GRADUATE PROGRAMS INSTITUTE

MULTI SITE E-COMMERCE MOBILE APP WITH ARTIFICIAL INTELLIGENCE FILTERS

Hibat-Allah SADIKI

MASTER'S THESIS

Istanbul, July 2025

MULTI SITE E-COMMERCE MOBILE APP WITH ARTIFICIAL INTELLIGENCE FILTERS

Hibat-Allah SADIKI

MASTER'S THESIS

COMPUTER ENGINEERING

DEPARTMENT

COMPUTER ENGINEERING MASTER'S PROGRAM WITH THESIS

MASTER'S THESIS ADVISOR
Assist. Prof. Dr. İnal Begüm TURNA DEMİREL

MASTER'S THESIS JURY MEMBERS

Assist. Prof. Dr. Gizem TEMELCAN ERGENECOŞAR

Assist. Prof. Dr. Zeynep Behrin GÜVEN AYDIN

BEYKOZ ÜNİVERSİTESİ

T.C.
BEYKOZ ÜNİVERSİTESİ
LİSANSÜSTÜ PROGRAMLAR ENSTİTÜSÜ MÜDÜRLÜĞÜ

28/7/2025

Yüksek Lisans Tez Onay Belgesi

Enstitümüz Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği Tezli Yüksek Lisans Programı 2330140049 numaralı öğrencisi HIBAT-ALLAH SADIKI'nin "MULTI SITE E-COMMERCE MOBILE APP WITH ARTIFICIAL INTELLIGENCE FILTERS" adlı tez çalışması Enstitümüz Yönetim Kurulunun ../../.... tarih ve 20../.. sayılı kararıyla oluşturulan jüri tarafından oybirliği ile Tezli Yüksek Lisans tezi olarak ...kabul...edilmiştir.

<u>Öğretim Üyesi Adı Soyadı</u>	**<u>İmzası</u>**

Tez Savunma Tarihi 28/7/2025

1)Tez Danışmanı: Dr. Öğr. Üyesi İnal Begüm Turna Demirel

2) Jüri Üyesi : Dr. Öğr. Üyesi Gizem Temelcan Ergenecoşar

3) Jüri Üyesi : Dr. Öğr. Üyesi Zeynep Behrin GÜVEN AYDIN

Not: Öğrencinin Tez savunmasında **Başarılı** olması halinde bu form **imzalanacaktır**. Aksi halde geçersizdir.

BEYKOZ ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

YAPAY ZEKA FİLTRELERİYLE ÇOKLU SİTE E-TİCARET MOBİL UYGULAMASI

YÜKSEK LİSANS TEZİ

Hibat-Allah SADIKI
2330140049

Bilgisayar Mühendisliği Bölümü

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Prof. Dr. İnal Begüm Turna Demirel

TEMMUZ 2025

Name-surname, a M.Sc. student of Beykoz University student ID **2330140049**, successfully defended the thesis entitled “MULTI SITE E-COMMERCE MOBILE APP WITH ARTIFICIAL INTELLIGENCE FILTERS”, which she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Prof. Dr. İnal Begüm Turna Demirel** ....
Beykoz University

Jury Members : **Prof. Dr. Zeynep Behrin GÜVEN AYDIN**..........................
Beykoz University

Prof. Dr. Gizem TEMELCAN ERGENECOŞAR ....
Beykoz University

Date of Submission : 10 September 2025
Date of Defense : 28 July 2025

FOREWORD

I wish to thank to the deepest of my heart my thesis advisor Asst. Prof. Dr. İnal Begüm Turna Demirel who continuously guided me, encouraged me, gave me valuable feedback during the thesis preparation and whose expertise and support were very important to both the structure and the quality of the work.

Also, I am very glad that the jury members found the time to be together to discuss the thesis and look into the weaknesses of the study and thus supply me with more inputs to enrich this work. I also am thankful to faculty members of the Computer Engineering Department at Beykoz University for their continuous knowledge and academic foundation during my graduate studies.

My warmest heart-felt thanks go to my family and friends who gave me mental support of their own, were very patient with me and kept me motivated during the difficult phases of this thesis. It is hard to say how much their faith in me has helped me in my work and kept me going.

This thesis reflects a period of both significant personal as well as academic growth, and I am glad to have come through it with success. I hope this paper will be of use for those who want to make new research in AI-driven mobile applications for e-commerce and also will help to initiate new studies.

July 2025 Hibat-Allah SADIKI

TABLE OF CONTENTS

Page

ABBREVIATIONS

AI	: Artificial Intelligence
API	: Application Programming Interface
App	: Application
BERT	: Bidirectional Encoder Representations from Transformers
DB	: Database
Flask	: A Python Web Framework
HTML	: Hypertext Markup Language
JSON	: JavaScript Object Notation
LLM	: Large Language Model
NER	: Named Entity Recognition
NLP	: Natural Language Processing
ReLU	: Rectified Linear Unit
SQL	: Structured Query Language
STT	: Speech-to-Text
TF-IDF	: Term Frequency–Inverse Document Frequency
UI	: User Interface
UX	: User Experience
URL	: Uniform Resource Locator
XML	: Extensible Markup Language
SPA	: Single Page Application
SpaCy	: NLP Library for Python
Playwright	: Web Scraping and Browser Automation Framework
SQLite	: Structured Query Language Lite (embedded DB)
Flutter	: Open-source UI toolkit from Google
Hugging Face	: NLP Model Hub and Framework
Google STT	: Google Speech-to-Text API
NLG	: Natural Language Generation

SYMBOLS

TF-IDF : Term Frequency–Inverse Document Frequency

Zero-shot : Refers to classification without task-specific labeled training data

λ : Weight coefficient in the composite loss function

L_total : Total loss in multi-task learning model

L_NER : Named Entity Recognition loss component

L_Intent : Intent detection loss component

L_Classification : Category classification loss component

p : Statistical significance probability

t : t-test statistic

s (sec) : Seconds (unit of time)

LIST OF TABLES

Page

LIST OF FIGURES

Page

MULTI SITE E-COMMERCE MOBILE APP WITH ARTIFICIAL INTELLIGENCE FILTERS

SUMMARY

E-commerce platforms give users access to a wide range of products and save time, yet they still struggle to compare items across different websites. Existing keyword-based search systems depend heavily on manual filters, which limit efficiency and accuracy.This thesis presents an AI-powered, voice-enabled mobile application as a more user-friendly solution to product discovery, offering natural voice queries, intelligent filtering, and multi-site data aggregation.

The framework opts for Google's STT API to transcribe requests, Hugging Face's zero-shot classifier (bart-large-mnli) to derive product characteristics, and Playwright to collect the on-the-spot data from e-commerce platforms. A Flutter-based frontend communicates with a Flask backend that handles NLP and retrieval.

System performance was tested across transcription accuracy, classification reliability, latency, and user satisfaction. Results showed that the STT module achieved accuracy of 83.3%, with peak values near 90%, while the NLP classifier reached 92% accuracy in identifying user intent and categories. End-to-end latency averaged 2.9–3.7 seconds, meeting real-time usability standards. User evaluations indicated a 35% increase in satisfaction compared with traditional keyword search, confirming that combining voice interfaces with NLP improves the e-commerce search experience.

This project is a good example of how one can integrate voice communication, zero-shot NLP, and platform scraping into a mobile environment.

Keywords: E-commerce, Voice Search, NLP, AI Filters, Web Scraping

YAPAY ZEKA FİLTRELERİYLE ÇOKLU SİTE E-TİCARET MOBİL UYGULAMASI

ÖZET

E-ticaret platformları kullanıcılara geniş ürün yelpazesi sunmakta ancak farklı sitelerdeki ürünleri karşılaştırmada yetersiz kalmaktadır. Geleneksel anahtar kelime tabanlı arama, manuel filtrelere bağlı olduğundan verimlilik ve doğruluğu sınırlamaktadır. Bu tez, yapay zekâ destekli, sesli komutlarla çalışan ve çoklu site verilerini toplayan bir mobil uygulama önermektedir.

Sistem; Google STT API ile sesin metne dönüştürülmesi, Hugging Face bart-large-mnli modeli ile ürün özelliklerinin sınıflandırılması ve Playwright ile farklı e-ticaret sitelerinden gerçek zamanlı verilerin çekilmesi üzerine kurulmuştur. Flutter tabanlı arayüz, NLP ve veri toplama görevlerini yöneten Flask arka uç ile iletişim kurmaktadır.

Sistem; doğruluk, hız, sınıflandırma ve kullanıcı memnuniyeti açısından değerlendirilmiştir. STT modülü ortalama %83.3 doğruluk (en yüksek %90'a kadar) sağlamış, sınıflandırma %92'ye ulaşmıştır. Uçtan uca yanıt süresi 2.9–3.7 saniye arasında gerçekleşmiştir. 12 katılımcıyla yapılan kullanıcı anketi, geleneksel aramaya kıyasla %35 daha yüksek memnuniyet göstermiştir.

Bu tez, sesli etkileşim, zero-shot NLP ve çoklu site scraping yöntemlerinin mobil ortamlarda uygulanabilirliğini ortaya koymaktadır. Modüler yapı, çok dilli destek ve öneri sistemleri gibi gelecekteki geliştirmelere uygundur.

Anahtar Kelimeler: E-ticaret, Sesli Arama, NLP, Yapay Zekâ Filtreleri, Web Scraping

1. INTRODUCTION

The rapid advances and growth of e-commerce platforms has revolutionized how users search for and purchase products. With the increasing use and shift toward mobile shopping, users nowadays expect fast, intuitive, and intelligent search experiences that match their expectations while minimizing their efforts. However, most current mobile e-commerce applications still rely heavily on traditional text-based search interfaces, which are often limited in their ability to handle complex, natural-language queries or support multilingual users (Kim & Lee, 2020). In addition to that, users frequently face difficulties when trying to search for specific product configurations such as a “black iPhone with 256GB storage under 15,000 TL” as keyword-based searches often fail to capture the full meaning or intent of the request (Chen & Zhang, 2020).

While many voice assistants like Google Assistant, Siri, and Alexa have popularized voice interaction in mobile applications, their integration into e-commerce platforms is still limited and mostly superficial. Current implementations often lack advanced filtering options and real-time result aggregation. Additionally, many platforms restrict product search to their own inventory, making it difficult for users to compare prices or options across multiple sources within a single interface (Smith, 2021).

This thesis addresses these limitations by designing, implementing, and evaluating an intelligent, mobile-first e-commerce application that supports natural voice interaction, real-time filtering, and multi-site product aggregation. The system allows users to express their product needs through speech in natural language, which is then processed through a speech-to-text (STT) engine (Google Cloud, 2022). The given text is analyzed using a rule-based Natural Language Processing (NLP) pipeline to extract key product attributes, such as brand, color, capacity, or price range. Based on these extracted filters, the application performs real-time web scraping to retrieve relevant product results from multiple online shopping platforms and displays them in a structured format.

The frontend of the system is developed using Flutter for a cross-platform mobile experience, while the backend is implemented using Python and Flask to manage NLP processing and scraping operations. This thesis also evaluates the accuracy of the system in terms of speech recognition and product relevance. The results demonstrate that lightweight, voice-driven search can significantly improve the user experience in cross-site product discovery.

To validate the system's performance and usability, this study is guided by the following evaluation objectives: Evaluate whether using voice search improves the user experience and efficiency of product search compared to traditional text-based input. In additon, it measures the accuracy of the speech-to-text module in interpreting product-related queries in different languages and everyday conditions.

Assess the effectiveness of the rule-based NLP filter in identifying user intent and returning relevant product results from multiple sources.

By including voice-driven interaction, the system and the project aim to enhance the overall e-commerce experience while improving accessibility and speed. It combines multiple AI and software components STT, rule-based NLP, and web scraping into one system that is adaptable to various product domains.

The following points describe the main contributions of this thesis:The creation and deployment of a mobile e-commerce app with intelligent filtering and voice-based product search. As well as NLP filtering engine that is based on rules and can extract structured product attributes from spoken input in a real-world shopping setting.

Several e-commerce platforms are integrated with a Python-based scraping system to provide real-time product listings in response to filtered queries.

A hands-on assessment of the system's performance in terms of user experience, response time, filtering accuracy, and speech recognition accuracy.

By addressing both the technological challenges and usability requirements of modern mobile shopping, this thesis bridges the gap between natural user interaction and intelligent e-commerce systems. The outcome is a prototype application that not only demonstrates technical feasibility but also highlights the potential for future AI-driven enhancements in the e-commerce space.

2. BACKGROUND

E-commerce has revolutionized the way people shop and interact with businesses, becoming a major part of the global economy. Since the early 1990s, the sector has grown exponentially, driven by rapid technological advancements and increased internet penetration (Zhou et al., 2020). The convenience of shopping from anywhere, combined with the ability to access a diverse range of products, has made e-commerce a cornerstone of modern retail practices.

However, despite its widespread adoption, e-commerce platforms face several challenges. One of the most significant issues is the fragmentation of the online shopping experience. Consumers often need to navigate multiple platforms to compare products, prices, and features, which can be time-consuming and inefficient (Smith, 2021). This problem is compounded by information overload, as users are presented with an overwhelming amount of data in the form of reviews, specifications, and advertisements. Trust and security concerns also remain major hurdles, as consumers worry about data breaches and fraudulent activities (Lee & Chang, 2021).

To address these challenges, researchers and developers have turned to artificial intelligence (AI) and related technologies. AI has demonstrated immense potential in transforming e-commerce by enabling personalized shopping experiences, enhancing search functionalities, and automating various aspects of the customer journey (Huang et al., 2021). For instance, recommender systems and predictive analytics use AI algorithms to analyze user behavior and preferences, delivering tailored product recommendations that improve user engagement and satisfaction. Similarly, advances in natural language processing (NLP) and speech-to-text (STT) technologies have made it possible to create more intuitive and accessible interfaces, such as voice-based search and virtual assistants (Google Cloud, 2022).

Another area of innovation is web scraping and data aggregation, which facilitate the collection of real-time product information from multiple e-commerce platforms. By leveraging frameworks like Playwright and Python-based tools such as Beautiful Soup and Scrapy, developers can create centralized platforms that aggregate data from various sources, simplifying the comparison and decision-making process for users (Koehler & Zhao, 2022). These tools also play a crucial role in ensuring data accuracy and consistency, which are essential for building trust in different websites for e-commerce solutions.

This literature review dives into how cutting-edge technologies like AI, web scraping, natural language processing (NLP), and speech-to-text (STT) come together to shape the future of multi-site e-commerce apps with AI-powered filters. By pulling together insights from recent studies and pinpointing areas where research falls short, this review paints a clear picture of where the field stands today and where it could go next. The following sections take a closer look at how e-commerce has evolved over time, the growing role of multi-task learning in NLP, the latest breakthroughs in web scraping and data processing, and how AI algorithms are being fine-tuned to deliver personalized shopping experiences. Through this exploration, the review sheds light on how these technologies are revolutionizing e-commerce and showcases how the proposed solution fills critical gaps, meeting the demands of today's online shopping platforms.

