



## Graduation Project Report



### ELEVATE

– Egyptian Local E-commerce Visualization and Augmented Technology Experience –

#### Presented by:

Adham Khaled Fahmy	20210050
Belal Ahmed Eid	20210092
Seif Yahia Al-Gohary	20210172
Shawky Ebrahim Ahmed	20210184
Nour Tarek Sayed	20210427
Nour Wael Elshaer	20210429

#### Under the supervision of:

Assoc. Prof. Iman Mohamed Atef Helal  
Eng. Dina Amr

Academic Year 2024/2025

## **Abstract**

ELEVATE introduces a new and innovative way of shopping online, especially for clothing, as it gives the user the ability to virtually try on clothes at home without the need to go to offline stores. Gathering Egyptian local brands in one place is the main idea of the project, where users can shop for clothing from multiple local brands all in one place.

The project aims to refresh the market of online shopping for clothing and to help more Egyptian local brands gain broader visibility and extended reach. Introducing a digital solution to the market of clothing is the most competitive edge in the project, as most clothing brands rely on offline stores, limiting their exposure.

ELEVATE addresses the need for easy online shopping experience, while current online shopping websites mostly have the usual e-commerce functionalities and do not offer extra features for the customer. The project provides an easy way for the customer to imagine how a piece of clothing would look on him/her without trying it on themselves.

Introducing a centralized platform for shopping from multiple local brands in Egypt with a focus on user experience and innovation is one of the main attempts to help fill the gap in the market. Integrating artificial intelligent features like personalized recommendations and augmented reality try on are also attempts to digitize the shopping experience and make it more engaging.

ELEVATE has the potential to become one of the most significant platforms in the clothing industry, as it offers a different and slightly new experience to the customer through integrating technology with real-life applications like shopping for clothing.

# Table of Contents

<b>Abstract.....</b>	i
<b>List of Figures .....</b>	iii
<b>List of Tables .....</b>	iv
<b>1 Chapter 1 – Introduction.....</b>	1
1.1    Background and Motivation.....	1
1.2    Problem Statement .....	1
1.3    Objectives .....	2
1.4    Project Scope and Limitations.....	2
1.5    Project Methodology .....	3
1.6    Project Report Outline.....	4
<b>2 Chapter 2 – Market and Literature Survey .....</b>	5
2.1    Market Analysis.....	5
2.2    Competitor Analysis .....	6
2.3    User Personas .....	7
<b>3 Chapter 3 – ELEVATE Analysis .....</b>	11
3.1    Functional Requirements .....	11
3.2    Non-Functional Requirements .....	14
3.3    System Architecture .....	15
3.4    Class Diagram .....	16
3.5    Database Structure (ERD) .....	17
3.6    Sequence Diagrams .....	18
3.7    Use Case Diagrams.....	23
3.8    Use Case Descriptions .....	25
<b>4 Chapter 4 – ELEVATE Design and Implementation .....</b>	55
4.1    Tools and Technologies Used.....	55
4.1.1    Backend Development.....	55
4.1.2    Data Storage and Management.....	55
4.1.3    Search and Recommendation Services.....	56
4.1.4    Serverless Data Platform and AI Integration.....	56
4.1.5    Deployment and Hosting.....	56
4.1.6    Frontend Development .....	57
4.1.7    Development Environment and Collaboration Tools .....	58

4.2 Application Screens (UI) .....	59
4.2.1 Customer Scenario .....	59
4.2.2 Brand Owner Scenario.....	64
<b>5 Chapter 5 – ELEVATE Testing and Evaluation.....</b>	<b>68</b>
5.1 Test Cases.....	68
5.2 Testing Results .....	78
<b>6 Chapter 6 – Conclusions and Future Work.....</b>	<b>81</b>
6.1 Conclusions .....	81
6.2 Lessons Learned .....	82
6.3 Future Work .....	83
<b>7 References.....</b>	<b>85</b>

## List of Figures

Figure 1.1. Star Life Cycle Model .....	3
Figure 1.2. Project Timeline .....	4
Figure 3.1. Architecture Diagram .....	15
Figure 3.2. Class Diagram.....	16
Figure 3.3. Database Structure (ERD) .....	17
Figure 3.4. Sign Up Sequence Diagram.....	18
Figure 3.5. Make Order Sequence Diagram.....	19
Figure 3.6. Refund Order Sequence Diagram.....	20
Figure 3.7. Rate Product Sequence Diagram.....	21
Figure 3.8. Add Product Sequence Diagram.....	21
Figure 3.9. Try On Sequence Diagram.....	22
Figure 3.10. Brand Owner Use Case Diagram (a) .....	23
Figure 3.11. Visitor Customer Use Case Diagram (b) .....	23
Figure 3.12. Logged Customer Use Case Diagram (c) .....	24

## List of Tables

Table 1.1. Work Breakdown Structure.....	2
Table 2.1. Related Works .....	6
Table 2.2. Customer Persona (Primary).....	7
Table 2.3. Customer Persona (Secondary).....	8
Table 2.4. Occasional Shopper Persona.....	9
Table 2.5. Brand Owner Persona.....	10
Table 3.1. Signup Scenario .....	25
Table 3.2. Login Scenario .....	26
Table 3.3. Reset Password Scenario .....	27
Table 3.4. Kids Section Scenario .....	28
Table 3.5. Gender-Based Sections Scenario .....	29
Table 3.6. List by Brands Use Case .....	30
Table 3.7. Search by Image Scenario .....	32
Table 3.8. Search by Brand Name Scenario .....	33
Table 3.9. Search Filters Scenario .....	34
Table 3.10. Manage Ratings Scenario .....	35
Table 3.11. Manage Orders Scenario .....	36
Table 3.12. Edit Profile Scenario.....	37
Table 3.13. AI Virtual Try-On Scenario.....	38
Table 3.14. AI Recommendations Scenario .....	39
Table 3.15. Add to Cart Scenario .....	40
Table 3.16. View Cart Summary Scenario .....	41
Table 3.17. Change Product Quantity Scenario .....	42
Table 3.18. Delete Products Scenario.....	43
Table 3.19. Manage Addresses Scenario .....	44
Table 3.20. Choose Shipping Method Scenario .....	46
Table 3.21. View Order Summary Scenario .....	47
Table 3.22. Access Products Dashboard Scenario .....	48
Table 3.23. View Brand Profile Scenario .....	48
Table 3.24. Edit Brand Profile Scenario .....	49
Table 3.25. Add New Product Scenario .....	51
Table 3.26. Edit Product Scenario.....	53
Table 3.27. Delete Product Scenario .....	54
Table 5.1. Customer Signup Test Case .....	68

Table 5.2. Third-Party Signup Test Case .....	68
Table 5.3. Customer Login Test Case.....	68
Table 5.4. Brand Owner Login .....	69
Table 5.5. Brand Manager Login Test Case.....	69
Table 5.6. Reset Password Test Case .....	69
Table 5.7. AI Virtual Try-On Test Case .....	70
Table 5.8. Add To Wishlist Test Case .....	70
Table 5.9. Remove From Wishlist Test Case .....	70
Table 5.10. Add to Cart Test Case .....	70
Table 5.11. Remove From Cart Test Case.....	71
Table 5.12. Create New Review Test Case .....	71
Table 5.13. Update Review Test Case.....	71
Table 5.14. Delete Review Test Case .....	71
Table 5.15. View Brand Orders Test Case.....	72
Table 5.16. View Customer Orders Test Case.....	72
Table 5.17. Create Draft Order Test Case .....	72
Table 5.18. Customer Order Details Test Case .....	72
Table 5.19. Discard Draft Order Test Case .....	73
Table 5.20. Place Order Test Case .....	73
Table 5.21. Cancel Order Test Case .....	73
Table 5.22. Refund Item Test Case .....	73
Table 5.23. Calculate Shipment Fees Test Case .....	74
Table 5.24. View Brand Data Test Case .....	74
Table 5.25. Update Brand Profile Test Case .....	74
Table 5.26. View Brand Owner Profile Test Case .....	75
Table 5.27. Update Brand Profile Test Case .....	75
Table 5.28. View Products List Test Case.....	75
Table 5.29. Add Product Test Case .....	75
Table 5.30. Edit Product Test Case.....	76
Table 5.31. View Product Test Case .....	76
Table 5.32. Delete Product Test Case .....	76
Table 5.33. View Processing Orders Test Case.....	76
Table 5.34. View Refunded Products Test Case.....	77
Table 5.35. View Brand Reviews Summary Test Case .....	77
Table 5.36. View Sales Statistics Test Case .....	77

# Chapter 1 – Introduction

This chapter gives an overview of the project, **ELEVATE**, which aims to support and promote Egyptian local brands by addressing the problem, objectives, scope, limitations, and the used methodology in this project.

## 1.1 Background and Motivation

The Egyptian market has a growing demand for local brands, driven by consumers who prioritize supporting local businesses and value "**Made in Egypt**" products. However, most local brands rely heavily on social media and physical stores, limiting their reach and visibility. This project aims to create a centralized e-commerce platform that brings together Egyptian brands under one digital umbrella, enabling these businesses to access a wider audience. This platform, targeted primarily at young Egyptian consumers, is motivated by the need to support local economic growth and promote national brands in a competitive and evolving market.

## 1.2 Problem Statement

Local Egyptian brands face a significant barrier to expanding their reach and visibility due to their reliance on social media and limited online presence. The lack of a unified digital platform for local brands creates a gap between consumer demand for accessible local products and the existing, scattered online presence of these businesses. Without a centralized platform, local brands struggle to attract a broader audience and miss the opportunity to benefit from collective marketing, exposure, and technological tools that enhance the user experience. Addressing this gap is critical for promoting local brands and supporting the Egyptian economy.

## 1.3 Objectives

Our main objective of this project is to create an e-commerce platform that will:

- Centralize Egyptian local brands, especially clothing brands, providing them with an extended reach.
- Offer features like brand directory, product listings, reviews, recommendations, and AI try-on options to improve customer experience.
- Develop a scalable platform that could expand globally, allowing Egyptian brands to gain international visibility.

## 1.4 Project Scope and Limitations

The scope of this project is to build a fully functional online e-commerce platform that showcases Egyptian local brands, primarily in the fashion and clothing sectors. The platform will initially target Egyptian consumers but is designed with scalability in mind to potentially reach international markets. Limitations include quality control for brand participation and the challenge of onboarding a wide range of local brands.

The initial team structure includes:

Table 1.1. Work Breakdown Structure

Role/Task	Adham	Belal	Seif	Shawky	Nour T.	Nour W.
Project Manager			✓	✓		
System Analysis	✓	✓	✓	✓	✓	✓
System Design	✓	✓	✓	✓	✓	✓
Backend Dev.				✓		
Frontend Dev.	✓	✓	✓		✓	✓
Graphic Design				✓	✓	
Documentation		✓	✓			

## 1.5 Project Methodology

The project will adopt the Star Life Cycle Model, which is a flexible and iterative approach that emphasizes adapting to changing requirements throughout the software development process. It allows for continuous refinement, accommodating evolving user needs and technologies. This makes it highly suitable for projects that require frequent updates and changes, ensuring responsiveness and flexibility in dynamic environments.

The Star Life Cycle Model is highly suitable for our project as it focuses on the consumer being at the center of all development stages as shown in *Figure 1.1*. This customer-centric approach ensures that the product continuously aligns with user needs and expectations. Also, it allows us to jump between different development stages such as analysis, design, and implementation and move back and forth ensuring the product remains relevant and responsive to changing demands.

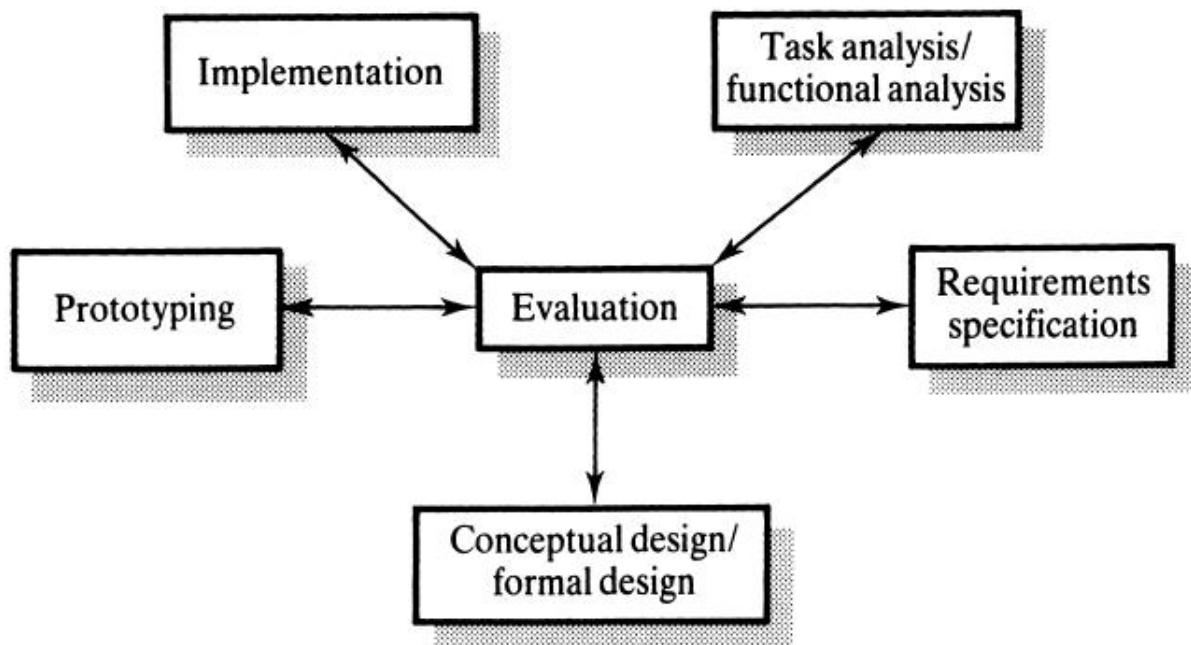


Figure 1.1. Star Life Cycle Model

The timeline for the project progress shows different development stages:

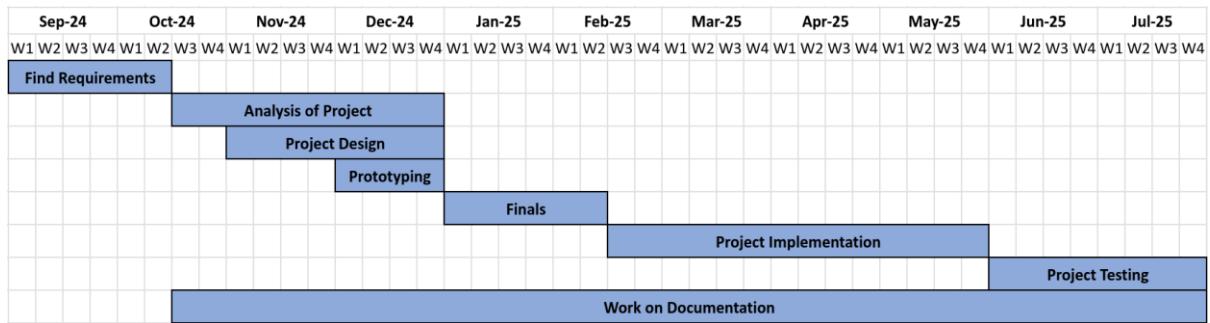


Figure 1.2. Project Timeline

## 1.6 Project Report Outline

- **Chapter 2** presents the review of the Egyptian market, competitors, existing studies and research, trends, and innovations in promoting local brands online.
- **Chapter 3** presents system analysis, functional and non-functional requirements, use case scenarios, and supporting UML diagrams.
- **Chapter 4** presents the design and development process, including the platform architecture, user interface, prototype screens, used tools, programming languages, and frameworks along with selection reasons.
- **Chapter 5** presents the testing, evaluation, and implementation results.
- **Chapter 6** concludes the work with key lessons learned, challenges faced, and suggestions for future work to expand or enhance the platform's capabilities.

## Chapter 2 – Market and Literature Survey

This chapter offers an overview analysis of the current market landscape for local Egyptian brands, examines competitor platforms. By analyzing existing platforms and market demand, this chapter identifies critical components for the successful implementation of the **ELEVATE** project.

### 2.1 Market Analysis

The market for local Egyptian brands has demonstrated significant potential, particularly due to the increasing consumer support for "**Made in Egypt**" initiatives. However, many local brands are limited by their reliance on social media and physical stores, which constrains their growth. Research indicates a substantial demand for a centralized platform where Egyptian brands, particularly small to medium-sized enterprises (SMEs), can showcase their products and expand their customer base.

#### – Stakeholders

Our main stakeholders are the Egyptian shoppers who are looking for a centralized platform to discover authentic local brands and are actively searching and supporting local businesses and boycott initiatives. SMEs in Egypt looking to expand their online presence and established Egyptian brands aiming to connect with a wider digital audience are also part of our stakeholders.

#### – Market Gaps

Current local competitors are lacking several key features that are popular on international platforms, such as:

**AI-Powered Recommendations:** The absence of personalized product suggestions based on user behavior.

**AI Try-Before-You-Buy:** The lack of virtual try-on capabilities for clothing, a feature which enhances the shopping experience and user engagement introducing a virtual fitting room experience.

## 2.2 Competitor Analysis

According to our recent analysis of existing studies on competitor e-commerce projects, we concluded that there are two main types of competitors:

### **Local Stores and Online Pages (Direct Competitors):**

Existing local stores such as Domanza, Go Native, Labelled, and Locally primarily depend on social media or physical stores for their presence, instead of having dedicated platforms. This limits user engagement and the availability of advanced features and limits their reach to different customers.

### **Larger E-commerce Sites (Indirect Competitors):**

While larger e-commerce sites like Jumia and Noon occasionally feature local clothing brands, they lack dedicated support or features tailored to the unique needs of smaller, emerging Egyptian businesses.

Table 2.1. Related Works

Feature / Platform	Domanza	Go Native	Labelled	Locally	ELEVATE
Offline Store	✓	✓	✓	✓	✗
Online Store	✓	✓	✗	✗	✓
Mobile App	✗	✗	✗	✗	✓
Subscription Plans	✗	✗	✗	✗	✓
Shop By Brand	✓	✓	✗	✗	✓
Visual Search	✗	✗	✗	✗	✓
AI Virtual Try-On	✗	✗	✗	✗	✓
AI-Powered Recommendations	✗	✗	✗	✗	✓

In summary, the Egyptian market is underserved by current platforms, providing a clear opportunity to support local brands through a dedicated e-commerce solution that emphasizes a seamless shopping experience and advanced digital features.

