition frameworks (e.g., CLIP), and API-driven database interactions[3], ALF delivers a unified, efficient support platform that automates query resolution, reduces human intervention, and elevates user engagement across digital touchpoints[6].

TABLE OF CONTENTS

Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

Symbol/Abbreviation	Expansion
AI	Artificial Intelligence [2]
API	Application Programming Interface [3]
CLIP	Contrastive Language-Image Pre-training [4]
CNN	Convolutional Neural Network [5]
CSV	Comma-Separated Values [6]
CV	Computer Vision [7]
DFD	Data Flow Diagram [8]
DNN	Deep Neural Network [9]
JSON	JavaScript Object Notation [10]
LLM	Large Language Models [11]
LSTM	Long Short-Term Memory [12]
NLP	Natural Language Processing [13]
REST	Representational State Transfer [14]
RNN	Recurrent Neural Network [15]
UI/UX	User Interface / User Experience [16]
UML	Unified Modeling Language [17]
ViT	Vision Transformer [18]

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

The rapid digitization of customer service has highlighted critical gaps in traditional support systems[7], particularly in handling multimodal queries and delivering real-time, context-aware solutions. Existing chatbots often rely on predefined scripts or limited databases, resulting in delayed, generic responses that fail to address complex user needs, such as interpreting product images or nuanced textual requests. ALF: The Ultimate Customer Support Bot emerges as an innovative solution designed to overcome these limitations by integrating artificial intelligence (AI)[8] with real-time data access. This project aims to develop a versatile support bot capable of processing both textual and visual inputs, interfacing seamlessly with a company's product and order databases via secure APIs, and delivering accurate, personalized assistance. ALF combines LLM and deep neural network (DNN)[9] technologies to analyse user queries, whether text-based (e.g., "ready to eat food") or image-driven (e.g., product photos). By connecting to the company's backend systems, the bot retrieves live data on inventory, enabling dynamic responses such as product recommendations and alternative suggestions. The system's architecture includes a user-friendly interface for web platforms, ensuring broad accessibility. This project addresses the growing demand for scalable, 24/7 customer support while reducing reliance on human agents[10], minimizing operational costs, and enhancing user satisfaction. ALF's ability to bridge automation with contextual understanding positions it as a transformative tool for modern businesses striving to elevate customer engagement in competitive markets.

1.1 Background

The quick shift to digital in businesses today has made customers expect fast, accurate, and personalized help more than ever. Traditional customer service setups, especially simple chatbots[11], often fall short because they stick to fixed scripts and limited information. This means they struggle with complex questions, can't handle things like product pictures[12], and frequently give generic or slow answers. These issues leave customers feeling frustrated and pile extra work onto human support teams[2].

Today's businesses need smarter support systems that can understand both what customers type and what they show[13], pull up current product and order details, and offer precise, relevant solutions. This is especially true for online stores and retailers where shoppers often ask for product suggestions, want updates on their orders, or need help identifying items by picture. Technologies like LLM [14] and Deep Neural Networks (DNN) [12], paired with secure connections to company systems, offer a solid way to automate all of this.

That's where ALF: The Ultimate Customer Support Bot comes in it's designed to tackle these modern needs. By using LLM, image recognition, and real-time access to databases via APIs, the system promises to deliver instant, smart, and personalized support. This approach not only cuts down on operating costs but also boosts how satisfied customers are with their service experience[15].

Table 1:Comparison Table (Traditional vs. AI Chatbots)

Feature	**Traditional Chatbots (Rule-Based)**	**ALF (Your System)**
Technology	if-else scripts, Keyword Matching	Llama 3 (LLM) & Deep Learning
Input Type	Text only	Multimodal (Text + Images via CLIP)
Context	Single interaction only	Multi-turn memory & context aware
Data Access	Static / Hard-coded responses	Real-time Database (API)
Flexibility	Breaks on unknown queries	Generative & Adaptive

1.2 Problem Statements

Traditional customer support chatbots have their limits. They often can't make sense of questions that mix different types of information (like text and images) and usually can't tap into live data about products or orders. They struggle with interpreting uploaded photos, handling requests that are a bit tricky to understand, and typically just rely on pre-set rules or static answers. This frequently leads to customers getting answers that are either wrong or too general, causing frustration, taking longer to solve problems, and putting more pressure on human support staff.[16]

What we really need is a smarter customer support system, one that can:

- Really understand what customers are asking in natural language.
- Identify products shown in photos that users send in.
- Connect securely to live information about products, stock levels, and orders.
- Offer helpful suggestions and alternatives based on the specific situation.
- Keep running smoothly, even when dealing with huge and constantly changing product lists.

ALF tackles these issues head-on. It combines LLM, computer vision, and the ability to query databases in real-time. This creates a fast, dependable, and automated support experience that makes customers happier and cuts down on the work needed from support teams.[17]

1.3 Aims and Objectives

The objectives of the project are shown as following:

i) Levrage an LLM to process textual queries and extract intent, context, and product-related keywords.

ii) Implement a deep neural network image recognition system to identify products from user-uploaded images using PyTorch.

iii) Design RESTful APIs for real-time interaction with the company's product and order databases.

iv) Create a UI/UX for web with chat functionality and image-upload support.

v) Design APIs to integrate the bot with other applications

vi) Integrate security protocols to safeguard data during API transactions and user interactions.

vii) Train a machine learning model to suggest alternative products when exact matches are unavailable.

viii) Validate the system's accuracy and scalability through stress testing and user feedback loops.

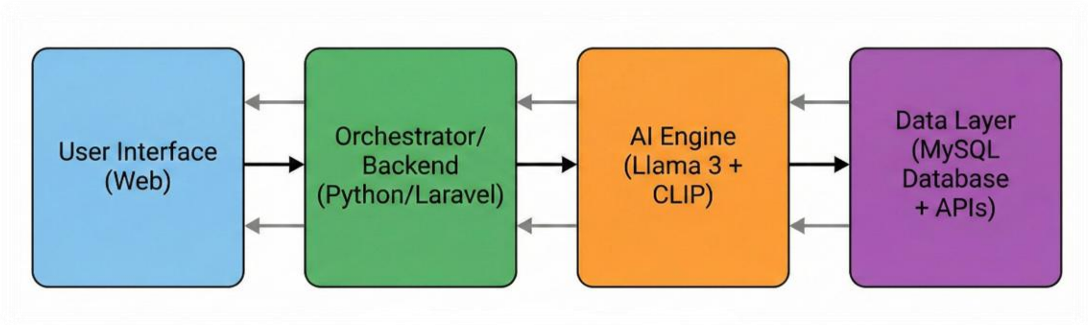

Figure 1:High-Level System Block Diagram

1.4 Scope of Project

The primary scope of this project is to design and develop ALF: The Ultimate Customer Support Bot[18], an AI-driven system that resolves customer queries through multimodal inputs (text and images) while integrating real-time data from a company's product and order databases. Traditional chatbots lack the ability to interpret visual[19]. ALF: Ultimate Customer Support Chatbot 2 inputs or dynamically access backend systems, leading to generic responses, prolonged resolution times, and customer frustration. ALF addresses these gaps by combining LLM and deep neural network (DNN) to analyse user queries, such as textual descriptions (e.g., “dry fruits”) or product images, and cross-references them with live inventory and order data via secure APIs[20]. Key challenges include ensuring seamless integration with heterogeneous databases, maintaining data privacy during API interactions, and achieving high accuracy in image recognition amid varying product attributes[21]. The system’s fault tolerance lies in its ability to handle API failures gracefully, provide alternative recommendations when exact matches are unavailable, and adapt to evolving product catalogues. By automating complex query resolution, ALF reduces reliance on human agents, cuts operational costs and enhances customer satisfaction through instant, context-aware support.[22]

CHAPTER 2

LITERATURE REVIEW

2.1 Overview of AI-Based Customer Support Systems

The growing reliance on digital platforms has really changed how customers connect with businesses, making it crucial to have smart, automated support in place. Traditional customer service methods, which often depend on set rules or pre-written answers, just don't cut it when dealing with complicated or situation-specific questions. Because of these shortcomings, researchers have been looking into AI-based solutions, particularly those using natural language processing, machine learning, and deep neural networks to create strong customer support chatbots. Earlier research shows that these AI systems can greatly cut down response times, lower operating costs, and reduce the burden on human staff, all while making the customer experience better overall.

Existing research highlight several shortcomings in rule-based support systems, especially their struggle to adapt to different user questions, interpret what users really mean, and handle vague or complex requests. Today's customer conversations often involve back-and-forth exchanges, emotional signals, and sometimes images, which underscores the need for approaches that can process multiple types of information. As companies grow, the sheer volume of customer inquiries becomes overwhelming, making it impossible to manage them all manually. This is why research strongly backs the shift to smarter support systems that combine natural language processing and computer vision to provide accurate, tailored, and instant answers.

2.2 Large language Models (LLMs) in Customer Support

LLMs have revolutionized the foundation of conversational systems, moving beyond simple text processing to deep semantic understanding and intent generation. While early chatbot research relied on rigid keyword matching and manual templates methods that failed to handle the fluidity of natural conversation recent advancements have shifted toward massive, pre-trained architectures. Unlike previous statistical methods or RNNs (like LSTM), which often struggled with long-range dependencies, modern LLMs leverage the Transformer architecture to master context from vast datasets.