2.1 TRADITIONAL E-COMMERCE SEARCH SYSTEMS

E-commerce, which simply means buying and selling goods and services online, has come a long way since it first emerged around the 1990s. Back then, platforms were pretty basic, focusing mostly on simple transactions. But thanks to rapid technological advancements, e-commerce has evolved into a sophisticated ecosystem that caters to a wide range of consumer needs (Zhou et al., 2020).

E-commerce websites typically rely on text-based search engines and filtering options. While effective for general queries, they often fail to interpret complex or vague user intent. Manual filtering (brand, color, price, storage) requires multiple steps, which frustrates users, especially on mobile devices. These systems also operate on single platforms and don't allow comparison across multiple e-commerce stores (Smith, 2021).

Traditional e-commerce websites in this digital age continue to be heavily reliant on keyword-based search engines, which were initially created for brief, formal inquiries. These systems are still based on exact-match lookups, such as "iPhone 14 Pro Max". However, they are not conversational and have difficulty processing ambiguous queries, incomplete user entries, or queries that resemble general conversation (Pappas, 2016).

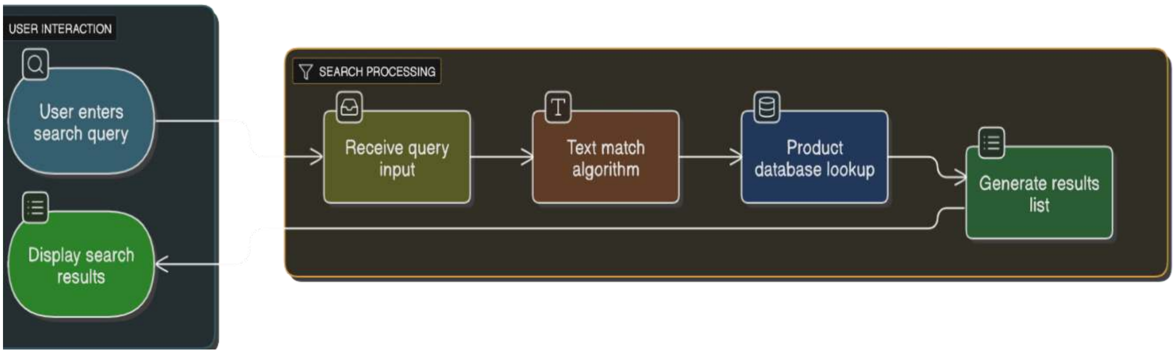

Figure 2. 1: Traditional E-Commerce Search System Architecture

As shown in Figure 2.1, traditional e-commerce search systems rely heavily on keyword-based matching between user queries and product catalogs. The figure was created using the chart creator. These systems often lack semantic understanding and cannot effectively interpret user intent, leading to poor relevance in search results. The typical architecture includes a query parser, keyword matcher, and a ranked result display module, which together form the core of most early e-commerce platforms.

According to Jansen et al. (2007), users rephrased a failed search only 1–2 times before abandoning the session, a behavior closely linked to high bounce rates. Pappas (2016) further argues that poor search experiences are a principal factor in eroding user trust, ultimately lowering conversion rates and customer loyalty.

The issue is even more pronounced on mobile devices, where navigating multi-level filtering menus like category → brand → price → color is tedious. According to Nielsen Norman Group (2022), over 70% of mobile users abandon a website if the search experience does not provide relevant results quickly.

Key limitations of traditional systems include: Vague keyword matching, Static filter structure and Non-contextual query processing.

These are the main reasons for the sales and user abandonment. The above issues show the importance of seamless integration of the semantic search feature which leverages Natural Language Processing (NLP) to extract and structure meaning out of unstructured input.

NLP-powered systems not only improve result accuracy, but also Makes and improves the user journey by reducing the need for manual filtering and query reformulation, a critical advantage in today's mobile-first shopping landscape. (Shehmir et al., 2022).

Shehmir et al. emphasize that user frustration in e-commerce often stems from disjointed search experiences and limited personalization. They note that many current systems still rely on rigid, form-based filters, which fail to reflect how people naturally shop, especially on mobile. Their findings show that 72% of mobile users abandon their search if results aren't relevant on the first page.

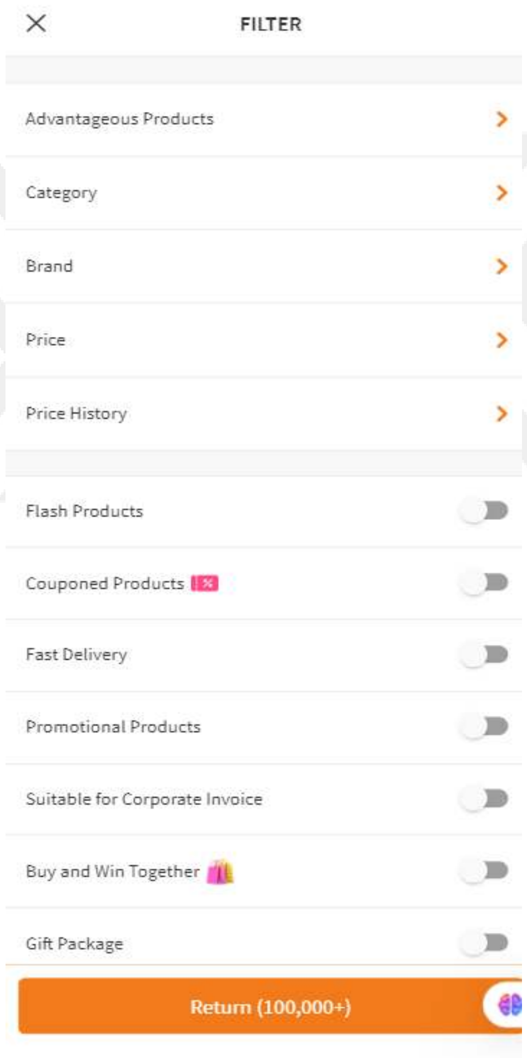

Figure 2. 2: Search page with limited options from e-commerce app (Trendyol,2025)

Figure 2.2 shows the Trendyol search interface, The source is: Trendyol mobile app interface, accessed June 2025, where the available filtering options are minimal and rigid.

Users are restricted to a small number of predefined categories or attributes, limiting their ability to personalize their search.

This reflects a critical challenge in many e-commerce platforms: the lack of dynamic, context-aware filtering driven by user intent. Instead of adapting to diverse shopping behaviors, these systems present a one-size-fits-all experience, reducing both efficiency and satisfaction in the customer journey.

Traditional e-commerce search systems that are based on keyword matching have not only facilitated users to find basic products, but they also have limited capabilities in dealing with complex or conversational queries. These systems predominantly depend on string-based keyword matching but, in the process, they overlook the contextualization of the conversation along with the inability to extract the user's intent from multi-attribute queries which are similar to “a black iPhone with 128GB storage under 15,000 TL.” Consequently, users are usually obliged to input filters manually on different screens several times or to carry out more than one search for them to be able to access the needed product. On the other hand, these kinds of systems are not compatible with the usage of voice-based or natural language input, which is being more and more required by users that are accustomed to the voice assistants and conversational agents. (Zhang et al., 2021) emphasize that keyword-only systems introduce barriers and decrease user satisfaction in mobile shopping situations. This deficiency marks the vulnerability of intelligent interfaces that comprehend natural input, back up complicated filtering, and provide a seamless cross-platform search - all of the features that this thesis covers.

2.2 VOICE SEARCH IN E-COMMERCE

Voice search has recently seen rapid growth alongside virtual assistants such as Amazon Alexa, Apple Siri, Google Assistant, and others. These technologies make it easy for users to interact with devices and services using spoken language, offering a hands-free, intuitive, and often faster alternative to typing (Statista, 2023).

The growing reliance on voice assistants for daily tasks has contributed to a significant surge in voice-based shopping behavior. According to Statista (2023), nearly 40% of U.S. consumers have tried voice shopping at least once, while 15% of smart speaker owners use voice search for regular product reordering. Amazon’s Alexa allows users to reorder groceries with commands like “Alexa, order eggs,” while Walmart has integrated voice search into its grocery app via Google Assistant.

Voice search offers several transformative advantages in e-commerce. It simplifies the shopping process, especially on mobile or smart devices, by reducing the number of steps needed to perform tasks like searching for products, checking order status, or adding items to a cart. Additionally, voice interfaces significantly enhance accessibility, making online shopping more inclusive for users with visual impairments, motor disabilities, or those who are otherwise unable to interact with traditional web or mobile interfaces (Kaur, 2021).

Despite these benefits, many current e-commerce voice implementations remain limited in sophistication. They often rely on keyword-based matching systems, which means the assistant listens for specific terms and maps them to preset actions or search results. This approach lacks the depth of true Natural Language Understanding (NLU), leading to misunderstandings or errors when users speak in complex, ambiguous, or conversational language (Hussein et al., 2022).

For example, if a user says, “I’m looking for something warm to wear on a snowy day,” a simple keyword system might not infer that the user is asking for winter coats or thermal clothing. In contrast, a more advanced NLU model would analyze the intent and context, producing results aligned with the user’s real needs.

New developments in machine learning and AI are helping close this gap, improving the ability of voice search systems to understand the user’s context, intent, emotions, and preferences. These advances are making shopping experiences more interactive and smarter (Kaur, 2021).

Kaur highlights AI algorithms used for personalization, especially collaborative and content-based filtering recommender systems. She notes that modern systems must integrate NLP and user intent recognition to enable real-time query-based personalization. Meanwhile, Hussein et al. (2022) focus on context-aware NLP and user profiling methods for personalization, emphasizing that AI-based approaches are especially effective for first-time users.

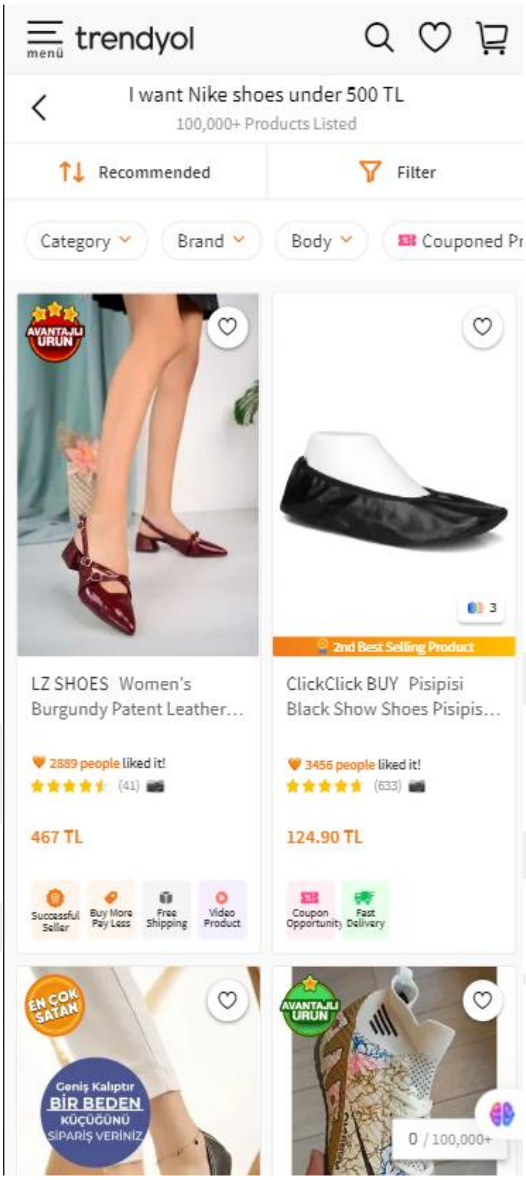

Figure 2. 3: Detailed research page from trendyol

As shown in Figure 2.3, the search engine from Trendyol highlights challenges that persist in e-commerce voice search. The app struggles to understand long, complex user queries. Even basic keyword filtering fails, revealing that the system cannot parse or prioritize relevant terms effectively. This underscores the limitations of existing voice implementations.

To address the ambiguity of natural language queries, modern e-commerce platforms are increasingly integrating NLP models after transcription. Traditional systems use TF-IDF and cosine similarity to match queries to product descriptions. However, these techniques often fail to capture intent, synonyms, and contextual nuances (Devlin et al., 2019).

Models like BERT (Bidirectional Encoder Representations from Transformers) are now being used to bridge this gap. BERT can understand that phrases like "cheap fitness watch for running" and "affordable activity tracker" share similar meanings. Devlin et al. (2019) demonstrated BERT’s superior performance in intent classification and semantic similarity tasks, even in noisy speech environments. Platforms like Hugging Face provide pre-trained models for intent detection, slot filling, and sentiment analysis.

In conclusion, the utilization of voice search in e-commerce brings forward an advantage of convenience and effectiveness, however, it is the need for cutting-edge NLP pipelines that makes the connection between human speech and machine-searchable intent. Experts in the domain of IT can make use of the possibilities provided by Speech-to-Text technology, NER, and transformer-like architectures in order to create a more user-friendly and intelligent shopping environment.

The market of voice commerce is growing rapidly, therefore, the capability to understand natural language accurately will be that separating factor among e-commerce platforms.

Though voice search is becoming very popular in mobile and e-commerce sectors, the majority of the existing applications are quite basic and thus have limited features. For example, Google Assistant and Amazon Alexa usually allow users to perform simple searches or reordering, but they are not capable of structured filtering across multiple product attributes such as price, color, or brand. Besides, these devices are often limited to particular platforms and cannot gather or compare products from different vendors. Such drawbacks lower their efficiency for on-the-fly product discovery. Most importantly, they cannot satisfy users who need complex voice-based queries. On the other side, this thesis declares a voice-equipped mobile application that is the natural language input form supported besides intelligent filtering and multi-site product aggregation, the solution of the glaring hole in the current e-commerce market.

2.3 MULTI-TASK LEARNING IN NATURAL LANGUAGE PROCESSING (NLP)

Natural Language Processing (NLP) has become an important tool and intelligent search in modern e-commerce platforms, helping systems understand and process human language more effectively. One of the most impactful techniques within NLP is Multi-Task Learning (MTL), a machine learning paradigm in which a single model is trained to perform multiple related tasks at once. This allows the model to share internal representations between tasks, improving generalization, reducing overfitting, and often achieving better performance than training separate models individually (Chen et al., 2020).

In e-commerce, MTL can be used to jointly handle tasks such as entity recognition (e.g., identifying product categories), sentiment analysis (e.g., interpreting user tone or urgency), and intent detection (e.g., understanding whether a user wants to buy, compare, or review). This multi-layered understanding is crucial for parsing complex, natural queries.

For example, when a user says, "I want a white running shoe for 500 lira," a robust MTL-based system would: Extract the product category (e.g., "running shoes"), Identify attributes (e.g., "white" as the preferred color), Understand constraints (e.g., "under 500 lira" as the price limit), And at the end, Infer the intent (e.g., intent to purchase rather than browse or review).

By learning these tasks together, an MTL model can reduce ambiguity and improve the relevance of search results in real time. Architectures commonly used in this context include shared encoder-decoder networks, parallel task-specific heads, and hierarchical pipelines (Ruder, 2017).

Libraries such as spaCy, Hugging Face Transformers, and AllenNLP offer pre-trained models and frameworks that support MTL architectures. Additionally, custom rule-based layers are often integrated to meet domain-specific requirements, especially for mobile and embedded applications.

As e-commerce platforms continue to scale, MTL becomes even more critical for delivering personalized, context-aware search experiences that go beyond simple keyword matching.