## 2.3 User Personas

Table 2.2. Customer Persona (Primary)

Name	Age	Location	Occupation	Marital status	Annual Income
Ahmed	18	Giza	Student	Single	N/A
<b>Quote</b>	I want to look stylish and unique, but I also want to support Egyptian designers. It's hard to find a platform where I can get all my favorite local brands in one place.				
<b>Interests</b>	Gaming, football, reading, psychology.				
<b>Goals</b>	<ul style="list-style-type: none"> <li>Stay fashionable and trendy by discovering unique clothing and accessories.</li> <li>Try on clothes before making a purchase by using an artificial intelligence (AI) app to let him try it virtually effortlessly.</li> <li>Easily find and purchase clothing from different local Egyptian brands.</li> <li>Support Egyptian designers and promote local fashion by sharing his purchases on social media</li> </ul>				
<b>Frustration</b>	<ul style="list-style-type: none"> <li>Difficulty finding all his favorite local brands in one place.</li> <li>Difficulty choosing clothes that would look good on him.</li> </ul>				
<b>Shopping Behavior</b>	Convenience-Driven shopper who prioritizes ease and speed of the shopping process.				
<b>Preferred Features</b>	<ul style="list-style-type: none"> <li>A well-curated homepage showing trendy Egyptian brands.</li> <li>A wish list or “favorites” section to save items for later.</li> <li>Quick access to a detailed size chart and reviews from other customers.</li> <li>The ability to pay using mobile payment services or upon delivery.</li> </ul>				
<b>Technology Use</b>	<ul style="list-style-type: none"> <li>Uses Instagram, Facebook, and TikTok to follow local brands.</li> <li>Actively uses e-commerce platforms like Amazon and Jumia online stores.</li> </ul>				
<b>Personality</b>	<ul style="list-style-type: none"> <li>Style: Fashion-forward, enjoys mixing modern and traditional elements.</li> <li>Values: Creativity, quality, and supporting the local community.</li> </ul>				

Table 2.3. Customer Persona (Secondary)

Name	Age	Location	Occupation	Marital status	Annual Income
Manal	45	Cairo	Teacher	Married	\$150,000
<b>Quote</b>		I want to find clothing for myself and my children without having to jump from one website to another. If I can get good quality and support local brands in the process, that's a big plus.			
<b>Interests</b>		Crochet, baking, watching the news, shopping.			
<b>Goals</b>		<ul style="list-style-type: none"> <li>Find quality clothing that suits the needs of both her and her children.</li> <li>Explore affordable fashion options from local brands.</li> <li>Support local businesses and follow boycott initiatives but prefers a straightforward shopping experience.</li> <li>Buy all family-related clothing in one place to avoid juggling multiple orders from different websites.</li> </ul>			
<b>Frustration</b>		<ul style="list-style-type: none"> <li>Difficulty in finding local platforms that provide a variety of clothing options for both adults and children.</li> </ul>			
<b>Shopping Behavior</b>		Convenience-Driven shopper who prioritizes ease and speed of the shopping process.			
<b>Preferred Features</b>		<ul style="list-style-type: none"> <li>Shops mostly during weekends when she has more free time.</li> <li>Looks for discounts and seasonal offers, as she often shops in bulk for the whole family.</li> <li>Reads reviews carefully before making a purchase, especially when shopping for her children.</li> <li>A well-curated homepage showing trendy Egyptian brands.</li> </ul>			
<b>Technology Use</b>		<ul style="list-style-type: none"> <li>Regularly browse through Facebook and Pinterest for clothing ideas.</li> <li>Uses platforms like Amazon Egypt and Jumia for variety and reliability.</li> </ul>			
<b>Personality</b>		<ul style="list-style-type: none"> <li>Style: Conservative yet trendy; prioritizes comfort and modesty.</li> <li>Values: Family, quality, and reliability.</li> </ul>			

Table 2.4. Occasional Shopper Persona

Name	Age	Location	Occupation	Marital status	Annual Income
Emily	31	USA	Blogger	Married	\$60,000
<b>Quote</b>	I want something that reminds me of Egypt — a piece of clothing that feels authentic, tells a story, and that I can gift to my friends.				
<b>Interests</b>	Photography, traveling, fashion, languages.				
<b>Goals</b>	<ul style="list-style-type: none"> <li>Buy authentic Egyptian-made clothing that reflects the culture and craftsmanship she experienced during her visit.</li> <li>Purchase unique pieces to gift her family and friends, showcasing the beauty of Egyptian fashion.</li> <li>Enjoy a seamless shopping experience that offers English language support and clear product descriptions.</li> </ul>				
<b>Frustration</b>	Concerns about receiving products that differ from what was advertised or of poor quality.				
<b>Shopping Behavior</b>	<ul style="list-style-type: none"> <li>Prefers to shop for clothing while reminiscing about her travel experiences and finding items that hold sentimental value.</li> <li>Looks for products that represent Egypt's culture and style, such as kaftans, scarves, or handwoven garments.</li> <li>Reviews product ratings, photos, and descriptions thoroughly before making a purchase.</li> </ul>				
<b>Preferred Features</b>	Clear Size Charts and Descriptions: Detailed descriptions, size charts, and care instructions in English to help her understand the product's fit and material.				
<b>Technology Use</b>	<ul style="list-style-type: none"> <li>Frequently shops online for travel-related products, souvenirs, and unique fashion items.</li> <li>Actively follows brands on Instagram, Pinterest, and Facebook to stay updated on local trends and unique finds from different countries.</li> <li>Rely on PayPal and credit card payments and expect a secure checkout process.</li> </ul>				
<b>Personality</b>	<ul style="list-style-type: none"> <li>Style: Adventurous, loves collecting unique and meaningful items from her travels.</li> <li>Values: Authenticity, cultural diversity, and supporting local communities.</li> </ul>				

Table 2.5. Brand Owner Persona

Name	Age	Location	Occupation
Tamara	24	Cairo	Co-Founder and Owner of "Not Found"
<b>Quote</b>	I need a platform that gives my brand more visibility and connects me with a broader scope of customers.		
<b>Background</b>	<ul style="list-style-type: none"> <li>• Tamara Sherif co-founded Not Found, a minimalist streetwear brand based in Cairo, Egypt, alongside Dina Tarek.</li> <li>• The brand focuses on creating essential, high-quality pieces that combine comfort and simplicity.</li> <li>• Sherif and her co-founder aim to make stylish and affordable basics that appeal to everyday wearers, emphasizing both practicality and minimalism.</li> <li>• The name "Not Found" was inspired by the difficulty they faced in sourcing high-quality essentials at reasonable prices. She is looking to expand the brand's online presence and reach a broader customer base.</li> </ul>		
<b>Goals</b>	<ul style="list-style-type: none"> <li>• Expand her brand's online presence.</li> <li>• Use analytics to understand customer behavior and optimize her product offerings.</li> <li>• Manage inventory more effectively based on demand.</li> <li>• Ensure his brand stands out with high visibility and excellent customer feedback on the platform.</li> </ul>		
<b>Frustration</b>	As the brand grows, logistical challenges can become overwhelming if the infrastructure isn't set up for rapid scaling.		
<b>Technology Use</b>	<ul style="list-style-type: none"> <li>• Uses Facebook and Instagram Ads to drive traffic to her store.</li> <li>• Operates her website using basic e-commerce tools.</li> </ul>		
<b>Personality</b>	<ul style="list-style-type: none"> <li>• Style: bold, creative, and passionate about promoting Egyptian craftsmanship.</li> <li>• Values: Quality, customer satisfaction, and staying connected to her cultural roots.</li> </ul>		

## Chapter 3 – ELEVATE Analysis

### 3.1 Functional Requirements

ID	Requirement Description
<b>User Registration and Login</b>	
#01	The system must allow users to register using third-party services (Google)
#02	Sign up using a valid email address, Phone number, Address (optional), and a strong password. <b>Strong password:</b> Should be at least 8 characters including uppercase, lowercase, numbers, and symbols.
#03	Login to access personalized features.
#04	<b>Forget Password:</b> The system must provide an option for users to recover their passwords via email or SMS.
#05	Manage user profiles, including updating their personal information, and shipping addresses.
#06	The system must allow any user to browse without an account, but they will need to sign up or log in to complete purchases.
#07	Brand owners can be added by the system admin through the admin's dashboard.
<b>Search and Filtering</b>	
#08	<b>Search by Brand Name:</b> Customers can search for products by entering the name of a specific brand
#09	<b>Search by Color:</b> Customers can filter products based on color preferences.
#10	<b>Search by Size:</b> Customers can filter products by size (XX Small / X Small / Small / Medium / Large / X Large).
#11	<b>Filtering by Product Type:</b> Customers can filter items by category, e.g., shirts, skirts, or any other clothing type.
#12	<b>Filtering by Price:</b> Customers can filter items by price, from low to high – high to low – specific price range.
#13	<b>Display Catalogs by Brand:</b> Each brand should have a catalog showcasing all available products.

ID	Requirement Description
#14	<b>Filter by Product Type within Catalog:</b> Customers should be able to filter within a brand's catalog to display specific product types (e.g., shirts, skirts, pants).
<b>Search by Image</b>	
#15	Upload images to search for similar products on the platform. The system should use an image recognition system to identify similar items across brands and categories.
<b>AI-based Recommendations</b>	
#16	The system should provide AI-based personalized product recommendations to users based on other customers' interactions and their browsing history.
#17	The homepage should display dynamically updated recommendations like "Most Liked," and "Picked Together" sections.
#18	Each product page must have a "Related Products" section showing items similar to the currently viewed product.
#19	The system offers a "Kids Section" where users can find recommendations.
<b>Order Life Cycle</b>	
#20	Add a product with a specified quantity and variant (if applicable) to the cart when clicking on "Add to Cart".
#21	View cart summary, showing product names, quantities, price per product, and subtotal.
#22	Change product quantities in the cart or remove products from the cart.
#23	Add new shipping addresses or select from previously saved addresses. Input fields should include full name, address, and phone number.
#24	The system displays one shipping method (e.g., standard) and calculates the cost based on the user's location and cart contents.
#25	View final order summary with itemized costs (product prices, shipping fees, discounts) before confirming the order.
#26	Cancel orders before shipping; a cancellation confirmation should be sent via email.

ID	Requirement Description
#27	Initiate a return/exchange request within 14 days of delivery. The system tracks the return status and confirms receipt.
<b>Ratings and Reviews</b>	
#28	<b>User Ratings:</b> Users can rate products out of 5 stars after they receive them
#29	<b>User Reviews:</b> Users can write reviews about products they have purchased before.
#30	<b>Brand Rating:</b> Brands will have a rating that's equal to the average rating of all their products. <ul style="list-style-type: none"> <li>The rating will be weighted ex: prod1(2 users rated 5), prod2(1 user rated 4) → Overall Brand Rating (<math>\frac{2 * 5 + 1 * 4}{3}</math>).</li> </ul>
<b>Brand Owner Interface</b>	
#31	Each brand owner has a separate profile page showcasing their identity and contact information
#32	Brand profile can contain the story behind the brand, the brand logo, and more
#33	Each brand owner has a dedicated page that gives him all his control over the brand products: add, edit, delete
#34	Each brand owner can add a new product: Name, Category, Color, Price, Size, Image, Quantity, Material, Department (Men, Women, Unisex).
#35	Each brand owner has a dashboard that shows insights into the brand's products: <ul style="list-style-type: none"> <li>Top products sold</li> <li>Products sold per month</li> <li>Customer reviews of brand products</li> <li>Total products sold monthly</li> <li>Total sales per month</li> </ul>
#36	Subscription plans are available to enable more features for the brand owner such as better search results exposure, more quota of products.
<b>AI Try On</b>	
#37	Click on the Try On button on products, ask for camera/storage permission so that users can upload images of themselves and be able to see the product on them virtually.

## 3.2 Non-Functional Requirements

### 1. Performance

- The system must handle up to 100 simultaneous user sessions without significant performance degradation.
- The system should load pages in less than 3 seconds under normal conditions.

### 2. Security

- All user passwords must be hashed and stored securely.
- User sessions must expire after a specified period (e.g., 60 minutes).

### 3. Usability

- The system must be accessible on both desktop and mobile browsers.
- The system interface should be intuitive and easy to navigate for both users and brand managers.
- User feedback should be considered to enhance usability over time.

### 4. Reliability

- The system must have an uptime of 99%.
- Data backups must be performed daily to prevent loss of orders and user data.

### 5. Scalability

- The system should be designed to scale as the number of users and brands grows.
- Database optimization techniques (e.g., indexing) should be used to handle increasing data volume.

### 6. Maintainability

- The codebase must be modular and well-documented to allow future enhancements.
- The system must be easy to update without disrupting user experience.

### 3.3 System Architecture

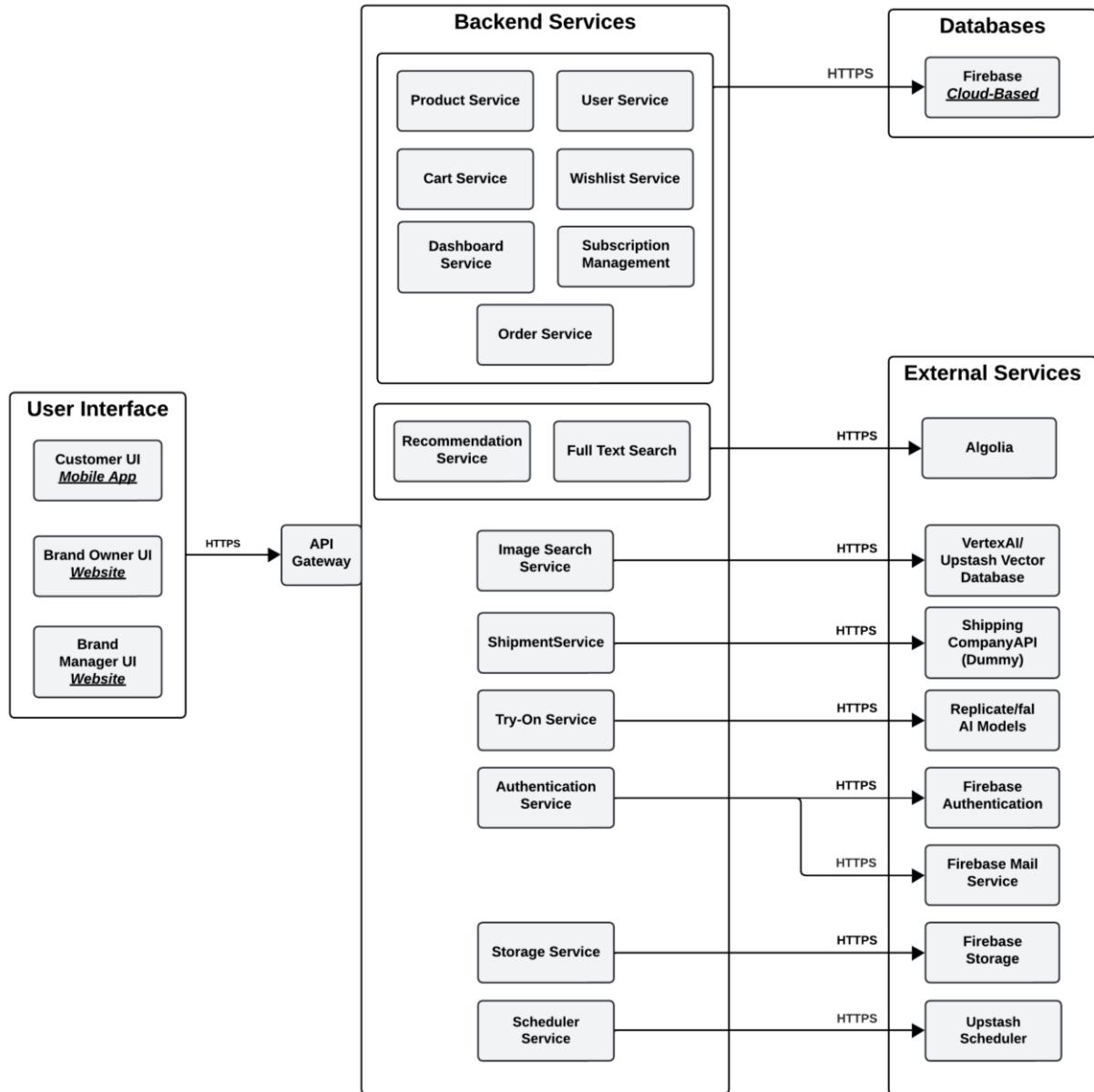


Figure 3.1. Architecture Diagram

The diagram provides an abstract view of the entire application by showing major components and their interactions. It does not delve into specific implementation details, making it suitable for anyone to understand the system architecture.

### 3.4 Class Diagram

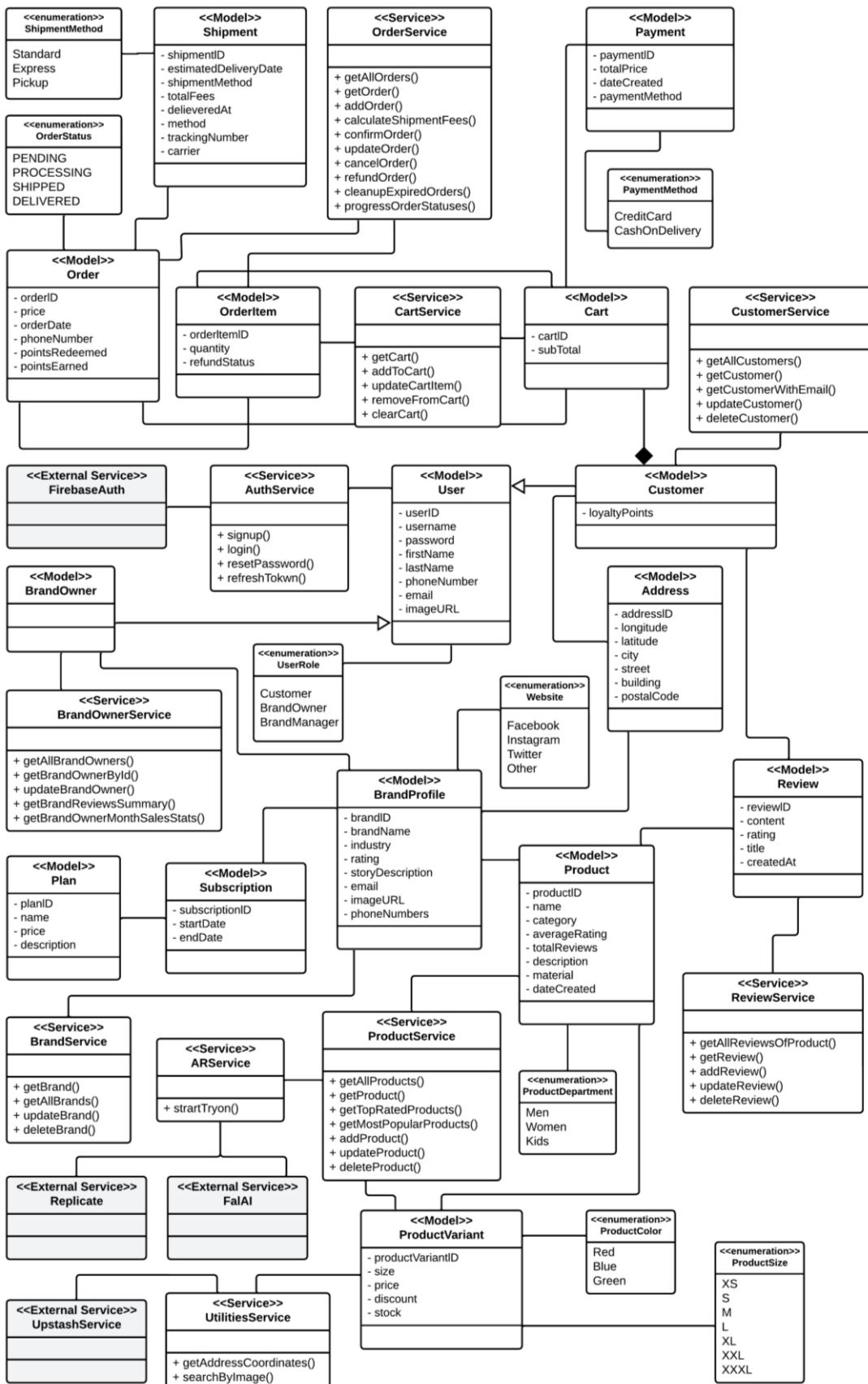


Figure 3.2. Class Diagram

### 3.5 Database Structure (ERD)

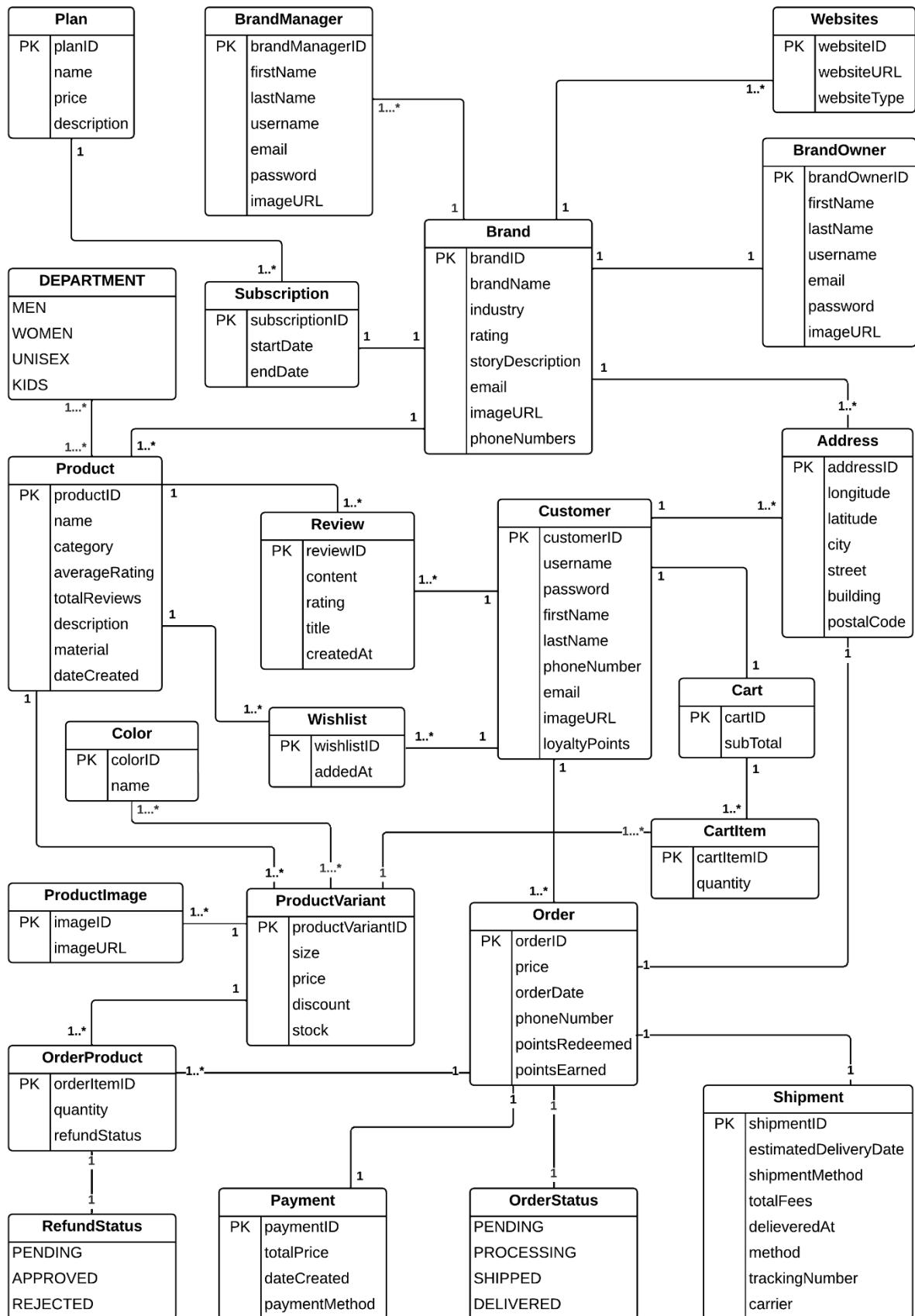


Figure 3.3. Database Structure (ERD)