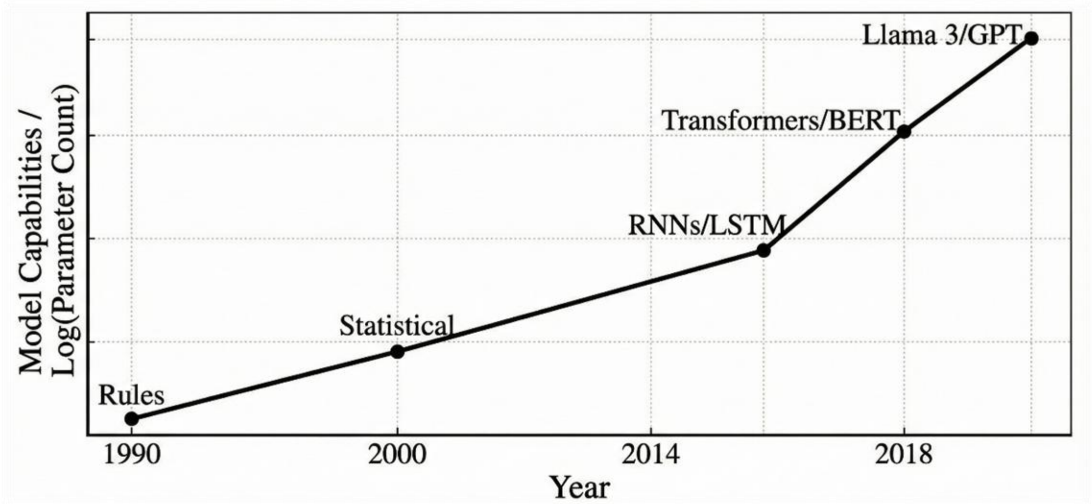

Figure 2:Evolution of NLP Models

The integration of transformer-based LLMs, particularly Llama 3 and GPT-style designs, has changed the landscape by enabling bidirectional context understanding and reasoning. Research confirms that these generative models far outperform traditional NLP pipelines in tasks like entity recognition, sentiment analysis, and maintaining complex conversation history. This is particularly vital in e-commerce, where users communicate in casual, fragmented, or descriptive language. These capabilities make an LLM-based approach perfect for ALF, ensuring that vague user queries like “dry fruits” or “hair oil for strong hair,” as well as multi-turn interactions requiring memory, are interpreted with high precision.

2.3 Computer Vision and Image-Based Product Recognition

Computer vision has become a really useful tool for helping customers, especially in industries where being able to identify products visually is super important. Studies show that types of deep neural learning models, like CLIP, and EfficientNet, are often used because they are great at accurately classifying images and picking out key features. CLIP is a popular choice among these because of its special way of learning that allows it to handle very deep networks without losing accuracy. Research has proven it’s effective at dealing with common issues in photos customers upload, like differences in lighting, rotation, things blocking the view, and messy backgrounds.

Image recognition tech is making waves in industries like retail, healthcare, logistics, and food. It's used for everything from spotting products and finding flaws to sorting stock and helping people find similar items. For chatbots that offer support, adding computer vision lets customers just upload a picture if they don't know the name of what they are looking for. This really boosts how easy it is to use the service. Experts point out that mixing computer vision with natural language processing creates systems that can understand more complex questions and give better answers. This blend is a perfect fit with the goals of ALF, making it possible to automatically spot products in user pictures and link them up with the database to suggest items.

2.2.1 OpenAI CLIP and Multimodal Understanding

OpenAI’s CLIP [23](Contrastive Language-Image Pre-training) addresses the limitations of traditional vision models by bridging the gap between visual data and natural language. Unlike standard convolutional networks that rely on fixed categories, CLIP uses a dual-encoder architecture to map images and text into a shared embedding space, enabling the system to understand visual content through semantic meaning.

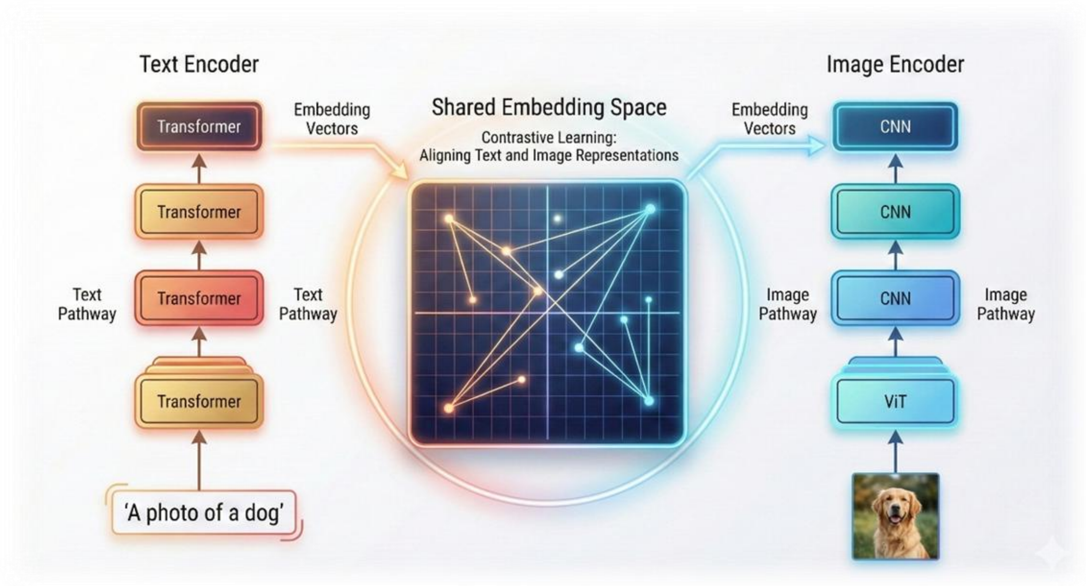

Figure 3:CLIP Architecture Diagram

This approach allows CLIP to perform "zero-shot" learning[24], meaning it can recognize objects it has never explicitly been trained on. Research indicates that CLIP is exceptionally robust when dealing with real-world photos—such as user-uploaded product images with inconsistent lighting, unique angles, or background clutter—making it far more effective than traditional models for e-commerce tasks where users describe visual items in natural language[25].

2.3 Real-Time Database Access and API Integration

To operate effectively, today's smart systems need quick and secure connections to databases that are constantly updated. This allows them to provide answers that are both correct and current. Research indicates that many business applications use RESTful APIs to pull in live information like whether products are in stock, current prices, order progress, and when items will be delivered. How well customer support chatbots perform really hinges on their skill in smoothly connecting with the services running behind the scenes. Earlier studies emphasize the value of creating API structures that minimize delays, using caching techniques, and guaranteeing the system stays dependable even when lots of people are using it at the same time.[26]

2.4.1 Role of RESTful APIs in Intelligent Support Systems

RESTful APIs [3]are now the go-to way for client apps to talk to backend services. Research really drives home that REST APIs offer scalability, lightweight communication, and they work across different platforms, which is perfect for customer support systems used on the web, mobile, and in the cloud. Because they have a stateless design, they can quickly pull live data like stock counts or shipping updates, ensuring chatbots can give real-time answers[27]. For systems like ALF, RESTful APIs serve as the essential link connecting the chatbot to things like product catalogs, customer order histories, and various backend tools.

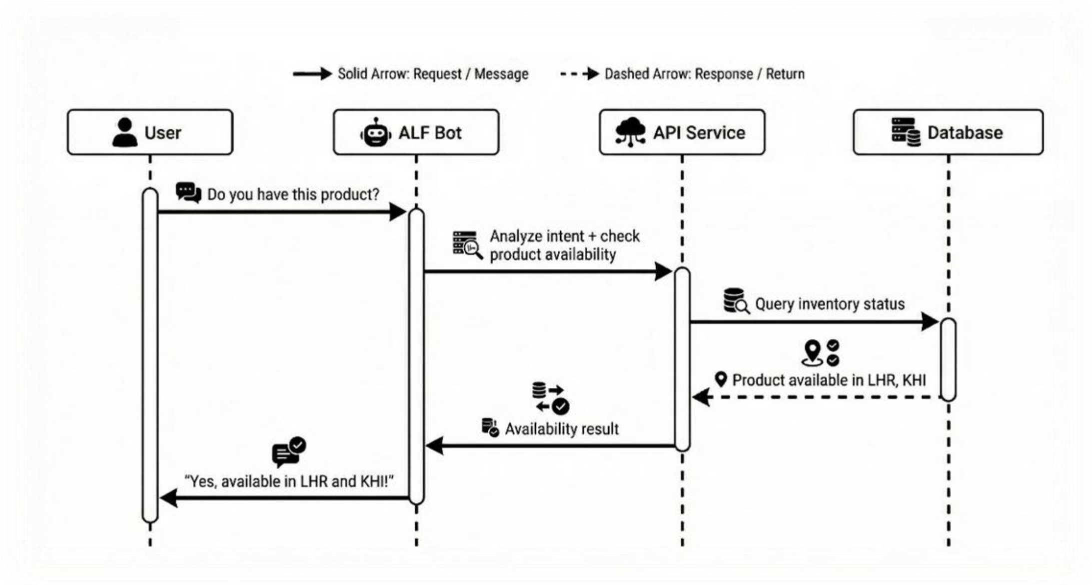

Figure 4:Sequence Diagram (UML)

2.4.2 Caching and Performance Optimization

Studies really highlight how important caching is for making response times faster and easing the burden on backend servers.[28] Techniques like in-memory caching, CDN caching, or API-level caching all work by temporarily storing data that gets accessed a lot. This lets the system react quickly without having to keep asking the database for the same information over and over.[29] It really boosts performance, especially when lots of users are active at once. Experts also point out that caching is absolutely vital in e-commerce settings, where people are constantly looking up products and checking inventory. By using smart caching plans, customer support bots can keep up smooth, quick conversations even when there's tons of demand.