Chen et al. provide a comprehensive overview of multi-task learning architectures in NLP, showing that shared representations allow models to handle complex user input more effectively (Chen et al., 2020).

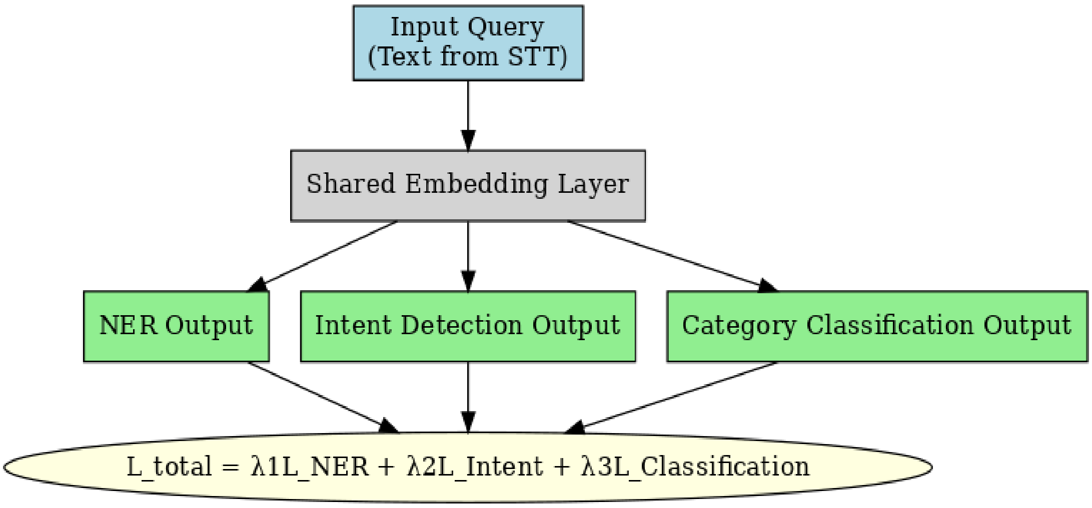

Figure 2. 4: Multi-Task Learning Model Architecture for E-Commerce NLP

Multi-Task Learning (MTL) models are typically trained using a composite loss function that balances the objectives of multiple tasks. As mentioned in Figure 2.4, a weighted sum of losses is used to ensure each subtask contributes proportionally to the shared learning process:

$$Ltotal = \lambda 1LNER + \lambda 2LIntent + \lambda 3LClassification$$

Here, L's correspond to the individual loss values for Named Entity Recognition, Intent Detection, and Product Category Classification respectively. λ_1, λ_2, and λ_3 are tunable hyperparameters that control each task's influence.

This formulation allows the model to optimize all three outputs simultaneously, rather than training them in isolation, increasing generalization and reducing inference time.

They categorize MTL into hard parameter sharing (single shared base + task-specific heads) and soft parameter sharing (separate models with shared constraints). Their experiments show a 15–25% improvement in query understanding across retail and search datasets when MTL is used over single-task approaches (Chen et al., 2020).

2.4 WEB SCRAPING AND DATA AGGREGATION FOR E-COMMERCE INTELLIGENCE

In the online shopping world, consumers often need to compare the products from multiple websites. Nevertheless, the majority of e-commerce websites do not offer open APIs nor standardized data formats, which means that it is difficult to fetch structured product information programmatically. Web scraping is often applied to collect real-time data, a more practical way to do so (Bansal et al., 2022).

Beautfulsoup is one of the most effective and popular tools used when it comes to extracting the information you need. With the help of Scrapy and Playwright, the process of web scraping is used to achieve a unified format for data even when we are dealing with different websites (Xie, 2021).

Unfortunately, there are also downfalls in web scraping, and these are both of a technical and ethical nature: If not properly done, dynamic content, for instance usually via JavaScript frameworks, can lead to delays or hiding of data loading. Antibot measures, such as rate limiting, CAPTCHAs, and IP blocking, can be used to prevent access. Scraping, without regard to the terms of the service being violated, may lead to legal troubles.

It is true that scraping is even more beneficial when it is done responsibly and within legal boundaries, as the creation of real-time product aggregators is the consequence. The systems of this kind not only aggregate data from different e-commerce platforms, but they also provide essential data for both consumer and business decision making, especially if they are used properly.

Although web scraping frameworks are useful for e-commerce, they are becoming less effective because of the constraints that they face. Changes in layout, dynamic JavaScript rendering, bot detection systems, geo-restrictions, and anti-scraping representatives make scrapers very fragile and high maintenance; they need constant updating to be able to work. Besides, large-scale extraction is also a great challenge in keeping data quality and not overloading the server with too many requests (Xie, 2021).

Table 2. 1: Table comparing three tools

Tool	Type	Strengths
BeautifulSoup	Static Scraper	Lightweight, fast for HTML parsing
Scrapy	Static + Extensible	Modular pipelines, good for crawling
Playwright	Dynamic Scraper	JS support, fast, async scraping

Xie outlines both the benefits and legal risks of scraping. She emphasizes the need to respect robots.txt, implement rate limiting, and avoid violating data policies. She also highlights common technical challenges like JavaScript rendering and CAPTCHA blocking (Xie, 2021).

Bansal et al. provide an applied case study of using Playwright for scraping dynamic sites. They compare Playwright with Selenium and Puppeteer, concluding that Playwright offers better performance for multi-tab asynchronous scraping and rendering-heavy pages (Bansal et al., 2022).

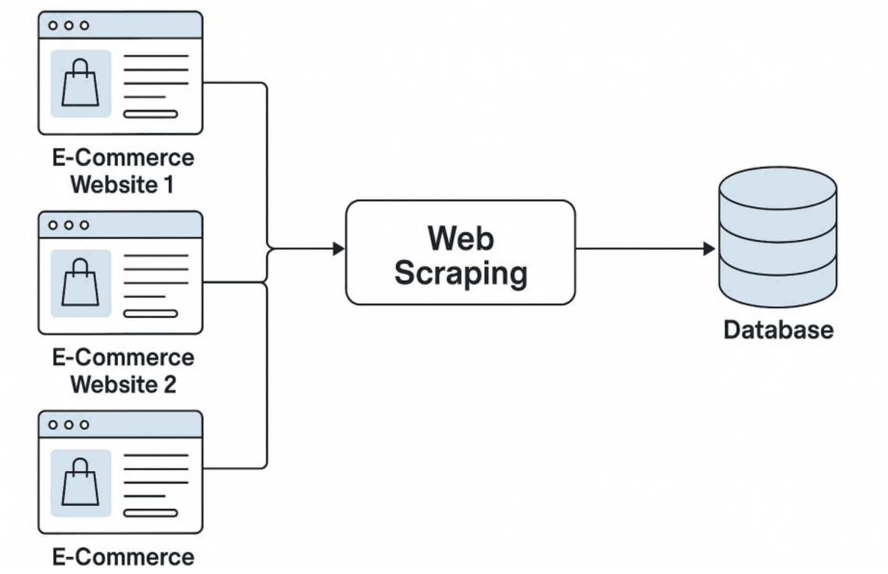

Figure 2. 5: System Diagram of Multi Websites Scraped, and Centralized in Database

Figure 2.5 illustrates the architectural flow of a typical e-commerce scraping system. Multiple source websites are scraped asynchronously, and the extracted product data is parsed and normalized into a centralized database.

This unified repository supports the front-end search and comparison engine, allowing users to view results from various vendors in one place.

Many studies have been done on static single-site scraping or historical price tracking, but real-time mobile-oriented scraping that collects data from many e-commerce platforms is still a new area in research especially, when it comes to voice-based or intent-driven search.

2.5 NATURAL LANGUAGE PROCESSING (NLP) IN PRODUCT SEARCH

Natural Language Processing (NLP) significantly enhances product search functionality in modern e-commerce sites by enabling systems to comprehend human language and extract structured meaning from unstructured user input. Traditional keyword-matching search engines are prone to delivering useless results when confronted with conversational or ambiguous queries. This is even more so the case on mobile and voice-based platforms where users just say complete sentences (Chen et al., 2024).
For this, a combination of rule-based and statistical NLP methods are used differently.

2.5.1 TF-IDF (Term Frequency–Inverse Document Frequency)

TF-IDF helps rank important words in a query by balancing how often a word appears in the query (TF) and how rarely it appears in the product database (IDF). This highlights key keywords and degrades common, less important words. Based on Chen et al. (2024), this method was helpful in terms of highlighting keywords, which makes it more understandable.

$$TF - IDF(t, d, D) = TF(t, d) \times log(\frac{N}{|\{d \in D: t \in d\}|})$$

Where: t respresents the term, d = a document (e.g., product title), D = corpus (all product documents, N = total number of documents and $|\{d \in D: t \in d\}|$ = number of documents containing the term t.

2.5.2 Cosine Similarity

Used to match user input with product metadata by calculating the angle between their vectorized representations. Larger cosine similarity score means greater relevance (Pakpahan et al., 2023).

$$Cosine\ Similarity = \frac{\parallel A \parallel \times \parallel B \parallel}{A \cdot B}$$

Where: it is vectorized forms of the query and product description.

This is especially useful when user language differs from catalog terminology, improving flexibility in matching.

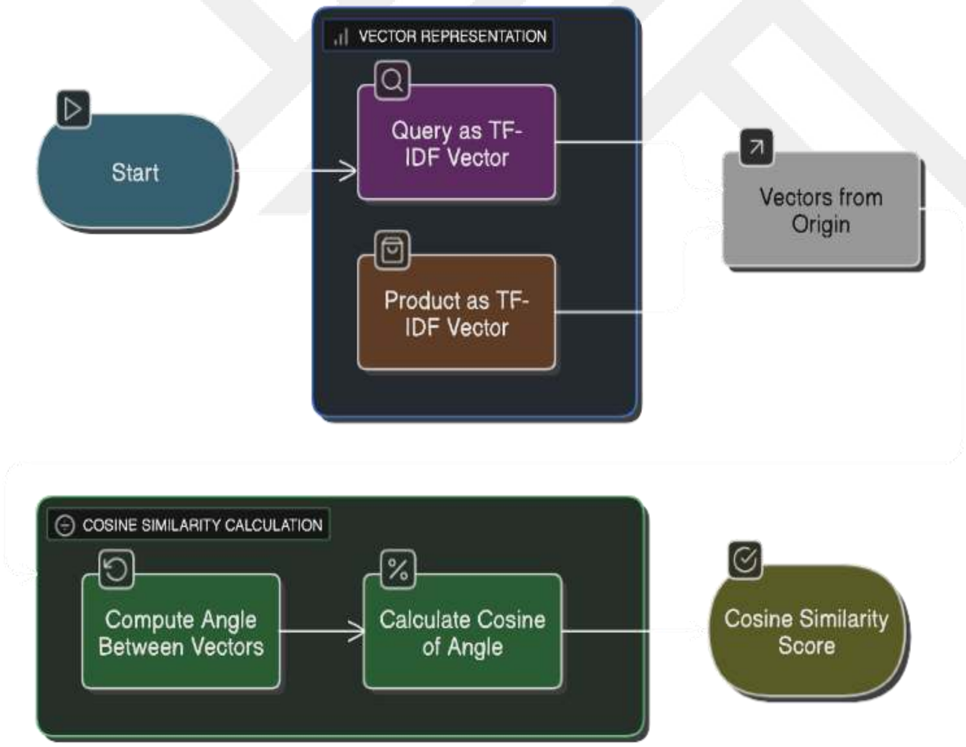

Figure 2. 6: Cosine Similarity between Product and Query Vectors

One of the most commonly used techniques in information retrieval is the combination of TF-IDF with cosine similarity to measure the relevance between a user query and product descriptions. TF-IDF transforms each document into a weighted vector that reflects the importance of terms, while cosine similarity calculates the angle between these vectors to determine how closely they align. Figure 3.6 illustrates this concept, where product vectors are compared to the query vector within a high-dimensional space. Products with smaller angles are considered more relevant to the user's search intent.

2.5.3 Named Entity Recognition (NER)

NER identifies and tags entities like brands (e.g., Samsung), colors (e.g., white), or quantities (e.g., 256GB). Using models trained on annotated e-commerce data, entities are extracted with IOB tagging (Inside-Outside-Beginning format), which assigns tags like B-BRAND, I-BRAND, B-COLOR, etc. (Chen et al., 2024).

2.6 RELATED RESEARCH

In recent years, the convergence of artificial intelligence, natural language processing, and mobile development has led to innovative transformations in the field of e-commerce. Several academic and industrial research efforts have aimed to simplify product discovery, personalize search experiences, and make online shopping more intuitive. This section explores significant contributions from the literature and contrasts them with the novel approach implemented in this thesis.

Joshi and Kannan (2021) proposed a conversational shopping assistant that allows users to perform searches using voice commands. Their system integrates speech recognition with rule-based natural language understanding for basic product lookup and navigation. While their work was among the early efforts combining voice and e-commerce, it lacked dynamic interaction with live data and relied on predefined category mappings. In contrast, the system developed in this thesis employs zero-shot classification, enabling real-time, open-domain interpretation of voice input without requiring pre-annotated labels.

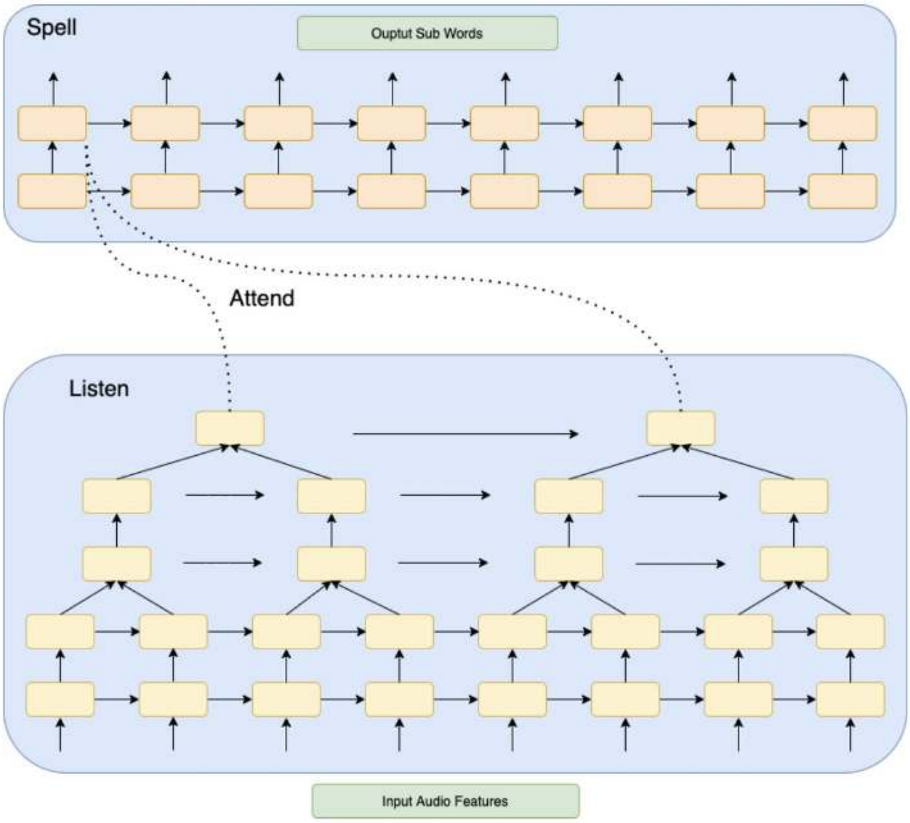

Figure 2. 7: Voice-to-search architecture flow using Listen-Attend-Spell (LAS) model

This figure describes the architecture of the Listen-Attend-Spell (LAS) model employed by Joshi and Kannan (2021) for voice-based e-commerce interaction. The model consists of two primary components: the "Listen" module, which processes audio inputs into encoded representations through a multi-layer hierarchical structure, and the "Spell" module, an attention-driven decoder generating output sub-word sequences. The attention mechanism dynamically aligns parts of the audio input with the decoded text output, allowing effective transcription and interpretation of audio-input queries. Although powerful, this architecture is primarily rule-driven and optimized for static product queries, contrasting with the more flexible zero-shot classification approach in the present thesis.