### 3.6 Sequence Diagrams

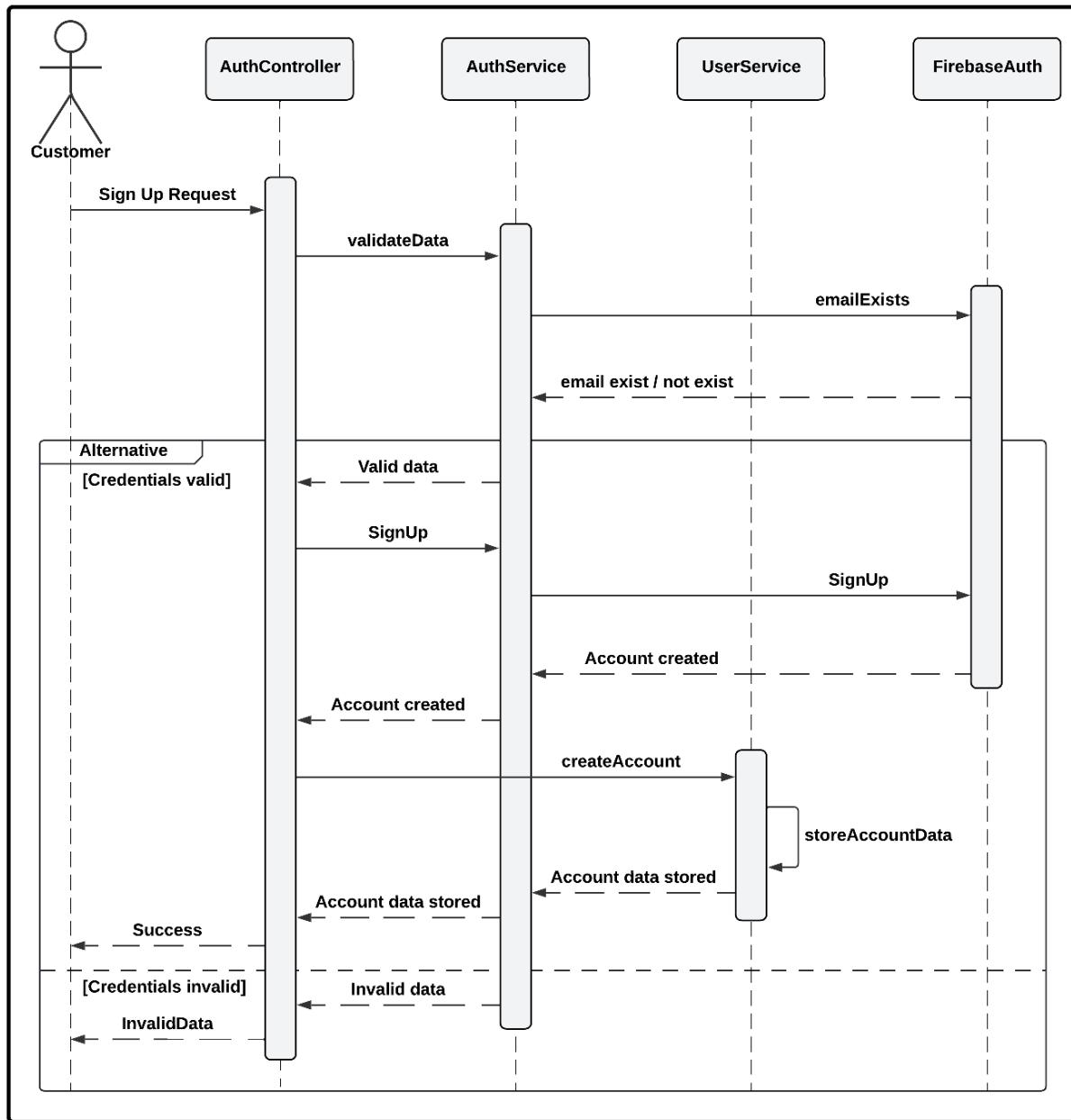


Figure 3.4. Sign Up Sequence Diagram

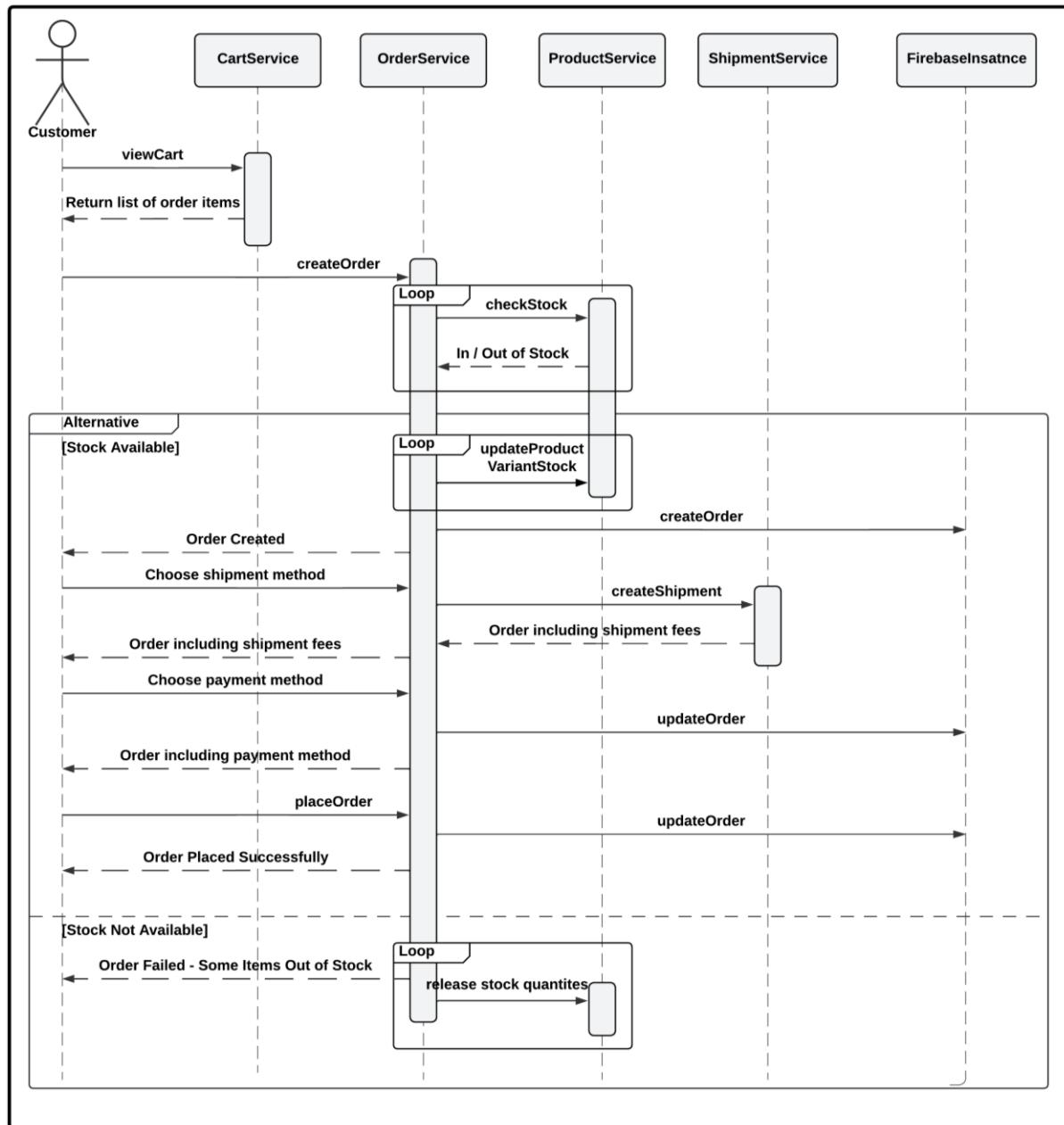


Figure 3.5. Make Order Sequence Diagram

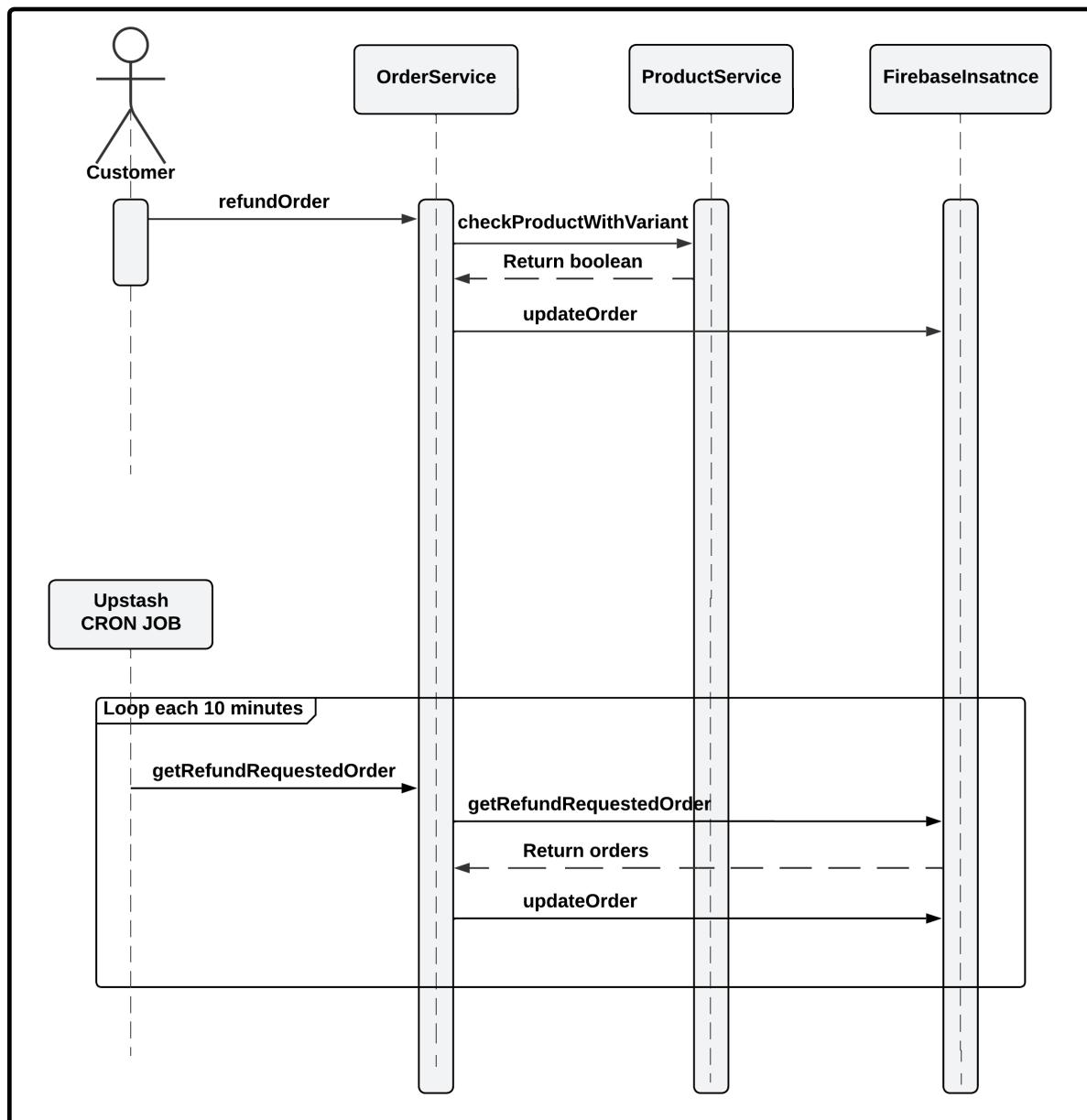


Figure 3.6. Refund Order Sequence Diagram

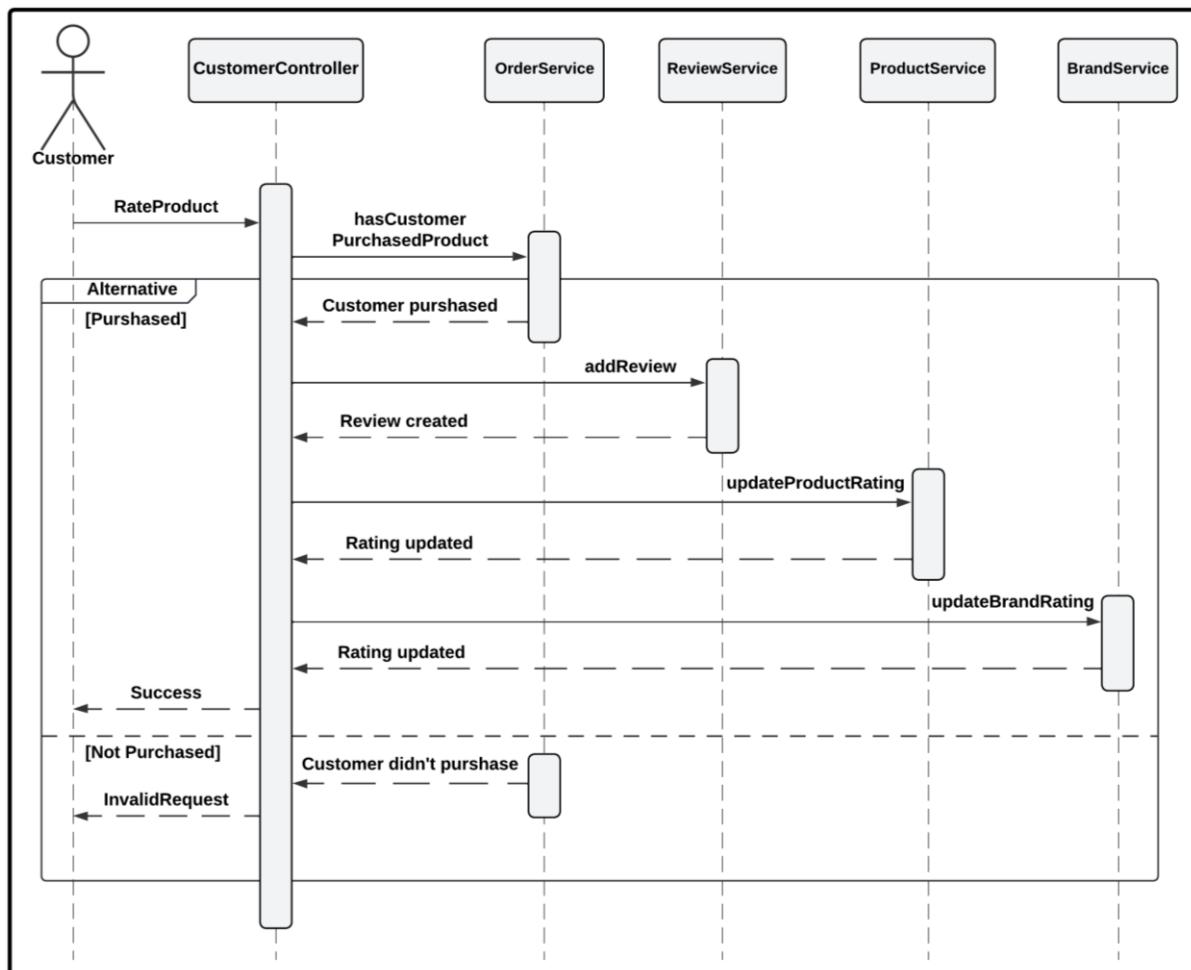


Figure 3.7. Rate Product Sequence Diagram

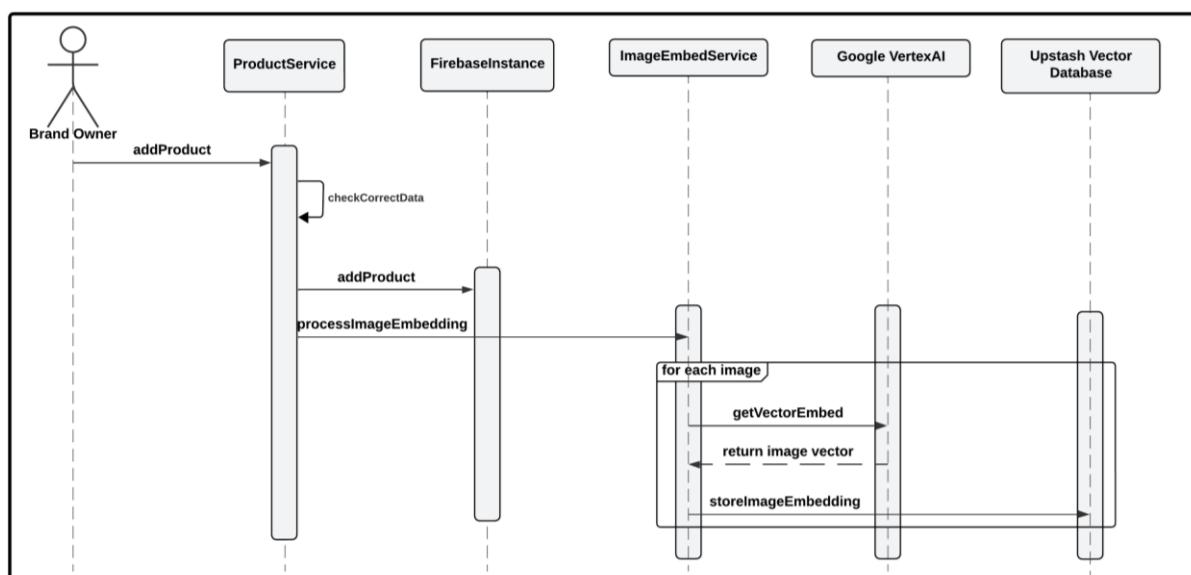


Figure 3.8. Add Product Sequence Diagram

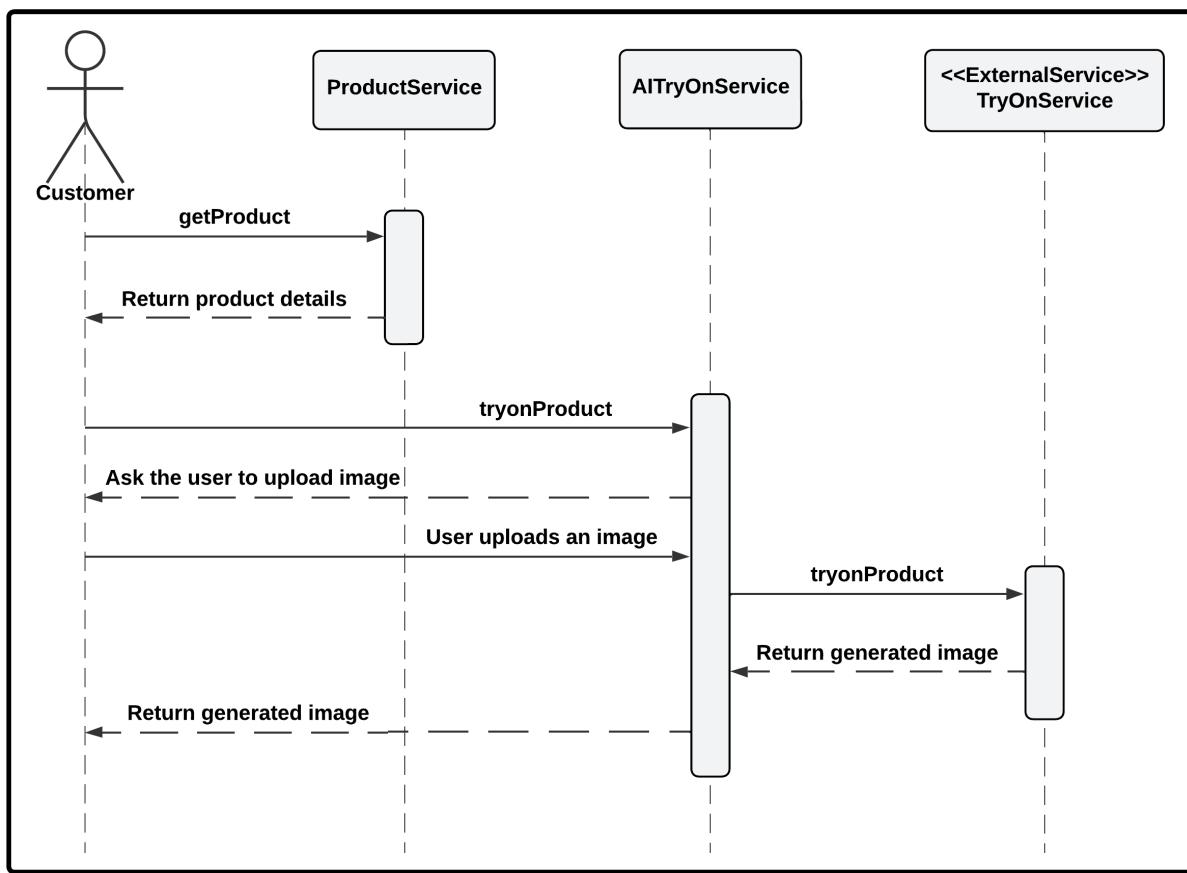


Figure 3.9. Try On Sequence Diagram

### 3.7 Use Case Diagrams

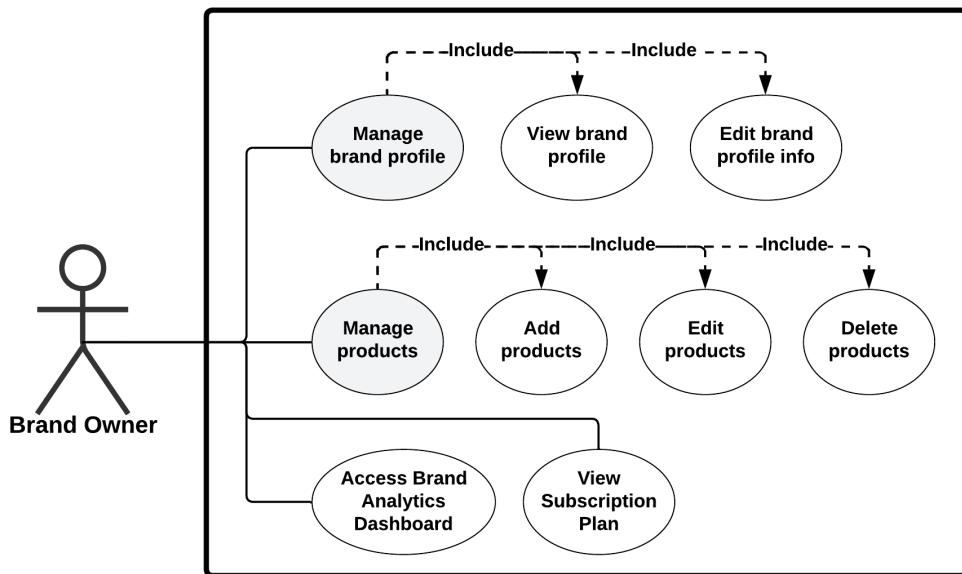


Figure 3.10. Brand Owner Use Case Diagram (a)

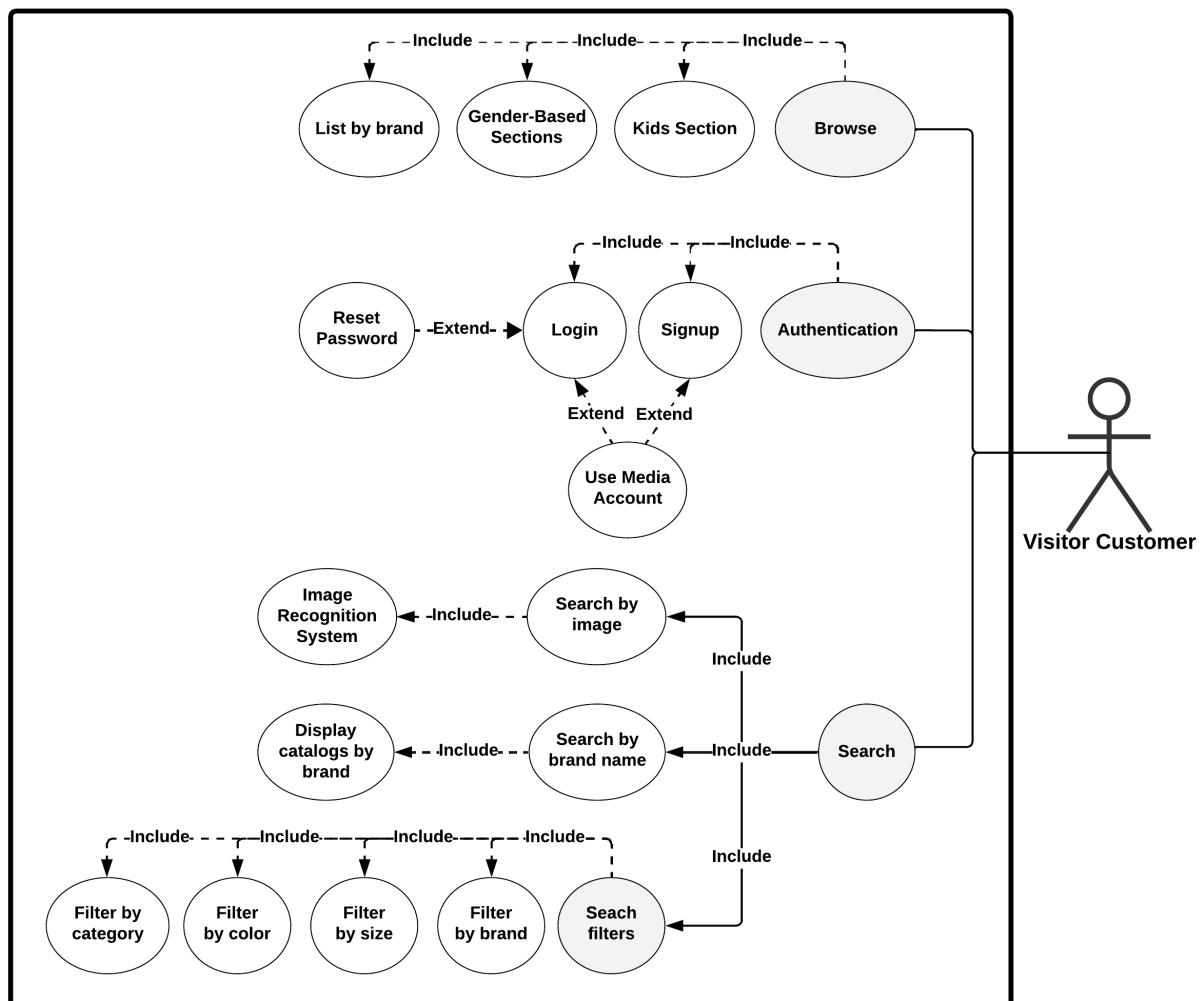


Figure 3.11. Visitor Customer Use Case Diagram (b)

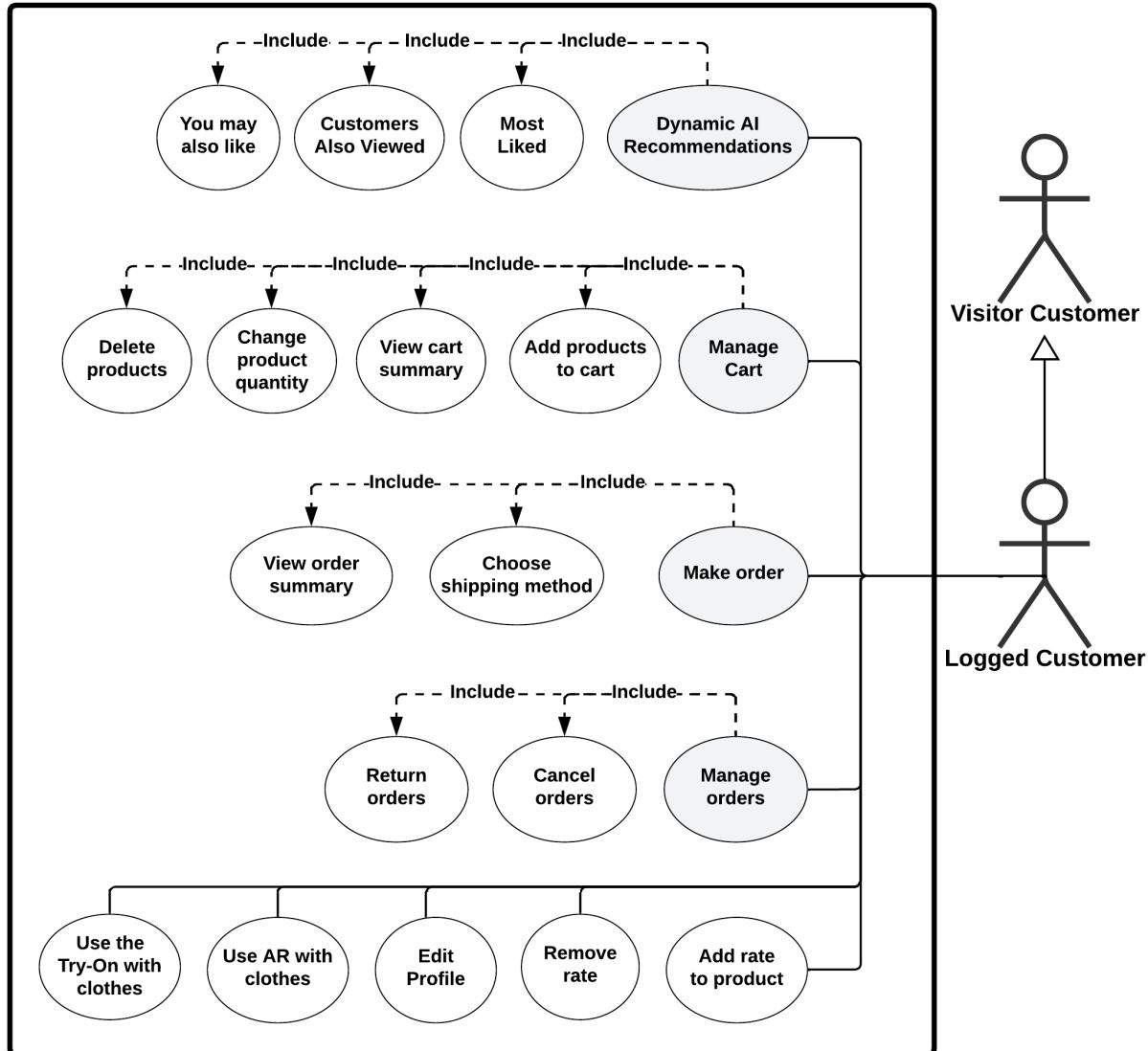


Figure 3.12. Logged Customer Use Case Diagram (c)

### 3.8 Use Case Descriptions

Table 3.1. Signup Scenario

<b>Use Case ID</b>	UC#1	
<b>Use Case Name</b>	Signup	
<b>Actors</b>	Customer	
<b>Primary Actor</b>	Customer	
<b>Description</b>	New users can create an account on the platform by signing up. This involves providing personal details or using their media Accounts.	
<b>Pre-Condition(s)</b>	The user does not have an account and wishes to create one.	
<b>Post-Condition(s)</b>	The user successfully created an account and is ready to log in.	
<b>Flow of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1- User clicks on "Sign Up" and selects the option to register using email or media Accounts (Facebook or Google).	
		2- System displays the signup form with fields for name, email, password, and other necessary details.
	3- User fills in the form and submits it.	
		4- System validates the information and creates the account.
		5- System displays a success message and redirects the user to the login page or logs the user in automatically.
<b>Exceptions</b>	<b>Actor Action</b>	<b>System Response</b>
		4- System recognizes that the email or phone number is already in use, then displays: "Email or phone number is already registered. Please log in or reset your password."

Table 3.2. Login Scenario

<b>Use Case ID</b>	UC#2	
<b>Use Case Name</b>	Login	
<b>Actors</b>	Customer	
<b>Primary Actor</b>	Customer	
<b>Description</b>	Registered users can log into their accounts to access personalized features.	
<b>Pre-Condition(s)</b>	The user already has an account and wants to access it.	
<b>Post-Condition(s)</b>	The user is logged in and can access their account features.	
<b>Flow of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1- User clicks on "Login."	
		2- System displays the login page.
		3- System shows boxes to input the username and password.
	4- User fills in the boxes with their credentials.	
		5- System verifies the user information.
		6- If verification is successful, the system grants the user access to the app's logged in customer functionalities.
<b>Exceptions</b>	<b>Actor Action</b>	<b>System Response</b>
		5- System doesn't recognize the user information.
		6- System shows: "Invalid username or password."

Table 3.3. Reset Password Scenario

<b>Use Case ID</b>	UC#3	
<b>Use Case Name</b>	Reset Password	
<b>Actors</b>	Customer	
<b>Primary Actor</b>	Customer	
<b>Description</b>	Users who have forgotten their password can reset it using this feature. It typically involves entering an email address and receiving instructions to set a new password.	
<b>Pre-Condition(s)</b>	The user already has an account and wants to access it but cannot remember their password and wants to reset it.	
<b>Post-Condition(s)</b>	The user successfully resets their password and can log in.	
<b>Flow of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1- User clicks on "Forgot Password."	
		2- System checks if the provided email or phone number is registered in the System before.
		3- System sends a password reset link.
	4- User follows the instructions to reset their password.	5- System verifies the reset and confirms the password change.
<b>Exceptions</b>	<b>Actor Action</b>	<b>System Response</b>
		2- System finds that the email or phone number is not registered, the system displays: "No account found with this information. Unable to send the reset link or Code"

Table 3.4. Kids Section Scenario

<b>Use Case ID</b>	UC#4	
<b>Use Case Name</b>	Kids Section	
<b>Actors</b>	Customer	
<b>Primary Actor</b>	Customer	
<b>Description</b>	Users can navigate products or brands intended for children. This section includes kids' clothing designed specifically for a family-centered audience.	
<b>Pre-Condition(s)</b>	The user is on the main browsing interface or has selected the option for Kids-specific products.	
<b>Post-Condition(s)</b>	The user is presented with a list of products or brands for kids.	
<b>Flow of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1- User clicks on Homepage.	
		2- System displays a page with all Brands.