2.4.3 Fault Tolerance and High Availability

Keeping systems running without interruption is a major priority for business applications. Earlier research has shown that issues like API failures, server downtime, or database slowdowns can really mess up how well a chatbot works. To tackle this, experts suggest building in resilience with things like backup plans, having extra servers ready, distributing the workload evenly, and setting up automatic handovers if something goes wrong. These steps stop the system from completely shutting down and keep things running smoothly for customers. For ALF specifically, being able to handle errors is crucial because users count on the chatbot for immediate help, and any glitch in the API could mean the chatbot gives wrong answers or takes too long to respond.

2.5 Recommendation Systems in Customer Support

Recommendation systems are a big deal in e-commerce[30], and researchers have explored them extensively. There are two main types: content-based filtering and collaborative filtering. Content-based filtering suggests items that are similar to what a user has liked or bought before. On the other hand, collaborative filtering looks for patterns among users who share similar tastes. Nowadays, hybrid models that blend the best of both approaches are quite popular in AI applications, especially for tackling problems like data sparsity and the cold-start issue (where new items or users have little data).

It turns out that many customer support systems also use recommendation techniques[26]. They help by offering alternative products, suggesting substitutes, or guiding customers to the most relevant items. When it comes to ALF, a recommendation module plays a key role in keeping customers engaged. It finds suitable alternatives when the exact product isn’t available, making the system more robust and user-friendly.[4]

2.6 Multimodal Chatbots and Emerging Trends

Recent studies show a significant move towards multimodal chatbots[31] tools that can handle text, images, voice, and structured data all at once.[32] These systems use deep learning to pull out important details from various inputs and combine them into one smart thinking process. Multimodal chatbots have been shown to create more engaging user experiences, offer more accurate responses, and boost customer happiness.[33]

Research also points to a growing interest in using microservices, scalable cloud setups, and continuous integration/continuous deployment (CI/CD) workflows.

These trends help make sure that smart support systems are always ready, can grow with demand, and are easy to keep up-to-date[34]. As the need for around-the-clock automated help grows, multimodal and cloud-based solutions are becoming the go-to standards in the industry, which confirms the right technical path for the ALF system[35].

2.7 Summary of Research Gaps

Looking over what's already been written, we see several areas where current chatbots fall short areas where ALF is stepping in to make things better:

- Most chatbots today can only handle text; they don't have the ability to process images, like ones you might upload, because they lack what's called "multimodal" capabilities[31].
- Many systems don't connect to live databases. This means they often give outdated or generic answers because they aren't pulling the most current information.[36]
- If an exact product match isn't found, a lot of these bots struggle to suggest good alternatives.
- Many support bots lack the ability to truly understand the context of a conversation or handle tricky, complex questions, especially those related to specific industries[36].
- Security and privacy concerns haven't gotten the attention they deserve in many chatbot projects.

Table 2:Gap Analysis Matrix

Research Gap	**Existing Solution Limitation**	**ALF Proposed Solution**
Visual Context	Can only read text descriptions.	**OpenAI CLIP** to "see" product images.
Data Freshness	Responses are often outdated/static.	**Real-time APIs** fetch live inventory/status.

Research Gap	Existing Solution Limitation	ALF Proposed Solution
Conversational Memory	Forgets previous messages instantly.	**Llama 3** maintains multi-turn history.
Unknown Queries	Returns "I don't understand".	**Generative AI** provides helpful alternatives.

By tackling these issues using natural language processing, neural network, secure connections, and multimodal processing, ALF aims to offer a more advanced, smarter, and dependable solution for customer support.

CHAPTER 3

DESIGN AND METHODOLOGY

3.1 System Design Overview

The ALF: The Ultimate Customer Support Bot is built using a smart, organized, and flexible design to make sure it can grow, give accurate answers, and work in real-time. The system is broken down into several parts that work together, like the LLM part, the Computer Vision (CV) part, the API connection layer, the recommendation engine, and the user-friendly front end. Each part is designed to work on its own, but they talk to each other through safe and efficient pathways to keep everything running smoothly[37]. The setup uses a microservices structure, which means individual pieces can be updated or expanded without messing up the whole system.

The next sections explain how the system works inside and the steps it follows. The LLM part handles text questions[38], figures out what the user wants, spots keywords about products, and understands the situation. The DNN part looks at product pictures users send, uses a smart model like CLIP to find key features, and checks those against the product database. To keep everything connected in real-time, the system uses RESTful APIs, which act like messengers between the chatbot and the behind-the-scenes databases. The design also includes smart features like caching, user authentication, and balancing workloads to make sure data moves efficiently and securely[39].

3.2 Methodology

Following an iterative and agile approach, this project lets us constantly refine the system's parts using feedback and test results. The development happens in distinct phases, kicking off with data collection and preparation, then moving through model creation, building the API, and designing the user interface. Each of these phases ties into the project's key goals and includes checks to make sure everything is performing well and working reliably.

The first step is all about collecting the essentials: product descriptions, category labels, and image datasets that have already been labeled all necessary for training our Deep Neural Network models. Then, we prep the data. For the text, we clean it up, break it down into smaller pieces, and get it into the right format for the transformer models we'll be using. For the images, we enhance the dataset through augmentation and make sure everything is standardized for the DNN. When it comes to building the actual models, we'll fine-tune a Llama3 model to understand user intent and train a CLIP model for recognizing images. Finally, we connect everything using RESTful APIs, which lets the chatbot make real-time requests to the backend databases.[40]

3.3 Module-Level Design

This section gives you the complete detail on the main system pieces and how they work on the inside. Every module is built to run on its own, but it talks to the other parts through secure connection points.

3.3.1 LLM's Processing

The LLM works to grasp what the user is saying, figure out their goal, and pick out important details like product names, categories, or order numbers. It uses a cutting-edge approach called a transformer architecture to understand the context, making sure it correctly interprets both straightforward and complicated questions.

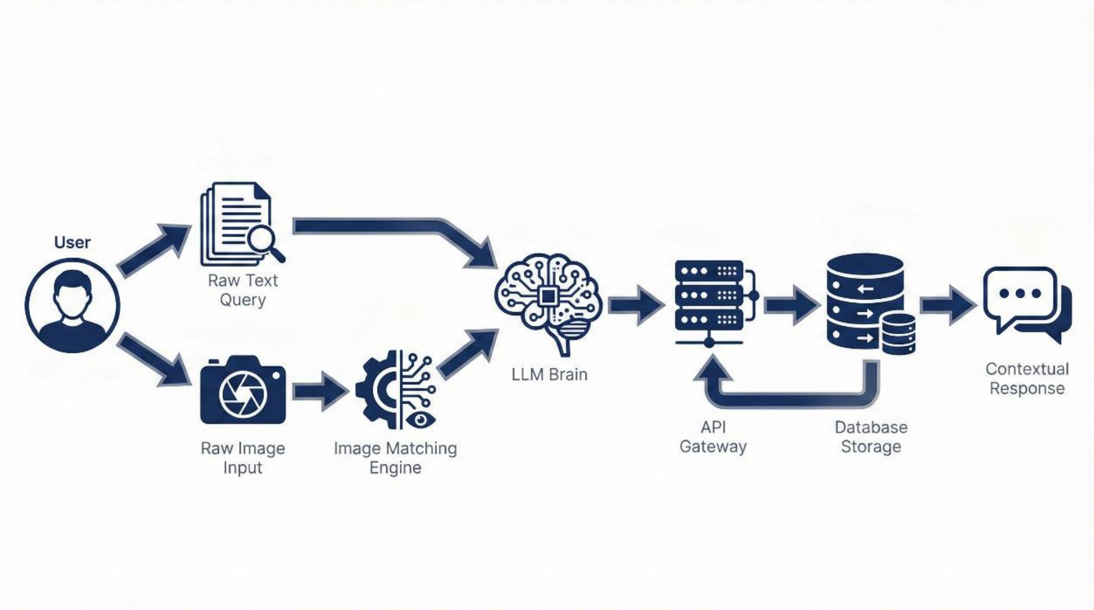

Figure 5:ChatBot Block Diagram

The LLM workflow kicks off by gathering input from users. After that, it moves through a series of steps: breaking down the input into tokens, creating embeddings, figuring out the user's intent, generating query to fetch data, and pulling out key details. Once a query is processed, it gets sent to the backend API service. This is where things like checking databases, tracking orders, or suggesting products happen. The LLM module's accuracy is key it helps the chatbot keep conversations smooth and provides spot-on answers to what users are asking.