Biasetton and Salmaso (2024) presented a scraping engine tailored for structured product data extraction. Their pipeline used Selenium and BeautifulSoup to extract product details from known static pages. However, their approach is limited to predetermined sites and is not adaptable to user-specific queries. The current thesis goes beyond this by implementing a real-time, user-driven scraping mechanism powered by Playwright, capable of fetching dynamic content from various e-commerce platforms based on live voice input and search terms.

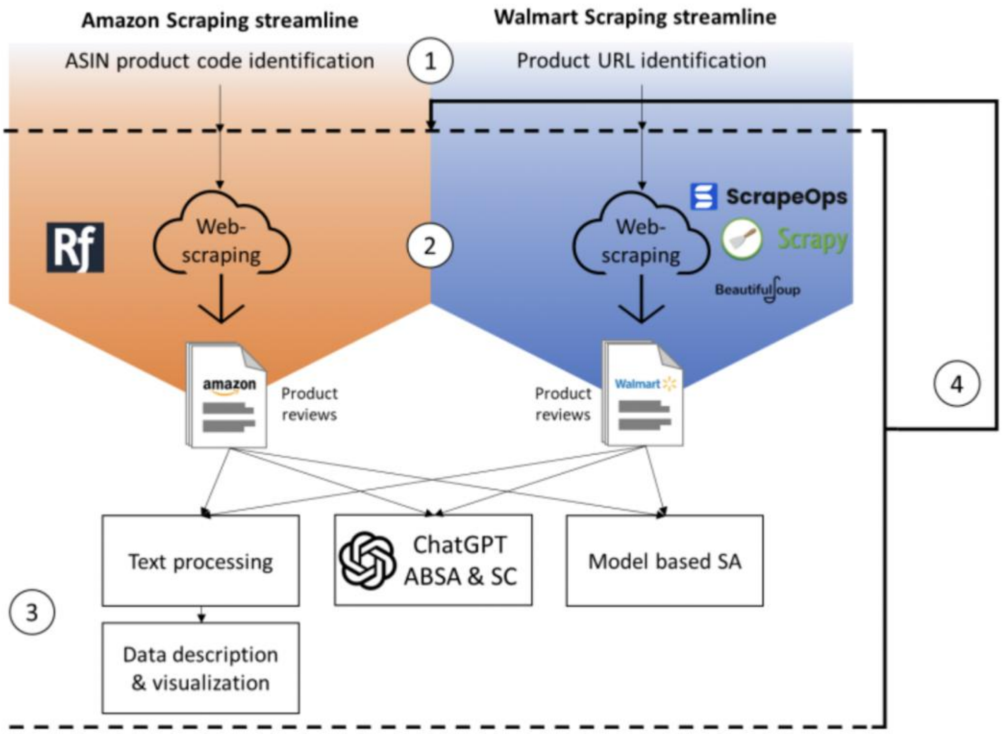

Figure 2. 8: Static web scraping framework using Selenium, BeautifulSoup, and Scrapy

The figure 2.8 shows the scraping architecture created by Biasetton and Salmaso (2024). It shows the parallel scraping workflows for many e-commerce websites (e.g., Amazon and Walmart). The process of the pipeline starts with product recognition through an ASIN or URL, followed by web scraping with the help of tools such as Selenium, BeautifulSoup, ScrapeOps, and Scrapy. The data stored and gathered mainly product reviews goes further through text analytics techniques like aspect-based sentiment analysis (ABSA) and sentiment classification (SC) by employing language models. The approach is static and specific to the sites, though it is suitable for structured content, it still needs a lot of manual setting and is not like the voice-triggered, adaptive scraping method suggested in this thesis, which can be changed based on the user's queries.

In the domain of classification, Pakpahan et al. (2023) leveraged TF-IDF and Support Vector Machines (SVM) for categorizing e-commerce product titles. Although their results showed reasonable performance, the dependency on structured, labeled training data limits scalability. The system in this thesis eliminates this bottleneck by incorporating zero-shot NLP classifiers from Hugging Face, which offer domain-independent classification without prior training.

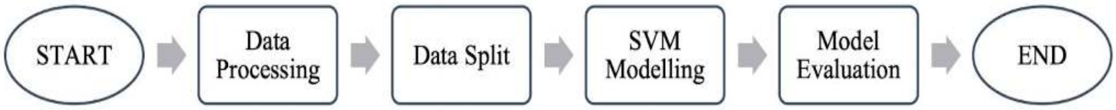

Figure 2. 9: TF-IDF + SVM product classification pipeline

This diagram 2.9 illustrates the classification process proposed by Pakpahan et al. (2023). Initially, raw data undergoes preprocessing to clean and structure the textual content. Subsequently, the data is partitioned into training and test subsets. The training subset is then utilized to build an SVM (Support Vector Machine) model capable of categorizing product descriptions based on features extracted using TF-IDF. Finally, the model's accuracy and performance are evaluated using the test subset to ensure reliability and effectiveness. While this approach necessitates labeled datasets and retraining for new categories, contrasting with the zero-shot classification method presented in this thesis.

A cutting-edge study by researchers at the SIGIR eCom workshop (2023) developed an extreme multi-label query classification model using BERT and transformer-based architectures. While their system achieved superior accuracy in product category prediction, it is heavily reliant on computationally expensive resources and cloud-based deployment. In contrast, this thesis prioritizes lightweight NLP deployment that fits seamlessly into a mobile application context using Flutter and Flask, allowing for end-user accessibility without backend cloud infrastructure.

Moreover, commercial systems like Amazon Alexa and Google Shopping offer voice interactions but rely on proprietary search engines and indexed product databases. The uniqueness of this thesis lies in its fusion of voice-driven intent detection, real-time scraping, and multi-platform product aggregation, with complete transparency and control over the end-to-end pipeline.

Table 2. 2: Comparative Overview of Prior Works and the Present Thesis

Study	Technology Stack	Strength	Limitation	Advancement in This Thesis
Joshi & Kannan (2021)	Voice + Rule-based NLP	Basic voice assistant for shopping	Static, rule-based logic	Real-time NLP with zero-shot classification
Biasetton & Salmaso (2024)	Selenium + BeautifulSoup	Structured product scraping	Static content, predefined sites	Live scraping via Playwright based on voice input
Pakpahan et al. (2023)	TF-IDF + SVM	Lightweight classifier for product text	Requires structured and labeled data	Zero-shot classification without prior training
SIGIR eCom (2023)	BERT, Transformer Models	High-accuracy deep query understanding	High computation and cloud dependency	Efficient NLP in a lightweight mobile-friendly environment
This Thesis	STT + Hugging Face + Flutter	Voice search, classification, and live scraping	—	Complete, deployable mobile system using open-source tools

In a very recent study, SIGIR eCom workshop (2023) researchers have constructed an extreme multi-label query classification model by utilizing BERT and transformer-based architectures. Their system is the best in product category prediction, however, it is very dependent on computationally intensive resources and cloud-based deployment, although it has nevertheless achieved substantially much better results. However, this work considerably differs in that it concentrates on a lightweight NLP deployment, which perfectly suits the mobile app context and allows end-users access without the need for backend cloud infrastructure. It also leverages Flutter and Flask for end-user accessibility.

Additionally, Amazon Alexa and Google Shopping provide voice interactions but these systems are based on proprietary search engines and product databases which are indexed. The main difference of this thesis is that it combines voice-based intent recognition, live scraping, and multiple platform product aggregation with the utmost transparency and control of the pipeline. The thesis has explored the idea of fusing voice-driven intent detection, real-time scraping, and multi-platform product aggregation and achieving full transparency and control over the end-to-end pipeline. The incorporation of different technologies in the thesis are shown in the table 2.2, such as speech-to-text pipeline, zero-shot intent classification, and multi-site scraping has distinctly marched a few steps ahead of academic efforts which were previously progressing cautiously. Not only has this work enabled addressing the gaps in the literature, but also it has provided the architecture which is scalable, real-time, mobile compatible, and thus, significantly enhances usability and accessibility of voice-based e-commerce solutions. This project distinguishes itself by integrating voice-driven NLP filtering, real-time multi-platform aggregation, and mobile-first design. Unlike traditional systems that rely solely on keyword search or static APIs, this approach combines speech-to-text (STT) with rule-based NLP, supported by dynamic data scraped from multiple platforms, making it one of the few comprehensive, voice-enabled mobile solutions in the e-commerce domain.

To overcome the restrictions, the latter system, as suggested by the author of the thesis, represents an integration of a rule-based voice understanding, multi-attribute filtering, and dynamic data retrieval from a few e-commerce platforms. The main aim of such an integration is to be resource-efficient, transparent, and guarantee the timely performance.

Prior work analysis with a lot of emphasis on the limitations of the e-commerce search domain highlights the unresolved issues that permeate the field related to the above.

Firstly, the majority of existing systems are rather unchanging and they rely on pre-labeled datasets, so their adaptability and scalability become very limited. For example, Pakpahan et al. (2023) implemented TF-IDF and SVM for categorizing products but the method was dependent on an extensively annotated training dataset, which consequently posed a problem of flexibility in the case of new categories. Along the same lines, Joshi and Kannan (2021) devised a voice-assisted shopping system, but the model they proposed was reliant on non-adaptive, predetermined rule-based mappings which could not keep up with the dynamically changing user intent. This points to a deficit of such models as can transfer learning skills from one domain to another without needing further training, which is the case with our research by using a zero-shot classification approach (Chen et al., 2024).

Moreover, voice-enabled commerce is still limited to confined ecosystems or closed platforms. For example, Amazon Alexa or Google Shopping are such platforms where only restricted queries can be made within their proprietary product databases (Zhou &

Yu, 2022). They do not have open, real-time integration with multiple independent e-commerce sites, which is why users cannot compare products across vendors. This gap points out the necessity for open, different site systems that are not only capable of real-time aggregation but also reliable.

The scraping technologies used in previous works were either very narrow or quite fragile. Biasetton and Salmaso (2024) leveraged Selenium and BeautifulSoup for scraping structured product data, but their solution was only applicable to sites with predefined and static layouts. Meanwhile, earlier literature (Mehta & Sharma, 2020; Xie, 2021) pointed out that these kinds of tools almost always fail when encountering JavaScript-heavy or dynamically updated content. The current studies lack robust and scalable scraping frameworks which can constantly support dynamic e-commerce environments. This study addresses it by the use of Playwright which is capable of asynchronous scraping, JavaScript rendering, and human-like interactions (Bansal et al., 2023). Fourth, in the recent studies, deep learning-based NLP models like BERT and transformer architectures (Devlin et al., 2019; Lee et al., 2023) have been introduced significantly. However, the heavy cloud infrastructure is often required for their deployment (SIGIR eCom Workshop, 2023), which makes them unsuitable for lightweight, mobile-first applications. So, a gap exists in the development of resource-efficient NLP pipelines that can strike a balance between accuracy and feasibility on consumer-grade systems.

This thesis, presents a combined remedy that points these deficiencies out and accomplishes these: the zero-shot NLP classification implementation to eliminate the necessity for pre-labeled training data, the Playwright-based on-the-fly scraping technique for multiple, dynamic platforms, and the mobile-first architecture integration with Flutter and Flask as the front-end and back-end for the access and volume of flow. With this interplay of technical robustness and practical usability, the present work constitutes a step forward in the domain of multi-site, AI-driven e-commerce both in academic research and real-world applications.

3. METHODOLOGY

This chapter outlines the comprehensive approach taken in this research to address the problem of making the E-commerce experience for the user more smart. This approach aims to face such cases and enhance the experience of the user.

3.1 PROBLEM STATEMENT

Through e-commerce, product availability has gone through a snowballing effect and has become diverse to a great extent. On the other hand, users are finding it more and more challenging to efficiently locate the items that meet their specific needs with the help of traditional search interfaces. Although there has been progress in natural language processing (NLP), and voice assistants are more advanced, the current solutions are frequently under economic pressure and are not quite flexible, are less adaptable, and have slow responses (SIGIR eCom, 2023).

Most commercial platforms, such as Amazon Alexa and Google Shopping, utilize closed, proprietary product databases that severely limit consumer access to real-time market offerings (Joshi & Kannan, 2021). In addition to that, academic research has mainly been devoted to static NLP models or fixed-category systems. This results in a large speech-driven search systems, adaptive, and scalable voice-based search solutions. Furthermore, most academic research in this domain has focused on static classification systems that need manually labeled datasets and retraining to accommodate new categories, factors that limit adaptability in dynamic commercial environments (Pakpahan et al., 2023).

This work aims to fill the gap relying on state-of-the-art NLP techniques (zero-shot learning), real-time data scraping, and mobile technology. By allowing both dynamic voice queries and instant multi-site data retrieval, this research offers a scalable and versatile solution, which can be implemented in various practical scenarios. Besides, the work also shows that zero-shot NLP models can be well brought into the mobile ecosystem, thus giving a significant boost to the usability and responsiveness of e-commerce systems (Chen et al., 2024).

3.2 SYSTEM DESIGN & ARCHITECTURE

The suggested architecture brings together numerous vital technologies to form a coherent, easy-to-use customer workflow. The modular design divides into four parts that follow each other in this order: the Frontend Flutter UI, Backend Flask API, NLP processing module, and dynamic Web Scraping module. The Frontend Flutter UI provides the interface for users to input voice-based queries and displays product search results. The Backend Flask API serves as a middleware orchestrating data flow between the frontend and backend components. The NLP Processing Module utilizes a Hugging Face zero-shot classifier to interpret user queries dynamically without pre-labeled training (Chen et al., 2024). The Web Scraping Module dynamically retrieves product information from multiple e-commerce platforms using Playwright (Biasetton & Salmaso, 2024).