	3- User clicks on the "Kids Section."	
		4- System loads and organizes kids' Products.
<b>Exceptions</b>	<b>Actor Action</b>	<b>System Response</b>
		4- There are no items, the system displays: "No items available in this section."

Table 3.5. Gender-Based Sections Scenario

<b>Use Case ID</b>	UC#5	
<b>Use Case Name</b>	Gender-Based Sections	
<b>Actors</b>	Customer	
<b>Primary Actor</b>	Customer	
<b>Description</b>	The user is on the main browsing interface or has selected the option for women/men-specific products.	
<b>Pre-Condition(s)</b>	The user clicks on "Women" or "Men" Section and explores various products available for women.	
<b>Post-Condition(s)</b>	The user is shown a list of products or brands for women.	
<b>Flow of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1- User clicks on Homepage.	
		2- System displays a page with all Brands.
	3- User clicks on the "Women" or "Men" Section.	
		4- System loads and organizes (shirts, pants, etc....) Products related to the Women or Men Department.
<b>Exceptions</b>	<b>Actor Action</b>	<b>System Response</b>
		4- There are no items, the system displays: "No items available in this section."

Table 3.6. List by Brands Use Case

<b>Use Case ID</b>	UC#6	
<b>Use Case Name</b>	List by Brands	
<b>Actors</b>	Customer	
<b>Primary Actor</b>	Customer	
<b>Description</b>	Users can explore products filtered by specific brands, making it easier for them to locate items from brands they trust or prefer.	
<b>Pre-Condition(s)</b>	The user has access to the "List by Brand" feature from the main navigation menu or search options.	
<b>Post-Condition(s)</b>	User successfully views a list of products associated with the selected brand.	
<b>Flow of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1- User selects the "List by Brand" option from the main navigation menu or search bar.	
		2- The system retrieves the list of available brands from the database.
		3- The system displays the brand list, organized alphabetically or by popularity, and provides a search bar to filter the list by brand name if desired.
	4- User scrolls through the brand list.	
	5- User clicks on the selected brand to view products associated with it.	
		6- The system retrieves all products associated with the selected brand from the database.
		7- The system displays the list of products for that brand, including product images, names, prices, and brief descriptions.

	8- User scrolls through the products under the selected brand and can click on individual items to view more details or return to the brand list.	
		9- The system provides detailed information for the selected product, including specifications, available sizes, colors, and stock status.
<b>Exceptions</b>	<b>Actor Action</b>	<b>System Response</b>
		6- The system retrieves all products associated with the selected brand from the database.

Table 3.7. Search by Image Scenario

<b>Use Case ID</b>	UC#7	
<b>Use Case Name</b>	Searching by image	
<b>Actors</b>	Customer, E-commerce System, Image Recognition System	
<b>Primary Actor</b>	Customer	
<b>Description</b>	Allowing users to search for similar clothes using a photo	
<b>Pre-Condition(s)</b>	The user already has an account and is logged in.	
<b>Post-Condition(s)</b>	Users will be able to view a list of products that match the uploaded user image.	
<b>Flow of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1- User opens the search page	
	2- User clicks on the image icon.	
	3- User chooses the image to be uploaded.	
		4- System recognizes the image
		5- System collects similar products and displays them on the search page.
<b>Exceptions</b>	<b>Actor Action</b>	<b>System Response</b>
	1- Uploading large image	1- Asks for smaller size
	2- Uploading unsupported format	2- Asks for supported formats
	3- Uploaded Image cannot be recognized	3- Shows warning message
		3- Shows an empty search page (Suggest similar products)
	4- Uploaded Image does not have similar products	4- Shows an empty search page (Suggest similar products)

Table 3.8. Search by Brand Name Scenario

<b>Use Case ID</b>	UC#8	
<b>Use Case Name</b>	Searching by brand name	
<b>Actors</b>	Customer, E-commerce System	
<b>Primary Actor</b>	Customer	
<b>Description</b>	Search for clothes produced by specific brands, showing brand catalogs, and the ability to filter the brand products by type.	
<b>Pre-Condition(s)</b>	—	
<b>Post-Condition(s)</b>	View a list of products produced by a specific brand.	
<b>Flow of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1- Open the search page.	
	2- Choose to search by brand name.	
	3- Enter the brand name.	
		4- Collect products from the brand.
	5- A specific brand catalog page is shown.	
<b>Exceptions</b>	<b>Actor Action</b>	<b>System Response</b>
	1- User does not want a brand with a specific name.	
		2- Suggest brands with similar words.