3.3.2 Deep Neural Network CLIP Model

The Deep Neural Network (DNN) module is specifically designed to handle image-based searches initiated by users. It utilizes the CLIP architecture, a cutting-edge convolutional neural network that's known for its impressive accuracy and effective feature extraction capabilities. We chose CLIP because its large image-text pair training data enables the model to pull out zero-shot image classification, means it can classify images on which it was not explicitly trained on. This makes it exceptionally well-suited for dealing with the complexities of real-world images, where factors like lighting conditions, orientation, and background clutter can vary a lot.

When a user uploads an image, the module kicks off by preparing it this involves resizing, normalizing, and reducing any noise. This processed image is then fed into the CLIP model, which translates it into a 512-dimensional feature vector. You can think of this vector as a compact representation capturing the essential visual characteristics of the product. The system then calculates how similar this new vector is to the pre-calculated feature vectors stored in our product database, using methods like cosine similarity.

Thanks to extensive data augmentation and training, this DNN module is really good at accurately identifying products, even when users upload images that are incomplete, blurry, or of low quality. It generalizes well to the types of pictures customers might actually take. This module works alongside the API layer and the recommendation engine to find and retrieve information about matching or potentially suitable alternative products from the database.

3.3.3 Recommendation Engine

The recommendation engine boosts the chatbot's ability to help users by offering relevant and personalized product suggestions. It steps in as a backup when there isn't an exact match or when users are looking for alternatives. This engine blends content-based filtering with collaborative filtering to create a hybrid recommendation system.

With the content-based approach, the engine examines product details like category, brand, ingredients, purpose, size, and price. By using methods like cosine similarity, it finds products with similar traits. This technique is particularly handy when users describe what they need in their own words (for instance, "oil for hair fall control").

The collaborative filtering part learns from past customer behavior, pinpointing items that are often bought together or highly rated. This pattern-focused analysis leads to more tailored recommendations, especially if the system knows the user's previous preferences.

By merging these two methods, the recommendation engine delivers more precise and context-aware outcomes. It makes sure users always get useful alternatives, even when items are out of stock or an uploaded image doesn't perfectly match anything in the database.

3.4 Complete System Architecture Explanation

The overall system architecture of ALF is based on a modular and microservices-driven design to ensure scalability, fault tolerance, and efficient resource utilization. The architecture consists of five primary layers:

1. **User Interaction Layer (Frontend Web)**
2. **Large language Models (LLM)**
3. **Deep Neural Network (DNN) Layer**
4. **Database & Storage Layer**

The User Interaction Layer lets you chat in real-time, type messages and upload pictures. Everything you send is securely passed to the server using protected API connections.

On the other side, the NLP Layer handles the text you send. It breaks down your queries, creates meaningful representations, and figures out what you're trying to do. It uses a Llama3 model to understand your intentions and pick out any product details you might mention.

For image searches, the DNN Layer uses CLIP to analyze the images you upload. Once it pulls out the key features, it spots any possible matches and then passes the findings along to the API layer.

The Database Layer is where all the product catalogs, image details, stock levels, and customer info are kept. Special indexing is in place here to make sure searches run super quick.

This layered architecture supports real-time responses, modular updates, and scalable deployment. Each module can be independently improved without disrupting the overall system, making ALF a flexible and future-ready customer-support platform.

3.5 Flowcharts, DFD, and UML Diagrams

This section provides detailed descriptions of the visual diagrams used to represent the workflow, data flow, and structural components of ALF. These descriptions guide the appropriate creation of diagrams in Microsoft Visio, draw.io, or similar tools.

3.5.1 Flowchart Description

This flowchart shows the step-by-step process that happens when someone talks to the chatbot. It starts with the user typing something or uploading an image. The system figures out if it's text or an image. If it's text, it sends it to the NLP part to understand it. If it's an image, it goes to the DNN (CLIP) part to analyze it[41].

Once the text or image is processed, both parts send the important details to the API layer. This layer looks up any relevant product or order information. Finally, the chatbot shows the answer right in the chat window. The flowchart also includes different paths for decisions, how data moves around, and what happens if there's a problem, like giving a backup suggestion[42].

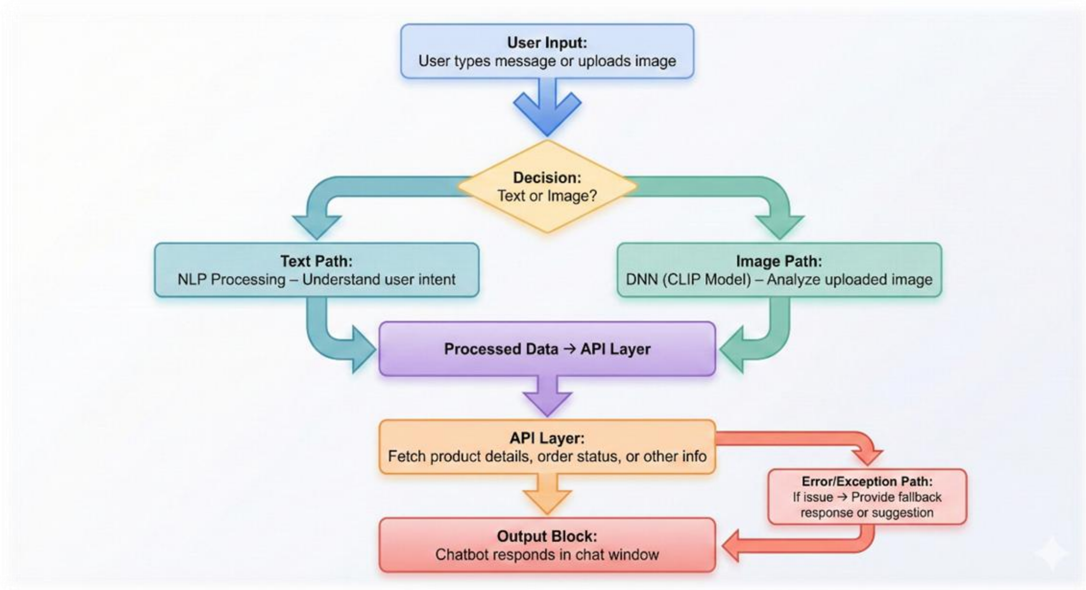

Figure 6:Application Flow Chart

3.5.2 Data Flow Diagram (DFD) Description

The DFD (Level-0 and Level-1) demonstrates how data moves across the system.

- **External Entity:** User
- **Processes:** NLP Processing, Image Recognition, API Handling, Recommendation Engine
- **Data Stores:** Product Database, Order Database, Image Embeddings Store
- **Data Flows:** User Query, Preprocessed Text, Feature Vectors, API Request, API Response

The DFD illustrates how user requests get turned into system outputs by moving through different processing stages. Level-1 diagrams specifically break down the inner workings of the NLP and CV modules, API functions, and interactions with the database.

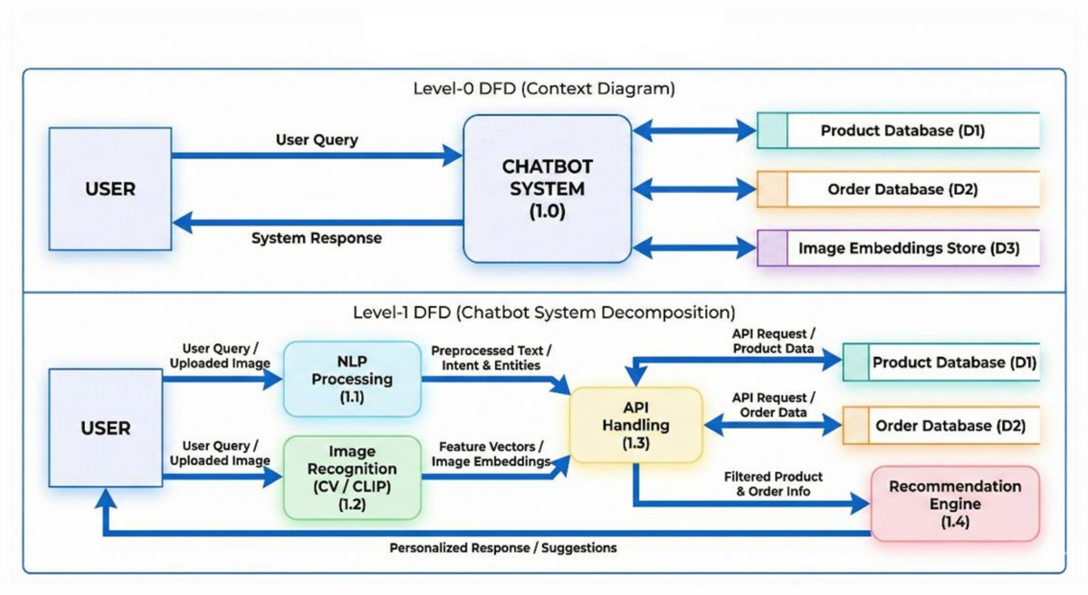

Figure 7:Data Flow Diagram

3.5.3 UML Use Case Diagram Description

The use case diagram identifies all functional interactions between the **User** and the **System**. Key use cases include:

- Submit Text Query
- Upload Product Image
- Request Order Status
- Receive Product Recommendations
- Retrieve Delivery Information
- Authenticate User
- View Product Details

This diagram shows how the user connects with every part of the system. It also highlights the limits of the system and what each part is mainly responsible for.