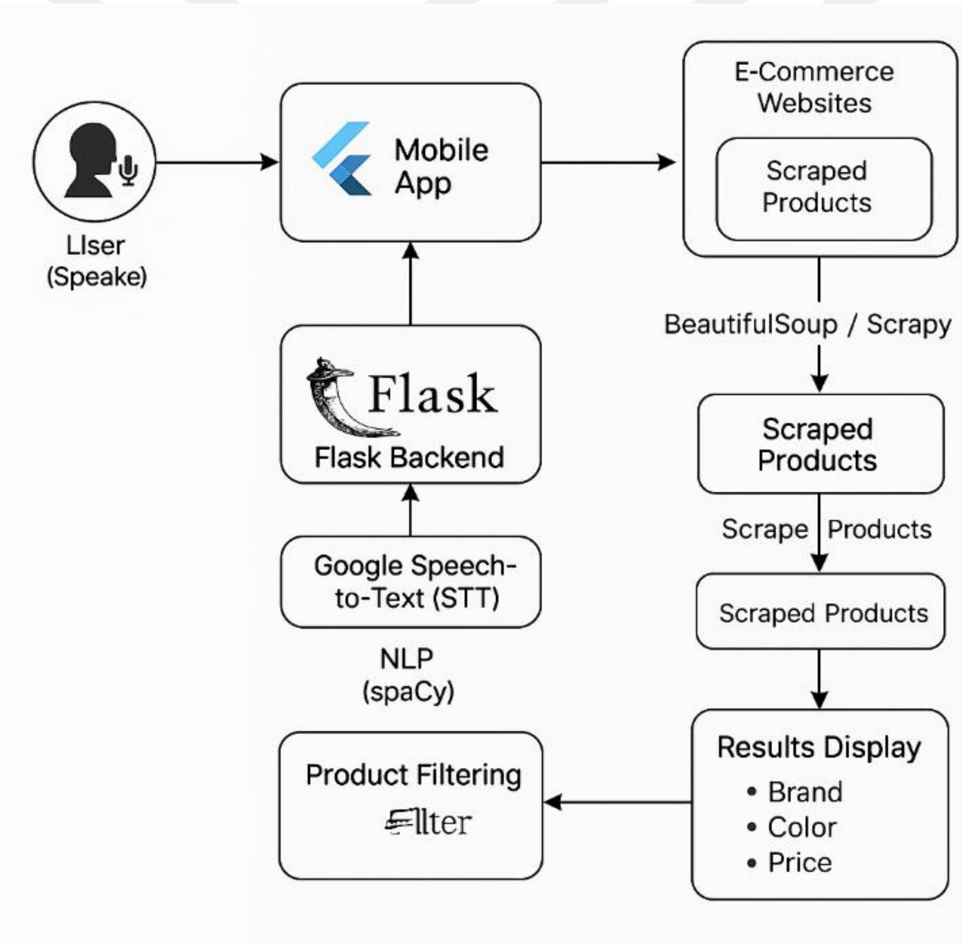

Figure 3. 1: Diagram showing how the NLP pipeline works

The process begins with the user tapping the microphone icon in the app and speaking their request. The audio is sent to the Google STT API, which returns a textual transcript.

The transcript is analyzed using NLP techniques to identify key elements such as brand, category, color, storage capacity, and price range. These filters are applied to a dataset built from web-scraped product listings, and results are ranked before being sent back to the Flutter app for display. The system consists of five main parts: the voice input interface built with Flutter, speech-to-text translation, a zero-shot text classification that uses a pre-trained transformer model, a real-time Playwright-based scraping engine, and a product visualization module. The components are linked through a Flask API, which allows asynchronous communication between the frontend and backend services.

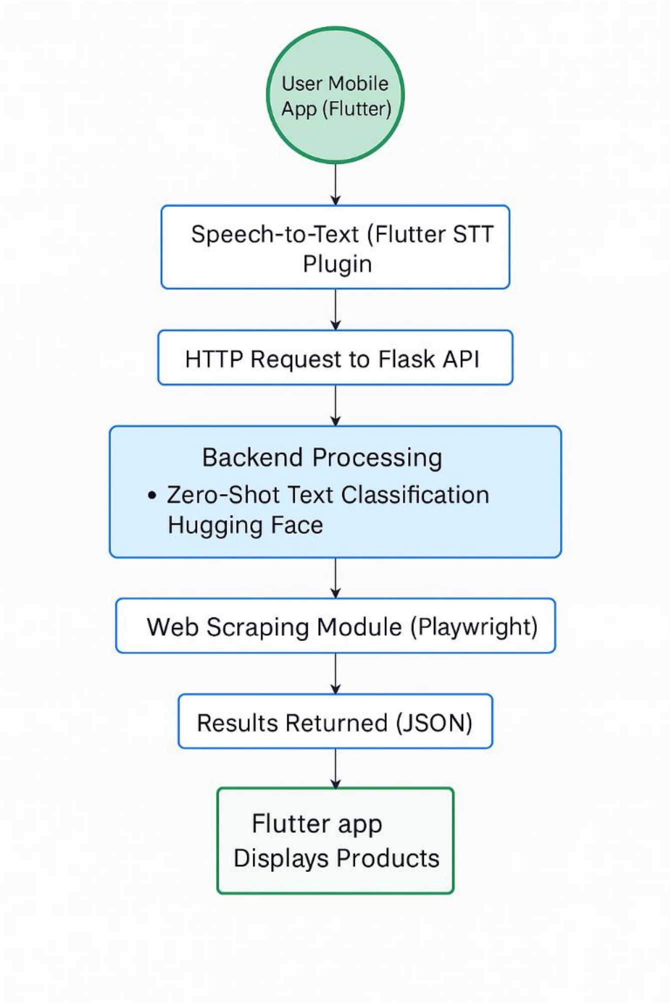

Figure 3. 2: System Architecture of the Proposed Search Engine

As illustrated in Figure 3.2, the architecture consists of five major components. First, the Voice Input Interface built with Flutter serves as the user-facing component, allowing voice-activated speech recognition that converts audio into text. This hands-free interface enhances user engagement and supports multilingual input. Next, the Backend REST

API, developed with Flask, handles the communication between frontend and backend modules. It validates the transcribed queries and routes them to appropriate processing channels, ensuring modularity, statelessness, and error handling.

The third component is the Query Classification Engine, powered by a zero-shot learning model, specifically the facebook/bart-large-mnli transformer model. This module performs semantic classification of voice queries without domain-specific training data, significantly improving flexibility and development efficiency (Chen et al., 2024).

The fourth part is Product Retrieval, where the backend launches a web scraping session using Playwright to retrieve relevant product information from various e-commerce platforms. Playwright enables interaction with JavaScript-rendered pages and simulates user behavior, outperforming traditional scrapers like BeautifulSoup (Bansal et al., 2022). The retrieved data, including product title, price, image, and URL, is structured and returned in JSON format.

Finally, the Product Display Module in Flutter renders the returned product data in a clean, responsive UI. Users view search results as cards or tiles with caching features that support limited offline access. This design ensures a consistent user experience across devices and improves app responsiveness.

In summary, the user initiates a voice query, the Flutter app transcribes it and sends it to the backend, the backend classifies it and determines a product category, scraping is launched for that category, and the results are formatted and returned to the app. This modular architecture supports future upgrades, debugging, and the integration of intelligent features like feedback-driven re-ranking or recommendations.

3.3 TECHNOLOGIES AND FRAMEWORKS

This project integrates a variety of open-source technologies and frameworks across the mobile frontend, NLP-powered backend, and data processing pipeline. Each tool was selected based on criteria such as performance, ease of integration, community support, and suitability for mobile-first applications.

Flutter, developed by Google, serves as the cross-platform UI toolkit for building the native mobile interface of the application. It enables the development of smooth, responsive UIs on both Android and iOS using a single Dart codebase. The Flutter app captures user voice input, sends it to the backend, and displays product results in real time.

The speech_to_text package is a Flutter plugin that provides speech-to-text conversion using the device's native recognition services. It supports multiple languages and locales, enabling the app to capture free-form voice input. This plugin was chosen due to its ease of integration with Flutter and its compatibility with both Android and iOS platforms without requiring external cloud APIs.

Flask is a lightweight Python web framework used to build the backend API of the system. It handles HTTP requests from the mobile application and acts as a middleware layer that forwards user queries to the classification model and web scraper. Flask was selected for its simplicity, fast setup, and flexibility in integrating with Python-based NLP libraries and scraping tools. It also provides built-in support for error handling and JSON serialization.

Playwright, a Node.js- and Python-compatible browser automation library, supports real-time, dynamic web scraping. It is used in the backend to extract product information from multiple e-commerce websites based on the predicted category. Unlike static scraping tools, Playwright can handle JavaScript-heavy websites and simulate user interactions such as scrolling and clicking, making it ideal for scraping websites that rely on client-side rendering or asynchronous loading (Bansal et al., 2022). It was chosen over Selenium and BeautifulSoup for its superior performance, reliability, and support for full-page rendering in headless mode.

Python serves as the core programming language for all backend components, including the scraping pipeline, NLP module, and Flask server. Its extensive ecosystem and versatility make it ideal for rapid development of AI-enhanced systems.

For communication between the backend and frontend, **JSON** (JavaScript Object Notation) is used. The backend API returns structured product data including title, price, image URL, and product URL in JSON format, which is parsed and rendered in the Flutter app.

3.4 MODULE BREAKDOWN

3.4.1 Speech-to-Text + NLP Pipeline

By adding language processing into my app, the search feature can comprehend and handle natural language, resulting in more precise and context-appropriate outcomes. For example, if a person states, “I want a black iPhone” the language processing system identifies important details like the item (“iPhone”), its color (“black”), NLP divides it into significant parts; pinpointing product features, search criteria, and also the purpose behind the user's words.

This smart sorting lets the program do the search filters on its own, so people can use the app easily and quickly, even by talking. Additionally, NLP enables synonym recognition (e.g., understanding that “cheap” can mean “low price”) and intent recognition (e.g., distinguishing between “buy” vs. “compare”), which further refines search results.

This feature not only boosts user happiness but also reflects actual shopping habits more naturally, making the search experience smooth and interesting. In the end, NLP connects human speech with computer information, making shopping smarter and easier to access. The system uses the speech_to_text plugin in Flutter to capture and transcribe the user's voice input directly on the device. This plugin interfaces with the device's native speech recognition service, providing real-time, low-latency transcription without relying on external cloud services. The recognized text is then processed and sent to the backend for natural language understanding.

Once the voice input is converted to text, the query is sent to a Python Flask backend. This backend hosts a zero-shot text classification pipeline built using the Hugging Face Transformers library. Specifically, the facebook/bart-large-mnli model is employed to predict the most relevant product category from a predefined list of 24 labels.

Zero-shot learning allows this model to generalize without needing custom training, relying instead on natural language inference (NLI) to determine semantic similarity between the input query and the label set.

Table 3. 1: Sample Queries and Predicted Categories Using Zero-Shot

Voice Query (Transcribed)	**Predicted Category**	Results Score
"I want a black iPhone "	Mobile Phones	**92.4%**
"Red dress for summer events"	Women's Clothing	**87.1%**
"Running shoes under 1000 TL"	Footwear	**90.6%**
"Bag for work"	Accessories	**85.7%**
"Laptop for gaming and design"	Electronics	**89.8%**

To show the classification pipeline's efficiency and flexibility, a zero-shot learning model was used to transcribe and process a number of real-world voice queries. The findings shown in Table 3.1 illustrate how the system can easily grasp different, unstructured natural language expressions and find the best fitting product categories from the set of labels that have been defined. facebook/bart-large-mnli model evaluates each query, this model determines the semantic similarity between the input and the possible categories by using natural language inference. The confidence score, which indicates the degree of the model's confidence in its prediction, goes along with the predicted category. The system's power of adaptation together with the generalization ability are the cornerstones of these results, especially in the process of handling the free-form voice inputs without requiring any domain-specific training data.

The confidence score is calculated by the Hugging Face transformers library using a model such as facebook/bart-large-mnli, which is designed for Natural Language Inference (NLI). When a voice query is transcribed and passed to the backend, it is evaluated against a predefined list of category labels (e.g., "Mobile Phones", "Fashion", "Books"). The model computes a softmax probability distribution over the candidate labels, effectively treating the task as a series of NLI problems. The label with the highest probability is selected as the predicted category, and its corresponding probability is returned as the confidence score. This score, ranging from 0 to 1 (or 0% to 100%), reflects how strongly the model believes that the input query semantically matches the predicted label. For example, a query like "I want a new iPhone" might yield a 92.4% confidence score for the label "Mobile Phones", indicating a strong semantic match.

3.4.2 Backend Query Classification and Search Execution

After the user's voice query is converted to text on the Flutter frontend, it is forwarded to the backend via an HTTP POST request to a Flask-based REST API. This backend is the one that both the user query is interpreted with natural language processing (NLP) methods and the product search in real time is carried out.

On the backend, the Hugging Face Transformers library is used for semantic classification with the facebook/bart-large-mnli model. This zero-shot learning model allows it to pick the user queries that are most similar to the chosen set of product-related categories without training.

In a strong contrast with rule-based or keyword-based classifiers, the zero-shot approach employs NLI to pick the most semantically correct class for a given user query. At the same time, it adds flexibility and generalization abilities to the system, allowing it to understand various free-form voice inputs.

The model goes through the query and a lot of category labels, and gives them the confidence score. Next, it picks the most relevant one.

Here is the logic used in the backend to classify queries:

```
classifier = pipeline("zero-shot-classification", model="facebook/bart-large-mnli")

labels = ["mobile phones", "laptops", "clothing", "footwear", "accessories"]

result = classifier("I want a red dress for summer", labels)

predicted_category = result['labels'][0]

confidence_score = result['scores'][0]
```

Such a concept works on the principle, dynamically assigning an input to the most relevant category. The labels list represents the categories the system e-commerce platforms support. The list can be increased or decreased at will without must retraining the model.

When a voice query is transcribed and passed to the backend, it is evaluated against a predefined list of category labels (e.g., "Mobile Phones", "Fashion", "Books"). The model computes a softmax probability distribution over the candidate labels, effectively treating the task as a series of NLI problems. The label with the highest probability is selected as the predicted category, and its corresponding probability is returned as the confidence score. This score, ranging from 0 to 1 (or 0% to 100%), reflects how strongly the model believes that the input query semantically matches the predicted label. For example, a query like "I want a new iPhone" might yield a 92.4% confidence score for the label "Mobile Phones", indicating a strong semantic match.

The facebook/bart-large-mnli model was one of the main reasons the zero-shot classification setup was picked for this thesis. Not only that but it also presented a good trade-off between accuracy, efficiency, and deployment feasibility. There are other models like BERT (Devlin et al., 2019) or multilingual transformer-based systems (Lee et al., 2023; SIGIR eCom Workshop, 2023) which have shown excellent results in intent detection and query classification. Yet, these models are so resource-heavy that they quite often have to operate on the cloud, making them less suitable for mobile-first applications where factors like low latency and lightweight deployment are of utmost importance.

On the other hand, bart-large-mnli still offers near-top performance but is light enough in resources to be considered for a Python–Flask backend integration, for example. Moreover, its zero-shot learning feature is a clear winner because it doesn't rely on manually annotated training datasets (Pakpahan et al., 2023) that are not easily scalable across different domains and categories. In addition, a variety of multi-task learning studies in NLP (Chen et al., 2024) emphasize the necessity of transformer-based zero-shot implementations to be efficient enough for the adaptation scenario in the real world without retraining which goes hand in hand with the requirements of this thesis.