Table 3.9. Search Filters Scenario

<b>Use Case ID</b>	UC#9	
<b>Use Case Name</b>	Searching Filtering	
<b>Actors</b>	Customer, E-commerce System	
<b>Primary Actor</b>	Customer	
<b>Description</b>	Allow users to filter products by: (Brand – Color – Size – Category – Department)	
<b>Pre-Condition(s)</b>	—	
<b>Post-Condition(s)</b>	Users will be able to filter the results of products with various filter options.	
<b>Flow of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1- Open the search page.	
	2- Search by any method.	
		3- Collect products based on the search method.
<b>Alternative</b>	<b>Actor Action</b>	<b>System Response</b>
	1- Choose the filter by price and define the price range.	1- Filter existing products that match the specific price range.
	2- Choose the filter by product type (shirts, skirts, etc....).	2- Filter existing products based on their types (shirts, skirts, etc....).
	3- Choose the filter by color (red, green, blue, etc....).	3- Filter existing products that match the chosen color.
	4- Choose the filter by size (X Small / XX Small / Small / Medium / Large / X Large).	4- Filter existing products that match the chosen size.
	5- Choose the filter by popularity.	5- Filter existing products based on the most purchased ones.
	6- Choose the filter by latest arrivals.	6- Filter existing products that have recently been published.

Table 3.10. Manage Ratings Scenario

<b>Use Case ID</b>	UC#10	
<b>Use Case Name</b>	Manage Customer Ratings	
<b>Actors</b>	Customer	
<b>Primary Actor</b>	Customer	
<b>Description</b>	Customers can add a rating to a product or remove a previous rating they had typed for a product.	
<b>Pre-Condition(s)</b>	Customers must be logged in to manage their product ratings. Customers must have purchased the product beforehand.	
<b>Post-Condition(s)</b>	Product rating is updated upon the latest customer rating.	
<b>Flow of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1- Customer purchases a product.	
	2- Customer adds a rating to the product.	
		3- Rating is added to the overall rating of the product, which is recalculated based on the number of users who rated this product.
	4- Customer removes a rating he/she has added to the product.	
		5- Rating is removed from the overall rating of the product, which is recalculated based on the number of users who rated this product.
<b>Exceptions</b>	<b>Actor Action</b>	<b>System Response</b>
	1- Customer tries to add a rating to a product which he/she has not purchased.	
		2- System does not allow users to rate products they have not purchased yet.

Table 3.11. Manage Orders Scenario

<b>Use Case ID</b>	UC#11	
<b>Use Case Name</b>	Manage Customer Orders	
<b>Actors</b>	Customer, Brand	
<b>Primary Actor</b>	Customer	
<b>Description</b>	Customers can cancel a pending order, track order status, view order history, and return an order.	
<b>Pre-Condition(s)</b>	Customers must be logged in to be able to manage their orders.	
<b>Post-Condition(s)</b>	Customers can manage their orders successfully.	
<b>Flow of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1- Customer wants to cancel an order.	
		2- System checks the order status, if it is not shipped it cancels the order, otherwise it will not allow the customer to cancel the order.
	3- Customer wants to view their order history.	
		4- System displays their previous orders. each with its details and its status.
	5- Customer wants to return an order.	
		6- System checks if the order has passed a certain time or not, if it's still in the time frame that is eligible to return, it begins the return process.
	7- Customer wants to track the order status.	
		8- System displays the current user orders' status.
<b>Exceptions</b>	<b>Actor Action</b>	<b>System Response</b>
	1- Customer tries to return an order after the period of return has passed.	
		2- System should not allow order returns after a certain period of the order date.

Table 3.12. Edit Profile Scenario

<b>Use Case ID</b>	UC#12	
<b>Use Case Name</b>	Edit Customer Profile	
<b>Actors</b>	Customer	
<b>Primary Actor</b>	Customer	
<b>Description</b>	Customers can edit their profile details like updating their username, full name, phone, or profile picture.	
<b>Pre-Condition(s)</b>	Customers must be logged in to be able to edit their profiles.	
<b>Post-Condition(s)</b>	Customer details are updated according to the edits made.	
<b>Flow of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1- Customer enters the new data to update.	
<b>Exceptions</b>	<b>Actor Action</b>	<b>System Response</b>
	1- Customer tries to change their phone number with an invalid number.	
		2- System saves changes to the database.
		2- System checks the phone number and detects that it is invalid, so it aborts the process of updating the number.

Table 3.13. AI Virtual Try-On Scenario

<b>Use Case ID</b>	UC#13	
<b>Use Case Name</b>	AI Virtual Try-On	
<b>Actors</b>	Customer, Brand	
<b>Primary Actor</b>	Customer	
<b>Description</b>	Customers should be able to view clothes virtually on them before buying clothes.	
<b>Pre-Condition(s)</b>	Customers must be logged in to be able to use AI.	
<b>Post-Condition(s)</b>	Customers can benefit from AI Try-On capabilities.	
<b>Flow of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1- Customers want to view a certain product on them through their camera.	
		2- System should allow logged in customers to view virtual mockups of clothes on them before buying real clothes.
	3- Customer wants to view a virtual model of a certain product, but it must be offered by the brand.	
<b>Exceptions</b>	<b>Actor Action</b>	<b>System Response</b>
	1- Customer wants to view a virtual model of a certain product, but it is not offered by the brand.	
		2- System should not allow the option for viewing virtual models for products that do not support this feature.

Table 3.14. AI Recommendations Scenario

<b>Use Case ID</b>	UC#14	
<b>Use Case Name</b>	Dynamic AI Recommendation	
<b>Actors</b>	Customer, Brand	
<b>Primary Actor</b>	Customer	
<b>Description</b>	Customers can view recommendations for the most liked products, picked together, and products they may also like.	
<b>Pre-Condition(s)</b>	Customers must be logged in to view AI recommendations.	
<b>Post-Condition(s)</b>	Customers can benefit from AI to ease their shopping.	
<b>Flow of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1- Customers browse the website normally.	
		2- System should display a recommended section that contains products they may like based on their purchase history.
	3- Customers browse for a certain product from a brand they've purchased from before.	
		4- System should display recommendations for the most picked together items to help customers buy suitable packages of products together.
	5- Customers browse the page of a certain brand.	
		6- System should display the most liked products from this brand to help them choose the best product to buy.

Table 3.15. Add to Cart Scenario

<b>Use Case ID</b>	UC#15	
<b>Use Case Name</b>	Add to cart	
<b>Actors</b>	Logged Customer	
<b>Primary Actor</b>	Logged Customer	
<b>Description</b>	Customer can add a product to the cart to complete a purchase.	
<b>Pre-Condition(s)</b>	Customer is logged in and the product exists in the inventory with the quantity specified by the user.	
<b>Post-Condition(s)</b>	The product with the specified quantity is added to the cart.	
<b>Flow of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1- Customer navigates to the product page.	
		2- System displays the product details, including available quantity.
	3- Customer selects the desired quantity and clicks "Add to Cart".	
		4- System checks if the selected quantity is available in inventory, then adds the specified quantity of the product to the customer's cart.
<b>Exceptions</b>	<b>Actor Action</b>	<b>System Response</b>
	3- Customer selects a quantity higher than what is available in the inventory and clicks "Add to Cart."	
		4- System checks if the selected quantity is available in inventory, then displays an error message indicating the quantity exceeds available stock.

Table 3.16. View Cart Summary Scenario

<b>Use Case ID</b>	UC#16	
<b>Use Case Name</b>	View Cart Summary	
<b>Actors</b>	Logged Customer	
<b>Primary Actor</b>	Logged Customer	
<b>Description</b>	The customer can view a summary of their cart, which displays each product's name, quantity, price per product, and the subtotal of all items in the cart.	
<b>Pre-Condition(s)</b>	The customer is logged in	
<b>Post-Condition(s)</b>	The customer's cart summary is displayed, listing all products with their respective quantities, prices, and the subtotal.	
<b>Flow of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1- Customer navigates to the "Cart" or "View Cart" option.	
<b>Alternative</b>		2- System displays a summary of the cart, including each product's name, quantity, price per product, and the subtotal for all items.
	<b>Actor Action</b>	<b>System Response</b>
		2- System checks the cart contents and finds it empty then displays a message indicating that the cart is empty, along with an option to continue shopping.

Table 3.17. Change Product Quantity Scenario

<b>Use Case ID</b>	UC#17	
<b>Use Case Name</b>	Change Product Quantity	
<b>Actors</b>	Logged Customer	
<b>Primary Actor</b>	Logged Customer	
<b>Description</b>	The customer can adjust the quantity of products already in their cart. The system updates the cart with the new quantities, ensuring the requested quantity is available in the inventory.	
<b>Pre-Condition(s)</b>	Customers are logged in and have at least one item in their cart.	
<b>Post-Condition(s)</b>	The updated quantity of each product is reflected in the customer's cart.	
<b>Flow of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1- Customer navigates to the "Cart" or "View Cart" option then selects a product and changes its quantity.	
<b>Exceptions</b>	<b>Actor Action</b>	<b>System Response</b>
	1- Customer navigates to the "Cart" or "View Cart" option then selects a product and changes its quantity.	
<b>Exceptions</b>		2- System checks the inventory and finds that the requested quantity exceeds available stock then displays an error message indicating that the requested quantity is not available.

Table 3.18. Delete Products Scenario

<b>Use Case ID</b>	UC#18	
<b>Use Case Name</b>	Delete Products	
<b>Actors</b>	Logged Customer	
<b>Primary Actor</b>	Logged Customer	
<b>Description</b>	Customers can remove products from their cart.	
<b>Pre-Condition(s)</b>	Customers are logged in and have at least one item in their cart.	
<b>Post-Condition(s)</b>	Selected products are removed from the customer's cart, and the subtotal is updated accordingly.	
<b>Flow of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1- Customer navigates to the "Cart" or "View Cart" option then selects a product to delete.	
		2- System removes the selected product from the cart then recalculates and displays the updated subtotal for the cart.

Table 3.19. Manage Addresses Scenario

<b>Use Case ID</b>	UC#19	
<b>Use Case Name</b>	Manage Addresses	
<b>Actors</b>	Logged Customer	
<b>Primary Actor</b>	Logged Customer	
<b>Description</b>	The customer can add new shipping addresses or edit previously saved addresses. Input fields for adding or editing an address include full name, address, and phone number.	
<b>Pre-Condition(s)</b>	The customer must be logged in	
<b>Post-Condition(s)</b>	<ul style="list-style-type: none"> <li>The customer's shipping address list is updated, either by adding a new address or editing an existing one.</li> <li>Customer can choose a certain address to use when placing an order.</li> </ul>	
<b>Flow of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1- Customer navigates to the "Account Settings" or "Manage Addresses" section.	
		2- System displays the customer's saved shipping addresses.
	3- Customer selects the option to add a new address.	
		4- System displays input fields for full name, address, and phone number.
	5- Customer enters the required details and submits the address.	
<b>Alternative</b>	<b>Actor Action</b>	<b>System Response</b>
	3- Customer selects an existing address to edit.	
		4- System displays the current details of the selected address.

	5- Customer updates the address, phone number, or name as needed.	
		6- System validates the input fields then saves the updated address to the customer's account.
<b>Exception</b>	<b>Actor Action</b>	<b>System Response</b>
	5- Customer enters invalid or incomplete address information	6- System displays an error message for the missing or invalid fields.

Table 3.20. Choose Shipping Method Scenario

<b>Use Case ID</b>	UC#20	
<b>Use Case Name</b>	Choose Shipping Method	
<b>Actors</b>	Logged Customer	
<b>Primary Actor</b>	Logged Customer	
<b>Description</b>	The system displays available shipping methods and calculates the cost based on the customer's location and cart contents. The customer selects a preferred shipping method for their order.	
<b>Pre-Condition(s)</b>	The customer is logged in and has added items to their cart and is ready to check out.	
<b>Post-Condition(s)</b>	The customer has selected a shipping method, and the system updates the order summary with the selected shipping cost.	
<b>Flow of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1- Customer navigates to the checkout page.	
		2- System displays the available shipping methods based on the customer's location and the items in their cart and calculates the cost for each shipping method.
	3- Customer selects a preferred shipping method.	
		4- System updates the order summary with the selected shipping method and cost.

Table 3.21. View Order Summary Scenario

<b>Use Case ID</b>	UC#21	
<b>Use Case Name</b>	View Order Summary	
<b>Actors</b>	Logged Customer	
<b>Primary Actor</b>	Logged Customer	
<b>Description</b>	The customer can view a final order summary, which includes itemized costs such as product prices, shipping fees, discounts, and the total cost before confirming the order.	
<b>Pre-Condition(s)</b>	The customer is logged in and has completed selecting products and is ready to review the final order.	
<b>Post-Condition(s)</b>	The customer has reviewed the order summary and can proceed to confirm or modify the order.	
<b>Flow of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1- Customer navigates to the "Checkout" page.	
		2- System displays the final order summary
	3- Customer reviews the details of the order summary and confirms that the order details are correct and proceeds to the next step (e.g., Place order).	

Table 3.22. Access Products Dashboard Scenario

<b>Use Case ID</b>	UC#22	
<b>Use Case Name</b>	Access Products Dashboard	
<b>Actors</b>	Brand Owner	
<b>Primary Actor</b>	Brand Owner	
<b>Description</b>	To enable the Brand Owner to view detailed analytics about the performance of his products.	
<b>Pre-Condition(s)</b>	The Brand Owner has an account and is logged in.	
<b>Post-Condition(s)</b>	The Brand Owner can view detailed analytics about their products through the dashboard.	
<b>Flow of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1- The brand owner logs in and selects the “Products Dashboard.”	
		2- The system displays a list of the brand's products with key performance indicators.
	3- The brand owner accesses detailed information about the performance of each product.	

Table 3.23. View Brand Profile Scenario

<b>Use Case ID</b>	UC#23	
<b>Use Case Name</b>	View Brand Profile	
<b>Actors</b>	Brand Owner	
<b>Primary Actor</b>	Brand Owner	
<b>Description</b>	Allow the brand owner to view their brand profile information.	
<b>Pre-Condition(s)</b>	The brand owner must be logged into the system.	
<b>Post-Condition(s)</b>	The brand profile is displayed.	
<b>Flow of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1- Brand owner navigates to the profile page.	
		2- System displays the brand profile information.

Table 3.24. Edit Brand Profile Scenario

<b>Use Case ID</b>	UC#24	
<b>Use Case Name</b>	Edit Brand Profile Information	
<b>Actors</b>	Brand Owner	
<b>Primary Actor</b>	Brand Owner	
<b>Description</b>	Allow the brand owner to update their brand profile information.	
<b>Pre-Condition(s)</b>	The brand owner must be logged into the system.	
<b>Post-Condition(s)</b>	The brand profile information is updated in the system and the changes are visible to customers who view the brand profile.	
<b>Flow of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1- Brand owner navigates to the profile page.	
		2- System displays the brand profile information.
	3- Brand owner selects "Edit Profile".	
		4- System displays fields for editing.
	5- Brand owner updates profile details (brand name, industry, contact information) and submits the updates.	
<b>Exception #1</b>	<b>Actor Action</b>	<b>System Response</b>
	5- Brand owner leaves some required fields empty (Brand name, industry, contact information) and submits the updates.	
		6- System displays error messages for the empty fields.
	7- Brand owner fills the empty fields and resubmits.	

	Actor Action	System Response
<b>Exception #2</b>	5- Brand owner enters an invalid email or phone number in the contact information fields and submits the updates.	
		6- System displays error messages for the invalid fields.
	7- Brand owner corrects the invalid inputs and resubmits.	

Table 3.25. Add New Product Scenario

<b>Use Case ID</b>	UC#25	
<b>Use Case Name</b>	Add New Product	
<b>Actors</b>	Brand Owner	
<b>Primary Actor</b>	Brand Owner	
<b>Description</b>	Allow the brand owner to add new products to their catalog.	
<b>Pre-Condition(s)</b>	The brand owner must be logged into the system.	
<b>Post-Condition(s)</b>	New products are added to the catalog and are available for customers to purchase.	
<b>Flow of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1- Brand owner navigates to "Manage Products".	
		2- System displays the product management interface for this brand.
	3- Brand owner selects "Add Product".	
		4- System displays fields for new product details.
	5- Brand Owner enters product details (name, category, color, size, price, quantity, description, department, material, image, discount if found) and submits the product.	
		6- System validates input fields. saves the product and updates the catalog.
<b>Exception #1</b>	<b>Actor Action</b>	<b>System Response</b>
	5- Brand owner leaves some required fields (product name, price, quantity, image, description) empty and submits the product.	
		6- System displays error messages for the empty fields.
<b>Exception #2</b>	<b>Actor Action</b>	<b>System Response</b>
	7- Brand owner fills the empty fields.	

	5- Brand owner enters non-numeric/negative values in a numeric field (price, quantity, discount), floating numbers in the quantity, or upload unsupported format images and submits the product.	
		6- System rejects the invalid inputs and displays error messages for the specific fields.
	7- Brand owner re-enters valid values.	
<b>Exception #3</b>	<b>Actor Action</b>	<b>System Response</b>
	5- Brand owner submits a product that exists in the catalog with the same name.	
		6- System displays an error message for the brand owner.

Table 3.26. Edit Product Scenario

<b>Use Case ID</b>	UC#26	
<b>Use Case Name</b>	Edit Product	
<b>Actors</b>	Brand Owner	
<b>Primary Actor</b>	Brand Owner	
<b>Description</b>	Allow the brand owner to modify details of an existing product.	
<b>Pre-Condition(s)</b>	The brand owner must be logged into the system and has products listed in their catalog.	
<b>Post-Condition(s)</b>	The selected product's information is updated in the catalog and the modified details are visible to customers.	
	<b>Actor Action</b>	<b>System Response</b>
<b>Flow of events</b>	1- Brand owner navigates to "Manage Products".	
		2- System displays the product management interface for this brand containing the catalog.
	3- Brand owner selects a product to edit.	
		4- System displays the product details.
	5- Brand owner modifies the product details and submits the updates.	
		6- System validates the input fields and saves the updates.
<b>Exceptions</b>	The same exceptions that could occur in the 'Add Product' UC.	

Table 3.27. Delete Product Scenario

<b>Use Case ID</b>	UC#27	
<b>Use Case Name</b>	Delete Product	
<b>Actors</b>	Brand Owner	
<b>Primary Actor</b>	Brand Owner	
<b>Description</b>	Allow the brand owner to delete a product from their catalog.	
<b>Pre-Condition(s)</b>	The brand owner must be logged into the system and has products listed in their catalog.	
<b>Post-Condition(s)</b>	The selected product is removed from the catalog, and it is no longer visible to customers.	
<b>Flow of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1- Brand owner clicks on "Manage Products".	
		2- System displays the product management interface for this brand containing the catalog.
	3- Brand owner selects a product to delete.	
		4- System displays a confirmation of the deletion.
	5- Brand owner confirms the deletion.	
		6- System deletes the product and updates the catalog.
<b>Alternative</b>	<b>Actor Action</b>	<b>System Response</b>
	5- Brand owner cancels the deletion.	
		6- System returns to the catalog.

# Chapter 4 – ELEVATE Design and Implementation

## 4.1 Tools and Technologies Used

The ELEVATE platform is engineered using a robust and modern technology stack, carefully selected for high performance, scalability, security, and exceptional user experience. This section details the core tools and technologies employed in the development of both backend services and user-facing applications, along with their respective rationales.

### 4.1.1 Backend Development

- **Node.js with Express.js:**
  - Node.js and Express.js were selected for their asynchronous, event-driven architecture, enabling efficient handling of concurrent requests crucial for e-commerce. This combination provides a robust foundation for rapid RESTful API development, further supported by a large community and broad hosting compatibility (e.g., Vercel).
- **Swagger (OpenAPI):**
  - Swagger (OpenAPI Specification) was integrated for API documentation and interactive testing. This streamlined development by providing a clear, standardized API description, enhancing team collaboration and accelerating frontend integration.

### 4.1.2 Data Storage and Management

- **Firebase (Firestore, Realtime Database, Firebase Storage, Firebase Authentication, Firebase Extensions):**
  - Firebase serves as the primary backend-as-a-service (BaaS), offering a comprehensive suite for robust, scalable infrastructure. Firestore, a flexible NoSQL database, handles core data with real-time synchronization. The Realtime Database provides low-latency updates, crucial for immediate user feedback on AI try-on model progress. Firebase Storage offers secure, scalable media storage, while Firebase Authentication manages user access securely. Firebase Extensions, particularly the Algolia extension, streamline development by automatically synchronizing Firestore data with search indexes.