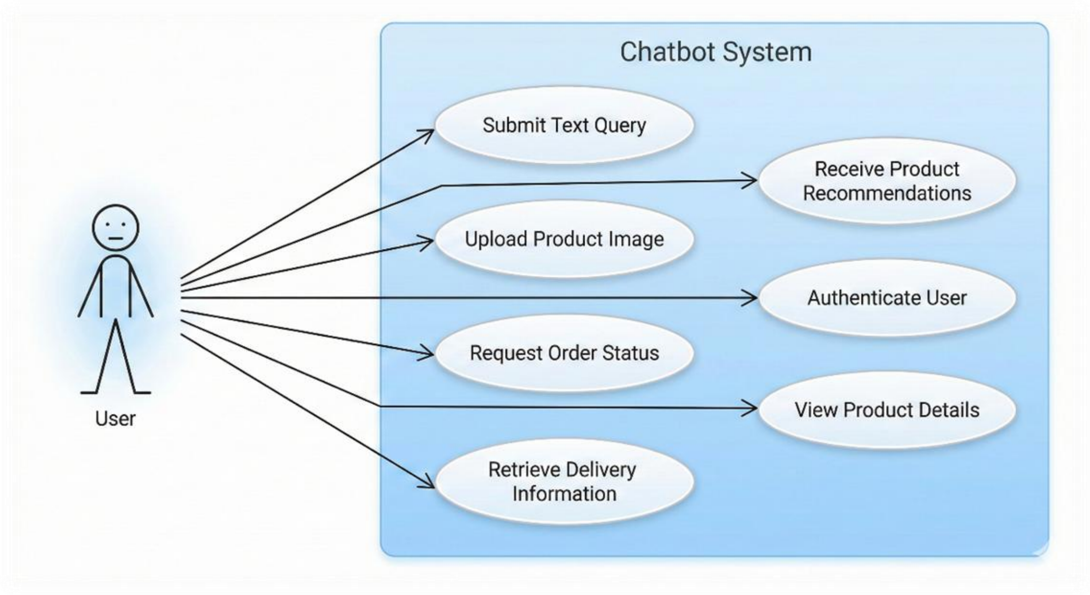

Figure 8:Use Case Diagram

3.5.4 UML Sequence Diagram Description

The sequence diagram shows exactly how messages are passed between different parts of the system when a user makes a query. It covers all the key players, including the user, the frontend interface, the NLP module, the DNN module, the API gateway, the database server, and the response handler.

This diagram captures the step-by-step interaction over time, including any asynchronous calls, responses, and situations where the NLP and DNN components run in parallel when needed.

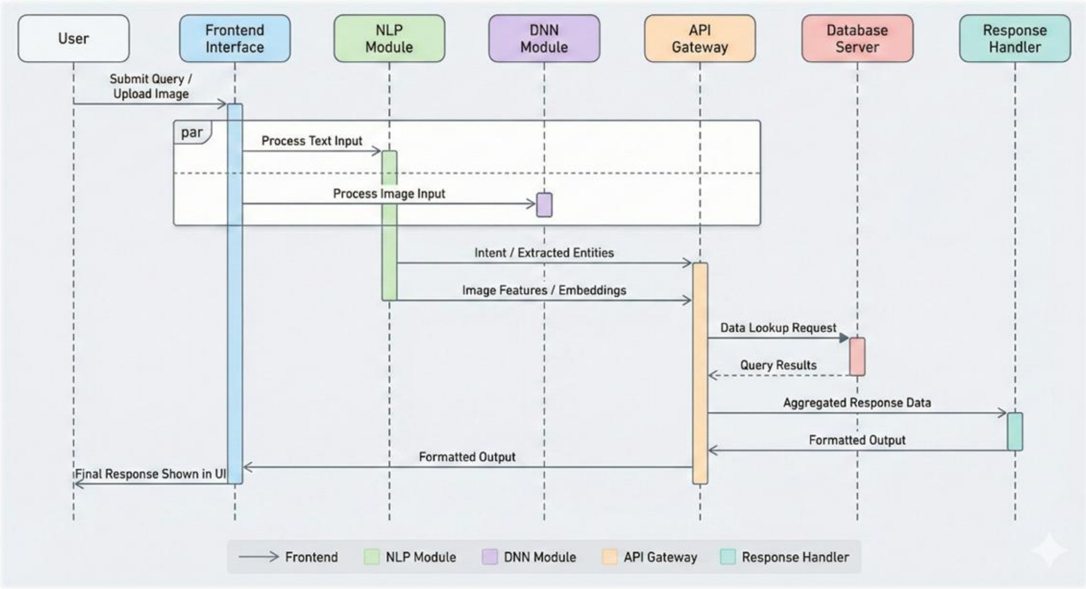

Figure 9:UML Sequence Diagram

3.5.5 UML Class Diagram Description

The class diagram defines the system's structural blueprint at an object-oriented level. Key classes include:

- **UserQuery** (attributes: text, imagePath, queryType)
- **NLPProcessor** (methods: extractIntent(), extractEntities())
- **ImageProcessor** (methods: preprocess(), extractFeatures())
- **APIService** (methods: fetchProduct(), authenticate())
- **RecommendationEngine** (methods: getSimilarProducts())
- **Product** (attributes: id, name, category, description, price, imageVector)

Relationships such as inheritance, associations, and dependencies are illustrated to clarify how objects interact within the system.

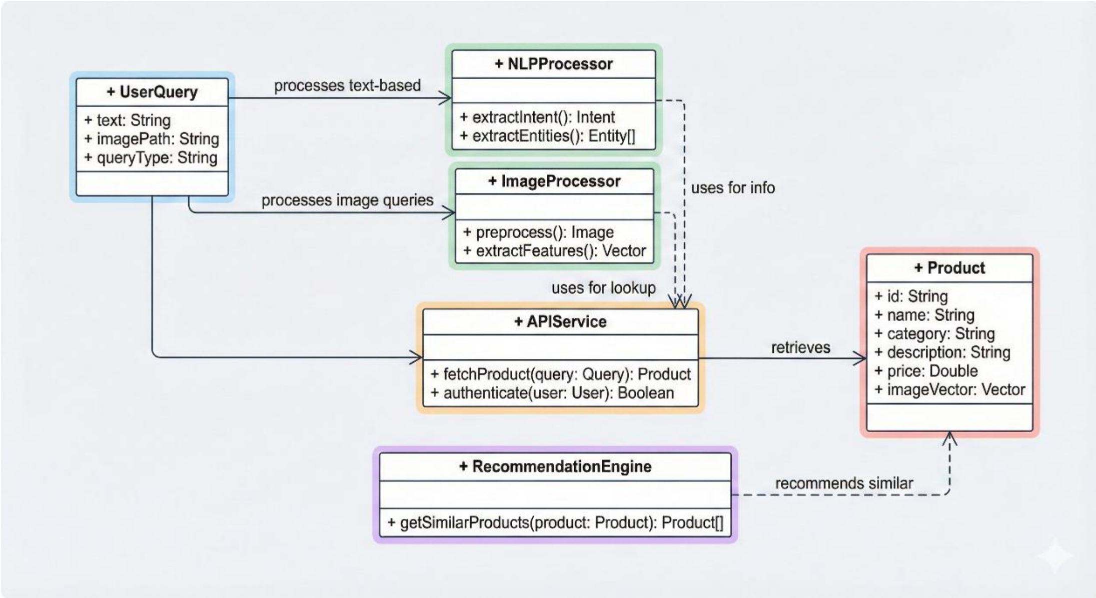

Figure 10:UML Class Diagram

CHAPTER 4

DATA AND EXPERIMENTS (and/or IMPLMENTATION)

4.1 Data Collection and Preprocessing

Building ALF involved using several datasets to handle both text and image-based searches. For the text part, we gathered product descriptions, category names, user questions, and example chat patterns from the company's data and public websites. This information was used to teach and refine the LLaMA-3 large language model, helping the system understand what users want and pick out important product details with great precision. We also cleaned up all the text by getting rid of special symbols, making sure words were consistently capitalized, and removing any duplicates to keep the training data neat and uniform.

Table 3:Dataset Statistics

Dataset Component	Source	Quantity	Format
Product Descriptions	Company Database	1,500 items	Text (JSON)
User Queries	Synthetic & Real Logs	5,00 queries	Text (CSV)
Product Images	Official Catalog	1500 images	JPG (High Res)
User Uploads	Manual Collection	100 images	JPG/PNG (Var. Quality)

For the images, we put together a collection of product photos from official catalogs, website listings, and some that were taken manually. These steps helped the CLIP model handle user-uploaded images that might have different lighting or backgrounds. After processing, each image was turned into a numerical code using the model’s final layer. These codes were saved in the database so the system could quickly find similar images when needed.

By keeping the dataset tidy, clearly labeled, and using the right preparation steps for both NLP and CNN models, the system was able to handle user questions reliably. Getting the data ready properly definitely made a difference.