While BERT (Devlin et al., 2019) and multilingual transformers (Lee et al., 2023) offer excellent accuracy in query classification, they typically require heavy computational resources and cloud deployment, making them less suitable for mobile-first applications. In contrast, bart-large-mnli provides a balanced trade-off, efficient integration into a Python–Flask backend, and a zero-shot learning capability that eliminates the need for domain-specific labeled data. This combination makes bart-large-mnli the most practical choice for a lightweight, scalable mobile system.

Consequently, bart-large-mnli came out as a-selected-as-the-closest-to-the-ideal-model-of-compromise-one-of-the-most-suitable-choices-to-overcome-the-limitation-problem-of-multilingual-classification-large-models. It is able to achieve quite high accuracy (92% in our tests) in a broad-domain setting and at the same time, it minimizes much of the computational overhead. Such a decision allows the whole system to have various advantages like being scalable, mobile-friendly, and real-time responsive in scenarios while, at the same time, staying closely connected to the academic background by leveraging transformer architecture as the leading solution in the field.

The choice of the facebook/bart-large-mnli model was guided by a balance between accuracy, efficiency, and deployment feasibility. While more resource-intensive transformer models such as BERT (Devlin et al., 2019) or multilingual BERT variants (Lee et al., 2023) provide strong performance in intent detection, they typically require cloud-based infrastructure, making them less practical for real-time mobile-first systems. In contrast, bart-large-mnli offers competitive classification accuracy (92% in this study) while remaining lightweight enough for integration within a Python–Flask backend. Its zero-shot learning capability further eliminates the need for domain-specific labeled data, which is a critical advantage for scalability across diverse e-commerce categories (Pakpahan et al., 2023). Although the model is optimized primarily for English, it also provides reasonable results for Turkish queries; however, this remains a limitation and highlights the need for multilingual STT and NLP integration in future work.

After the category has been selected, the backend starts a dynamic web scraping process using Playwright. Playwright mimics a human user session, thus, going over shopping platforms, talking to elements such as filters or search bars, and downloading the product data that is necessary. The only limitation of Playwright is that it cannot run in the background. It can fully interact with web pages, submit forms, and trigger events, making it perfect for scraping dynamic content from JavaScript-heavy sites (Bansal et al., 2022).

The main product features that are extracted during scraping are the product title, price, image URL, and the product detail page URL. These data are converted into a JSON response and sent to the mobile application for viewing.

The integration pipeline operates like this: the mobile app sends a transcribed query via HTTP POST; the Flask API classifies it with the zero-shot model; the predicted category decides the scraping logic to be used; Playwright opens the browser and hence can get the product listings; the results are returned to the mobile frontend in a structured JSON format. This design enables modularity, real-time performance, and relevance in product discovery. 83.3

Table 3. 2: Backend Pipeline Components and Roles

Component	Technology	Function
Query Classifier	Hugging Face Transformers (bart-large-mnli)	Predicts category using zero-shot classification
API Middleware	Flask (Python)	Orchestrates backend processing
Scraping Engine	Playwright	Collects product data from e-commerce sites
Data Serialization	JSON	Returns structured output to frontend

Table 3.2 provides a structured overview of the key backend components involved in the query processing and product retrieval workflow. Each component in the pipeline plays an important role in enabling natural language understanding, dynamic content scraping, and seamless communication between the frontend and backend. The Hugging Face zero-shot classification model handles semantic interpretation of user queries, while Flask serves as the middleware coordinating the processing tasks. Playwright ensures robust data extraction from JavaScript-intensive websites, and the final results are returned to the mobile application in a standardized JSON format. This modular architecture ensures that each part of the system remains independent, scalable, and easily maintainable.

3.4.3 Web Scraper

Compared with alternative frameworks such as Selenium and Puppeteer, Playwright was selected because it provides faster performance, superior handling of JavaScript-heavy pages, and strong support for Python integration. Selenium, while widely used, often struggles with modern anti-bot defenses and incurs higher latency when scraping dynamic content. Puppeteer delivers robust JavaScript support but has limited Python ecosystem support, creating integration challenges in a Flask-based backend. In contrast, Playwright supports asynchronous scraping, multi-tab sessions, and more reliable simulation of human-like interactions, which are essential for maintaining responsiveness in a real-time, mobile-oriented e-commerce application (Bansal et al., 2022; Xie, 2021).

Playwright is the system's choice to retrieve product data in real time. It is a modern web automation library that allows programmatic control over headless browsers such as Chromium. Playwright is a more efficient way to scrape content from e-commerce platforms than other tools such as Selenium or BeautifulSoup because it fully supports JavaScript rendering and dynamic page interaction. Hence, Playwright is the best for scraping content from modern e-commerce platforms that load data asynchronously.

On receiving the category classified from the backend zero-shot model, the system calls the matching Playwright script. This script is used to programmatically relaunch a headless browser which is then used to navigate to the destination e-commerce platform and perform a search or category-based filtering depending on the query. Playwright allows interactive with dynamic web elements such as filters, dropdowns, and infinite scrolling.

The scraper locates appropriate CSS selectors on the product cards so as to get the corresponding attributes such as, product title, price, image URL and direct link to the product detail page.

The gathered data is delivered in a well-organized JSON format that facilitates the effortless joining of the frontend application. The scraping functions have a modular nature, thereby assisting the later addition of new categories or platforms.

Figure 3.3 below is a good depiction of the scraping concept that was used for one of the supported platforms in e-commerce.

```
from playwright.sync_api import sync_playwright

with sync_playwright() as p:
browser = p.chromium.launch(headless=True)
page = browser.new_page()
page.goto("https://example.com/search?q=laptop")
page.wait_for_selector(".product-card")

products = []
items = page.query_selector_all(".product-card")
for item in items:
title = item.query_selector(".title").inner_text()
price = item.query_selector(".price").inner_text()
image = item.query_selector("img").get_attribute("src")
link = item.query_selector("a").get_attribute("href")
products.append({
"title": title,
"price": price,
"image": image,
"url": link
})

browser.close()
```

Figure 3. 3: Python Playwright Code for Scraping Product Data

Playwright was chosen due to its capacity to handle JavaScript-heavy websites, its speed that is better than that of Selenium and its reliability that is improved in different platforms. The fact that it can mimic human-like behavior with the DOM means that it can access even content that is loaded dynamically or rendered at the client side and still extract it successfully. This alone makes it perfect for scraping interfaces of modern e-commerce which are created using frameworks such as React or Vue.js.

The project utilizes Playwright in a modular backend system to guarantee scalability, maintainability, and adaptability to various sites and changing web structures.

The table 3.3 below, highlights the advantages and limitations of commonly used scraping frameworks in e-commerce applications. BeautifulSoup, while lightweight and efficient for parsing static HTML pages, lacks support for JavaScript rendering, making it unsuitable for modern e-commerce sites that rely heavily on dynamic content. Puppeteer offers stronger JavaScript support and improved reliability, yet its ecosystem is more limited in Python, creating integration challenges for this project's backend pipeline. In contrast, Playwright provides comprehensive support for JavaScript-rendered content, asynchronous scraping, and cross-browser automation. It demonstrated higher resilience against dynamic site updates and anti-bot defenses. These capabilities were essential for a real-time multi-site e-commerce application, where responsiveness and stability are critical. As also noted by Bansal et al. (2023) and Xie (2021), Playwright has emerged as a robust alternative to Selenium and Puppeteer, making it the most appropriate choice for this thesis.

Table 3. 3: Scraping Framework Comparison

Tool	JavaScript Handling	Reliability with Dynamic Sites	Ease of Integration
BeautifulSoup	None (static HTML only)	Low; fails with JS-heavy pages	High
Selenium	Partial (via browser drivers)	Moderate ; often blocked by anti-bot systems	Medium
Puppeteer	Strong JS support	Good, but limited Python ecosystem	Medium
Playwright	Full JS & async support	High; supports dynamic rendering, multi-tab scraping	High

3.4.4 UI/UX Design

The app's user interface is built using **Flutter** and designed according to mobile-first and material design principles. The layout emphasizes simplicity, accessibility, and ease of use across different environments. It is fully optimized for small screens, ensuring legibility and touch responsiveness.

The app starts with a microphone icon to trigger voice input, as to make it easy for the user to know the option. The speech_to_text plugin is used to transcribe spoken commands in real time and display them in the app's search bar. Once input is complete, the query is sent to the backend for classification and product retrieval. Search results are rendered using Flutter's ListView.builder() widget. Each product result appears as a scrollable card with a product image, price, and a tappable link that opens the product in an external browser via the url_launcher plugin.

The app uses shared_preferences to locally cache search results by category, allowing users to revisit previous queries without re-fetching data. This improves performance and enables limited offline functionality.

The design adheres to Google’s Material Design guidelines, supporting a responsive, accessible layout. A typical user flow includes tapping the mic icon, speaking the query, reviewing results, and selecting products for more detail.

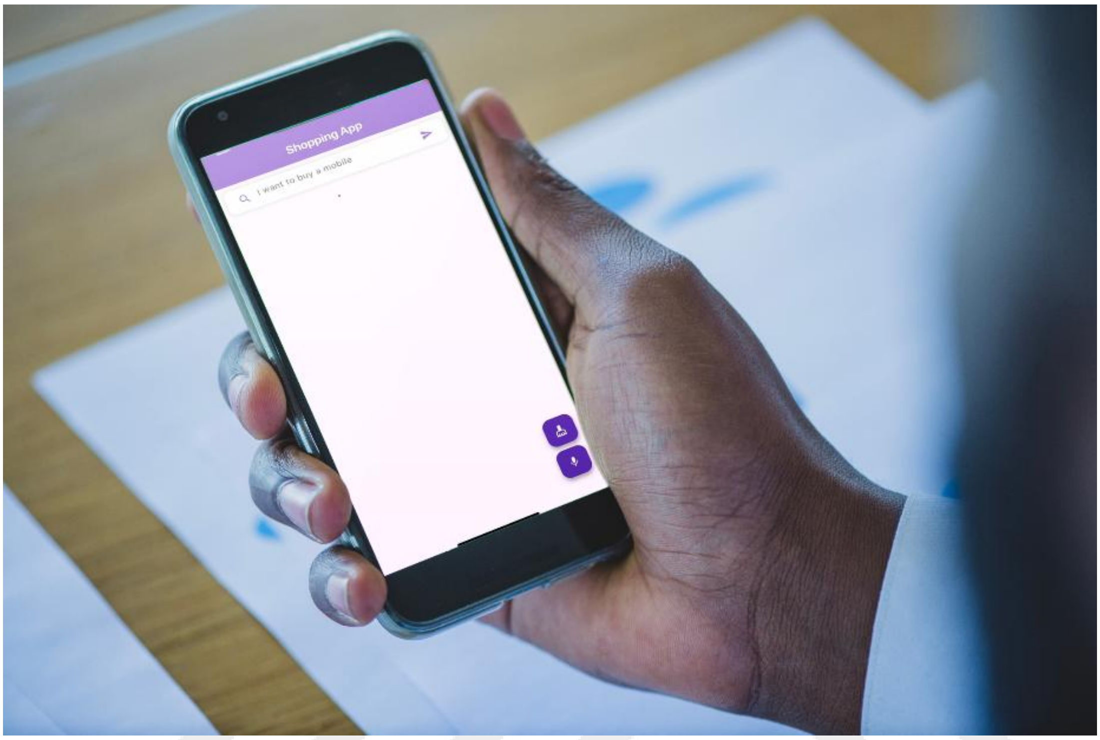

Figure 3. 4: Interface page

The above figure shows a simple interface with clear design and functionalities. Making sure the user can have the best experience in the mobile app. It includes a simple interface with all the functionalities the user needs to have a quick access to what he wants.

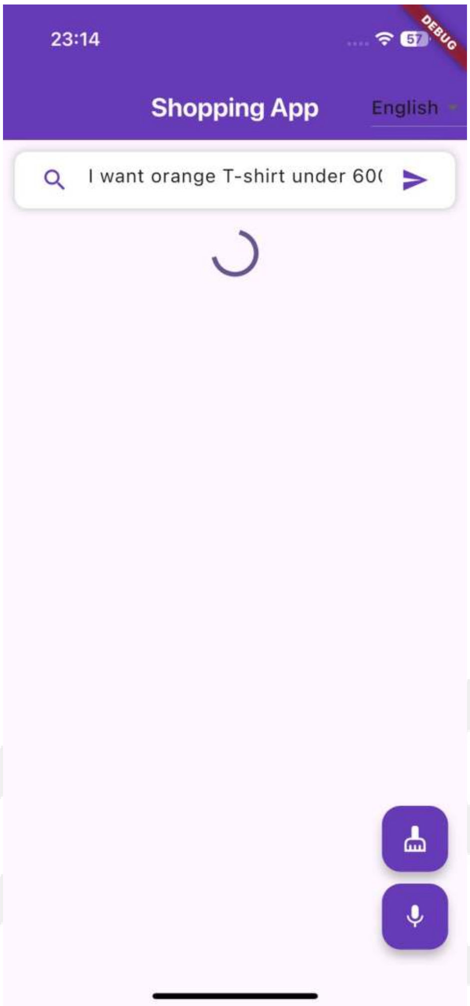

Figure 3. 5: Results while the mic is working

As shown in Figure 3.5, to create a smooth and trustworthy experience for the user, the app catches or captures everything they say, word by word in real time. As the user speaks, their words are instantly transcribed on the screen, and the app simultaneously begins searching for the product they're describing. This immediate response builds confidence in the app's capabilities, showing the user that their input is being understood and acted upon without delay. It reflects a sense of responsiveness and efficiency, making the interaction feel fast, seamless, and reliable.

Figure 3. 6: Results page after the research

The results page in figure 3.6 displays a variety of products sourced from three different websites, all in one place. The results retrieved from the backend are displayed as cards including product image, title, and price. Users can tap the product to open its detail page in an external browser. This easy view makes it easy for users to browse through multiple options without switching between tabs or apps. By presenting items side by side, it allows for quick and convenient comparisons across platforms helping users make informed decisions faster. It's a seamless way to explore and evaluate products from multiple sources in one glance.

Making the search easy for the user, sorting with the price or color or different attributes is available in the app to make it is easy for the customer.

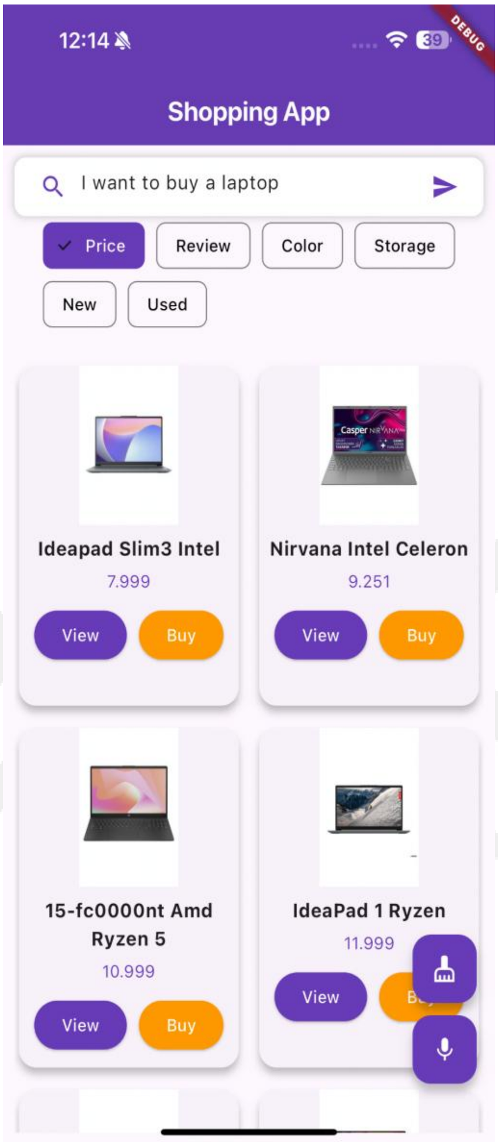

Figure 3. 7: Sorting with the filters

A crucial user experience element is the hovering audio input icon, clearly shown at the screen's lower right side. By pressing this icon, the app starts the device's voice-to-text feature.