#### 4.1.3 Search and Recommendation Services

- **Algolia (Search as a Service):**
  - Algolia was adopted for its powerful and fast search capabilities, delivering a superior full-text search experience with features like typo tolerance and instant results. Its recommendation models (e.g., "Looking Similar" and "Related Items") also provide personalized product suggestions, enhancing user engagement.

#### 4.1.4 Serverless Data Platform and AI Integration

- **Upstash Serverless Data Platform (CRON Jobs, Vector Database):**
  - Upstash provides serverless infrastructure for data operations. Its CRON Jobs automate periodic tasks like deleting expired orders and updating order states from external shipment companies. The Vector Database is crucial for image search, storing vector embeddings for efficient similarity lookups.
- **Google Services:**
  - **Vertex AI:** Generates high-quality image embeddings for visual search.
  - **Geolocation Services:** Google's Geolocation Services enable location-based features like shipping cost calculation.
- **AI Try-On Models (Replicate and Fal.AI):**
  - Replicate and Fal.AI provide AI models for realistic virtual try-on experiences. This core ELEVATE feature significantly enhances the online shopping experience and helps reduce return rates.

#### 4.1.5 Deployment and Hosting

- **Vercel** was chosen for hosting backend serverless functions and frontend websites due to its developer-centric approach, automatic deployments, and global CDN. Its seamless Git integration ensures fast and reliable updates.

#### 4.1.6 Frontend Development

- **Flutter with Cubit State Management (Mobile Application):**
  - Flutter, Google's cross-platform UI toolkit, was chosen for the mobile application to deliver a consistent, high-performance, and visually appealing experience across Android and iOS. Cubit (Bloc) was adopted for its simple, predictable, and testable state management, enabling efficient application scalability.
- **React.js with Vite, TypeScript, and MUI (Brand Owner Website):**
  - **React.js** - Selected as the foundation for the brand owner's website due to its component-based architecture, which enables the development of dynamic and interactive user interfaces.
  - **Vite** - Leveraged as the build tool for exceptional development speed and optimized production builds.
  - **TypeScript** - Adds type safety and enhances code maintainability, reducing runtime errors and improving developer productivity.
  - **Material-UI (MUI)** - Provides a comprehensive library of pre-built, accessible components that ensure design consistency across the platform.
  - **React Query** - Handles server state management with intelligent caching and background synchronization, minimizing unnecessary API calls.
  - **Formik & Yup** - Delivers a robust solution for complex form logic and schema-based validation, with Formik managing form state and Yup providing validation schemas.
  - **Snackbar** - Enables non-intrusive user notifications for real-time feedback and system alerts.

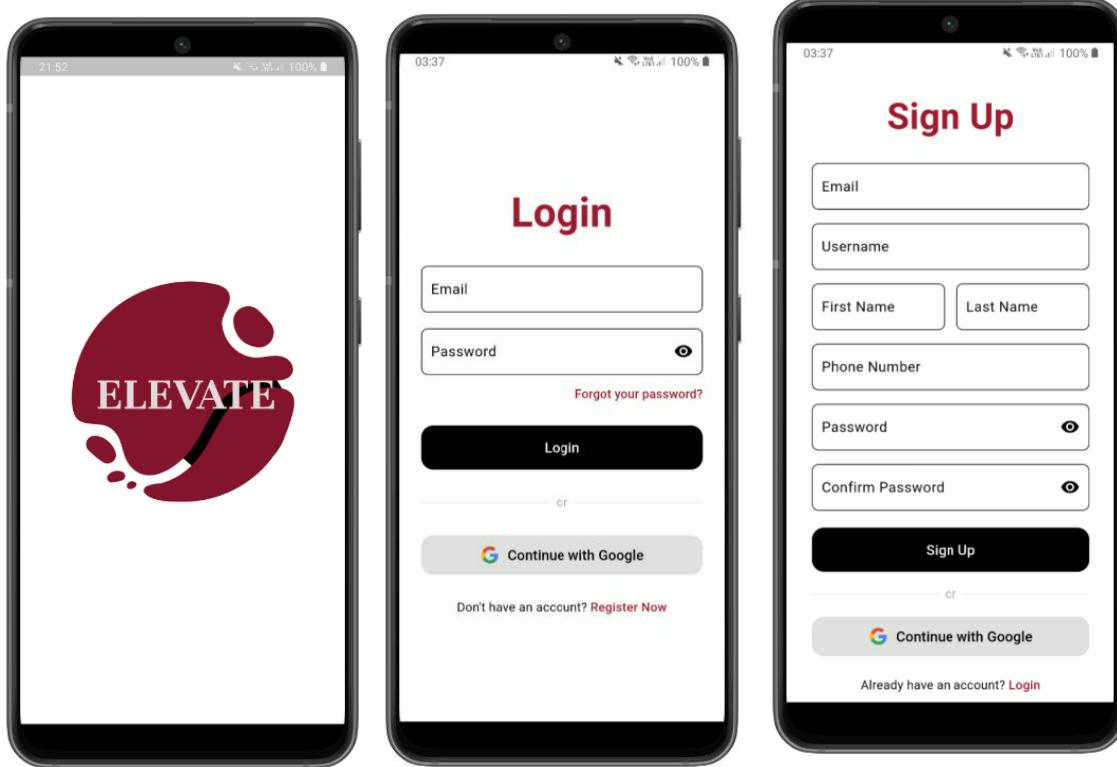
#### 4.1.7 Development Environment and Collaboration Tools

- **VS Code with Copilot:**
  - Visual Studio Code (VS Code) served as the primary Integrated Development Environment (IDE) for its extensive features, customization options, and broad language support. GitHub Copilot, an AI-powered coding assistant, was integrated to enhance developer productivity by providing real-time code suggestions and accelerating the development process.
- **Discord:**
  - Discord was utilized as the main communication and collaboration platform for the project team. Its features, including voice channels, text chat, and file sharing, facilitated efficient session meetings, real-time discussions, and seamless exchange of project-related documents.
- **Lucid:**
  - Lucid (e.g., Lucidchart) was employed for creating various project diagrams, including sequence diagrams, architecture diagrams, and use case diagrams. Its intuitive interface and collaborative features enabled the team to visually represent system flows and structures effectively.
- **Microsoft Word:**
  - Microsoft Word was used for authoring and managing the project's comprehensive documentation. Its robust features for text formatting, table creation, and collaborative editing were essential for producing a professional and well-structured project report.

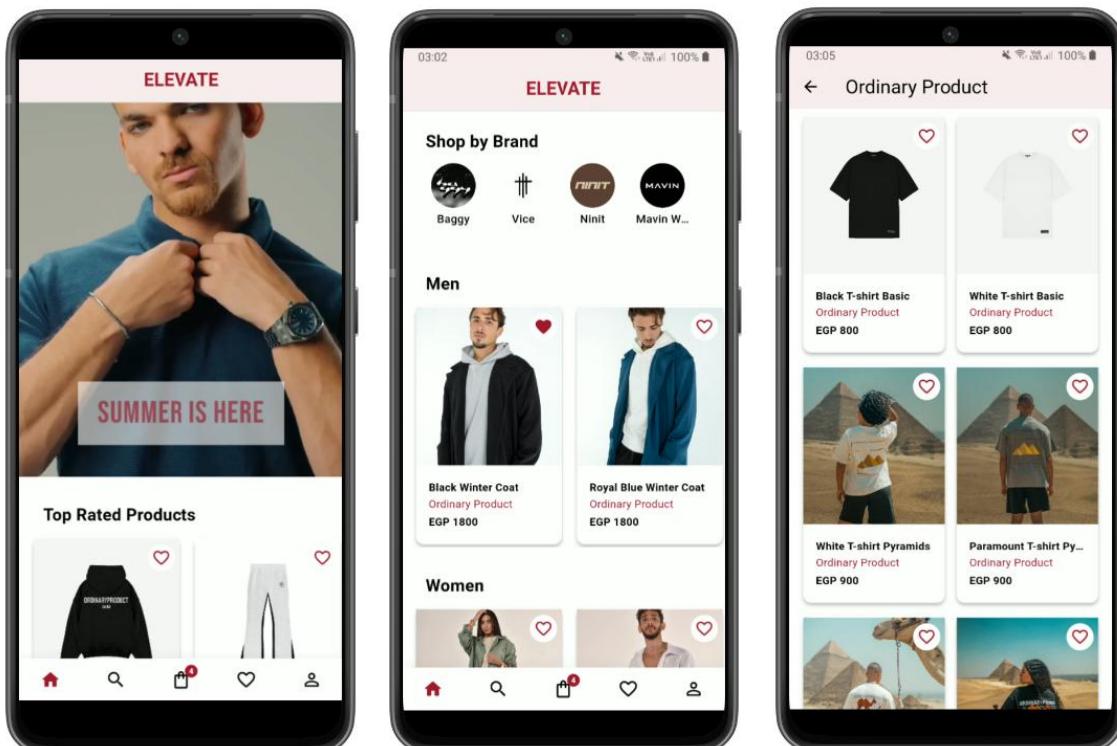
## 4.2 Application Screens (UI)

### 4.2.1 Customer Scenario

Register/Login to the mobile application by entering valid credentials

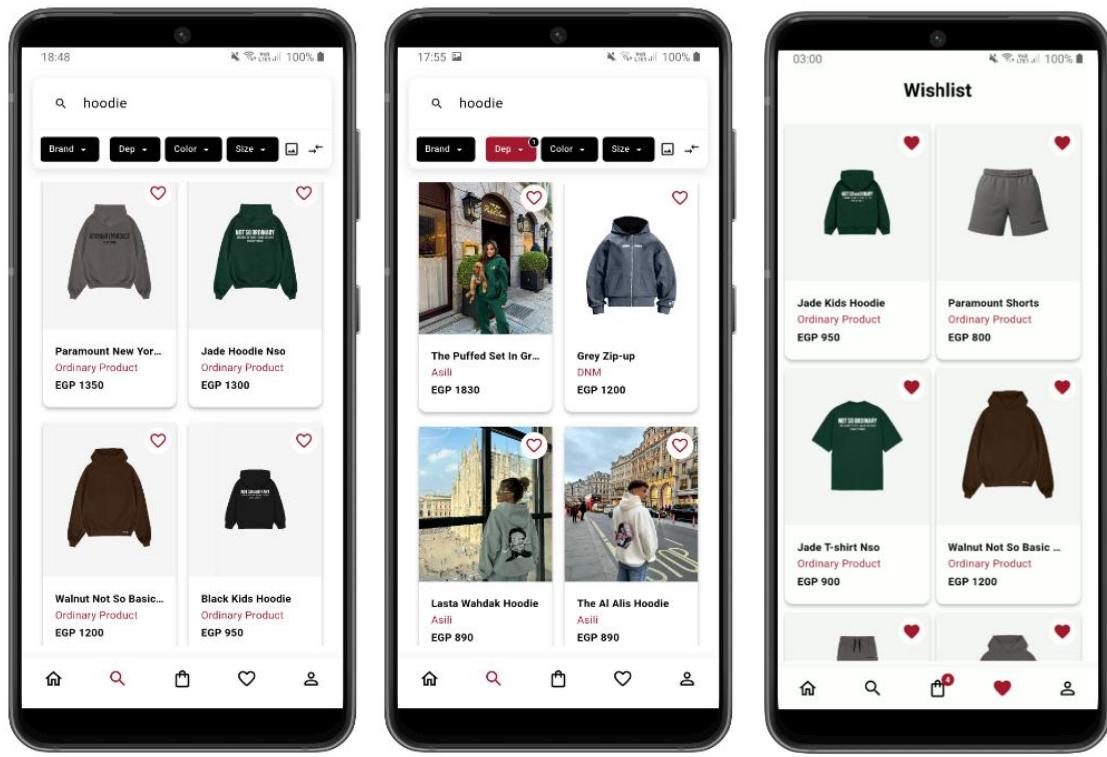


Explore the home page to shop for top products, see different brands and shop for products by a specific brand

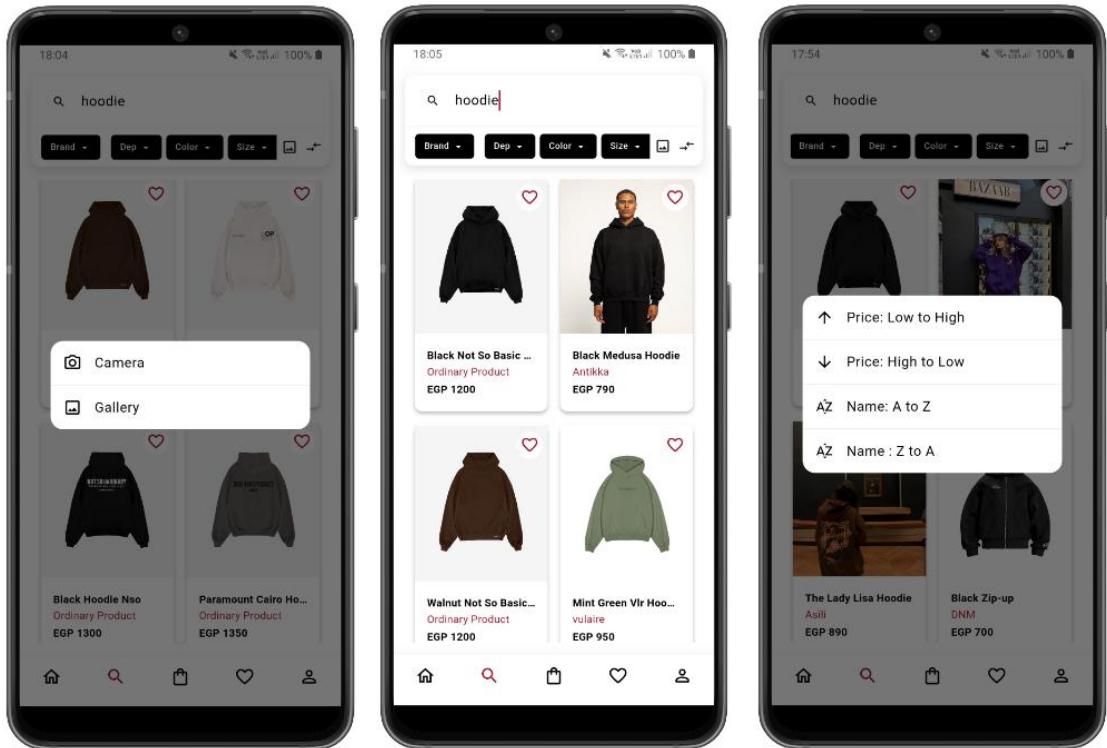


## Chapter 4 – ELEVATE Design and Implementation

Search for products, filter the results, and add a product to wishlist for later purchase