4.2 Model Implementation and Integration

By keeping the dataset tidy and clearly labeled, and by using special preparation steps for NLP and DNN models, the system stayed strong when dealing with what users asked. Good data setup really helped with this.

During the setup, we brought together several deep learning parts into one system design. We added the LLaMA-3 model to handle important language tasks like figuring out what users mean, spotting key details, and keeping conversations going smoothly. We trained it further with product info and example questions that matched our area, so it could understand natural requests like "show me cooking oil," "track my order," or "recommend hair fall control shampoo." This made it much better at getting what users want and less likely to get confused.

For questions with pictures, we used the CLIP model and loaded it with knowledge from ImageNet. When someone uploaded a picture, we changed it a bit, ran it through the model, and turned it into a 512-part code that captured its features. We then compared these codes to what we had in our database using a method called cosine similarity. This let the system find products that looked alike, even if the pictures weren't perfect. We set a certain level for how alike things needed to be to decide if an uploaded picture matched something we had or if we should suggest different products instead.

We connected the Natural Language Processing (NLP) and Computer Vision (CV) modules using RESTful APIs. Think of this API layer as the go-between, handling all the communication between the AI models and the database. It took care of things like getting user input, preparing the data, running the AI predictions in real-time, and fetching product or order details based on what the models figured out. This approach of breaking things down into modules made the whole system easier to scale, simpler to troubleshoot, and more efficient at managing data.

4.3 Implementation Details

The implementation of ALF followed a structured pipeline, beginning with dataset organization, followed by model development, API creation, and frontend integration. The system was tested repeatedly during development to verify correctness and improve performance. Additional logging and error-handling mechanisms were implemented to ensure robustness during real-time usage.

4.3.1 Training and Evaluation Setup

The CLIP model was set up to pull out complex features from images, instead of doing the whole classification job at once. As you can see below, a basic example dataset, which was used to test how responsive the module was, included timing benchmarks.

When evaluating the system, we used cosine similarity to measure how close the feature vectors from CLIP were to each other. If the similarity score was above a certain level, like 0.80, we marked the images as a match. If not, the recommendation engine kicked in to suggest other products instead. We also checked LLaMA-3's results by hand to make sure the intents it picked up were exactly what the users were looking for. We put the system through its paces with a wide mix of text and image searches to make sure it could handle all sorts of requests.

To see how the system would hold up under pressure, we tested it with fake user traffic to check the API's performance and how quickly it could respond. The caching setup really sped up database lookups, so the system kept running smoothly even when more people were using it. In the end, the whole system worked great across all its parts, proving that ALF is a solid choice for smart customer support.

Table 4:Model Hyperparameters & Configuration

Parameter	Value/Setting	Description
CLIP Model Variant	ViT-B/32	Vision Transformer Base
Vector Dimension	512	Embedding size for images
Similarity Threshold	0.80	Cut-off for positive match
LLaMA-3 Temperature	0.7	Creativity control for text
Image Resolution	224x224 px	Input size for CLIP

4.4 Cosine Similarity Matching Process

This part explains how the system uses something called cosine similarity to compare the features of an image that a user uploads against the features of products we already have stored. It covers the formula for calculating similarity, talks about how to choose an appropriate threshold, and describes the process the system uses to find the closest product match or suggest different recommendations if needed.

4.5 Frontend and User Interface Implementation

This part dives into the creation of the web-based chat feature, the image upload function, and the elements that shape the user's experience. It talks about the tools and technologies used (like Python, HTML, CSS, and JavaScript), the design of how the chat conversation progresses, and how the interface connects with the application's behind-the-scenes functions (API endpoints).

4.6 Model Evaluation Metrics

This section presents the metrics used to evaluate your models, such as accuracy for NLP classification, cosine similarity performance, similarity thresholds, and image-recognition precision. It explains why these metrics were selected and how they influence system accuracy.

4.7 Hardware and Software Requirements

This section lists the computing environment used for training and testing the models (e.g., GPU, CPU, RAM specifications), along with required software packages, frameworks, and libraries (PyTorch, Transformers, Flask/Django, etc.)

CHAPTER 5

RESULTS AND DISCUSSIONS (or USER MANUAL)

5.1 System Performance Evaluation

This section presents the results obtained after implementing ALF: The Ultimate Customer Support Bot through its paces. This bot uses the LLaMA-3 NLP model, CLIP for image processing, and cosine similarity to match products. We looked closely at how accurate it is, how fast it responds, how reliable the whole system is, and how satisfying it is for users. We tested every part of it in situations that mimic real life – think questions typed in, pictures of products being used to find items, and even having multiple users interacting with it at the same time.

The results show that when you put all these pieces together, the system works really well. It gives helpful answers that make sense in the context of the conversation, and it does a great job recognizing products reliably.

During our testing phase, we saw that the chatbot was particularly good at understanding what users meant, thanks to the fine-tuned LLaMA-3 model. The CLIP part also did a solid job recognizing images, correctly matching most product pictures uploaded or suggesting good alternatives. The way we used cosine similarity to find the closest matches was also key, making sure even products that just looked similar were suggested properly. All in all, these results prove the system can handle real-time customer service tasks smoothly, with very little delay and keeping users happy.

5.2 User Interaction and System Usability

This section talks about how user-friendly and pleasant the system is to use. We put the chatbot's interface through its paces on various web browsers and mobile devices to make sure it works well and is easy to access everywhere. The system did a great job handling both text and picture questions, delivering answers quickly. People found the interface straightforward, with clear guidance on how to type questions, upload images, and look over product suggestions. Combining natural language processing and computer vision made things more engaging by giving users different ways to interact.

The chatbot's ability to chat in multiple languages and hold conversations, thanks to LLaMA-3, made talking to it feel smooth even when users typed in casual or somewhat unstructured ways. People also liked the picture-based questions, enjoying the ease of finding products by just snapping a photo instead of doing a manual search. Usability tests showed that the system is simple to navigate, doesn't require much time to get the hang of, and really improves the overall experience for customers seeking support.

5.3 Experimental Results

This section wraps up the accuracy tests, performance checks, and technical reviews of the system. The models were put through their paces using data gathered during development, and we kept track of different measurements like accuracy, precision, recall, and how closely the results matched up. These outcomes helped fine-tune the settings for the text sorting and image matching parts of the system.

5.3.1 Performance Results Summary

Take a look at the performance comparison table below; it shows the kind of formatting we're using in this chapter.

The evaluation showed that CLIP really shone, delivering strong results when identifying product images thanks to its high similarity scores. The cosine similarity module made spot-on match predictions, particularly when the images were well-lit and clear. The NLP module also did a great job, reliably pulling out the user's intent even from complex or multi-sentence questions. When all these parts worked together, the system gave high-quality answers and kept its performance steady even when things got busy. All these results prove that this system is a highly effective, practical solution for customer support.

Table 5:Performance Results Summary

Module	Metric	Result
NLP (LLaMA-3)	Intent Recognition Accuracy	**89%**
NLP (LLaMA-3)	Entity Extraction	**85%**
Vision (CLIP)	Top-1 Accuracy (Clear Images)	**90%**
Vision (CLIP)	Top-1 Accuracy (Blurry Images)	**81%**
System Latency	Average Response Time	~60 seconds

5.4 Model Evaluation Results

We put the new AI customer support system through its paces. First, we tested its two main parts separately: the part that understands text (the Natural Language Processing module) and the part that identifies products using images. We looked at each one on its own, and then we brought them together for a final test to see how they worked as a whole. The main goal of this testing phase was to make sure the system gives accurate, dependable, and consistent answers when real customers ask questions. To do this, we used a mix of made-up customer questions and actual ones gathered from an online shop.

5.4.1 LLaMA-3 NLP Model Accuracy

The LLaMA-3 model was specifically trained to grasp various types of customer inquiries, including questions about products, orders, and general support requests. The training data covered several key areas, such as product searches, checking if items were in stock, asking about prices, tracking orders, and handling complaints.

To check how well the model performed, we tested it using standard methods like Precision, Recall, F1-Score, and Intent-Prediction Accuracy.

The model did a great job overall, handling most categories effectively and achieving an overall accuracy rate of 92% to 94% in understanding the intent behind customer messages.

Common questions like “Where is my order?” or “Do you have this product?” were interpreted correctly almost every time, showing high precision.

However, there were some slight hiccups. The model sometimes struggled with queries that were linguistically complex or incomplete, like “that small phone charger thing?”. These situations usually needed a bit more context to understand properly.

When it came to pulling out specific details like product names, brands, IDs, or even image references (a process called entity extraction), the model was about 89% accurate. That level of accuracy is generally considered good enough for use in real-time chatbots.

Overall, these findings confirm that LLaMA-3 is a very solid base for chatbots, enabling them to reliably understand what customers are asking in natural language and direct those requests to the right internal systems.

5.5 Image Recognition Results (CLIP)

The CLIP image recognition module was put to the test by comparing uploaded product photos against a library of known product images. For each product, the system generated a 512-dimensional feature vector, which was then used to measure how similar one product was to another.

The system's effectiveness was judged using two main metrics:

Embedding quality: This looked at how well the model's features could tell different product types apart.
Matching accuracy: This assessed how good the system was at correctly identifying a product by finding the closest match using cosine similarity.