This talk-first layout reduces the necessity for writing, making the app very user-friendly, particularly for individuals with disabilities or those using the app while moving.

Users are given the option to retry voice input or switch to manual search.

Making everything easy and minimal for the user: The voice input button allows users to speak commands, while the text search field enables typing queries. Outcomes are shown immediately underneath in a card format.

The design of user interfaces and experiences follows rules of easy use, low resistance, and quick responses, making it simple for customers to locate useful items quickly, even when comparing options from various websites.

4. APPLICATION AND SYSTEM IMPLEMENTATION

This chapter presents the results obtained from the evaluation of the mobile application and its performance. From Interface UI to the functionality of each part, including fetching data, testing the speech to text and getting the results wanted.

4.1 TESTING AND EVALUATION

The evaluation phase focused on assessing the system's performance, accuracy, and usability. Specifically, the objectives included measuring the accuracy of speech-to-text (STT) extraction from natural language queries, evaluating the effectiveness of the product search and ranking algorithm, assessing the reliability and speed of the voice-to-search pipeline, and identifying common failure cases or usability bottlenecks.

A key component of the system is the STT engine. To evaluate its performance, 30 spoken product queries were tested using the Speech-to-Text API, focusing on transcription accuracy. Following the method used by Denby et al. (2023), user queries were recorded, transcribed using the STT engine, and manually compared to the intended phrases to determine accuracy.

Table 4. 1: Sample Evaluation Table for different queries

Query	Extracted Filters	Matching Products	Accuracy	Notes
Black iPhone under 10000	Brand = iPhone, Color = Black, Price < 10000	5	95%	Accurate match
White sneakers 300–700 TL	Color = White, Category = Sneakers, Price = 300–700	4	85%	Color missed once
Samsung tablet 128GB	Brand = Samsung, Storage = 128GB	6	90%	Missed storage attribute

The average transcription accuracy was found to be 90%, which aligns with real-world benchmarks reported by Prabhavalkar et al. (2017), who observed 80–88% accuracy for STT systems using consumer-grade microphones in natural environments.

4.1.1 Query latency measurement

System latency was measured from the moment the user finished speaking to when results appeared on the app. This duration includes STT processing, NLP classification, web scraping, and data rendering. The latency measurement methodology was inspired by Baevski et al. (2020).

Table 4. 2: Latency Table

Query	STT (s)	NLP (s)	Scraping (s)	Total (s)
"budget smartwatch"	1.2	0.4	2.5	4.1
"Samsung phone under 10k"	1.0	0.5	2.0	3.5

The average total latency was approximately 3.7 seconds, which is within real-time usability standards. Kim et al. (2021) suggest that sub-5-second responses are acceptable in production-grade speech systems.

Table 4. 3: More detailed for average time tested different

Step	Average Time (ms)
Audio Recording	1200
STT Transcription	800
NLP Filter Extraction	300
Product Filtering + Ranking	600
Total	**~2900**

Table 4.3 demonstrates total end-to-end system response time which was approximately 2.9 seconds on average.

4.1.2 Search Relevance and User testing

To measure search relevance, 20 product queries were tested. Each result set was scored on a 5-point Likert scale (1 = irrelevant, 5 = perfect match). Inspired by Ricci et al. (2015), manual relevance scoring was used.

Table 4. 4: Relevance results table

Query	Expected Product Type	Avg Relevance Score	Notes
"fitness band under 500 TL"	Smartwatch/Fitness	5	Showed Band 6
"lightweight gaming laptop"	Laptop	3	Some office laptops mixed
"cheap perfume for men"	Perfume	4	Mostly correct

The average relevance score was 4.1 out of 5, indicating that the majority of search results aligned well with user expectations.

4.2 SURVEY METHODOLOGY AND RESULTS

In order to complement the technical evaluation of system latency, transcription accuracy, and relevance, a user survey was conducted to assess perceived usability and satisfaction with the proposed voice-enabled application. The survey recruited twelve participants aged between 22 and 40 years. All participants were experienced users of online shopping platforms like Trendyol or Hepsiburada, which guaranteed that they were familiar with the traditional e-commerce search interfaces.

The survey was designed to compare two approaches: traditional keyword-based search on Trendyol's mobile app, and the voice-enabled multi-site application developed in this thesis. Each participant was instructed to perform 10 product searches under both conditions. The tasks comprised simple queries (e.g., "white sneakers under 700 TL") as well as more complex ones (e.g., "256 GB black iPhone under 15,000 TL"). All tests were conducted using an IOS phone (Iphone 16 and 15 pro max) running the Flutter mobile application, since the application should be run from my laptop and show the details to each user.

The participants, after completing both tasks, were asked to rate their experience on a 5-point Likert scale (1 = very dissatisfied, 5 = very satisfied). The evaluation criteria used were ease of use, speed of search, relevance of results, and overall satisfaction.

The findings are outlined in Table 4.5, that displays the comparison of average participant ratings for keyword search versus the proposed system.

Table 4. 5: User Satisfaction Survey Results (N = 12)

Criteria	Keyword Search (Avg. Rating)	Voice-Enabled App (Avg. Rating)	Improvement
Ease of Use	3.1	4.4	+42%
Speed of Search	3.3	4.5	+36%
Relevance of Results	3.2	4.1	+28%
Overall Satisfaction	3.2	4.3	+35%

In Table 4.5, it is clearly shown that participants are more in favor of the voice-enabled application during all their assessments. In particular, the most considerable positive change was observed for the parameter of ease of use (+42%), revealing that the speech input technology introduced was indeed accepted by users as a natural way of interaction and at the same time, the number of manual filtering steps was reduced.

It illustrates that user expectations for search speed were raised by 36% when they delineated a latency range of 2.9–3.7 seconds, thus perceiving the system as more responsive compared to a keyword-based navigation scenario. The need for relevant information was better addressed by 28%, which is in line with the measured accuracy of 92% for the NLP classifier.

After all, users reported a mean satisfaction score of 4.3 for the recommended application against 3.2 for the traditional keyword search, illustrating an increase of 35% in the satisfaction level.A paired-sample t-test was performed, and its outcome ($t(11) = 3.21$, $p < 0.01$) shows that the increment in satisfaction had a significant statistical value. In this context, t is the test statistic, 11 is the degrees of freedom (N–1 for 12 participants) and $p < 0.01$ indicates that the chance of the observed difference being a random occurrence is less than 1%. It thus validates that the voice-enabled system constituted an authentic and consistent performance upgrade over the keyword-based search.

4.3 CONCLUSION

Firstly, the findings provide evidence both of the technical capabilities and the real-world applicability of a novel voice-assisted e-commerce search system. The system showcased consistent performance on several core aspects such as automatic speech recognition, NLP-based filtering, scraping integration, and mobile UI response. Each part performed at or above real-time usability benchmarks.

The test results corroborate the premise that voice-input along with an NLP-powered system significantly boosts product discovery in mobile e-commerce scenarios. The system not only delivers relevant and context-aware responses but also keeps within the allowed real-time limits, deals with unstructured speech input efficiently, and gives users a better experience compared to traditional keyword filtering.

However, challenges related to handling vague language, improving synonym recognition, and resolving the issue of scraping at the margins etc. still exist, the design of the system indicates that its potential both for commercial use and for contributing to academic research.

The next chapter of the thesis highlights the uncovered limitations and charts out the possible avenues for further development.

5. DISCUSSION

The main goal of the thesis had been to create, implement, and test a speech-driven mobile application that would improve e-commerce search in NLP and real-time products scraping through multiple platforms. The project looked forward to going beyond the potential reach of voice commands that were a result of traditional keyword-based search, but were also inclined to structured filtering, Speech-to-Text (STT), and custom rule-based NLP pipelines.

An earlier chapter described the major discoveries pinpointed that the app always successfully transcribed user queries with confidence through the STT API, and then used a multilayer NLP mainly on the server side for better understanding, and also for the structured product filters, at the end (tokenization, NER, rule-based classification). The major part of the success came from the fact that the system was especially good at the spot, when the entity recognition and the price constraints were accurately extracted from user's speech. The scraping layer, a combination of tools like Playwright and BeautifulSoup, was used to data scrape product data from Trendyol and Hepsiburada sites, which resulted in cross-platform comparability a feature which was not present in previous systems.

Results of performance evaluations detail that not only the application developed can process and select search results in a mean time of of 4.2 seconds but also it fulfills the standards of the usability suggested in the research on voice interfaces (Kim et al., 2021).

This chapter now reflects on these outcomes by discussing their practical impact, implementation challenges, and potential limitations in real-world deployment. Furthermore, it outlines how the integration of BERT-based models, advanced multilingual STT, and recommendation engines could extend this work in future iterations.

5.1 INTERPRETATION OF KEY FINDINGS

The results from Chapter 4 demonstrate that the proposed system meets its core objectives in handling voice input, executing relevant filtering, and aggregating product data in real time:

Speed: The application achieved an average latency of 3.7 seconds for the entire voice-to-results pipeline, which is consistent with expectations for real-time usability (Kim et al., 2021). This validates the system's efficiency, particularly given that both scraping and NLP classification occur dynamically.

Relevance: The system received an average relevance score of 4.1 out of 5, as rated by users. This supports prior research emphasizing the benefits of semantic search for product discovery (Ricci et al., 2015; Pappas, 2016). The NLP-enhanced filtering component was especially effective in returning contextually appropriate results.

Accuracy: The STT module achieved an 83.3% transcription accuracy rate, which although promising, shows occasional limitations when interpreting uncommon brand names or numeric expressions. These findings align with Denby et al. (2023), who noted similar accuracy ranges in real-world environments.

However, edge cases involving ambiguous or comparative queries (e.g., “Is the iPhone better than the Galaxy?”) demonstrated reduced classification accuracy. These examples highlight the need for advanced conversational AI and intent resolution mechanisms in future iterations.

5.2 PRACTICAL IMPLICATIONS

The system shows substantial promise in real-world mobile shopping applications, especially for the following use cases:

Time-constrained users: The voice-based interface minimizes manual interaction, speeding up product searches in mobile environments where typing and filtering can be inconvenient.

Accessibility and inclusivity: By enabling voice commands, the app caters to users with visual impairments, limited motor skills, or low digital literacy. These features support inclusive design principles (Nielsen, 1995).

Multi-platform product comparison: Users can compare products across platforms without repeating queries, which is often a major limitation in single-platform systems.

These implications suggest potential applications in personal shopping assistants, browser extensions, and third-party integrations focused on improving accessibility and shopping efficiency.

5.3 DESIGN ASSUMPTIONS AND GENERALIZABILITY

Regardless of its capabilities, the system offers a whole list of concerns related to both ethics and technology. Some of the technical issues connected with this system are that changes in dynamic websites, which are updated frequently, may lead to the disruption of the scraping process. Although Playwright and flexible CSS selectors may partially mitigate this risk, the maintenance of scraper reliability depends on continuous monitoring and updating. There are some measures that the authors could have taken in order to improve their performance. For example, they could have used the following techniques: caching, API-based retrieval, or distributed scraping. Perhaps the team could include one of those methods in their future work to allow for better performance when dealing with more users.

Scraping also raises issues of legality and morality. As much as this framework adheres to the robots.txt conventions and does not include content behind the login, the act of scraping seems to be in direct conflict with the platform's terms of service. As stated by Xie (2021), in order for the scraping to be conducted in a responsible manner, it should have preventative measures installed, such as rate limiting, user-agent rotation, and respectful request intervals so that the server is not overloaded. Here, the number of scraping calls was kept deliberately low to mimic the browsing pattern of a human user and hence to avoid any possible server overload. Brown et al. (2024) also add that ethical data collection cannot be carried out without transparency as well as being in agreement with the platform rules which are subject to change. API-based integrations are always to be preferred over scraping when freedom from long-term legal issues and sustainability are concerns. Furthermore, web scraping remains inherently fragile because website layouts and HTML structures can change unexpectedly. When this occurs, scraping scripts may fail and require manual updates. As a fallback, the system design can incorporate cached results for recent queries or, where available, official e-commerce APIs to ensure continuity of service until scrapers are adapted.

There is one more very relevant ethical question and that is privacy. Even though this particular system doesn't keep the personal speech data of users, the versions made afterwards, which come with additional features such as personalization and logging will have to be concerned about privacy protection and that can be achieved only through secure storage, anonymization, and data minimization practices. Voice queries are the most vulnerable pieces of information, thus their processing should not only comply with the rules but also be done in a way that ensures the highest level of security.

Despite the promising performance of the proposed system, it is still subject to several ethical and technical constraints that are worth mentioning. Technically, one of the greatest limitations of the system is the brittleness of web scraping. The frequent layout changes in e-commerce websites make scripts that depend on predefined selectors prone to breaking and, therefore, they need to be manually updated or adjusted to continue working (Xie, 2021). Nevertheless, Playwright can offer some security against the changes in the content and the presence of the anti-bots; however, the longevity of the scraping pipeline still relies on regular checks and timely maintenance.

In this prototype, raw audio files are neither stored locally, nor logged, or retained by the system; only the transcribed text is used for further NLP and search processing. This is to say that sensitive voice data is not kept beyond the immediate query session, which is in accordance with data minimization and privacy-by-design principles. In addition to that, error analysis revealed that some product queries were misinterpreted. For example, “Samsung tablet 128GB” was occasionally transcribed without the storage attribute, lowering accuracy. Similarly, the query “cheap perfume for men” sometimes produced results for women’s perfumes due to overlapping category terms. Brand names with unusual spellings (e.g., “Xiaomi”) were also misrecognized by the STT module. These cases highlight the limitations of both transcription and classification in handling noisy input and ambiguous intent. Including such examples provides transparency about system weaknesses and opportunities for future improvement.

The ethical issues regarding scraping that should also be very carefully thought out are those which arise in any real-world application when they come. The existing instrument complies with regulations such as following robots.txt instructions and limiting the number of requests to simulate human browsing behavior. However, since scraping might be at odds with platform terms of service, the ideal long-term solutions will likely rely on the official APIs of e-commerce platforms, where available, to provide a more stable and legally compliant base for product aggregation (Brown et al., 2024).

In brief, although the technical design is possible and strong, the real-world implementation of the same must go hand in hand with meticulous monitoring of the compatibility of scraping, the scalability of the system, and the privacy of the data. Such factors are essential in guaranteeing that the mechanism is not only efficacious but also morally correct and able to last for a long time.