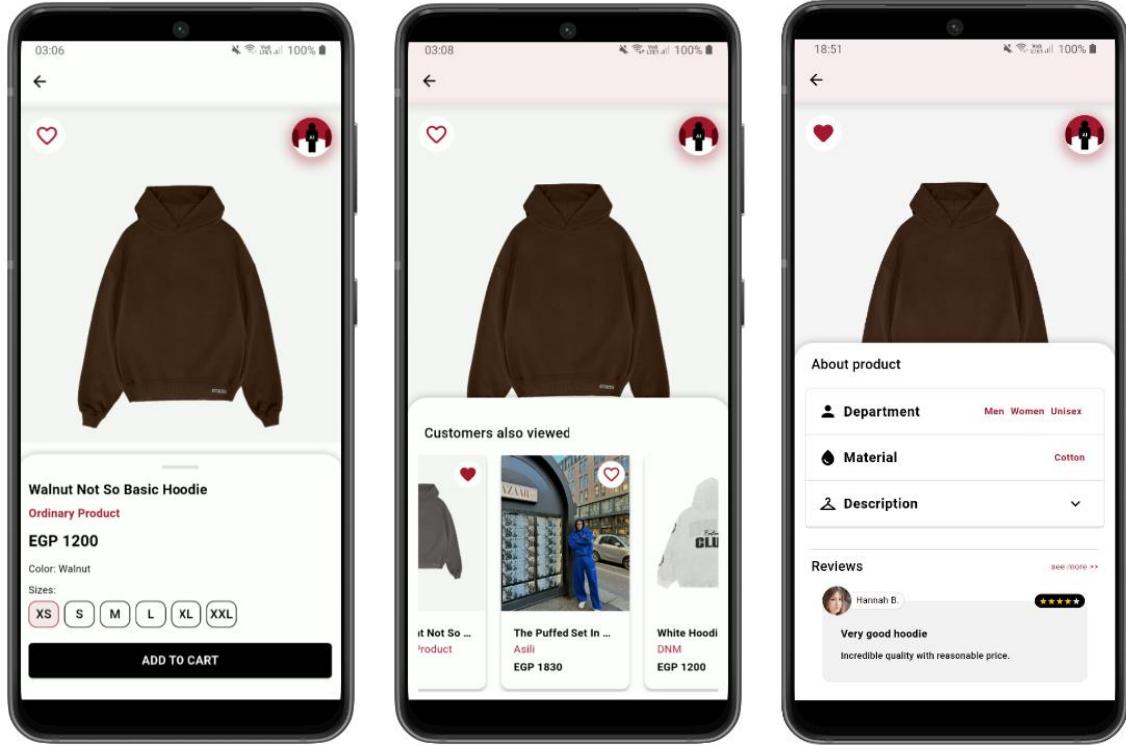


Visual search and sorting results by price or name

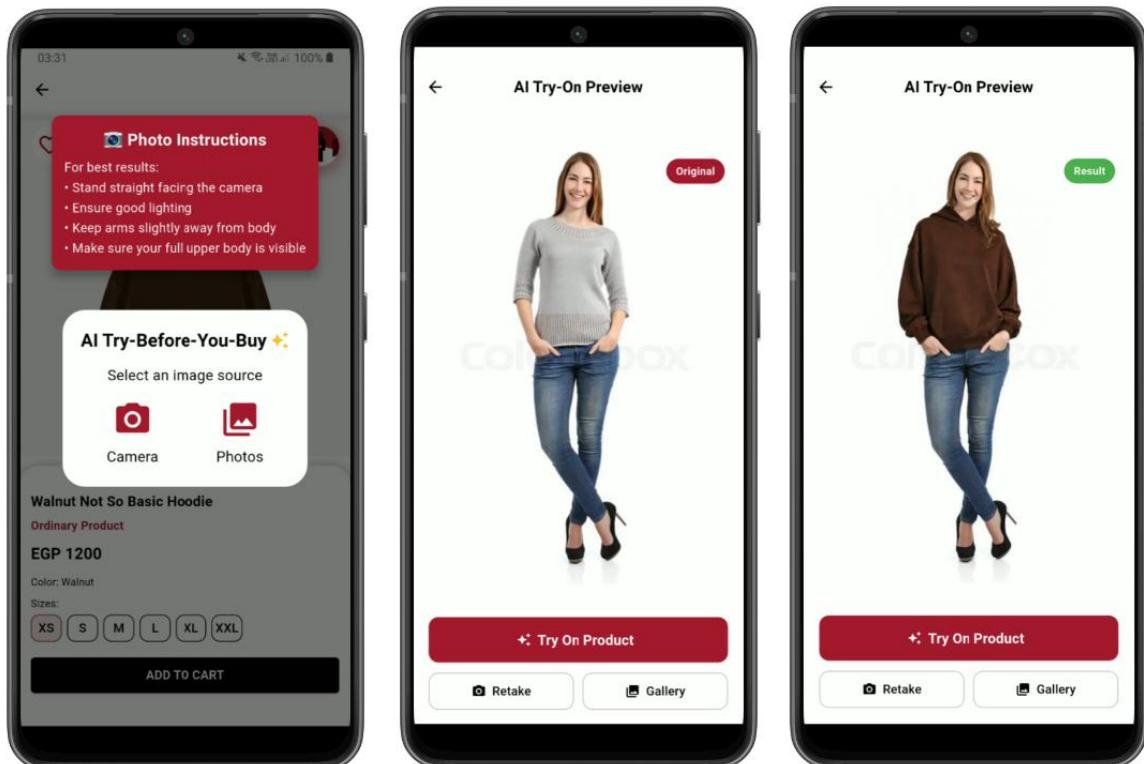


## Chapter 4 – ELEVATE Design and Implementation

Display product details, reviews, and AI recommendations

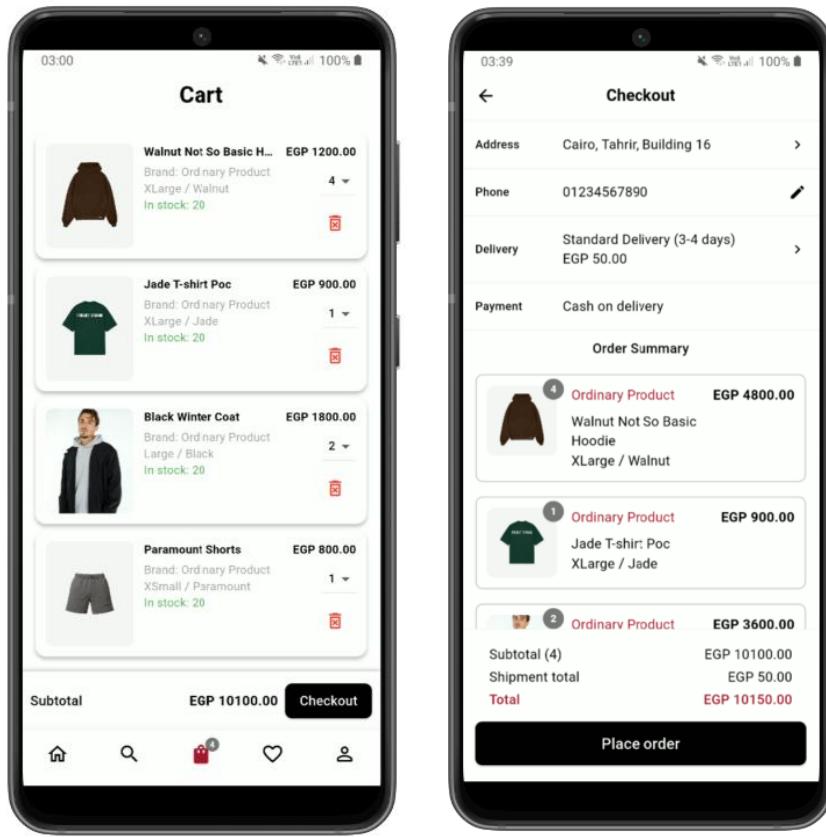


Virtually Try-On product using our fabulous Try-Before-You-Buy feature

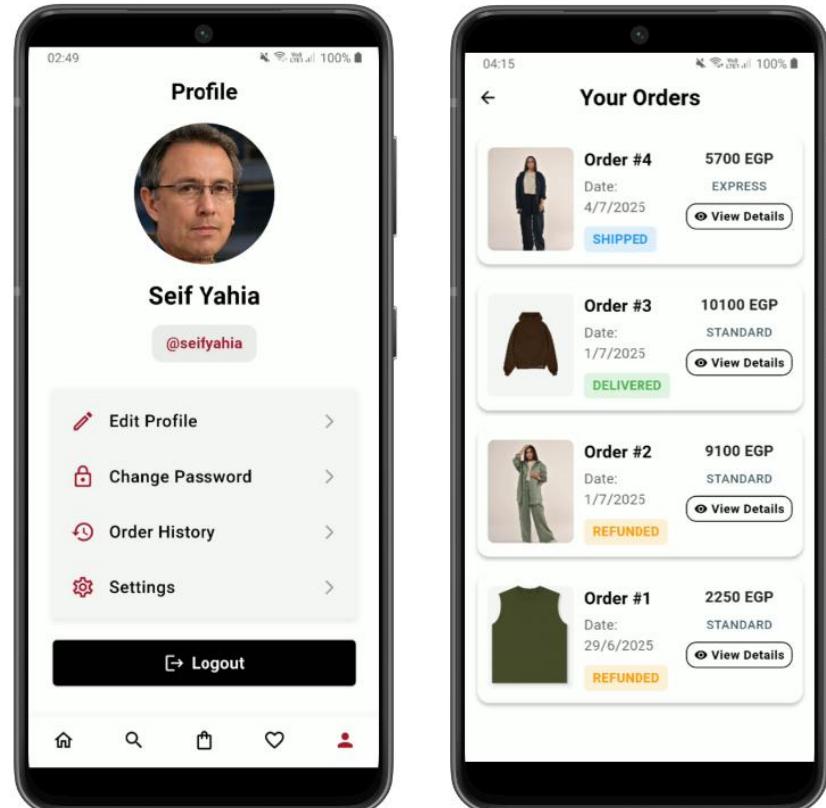


## Chapter 4 – ELEVATE Design and Implementation

Add products to cart and proceed to checkout to place an order

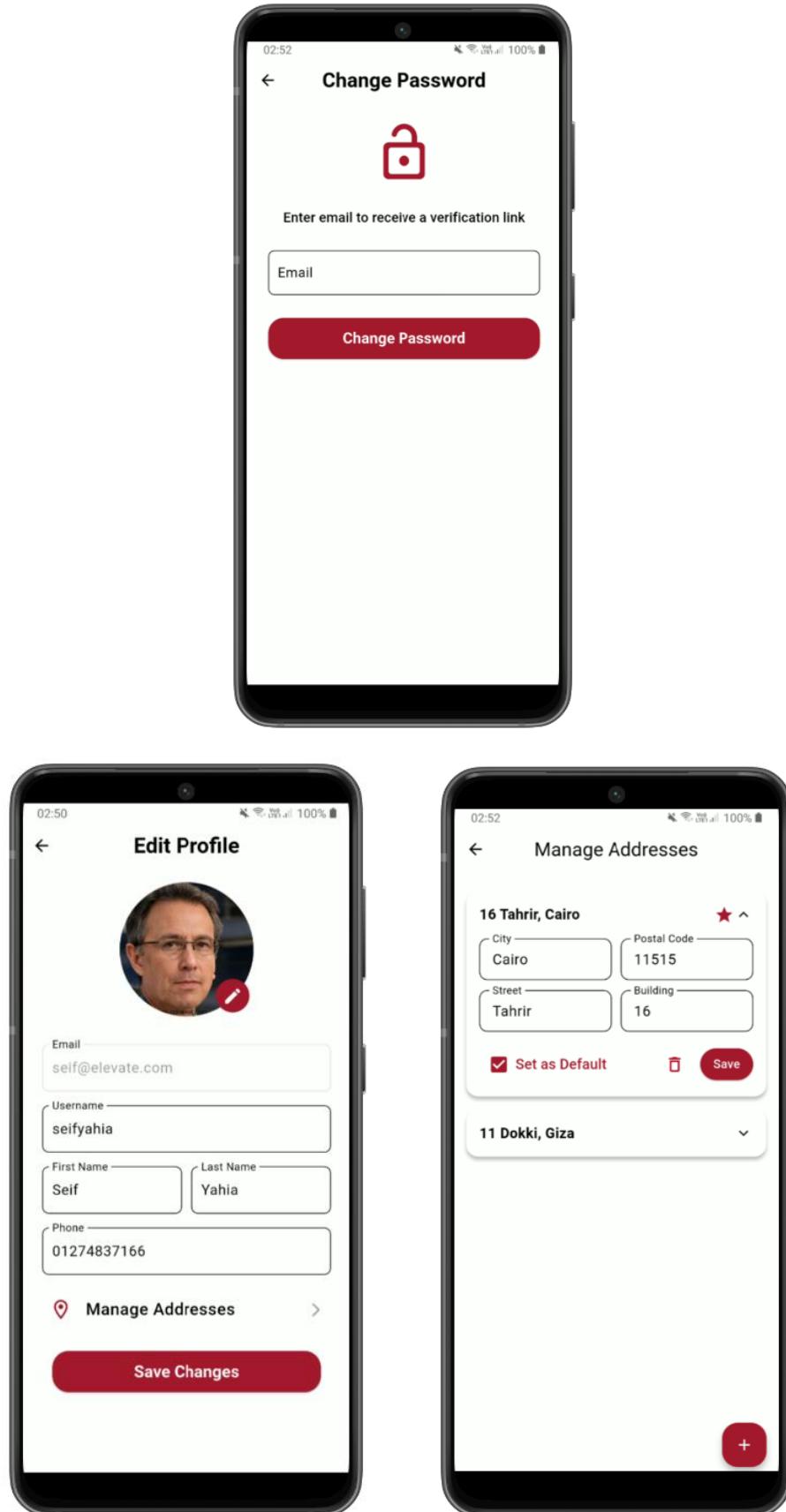


Check previous orders through the order history section in the profile page



## Chapter 4 – ELEVATE Design and Implementation

Update personal information through the edit profile page and update your addresses, profile photo, or change password



### 4.2.2 Brand Owner Scenario

View brand analytics through a dashboard of monthly sales

The screenshot shows the ELEVATE brand owner dashboard. On the left, a sidebar menu includes Dashboard, Profile, Orders, Manage Products, Settings, and Sign Out. The main area displays July Sales (13 products sold, 15100 EGP total sales, Black Sweats Cuffed as top product), Top Products (a table with data from Black Sweats Cuffed to Black Cairo Hoodie), Customer Reviews (a bar chart showing a rating of 4 from 1 to 5 based on 11 reviews), and Total Sales By Month (a bar chart showing sales from August 2025 to July 2025, with a total of 15.1k).

View brand profile details, contact information, and brand story

The screenshot shows the ELEVATE brand profile page. The sidebar menu includes Profile, Orders, Manage Products, Settings, and Sign Out. The main area shows the brand profile details: Brand Name (Ordinary Product), Brand Logo (a placeholder image labeled ORDINARYPRODUCT), Contacts (phone number 01234567890), and Socials (Instagram link https://www.instagram.com/ordinaryproduct\_). The brand story is listed as "NOT SO ORDINARY".

View brand subscription plan

The screenshot shows the ELEVATE settings page. The sidebar menu includes Dashboard, Profile, Orders, Manage Products, Settings, and Sign Out. The main area has tabs for ACCOUNT and SUBSCRIPTIONS. The SUBSCRIPTIONS tab is active, showing Subscriptions Info (Current Subscription: PREMIUM, Expires On: 24/07/2025).

## Chapter 4 – ELEVATE Design and Implementation

Update brand contact information, logo, and other details

Dashboard

**Profile**

Brand Name: Ordinary Product

Brand Email: self\_yahia\_op@elevate.com

Industry: clothes

Addresses: 1 Al Amal, New Cairo 1, Cairo 4722022

Websites: https://ordinaryproduct.com/

Contacts: 01234567890

Socials: https://www.instagram.com/ordinaryproduct\_/\_

Brand Story: NOT SO ORDINARY

Brand Logo: ORDINARYPRODUCT

Upload Images

Accepted formats: JPG, JPEG, PNG, WEBP • Maximum 5MB per image

Cancel Save

Display brand orders to be prepared for shipment

**ELEVATE**

**Orders**

**Self**  
Brand Owner

Dashboard

Profile

**Orders**

Manage Products

Settings

Sign Out

ORDERS

REFUNDS

Nude Linen Shirt  
2 orders

Pink Lemonade Sweats Cuffed  
1 orders

White Not So Basic Hoodie  
1 orders

View brand orders that will be received through refund requests

**ELEVATE**

**Orders**

**Self**  
Brand Owner

Dashboard

Profile

**Orders**

Manage Products

Settings

Sign Out

ORDERS

REFUNDS

Black Kids T-shirt  
3 refunds

Pearl Linen Shirt  
2 refunds

Cream Linen Shirt  
1 refunds

## Chapter 4 – ELEVATE Design and Implementation

Display current products of the brand

**ELEVATE**

**Manage Products**

Showing 1-9 of 117 products

- Pistachio Linen Shirt (Thumbnail: Woman in green shirt)
- Pearl Linen Shirt (Thumbnail: Man in white shirt)
- Midnight Linen Shirt (Thumbnail: Man in dark shirt)
- Cream Linen Shirt (Thumbnail: Woman in white shirt)
- Pistachio Linen Pants (Thumbnail: Woman in white pants)

Add a new product to the brand catalog

**Add New Product**

**Product Details**

- Product Name: White Shirt
- Description: Casual white shirt suitable for any occasion
- Material: Cotton
- Category: tops - shirts
- Department: men

**Product Variants**

Size	Colors	Price	Stock	Discount	Images	Actions
small	White	435 EGP	35	5%	1	

**Add Product**

Delete an existing or old product from the brand catalog

**ELEVATE**

**Manage Products**

Showing 1-9 of 117 products

- White Shirt (Thumbnail: White shirt)
- Pistachio Linen Shirt (Thumbnail: Man in white shirt)
- Pearl Linen Shirt (Thumbnail: Man in white shirt)
- Midnight Linen Shirt (Thumbnail: Man in dark shirt)
- Cream Linen Shirt (Thumbnail: Woman in white shirt)
- Pistachio Linen Pants (Thumbnail: Woman in white pants)

Confirm Delete

Are you sure you want to delete "White Shirt"?

⚠ This action cannot be undone. All product data including variants will be permanently removed.

**Delete**

## Chapter 4 – ELEVATE Design and Implementation

Edit specific product details (add more images, sizes, or discount)

Product Variants						
Size	Colors	Price	Stock	Discount	Images	Actl
Small	Pistachio	1200 EGP	2	0%	2	
Medium	Pistachio	1200 EGP	20	0%	0	
Large	Pistachio	1200 EGP	20	0%	0	

View brand owner/manager profile information

Edit brand owner/manager profile information

## Chapter 5 – ELEVATE Testing and Evaluation

### 5.1 Test Cases

Table 5.1. Customer Signup Test Case

<b>Test Case ID</b>	TC#1
<b>Test Case Name</b>	Customer Signup
<b>Endpoint</b>	POST /customers/signup
<b>Input Data</b>	Email, Username, First Name, Last Name, Phone Number, Password, Confirm Password
<b>Description</b>	New users can create an account through filling in personal details manually in the form.
<b>Expected Result</b>	Account is created successfully, and the user is ready to log in.

Table 5.2. Third-Party Signup Test Case

<b>Test Case ID</b>	TC#2
<b>Test Case Name</b>	Customer Third-Party Signup
<b>Endpoint</b>	POST /customers/third-party-signup
<b>Input Data</b>	Email is extracted from Google Sign-in prompt.
<b>Description</b>	New users can create an account on the platform by signing up through third party services like Google.
<b>Expected Result</b>	Account is created successfully, and the user is ready to log in.

Table 5.3. Customer Login Test Case

<b>Test Case ID</b>	TC#3
<b>Test Case Name</b>	Customer Login
<b>Endpoint</b>	POST /customers/login
<b>Input Data</b>	Email, Password
<b>Description</b>	Customers can log in by providing personal details manually in the form or signing in via third party services like Google.
<b>Expected Result</b>	Customer is logged in and can access all features.

Table 5.4. Brand Owner Login

<b>Test Case ID</b>	TC#4
<b>Test Case Name</b>	Brand Owner Login
<b>Input Data</b>	Email, Password
<b>Endpoint</b>	POST /brand-owners/login
<b>Description</b>	Brand owners can log in to the system to manage the products of the brand and access dashboard analytics.
<b>Expected Result</b>	Brand owner is logged in and can access all features.

Table 5.5. Brand Manager Login Test Case

<b>Test Case ID</b>	TC#5
<b>Test Case Name</b>	Brand Manager Login
<b>Input Data</b>	Email, Password
<b>Endpoint</b>	POST /brand-managers/login
<b>Description</b>	Brand managers can log in to the system to manage product refund requests and monitor the products stock.
<b>Expected Result</b>	Brand manager is logged in and can access selected features.

Table 5.6. Reset Password Test Case

<b>Test Case ID</b>	TC#6
<b>Test Case Name</b>	Forget / Change Password
<b>Input Data</b>	Email, New Password (via external link)
<b>Endpoint</b>	POST /utilities/send-password-reset
<b>Description</b>	Customers and brand owners can send a request with their email to receive a reset link to update their password.
<b>Expected Result</b>	The password is updated, and the user can login again.

Table 5.7. AI Virtual Try-On Test Case

<b>Test Case ID</b>	TC#7
<b>Test Case Name</b>	AI Virtual Try-On
<b>Input Data</b>	Customer Image
<b>Endpoint</b>	POST /utilities/try-on/start
<b>Description</b>	Customers can view clothes virtually in their photos before completing a purchase.
<b>Expected Result</b>	Customer view the result like a virtual fitting room.

Table 5.8. Add To Wishlist Test Case

<b>Test Case ID</b>	TC#8
<b>Test Case Name</b>	Add to wishlist
<b>Endpoint</b>	POST /customers/me/wishlist/items
<b>Input Data</b>	Product ID
<b>Description</b>	Customers add a product to their wishlist to access it later.
<b>Expected Result</b>	The product is added to the logged in customer's wishlist.

Table 5.9. Remove From Wishlist Test Case

<b>Test Case ID</b>	TC#9
<b>Test Case Name</b>	Remove from wishlist
<b>Endpoint</b>	DELETE /customers/me/wishlist/items
<b>Input Data</b>	Product ID
<b>Description</b>	Customers remove a product from their wishlist.
<b>Expected Result</b>	The product is removed from the logged in customer's wishlist.

Table 5.10. Add to Cart Test Case

<b>Test Case ID</b>	TC#10
<b>Test Case Name</b>	Add to Cart
<b>Endpoint</b>	POST /customers/me/cart/items
<b>Input Data</b>	Product ID, Variant ID, Quantity
<b>Description</b>	Customers add a product to their Cart to proceed to checkout.
<b>Expected Result</b>	The product is added to the logged in customer's Cart.

Table 5.11. Remove From Cart Test Case

<b>Test Case ID</b>	TC#11
<b>Test Case Name</b>	Remove from cart
<b>Endpoint</b>	DELETE /customers/me/cart/items/{id}
<b>Input Data</b>	Cart Item ID
<b>Description</b>	Customers remove a product from their cart.
<b>Expected Result</b>	The product is removed from the logged in customer's cart.

Table 5.12. Create New Review Test Case

<b>Test Case ID</b>	TC#12
<b>Test Case Name</b>	Create new review
<b>Endpoint</b>	POST /products/{productId}/reviews
<b>Input Data</b>	Product ID
<b>Description</b>	Customers create a new review on a product.
<b>Expected Result</b>	The review is added to the reviews section in the product details page.

Table 5.13. Update Review Test Case

<b>Test Case ID</b>	TC#13
<b>Test Case Name</b>	Update Review
<b>Endpoint</b>	PUT /reviews/{id}
<b>Input Data</b>	Review ID
<b>Description</b>	Customers update their review on a product.
<b>Expected Result</b>	The review is updated in reviews section in product details page.

Table 5.14. Delete Review Test Case

<b>Test Case ID</b>	TC#14
<b>Test Case Name</b>	Delete Review
<b>Endpoint</b>	DELETE / reviews/{id}
<b>Input Data</b>	Review ID
<b>Description</b>	Customers delete their review of a specific product.
<b>Expected Result</b>	Review is deleted from the reviews section in product details page.

Table 5.15. View Brand Orders Test Case

<b>Test Case ID</b>	TC#15
<b>Test Case Name</b>	View Brand Orders
<b>Endpoint</b>	GET /orders
<b>Input Data</b>	Order Status
<b>Description</b>	Brands retrieve all orders that include their products filtered by status.
<b>Expected Result</b>	A list of orders related to the brand's products is returned and filtered by order status.

Table 5.16. View Customer Orders Test Case

<b>Test Case ID</b>	TC#16
<b>Test Case Name</b>	View Customer Orders
<b>Endpoint</b>	GET /customers/me/orders
<b>Input Data</b>	Customer Authentication Token
<b>Description</b>	Customers retrieve a list of their orders.
<b>Expected Result</b>	A list of the customers' orders is returned, including relevant details such as order id, status, order items, total price.

Table 5.17. Create Draft Order Test Case

<b>Test Case ID</b>	TC#17
<b>Test Case Name</b>	Create Draft Order
<b>Endpoint</b>	POST /customers/me/orders
<b>Input Data</b>	List of products: {Product ID, Variant ID, Quantity}
<b>Description</b>	Customers create an order draft and reserve the specified quantities of products.
<b>Expected Result</b>	An order draft is created, and the specified quantities of order items are reserved.

Table 5.18. Customer Order Details Test Case

<b>Test Case ID</b>	TC#18
<b>Test Case Name</b>	View Customer Order
<b>Endpoint</b>	GET /customers/me/orders/{id}
<b>Input Data</b>	Order ID
<b>Description</b>	Customers obtain detailed information about a specific order they placed.
<b>Expected Result</b>	The system returns the order details.

Table 5.19. Discard Draft Order Test Case

<b>Test Case ID</b>	TC#19
<b>Test Case Name</b>	Discard Draft Order
<b>Endpoint</b>	DELETE /customers/me/orders/{id}
<b>Input Data</b>	Order ID
<b>Description</b>	Customers discard draft orders on the checkout screen.
<b>Expected Result</b>	The order is removed from the system and the reserved stock quantities for the order items are released.