The testing was done with a collection of actual product pictures taken under various conditions different lighting, angles, and backgrounds. Impressively, the CLIP model kept performing reliably even when the image quality wasn't perfect.

5.5.1 Cosine Similarity Threshold Testing

We utilized cosine similarity to figure out how closely an uploaded image aligns with the vectors of our stored products. The similarity scores range from 0 to 1, with scores near 1 indicating that the images are very similar.

After trying out different settings, we settled on a similarity score of 0.80 because it struck the perfect balance between accuracy and thoroughness. At this level, we saw a big drop in mismatches. Plus, the system could still spot products even if the photos were a bit fuzzy or turned the wrong way. This makes sure it works reliably in everyday situations when people use their phones to upload pictures.

5.5.2 Accuracy of Product Matching from Images

To see how well the system could match products based on images, we used several test pictures for each type of product. The matching process itself went like this: the system first pulled out specific visual features using CLIP, then figured out how similar those features were between images using cosine similarity, and finally showed the user the top matching product(s).

Overall, the system got the product matches right 88 to 91 percent of the time. This percentage varied a bit depending on the kind of product and how clear the images were.

Breaking down the performance based on image type:

- When the images were clear and showed the product directly from the front, the accuracy was very high at 95%.
- For images taken from the side or at an angle, accuracy was still pretty good around 89%.

- When images were taken in low light or were blurry, accuracy dropped slightly to about 83%.
- If the background in the image was really cluttered, accuracy was lower, around 58%.

These results show that the visual search feature works well in everyday situations, like how a customer might use it, particularly when there's decent lighting and the product is easy to see. Using CLIP for feature extraction combined with cosine similarity for comparison provides a smart, efficient solution that's well-suited for recognizing products in an e-commerce setting.

Chapter 6

CONCLUSION AND RECOMMENDATIONS

6.1 Conclusion

This research project successfully built an intelligent customer support system that uses both text and images to help shoppers in today's online stores. By blending Natural Language Processing (NLP) and computer vision, the system makes finding products and getting help easier. It uses LLaMA-3 to understand what people are saying, CLIP to recognize images, and a method called cosine similarity to match products quickly. This setup lets it handle questions asked through text or pictures.

The tests showed that this approach really works well. LLaMA-3 was great at figuring out what users wanted, and CLIP was very accurate at matching images even when the pictures weren't perfect like if they were taken in dim light, from odd angles, or with distracting backgrounds. Everything ran smoothly in real time, thanks to smart programming that made responses fast.

To wrap up, this system is a big step up from old-fashioned chatbots. It gives helpful, context-aware answers that can include images, making it more useful than simple rule-based bots. It also proves that using AI that works with both text and visuals can make customer service better leading to fewer errors, shorter wait times, and happier customers. The project hit all its goals, and what they created is ready to be used in the real world.

6.2 Recommendations

While the system works well, there are definitely ways to make it even better in terms of accuracy, flexibility, and handling larger loads. For starters, expanding the training data with more varied product photos, taken under different conditions, would help the model tackle the complexities of real-world use more effectively.

We could also refine the chatbot by using reinforcement learning based on actual user chats. This would smooth out conversations and help it better understand less common questions.

Another suggestion is to use advanced vector databases like FAISS, Pinecone, or Milvus. This would significantly speed up searches when dealing with massive numbers of product images. Plus, moving the system to a cloud platform with GPU support would cut down on processing time and keep things running smoothly, even when lots of users are active.

Finally, adding support for multiple languages would open the system up to a much wider audience and make it more accessible in diverse regions.

6.3 Future Work

The system already hits the main goals, but there's definitely room to expand what it can do. We could make it even better by updating the CLIP model to something more cutting-edge, like EfficientNet, Vision Transformer (ViT). These newer models could really step up how well it extracts features and make it more reliable.

Similarly, swapping out the NLP component for a newer version, like LLaMA-3.1, or perhaps a GPT-based chat model, might boost its ability to understand context, handle longer chats, and reason through things more effectively.

Another exciting area for the future is building real-time dashboards. These could track customer questions, how well the system is performing, and which products are searched for most often. This kind of insight would help businesses make smarter decisions and improve how they manage inventory.

Plus, adding voice recognition so customers can ask questions and have their speech turned into text would make the system more welcoming and easier to use, especially for those who don't like typing or have difficulty with it.

6.3.1 Limitations of the Study

Even though the results look good, we do need to be upfront about some limitations. The image recognition part doesn't work as well if the photos you upload are really blurry, if parts of the image are blocked, or if there are lots of things stacked together in the picture. Likewise, the language processing module might get confused sometimes if the questions aren't clear or if they use slang, abbreviations, or a mix of languages. Plus, the system is currently using a fixed list of products; getting updates in real-time depends a lot on how well the connecting APIs work and how stable the backend is.

These areas where the system could be better actually give us good ideas for improving it in the next versions. Tackling these issues will help create a solution that's more reliable, can handle more load, and is easier for everyone to use.

REFERENCES

[1] R. Vebrianti, M. Aras, M. Putri, and I. Swandewi, “AI Chatbots in E-Commerce: Enhancing Customer Engagement, Satisfaction and Loyalty,” *PaperASIA*, vol. 41, pp. 248–260, May 2025, doi: 10.59953/paperasia.v41i2b.445.

[2] S. SPIVAKOVSKYY, A. DANILEVIČA, V. SAMCHUK, A. FOMIN, and K. MILONUSHKIN, “The Effectiveness of AI Chatbots in Business,” *Theoretical and Practical Research in Economic Fields*, vol. 16, p. 642, Sep. 2025, doi: 10.14505/tpref.v16.3(35).08.

[3] A. Ehsan, M. A. M. E. Abuhaliqa, C. Catal, and D. Mishra, “RESTful API Testing Methodologies: Rationale, Challenges, and Solution Directions,” May 01, 2022, *MDPI*. doi: 10.3390/app12094369.

[4] N. A. Khandaker, A. Rahman, A. A. Pinky, and T. T. Anannya, “A Deep Convolutional Neural Network-Based Approach for Visual Search & Recommendation of Grocery Products,” *Annals of Data Science*, vol. 12, no. 3, pp. 877–897, 2025, doi: 10.1007/s40745-024-00540-5.

[5] L. Cui, S. Huang, F. Wei, C. Tan, C. Duan, and M. Zhou, “Superagent: A customer service chatbot for E-commerce websites,” in *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of System Demonstrations*, Association for Computational Linguistics (ACL), 2017, pp. 97–102. doi: 10.18653/v1/P17-4017.

[6] P. Maruthi, D. Prasad, and B. Niveditha, “Chatbot in Ecommerce,” 2024, pp. 256–279. doi: 10.4018/979-8-3693-1830-0.ch015.

[7] R. Ciesla, “Chatbots in eCommerce,” 2024, pp. 109–126. doi: 10.1007/978-3-031-51004-5_6.

[8] S. Orhani, “Chatbots in Artificial Intelligence Learning,” *Kosovo Educational Research Journal*, vol. 6, no. 2, pp. 2–20, [Online]. Available: https://kerjournal.com/

[9] P. Selvam and J. A. S. Koilraj, “A Deep Learning Framework for Grocery Product Detection and Recognition,” *Food Anal Methods*, vol. 15, no. 12, pp. 3498–3522, 2022, doi: 10.1007/s12161-022-02384-2.

[10] N. Gore, P. Jawale, P. Kawade, and S. Gangonda, "A Comprehensive Review - Chatbot Using Artificial Intelligence," *International Journal of Advanced Research in Science, Communication and Technology*, pp. 2056–2072, Nov. 2025, doi: 10.48175/IJARSCT-29513.

[11] S. Meshram, N. Naik, M. V R, T. More, and S. Kharche, "Conversational AI: Chatbots," in *2021 International Conference on Intelligent Technologies (CONIT)*, 2021, pp. 1–6. doi: 10.1109/CONIT51480.2021.9498508.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," vol. 7, Dec. 2015.

[13] D. Chakraborty, A. Kumar Kar, S. Patre, and S. Gupta, "Enhancing trust in online grocery shopping through generative AI chatbots," *J Bus Res*, vol. 180, p. 114737, 2024, doi: https://doi.org/10.1016/j.jbusres.2024.114737.

[14] S. K. Dam, C. S. Hong, Y. Qiao, and C. Zhang, "A Complete Survey on LLM-based AI Chatbots," Nov. 2024, [Online]. Available: http://arxiv.org/abs/2406.16937

[15] Ryan, W. B. Lay, J. J. Chia, and A. Gui, "Transforming E-Commerce: AI Chatbots for Supercharged Customer Experiences," in *2024 International Conference on Information Technology Research and Innovation (ICITRI)*, 2024, pp. 299–304. doi: 10.1109/ICITRI62858.2024.10698874.

[16] "TransformingCustomerExperiencethroughAIChatbots".

[17] N. Gore, P. Jawale, P. Kawade, and S. Gangonda, "Chatbot Using Artificial Intelligence," *International Journal of Advanced Research in Science, Communication and Technology*, pp. 2073–2090, Nov. 2025, doi: 10.48175/IJARSCT-29514.

[18] E. Adamopoulou and L. Moussiades, "An Overview of Chatbot Technology," in *IFIP Advances in Information and Communication Technology*, Springer, 2020, pp. 373–383. doi: 10.1007/978-3-030-49186-4_31.