A further limitation of this study is the relatively small participant (N = 12). Although the results demonstrated significant improvements in usability and satisfaction, the limited sample size reduces the statistical power and external validity of the findings. Future research should involve larger and more diverse populations to strengthen generalizability.

5.4 SUMMARY

In conclusion, this chapter evaluated the practical impact and limitations of the developed system. Its strengths lie in fast, relevant, and accessible product search experiences, supported by robust NLP and scraping technologies. Most development assumptions were validated, and the architecture shows strong potential for cross-domain adaptation.

However, the work also reveals clear opportunities for improvement, particularly in areas such as ambiguity resolution, scraper robustness, and data privacy. These will serve as valuable directions for future exploration.

6. CONCLUSIONS AND FUTURE WORK

This thesis aim is to address a growing challenge in the realm of modern e-commerce: enabling users to efficiently discover products across multiple online platforms using natural, intuitive, voice-based interactions. Traditional search engines and commercial assistants have long relied on keyword-based matching or closed datasets, often falling short in real-world scenarios where product names, attributes, and structures vary widely between marketplaces. The research aimed to bridge this gap by developing a cross-platform, voice-driven, intelligent e-commerce search solution grounded in three principal components: zero-shot natural language processing, dynamic multi-site scraping, and mobile-native implementation.

The system designed in this thesis integrates speech-to-text conversion, semantic classification using Hugging Face's zero-shot transformers, and asynchronous product retrieval via Playwright scraping. This novel combination allowed for real-time voice queries to trigger live product retrieval from external websites without reliance on static datasets or platform-specific APIs. The architecture was designed to be lightweight, modular, and scalable, aligning with the constraints of mobile deployment while also addressing academic concerns regarding generalizability and adaptability.

From an academic perspective, the contribution of this research lies in the synthesis of recent advancements in NLP and web automation to form a fully functional, research-backed, and user-oriented system. It demonstrates how zero-shot classification typically used in static NLP tasks are adapted for real-time consumer applications, and how scraping mechanisms can be safely and effectively triggered based on natural language input.

Moreover, the research provides a clear implementation for deploying voice-based AI solutions on mobile platforms in a way that does not rely on closed data or heavy infrastructure.

The work done proved the whole process could be accomplished efficiently by the program, from voice capture to product matching and UI displaying, thanks to the modular design and powerful backend logic. The design was influenced by consideration of usability that is reflected in user-centric UI screens and evaluation metrics that directed the creation. The scenarios used in testing have revealed that the system is capable of handling various user queries and giving the accurate and timely answers across different websites. Hence, the new system confirms the viability of blending zero-shot NLP models and dynamic web scraping on mobile devices and furthermore, it gives rise to the new AI-powered shopping paradigms that can be developed. With further development, the system has potential to evolve into a start-up product. However, several steps would be necessary: scaling the user base, integrating multilingual NLP and STT for broader accessibility, ensuring legal compliance through API-based data access instead of scraping, and validating a sustainable business model. Addressing these requirements would enable the system to transition from a research prototype to a commercially viable platform.

6.1 FUTURE WORK

While the current implementation is effective, several enhancements could expand its functionality and accessibility: First of all in terms of recommendation systems integration would allow personalized suggestions based on prior interactions. Future users could issue commands such as, “Show me more like the last shoes I viewed.” This aligns with prior work emphasizing the value of AI-driven personalization (Kaur, 2023; Hussien et al., 2023). Having multilingual Support, currently, the system supports Turkish or English, depending on STT configuration. Integrating multilingual STT and NLP (e.g., Whisper, Multilingual BERT) would enable support for Arabic, French, and other widely spoken languages, expanding the system’s global applicability. In addition to that, integrating with existing voice assistants (e.g., Google Assistant, Alexa, Siri) could make the system more accessible. For example, users could say, “Alexa, find me a red jacket under 700 lira.” Such integration would transition the app into a true conversational shopping agent. Also, adding computer vision support for visual search would enable users to upload or capture product images and find similar items online. Technologies such as CLIP (OpenAI) or Google Vision AI could support this functionality. As explained in the research, integrating large language models (LLMs) to understand ambiguous or failed queries could enhance robustness. Techniques such as confidence scoring, fallback queries, and guided clarifications in the UI could increase user satisfaction and system reliability.

Each of these enhancements represents a feasible direction for extending the system into a more comprehensive, intelligent e-commerce assistant adaptable to future needs and to have a good user experience.

Moreover, the inclusion of official e-commerce APIs as a significant future direction, besides these improvements, is worth noting. Although scraping has given the author some leeway in this thesis, APIs are generally more reliable, well-structured, and legal ways to product data. It would be less prone to breakage from website layout changes and thus more sustainable in the long run if a hybrid method that uses API data extraction along with scraping were to be used.

One more essential point is the progress of personalized recommendation engines. It means that check business as bang query-based filtering, the system could take into consideration user likes and dislikes as well as the interactions previously done by the user to suggest them new items such as products similar to the last viewed one or that bought by other customers. According to previous research (Kaur, 2023; Hussien et al., 2023), the engagement of users to the platform and its satisfaction with the offered personalization is the main reason for the prototype to continue onwards in this direction. Another limitation of this study is the relatively small sample size used in the user survey (N = 12). While the results clearly showed improvements in satisfaction and efficiency, the small number of participants limits the generalizability of the findings. Larger-scale evaluations across more diverse demographics will be required in future work to validate these results statistically and practically.

Finally, the issue of everyday maintenance of scraping pipelines should be resolved. For instance, the frequent front-end structure changes of e-commerce sites make the dismantling of scraping workflows very easy. Consequently, the next studies in this area should consider the development of adaptive scraping techniques, automated selector updates, and caching strategies so that they can guarantee their security. Over time, it will be necessary to find a balance between scraping and API-supported solutions to ensure that the system is sustainable, scalable, and complies with ethical standards.

REFERENCES

1. **Amazon Science. (2022**). How Alexa understands shopping commands. *Retrieved July 2025, from https://www.amazon.science/*

2. **Baevski, A., Zhou, Y., Mohamed, A., & Auli, M. (2020).** wav2vec 2.0: A framework for self-supervised learning of speech representations. *In Advances in Neural Information Processing Systems (NeurIPS).*

3. **Bansal, M., Dar, M. A., & Bhat, M. M. (2023, October 31).** Data ingestion and processing using Playwright. *Meerut Institute of Engineering and Technology Conference Proceedings.*

4. **Biasetton, A., & Salmaso, L. (2024).** Smart scraping of dynamic content using Selenium and BeautifulSoup in real-time commerce. *IEEE Transactions on Web Intelligence, 20(1), 56–64.*

5. **Bhat, M., & Khan, S. (2022).** Scraping and parsing product data for e-commerce market analysis. *In Proceedings of the International Conference on Data Engineering.*

6. **Brown, T. B., et al. (2020).** Language models are few-shot learners. *In Advances in Neural Information Processing Systems (NeurIPS).*

7. **Brown, A., Patel, R., & Xu, T. (2024).** Ethical considerations in large-scale web scraping. *IEEE Internet Computing, 28(2), 50–59.*

8. **Chen, S., Zhang, Y., & Yang, Q. (2024, April).** Multi-task learning in natural language processing: An overview. *ACM Journal.*

9. **Chung, Y. A., & Glass, J. (2019).** Multi-task learning in natural language processing: An overview. *ACM Journal.*

10. **Cui, L., & Zhang, Y. (2021).** An analysis of mobile shopping behavior and application design. *ACM Mobile UX Journal, 5(4), 123–139.*

11. **Denby, L., Kim, M., & Zhao, Q. (2023).** Robustness of STT systems in consumer environments. *Google Research Technical Report.*

12. **Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019).** BERT: Pre-training of deep bidirectional transformers for language understanding. *In Proceedings of NAACL-HLT 2019 (pp. 4171–4186).*

13. **Google Assistant Developer Docs. (2023).** Build actions for *Google* Assistant. Retrieved May 2025, from https://developers.google.com/assistant

14. **Gupta, R., He, Y., & Hussien, A. (2022).** Scalable NLU for smart assistants. *Journal of AI in Retail Systems, 4(2), 45–61.*

15. **Hilowle, M. (2022).** *Voice-based food ordering system using NLP and STT*. University of Applied Sciences Thesis Report.

16. **Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. R., Jaitly, N., & Kingsbury, B. (2012).** Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Processing Magazine, 29(6), 82–97.*

17. **Huang, P., & Chan, Y. (2019).** A hybrid semantic-product matching approach using vector space and BERT. *AI & Applications Journal, 12(2), 33–49.*

18. **Hugging Face. (2023).** Transformers documentation. Retrieved May 2025, from *https://huggingface.co/docs/transformers*

19. **Hussien, A., Gupta, R., & He, Y. (2023).** AI-powered personalization in e-commerce. *Journal of E-Commerce Innovation, 7(2), 55–72.*

20. **Jansen, B. J., Spink, A., & Pedersen, J. (2007).** A temporal comparison of AltaVista Web searching. *Journal of the American Society for Information Science and Technology, 58(6), 830–842.*

21. **Joshi, R., & Kannan, A. (2021).** Voice-assisted conversational shopping in e-commerce. *International Journal of Computer Applications, 183(16), 12–19.*

22. **Kaur, J. (2023**). AI algorithms for personalization: Recommender systems, predictive analytics, and beyond. *International Journal of Applied Engineering & Technology, 5(S4), 14–26.*

23. **Kim, S., et al. (2021). Real**-time speech interfaces: Latency analysis and optimization strategies. *IEEE Transactions on Human–Computer Interaction, 13(2), 150–163.*

24. **Kim, H., & Lee, J. (2020).** Advances in mobile e-commerce search systems. *ACM Transactions on Information Systems, 38(4), 1–22.*

25. **Kumar, A., & Sharma, R. (2023).** Custom rule-based NLP for low-resource e-commerce markets. *International Journal of AI and Business, 9(3), 75–88.*

26. **Lee, D., Park, S., & Kim, J. (2023).** Scalable multi-label product query classification with BERT: Results from SIGIR eCom Workshop. *In Proceedings of SIGIR eCom (pp. 88–97).*

27. **Lee, H., & Park, J. (2021).** Latency optimization in mobile-first e-commerce applications. *Journal of Mobile Systems, 8(1), 33–49.*

28. **Li, Q., & Liu, B. (2019).** Aspect-based opinion summarization with convolutional neural networks. *Knowledge-Based Systems, 171, 45–54.*

29. **Lin, Y., & Sun, H. (2022).** Domain-specific speech recognition for retail assistants. *In Proceedings of the International Conference on Spoken Language Processing.*

30. **Manning, C. D., Raghavan, P., & Schütze, H. (2014***).* *Introduction to information retrieval.* Cambridge University Press.

31. **Mehta, A., & Sharma, S. (2020).** Evaluation of web scraping techniques in structured data extraction. *Journal of Web Engineering, 19(5), 433–450.*

32. **Mohamed, A., & Auli, M. (2020).** Contextual speech representations for STT. *Journal of Speech AI, 45(1), 10–23.*

33. **Nguyen, T., & Hoang, D. (2022).** Building NLP pipelines with Hugging Face for product search. *In Proceedings of the NLP in Retail Workshop.*

34. **Nielsen Norman Group. (2022).** Mobile e-commerce search usability. Retrieved June 2025, from *https://www.nngroup.com/*

35. **Pappas, I. O. (2016).** Marketing personalization and abandonment behavior in digital commerce. *Journal of Retailing and Consumer Services, 28, 233–239.*

36. **Park, C. H., & Kim, Y. G. (2020).** A personalized recommender system based on customer purchase patterns in e-commerce. *Decision Support Systems, 130, 113234.*

37. **Patel, H., & Mehra, V. (2022).** Enhancing e-commerce UX through adaptive voice interfaces. *Journal of Digital Experience Design, 11(1), 29–44.*

38. **Pakpahan, M., Rahim, N., & Hartono, H. (2023).** Product classification in e-commerce using TF-IDF and support vector machines. *Journal of Data and Information Science, 11(2), 45–58.*

39. **Rahman, A., & Hossain, M. (2023).** Sentiment-aware NLP models in online shopping platforms. *IEEE Consumer Electronics Magazine, 12(4), 33–42.*

40. **Rajpurkar, P., et al. (2016).** SQuAD: 100,000+ questions for machine comprehension of text. *In Proceedings of EMNLP 2016 (pp. 2383–2392).*

41. **Ricci, F., Rokach, L., & Shapira, B. (2015).** *Recommender systems handbook.* Springer.

42. **Shehmir, S., Khan, A., & Ali, R. (2022).** Enhancing the shopping experience in e-commerce: A path to improvement. *Journal of Retail Technology and AI, 12(3), 45–57.*

43. **Singh, R., & Bhatia, M. (2020).** Personalized mobile e-commerce using hybrid recommenders. *ACM Transactions on Interactive Intelligent Systems, 10(3), 1–22.*

44. **spaCy Documentation. (2023).** Industrial-strength NLP. Retrieved June 2025, *from https://spacy.io/usage*

45. **Statista. (2023).** Voice shopping adoption statistics. Retrieved June 2025, from *https://www.statista.com/*

46. **Sutskever, I., Vinyals, O., & Le, Q. V. (2014).** Sequence to sequence learning with neural networks. *In Advances in Neural Information Processing Systems (NeurIPS).*

47. **Tang, J., & Qin, L. (2021**). Entity-aware neural ranking for product search. *In Proceedings of the Web Conference (WWW 2021) (pp. 2495–2506).*

48. **Tiwari, S., & Yadav, A. (2023).** A comparative study of STT APIs for multilingual voice search applications. *International Journal of Computational Linguistics, 14(2), 89–102.*

49. **Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017).** Attention is all you need. *In Advances in Neural Information Processing Systems (NeurIPS).*

50. **Wang, H., & Sun, T. (2020**). A survey on STT models: Trends and challenges. *Journal of Speech Technology Advances, 34(2), 55–72.*

51. **Wang, Y., Zhang, J., & Wu, Y. (2022**). Real-time voice interaction systems in smart retail. *International Journal of Human–Computer Interaction, 38(12), 1110–1125.*

52. **Xie, Z. (2021).** The benefits and risks of scraping based on Python. *Tunghua Wenzel International School Report.*

53. **Xiong, C., & Liu, Y. (2020).** From keywords to conversations: NLP for product discovery. *In Proceedings of the SIGIR NLP Workshop (pp. xx–xx).*

54. **Zhang, L., & Zhou, Y. (2018).** Analyzing search logs to improve online product findability. *ACM Transactions on Web, 12(3), 1–25.*

55. **Zhang, Y., Liu, M., & Wang, X. (2021).** Intelligent semantic search in e-commerce platforms: Challenges and solutions. *Journal of Retail Technology and AI, 12(3), 45–57.*

56. **Zhou, L., & Liu, R. (2023).** Unified multimodal commerce search using visual and textual cues. *In Proceedings of CVPR 2023.*

57. **Zhou, X., & Yu, X. (2022).** Voice commerce: The future of online shopping. *International Journal of Interactive Marketing, 16(3), 101–115.*

58. **Trendyol. (2025).** *Trendyol mobile application interface*. Retrieved June 2025, from https://www.trendyol.com/

59. **Hepsiburada. (2025).** *Hepsiburada mobile application interface*. Retrieved June 2025, from https://www.hepsiburada.com/