Table 5.20. Place Order Test Case

<b>Test Case ID</b>	TC#20
<b>Test Case Name</b>	Place Order
<b>Endpoint</b>	PUT /customers/me/orders/{id}/confirm
<b>Input Data</b>	Order ID, Phone Number, Payment Method
<b>Description</b>	Customers confirm and place an order after reviewing its details.
<b>Expected Result</b>	The system begins processing the order and updates the order details.

Table 5.21. Cancel Order Test Case

<b>Test Case ID</b>	TC#21
<b>Test Case Name</b>	Cancel Order
<b>Endpoint</b>	PATCH /customers/me/orders/{id}/cancel
<b>Input Data</b>	Order ID
<b>Description</b>	Customers cancel an order before shipping.
<b>Expected Result</b>	The order status is updated to “cancelled” and the reserved stock quantities for the order items are released.

Table 5.22. Refund Item Test Case

<b>Test Case ID</b>	TC#22
<b>Test Case Name</b>	Refund Order Item
<b>Endpoint</b>	PATCH /customers/me/orders/{id}/refund
<b>Input Data</b>	Order ID, Product ID, Variant ID
<b>Description</b>	The customer requests a refund for a specific item in a delivered order.
<b>Expected Result</b>	The system initiates the refund process, and the order status is updated to “refund requested”.

Table 5.23. Calculate Shipment Fees Test Case

<b>Test Case ID</b>	TC#23
<b>Test Case Name</b>	Calculate Shipment Fees
<b>Endpoint</b>	PUT /customers/me/orders/{id}/calculate-shipment-fees
<b>Input Data</b>	Order ID, Customer Address, Shipment Type
<b>Description</b>	Calculate shipment fees given the customer address and shipment type.
<b>Expected Result</b>	The system returns calculated shipment fees and estimated delivery days and updates the order with the shipment type, the shipment fees, and customer address.

Table 5.24. View Brand Data Test Case

<b>Test Case ID</b>	TC#24
<b>Test Case Name</b>	View Brand Data
<b>Endpoint</b>	GET /brands/me
<b>Input Data</b>	Authorization Token
<b>Description</b>	Retrieve brand data for the authenticated brand owner
<b>Expected Result</b>	The system returns brand profile including brand name, story description, image URL, phone numbers, industry, rating, addresses, websites, and subscription details

Table 5.25. Update Brand Profile Test Case

<b>Test Case ID</b>	TC#25
<b>Test Case Name</b>	Update Brand Profile
<b>Endpoint</b>	PUT /brands/me
<b>Input Data</b>	Brand Name, Story Description, Image URL, Phone Numbers, Industry, Rating, Addresses, Websites
<b>Description</b>	Brand owner updates their brand profile.
<b>Expected Result</b>	The system updates the brand profile information.

Table 5.26. View Brand Owner Profile Test Case

<b>Test Case ID</b>	TC#26
<b>Test Case Name</b>	View Brand Owner Profile
<b>Endpoint</b>	GET /brand-owners/me
<b>Input Data</b>	Authorization Token
<b>Description</b>	Retrieve profile information for the authenticated brand owner
<b>Expected Result</b>	The system returns user data including email, role, first name, last name, username, and image URL.

Table 5.27. Update Brand Profile Test Case

<b>Test Case ID</b>	TC#27
<b>Test Case Name</b>	Update Brand Owner Profile
<b>Endpoint</b>	PUT /brand-owners/me
<b>Input Data</b>	First name, last name, username, and image URL.
<b>Description</b>	Brand owner updates their profile.
<b>Expected Result</b>	The system updates the brand owner profile information.

Table 5.28. View Products List Test Case

<b>Test Case ID</b>	TC#28
<b>Test Case Name</b>	View Products List
<b>Endpoint</b>	GET /brand-owners/me/products
<b>Input Data</b>	Authorization Token, Page Number.
<b>Description</b>	Retrieve paginated list of products for the brand owner
<b>Expected Result</b>	The system returns an array of products including category, department, description, material, name, and review summary

Table 5.29. Add Product Test Case

<b>Test Case ID</b>	TC#29
<b>Test Case Name</b>	Add Product
<b>Endpoint</b>	POST /brand-owners/me/products
<b>Input Data</b>	Category, Department, Description, Material, Name, Variants
<b>Description</b>	Add a new product to the brand's catalog
<b>Expected Result</b>	A new product is created and added to the brand's catalog.

Table 5.30. Edit Product Test Case

<b>Test Case ID</b>	TC#30
<b>Test Case Name</b>	Edit Product
<b>Endpoint</b>	PUT /brand-owners/me/products/{id}
<b>Input Data</b>	Product ID, Category, Department, Description, Material, Name, Variants
<b>Description</b>	Update existing product information
<b>Expected Result</b>	The product information is updated

Table 5.31. View Product Test Case

<b>Test Case ID</b>	TC#31
<b>Test Case Name</b>	View Product
<b>Endpoint</b>	GET /products/{id}
<b>Input Data</b>	Product ID
<b>Description</b>	Retrieve detailed information about a specific product
<b>Expected Result</b>	The system returns product data including category, department, description, material, name, reviewSummary, and variants with stock, images, discount, price, size, and colors.

Table 5.32. Delete Product Test Case

<b>Test Case ID</b>	TC#32
<b>Test Case Name</b>	Delete Product
<b>Endpoint</b>	DELETE /brand-owners/me/products/{id}
<b>Input Data</b>	Product ID
<b>Description</b>	Remove a product from the brand's catalog
<b>Expected Result</b>	The product is deleted from the system

Table 5.33. View Processing Orders Test Case

<b>Test Case ID</b>	TC#33
<b>Test Case Name</b>	View Processing Orders
<b>Endpoint</b>	GET /brand-owners/me/orders/processing-products
<b>Input Data</b>	Authorization Token
<b>Description</b>	Retrieve list of products currently being processed in orders
<b>Expected Result</b>	The system returns products data including Product Name, Total Quantity, and Product URL

Table 5.34. View Refunded Products Test Case

<b>Test Case ID</b>	TC#34
<b>Test Case Name</b>	View Refunded Orders
<b>Endpoint</b>	GET /brand-owners/me/orders/refunded-products
<b>Input Data</b>	Authorization Token
<b>Description</b>	Retrieve list of products currently being refunded
<b>Expected Result</b>	The system returns products data including Product Name, Total Quantity, and Product URL

Table 5.35. View Brand Reviews Summary Test Case

<b>Test Case ID</b>	TC#35
<b>Test Case Name</b>	View Brand Reviews Summary
<b>Endpoint</b>	GET /brand-owners/me/dashboard/reviews-summary
<b>Input Data</b>	Authorization Token
<b>Description</b>	Retrieve overall brand review statistics
<b>Expected Result</b>	The system returns Review Data including Total Reviews, Average Rating, and Rating Distribution.

Table 5.36. View Sales Statistics Test Case

<b>Test Case ID</b>	TC#36
<b>Test Case Name</b>	View Sales Statistics
<b>Endpoint</b>	GET /brand-owners/me/dashboard/months-sales
<b>Input Data</b>	Authorization Token
<b>Description</b>	Retrieve monthly sales statistics and current month performance
<b>Expected Result</b>	The system returns Current Month Stats including Total Products Sold, Total Sales, Top Product, and Monthly Sales data.

## 5.2 Testing Results

Test Case ID	Test Case Name	Actual Result	Defect	Defect Description	Fixed
TC#1	Customer Signup	Customer account created successfully.	No		
TC#2	Customer Third-Party Signup	Google data is parsed, and customer account is created.	Yes	Google data was not correctly parsed, and data was extracted incorrectly.	Yes
TC#3	Customer Login	Customer is logged in and profile data is loaded.	No		
TC#4	Brand Owner Login	Brand owner is logged in to access all features.	No		
TC#5	Brand Manager Login	Brand manager is logged in with access to specific features.	No		
TC#6	Forget/Change Password	Email is sent to the customer and enters the new password to be updated.	No		
TC#7	AI Try-On	The result with the selected product being viewed on the customer's provided image.	Yes	Poor lighting or invalid photos may produce unexpected results.	No
TC#8	Add to wishlist	Product is added to the customer's wishlist for future purchase decisions.	No		
TC#9	Remove from wishlist	Product is removed from the customer's wishlist.	No		
TC#10	Add to Cart	The product is added to the logged in customer's Cart.	No		
TC#11	Remove from cart	The product is removed from the logged in customer's cart	No		
TC#12	Create new review	The review is added to the reviews section in the product details page.	Yes	Validation is needed to allow customer to add one review per product.	Yes
TC#13	Update Review	Customers update their review on a product.	No		
TC14	Delete Review	Review is deleted from the reviews section in product details page.	No		
TC#15	View Brand Orders	A list of orders related to the brand's products is returned.	No		
TC#16	View Customer Orders	A list of customers' orders is returned.	No		

## Chapter 5 – ELEVATE Testing and Evaluation

<b>Test Case ID</b>	<b>Test Case Name</b>	<b>Actual Result</b>	<b>Defect</b>	<b>Defect Description</b>	<b>Fixed</b>
TC#17	Create Draft Order	An order draft is created, and the specified quantities of order items are reserved.	No		
TC#18	View Customer Order	The system returns customer order details	No		
TC#19	Discard Draft Order	The order is removed from the system and the reserved stock quantities for the order items are released.	No		
TC#20	Place Order	The system begins processing the order and updates the order details.	No		
TC#21	Cancel Order	The order status is updated to “Cancelled” and the reserved stock quantities for the order items are released.	No		
TC#22	Refund Order	The system initiates the refund process, and the order status is updated to “Refund Requested”.	Yes	The customer can't refund more than one product in the same order.	Yes
TC#23	Calculate Shipment Fees	The system returns calculated shipment fees and updates the order with the shipment type, shipment fees, and customer address.	No		
TC#24	View Brand Data	The system returns brand profile including all brand details	No		
TC#25	Update Brand Profile	The system updates the brand profile information.	No		
TC#26	View Brand Owner Profile	The system returns user data including email, role, first name, last name, username, and image URL.	No		
TC#27	Update Brand Owner Profile	The system updates the brand owner profile information.	No		
TC#28	View Products List	The system returns an array of products including category, department, description, material, name, and review summary	No		
TC#29	Add Product	A new product is created	No		

## Chapter 5 – ELEVATE Testing and Evaluation

<b>Test Case ID</b>	<b>Test Case Name</b>	<b>Actual Result</b>	<b>Defect</b>	<b>Defect Description</b>	<b>Fixed</b>
TC#30	Edit Product	The product is updated	Yes	After updating a product, it would give a 5-minute timeout on update/delete.	Yes
TC#31	View Product	The system returns product data including category, department, description, material, name, reviewSummary, and variants with stock, images, discount, price, size, and colors.	No		
TC#32	Delete Product	The product is deleted from the system	Yes	After deleting a product, it would give a 5-minute timeout on update/delete.	Yes
TC#33	View Processing Orders	The system returns products data including Product Name, Total Quantity, and Product URL	No		
TC#34	View Refunded Orders	The system returns products data including Product Name, Total Quantity, and Product URL	No		
TC#35	View Brand Reviews Summary	The system returns Review Data including Total Reviews, Average Rating, and Rating Distribution.	No		
TC#36	View Sales Statistics	The system returns Current Month Stats including Total Products Sold, Total Sales, Top Product, and Monthly Sales data.	No		

## Chapter 6 – Conclusions and Future Work

### 6.1 Conclusions

Throughout the development of the **ELEVATE** platform, the team successfully achieved several key milestones and here are some of the conclusions:

- **Centralized Platform:** Developed a fully functional e-commerce platform that brings together a wide range of Egyptian local clothing brands in one accessible digital space.
- **Augmented Reality Integration:** Implemented AI-based try-on features that allow users to visualize clothing on themselves, significantly enhancing the online shopping experience.
- **AI-Powered Personalization:** Integrated Algolia-based recommendations that provide customers with smart, personalized product suggestions.
- **Advanced Search Capabilities:** Enabled visual search by image and advanced filtering options, making product discovery more intuitive and efficient.
- **Brand Owner Dashboard:** Provided brand owners with a dedicated dashboard to manage their products, monitor sales, and analyze customer feedback.
- **Robust and Scalable Architecture:** Built the platform using scalable technologies like Node.js, Firebase, and Algolia to handle growth in users and products efficiently.
- **Mobile-Friendly Application:** Created a cross-platform mobile app using Flutter to ensure seamless access for users on both Android and iOS devices.
- **Comprehensive Testing:** Conducted extensive system and user testing to ensure functionality, security, and smooth user experience.

## 6.2 Lessons Learned

Through the journey of this project, the team gained valuable lessons in both technical and project management aspects:

- **Understanding Local Market Needs:** Building a solution tailored to the unique needs of Egyptian local brands and shoppers required a deep understanding of the local e-commerce landscape.
- **The Importance of User-Centric Design:** Prioritizing ease of use, intuitive navigation, and real-time feedback significantly impacts user satisfaction and platform adoption.
- **Agile Flexibility is Crucial:** Adopting an iterative approach through the Star Life Cycle Model allowed the team to continuously adapt to challenges and refine the system based on user feedback.
- **Integrating New Technologies Requires Deep Research:** Working with AI and visual search highlighted the importance of thorough research and testing to ensure performance and accuracy.
- **Team Collaboration and Task Distribution:** Effective coordination and clear role assignments were essential in managing the complex and parallel development tracks of the system.
- **Testing Early and Often Matters:** Early testing helped identify issues before they escalated, leading to smoother final integrations and a more reliable platform.

### 6.3 Future Work

To further enhance the **ELEVATE** platform and expand its capabilities, several key areas have been identified for future development. These initiatives aim to improve user experience, streamline operations, broaden market reach, and integrate more advanced technologies.

- **Global Audience Expansion:** Extend the platform's reach beyond Egypt, enabling international customers to seamlessly order products from local brands, thereby unlocking new markets and increasing brand visibility on a global scale.
- **Push Notifications Support:** Integrate Firebase Cloud Messaging to enable real-time push notifications for users, enhancing engagement by providing instant updates on order status, promotions, and personalized recommendations.
- **Comment Sentiment Analysis:** Utilize natural language processing (NLP) to perform sentiment analysis on customer reviews, classifying them into categories such as "Admiration," "Approval," or "Sadness" to provide brands with deeper insights into customer feedback.
- **Real Payment Gateway Integration:** Integrate with prominent local payment gateways such as Fawry and PayMob to offer secure, diverse, and convenient payment options for customers, enhancing the checkout experience.
- **Real Shipment Company Integration:** Establish direct integrations with local shipment companies like Bosta to automate logistics, provide real-time tracking, and ensure efficient and reliable product delivery.
- **Granular Access Controls for Brand Managers:** Implement advanced access control mechanisms, allowing brand owners to define specific permissions and roles for individual brand managers, ensuring secure and tailored access to brand data and functionalities.
- **Dedicated Brand Manager Dashboard:** Develop a separate, tailored dashboard for brand managers, providing them with specific tools and insights relevant to their responsibilities, distinct from the brand owner's comprehensive dashboard.
- **AI Recommendations for Brand Owners:** Leverage Large Language Models (LLMs) and advanced AI to provide brand owners with data-driven recommendations, analyzing sales trends and customer histories to suggest strategies for improving product offerings and boosting sales.

- **Message Broker Integration for Database Updates:** Integrate message brokers to manage asynchronous database updates, allowing for periodic batch uploads rather than per-request updates. This optimizes resource utilization and reduces the load on the database, improving overall system performance.
- **Customer App Chatbot with LLM and RAG:** Integrate a chatbot into the customer mobile application for instant support. This chatbot will utilize LLMs combined with Retrieval-Augmented Generation (RAG) to provide accurate answers to customer questions based on the project's data.
- **Exploration of Advanced AI Models:** Continuously research and integrate new AI models to further enhance the customer experience, exploring innovative applications that can improve personalization, visual search, and virtual try-on features.
- **Mobile App Personalization and History Features:** Implement user-centric enhancements in the mobile app, including Light/Dark mode, multi-language support (Arabic and English), and dedicated screens for try-on histories and other user activities.
- **Snapchat AR Lens Integration:** Explore integration with Snapchat AR Lenses to expand the augmented reality try-on experience, potentially reaching a wider audience and offering more interactive virtual try-on options.
- **Dark Mode Theme:** Enhance the look and feel of the application through a dark mode theme to fulfill different customer needs and enable theme configurations for further customization.

## References

Listed below links to the set of tools we used throughout the project along with other resources that helped us complete the implementation:

- [1] Node.js, “Node.js JavaScript runtime,” [Online]. Available: <https://nodejs.org/en>. [Accessed: Feb. 2025].
- [2] Express.js, “Fast, unopinionated, minimalist web framework for Node.js,” [Online]. Available: <https://expressjs.com/>. [Accessed: Feb. 2025].
- [3] Swagger Editor, “Design, describe, and document your API,” [Online]. Available: <https://editor.swagger.io/>. [Accessed: May. 2025].
- [4] Firebase, “Firebase app development platform by Google,” [Online]. Available: <https://firebase.google.com/>. [Accessed: Jul. 2025].
- [5] Algolia, “AI-powered search and discovery platform,” [Online]. Available: <https://www.algolia.com/>. [Accessed: Jul. 2025].
- [6] Upstash, “Serverless database for Redis and Kafka,” [Online]. Available: <https://upstash.com/>. [Accessed: Jul. 2025].
- [7] Google Cloud, “Vertex AI platform,” [Online]. Available: <https://console.cloud.google.com/vertex-ai>. [Accessed: Jul. 2025].
- [8] Google Cloud, “Vertex AI documentation,” [Online]. Available: <https://cloud.google.com/vertex-ai/docs>. [Accessed: Jul. 2025].
- [9] Google Cloud, “Official cloud services and tools,” [Online]. Available: <https://cloud.google.com/?hl=en>. [Accessed: Jul. 2025].
- [10] Replicate, “Run machine learning models in the cloud,” [Online]. Available: <https://replicate.com/>. [Accessed: Jul. 2025].
- [11] Fal.ai, “AI inferencing platform for developers,” [Online]. Available: <https://fal.ai/>. [Accessed: Jul. 2025].
- [12] Vercel, “Frontend cloud platform for developers,” [Online]. Available: <https://vercel.com/>. [Accessed: Jun. 2025].
- [13] Flutter, “Build apps for mobile, web & desktop from a single codebase,” [Online]. Available: <https://flutter.dev/>. [Accessed: Jan. 2025].
- [14] ReactJS, “A JavaScript library for building user interfaces,” [Online]. Available: <https://react.dev/>. [Accessed: Feb. 2025].
- [15] Material UI, “React UI library with Material Design,” [Online]. Available: <https://mui.com/material-ui/>. [Accessed: Mar. 2025].

- [16] Visual Studio Code, “Code editor redefined and optimized for building and debugging modern web and cloud applications,” [Online]. Available: <https://code.visualstudio.com/>. [Accessed: Jul. 2025].
- [17] GitHub Copilot, “AI pair programmer by GitHub and OpenAI,” [Online]. Available: <https://github.com/features/copilot>. [Accessed: Jul. 2025].
- [18] Discord, “Voice, video, and text communication service,” [Online]. Available: <https://discord.com/>. [Accessed: Jul. 2025].
- [19] Lucidchart, “Visual workspace for diagramming,” [Online]. Available: <https://www.lucidchart.com/pages>. [Accessed: Jul. 2025].
- [20] Microsoft Word Online, “Word processing software,” [Online]. Available: <https://word.cloud.microsoft/>. [Accessed: Jul. 2025].
- [21] Stack Overflow, “Question-and-answer site for programming,” [Online]. Available: <https://stackoverflow.com/>. [Accessed: Apr. 2025].
- [22] Hugging Face, “Platform for sharing and running AI models,” [Online]. Available: <https://huggingface.co/>. [Accessed: Dec. 2024].
- [23] Shopify, “E-commerce platform for online stores,” [Online]. Available: <https://www.shopify.com/>. [Accessed: Mar. 2025].
- [24] Domanza, “Creative agency and branding platform,” [Online]. Available: <https://domanza.co/>. [Accessed: Mar. 2025].
- [25] Noon, “E-commerce platform in the Middle East,” [Online]. Available: <https://www.noon.com/egypt-en/>. [Accessed: Mar. 2025].