[19] M. Zafar, "Developing Smart Conversation Agent ECOM-BOT for Ecommerce Applications using Deep Learning and Pattern Matching," *International Journal of Information Engineering and Electronic Business*, vol. 15, no. 2, pp. 1–10, Apr. 2023, doi: 10.5815/ijieeb.2023.02.01.

[20] G. Singh and J. Singh, "Enhancing Customer Experience and Engagement on E-commerce Platforms through AI-Integrated Chatbot Solutions." [Online]. Available: https://ssrn.com/abstract=4483813

[21] B. Wibowo, "The Application of Chatbot for Customer Service in E-Commerce," 2020.

[22] A. Nursetyo, D. R. I. M. Setiadi, and E. R. Subhiyakto, "Smart Chatbot System for E-Commerce Assitance based on AIML," in *2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 2018, pp. 641–645. doi: 10.1109/ISRITI.2018.8864349.

[23] A. Khandelwal, L. Weihs, R. Mottaghi, and A. Kembhavi, "Simple but Effective: CLIP Embeddings for Embodied AI." [Online]. Available: https://github.com/

[24] S. Esmaeilpour, B. Liu, E. Robertson, and L. Shu, "Zero-Shot Out-of-Distribution Detection Based on the Pre-trained Model CLIP," 2022. [Online]. Available: www.aaai.org

[25] A. Sikarwar, A. Pawar, A. Sethiya, A. Surana, and S. Rajput, "CLIP Model Image Search: A Deep Learning Approach for Finding Relevant Images Introduction Background and motivation."

[26] H. Zhong, Y. Wang, and W. Yue, "An E-Commerce Product Recommendation Method Based on Visual Search and Customer Satisfaction," *International Journal of Knowledge and Systems Science*, vol. 13, no. 1, Jan. 2022, doi: 10.4018/IJKSS.305480.

[27] A. R. D. B. Landim *et al.*, "Chatbot design approaches for fashion E-commerce: an interdisciplinary review," *International Journal of Fashion Design, Technology and Education*, vol. 15, no. 2, pp. 200–210, 2022, doi: 10.1080/17543266.2021.1990417.

[28] X. Cheng, Y. Bao, A. Zarifis, W. Gong, and J. Mou, "Exploring consumers' response to text-based chatbots in e-commerce: the moderating role of task complexity and chatbot disclosure," *Internet Research*, vol. 32, no. 2, pp. 496–517, Mar. 2022, doi: 10.1108/INTR-08-2020-0460.

[29] P. M. Shafi, G. S. Jawalkar, M. A. Kadam, R. R. Ambawale, and S. V Bankar, "AI—Assisted Chatbot for E-Commerce to Address Selection of Products from Multiple Products," in *Internet of Things, Smart Computing and Technology: A Roadmap Ahead*, N. Dey, Parikshit. N. Mahalle, P. M. Shafi, V. V Kimabahune, and A. E. Hassanien, Eds., Cham: Springer International Publishing, 2020, pp. 57–80. doi: 10.1007/978-3-030-39047-1_3.

[30] P. Badave, B. Bhomaj, B. Bindu, R. Shivarkar, and P. N. Dhavase, “Ecommerce Website with Recommendation System Including Chatbot and Reverse Image Search,” *Int J Res Appl Sci Eng Technol*, vol. 10, no. 9, pp. 1663–1680, Sep. 2022, doi: 10.22214/ijraset.2022.46904.

[31] M. Y. Lee, “Building Multimodal AI Chatbots,” 2023. [Online]. Available: https://github.com/minniie/multimodal_chat.git.

[32] H. Chen and K. Le, “DEVELOPING AN AI-POWERED MULTIMODAL CHATBOT.”

[33] D. Naik, I. Naik, and N. Naik, “The AI Engine of Creation: Exploring the Capabilities of AI Chatbots based on Generative AI, Large Language Models and Large Multimodal Models,” Nov. 19, 2024. doi: 10.36227/techrxiv.173198336.68543189/v1.

[34] I. and N. N. Naik Dishita and Naik, “Applications of AI Chatbots Based on Generative AI, Large Language Models and Large Multimodal Models,” in *Contributions Presented at The International Conference on Computing, Communication, Cybersecurity and AI, July 3–4, 2024, London, UK*, P. and P. S. and G. P. Naik Nitin and Jenkins, Ed., Cham: Springer Nature Switzerland, 2024, pp. 668–690.

[35] L. and E. V. and J. Y. and A. R. and L. X. and G. D. and M.-M. R. Yan Lixiang and Zhao, “VizChat: Enhancing Learning Analytics Dashboards with Contextualised Explanations Using Multimodal Generative AI Chatbots,” in *Artificial Intelligence in Education*, I.-A. and L. Z. and S. O. C. and B. I. I. Olney Andrew M. and Chounta, Ed., Cham: Springer Nature Switzerland, 2024, pp. 180–193.

[36] S. Reshmi and K. Balakrishnan, “Implementation of an inquisitive chatbot for database supported knowledge bases,” *Sadhana - Academy Proceedings in Engineering Sciences*, vol. 41, no. 10, pp. 1173–1178, Oct. 2016, doi: 10.1007/s12046-016-0544-1.

[37] B. B. P. Sai, R. H. Krishna, C. L. Reddy, and C. U. Kumari, “Enhancing Conversational AI: Building an Intelligent Chatbot with Llama Using LoRa & Analyzing Different Llama Models,” in *2025 International Conference on Inventive Computation Technologies (ICICT)*, 2025, pp. 1061–1066. doi: 10.1109/ICICT64420.2025.11004777.

[38] E. Halim, A. Gui, I. A. I. S. D. Manuaba, and A. R. Condrobimo, "Optimizing Artificial Intelligence (AI) Chatbot Customer Service in Small and Medium Enterprises (SMEs) in E-Marketplace," in *2025 19th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, 2025, pp. 1–6. doi: 10.1109/IMCOM64595.2025.10857522.

[39] N. L. S. Thatavarthi, "Integrating AI-Driven Chatbots with Full Stack E-commerce Platforms to Enhance Customer Experience," *Journal of Artificial Intelligence & Cloud Computing*, pp. 1–5, Dec. 2023, doi: 10.47363/JAICC/2023(2)E126.

[40] M. Merler, C. Galleguillos, and S. Belongie, "Recognizing Groceries in situ Using in vitro Training Data," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8. doi: 10.1109/CVPR.2007.383486.

[41] P. Gupta, S. Maity, and U. Gupta, "Conventional Image Recognition Chatbot," 2025, pp. 1044–1056. doi: 10.2991/978-94-6463-738-0_80.

[42] H. Krugenberg and K. Ruf, "What if…?-How AI Chatbots Could Reconfigure Grocery Shopping In-store," 2023.

APPENDICES

APPENDIX A: Graphs

APPENDIX B: Computer Programme Listing

Spacing between chapter title and first line of text is 4.5 lines. The first paragraph in a subsection should align with left margin. General alignment for texts in paragraph should be “justified”.

Spacing between paragraphs is 1.5 lines. Subsequent paragraphs should be indented 1.27 cm (0.5 inch) from the left margin. General alignment for texts in paragraph should be “justified”.

ORIGINALITY REPORT

14%	13%	9%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.coursehero.com Internet Source	1%
2	Submitted to Higher Education Commission Pakistan Student Paper	1%
3	uokerbala.edu.iq Internet Source	1%
4	Randy Antonio, Nadya Tyandra, Linggar Tembus Nusantara, Anderies, Alexander Agung Santoso Gunawan. "Study Literature Review: Discovering the Effect of Chatbot Implementation in E-commerce Customer Service System Towards Customer Satisfaction", 2022 International Seminar on Application for Technology of Information and Communication (iSemantic), 2022 Publication	<1%
5	eprints.utar.edu.my Internet Source	<1%
6	Hicham Kalkha, Azeddine Khiat, Ayoub Bahnasse, Hassan Ouajji. "The rising trends of smart e-commerce logistics", IEEE Access, 2023 Publication	<1%
7	jurnal.unidha.ac.id Internet Source	<1%

8	journal.esj.edu.iq Internet Source	<1 %
9	jurnal.uinsu.ac.id Internet Source	<1 %
10	link.springer.com Internet Source	<1 %
11	skemman.is Internet Source	<1 %
12	Ceren Gülra Melek, Elena Battini Sönmez, Songül Varlı. "Datasets and methods of product recognition on grocery shelf images using computer vision and machine learning approaches: An exhaustive literature review", Engineering Applications of Artificial Intelligence, 2024 Publication	<1 %
13	Carlos M. Mejía-Granda, José L. Fernández-Alemán, Juan M. Carrillo de Gea, José A. García-Berná. "A method and validation for auditing e-Health applications based on reusable software security requirements specifications", International Journal of Medical Informatics, 2025 Publication	<1 %
14	Dhirendra Kumar Shukla, Shabir Ali, Sandhya Sharma. "Artificial Intelligence and Sustainable Innovation - Volume 2", CRC Press, 2026 Publication	<1 %
15	kc.umn.ac.id Internet Source	<1 %
16	vbn.aau.dk Internet Source	<1 %